

# Strojové učení a rozpoznávání (2020/2021)

## Projekt

Ondrej Šajdík, Jozef Horvát, Miroslav Bulíčka

12. mája 2021

## 1 Rozpoznávanie tváre

Pre natrénovanie klasifikátora pre rozoznávanie tvárí sme použili predtrénovanú VGG16 sieť s vlastnou sekvenčnou časťou. Túto sieť sme implementovali pomocou nástroja Keras.

### 1.1 Implementácia

Obrázky sa načítavajú do tréovacích a validačných datasetov (z priechkov `train` a `dev`), ktoré sú následne použité pri tréovaní.

Model sa skladá z predspracovacej vrstvy, zahŕňajúcej image rescaling a data augmentation. Následne sú takto predspracované obrázky vložené do predtrénovanej VGG16 <sup>1</sup> siete (všetky vrstvy sú zmrazené). Časť modelu, ktorú tréujeme je naša sekvenčná vrstva, ktorá sa skladá z jednej Flatten vrstvy, z troch Dense vrstiev (s 1024 a 512 neurónmi) a z prekladaných Dropout vrstiev. V prvých dvoch Dense vrstvách sme použili regularizáciu L2.

Pre učenie používame `SparseCategoricalCrossEntropy` loss funkciu a optimalizátor Nadam. Model je natrénovaný v 50 epochách s batch size 8.

V kóde je možné odkomentovať funkciu `plot_training_results`, ktorá zobrazí výsledky tréovanie v grafoch.

Pre dotrénovanie modelu sme skúsili vymeniť tréovacie a validačné datasety (tréovali sme na validačných datasetoch a a validovali na testovacích) alebo tieto datasety navzájom premiešať. Dotrénovanie modelu je implementované v súbore `second_learning.py`. Používa sa ten istý (ale už predtrénovaný) model. Tréuje sa, len naša sekvenčná časť. Ostatné vrstvy sú zamrazené.

Časť kódu, kde tréujeme klasifikátor sa nachádza v súbore `face_rec.py`. Natrénovaný model je uložený v súbore `face_predictions`. Kód pre predikciu tvárí je implementovaný v súbore `make_predict.py`. Skripty pre rozoznávanie tvárí sú uložené v priečinku `face_recognition.py`

---

<sup>1</sup><https://github.com/rcmalli/keras-vggface>

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
rescaling (Rescaling)	(None, 80, 80, 3)	0
vggface_vgg16 (Functional)	(None, 2, 2, 512)	14714688
dropout (Dropout)	(None, 2, 2, 512)	0
flatten (Flatten)	(None, 2048)	0
fc6 (Dense)	(None, 1024)	2098176
dropout_1 (Dropout)	(None, 1024)	0
fc7 (Dense)	(None, 512)	524800
dropout_2 (Dropout)	(None, 512)	0
fc8 (Dense)	(None, 31)	15903

```

Total params: 17,353,567
Trainable params: 2,638,879
Non-trainable params: 14,714,688

```

Obr. 1: Architektúra modelu

## 1.2 Úspešnosť modelu na validačných dátach

Úspešnosť modelu po prvom trénoch sa pohybovala okolo 40%. Po dotrénoch (spustili sme skript 3 až 5-krát) sme obdržali následovné výsledky:

Accuracy: 0.7678571343421936

Loss: 1.5052175521850586

## 1.3 Závislosti

Závislosti je možné nájsť v súbore *requirements.txt*. Je ich možné nainštalovať pomocou príkazu (je potrebné byť v pracovnom priečinku):

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

## 1.4 Spustenie aplikácie

Trénoch modelu a následná klasifikácia (je potrebné byť v pracovnom priečinku):

```
python face_rec.py
python second_learning.py
python make_predict.py
```

Priečinky s trénochmi, validačnými a testovacími dátami musia byť na rovnakej úrovni ako je pracovný priečinok. Predikcie tváří a natrénochovaný model sa ukladajú v rámci pracovného priečinku.

## 2 Rozpoznávanie hlasu

Klasifikátor sa skladá z množiny multivariačných Gaussových rozdelení, kde každé rozdelenie predstavuje jednu triedu.

### 2.1 Implementácia

Pre extrakciu príznačkov bola použitá knižnica *librosa*. Z wav súborov boli získané MFCC príznaky. Sampling rate bol nastavený na 16kHz a u každého vzorku bolo extrahované 40 koeficientov. S týmito konštantami sa nám podarilo získať najlepšie výsledky. Vzorky rovnakej triedy zo všetkých súborov boli spojené do jedného.

Takto vytvorené dáta boli použité pre natrénovanie množiny multivariačných Gaussových rozdelení. **Gaussove rozdelenia** sú implementované pomocou *GaussianMixture* z knižnice *sklearn.mixture*. Jedná sa o zmes gausoviek ale počet komponent je nastavený na jedna. *GaussianMixture* bol vybraný pretože obsahuje implementované funkcie *fit* pre odhad most-likelyhood parametrov a funkciu *score* pre vyhodnotenie priemerného log-likelihoodu pre zadané data X.

Trieda nahrávky je získaná vypočítaním score nahrávky u každej gausovky. Výsledná trieda je tá , ktorá má najvyššie skóre.

### 2.2 Spôsob vyhodnotenia

Kvalita modelu je vyhodnotená ako pomer správne klasifikovaných nahrávok ku celkovému počtu nahrávok na dátach v priečinku *dev*.

### 2.3 Zavislosti a spustenie

Závislosti:

```
Python 3.6.0
librosa 0.8.0
numpy 1.19.5
```

Spustenie:

```
python3 audio.py
```

Výstup

```
Hits: 46 / 62
Accuracy: 0.7419354838709677
```

Výsledky klasifikácie nahrávok zo zložky *eval* je možné nájsť v súbore *voice\_predictions* a sú vytvorené na konci behu programu.