

**Wydział
Transportu**

POLITECHNIKA WARSZAWSKA

Zakład Telekomunikacji
w Transporcie

Praca dyplomowa inżynierska

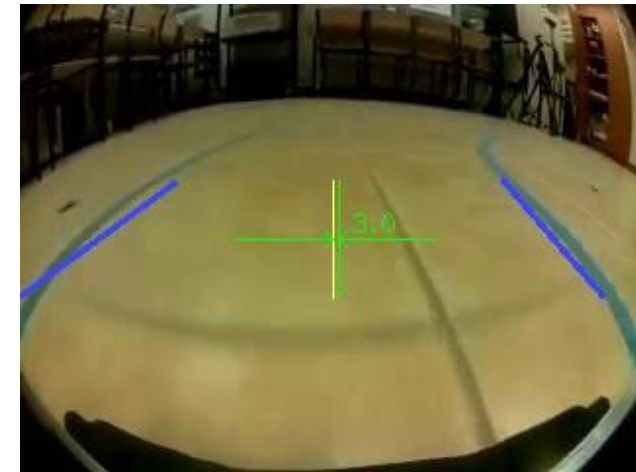
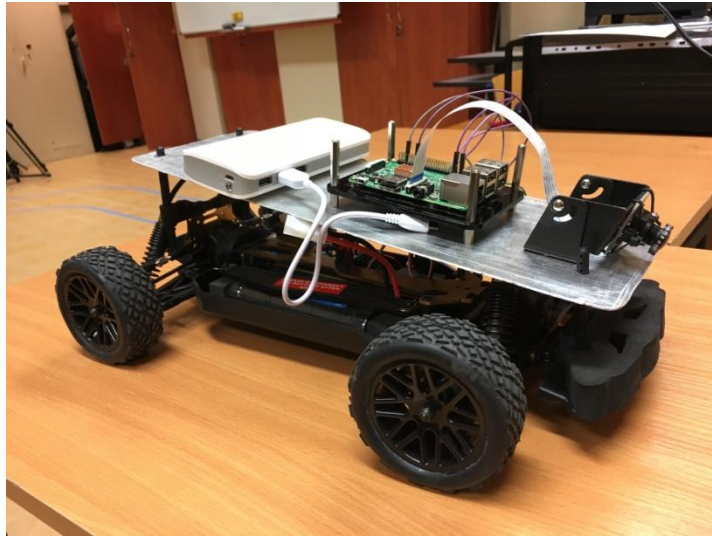
Temat: Projekt aplikacji do rozpoznania wizyjnego
wyszukiwania krawędzi pasa ruchu dla modelu
samochodu w skali

**Politechnika
Warszawska**

Autor: Serhiy Mytrovtsiy
Kierownik pracy: dr hab. inż. Maciej Kozłowski

Cel pracy

Celem pracy jest zapoznanie się z dostępnymi narzędziami do przetwarzania obrazu i zastosowania ich w branży samochodowej oraz opracowanie aplikacji do rozpoznania wizyjnego wyszukiwania krawędzi pasa ruchu dla samochodu w odpowiedniej skali.



Przegląd algorytmów i narzędzi programistycznych możliwych do wykorzystania

Języki programowania:

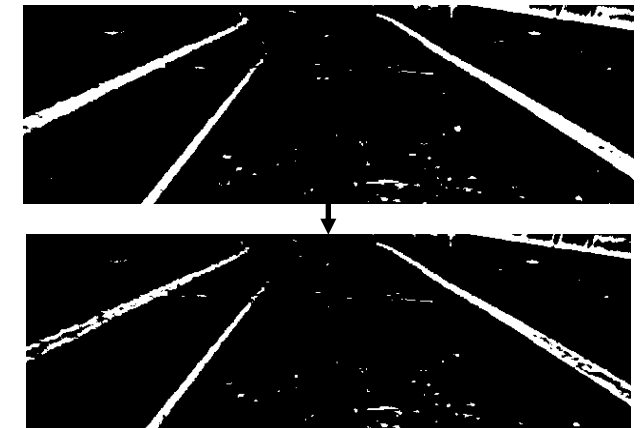
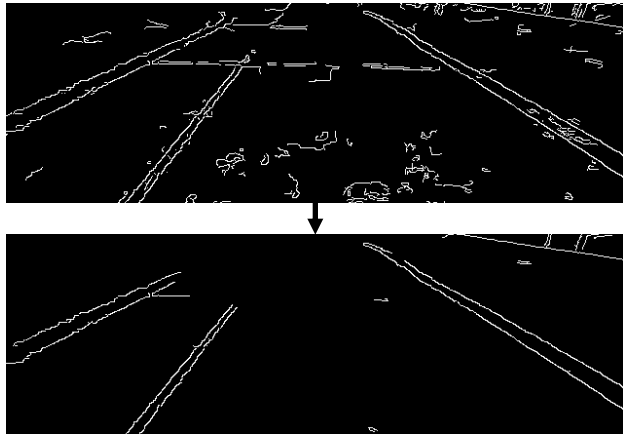
- C
- C++
- C#
- Python
- Go
- Java

Biblioteki:

- OpenCV
- AForge.Net
- VXL
- ImageJ

Algorytmy:

- Thresholding:
 - Global thresholding
 - Local thresholding
- Detektor krawędzi Canny
- Transformacja Hough
- Undistort
- Blur



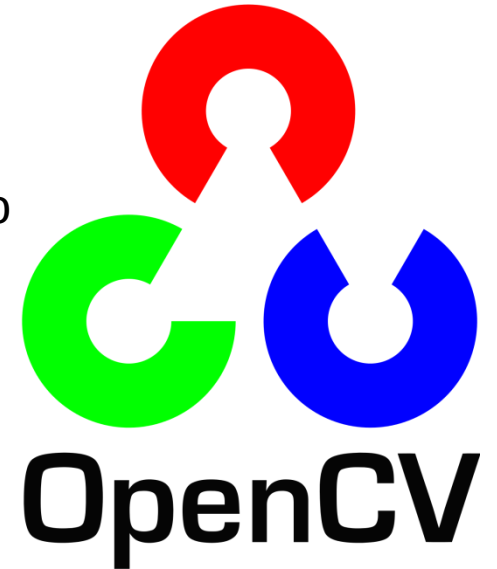
Wybór języka programowania oraz biblioteki

Język programowania, który został wybrany dla realizacji programu to Python (wersji 3.5.2).

W języku Python jest dostępnych kilka bibliotek do przetwarzania obrazu:

- OpenCV
- PIL
- Scikit-image

Tylko jedna z bibliotek ma zaimplementowane niezbędne algorytmy, które są potrzebne do przetwarzania obrazu w programie – OpenCV (wersji 3.2).



Cel i założenie projektu

Cel projektu: stworzyć program do rozpoznawania wizyjnego wyszukiwania krawędzi pasa ruchu dla samochodu w skali.

Założenie projektu:

- Program musi udostępniać dane o położeniu samochodu względem krawędzi;
- Program musi być niezależny od platformy;
- Program musi działać niezależnie od rozmiaru podawanego obrazu;
- Program musi umożliwiać przetwarzania minimalnie 60 fps (klatek na sekundę);
- Program musi sterować samochodem badawczym;
- Program musi włączać i wyłączać się za pomocą przycisku;
- Program musi składać się z oddzielnych modułów, które muszą pracować niezależnie;
- Program musi umożliwiać przesyłania obrazu na komputer zdalny dla przetwarzania (na żądanie);
- Program musi sygnalizować o gotowości do działania.

Algorytmy realizowane oraz struktura programu

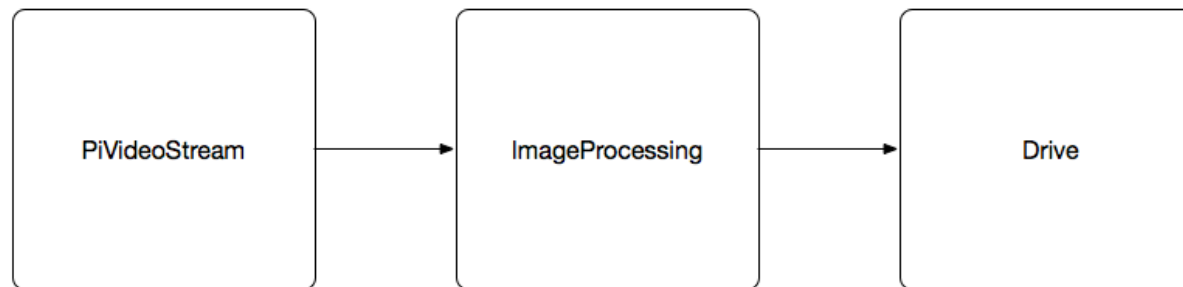
Zostały zaimplementowane dwa algorytmy dla wyznaczenia położenia pojazdu względem krawędzi pasa ruchu:

- Punkt docelowy;
- Algorytm własny.

Program składa się z trzech modułów:

- Moduł pobierania obrazu z kamery;
- Moduł przetwarzania obrazu;
- Moduł sterowania samochodem.

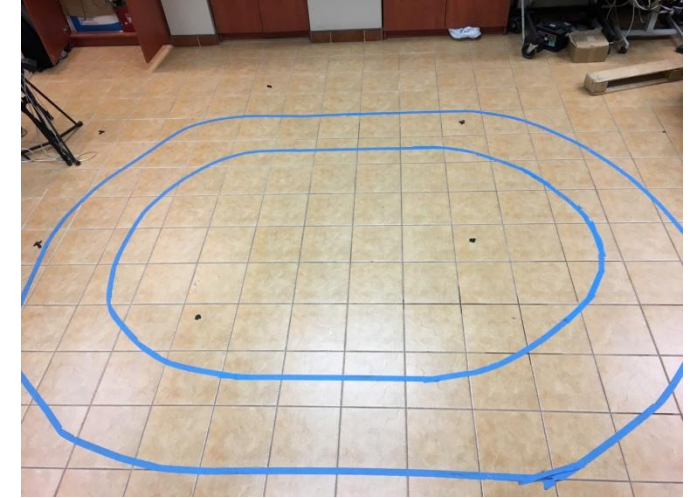
Każdy moduł jest niezależny i może być zastąpiony lub działać oddzielnie.



Stanowisko badawcze

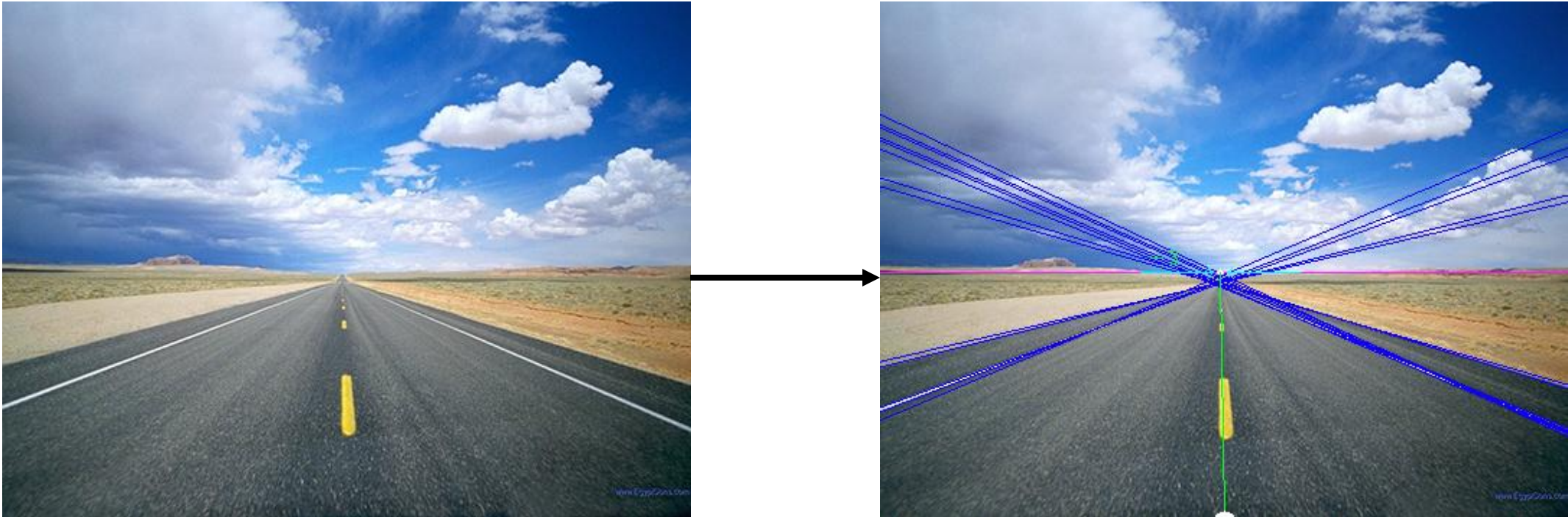
Stanowisko badawcze składa się z następujących elementów:

- Samochód w skali 1:10;
- Raspberry Pi 3 (komputer pokładowy);
- Kamera szerokokątna z kątem widzenia 170 stopni oraz matryca 5 MPx;
- Power bank o pojemności 8000 mAh dla zasilania komputera pokładowego
- Tor testowy.



Algorytm wyszukiwania punktu docelowego

Dany algorytm polega na tym, że na zdjęciu wyszukujemy wszystkie linie. Następnym krokiem jest odnalezienie punktu przecięcia jak największej liczby linii. Znaleziony punkt jest punktem docelowym. Wskazuje on na to, w którą stronę skręca pas ruchu oraz daje możliwość określenia położenia samochodu względem krawędzi.



Algorytm wyszukiwania punktu docelowego



Źródłowe zdjęcie



Canny edge detector



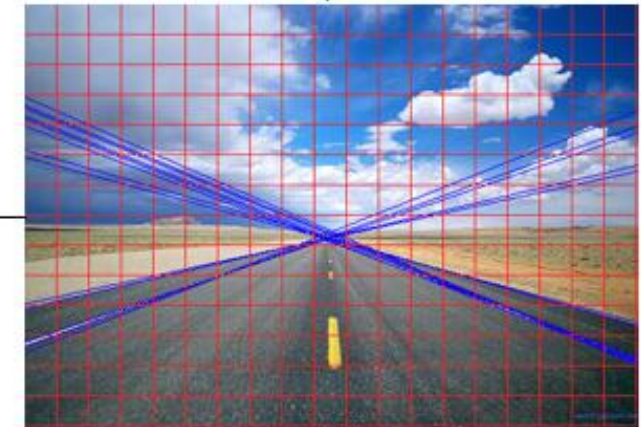
Transformacja Hough'a



Określenie różnicy między punktem przecięcia a środkiem zdjęcia



Dodanie punktu przecięcia na zdjęcie

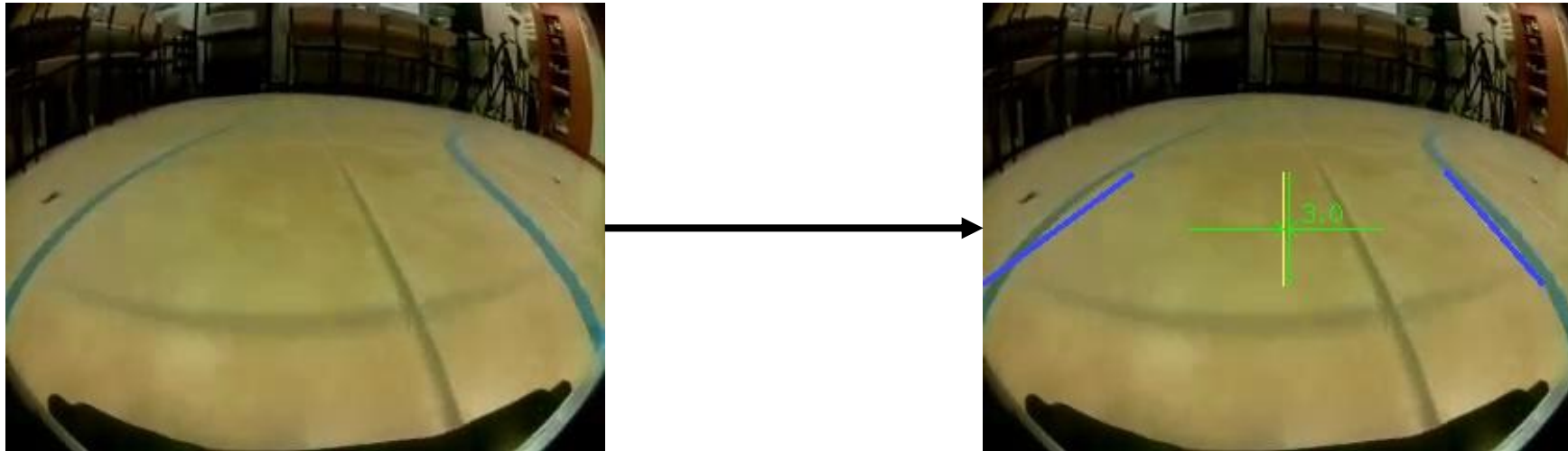


Budowa siatki dla odnalezienia punktu przecięcia linii

Algorytm własny

Dany algorytm składa się z 5 etapów:

- Obcięcie obrazu i pozostawienie tylko tego obszaru, który zawiera krawędzie. Zastosowanie efektu rozmycia;
- Usunięcie wszystkich kolorów, które nie mogą być krawędzią (czerwony, niebieski, zielony itd.);
- Usunięcie linii, które są mniejsze niż minimalnie możliwe żeby być krawędzią, farbowanie krawędzi w biały kolor oraz detekcja lewej i/lub prawej krawędzi;
- Wyznaczenie położenia samochodu na podstawie obliczenia odległości od lewej i/lub prawej krawędzi;
- Przenoszenie informacji na obraz źródłowy.



Algorytm własny



Zdjęcie źródłowe



Obcinanie obrazu i zastosowanie algorytmu Blur



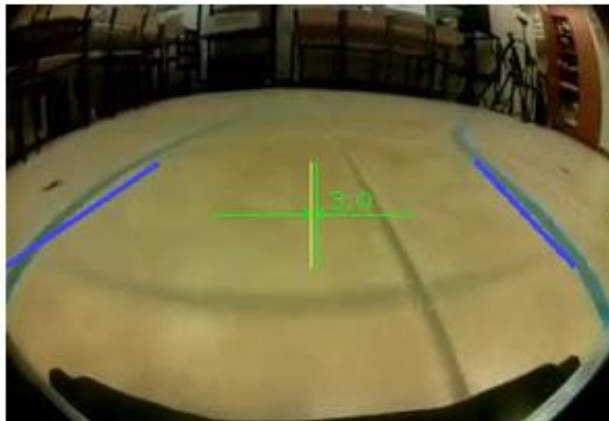
Usuwanie niepotrzebnych kolorów



Usuwanie niepotrzebnych linii oraz farbowanie krawędzi.
Poszukiwanie lewej i prawej krawędzi.



Rysowanie linii środkowej i odległości od
znalezionych krawędzi



Przenoszenie informacji na obraz źródłowy

Badanie

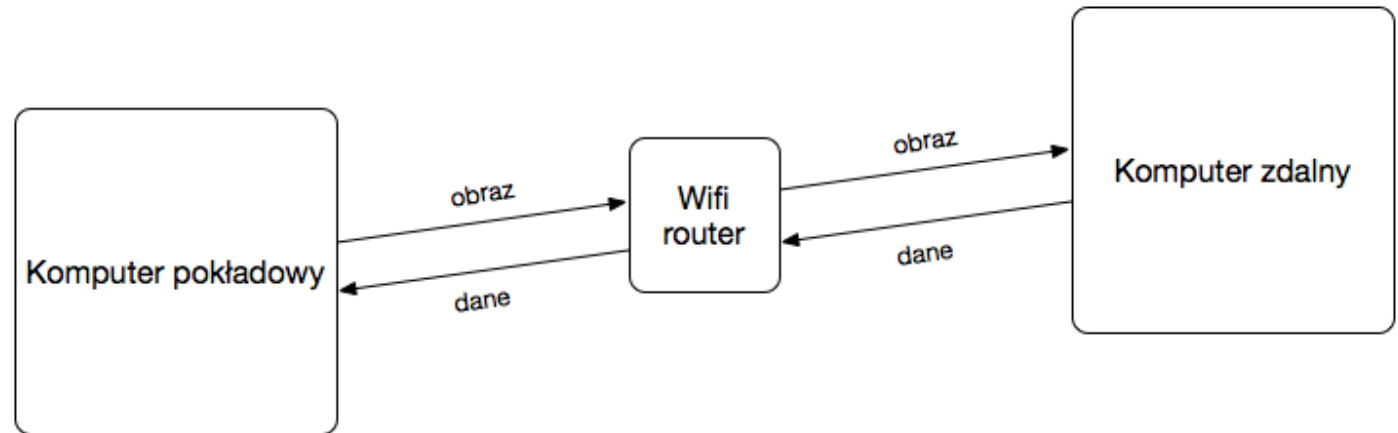
Badanie polega na sprawdzeniu, czy stworzony program spełnia założone wymogi. Przez porównanie dwóch zaimplementowanych algorytmów oraz zestawienie z wydajnością własnego algorytmu.

Porównanie wydajności polega na analizie/zestawieniu trzech wskaźników przy użyciu dwóch metod przetwarzania obrazu:

- Przetwarzanie obrazu na komputerze pokładowym;
- Wysyłka obrazu na komputer zdalny i wysyłanie z powrotem danych o położeniu samochodu przez sieć Wifi IEEE 802.11ac

Wskaźniki dla porównywania wydajności:

- Ilość klatek ze sterowaniem samochodem;
- Ilość klatek bez sterowania samochodem;
- Średni czas przetwarzania obrazu.



Podsumowanie

Stworzony program spełnia następujące założenia:

- ✓ Program musi udostępniać dane o położeniu samochodu względem krawędzi;
- ✓ Program musi być niezależny od platformy;
- ✓ Program musi działać niezależnie od rozmiaru podawanego obrazu;
- Program musi umożliwiać przetwarzania minimalnie 60 fps (klatek na sekundę);
- ✓ Program musi sterować samochodem badawczym;
- ✓ Program musi włączać i wyłączać się za pomocą przycisku;
- ✓ Program musi składać się z oddzielnych modułów, które muszą pracować niezależnie;
- ✓ Program musi umożliwiać przesyłania obrazu na komputer zdalny dla przetwarzania (na żądanie);
- ✓ Program musi sygnalizować o gotowości do działania.

Dziękuję za uwagę

Email: mitrovtsiy@ukr.net

Web: <https://github.com/exelban>