Dawid Michalak

Nr 124027

Imformatyka sem VII

Face Recognition – Project

Dokumentacja projektu

Spis treści

[1. Opis projektu 3](#_Toc504062037)

[2. Podział prac 3](#_Toc504062038)

[3. Funkcjonalności programu 3](#_Toc504062039)

[4. Wybrane technologie 4](#_Toc504062040)

[OpenCV 4](#_Toc504062041)

[EmguCV 5](#_Toc504062042)

[5. Architektura rozwiązania 5](#_Toc504062043)

[Architektura OpenCV 5](#_Toc504062044)

[FrameGrabber 6](#_Toc504062045)

[Gray Convert 6](#_Toc504062046)

[Klasyfikator Haar 6](#_Toc504062047)

[Klasyfikator Haar – opis ogólny 7](#_Toc504062048)

[FaceDetection i Face Recongition 8](#_Toc504062049)

[PersonDataBase 8](#_Toc504062050)

[6. Instrukcja obsługi programu 9](#_Toc504062051)

[Pierwsze uruchomienie 9](#_Toc504062052)

[Główne okno 9](#_Toc504062053)

[Strumień z kamery 10](#_Toc504062054)

[Baza twarzy 10](#_Toc504062055)

[Rozpoznane twarze - informacja 12](#_Toc504062056)

[Przeglądarka bazy twarzy 12](#_Toc504062057)

[Rozpoznawanie twarzy ze zdjęć 13](#_Toc504062058)

[7. Usprawnienia i pomysły na przyszłość 15](#_Toc504062059)

# Opis projektu

Oprogramowanie służy do detekcji i rozpoznawania twarzy. Użytkownik zapisuje swoją twarz do bazy programu po czym aplikacja ma za zadanie przy wykryciu i rozpoznaniu twarzy wyświetlić imię użytkownika.

Zdecydowałem wybór tego tematu ponieważ mnie on interesuje i myślę że w dzisiejszych czasach jest duże zainteresowanie i jest to mocno rozwijana technologia. Śledzenie i szpiegowanie ludzi stało się popularne. Zapoznanie się z algorytmami i strukturą oprogramowania do rozpoznawania twarzy pozwoli zrozumieć chociaż po części jak to działa jakie są możliwości i ograniczenia - co może być interesujące.

# Podział prac

|  |  |
| --- | --- |
| Praca | Osoba realizująca |
| Wybór bibliotek  Języka oprogramowania | Grupa jednoosobowa - Dawid Michalak |
| Stworzenie wyglądu aplikacji |
| Kod:  - obsługa kamery  - detekcja twarzy  - stworzenie bazy twarzy – dodawanie  - rozpoznawanie twarzy  - biblioteka twarz  - ilość wykrytych twarzy  - rozpoznawanie twarzy w czasie rzeczywistym  - rozpoznawanie twarzy i oczu ze zdjęć |
| Testy |
| Naprawa błędów |
| Optymalizacja |

# Funkcjonalności programu

* Obsługa kamery

Program obsługuje kamerę, z której czerpie źródło do dalszej obróbki

* Detekcja wielu twarzy naraz

Analiza obrazu wejściowego w poszukiwaniu twarzy. Zaznacza rozpoznane twarze na obrazie rysując kwadrat

* Opcja wykrywania oczu z kamery
* Stworzenie bazy twarzy – dodawanie twarzy

Użytkownik po detekcji twarzy może ją zapisać do bazy programu.

* Rozpoznawanie twarzy

Aplikacja przy następnej detekcji twarzy sprawdza bazę, jeżeli osoba jest w bazie to wyświetli imię danego użytkownika nad narysowanym kwadratem

* Biblioteka twarz

Funkcjonalność która pozwala przejrzeć zapisane osoby w bazie wraz ze zdjęciem i imieniem danej osoby

* Ilość wykrytych twarzy

Program pozwala wykrywać wiele twarzy jednocześnie w czasie rzeczywistym. Wyświetla ilość rozpoznanych twarzy wraz z liczbą ogólną wykrytych twarzy

* Rozpoznawanie twarzy w czasie rzeczywistym

Pobieranie z kamery obrazu i obróbka w czasie rzeczywistym

* Rozpoznawanie twarzy i oczu ze zdjęć

Użytkownik może wprowadzić zdjęcie po czym program rozpocznie analizę i zaznaczy wykryte twarze

# Wybrane technologie

## OpenCV

(Open Source Computer Vision library) jest bezpłatną, open-source’ową biblioteką do użytku komercyjnego i edukacyjnego. Została napisana w języku C i C++ i rozwijana jest za pomocą wrapperów w wielu językach programowania takich jak C#, Python, Ruby, Matlab czy Java. Posiada wielu użytkowników na całym.

Biblioteka ta została stworzona na potrzeby aplikacji czasu rzeczywistego gdzie wydajność obliczeniowa programów jest bardzo istotna. Napisana jest w zoptymalizowanym języku C/C++ i wykorzystuje możliwości jakie dają popularne od kilku lat procesory wielordzeniowe.

Jednym z głównych celów biblioteki OpenCV jest dostarczenie narzędzia, które pozwoli tworzyć zarówno proste programy jak również zaawansowane projekty ludziom z różnym poziomem wiedzy na temat programowania i przetwarzania obrazów. Amatorzy za pomocą OpenCV i kamery internetowej mogą poznawać elementy przetwarzania obrazów w zakresie: wykrywania krawędzi, segmentacji czy filtracji obrazów bez większego wysiłku. Zaś sami specjaliści nie muszą poświęca czasu na pisanie bazowych funkcji w zaawansowanych w projektach – nigdy więcej wymyślania koła! Możliwe jest to dzięki wbudowanym w bibliotekę ponad 500 funkcjom, które obejmują wiele obszarów komputerowej wizji takich jak robotyka, stereowizja, bezpieczeństwo, analiza obrazów medycznych czy kalibracja kamery. Wiele projektów wykorzystujących zdjęcia z lotu ptaka czy mapowanie ulic takie jak – „Google Maps”, czy „Google Street’s View”, wykorzystuje kalibracje kamery oraz metody „zszywania” obrazów, które są częścią OpenCV. Bezzałogowe samoloty, urządzenia do detekcji obiektów czy układy monitoringu również oparte są o wizje komputerową.

OpenCV – Powodem wybrania tej biblioteki był taki że jest ona bardzo rozbudowana, prosta i bardzo popuarna. Możliwości biblioteki są ogromne, począwszy od prostych operacji na pojedynczych pikselach przez zaawansowane algorytmy przetwarzania obrazów, po algorytmy uczenia maszynowego, wykorzystywane w takich zagadnieniach jak detekcja twarzy. Niewątpliwą zaletą jest możliwość wykonywania wszystkich operacji “w locie”, bezpośrednio na strumieniu wideo.

## EmguCV

EmguCV jest .Netowym wrapperem do OpenCV - potężnej biblioteki do przetwarzania i analizy obrazów (zarówno statycznych jak i ruchomych) oraz innych sygnałów cyfrowych.

Wykorzystałem Emgu ponieważ jest to mój pierwszy projekt z wykorzystaniem C#. Chciałem się zapoznać z językiem dlatego też wybrałem Emgu – Wrapper OpenCV wspierający ten język.

Ważną cechą biblioteki jest fakt iż stara się ona wykorzystać jak najlepiej zasoby dostępne na maszynie na której jest ona uruchomiona, ze szczególnym uwzględnieniem karty grafiki.

# Architektura rozwiązania

## Architektura OpenCV

* OpenCV składa się z pięciu podstawowych komponentów:

CV oraz CVaux – komponenty zawierające funkcję transformacji, filtracji oraz konwersji przestrzeni obrazów, funkcję analizy obrazów takie jak selekcja, operacje morfologiczne, detekcję krawędzi oraz obsługę histogramów, detekcję obiektów, kalibrację kamery, rekonstrukcję sceny 3D i inne,

* MLL – Machine Learning Library, jak sama nazwa wskazuje zawiera funkcje tworzenia klasyfikatorów bazujących na statystyce odgrywających znaczącą rolę w uczeniu maszyn sposobu detekcji,
* HighGUI – zawiera metody akwizycji i zwalniania obrazów, sekwencji wideo, narzędzia tworzenia interfejsu okienkowego, suwaków, obsługi myszy etc.
* CxCore – Podstawowy komponent biblioteki zawiera operacje na tablicach, algebrę macierzy, funkcje matematyczne, transformacje Fouriera, wsparcie dla plików XML, narzędzia rysowania obiektów 2D i inne.

Funkcje OpenCV, które pozwalają na współdziałanie z systemem operacyjnym, systemem plików, czy elementami sprzętowymi takimi jak kamera, są zebrane w bibliotece HighGUI (High-Level Graphical User Interface). Biblioteka ta pozwala na tworzenie okien, oraz wyświetlanie na nich obrazów, wczytanych dzięki możliwościom HighGUI w zakresie obsługi plików graficznych (obrazów i sekwencji wideo). Wspomaga również obsługę myszy, którą można wykorzystywać do obsługi przełączników oraz suwaków.

Tworzenie interfejsu użytkownika możemy podzieli na trzy części: sprzętową, obsługi plików oraz interfejsu użytkownika.

Pierwszą z nich jest komunikacja z urządzeniami takimi jak kamera. Biblioteka HighGUI dostarcza narzędzi, które w prosty sposób pozwalają na komunikację z kamerą, co zazwyczaj było zmorą programistów wykorzystujących obrazy z kamery.

Drugą część stanowi obsługa plików graficznych. Skupia się ona głównie na załadowaniu oraz zapisaniu obrazów. Udostępnione narzędzia pozwalają na pracę zarówno z obrazami jak i sekwencjami wideo. Wykorzystywane funkcje wspierają wiele formatów graficznych wraz z ich kodowaniem i dekodowaniem, gdy tego wymagają.

Ostatnia część skupia się na właściwym interfejsie użytkownika (GUI). Wbudowane funkcję pozwalają na otwieranie zdjęć w stworzonych przez nie oknach. Ułatwiają modyfikację parametrów obrazów za pomocą funkcji tworzących suwaki, przełączniki oraz wspomagających obsługę myszy.

## FrameGrabber

Ładowanie plików wideo do OpenCV odbywa się za pomocą funkcji Capture(). Funkcja ta ma za zadanie uruchomić kamerę i przygotować stream. Poniżej deklaracja funkcji:

//Inicjalizacja obsługi kamery

grabber = new Capture();

grabber.QueryFrame();

Application.Idle += new EventHandler(FrameGrabber);

Przy poprawnym załadowaniu funkcja grabber.QueryFrame() odczyta klatki z kamery.

## Gray Convert

Podstawowy tryb koloru to skala szarości (ang. grey scale). Zawiera 256 tonów koloru, zaczynając od 0 – koloru czarnego, przez 254 odcienie szarości i koocząc na 256 – kolorze czarnym. Zawiera 8-bitową głębie kolorów. Obrazy w trybie skali szarości nazywane są monochromatycznymi. Konwersją obrazu w skale szarości, odpowiada funkcja:

currentFrame.Convert();

## Klasyfikator Haar

Na początku program ładuje gotowy klasyfikator Haar’a do wykrywania frontowej twarzy, który umieszczony jest w pliku XML (haarcascade\_frontalface\_default.xml).

Następnie uruchamia się funkcja wykrywająca twarz na podstawie klasyfikatorów Haar

DetectHaarCascade();

### Klasyfikator Haar – opis ogólny

Wykrywanie obiektów na obrazie, należy do zadań, dla których ciężko jest stworzyć algorytm. Problem ten jest bardzo złożony. Dlatego też do jego rozwiązania stosuje się elementy inteligencji obliczeniowej. Dziedzina ta zajmuje się między innymi uczeniem maszynowym, czyli grupą algorytmów które uczy się pewnych zadań na podstawie wzorców.

W obecnym czasie do rozpoznawania obiektów, szeroko-stosowane są dwa rozwiązania. Pierwsze z nich to **eigenface**, a drugie to **haar**. Biblioteka korzysta z tego drugiego rozwiązania. Metoda ta została opracowana przez Paula Viol’a. Opiera się ona na wirtualnym oknie, które przesuwa się po obrazie, w celu znalezienie pasujących regionów.

#### Trening klasyfikatora

Biblioteka udostępnia szereg wytrenowanych już klasyfikatorów. Zapisywane są one w formacie XML. W ostatniej wersji biblioteki można między innymi:

* haarcascade eye tree eyeglasses.xml
* haarcascade eye.xml
* haarcascade frontalface alt2.xml
* haarcascade frontalface alt tree.xml
* haarcascade frontalface alt.xml
* haarcascade mcs mouth.xml
* haarcascade mcs nose.xml

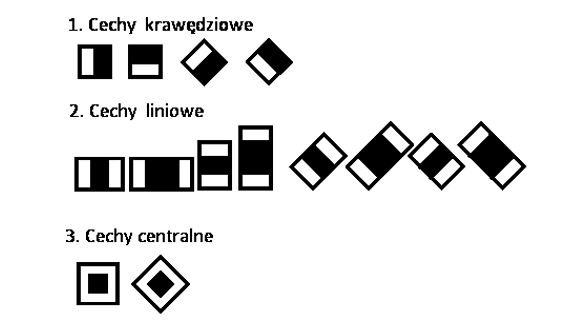
Widzimy tu klasyfikatory wytrenowane w wyszukiwaniu twarzy (frontalnie i z profilu) oraz jej części składowych takich jak uszy oczy itp.

Algorytm funkcji „Haar-like features” opiera się na trzech podstawowych fazach:

Wybór zestawu cech

W metodzie Viola-Jenes cechy traktowane były jako kombinacja dwóch, trzech lub czterech prostokątów. Ilość możliwych kombinacji połączenia prostokątów była bardzo duża, dlatego przy określonych założeniach została wyselekcjonowana grupa.

Następnie udoskonalony zestaw cech jeszcze lepiej identyfikujący poszukiwane obiekty został zaproponowany w metodzie LienhartMaydt. Grupa ta została podzielona na cechy krawędziowe, liniowe oraz centralnie otoczone. Właśnie z tego zestawu cech korzysta funkcja z OpenCV.



Zasada działania cech polega na odnajdywaniu obszarów, na których różnica pomiędzy sumą pikseli regionów ograniczonych czarnymi i białymi prostokątami znajduję się powyżej pewnego progu. Na podstawie tej wartości algorytm może zidentyfikować czy badany obszar jest poszukiwanym obiektem czy jego tłem. Dla przykładu, prostokątne regiony oczu mają dużo mniejszą intensywność niż obszar czoła, ten zaś dużo większą od obszaru włosów.

## FaceDetection i Face Recongition

Program wykorzystując klasyfikatory wykrywa twarz i rysuję na niej kwadrat. Zapisując twarz aplikacja tworzy zdjęcie służące później do porównania z obrazem z kamery w czasie rzeczywistym wykorzystując funkcje:

EigenObjectRecognizer();

Jako parametry funkcja przyjmuje wytrenowane zdjęcia osób oraz aktualną ramkę z kamery. Gdy osoba jest w bazie nad kwadratem dodawany jest podpis.

## PersonDataBase

Proces dodawania twarzy do bazy:

Przechwycony kwadrat z twarzą podlega konwersji do skali monochromatycznej i przeskalowaniu do 100x100px. Gotowy obraz zapisywany jest w folderze TrainedFace i dodawany jest label z podpisem do pliku TrainedLabels.txt w poniższym formacie :

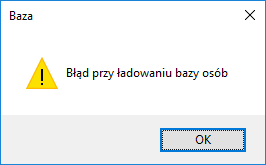
LiczbaZapisanychTwarzy%Imie1%Imie2%...%...

Pierwsza liczba pomaga w listowaniu twarzy a następnie są znaki odizolowania „%”.

# Instrukcja obsługi programu

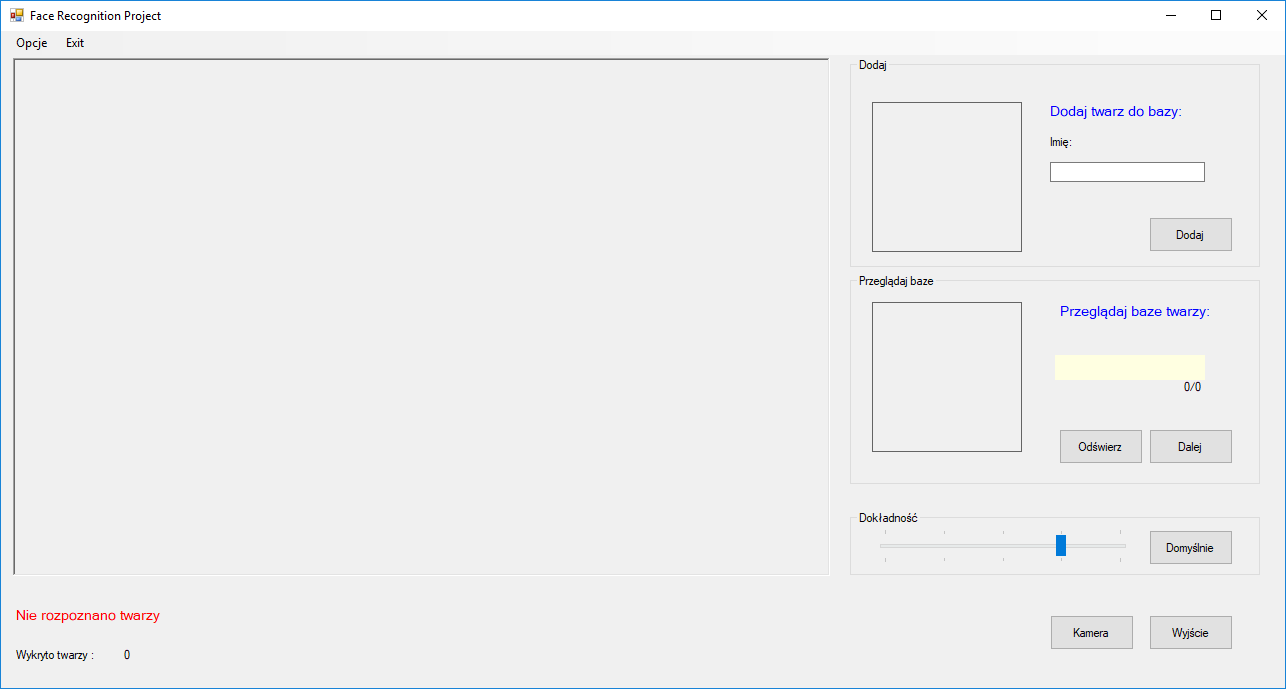
## Pierwsze uruchomienie

Jeżeli żadna osoba nie jest zapisana w bazie programu tj. zdjęcie rozpoznanej osoby i imię to program przed uruchomieniem zawiadomi użytkownika odpowiednim komunikatem



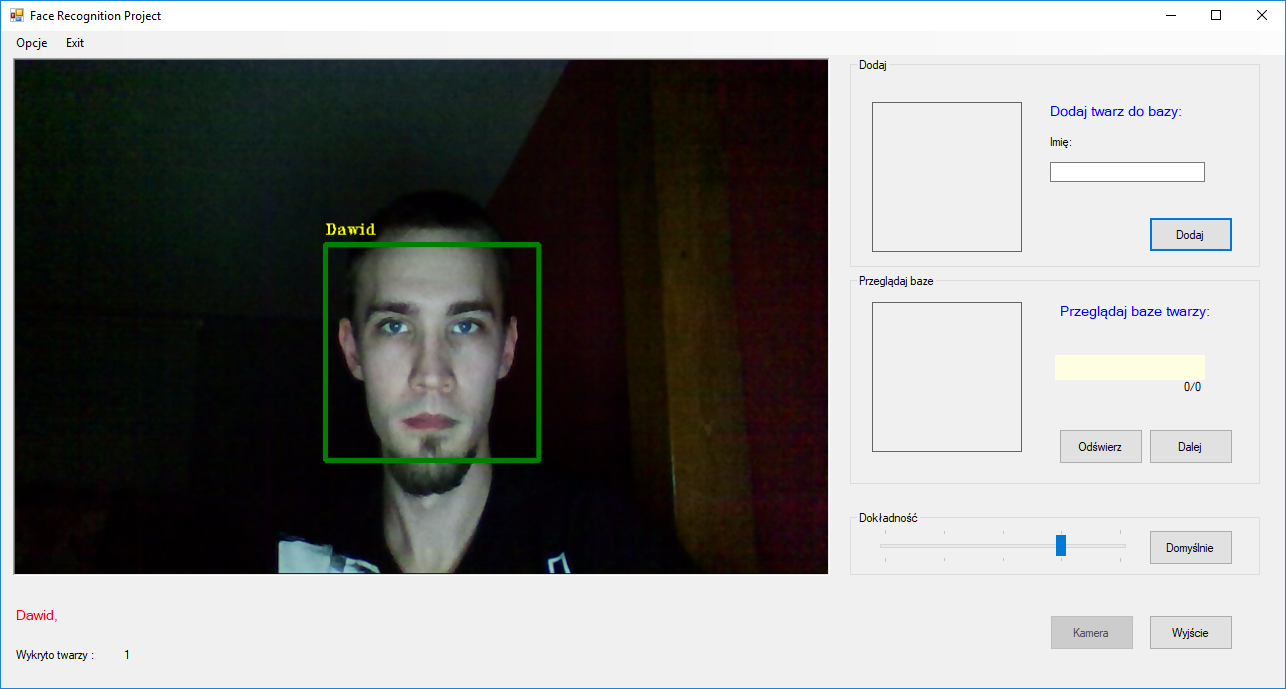
## Główne okno

Po uruchomieniu aplikacji otworzy się główne okno programu na którym możemy m.in. uruchomić kamerę:



## Strumień z kamery

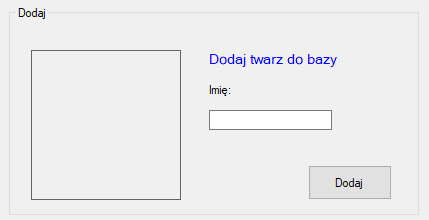
Po uruchomieniu kamery rozpocznie się automatyczne rozpoznawanie obrazu ze strumienia kamery w czasie rzeczywistym.



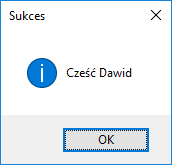
Jak widać na zdjęciu program odpowiednio zaznaczył twarz odpowiednim zielonym kwadratem.

## Baza twarzy

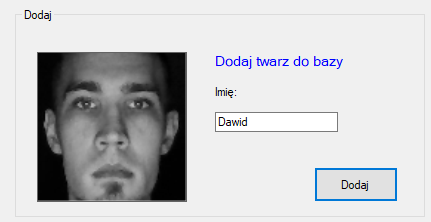
Następnie rozpoznaną twarz możemy zapisać do bazy programu wybierając przycisk Dodaj w prawym górnym rogu. Przed wciśnięciem przycisku powinniśmy wpisać do TextBox’a swoje imię w celu dalszego wykorzystania.



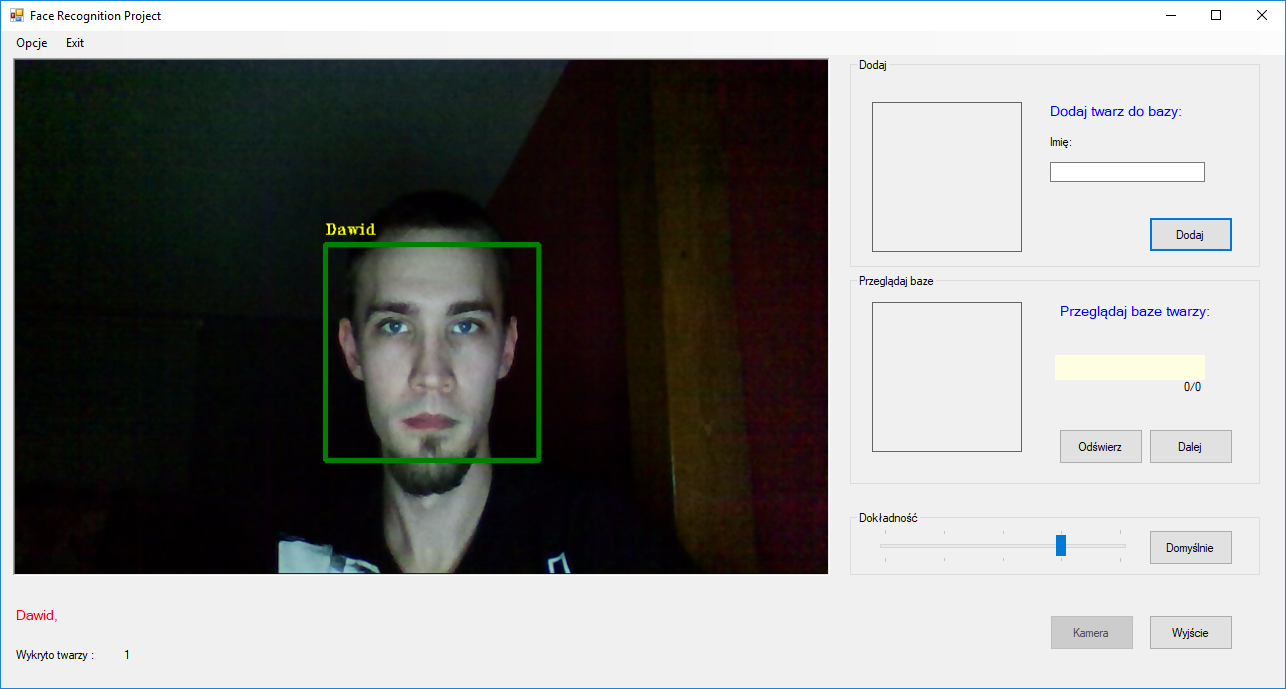
Twarz zostanie zapisana do bazy programu informując użytkownika stosownym komunikatem:



A oto efekt



Gdy twarz istnieje w bazie to program automatycznie rozpozna twarz ze strumienia z kamery i doda do kwadratu stosowny podpis.



## Rozpoznane twarze - informacja

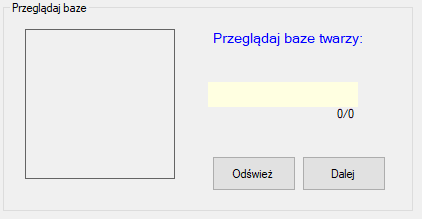
W dolnej części okna aplikacja informuje o ilości twarzy wykrytej na obrazie oraz imionach rozpoznanych twarzy.



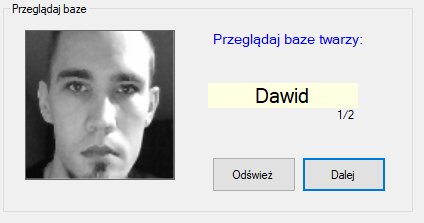
## Przeglądarka bazy twarzy

Gdy już mamy rozbudowaną bazę osób możemy ją przeglądać wciskając przycisk Dalej.

Możemy podejrzeć jakie mamy zdjęcia / próbki w bazie. Przy większej ilości zdjęć i dużej ilości próbek danej osoby ta opcja nie ma większego sensu ale przy procesie tworzenia aplikacji pomogła ona lepiej zrozumieć algorytm.



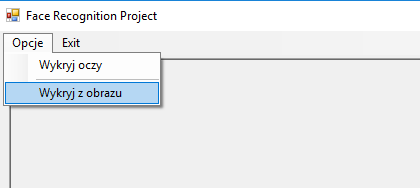
Przycisk Odśwież służy do odświeżenia bazy.



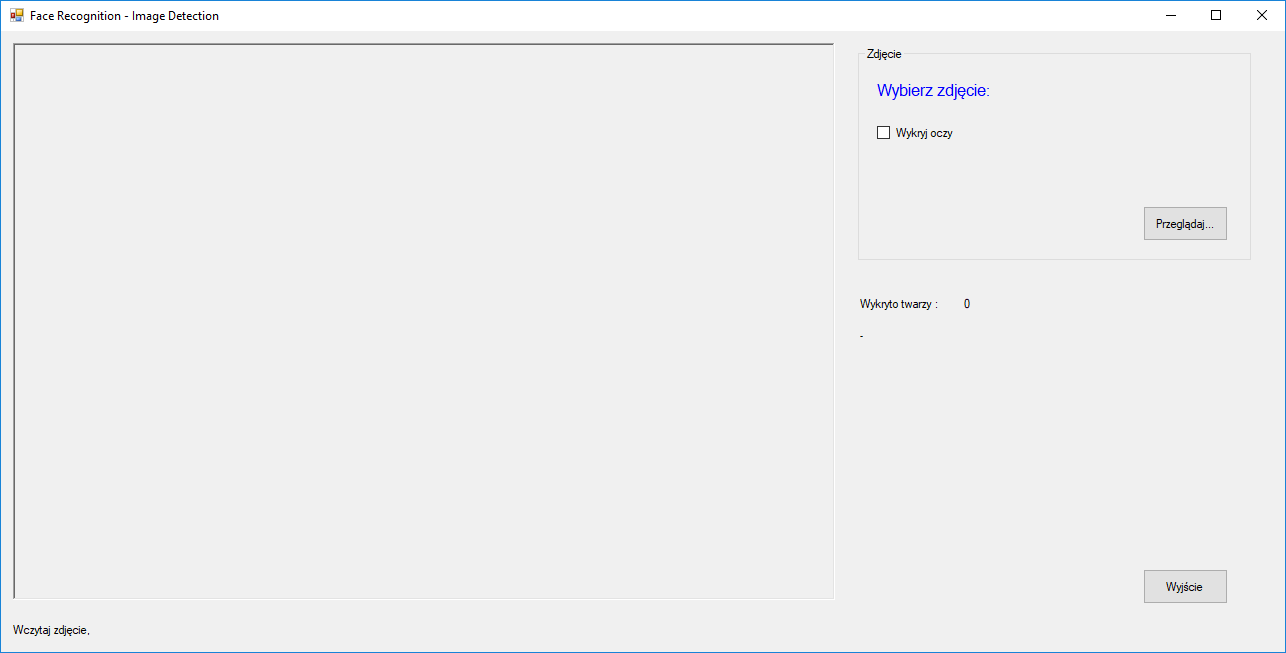
Pojawia się informacja dotycząca ilości próbek w bazie programu.

## Rozpoznawanie twarzy ze zdjęć

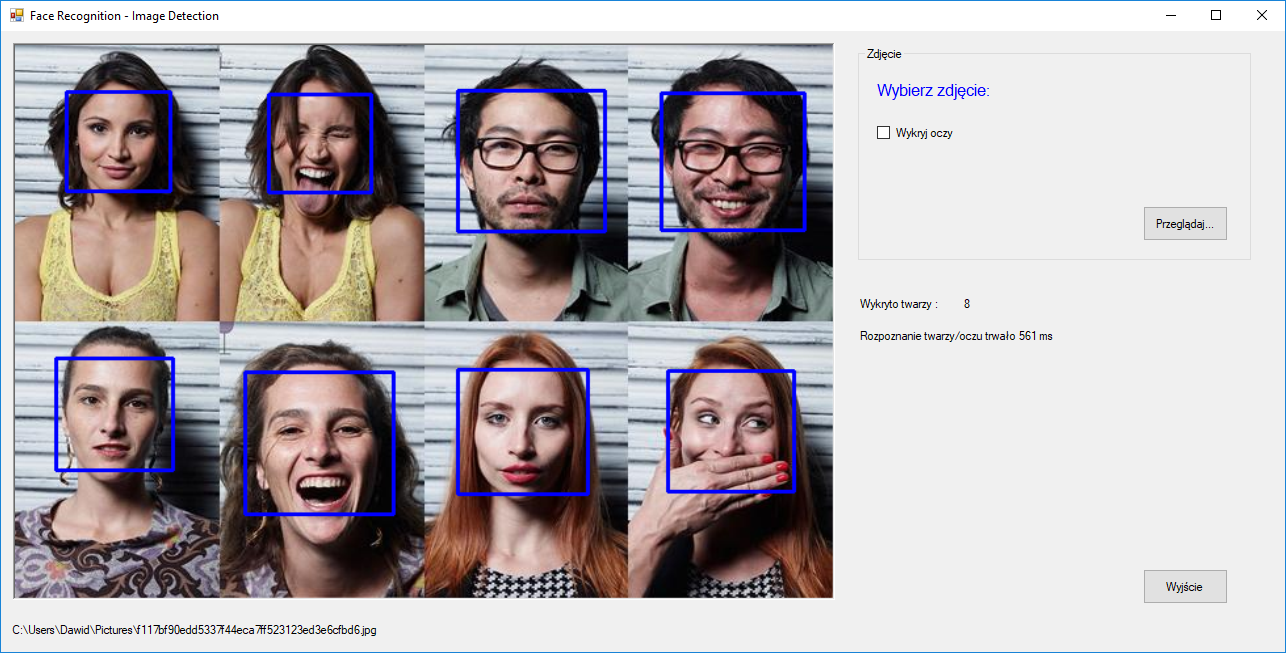
Wciskając opcje „Wykryj z obrazka” w menu „Opcje” możemy uruchomić algorytm detekcji twarzy na zdjęciach:



Otworzy nam się drugie okno:

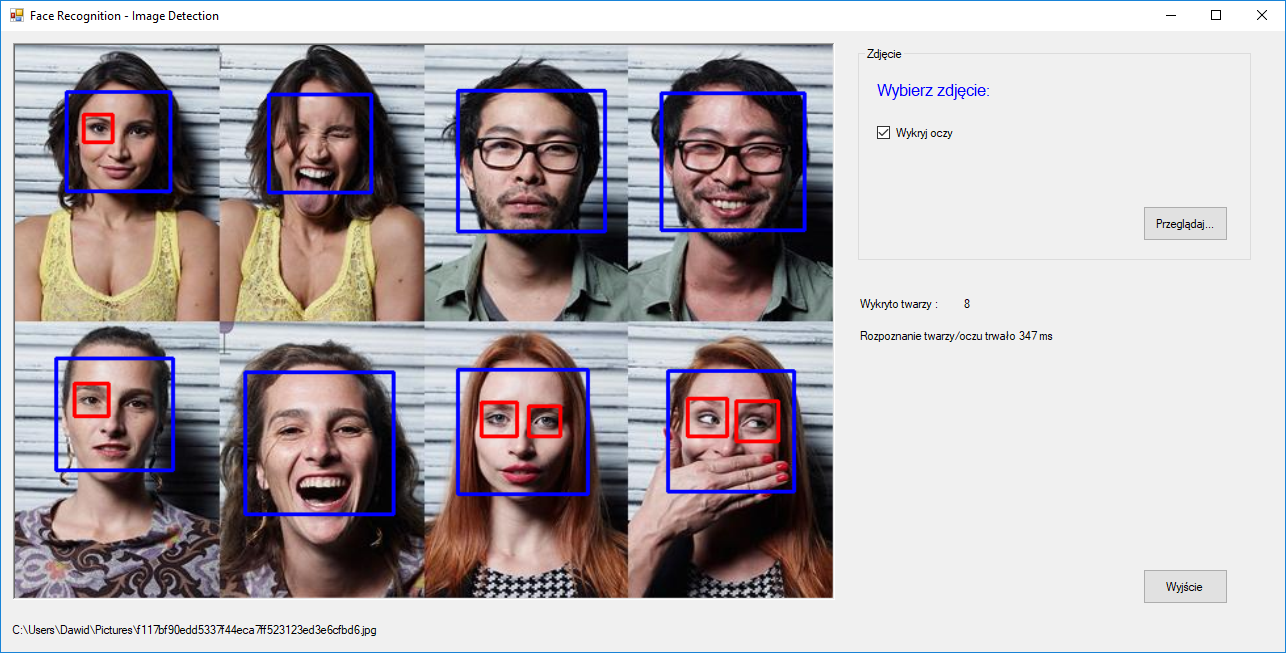


Wciskamy przeglądaj w celu odnalezienia zdjęcia z wewnętrznego źródła:



Program wykryje twarze i je zaznaczy na obrazie. Algorytm w tym oknie również zaznacza oczy. Aplikacja informuje ile wykryła twarzy oraz ile czasu to trwało. Pojawia się również ścieżka do wybranego zdjęcia.

Możemy również zaznaczyć opcje wykrywania oczu:



Jak widać algorytm nie radzi sobie gdy oczy są nie wyraźne lub słabo widoczne zza okularów.

Po zakończeniu pracy wciskamy przycisk wyjście po czym program zakańcza swoją pracę.

# Usprawnienia i pomysły na przyszłość

Program jest sprawny i funkcjonalny jednak zawsze można udoskonalić, oto pomysły na które brakło czasu aby zostały zrealizowane:

- zapisywanie próbek i labeli do pliku .xml

- zapisywanie ~10 próbek naraz

- wykorzystanie najnowszej wersji biblioteki. W projekcie została użyta wersja 2.4 ponieważ nowsze wersje są dużo bardziej rozbudowane mają więcej opcji i technologii przez co aplikacja wykorzystuje 100% zasobów z niewiadomej przyczyny. Dlatego postanowiłem rozpocząć przygodę z OpenCV od starszej wersji biblioteki, która wspierała by wszystkie moje wymagania.

- śledzenie obiektów / rozpoznawanie przedmiotów.