

Analiza obrazów
Sprawozdanie nr 2 z ćwiczeń laboratoryjnych
Mikołaj Marchewa

Wydział Fizyki i informatyki Stosowanej
Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
14 grudnia 2020

1. Laboratorium 5

W trakcie trwania laboratorium nr 5 zapoznaliśmy się z operacjami morfologicznymi, które pozwoliły nam na analizę kształtów, segmentację, a także zliczenie podobnych do siebie obiektów znajdujących się na obrazie. By wykonać kolejne ćwiczenia wykorzystaliśmy poniższe zdjęcie kaczek:

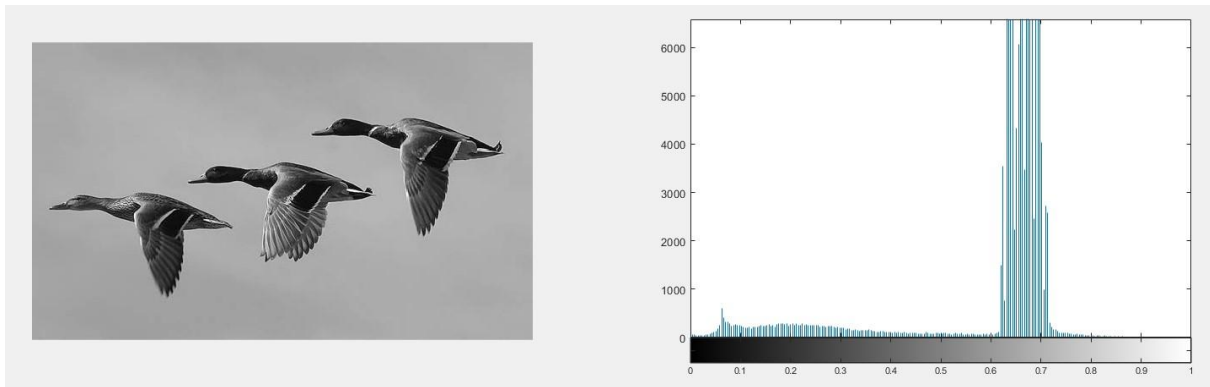
```
im = double(imread('kaczki.jpg'))/255;
```



Rys. 1: Oryginał zdjęcia wykorzystanego do ćwiczeń.

Pierwszym krokiem była zmiana kolorów na skalę szarości. Dodatkowo utworzyliśmy także histogram, dzięki któremu uzyskaliśmy wskazówki do dalszej analizy.

```
gim = rgb2gray(im);  
  
subplot(1,2,1);  
imshow(gim);  
subplot(1,2,2);  
imhist(gim);
```



Rys. 2: Oryginał obrazu w skali szarości wraz z histogramem

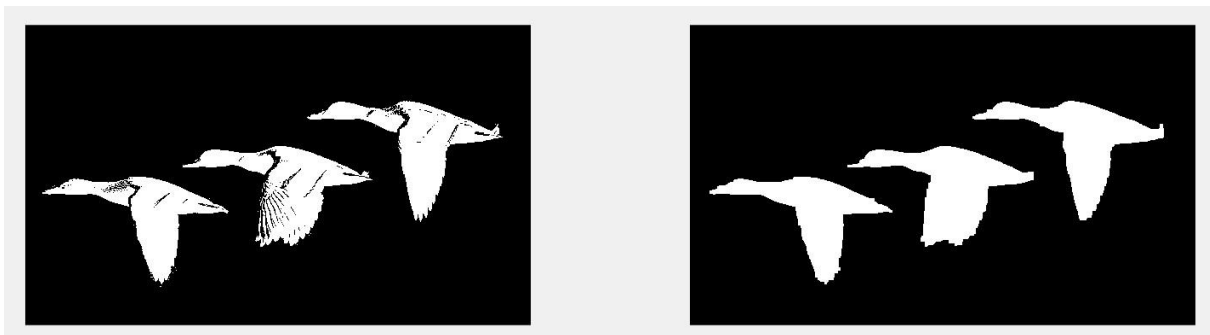
Histogram z Rys.2 pozwolił nam na intuicyjne wyznaczenie odpowiedniego progu binaryzacji obrazu (0.61).

```
bim = ~imbinarize(gim, 0.61);
subplot(1,2,1);
imshow(bim);
```

Dodatkowo by uzyskać jedynie kształty interesujących nas obiektów, bez zanieczyszczeń w postaci pojedynczych czarnych pikseli wykonaliśmy kolejno operacje zamknięcia i otwarcia. Rezultat znajduje się na Rys.3.

```
bim = imclose(bim,ones(7));
bim = imopen(bim,ones(3));

subplot(1,2,2);
imshow(bim);
```



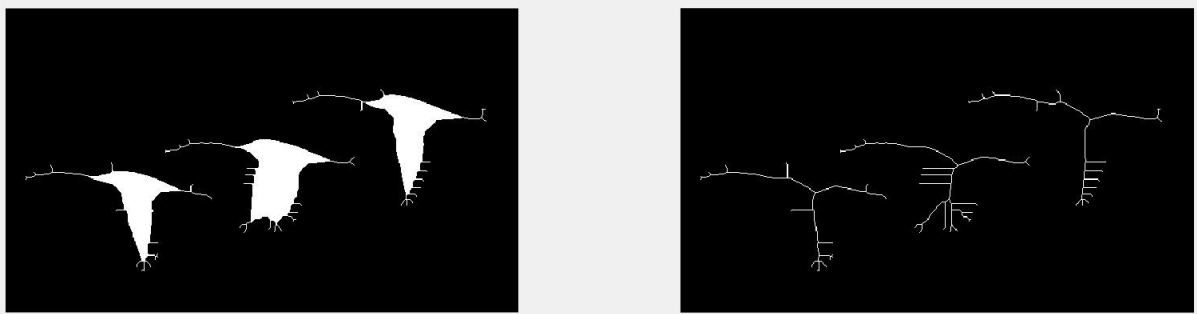
Rys. 3: Binarizacja z progiem ustalonym na podstawie histogramu obrazu (po lewo) oraz oczyszczony obraz po operacji binaryzacji, zamknięcia i otwarcia (po prawo).

1.1 Szkieletyzacja

W kolejnej części zajęć laboratoryjnych z wykorzystaniem funkcji *bwmorph* poznaliśmy działanie operacji szkieletyzacji ('skel'). Operacja ta polega usuwaniu pikseli, a jej wynikiem jest zbiór punktów znajdujący się w takiej samej odległości od krawędzi. Trzeci argument, które przyjmuje ta funkcja jest n – liczba powtórzeń wykonywanej operacji. Pozwala na przyjrzenie się dokładniej stopniowym zmianom na obrazie.

```
ten = bwmorph(bim, 'skel', 10);
subplot(1,2,1);
imshow(ten);

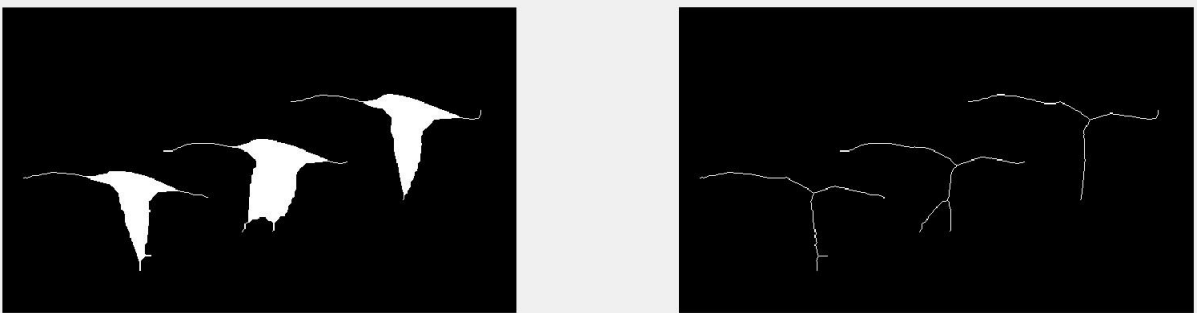
infi = bwmorph(bim, 'skel', Inf);
subplot(1,2,2);
imshow(infi);
```



Rys. 4: Po lewo: uzyskany szkielet po 10 powtórzeniach operacji 'skel'. Po prawo: szkielet kaczek uzyskany po uzyskaniu momentu, w którym kolejne wykonanie operacji 'skel' nie zmienia obrazu.

1.2 Operacja „odchudzania”

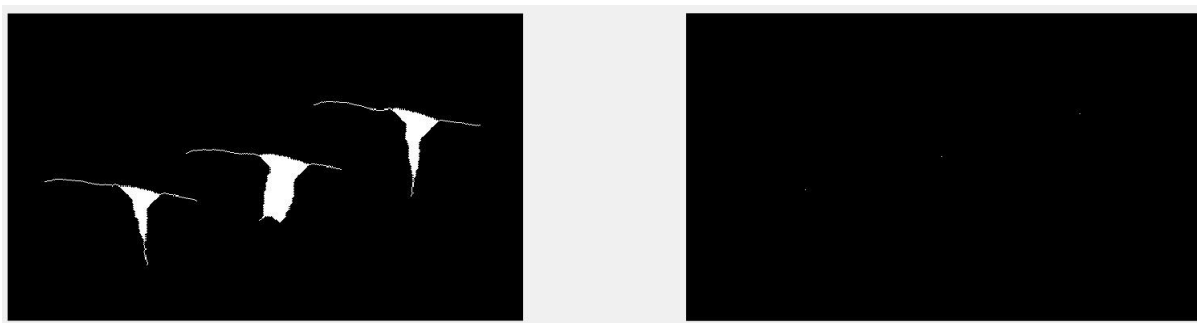
Zbliżony efekt do szkieletyzacji daje operacja 'thin' – jest to erozja z zapewnieniem, że pozostała linia będzie nieprzerwana. Kod jest analogiczny do podpunktu 1.1.



Rys. 5: Po lewo: uzyskany szkielet po 10 powtórzeniach operacji 'thin'. Po prawo: szkielet kaczek uzyskany po uzyskaniu momentu, w którym kolejne wykonanie operacji 'thin' nie zmienia obrazu.

1.3 Operacja „zwężania”

Operacja 'shrink' zwęża obiekty na obrazie do uzyskania pojedynczych punktów - dla obiektów bez „dziur” wewnątrz oraz do uzyskania pierścieni dla obiektów zawierających „dziurę”. Kod również analogiczny do podpunktu 1.1.



Rys. 6: Po lewo: uzyskany szkielet po 10 powtórzeniach operacji 'skel'. Po prawo: szkielet kaczek uzyskany po uzyskaniu momentu, w którym kolejne wykonanie operacji 'skel' nie zmienia obrazu.

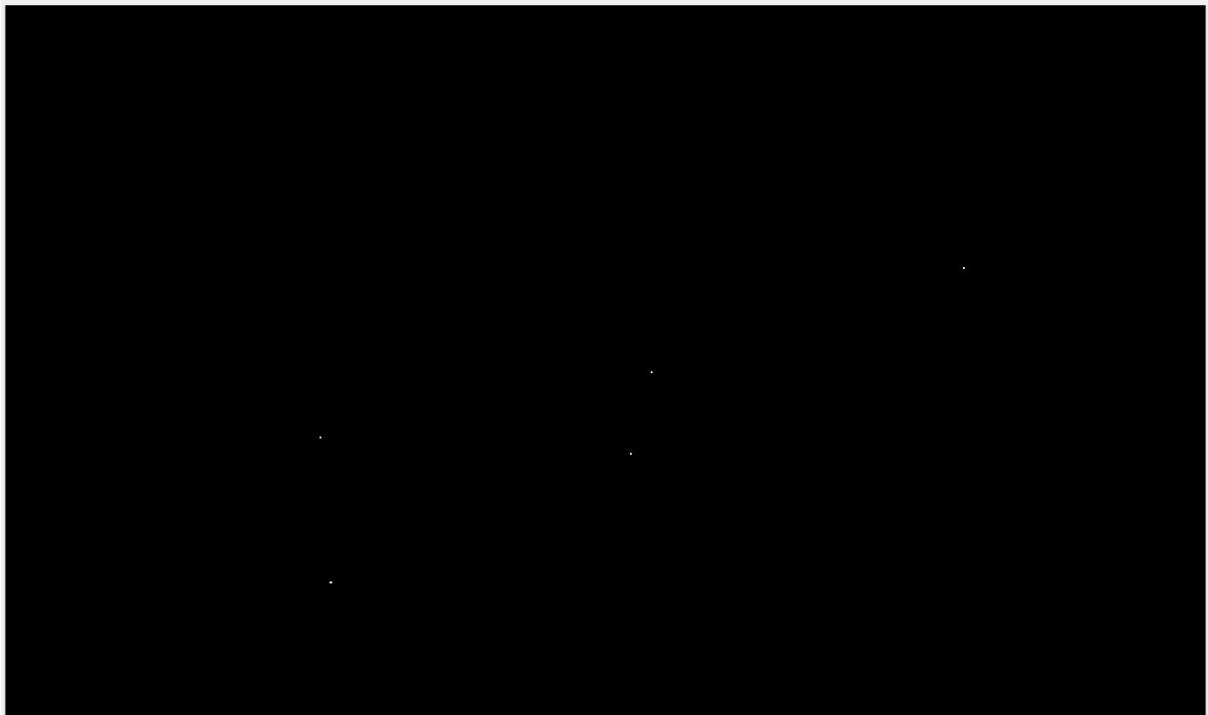
1.4 Punkty odgałęziania szkieletu

Operacja 'branchpoints' wykonana po szkieletyzacji pozostawia na obrazie punkty będące miejscami powstawania odgałęzień szkieletu.

```
infi = bwmorph(infi, 'branchpoints');
```



Rys. 7: Punkty odgałęzień szkieletu powstałem po wykorzystaniu operacji 'skel'.



Rys. 8: Punkty odgałęzień szkieletu powstałe po wykorzystaniu operacji 'thin'.

1.5 Punkty końcowe

Ponownie korzystając z funkcji *bwmorph*, sprawdziliśmy działanie kolejnej operacji – 'endpoint'. Jej celem jest pozostawienie na obrazie zakończeń każdej z gałęzi uzyskanych po szkieletyzacji.

```
infi = bwmorph(infi, 'endpoints');
```



Rys. 9: Punkty końcowe powstałe po wykorzystaniu operacji 'skel'.

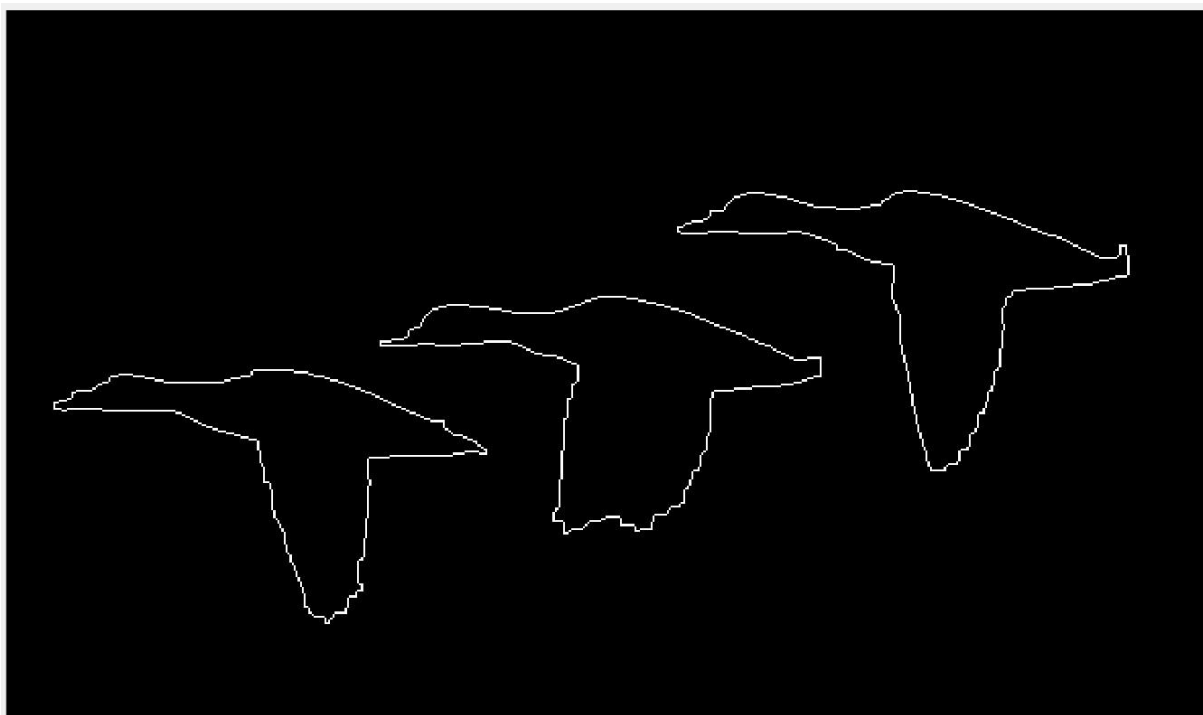


Rys. 10: Punkty końcowe powstałe po wykorzystaniu operacji 'thin'.

1.6 Operacja 'remove'

Operacja 'remove' usuwa wewnętrzne piksele obiektów na obrazie. Wykorzystywane jest tu sąsiedztwo von Neumanna – jeśli 4 sąsiadujące ze sobą piksele przyjmują wartość 1 – wówczas piksel środkowy jest ustawiany na 0.

```
bim = bwmorph(bim, 'remove');
```



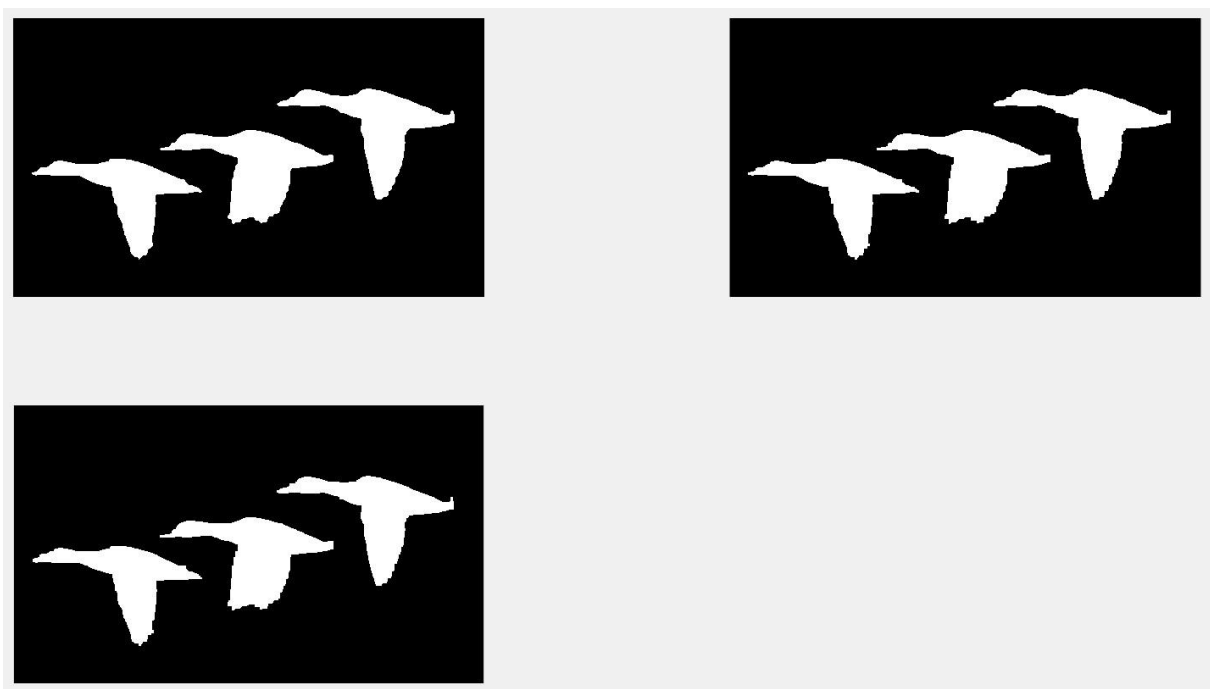
Rys. 11: Efekt działania operacji 'remove'.

1.7 Operacja oczyszczające obraz

W trakcie trwania laboratoriów poznaliśmy również inne wbudowane operacje pozwalające na oczyszczenie obrazu przed jego analizą.

- 'majority' – ustawia piksel środkowy na wartość 1, gdy pięć lub więcej pikseli w jego otoczeniu 3x3 również ma wartość 1,
- 'fill' – ustawia piksel środkowy na wartość 1, gdy piksele w jego otoczeniu przyjmują wartość 1,
- 'clean' – odwrotność operacji 'fill'.

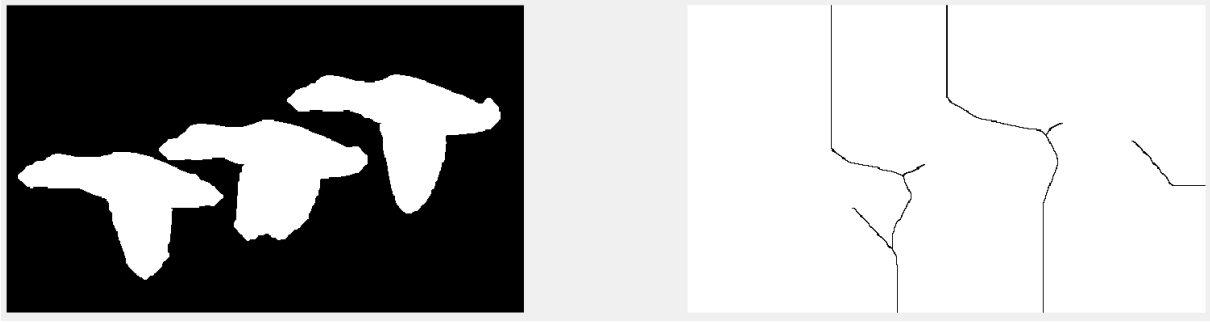
Efekty działania powyższych operacji przedstawiono poniżej:



Rys. 12: Efekt działania operacji: 'majority' – lewy górny róg, 'fill' – prawy górny róg, 'clean' – lewy dolny róg rysunku.

1.8 Operacja pogrubienia

Przeciwieństwem operacji zwężania jest operacja pogrubienia – 'thicken'. Jej wynikiem działania są obiekty pogrubione o swoje zewnętrzne krawędzie, z zachowaniem rozdzielności sąsiadujących ze sobą obiektów.

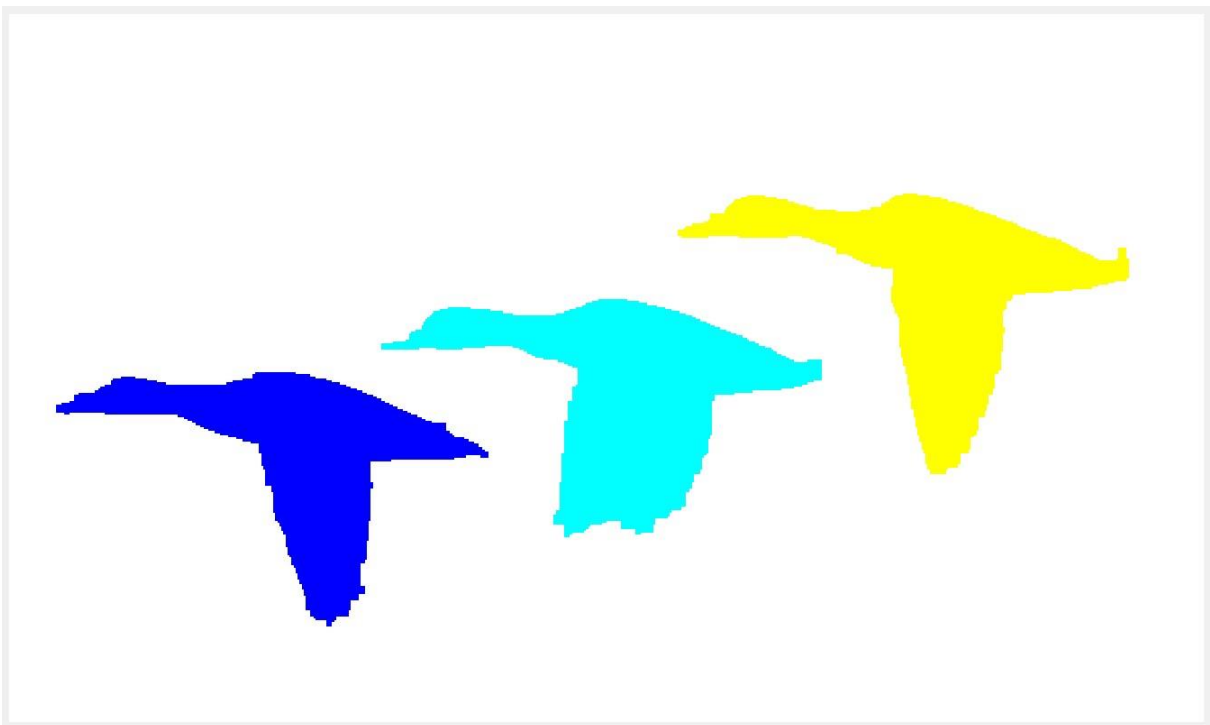


Rys. 13: Efekt działania operacji 'thicken' po 10 powtórzeniach (lewo) i po ilości powtórzeń dla których wykonanie każdej kolejnej operacji 'thicken' nie zmienia obrazu.

1.9 Segmentacja obrazu

Segmentacja obrazu polega na podziale obrazu na obszary jednorodne zgodnie z pewnym wybranym kryterium. Do ćwiczeń wykorzystaliśmy funkcję `bwlabel`, która pozwoliła nam na nadanie etykiety poszczególnym segmentom obrazu, co z kolei symbolizują różne kolory obiektów na Rys.14.

```
rim = bwlabel(bim);
imshow(label2rgb(rim));
```



Rys. 14: Efekt działania funkcji `bwlabel`.

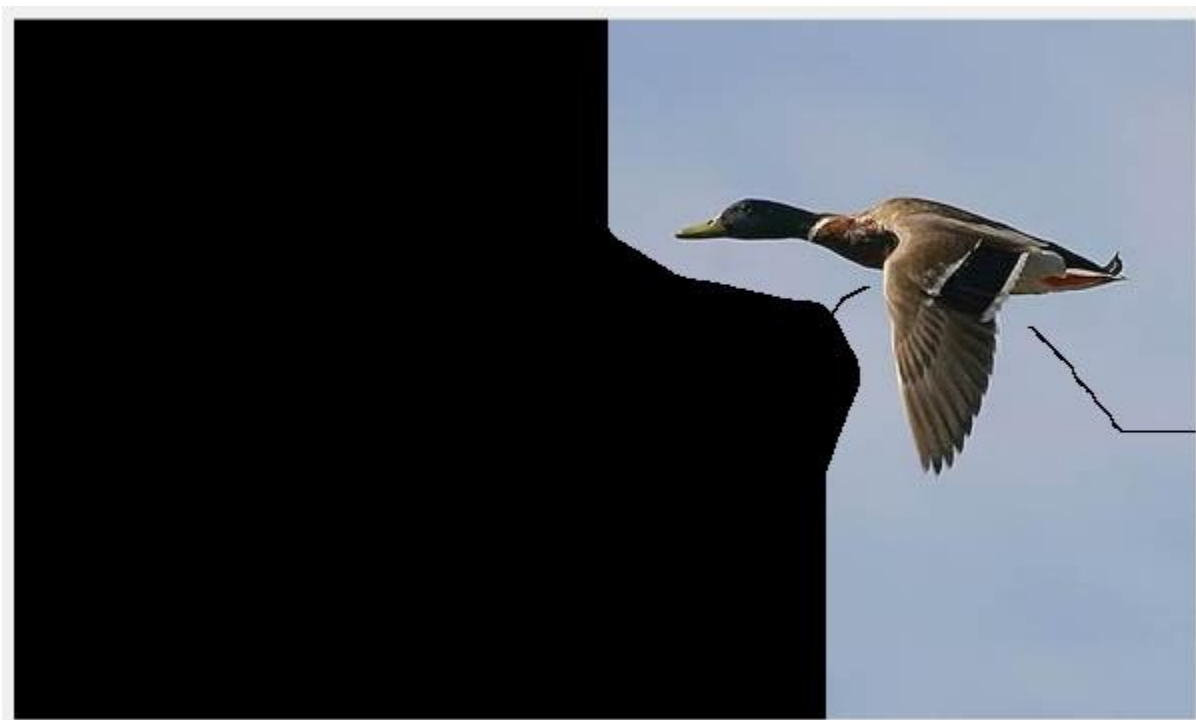
Wykorzystując bitmapę z etykietami `rim`, możemy wyodrębnić dowolnie wybraną z kaczek na oryginalnym obrazie.

```
imshow(im.*(rim == 3));
```




Rys. 15: Wyodrębniony obiekt z oryginalnego obrazu, któremu funkcja `bwlabel` nadana indeks 3.

Po połączeniu wyniku działania operacji pogrubiania oraz funkcji `bwlabel` możemy w prosty sposób uzyskać wybraną kaczkę wraz z jej otoczeniem:



Rys. 16: Wyodrębniony obiekt wraz z otoczeniem z oryginalnego obrazu, któremu funkcja `bwlabel` nadana indeks 3.

1.10 Transformata odległościowa

Kolejnym zagadnieniem poznany w czasie zajęć była transformata odległościowa. Wynikiem działania tej operacji jest bitmapa zawierająca informację o odległości poszczególnych pikseli na obrazie od obiektu. Tworzymy w ten sposób „mapę topograficzną” naszego obrazu. Transformatę odległościową możemy wyliczyć z wykorzystaniem różnych metryk, z których najbardziej powszechną jest metryka Euklidesowa.

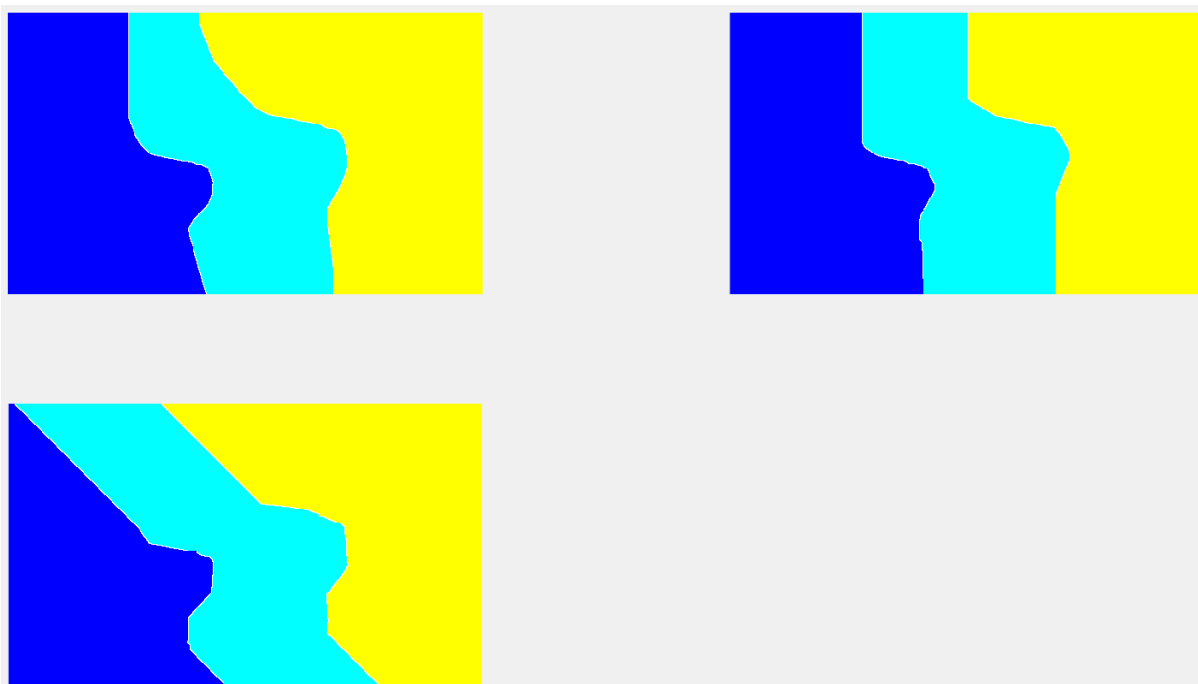


Rys. 17: Transformata odległościowa z metryką Euklidesową.

1.10 Segmentacja wododziałowa

Ostatnim z poznanych zagadnień była segmentacja wododziałowa. Pełni ona podobną funkcję do operacji ‘thicken’ dzieląc obraz na obszary w ilości równej ilości obiektów na danym obrazie.

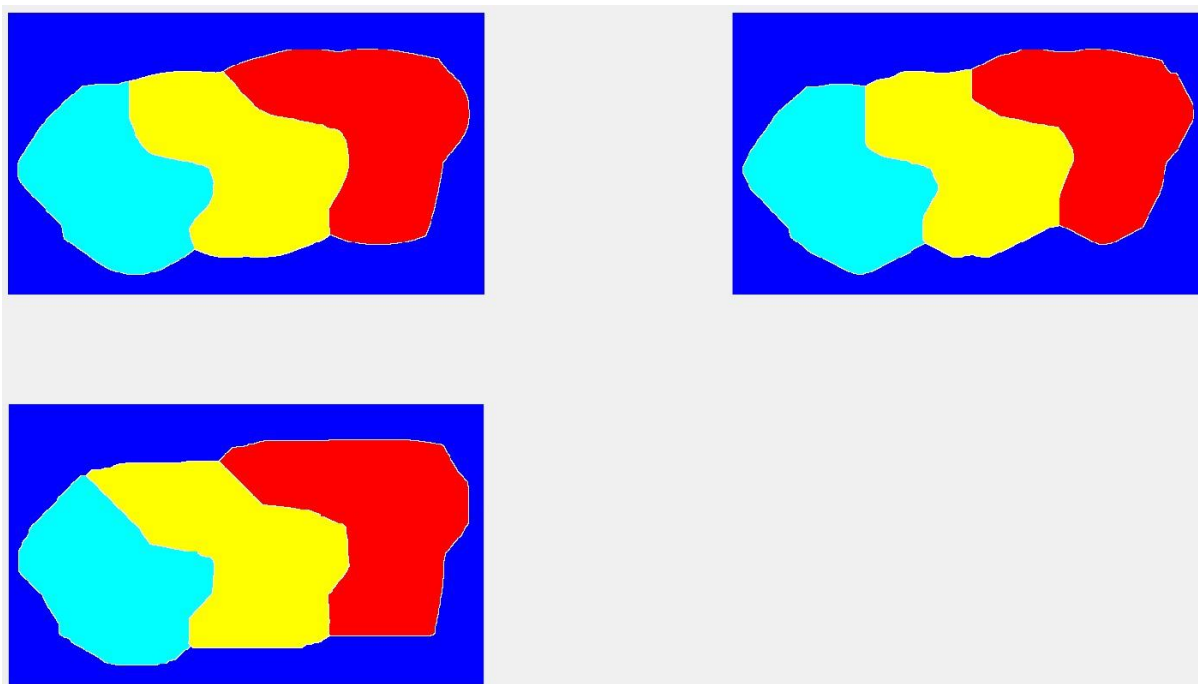
```
a = bwdist(bim, 'cityblock');  
a = watershed(a);  
subplot(2,2,1);  
imshow(label2rgb(a));
```



Rys. 18: Segmentacja wododziałowa w metryce Euklidesowej – lewy górny róg, w metryce Manhattan – prawy górny róg, w metryce Chebysheva – lewy dolny róg rysunku.

Dodając ramkę wokół obrazu złożoną z białych pikseli finalnie otrzymaliśmy segmentację wododziałową z czterema wydzielonymi obszarami, co przedstawia Rys.19.

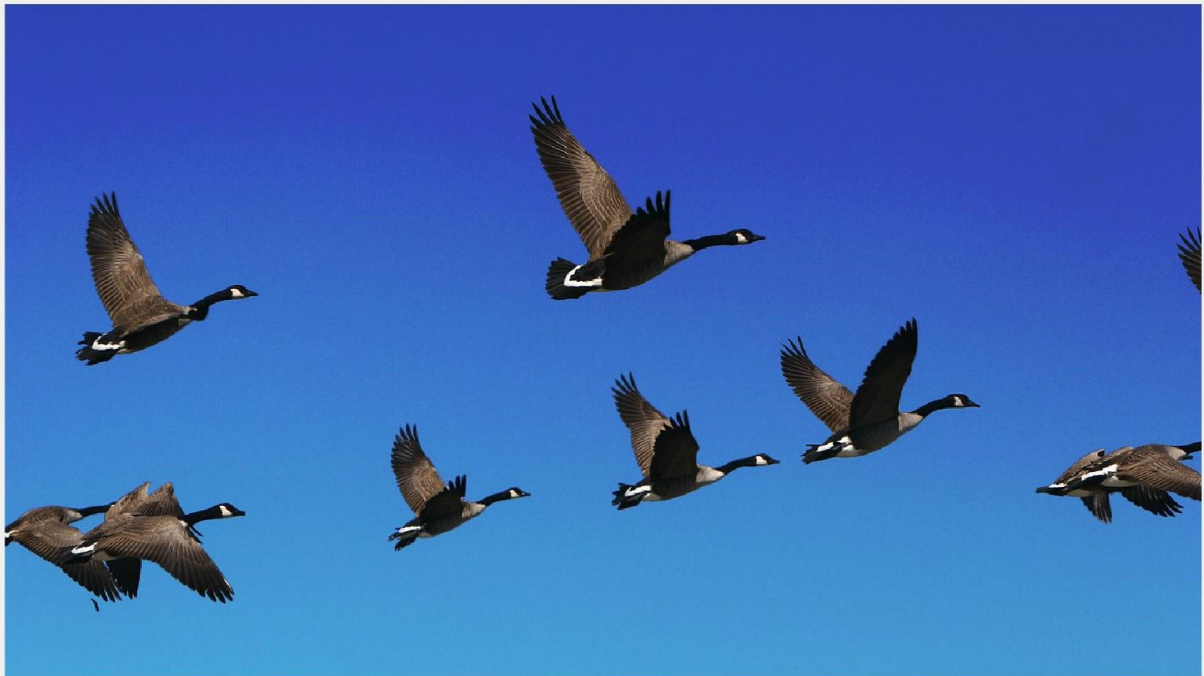
```
bim(:, [1, end]) = 1;  
bim([1, end], :) = 1;
```



Rys. 19: Segmentacja wododziałowa z nałożoną na obraz białą ramką oraz w metryce Euklidesowej – lewy górny róg, w metryce Manhattan – prawy górny róg, w metryce Chebysheva – lewy dolny róg rysunku.

1. Laboratorium 6

Laboratorium nr 6 było kontynuacją poznanych technik analizy obrazu z poprzedniego laboratorium, a także okazją do przyjrzenia się bliżej współczynnikom geometrycznym obiektów znajdujących się na zdjęciu. Za punkt wyjściowy przyjęliśmy zdjęcie z Rys.20.



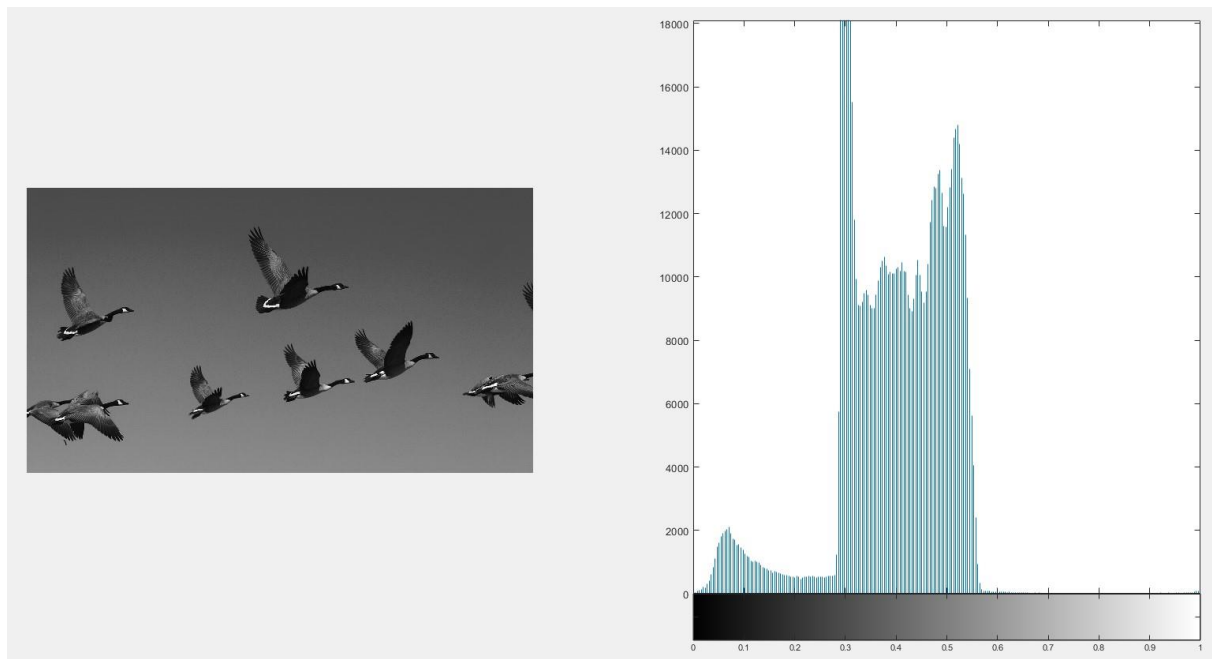
Rys. 20: Obraz oryginalny.

Postępując analogicznie do początku poprzednich laboratoriów dla oryginalnego obrazu sprawdziliśmy jego histogram po uprzedniej zmianie koloru na odcienie szarości. Tym razem jednak obraz okazał się mniej trywialny w binaryzacji. Histogram z Rys.21 ukazał szumy, których chcemy uniknąć w trakcie binaryzacji i późniejszej segmentacji.

```
im = double(imread('ptaki.jpg'))/255;
imshow(im);
gim = rgb2gray(im);

subplot(1,2,1);
imshow(gim);

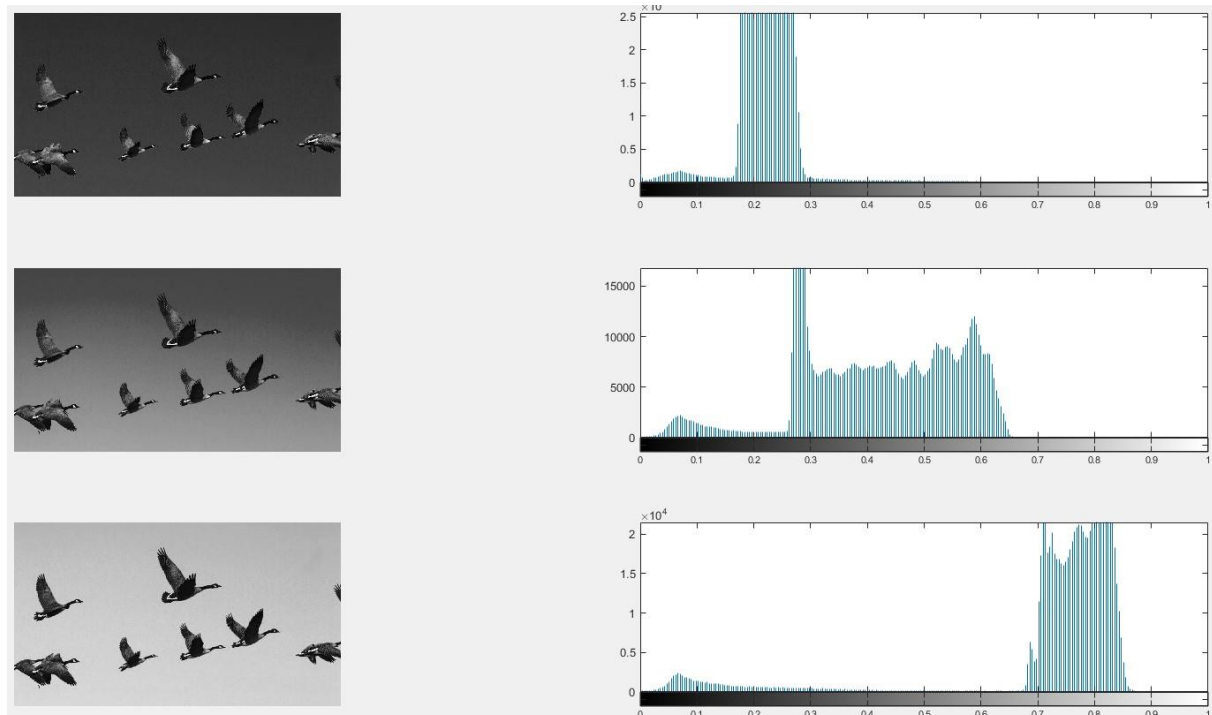
subplot(1,2,2);
imhist(gim);
```



Rys. 21: Oryginał obrazu w skali szarości wraz z histogramem.

By uniknąć zniekształcenia ostatecznego wyniku sprawdziliśmy jak kształtują się wykresy poszczególnych kanałów kolorów:

```
for i = 1:3
    subplot(3,2,2*i-1);
    imshow(im(:,:,i));
    subplot(3,2,2*i);
    imhist(im(:,:,i));
end
```



Rys. 22: Kolejne kanały kolorów RGB w skali szarości i histogramy, pochodzące z oryginalnego obrazu.

Rezultat uzyskany na Rys.21 pozwolił na lepszą ocenę wpływu poszczególnych kanałów kolorów na obraz. Z punktu widzenia dalszej analizy najbardziej przydatne okazały się kanał czerwony i niebieski, dla których ustaliliśmy progi binaryzacji, zbinaryzowaliśmy je, a następnie połączyliśmy w jeden binarny obraz. Rys. 22 doskonale obrazuje na ile powyższe operacje pomogły w uzyskaniu lepszego materiału do segmentacji i analizy obrazu.



Rys. 23: Zbinaryzowany i oczyszczony obraz oryginalny z wykorzystaniem wszystkich kanałów RGB (po lewo) oraz zbinaryzowany i oczyszczony obraz po dobraniu odpowiednich kanałów do binaryzacji (po prawo).

Z tak przygotowanym obrazem binarnym przeszliśmy do zapoznania się ze współczynnikami geometrycznymi. W tym celu użyliśmy funkcji *regionprops*, by odnaleźć lokalizację i wymiary „okienek” zawierających poszczególne obiekty na obrazie. Kolejnym krokiem było utworzenie macierzy do której wpisywaliśmy kolejne wartości obliczonych współczynników geometrycznych. Poznanymi współczynnikami były kolejno:

1. Współczynnik Blair-Bliss’a → stosunek średniej odległości pikseli od środka obiektu do średniej odległości pikseli od środka koła o tym samym polu,
2. współczynnik Danielsson’a → stosunek średniej odległości pikseli od krawędzi obiektu do średniej odległości pikseli od krawędzi koła o tym samym polu,
3. współczynnik Feret’a → rozpiętości w układzie współrzędnych/długości,
4. współczynnik Haralick’a → stosunek średniej odległości pikseli na krawędzi od środka obiektu do średniej odległości pikseli na krawędzi od środka koła o tym samym polu,
5. współczynnik Malinowskiej → stosunek obwodu obiektu do pierwiastka z $4 * \pi * S$ pomniejszonego o 1, gdzie S jest powierzchnią badanego obiektu.

```
a = regionprops(bim, 'all');
fun = {@BlairBliss, @Danielsson, @Feret, @Haralick, @Malinowska};
w = zeros(length(a), length(fun));

for i = 1:length(a)
    for j = 1:length(fun)
        w(i,j) = fun{j}(a(i).Image);
    end
end
```

	1	2	3	4	5
1	6.8833	107.0568	0.5078	80.3237	0.8806
2	6.0071	93.1480	0.9637	73.8036	0.7699
3	1.6442	14.9911	1.6250	4.6556	0.1801
4	5.3945	89.2812	0.8882	55.3935	0.8187
5	6.6645	106.3109	0.8651	91.3402	0.9176
6	5.7364	85.6468	0.8258	64.5639	0.8612
7	6.2503	116.7577	0.7324	71.1582	0.9834
8	5.8982	77.5783	0.4972	58.7440	0.6051
9	3.3761	27.7870	2.7692	22.1400	0.7882

Rys. 24: Wartości kolejnych współczynników geometrycznych (kolumny) dla 9 odnalezionych niezależnych obiektów (wiersze).

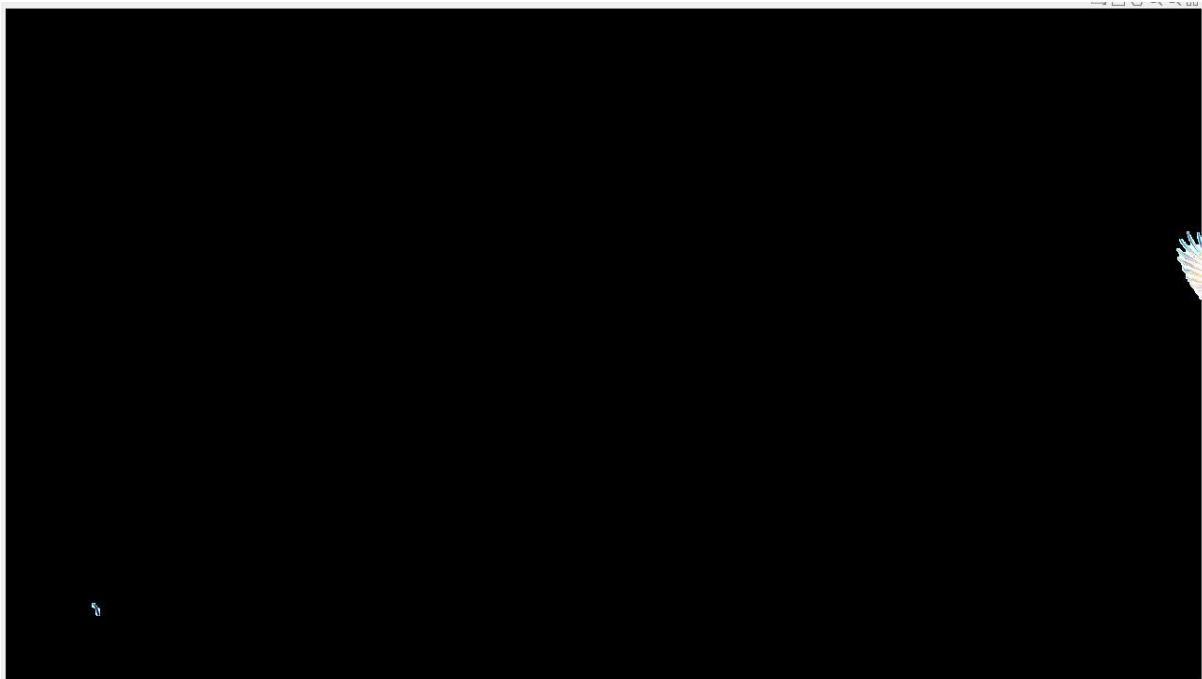
Wykorzystując obliczone współczynniki możemy odnaleźć obiekty o nietypowym kształcie, a dzięki temu wyselekcjonować z obrazu tylko te obiekty, które pod względem analizy są dla nas istotne. Wyliczamy więc kolejno średnią wartość kolejnych współczynników, odchylenie standardowe, a następnie błąd dla którego dany obiekt uznajemy za nietypowy.

```

m = mean(w);
s = std(w);
ws = (w-m)./s;

out = ~max(abs(ws) > 2, [], 2);

```



Rys. 25: Odnalezione nietypowe obiekty na oryginalnym obrazie.

1. Laboratorium 7

W trakcie trwania tych laboratoriów wykorzystaliśmy poniższy obraz do analizy:



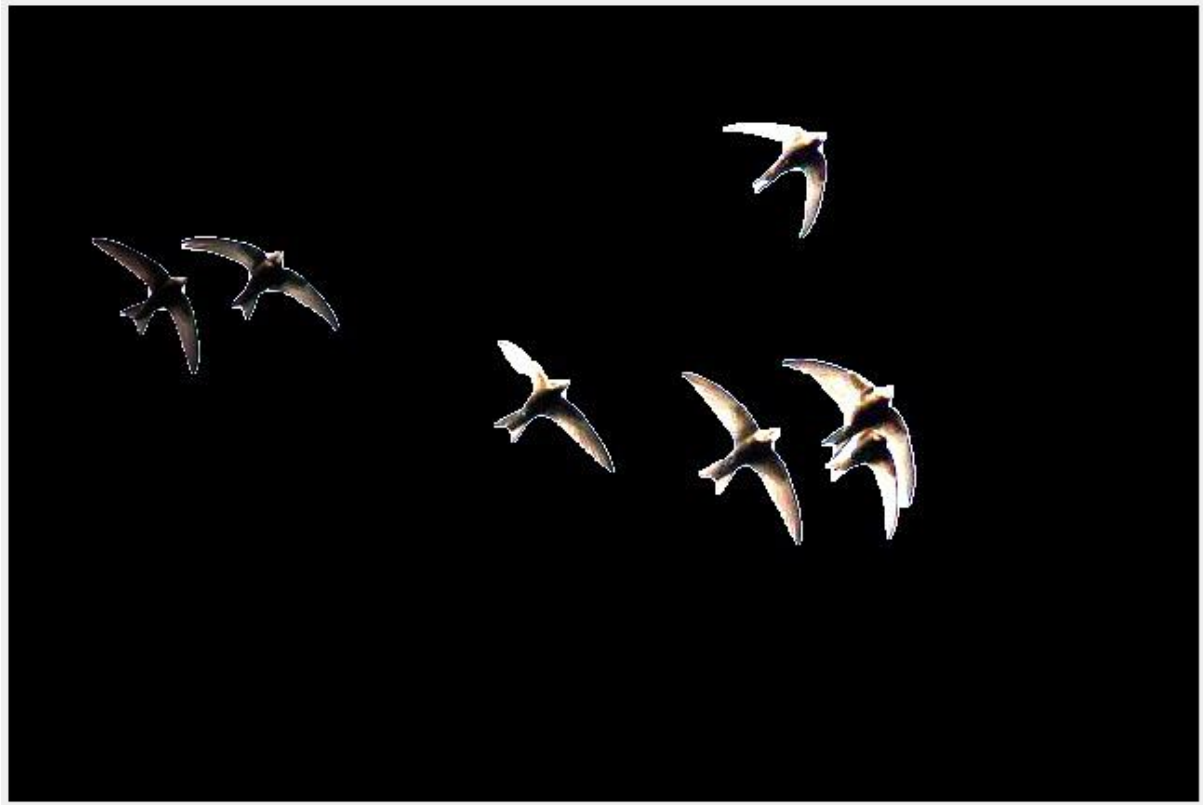
Rys. 26: Oryginał obrazu.

Dokonaaliśmy binaryzacji i odszumienia obrazu, a także nadaliśmy każdemu znalezionemu obiektowi etykiety:



Rys. 27: Oryginał obrazu po binaryzacji, odszumianiu i nadaniu obiektom etykiet.

Kolejnym etapem było wykonanie analogicznych kroków jak w trakcie ćwiczeń laboratoryjnych nr 6 w celu pozbycia się elementów na obrazie o nietypowych kształtach, a także obiektów znajdujących się na krawędziach obrazu (są to końcówki skrzydeł ptaków), czyli takich obiektów, które są nieistotne z punktu widzenia analizy naszego zdjęcia.

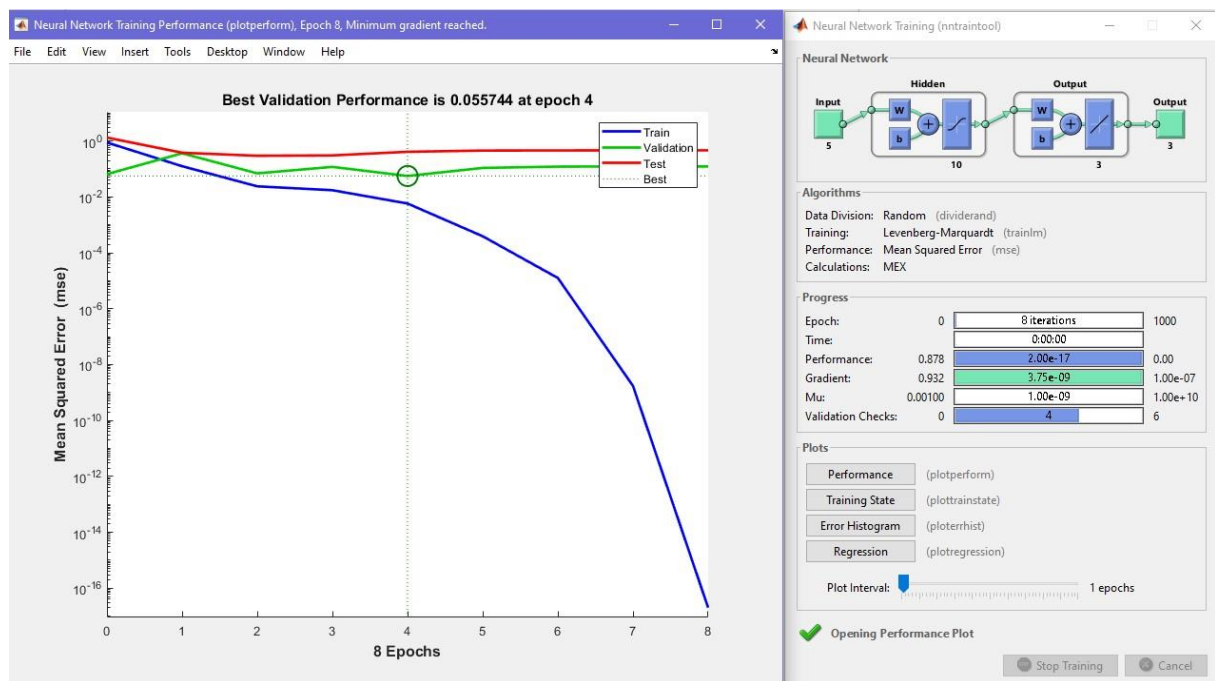


Rys. 28: Obiekty pozostałe na obrazie po dokonaniu analizy obrazu.

Na koniec poznaliśmy koncept działania sieci neuronowych. Wykorzystując zdobytą wiedzę przygotowaliśmy odpowiedni dataset, utworzony z wyników naszej własnej analizy obrazów ptaków, po czym użyliśmy go do wytrenowania naszej własnej sieci neuronowej. Ze względu na niewielką ilość danych sieć miała mocno ograniczone możliwości, przez co nie każda próba trenowania dawała satysfakcjonujące rezultaty. Jednakże nawet z tak ubogą liczbą informacji byliśmy w stanie nauczyć sieć rozpoznawać nietypowe elementy na wykorzystywanych przez nas obrazach ptaków (np. końcówki skrzydeł). Na Rys.29 widoczne są wektory trzelementowe (kolejne kolumny). Wartość bliska 1 oznacza, że udało nam się sklasyfikować testowany obiekt, lecz nie w każdym przypadku się to udaje o czym świadczy wektor z trzeciej kolumny.

ans =					
0.0435	-0.6063	0.2434	-0.3848	0.0626	-0.0054
0.0260	0.0400	0.4694	0.0408	-0.0324	0.0988
0.8828	1.2392	0.1842	1.2499	0.8645	0.9702

Rys. 29: Efekty trenowania sieci neuronowej.



Rys. 30: Efekty trenowania sieci neuronowej.