

# Detekcja tablic rejestracyjnych

Dawid Rombel

Jest to projekt zaliczeniowy z cyfrowego przetwarzania obrazów w programie Matlab. Celem projektu jest wykrycie tablic rejestracyjnych na zdjęciach w których znajduje się pojazd.

Moje rozwiązanie polega na wykorzystaniu morfologicznej rekonstrukcji, dylatacji, erozji, domknięcia oraz metody Canniego do wykrycia krawędzi. Do sprawdzenia programu wykorzystany został zbiór 113 zdjęć.

Cały program ma dwie główne funkcje **getImageTransform** oraz **getPlate**. **getImageTransform** odpowiada za przekształcenie obrazu początkowego w obraz binarny na którym widać tablicę rejestracyjną, zaś **getPlate** analizuje obszary i wybiera ten który najbardziej przypomina wygląd tablicy rejestracyjnej.

## 1. Opis rozwiązania

### Opis **getImageTransform(imgOrg)**

W funkcji obraz wejściowy jest zamieniany na obraz w skali szarości, ma zmieniany rozmiar na 400x600 pikseli i jest używana funkcja **imadjust** dla zwiększenia wyrazistości krawędzi pomiędzy białym tłem tablicy, a jej napisami.

```
function imgRes = getImageTransform(imgOrg)
    img = rgb2gray(imgOrg);

    img = imresize(img, [400 600]);
    img = imadjust(img);
```



Elementy strukturalne oraz próg dla funkcji Canniego które później zostaną wykorzystane do operacji morfologicznych.

```
seRecon1 = ones(31,1);  
seRecon2 = ones(17,2);  
seClose = ones(1,12);  
cannyValue = 0.63;
```

Operacje morfologiczne

Każdy obraz przedstawia linię kodu, obrazy od lewej są adekwatne liniom kodu od góry.

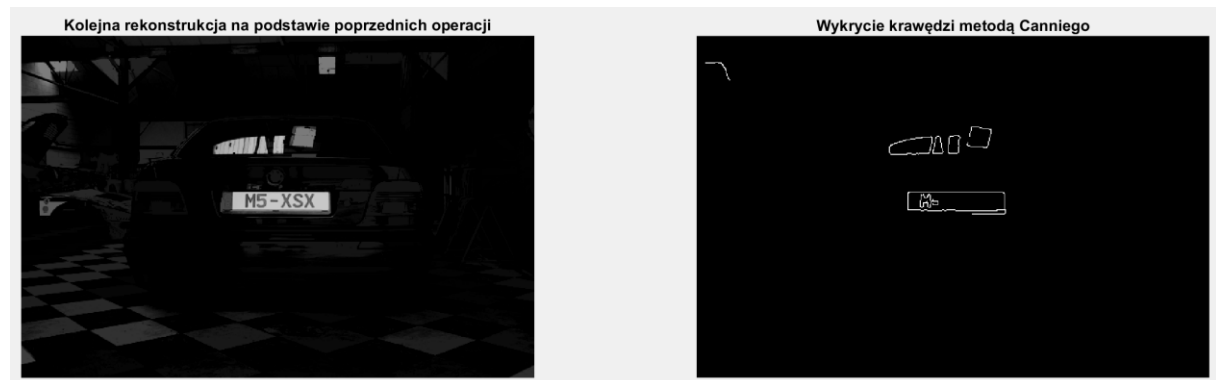
Etap odpowiadający za pozbycie się tła w obrazie.

```
imgRecon = imreconstruct(imerode(img, seRecon1), img);  
imgSub = imsubtract(img, imgRecon);  
imgRecon2 = imreconstruct(imerode(imgSub, seRecon2), imgSub);  
imgDilate = imdilate(imgRecon2, seRecon2);
```



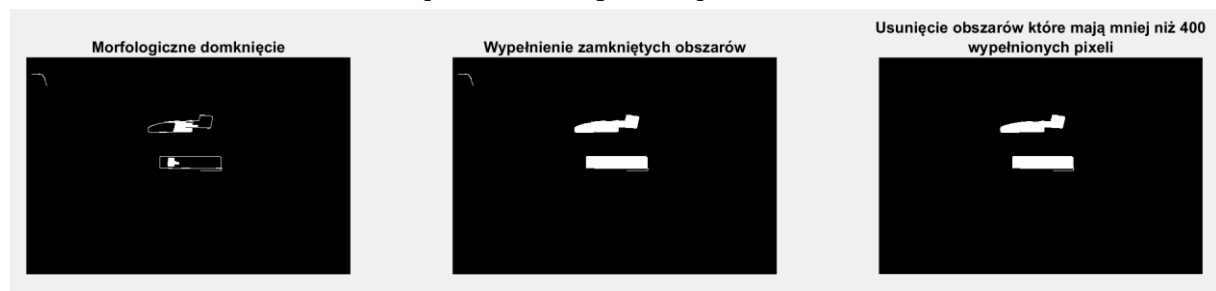
Etap wykrywania krawędzi.

```
img = imreconstruct(min(imgSub, imgDilate), imgSub);  
img = edge(img, 'canny', cannyValue);
```



Etap wypełnienia i selekcji obszarów.

```
img = imclose(img, seClose);  
img = imfill (img, "holes");  
img = bwareaopen(img, 400);
```



```

        Następnie jest przywracany rozmiar pierwotny obrazu
        sizeOrg = size(imgOrg);
        img = imresize(img, [sizeOrg(1) sizeOrg(2)]);
        imgRes = img;

```

Zwracany wynik to obraz binarny z obszarami które mogą być tablicami rejestracyjnymi.

Funkcja **getPlate(img)**

```

function best = getPlate(img)
    stats = regionprops(img, 'FilledArea', 'Area', 'BoundingBox', 'Centroid');
    best = [];
    bestScore = 0;
    for j = 1:size(stats)
        stat = stats(j);
        m = stat.BoundingBox;
        score = stat.FilledArea / (m(3) * m(4));

        if ( m(3)/m(4) > 2.3 && score > bestScore )
            best = stat;
        end
    end
end

```

Sprawdza rejony w obrazie, i szuka tego najbardziej przypominającego tablice rejestracyjną. Funkcja uwzględnia stosunek wysokości do długości oraz wypełnienie obszaru.

Po uzyskaniu wszystkich potrzebnych danych, sprawdzane one są z oczekiwanym wynikiem.

```

centroid = best.Centroid;

posX = str2num(text{2}); posY = str2num(text{3});
width = str2num(text{4}); height = str2num(text{5});
textCentorid = [(width/2)+posX (height/2)+posY];

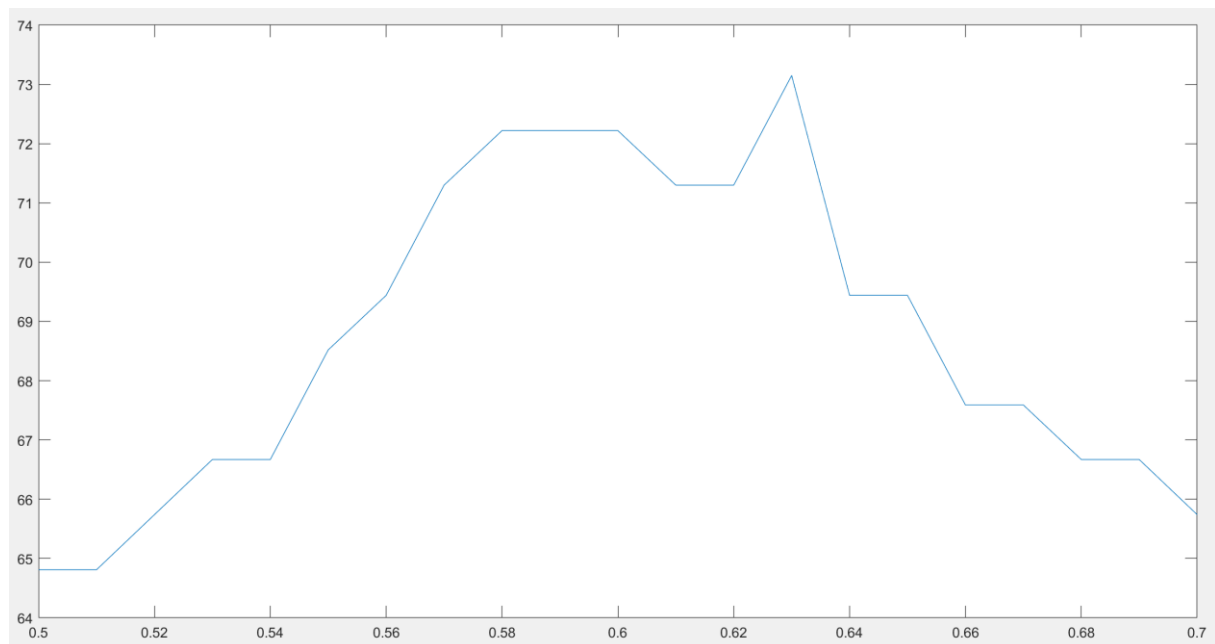
marginV = 0.1;
margin = [width*marginV height*marginV];
difference = centroid - textCentorid;
result = margin > difference;
if (result == true)
    counter = counter + 1;
end

```

Po obliczeniu centroida z wyniku, jest on porównywany z aktualnym. Jeżeli margines błędu nie jest większy niż 10% z pozycji wczytanego wyniku, to uznaje się że pozycja jest prawidłowa.

## 2. Ilościowa i jakościowa analiza wyników

Algorytm ten pokazuje że ma 73,15% skuteczność w prawidłowym wykrywaniu tablic. Dla analizy specjalnie wykonałem pomiar dla progu CANNIEgo od 0,5 do 0,7 z przeskokiem 0,01.



Na osi X jest przeskok o 0,01. Na osi Y rezultaty pomiarów w procentach.

### 3. Wnioski, propozycje ulepszeń

Zauważyłem że w zależności jak dostosujemy elementy strukturalne wykrywanie dla różnych obrazów może być lepsze, a dla niektórych gorsze. Ulepszenie mogłoby polegać na zmianie istniejącego sposobu wykrywania tej tablicy, albo przeprowadzenie analizy gdzie są wszystkie parametry testowane z każdym, np. sprawdzić elementy strukturalne w (1-30, 1-5) i Caniego 0,3-0,7 oraz parę innych parametrów. Gdy policzyłem wszystkie kombinacje to wyszło ich ok 3 milionów, a na to potrzeba bardzo dużo czasu. Innym ulepszeniem mogłoby być użycie sztucznej inteligencji lub klasyfikatorów.