

Studia niestacjonarne
Wydział Elektryczny
Kierunek Informatyka

Projekt: Rozpoznawanie Twarzy

Przedmiot : Podstawy Telekomunikacji

Wykonali:
Robert Frysiak
Przemysław Olchowik

SPIS TREŚCI

1. Opis i uzasadnienie dlaczego wybrany został dany temat,
2. Podział prac pomiędzy członków zespołu,
3. Wylistowanie i opisanie funkcjonalności oferowanej przez aplikację,
4. Wybrane technologie wraz z uzasadnieniem dlaczego,
5. Architektura rozwiązania (jak jest zbudowane),
6. Interesujące problemy i rozwiązania ich na jakie się natknęliście
7. Instrukcja użytkowania aplikacji,

Wstęp

System rozpoznawania twarzy to obecnie jedna z najszybciej rozwijających się technologii sztucznej inteligencji. Powstał dla poprawy bezpieczeństwa w miejscach publicznych. Weryfikowanie osoby odbywa się poprzez porównywanie i analizowanie wzorców na podstawie charakterystycznych cech i konturów twarzy.

W rzeczywistości technologia rozpoznawania twarzy zyskała znaczną uwagę, ponieważ ma potencjał w szerokim zakresie zastosowań związanych z egzekwowaniem prawa i kontrolowaniem zachowań społeczeństwa.

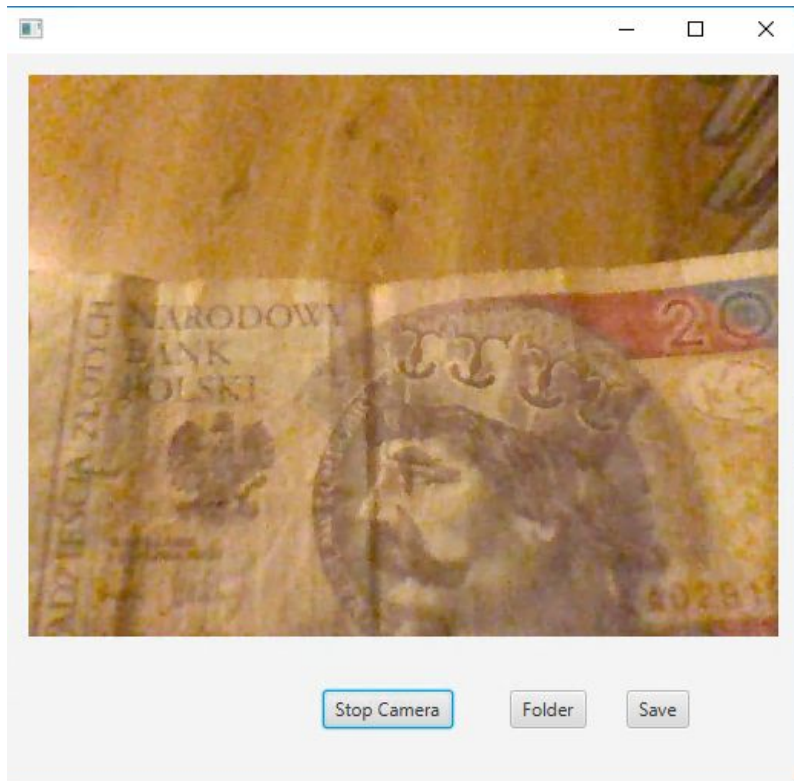
Istnieje wiele zalet związanych z rozpoznawaniem twarzy. W porównaniu z innymi technikami biometrycznymi ma charakter bezkontaktowy. Obrazy twarzy można uchwycić z dystansu i przeanalizować bez konieczności jakiegokolwiek interakcji z użytkownikiem / osobą. W rezultacie żaden użytkownik nie może z powodzeniem naśladować innej osoby. Rozpoznawanie twarzy może być doskonałym środkiem bezpieczeństwa do śledzenia czasu i obecności. Rozpoznawanie twarzy to również tania technologia.

Istnieją również pewne wady. Rozpoznawanie twarzy może identyfikować osoby, gdy oświetlenie i warunki atmosferyczne są korzystne.

Wybierając ten projekt chcemy stworzyć aplikację czasu rzeczywistego wykorzystując fizyczny sprzęt z dziedziny Internet of Things (IoT). Nie mieliśmy wcześniej styczności z przetwarzaniem obrazu powiązanym z językiem programowania.

Funkcjonalności oferowane przez aplikację.

- Interaktywny interfejs graficzny JavaFX.



Okno aplikacji jest uruchamiane z wyłączonym przechwytem obrazu. Kiedy program jest już w stanie running użytkownik uruchamia przycisk Start Camera rozpoczynając inicjację kamery i przechwytywanie obrazu. Ładowanie pozostałych komponentów odbywa się automatycznie.

- Dodawanie folderów i umieszczanie zdjęć w katalogu.

Przycisk folder otwiera eksplorator plików Windows, gdzie możemy dodać nowy katalog twarzy do bazy danych. Przycisk save zapisuje bieżące zdjęcie do wybranego katalogu.

- Odczyt bazy danych w formacie .csv do aplikacji.

Baza danych składa się z rekordów zawierających dokładnie dwie informacje rozdzielone średnikiem tj. ścieżka do pliku ze zdjęciem oraz index osoby w formie liczby naturalnej.

```

1 src\\facedetect\\Baza\\s1\\1.pgm ; 0
2 src\\facedetect\\Baza\\s1\\10.pgm ; 0
3 src\\facedetect\\Baza\\s1\\2.pgm ; 0
4 src\\facedetect\\Baza\\s1\\3.pgm ; 0
5 src\\facedetect\\Baza\\s1\\4.pgm ; 0
6 src\\facedetect\\Baza\\s1\\5.pgm ; 0
7 src\\facedetect\\Baza\\s1\\6.pgm ; 0
8 src\\facedetect\\Baza\\s1\\7.pgm ; 0
9 src\\facedetect\\Baza\\s1\\8.pgm ; 0
10 src\\facedetect\\Baza\\s1\\9.pgm ; 0
11 src\\facedetect\\Baza\\s2\\1.pgm ; 1
12 src\\facedetect\\Baza\\s2\\10.pgm ; 1
13 src\\facedetect\\Baza\\s2\\2.pgm ; 1
14 src\\facedetect\\Baza\\s2\\3.pgm ; 1
15 src\\facedetect\\Baza\\s2\\4.pgm ; 1
16 src\\facedetect\\Baza\\s2\\5.pgm ; 1
17 src\\facedetect\\Baza\\s2\\6.pgm ; 1

```

- Uczenie modelu rozpoznającego twarze osób z bazy.

Poznanie przez program nowych twarzy odbywa się w przy pomocy metody “train()”, która dołączana jest wraz z klasą FaceRecognizer. Jako parametry przekazujemy dwa kontenery:

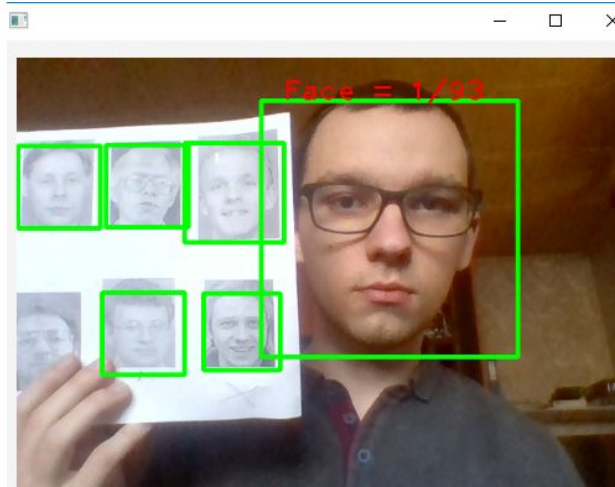
- images ze zdjęciami
- labels z etykietami osób

```

ArrayList aList= new ArrayList(Arrays.asList(data.split(" ; ")));
for(int j=0;j<aList.size();j++)
{
    if(j==0)
    {
        String load0=(String) aList.get(j);    //Ładowanie lewej części split-a
        images.add(Imgcodecs.imread(load0, 0));
    }
    else
    {
        String load1=(String) aList.get(j);    //Ładowanie prawej części split-a
        inty.add(Integer.parseInt(load1));
    }
}

```

- Porównywanie twarzy w czasie rzeczywistym i wyświetlanie kwadratów z twarzami i podpisem.



Nakładanie tekstu na ramkę jest ograniczone parametrem prawdopodobieństwa. Jeżeli prawdopodobieństwo danej twarzy wyniesie więcej niż 100, to tekst nie zostanie wyświetlony.

Podział prac

Przemek:

- Stworzenie bazy znanych twarzy.
- Przetworzenie i import danych z bazy do programu.
- Dodanie funkcjonalności dodawania nowych osób do bazy.
- Porównanie pobranych klatek obrazu ze znanymi twarzami.

Robert

- Kompilacja źródeł biblioteki OpenCV
- Przygotowanie interfejsu graficznego.
- Pobranie obrazu z kamery do GUI (Graficzny interfejs użytkownika).
- Wyznaczenie obszaru twarzy na przechwyczonej klatce obrazu za pomocą klasyfikatora.
- Wyświetlenie rezultatu rozpoznawania na ekranie wraz ze wskaźnikiem prawdopodobieństwa.

Wybrane technologie



Język programowania: Java

Wybraliśmy ten język programowania, ponieważ mieliśmy okazję zapoznać się z nim na innym przedmiocie z dziedziny programowania.

Klarowność tworzenia interfejsu, dodawanie do niego kolejnych obiektów graficznych JavaFX oraz dostępność poniższej biblioteki OpenCV zdecydowanie nakreśliły nam wstępny szkielet projektu.

Biblioteka rozpoznawania twarzy: OpenCV

OpenCV jest bezpłatną, open-source'ową biblioteką do użytku komercyjnego i edukacyjnego. Została napisana w języku C i C++. Rozwijana jest za pomocą wrapperów w wielu językach programowania, jednym z nich jest wybrana przez nas Java. Na oficjalnej stronie znajduje się pełna dokumentacja zawartych w niej klas i funkcji.

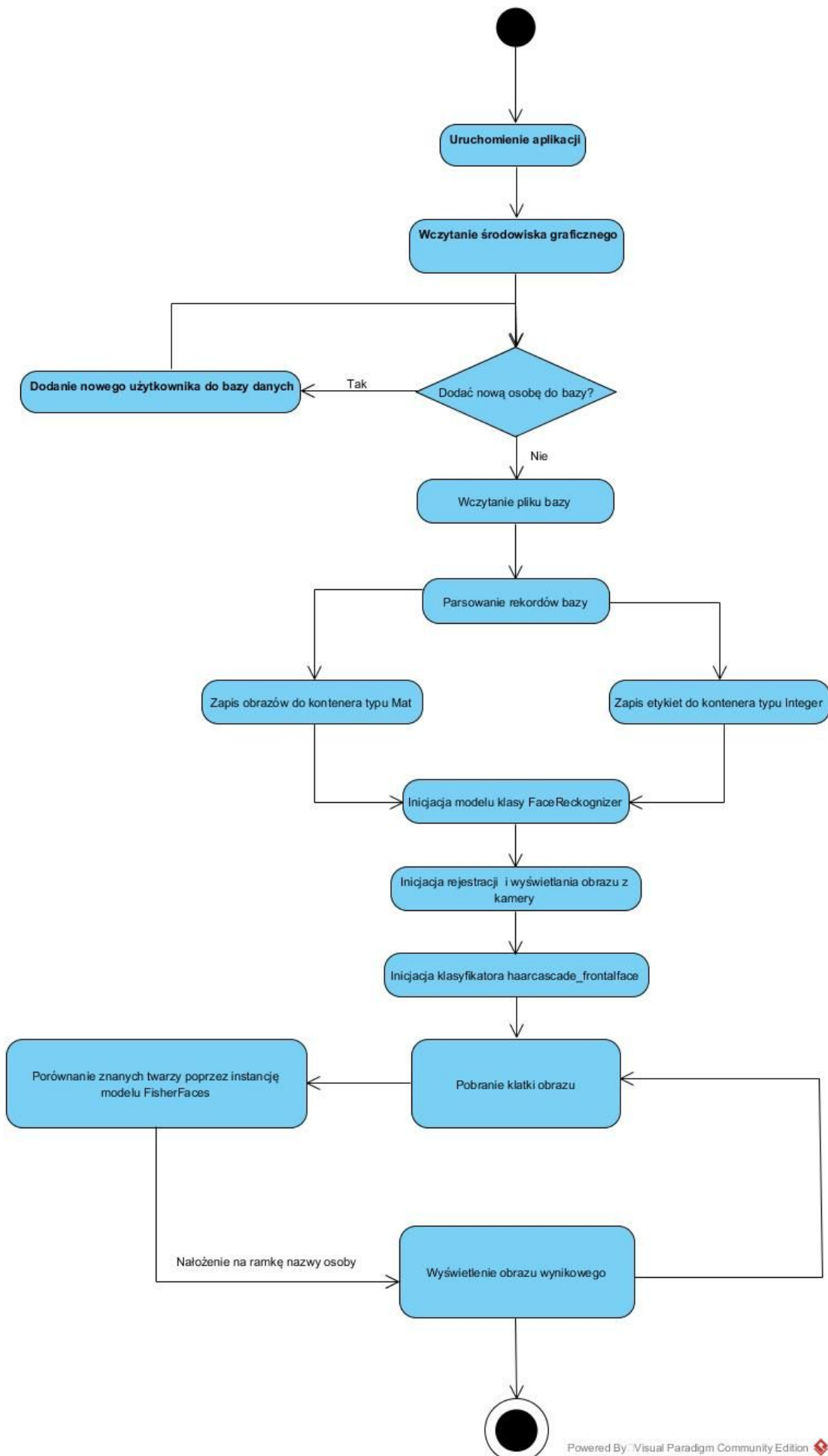


Repozytorium Github

Github umożliwia natychmiastową synchronizację tworzonego projektu w chmurze. Dzięki niemu możemy dzielić się opracowanymi rozwiązaniami, ewidencjonować poczynione zmiany, przywracać je i przeglądać różnice w kodzie względem naszego kodu. Wszystkie te cechy przyczyniają się do przyspieszenia rozwoju prac nad projektem.



Architektura rozwiązania



Moduł video - biblioteka posiada wbudowany moduł do obsługi kamery. Pozwala on na przechwycenie ramki i zapisanie jej jako obiekt Mat.

Klasyfikator kaskadowy - służy do wykrywania obiektów na obrazie. Biblioteka OpenCV posiada kilka zdefiniowanych klasyfikatorów które potrafią zaznaczać na obrazie obszary przedstawiające:

- oczy
- nos
- usta
- całą twarz
- całą postać
- uśmiech
- tablice rejestracyjne

Po przepuszczeniu obrazu przez klasyfikator, zwraca on tablicę prostokątów, które określają położenie wykrytego obiektu na obrazie. Taki prostokąt jest reprezentowany jako współrzędne górnego lewego boku oraz jego szerokość i wysokość.

Algorytmy rozpoznawania twarzy dostępne w OpenCV:

- FisherFace
- EigenFace
- LBPHFace

Po wytrenowaniu modelu, podajemy mu na wejście obraz twarzy do rozpoznania. Model zwraca dwie wartości, tj. label i confidence. Label określa numer twarzy z załadowanej bazy, a confidence pewność dopasowania. Im wartość pewności jest mniejsza tym dopasowanie pewniejsze. Pewność dąży do zera.

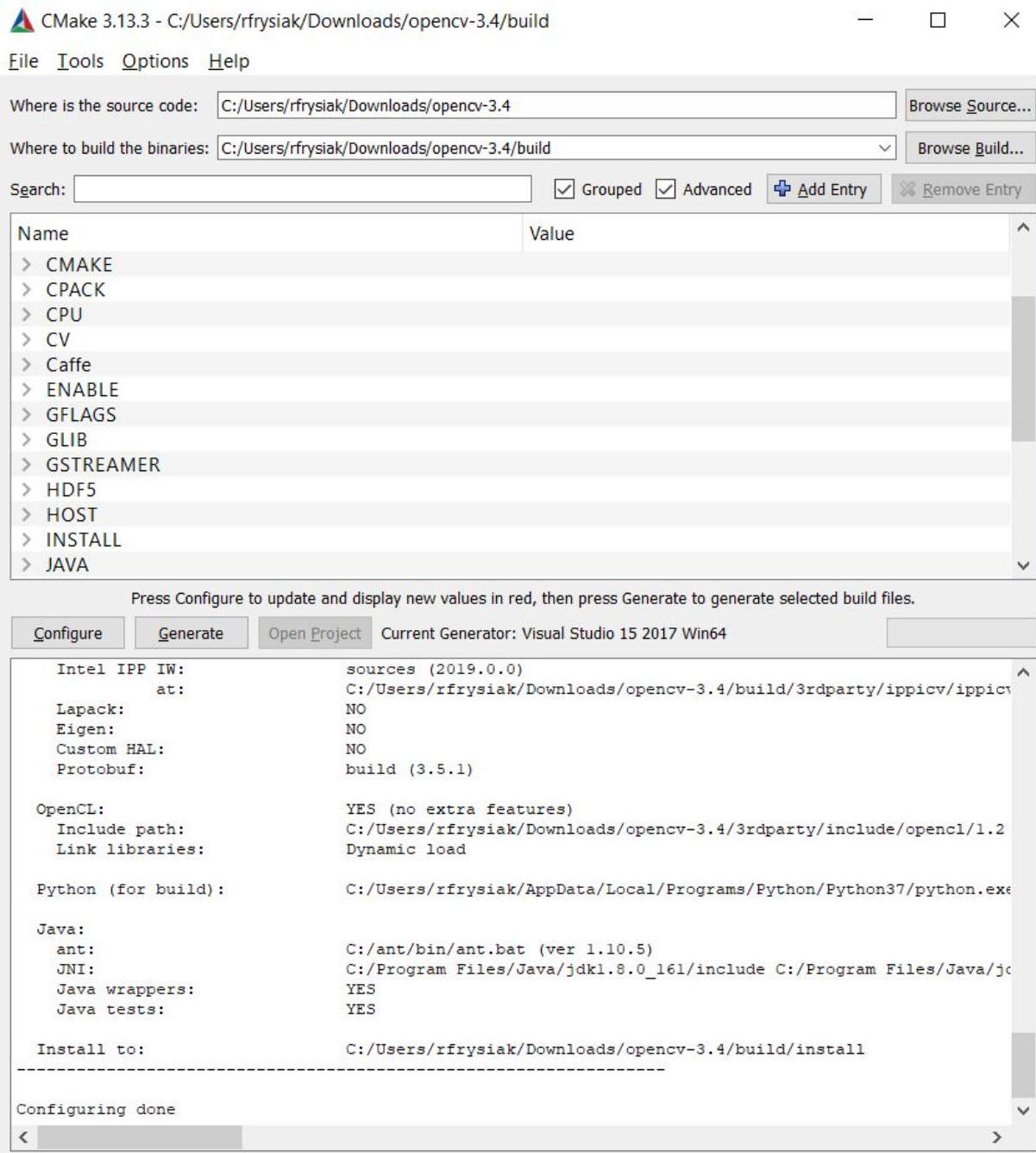
Interesujące problemy i ich rozwiązania

Podczas realizacji projektu okazało się, że biblioteka udostępniona na stronie projektu opencv.org nie zawiera modułu do rozpoznawania twarzy. Moduł ten o nazwie **face** jest rozwijany w osobnym repozytorium projektu **opencv_contrib** wraz z innymi dodatkowymi modułami, które nie są skompilowane w podstawowej bibliotece. Jedynym rozwiązaniem było ręczne przekompilowanie całej biblioteki.

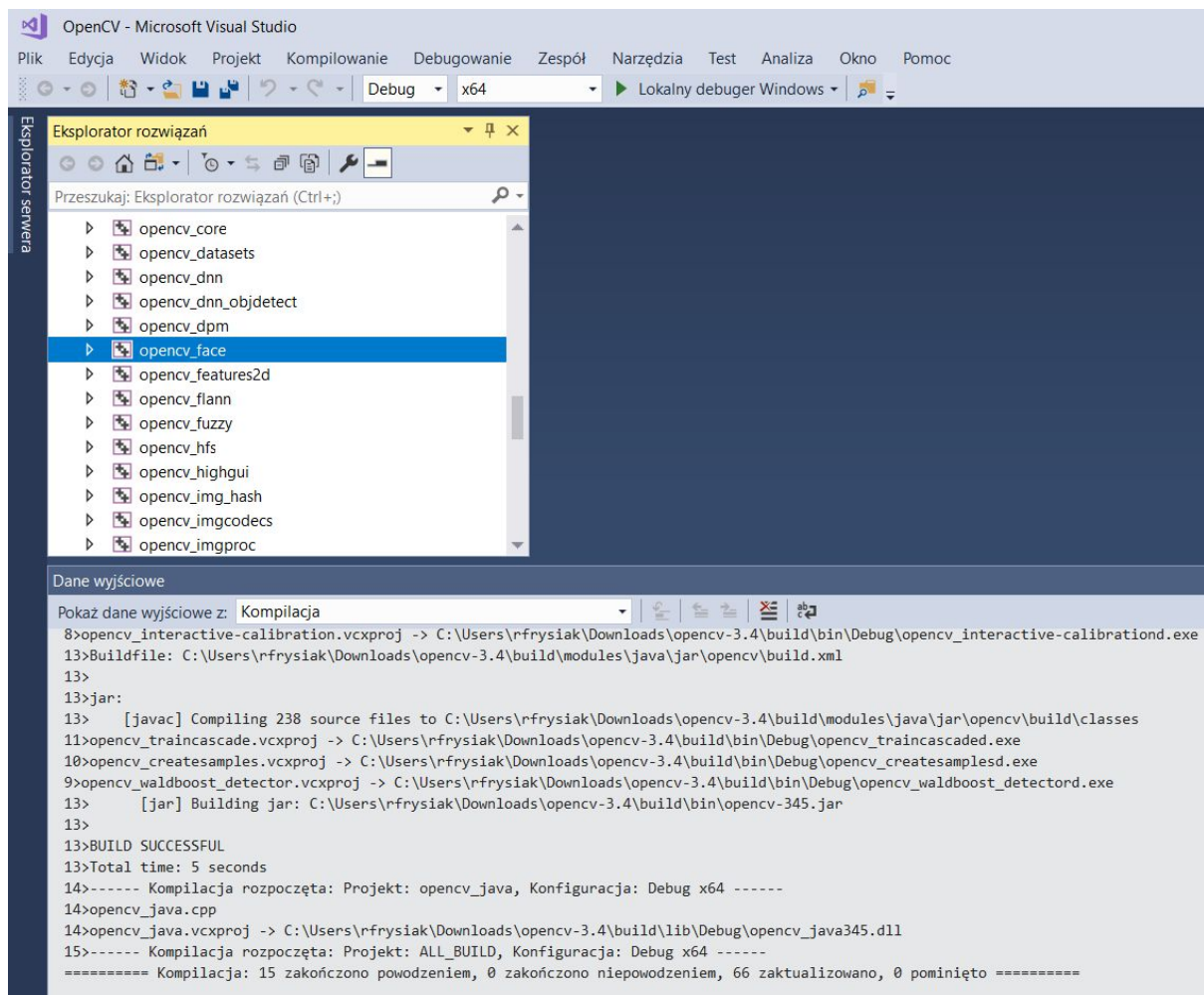
Pierwszym krokiem podczas kompilacji była konfiguracja ściągniętych źródeł przy pomocy programu CMake, co obejmowało:

- wskazanie kompilatora - do kompilacji użyliśmy programu Visual Studio 2017
- wskazanie ścieżki do środowiska Java - użyliśmy środowiska Oracle Java JDK 1.8
- instalacja Apache Ant - narzędzia służącego do zautomatyzowania procesu budowy oprogramowania

W kolejnym kroku wygenerowaliśmy pliki dla projektu, aby było można je wczytać do Visual Studio 2017 i przystąpić do kompilacji.



Ostatnim krokiem była kompilacja biblioteki w programie Visual Studio 2017.



Klasyfikator zaznaczający obszar twarzy bardzo go ogranicza, do tego stopnia, że w kwadracie nie mieści się wierzchołek głowy i dolna część brody. Powodowało to gorsze rozpoznawanie twarzy. Aby ten efekt zminimalizować, kwadrat zwrócony przez klasyfikator powiększamy o 10% i taki większy wycinek podajemy do modelu rozpoznawania. W ten sposób dokładność rozpoznawania się poprawiła.

W bazie zdjęcia przechowywane są w rozmiarze 92x112 pixeli. Model rozpoznawania FisherFaceRecognizer wymaga, aby rozpoznawany obraz był tej samej wielkości co obrazy w bazie. Klasyfikator zaznacza twarz jako kwadrat o różnej wielkości w zależności od tego jak daleko znajduje się obiekt od kamery. Wymaga to dodatkowego przeskalowania wykrytej twarzy. Inne modele nie mają takiego ograniczenia, ale zauważyliśmy, że przeskalowanie obrazu do rozmiaru z bazy poprawia również jakość rozpoznawania dla reszty modeli.

Instrukcja użytkowania aplikacji

Użytkownik po uruchomieniu programu ujrzy główne okno programu z trzema dostępnymi przyciskami :

- Start kamery
- Dodaj folder
- Dodaj zdjęcie

Przed uruchomieniem można dodać nowy folder, a następnie umieścić w nim zdjęcia wykonane za pomocą przycisku “Dodaj zdjęcie”.

Po poprawnym zapisie użytkownik wciska przycisk Start kamery i program prezentuje przed nim swoją pełną funkcjonalność.