

Zaprojektowanie i zaimplementowanie algorytmu do rozpoznawania koloru samochodów osobowych

Krzysztof Król

Promotor: Michał Grega

25.03.2022 r.

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
AGH University of Science and Technology



Potencjalne problemy

- Poprawa bezpieczeństwa ruchu drogowego
 - Pewniejsza identyfikacja uczestników ruchu nie stosujących się do przepisów
 - Aktywna pomoc organom ścigania
- Zautomatyzowanie pomiarów statystycznych
- Zastąpienie obecnych, podobnych rozwiązań
 - Wysoki koszt profesjonalnych rozwiązań
 - Dostępność kodu na podstawie otwartego źródła
- Optymalizacja wyszukiwarek pod kątem wyceny aut na podstawie koloru na portalach e-commerce

Potencjalne problemy

- Poprawa bezpieczeństwa ruchu drogowego
 - Pewniejsza identyfikacja uczestników ruchu nie stosujących się do przepisów
 - Aktywna pomoc organom ścigania
- Zautomatyzowanie pomiarów statystycznych
- Zastąpienie obecnych, podobnych rozwiązań
 - Wysoki koszt profesjonalnych rozwiązań
 - Dostępność kodu na podstawie otwartego źródła
- Optymalizacja wyszukiwarek pod kątem wyceny aut na podstawie koloru na portalach e-commerce

Potencjalne problemy

- Poprawa bezpieczeństwa ruchu drogowego
 - Pewniejsza identyfikacja uczestników ruchu nie stosujących się do przepisów
 - Aktywna pomoc organom ścigania
- Zautomatyzowanie pomiarów statystycznych
- Zastąpienie obecnych, podobnych rozwiązań
 - Wysoki koszt profesjonalnych rozwiązań
 - Dostępność kodu na podstawie otwartego źródła
- Optymalizacja wyszukiwarek pod kątem wyceny aut na podstawie koloru na portalach e-commerce

Potencjalne problemy

- Poprawa bezpieczeństwa ruchu drogowego
 - Pewniejsza identyfikacja uczestników ruchu nie stosujących się do przepisów
 - Aktywna pomoc organom ścigania
- Zautomatyzowanie pomiarów statystycznych
- Zastąpienie obecnych, podobnych rozwiązań
 - Wysoki koszt profesjonalnych rozwiązań
 - Dostępność kodu na podstawie otwartego źródła
- Optymalizacja wyszukiwarek pod kątem wyceny aut na podstawie koloru na portalach e-commerce

Potencjalne problemy

- Poprawa bezpieczeństwa ruchu drogowego
 - Pewniejsza identyfikacja uczestników ruchu nie stosujących się do przepisów
 - Aktywna pomoc organom ścigania
- Zautomatyzowanie pomiarów statystycznych
- Zastąpienie obecnych, podobnych rozwiązań
 - Wysoki koszt profesjonalnych rozwiązań
 - Dostępność kodu na podstawie otwartego źródła
- Optymalizacja wyszukiwarek pod kątem wyceny aut na podstawie koloru na portalach e-commerce

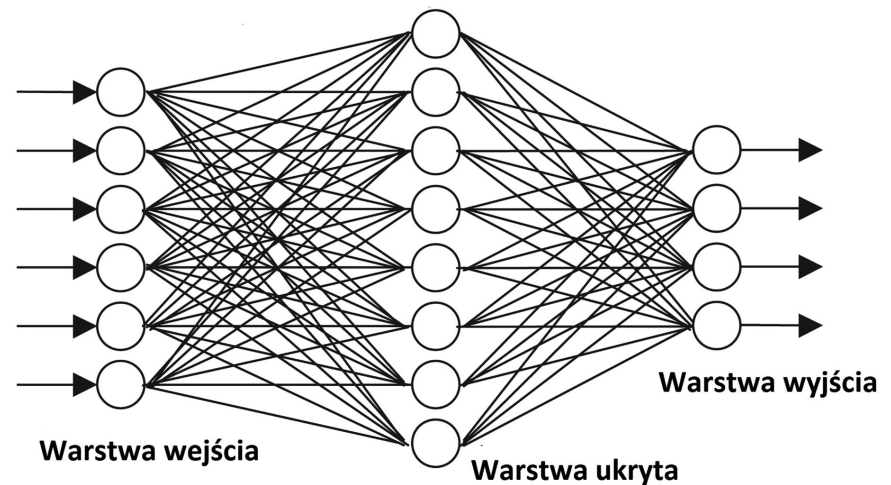
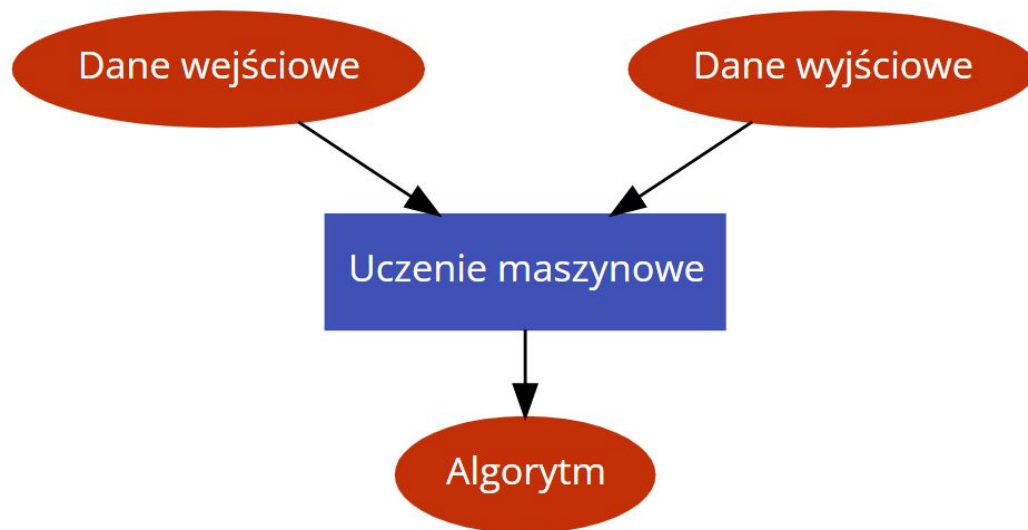
Potencjalne problemy

- Poprawa bezpieczeństwa ruchu drogowego
 - Pewniejsza identyfikacja uczestników ruchu nie stosujących się do przepisów
 - Aktywna pomoc organom ścigania
- Zautomatyzowanie pomiarów statystycznych
- Zastąpienie obecnych, podobnych rozwiązań
 - Wysoki koszt profesjonalnych rozwiązań
 - Dostępność kodu na podstawie otwartego źródła
- Optymalizacja wyszukiwarek pod kątem wyceny aut na podstawie koloru na portalach e-commerce

Uczenie maszynowe

Rozwiązanie opiera się na uczeniu maszynowym.
Wykorzystany został model sieci neuronowej MLP.

Sieci neuronowe to dominujący model w sferze przetwarzania danych wejściowych o wielu cechach (np. obrazy).



Dane

Przykładowe obrazy z używanego w projekcie zbioru danych i ich przypisy. [1]



[1] Zbiór danych zaczerpnięty z projektu: J.-W. Hsieh, L.-C. Chen, S.-Y. Chen, D.-Y. Chen, S. Alghyaline, and H.-F. Chiang, Vehicle Color Classification Under Different Lighting Conditions Through Color Correction, IEEE Sensors Journal, 2015, Vol: 15, Issue: 2, pp: 971–983.

Dane

Zbiór danych jest nieidealny z wielu powodów:

- Błąd ludzki
- Jakość zdjęcia (rozmycie, prześwietlenie)
- Różnokolorowe karoserie aut, naklejki
- Warunki atmosferyczne, brud, kurz i inne naloty

-58% !

Dlatego zbiór został manualnie **“wyczyszczony”**, a co za tym idzie - pomniejszony.

Biały	Czarny	Szary	Czerwony	Niebieski	Żółty	Zielony	Cyjanowy
4742	3418	3046	1941	1086	581	482	281

Tablica 3: Ilość elementów w każdej z klas



Biały	Czarny	Szary	Czerwony	Niebieski	Żółty	Cyjanowy	Zielony
2153	1972	1318	449	217	121	107	78

Tablica 6: Ilości elementów poszczególnych klas zbioru po jego czyszczeniu

Problematiczne dane



Czołowa dziesiątka najpopularniejszych kolorów samochodów osobowych:

1. Srebrny – 25%
2. Czarny – 23%
3. Biały – 16%
4. Szary – 13%
5. Niebieski – 9%
6. Czerwony – 8%
7. Brązowy i beżowy – 4%
8. Zielony – 1%
9. Żółty i złoty – 1%
10. Inne – poniżej 1%

Czołowa dziesiątka najpopularniejszych kolorów samochodów osobowych:

1. Srebrny – 25%
2. Czarny – 23%
3. Biały – 16%
4. Szary – 13%
5. Niebieski – 9%
6. Czerwony – 8%
7. Brązowy i beżowy – 4%
8. Zielony – 1%
9. Żółty i złoty – 1%
10. Inne – poniżej 1%

≈ 93% pokrycia

Podstawa działania algorytmu

1. Wykrycie i lokalizacja samochodów
2. Wycięcie i skalowanie obrazu
3. Wyodrębnienie cech znaczących (histogramy kolorów)
4. Wytrenowanie modelu MLP
5. Generalizacja na podstawie nowych histogramów
6. Wizualizacja

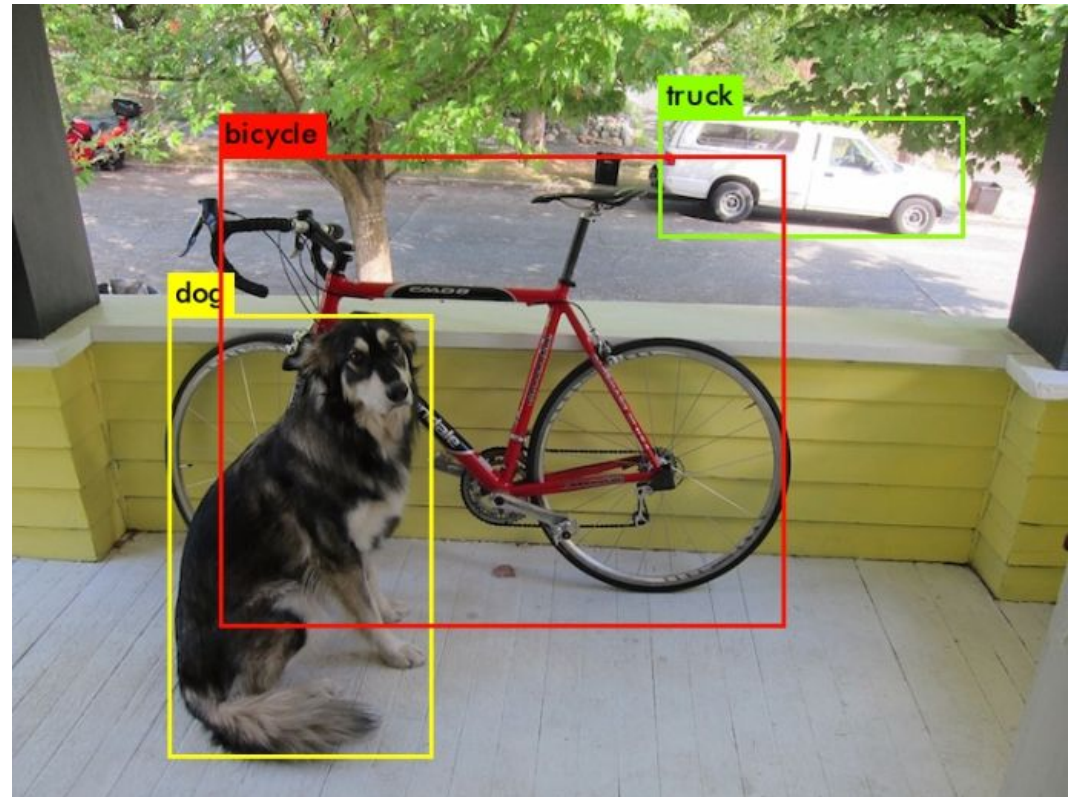
Użyte technologie



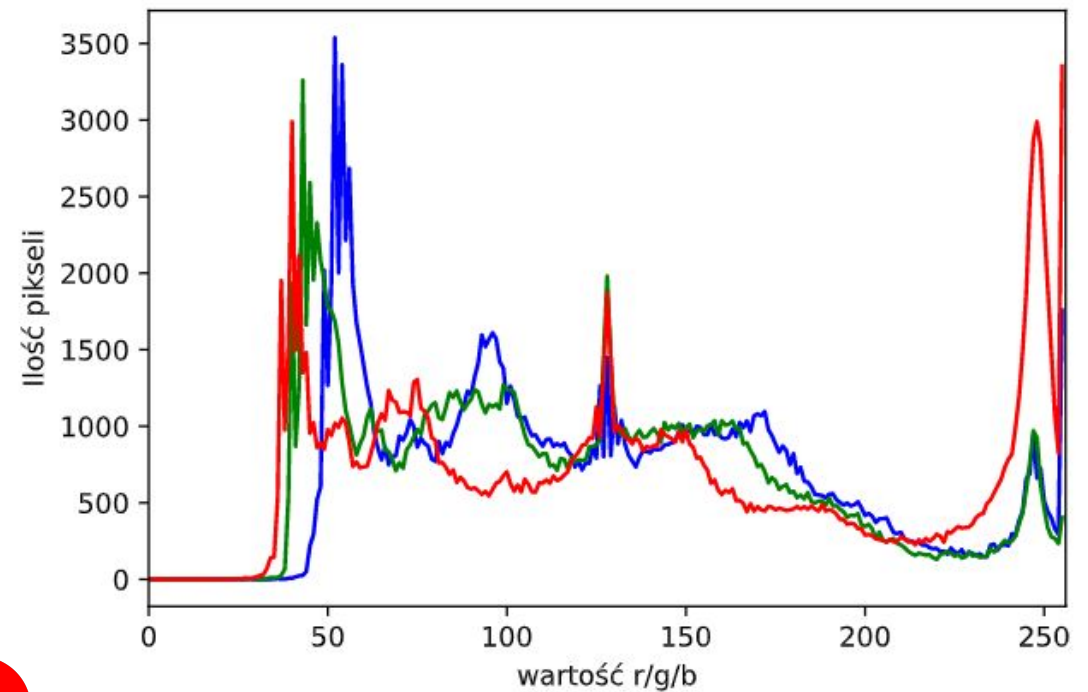
Wykrywanie samochodu

YOLO (you only look once) system pozwalający na wykrywanie obiektów z wysoką dokładnością w krótkim czasie (w tym aut)

YOLO



Wyodrębnienie cech znaczących



Parametry modelu



Aby odnaleźć parametry modelu najbliższe optymalnym, użyte zostało wyszukiwanie macierzowe parametrów (z ang. grid search).

To automatyczny proces trenowania i walidacji modelu z różnymi parametrami z ich zadanej przestrzeni.

```
parameter_space = {  
    'hidden_layer_sizes': [(100, 50, 25),  
                           (100, 70, 40, 10),  
                           (100, 45, 10),  
                           (70, 35, 17),  
                           (70, 52, 35, 17)],  
    'max_iter': [5000],  
    'activation': ['tanh', 'relu'],  
    'solver': ['sgd', 'adam'],  
    'alpha': [0.0001, 0.05, 1e-3],  
    'learning_rate': ['constant', 'adaptive'],  
}
```



```
mlp = MLPClassifier(  
    hidden_layer_sizes=(100, 50, 25),  
    max_iter=5000,  
    alpha=0.05,  
    solver="adam",  
    activation="relu",  
    learning_rate="adaptive",  
)
```

Wynik działania programu



Wynik działania programu

