

# SUR PROJEKT

## Dokumentace

Tým: xbromn00, xkonec86

### Příprava dat a extrakce příznaků

Data ze souborů *train* a *dev* byla sloučena a následně náhodně rozdělena na trénovací a testovací sadu s rovnoměrným zastoupením tříd. Pro vývoj jsme používali 75 % dat na trénování (6 vzorků na třídu) a 25 % na testování (2 vzorky na třídu). Byly vytvořeny samostatné datové sady pro klasifikaci podle zvuku, podle obrázků a pro multimodální klasifikaci.

Při trénování finálního klasifikátoru byla využita všechna data z *train* i *dev*.

Na začátku jsme zvažovali extrakci příznaků z obrázků pomocí LBP (Local Binary Patterns), ze kterých byl následně sestaven histogram ke snížení množství příznaků. Tato metoda se však neosvědčila – kvůli tvorbě histogramu pravděpodobně docházelo k příliš velké ztrátě informace. Ani samotné použití LBP bez histogramu nepřineslo lepší výsledky než jiné metody.

Nejllepších výsledků jsme nakonec dosáhli použitím PCA nad původními (nerozpoznanými) obrázky, kdy byla každá obrazová matice převedena na jednorozměrný vektor.

U zvukových dat jsme použili MFCC, na které jsme následně aplikovali PCA. Z výsledné matice jedné nahrávky jsme extrahovali průměr a rozptyl jednotlivých koeficientů, které byly následně spojeny do jednoho vektoru příznaků.

Pro multimodální klasifikátor (využívající zvuk i obraz) jsme použili výše popsané postupy a následně zkombinovali příznaky ze zvuku a obrazu do jednoho vektoru, na který byla aplikována PCA.

Při párování zvukových a obrazových vzorků jsme provedli kartézský součin mezi vektory nahrávek a obrázků náležejících ke stejné třídě.

Všechny PCA transformace byly nastaveny tak, aby zachovávaly 95 % rozptylu původních dat.

### Použité modely

Pro klasifikaci jsme využili dva typy modelů z knihovny scikit-learn – SVM a GMM. Oba modely byly testovány napříč všemi modalitami (obraz, zvuk, multimodální). SVM dosahovalo v testovací sadě mírně lepších výsledků než GMM, rozdíl však nebyl výrazný.

Při trénování GMM jsme ladili kombinace dvou parametrů: počet komponent na třídu (1, 2, 4, 8) a typ kovarianční matice (diagonální nebo plná). Hodnocení kombinací parametrů probíhalo na jedné třídě, kde jsme počítali skóre podle likelihoodu náležitosti vzorku ke třídě.

V případě SVM jsme zkoušeli různé typy kernelů a hodnoty regularizačního parametru.

## Prostředí a knihovny

Projekt byl vyvíjen v jazyce Python ve verzi 3.10.12. Seznam potřebných knihoven je uveden v souboru *src/requirements.txt*.

Vytvoření virtuálního prostředí a instalace závislostí:

```
python -m venv sur-env
```

```
source sur-env/bin/activate
```

```
pip install -r requirements.txt
```

Skripty předpokládají, že ve stejné složce existuje adresář *SUR\_projekt2024-2025* obsahující složky *train* a *dev*. Pro evaluaci je třeba, aby ve stejném adresáři byla i složka *eval*.

## Spouštění skriptů

### 1. Generování datových sad

Skript *data.py* slouží k vytvoření trénovací a testovací sady. Je možné zde nastavit velikost jednotlivých sad i náhodný seed pro zajištění konzistence výsledků mezi běhy. Výstupem jsou pickle soubory s připravenými daty a extrakční pipeline pro případné nové vstupy.

### 2. Trénování klasifikátorů

Skript *classifier\_training.py* trénuje jednotlivé modely (všechny kombinace model–modalita), provádí vyhodnocení accuracy nad testovacími daty (pokud *TEST\_DATA* není nastaveno na *False*) a výsledné modely ukládá do pickle souborů.

### 3. Klasifikace jednotlivých vzorků

Klasifikaci jednoho nebo dvou vzorků (obrázek a/nebo zvuk) provádí skript *classify.py*.

Parametry:

```
--gmm nebo --svm
```

```
--mm, --aud nebo --img
```

```
--llh (volitelné pro výpis log likelihoodů)
```

```
<img_file> <audio_file>
```

Při použití *--mm* je třeba zadat jak obrázkový, tak zvukový soubor. S parametrem *--svm* a *--llh* skript nevypisuje log-likelihoody, ale vrací hodnoty *NaN*. Výstupem je klasifikace a (pokud relevantní) log-likelihood ve formátu definovaném v zadání.

### 4. Evaluace nad dávkou souborů

Pro klasifikaci více vzorků ze složky *eval* slouží bash skript *run\_eval.sh*, který spouští multimodální klasifikaci pomocí SVM a GMM. Výsledky jsou ukládány do souborů *SVM\_Audio\_Image* a *GMM\_Audio\_Image*.