

**Politechnika Poznańska**

Wydział Elektryczny

Instytut Automatyki, Robotyki i Inżynierii  
Informatycznej

**Projekt z Podstaw Teleinformatyki**

Rozpoznawanie twarzy i śledzenie ruchu

Jakub Duński 121903

[jakub.dunski@student.put.poznan.pl](mailto:jakub.dunski@student.put.poznan.pl)

Paweł Korczak 116283

[pawel.korczak@student.put.poznan.pl](mailto:pawel.korczak@student.put.poznan.pl)

Remigiusz Wróblewski 125903

[remigiusz.wroblewski@student.put.poznan.pl](mailto:remigiusz.wroblewski@student.put.poznan.pl)

Marian Wojtkowiak 116459

[marian.wojtkowiak@student.put.poznan.pl](mailto:marian.wojtkowiak@student.put.poznan.pl)

19.03.2019

## Spis treści

1. Charakterystyka ogólna	2
2. Uzasadnienie wyboru tematu	4
3. Podział pracy	5
4. Wymagania	6
4.1 Funkcjonalne	6
4.2 Niefunkcjonalne	7
5. Narzędzia, środowisko, biblioteki	8
5.1 Narzędzia	8
5.2 Technologie i materiały powiązane	8
5.3 Środowisko	9
6. Przypadki użycia	10
7. Schemat bazy danych	15
8. Architektura systemu	16
9. Napotkane problemy	17
10. Instrukcja użytkowania	18
11. Dalszy rozwój	23
12. Bibliografia	26

# 1. Charakterystyka ogólna

System aktywnego monitoringu. Program pracuje na nagraniu (domyślnie pochodzącym z kamery) i rozpoznaje twarze wchodzących do pomieszczenia. Wykorzystywana jest biblioteka OpenCV, która dostarcza wiele funkcji przydatnych w pracy nad obrazem. Z pomocą odpowiedniego pakietu tych funkcji program wykrywa ruch w nagraniu.

W tym czasie sprawdzana jest możliwość pełnego zarejestrowania twarzy (z wykorzystaniem kolejnych funkcji OpenCV), a gdy stanie się to możliwe – zapisywany jest aktualny obraz. Następnie wykryta twarz porównywana jest z zapisanymi w bazie danych zdjęciami (na ich podstawie przeprowadzana jest próba rozpoznania). Dodatkową funkcją jest zliczanie osób wchodzących do pomieszczenia.

## 1.1 Zastosowane algorytmy

- **Wykrywanie twarzy:**

Klasyfikator kaskad Haara

OpenCV do wykrywania twarzy używa klasyfikatora kaskad Haara (Haar Cascade Classifier). Przeprowadza on na danym obrazku analizę każdej jego części i klasyfikuje ją jako zawierającą twarz bądź nie. Klasyfikator wyodrębnia twarze na podstawie danych zapisanych w pliku XML, w którym to znajdują się definicje poszukiwanych przez nas twarzy.

- **Rozpoznawanie twarzy:**

Metoda LBPH

Local Binary Pattern Histogram jest metodą pozwalającą na zmianę zdjęcia w macierz wartości całkowitych opisujących mało-wymiarowe cechy obrazu. Histogram tych wartości jest następnie wykorzystywany do dalszej analizy. LBPH nie bazuje na statystyce, więc nie jest potrzebna duża ilość zdjęć uczących.

- **Wykrywanie ruchu:**

MOG2

Jest to algorytm oparty na mieszaninie Gaussa i algorytmu segmentacji pierwszego planu. Ważną cechą tego algorytmu jest to, że wybiera odpowiednią liczbę rozkładów gaussowskich dla każdego piksela. Zapewnia lepszą adaptację do różnych scen z powodu zmian oświetlenia itp

## **2. Uzasadnienie wyboru tematu**

Temat zainteresował nas ze względu na przedmiot obieralny z trzeciego semestru. W ramach tego przedmiotu uczyliśmy się przetwarzania obrazów gdzie poznaliśmy bibliotekę OpenCV z języka Python. Zauważyliśmy ogrom funkcjonalności jaki daje ta biblioteka. Podczas zajęć nie udało nam się zgłębić pełni jej możliwości więc postanowiliśmy aby nasz projekt bazował na tej bibliotece.

### **3. Podział pracy**

#### **I. Paweł Korczak**

- Rozpoznawanie twarzy - wykrywanie osób spoza bazy danych
- Wykrywanie ruchu
- Interfejs graficzny

#### **II. Remigiusz Wróblewski**

- Algorytm wykrywania i rozpoznawania twarzy
- Połączenie aplikacji z bazą danych
- Połączenie aplikacji z kamerami IP

## **4. Wymagania**

Wymagania określają jakie funkcje ma spełniać system oraz sposób wykonywania określonych zadań. Wymagania można podzielić na:

### **4.1 Funkcjonalne**

- identyfikacja osób wchodzących do pomieszczenia obserwowanego przez kamerę IP
- wykrywanie twarzy
- rozpoznawanie twarzy
- administrator ma możliwość dodawania i usuwania osób z bazy danych
- administrator ma możliwość włączania i wyłączania kamer
- celem dodatkowym jest określenie ile osób aktualnie znajduje się w danym pomieszczeniu

## 4.2 Niefunkcjonalne

- aplikacja napisana w języku Python
- wykorzystanie biblioteki OpenCV do pracy na obrazie
- aplikacja wykrywa twarze w czasie rzeczywistym
- wykorzystanie SQLite do przechowywania danych o użytkownikach
- jako kamera IP wykorzystana aplikacja na telefony z systemem android "IP Webcam"
- możliwość zapisu wideo
- do obsługi programu wymagana jest mysz komputerowa oraz klawiatura



## 5. Narzędzia, środowisko, biblioteki

Rozdział przedstawia narzędzia i technologie, które zostały wykorzystane przy tworzeniu projektu.

### 5.1 Narzędzia

- GitHub – hostingowy serwis internetowy przeznaczony dla projektów programistycznych wykorzystujących system kontroli wersji Git.
- SQLite - wolnodostępny system zarządzania relacyjnymi bazami danych. Bazy SQLite przechowywane są w pojedynczych plikach, które łatwo archiwizować, przenosić czy badać, podglądając ich zawartość. Podstawowym narzędziem jest interpreter sqlite3 (sqlite3.exe w Windows).
- Qt Designer - aplikacja graficzna do definiowania graficznego interfejsu użytkownika (okien dialogowych itp.)

### 5.2 Technologie i materiały powiązane

- Biblioteka OpenCV – jest to biblioteka wykorzystywana podczas obróbki obrazu w czasie rzeczywistym. Biblioteka ta jest wieloplatformowa, można z niej korzystać w Mac OS X, Windows jak i Linux.
- PyQt5 - piąta wersja nakładki na bibliotekę Qt umożliwiająca tworzenie interfejsu graficznego dla programów komputerowych pisanych w języku Python.

## 5.3 Środowisko

PyCharm – zintegrowane środowisko programistyczne (IDE) dla języka programowania Python firmy JetBrains. Zapewnia m.in.: edycję i analizę kodu źródłowego oraz graficzny debugger.

## 6. Przypadki użycia

Przypadek użycia przedstawia interakcję pomiędzy aktorem (użytkownikiem systemu), który inicjuje zdarzenie oraz samym systemem jako sekwencję prostych kroków.

<b>Przypadek użycia:</b> Wykrywanie twarzy
<b>Aktorzy:</b> Użytkownik
<b>Scenariusz główny</b>
<ol style="list-style-type: none"><li>1. Użytkownik chce wykrywać twarze w czasie rzeczywistym</li><li>2. Użytkownik wybiera z której kamery chce wczytywać obraz</li><li>3. Użytkownik wykrywa twarze</li></ol>
<b>Rozszerzenia:</b>
2.A.Użytkownik wybiera pomiędzy kamerą wbudowaną w komputer a kamerą IP

<b>Przypadek użycia:</b> Rozpoznawanie twarzy przez kamerę wbudowaną
<b>Aktorzy:</b> Użytkownik
<b>Scenariusz główny</b>
<ol style="list-style-type: none"><li>1. Użytkownik chce rozpoznawać twarze w czasie rzeczywistym</li><li>2. Użytkownik wybiera opcję z menu z rozpoznawaniem twarzy</li><li>3. Użytkownik rozpoznaje twarze</li></ol>
<b>Rozszerzenia:</b>
2.A.Użytkownik wybiera pomiędzy kamerą wbudowaną w komputer a kamerą IP

<b>Przypadek użycia:</b> Rozpoznawanie twarzy przez kamerę IP
<b>Aktorzy:</b> Użytkownik
<b>Scenariusz główny</b>
<ol style="list-style-type: none"> <li>1. Użytkownik chce rozpoznawać twarze w czasie rzeczywistym</li> <li>2. Użytkownik wybiera opcję wykrywania twarzy przez kamerę IP</li> <li>3. Użytkownik podaje URL kamery IP</li> <li>4. Użytkownik rozpoznaje twarze przez kamerę IP</li> </ol>
<b>Rozszerzenia:</b>

<b>Przypadek użycia:</b> Dodawanie twarzy nowych użytkowników do bazy danych
<b>Aktorzy:</b> Użytkownik
<b>Scenariusz główny</b>
<ol style="list-style-type: none"> <li>1. Użytkownik chce dodać nową twarz do bazy danych użytkowników</li> <li>2. Użytkownik wpisuje ID i nazwę dodawanego użytkownika</li> <li>3. Użytkownik ma robionych 20 zdjęć</li> <li>4. Użytkownik zostaje dodany do bazy danych</li> </ol>
<b>Rozszerzenia:</b>

<b>Przypadek użycia:</b> Tworzenie pliku treningowego z którego program rozpoznaje twarze
<b>Aktorzy:</b> Użytkownik
<b>Scenariusz główny</b>
<ol style="list-style-type: none"> <li>1. Użytkownik chce stworzyć plik treningowy</li> <li>2. Użytkownik wybiera opcję z menu</li> <li>3. Użytkownik tworzy plik treningowy</li> </ol>
<b>Rozszerzenia:</b>
2.A.W celu lepszego wykrywania twarzy zalecane jest usunięcie błędnych zdjęć twarzy z pliku

<b>Przypadek użycia:</b> Modyfikowanie bazy danych przez program
<b>Aktorzy:</b> Użytkownik
<b>Scenariusz główny</b>
<ol style="list-style-type: none"> <li>1. Użytkownik chce zmodyfikować bazę danych</li> <li>2. Użytkownik wybiera opcję modyfikacji bazy danych z menu</li> <li>3. Użytkownik wprowadza ID, nazwę użytkownika i opcjonalnie wiek i płeć</li> <li>4. Użytkownik zostaje zmodyfikowany w bazie danych</li> </ol>
<b>Rozszerzenia:</b>
3.A. Nazwa użytkownika oraz płeć musi być wpisana w cudzysłowie

<b>Przypadek użycia:</b> Modyfikowanie bazy danych przez SQLite
<b>Aktorzy:</b> Użytkownik
<b>Scenariusz główny</b>
<ol style="list-style-type: none"> <li>1. Użytkownik chce zmodyfikować bazę danych</li> <li>2. Użytkownik otwiera program SQLite</li> <li>3. Użytkownik modyfikuje dane przez SQLite</li> <li>4. Użytkownik zostaje zmodyfikowany w bazie danych</li> </ol>
<b>Rozszerzenia:</b>




<b>Przypadek użycia:</b> Wykrywanie ruchu
<b>Aktorzy:</b> Użytkownik
<b>Scenariusz główny</b>
<ol style="list-style-type: none"> <li>1. Użytkownik chce wykrywać ruch w czasie rzeczywistym</li> <li>2. Użytkownik wybiera opcję wykrywania ruchu z menu</li> <li>3. Użytkownik wykrywa ruch</li> </ol>
<b>Rozszerzenia:</b>

<b>Przypadek użycia:</b> Wyświetlanie bazy danych w programie
<b>Aktorzy:</b> Użytkownik
<b>Scenariusz główny</b>
<ol style="list-style-type: none"> <li>1. Użytkownik chce wyświetlić bazę danych</li> <li>2. Użytkownik wybiera opcję wyświetlania bazy danych z menu</li> <li>3. Baza danych zostaje wyświetlona w terminalu</li> </ol>
<b>Rozszerzenia:</b>

## 7. Schemat bazy danych

Baza danych aplikacji realizowana jest za pomocą SQLite i służy do gromadzenia informacji, które będą wykorzystywane do identyfikacji osób rozpoznanych przez kamerę. W bazie danych znajduje się jedna tabela, która zawiera takie informacje jak:

- ID
- Nazwę
- Płeć
- Wiek

	Nazwa	Typ danych	Klucz Główny	Klucz Obcy	Wartości unikalne	Warunek	Niepuste	Zestawienie
1	ID	INT						
2	Name	STRING						
3	Age	INT						
4	Gender	TEXT						



## **8. Architektura systemu**

Aplikacja desktopowa dedykowana na komputery z systemem operacyjnym z rodziny Microsoft Windows. Program wykorzystuje bibliotekę OpenCV do przetwarzania obrazu oraz bibliotekę PyQt do tworzenia interfejsu graficznego.

Aplikacja korzysta z bazy danych SQLite w której znajdują się informacje na temat osób, które chcemy rozpoznać za pomocą kamery. Dane te można wprowadzić ręcznie lub za pomocą aplikacji.

Program daje możliwość podłączenia wielu kamer IP działających w sieci lokalnej. Po wykryciu twarzy za pomocą klasyfikatora kaskad Haara, wykorzystywany jest algorytm LBPH (Local Binary Patterns Histogram) w celu identyfikacji wykrytej twarzy. Program wyświetla na ekranie informacje na temat rozpoznanych osób, a także ich liczbę.

## 9. Napotkane problemy

Pierwszym problemem jaki napotkaliśmy była identyfikacja osób nie znajdujących się w bazie danych. W przypadku wykrycia twarzy takiej osoby, program przypisywał jej dane innej osoby, która znajdowała się w bazie. Rozwiązaniem tego problemu okazało się użycie wartości określającej prawdopodobieństwo rozpoznania twarzy zwracanej przez algorytm. Dzięki temu twarze, dla których ta wartość była niska można było określić jako “niezidentyfikowane”.

Rozwiązanie pierwszego problemu doprowadziło jednak do pogłębienie drugiego jakim jest wrażliwość na warunki oświetleniowe. Nie udało nam się znaleźć rozwiązanie, które sprawiłoby, żeby program działał tak samo, niezależnie od światła. Jedyne co można zrobić to dostosować próg przy którym dana osoba zostałaby zaklasyfikowana jako “niezidentyfikowana” do warunków panujących w danym pomieszczeniu.

Kolejnym problemem jaki napotkaliśmy to tworzenie pliku treningowego z którego program rozpoznawał twarze. Pierwszą rzeczą którą robimy to tworzenie tzw. “Dataset” w którym znajdują się zdjęcia na podstawie których program rozpoznaje twarz. Następnie, żeby skuteczność rozpoznawania była wyższa musimy wejść w plik gdzie zapisane są zdjęcia i ręcznie usunąć złe zdjęcia. Zauważyliśmy, że dzięki temu znacząco poprawiła się skuteczność w rozpoznawaniu.

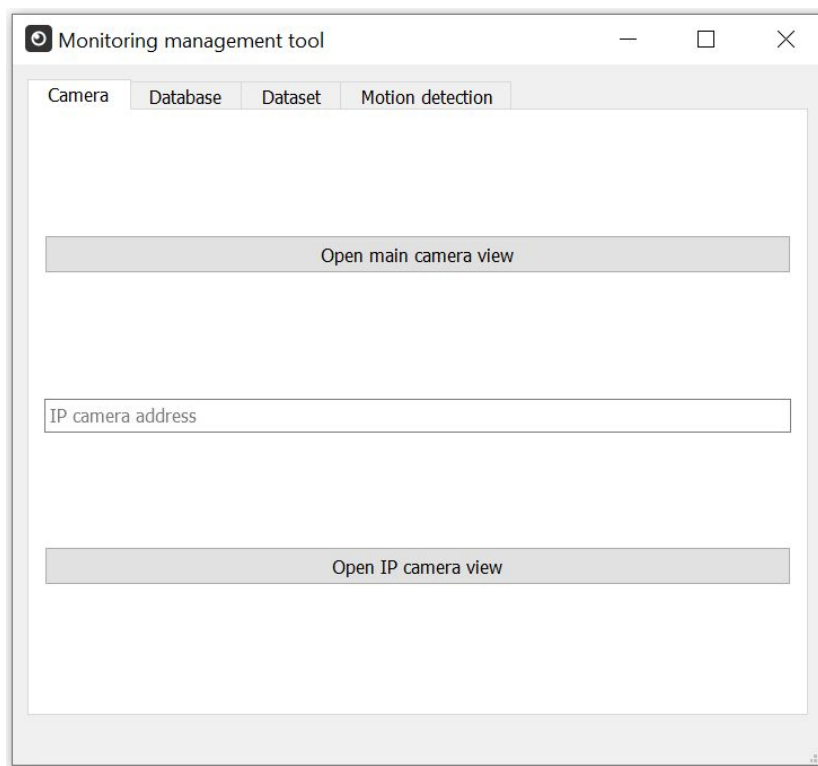


## 10. Instrukcja użytkowania

W tym rozdziale przedstawiono instrukcję użytkowania aplikacji.

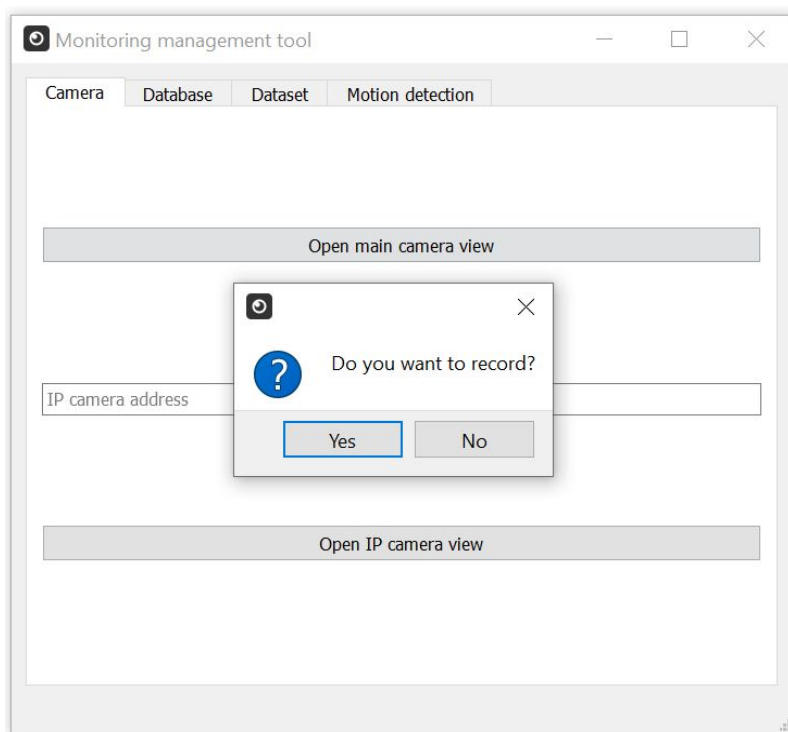
### I. Rozpoznawanie twarzy

Menu główne posiada cztery zakładki. Pierwsza z nich (Camera) wiąże się z podstawowym zadaniem aplikacji czyli wykrywaniem oraz rozpoznawaniem twarzy osób znajdujących się w zasięgu widzenia kamery.



Rysunek 10.1: Okno główne aplikacji

Mamy do wyboru dwie opcje. Po wybraniu pierwszej z nich (Open main camera view) aplikacja pyta czy chcemy rozpocząć nagrywanie. Po potwierdzeniu możemy nadać nazwę plikowi z nagraniem lub pozostawić nazwę domyślną.

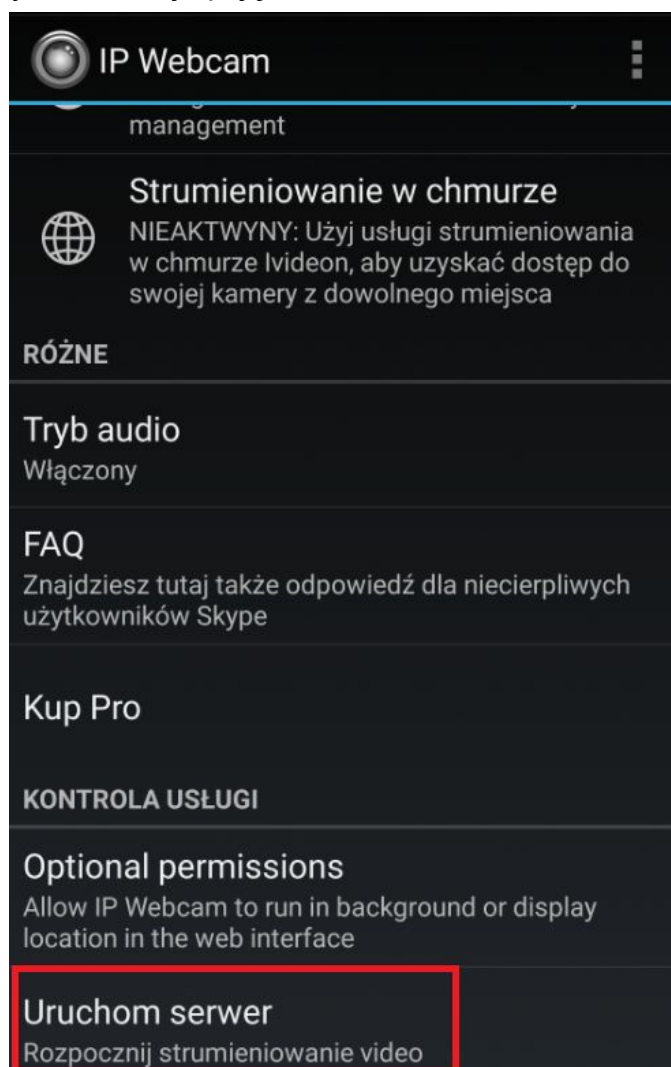


Rysunek 10.2: Rozpoczęcie nagrywania

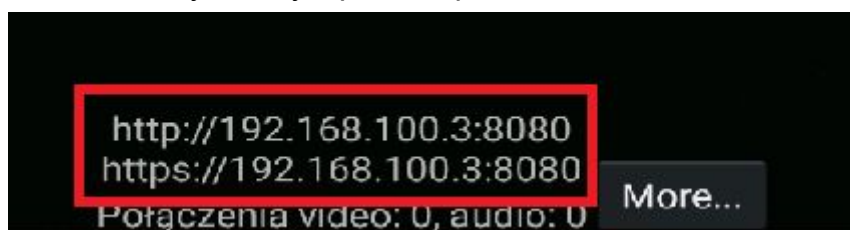
Kolejno mamy opcje pracy na nagraniu z kamery IP. W tym celu należy ściągnąć aplikację na telefon ze sklepu Google Play aplikacje “IP Webcam” (<https://play.google.com/store/apps/details?id=com.pas.webcam>)

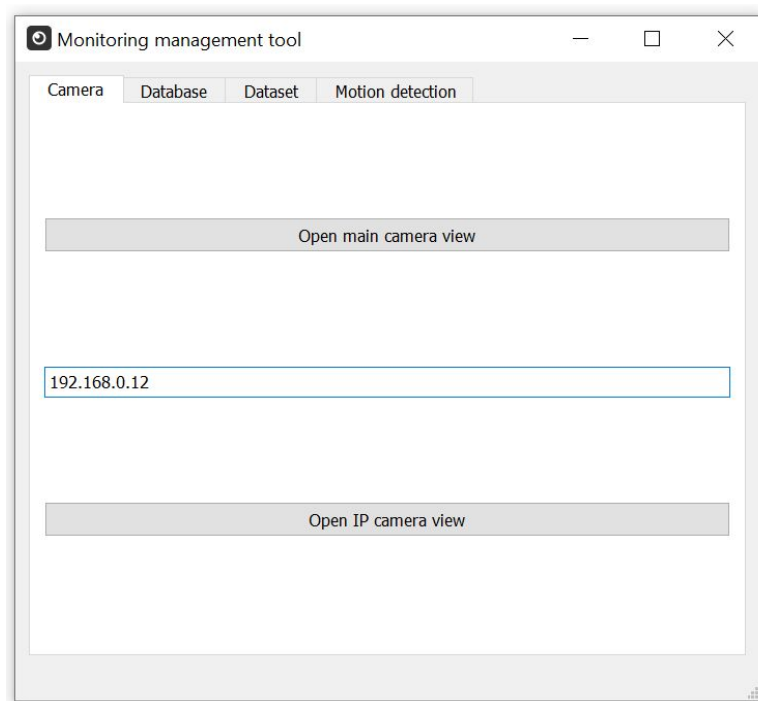


Następnie wybrać w niej opcję “Uruchom serwer”.

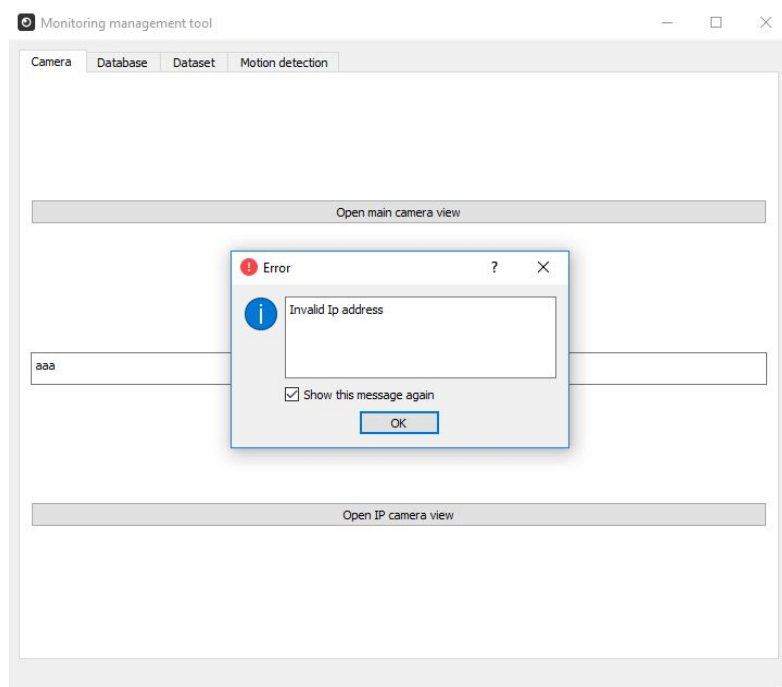


Telefon zacznie nagrywać obraz jednocześnie wyświetlając na ekranie swój adres IP, który należy wpisać w polu “IP camera address”.





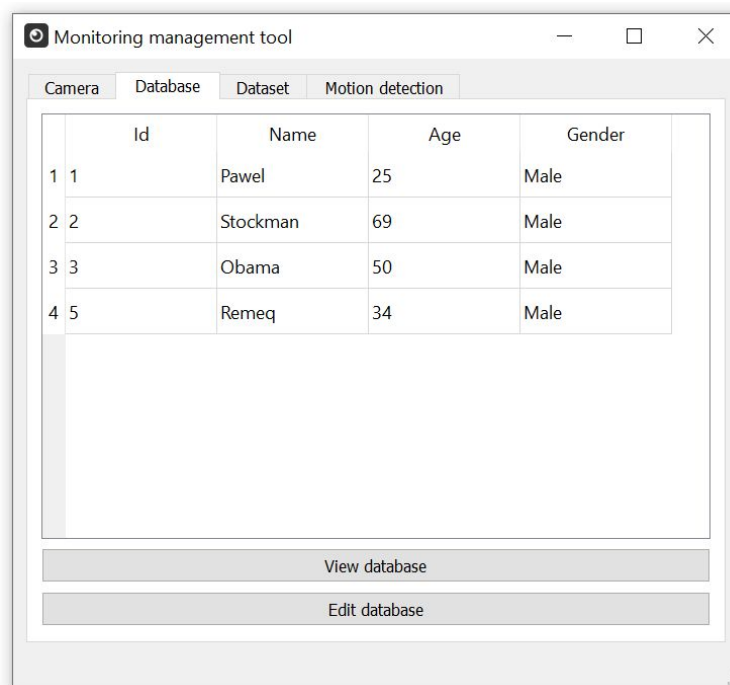
Rysunek 10.3: Połączenie z kamerą IP



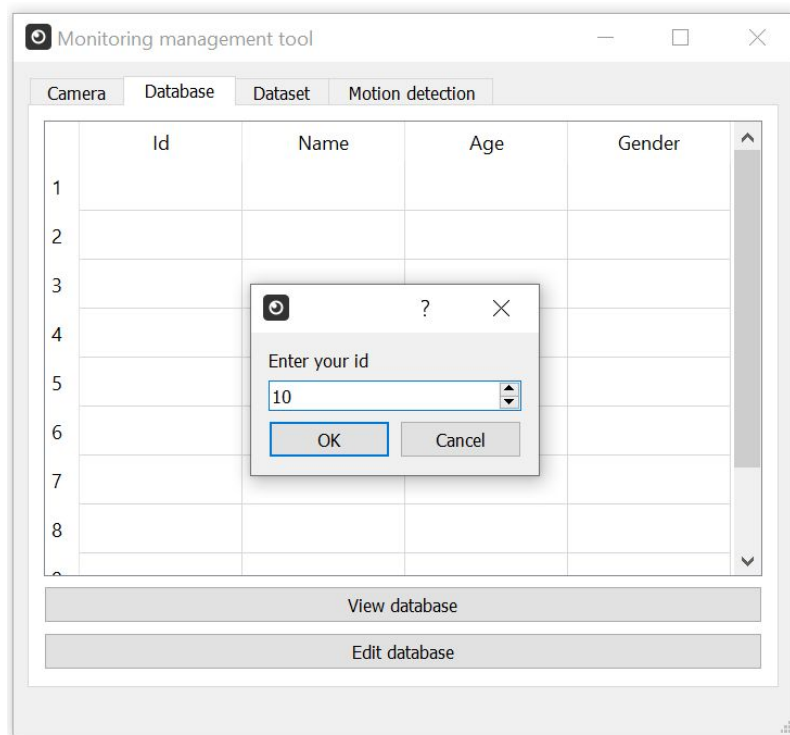
Rysunek 10.4: Błędnie wpisany adres IP

## II. Baza danych

Kolejna zakładka (Database) związana jest z bazą danych aplikacji. Po wybraniu opcji “View database” ukazuje się tabela z osobami znajdującymi się w naszej bazie.



Druga opcja w tej zakładce daje nam możliwość modyfikacji bazy danych. Po jej wybraniu program będzie nas po kolei prosił o podanie numeru id, nazwy, wieku oraz płci.



Rysunek 10.4: Modyfikacja bazy danych

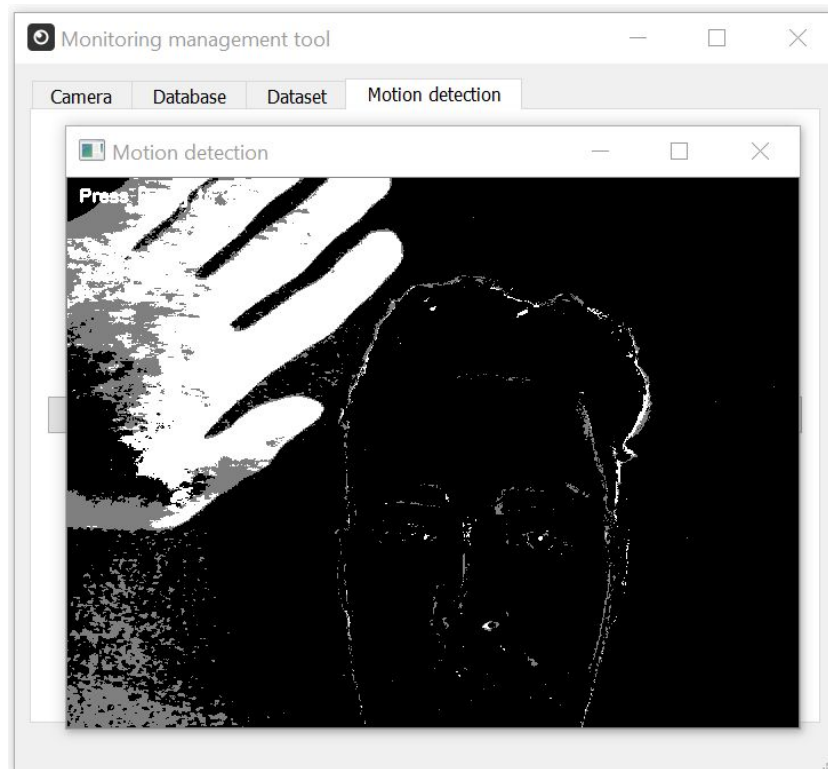
### III. Tworzenie datasetu

Trzecia zakładka pozwala nam na dodanie do bazy danych nowej osoby wraz ze zdjęciami potrzebnymi do jej rozpoznania przez kamerę. W tym celu wybieramy "Create data set" i podajemy kolejne dane po czym program uruchamia kamerę i robi nam serię zdjęć.



#### IV. Wykrywanie ruchu

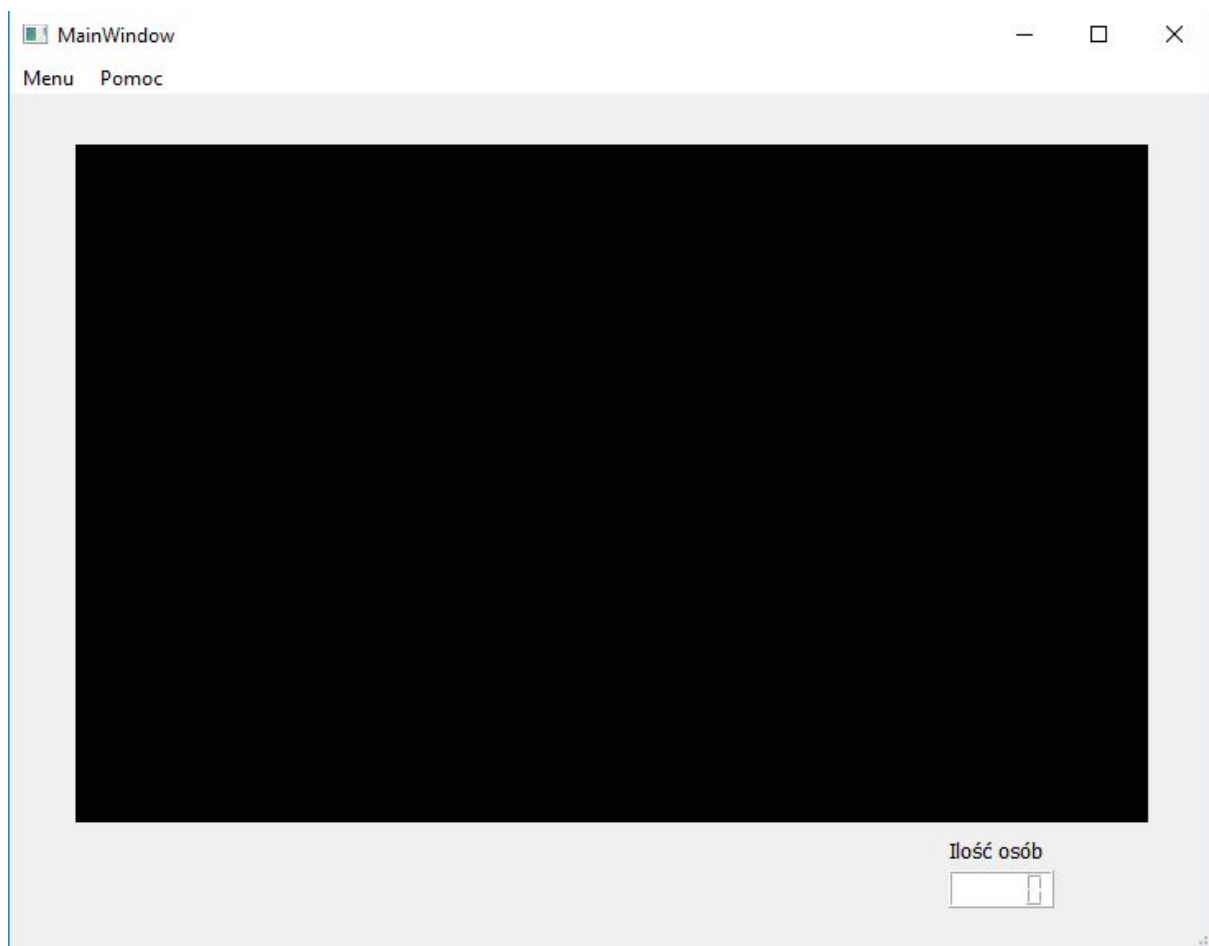
Ostatnią funkcją aplikacji jest wykrywanie ruchu. Aby jej użyć wybieramy zakładkę "Motion detection" a następnie wciskamy przycisk "Open motion detection camera". Aplikacja otworzy kamerę z funkcją wykrywania ruchu.



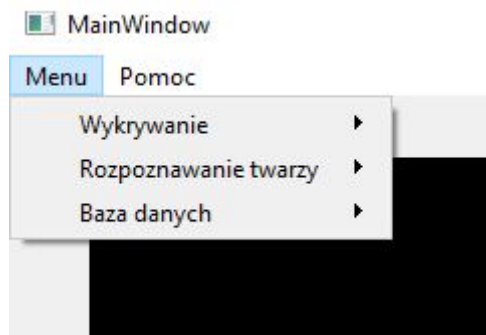
Rysunek 10.4: Wykrywanie ruchu

## 11. Dalszy rozwój aplikacji

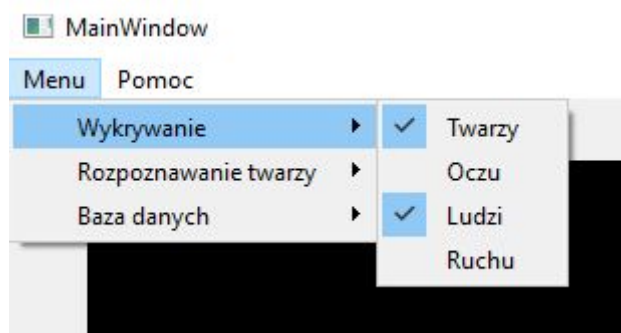
W dalszym rozwoju aplikacji chcielibyśmy stworzyć bardziej rozbudowany interfejs.



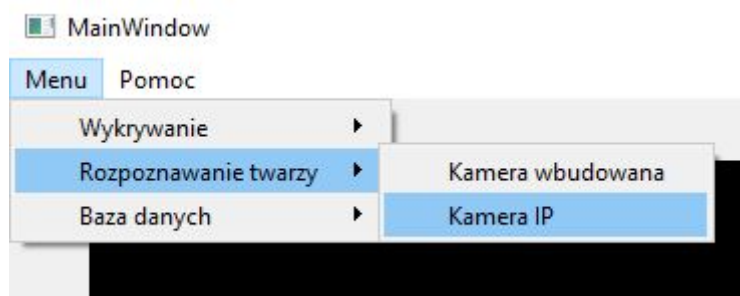
Koncepcja menu głównego aplikacji w której będzie do wybrania kilka opcji.



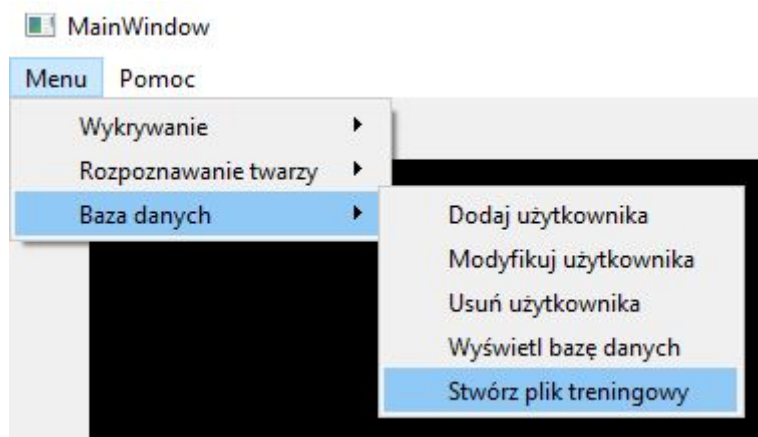
Możliwość wyboru co ma być wykrywane w czasie rzeczywistym.



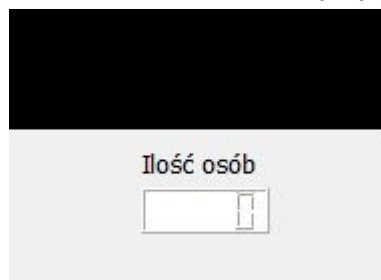
Możliwość wybrania z której kamery ma być przechwytywany obraz podczas rozpoznawania twarzy.



Możliwość modyfikacji bazy danych.



Zliczanie ilości osób znajdujących się w danej chwili na wideo.



## **12. Bibliografia**

1. OpenCV, <https://opencv.org/>
2. Local Binary Patterns, [https://en.wikipedia.org/wiki/Local\\_binary\\_patterns](https://en.wikipedia.org/wiki/Local_binary_patterns)
3. Wykrywanie ruchu, [https://docs.opencv.org/3.1.0/d1/dc5/tutorial\\_background\\_subtraction.html](https://docs.opencv.org/3.1.0/d1/dc5/tutorial_background_subtraction.html)
4. Qt Designer, <https://doc.qt.io/qt-5/qtdesigner-manual.html>
5. PyQt, <https://wiki.python.org/moin/PyQt/Tutorials>