

---

## **Dokumentacja Projektu**

Kamera samochodowa z funkcją wykrywania obiektów i panelem konfiguracyjnym

---

**Autorzy:** Michał Dziedziak 263901, Michał Zychowicz 263950  
**Imię i Nazwisko prowadzącego kurs:** mgr inż. Tomasz Serafin  
**Dzień i godzina zajęć:** Wtorek NP, 7:30 - 11:00

# **Spis treści**

<b>1 Wstęp</b>	<b>3</b>
1.1 Opis projektu i systemu . . . . .	3
1.2 Analiza istniejących rozwiązań . . . . .	3
<b>2 Case studies</b>	<b>4</b>
2.1 Sytuacje obsługiwane przez system . . . . .	4
<b>3 Opis funkcjonalny</b>	<b>7</b>
<b>4 Architektura wysokopoziomowa systemu</b>	<b>8</b>
4.1 Opis . . . . .	8
4.2 Schemat . . . . .	8
<b>5 Architektura logiczna systemu</b>	<b>8</b>
5.1 Opis . . . . .	8
5.2 Schemat . . . . .	9
<b>6 Dobór technologii</b>	<b>9</b>
<b>7 Planowany zakres prac rozwojowych</b>	<b>11</b>
<b>8 Plan testów systemu</b>	<b>12</b>
8.1 Testy systemu . . . . .	12
8.2 Opis testowania funkcjonalności . . . . .	12
8.3 Źródła materiałów wideo . . . . .	12

# **Spis tabel**

1 Tabela przedstawiająca harmonogram pracy na przestrzeni kolejnych tygodni . . . . .	11
---	----

# **List of Listings**

# **Spis rysunków**

1 Linie pomocnicze są nakładane na obraz z kamery . . . . .	4
---	---

2	Wykryty pieszy zostaje oznaczony przez system . . . . .	5
3	Wykryty znak ostrzegawczy zostaje oznaczony przez system . . . . .	6
4	Wykryty samochód zostaje oznaczony przez system . . . . .	7
5	Schemat architektury wysokopoziomowej systemu: . . . . .	8
6	Schemat architektury logicznej systemu: . . . . .	9

# 1 Wstęp

## 1.1 Opis projektu i systemu

Projekt ma na celu opracowanie systemu obsługującego kamerę samochodową, wyposażoną w funkcję wykrywania określonych obiektów, z możliwością konfiguracji jej parametrów. Głównym zadaniem systemu jest analiza obrazu z kamery oraz sygnalizacja użytkownikowi wykrycia obiektu spełniającego określone kryteria. W ramach projektu przewiduje się wykrywanie takich obiektów jak: samochody, piesi i znaki drogowe.

W skład systemu wchodzi również edytor umożliwiający konfigurację oprogramowania wykonawczego. Edytor ten umożliwia kadrowanie obrazu, edycję linii, wybór obiektów do wykrywania oraz ustawianie parametrów wykrywania obiektów. Dodatkowo, użytkownik może ustawić metodę informowania o wykryciu obiektów, taką jak ramka wokół obiektu, alert na ekranie lub alert dźwiękowy.

System zapewnia obsługę zarówno nagrania wideo z kamery, jak i obrazu w czasie rzeczywistym. Dzięki temu użytkownik ma możliwość monitorowania otoczenia pojazdu w czasie rzeczywistym oraz analizy wcześniej zarejestrowanych nagrani.

## 1.2 Analiza istniejących rozwiązań

### FineVu GX1000

- Zalety
  - Wysoka rozdzielcość kamer: Quad HD + Quad HD
  - Szeroki kąt widzenia: 122° / 122°
  - Funkcja detekcji ruchu
  - Ostrzeżenia o fotoradarach, kamerach średniej prędkości i innych kontrolach drogowych
  - Auto Night Vision - tryb nocny i automatyczna poprawa obrazu w dzień
- Wady
  - Wysoka cena: 1349 zł lub 1399 zł w zależności od rozmiaru karty pamięci
  - Ryzyko przegrzania: W niektórych warunkach klimatycznych lub przy długotrwałym użytkowaniu istnieje ryzyko przegrzewania się urządzenia, co może wpływać na jego wydajność.
  - Brak funkcji wykrywania statycznych obiektów (pachołki, krawężniki itp.)

Link: <https://finevu.pl/kamery-samochodowe/finevu-gx1000/>

### M3S/M6S Infrared Night Vision Camera for Cars

- Zalety
  - Technologia termowizyjna: umożliwia użytkownikom obserwację otoczenia pojazdu w warunkach słabego oświetlenia i nawet w całkowitej ciemności.
  - Wykrywanie obiektów: Produkt może być używany do wykrywania obiektów na drodze, włączając w to zwierzęta, osoby poruszające się pieszo czy inne pojazdy, co może pomóc w unikaniu kolizji.
- Wady
  - Niska dostępność

Link: <https://www.infiray.com/products/m6s-infrared-night-vision-camera-for-cars.html>

## 2 Case studies

### 2.1 Sytuacje obsługiwane przez system

Tworzony system będzie w stanie obsługiwać następujące sytuacje:

- Rysowanie pomocniczych linii poziomych
  - Na obrazie z kamery rysowane są statyczne linie pomocnicze mające na celu wspomaganie kierowcy przy parkowaniu

Zdjęcie poniżej przedstawia przykładową wizualizację linii pomocniczych



Rysunek 1: Linie pomocnicze są nakładane na obraz z kamery

- Wykrywanie pieszych znajdujących się przed lub za samochodem
  - Po wykryciu pieszego w polu widzenia kamery system informuje o tym kierowcę w wybrany w konfiguratorze sposób.

Zdjęcie poniżej pokazuje przykład tego typu sytuacji:



Rysunek 2: Wykryty pieszy zostaje oznaczony przez system

- Wykrywanie znaku Stop oraz znaków z grupy znaków ostrzegawczych
  - Kiedy w polu widzenia kamery zostanie wykryty jeden z rozpoznawanych znaków kierowca zostanie o tym poinformowany w wybrany w konfiguratorze sposób.

Zdjęcie poniżej pokazuje przykład tego typu sytuacji:



Rysunek 3: Wykryty znak ostrzegawczy zostaje oznaczony przez system

- Wykrywanie innych samochodów
  - Kiedy system wykryje samochód w polu widzenia kamery, poinformuje on o tym kierowcę, w sposób wybrany w konfiguratorze.

Zdjęcie poniżej pokazuje przykład tego typu sytuacji:



Rysunek 4: Wykryty samochód zostaje oznaczony przez system

### 3 Opis funkcjonalny

System został zaprojektowany w celu skutecznego wykrywania i identyfikacji różnych obiektów na drodze, w tym znaków drogowych, pojazdów oraz pieszych. Oprócz tego, system oferuje dodatkowe możliwości, które pozwalają na bardziej kompleksową obsługę sytuacji drogowych. Poniżej przedstawiamy szczegółowy opis funkcjonalny systemu

- **Wykrywanie pojazdów:** System jest zdolny do wykrywania różnych typów pojazdów poruszających się na drodze, włączając w to samochody osobowe, ciężarówki oraz inne pojazdy.
- **Wykrywanie pieszych:** System jest wyposażony w funkcję wykrywania pieszych poruszających się wzduż drogi lub przechodzących przez nią.
- **Wykrywanie ostrzegawczych znaków drogowych:** System umożliwia identyfikację i klasyfikację ostrzegawczych znaków drogowych.
- **Alerty:** System może dawać znać o wykrytych obiektach na różne sposoby, takie jak ramka wokół wykrytego obiektu, czy ostrzeżenie dźwiękowe.
- **Rysowanie linii:** Na ekranie kamery rysowane są linie ułatwiające parkowanie.
- **Konfigurowalne parametry:** Użytkownik ma możliwość konfiguracji parametrów dotyczących wykrywania obiektów, rysowania linii oraz wyświetlania alertów. Może dostosować te parametry do swoich indywidualnych preferencji i potrzeb.

## 4 Architektura wysokopoziomowa systemu

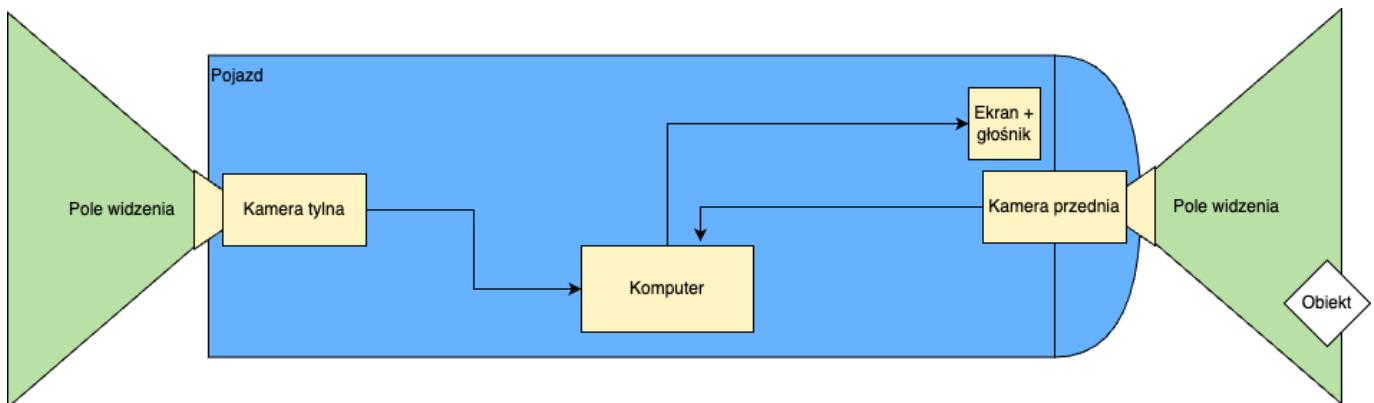
### 4.1 Opis

System zawiera następujące elementy:

- Kamera przednia
- Kamera tylna
- Komputer
- Ekran
- Głośnik

Obraz z kamer przesyłany jest do komputera, gdzie zostaje on przetworzony, oraz przeprowadzany jest proces wykrywania obiektów. Następnie komputer przesyła przetworzony obraz na ekran, oraz może uruchomić sygnał dźwiękowy jeśli wykryty zostanie określony typ obiektu.

### 4.2 Schemat



Rysunek 5: Schemat architektury wysokopoziomowej systemu:

## 5 Architektura logiczna systemu

### 5.1 Opis

Architektura logiczna systemu składa się z trzech modułów, które współpracują ze sobą w celu zapewnienia kompleksowej funkcjonalności systemu.

- **Konfigurator**

- Odpowiada za interakcję użytkownika z systemem poprzez interfejs graficzny. Użytkownik ma możliwość dokonywania modyfikacji ustawień dotyczących działania oprogramowania wykonawczego za pomocą interfejsu.
- Umożliwia wprowadzanie ustawień takich jak kadrowanie obrazu, edycja linii rysowanych na obrazie, wybór wykrywanych obiektów oraz parametryzacja procesu wykrywania i informowania o obiektach.

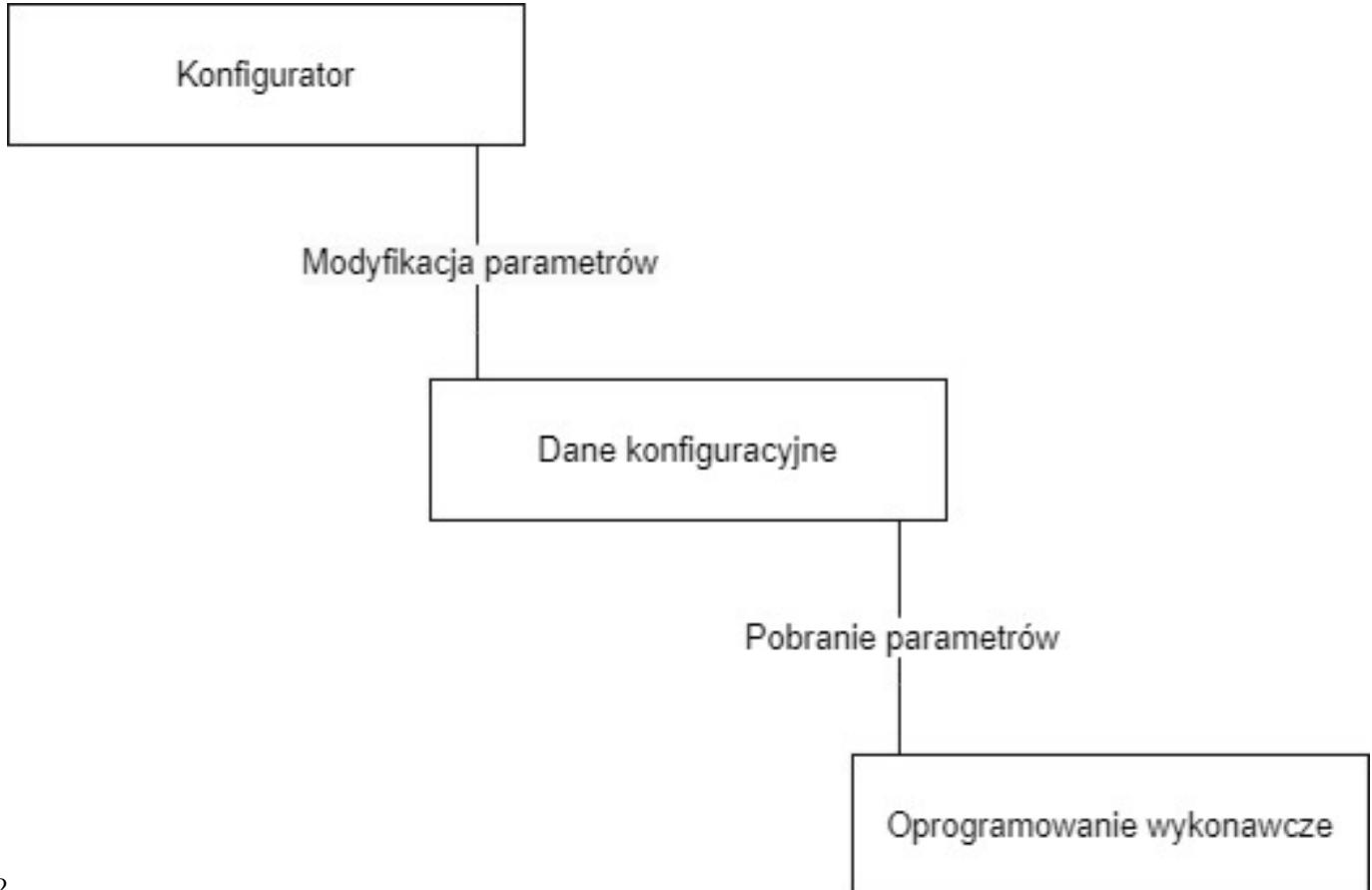
- **Dane konfiguracyjne**

- Stanowi pośrednią warstwę między konfiguratorem a oprogramowaniem wykonawczym. Jest odpowiedzialny za przechowywanie ustawień konfiguracyjnych.

- **Oprogramowanie wykonawcze**

- Jest odpowiedzialny za przetwarzanie obrazu z kamery oraz realizację funkcji związanych z wykrywaniem i identyfikacją obiektów na drodze.
- Podczas działania oprogramowania wykonawczego, na ekranie wyświetlany jest obraz z kamery lub nagranie wraz z elementami graficznymi, takimi jak linie i ostrzeżenia.

## 5.2 Schemat



/2

Rysunek 6: Schemat architektury logicznej systemu:

## 6 Dobór technologii

W niniejszym punkcie przedstawiamy listę wymagań technicznych dla tworzonego systemu. Przed rozpoczęciem implementacji oprogramowania kluczowe jest dokładne określenie środowiska, identyfikacja kluczowych komponentów technologicznych oraz uwzględnienie aspektów związanych z systemem operacyjnym, infrastrukturą sprzętową, bezpieczeństwem danych oraz wyborem technologii programistycznych.

### System Operacyjny

Oprogramowanie będzie kompatybilne z systemem operacyjnym Windows.

## **Wymagania Sprzętowe**

- Procesor: Intel Core i5 lub równoważny
- Pamięć RAM: Minimum 8GB
- Karta graficzna: Zintegrowana lub dedykowana wspierająca OpenGL
- System operacyjny: Windows 10 lub nowszy

## **Podstawowe Predykcje Wydajnościowe**

Oprogramowanie powinno działać płynnie na wymaganym sprzęcie, zapewniając responsywność interfejsu użytkownika oraz szybką obróbkę danych w czasie rzeczywistym.

## **Kwestie Sieciowe i Bezpieczeństwa**

- Oprogramowanie będzie działać w trybie lokalnym, nie wymagając połączenia z internetem.
- System Windows będzie wyposażony w aktualną ochronę antywirusową w celu zapewnienia bezpieczeństwa danych.

## **Wymagania dotyczące Kamer**

- Oprogramowanie będzie kompatybilne z różnymi modelami kamer kompatybilnymi z systemem Windows.
- Wsparcie dla standardowych rozdzielcości obrazu oraz prędkości klatek.

## **Język Programowania i Biblioteki**

- Oprogramowanie będzie napisane w języku Python.
- Do przetwarzania obrazu będzie wykorzystywana biblioteka OpenCV.
- Interfejs zostanie zaimplementowany przy użyciu biblioteki Tkinter.

## **Konteneryzacja**

Oprogramowanie zostanie skonteneryzowane przy użyciu narzędzia Docker, aby zapewnić łatwość w wdrażaniu oraz zarządzaniu środowiskiem aplikacyjnym.

## 7 Planowany zakres prac rozwojowych

numer	Tydzień		Zadania
	od	do	
1	2024-03-25	2024-03-31	- Omówienie tematu - Poszukiwanie pomysłów do realizacji w ramach projektu. - Wybór narzędzi i technologii
2	2024-04-01	2024-04-07	- Uszczegółowianie założeń projektowych - Tworzenie dokumentacji.
3	2024-04-08	2024-04-14	- Nauka obsługi wybranych narzędzi.
4	2024-04-15	2024-04-21	- Produkcja modelu zdolnego do wykrywania obiektów
5	2024-04-22	2024-04-28	- Produkcja modelu zdolnego do wykrywania obiektów
6	2024-04-29	2024-05-05	- Stworzenie szkieletu aplikacji z funkcjonalnością wykrywania obiektów
7	2024-05-06	2024-05-12	- Implementacja funkcjonalności rysowania linii pomocniczych
8	2024-05-13	2024-05-19	- Ogólna weryfikacja spełnienia założeń projektowych. - Kontrola jakości i poprawki związane z użytkowaniem aplikacji
9	2024-05-20	2024-05-26	- Napisanie i przeprowadzenie testów aplikacji
10	2024-05-27	2024-06-02	- Naprawianie błędów
11	2024-06-03	2024-06-09	- Wdrażanie poprawek

Tabela 1: Tabela przedstawiająca harmonogram pracy na przestrzeni kolejnych tygodni

### Tydzien 1.

Pierwszy tydzień zostanie poświęcony na poszukiwanie pomysłów do realizacji w ramach projektu. Po wybraniu kandydatów rozważymy narzędzia i technologie, które zostaną użyte w projekcie.

### Tydzien 2.

W drugim tygodniu uszczegółowimy założenia projektowe.

- Wybierzemy jakie dokładnie obiekty będą identyfikowane przez nasz system.
- Ustalimy w jak będzie informować kierowce o wykrytych obiektach.
- Rozważymy struktury aplikacji.
- Ustalimy koncept wyglądu aplikacji.

Następnie przystąpimy do tworzenia dokumentacji.

### Tydzien 3.

Trzeci tydzień poświęcimy na naukę obsługi wybranych narzędzi.

### Tydzien 4., 5.

W czwartym i piątym tygodniu skupimy się na produkcji modelu zdolnego do wykrywania interesujących nas obiektów. Pod koniec tego etapu będziemy mieli prostą aplikację wyświetlającą nagranie video z alertami o wykrytych obiektach.

### Tydzien 6.

Szósty tydzień poświęcimy na stworzenie szkieletu aplikacji. Będzie ona bliska wyglądem docelowej wersji i będzie miała zaimplementowany model wykrywający obiekty.

## Tydzień 7.

W siódmym tygodniu dodamy do aplikacji możliwość rysowania linii pomocniczych.

## Tydzień 8.

Podczas ósmego tygodnia zajmiemy się intuicyjnością i wygodą użytkowania aplikacji. Dokonamy szlifów interfejsu użytkownika.

## Tydzień 9.

W dziewiątym tygodniu przeprowadzimy testy aplikacji.

## Tydzień 10.

Dziesiąty tydzień poświęcimy na naprawianie błędów wykrytych podczas poprzedniego tygodnia.

## Tydzień 11.

Ostatni tydzień poświęcimy na wdrażanie poprawek zleconych przez klienta.

## 8 Plan testów systemu

### 8.1 Testy systemu

Testowane będą funkcjonalności systemu wyszczególnione w sekcji 3.

Testy będą przeprowadzane przez testera w sposób manualny.

### 8.2 Opis testowania funkcjonalności

- **Wykrywanie pojazdów** - Po poprawnym skonfigurowaniu systemu, tester uruchomi przykładowy materiał wideo. W trakcie oglądania go będzie obserwował czy pojazdy na nagraniu są poprawnie wykrywane.
- **Wykrywanie pieszych** - Po poprawnym skonfigurowaniu systemu, tester uruchomi przykładowy materiał wideo. W trakcie oglądania go będzie obserwował czy piesi na nagraniu są poprawnie wykrywani.
- **Wykrywanie ostrzegawczych znaków drogowych** - Po poprawnym skonfigurowaniu systemu, tester uruchomi przykładowy materiał wideo. W trakcie oglądania go będzie obserwował czy znaki drogowe na nagraniu są poprawnie wykrywane.
- **Alerty** - Tester, przy użyciu programu konfiguracyjnego, będzie po kolej wybierał wszystkie dostępne rodzaje alertów a następnie oglądając materiał testowy, będzie sprawdzał czy są one poprawnie uruchamiane.
- **Rysowanie linii** - Tester zaobserwuje, czy linie pomocnicze są nakładane na nagranie zgodnie z ustawieniami wybranymi w programie konfiguracyjnym.
- **Konfigurowalne parametry** Tester, wprowadzając zmiany w programie konfiguracyjnym, będzie sprawdzał czy są one poprawnie zapisywane.

### **8.3 Źródła materiałów wideo**

Materiały wideo na których przeprowadzane będą testy, będą pochodzić z własnych nagrań oraz z internetu. Kilka przykładowych nagrań z serwisu YouTube:

- <https://www.youtube.com/watch?v=PNDWr5FgBA8>
- <https://www.youtube.com/watch?v=DF2d4GqSNQ4>
- <https://www.youtube.com/watch?v=WC0GY7T3ZlM>