

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

SUR – Strojové učení a rozpoznávání
Klasifikace obličejů a řeči

8. května 2020

Martin Kostelník (xkoste12)
Ivo Meixner (xmeixn00)
Adam Gajda (xgajda07)

1 Rozpoznávání obličejů

1.1 Klasifikátor

Původní plán byl vytvořit klasifikátor obličejů pomocí knihovny TFLearn. Úspěšně jsem model vytvořil a otestoval. Bohužel jsem ale nemohl najít optimální konfiguraci neuronové sítě a model nefungoval úplně správně. Většinou označil úplně všechny data jako non-target a výsledná přesnost byla tedy 0.8571, což je ale poměr obrázků non-targetu ku všem datům. Tento model tedy není funkční a ve finálním odevzdání nebyl použit. Z kódu ovšem odstraněn nebyl, byl pouze zakomentován a všechny jeho části jsou označeny komentáři.

Rozhodli jsme se tedy použít knihovnu Keras. Vytvoření a otestování jednoduchého modelu bylo velice podobné modelu TFLearn. Nejdůležitější faktor pro použití této knihovny byl však nástroj Keras Tuner, který nám umožnil najít funkční (a v rámci možností neoptimálnější) model. Nejprve jsme se seznámili s tím, jak Tuner funguje a vyzkoušeli jsme si najít několik málo modelů a z nich vybrat ten nejlepší. Jakmile jsme měli s Tunerem nějakou zkušenost, rozšířili jsme hodnoty hyper parametrů aby bylo vyzkoušeno více modelů a jednotlivé modely vyzkoušeny vícekrát. Hledání nejlepšího modelu nakonec trvalo asi šest hodin. Po dokončení ladění jsme měli k dispozici model, který byl nakonec využit k finální klasifikaci a generování výsledků. Samotný model je popsán v souboru model_shape.

1.2 Reprodukce výsledků

1. Použít python 3.6.9
2. Instalace potřebných balíčků pomocí příkazu `pip install -r requirements.txt`
3. Nastavit globálních proměnných v souboru main.py
 - TRAIN_DIR – Adresář se všemi trénovacími daty
 - VALIDATION_DIR – Adresář se všemi validačními daty
 - TARGET_LABEL – Prefix obrázků obsahujících target
 - TEST_DIR – Adresář s testovacími daty
 - LOG_DIR – Adresář s výsledky nástroje Keras Tuner
4. Připravit trénovací data. Při prvním použití je třeba spustit funkce `create_data()` pro vytvoření trénovacích a validačních dat a funkci `process_test_data()` pro vytvoření testovacích dat. Tyto funkce vytvoří data v přípustném formátu pro model. Data jsou poté uloženy v souborech `testing_data.npy` a `validation_data.npy`. Při opakovaném použití není třeba data znovu vytvářet a stačí je načíst.
5. Při nepřítomnosti použitého modelu je dále třeba odkomentovat příslušné řádky (LINE 187-201). Nicméně hledání nejlepšího modelu může trvat i několik hodin. Další možností je model ručně vytvořit podle popisu v souboru `model_shape`, ale to může být problematické. Pro použití totožného modelu jako jsme použili my, je náš model dostupný v GitHub repozitáři na adrese: https://github.com/natiix/vut-fit-sur/blob/master/facial_recognition/best_model.h5. Tento model lze jednoduše načíst provedením funkce `load_model()`. Z důvodu velikosti jsme jej nemohli přiložit do odevzdaného archivu.
6. Pokud je přítomen model a jsou nachystány testovací data, stačí provést funkci `recognize()` a výsledky budou vypsány na standardní výstup.