

Dokumentacja projektu wykonywanego w ramach zajęć ANALIZA OBRAZÓW

Dominik Smęda

Wojciech Maciąg

Bartosz Gawron

14.01.2024

1. Opis projektu

Program ma na celu wczytanie obrazu wskazanego przez użytkownika, a następnie na danym pliku sprawdzić, czy wykrywalna jest ludzka twarz – jeżeli tak, to określa pozycję oczu, aby na nich zastosować określone przez użytkownika operacje morfologiczne.

Technologie zastosowane w projekcie to języki JavaScript oraz HTML, a aplikacja użytkownika stworzona jest jako serwis internetowy.

Program wykorzystuje sieć neuronową do detekcji twarzy.

Funkcjonalności zaimplementowane w projekcie:

- wczytywanie obrazu (Input Source na rysunku 1), gdzie użytkownik wybiera obraz znajdujący się na jego urządzeniu, bądź rejestruje obraz z kamery użytkownika,
- określenie dla ilu twarzy mają zostać wykonane operacje morfologiczne (Detection Options na rysunku 1):
 - opcja All faces aplikuje operacje na wszystkie rozpoznane twarze,
 - opcja Faces with labels natomiast pozwala na wybór przez użytkownika określonych twarzy ze wskazaniem nazw dla każdej z nich (Rysunek 1),
- wskazanie operacji morfologicznej, która ma zostać wykonana na wskazanych przez użytkownika twarzach (Mask Options na rysunku 1):
 - opcja Blur rozmywa obiekt wykryty przez program jako obszar oczu,
 - opcja Pixelized przekształca wykryty obiekt w obraz pikseli,
 - opcja Contrast ustawia kontrast na obiekcie,
 - opcja Invert dokonuje inwersji kolorów na danym obiekcie,
 - opcja GrayScale przedstawia obiekt w odcieniach szarości,dodatkowo użytkownik ma możliwość wyboru koloru rgb (Color Map na rysunku 1), w którym wskazany obiekt ma zostać przedstawiony po wykonaniu przez program operacji morfologicznej, ponadto aplikacja daje możliwość ustawienia efektywności danej operacji morfologicznej (Factor na rysunku 1),
- sprawdzanie poprawności programu względem naszego obrazu (Detection Debug na rysunku 1):
 - opcja Show Face Box pokazuje miejsce, w którym program wykrył twarz,
 - opcja Show Face Landmarks pokazuje punkty, na podstawie których sieć neuronowa sklasyfikowała dany obiekt jako twarz,
 - opcja Show Face Label wyświetla nazwę przypisaną do danego obiektu.

Input Source

Image

Wybierz plik Nie wybrano pliku

Detection Options

Mode: All faces

Labels: label1, label2, ...

Mask Options

Effect: Blur

Factor:

Color Map: None

Detection Debug

☐ Show Face Box

☐ Show Face Landmarks

☐ Show Face Label

Rysunek 1: Panel użytkownika

2. Wymagania oraz uruchamianie programu

Aby uruchomić program potrzebne jest zainstalowanie IDE Visual Studio Code, w którym należy otworzyć folder zawierający pliki projektu.

Następnym krokiem jest zainstalowanie w pobranym IDE rozszerzenia Live Server. Aby to zrobić należy wybrać zakładkę Exstensions (CTRL + SHIFT + X) w Visual Studio Code, a w niej wyszukać interesujące nas rozszerzenie i zainstalować je w naszym IDE.

Ostatnim krokiem jest wyszukanie pliku index.html, na którego w IDE klikamy prawym przyciskiem myszki i wybieramy opcje Open with Live Server (ALT + L a następnie ALT + O). Aplikacja użytkownika uruchomi się w oknie naszej domyślnej przeglądarki.

3. Problemy oraz możliwe rozwinięcie programu

Opcja Faces with labels została zaimplementowana tylko dla obrazków znajdujących się w folderze labeled_images, które „na sztywno” wczytywane są przez program do rozpoznania. Aplikacja nie daje możliwości przypisania nazwy dla danej twarzy, która następnie byłaby rozpoznawalna przez program, działa to tylko dla obrazków, w których nazwy obiektów zostały zdefiniowane w programie. Opcją rozszerzającą program jest właśnie dodanie funkcjonalności przypisywania nazw do obiektów znajdujących się na obrazie przesłanym przez użytkownika.

Innym możliwym rozwinięciem projektu jest implementacja funkcjonalności, która rysowałaby na obrazie wczytanym przez użytkownika na przykład okulary. Program wykorzystywałby odległości pikseli lewego i prawego oka od wierzchołka nosa, których używa do detekcji twarzy, i na ich podstawie określał kierunek skierowania twarzy, a następnie rysował obiekt w znalezionym obszarze.

4. Podział pracy

- Dominik Smęda – zaprojektowanie GUI i implementacja funkcjonalności wczytywania obrazu z pliku bądź kamery,
- Wojciech Maciąg – implementacja funkcjonalności Detection Options oraz Detection Debug,
- Bartosz Gawron – implementacja operacji morfologicznych, dokumentacja.