

TASK 2  
Dokumentacja  
MSKN Decybel  
02.12.2023

===

Program utworzono zgodnie z wymogami zadania. Kolejne fragmenty zawierają:

```
%%Wczytanie do zmiennej obrazków z aktualnego folderu
% Autor: Wiktoria Marczuk, MSKN Decybel
% Skrypt wymaga umieszczenia go bezpośrednio w folderze ze zdjęciami do
% testów
imagefiles = dir('*.png');
n_files=length(imagefiles);
saveToFile=struct;
```

Funkcje wczytania struktury plików

```
%% pętla główna iterująca po obrazkach z folderu
for ii=1:n_files
    currentfilename = imagefiles(ii).name;
    przykladowy_obrazek = imread(currentfilename);

    %% pierwszy thresholding i tworzenie maski całego obrazka przy użyciu Color Thresholder App
    I = przykladowy_obrazek;
    % Define thresholds for channel 1 based on histogram settings
    channel1Min = 142.000;
    channel1Max = 187.000;

    % Define thresholds for channel 2 based on histogram settings
    channel2Min = 109.000;
    channel2Max = 164.000;

    % Define thresholds for channel 3 based on histogram settings
    channel3Min = 58.000;
    channel3Max = 130.000;

    % Create mask based on chosen histogram thresholds
    sliderBW = (I(:, :, 1) >= channel1Min ) & (I(:, :, 1) <= channel1Max) & ...
        (I(:, :, 2) >= channel2Min ) & (I(:, :, 2) <= channel2Max) & ...
        (I(:, :, 3) >= channel3Min ) & (I(:, :, 3) <= channel3Max);
    BW = sliderBW;

    %% operacje morfologiczne
    SE = strel("disk",2);
    otwarcie_obrazka = imopen(BW, SE);
    SE = strel("disk",5);
    zamkniecie_przykladowego_obrazka = imclose(otwarcie_obrazka, SE);
```

Następnie w pętli for następuje analiza pod kątem koloru żółtego i innych specyficznych cech charakterystycznych dla znaku „Droga z pierwszeństwem D-1”. Dodatkowo zawarto operację otwarcia i zamknięcia.

```
%% wyszukiwanie obszarów na obrazie mogących być znakiem drogowym
hblobanalysis = vision.BlobAnalysis('MinimumBlobArea',200,'MaximumBlobArea',3000);
[obj_area, obj_centroid, bbox_oryginalny_obrazek] = step(hblobanalysis,
zamkniecie_przykladowego_obrazka);

%% zerowanie bounding box które nie są kwadratami
for i=1:height(bbox_oryginalny_obrazek)
    if bbox_oryginalny_obrazek(i,3)<=bbox_oryginalny_obrazek(i,4)*1.1 &&
bbox_oryginalny_obrazek(i,3)>=bbox_oryginalny_obrazek(i,4)*0.9
        bbox_oryginalny_obrazek(i,:);
    else
        bbox_oryginalny_obrazek(i,:)=0;
    end
end

%% usuwanie wierszy które są zerami z wektora bounding boxów
bbox=bbox_oryginalny_obrazek.';
SBW_col=sum(bbox,1);
col_left=find(SBW_col,1,"first");
col_right=find(SBW_col,1,"last");
bbox=bbox(:,col_left:col_right);
bbox_oryginalny_obrazek=bbox.';

%% deklaracja wektora pomocniczego
procentowy_udzial_bieli_w_calej_masce = zeros(1, height(bbox));
```

Oczywistym jest, że na obrazie wystąpić może kilka obszarów z kolorem żółtym. Należy sprawdzić, które z nich są zbliżone do kwadratu, następnie ma miejsce operacja szukania bieli – jeśli w okolicy można znaleźć sporo pikseli białych, jest to prawdopodobnie szukany znak. Należy dodać, że zdecydowano się na konwencjonalne operacje macierzowe, bez wykorzystania sieci neuronowych – rozwiązanie to jest nieco trudniejsze w implementacji, ale może dać porównywalną, a nawet lepszą dokładność przy wielokrotnie mniejszym koszcie obliczeniowym i wymaganiach sprzętowych.

```
%% pętla iterująca po bounding boxach zakategoryzowanych jako mogące zawierać znak
drogowy
for i=1:height(bbox_oryginalny_obrazek)

    bbox = bbox_oryginalny_obrazek;
    bbox(i,1) = bbox(i,1);
    bbox(i,2) = bbox(i,2);
    bbox(i,3) = bbox(i,3);
    bbox(i,4) = bbox(i,4);
```

```

obrazek_przyciety_w_miejscu_gdzie_maska = imcrop(przykladowy_obrazek,bbox(i,:));

%% ponowny thresholding kolorów dla mniejszych obrazków
I = obrazek_przyciety_w_miejscu_gdzie_maska;

% Define thresholds for channel 1 based on histogram settings
channel1Min = 147.000;
channel1Max = 161.000;

% Define thresholds for channel 2 based on histogram settings
channel2Min = 127.000;
channel2Max = 167.000;

% Define thresholds for channel 3 based on histogram settings
channel3Min = 153.000;
channel3Max = 166.000;

% Create mask based on chosen histogram thresholds jeśli nie ma obrazka
% to przypisujemy wartości bbox na zera bo nie jest możliwe wykrycie
try
sliderBW = (I(:, :, 1) >= channel1Min ) & (I(:, :, 1) <= channel1Max) & ...
(I(:, :, 2) >= channel2Min ) & (I(:, :, 2) <= channel2Max) & ...
(I(:, :, 3) >= channel3Min ) & (I(:, :, 3) <= channel3Max);
BW = sliderBW;
catch
powiekszona_granica=[0 0 0 0];
end

%% operacje morfologiczne
SE = strel("disk",2);
maska_malego_obrazka_w_iteracji = imclose(BW, SE);

s = size(maska_malego_obrazka_w_iteracji);
procentowy_udzial_bieli_w_calej_masce(i) =
sum(sum(maska_malego_obrazka_w_iteracji))/(s(1)*s(2));

end

% obliczenie pomocniczej wartości do wyboru bardziej prawdopodobnych
% zdjęć
[maksymalna_wartosc_bialych_pikseli, indeks_obrazka_z_maksymalna_wartoscia_bieli] =
max(procentowy_udzial_bieli_w_calej_masce);

%% skalowanie bboxa aby pokrywał cały znak (łącznie z białą obwódką)
try
przykladowy_obrazek_z_wybrany_bbox = insertShape(przykladowy_obrazek,
"rectangle",bbox_oryginalny_obrazek(indeks_obrazka_z_maksymalna_wartoscia_bieli,:),
"LineWidth",4);

```

```

%% obliczone powiększone granice dla białej obwódki na podstawie proporcji znaku
powiekszona_granica =
[bbox_oryginalny_obrazek(indeks_obrazka_z_maksymalna_wartoscia_bieli,1) -
bbox_oryginalny_obrazek(indeks_obrazka_z_maksymalna_wartoscia_bieli,3)*0.37, ...
    bbox_oryginalny_obrazek(indeks_obrazka_z_maksymalna_wartoscia_bieli,2) -
bbox_oryginalny_obrazek(indeks_obrazka_z_maksymalna_wartoscia_bieli,4)*0.37, ...

bbox_oryginalny_obrazek(indeks_obrazka_z_maksymalna_wartoscia_bieli,3)*1.74,
bbox_oryginalny_obrazek(indeks_obrazka_z_maksymalna_wartoscia_bieli,4)*1.74];

catch
% jeżeli nie wykryto żadnego bbox to ustalamy pustą wartość
powiekszona_granica=[0 0 0 0];

end

```

Dla wyciętego ROI ma miejsce ponowny thresholding, ma miejsce ponowna kontrola i przeskalowanie obrazka zgodnie z wymiarami określonymi w normach – pozwala to objąć całość znaku, czyli kolor żółty i biały

```

%% Zapisywanie do struktury według wytycznych
saveToFile(ii).image=currentfilename;
saveToFile(ii).BoundingBox=powiekszona_granica;
saveToFile=transpose(saveToFile);

end

%% Zapisywanie do pliku
save Task_2_MSKNDecybel.mat saveToFile

```

Ostatni fragment programu to zapisanie wyników zgodnie z wymogami Organizatorów