

# Sprawozdanie z laboratorium:

## Komunikacja człowiek-komputer

### Sprawozdanie z przetwarzania obrazu

Prowadzący: Agnieszka Mensfelt

Autorzy:

**Jan Ciechanowicz 141202**

**Łukasz Frątczak 141213**

Zajęcia czwartkowe 16:50

## 1. Wstęp

Celem projektu było zaimplementowanie algorytmu rozpoznającego tablice rejestracyjne samochodów na zdjęciach oraz odczytanie tekstu z odnalezionych tablic. Dane zdobyliśmy głównie z reklam, ogłoszeń oraz fanpejdżów poświęconych samochodom.

## 2. Sposób działania

Na początku wczytujemy obraz i dostosowujemy odpowiednio jego rozdzielczość, po czym dokonujemy przetworzenia obrazu do skali szarości. Następnie po wstępnym usunięciu szumu dokonujemy morfologicznej operacji blackhat by wyróżnić litery i znaki na interesującym nas obrazie.

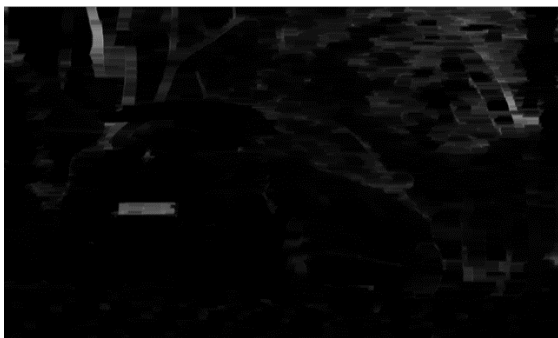


*Rysunek 1 Oryginalny obraz*



*Rysunek 2 Obraz po operacji blackhat*

W Kolejnym kroku używamy filtra Sobel przy zastosowaniu odpowiedniego progu (`cv2.threshold`) by wyróżnić miejsca, w których mogą znajdować się tablice, które następnie odszumiamy filtrem Gaussa, konwertujemy na czarno-biały oraz podkreślamy uzyskane kształty poprzez użycie serii transformacji erozji i dylatacji. Następnie używamy metody Otsu do znalezienia źródeł światła na obrazie i wykonujemy bitową operację AND pomiędzy uzyskanymi obrazami.



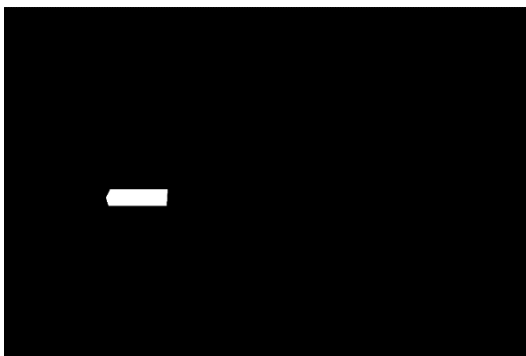
Rysunek 3 Obraz po operacji Sobel



Rysunek 4 Obraz po serii dylatacji i erozji oraz po porównaniu obrazu do obrazu z światłem

Przy tak uzyskanym obrazie przechodzimy do kolejnego działania, którym jest odnalezienie odpowiednich konturów. Wśród konturów uzyskanych przez cv2.findcontours. Wybieramy 20 największych konturów, aby pominąć pozostałe szumy na obrazie.

Wybieramy te których kształt po uproszczeniu ma proporcje zbliżone do proporcji typowych rejestracji samochodowych. Następnie nakładamy maskę na nasz obraz, aby wyciąć interesujący nas fragment z rejestracją,



Rysunek 5 Obraz po znalezieniu rejestracji



Rysunek 6 Wycięty fragment obrazu z rejestracją

Obraz z uzyskanego obszaru wczytujemy w skali szarości, odszumiamy filtrem dwustronnym oraz używamy na nim filtru Canny by znaleźć krawędzie liter. Szukamy konturów na obrazie i wybieramy te kontury, które mają odpowiednią wielkość i proporcje zbliżone do proporcji liter.



Rysunek 7 Przetworzona tablica do szukania liter



Rysunek 8 Znalezione litery na tablicy



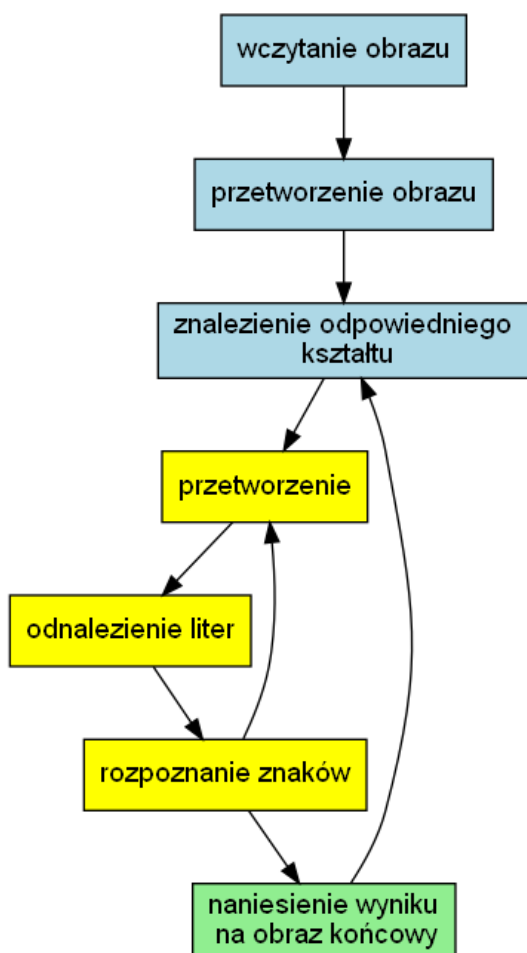
Rysunek 9 Wycięte litery do odczytania

Znalezione litery wycinamy z obrazu i konwertujemy go do czarni i bieli. Litery rozpoznajemy przy wykorzystaniu pytesseract OCR. Szukamy wyłącznie liter i cyfr. Odczytane litery wraz z konturem rejestracji umieszczamy na obrazie wynikowym.

Po tych operacjach rozpoczynamy poszukiwanie kolejnych potencjalnych rejestracji na obrazie i powtarzamy cały proces



Rysunek 10 Ostateczny wynik działania programu



Graf przekształceń przedstawiający plan działania naszego algorytmu. Kolor niebieski przedstawia elementy które zajmują się znalezieniem potencjalnych kandydatów na tablice, natomiast kolor żółty odpowiada za elementy obsługujące odpowiednie dostosowanie oraz odczytanie tekstu z potencjalnych tablic rejestracyjnych

### 3. Wyniki działania programu

Poniżej przedstawiamy wyniki działania naszego algorytmu dla zebranego zbioru 70 zdjęć.

Nasz algorytm poprawnie rozpoznał poprawnie 69 tablic rejestracyjnych, 4 tablice nie zostały rozpoznane oraz 4 obiekty zostały błędnie zinterpretowane jako tablica.

W przypadku rozpoznawaniu znaków braliśmy pod uwagę tylko te obszary które zostały wytypowane jako tablica, tzn. nie uznawaliśmy za błąd nieodczytanie liter z nieznalesionej tablicy. Algorytm odczytał poprawnie 492 znaki, błędnie odczytał 83 znaki, a 3 obrazy zostały błędnie zinterpretowane jako znak.

Rozpoznawanie tablic			
		Klasa rzeczywista	
Klasa przewidziana		pozytywna	negatywna
	pozytywna	69	4
	negatywna	4	0

**Skuteczność 90%**

Rozpoznawanie znaków			
		Klasa rzeczywista	
Klasa przewidziana		pozytywna	negatywna
	pozytywna	492	3
	negatywna	83	0

**Skuteczność 85%**



## Przykładowe wyniki działania algorytmu:





Przykładowe niedociągnięcia algorytmu:



## 4. Wnioski

O ile rozpoznanie tablicy w dobrych warunkach, przy zapewnieniu odpowiedniej rozdzielczości zdjęć, nie jest problemem, to jednak należy zwrócić uwagę, że algorytmy takie głównie są stosowane w zróżnicowanych warunkach – tj. Tablice odczytywane są z różnej odległości oraz często trzeba radzić sobie z elementami pogody, różnym oświetleniem a także rejestracjami znajdującymi się pod większym kontem względem obserwatora. Należy jednak też zwrócić uwagę na fakt, iż zdjęcia były różnych rozdzielczości i ogólnej jakości, co negatywnie wpłynęło na nasze osiągi, sytuacja taka nie występuje w faktycznych miejscach użytkowania tego typu algorytmów - dana kamera ma zazwyczaj stałe parametry rejestrowania obrazu.