

ROZPOZNAWANIE I PRZETWARZANIE OBRAZÓW

GENERATOR MASEK REAL-TIME

ZADANIE 1 - PLAN REALIZACJI PROJEKTU

Maja Bojarska, 241287

Wiktor Pieklik, 241282

Grupa projektowa nr 2

7 kwietnia 2020

1 Cele projektu

Celem naszego projektu jest poznanie technik i narzędzi do rozpoznawania twarzy oraz śledzenia jej w czasie rzeczywistym, w celu rysowania na niej maski. Wymaga to określenia potrzebnych punktów charakterystycznych twarzy. Aby wykrywać ich położenie w krótkim czasie (bez zauważalnego opóźnienia), konieczne jest przygotowanie wyspecjalizowanego modelu predyktora cech. Ostatecznie, aplikacja ma być naszą interpretacją ogólnodostępnych, komercyjnych narzędzi, takich jak Instagram lub Snapchat.

2 Sposoby dojścia do celu

Podczas realizacji projektu wprowadziliśmy konwencje i założenia dotyczące organizacji pracy:

- Kod źródłowy przetrzymywany jest w serwisie GitHub,
- Definiowanie nowych zadań i ich przydział odbywa się w serwisie Trello,
- Głównym kanałem komunikacji i konsultacji jest komunikator Slack,
- Przyjętym zintegrowanym środowiskiem programistycznym jest PyCharm.

2.1 Narzędzia

- język Python ≥ 3.7 .
 - Narzędzia wykorzystane do przetwarzania obrazów:
 - Pillow - zbiór narzędzi do obróbki grafiki[1],
 - numpy - obliczenia naukowe, pythonowa alternatywa dla języka MATLAB[2],
 - opencv-python - "Open Source Computer Vision Library"[3],
 - dlib - zbiór narzędzi uczenia maszynowego[4].
 - Inne narzędzia:
 - PyQt5 - framework GUI[5].

2.2 Algorytmy

- Algorytm "Ensemble of Regression Trees (ERT)"[7].

2.3 Dane

- Zestaw danych iBUG 300-W[6].

2.4 Plan realizacji

- Rozpoznawanie twarzy:
 1. określenie potrzebnych punktów charakterystycznych.
 2. Wytrenowanie modelu predyktora na podstawie zestawu danych iBUG 300-W[6], mając na uwadze szybkość predykcji, potrzebną do rysowania masek w czasie rzeczywistym.
 3. Zastosowanie modelu wynikowego do predykcji położenia punktów charakterystycznych na obrazie pobieranym ze źródła obrazu (docelowo kamery).
- Maski:
 1. Określenie interfejsów wejścia/wyjścia generycznej maski.
 2. Określenie wymagań co do rodzajów masek, w tym aspektów estetycznych.
 3. Implementacja rysowania konkretnych masek, zgodnych z ustalonym interfejsem.
- Interfejs graficzny:
 1. określenie wymagań - funkcjonalności interfejsu graficznego oraz jego widoków,
 2. projekt położenia elementów typu “wireframe”, dla poszczególnych widoków.
 3. Implementacja widoków.
 4. Integracja GUI z logiką programu.

Literatura

- [1] Pillow 7 Documentation: <https://pillow.readthedocs.io/en/stable/>
- [2] Numpy Reference: <https://docs.scipy.org/doc/numpy/reference/>
- [3] OpenCV-Python: https://docs.opencv.org/master/d6/d00/tutorial_py_root.html
- [4] Dlib C++ Library: <http://dlib.net/>
- [5] PyQt5 Reference Guide: <https://www.riverbankcomputing.com/static/Docs/PyQt5/>
- [6] 300 Faces In-the-Wild Challenge: <https://ibug.doc.ic.ac.uk/resources/300-W/>
- [7] Kazemi, Vahid and Sullivan, Josephine - One Millisecond Face Alignment with an Ensemble of Regression Trees: http://www.csc.kth.se/~vahidk/face_ert.html
- [8] Jacqueline G. Cavazos, P. Jonathon Phillips, Carlos D. Castillo, Alice J. O'Toole - Accuracy comparison across face recognition algorithms: <https://arxiv.org/abs/1912.07398>