Podstawy Teleinformatyki

SFR – Student Face Recognition

AUTORZY:

Mikołaj Drygas 116216

Patryk Dzwoniarski 121997

Krzysztof Figiel 123652

Spis treści

[**1. Wstęp** 3](#_Toc477792333)

[1.1. Dlaczego wybraliśmy ten temat? 3](#_Toc477792334)

[**2. Cel i zakres pracy** 3](#_Toc477792335)

[2.1. Zadania SFR – Student Face Recognition 3](#_Toc477792336)

[2.2. Podział prac 3](#_Toc477792337)

[3. konstruowanie systemu 4](#_Toc477792338)

[3.1. środki implementacji 4](#_Toc477792339)

[3.2. Środowisko 4](#_Toc477792340)

[3.3. Narzędzia 4](#_Toc477792341)

# **1. Wstęp**

Celem naszego projektu jest stworzenie aplikacji do rozpoznawania twarzy osób wchodzących do pomieszczenia, którym może być np. laboratorium. Aplikacja przyda się do selekcji osób wchodzących na salę przed ważnym egzaminem, bądź zwyczajnie, do rozpoznawania studentów przed wejściem do sali laboratoryjnej i wykrywania osób spoza roku.

## 1.1. Dlaczego wybraliśmy ten temat?

Głównym powodem wybrania tego tematu była chęć zapoznania się z biblioteką *OpenCV* i bibliotekami podobnymi, które są dziś powszechnie używane w wielu przydatnych aplikacjach i systemach informatycznych.

Dodatkowo chcieliśmy podnieść swoje umiejętności w pisaniu aplikacji w technologii *C# WPF*, z którą mieliśmy okazję zapoznać się w trakcie pisania różnych projektów w czasie studiów. Uważamy, że jest ona dobrym rozwiązaniem, głównie ze względu na prostotę tworzenia interfejsu graficznego programu i jego szybkiej edycji.

# **2. Cel i zakres pracy**

## 2.1. Zadania SFR – Student Face Recognition

* Wykrywanie twarzy osoby, bądź wielu osób wchodzących do sali, stojących naprzeciwko kamerki IP
* Możliwość zapisania rozpoznanej twarzy osoby do lokalnej bazy danych poprzez podanie jej imienia, nazwiska i numeru indeksu
* Wyświetlenie imienia, nazwiska i numeru indeksu osoby w przypadku rozpoznania jej twarzy
* W przypadku nierozpoznania osoby wchodzącej do sali – wyświetlenie alertu ostrzegawczego

## 2.2. Podział prac

Aplikacja tworzona będzie w trakcie wspólnych spotkań grupowych, a podział zadań będzie ustalany w momencie tworzenia projektu.

|  |  |
| --- | --- |
| Osoba | Zadania |
| Patryk Dzwoniarski | Aplikacja |
| Mikołaj Drygas | Aplikacja |
| Krzysztof Figiel | Aplikacja |

# 3. konstruowanie systemu

## 3.1. środki implementacji

* *C#, WPF* – technologia ta umożliwia wygodną pracę nad aplikacją z interfejsem graficznym
* *Emgu CV* – jest to .*Net’owy* wrapper do *OpenCV*, służący do przetwarzania i analizy obrazów (statycznych i ruchomych). Biblioteka ta zawiera ogrom funkcji służących do pracy z CPU oraz GPU. Wrapper ten może być kompilowany do takich systemów jak *Linux*, czy *Mac OS X*. Największą zaletą tej biblioteki jest to, że stara się ona jak najlepiej wykorzystać zasoby, które oferuje nam nasza maszyna, a głównie jej karta graficzna.

## 3.2. Środowisko

* *Microsoft Visual Studio 2015 Enterprise* – rozbudowane środowisko stworzone przez firmę *Microsoft* idealne do stworzenia aplikacji typu *WPF*.

## 3.3. Narzędzia

* *GitHub –* system kontroli wersji, który umożliwi nam dostęp do kodu aplikacji z każdego komputera, na którym akurat pracujemy. Odnotowanie wszystkich zmian z notatką przy zapisywaniu postępów pozwala na łatwe analizowanie historii kodu.