Dokumentace k projektu SUR

Autoři: Rudolf Jurišica (xjuris02), Rostislav Červenka (xcerve30)

Projekt byl implementován v jazyce Python 3.12.

Společné vyhodnocení výsledků je implementováno v souboru `combined_evaluation.py`. Veškeré zdrojové soubory se nacházejí ve složce ‚src/'.

Potřebné knihovny k doinstalování jsou vypsány v souboru `requirements.txt`.

Pro správné spuštění se očekává, že se složky trénovací train a dev vyskytují ve stejném adresáři.

Spuštění kombinovaného vyhodnocení lze provést příkazem:

```
python3 src/combined_evaluation.py [test_dir] [output_file]
[is_final]
```

`test_dir` značí složku se soubory, které se budou rozpoznávat.
`results_file` značí soubor, do kterého se budou zapisovat výsledky. Jejich formát (formát řádků) je následující:
- jméno segmentu
- tvrdé rozhodnutí o třídě (hodnota 1 – 31)
- 31 polí obsahující číselné skóre odpovídající logaritmickým pravděpodobnostem jednotlivých tříd 1 až 31
`is_final` je přepínač (zadávejte pouze hodnoty „true“ nebo „false“), kterým se nastavuje, jestli jde o konečné vyhodnocování. Nastavuje to způsob čtení souborů pro rozpoznávání. Trénovací data jsou obsaženy ve složce a jednotlivé třídy v dalších složkách, zatímco finální soubory (složka eval) pro rozpoznávání pouze v jedné složce.

Rozpoznávání obrazu (Rudolf Jurišica)

Implementace se nachází v souboru `train_image.py`.
Spuštění daného skriptu lze pomocí:
`python3 src/train_image.py [test_dir] [results_file] [is_final]`

Pro vytvoření modelu pro rozpoznávání obrazu byla primárně využita knihovna scikit. Vzhledem k malému množství trénovacích dat je s daty pracováno v čenobílém formátu. Byla zkoušena augmentace dat (jejich pootočení, oříznutí apod.). Pro každý obrázek se vytvořilo 10 nových augmentovaných obrázků. Po několika otestování ale vycházely výsledky ještě hůře, takže se následně s augmentací již nepracovalo a byla smazána. Stejně tak se testovalo učení na barevných obrázcích, což také produkovalo daleko horší výsledky.

Model se trénoval na datech ze složky dev. Ověření jeho přesnosti se provedlo na souborech .png ze složky dev. Konečná přesnost byla kolem 75%.

Trénování modelu probíhá následovně. Extrahují se třídy z názvů souborů (např. ‚f404‘), obrázek se načte ve stupních šedi, převzorkuje na velikost 80x80 (pokud tak již není), a spočítají se HOG příznaky.
Názvy tříd se zakódují pomocí `LabelEncoder` na číselné hodnoty a vytvoří se trénovací pipeline. Skládá se z LDA a Support Vector Classifier (SVC).
Stejný postup je i pro konečné vyhodnocení na datech ze složky eval. Tentokrát se ale model trénuje již na obrázcích z dev i train adresářů, pro lepší přesnost. Tato verze je i odevzdána. V případě trénování jen na datech z ‚train‘ a ověření přesnosti na datech z ‚dev‘ je potřeba zakomentovat a odkomentovat příslušné vyznačené části v souboru train_image.py ve funkci `evaluate`.

Vylepšení daného modelu by se mohlo provést zvýšením počtu komponent v LDA (to, které je tam nyní nastavené, vycházelo z cca. 6 možností nejlépe). Případně umožnit ukládání modelu a jeho pozdější načtení (v tomto zadání nebylo potřeba, alespoň se vždy natrénuje na nejaktuálnějších datech).

Rozpoznávání audia (Rostislav Červenka)

Implementace se nachází v souboru train_voice.py.
Spuštění daného skriptu lze pomocí:
python3 src/train_voice.py [test_dir] [results_file] `[is_final]`

Pro vytvoření modelu pro rozpoznávání mluvčích byla využita knihovna scikit-learn, konkrétně GaussianMixture modely. Vstupními daty jsou .wav soubory s řečí jednotlivých osob.

Každý záznam je předzpracován – je převeden na mono a resamplován na 16 kHz- inspirováno funkcí wav16khz2mfcc. Pomocí librosa.effects.trim() jsou odstraněny tiché části nepřesahující 20db.

Zvukové soubory jsou nejprve převedeny na 13 MFCC příznaků, ke kterým jsou dále dopočítány delta a delta-delta koeficienty, čímž vzniká kombinovaný příznakový vektor délky 39.

Modely se trénují zvlášť pro každou osobu. K trénování se využívají záznamy ze složek train a dev. Trénovací příznaky z obou těchto složek se sloučí, čímž se zvyšuje množství dat pro každého mluvčího. Pro každý z 31 mluvčích je vytvořen samostatný GMM model o 32 komponentách. Tento počet komponent vykazoval nejlepší výsledky během testování.

Vyhodnocení se provádí pomocí funkce evaluate_all, která načte všechny modely, zpracuje .wav soubory z předložené složky a předpoví identitu mluvčího. Výsledky jsou uloženy do výstupního souboru v podobě řádků obsahujících název souboru, predikovaný štítek a skóre pro každý model.

Při trénování na složce train a testování na složce dev dosahovala tato implementace 80-85% úspěšnost. Pro evaluaci na složce eval, se se jako trénovací data použily obě složky train a dev.

Kombinování evaluací

Implementace se nachází v souboru combined_evaluation.py.
Spuštění je popsáno na začátku dokumentu.

Tento skript slouží k vyhodnocení kombinovaného systému pro rozpoznávání osob na základě hlasu a obrazu. Při jeho spuštění se nejprve natrénují potřebné modely pro jednotlivé modality a poté se na zadaných testovacích datech provedou predikce.

Pro každý testovací vzorek se načtou skóre obou systémů. Aby bylo možné je porovnat a zkombinovat, jsou skóre nejprve normalizována do intervalu 0–1. Následně jsou převedena na logaritmické pravděpodobnosti (log-likelihoody), čímž se zvýrazní rozdíly mezi jednotlivými třídami a předejde se zkreslení extrémními hodnotami. Obě logaritmická skóre se poté jednoduše sečtou. Výsledná predikce je určena jako třída s nejvyšším kombinovaným skóre.

Během testování bylo zjištěno, že bez použití normalizace výsledky z rozpoznávání audia přebíjely výsledky z obrázků, kvůli jejich větší velikosti. Díky normalizaci se úspěčnost správné identifikace zvětšila až o pět procent na 95%.

Kombinované výsledky se uloží do výstupního souboru, který obsahuje pro každý vzorek jeho název, predikovanou třídu a kombinovaná skóre pro všechny osoby. Tento přístup umožňuje efektivně kombinovat dva nezávislé systémy do jednoho robustnějšího řešení. Výsledná přesnost je typicky vyšší než u jednotlivých systémů samostatně, zejména v případech, kdy jeden z nich selhává, ale druhý podává kvalitní výsledky.