

Analiza obrazów

Sprawozdanie z ćwiczeń laboratoryjnych 5,6,7.

Patryk Śledź

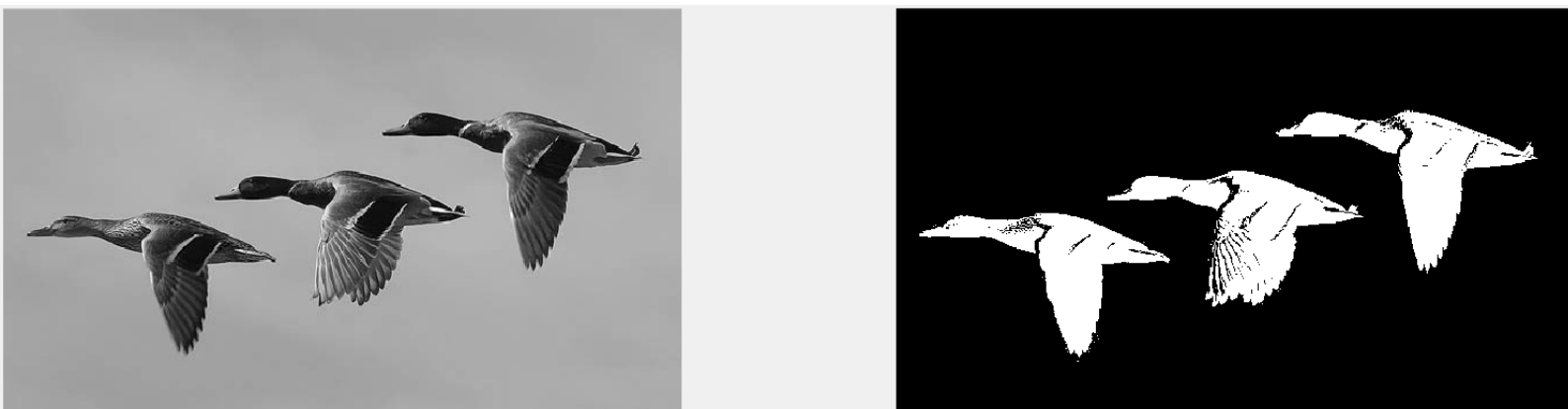
10 grudnia 2020

• Laboratorium nr 5

Na laboratorium 5 zajmowaliśmy się operacjami morfologicznymi. W naszych przykładach zajmowaliśmy się obrazami binarnymi, aby skupić się w szczególności na kształcie bez wpływu innych składowych m.in. jasności. Korzystaliśmy z funkcji **bwmorph (Image, operation, n)**, gdzie w naszym przypadku **Image** – obraz binarny, **operation** – przeprowadzane operacje oraz **n** – ilość powtórzeń operacji.

a) Binaryzacja i oczyszczanie obrazu.

Wszystkie operacje morfologiczne będziemy przeprowadzać na obrazie binarnym. Dlatego wczytany obraz kolorowy w pierwszej kolejności przekształciliśmy do skali szarości, a następnie dokonaliśmy binaryzacji z progiem 0.61 (rys.1).



Rys. 1. Po lewej obraz przekształcony do skali szarości, po prawej po operacji binaryzacji i inwersji kolorów.



Rys. 2. Po lewej obraz binarny, po prawej obraz po operacjach zamknięcia i otwarcia.

Na obrazie (rys. 1) widoczne są pewne niedoskonałości (szum na obiektach oraz przy krawędziach obiektów). Korzystając z operacji zamknięcia udało nam się uzyskać spójną binaryzację obiektów bez szumu. Dodatkowo skorzystaliśmy z operacji zamknięcia w celu poprawienia jakości krawędzi (rys.2).

b) Operacje *clean* oraz *fill*.



Rys. 3. Po lewej obraz binarny, po prawej obraz po wykorzystaniu operacji clean.

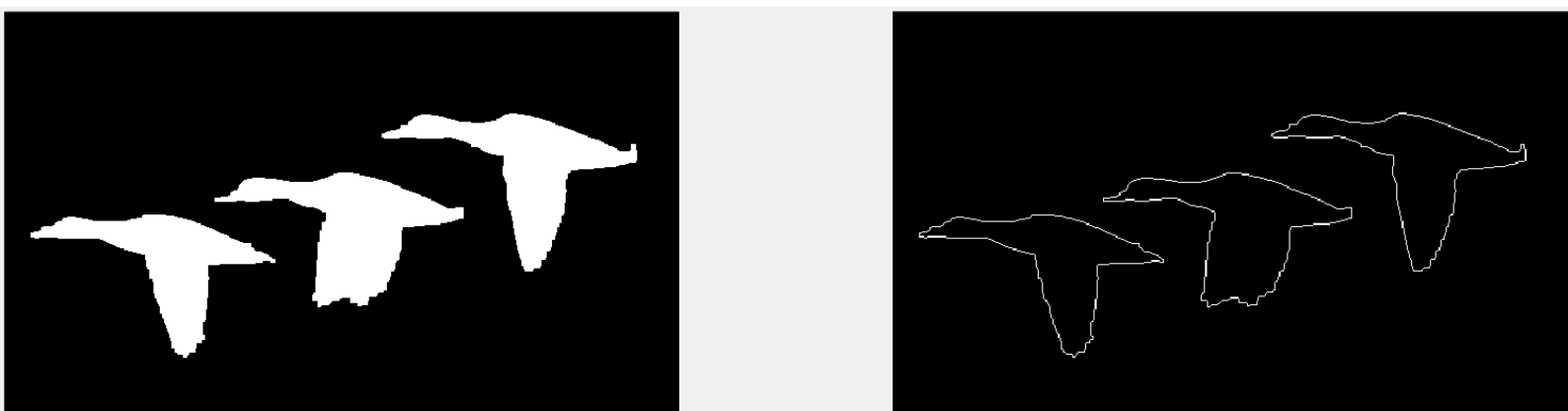


Rys. 4. Po lewej obraz binarny, po prawej obraz po wykorzystaniu operacji fill.

Operacja **clean** oczyszcza obraz z białych pikseli, które są odizolowane od pozostałych. Na rys. 3 pokazany został powiększony fragment obiektu, którego dolna krawędź jest wyraźnie zaszumiona. Operacja z powodzeniem usunęła pojedyncze piksele.

Operacja **fill** oczyszcza obraz z pojedynczych czarnych pikseli. Efekt działania jest widoczny na rys. 4.

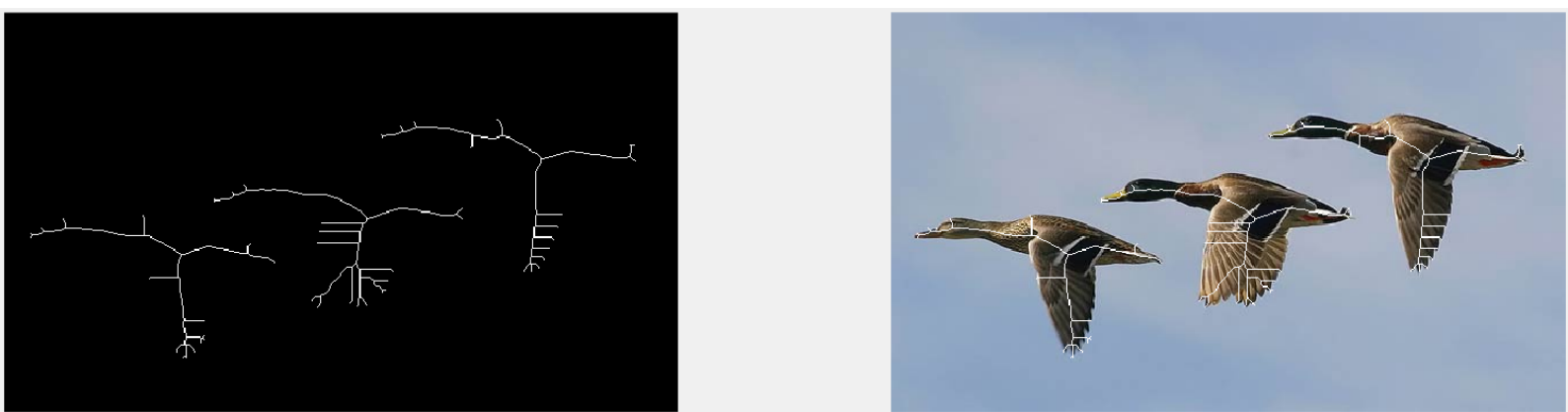
c) Operacja *remove*.



Rys. 5. Po lewej obraz binarny, po prawej obraz po wykorzystaniu operacji *remove*.

Operacja **remove** usuwa piksele białe, które nie sąsiadują z żadnym pikselem czarnym. W efekcie otrzymujemy krawędzie obiektów rys.5.

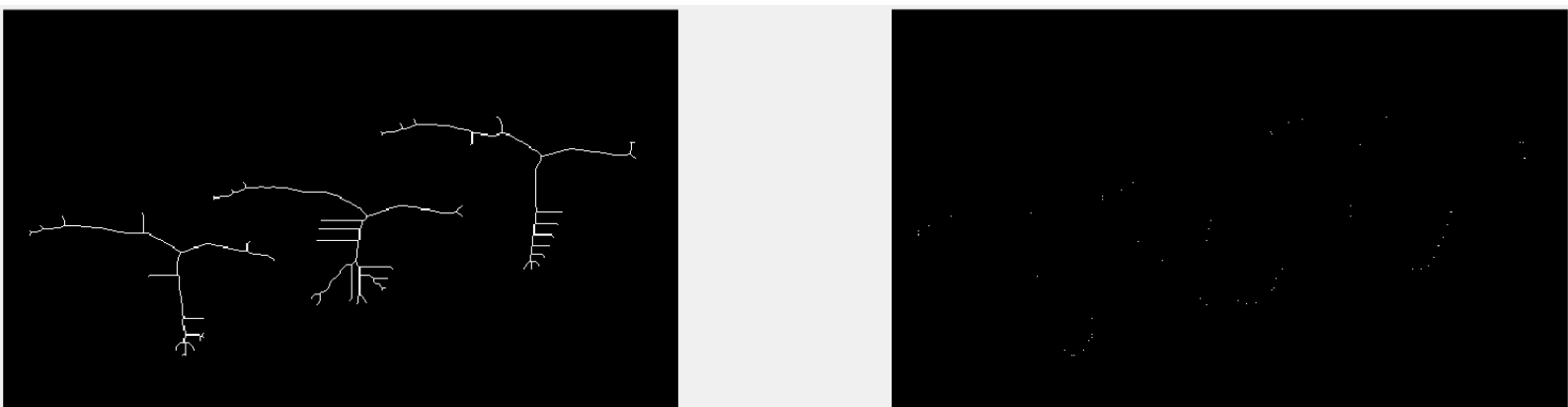
d) Szkieletyzacja obrazu - operacja *skel*.



Rys. 6. Po lewej obraz po operacji szkieletyzacji, po prawej oryginalny obraz z nałożonym szkieletem.

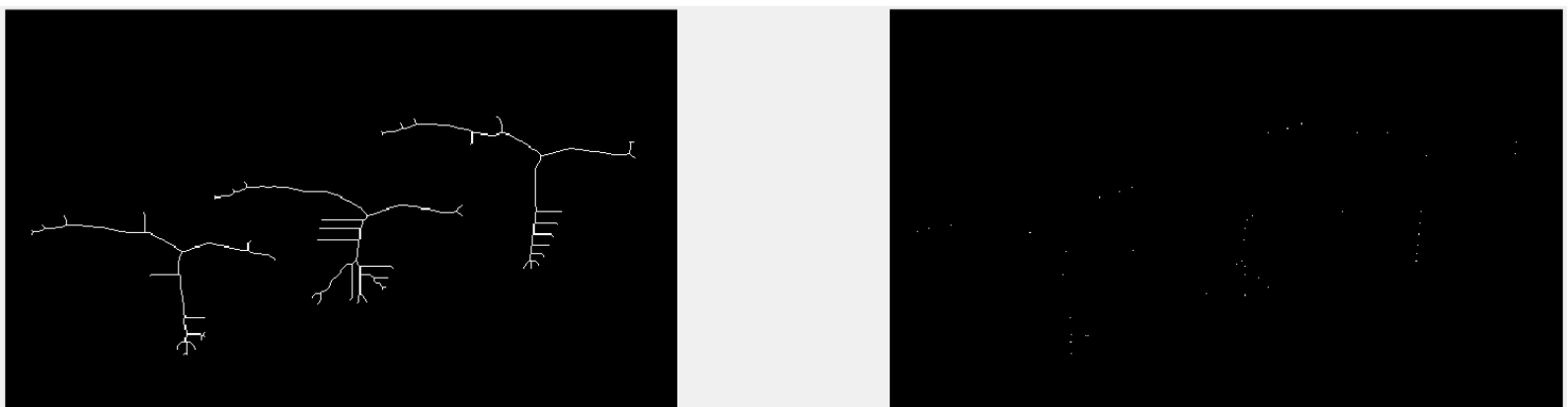
Efekt widoczny na rys. 6 otrzymaliśmy wykonując wielokrotnie funkcję **bwmorph** z parametrem **skel** dopóki występowały jakiegokolwiek zmiany na obrazie.

Szkielet obrazu jest to zbiór wszystkich punktów równoodległych od co najmniej dwóch brzegów. Szkielet odzwierciedla podstawowe właściwości topologiczne obiektu.



Rys. 7. Po lewej szkielet obrazu, po prawej punkty końcowe szkieletu.

Korzystając z operacji **endpoints** wykonanej na uzyskanym szkielecie uzyskaliśmy jego punkty końcowe. Przy pomocy funkcji **find** możemy odnaleźć współrzędne $[x, y]$ punktów końcowych.



Rys. 8. Po lewej szkielet obrazu, po prawej punkty przecięcia szkieletu.

Ponownie korzystając z funkcji **bwmorph** tym razem z parametrem **branchpoints** otrzymaliśmy punkty będące punktami przecięcia.

Podsumowując, przy pomocy punktów końcowych i przecięcia jesteśmy w stanie znacznie zredukować ilość danych potrzebnych do opisanie obiektu. Dodatkowo po odnalezieniu współrzędnych tych punktów (funkcja **find**) możemy w sposób analityczny przedstawiać i przekształcać nasz obiekt.

e) Operacja shrink.



Rys. 9. Po lewej obraz binarny, po prawej zaznaczono uzyskane punkty po operacji bwmorph.

Operacja ta redukuje obiekt, który nie ma posiada dziur do punktu. Dodatkowo operacja gwarantuje nam, że obiekt nie zostanie całkowicie zredukowany, a punkt będzie znajdował się wewnątrz wcześniejszego obiektu. **Shrink** może nam pomóc dowiedzieć się w którym miejscu na obrazie znajduje się obiekt.

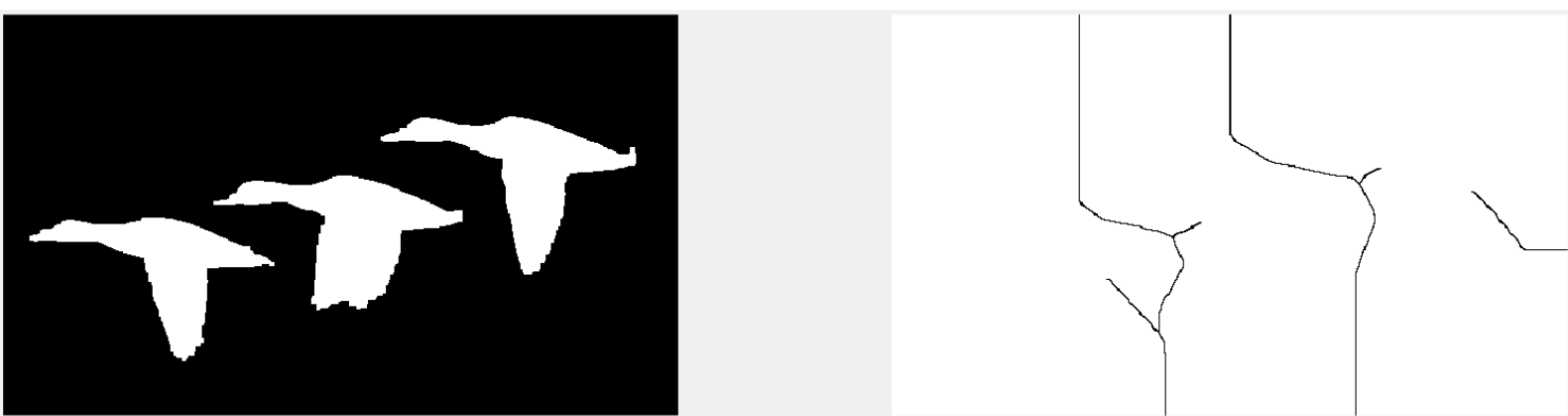
f) Operacja thin.



Rys. 10. Po lewej obraz binarny, po prawej obraz po operacji thin.

Operacja **thin** przeprowadzając erozję redukuje obiekt, który nie posiada dziur do linii. Dodatkowo gwarantując, że linie na obiektach zostaną zachowane.

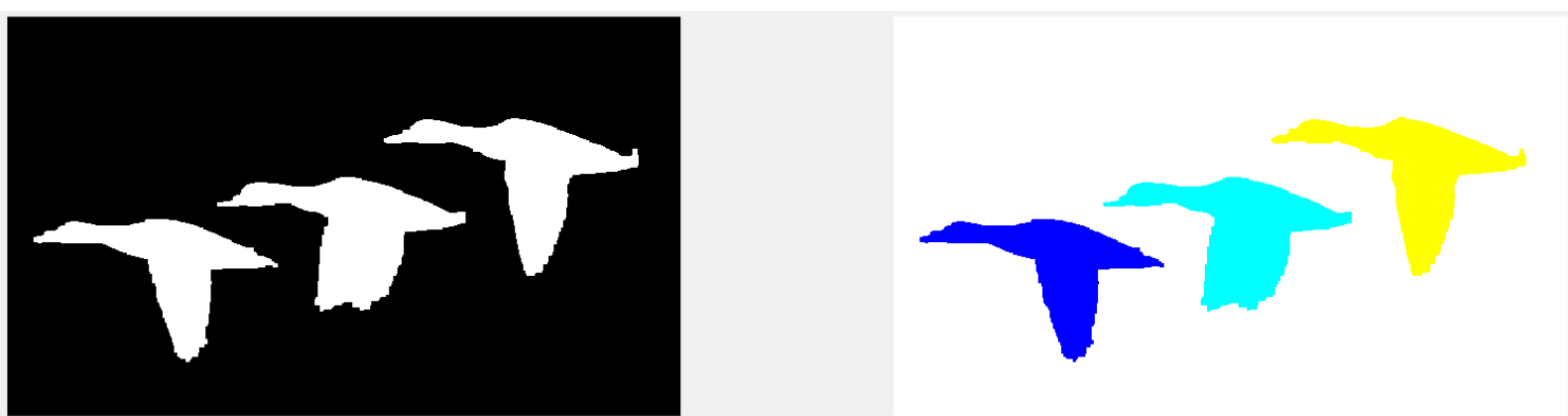
g) Operacja **thicken**.



Rys. 11. Po lewej obraz binarny, po prawej obraz po operacji **thicken**.

Operacja **thicken** przeprowadzając dylatację zwróciła nam 3 oddzielone od siebie segmenty. W każdym z poszczególnych segmentów znajduje się cały uprzedni obiekt oraz mamy pewność że żaden z obiektów nie połączył się z innym.

h) Segmentacja obrazu.



Rys. 12. Po lewej obraz binarny, po prawej wydzielone segmenty obrazu – każdy kolor to inna etykieta obiektu.

Przy pomocy funkcji **bwlabel** otrzymujemy macierz, której wartości zerowe wskazują nam tło, a każdy obiekt na obrazie posiada etykietę – wartość liczbową.

Rezultat widoczny na rys. 12 otrzymano przy użyciu funkcji **label2rgb**, która przypisuje każdej etykietce pewien kolor.



Rys. 13. Po lewej obraz binarny, po prawej wydzielony obiekt o etykiecie równej 2.

Mnożąc wartości macierzy oryginalnego obrazu przez wartości o etykiecie numer 2, otrzymaliśmy wyodrębniony obiekt bez tła (rys. 13).



Rys. 14. Po lewej obraz binarny, po prawej wydzielone segmenty obrazu przy użyciu operacji thicken.



Rys. 15. Po lewej obraz binarny, po prawej wydzielony segment o etykiecie równej 2.

Przy użyciu operacji **thicken**, która została przedstawiona w podpunkcie g) przeprowadziliśmy operacje segmentacji przez pogrubianie. W wyniku operacji thicken otrzymaliśmy segmenty, które zawierają obiekty. Następnie korzystając z funkcji bwlabel przypisaliśmy etykietę dla każdego segmentu.

Efekty segmentacji widoczne są na rys. 14 i 15.

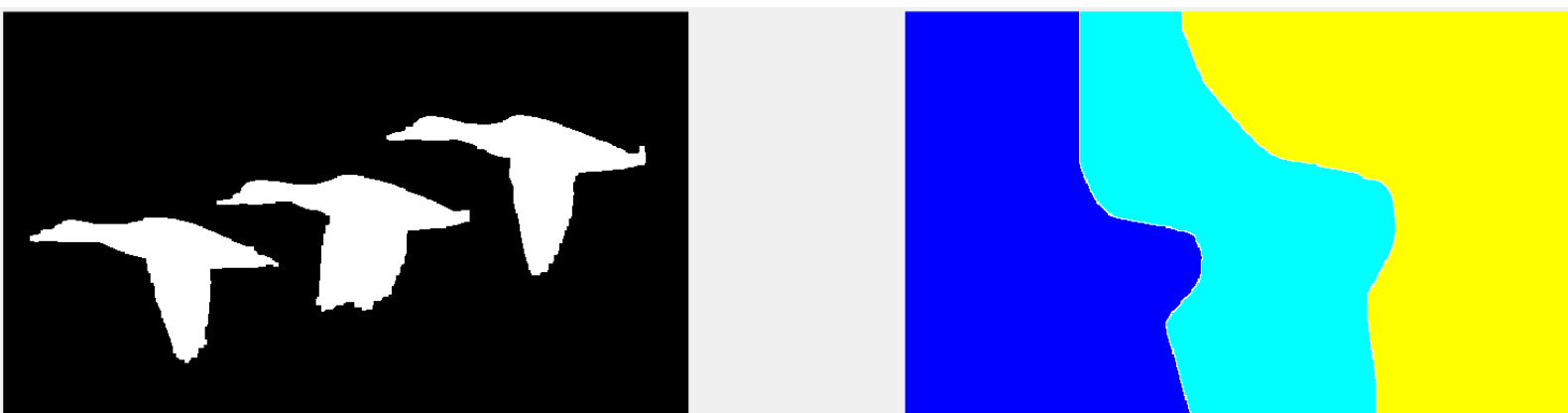
Segmentacja przez pogrubianie w porównaniu do segmentacji przeprowadzanej bez operacji thicken umożliwia nam dokładniejsze wydzielenie obiektu z obrazu. Obiekty na obrazie nie zawsze są do siebie zbliżone pod względem m.in. kolorów. Z tego powodu binaryzacja może być utrudniona i jej wynik może być niezadawalający, a wydzielając w pierwszej kolejności obszar z obiektem możemy dokładniej zbinaryzować obraz.

i) Segmentacja wododziałowa przy użyciu transformaty odległościowej.



Rys. 16. Po lewej obraz binarny, po prawej obraz przekształcony przy pomocy transformaty odległościowej.

Pracując na obrazie binarnym, każdemu pikselowi obrazu przypisujemy odległość od najbliższego obiektu. Dlatego na rys.16 obiekty są reprezentowane przez czarne piksele ponieważ odległość białego piksela od obiektu jest równa 0. Obraz na rys. 16 można interpretować jako zlewiska wody, a krawędź oddzielająca dwa zlewiska nazywamy wododziałem.



Rys. 17. Po lewej obraz binarny, po prawej wydzielone segmenty.



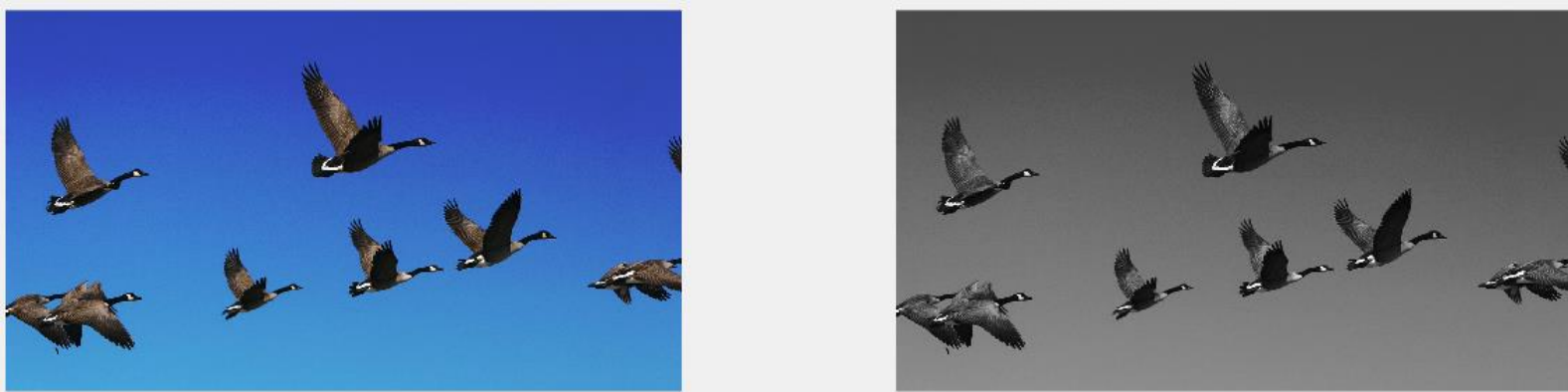
Rys. 18. Po lewej obraz binarny, po prawej wydzielony segment o etykiecie równej 2.

Przy pomocy funkcji **watershed** otrzymujemy etykiety segmentów obrazu, które widoczne są na rys. 17. W porównaniu do segmentacji uzyskanej na rys. 14, wododziałowa jest dokładniejsza, a krawędzie są mniej poszarpane.

• Laboratorium nr 6

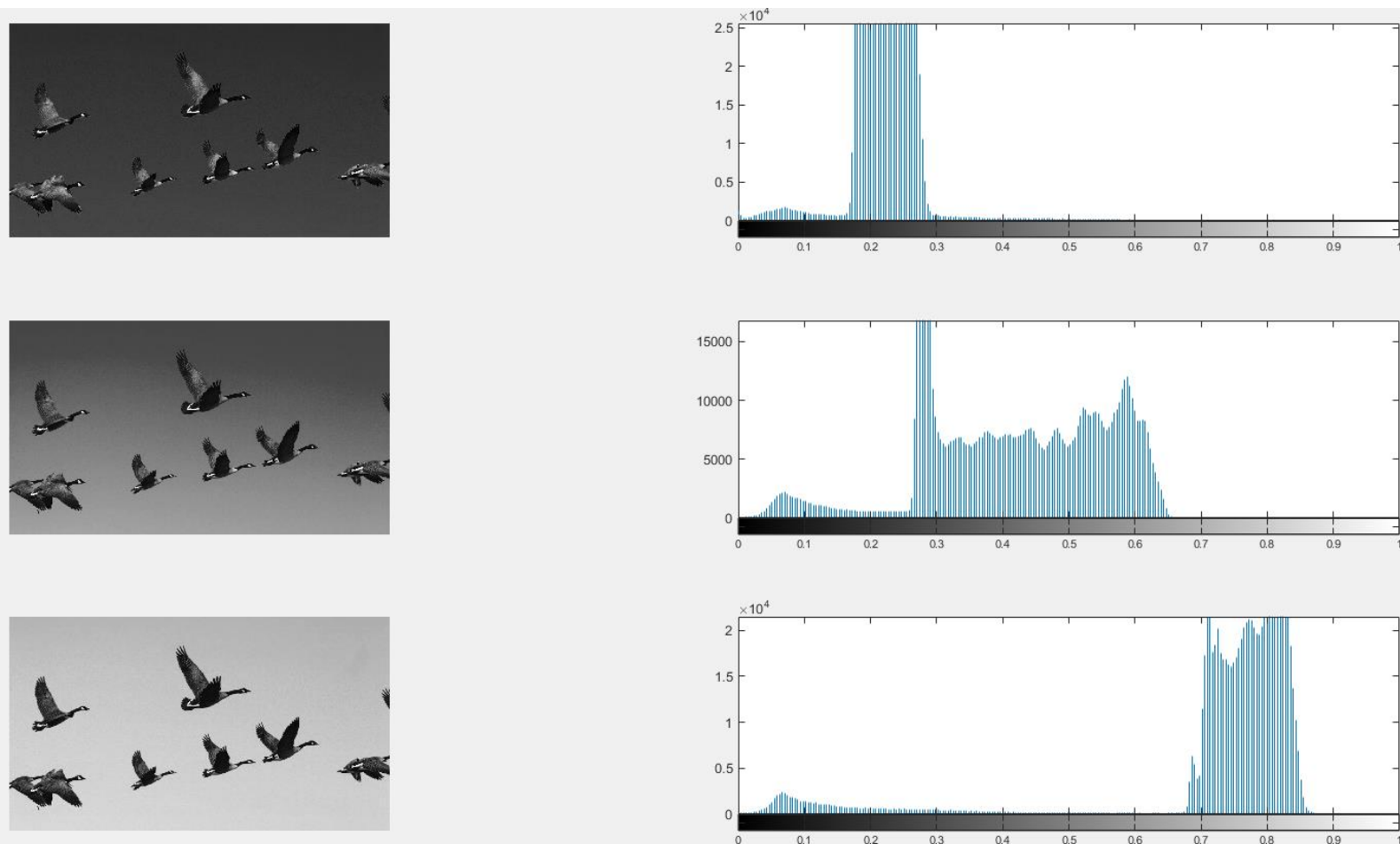
Na laboratorium 6 zajmowaliśmy się współczynnikami geometrycznymi obrazu.

Przypisywaliśmy obiektom wartości liczbowe, które mają go opisać, aby w dalszych etapach móc m.in. porównywać obiekty.



Rys. 19. Obraz na którym pracowaliśmy. Po lewej oryginalny, po prawej w odcieniach szarości.

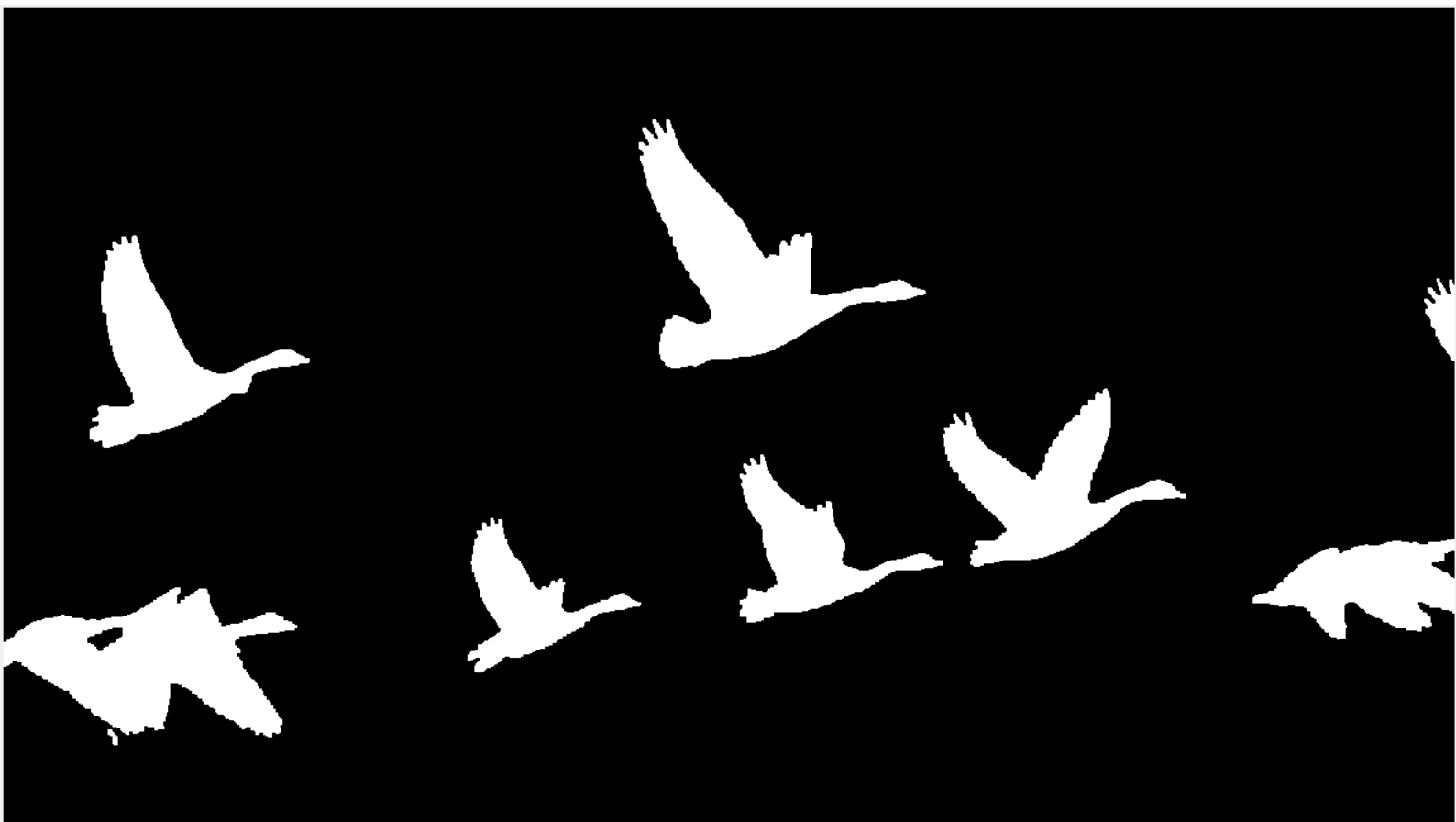
a) Binarizacja obrazu wykorzystując kanały RGB.



Rys. 20. Każdy wiersz zawiera obraz w odcieniach szarości danego kanału RGB wraz z histogramem.

Obserwując obraz na rys. 19 można zauważyć, że tło nie jest jednolite, a gradientowe. Przekształcając obraz do odcieni szarości wiele wartości pikseli tła jest podobna do wartości obiektu co powoduje trudność binaryzacji. Warto zauważyć, że na obrazie kolorowym bardzo dobrze są wydzielone obiekty od tła, dlatego więc skorzystaliśmy z binaryzacji wydzielając odpowiednie kanały RGB.

Skorzystaliśmy z kanałów R oraz B. Osobno przeprowadzając binaryzacje w każdym z nich, a wyniki połączyliśmy. Dodatkowo przeprowadziliśmy operacje otwarcie i zamknięcia, aby usunąć szumy i niedoskonałości. Efekty widoczne są na rys. 21.



Rys. 21. Wynik połączenia binaryzacji na kanale R i B.

b) Cechy obiektów – funkcja regionprops.

Funkcja **regionprops** do której przekazujemy macierz obrazu binarnego jest w stanie zwrócić nam wiele cech obrazu.

Field ▲	Value		
Area	11955	Extrema	8x2 double
Centroid	[137.4514,307.9989]	EquivDiameter	123.3758
BoundingBox	[76.5000,200.5000,194,187]	Solidity	0.5767
SubarrayIdx	1x2 cell	Extent	0.3295
MajorAxisLength	187.7972	PixelIdxList	11955x1 double
MinorAxisLength	142.8075	PixelList	11955x2 double
Eccentricity	0.6494	Perimeter	732.1730
Orientation	-59.9836	PerimeterOld	775.4701
ConvexHull	39x2 double	MaxFeretDiameter	206.9299
ConvexImage	187x194 logical	MaxFeretAngle	159.6384
ConvexArea	20729	MaxFeretCoordin...	[270.5000,308.5000;76.5000,380.5000]
Circularity	0.2802	MinFeretDiameter	170.6982
Image	187x194 logical	MinFeretAngle	125.2176
FilledImage	187x194 logical	MinFeretCoordina...	[76.5000,380.5000;174.9388,241.0450]
FilledArea	11955		
EulerNumber	1		

Rys. 22. Wartości struktury zwróconej przez funkcję regionprops dla obiektu o etykiecie nr 2.

Wybrane właściwości:

- **Area** – jest to rzeczywista liczba pikseli w danym regionie,
- **Centroid** – środek masy zwrócony jako wektor, gdzie pierwsza wartość współrzędna x (lub y), a druga wartość to współrzędna y (lub x),
- **BoundingBox** – położenie i rozmiar najmniejszego prostokąta w którym można umieścić obiekt. Prostokąt ten jest zgodny z układem odniesienia,
- **MajorAxisLength** i **MinorAxisLength** – długości głównej i mniejszej osi elipsy, która przybliża krawędzie obiektu,
- **Orientation** – kąt pomiędzy osią X układu odniesienia a główną osią (MajorAxis),
- **Circularity** – miara podobieństwa obiektu do koła. Dla idealnego koła wartość ta wynosi 1,
- **Image** – zawiera macierz obrazu przyciętego do BoundingBox,
- **EulerNumber** – wartość liczby obiektów pomniejszona o liczbę otworów w tych obiektach,
- **Perimeter** – obwód, długość krawędzi obiektu.

c) Miara podobieństwa obiektu do koła – obliczanie współczynników.

6.8833	151.8289	285.5240	3.5365	107.0568	0.5078	80.3237	0.8806
6.0071	123.3758	218.3606	3.1325	93.1480	0.9637	73.8036	0.7699
1.6442	9.4407	11.1408	1.3926	14.9911	1.6250	4.6556	0.1801
5.3945	95.3864	173.4789	3.3077	89.2812	0.8882	55.3935	0.8187
6.6645	151.3837	290.2986	3.6773	106.3109	0.8651	91.3402	0.9176
5.7364	107.7469	200.5352	3.4640	85.6468	0.8258	64.5639	0.8612
6.2503	127.4269	252.7380	3.9339	116.7577	0.7324	71.1582	0.9834
5.8982	108.8754	174.7521	2.5762	77.5783	0.4972	58.7440	0.6051
3.3761	37.9142	67.8000	3.1978	27.7870	2.7692	22.1400	0.7882

Rys. 23. Współczynniki obliczone dla każdego obiektu na obrazie.

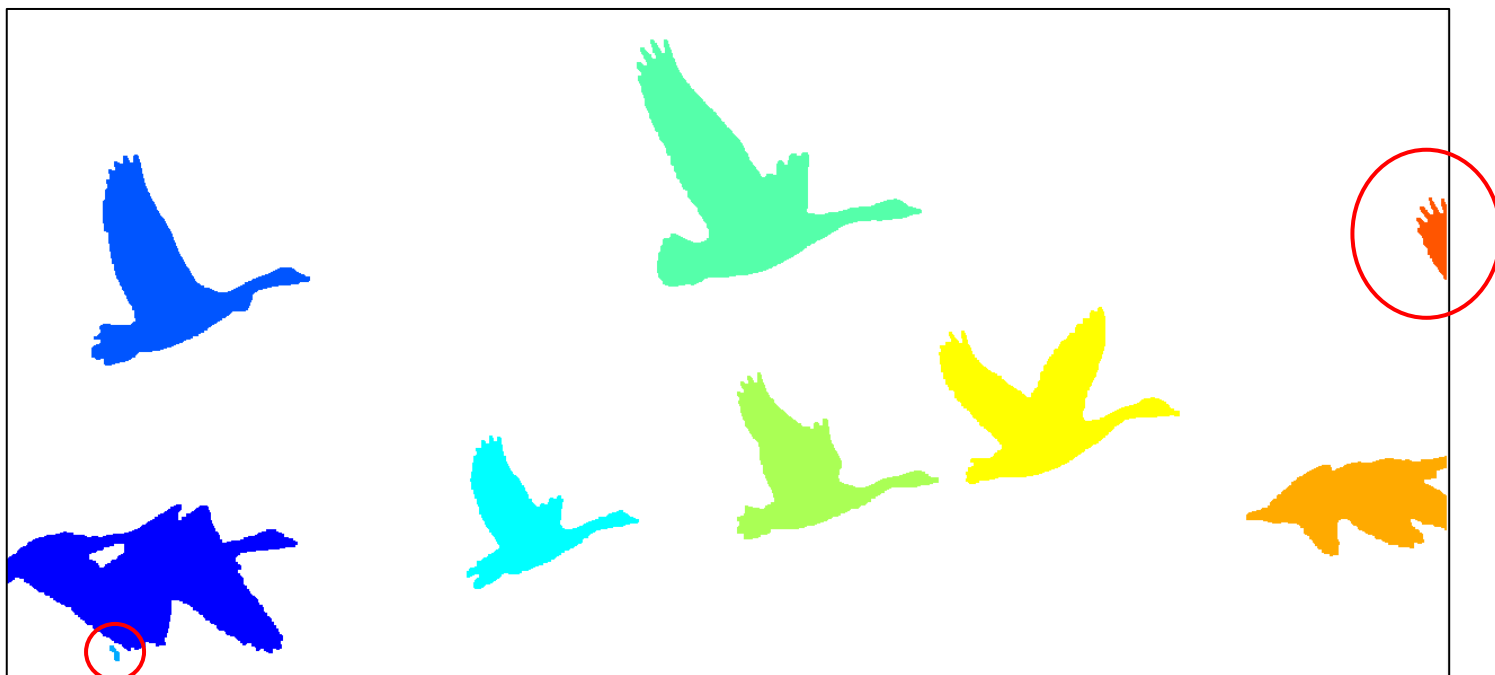
0.9159	1.0379	1.0533	0.5299	0.7689	-0.7923	0.8004	0.5169
0.4035	0.4513	0.3420	-0.0038	0.3760	-0.1553	0.5665	0.0573
-2.1480	-1.8977	-1.8526	-2.3024	-1.8320	0.7684	-1.9144	-2.3916
0.0452	-0.1258	-0.1333	0.2276	0.2667	-0.2609	-0.0940	0.2600
0.7879	1.0287	1.1038	0.7160	0.7478	-0.2932	1.1957	0.6708
0.2451	0.1291	0.1532	0.4341	0.1640	-0.3480	0.2350	0.4363
0.5457	0.5348	0.7061	1.0549	1.0429	-0.4785	0.4716	0.9438
0.3398	0.1523	-0.1199	-0.7387	-0.0639	-0.8071	0.0262	-0.6270
-1.1351	-1.3107	-1.2526	0.0825	-1.4705	2.3669	-1.2871	0.1336

Rys. 24. Macierz współczynników po operacji standaryzacji.

Dla każdego obiektu o danej etykiecie obliczyliśmy współczynniki. Każdy wiersz na rys. 23 oraz 24 symbolizuje obiekt, natomiast każda kolumna to inny współczynnik. Patrząc kolejno od prawej do lewej są to:

- **współczynnik Blaira – Blissa** – jest stosunkiem średniej odległości piksela od środka masy obiektu do średniej odległości piksela od środka masy dla koła,
- **CircularityS** – odpowiednik EquivDiameter czyli promień koła o tym samym polu,
- **CircularityL** – promień koła o tym samym obwodzie,
- **Shape** – będący odwrotnością współczynnika Circularity,
- **współczynnik Danielssona** – jest stosunkiem średniej odległości piksela od najbliższej krawędzi obiektu do średniej odległości piksela od najbliższej krawędzi koła,
- **współczynnik Fereta** – stosunek długości osi Fereta,
- **współczynnik Haralicka** - jest stosunkiem średniej odległości piksela krawędzi od środka masy obiektu do średniej odległości piksela krawędzi od środka masy koła,
- **współczynnik Malinowskiej** – przekształcenie współczynnika Circularity, przyjmuje wartości od 0 do +nieskończoności- w zerze obiekt jest kołem.

Analizując wartości otrzymane na rys. 23. Możemy zauważyć, współczynniki otrzymane dla obiektów w wierszu 3 oraz 9 znacznie odbiegają od pozostałych wartości. Jeszcze lepiej jest to widoczne na rys. 24 gdzie przedstawione są ustandaryzowane wyniki. Na rys. 25 zaznaczono obiekty, które są opisywane przez te wiersze. Bez problemów można zauważyć dlaczego wartości tak bardzo się różniły.



Rys. 25. Obraz przedstawiający obiekty z przypisanymi etykietami.

d) Wyznaczanie nietypowych obiektów poprzez standaryzację.

Aby bardziej zautomatyzować sposób sprawdzania rozbieżności między obiektami możemy posłużyć się rozkładem normalnym. Po standaryzacji wyników otrzymaliśmy dane na temat tego jak bardzo dany wynik jest różny od wartości oczekiwanej rys. 24.

Z powodu zbyt małej liczby danych przyjęliśmy wartość 2 sigm jako zakres odchylenia dla którego dany wynik jest prawidłowy.

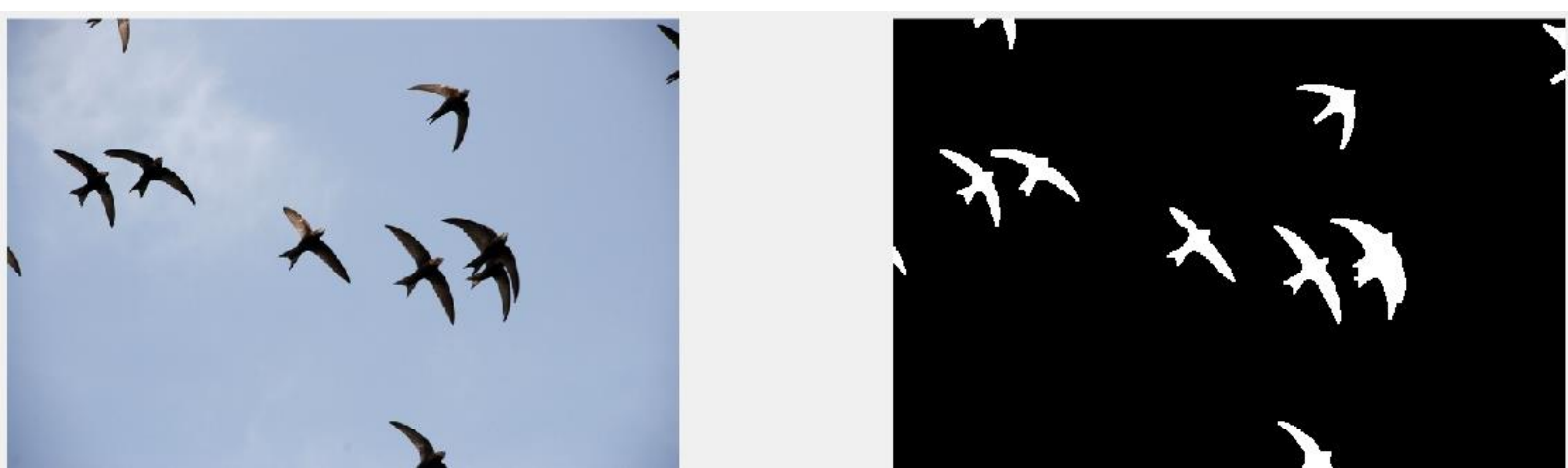
Otrzymaliśmy następujący wektor wartości logicznych rys. 26, nietypowe elementy są symbolizowane przez wartość 1.

0
0
1
0
0
0
0
0
1

Rys. 26. Wektor wartości logicznych.

- **Laboratorium nr 7**

Na laboratorium 7 zajmowaliśmy się klasyfikacją obiektów przy użyciu sieci neuronowych.



Rys. 27. Po lewej obraz na którym pracowaliśmy, po prawej obraz po binaryzacji.



Rys. 28. Obraz przedstawiający obiekty wraz z etykietami.

Po wczytaniu obrazu dokonaliśmy binaryzacji, a następnie przypisaliśmy etykiety do obiektów. Następnie podobnie jak podczas laboratorium 6 obliczyliśmy następujące współczynniki: **Blaira – Blissa**, **CircularityS**, **CircularityL**, **Shape**, **Danielssona**, **Fereta**, **Haralicka**, **Malinowskiej** (rys. 29).

2.0889	14.9696	17.8254	1.4179	26.1429	2.1667	8.3792	0.1908
3.2012	39.8463	73.8479	3.4348	117.3958	1.2632	23.8843	0.8533
1.7692	9.3049	9.5493	1.0532	13.2364	0.5833	4.5113	0.0263
3.1999	40.4804	73.2113	3.2709	114.4489	0.5952	25.4188	0.8086
2.2761	17.8055	22.6000	1.6110	24.7535	2.4167	9.6789	0.2693
3.2085	40.5432	71.6197	3.1205	123.6877	1.0938	24.7295	0.7665
3.4586	47.0549	83.7155	3.1652	127.0626	1.4375	28.9377	0.7791
2.9325	32.5474	45.5183	1.9559	48.6244	0.8462	19.6714	0.3985
3.2505	38.5635	75.1211	3.7946	95.2744	1.1091	22.7879	0.9480
4.2063	58.1745	92.9465	2.5527	77.0866	1.3521	33.7149	0.5977
2.3692	17.9834	21.0085	1.3647	23.9165	1.2000	9.7947	0.1682
1.9175	11.3401	11.4592	1.0211	23.9319	0.9231	5.6697	0.0105

Rys. 29. Współczynniki obliczone dla każdego obiektu na obrazie.

Jak można zauważyć na rys. 28 mamy wiele obiektów które są przy krawędziach, dlatego przy użyciu standaryzacji i rozkładu normalnego możemy nie dostać zadawalającego efektu wyznaczenia elementów nietypowych.

Chcemy więc podzielić obiekty na te które są położone przy krawędzi oraz te, które nie przylegają do niej. Do tego zadania użyliśmy funkcji **unique** oraz **setdiff** operując na etykietach.

a) Uczenie sieci neuronowej.

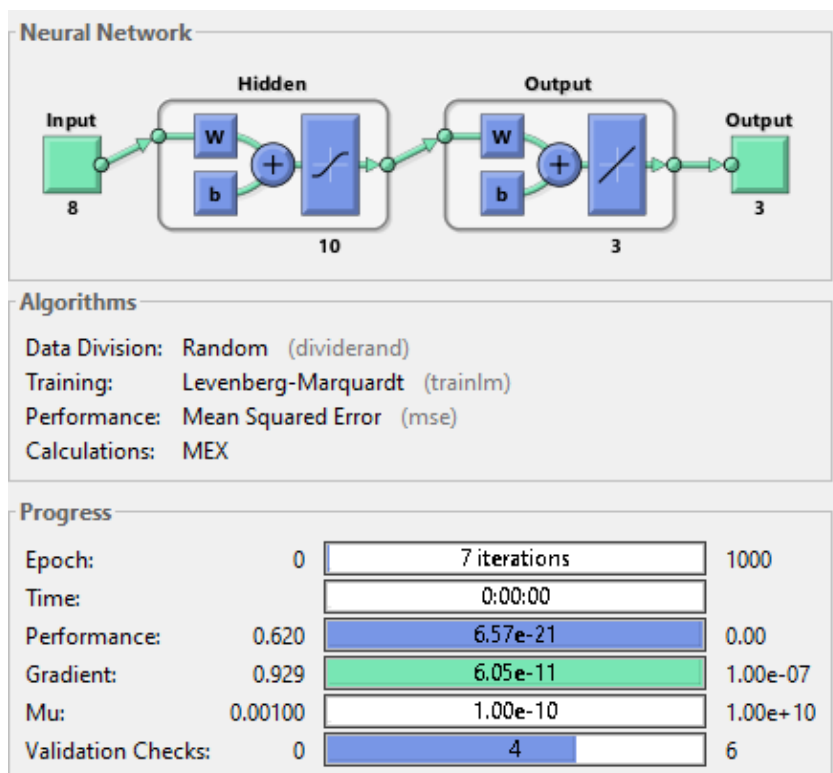
Danymi wejściowymi będą trzy grupy obiektów, dane te uzyskamy ze współczynników uzyskanych podczas laboratorium 6 (rys. 23) oraz współczynników z rys. 29.

Grupa pierwsza będzie składała się z obiektów (gęsi) z laboratorium 6.

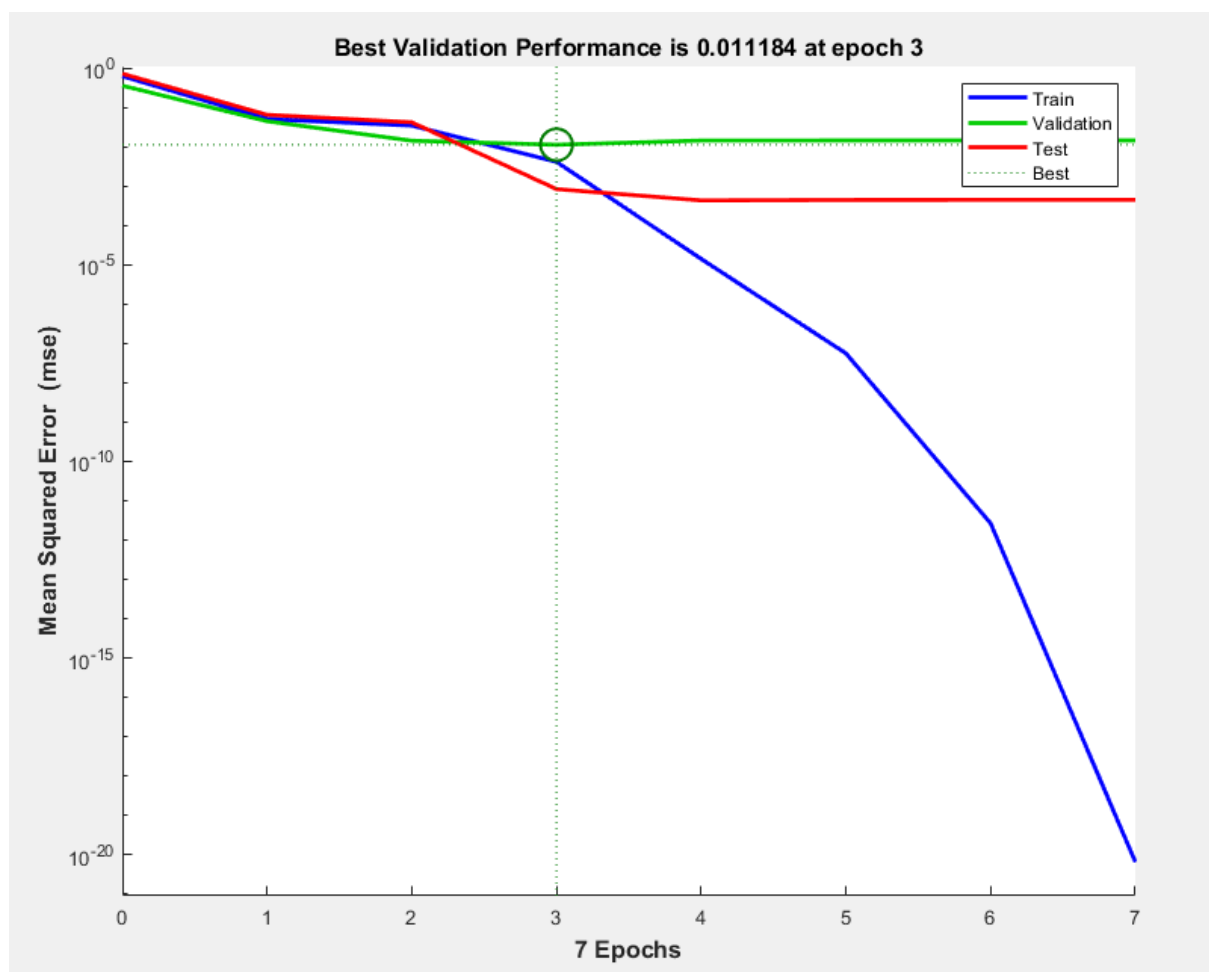
Grupa druga zawiera ptaki z laboratorium 7, które uznajemy za typowe czyli zakładamy, że to te w środkowej części obrazu.

Grupa trzecia będzie składać się z gęsi, które są nietypowe z laboratorium 6 oraz z ptaków, które uznajemy za nietypowe czyli te, które leżą przy krawędzi obrazu.

Na rys. 30 otrzymaliśmy graficzną reprezentację sieci neuronowej. Wartościami Input jest osiem współczynników, które obliczyliśmy. Sieć przekształcała dane wejściowe sumując je z odpowiednimi wagami i wyrazami wolnymi.



Rys. 30. Okno sieci neuronowej feedforwardnet.



Rys. 31. Wykres kolejnych iteracji uczenia.

Proces uczenia się na naszym zbiorze danych zakończył się po 7 iteracjach. Po 3 iteracji wynik zaczął się pogarszać więc przyjęto rezultat otrzymany w 3 iteracji. W przedstawionym przykładzie uczenie zostało zatrzymane po osiągnięciu wymaganego gradientu.

W celu sprawdzenia jak dobrze nasza sieć rozpoznaje obiekty, przekazaliśmy do niej te elementy, których nie użyliśmy do procesu uczenia (mamy w sumie 6 obiektów, których nie użyliśmy).

0.9313	1.1346	-0.0069	0.1814	0.0578	0.0305
0.1270	-0.3005	0.9076	0.3485	-0.1371	-0.0726
-0.0070	0.2892	0.0808	0.4662	1.0660	1.0495

Rys. 32. Wartości zwrócone przez sieć neuronową.

1	1	0	0	0	0
0	0	1	0	0	0
0	0	0	0	1	1

Rys. 33. Zaokrąglone wartości z rys. 32.

Za przewidywane wartości obraliśmy wektor [1,0,0] dla grupy 1, wektor [0, 1, 0] dla grupy 2 oraz [0, 0, 1] dla grupy 3.

Analizując otrzymane zaokrąglenia wyników (rys. 32), biorąc pod uwagę ograniczoną ilość danych uczących, nasza sieć w dość zadowalającym stopniu rozpoznała przynależność elementów. Jedynie element z kolumny 4 nie został odpowiednio rozpoznany.