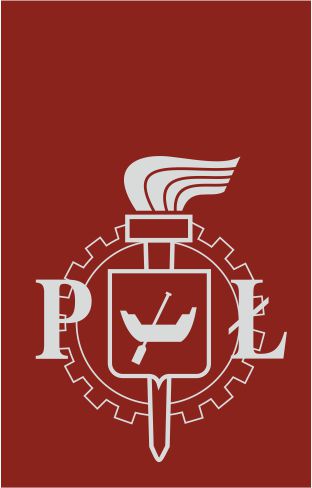
**20.09.2018, Łódź**



**System rozpoznawania twarzy**

**Dokumentacja Techniczna**

# **Python i biblioteki**

## **Windows**

Aplikacja została stworzona z użyciem języka programowania **Python w wersji 3.6.6 dla systemów 64 bitowych**. Główne biblioteki użyte w programie są zamieszczone poniżej.



Przy instalacji wyżej zamieszczonych bibliotek istnieje potrzeba zainstalowania zależnych bibliotek, które w pewnych przypadkach mogą nie zostać zainstalowane automatycznie. Szczególnie istotne jest to dla biblioteki Keras (nakładka na Tensorflow) umożliwiającej tworzenie sieci neuronowych. Z tego też względu poniżej zamieszczone są wszystkie biblioteki wraz z zależnościami wymaganymi do uruchomienia oprogramowania – spis bibliotek został wygenerowany na podstawie środowiska wirtualnego stworzonego na potrzeby stworzenia oprogramowania.



## **Linux**

W przypadku systemu linux aplikacja była testowana na **ubuntu 18.04 LTS** z **użyciem Python 3.6.5**, czyli domyślnego pythona zapisanego jako python3. W razie problemów zamieszczona poniżej lista zawiera wszystkie biblioteki zainstalowane dla python3 wraz z domyślnymi bibliotekami zainstalowanymi przy tworzeniu systemu.



## **Instalacja bibliotek**

Wskazanym jest instalowanie wymaganych bibliotek używając:

* ***pip install -r nazwa\_pliku.txt*** dla pythona globalnie
* bądź jako: ***sciezka\do\virtualenvironment\Scripts\python –m pip install –r nazwa\_pliku.txt*** w przypadku konkretnego środowiska wirtualnego (oczywiście niezbędne jest wcześniejsze dodanie pythona do PATH w zmiennych środowiskowych).

Instalacja tych bibliotek jest wystarczająca do uruchomienia aplikacji z obliczeniami sieci neuronowej z użyciem CPU (używana jest biblioteka tensorflow, a nie biblioteka tensorflow-gpu).

# **CUDA – obliczenia na GPU**

Niezależnie od systemu operacyjnego zawsze istotnym jest zainstalowanie aktualnych sterowników odpowiednich dla danej karty graficznej.

## **WINDOWS**

W przypadku obliczeń z użyciem GPU wymagana jest karta graficzna z CUDA. Biblioteka tensorflow i keras nie jest kompatybilna ze wszystkimi wersjami NVIDIA CUDA Toolkit, dlatego też wskazane jest pobranie wersji **CUDA Toolkit 9.0** wraz z **cuDNN v7.2.1 (August 7, 2018), dla CUDA 9.0.** Instalacja przebiega następująco:



Następnie należy zainstalować bibliotekę **tensorflow-gpu** (preferowalnie w wersji 1.10.0), a następnie bibliotekę **keras** (przykładowo z użyciem *pip install nazwa\_bibioteki*). Istnieje możliwość, że instalacja biblioteki keras zainstaluje tensorflow dla CPU, w takiej sytuacji należy odinstalować tensorflow i zainstalować tensorflow-gpu ponownie. Dodatkowo, w niektórych przypadkach wymagany może być **Microsoft Visual Studio 2015**.

## **LINUX**

W przypadku systemów linux procedura jest analogiczna. **Trzeba jednak samemu zadbać o ustawienie ścieżek na koniec ~/.bashrc lub ~/.zshrc** po instalacji CUDA Toolkit i cuDNN**:**

export PATH=/usr/local/cuda-9.0/bin${PATH:+:${PATH}}

export LD\_LIBRARY\_PATH=/usr/local/cuda/lib64:${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

Dodatkowo, wymagane jest zainstalowanie **libcupti**:

sudo apt-get install libcupti-dev

Zdając sobie sprawę z napisania jedynie krótkiej instrukcji nie zakładającej ewentualnych komplikacji, które mogą wystąpić z różnych przyczyn chcę zaznaczyć, że procedury te są bardzo dokładnie opisane przez NVIDIA wraz z rozwiązywaniem problemów.

# **System operacyjny**

Jedyny wymóg co do systemu operacyjnego jest związany z CUDA, która wymaga aby system operacyjny był 64 bitowy.

# **Sprzęt**

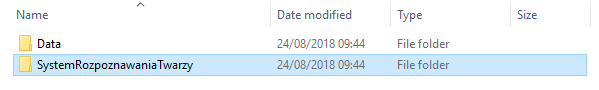
Istnieją cztery wytyczne:

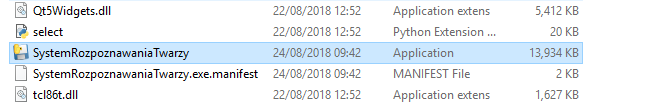
* Podłączenie sprawnej kamery do komputera. Niezależnie czy jest to kamera internetowa w laptopie, czy dokupiona kamera do komputera stacjonarnego – oprogramowanie powinno działać tak samo.
* Podłączenie Internetu. Aplikacja przy pierwszym użyciu musi ściągnąć model vgg-face z Internetu aby transfer wiedzy był możliwy (czynność jednorazowa).
* Karta graficzna z CUDA jeśli na stanowisku mają być wykonywane obliczenia z użyciem GPU.
* Minimum 4 GB RAM. Wszystkie zdjęcia przechowywane są w pamięci RAM i dlatego też przy większej ilości dodanych osób należałoby zapewnić nawet 8 GB RAM.

# **Uruchamianie**

## **Wersja bez instalacja – CPU - standalone**

Dla aplikacji używających jedynie CPU zostały przygotowane dwie dystrybucje (na Windows i Linux), które powinny działać bez instalowania żadnych dodatkowych bibliotek, a jedynie przez uruchomienie „SystemRozpoznawaniaTwarzy” w folderze „SystemRozpoznawaniaTwarzy”.





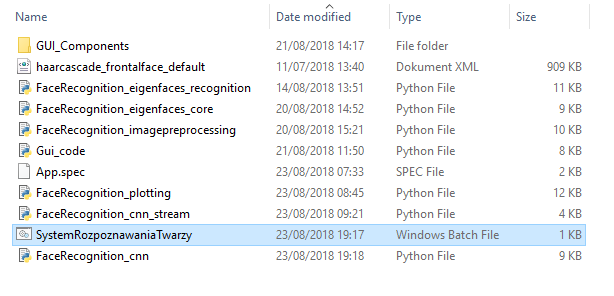
W przypadku wersji dla linuxa – możliwość stworzenia pliku, który będzie uruchamiany podwójnym kliknięciem istnieje – ale należy to zrobić po skopiowaniu aplikacji do systemu (wymagana jest stała ścieżka do pliku).

Niestety niemożliwym jest zrobienie wyżej opisanych dystrybucji dla GPU, jako że jest zbyt wiele zależności, które uniemożliwiają powiązanie i skopiowanie istotnych elementów z NVIDIA Toolkit etc.

**Jedyną różnicą pomiędzy dystrybucjami** jest pewien brak kompatybilności dla linuxa z PyQt5 i z tego też względu obrazy z kamery nie są wyświetlane w oknie aplikacji. Jest to związane z brakiem możliwości osadzenia obrazu z kamery bezpośrednio w aplikacji (wyświetlanie obrazu przez dodatkowe okno) – kod dla wersji linux i dla wersji windows różni się jedynie małą zmianą w jednym pliku (SystemRozpoznawaniaTwarzy.py) gdzie w przypadku windowsa w 4 funkcjach argumentem jest (gui=self), zaś w przypadku linuxa linie te musiały zostać usunięte.

## **Wersja do instalacji**

W przypadku instalacji bibliotek w systemie bez użycia wcześniej wspomnianego rozwiązania w wersji dla Windows aplikacja uruchamiana jest przez „SystemRozpoznawaniaTwarzy.bat”.



W przypadku linuxa zaś stworzenie takiego pliku nie znając miejsca docelowego gdzie zostanie skopiowana aplikacja nie jest możliwa (tak jak wspomniano przy wersji standalone). W tej sytuacji zostaje użycie terminalu i wywołanie kodu za pomocą python3 sciezka/do/pliku/Gui\_code.py, bądź stworzenie skrótu uruchamiającego po instalacji.