Uczenie Maszynowe

Projekt

Zespół:

Alan Kasperczak, 107265

Maciej Kowal, 107235

Jakub Wiśniewski, 107251

Bartłomiej Leśniak, 107236

Laboratoria prowadzone przez:

Dr inż. Natalia Piórkowska

WSB Merito

we Wrocławiu

Wrocław 23 czerwiec 2025r.

Spis treści

1 Słownik

Epoka – to jedno pełne przejście całego zbioru danych uczących przez algorytm, podczas którego każdy przykład jest używany do aktualizacji wag sieci neuronowej. W każdej epoce obliczany jest osobny gradient funkcji straty, który pozwala dostosować wagi, aby zminimalizować błędy. [1]

Funkcja straty - funkcja matematyczna pozwalająca określić różnicę między przewidywaniami modelu, a wartościami docelowymi. Stosowana w procesie optymalizacji do dopasowania parametrów uczenia. Wynikiem obliczenia tej funkcji jest wartość nazywana stratą. Jeżeli jej wartość zmniejsza się w miarę upływu kolejnych epok, oznacza to poprawę skuteczności uczenia modelu. [2]

Gradient - wektor pochodnych cząstkowych funkcji straty względem wag sieci neuronowej. Wskazuje kierunek, w którym należy zmienić wagi, aby zminimalizować funkcję straty. W każdej epoce, po obliczeniu gradientu na podstawie danych treningowych, wagi są aktualizowane w kierunku przeciwnym do gradientu. Im większa wartość gradientu, tym większe zmiany wag, co skutkuje szybszym osiągnięciem minimum funkcji straty.[3]

Krzywa uczenia - graficzna reprezentacja wydajności modelu podczas treningu. Poprzez jej analizę można zidentyfikować problemy takie jak niedouczenie lub przeuczenie. [4]

Przeuczenie – zjawisko, w którym model za bardzo dostosowuje się do danych treningowych. Traci przez to zdolność do generalizacji, co oznacza, że nie radzi sobie dobrze z nowymi, nieznanymi danymi. Model może osiągać wysoką dokładność na zbiorze treningowym, ale jego wydajność na danych testowych może być znacznie gorsza.

Niedouczenie - zjawisko, w którym model jest zbyt prosty, aby uchwycić istotne wzorce w danych treningowych. Niedouczenie prowadzi do niskiej dokładności na zbiorze treningowym, jak i na zbiorze testowym.

---

[1] M. Szeliga, “Praktyczne uczenie maszynowe”. Wydawnictwo PWN, Warszawa 2019

[2] https://pl.eitca.org/artificial-intelligence/eitc-ai-tff-tensorflow-fundamentals/tensorflow-in-google-colaboratory/building-a-deep-neural-network-with-tensorflow-in-colab/examination-review-building-a-deep-neural-network-with-tensorflow-in-colab/what-is-the-role-of-the-loss-function-and-optimizer-in-the-training-process-of-the-neural-network/ [dostęp: 24.01.2025]

[3] R. Tadeusiewicz, "Odkrywanie właściwości sieci neuronowych przy użyciu programów w języku C#". Polska Akademia Umiejętności, Warszawa 2007

[4] A. Król-Nowak, "Podstawy uczenia maszynowego". Wydawnictwo AGH, Kraków 2022

Walidacja – proces oceny wydajności modelu na innych danych niż te, użyte podczas treningu. Jej celem jest ocenić, jak dobrze model generalizuje na nowych danych oraz identyfikacja potencjalnych problemów, takich jak przeuczenie czy niedouczenie. [5]

Dataset - zbiór danych, używany do trenowania, walidacji i testowania modelu. Może się składać z np. Obrazów, plików audio, tekstu, danych liczbowych lub danych bazodanowych.

Model - wyspecjalizowany program komputerowy przeznaczony do analizowania danych, rozpoznawania wzorców i dokonywania przewidywań lub podejmowania decyzji na podstawie tych danych bez konieczności stosowania wyraźnych instrukcji ze strony człowieka w przypadku każdego scenariusza.

Etykieta - znacznik identyfikacyjny lub adnotacja dla punktów danych, wskazujący co konkretny fragment danych reprezentuje. Etykiety zapewniają kontekst, który pomaga modelom sztucznej inteligencji nauczyć się rozpoznawać wzorce i dokonywać prognoz na podstawie danych, na których są szkolone. [6]

SI (Sztuczna Inteligencja) – to techniczne rozwiązanie zaprojektowane przez ludzi, interpretujące zebrane dane oraz wyciągające z nich wnioski w celu wyboru najlepszych działań do podjęcia, aby osiągnąć wyznaczony cel. Systemy te wykonują czynności zazwyczaj wykonywane przez ludzi bądź wymagające ludzkiego intelektu. [7]

Sieć neuronowa – bardzo uproszczony model mózgu. Składa się ona z kilkuset do kilkudziesięciu elementów przetwarzających informacje. Elementy te nazywane są neuronami, jednakże w porównaniu do rzeczywistych komórek nerwowych mają bardzo uproszczone funkcje. Są one powiązane w sieć za pomocą parametrów modyfikowanych w trakcie uczenia, zwanymi wagami. [8]

YOLO (You Only Look Once) – Model sztucznej inteligencji o otwartym kodzie źródłowym, służący do analizy obrazu w czasie rzeczywistym. Analizuje obraz tylko raz, w celu wyznaczenia klasy i położenia obiektu.

OCR (Optical Character Recognition)– Optyczne rozpoznawanie znaków.

[5] https://www.deeptechnology.ai/walidacja-modelu-czy-jest-potrzebna/ [dostęp: 22.01.2025]
[6] Robert A. Kosiński, “Sztuczne sieci neuronowe. Dynamika nieliniowa i chaos”. Wydawnictwo Naukowe PWN, 2017
[7] Monografie Prwnicze, "Prawo sztucznej inteligencji". Wydawnictwo C.H.Beck Sp. z o.o., 2020
[8] R. Tadeusiewicz, "Sieci neuronowe". Akademicka Oficyna Wydawnicza, Warszawa 1993

IoU (Intersection over Union) - sposób zmierzenia, jak bardzo ramka rzeczywista i ramka przewidziana przez model pokrywają się.

ALPR (Automatic License Plate Recognition)- automatyczne rozpoznawanie tablic rejestracyjnych.

GUI (Graphical user interface) - graficzny interfejs użytkownika.

Preprocessing – proces mający na celu przygotować dane przed dalszym przetwarzaniem. Transformuje on surowe dane w formę odpowiednią dla algorytmu. [9]

Postprocessing – to działania podejmowane po przetworzeniu danych przez model, mający na celu poprawę wyników, interpretację danych lub dalszą analizę. Po przeanalizowaniu danych przez algorytm, wyniki można poddać postprocessingowi aby je zwizualizować lub wygenerować raport. [10]

RPI (Raspberry PI) - komputer jednopłytkowy, którego system jest oparty na Linuxie.

[9] https://ekordo.pl/uczenie-maszynowe-przygotowanie-danych-wprowadzenie/ [dostęp: 23.01.2025]
[10] https://langas.pl/kluczowe-terminy-dotyczace-zarzadzania-sztuczna-inteligencja/ [dostęp: 22.01.2025]

2 Cel i opis projektu

Program ma za zadanie zlokalizować tablicę rejestracyjną przy pomocy wyszkolonego specjalnie do użytku w projekcie modelu YOLOv8, a następnie odczytać ją używając EasyOCR. Dodatkowo połączona z projektem jest baza danych w której wpisane są "upoważnione" do wjazdu tablice rejestracyjne oraz logi wydarzeń przechowujące godzinę zdarzenia, odczyt tablicy rejestracyjnej oraz komentarz czy tablica jest zautoryzowana do wjazdu czy nie. Projekt przewiduje możliwość odczytu tablicy zarówno ze zdjęcia jak i w czasie rzeczywistym, używając kamery.

Według polskich przepisów, każdy pojazd o napędzie mechanicznym musi zostać zarejestrowany. Tablice rejestracyjne natomiast dzielą się na kilka rodzajów. Na każdej z nich znajduje się symbol Unii Europejskiej wraz z oznaczeniem kraju (w tym wypadku PL) oraz litery reprezentujące powiat albo województwo w połączeniu z mieszanką liczb i cyfr, służących jako identyfikator. Jednakże kolor tła oraz napisów różni się w zależności od wariantu tablicy. Standardowa tablica ma czarne znaki na białym tle. Tablica tymczasowa - czyli taka którą otrzymuje się na okres 30 dni np. w sytuacji, gdy sprowadzone auto nie przeszło jeszcze przeglądu w Polsce - ma znaki w kolorze czerwonym na białym tle. Rejestracja dla samochodu elektrycznego bądź napędzanego wodorem wyróżnia się zielonym kolorem tła (kolor znaków pozostaje czarny). Jeżeli auto jest w obiegu od dwudziestu trzech lat, otrzymuje ono tytuł zabytku i tablicę w kolorze żółtym (znaki ponownie pozostają czarne). Kolejnym rodzajem tablicy są tablice dyplomatyczne. Otrzymują je osoby wysłane na misje dyplomatyczne lub pojazdy należące do organizacji międzynarodowych. Cechują się one niebieskim tłem oraz białymi znakami oraz oznakowaniem; zamiast liter odpowiedzialnych za określenie powiatu, mają trzy cyfry reprezentujące kraj (np. USA - 001) i nie posiadają niebieskiego paska z logiem Unii Europejskiej. Osobną kategorią są rejestracje wojskowe, jednakże kolor tła oraz znaków pokrywa się ze standardową rejestracją[11]. W Polsce można jeszcze spotkać czarne tablice z białymi znakami. Wydawane były do roku dwutysięcznego, jednakże mogą być używane, jeżeli auto było zarejestrowane jeszcze podczas okresu ich użytkowania i ma wciąż miejsce na pieczątki z przeglądów w dowodzie rejestracyjnym.

[11] https://www.autodna.pl/blog/tablice-rejestracyjne-rodzaje-oznaczenia-skroty-i-inne/ [dostęp: 23.01.2025]

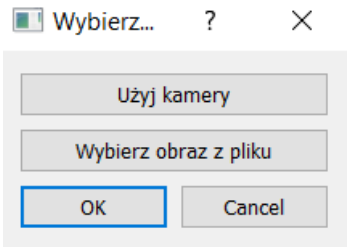
Dodatkowo tablice rejestracyjne muszą spełniać wymogi wymiarowe[12]. Podział jest następujący:

Tabela 2.1. Wymiary tablic rejestracyjnych używanych na terenie Polski

samochodowe jednorzędowe	520×114 mm
samochodowe dwurzędowe	305×214 mm
samochodowe zmniejszone	305×114 mm
motocyklowe	190×150 mm
motorowerowe	140×114 mm

Źródło: opracowanie własne.

Na przedstawionym poniżej zrzucie ekranu widać główne okno aplikacji do rozpoznawania tablic rejestracyjnych. Obszar oznaczony numerem jeden odpowiada za wyświetlenie obrazu z wybranego źródła. Po prawej stronie, w sekcji oznaczonej numerem dwa umieszczono logo Akademii Wojsk Lądowych. Tuż pod nią znajduje się pole oznaczone numerem trzy w którym program wyświetla odczytany numer rejestracyjny pojazdu. W obszarze numer cztery podana jest informacja czy tablica rejestracyjna znajduje się na liście upoważnionych do wjazdu, czy wręcz przeciwnie. Ostatnim elementem, oznaczonym numerem pięć, jest przycisk służący do wyboru źródła obrazu. Wywołuje on okienko (jak na Rys. 2.1.), w którym użytkownik może zdecydować, czy chce odczytać tablicę rejestracyjną z pliku, czy też użyć kamery do detekcji w czasie rzeczywistym.



Rysunek 2.1. Okno wyboru źródła obrazu

Źródło: opracowanie własne.

[12] Rozporządzenie Ministra Infrastruktury z dnia 31 sierpnia 2022 w sprawie rejestracji i oznaczania pojazdów, wymagań dla tablic rejestracyjnych oraz wzorów innych dokumentów zwiazanych z rejestracją pojazdów (Dz. U. 2022 poz. 1847)

Rysunek 2.2. Prezentacja GUI stworzonego dla aplikacji
Źródło: opracowanie własne.

3 Użyte narzędzia

PyCharm – narzędzie programistyczne do pisania w języku Python, obsługujące sporą liczbę przydatnych funkcji i ułatwień, takich jak edytor kodu, automatyczne uzupełnianie kodu, narzędzia do testowania oprogramowania oraz debugowanie. Program został stworzony przez JetBrains i jest jednym z popularniejszych narzędzi do programowania we wspomnianym języku. Podczas tworzenia projektu użyto wersji Professional, różniącej się od swojego darmowego odpowiednika między innymi ilością obsługiwanych framework'ów (np. Django), zintegrowanymi narzędziami do obsługi baz danych czy rozbudowanym wsparciem dla bibliotek naukowych. [13]

YOLOv8n (You Only Look Once) - Jest to wersja "nano" modelu wersji ósmej, cechująca się duża lekkością oraz szybkością, zachowując przy tym satysfakcjonującą dokładność. Jest to istotne przy użytkowaniu na urządzeniach z ograniczoną mocą obliczeniową. [14]

[13] https://www.jetbrains.com/products/compare/?product=pycharm&product=pycharm-ce [dostęp: 13.01.2025]
[14] https://github.com/ultralytics/ultralytics [dostęp: 23.01.2025]

EasyOCR – biblioteka open-source służąca do rozpoznawania tekstu w obrazach. Została napisana w Pythonie i wykorzystuje framework PyTorch do uczenia maszynowego. Wspiera domyślnie ponad 80 języków, pozwalając użytkownikom rozpoznawać tekst w róznych alfabetach. Cechuje się aktywną społecznością, która regularnie dzieli się nowymi przykładami i odpowiedziami na pojawiające się problemy. Gwarantuje on przyzwoitą dokładność rozpoznania, będąc przy tym lekkim i łatwym w konfiguracji. [15]

SQLite3 – lekka, wbudowana baza danych, oparta na języku SQL, cechująca się prostotą i brakiem konieczności konfiguracji serwera Cały jej mechanizm mieści się w jednej bibliotece, natomiast dane są przechowywane w jednym pliku na dysku. Pomimo swoich niewielkich rozmiarów SQLite udostępnia pełen zestaw funkcjonalności SQL.[16]

Stacja robocza – Laptop Lenovo ideapad Gaming o następujących parametrach technicznych:

- CPU – AMD Ryzen 7 4800H with Radeon Graphics
- Pamięć RAM – DDR4, 8 GB, 3200 MHz
- GPU – NVIDIA GeForce GTX 1650

Dataset – Udostępniony na platformie Kaggle w ramach licencji open-source zestaw danych użytych do wytrenowania modelu projektowego. Składa się z 25470 obrazów i ramek treningowych oraz 1073 walidacyjnych.

4 Porównanie różnych wersji YOLO

Pierwszym krokiem do wyboru modelu, który zostanie użyty w aplikacji było zapoznanie się z parametrami kilku różnych wersji YOLO. Rozważano wersję 11 oraz 8, ostatecznie decydując się na drugą z wspomnianych w wersji n. Powodem takiej decyzji był najlepszy czas przetwarzania obrazu, który osiągnął wspomniany model. Analizując poniższe porównanie można zauważyć, że YOLOv8 w konfiguracji “nano” wykazało się najszybszą prędkością przetwarzania obrazu (3.3 ms). Wynik ten jest korzystny dla aplikacji korzystających z wykrywania w czasie rzeczywistym. Dodatkowo konfiguracja

[15] https://github.com/JaidedAI/EasyOCR [dostęp: 23.01.2025]
[16] https://docs.python.org/3/library/sqlite3.html [dostęp: 23.01.2025]

ta jest bardzo lekka, co ułatwia jej użycie na słabszych pod względem obliczeniowym urządzeniach. [17]

Tabela 4.1. Wyniki testów dla YOLOv11

Konfiguracja	YOLO 11		
	preprocessing	Wnioskowanie	postprocessing
n	0.3	4.8	56.7
s	0.3	6.0	2.6
m	0.2	11.7	2.0
l	0.3	14.5	1.9
x	0.2	25.0	1.9

Źródło: Sapkota R. and Karkee M. Comparing YOLO11 and YOLOv8 for instance segmentation of occluded and non-occluded immature green fruits in complex orchard environment.

Tabela 4.2. Wyniki testów dla YOLOv8

Konfiguracja	YOLO 8		
	preprocessing	Wnioskowanie	postprocessing
n	0.3	3.3	2.5
s	0.3	5.1	3.6
m	0.3	10.1	2.4
l	0.3	14.6	1.9
x	0.3	20.8	2.0

Źródło: Sapkota R. and Karkee M. Comparing YOLO11 and YOLOv8 for instance segmentation of occluded and non-occluded immature green fruits in complex orchard environment.

5 Przedstawienie wyników szkolenia

W tej części dokumentu omówiono wyniki przeprowadzonego szkolenia modelu YOLOv8. Model ten szkolony był, używając datasetu open source, na który składało się 25470 obrazów treningowych oraz 1073 walidacyjnych.

5.1 Omówienie results.csv

Poniżej przedstawiono tabelę prezentującą wartości z pliku results.csv w której znajduje się zapis procesu szkolenia użytego modelu YOLO.

[17] R. Sapkota, M. Karkee, Comparing YOLO11 and YOLOv8 for instance segmentation of occluded and non-occluded immature green fruits in complex orchard environment. arXiv preprint arXiv:2410.19869, 2024.

Objaśnienie terminologii w kolumnach:

Epoch: numer epoki treningu;

Time: Czas który upłynął od rozpoczęcia szkolenia modelu, mierzony w sekundach;

Train/box_loss: Strata powiązana z predykcją ramki podczas treningu. Im niższa wartość, tym lepsza dokładność lokalizacji obiektu;

Train/cls_loss: Strata klasyfikacyjna. Określa jak dobrze model przewiduje klasę w zbiorze, na którym był trenowany;

Train/dfl_loss: Strata związana z jakością predykcji lokalizacji obiektu. Im niższa wartość, tym lepsza dokładność;

Metrics/precision(B): Obliczona dla danych treningowych miara precyzji. Określa proporcję trafionych predykcji pośród wszystkich pozytywnych predykcji;

metrics/recall(B): Miara czułości. Pokazuje proporcję poprawnych detekcji względem wszystkich pozytywnych przykładów;

metrics/mAP50(B): Średnia precyzja dla progu IoU=0.5. Standardowa miara oceny detekcji;

metrics/mAP50-95(B): Średnia precyzja dla różnych progów IoU. Pozwala kompleksowo ocenić skuteczność detekcji;

val/box_loss: Strata związana z ramkami w zbiorze walidacyjnym. Nizsza wartość oznacza lepszą zdolność do generalizacji;

val/cls_loss: Strata związana z klasyfikacją w zbiorze walidacyjnym. Pozwala ocenić jak dobrze model przewiduje etykiety klas wcześniej niewidzianych danych;

val/dfl_loss: Strata związana z jakością lokalizacji dla danych walidacyjnych. Im niższa wartość tym lepsza jakość lokalizacji;

Lr/pg0, lr/pg1, lr/pg2: Współczynniki uczenia dla różnych grup parametrów. Wskazują prędkośc optymalizacji wag modelu podczas szkolenia.

Analizując poniższe tabele, można wywnioskować, że w miarę upływu epok model poprawia się. Precyzja osiąga przyzwoity wynik 83,7% w porównaniu do średnich wyników wynoszących od 75% do 85%. Czułość z kolei przekroczyła średnią, wahającą

się od 65% do 75%, osiągając wynik 76,6%. Wartość mAP50 (80,7%) mieści się w kategorii dobrze działających modeli, których zakres to 75%–85%. Jednakże model wciąż może zostać udoskonalony — wartość mAP50-95 (42,3%), pomimo że akceptowalna i wskazująca na umiarkowaną skuteczność, odbiega od dobrych wyników, które powinny wynosić od 50% do 60%.

Tabela 5.1. Przedstawienie wyników szkolenia z pliku results.csv

epoch	time	train/box_loss	train/cls_loss	train/dfl_loss	metrics/precision(B)	metrics/recall(B)	metrics/mAP50(B)	metrics/mAP50-95(B)
1	83.1956	1.27205	1.2591	1.15736	0.68813	0.54609	0.58145	0.26258
2	161.31	1.25696	0.71137	1.17891	0.71981	0.61157	0.6328	0.28498
3	238.157	1.23096	0.66642	1.17332	0.76688	0.60648	0.65442	0.28942
4	315.775	1.20263	0.61513	1.16058	0.78296	0.60521	0.67242	0.32444
5	394.736	1.17839	0.57413	1.14731	0.79282	0.67769	0.72923	0.35623
6	478.691	1.1558	0.54325	1.13709	0.85028	0.68786	0.74846	0.376
7	560.528	1.13758	0.51322	1.12394	0.83609	0.72155	0.7655	0.39076
8	636.758	1.1151	0.48826	1.10882	0.82644	0.74698	0.78773	0.40231
9	713.017	1.09799	0.46718	1.1019	0.83644	0.74317	0.80225	0.42168
10	789.124	1.08121	0.44547	1.0888	0.84161	0.76003	0.80659	0.42167

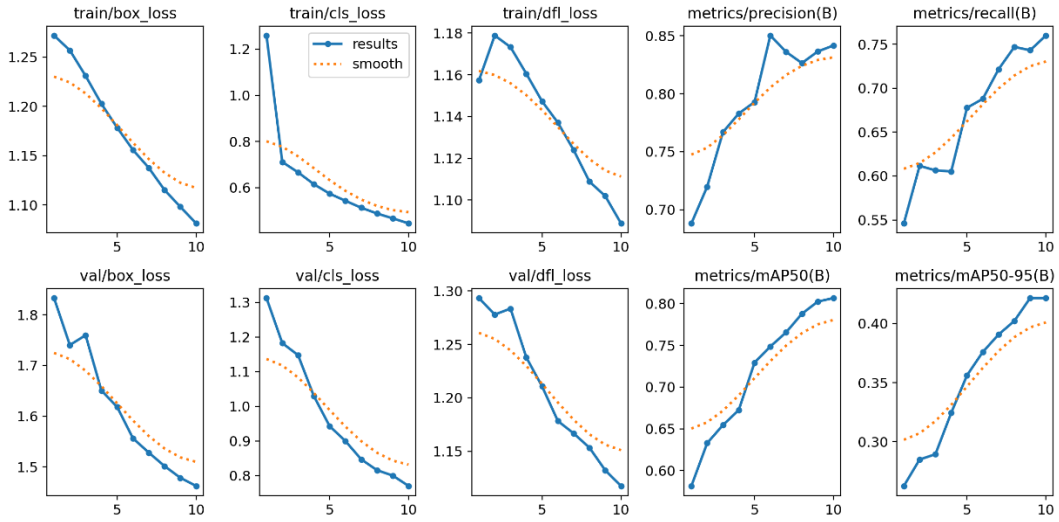
Źródło: opracowanie własne.

Tabela 5.2. Przedstawienie wyników szkolenia z pliku results.csv, ciąg dalszy

epoch	val/box_loss	val/cls_loss	val/dfl_loss	lr/pg0	lr/pg1	lr/pg2
1	1.83277	1.31288	1.29331	0.000666248	0.000666248	0.000666248
2	1.74013	1.18276	1.27788	0.00120096	0.00120096	0.00120096
3	1.75958	1.14796	1.28343	0.00160366	0.00160366	0.00160366
4	1.65062	1.02978	1.23783	0.001406	0.001406	0.001406
5	1.61799	0.94289	1.2107	0.001208	0.001208	0.001208
6	1.5564	0.89989	1.17836	0.00101	0.00101	0.00101
7	1.52826	0.84715	1.16679	0.000812	0.000812	0.000812
8	1.5017	0.81584	1.15342	0.000614	0.000614	0.000614
9	1.47864	0.80018	1.13232	0.000416	0.000416	0.000416
10	1.46257	0.77029	1.11767	0.000218	0.000218	0.000218

Źródło: opracowanie własne.

W celu zwiększenia przejrzystości danych, poniżej zamieszczono wykresy wizualizujące ich wartości.



Rysunek 5.1.Wykresy wizualizujące wartości z pliku results.csv, przedstawione w tabelach 5.1. i 5.2. Źródło: opracowanie własne.

Po analizie zamieszczonych wyżej wykresów zauważono, że:

train/box_loss: Pokazuje zmniejszającą się stratę dla ramek w trakcie treningu. Można odnotować systematyczny spadek wraz ze wzrostem epok. Oznacza to, że model poprawnie uczy się lokalizować obiekty.

train/cls_loss: Obrazuje stratę dla klasyfikacji obiektów podczas treningu. Podobnie jak w przykładzie wcześniejszym, tutaj również można zaobserwować trend malejący, co oznacza, że model coraz lepiej rozpoznaje klasy obiektów.

train/dfl_loss: Dotyczy strat związanych z lokalizacją. Wartość maleje, co świadczy o coraz lepszej precyzji w określaniu pozycji ramek. Przyglądając się jednak pierwszym epokom, zobaczyć można wzrost tej wartości. Zjawisko to spowodowane jest najprawdopodobniej eksploracją wag przez model. Wszystko jednak jest w normie, tak długo jak w późniejszych epokach wartość spada.

metrics/precision(B): Precyzja modelu dla danych walidacyjnych poprawia się wraz z kolejnymi epokami. Wykres rośnie, co oznacza, że model coraz lepiej przewiduje odpowiednie klasy dla wykrywanych obiektów. Widoczny wzrost, spadek i ponowny wzrost precyzji odzwierciedlają proces dostosowywania się modelu do różnych aspektów danych i funkcji straty, przez co nie powinien budzić niepokoju, o ile końcowa wartość precyzji jest wysoka.

metrics/recall(B): Wskaźnik czułości wzrasta, co oznacza, że model wykrywa coraz większy odsetek rzeczywistych obiektów.

val/box_loss: Strata dla ramek na danych walidacyjnych obniża się z każdą epoką, co świadczy o poprawie jakości przewidywań również na danych testowych.

val/cls_loss: Malejąca wartość strat dla klasyfikacji w walidacji, podobna do strat treningowych oznacza stabilność modelu w trakcie treningu.

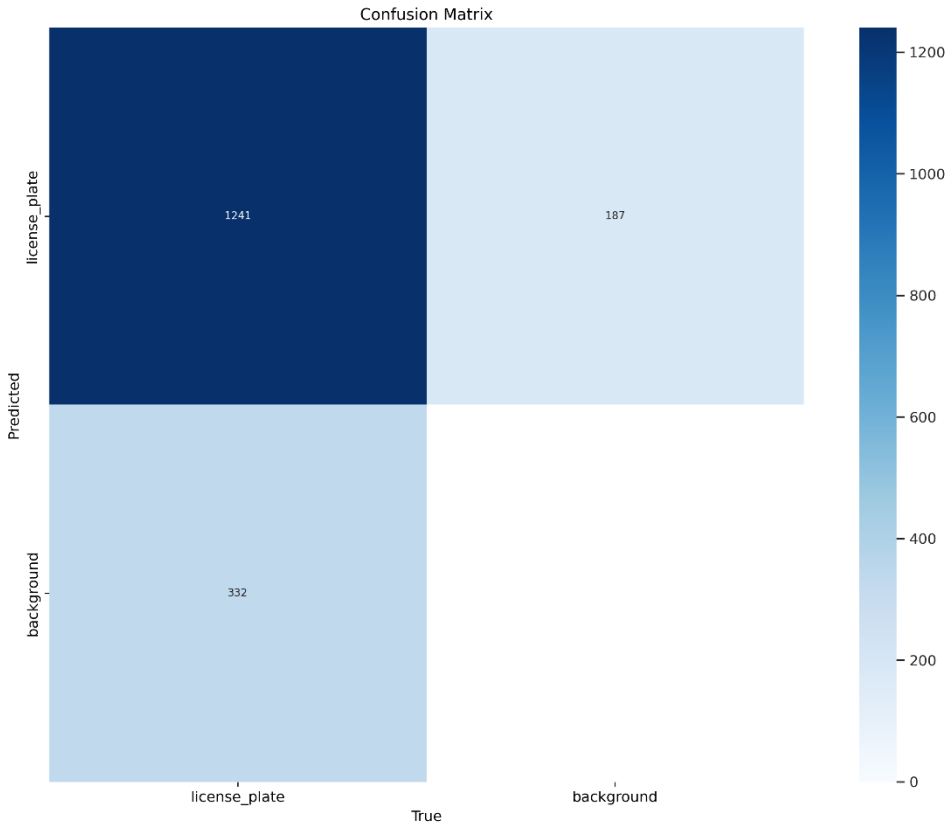
val/dfl_loss: Również tutaj można zobaczyć zmniejszenie strat lokalizacyjnych w danych walidacyjnych. Oznacza to, że model dokonuje postępów w szkoleniu.

metrics/mAP50(B): Miara mAP50, czyli średnia precyzja dla progu IoU 50%, systematycznie wzrasta. Można z tego wnioskować poprawę skuteczności modelu w detekcji obiektów

metrics/mAP50-95(B): Wzrost wartości w tej kategorii oznacza, że model staje się bardziej wszechstronny.

5.2 Macierz pomyłek

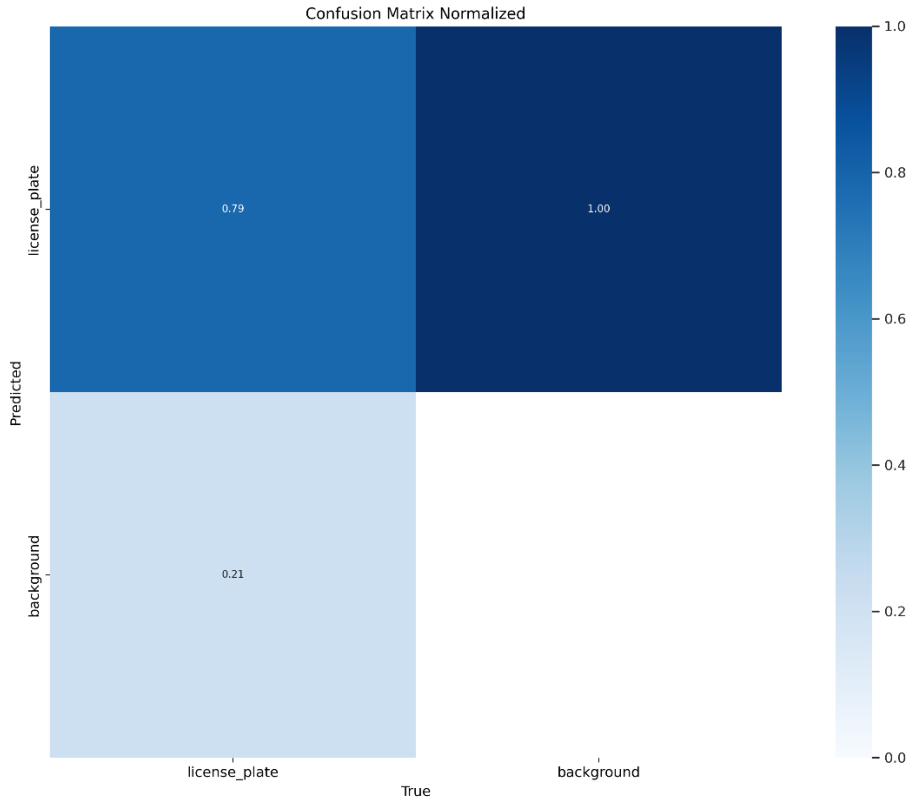
Poniższy rysunek przedstawia macierz pomyłek uzyskaną w trakcie szkolenia. Na osi poziomej znajdują się klasy obiektów, natomiast na osi pionowej przedstawione są wyniki przewidziane przez model. Powstają dzięki temu cztery pola: w lewym górnym rogu znajduje się liczba przykładów, które faktycznie były tablicami rejestracyjnymi i zostały przez model zidentyfikowane prawidłowo. W prawym górnym rogu przedstawiona jest ilość przypadków, w których model błędnie sklasyfikował tło jako tablicę rejestracyjną. W lewym dolnym rogu można zauważyć ilość tablic rozpoznanych przez model jako tło. Natomiast w prawym dolnym rogu podana powinna być informacja, ile razy model poprawnie zidentyfikował tło. Jak widać na podanym wykresie najciemniejszym odcieniem niebieskiego, a zatem największym skupiskiem danych, jest obszar w lewym górnym rogu. Wskazuje to na dużą poprawną ilość klasyfikacji tablic rejestracyjnych. Obszar w prawym górnym rogu pokazuje jednak, że w zdarzyły się przypadki, w których model błędnie zinterpretował tło jako tablicę, czyli tak zwane fałszywe alarmy. Skala znajdująca się po prawej stronie, ma na celu umożliwić szybką ocenę, gdzie występuje największa ilość próbek danego typu. Im ciemniejszy kolor, tym więcej próbek. Taka wizualizacja umożliwia łatwe określenie czy model częściej myli tło z tablicą, czy na odwrót. Pomaga to zidentyfikować w których obszarach działa on poprawnie, a gdzie wymagana jest poprawa.



Rysunek 5.2. Wizualizacja macierzy pomyłek
Źródło: opracowanie własne.

5.3 Znormalizowana macierz pomyłek

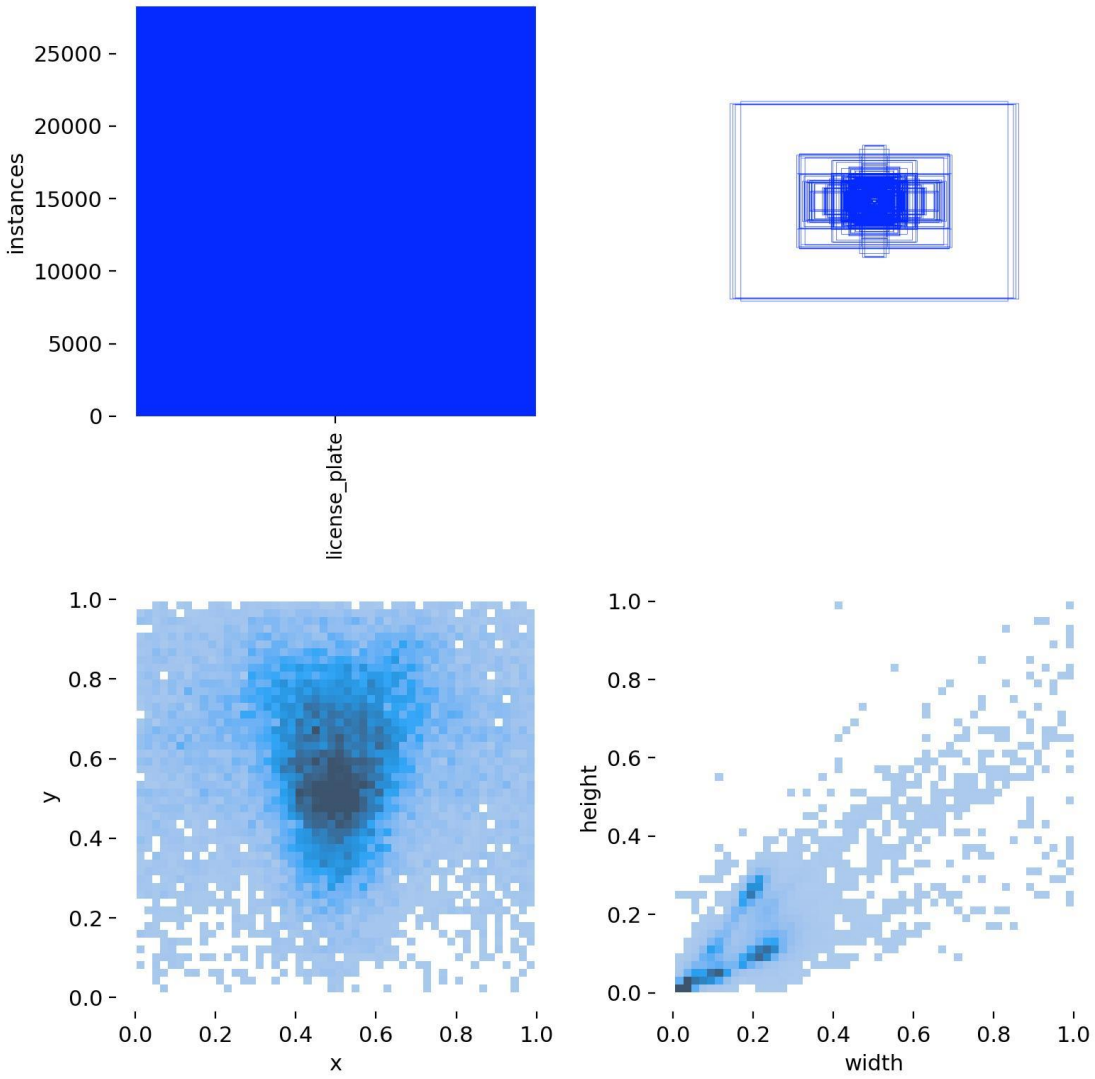
W odróżnieniu od przedstawionej wcześniej macierzy pomyłek, w której przedstawiona została konkretna liczba poprawnych oraz błędnych klasyfikacji, poniżej znajduje się wersja znormalizowana. Różnica polega na tym, że wartości w komórkach nie reprezentują liczby przykładów, tylko odsetek próbek danej klasy. Można dzięki temu łatwiej ocenić skalę błędu jaką model uzyskał podczas szkolenia. Wartość 0.79, znajdująca się w lewym górnym rogu oznacza, że 79% tablic została prawidłowo rozpoznana jako rejestracja. Natomiast 0.21 odnosi się do 21% które model pomylił z tłem.



Rysunek 5.3. Znormalizowana macierz pomyłek
Źródło: opracowanie własne.

5.4 Charakterystyka ramek

Poniżej zaprezentowano zestaw wykresów przedstawiający charakterystykę zbioru ramek dla klasy tablic rejestracyjnych. W lewym górnym rogu znajduje się pojedynczy słupek pokazujący ilość instancji tej klasy. Po prawo, u góry można zauważyć nałożone na siebie kontury wszystkich ramek, co obrazuje w jakim miejscu się znajdują oraz jaki kształt mają. Na dole, po lewej znajduje się mapa cieplna obrazująca zależność między pozycjami x oraz y. Obszar, w którym ramki pojawiają się najczęściej charakteryzuje się ciemniejszym kolorem niż reszta mapy. Wynika z tego, że tablica najczęściej znajdowała się w okolicy środka obrazu. Druga mapa, znajdująca się po prawo odnosi się do szerokości i wysokości. Można na niej zauważyć zależność wskazującą, że wraz z wzrostem szerokości, wysokość również zwiększała się. Wskazuje to na podobieństwo w wymiarach obiektów wykrytych przez model.

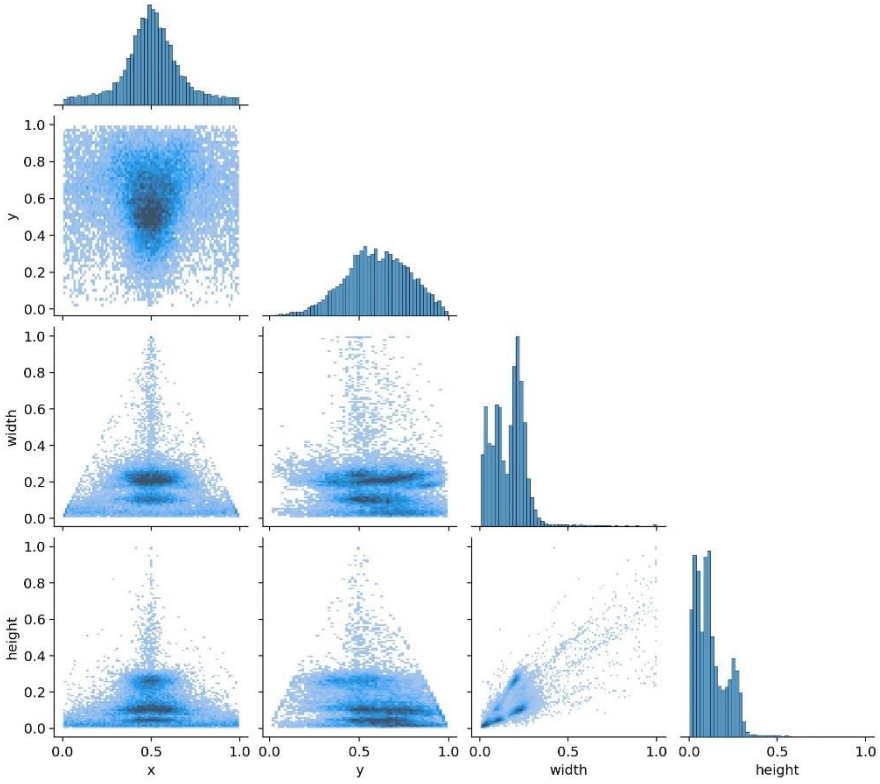


Rysunek 5.4. Charakterystyka ramek przedstawiona na wykresach
Źródło: opracowanie własne.

5.5 Powiązania oraz parametry ramek

Zamieszczony na rysunku 5.5 zbiór wykresów pokazuje powiazania i rozkłady parametrów ramek. Konkretnie współrzędne x i y oraz wysokość i szerokość. Na przekątnej wspomnianego zestawu zauważyć można wykresy słupkowe, pokazujące rozkład poszczególnych zmiennych, pod nimi z kolei znajdują się mapy cieplne. Na wykresach można zobaczyć, że wartości x i y przeważnie skupiają się w okolicy środka, podczas gdy szerokość i wysokość ramek skupia się u dołu wykresów. Warto odnotować, że szerokość osiąga większe wartości niż wysokość. Jest to spowodowane kształtem tablicy rejestracyjnej, która w większości przypadków jest prostokątna. Na mapach znajdujących się pod przekątną, pokazana jest relacja między parami zmiennych. Pozycja największe zagęszczenie osiąga w centrum wykresu, ponieważ przeważnie tablica

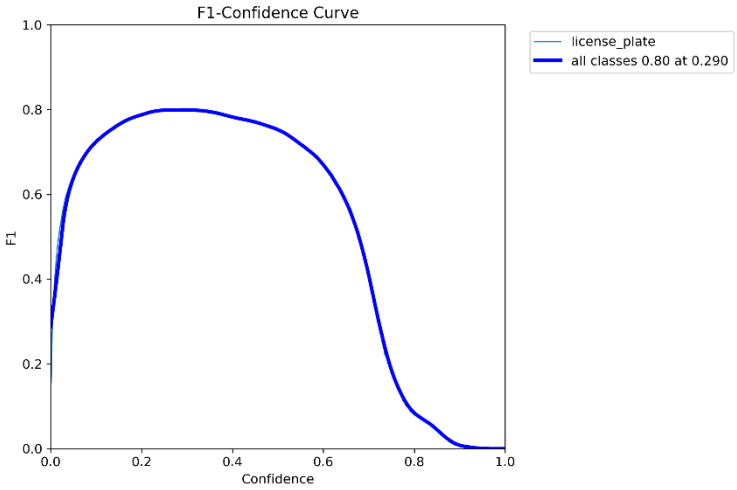
rejestracyjna występuje w środkowej strefie obrazu. Dodatkowo, porównując wykres szerokości i wysokości można odnotować, że gdy jedna z wartości rośnie, druga również ulega zwiększeniu.



Rysunek 5.5. Wizualizacja powiązań i parametrów ramek
Źródło: opracowanie własne.

5.6 Zależność F1 od pewności

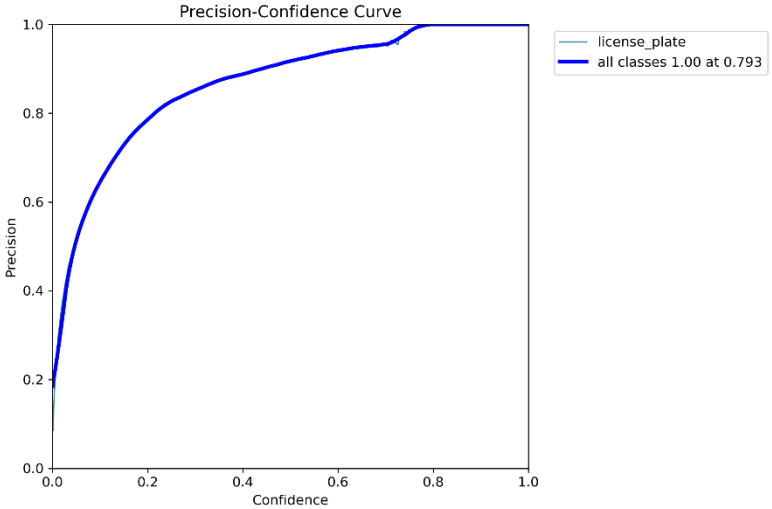
Poniższy wykres przedstawia zależność metryki F1 od pewności detekcji. Krzywa osiąga swoje maksimum wynoszące około 0.80 przy pewności równej 0.29. Jeżeli pewność jest zbyt niska model generuje wiele fałszywych alarmów, czyli rośnie liczba nieprawidłowych detekcji, obniża to jego precyzję i skutkuje spadkiem F1. Jeżeli natomiast pewność jest bardzo wysoka, spada jego czułość, jest to spowodowane odrzucaniem przez model większej liczby prawidłowych detekcji. W podanym przykładzie najlepszy kompromis występuje dla wspomnianej wcześniej wartości progu 0.29.



Rysunek 5.6. Wykres prezentujący zależność metryki F1 od pewności
Źródło: opracowanie własne.

5.7 Zależność precyzji od pewności

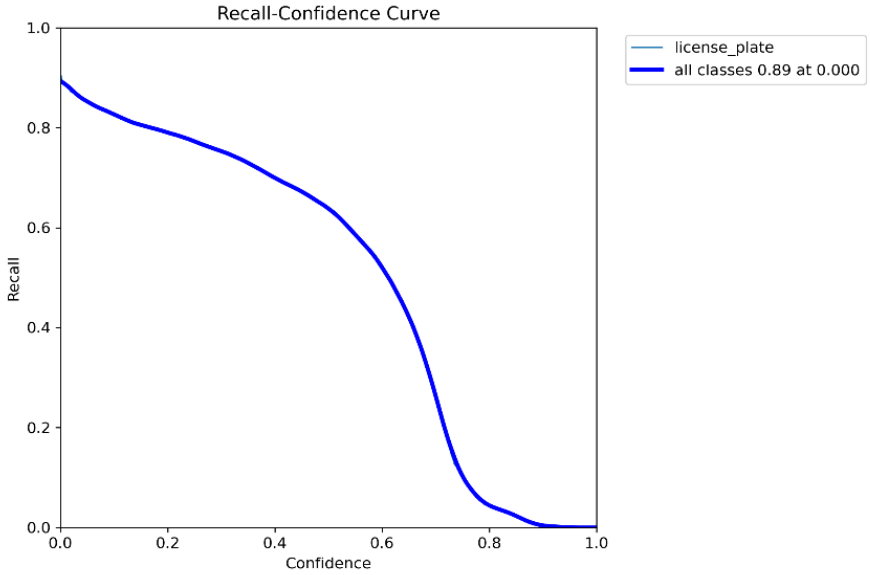
Na wykresie przedstawiono zależność precyzji od progu pewności. Widać na nim, że wraz z wzrostem pewności precyzja systematycznie rośnie, a przy wartości około 0,79 osiąga maksimum. Gdy wartość pewności jest niższa, model generuje więcej detekcji, jednak proporcjonalnie rośnie też liczba fałszywych alarmów, które obniżają precyzję. Natomiast gdy pewność jest wysoka, skutkuje to wyeliminowaniem niemal wszystkich błędnych detekcji, co w efekcie daje pełną precyzję, kosztem mniejszej liczby poprawnie wykrytych obiektów.



Rysunek 5.7. Wizualizacja zależności precyzji od pewności
Źródło: opracowanie własne.

5.8 Zależność czułości od pewności

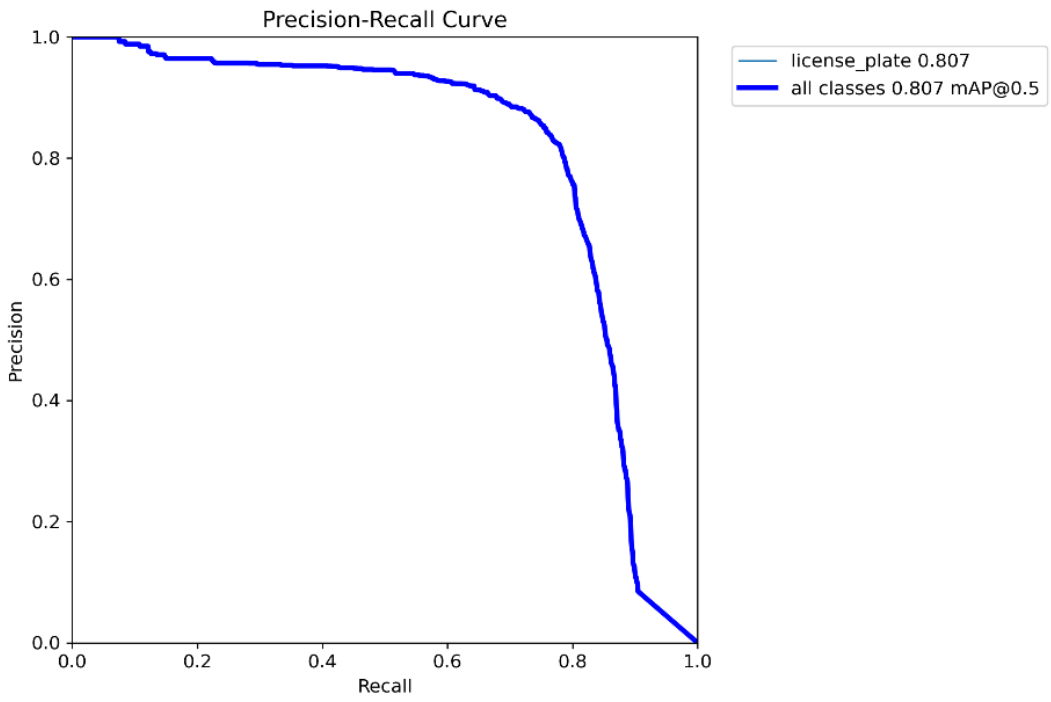
Na wykresie przedstawiono zależność czułości od progu pewności. Przy najniższym progu pewności osiągana jest największa czułość - około 0,89. Zwiększanie się progu powoduje systematyczny spadek czułości, co jest powodowane tym, że model staje się bardziej restrykcyjny i odrzuca część poprawnych detekcji. Przy bardzo wysokich wartościach progu pewności wykrywane są już tylko najpewniejsze przykłady. Skutkuje to niemal całkowitą eliminacją fałszywych alarmów, ale też znacznym obniżeniem czułości modelu.



Rysunek 5.8. Wizualizacja krzywej Recall-Confidence, czyli zależność czułości od pewności
Źródło: opracowanie własne.

5.9 Zależność precyzji do czułości

Na wykresie przedstawiona została krzywa zależności precyzji względem czułości. Na osi X przedstawiona jest czułość, natomiast na osi Y precyzja. Z legendy po prawo wykresu można odczytać wartość 0,807 którą model osiąga dla klasy "license_plate", jak i dla zbioru wszystkich klas. Przy niższych wartościach czułości precyzja jest prawie maksymalna, natomiast w miarę wzrostu czułości spada. Przy wysokiej czułości można zauważyć znaczne obniżenie precyzji, powodem tego jest trudność w zachowaniu dużej precyzji przy próbie wykrycia wielu obiektów.



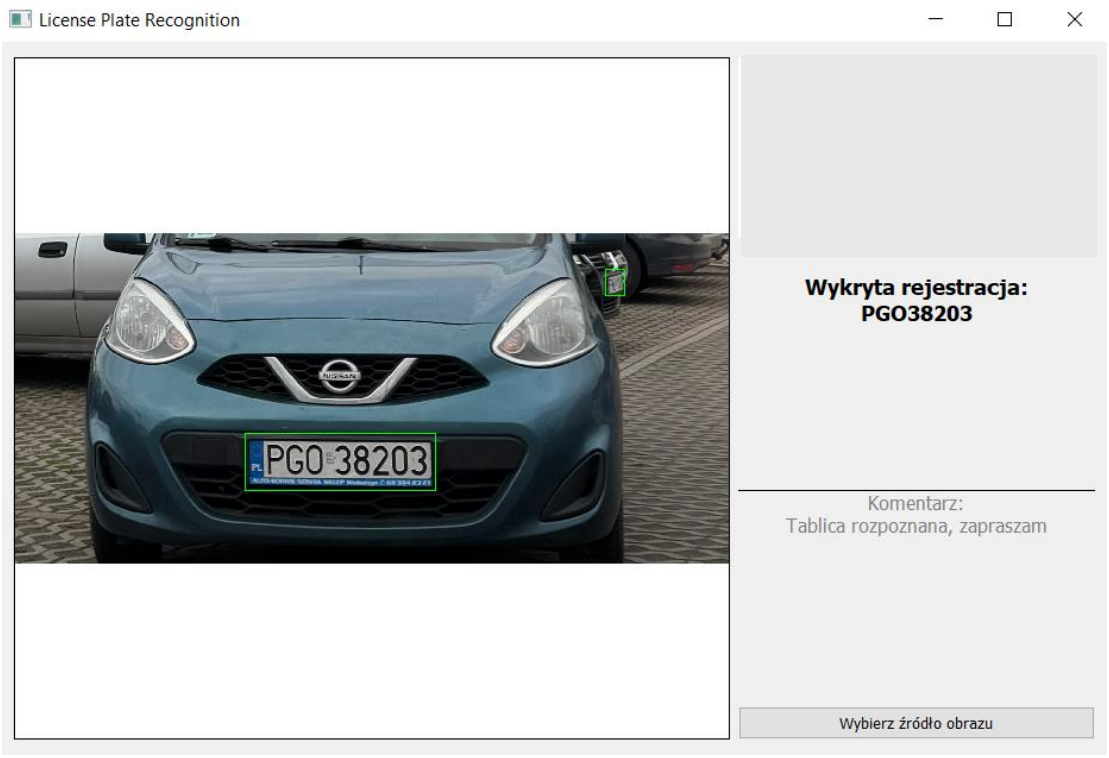
Rysunek 5.9. Wizualizacja krzywej Precision-Recall, pokazująca zależność precyzji do czułości
Źródło: opracowanie własne.

6 Prezentacja działania projektu

W tej części dokumentu przedstawiono działanie programu. W podrozdziale 6.1 zamieszczono obrazy, pokazujące, że program radzi sobie z większością tablic rejestracyjnych używanych w Polsce. Do tablic branych pod uwagę zaliczają się tablice: standardowe, zmniejszone, elektryczne, czarne, dyplomatyczne, zabytkowe oraz tymczasowe. Dodatkowo zaprezentowano działania detekcji w czasie rzeczywistym oraz przybliżono działanie bazy danych.

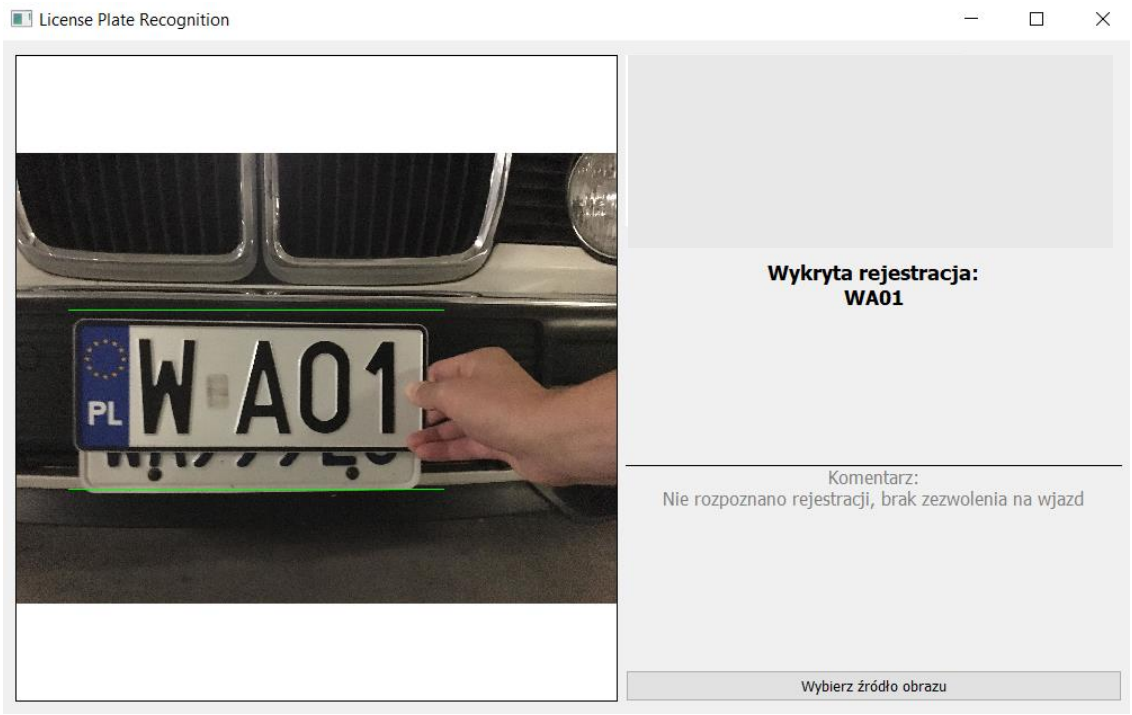
6.1 Wykrywanie tablic rejestracyjnych o różnych kolorach i wymiarach.

Na poniższym obrazie zaprezentowano jak program radzi sobie z wykryciem standardowej tablicy rejestracyjnej o wymiarach 520 x 144 mm. Tablicę charakteryzują czarne znaki na białym tle.



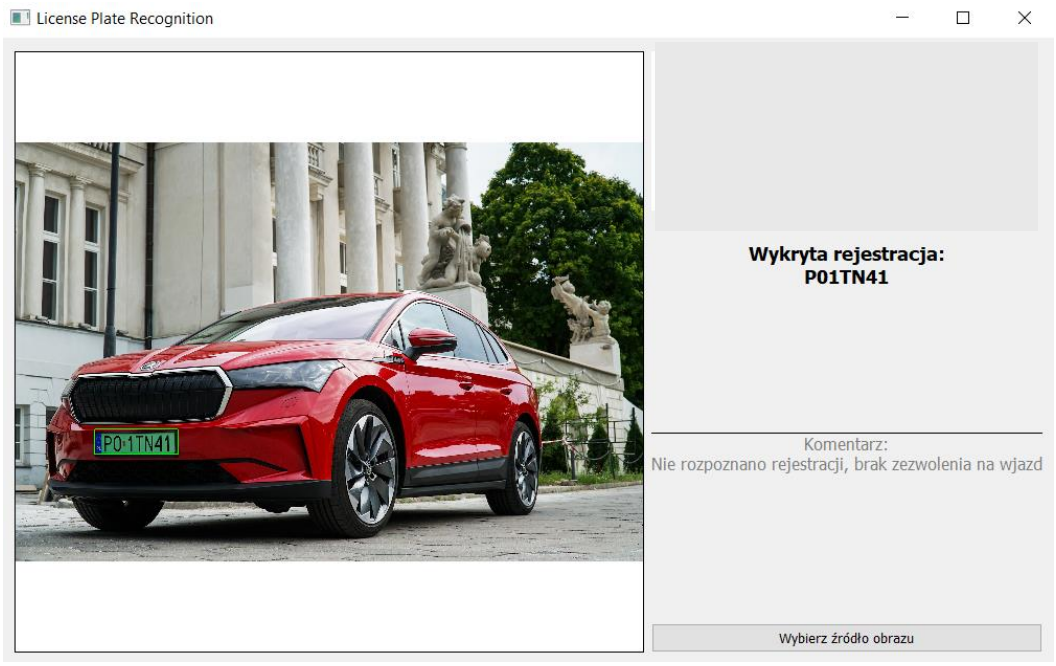
Rysunek 6.1. Przedstawienie wykrycia standardowej tablicy
Źródło: opracowanie własne.

Obraz poniżej przedstawia wynik wykrycia tablicy zmniejszonej, charakteryzującej się wymiarami 305 x 114 mm.



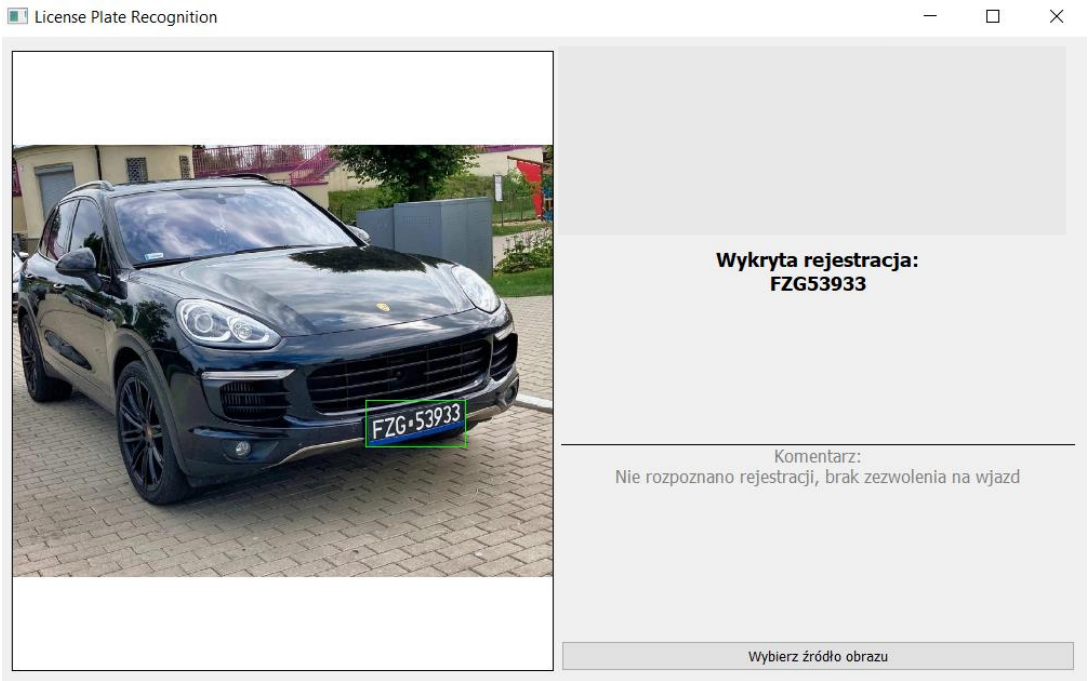
Rysunek 6.2. Wykrycie tablicy pomniejszonej
Źródło: opracowanie własne.

Na obrazie 6.3. można zauważyć wykrycie przez program tablicy dedykowanej dla pojazdów elektrycznych i wodorowych, wyróżniającej się zielonym tłem.



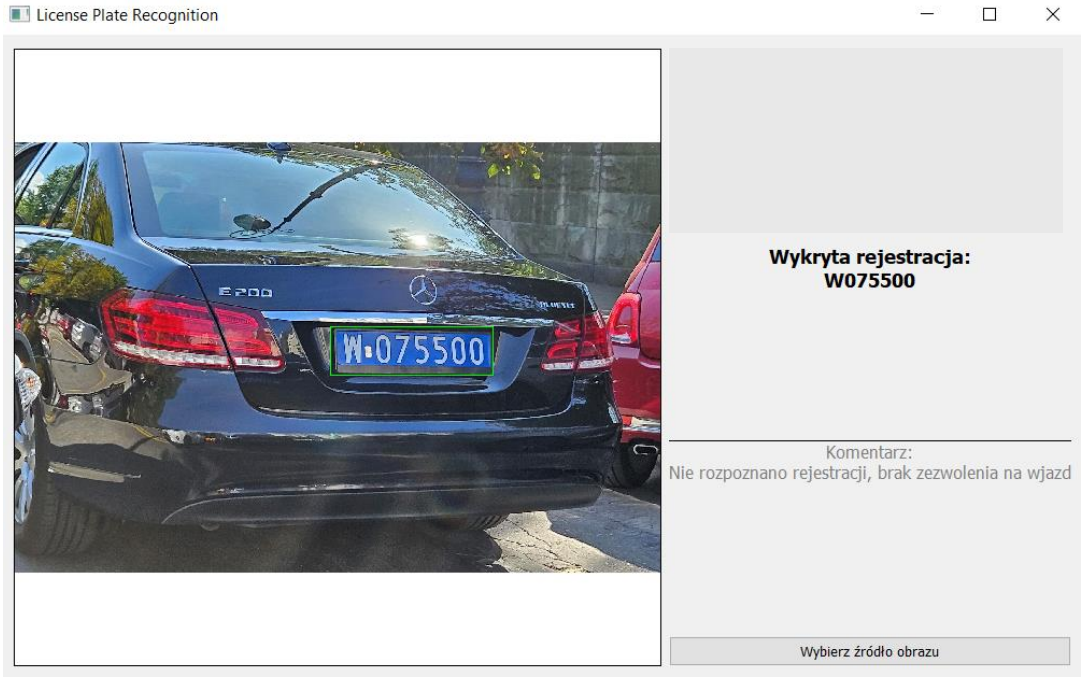
Rysunek 6.3. Wykrycie tablicy dla pojazdów elektrycznych
Źródło: opracowanie własne.

Obraz poniżej prezentuje wykrycie i odczyt tablicy o czarnym tle i białych znakach.



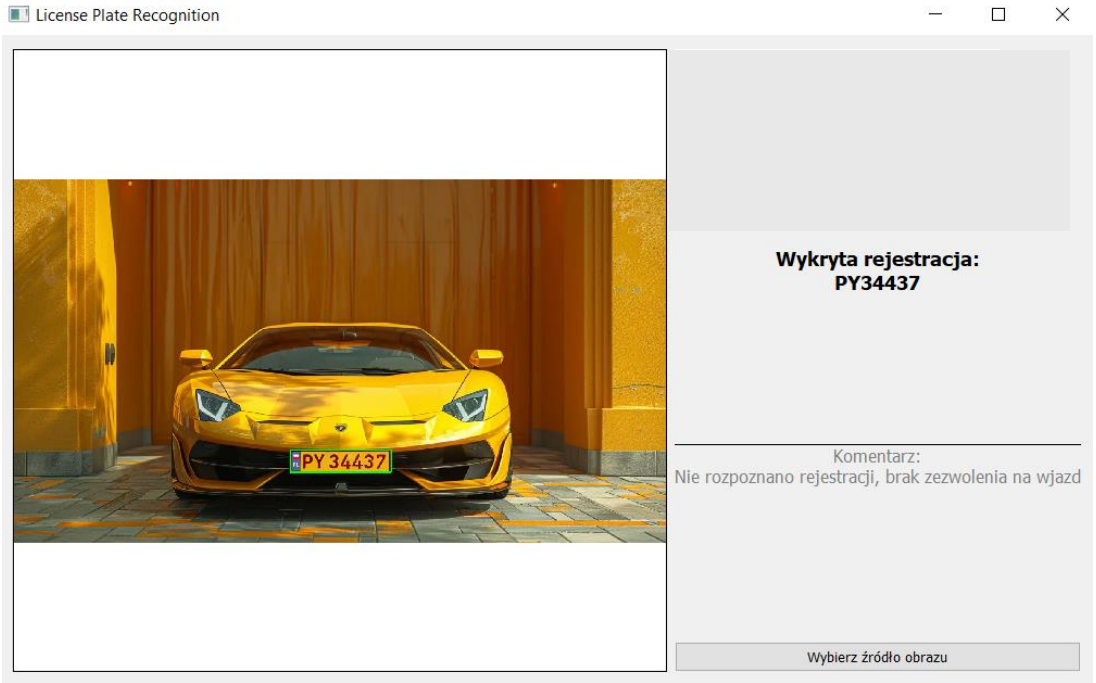
Rysunek 6.4. Wykrycie tablicy czarno-białej
Źródło: opracowanie własne.

Na obrazie 6.5. przedstawiono wykrycie tablicy dyplomatycznej, którą cechują białe znaki na niebieskim tle.



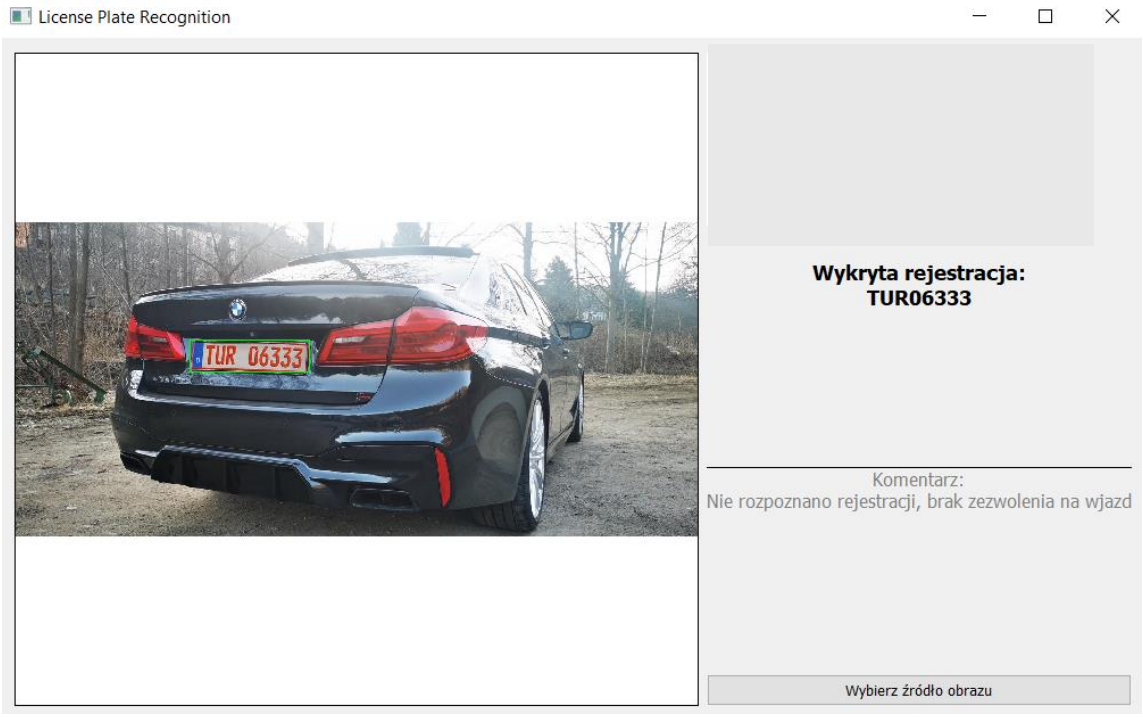
Rysunek 6.5. Wykrycie tablicy dyplomatycznej
Źródło: opracowanie własne.

Obraz poniżej przedstawia detekcję i odczyt tablicy z żółtym tłem przeznaczoną dla pojazdów zabytkowych.



Rysunek 6.6. Wykrycie tablicy zabytkowej
Źródło: opracowanie własne.

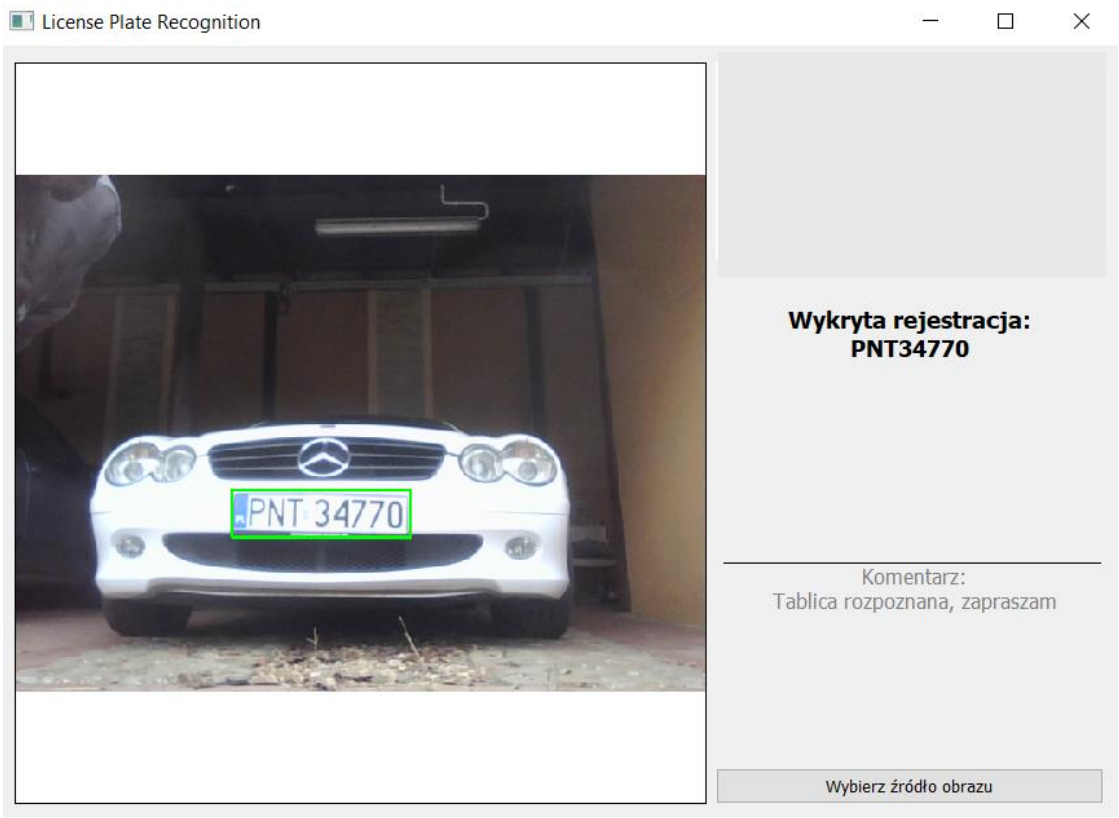
Na przedstawionym poniżej obrazie widać zlokalizowanie i odczyt tablicy w kolorach czerwono-białych (czerwone znaki na białym tle), którymi charakteryzują się tablice tymczasowe.



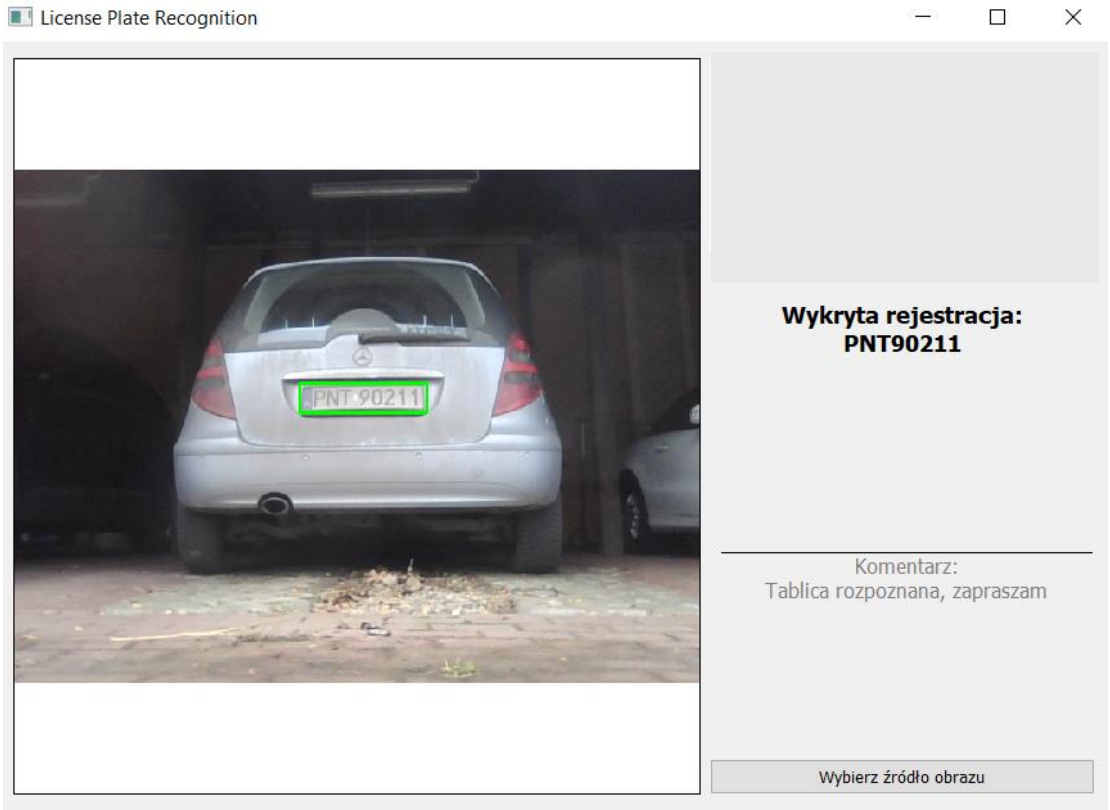
Rysunek 6.7. Wykrycie tablicy tymczasowej
Źródło: opracowanie własne.

6.2 Wykrywanie w czasie rzeczywistym

Poniżej udokumentowano dwa wykrycia w czasie rzeczywistym. Do rozpoznania użyto kamery domyślnie zamontowanej w laptopie, co widać, gdy zwróci się uwagę na lekkie przymglenie obrazu. Obraz 6.8. przedstawia wykrycie w warunkach idealnych, czyli przy dobrym oświetleniu oraz z tablicą na wprost kamery. Obraz 6.9. natomiast przedstawia rezultat wykrycia na zabrudzonej tablicy rejestracyjnej. Pomimo zakurzenia program dalej jest w stanie zidentyfikować znaki na tablicy. Warto jednak odnotować, że zakurzenie nie zniekształca w poważny sposób żadnego ze znaków, jak zrobiłoby to na przykład błoto lub śnieg.



Rysunek 6.8. Detekcja w czasie rzeczywistym w warunkach optymalnych
Źródło: opracowanie własne.



Rysunek 6.9. Detekcja w czasie rzeczywistym przy zabrudzeniu tablicy rejestracyjnej
Źródło: opracowanie własne.

6.3 Baza danych

Przedstawiona poniżej baza ma dwa zastosowania. Pierwsze z nich to nadanie prawa wjazdu pojazdom z numerami rejestracyjnymi wpisanymi do tabeli “authorised”. Drugim zastosowaniem jest działanie jako system logów, w którym system wpisuje w tabeli “detection_logs” dane ze zdarzenia, takie jak: numer rejestracji, datę oraz komentarz. Tabelę z autoryzowanymi rejestracjami można swobodnie rozszerzyć o kolejne rekordy, jednakże na potrzeby testów dodano pięć wpisów.

	id	plate_number
1	1	PNT34770
2	2	PNT90211
3	3	P04PJ54
4	4	PG038203
5	5	DDZ35150

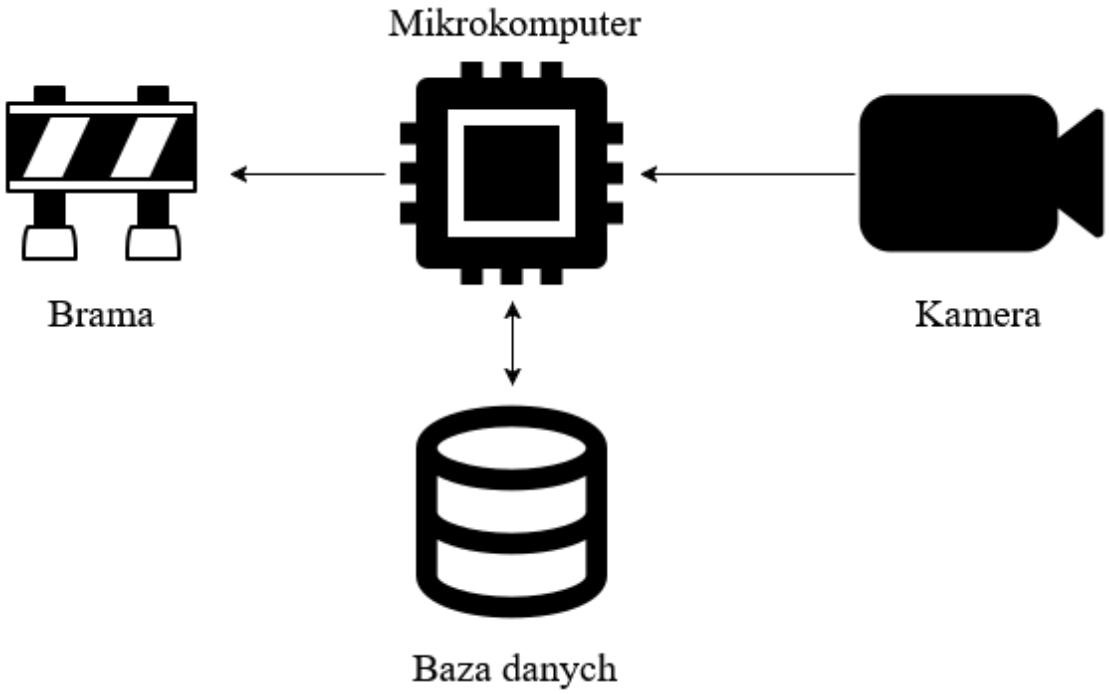
Rysunek 6.10. Przedstawienie tabeli z autoryzowanymi tablicami rejestracyjnymi
Źródło: opracowanie własne.

	id	timestamp	plate_number	comment
157	157	2025-01-22 14:27:48	W075500	Nie rozpoznano rejestracji, brak z…
158	158	2025-01-22 14:59:55	PY34437	Nie rozpoznano rejestracji, brak z…
159	159	2025-01-23 14:34:25	PG038203	Tablica rozpoznana, zapraszam
160	160	2025-01-24 00:00:58	PY34437	Nie rozpoznano rejestracji, brak z…
161	161	2025-01-24 00:01:07	PL	Nie rozpoznano rejestracji, brak z…
162	162	2025-01-24 00:01:07	WA01	Nie rozpoznano rejestracji, brak z…
163	163	2025-01-24 00:01:16	PL	Nie rozpoznano rejestracji, brak z…
164	164	2025-01-24 00:01:16	PNT34770	Tablica rozpoznana, zapraszam
165	165	2025-01-24 00:01:23	PL	Nie rozpoznano rejestracji, brak z…
166	166	2025-01-24 00:01:23	PNT90211	Tablica rozpoznana, zapraszam
167	167	2025-01-24 00:01:32	P02PS91	Nie rozpoznano rejestracji, brak z…
168	168	2025-01-24 00:01:41	PG038203	Tablica rozpoznana, zapraszam
169	169	2025-01-24 00:01:50	PL	Nie rozpoznano rejestracji, brak z…
170	170	2025-01-24 00:01:50	P04PJ54	Nie rozpoznano rejestracji, brak z…
171	171	2025-01-24 00:02:07	W075500	Nie rozpoznano rejestracji, brak z…
172	172	2025-01-24 00:02:15	NLE003	Nie rozpoznano rejestracji, brak z…
173	173	2025-01-24 00:02:22	WW625XP	Nie rozpoznano rejestracji, brak z…
174	174	2025-01-24 00:02:33	DW	Nie rozpoznano rejestracji, brak z…
175	175	2025-01-24 00:02:33	2DK01	Nie rozpoznano rejestracji, brak z…
176	176	2025-01-24 00:04:09	PY34437	Nie rozpoznano rejestracji, brak z…

Rysunek 6.11. Tabela pełniąca funkcję systemu logów
Źródło: opracowanie własne.

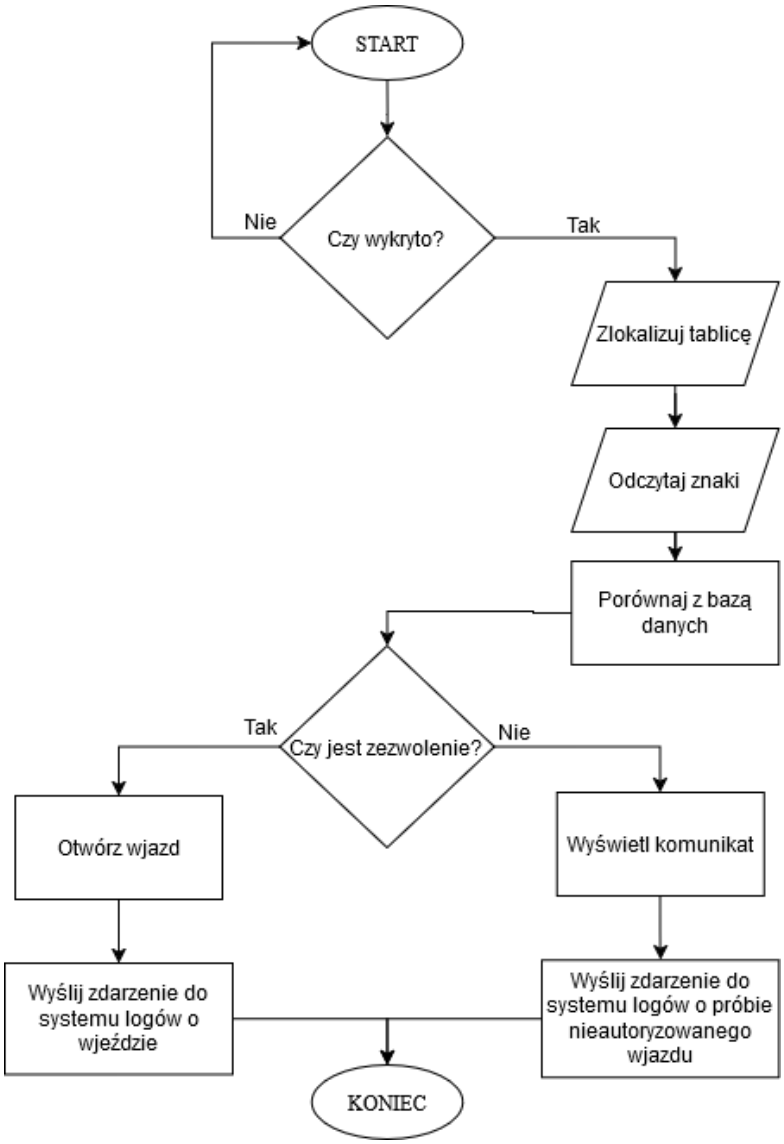
7 Przykładowa implementacja sprzętowa.

W aktualnym rozdziale zaproponowano, jak mógłby wyglądać w pełni działający system korzystający z aplikacji. Składałby się on z mikrokomputera, na którym działa program (np. Raspberry Pi), oraz kamery. Schemat systemu przedstawiono na rysunku 7.1. W centrum układu znajduje się mikrokomputer, który po otrzymaniu danych z kamery przetwarza je i wysyła zapytanie do bazy danych, czy wykryta rejestracja figuruje na liście pojazdów autoryzowanych do wjazdu. Jeśli informacja zwrotna potwierdzi autoryzację, aplikacja otwiera bramę. W przeciwnym razie szlaban pozostaje zamknięty, a program wyświetla komunikat o braku zezwolenia na wjazd.



Rysunek 7.1. Schemat prezentujący działanie systemu
Źródło: opracowanie własne.

Dodatkowo, na poniższym obrazie przedstawiono schemat blokowy działania programu, który obrazuje poszczególne etapy przetwarzania danych. Schemat ten pozwala na zrozumienie przepływu informacji i sposobu podejmowania decyzji przez system.



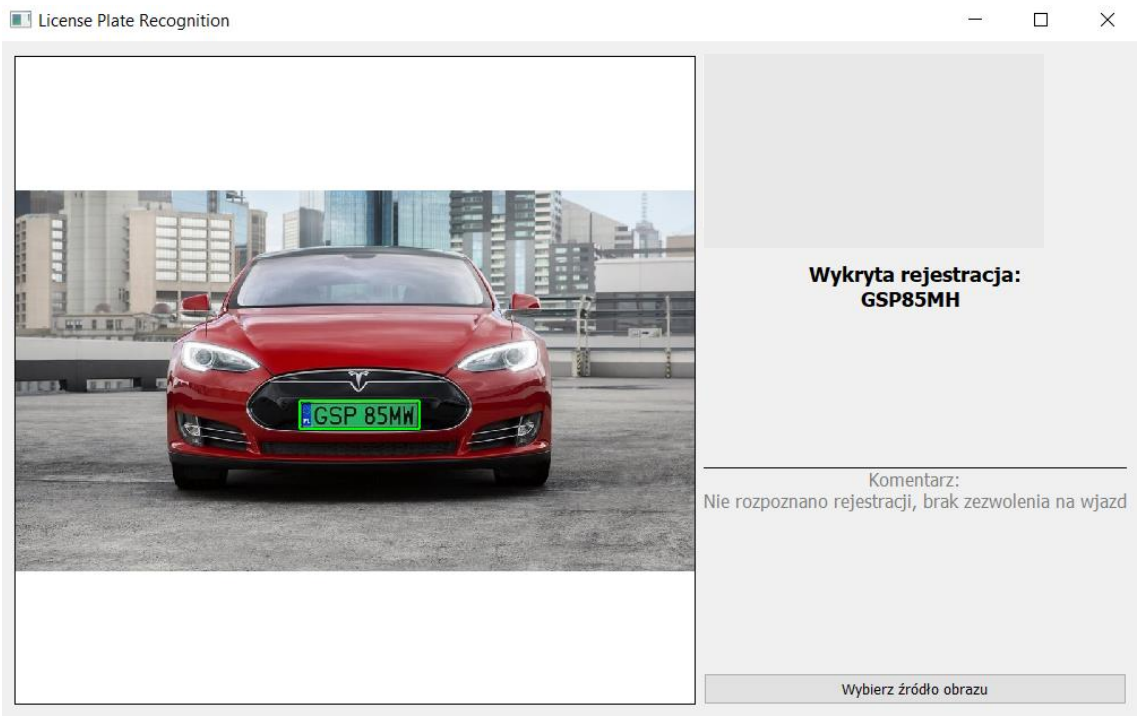
Rysunek 7.2. Schemat blokowy prezentujący działanie programu
Źródło: opracowanie własne.

8 Trudności

Program pomimo zadowalającego działania, jest daleki od ideału. Podczas testowania napotkano kilka problemów, które opisano poniżej.

8.1 Problemy z jakością

Pierwszym i najpoważniejszym problemem jest jakość obrazu. Jeżeli obraz będzie zbyt słaby pod względem jakości, program zwróci błędne rozpoznanie. Jest to spowodowane przybliżaniem obrazu, który już na starcie ma słabą jakość, a dalsze przybliżanie go powoduje zniekształcenie znaków. W związku z czym program nie rozpoznaje ich prawidłowo. Przykład zaprezentowano poniżej. Jak można zauważyć program odczytał ostatnią literę rejestracji “W” jako “H”.



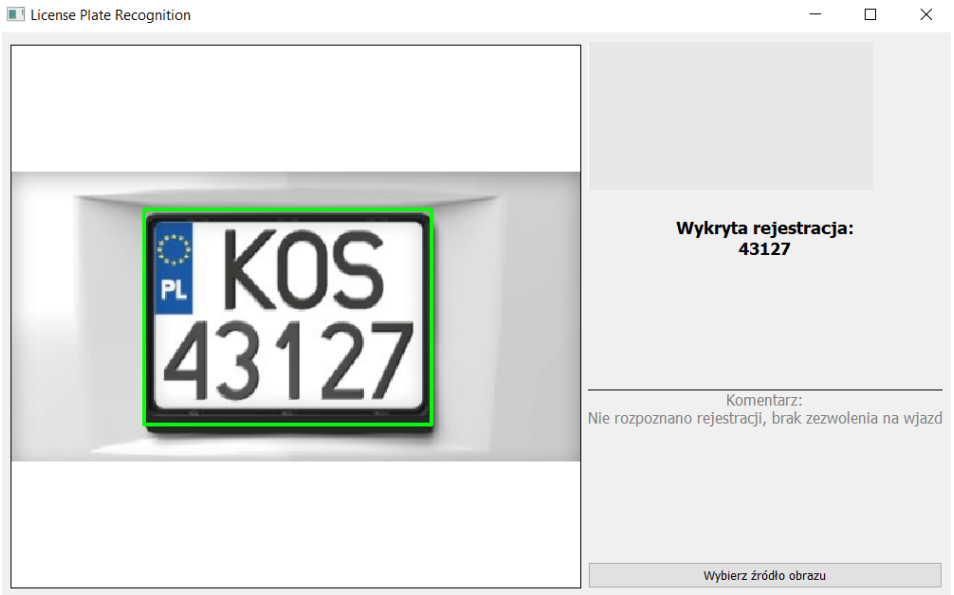
Rysunek 8.1. Prezentacja błędnego odczytu spowodowanego zbyt niską jakością obrazu
Źródło: opracowanie własne.

Jeżeli natomiast obraz będzie zbyt dokładny, program może skupić się na niewłaściwych znakach do odczytania i, zamiast zwrócić znaki wypisane na tablicy, będzie próbował odczytać tekst z dołu rejestracji, np. adres salonu, w którym auto jest serwisowane. Można to zauważyć na poniższym przykładzie, który przedstawia sytuację, w której program napotkał trudności. Problem ten częściowo łączy się z wcześniej omówionym zagadnieniem dotyczącym zbyt niskiej jakości obrazu. W tym przypadku, po przybliżeniu tekstu znajdującego się na dole tablicy, okazuje się, że nie jest on wystarczająco wyraźny, aby program mógł go prawidłowo odczytać. Jak można wnioskować po zwróconym tekście, program próbuje odczytać “Group Polska”. Sugerują to pierwsze cztery litery wykrycia, w których program pomylił “O” z “Q”.

Rysunek 8.2. Prezentacja błędnego odczytu spowodowanego zbyt dużym przybliżeniem obrazu
Źródło: opracowanie własne.

8.2 Trudności z odczytem rejestracji dwurzędowych.

Następnym napotkanym problemem jest nieprawidłowość w odczycie tablic dwurzędowych. Program traktuje to jako dwa osobne byty do odczytu, w związku z czym pomimo prawidłowej interpretacji znaków, zwracane jest błędne rozpoznanie. Jak można zobaczyć poniżej, program zwrócił jako rozpoznanie tylko dolną część tablicy, jednakże w logach systemu (Rys. 8.3) można odnotować zarejestrowanie obu części jako dwa osobne wykrycia.



Rysunek 8.3. Prezentacja błędu spowodowanego wykryciem tablicy dwurzędowej z perspektywy programu
Źródło: opracowanie własne.

243	243	2025-01-25 14:53:42	K0S	Nie rozpoznano rejestracji, brak z…
244	244	2025-01-25 14:53:42	43127	Nie rozpoznano rejestracji, brak z…

Rysunek 8.4. Prezentacja błędu spowodowanego wykryciem tablicy dwurzędowej z perspektywy logów
Źródło: opracowanie własne.

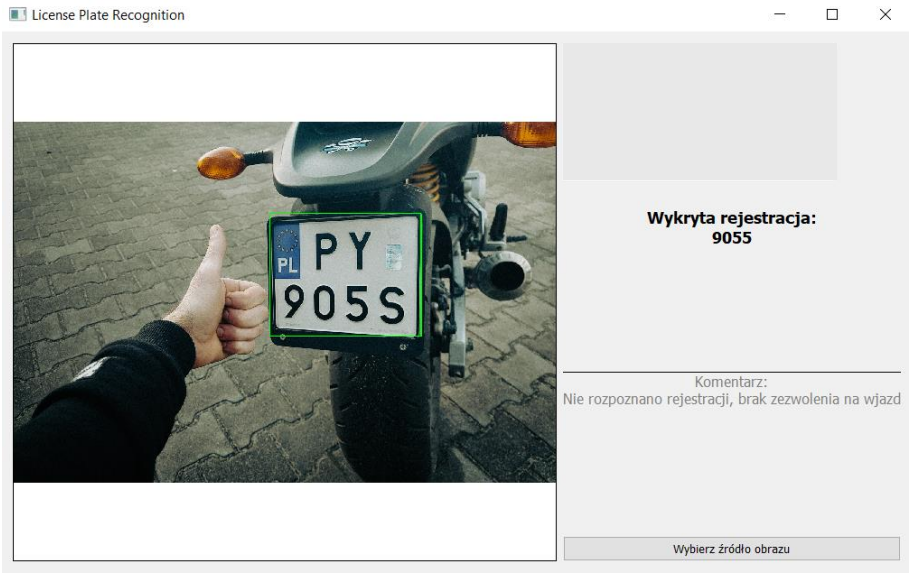
8.3 Pomyłki związane ze znakami “0” oraz “O”.

Podczas testowania napotkano również problemy, w których program mylił niektóre znaki. Jednym z nich jest para “O” oraz “0”, które z powodu na podobną konstrukcję są mylone w niektórych detekcjach. Przykład przedstawiono poniżej. Na potrzebę testu zamazano dolną część, w której znajdywała się informacja o serwisie samochodowym. Jak widać na obrazie tablica rozpoczyna się znakami “PO” które wskazują na rejestrację poznańską, jednakże program zidentyfikował to jako “P0”. Problem ten jest popularny i występuje w wielu programach do rozpoznawania znaków.

Rysunek 8.5. Przykładowy błąd interpretacji znaku “O” jako ”0”
Źródło: opracowanie własne.

8.4 Pomyłki związane ze znakami “5” oraz “S”

Problematyczne pary znaków nie kończą się jednak tylko na wspomnianych już przypadkach. Niewygodne do rozpoznania są również znaki “5” oraz “S”. Przykład błędnego rozpoznania tych znaków przedstawiono poniżej. Jak można zauważyć program odczytał “905S” jako “9055”.



Rysunek 8.6. Przykładowy błąd, w którym program zinterpretował znak "S" jako "5"
Źródło: opracowanie własne.

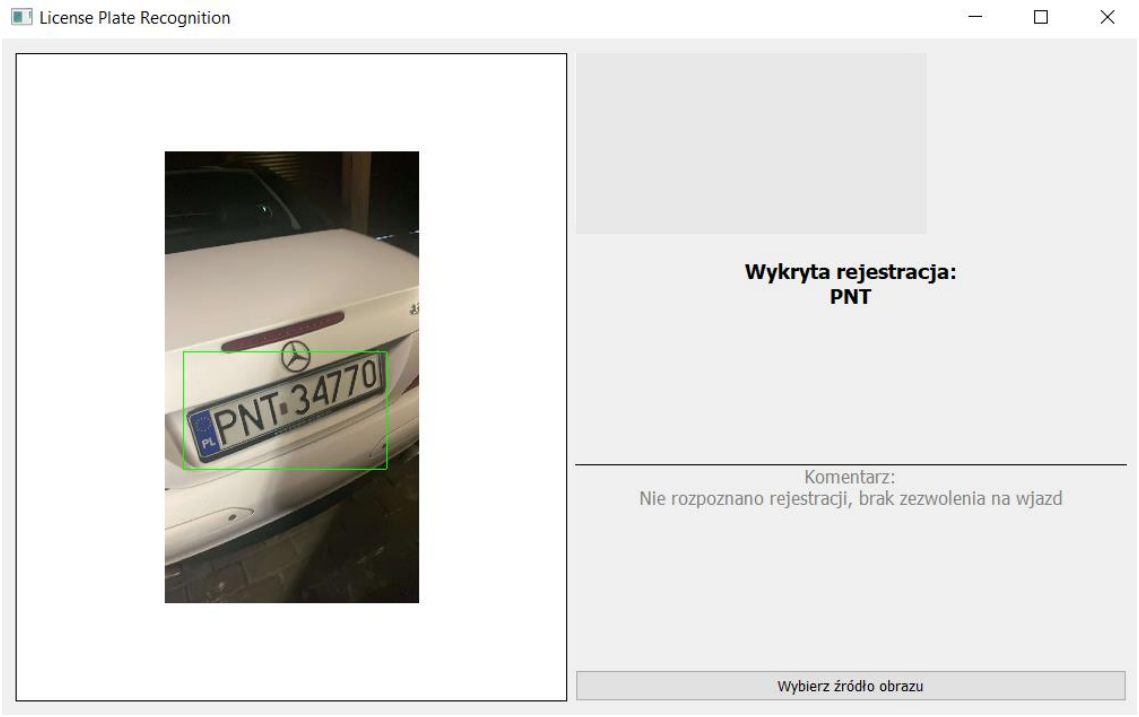
8.5 Pomyłki związane ze znakami "I" oraz "1"

Kolejną problematyczną parą znaków są symbole "I" oraz "1", które jak w poprzednim przykładzie, charakteryzują się podobną konstrukcją, przez co program czasem myli je w odczycie. Na obrazie poniżej widać przykład takiej pomyłki. Problem odnotowano podczas testowania jak program zachowa się podczas wykrycia zagranicznych rejestracji. W tym wypadku jest to tablica brytyjska.

Rysunek 8.7. Przykładowy błąd odczytu znaku "I" jako "1"
Źródło: opracowanie własne.

8.6 Problem z odczytem rejestracji pod kątem

Program napotyka problemy z odczytem rejestracji, kiedy ta nie jest skierowana wprost w obiektyw. Jest to ponownie spowodowane zniekształceniem znaków, ale również tym, że aplikacja czasem traktuje obróconą rejestrację jako dwa osobne wykrycia. Nawiązuje to do problemu z odczytem tablic dwurzędowych. Przykład przedstawiono niżej. Można zauważyć, że wyszkolony specjalnie do projektu model YOLO wykrył, gdzie znajduje się tablica. Wywnioskowano więc, że problem leży w używanym do projektu EasyOCR, który nie daje sobie z taką trudnością rady.

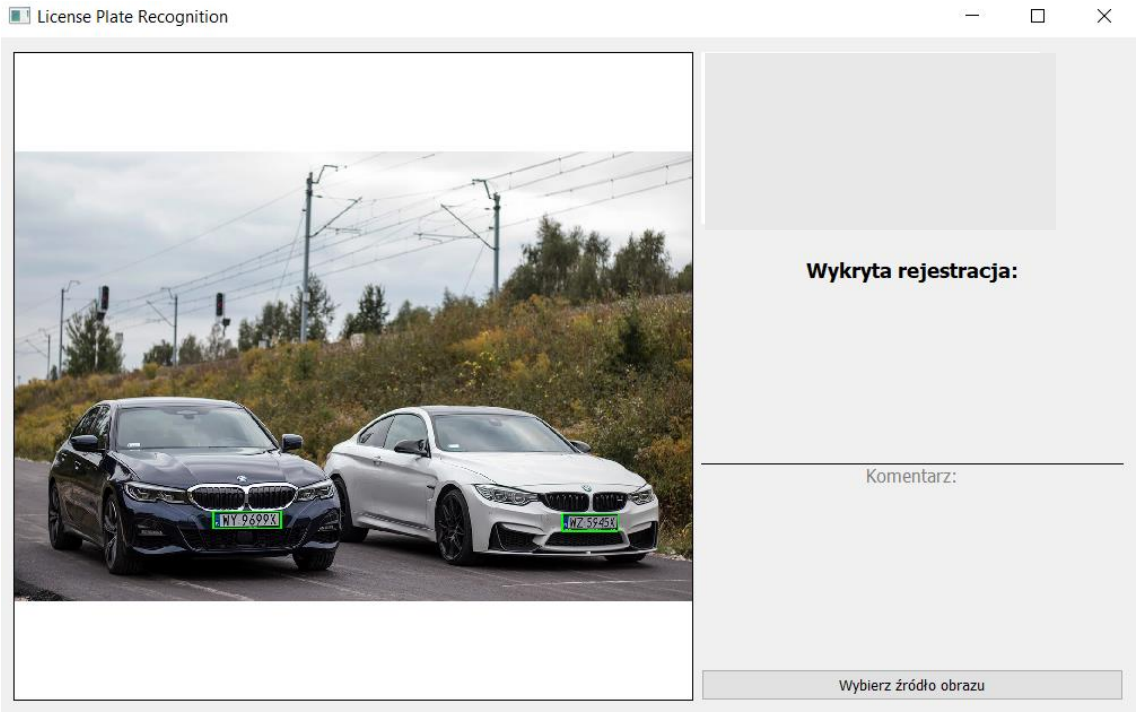


Rysunek 8.8. Prezentacja błędnego odczytu tablicy ustawionej pod kątem
Źródło: opracowanie własne.

8.7 Problem z odczytem dwóch rejestracji obok siebie.

Ostatnim udokumentowanym problemem, napotkanym podczas szkolenia, jest nieporadność programu w sytuacji, w której obok siebie staną dwa pojazdy. YOLO wykryje obie rejestracje, ale program nie będzie wiedział, którą odczytać, przez co nie odczyta żadnej. Warto jednak pamiętać, że program został stworzony z myślą o kontroli wjazdu na dany teren, poprzez wpuszczanie pojazdów pojedynczo, a więc nie zawarto w kodzie funkcji odczytu kilku tablic na raz. Problem ten różni się od błędnych odczytów tablic dwurzędnych tym, że w przypadku tych tablic następuje jedno wykrycie, w którym jest kilka poziomów tekstu. Tutaj natomiast program otrzymuje dwa różne źródła tekstu.

Pomimo że pierwotnie funkcja nie była przewidziana do dodania, warto jednak odnotować ten błąd, aby w przyszłości dodać taką funkcjonalność w celu usprawnienia aplikacji. Udokumentowanie problemu przedstawiono na obrazie poniżej.



Rysunek 8.9. Błąd w odczycie, przy wykryciu dwóch tablic na raz
Źródło: opracowanie własne.

8.8 Metody rozwiązania wspomnianych problemów

Aby zneutralizować wspomniane wcześniej błędy albo przynajmniej zmniejszyć ryzyko ich wystąpienia zasugerowano następujące działania:

W celu zapobiegania problemom związanym z jakością obrazu oraz odległością od kamery, wystarczy wyznaczyć obszar, w którym pojazdy musiałyby się zatrzymać w celu zidentyfikowania. Rozwiązałoby to również problemy programu z odczytem tablic obróconych pod kątem. Taka standaryzacja warunków zapewniłaby jednakowe warunki oświetlenia dla wszystkich pojazdów oraz kąt nachylenia kamery. Ponadto zagwarantowałoby to prostopadły kadr na tablicę rejestracyjną.

Jeżeli chodzi o problemy z tablicami dwurzędowymi, wymagane będą poprawki w kodzie. Należałoby traktować oba odczyty jako jedno wykrycie i przekazać je do bazy danych jako jeden połączony tekst, pamiętając jednak o kolejności. Tekst musiałby zostać wysłany, zaczynając od góry i łączony z dolną częścią. Alternatywnym rozwiązaniem byłoby wyszkolenie modelu, który poradziłby sobie z odczytem kilkupoziomowego

tekstu. Istnieje też możliwość obliczenia średniej odległości pionowej między rzędami i jeżeli mieściłaby się ona w określonym zakresie, program zakładałby, że są to wiersze jednej tablicy.

W celu uniknięcia problemu z pomyłkami podczas odczytu znaków, najlepszym krokiem byłoby wytrenowanie własnego modelu OCR. Podjęto próby takiego rozwiązania, jednakże ograniczenia sprzętowe nie pozwoliły na odpowiednie przeszkolenie modelu, w związku z czym jego wyniki były niewystarczające i generowały dużo błędnych wykryć. Podczas szkolenia można by wtedy zwiększyć nacisk na problematyczne znaki. Dodatkowo można by wtedy nałożyć filtry w celu redukcji szumu lub zwiększyć kontrast między znakami a tłem używając preprocessingu.

9 Podsumowanie

W niniejszej pracy zaprezentowano system automatycznego rozpoznawania tablic rejestracyjnych, który wykorzystuje technologie sztucznej inteligencji, takie jak model YOLOv8 oraz bibliotekę EasyOCR. Zadaniem projektu było opracowanie systemu, który automatyzuje wjazd samochodów na obszar zamknięty poprzez identyfikację i odczyt numerów rejestracyjnych. W czasie realizacji projektu opracowano aplikację, która korzysta z algorytmów do rozpoznawania obrazów. System został zaprojektowany do działania w dwóch trybach: na podstawie statycznych obrazów oraz w trybie rzeczywistym z wykorzystaniem kamery. Aplikacja została połączona z bazą danych, która spełnia dwie role. Pierwszą z nich jest trzymanie listy pojazdów upoważnionych do wjazdu, co umożliwia automatyczne podejmowanie decyzji o otworzeniu bramy. Drugą funkcją jest zapisywanie zdarzeń, obejmujących numery tablic, datę i godzinę detekcji oraz rezultat weryfikacji autoryzacji. Model był uczony na zbiorze danych, który zawierał tysiące zdjęć, aby zagwarantować właściwą zdolność generalizacji. Uzyskane wyniki pokazują zadowalającą efektywność w większości przypadków, lecz nadal można dostrzec obszary, które wymagają udoskonalenia, takie jak rozpoznawanie tablic dwurzędowych, identyfikacja podobnych znaków czy detekcja pod nietypowym kątem.

Bibliografia

a. Pozycje zwarte:

A. Król-Nowak, "Podstawy uczenia maszynowego". Wydawnictwo AGH, Kraków 2022

Monografie Prawnicze, "Prawo sztucznej inteligencji". Wydawnictwo C.H.Beck Sp. z o.o., 2020

R. Tadeusiewicz, "Odkrywanie właściwości sieci neuronowych przy użyciu programów w języku C#". Polska Akademia Umiejętności, Warszawa 2007

R. Tadeusiewicz, "Sieci neuronowe". Akademicka Oficyna Wydawnicza, Warszawa 1993

Robert A. Kosiński, “Sztuczne sieci neuronowe. Dynamika nieliniowa i chaos”. Wydawnictwo Naukowe PWN, 2017

Rozporządzenie Ministra Infrastruktury z dnia 31 sierpnia 2022 w sprawie rejestracji i oznaczania pojazdów, wymagań dla tablic rejestracyjnych oraz wzorów innych dokumentów związanych z rejestracją pojazdów (Dz. U. 2022 poz. 1847)

b. Artykuły:

R. Sapkota, M. Karkee, Comparing YOLO11 and YOLOv8 for instance segmentation of occluded and non-occluded immature green fruits in complex orchard environment. arXiv preprint arXiv:2410.19869, 2024.

c. Źródła internetowe:

https://github.com/JaidedAI/EasyOCR [dostęp: 23.01.2025]

https://github.com/ultralytics/ultralytics [dostęp: 23.01.2025]

https://langas.pl/kluczowe-terminy-dotyczace-zarzadzania-sztuczna-inteligencja/ [dostęp: 22.01.2025]

https://pl.eitca.org/artificial-intelligence/eitc-ai-tff-tensorflow-fundamentals/tensorflow-in-google-colaboratory/building-a-deep-neural-network-with-tensorflow-in-colab/examination-review-building-a-deep-neural-network-with-tensorflow-in-colab/what-is-the-role-of-the-loss-function-and-optimizer-in-the-training-process-of-the-neural-network/ [dostęp: 24.01.2025]

https://www.autodna.pl/blog/tablice-rejestracyjne-rodzaje-oznaczenia-skroty-i-inne/ [dostęp: 23.01.2025]

https://www.deeptechnology.ai/walidacja-modelu-czy-jest-potrzebna/ [dostęp: 22.01.2025]
https://www.jetbrains.com/products/compare/?product=pycharm&product=pycharm-ce [dostęp: 13.01.2025]

Spis rysunków

Spis tabeli

Streszczenie w jęz. polskim

Projekt sieci neuronowej do rozpoznawania tablic rejestracyjnych

Zadaniem przedstawianego projektu jest automatyzacja wjazdu pojazdów na zamknięty teren poprzez zlokalizowanie i odczytanie tablic rejestracyjnych pojazdów. Przy projekcie został zastosowany model sztucznej inteligencji YOLOv8, który został wyszkolony specjalnie do celów projektowych. Jego zadaniem jest zlokalizowanie tablicy rejestracyjnej na pojeździe, a następnie przekazanie informacji do programu, który przy użyciu EasyOCR odczytuje numer rejestracyjny pojazdu. Odczyt tablic funkcjonuje w dwóch modułach: rozpoznanie ze zdjęcia oraz w czasie rzeczywistym przy użyciu kamery. Po odczytaniu tablic przez EasyOCR, dane zostają porównane z bazą danych, która ma podwójne zastosowanie: pierwsze z nich to nadanie pojazdom statusu uprawniającego je do wjazdu, a drugi to dziennik logów, w których są zapisywane dane takie jak: numer tablicy rejestracyjnej, data zdarzenia oraz komentarz. Po porównaniu i sprawdzeniu danych, udzielana jest informacja zwrotna o autoryzacji wjazdu lub jego braku.

Słowa kluczowe: Rozpoznawanie, tablice rejestracyjne, sztuczna inteligencja

Streszczenie w jęz. angielskim

Neural Network Project for License Plate Recognition

The task of the presented project is to automate the entry of vehicles into a closed area by locating and reading vehicle license plates. The project used the YOLOv8 artificial intelligence model, which was trained specifically for the project. Its task is to locate the license plate on the vehicle and then transmit the information to a program that reads the vehicle's license plate number using EasyOCR. Plate reading works in two modules: recognition from a photo and in real time using a camera. After EasyOCR reads the plates, the data is compared to a database, which has a dual purpose: the first is to give vehicles a status that entitles them to enter, and the second is a log book that records data such as the license plate number, the date of the incident and a comment. After comparing and checking the data, feedback is given on whether or not entry is authorized.

Keywords: Recognition, license plates, artificial intelligence