

Michał Marciniak

PROJEKT

"Dogs vs Cats"

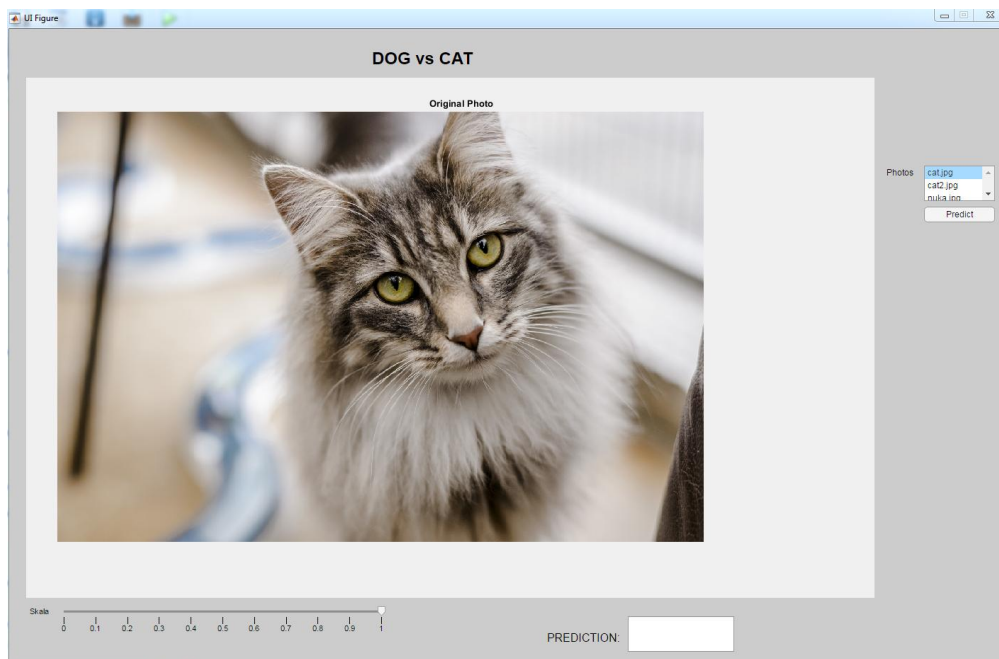
Inżynieria Biomedyczna: rok akademicki 2018/2019, III semestr

Kurs: "Informatyka", data oddania: 20.01.2019

Cel Projektu:

Założeniem było zwieńczenie ostatnich miesięcy nauki w dziedzinie głębokiego uczenia i wytrenowanie sieci neuronowej zdolnej do zadania klasyfikacji (rozpoznawanie czy na obrazie znajduje się pies czy kot). Główny cel to nabycie doświadczenia w dziedzinie uczenia maszynowego, oraz tworzenia atrakcyjnego interfejsu umożliwiającego wygodną obsługę programu.

Opis działania:



Rys. 1 Wygląd aplikacji

W aplikacji wybieramy interesujące nas zdjęcie. Przy użyciu suwaka możemy zmienić jego rozmiar (w przypadku wykorzystania GUI jest to ograniczone ze względu na automatyczne dopasowanie obrazu do rozmiaru obiektu Axes appdesigner'a). Po wybraniu zdjęcia i kliknięciu przycisku Predict zdjęcie przesyłane jest na wejście sieci neuronowej przy wcześniejszej obróbce. Wytrenowana sieć wyznacza prawdopodobieństwo przynależności do jednej z dwóch klas. Wynik zwracany jest do interfejsu i wyświetlany w okienku Prediction.

Funkcje w GUI:

startupFcn(app) - wywoływana przy uruchomieniu aplikacji, inicjuje początkowy stan programu poprzez zapełnienie PhotoBoxa zdjęciami, wyświetlenie domyślnego zdjęcia i ustawienie domyślnych wartości widgetów.

PhotosListBoxValueChanged(app, event) - Funkcja wywoływana przy zmianie wybranego elementu w PhotoBox'ie. Wyświetlane jest wybrane zdjęcie.

PredictButtonPushed(app, event) - Funkcja służąca do przesłania danych do skryptu Pythona (psyikoty.py) gdzie następuje klasyfikacja zwierzęcia na zdjęciu. Następnie wynik wyświetlany jest w polu Prediction.

SkalaSliderValueChanging(app, event)- Funkcja wywoływana przy zmianie wartości suwaka. Służy do skalowania zdjęcia w stosunku do jego rozmiaru oryginalnego. Wartości na suwaku odpowiadają wartością procentowym.

Szczegółowy poszczególnych funkcji został zawarty jako komentarze w kodzie źródłowym.

Informacje o sieci:

Do treningu wykorzystano zbiór danych z strony kaggle.com: <https://www.kaggle.com/c/dogs-vs-cats/data>

Jest to 25.000 zdjęć psów i kotów z odpowiadającymi im etykietami.

Obróbka danych, przygotowanie zbiorów i trening odbył się w google collaboratory (przy wykorzystaniu darmowego GPU). Kod z tej sekcji znajduje się w pliku Projekt.ipynb, który należy otworzyć właśnie w collabie.

Na dzień dzisiejszy (tj. 20.01.2019) skrypt pythona należy odpalać w wersji 3.6 (ze względu na problem z biblioteką tensorflow).

Wnioski: Można usprawnić działanie modelu poprzez zastosowanie między innymi standaryzacji danych, dzięki czemu sieć szybciej uzyska zbieżność do minimum co przyspieszy proces treningu. Dodatkowo pozwala to rozwiązać problem "zanikających" i "wybuchających" gradientów.

Można również zastosować augmentację danych (data augmentation) poprzez manipulację na obrazach z zbioru treningowego. Umożliwia to wygenerowanie dodatkowych danych, co usprawni proces treningowy i pozwoli zmniejszyć szansę na przetrenowanie modelu.

W tworzeniu projektu wykorzystana została dokumentacja:

-Python: <https://docs.python.org/3/>

-Tensorflow: <https://www.tensorflow.org/>

-Numpy: <https://docs.scipy.org/doc/>

-Matlab: https://www.mathworks.com/?s_tid=gn_logo

oraz:

<https://stackoverflow.com/>

Projekt można znaleźć na moim profilu github: <https://github.com/MMarciniak103>