

Przetwarzanie Sygnałów i Obrazów

Projekt pt. Wirtualny parking

Prowadzący:

Dr inż. Wojciech Bieniecki

Autorzy:

Kamil Kaźmierczak 245839

Szymon Kobus-Puchała 245843

Wprowadzenie

Celem projektu było stworzenie inteligentnego systemu monitorowania i zarządzania parkingiem, który wykorzystuje kamery oraz algorytmy przetwarzania obrazu do analizy zajętości miejsc i ruchu na parkingu.

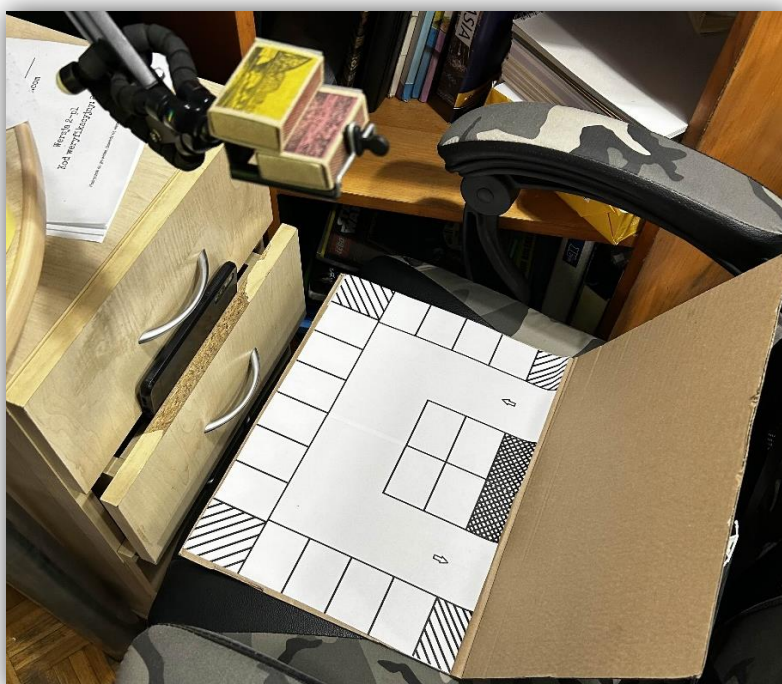
- System w czasie rzeczywistym rozpoznaje, czy miejsca są wolne czy zajęte.
- Dodatkowo system posiada moduł rozpoznawania tablic rejestracyjnych, który automatycznie weryfikuje wjeżdżające pojazdy.
- Logi z wjazdów, wyjazdów oraz statusów miejsc parkingowych są zapisywane w pliku tekstowym.

Projekt został zrealizowany w oparciu o kamery monitorujące makietę parkingu oraz skrypty w języku Python z użyciem bibliotek takich jak OpenCV i NumPy. Stanowi on praktyczny przykład wykorzystania technologii OCR i analizy obrazu w codziennym życiu.

Etapy realizacji projektu

Projektowanie i przygotowanie makiety

- Na etapie wstępnym została stworzona makietę parkingu z oznaczonymi miejscami parkingowymi.
- W makiecie uwzględniono kamerę ustawioną nad parkingiem, aby monitorowała cały obszar oraz kamerę skierowaną na wjazd i wyjazd z parkingu.



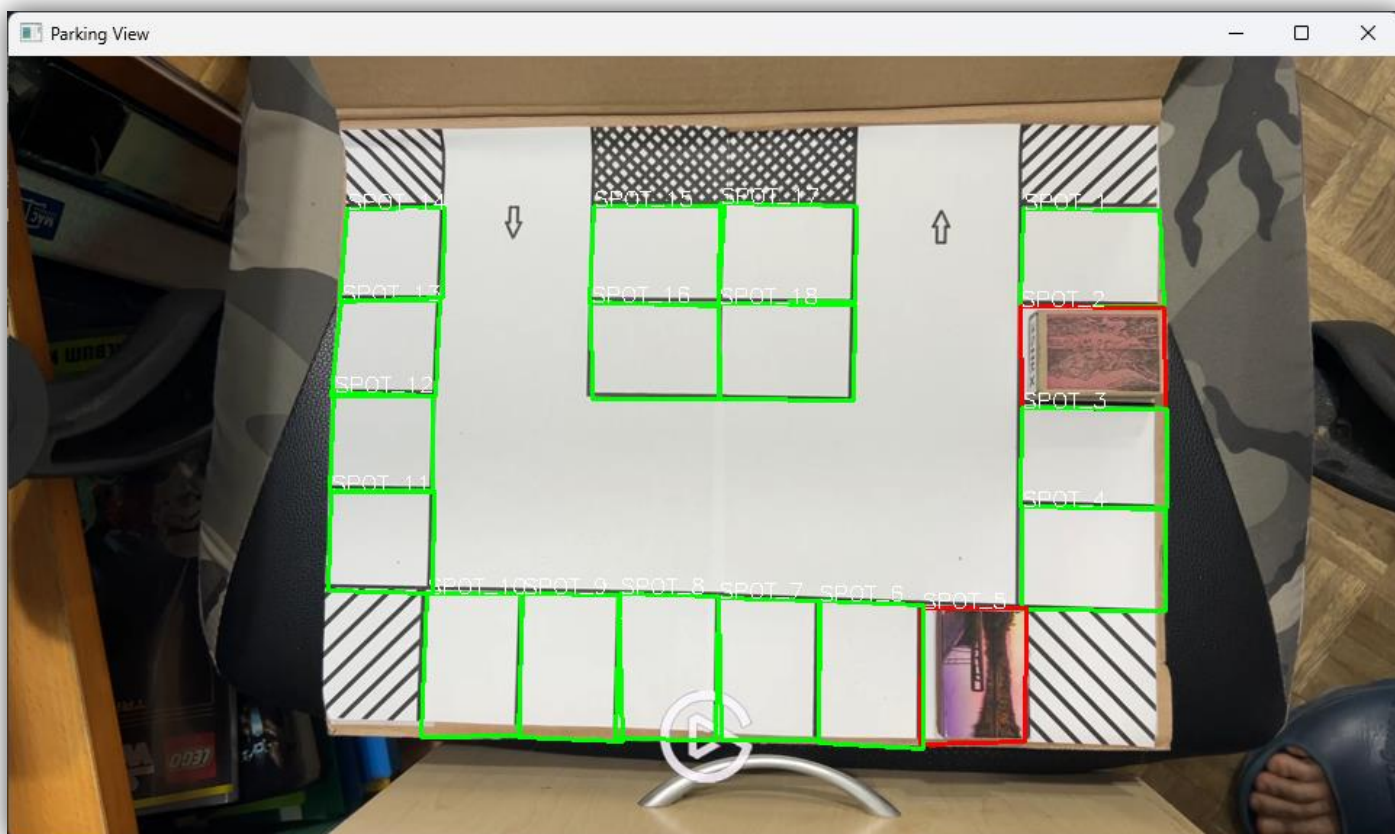
Tworzenie programu do oznaczania miejsc parkingowych

- Stworzono skrypt pozwalający użytkownikowi na interaktywne zaznaczenie miejsc parkingowych za pomocą myszy.
- Każde miejsce parkingowe zostało zapisane jako współrzędne punktów w pliku „parking_coords.json”, z identyfikatorami w formacie SPOT_X.

```
[
  {
    "id": "SPOT_1",
    "coordinates": [
      [
        690,
        73
      ],
      [
        792,
        76
      ],
      [
        790,
        146
      ],
      [
        688,
        143
      ]
    ]
  },
  {
    "id": "SPOT_2",
    "coordinates": [
      [
        471,
        138
      ],
      [
        568,
        141
      ],
      [
        565,
        213
      ],
      [
        468,
        210
      ]
    ]
  }
]
```

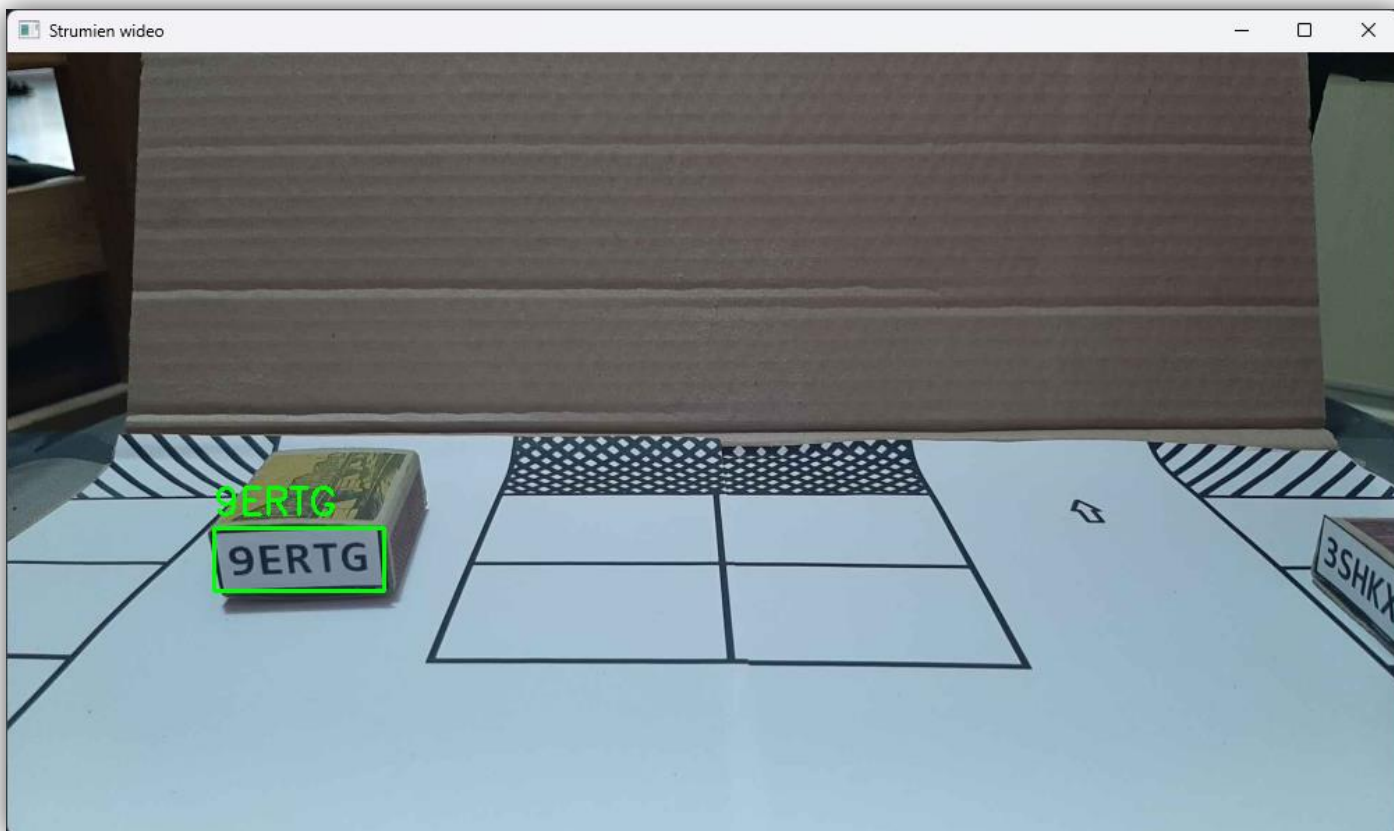
Wykrywanie zajętości miejsc parkingowych

- Na podstawie współrzędnych z JSON, program analizował zmiany w regionach odpowiadających miejscom parkingowym.
- Wykryto zajętość miejsca, jeśli różnica na obszarze wzrosła powyżej określonego progu.
- Dla lepszej wizualizacji wolne miejsca były oznaczane na zielono, a zajęte na czerwono.



System rejestracji wjazdów i wyjazdów

- Opracowano system do rozpoznawania tablic rejestracyjnych pojazdów z wykorzystaniem biblioteki do OCR (Tesseract).
- Stworzono plik tekstowy z listą numerów rejestracyjnych pojazdów uprawnionych do wjazdu. Program czytając dany numer rejestracyjny analizował czy znajduje się on na liście pojazdów uprawnionych do wjazdu.
- Program rejestrował wjazdy i wyjazdy pojazdów na podstawie wykrytych tablic i aktualizował listę zajętych miejsc parkingowych.



Logowanie zdarzeń

- W systemie wprowadzono logi, które rejestrowały:
 - Czas wjazdu i wyjazdu pojazdu
 - Identyfikator miejsca parkingowego
 - Status: wolne/zajęte
- Logi zapisywano w pliku tekstowym „parking_log.txt” w celu archiwizacji danych.

```
[2025-01-27 20:36:35] Miejsca na parkingu: 18
[2025-01-27 20:37:30] Auto z tablica 9ERTG wjechało na parking.
[2025-01-27 20:37:37] Miejsce SPOT_9 zajęte. Wolne miejsca: 17/18
[2025-01-27 20:37:48] Auto z tablica 3SHKX wjechało na parking.
[2025-01-27 20:37:52] Miejsce SPOT_13 zajęte. Wolne miejsca: 16/18
[2025-01-27 20:38:04] Miejsce SPOT_9 zwolnione. Wolne miejsca: 17/18
[2025-01-27 20:38:11] Auto z tablica 9ERTG wyjechało z parkingu.
[2025-01-27 20:38:15] Miejsce SPOT_13 zwolnione. Wolne miejsca: 17/18
[2025-01-27 20:38:20] Auto z tablica 3SHKX wyjechało z parkingu.
```

Napotkane problemy i ich rozwiązania

Dobór odpowiedniego algorytmu rozpoznawania tablic

- Na zajęciach poznaliśmy metody OCR jednak zastosowanie odpowiedniej w praktycznym rozwiązaniu okazało się problematyczne.
- **Rozwiązanie:** Przeszukanie informacji o skutecznych algorytmach OCR i znalezienie biblioteki Tesseract, która nadawała się idealnie.

Rozpoznawanie tablic rejestracyjnych

- OCR miał trudności z dokładnym odczytywaniem numerów tablic w przypadku niezłapanej ostrości przez autofokus w kamerze.
- **Rozwiązanie:** Dodanie tekstury za makietą sprawiało że ostrość skupiała się na niej i tablice były lepiej czytane. W przyszłości można rozważyć możliwość zainstalowania dodatkowej kamery (osobna na wjazd i wyjazd) jak w komercyjnych rozwiązaniach.

Organizacja współrzędnych miejsc parkingowych

- Dodawanie współrzędnych do pliku JSON okazało się problemem ponieważ JSON nie obsługuje krotek (tuple) i zamienia je w puste listy.
- **Rozwiązanie:** Trzeba było przekonwertować krotki na listy przed dodaniem ich do pliku.

Wyniki i wnioski

- System poprawnie rozpoznaje zajętość miejsc parkingowych oraz wjazdy i wyjazdy pojazdów.
- Logi umożliwiają analizę wykorzystania parkingu w czasie rzeczywistym.
- Program działa płynnie, choć wymaga odpowiednich warunków oświetleniowych dla skutecznego działania.

Podsumowanie

Projekt spełnił założenia i umożliwia skuteczne monitorowanie miejsc parkingowych oraz kontrolowanie wjazdów i wyjazdów pojazdów. Mimo napotkanych trudności udało się opracować stabilne rozwiązanie.