SUR – Dokumentace k projektu

Dalibor Kříčka (xkrick01)

květen, 2025

1 Rozpoznávání obličeje

1.1 Řešení pomocí Convolutional Neural Network (CNN)

Struktura konvoluční neuronové sítě

Při konstrukci CNN bylo experimentováno především s počtem konvolučních vrstev, počtem filtrů v jednotlivých vrstvách, velikostí konvolučních jader, použitím dropout vrstev a velikostí plně propojených vrstev. Nejoptimálnější řešení bylo nalezeno pomocí nástroje *Keras Tuner*, což je nástroj pro automatizované hledání nejvhodnějších hyperparametrů neuronových sítí. Testována byla také velikost jedné dávky (batch) nebo různé optimalizátory (Adam, RMSprop a Adagrad). Samozřejmě nebyly vyzkoušeny všechny možné kombinace zmíněných parametrů z důvodu velkého množství možností, ale systematicky byla nastavení měněna a pozorována zlepšení modelu. Výsledný systém viz soubor `SRC/cnn_img.py`.

Augmentace

Trénování sítě nad náhodně pozměněnými daty vedlo ke zlepšení při vyhodnocení přesnosti modelu na testovacích datech. Dá se předpokládat, že trénování sítě nad pozměněnými vstupy, může pomoci také přesnosti při klasifikaci obličejů ve finální fázi projektu. Změny však byly aplikovány pouze v rozumné míře z důvodu malého množství vstupních dat.

Konkrétně byly použity následující úpravy obrázků: posunutí vlevo/vpravo/nahoru/dolů, přiblížení, rotace a horizontální převrácení.

1.2 Řešení pomocí Support Vector Machine (SVM)

Model používá lineární jádro (kernel='linear'), které hledá rozhodovací hranici ve formě přímky (hyperroviny) v prostoru příznaků. Parametr $C = 0.1$ zajišťuje vyšší regularizaci, díky čemuž model více toleruje chyby a lépe generalizuje. Tyto parametry byly zvoleny na základě výsledků získaných při vyhodnocení přesnosti SVM klasifikátoru pomocí nástroje GridSearchCV, který slouží pro automatické ladění hyperparametrů modelu.

Volba `probability=True` umožňuje, aby model vracel pravděpodobnosti příslušnosti ke třídám a ne pouze konečnou predikci. To ovšem není typická vlastnost pro SVM. I když samotná dokumentace uvádí, že výsledky pravděpodobností mohou být nepřesné při malých trénovacích datových sadách, v tomto případě poměrně přesné byly. Pravděpodobnosti jsou vypsány v souboru s výsledky a dále jsou použity pro kombinaci s klasifikátorem hlasu. Výsledný systém viz soubor `SRC/svm_img.py`.

Extrakce příznaků

Ve výsledném řešení se pro extrakci příznaků z obrazových dat používá technika HOG (Histogram of Oriented Gradients). HOG je metoda pro extrakci tvarových rysů z obrazu, která rozdělí obraz do malých buněk (např. 4x4 pixely). Následně pro každou buňku spočítá histogram gradientů (směrů a velikostí změn jasu). Histogramy sousedních buněk jsou následně normalizovány do tzv.

bloků (např. 2x2 buňky) a ty se spojí do jednoho dlouhého vektoru příznaků. Experimentováno bylo s velikostmi buněk a bloků.

Při vývoji modelu byly zkoušeny i další techniky jako je PCA (Principal Component Analysis) a LBP (Local Binary Patterns). Avšak tyto metody nevedly ke zlepšení přesnosti SVM klasifikátoru, a proto v závěru nebyly použity.

Augmentace

I zde vedlo přidání náhodně pozměněných dat ke zlepšení přesnosti modelu na testovacích datech. Konkrétně je přidáno do trénovací sady 10 náhodně mírně pozměněných obrázků pro každý původní obrázek. Experimentováno bylo s různými počty augmentací na obrázek.

Pro doplnění, použity byly následující úpravy obrázků: výřez původního obrázku, rotace, horizontální převrácení, zvýšení/snížení jasu, kontrastu a sytosti.

2 Rozpoznávání hlasu

2.1 Řešení pomocí Gaussian Mixture Models (GMM)

Pro trénování GMM bylo využito EM (Expectaion Maximization) algoritmu s diagonálními kovariančními maticemi (kvůli rychlosti a lepší generalizaci pro malé množství trénovacích dat). Počet GMM komponent byl stanoven na 16 pro poskytnutí nejvyšší získané přesnosti na testovacích datech. Výsledný systém viz soubor `SRC/gmm_voice.py`.

Extrakce příznaků

Pro extrakci příznaků jsou využity Mel-frequency cepstral coefficients (MFCC), které se používají pro převod audio signálu z časové oblasti na kompaktní reprezentaci hlasového spektra. Na výsledné koeficienty je aplikována normalizace odečtením střední hodnoty všech koeficientů. Tím je možné redukovat rozdíly v nahrávkách vzniklé použitím jiného nahrávacího zařízení či ekvalizéru. Počet zpracovávaných koeficientů byl stanoven na 12.

Předzpracování a augmentace

Každá nahrávka začíná stejným zvukovým signálem, který je odstraněn (počáteční 1 sekunda). Dále jsou z každé nahrávky vyříznuty tiché úseky (začátek, konec, mezi větami, ...), kde za ticho je považováno vše, co je o více než 17 dB tišší než maximální hlasitost, jelikož tiché segmenty jsou irelevantní pro vyhodnocení a mohou model zmást.

Trénovací sada byla rozšířena o 2 náhodně upravené záznamy pro každou nahrávku. I v tomto případě přidání augmentací vedlo k lepším výsledkům při vyhodnocení modelu. Využité jsou následující augmentace: posun výšky hlasu, zrychlení/zpomalení záznamu, přidání šumu.

3 Systémy kombinující rozpoznávání obrazu a hlasu

Další dva systémy vznikly kombinací výsledků klasifikátorů obrazu a hlasu. Predikovaná třída na základě dvou klasifikátorů je získána následovně. Jsou získány výsledky obou klasifikátorů v podobě logaritmů pravděpodobností pro jednotlivé třídy. Ty jsou pak standardizovány odečtením střední hodnoty a vydělením směrodatnou odchylkou pravděpodobností pro všechny třídy a všechny záznamy. Díky standardizaci mají nyní výsledky podobně velké hodnoty, které pak stačí sečíst a za výslednou třídu prohlásit tu s největší hodnotou (nejedná se však už o logaritmy pravděpodobností). Zmíněné dva systémy vznikly kombinací:

1. CNN modelu pro obraz a GMM modelu pro hlas
2. SVM modelu pro obraz a GMM modelu pro hlas

4 Spuštění a reprodukce výsledků

Pro spuštění jednotlivých skriptů je potřeba použít Python verze 3.11, který podporuje všechny použité knihovny. Ty je možné nainstalovat pomocí příkazu `pip3.11 install -r requirements.txt`.

4.1 Trénování modelů

Pro všechny dříve zmíněné modely existuje právě jeden zdrojový soubor – `cnn_img.py`, `svm_img.py` a `gmm_voice.py`. V těchto souborech je možné v hlavičce souboru nastavit cesty k adresářům s trénovacími a testovacími daty (konstanty `TRAIN_DIR` a `DEV_DIR`). Při konfiguraci modelů byla data dělena do trénovací a testovací sady dle čísla sezení, tedy 6 souborů v trénovací (3 sezení) a 2 soubory v testovací (1 sezení). Dále je možné nastavit cestu pro uložení výsledných modelů, případně souborů mapovacích indexy pravděpodobností na indexy jednotlivých osob (`OUTPUT_MODEL_PATH` a `OUTPUT_MAPPER_PATH`). U modelů SVM a GMM není evaluace přesnosti modelů ve výchozím stavu nastavena a pro její vyhodnocení je třeba odkomentovat volání funkce `evaluate_model()`, není tedy nutné mít v případě potřeby natrénování modelu k dispozici testovací data. Model CNN potřebuje testovací data, jelikož je využívá jako validační při trénování modelu. Uložené natrénované modely se poté používají pro vyhodnocení výsledků (klasifikaci) neznámých dat. Pro natrénovaní modelů, ze kterých byly získány odevzdané výsledky byla použita všechna dostupná data.

4.2 Reprodukce výsledků

Pro získání výsledků klasifikace pro neznámá data je k dispozici celkem 5 systémů. První tři systémy vychází pouze z jednoho modelu (CNN, SVM nebo GMM) a následující dva tyto modely kombinují. Pro vyhodnocení výsledků klasifikace existuje pro každý systém skript, který klasifikuje všechna data ve zvolené složce a výsledky uloží v zadáním požadovaném formátu do vybraného souboru. Zdrojové soubory pro vyhodnocení systémů:

- rozpoznání obrazu pomocí CNN – `results_gen_cnn_img.py`
- rozpoznání obrazu pomocí SVM – `results_gen_svm_img.py`
- rozpoznání hlasu pomocí GMM – `results_gen_gmm_voice.py`
- kombinace rozpoznání obrazu (CNN) a hlasu (GMM) – `results_gen_cnn_img_gmm_voice.py`
- kombinace rozpoznání obrazu (SVM) a hlasu (GMM) – `results_gen_svm_img_gmm_voice.py`

V jednotlivých skriptech v sekci `main` je možné zadat cestu ke vstupním datům (`input_dir`), k natrénovanému modelu a případně k mapovacímu souboru (`model_dir`/`model_path` a `mapper_path`) a k výslednému ASCII souboru (`result_file_path`).

Odevzdané zdrojové soubory mají cesty předvyplněné tak, aby stačilo pouze přidat do kořenového adresáře data dostupná k tomuto projektu (složky `train/`, `dev/`, `eval/` a dodatečně `merged_data/`, která obsahuje sloučená `train` a `dev` data), poté postupně spouštět skripty pro trénování (bez argumentů) a následně pro generování výsledků. Je možné, že se zreprodukované výsledky budou mírně lišit, kvůli náhodně voleným augmentacím při trénování.

Odevzdaná složka `models/` je prázdná pro uložení natrénovaných modelů. V případě potřeby je možné natrénované modely, na kterých byly vyprodukovány přiložené výsledky stáhnout zde – Google drive link a následně nahradit prázdnou složku `models/`.