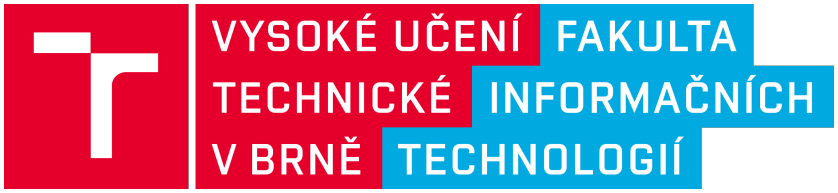

Dokumentácia k projektu z predmetu SUR

Model pro identifikaci 31 různých osob z obrázku obličeje a hlasové nahrávky

2024/2025

Patrik Gáfrik (xgafri00)
Adrián Horváth (xhorva14)

1 Úvod

Cieľom tejto práce je vytvoriť systém na identifikáciu osôb na základe multimodálnych vstupov – konkrétne z obrázkov tvárí (vo formáte PNG) a hlasových nahrávok (vo formáte WAV). Úloha spočíva v klasifikácii 31 rôznych osôb na základe dodaných trénovacích dát, pričom pre každú osobu sú k dispozícii viaceré vzorky z rôznych nahrávacích sedení. Vstupom do systému sú neznáme obrázky a zvukové súbory, ktoré majú byť automaticky priradené k jednej z 31 tried. Výstupom je textový súbor obsahujúci pre každý segment identifikátor súboru, predikovanú triedu a skóre pre všetky triedy vo forme logaritmických pravdepodobností.

2 Použité dáta

Na trénovanie a vyhodnocovanie rozpoznávacích systémov boli poskytnuté dáta v archíve **SUR_projekt2024-2025.zip**, ktorý obsahuje dva hlavné adresáre: **train** a **dev**. Každý z nich je ďalej rozdelený do 31 podadresárov, pričom každý podadresár zodpovedá jednej osobe (triede) identifikovanej číslom od 1 do 31. V rámci každej triedy sa nachádzajú vzorky vo formáte PNG (obrázky tváre) a WAV (hlasové nahrávky). Názvy súborov obsahujú štruktúrované informácie rozdelené podčiarkovníkmi, napríklad: **f401_01_f12_i0_0**, kde **f401** je identifikátor osoby, **01** označuje číslo nahrávacieho sedenia. V súlade s pravidlami zadania projektu sme nepoužili žiadne externé obrazové alebo zvukové dáta, ani predtrénované modely. Všetky trénovacie a testovacie dáta pochádzajú výhradne z poskytnutého archívu.

3 Spracovanie obrázkov

Každú snímku z trénovacej sady sme previedli na škálu šedej, normalizovali a zmenili na jednotnú veľkosť, z ktorej sme extrahovali HOG príznaky. Pomocou mriežkového vyhľadávania sme optimalizovali rozlíšenie, veľkosť buniek, blokov a počet orientačných binov podľa presnosti na validačnej sade. S najlepším nastavením sme príznaky štandardizovali, natrénovali lineárny SVM a overili jeho výkonnosť na dev sade.

3.1 Extrakcia HOG príznakov

Každý obrázok sa načíta cez **skimage.io.imread()**. Ak je farebný, transformuje sa na odtiene šedej (rgb2gray). Následne sa obrazu priradí float repre-

zentácia (img_as_float) a preškáluje sa na pevné rozmery (štvorec) pomocou **skimage.transform.resize()**. Parametre HOG:

- **resize_dim** – rozmery výsledného obrázka (výška, šírka); default (48, 48)
- **orientations** – počet orientačných binov histogramu gradientov; default 12
- **pixels_per_cell** – veľkosť bunky v pixeloch; default (4, 4)
- **cells_per_block** – veľkosť normalizačného bloku v bunkách; default (2, 2)

Výsledkom extrakcie je vektor dĺžky závislej od vyššie uvedených parametrov (niekoľko tisíc prvkov). Rôzne kombinácie týchto parametrov výrazne ovplyvňujú veľkosť vektoru. Optimalizácia HOG príznakov prebiehala pomocou mriežkového vyhľadávania štyroch kľúčových parametrov – rozlíšenia obrázku, veľkosti priestorovej bunky, veľkosti normalizačného bloku a počtu orientačných binov. Pre každú kombináciu sme jednorazovo extrahovali príznaky z trénovacej aj validačnej množiny, natrénovali lineárny klasifikátor s pevnou reguláciou a zmerali presnosť na validačných dátach. Na základe týchto výsledkov sme vybrali najúspešnejšie nastavenie, ktoré sme následne skontrolovali retréningom na celej trénovacej sade a použili vo finálnom modeli.

3.2 Tréning SVM modelu

Pred trénovaním klasifikátora sa každá dimenzia HOG-vstupného vektora upraví tak, aby mala nulovú strednú hodnotu a jednotkovú varianciu – táto normalizácia zabezpečuje, že jednotlivo extrahované gradientovo-orientované príznaky prispievajú k rozhodnutiu modelu v rovnakej miere, bez dominancie tých s väčším numerickým rozsahom. Upravené vektory slúžia ako vstup do lineárneho klasifikátoru SVM (funkcia **LinearSVC(C=...)**, ktorý sa snaží nájsť hyperrovinu maximalizujúcu šírku okraja medzi vzorkami rôznych tried. Regularizačný parameter C v optimalizačnej úlohe balancuje medzi šírkou okraja a penalizáciou chýb na trénovacích príkladoch, čím zabraňuje nadmernému prispôsobeniu sa šumu či odľahlým vzorkám.

Optimalizačný algoritmus rieši convexnú úlohu s hrotovým (hinge) kritériom spoil-loss a L2-normou regularizácie, pričom iteratívne hľadá riešenie až do dosiahnutia konvergenčného kritéria (maximálny počet iterácií stanovený na 5000 krokov). Po ukončení procesu trénovania model obsahuje parametre

hyperroviny (váhy a posun) pre každú triedu a je pripravený na predikciu. Nakoniec sa vytrénované parametre uložia vo forme binárneho súboru, ktorý môže byť v následných krokoch načítaný a aplikovaný na nové vstupné dáta bez potreby opätovného trénovania.

4 Spracovanie nahrávok

V zvukovej časti súbory načítame pri 16kHz a rozdelíme na prekryté rámce dĺžky 25ms, z ktorých extrahujeme mel-cepstrálne koeficienty (13 alebo 16), voliteľne rozšírené o delta-koeficienty a upravené cepstrálnou normalizáciou. Optimalizáciou kľúčových parametrov MFCC (počet koeficientov, filtrov, FFT, delty, CMVN) a počtu Gaussových zložiek (16, 32, 64) vyberieme nastavenie s najvyššou presnosťou na validačnej sade. S týmto nastavením natrénujeme pre každú osobu samostatný GMM pomocou EM algoritmu (max. 20 iterácií, tolerancia 1e-4). Modely sa potom vyhodnotia na trénovacej a validačnej množine a nasadia pre predikciu neoznačených nahrávok, pričom pre každý segment generujú rozhodnutie aj log-pravdepodobnostné skóre všetkých 31 tried.

4.1 Extrakcia MFCC príznakov

Pre každú nahrávku sa najprv segmentuje signál do prekrytých rámcov dĺžky 25ms s posunom 10ms. Na každý rámec sa aplikuje krátkodobá Fourierova transformácia s dĺžkou FFT 512 (príp. 1024 pri optimalizácii). Výsledné spektrá sa projekčne prevádzajú na melovú stupnicu s 23 alebo 30 filtrovými bankami, z ktorej sa počíta prvých 13 alebo 16 cepstrálnych koeficientov. V prípade potreby sa do reprezentácie pridajú aj prvé a druhé delta-koeficienty, ktoré zachytávajú dynamiku cepstrálnych príznakov. Nakoniec sa na každý koeficient aplikuje normalizácia jeho strednej hodnoty a rozptylu v rámci celej nahrávky, čím sa kompenzujú rozdiely v celkovej energetickej hladine a mikrofónnych vlastnostiach.

Pre nájdenie optimálnej kombinácie základných MFCC parametrov (počet koeficientov, počet mel-filter, dĺžka FFT, použitie delta-príznakov, CMVN) a počtu zložiek GMM (16, 32, 64) sme realizovali mriežkové vyhľadávanie. Pre každé nastavenie sme extrahovali príznaky z celej trénovacej a validačnej množiny, natrénovali samostatné GMM pre každú osobu a na validačnej sade vyhodnotili presnosť priemerného log-pravdepodobnostného skóre segmentu. Kombinácia s najvyššou presnosťou bola zvolená pre finálny tréning.

4.2 Tréning GMM modelu

Pre každú osobu sa všetky rámcové príznaky z nahrávok tejto triedy zreťazia do jednej matice. GMM sa inicializuje rovnomernými váhami, prostriedkami náhodne vybranými z dát a diagonálnymi kovarianciami odhadnutými z celkovej variability. Algoritmus očakávacej-maximalizačnej (EM) iterácie potom aktualizuje parametre až do splnenia tolerančného kritéria (prírastok log-pravdepodobnosti $< 1e$-4) alebo prekročenia maximálneho počtu krokov (20). Použitie diagonálnej kovariančnej matice znižuje riziko numerických problémov a urýchľuje výpočty.

Vo finálnej fáze sa pre nové, neoznačené nahrávky aplikuje ten istý postup extrakcie aj GMM scoringu. Výstupný súbor obsahuje pre každý segment názov, predikovanú triedu a log-pravdepodobnosti voči všetkým 31 triedam v poradí.

5 Spustenie systému

5.1 Použité knižnice a prostredie

```
python -m venv venv
source venv/bin/activate
pip install numpy scipy scikit-image scikit-learn librosa tabulate
```

Knižnica `ikrlib` musí byť prístupná v `PYTHONPATH`.

5.2 Spustenie obrazového modulu

Tréning a vyhodnotenie

```
python3 image_classifier.py \
  --train \
  --resize 48 48 \
  --ppc 4 4 \
  --cpb 2 2 \
  --orientations 12 \
  --svm-c 0.1 \
  --train-dir path/to/train \
  --dev-dir   path/to/dev \
  --output-model image_model.pkl
```

- načíta `train` a `dev` obrázky,

- extrahuje HOG príznaky,
- natrénuje SVM a uloží pipeline do `image_model.pkl`,
- vyhodnotí model na validačnej sade (classification report).

Predikcia

```
python3 image_classifier.py \
  --predict \
  --input-model       image_model.pkl \
  --input-data        path/to/unlabeled_images \
  --output-predictions image_predictions.txt
```

5.3 Spustenie audio modulu

Tréning a vyhodnotenie

```
python3 audio_classifier.py \
  --train \
  --nceps 16 \
  --nbanks 23 \
  --nfft   512 \
  --add-deltas \
  --use-cmvn \
  --gmm-components 32 \
  --max-iter        20 \
  --tol             1e-4 \
  --train-dir       path/to/train \
  --dev-dir         path/to/dev \
  --output-model    audio_models.pkl
```

- načíta WAV nahrávky pri 16 kHz,
- extrahuje MFCC príznaky
- trénuje GMM pre každú triedu,
- ukladá slovník modelov do `audio_models.pkl`,
- vyhodnocuje na tréningovej aj validačnej sade.

Predikcia

```
python3 audio_classifier.py \
  --predict \
  --input-model       audio_models.pkl \
  --input-data        path/to/unlabeled_wavs \
  --output-predictions audio_predictions.txt
```

5.4 Formát výstupných súborov

Výstupné textové súbory s výsledkami predikcií majú formát:

1. **Segment name** (názov súboru bez prípony),
2. **Hard label** (trieda 1–31),
3. **31 log-pravdepodobností** pre triedy 1 – 31, oddelených medzerou, alebo NaN v prípade klasifikátora obrázkov.

Príklad:

```
eval_00444 29 -65.403634 -64.368401 ... -62.528089
eval_00420 16 nan nan nan nan ... nan
```

6 Vyhodnotenie a záver

Vyhodnotenie na validačnej množine ukázalo, že z obrazovej modality naše riešenie dosahuje presnosť približne 75,8% (47/62). Hoci niektoré triedy dosahujú perfektnú priepustnosť a precíznosť (napr. triedy 2, 3, 9, 10, 11, 13, 14, 16, 25, 28), u viacerých tried (1, 7, 20, 27) systém stále nerozpoznal ani jeden správny prípad. Macro-priemer (precision = 0,72, recall = 0,76, F1 = 0,71) naznačuje, že výkon sa líši medzi triedami, a niektoré tváre sú pre HOG – SVM extrémne náročné na odlíšenie.

Naopak, audio systém pracuje výrazne lepšie—celková presnosť je 87,1% (54/62) a macro-priemer (precision = 0,91, recall = 0,87, F1 = 0,87) potvrdzuje vyváženejší výkon naprieč všetkými 31 triedami. Takmer dve tretiny tried (17 zo 31) dosahujú dokonalú citlivosť aj presnosť, a i tie, pri ktorých sa systém menej trafí (napr. trieda 1: precision = 1,00, recall = 0,50; trieda 28: precision = 0,50, recall = 1,00), stále prinášajú adekvátne F1-skóre.

Hlavnou silnou stránkou audio modelu je možnosť kvantifikovať stupeň istoty—priemerné log-pravdepodobnostné skóre poskytuje nielen najpravdepodobnejší odhad identity, ale aj dôveru v rozhodnutie, čo je kľúčové pri

budúcej multimodálnej fúzii. Obrazový model túto funkciu postráda, pretože SVM nie je pravdepodobnostný model.

Pre ďalší rozvoj obrazovej časti systému by bolo vhodné nasadiť klasifikátor, ktorý dokáže priamo odhadovať pravdepodobnosti výstupných tried – napríklad SVM s rbf kernelom a povolenou kalibráciou (probability=True) alebo alternatívne modely s natívnou podporou pravdepodobnostných predikcií.

V audio časti by sa oplatilo jemne vyladiť počet Gaussových zložiek v zmesiach a rozmer MFCC reprezentácie okolo už overených hodnôt – napríklad vykonať menší grid-search pre 24, 32 a 40 komponentov či rozšíriť množinu delta-koeficientov.