



Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

Wydział Elektrotechniki, Automatyki,  
Informatyki i Inżynierii Biomedycznej

## Praca dyplomowa inżynierska

Wideo detekcja pojazdów samochodowych w ruchu  
drogowym

Video-based vehicle detection for road traffic  
applications

Autor:

Kierunek studiów:

Opiekun pracy:

Rafał Prusak

Automatyka i Robotyka

dr inż. Zbigniew Marszałek

Kraków, 24 grudnia 2015

Oświadczam, świadomy(-a) odpowiedzialności karnej za poświadczenie nieprawdy, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie, i nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.

.....

## Spis treści

1. Wstęp .....	2
2. Metody pomiarów parametrów ruchu drogowego .....	3
3. Systemy wizyjne w zastosowaniach komercyjnych .....	7
4. Zastosowane narzędzia. ....	11
5. Opis programu.....	15
6. Algorytm detekcji i klasyfikacji pojazdów. ....	22
7. Analiza wyników .....	24
8. Rozwój oprogramowania .....	25
9. Podsumowanie.....	27
Bibliografia .....	27

# 1. Wstęp

Tutaj będzie wstęp do pracy inżynierskiej  
Burzliwy rozwój nauki i techniki w XX wieku...

## 2. Metody pomiarów parametrów ruchu drogowego

W wyniku rozwoju motoryzacji oraz rozbudowy infrastruktury drogowej pojawiła się konieczność zbierania i analizy informacji o ruchu pojazdów.

### Generalny Pomiar Ruchu Drogowego

W Polsce od roku 1965, co 5 lat, dokonywany jest Generalny Pomiar Ruchu Drogowego (GPRD). Ma on na celu pozyskanie charakterystyk ruchu drogowego na wszystkich odcinakach dróg krajowych i wojewódzkich. Uzyskiwane w wyniku GPRD pomiary są wykorzystywane przy planowaniu rozbudowy infrastruktury drogowej, organizacji ruchu, utrzymaniu dróg, określaniu przyczyn wypadkowości czy badaniu wpływu ruchu na komfort życia mieszkańców i środowisko naturalne.

W czasie pomiarów pojazdy są klasyfikowane do jednej spośród 8 kategorii:

1. motocykle,
2. samochody osobowe,
3. lekkie samochody ciężarowe,
4. samochody ciężarowe bez przyczepy,
5. samochody ciężarowe z przyczepami/ naczepami,
6. autobusy,
7. ciągniki rolnicze,
8. rowery.

Metodologię dokonywania pomiarów, terminy pomiarów, zadania obserwatorów, etc reguluje zarządzenie nr 38 Generalnego Dyrektora Dróg Krajowych i Autostrad z dnia 1 września 2014 r #dodać odnośnik do bibliografii

## Parametry ruchu drogowego

Podczas dokonywania pomiarów dąży się do pozyskania zarówno indywidualnych informacji o pojazdach jak również ogólnej charakterystyki ruchu. Wyróżnia się w tym celu szereg parametrów:

- Prędkość - chwilowa, jazdy, podróży, miarodajna.
- Czas podróży - czas pokonania odcinka drogi.
- Natężenie ruchu - liczba pojazdów na jednostkę czasu.
- Gęstość ruchu - liczba pojazdów znajdujących się na odcinku drogi na jednostkę czasu.
- Zajętość drogi - procent drogi zajęty przez pojazdy.
- Bezpieczeństwo ruchu.
- Wpływ na środowisko.
- Zużycie paliwa.

Pomiarów ruchu drogowego dokonuje się w punkcie, na krótkim odcinku drogi, w zdefiniowanym obszarze, na całej trasie lub za pomocą pojazdu/pojazdów próbnych poruszających się po drodze. Rozróżnia się 4 typy ruchu pojazdów: jazda swobodna, jazda przy maksymalnym przepływie, jazda w warunkach dużej gęstości oraz zator.

Indywidualny pojazd można opisać szeregiem wartości: klasą pojazdu, numerem rejestracyjnym, liczbą osi, odległościami między osiami, długością, szerokością, wysokością, prędkością czy naciskiem na jezdnię. Stosuje się także opis strefowy, środowiskowy, pogodowy i opis jezdni.

## Detektory

Detektor jest to urządzenie składające się z czujnika, układu kondycjonowania, dyskryminatora oraz elementu wykonawczego. Jego celem jest wykrycie obecności pojazdu w punkcie bądź strefie pomiaru. Detektor powinien być urządzenie prostym w obsłudze i montażu, bezawaryjnym, pozwalającym na łatwy odczyt informacji. Stosowane są dwa rodzaje detektorów: bezinwazyjne i wbudowane (instalowane w nawierzchni).

Klasyfikacja detektorów wbudowanych:

- Linowy - drut stalowy lub linka nylonowa zamontowana jest na wysokości kilku centymetrów na jezdni. Nacisk przejeżdżającego pojazdu powoduje naciąg linki i uruchomienie zestyku, który generuje impuls. Ten rodzaj detektory może generować fałszywe impulsy, wymaga regulacji i poprawy naciągu linki, ma ograniczoną maksymalną prędkość pomiarów. Jego zaletą jest prosta budowa.

- Taśmowy - dwie taśmy z gumowymi wkładkami, wypełnione gazem elektrycznie obojętnym. Nacisk pojazdu powoduje zetknięcie się ze sobą taśm. Do działania wymaga specjalnego osprzętu. Występują niekorzystne drgania styków. Montaż wymaga interwencji w nawierzchnię.
- Hydrauliczny - nacisk pojazdu na wąż wypełniony olejem powoduje podniesienie się poziomu w kapilarze i uruchomienie zestyku. Pojazdy o dużej masie mogą uszkodzić ten typ detektora.
- Pneumatyczny - działa na podobnej zasadzie co hydrauliczny.
- Reluktancyjny - nacisk samochodu powoduje zwierzenie się szczeliny. Następuje zmiana reluktancji obwodu magnetycznego w wyniku czego zmienia się częstotliwość sygnału wyjściowego.
- Rezystancyjny - nacisk pojazdu powoduje nieliniową zmianę rezystancji czujnika. Detektor ma ograniczoną prędkość pomiaru do max 80 km/h, więc jest rzadko wykorzystywany.
- Pojemnościowy - dwie elektrody oddzielone dielektrykiem. Nacisk pojazdu powoduje zmniejszenie szerokości kondensatora i zmianę jego pojemności. Czujniki te mają kiepską odporność na uszkodzenia mechaniczne i zasolenie. Ich montaż wymaga kosztownej interwencji w nawierzchnię.
- Piezoelektryczny - jest to taśma wykonana z materiału piezoelektrycznego lub kabel koncentryczny z umieszczoną w środku elektrodą. Czujniki są wrażliwe na zmiany temperatury, której wzrost powoduje pogorszenie jakości pomiaru. Mierzony może być jedynie nacisk zmienny w czasie, co ogranicza prędkość pomiaru do zakresu 20-110 km/h. Ten sposób pomiaru jest szeroko stosowany z powodu dużej czułości, szerokiemu zakresowi temperatury oraz niskiej cenie wykonania.
- Kvarcowe - działa na podobnej zasadzie co piezoelektryczny: nacisk samochodu na aluminiowy profil powoduje nacisk na element kwarcowy. Jest często stosowany w systemach ważenia pojazdów. W przeciwieństwie do poprzedniego typu, cechuje się odpornością na oddziaływanie sił podłużnych i większą wrażliwością na oddziaływanie sił poprzecznych. Czujnik może pracować w szerokim zakresie temperatur i prędkości. Jest odporny na zakłócenia temperaturowe i elektromagnetyczne. Jego trwałość osiąga nawet 10 lat.
- Tensometryczny - używany jest zarówno w pomiarach nacisków osi pojazdów statycznych i dynamicznych. Ma postać platformy na której zamontowywane są tensometry, najczęściej w układzie pełnego mostka. Charakteryzuje się dużą dokładnością pomiaru (błędy rzędu 2% przy pomiarze statycznym). Może mierzyć pojazdy o prędkości do 200 km/h. Jego trwałość sięga 10 lat.

- Indukcyjny pętlowy - jest najbardziej rozpowszechnionym typem detektora. Może być stosowany w pomiarach prędkości, wysokości zawieszenia, ilości osi, detekcji incydentów. Uzyskiwany sygnał jest nazwany profilem magnetycznym pojazdu. Czujnik jest odporny na czynniki zewnętrzne, ma dużą trwałość, jego instalacja na jezdni jest prosta i mało kosztowna.
- Magnetyczne - reaguje na zmiany pola magnetycznego. Jest bardziej czuły na zakłócenia niż pętlowy, ale jego instalacja w nawierzchni jest znacznie prostsza.

Drugim rodzajem detektorów są detektory nieinwazyjne. Spośród nich wyróżniamy:

- Laserowe aktywne - zbudowane są z nadajnika promieniowania, układu optycznego oraz odbiornika promieniowania. Mogą być wykorzystane do pomiaru prędkości, wysokości czy, przy odpowiedniej częstotliwości próbkowania, budowaniu profilu samochodu. Czujnik tej klasy mogą pracować w szerokim zakresie temperatur, wykazując odporność na warunki atmosferyczne oraz są niezależne od zewnętrznego oświetlenia.
- Laserowa pasywne - reagują na zmiany w promieniowaniu podczerwonym wywołane ruchem obiektów w obszarze detekcji. Detektor ten jest bardzo łatwy w instalacji - nie wymaga żadnej interwencji w ruchu drogowym. Detektory tego typu nie oddziałują na siebie nawzajem.
- Akustyczne - składają się z matrycy mikrofonów, które zbierają dźwięki przejeżdżających samochodów: opony, silnik, ruch powietrza.
- Ultradźwiękowe - generują falę dźwiękową niesłyszalną dla człowieka, która służy wykrywaniu obecności obiektów w sferze detekcji.
- Mikrofalowe - działają na zasadzie radaru, generują fale wysokiej częstotliwości. Mogą pracować praktycznie w każdych warunkach.
- Kamery z systemami przetwarzania obrazu.



### 3. Systemy wizyjne w zastosowaniach komercyjnych

Rozdział ten stanowi przegląd dostępnych na rynku systemów pomiaru ruchu drogowego, w których zastosowanie znalazły algorytmy wizyjne.

#### Charakterystyka systemów wizyjnych

Systemy VIP (Video Image Processing) są coraz częściej stosowane przy pomiarach parametrów ruchu drogowego. Początkowo służyły one jedynie wizualizacji pomiarów bądź jako pomoc dla operatora w procesie weryfikacji poprawności pomiarów dokonywanych „standardowymi” metodami. Rozwój mocy obliczeniowych komputerów sprawił, że pojawiła się możliwość automatycznej analizy obrazu i wydobywania z niego użytecznych informacji, takich jak:

- obecność pojazdów na skrzyżowaniach, drogach, tunelach, etc,
- kierunek ruchu,
- klasyfikacja pojazdów,
- pomiar natężenia ruchu,
- informacje o obecności pieszych i rowerzystów,
- detekcja incydentów drogowych.

Obecnie wykorzystywane aplikacje wizyjne mogą działać zarówno jako samodzielne metody pomiaru lub stanowić część systemów kontroli wykroczeń, pomiaru prędkości czy ważenia samochodów.

Efektywność systemów wizyjnych szacuje się na około 90%. Posiadają one wiele zalet:

- są całkowicie bezinwazyjne,
- mimo wysokiej ceny budowy i instalacji systemu, jego dalsza eksploatacja odbywa się niskim kosztem,
- mają szerokie możliwości konfiguracji i dostosowywania strefy detekcji,
- mogą być zintegrowane z systemami pomiaru wbudowanymi w jezdnię,

- w zaawansowanych aplikacjach kamera może dokonywać auto-pozycjonowania w celu kompensacji zakłóceń wynikających ze zmiany położenia słońca czy spadku jakości oświetlenia,
- istnieje możliwość streamowania obrazu przez sieć Internet.

Największymi wadami metod wizyjnych są:

- niekorzystny wpływ warunków atmosferycznych: deszcz, śnieg, mgła, zachmurzenie,
- zjawiska związane ze wzajemnym mijaniem i zasłanianiem się pojazdów,
- zabrudzenia odkładające się na układzie optycznym.

Ważnym czynnikiem mającym wpływ na działanie systemu jest sposób montażu kamery względem nadzorowanego pasa ruchu. Najczęściej umieszcza się kamerę wysoko nad pasem ruchu. Umieszczenie punktu pomiaru z boku drogi zmniejsza skuteczność pomiaru o około 10%.

## NeuroSoft

Rozwiązania wrocławskiej firmy NeuroSoft są jednymi z najpopularniejszych w kategorii systemów pomiaru i zarządzania ruchem drogowym. Firma ta oferuje system o nazwie „Neuro Car” oraz bardziej rozbudowany „Neuro Car 2.0”. Programy te dokonują wideo rejestracji pojazdów, odczytów tablic rejestracyjnych i wstępnej selekcji danych. System jest dostarczany w postaci terminala zbudowanego z kamery przemysłowej, komputera, karty WLAN, modemu GPRS i zalicza. Instalacja systemu jest bardzo prosta: polega na ustawieniu kamery, podpięciu zasilania i konfiguracji parametrów.

Zaletami systemu są:

- niska cena,
- możliwość integracji z innymi systemami
- wysoka skuteczność rozpoznawania,
- praca w trybie rzeczywistym,
- odporność na warunki zewnętrzne,
- rozpoznawanie marki, modelu, koloru,
- zastosowanie kamer z dużą dynamiką jasności,
- funkcjonalność zdalnego dostępu do danych.

W systemie Neuro Car zastosowano procesory wielordzeniowe, DSP oraz system operacyjny Linux, umożliwiając w ten sposób pomiar w stałej ilości 25 klatek na sekundę. Rozpoznanie, klasyfikacja i generacja opisu trwa nie mniej niż 100 ms. Dodatkowo program generuje od 3 do 8 zdjęć dla zidentyfikowanego pojazdu. Producent określa skuteczność detekcji na 99%.

W skład systemu wchodzi wiele dodatkowych aplikacji:

- Velocity - pomiar prędkości chwilowej na podstawie serii zdjęć.
- Dangerous goods - identyfikacja samochodów przewożących materiały niebezpieczne.
- Classification - klasyfikacja pojazdów wg marki i typu(jednośladowy, osobowy, ciężarowy).
- PDA - aplikacja na platformy mobilne do rozpoznawania numerów tablic rejestracyjnych.
- SectionControl - pomiar prędkości średniej na odcinku trasy.
- RedLight - detekcja przejazdu na czerwonym świetle.
- City - tworzenie płatnych stref w miastach i pobieranie opłat za przebywanie w nich.
- Parking - monitorowanie parkingu, rozliczenie opłat za parkowanie.
- Monitoring - uniwersalna aplikacja do obserwacji przepływu pojazdów.
- Base - integracja z basami mySQL lub postgrSQL.
- Monitor - oprogramowanie do wizualizacji wyników pomiaru.
- ScaleGate - system rozpoznawania tablic i ważenia pojazdów, stosowany w hurtowniach i magazynach w celu pozyskiwania statystyk o tonażu materiałów przychodzący i opuszczających teren zakładu.
- Neurocar WIM&MIM - automatyczna detekcja pojazdów ponadgabarytowych(przeciążonych oraz za wysokich), których obecność powoduje pogorszenie właściwości nawierzchni oraz stwarza zagrożenie na mostach, pod wiaduktami i w tunelach.

## Cat Traffic

Poznańska firma Cat Traffic zajmuje się systemami stanowiącymi połączenie metod wizyjnych z wykorzystaniem płyt warzących. Firma wdrożyła kilkanaście stacji warzących

na terenie Niemiec, Polski, Austrii i Szwajcarii. W Polsce CatTraffic wdrożyło na autostradzie A4 systemy pomiaru wagi i badania statystyki ruchu pojazdów, oraz system czasu przejazdu na „Zakopiance”.

Rozwiązania oferowane przez firmę:

- System pomiaru prędkości EasyCount oprócz pomiaru prędkości może badać również kierunek jazdy, odległości między pojazdami, i długość pojazdu. Przy użyciu dodatkowego osprzętu, EasyCount umożliwia pomiar na dwóch pasach ruchu. System ma do dyspozycji pamięć, która oferuje miejsce na dane o 800 000 przejazdach. Można dokonywać jego konfiguracji za pomocą modułu GSM. Zasilanie zapewniają akumulatory lub baterie słoneczne.
- Easy Count-Bike- oferuje podobne funkcje co Easy Count, ale w odniesieniu do rowerzystów.
- Licznik AVC - to pętlowy system pomiaru ruchu z funkcjonalnością zapisu, transferu i analizy danych.
- V-REX 1000 - mobilny radar do odczytu tablic i badania struktury ruchu.
- VIM-VIVER - automatyczna stacja do ważenia pojazdów w ruchu, wraz z pozyskiwaniem profilu pojazdu i numeru tablic rejestracyjnych.
- SAW - przenośna waga, umożliwiająca pomiar pojazdów o nacisku do 15 ton. Wykorzystywana przez Państwową Inspekcję Transportu Drogowego.
- ANPR - automatyczne rozpoznawanie tablic, z maksymalnie 8 kamer jednocześnie.

## 4. Zastosowane narzędzia.

Poniższy rozdział zawiera krótki opis języka programowania, w którym została zaimplementowana praca inżynierska, zastosowanych zewnętrznych bibliotek oraz innych narzędzi użytych przy wykonaniu programu.

### Python

Algorytm wideo detekcji pojazdów został zaimplementowany w języku programowania Python, w wersji 3.4. Język ten posiada wiele korzystnych cech, które miały istotny wpływ na proces powstawania programu:

- Jest językiem obiektowym wysokiego poziomu. Obiektość kodu, w przeciwieństwie do takich popularnych języków jak Java czy C#, nie jest jednak odgórnie wymuszana przez składnię języka. Powoduje to, że w Pythonie można pisać proste liniowe skrypty, służące np. do testowania funkcjonalności programu.
- Gramatyka języka jest bardzo prosta, podobna do składni języka angielskiego, przez co kod jest jasny i zrozumiały. Dodatkowo język ten wymusza na programiście stosowanie standardu PEP8, co powoduje jednolity i przejrzysty styl kodu w powstającym programie.
- Posiada bogatą bibliotekę standardową zawierającą moduły do programowania sieciowego, pracy z plikami html, xml, json, tworzenia interfejsów gui czy przetwarzania wielowątkowego.
- Jest językiem interpretowanym, więc programy wykonane w tym języku są niezależne od platformy i mogą być łatwo przenoszone na inne systemy operacyjne bądź platformy sprzętowe (komputer PC, systemy embeded).
- Ogromna popularność języka Python i szeroka społeczność skupiona wokół niego, skutkuje w ogromnej bazie gotowych rozwiązań, porad i tutoriali dostępnych w sieci.

### OpenCV

OpenCV jest biblioteką „open source” służącą do cyfrowego przetwarzania obrazu wizyjnego. Jest ona bezpośrednio zaimplementowana w języku C/C++. Dostępna jest na

wszystkich systemach operacyjnych stacjonarnych i mobilnych: Windows, Linux, MacOS, Android, iOS. Biblioteka zapewnia API dla wielu języków, takich jak: Python, Ruby, Java, Matlab. Została napisana z myślą o programach potrzebujących najwyższej wydajności i wykorzystaniu w aplikacjach czasu rzeczywistego. Najnowsze wydanie, w wersji 3.0, przynosi optymalizację biblioteki na wielu poziomach: algorytmów, wykorzystania wątków oraz rdzeni czy rozkazów dla CPU. Najwięksi producenci sprzętu komputerowego, tacy jak Intel czy Nvidia, także wzięli udział w tworzeniu biblioteki, dostarczając własne niskopoziomowe programy do wykonywania operacji na GPU.

Biblioteka jest podzielona na kilkanaście modułów, dostępnych w postaci bibliotek statycznych lub dynamicznych. Poniżej zostały omówione najważniejsze moduły, które znalazły zastosowanie w implementacji pracy inżynierskiej:

- core - podstawowe funkcje do działań na tablicach wielowymiarowych,
- highgui - obsługa graficznego interfejsu użytkownika, wyświetlanie zdjęć i wideo ,
- imgproc - funkcje do przetwarzania obrazu: filtracja, transformacje geometryczne, konwersje między przestrzeniami barw,
- imgcodecs - interfejs do odczytu i zapisu zdjęć,
- videoio - przechwytywanie obrazu wideo,
- video - analiza obrazu wideo(śledzenie ruchu, wyodrębnianie tła),
- features2D - znajdowanie krawędzi, dopasowywanie szablonów,
- objectdetect - dopasowywanie i wykrywanie obiektów.

OpenCv znalazło wiele zastosowań w aplikacjach przemysłowych, takich jak: inspekcja gotowych produktów w fabrykach, medycyna, bezpieczeństwo, systemy wizyjne „wielowymiarowe”, robotyka.

## NumPy

NumPy jest zewnętrzną biblioteką „open source”, służącą do przeprowadzania obliczeń na n-wymiarowych tablicach. Zapewnia ona funkcjonalność podobną do środowiska Matlab. Podstawowa typ tablicy, używany przez bibliotekę znacząco różni się od tablicy(listy) samego Pythona: lista jest kontenerem heterogenicznym, mogącym dynamicznie zmieniać swój rozmiar, natomiast „ndarray” jest tablicą homogeniczną o stałym rozmiarze. Rozwiązanie takie zapewnia wysoką wydajność obliczeń i możliwość bezpośredniej integracji z plikami binarnymi biblioteki napisanymi w języku C/C++.

## SciPy

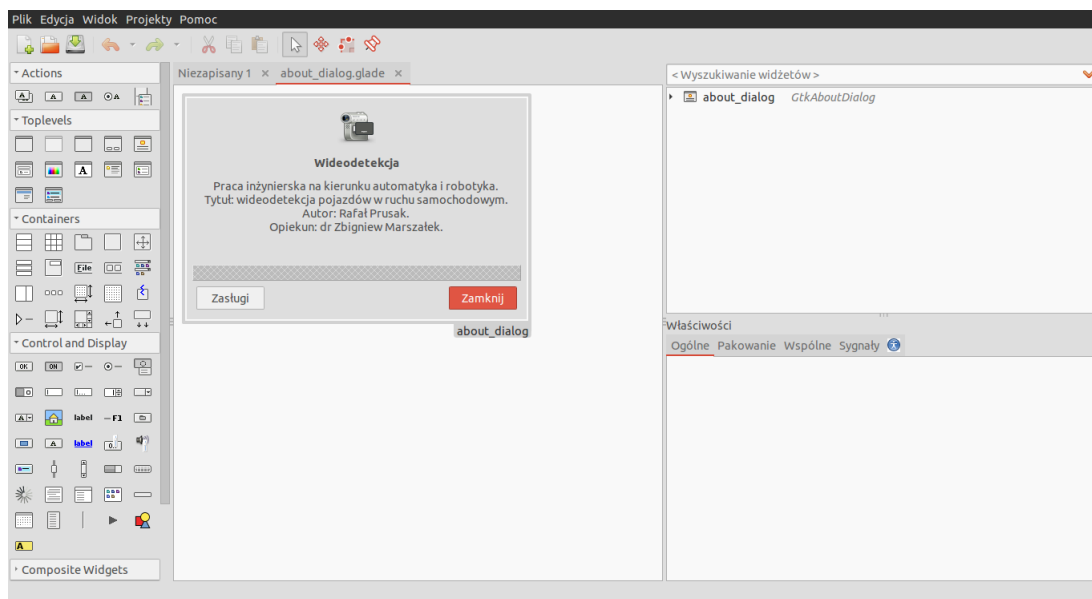
Biblioteka SciPy (skrót od „Scientific Python”) jest biblioteką „open-source” zawierającą funkcje do rozwiązywania zadań optymalizacji, algebry liniowej, przetwarzania sygnałów czy rozwiązywania równań różniczkowych. Biblioteka pozwala zrealizować większość funkcjonalności środowiska Matlab.

## PyGtk + Glade

W celu zaprojektowania i stworzenia interfejsu graficznego użytkownika zostało wykorzystane PyGtk oraz program Glade.

PyGtk jest nakładką na bibliotekę Gtk+, pozwalającą na tworzenie graficznego interfejsu użytkownika przy użyciu języka Python. Znalazła ona zastosowanie w wielu popularnych aplikacjach: BitTorrent, Gedit, GIMP. Jest częścią linuksowego środowiska graficznego GNOME.

Glade jest programem do projektowania interfejsu graficznego użytkownika z wykorzystaniem „widgetów” z biblioteki Gtk. Stworzony projekt okna interfejsu jest zapisywany do pliku xml. Spreparowany w ten sposób plik, może być wykorzystany do zbudowania okna za pomocą klasy GtkBuilder.

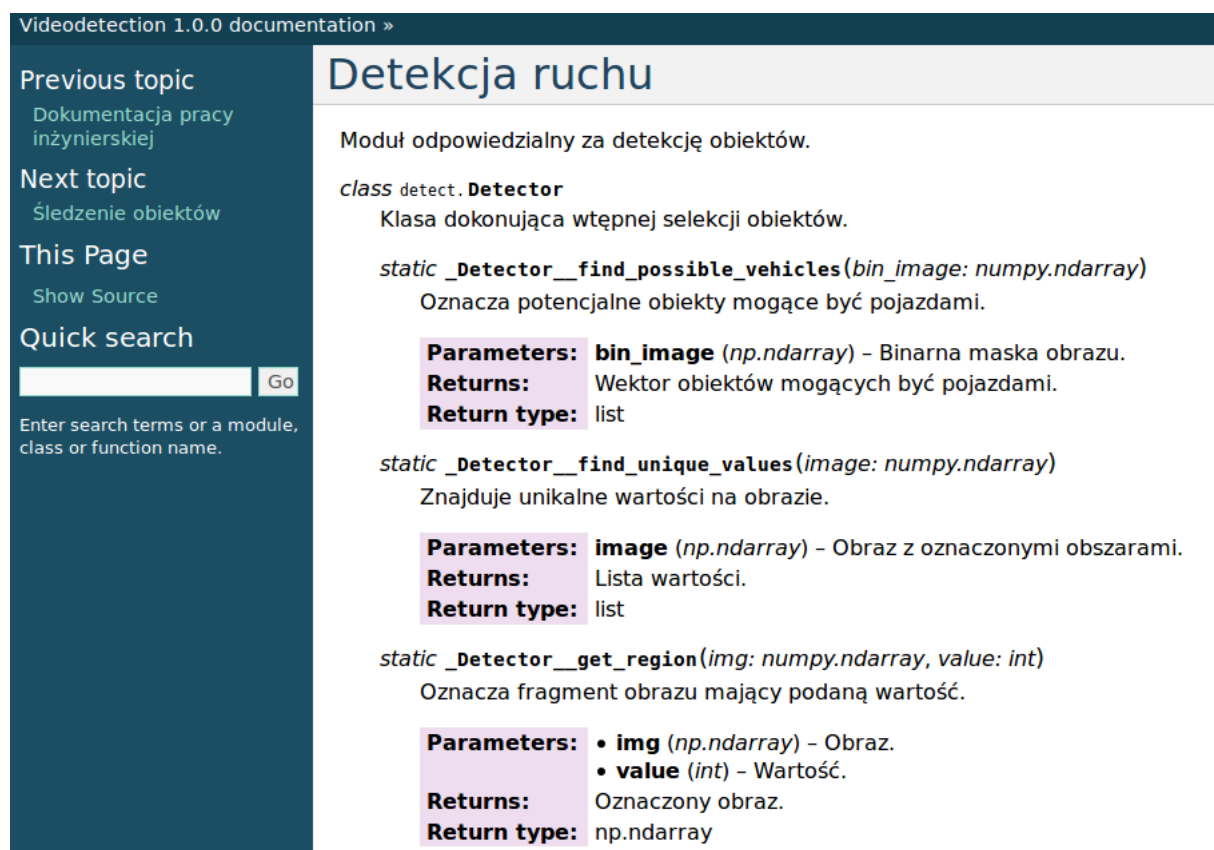


Rysunek 4.1. Widok okna programu Glade

## Sphinx

Sphinx jest to narzędzie służące do automatycznej generacji dokumentacji programu na podstawie komentarzy zawartych w kodzie. Program ten pozwala na tworzenie dokumentacji programów napisanych w Pythonie bądź C++. Aby dokumentacja mogła zostać

wygenerowana, komentarze muszą być zgodne ze standardem „reStructuredText”. Dokumentacja może być wygenerowana w postaci tekstu, htmla, w formacie pdf czy Latex.



Videodetection 1.0.0 documentation »

## Detekcja ruchu

Moduł odpowiedzialny za detekcję obiektów.

**class** `detect.Detector`  
Klasa dokonująca wstępnej selekcji obiektów.

**static** `_Detector__find_possible_vehicles(bin_image: numpy.ndarray)`  
Oznacza potencjalne obiekty mogące być pojazdami.

**Parameters:** `bin_image` (*np.ndarray*) – Binarna maska obrazu.  
**Returns:** Wektor obiektów mogących być pojazdami.  
**Return type:** list

**static** `_Detector__find_unique_values(image: numpy.ndarray)`  
Znajduje unikalne wartości na obrazie.

**Parameters:** `image` (*np.ndarray*) – Obraz z oznaczonymi obszarami.  
**Returns:** Lista wartości.  
**Return type:** list

**static** `_Detector__get_region(img: numpy.ndarray, value: int)`  
Oznacza fragment obrazu mający podaną wartość.

**Parameters:** • `img` (*np.ndarray*) – Obraz.  
• `value` (*int*) – Wartość.  
**Returns:** Oznaczony obraz.  
**Return type:** *np.ndarray*

Rysunek 4.2. Wygenerowana dokumentacja.

## Pycharm

Pycharm jest środowiskiem deweloperskim dla języka Python, tworzonym przez firmę JetBrains. Zapewnia ono szereg przydatnych funkcji, takich jak: podpowiadanie składni, detekcję błędów, proponowanie poprawek i rozbudowany debugger. Działa zgodnie, ze wszystkimi popularnymi implementacjami języka Python: CPython, IronPython, Jython. Środowisko oferuje wsparcie dla innych technologii, np. Html5, Css3, JavaScript, Angular.js. Za pomocą Pycharma programista może korzystać z frameworków webowych takich jak Django czy Flask. Środowisko wspiera również systemy kontroli wersji (Git, SVN), posiada wsparcie dla narzędzi do pracy z bazami SQL oraz umożliwia zdalne edytowanie kodu w oparciu o protokół SSH. Firma JetBrains udostępnia 3 wersje IDE: profesjonalną (płatną), edukacyjną (dla studentów) oraz „społecznościową”.



## 5. Opis programu

Poniższy rozdział opisuje interfejs i sposób działania zaimplementowanego programu.

### Instalacja

Poniżej został opisany sposób instalacji programu oraz wymaganych bibliotek na systemie Linux.

1. Wchodzimy na stronę, i pobieramy najnowszą wersję programu:

```
https://github.com/artven/videodetection/releases/latest
```

2. Przechodzimy do folderu zawierające pobrany plik:

```
cd ~/Pobrane
```

3. Rozpakowujemy i kopiujemy program do katalogu domowego:

```
unzip videodetection.v1.0.zip
mkdir ~/videodetection
cp -r cp -r 'videodetection v1.0'/* ~/videodetection/
cd ~/videodetection
```

4. Instalujemy zależności dla biblioteki OpenCV:

```
sudo apt-get update && sudo apt-get upgrade && sudo apt-get
dist-upgrade && sudo apt-get autoremove
sudo apt-get install build-essential cmake qt5-default
libvtk6-dev install zlib1g-dev libjpeg-dev libwebp-dev
libpng-dev libtiff5-dev libjasper-dev libopenexr-dev
libgdal-dev libdc1394-22-dev libavcodec-dev libavformat-
dev libswscale-dev libtheora-dev libvorbis-dev
libxvidcore-dev libx264-dev yasm libopencore-amrnb-dev
libopencore-amrwb-dev libv4l-dev libxine2-dev libtbb-dev
libeigen3-dev python-dev python-tk python-numpy python3-
dev python3-tk python3-numpy ant default-jdk doxygen
```

5. Pobieramy bibliotekę z oficjalnej strony:

```
cd /tmp
wget https://github.com/Itseez/opencv/archive/3.1.0.zip
unzip 3.1.0.zip
cd opencv-3.1.0/
```

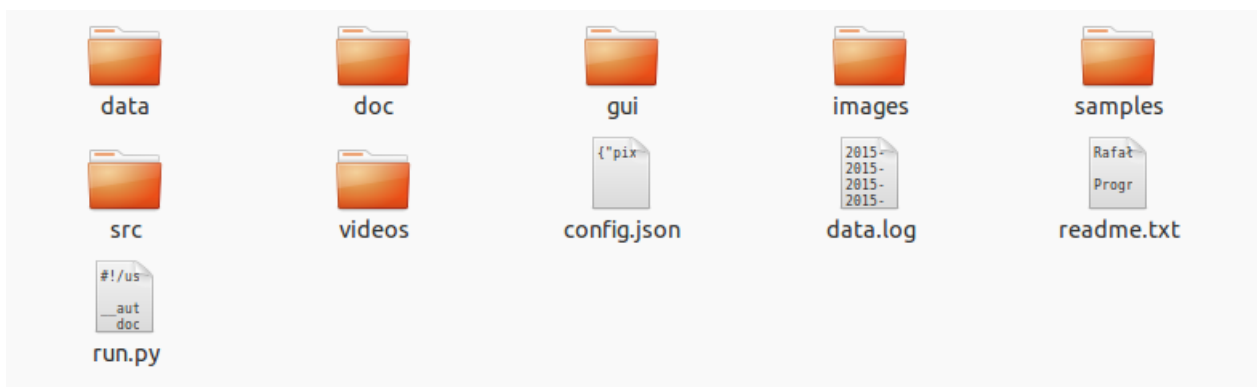
6. Kompilujemy i instalujemy OpenCV:

```
mkdir build
cd build
cmake -DWITH_QT=ON -DWITH_OPENGL=ON -DFORCE_VTK=ON -
      DWITH_TBB=ON -DWITH_GDAL=ON -DWITH_XINE=ON -
      DBUILD_EXAMPLES=ON ..
make -j4
sudo make install
sudo ldconfig
```

7. Instalujemy sklearn:

```
pip3 install -U scikit-learn
```

## Struktura programu



Rysunek 5.1. Widok struktury programu

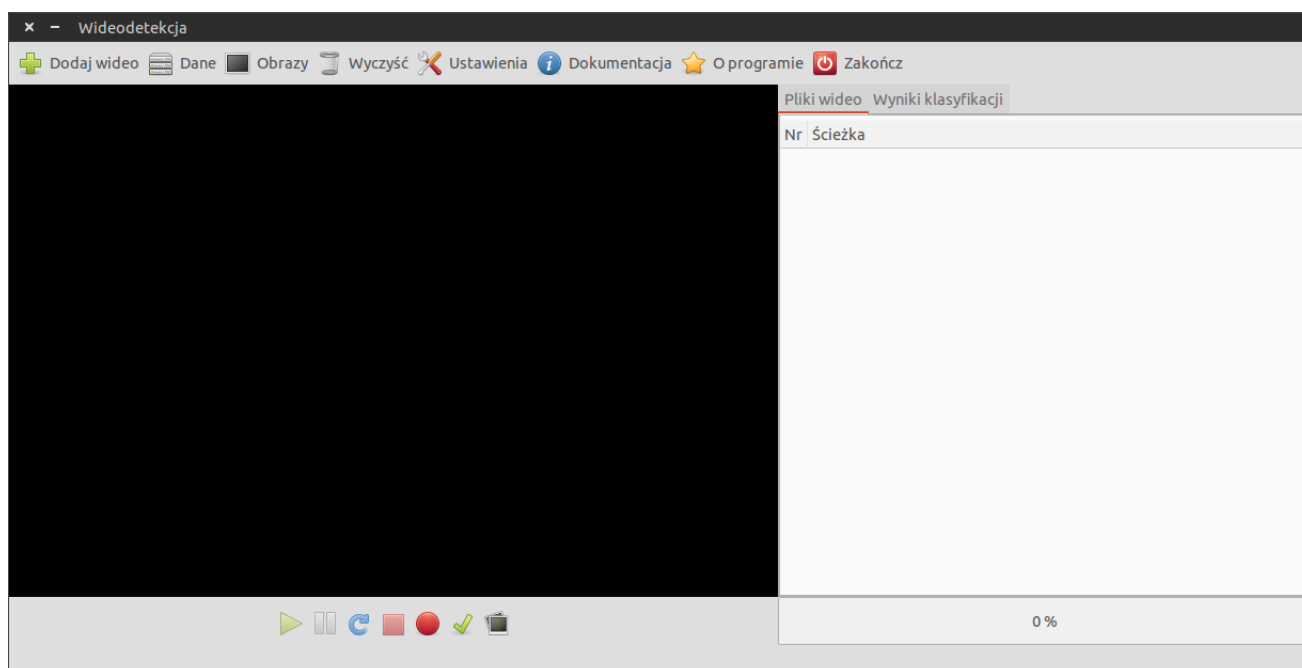
Program został podzielony na następujące pliki oraz foldery:

- data - folder do którego program zapisuje dane o wykrytych pojazdach do baz danych SQLite,
- doc - dokumentacja programu w formacie .html wygenerowana przy użyciu narzędzia Sphinx,

- gui - pliki w formacie .glade używane do wyświetlania interfejsu graficznego użytkownika,
- images - zdjęcia wykrytych samochodów,
- samples - przykładowe próbki nagrań ruchu samochodów,
- src - pliki źródłowe programu,
- videos - zapisane przez program wyniki analizy w postaci plików wideo,
- config.json - plik konfiguracyjny programu, zawiera parametry algorytmów oraz opcje dotyczące prezentacji wyników programu,
- data.log - plik gdzie są odkładane komunikaty diagnostyczne programu,
- readme.txt - plik tekstowy zawierający informacje o programie,
- run.py - skrypt uruchamiający program.

## Interfejs graficzny

Po uruchomieniu programu skryptem z pliku run.py pojawia się główne okno programu.

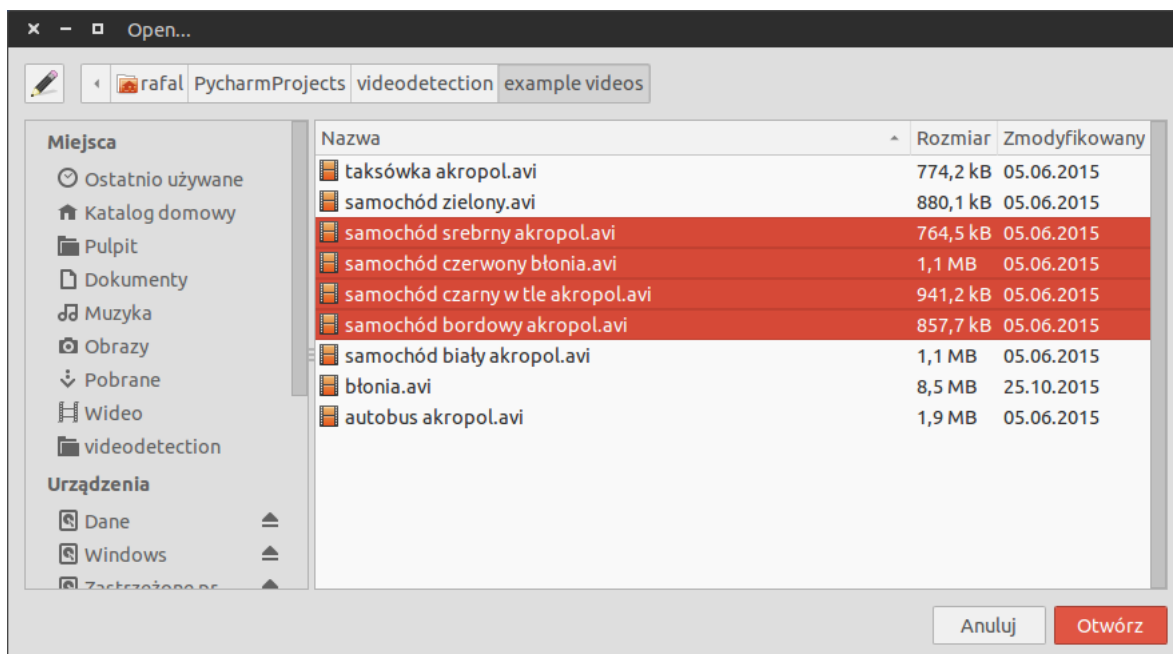


Rysunek 5.2. Główne okno programu

W skład głównego okna wchodzi górne menu, obszar odtwarzania pliku wideo, lista przetwarzanych plików, lista wyników klasyfikacji, menu odtwarzania i pasek informujący o postępie przetwarzania.

Funkcje elementów górnego menu:

- „Dodaj wideo” - otwiera poniższe okno dialogowe, umożliwiające wybór jednego bądź większej ilości plików wideo do analizy. Wybrane pliki są dodawane do listy po prawej stronie.



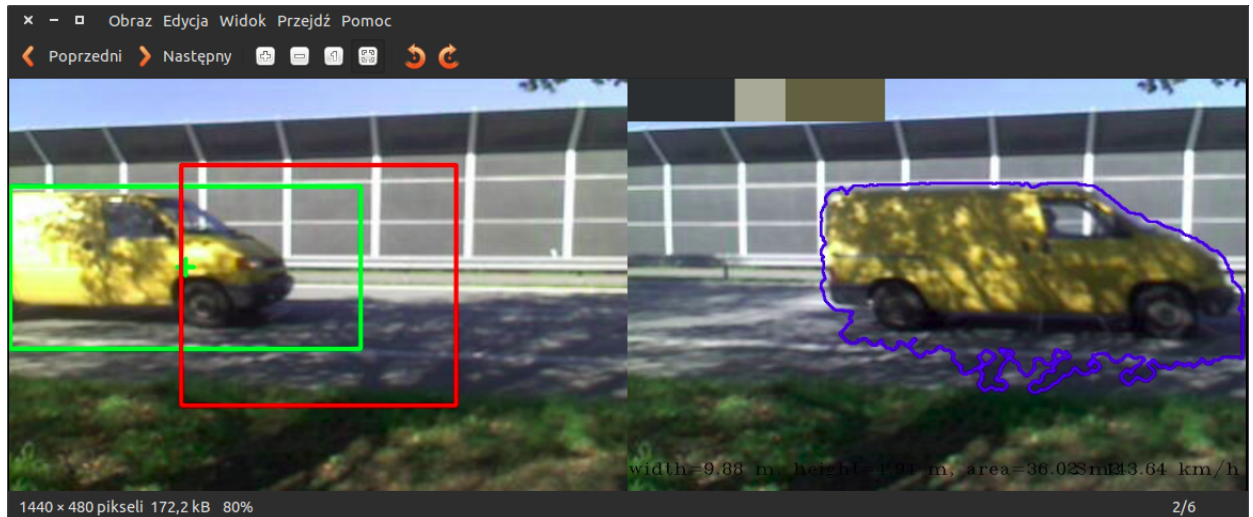
Rysunek 5.3. Okno wyboru plików

- „Dane” - umożliwia wybranie bazy danych i podejrzenie jej zawartości.

/home/rafal/PycharmProjects/videodetection/test.db						
Nr	Długość	Wysokość	Pole	Prędkość	Plik	Data
1	8.10 m	3.42 m	21.68 m2	165.24 km/h	M151001_124352.avi	2015-12-20 20:10:27
2	1.12 m	1.50 m	0.51 m2	570.24 km/h	M151001_124352.avi	2015-12-20 20:10:28
3	9.88 m	5.08 m	36.93 m2	143.64 km/h	M151001_124356.avi	2015-12-20 20:10:36
4	1.06 m	1.10 m	0.37 m2	207.36 km/h	M151001_124356.avi	2015-12-20 20:10:41
5	8.18 m	3.16 m	17.73 m2	154.98 km/h	M151001_124400.avi	2015-12-20 20:10:50
6	6.82 m	3.14 m	12.38 m2	82.08 km/h	M151001_124400.avi	2015-12-20 20:10:53
7	9.38 m	3.40 m	22.69 m2	172.08 km/h	M151001_124410.avi	2015-12-20 20:11:01
8	9.88 m	6.00 m	52.28 m2	75.91 km/h	M151001_124414.avi	2015-12-20 20:11:12
9	9.22 m	3.68 m	24.53 m2	12.86 km/h	M151001_124422.avi	2015-12-20 20:11:27
10	4.36 m	1.20 m	1.57 m2	326.16 km/h	M151001_124442.avi	2015-12-20 20:11:37
11	9.86 m	3.86 m	28.45 m2	576.72 km/h	M151001_124442.avi	2015-12-20 20:11:41
12	9.58 m	3.42 m	24.22 m2	175.68 km/h	M151001_124507.avi	2015-12-20 20:11:49
13	9.52 m	2.44 m	9.93 m2	546.48 km/h	M151001_124516.avi	2015-12-20 20:12:00
14	7.50 m	3.14 m	16.09 m2	150.66 km/h	M151001_124520.avi	2015-12-20 20:12:10
15	8.72 m	3.30 m	20.97 m2	194.40 km/h	M151001_124529.avi	2015-12-20 20:12:18
16	8.64 m	5.34 m	40.28 m2	99.00 km/h	M151001_124543.avi	2015-12-20 20:12:29

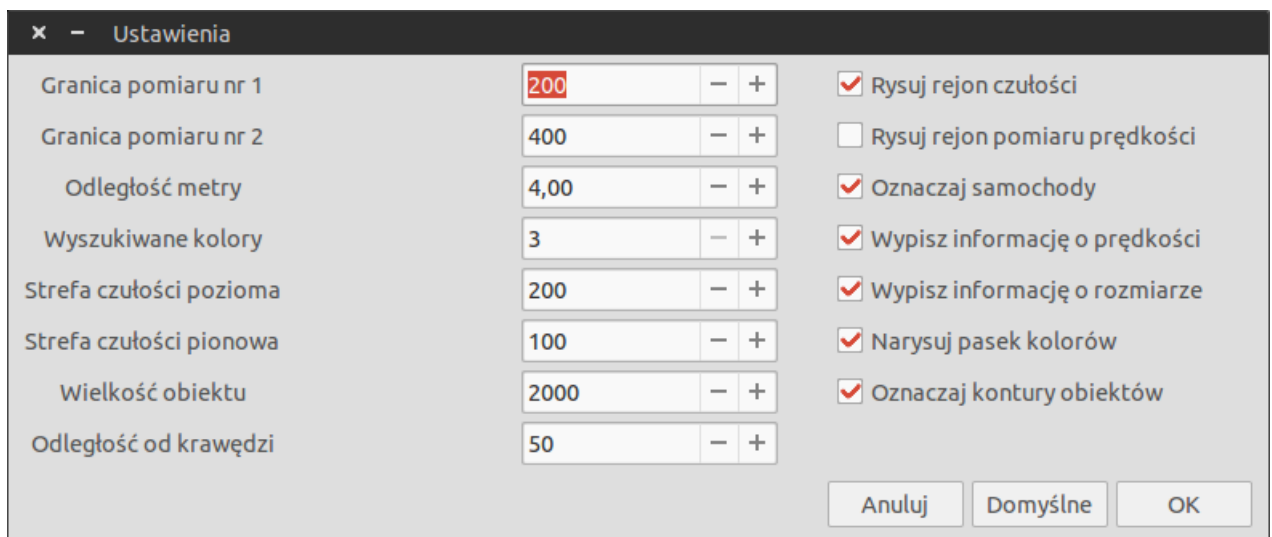
Rysunek 5.4. Podgląd bazy danych

- „Obrazy” - włącza przeglądarkę zapisanych przez program zdjęć pojazdów.



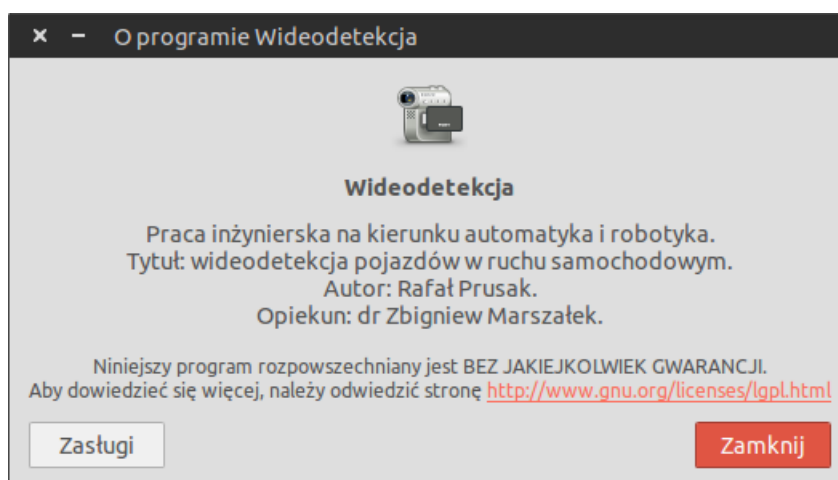
Rysunek 5.5. Widok przeglądania obrazów.

- „Wyczyść” - usuwa zdjęcia, filmy, bazy danych oraz wyniki klasyfikacji.
- „Ustawienia” -pokazuje okno ustawień programu.



Rysunek 5.6. Okno ustawień programu.

- „Dokumentacja” - uruchamia stronę html z dokumentacją.
- „O programie” - uruchamia okienko dialogowe pokazujące krótką informację o programie.



Rysunek 5.7. Informacja o programie

- „Zakończ” - wyłącza program.

## Ustawienia

Okno ustawień umożliwia dostosowywanie parametrów algorytmu oraz wizualizacji danych.

- „Odległość pixele” - wartość tego pola służy do wyliczynie przelicznika pixeli na obrazie na rzeczywistą odległość w metrach.
- „Odległość metry” - wartość tego pola ma również zastosowanie w przeliczaniu pixeli na metry. Procedura kalibracji pomiaru została opisana w poprzednim rozdziale
- „Liczba kolorów” - określa liczbę kolorów jaką program ma rozpoznać na obiekcie.
- „Strefa czułości pozioma” - określa ona obszar w poziomie, w którym obiekty są wykrywane i śledzone.
- „Strefa czułości pionowa” - określa ona obszar w pionie, w którym obiekty są wykrywane i śledzone.
- „Wielkość obiektu” - obiekt o mniejszej wartości są ignorowane po fazie binaryzacji
- „Odległość od krawędzi”
- „Rysuj rejon czułości” - oznacza na obszarze czerwonym prostokątem ekran będący strefą detekcji kamery.
- „Rysuj rejon pomiaru prędkości” - rysuje na obszarze detekcji dwa pionowe różowe paski służące do kalibracji przelicznika pixeli na metry, w celu obliczenia prędkości.
- „Oznaczaj samochody” - oznacza samochody zielonym prostokątem i krzyżykiem symbolizującym środek ciężkości pojazdu.

- „Wypisz informację o prędkości” - podpisuje obraz zidentyfikowanego pojazdu informacją o prędkości.
- „Wypisz informację o rozmiarze” - podpisuje obraz zidentyfikowanego pojazdu informacją rozmiarze: długości, wysokości, polu powierzchni bocznej.
- „Narysuj pasek kolorów” - rysuje na obrazie zidentyfikowanego pojazdu pasek informujący o kolorze tego pojazdu.

## 6. Algorytm detekcji i klasyfikacji pojazdów.

W tym rozdziale zostały opisane kolejne etapy zaimplementowanego algorytmu.

### Kalibracja programu

Przed rozpoczęciem analizy próbek wideo należy dobrać odpowiednie ustawienia. W tym celu istnieje możliwość wyłączenia przetwarzania obrazu przez algorytm, co znacząco polepsza wydajność programu na tym etapie. Pierwszym krokiem konfiguracji programu, powinno być dostosowanie rozmiaru strefy czułości pojazdu. Należy przy tym stosować się do następujących zasad:

1. Strefa czułości powinna być tak dobrana aby obejmowała obszar obrazu, będący faktycznym rejonem przejazdu pojazdów. Wybranie zbyt dużej strefy, obejmującej np. pobocze, tło, etc, może powodować pojawienie się błędnych odczytów spowodowanych klasyfikacją odbić światła i cieni jako poruszających się pojazdów.
2. Strefa powinna być na tyle wąska aby pojazdy były wykrywane dopiero gdy znajdują się całkowicie w kadrze. W przeciwnym razie tracona jest dokładność pomiaru prędkości.

Kolejnym etapem jest wprowadzenie granic pomiaru prędkości. Przed nagraniem danych z ruchu drogowego operator powinien dokonać ręcznego pomiaru odległości, następnie w trakcie konfiguracji programu oznaczyć zmierzony odcinek granicami pomiaru i podać zmierzoną wartość. Wartość ta stanowi podstawę przelicznika pixeli na obrazie na rzeczywistą odległość.

Ostatnim etapem powinno być dobranie minimalnego rozmiaru obiektu. Jest to ważna opcja, kluczowa podczas selekcji obiektów na obrazie. Umożliwia odrzucenie obiektów będących zakłóceniami. Minimalna wielkość obiektu powinna być dobrana z uwzględnieniem odległości urządzenia nagrywającego od jezdni - im urządzenie było bliżej szosy, dolna granica powinna być większa.

W programie, na chwilę obecną, nie ma możliwości ustalenia



Pobranie obrazu

Wyodrębnianie tła

Detekcja obiektów

Śledzenie pojazdów

Klasyfikacja

Zapis danych

## 7. Analiza wyników

## 8. Rozwój oprogramowania

### Nie zrealizowane funkcjonalności

W trakcie realizacji pracy inżynierskiej nie udało się autorowi zrealizować wszystkich pierwotnych założeń.

W trakcie pracy nad algorytmem okazało się, że detekcja pojazdów w ruchu na drodze dwupasmowej jest zadaniem bardzo trudnym, jeżeli program ma do dyspozycji jedynie obraz wideo z pobocza drogi. Zachodzi wtedy wiele „niekorzystnych” zjawisk:

- Mijanie się samochodów - w tej sytuacji dochodzi do zlewania się obiektów w jeden na obrazie binarnym a pojazdy mogą dotrzeć do przeciwnych końców drogi w odwrotnej kolejności. W celu zabezpieczenia programu przed wynikającymi z tej sytuacji przekłamaniami, należałoby dokonać poważnych zmian w algorytmie: zamiast pobierać informację o pojeździe na końcach obszaru „czułości” kamery, program musiałby śledzić obiekt w całym polu, wykryć kierunek ruchu i nadać mu szacowaną prędkość (przyrost pozycji w pixelach na kratkę obrazu).
- Wyprzedzanie się pojazdów - ta sytuacja może powodować, że dwa obiekty staną się dla programu jednym, jeżeli manewr zakończy się po za polem widzenia kamery. Skutkiem tej sytuacji będzie całkowita utrata wiadomości o obiekcie.
- Wzajemne rzucanie cienia i odbicia światła od samochodów - Powoduje to zniekształcenia na obrazie po wyodrębnianiu tła, takie jak połączenia się dwóch pojazdów w jeden w skutek odbicia światła, albo podział pojazdu na dwa niezwiązane ze sobą obiekty.

Rozwiązaniem powyższych problemów mogłoby być całkowite przemodelowanie algorytmu: odejście od prostej, ale wydajnej metody wyodrębniania tła na rzecz bardziej skomplikowanych algorytmów np. „cam shift”, „mean shift” czy „optical-flow”.

Kolejnym założeniem było zrealizowanie funkcjonalności do pracy z obrazem ruchu drogowego „on-line” streamowanym w sieci Internet. Niepowodzenie było spowodowane dwoma faktami :

- Znacząca większość dostępnych w sieci obrazów z kamer monitorujących filmuje ruch drogowy z pozycji „nad drogą” a nie z pobocza drogi.

- Udostępniany w sieci obraz jest obrazem w skali szarości, niskiej jakości i o bardzo małej ilości klatek, rzędu 2-3 na sekundę, co utrudnia detekcję poruszających się obiektów.

## Przyszłe zmiany w programie

W ciągu dalszego rozwoju oprogramowania możliwe byłyby następujące zmiany:

- Podział kodu na klasy i metody operujący na tych samych „poziomach” abstrakcji - kod programu miejscami jest niezrozumiały, kilka metod jest zdecydowanie za długie, mieszają się operacje o różnej abstrakcji, co skutkuje niestety w trudnościach w modyfikacji i dodawaniu do programu nowych funkcjonalności.
- Dodanie obsługi baz MySQL - dodanie tej funkcjonalności umożliwiłoby automatyczne dokonywanie pomiarów „w terenie” po przez użycie mini-komputera klasy RaspberryPi i kamery, natomiast obserwacja wyników mogłaby odbywać się zdalnie za pomocą strony WWW.
- Zbudowanie funkcjonalności streamowanie obrazu z programu na stronie WWW.
- Dodanie konfiguracji przez WWW.
- Dodatkowe opcje - ustawienia programu należało by poszerzyć o parametry wszystkich algorytmów.
- Zmiany w GUI - istnieje możliwość dodania do gui rysowania wykresów/opisywania statystyk przejechanych pojazdów.

## 9. Podsumowanie

A tutaj będzie podsumowanie pracy inżynierskiej, gdy już napiszę to gównie.

## Spis rysunków

4.1	Widok okna programu Glade . . . . .	13
4.2	Wygenerowana dokumentacja. . . . .	14
5.1	Widok struktury programu . . . . .	16
5.2	Główne okno programu . . . . .	17
5.3	Okno wyboru plików . . . . .	18
5.4	Podgląd bazy danych . . . . .	18
5.5	Widok przeglądania obrazów. . . . .	19
5.6	Okno ustawień programu. . . . .	19
5.7	Informacja o programie . . . . .	20

## Bibliografia