



AKADEMIA GÓRNICZO-HUTNICZA W KRAKOWIE

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ

Analiza obrazów

Laboratorium 5-7: Operacje morfologiczne

Andrzej Świętek

02.12.2024

Contents

1	Operacje morfologiczne	2
1.1	Znajdowanie samych krawędzi = BIM - Eroza(BIM)	2
1.2	Szkieletyzacja - znajdowanie szkieletu obiektu	3
1.3	Znajdowanie punktów końcowych szkieletu	3
1.4	Znajdowanie punktów gdzie linie szkieletu się rozgałęziają	4
1.5	Funckja SHRINK	5
1.6	Funckja THIN	6
1.7	Funckja THICKEN	8
1.8	Segmentacja	10
1.8.1	Segmentacja przez pogrubianie - bwlabel	10
1.8.2	Segmentacja przez transformatę odległościową - bwdist	12
1.9	Wnioski	16
2	Właściwości obiektu	17
2.1	Przygotowanie do badania właściwości obiektu	17
2.2	Wyodrębnienie obiektu	20
2.3	Właściwości obiektu na obrazie	21
2.4	Współczynniki	22
2.5	Wnioski	25
3	Sieć neuronowa	26
3.1	Perceptron	27
3.2	Wstępne przetworzenie obrazów	27
3.3	Ekstrakcja cech	28
3.4	Zapis wektorowy danych wejściowych i wyjściowych	29
3.5	Trenowanie sieci neuronowej	30
3.6	Testowanie sieci neuronowej	33
3.7	Wnioski	33

1. Operacje morfologiczne

W ramach zajęć wszystkie operacje morfologiczne wykonywaliśmy na zbinaryzowanej wersji obrazu kaczki.jpg za pomocą funkcji `imbinarize` dla wartości progu 0.6 (wyznaczony eksperymentalnie) po uprzedniej odbrazu do skali szarości.



Figure 1: Obraz oryginalny



Figure 2: Efekt binaryzacji

1.1. Znajdowanie samych krawędzi = BIM - Eroza(BIM)

Operacja ta polega na znajdowaniu krawędzi obiektów na obrazie binarnym. Wykorzystuje różnicę pomiędzy obrazem binarnym (BIM) a jego erozją. Erozja usuwa piksele na brzegach obiektu, a różnica tych dwóch obrazów uwidacznia tylko krawędzie. Efekt tej operacji to zbiór pikseli reprezentujących granice obiektów w obrazie.

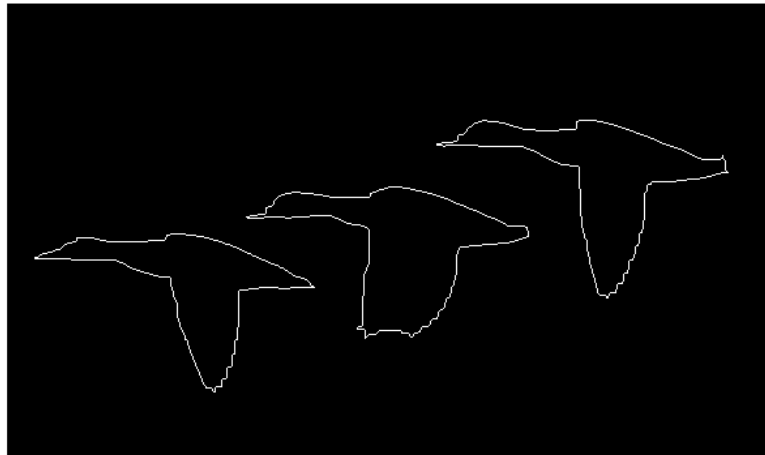


Figure 3: Efekt operacji znajdowania krawędzi

1.2. Szkieletyzacja - znajdowanie szkieletu obiektu

Szkieletyzacja jest procesem przekształcenia obiektów binarnych na obrazie w ich struktury szkieletowe. Szkielet obiektu reprezentuje najcieńszą możliwą wersję obiektu, zachowującą jego topologię. Proces ten jest używany do analizy kształtów, upraszczania reprezentacji obiektów oraz znajdowania ich podstawowej struktury geometrycznej.

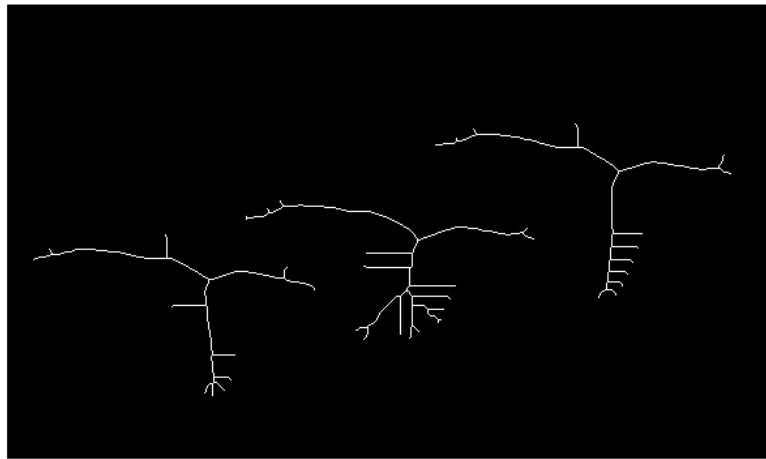


Figure 4: Efekt operacji szkieletyzacji

1.3. Znajdowanie punktów końcowych szkieletu

Operacja ta polega na identyfikacji pikseli końcowych w szkielecie obiektu. Punkty końcowe to takie piksele, które mają tylko jednego sąsiada należącego do szkieletu. Znajdowanie punktów końcowych jest kluczowe w analizie i przetwarzaniu kształtów, np. w celu detekcji końców linii w obrazach

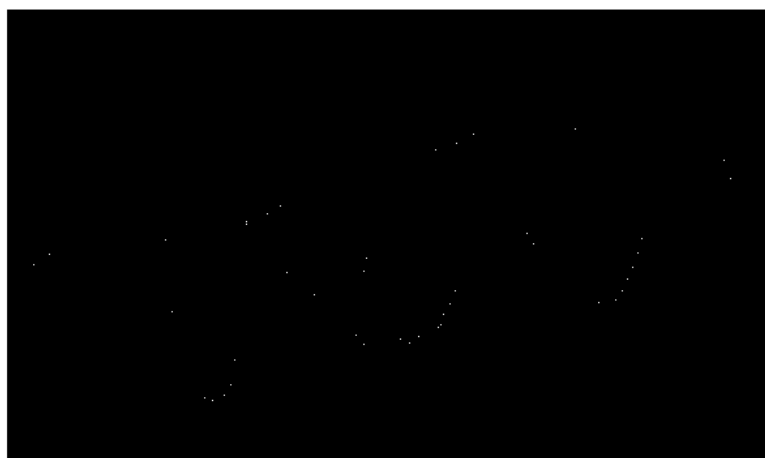


Figure 5: Efekt operacji znajdowania punktów końcowych

1.4. Znajdowanie punktów gdzie linie szkieletu się rozgałęziają

Znajdowanie punktów rozgałęzienia pozwala zidentyfikować miejsca, gdzie linie szkieletu rozchodzą się w różnych kierunkach. Są to piksele mające więcej niż dwóch sąsiadów należących do szkieletu. Taka analiza jest używana np. w analizie sieci dróg, gałęzi drzew lub struktur biologicznych.

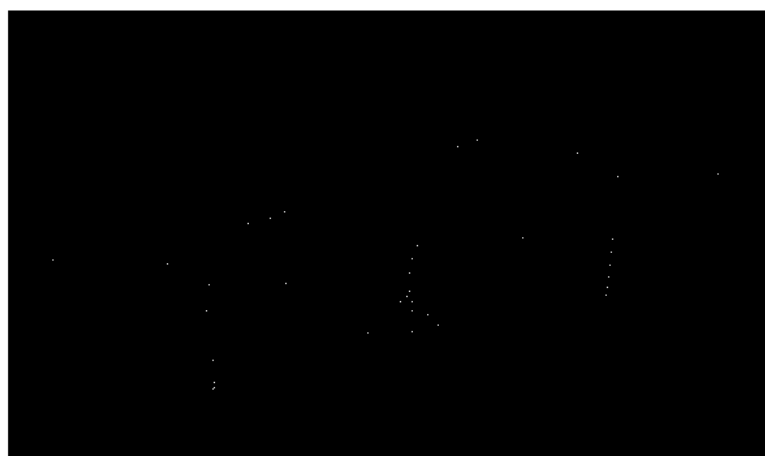


Figure 6: Efekt operacji bmorth branchpoints

1.5. Funkcja SHRINK

Funkcja ta zmniejsza obiekty binarne na obrazie poprzez iteracyjne usuwanie pikseli z zewnętrznych warstw obiektu, zachowując jego ogólny kształt. Parametr liczby iteracji kontroluje, jak bardzo obiekt zostanie zmniejszony:

- SHRINK 10 – wykonuje 10 iteracji zmniejszania obiektu.
- SHRINK 20 – wykonuje 20 iteracji zmniejszania obiektu.
- SHRINK Infinity – iteracje są wykonywane aż do osiągnięcia minimalnego kształtu.

Efektem operacji SHRINK jest zmniejszona wersja obiektów, przydatna np. do redukcji szumu lub upraszczania kształtów przed dalszą analizą.



Figure 7: Efekt operacji shrink

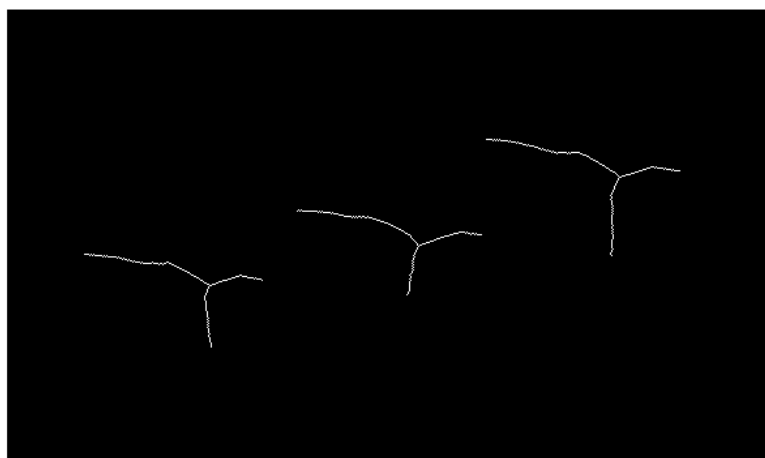


Figure 8: Efekt operacji shrink

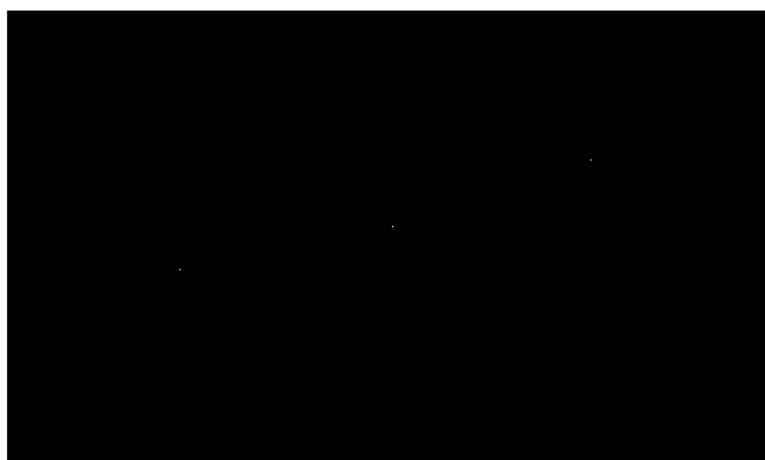


Figure 9: Efekt operacji shrink infinity

1.6. Funckja THIN

Operacja THIN jest procesem cienienia obiektów na obrazie binarnym. Polega na iteracyjnym usuwaniu pikseli z krawędzi obiektów, ale w sposób bardziej kontrolowany niż w przypadku erozji. Celem operacji jest uzyskanie cenniejszej wersji obiektu, która zachowuje jego ogólną topologię, ale zredukowaną do najcieńszej możliwej formy, bez zamiany go w punkt.

- THIN 10 – operacja cienienia jest wykonywana przez 10 iteracji. Obiekt staje się zauważalnie cieńszy, ale nadal rozpoznawalny.
- THIN 20 – cienienie kontynuuje przez 20 iteracji, powodując dalsze redukcje grubości obiektu.
- THIN Infinity – proces trwa aż do osiągnięcia minimalnej grubości obiektu. Efektem jest najcieńsza możliwa wersja obiektu, zachowująca jego spójność.

Funkcja THIN jest często używana w przetwarzaniu obrazu w celu upraszczania kształtów przed dalszą analizą, np. w rozpoznawaniu wzorców lub badaniach struktur.



Figure 10: Efekt operacji thin



Figure 11: Efekt operacji thin

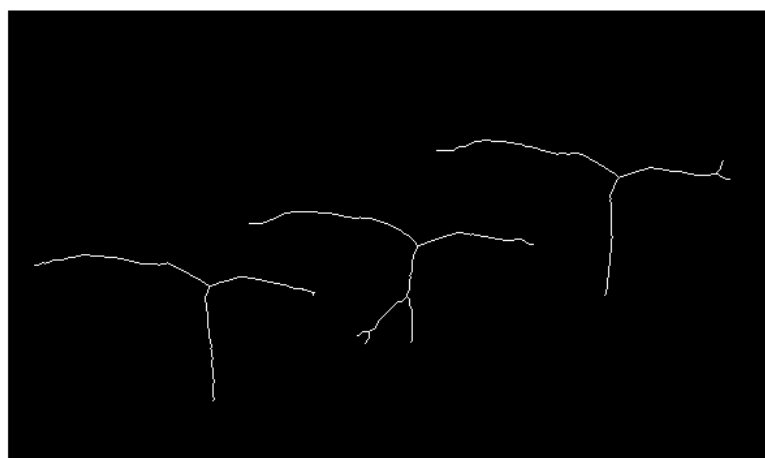


Figure 12: Efekt operacji thin infinity

1.7. Funkcja THICKEN

Operacja THICKEN działa odwrotnie do cieniowania, powodując stopniowe pogrubianie obiektów na obrazie binarnym. Proces polega na iteracyjnym dodawaniu pikseli wokół istniejących obiektów, co skutkuje ich wzrostem. Pogrubianie może być używane w celu wzmocnienia struktur obiektów lub wypełniania drobnych luk.

- THICKEN 10 – pogrubianie obiektów jest wykonywane przez 10 iteracji, powodując zauważalny wzrost obiektów.
- THICKEN 20 – proces kontynuuje przez 20 iteracji, dodatkowo zwiększając rozmiary obiektów.
- THICKEN Infinity – operacja trwa do momentu, aż obiekt osiągnie maksymalny możliwy rozmiar, wypełniając cały dostępny obszar lub przestrzeń między sąsiednimi obiektami.

Funkcja THICKEN znajduje zastosowanie np. w wypełnianiu przerw w strukturach, analizie kształtów oraz w procesach segmentacji obrazu.



Figure 13: Efekt operacji thicken



Figure 14: Efekt operacji thicken

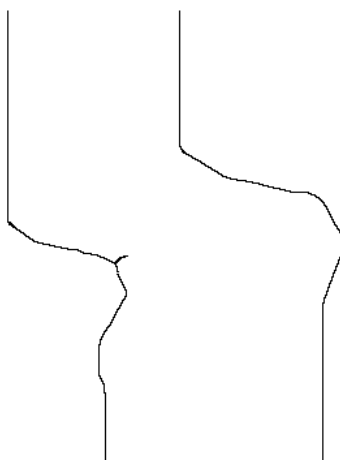


Figure 15: Efekt operacji thicken infinity

1.8. Segmentacja

1.8.1. Segmentacja przez pogrubianie - bwlabel

Segmentacja przez pogrubianie przy użyciu funkcji `bwlabel` polega na identyfikacji i oznaczaniu poszczególnych obiektów w obrazie binarnym. Każdemu zidentyfikowanemu obiektowi przypisywany jest unikalny indeks, co umożliwia dalszą analizę i

wyodrębnianie tych obiektów.

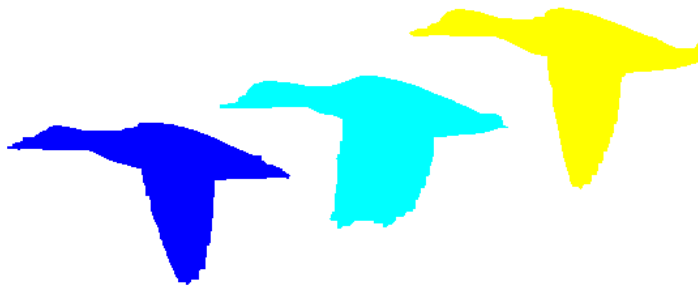


Figure 16: Efekt operacji segmentacji poprzez bwlabel



Figure 17: Wyodrębniona kaczka nr 2

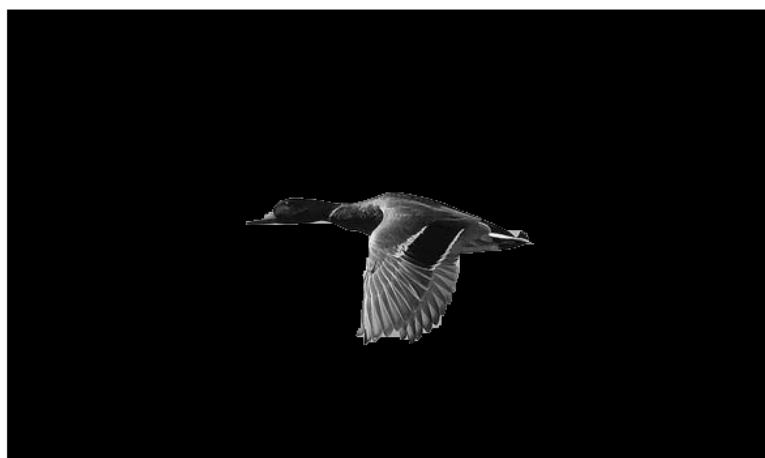


Figure 18: Wyodrębniona kaczka nr 2 naniesiona na obraz oryginalny

1.8.2. Segmentacja przez transformację odległościową - bwdist

Segmentacja przez transformację odległościową opiera się na obliczaniu odległości każdego piksela od najbliższego pikselu tła. Powstałe mapy odległości pozwalają na przeprowadzenie segmentacji obiektów. Transformacja odległościowa znajduje zastosowanie w problemach, gdzie ważne jest precyzyjne rozdzielenie nakładających się obiektów.



Figure 19: Efekt operacji segmentacji odległościowej



Figure 20: Efekt operacji segmentacji odległościowej w negatywie

Podstawowe pojęcia do których algorytm segmentacji działa analogicznie

- Zlewisko - obszar do którego wszystko się zlewa - jak w gorach i woda
- Wododział - granica między nimi
- Jeśli spływa do jednego zbiornika, znaczy to że jest to jeden segment

Rodzaje odległości

- Odległość euklidesowa

$$L_1 = d = \sqrt[1]{a^1 + b^1}$$

$$L_2 = d = \sqrt{a^2 + b^2}$$

$$L_\infty = d = \sqrt[\infty]{a^\infty + b^\infty} = \max\{a, b\}$$

- Odległość Manhattan - City block Mierzona jako suma różnic współrzędnych wzdłuż osi x i y.
- Odległość Kwazi-Eulidesowa - quasi-euclidean
- Odległość Chebysheva - chessboard



Figure 21: Podstawowy efekt segmentacji metodą wododziału.

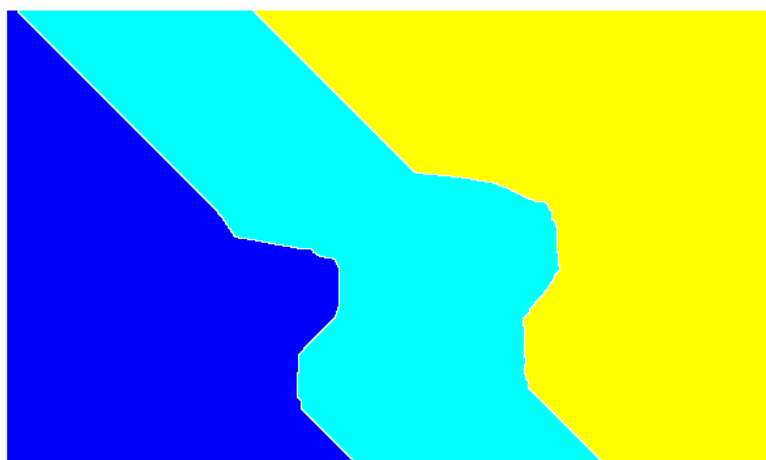


Figure 22: Segmentacja z użyciem odległości typu Chebysheva.

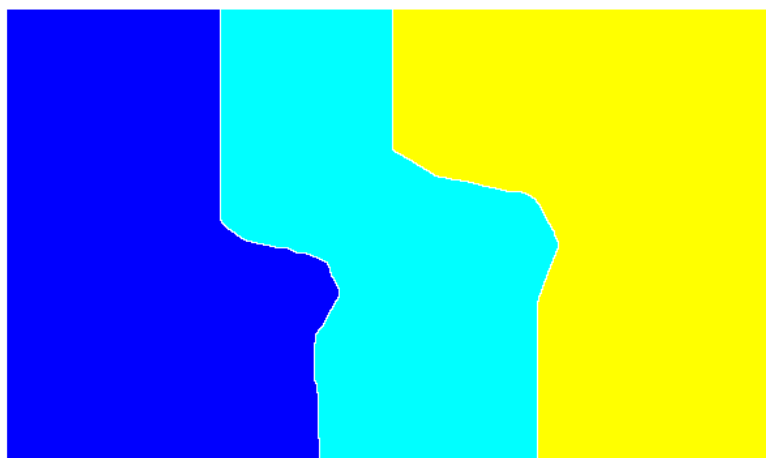


Figure 23: Segmentacja z użyciem odległości Manhattan (City Block).

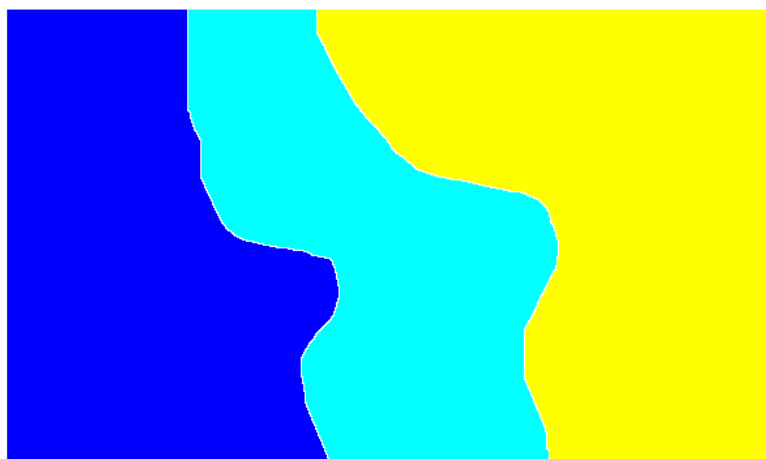


Figure 24: Segmentacja z użyciem odległości Kwazi-Euklidesowej.

1.9. Wnioski

- Operacje morfologiczne są kluczowymi narzędziami w przetwarzaniu obrazów, umożliwiającymi efektywne przetwarzanie i analizę danych przestrzennych. Na przykład operacje takie jak erozja i dylatacja umożliwiają kształtowanie i modyfikowanie struktur obrazu poprzez rozszerzanie lub kurczenie obiektów w zależności od ich cech geometrycznych. Wyszukiwanie samych krawędzi obrazu za pomocą BIM (Binary Image Morphology) pozwala na precyzyjne oddzielenie obiektów od tła, co jest kluczowe w procesie segmentacji.
- Szkieletyzacja i funkcje związane z nią, jak SHRINK, THIN, oraz THICKEN, są wykorzystywane do zmniejszania lub rozszerzania obiektów w celu uzyskania ich optymalnych rozmiarów i kształtów do dalszej analizy. Dzięki nim możliwe jest precyzyjne określenie punktów końcowych szkieletu czy miejsc, gdzie linie szkieletu się rozgałęziają, co może być użyteczne w analizie strukturalnej i przestrzennej.
- Segmentacja obrazu za pomocą funkcji bwlabel i bwdist umożliwia podział obrazu na różne obszary i identyfikację poszczególnych obiektów na podstawie ich cech morfologicznych. Te operacje są nieocenione w klasyfikacji i identyfikacji obiektów w skomplikowanych obrazach, umożliwiając efektywne rozróżnianie struktur i poprawę jakości przetwarzania obrazu.

2. Właściwości obiektu

2.1. Przygotowanie do badania właściwości obiektu

Żeby policzyć niektóre właściwości obiektów użyliśmy obrazu z ptakami, który zbinaryzowaliśmy i wyodrębniliśmy obiekty. Było to znacząco tródniejsze ze względu na "gradientowe" tło obrazu przez co skuteczna binaryzacja jest uniemożliwiona.

Jasność gęsi pokrywa się z jasnością tła nie da się postawić dobrze progu ponieważ niektóre kanały nadrabiają za inne to. Na niebieskim kanale spodziewamy się zobaczyć gęsi najlepiej.

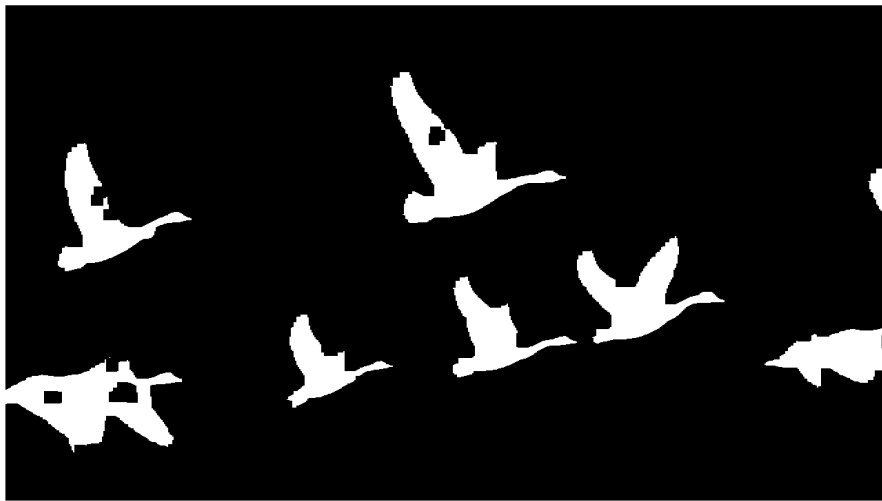


Figure 25: Nieudana próba binaryzacji

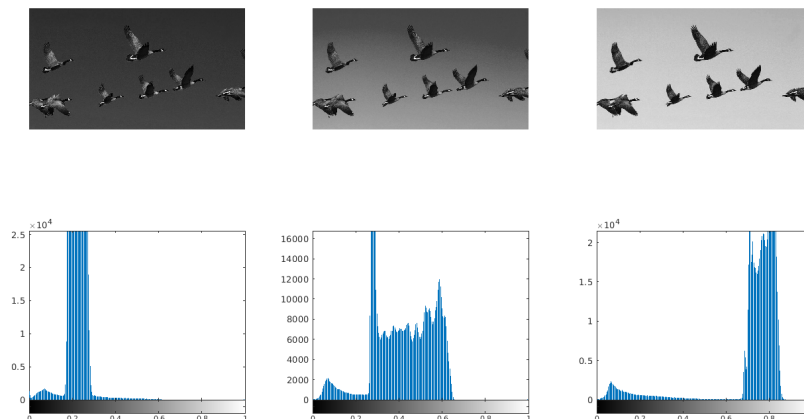


Figure 26: Histogramy poszczególnych kanałów

Na podstawie histogramu dokonaliśmy następujących obserwacji:

- czerwonego mało - prawie wcale
- to co się zmienia w obrazie to ilość zielonego - gradient który nam uniemożliwia prostą binaryzację
- niebieska warstwa daje nam bardzo dużo istotnych informacji

Celem uzyskania najlepszego wyniku binaryzacji możemy złożyć obraz z 2 części: z warstwy Niebieskiej i Czerwonej - pomijając "szkodliwą" zieloną.

```

1      r = imbinarize(im(:,:,1),.3);
2      b = imbinarize(im(:,:,3),.5);
3      bim = r|~b;
4      bim = imopen(bim,ones(7));
5

```

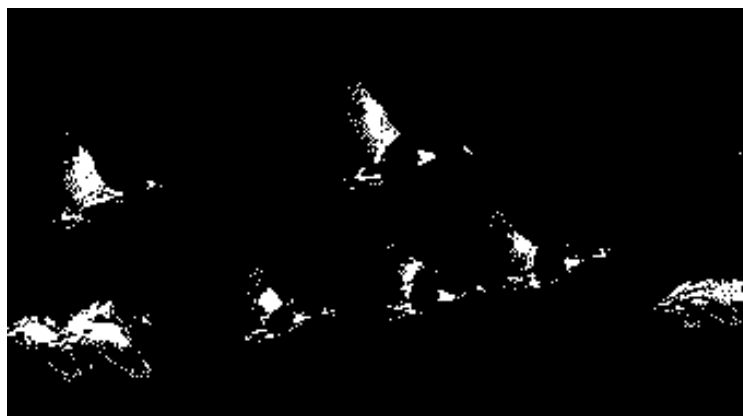


Figure 27: Obraz zbinaryzowany - części

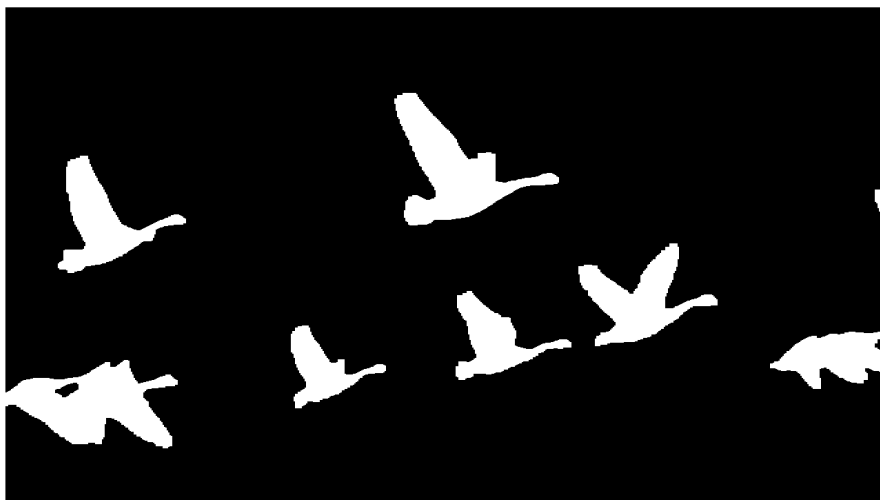


Figure 28: Obraz zbinaryzowany po użyciu operacji otwarcia i zamknięcia

2.2. Wyodrębnienie obiektu

Po powyższym przetworzeniu wstępnym obrazu dokonaliśmy segmentacji i zetyketowaliśmy obiekty na obrazie.



Figure 29: Wyodrębniony obiekt obrazu - przycięty jego bounding boxem

2.3. Właściwości obiektu na obrazie

Parametr	Opis
Area	Pole powierzchni w pikselach, liczy ilość białych pikseli.
Centroid	Środek masy figury, zakładając jednorodność materiału.
Bounding Box	Najmniejszy prostokąt opisujący figurę (x1, x2, y1, y2).
Major Axis Length	Długość osi głównej, największa rozpiętość figury.
Minor Axis Length	Długość osi prostopadłej do głównej osi figury.
Eccentricity	Miara przesunięcia środka masy względem Bounding Boxa (0-1).
Orientation	Kąt orientacji głównej osi figury.
Convex Hull	Zewnętrzna powierzchnia figury (dla 3D).
Image	Wycięty fragment obrazu zawierający figurę.
Filled Image	Obraz z wypełnionymi dziurami (np. tomografia).
Euler Number	Liczba obiektów pomniejszona o liczbę dziur.
Extrema	Współrzędne najbardziej wysuniętych punktów figury.
Solidity	Stosunek pola figury do pola Bounding Boxa.
Perimeter	Obwód figury.
Axes of Feret	Długości osi Bounding Boxa.
Circularity	Miara podobieństwa figury do koła: R_a/R_p .
Shape Coefficients	Współczynniki opisujące kształt, np. Malinowskiej, Danielssona.

Table 1: Parametry opisujące figurę

2.4. Współczynniki

Współczynniki kształtu umożliwiają matematyczne opisanie cech geometrycznych obiektów na obrazie, takich jak ich regularność, podobieństwo do idealnych figur geometrycznych (np. koła, prostokąta), czy też struktura krawędzi. Są używane w analizie obrazu do klasyfikacji i identyfikacji obiektów na podstawie ich kształtu.

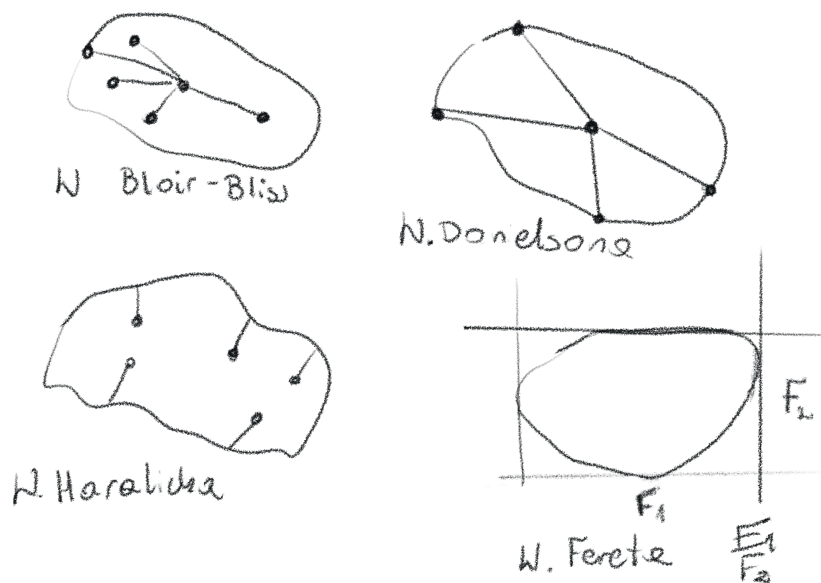


Figure 30: Współczynniki

Jednym z najważniejszych jest współczynnik cyrkularności - ocenia, jak bardzo figura przypomina idealne koło, gdzie wynik bliski 1 oznacza wysokie podobieństwo.

$$C = \frac{4\pi \cdot \text{Pole}}{\text{Obwód}^2}$$

1. Współczynnik Blaira-Blissa

Miara wydłużenia figury, porównująca długość największej osi do rozciągłości figury w prostym kierunku. Często stosowany do rozróżniania kształtów podłużnych i bardziej zbliżonych do koła.

2. Współczynnik Danielsona

Miara regularności figury, bazująca na odległości krawędzi figury od jej środka masy. Figura regularna, np. koło, osiąga niski współczynnik, natomiast figury nieregularne mają wyższe wartości.

$$D = \frac{\sum_{i=1}^n |d_i - \bar{d}|}{n}$$

gdzie d_i to odległość od środka masy do punktu na krawędzi, a \bar{d} to średnia odległość

3. Współczynnik Harlick'a

Średnia odległość punktów figury od najbliższej krawędzi. Stosowany w analizie tekstur i struktur o różnym stopniu "ziarnistości".

$$H = \frac{\sum_{i=1}^n d_{i,k}}{n}$$

gdzie $d_{i,k}$ to o odległość punktu i od najbliższej krawędzi k

4. Współczynnik Fereta

Odnosi się do maksymalnej i minimalnej odległości między dwoma punktami na krawędzi figury (tzw. odległości Fereta). Stosowany do analizy wymiarów nieregularnych kształtów. Jakościowo stosunek boków boundingbox'a.

$$F = \frac{F_1}{F_2}$$

5. Współczynnik Malinowskiej

Miara nieregularności kształtu, porównująca obwód figury do obwodu okręgu o tej samej powierzchni.

$$M = \frac{\text{Obwód}}{2\sqrt{t\pi} \cdot \text{Pole}}$$

6. Współczynnik cyrkularności typu W1

Klasyczny współczynnik cyrkularności, mierzący podobieństwo figury do idealnego koła.

$$C_{w_1} = \frac{4\pi \cdot \text{Pole}}{\text{Obwód}^2}$$

7. Współczynnik cyrkularności typu W2

Miara porównująca powierzchnię figury do powierzchni koła wpisanego w jej Bounding Box.

$$C_{w_2} = \frac{\text{Pole}}{\pi \left(\frac{d}{2}\right)^2}$$

gdzie d to długość średnicy Bounding Boxa.

Nazwa współczynnika	Wartość
Współczynnik Blaira-Blissa (średnia odległość pikseli od środka masy)	6.0071
Współczynnik cyrkularności typu W1	123.3758
Współczynnik cyrkularności typu W2	218.3606
Jeszcze inny współczynnik cyrkularności	3.1325
Współczynnik Danielssona	93.148
Współczynnik Fereta	0.9637
Współczynnik Haralicka	73.8036
Współczynnik Malinowskiej	0.7699

Table 2: Wartości współczynników opisujących obiekt

Nazwa właściwości	Wartość
Area	11955
Centroid	[137.4514; 308.9989]
BoundingBox	[76.5, 200.5], 194×187
MajorAxisLength	187.7972
MinorAxisLength	142.8075
Orientation	-59.9836
ConvexArea	20729
Circularity	0.2802
EulerNumber	1
Solidity	0.5767
Perimeter	732.173

Table 3: Właściwości obiektu na obrazie

2.5. Wnioski

- W trakcie wstępnej obróbki obrazów za pomocą binaryzacji, istotne jest uwzględnienie różnych kanałów kolorów (R, G, B), aby poprawić jakość i dokładność wyników. Daje to możliwość oddzielenia obiektów od tła oraz eliminacji szumów, które mogą zakłócać dalszą analizę.
- Obiekty na obrazie mają swoje właściwości które pozwalają dokładnie w sposób matematyczny określić z jakiego rodzaju figurą mamy do czynienia.
- Właściwości obiektów na obrazach, takie jak pole powierzchni, momenty statystyczne czy długość osi, są istotne dla skutecznej segmentacji i identyfikacji. Analiza tych właściwości pozwala na precyzyjne opisanie obiektów oraz ich klasyfikację, co jest kluczowe dla rozróżniania i klasyfikowania obiektów na obrazach.
- Taka analiza pozwala znajdować zakłucenia albo obiekty outlier'y a następnie je usuwać jeśli są nie porządane na obrazie np.: fragment skrzydła ptaka
- Jest to jeden ze sposobów feature extraction i pozwala to porównywać ze sobą różne figury i określać ich podobieństwo co docelowo prowadzi do umożliwienia prostej klasyfikacji.

3. Sieć neuronowa

Podczas laboratorium zaprojektowano i przeszkolono prostą sieć neuronową w celu klasyfikacji obrazów przedstawiających gęsi i kaczki. Proces ten obejmował wstępną obróbkę obrazów, ekstrakcję cech oraz naukę modelu sieci neuronowej.



Figure 31: Obiekty klasy 0 - gęsi Bernikle kanadyjskie



Figure 32: Obiekty klasy 1 - kaczki/jerzyki

3.1. Perceptron

W projekcie wykorzystano perceptron jako podstawowy element sieci neuronowej, który realizuje klasyfikację na podstawie wyuczonych cech.

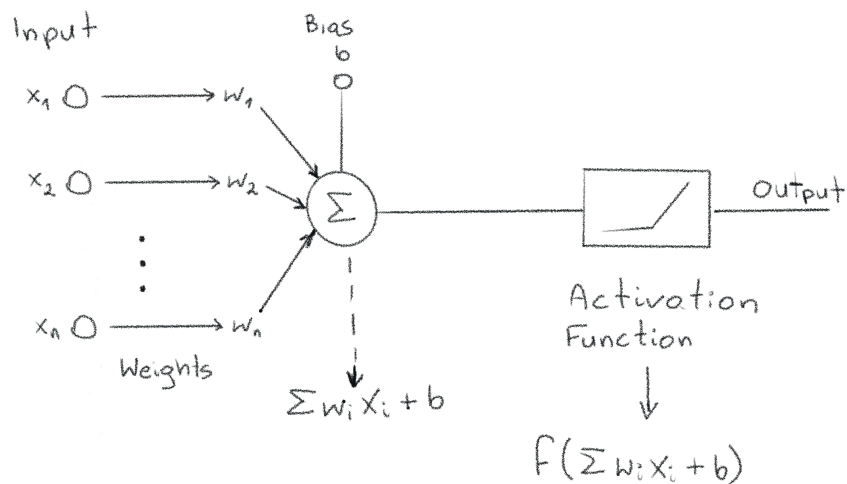


Figure 33: Schemat Perceptronu

3.2. Wstępne przetworzenie obrazów

W ramach wstępnej obróbki obrazy zostały odpowiednio przetworzone, aby uzyskać dane wejściowe dla modelu. Przeprowadzono następujące kroki:

1. Zaimportowano obrazy ptaki.jpg i ptaki2.jpg do programu MATLAB.
2. Wykonano segmentację obrazów, wykorzystując kanały kolorów R i B, co pozwoliło na uzyskanie binarnych masek.
3. W celu usunięcia zakłóceń zastosowano operację morfologiczną otwarcia.
4. Obiekty zostały zidentyfikowane oraz opisane na podstawie wyekstrahowanych cech, takich jak kształt i wielkość.
5. Nietypowe obiekty, odbiegające od wzorca, zostały odfiltrowane na podstawie znormalizowanych odchyleń od średnich wartości.
6. Przetworzone obrazy zbinaryzowano i zapisano do dalszej analizy.



Figure 34: Po wstępnej obróbce - "szczępy" ptaków zostały usunięte a sam obraz jest odpowiednio zbinaryzowany

3.3. Ekstrakcja cech

Wykorzystano zestaw funkcji opisujących właściwości obiektów na obrazie, takich jak:

1. Współczynnik Blaira-Blissa,
2. Współczynniki cyrkularności (W1, W2, Malinowskiej),
3. Parametry Haralicka, Danielssona, Fereta,
4. Shape Coefficients.

Cechy te zostały wyliczone dla każdego obiektu i zapisane w macierzy, która stanowiła dane wejściowe dla sieci neuronowej.

3.4. Zapis wektorowy danych wejściowych i wyjściowych

Dane wejściowe do modelu Każdy obiekt na obrazie jest reprezentowany jako wektor cech wyekstrahowanych na podstawie właściwości geometrycznych i morfologicznych. Niech \mathbf{x}_i będzie wektorem cech dla i -tego obiektu:

$$\mathbf{x}_i = \begin{bmatrix} \text{Blair-Bliss} \\ \text{Circularity (W1)} \\ \text{Circularity (W2)} \\ \text{Danielsson} \\ \text{Feret} \\ \text{Haralick} \\ \text{Malinowska} \\ \text{Shape} \end{bmatrix}_i$$

Dane wejściowe do sieci są macierzą \mathbf{X} , w której każda kolumna to wektor cech dla jednego obiektu:

$$\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \dots \quad \mathbf{x}_n]$$

Dla n obiektów z obu obrazów:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{81} & x_{82} & \dots & x_{8n} \end{bmatrix}$$

gdzie x_{ij} oznacza wartość j -tej cechy dla i -tego obiektu.

Dane wyjściowe (etykiety klas) Wyjściem sieci jest macierz \mathbf{Y} , w której każda kolumna to wektor wskazujący przynależność do klasy. Na przykład, dla dwóch klas: - Gęś: $[1; 0]$ - Kaczka: $[0; 1]$

$$\mathbf{Y} = \begin{bmatrix} 1 & 1 & \dots & 0 & 0 \\ 0 & 0 & \dots & 1 & 1 \end{bmatrix}$$

Dane testowe Podczas testowania wejścia sieci stanowią nowe wektory cech \mathbf{t}_i , które tworzą macierz testową \mathbf{T} :

$$\mathbf{T} = [\mathbf{t}_1 \quad \mathbf{t}_2 \quad \dots \quad \mathbf{t}_m]$$

Analogicznie, etykiety klas dla danych testowych to macierz \mathbf{Y}_{test} :

$$\mathbf{Y}_{\text{test}} = \begin{bmatrix} 1 & 0 & \dots & 1 \\ 0 & 1 & \dots & 0 \end{bmatrix}$$

3.5. Trenowanie sieci neuronowej

Dane wejściowe składały się z cech wyekstrahowanych z przetworzonych obrazów, a etykiety określały przynależność do klasy (gęś lub kaczka). Sieć neuronowa została wytrenowana z wykorzystaniem funkcji `feedforwardnet`. Proces treningu obejmował następujące kroki:

1. Przygotowano dane wejściowe *uin* i etykiety wyjściowe *uout* dla zbioru treningowego.
2. Sieć neuronowa została zainicjalizowana i wytrenowana za pomocą funkcji `train`.
3. Wartości progowe oraz nietypowe dane zostały odpowiednio obsłużone w celu poprawy jakości klasyfikacji.

```
1 %   siec neuronowa
2 uin = [M(1:end-2,:);M2(1:end-2,:)]'; % odcinamy dane testowe
3 n = max(1,[],'all');
4 n2 = max(12,[],'all');
5
6 uout = [repmat([1;0],1,n-2), repmat([0;1],1,n2-2)];
7 tin = [M(end-1:end,:);M2(end-1:end,:)]'; % testowe wejsciowe
8 tout = [1,1,0,0;0,0,1,1];                % testowe wyjsciowe
9
10 nm = feedforwardnet;
11 nm = train(nm,uin,uout);
```

Proces uczenia sieci neuronowej obejmuje dwa główne etapy, które wspólnie pozwalają na dostosowanie wag modelu w celu poprawienia jakości jego predykcji:

1. Feed-forward

Podczas etapu feed-forward dane wejściowe są propagowane przez warstwy sieci neuronowej od wejścia do wyjścia.

- Na podstawie początkowych wag (zazwyczaj zainicjalizowanych losowo) obliczane są odpowiedzi neuronów w kolejnych warstwach, aż do uzyskania wyników na warstwie wyjściowej.
- Na tym etapie sieć dokonuje wstępnych predykcji na podstawie bieżących ustawień wag, które zazwyczaj są mało dokładne na początku procesu uczenia.

2. Backpropagation (propagacja wsteczna)

W etapie propagacji wstecznej obliczany jest błąd predykcji, czyli różnica między wyjściem sieci a wartościami oczekiwanymi (tzw. labelami). Następnie:

- Za pomocą algorytmu opierającego się na regule łańcuchowej (ang. chain rule), gradienty błędów są propagowane wstecz od warstwy wyjściowej przez wszystkie warstwy ukryte do warstwy wejściowej.
- Gradualnie dostosowywane są wagi połączeń między neuronami w taki sposób, aby minimalizować funkcję kosztu (np. średni kwadrat błędów czy entropię krzyżową). Wykorzystuje się w tym celu metodę gradientu prostego lub jej warianty, np. Adam czy RMSProp.

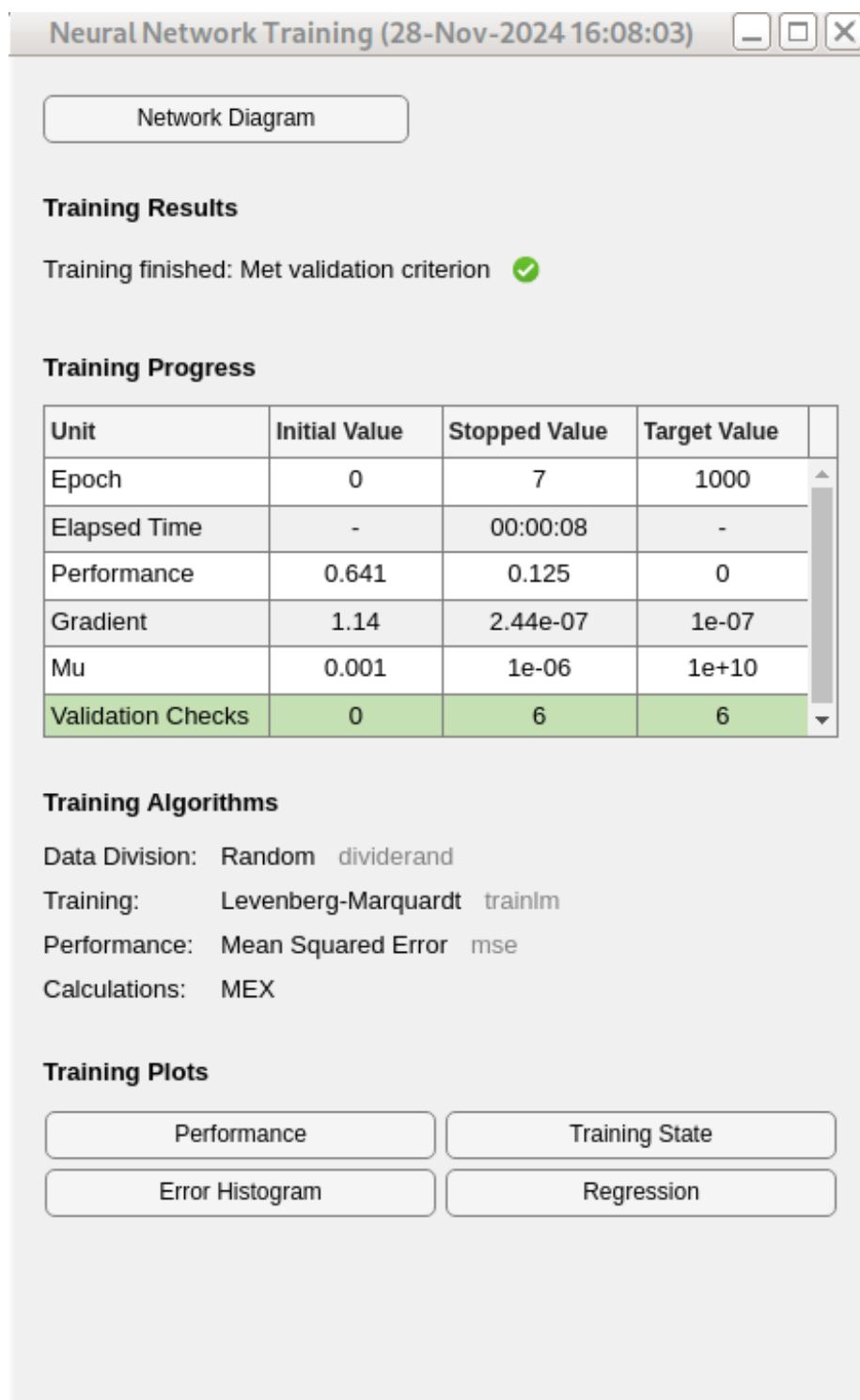


Figure 35: Widok panelu do trenowania

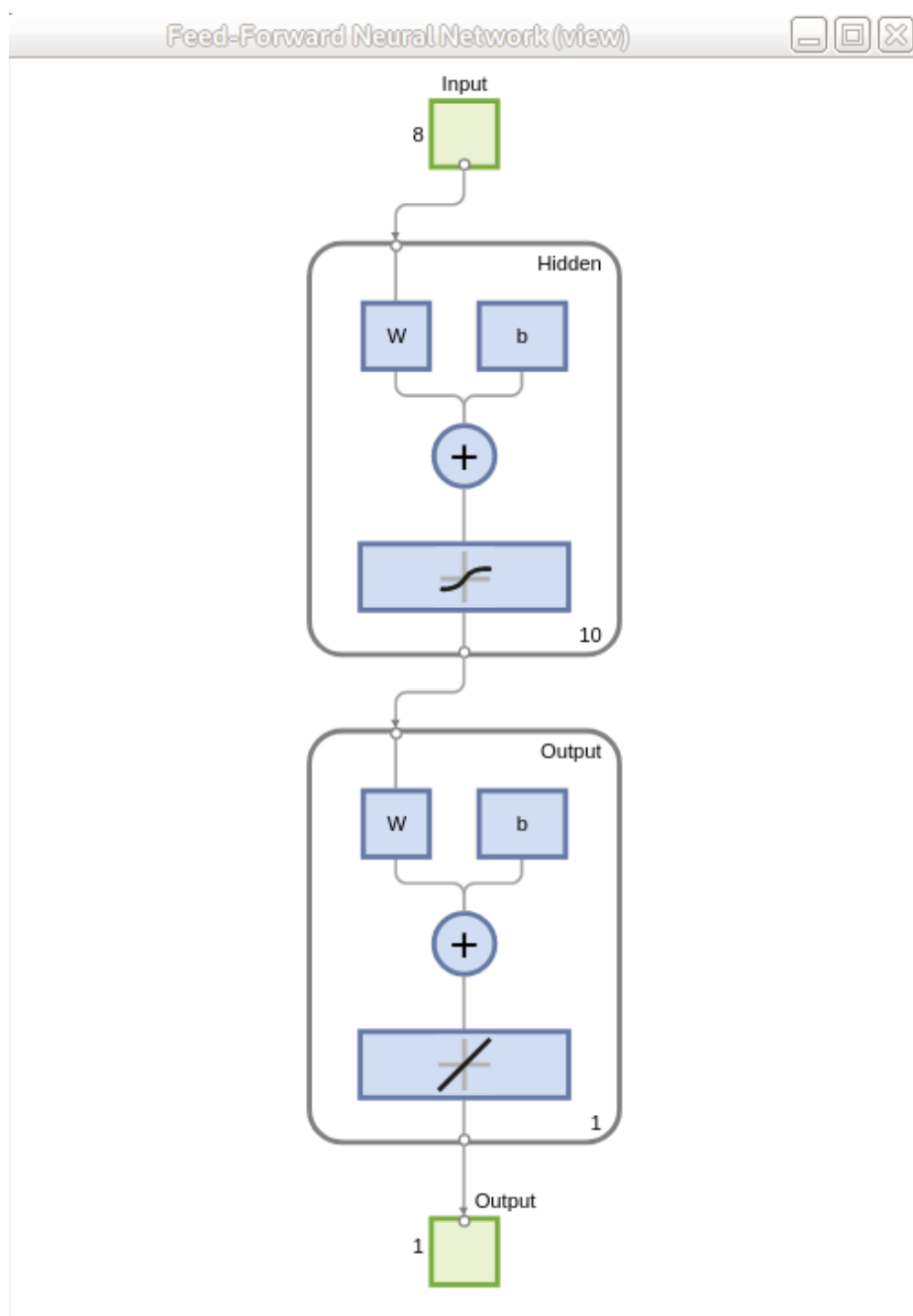


Figure 36: Widok panelu do trenowania

3.6. Testowanie sieci neuronowej

Testowanie sieci obejmowało ocenę klasyfikacji dla danych nowych i nieznanych wcześniej sieci. Wyniki klasyfikacji przedstawiono jako wartości zaokrąglone do najbliższych klas, a dokładność sieci oceniono na podstawie liczby poprawnych predykcji.

Przykład testowania dla danych z macierzy wejściowej i danych testowych:

```
1 round(nm(M(5,:))) % Wynik dla danych treningowych
2 round(nm(tin))    % Wynik dla danych testowych
```

Dzięki tym działaniom możliwe było skuteczne rozróżnienie gęsi od kaczek na podstawie przetworzonych obrazów i wytrenowanej sieci neuronowej.

Po przeprowadzeniu testu modelu na zbiorze testowym ($tin, tout$) otrzymaliśmy następujące wyniki:

$$\mathbf{Output} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Interpretacja wyników:

- Pierwsze dwa obiekty (ptaki) sklasyfikowano jako gęsi (klasa 0).
- Ostatnie dwa obiekty (ptaki) sklasyfikowano jako kaczki (klasa 1).

3.7. Wnioski

- Wytrenowana sieć neuronowa jest w stanie rozróżniać kaczki i gęsi
- Aby zapewnić wyższą jakość klasyfikacji i niezawodność należy użyć większej ilości danych treningowych
- Choć w tym przypadku nie było to konieczne można rozważyć użycie Convolutional Neural Networks - sieć sama będzie sobie skutecznie znajdować i wyciągać cechy (niekoniecznie zrozumiałe dla człowieka)
- Głębsze modele mogą nadawać się lepiej dla bardziej skomplikowanych wzorców