

Analiza Obrazów

Sprawozdanie II

Yuliya Zviarko

Wydział Fizyki i Informatyki Stosowanej

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

2 grudnia 2024

Spis Treści

Spis treści

1	Wprowadzenie	3
1.1	Rozpoznawanie obiektów	3
1.2	Analiza obiektów w języku Matlab	4
2	Laboratorium 5	7
2.1	Binaryzacja: czarne tło i białe kaczki	7
2.2	Szkieletowanie obrazu	8
2.2.1	Punkty końcowe szkieletu	9
2.2.2	Punkty rozgałęzień szkieletu	9
2.3	Modyfikacja kształtu i rozmiaru obiektów binarnych	10
2.3.1	Redukcja obiektów – funkcja <code>shrink</code>	10
2.3.2	Cienkowanie obiektów – funkcja <code>thin</code>	11
2.3.3	Pogrubianie obiektów – funkcja <code>thicken</code>	11
2.4	Segmentacja obrazu binarnego	12
2.4.1	Wizualizacja wybranego obiektu	13
2.5	Transformata odległościowa	13
2.5.1	Transformata odległościowa z metryką chessboard	14
2.5.2	Segmentacja z wykorzystaniem metody watershed	15
2.5.3	Zastosowanie segmentacji na obrazie oryginalnym	15
3	Wnioski	16
4	Laboratorium 6	17
4.1	Binaryzacja obrazu	17
4.2	Operacje morfologiczne	18
4.3	Segmentacja i analiza obiektów	18
4.4	Obliczanie współczynników geometrycznych	20
4.5	Normalizacja i eliminacja odstających wartości	21
4.6	Wyniki analizy obiektów	22
4.7	Wnioski	22
5	Laboratorium 7	24
5.1	Wprowadzenie do sztucznych sieci neuronowych	24
5.1.1	Model matematyczny neuronu	24
5.1.2	Perceptron	25
5.1.3	Sztuczna sieć neuronowa	25
5.1.4	Zastosowanie i proces uczenia sieci neuronowych	26
5.2	Opis eksperymentu	27
5.2.1	Przygotowanie danych wejściowych	27
5.2.2	Trenowanie sieci neuronowej	27
5.3	Testowanie sieci neuronowej	28
5.4	Wnioski	29

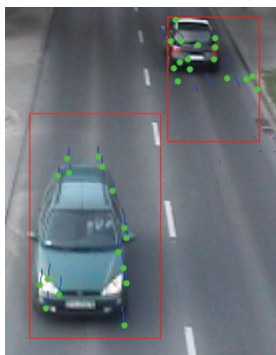
1 Wprowadzenie

1.1 Rozpoznawanie obiektów

Rozpoznawanie obiektów na obrazach stanowi jedno z kluczowych zagadnień w analizie obrazów i znajduje zastosowanie w wielu dziedzinach nauki oraz przemysłu. Proces ten wymaga odpowiedniej „wiedzy”, która umożliwia maszynie prawidłową interpretację przedstawionych danych wizualnych. Podczas gdy człowiek nabywa umiejętność rozpoznawania obiektów w sposób intuicyjny i ciągły w trakcie życia, maszynę należy wyposażyć w taką wiedzę poprzez proces uczenia.

Uczenie maszynowe, stosowane w analizie obrazów, polega między innymi na budowie bazy danych zawierającej reguły i cechy charakterystyczne obiektów. Przykładowo, system może zakładać, że elementy o barwie niebieskiej z prawdopodobieństwem 80% reprezentują niebo, a pionowe linie mogą być interpretowane jako litera „l”. Wiedza systemu rozpoznającego obrazy powinna obejmować zarówno wiedzę ekspercką, specyficzną dla danej dziedziny, jak i dane empiryczne pozyskane w wyniku obserwacji obiektów w różnych warunkach, takich jak zmienne kąty widzenia, oświetlenie czy kontekst.

W rozpoznawaniu obrazów opartym na metodach statystycznych celem jest oszacowanie prawdopodobieństwa, że analizowany obiekt należy do określonej klasy. Wykorzystuje się przy tym porównanie wektorów cech lub elementów składowych (podwzorców) zgodnych z określoną gramatyką. Choć dążymy do maksymalizacji pewności klasyfikacji, osiągnięcie 100% pewności jest praktycznie niemożliwe, co stanowi ograniczenie zarówno dla systemów automatycznych, jak i dla człowieka.



Rysunek 1: Przykład systemu rozpoznającego obiekty, który zidentyfikował samochód na obrazie.

Podobnie jak maszyny, również ludzie mogą błędnie klasyfikować obiekty w nietypowych warunkach, np. przy słabym oświetleniu lub nietypowej perspektywie. Mimo to rozwój systemów analizy obrazów umożliwia ich szerokie zastosowanie w wielu dziedzinach, takich jak:

- wykrywanie pieszych i pojazdów w systemach monitoringu wideo,
- identyfikacja logo w systemach e-commerce,
- ocena uszkodzeń pojazdów w procesach ubezpieczeniowych,
- przyspieszenie diagnoz medycznych poprzez analizę zdjęć rentgenowskich,
- systemy monitoringu całodobowego, wykrywające osoby i niepożądane aktywności oraz powiadamiające odpowiednie służby,
- identyfikacja pracowników na budowie za pomocą kodów QR umieszczonych na kaskach.

Podczas naszych zajęć laboratoryjnych realizowanych w ramach naszego przedmiotu (laboratoria 5–7), koncentrowaliśmy się na wdrażaniu zagadnień związanych z rozpoznawaniem obiektów, w tym:

- identyfikacji obiektów na obrazie,
- binaryzacji obrazów o niejednorodnym oświetleniu,
- analizie obrazów z wykorzystaniem dedykowanych funkcji,
- wprowadzeniu do uczenia maszynowego, w tym sieci neuronowych.

1.2 Analiza obiektów w języku Matlab

W poniższym zestawieniu przedstawiono komendy, które zostały zastosowane w ramach realizowanych ćwiczeń. Komendy te obejmują różne operacje morfologiczne oraz analityczne, które mają na celu przetwarzanie obrazów, detekcję obiektów oraz segmentację.

- **imclose** – $J = \text{imclose}(I, SE)$ wykonuje operację zamknięcia morfologicznego na obrazie w odcieniach szarości lub obrazie binarnym I , przy użyciu elementu strukturalnego SE . Operacja zamknięcia morfologicznego jest dilatacją, a następnie erozją, wykorzystując ten sam element strukturalny dla obu operacji.
- **bwmorph** – $BW2 = \text{bwmorph}(BW, operation)$ stosuje określoną operację morfologiczną na obrazie binarnym BW .
 - **bwmorph(BW, 'shrink', n)** – Zmniejsza obiekty do punktów, usuwając piksele z granic obiektów. Obiekty bez dziur zmniejszają się do punktu, a obiekty z dziurami zmniejszają się do połączonego pierścienia pomiędzy każdą dziurą a zewnętrzną granicą. Ta opcja zachowuje liczbę Eulera.

- **bwmorph(BW, 'thin', n)** – Przekształca obiekty do ich centralnych linii, usuwając piksele z granicy obiektów. Obiekty bez dziur redukują się do minimalnie połączonego szlaku, a obiekty z dziurami przekształcają się do połączonego pierścienia pomiędzy każdą dziurą a zewnętrzną granicą. Ta opcja zachowuje liczbę Eulera.
- **bwmorph(BW, 'thicken', n)** – Pogrubia obiekty przez dodanie pikseli do zewnętrznej części obiektów, aż do momentu, w którym nie połączą się one z innymi obiektami (8-połączenie). Ta opcja zachowuje liczbę Eulera.
- **bwmorph(BW, 'endpoint')** – **bwmorph(sk, 'endpoint')** znajduje punkty końcowe szkieletu. Na przykład:

$$\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array} \rightarrow \begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}$$

Uwaga: Aby znaleźć punkty końcowe, obraz musi być zdyskretyzowany (szkieletowany) przy użyciu **bwmorph(BW, "skel")**.

- **bwmorph(BW, 'branchpoint')** – **bwmorph(sk, 'branchpoint')** znajduje punkty rozgałęzienia szkieletu. Na przykład:

$$\begin{array}{ccccc} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{array} \rightarrow \begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}$$

Uwaga: Aby znaleźć punkty końcowe, obraz musi być zdyskretyzowany (szkieletowany) przy użyciu **bwmorph(BW, "skel")**.

- **bwdist** – **bwdist(bim)** oblicza transformację odległościową obrazu **bim**, przypisując każdemu pikselowi odległość do najbliższego tła. Funkcja ta znajduje zastosowanie w analizie kształtu obiektu.
- **bwdist ('chessboard')** – Oblicza odległość w oparciu o dystans typu „szachownica”. W 2D odległość szachownicową między punktami (x_1, y_1) oraz (x_2, y_2) oblicza się jako:

$$\max(|x_1 - x_2|, |y_1 - y_2|)$$

- **watershed** – **watershed(d)** transformacja wszechstronna, znajdująca „catchment basins” lub „watershed ridge lines” w obrazie, traktując obraz jako powierzchnię, w której jasne piksele reprezentują wysokie wzniesienia, a ciemne piksele reprezentują niskie wzniesienia. Transformacja wododziału może być wykorzystana do segmentacji kontynuowanych obszarów zainteresowania w oddzielne obiekty.

- **regionprops** Funkcja `regionprops` umożliwia wyznaczanie właściwości takich jak pole powierzchni (*area*), środek masy (*centroid*) oraz prostokąt otaczający (*bounding box*) dla każdego obiektu na obrazie. Obsługuje zarówno regiony sąsiadujące, jak i regiony nieciągłe.

Funkcja identyfikuje unikatowe obiekty na obrazach binarnych, wykorzystując 8-połączone sąsiedztwo dla obrazów 2D oraz maksymalną łączność dla obrazów o wyższych wymiarach.

- **feedforwardnet** Funkcja `feedforwardnet` pozwala na tworzenie i trenowanie sieci neuronowych typu feedforward.

Wywołanie `net = feedforwardnet(hiddenSizes, trainFcn)` tworzy sieć z warstwą ukrytą o rozmiarze `hiddenSizes` i funkcją treningową określoną przez `trainFcn`.

2 Laboratorium 5

W ramach zajęć analizowano obraz *kaczki.jpg*, przedstawiający scenę z trzema latającymi kaczkami. Pierwsza z nich, znajdująca się po lewej stronie obrazu, znajduje się poniżej poziomu pozostałych. Druga kaczka, umieszczona w centralnej części obrazu, jest na poziomie środkowym, natomiast trzecia kaczka, znajdująca się po prawej stronie, jest umieszczona nieco wyżej od środkowej. Na obrazie wyraźnie widoczna jest struktura skrzydeł kaczek, zróżnicowane odcienie piór, zaczynając od ciemno-zielonych, czarnych tonów, a kończąc na jasno-brązowych oraz białych fragmentach. W szczególności, widoczna jest wyraźna rozdzielność piór w dolnej części skrzydeł każdej z kaczek. Tło obrazu przedstawia niebo, które nie jest jednolite, widoczne są na nim delikatne odcienie szarości oraz fragmenty słabnących chmur.



Rysunek 2: Oryginalny obraz kaczek wykorzystywany w trakcie zajęć.

Na początku procesu analizowania obraz został wczytany, a następnie znormalizowany do przedziału $[0, 1]$, zgodnie ze standardem wykorzystywanym w języku Matlab, który jest stosowany na wszystkich zajęciach laboratoryjnych.

2.1 Binaryzacja: czarne tło i białe kaczki

Pierwszym krokiem w przetwarzaniu obrazu było zastosowanie binaryzacji i operacji zamknięcia w celu przygotowania danych do dalszej analizy. Funkcja *imbinarize* została wykorzystana do przekształcenia obrazu w formę binarną. Następnie użyto operacji zamknięcia (*imclose*), która łączy dylatację i erozję, pozwalając na wygładzenie kształtów obiektów oraz eliminację drobnych artefaktów.

W efekcie powyższych działań tło obrazu zostało zredukowane do koloru czarnego, a kaczki – obiekty główne – przedstawiono w kolorze białym. Tak przetworzony obraz umożliwia wyraźne rozróżnienie obiektów od tła, co jest kluczowe dla dalszych analiz.

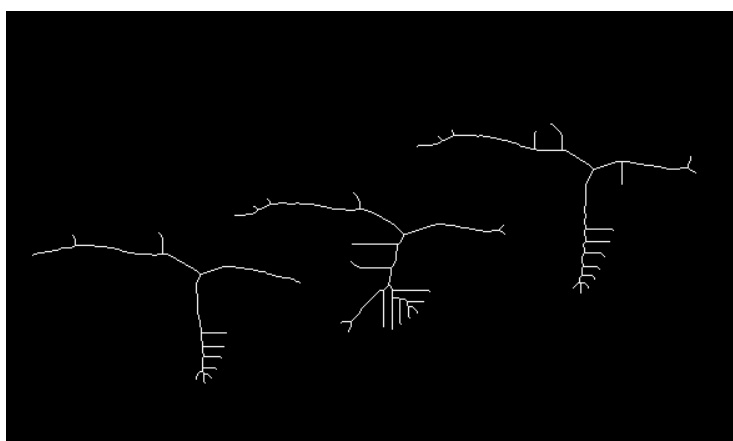


Rysunek 3: Obraz binarny po operacji zamknięcia.

2.2 Szkieletowanie obrazu

Następnym etapem przetwarzania było szkieletowanie obrazu binarnego, przeprowadzone za pomocą funkcji *bwmorph* z parametrem *skel*. Proces ten pozwolił na zredukowanie obiektów do ich podstawowych struktur geometrycznych – szkieletów. Szkielety reprezentują obiekty w postaci cienkich linii jedno- lub kilkupikselowych, zachowując ich topologię oraz kluczowe cechy geometryczne.

Rezultatem szkieletowania jest wyraźny obraz szkieletów kaczek, na którym łatwo rozpoznać główne struktury obiektów. Takie uproszczone reprezentacje mogą być używane np. w rozpoznawaniu kształtów czy identyfikacji obiektów.



Rysunek 4: Szkielet obrazu przedstawiającego kaczki.

2.2.1 Punkty końcowe szkieletu

Kolejnym krokiem analizy było wyznaczenie punktów końcowych na szkieletach obiektów. Proces ten przeprowadzono za pomocą funkcji *bwmorph* z parametrem *endpoint*. Punkty końcowe to zakończenia szkieletów obiektów, które reprezentują miejsca, w których struktury przestają być kontynuowane. Są one szczególnie użyteczne w analizie morfologicznej, ponieważ umożliwiają rozpoznawanie charakterystycznych cech obiektów, takich jak rozgałęzienia, granice czy miejsca styku. Mogą również być wykorzystywane do wyznaczania trajektorii czy analizy topologii.

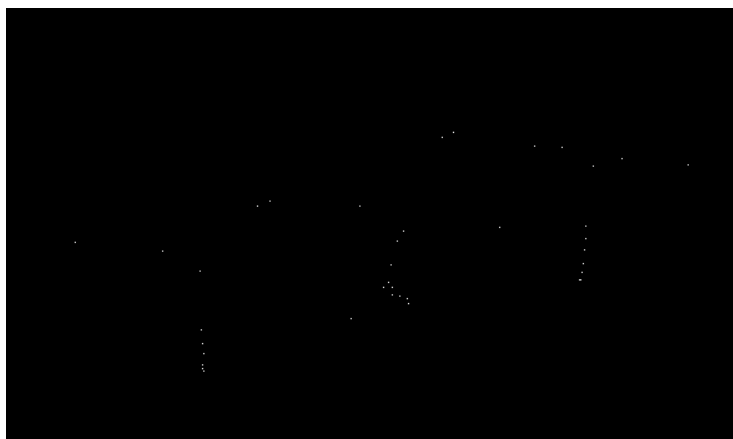
W rezultacie przekształcenia uzyskano obraz zawierający jedynie punkty końcowe, które wizualizowane są jako jasne punkty na czarnym tle.



Rysunek 5: Punkty końcowe szkieletu obrazu kaczek.

2.2.2 Punkty rozgałęzień szkieletu

Za pomocą funkcji *bwmorph* z opcją *branchpoint* zidentyfikowano punkty rozgałęzień na szkieletach obiektów. Punkty te wskazują miejsca, w których dochodzi do podziału szkieletu na różne gałęzie, co odgrywa kluczową rolę w procesach odwzorowywania struktur oraz analizie ich właściwości topologicznych.



Rysunek 6: Punkty rozgałęzień szkieletu obrazu kaczek.

2.3 Modyfikacja kształtu i rozmiaru obiektów binarnych

2.3.1 Redukcja obiektów – funkcja shrink

Po zastosowaniu funkcji *bwmorph* z parametrem *shrink* i liczbą iteracji ustawioną na 10 uzyskano obraz, na którym rozmiar obiektów został zauważalnie zmniejszony. Jednak nie wszystkie obiekty zostały zredukowane do pojedynczych pikseli, ponieważ nadal zachowują niewielkie, zwarte struktury.

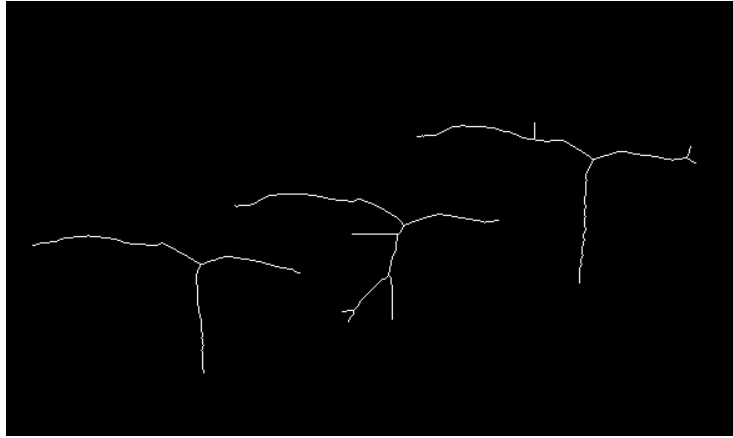
Dzięki regulacji liczby iteracji w funkcji *shrink* możliwe jest dalsze zmniejszanie obiektów, aż do uzyskania pełnej redukcji do pojedynczych pikseli. W przypadku przedstawionego wyniku zwiększenie wartości iteracji może być konieczne do osiągnięcia pożądanego efektu.



Rysunek 7: Obraz po redukcji obiektów do pojedynczych pikseli przy użyciu funkcji *shrink*.

2.3.2 Cienkowanie obiektów – funkcja *thin*

Zastosowanie funkcji *bwmorph* z parametrem *thin* pozwoliło na przekształcenie obiektów w ich uproszczone wersje, zredukowane do pojedynczej warstwy pikseli. Uzyskany obraz wiernie odwzorowuje kontury oraz strukturę szkieletową kaczek, zachowując ich kluczowe cechy geometryczne.



Rysunek 8: Obraz po cienkowaniu obiektów.

2.3.3 Pogrubianie obiektów – funkcja *thicken*

W realizacji pogrubiania obiektów przy użyciu funkcji *bwmorph* z parametrem *thicken* uzyskano obraz, na którym jasne obszary (reprezentujące obiekty, w tym kaczkę) zostały wyraźnie powiększone. Pogrubianie obiektów pozwala na rozszerzenie ich granic, dzięki czemu prawie każdy piksel na obrazie zostaje przypisany do najbliższego obiektu, z wyjątkiem granic między obiektami. Ta operacja zachowuje liczbę obiektów, ale umożliwia lepsze uwidocznienie struktury, co może być przydatne w analizie przestrzennej i identyfikacji obszarów zajmowanych przez obiekty.

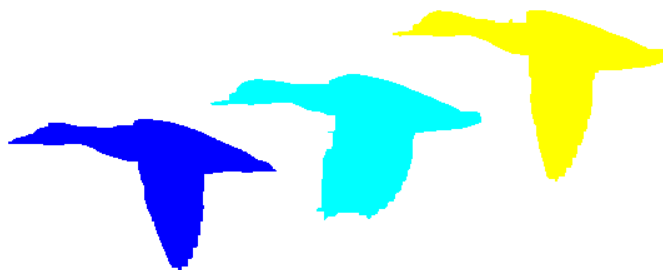


Rysunek 9: Obraz po pogrubieniu obiektów.

2.4 Segmentacja obrazu binarnego

Do przeprowadzenia segmentacji obrazu wykorzystano funkcję *bwlabel*, która przypisuje unikalne etykiety poszczególnym obiektom na obrazie binarnym. Funkcja ta przyjmuje obraz binarny, a zwraca obraz, w którym tło jest oznaczone zerem, a każdy piksel obiektu posiada przypisaną liczbę całkowitą identyfikującą dany obiekt. Wszystkie piksele danego obiektu mają tę samą wartość etykiety, a żadne inne piksele nie mają tej samej wartości.

Operacja ta umożliwia jednoznaczną identyfikację obiektów oraz ich późniejszą analizę, co jest kluczowe w wielu zastosowaniach analizy morfologicznej i rozpoznawania obiektów na obrazach binarnych.



Rysunek 10: Wynik segmentacji obrazu binarnego.

2.4.1 Wizualizacja wybranego obiektu

Po segmentacji możliwe jest wyróżnienie dowolnego obiektu na podstawie przypisanej etykiety. W przedstawionym przykładzie zwizualizowano obiekt o etykiecie numer 2 przy użyciu funkcji pomocniczej *label2rgb*:



Rysunek 11: Wizualizacja obiektu o etykiecie numer 2 po segmentacji.

2.5 Transformata odległościowa

Transformata odległościowa jest techniką morfologiczną, która przypisuje każdemu pikselowi obrazu wartość odległości do najbliższego piksela należącego do tła, obliczaną za pomocą funkcji *bwdist*. Proces ten tworzy mapę, na której wartości odległości są reprezentowane w różnych odcieniach szarości – im piksel jest dalej od obiektów, tym jego wartość jest wyższa (jaśniejszy).

Uzyskane wyniki pozwalają na analizę kształtu obiektów, umożliwiając m.in. identyfikację granic obiektów, ocenę ich rozmiaru, kształtu czy lokalizacji względem tła. Transformata odległościowa może być również wykorzystywana do dalszej segmentacji lub analizy morfologicznej, szczególnie przy rozpoznawaniu obiektów o bardziej złożonych strukturach.



Rysunek 12: Transformata odległościowa obliczona dla obrazu binarnego.

2.5.1 Transformata odległościowa z metryką chessboard

Transformata odległościowa z metryką *chessboard* różni się od tradycyjnej metryki tym, że odległość każdego piksela od tła jest obliczana z uwzględnieniem metryki naśladującej ruchy na planszy szachowej, gdzie odległość mierzy się jako największą z różnic współrzędnych w kierunku poziomym lub pionowym. Takie podejście pozwala na uzyskanie dokładniejszego odwzorowania geometrycznego obiektów o kanciastych, prostokątnych krawędziach.

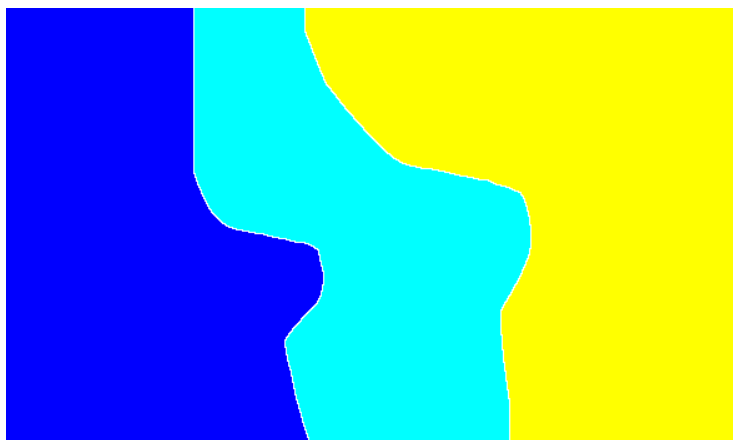
Metryka *chessboard* jest najbardziej efektywna w analizie obiektów o wyraźnych, prostokątnych krawędziach, ponieważ pozwala na precyzyjne odwzorowanie ich struktury, lepiej uwzględniając charakterystyczne prostokątne formy obiektów.



Rysunek 13: Transformata odległościowa z metryką "chessboard".

2.5.2 Segmentacja z wykorzystaniem metody watershed

Metoda *watershed* (wododziału) jest techniką segmentacyjną, która traktuje obraz jako powierzchnię, gdzie jasne piksele reprezentują wyższe elewacje, a ciemne piksele – niższe. Transformata w połączeniu z metodą *watershed* umożliwia dokładne wyznaczenie granic między obiektami, przypisując im unikalne etykiety identyfikujące poszczególne segmenty obrazu.



Rysunek 14: Segmentacja obiektów z wykorzystaniem metody wododziału.

2.5.3 Zastosowanie segmentacji na obrazie oryginalnym

Na końcowym etapie segmentacji zastosowano wyniki analizy do oryginalnego obrazu. W tym przypadku, wyróżniono obiekt o etykiecie numer 2, umożliwiając jego wizualną identyfikację w kontekście pełnej sceny.



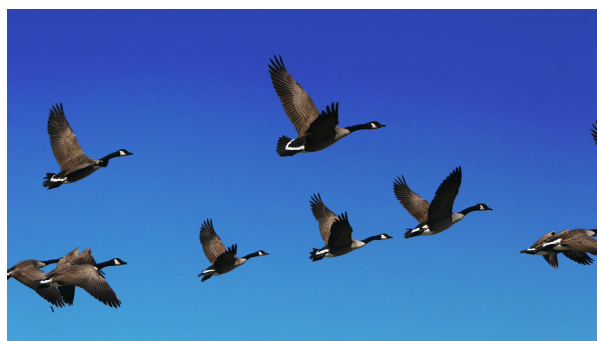
3 Wnioski

W ramach tych zajęć laboratoryjnych przeprowadzono szereg operacji, które pozwoliły na skuteczną analizę obrazu przedstawiającego kaczki. Proces ten obejmował binaryzację, szkieletowanie, modyfikację kształtu obiektów, segmentację, oraz zastosowanie transformacji odległościowej. Każdy z tych kroków umożliwił uzyskanie coraz bardziej szczegółowych i precyzyjnych reprezentacji obiektów, co stanowiło podstawę do dalszych analiz morfologicznych i rozpoznawczych.

Rezultaty wskazują na przydatność tych technik, szczególnie w kontekście identyfikacji obiektów oraz rozpoznawania ich kształtu i struktury.

4 Laboratorium 6

W ramach szóstych zajęć laboratoryjnych analizowano obraz przedstawiający dziewięć ptaków, z których jeden znajduje się w prawym środkowym rogu i jest prawie niewidoczny – widoczny jest jedynie niewielki fragment skrzydła. Tło obrazu to niebo o gradientzie kolorów, przechodzące od ciemnego niebieskiego do jaśniejszych odcieni, co utrudniało wyizolowanie ptaków za pomocą standardowej binaryzacji.



Rysunek 15: Obraz przedstawiający ptaki na tle gradientowego nieba.

4.1 Binarizacja obrazu

Problem ten rozwiązano poprzez zastosowanie binaryzacji kanałów obrazu w przestrzeni RGB. Wykorzystano binarizację kanału czerwonego oraz inwersję binarizacji kanału niebieskiego, a następnie połączono oba wyniki za pomocą operacji logicznej OR, co pozwoliło skutecznie wyodrębnić ptaki na tle gradientowego nieba.



Rysunek 16: Wynik binaryzacji kanałów obrazu z zastosowaniem operacji logicznej OR.

4.2 Operacje morfologiczne

W celu poprawy jakości binaryzacji i eliminacji zakłóceń zastosowano operacje morfologiczne: otwarcie (*imopen*), która usuwa małe obiekty i zakłócenia w obrazie, które nie należą do pożądaných struktur (np. szum lub niewielkie fragmenty tła), oraz zamknięcie (*imclose*), służącą do wypełnienia drobnych ubytków w obiektach, takich jak niewielkie dziury lub przerwy w krawędziach, wykorzystując strukturalny element w kształcie kwadratu o rozmiarze 5x5 pikseli.

Uzyskano bardziej spójny obraz z wyraźnie odseparowanymi obiektami (ptakami), co umożliwiło ich precyzyjną segmentację i analizę w kolejnych krokach.



Rysunek 17: Obraz po zastosowaniu operacji otwarcia i zamknięcia.

4.3 Segmentacja i analiza obiektów

Po przetworzeniu obrazu dokonano segmentacji, czyli etykietowania obiektów w obrazie. W wyniku tego procesu uzyskano mapę etykiet, gdzie każdemu obiektowi przypisano unikalną liczbę. Następnie, dla obiektu o etykiecie 4, wyodrębniono szereg właściwości geometrycznych, takich jak pole powierzchni, centroid, obwód, prostokąt ograniczający, ekscentryczność oraz orientację.



Rysunek 18: Wyodrębniony obiekt (po lewej) i jego maska (po prawej).

Funkcja `regionprops` w MATLAB umożliwia wyznaczanie właściwości geometrycznych oraz intensywnościowych dla każdego obiektu w obrazie. Może być stosowana zarówno do obrazów binarnych, jak i do obrazów z etykietami (label matrix).

Tabela poniżej przedstawia kluczowe właściwości, które można uzyskać za pomocą tej funkcji:

Tabela 1: Najważniejsze właściwości dostępne w `regionprops`.

Nazwa właściwości	Opis
Area	Liczba pikseli w regionie.
BoundingBox	Współrzędne i rozmiar prostokąta ograniczającego.
Centroid	Środek masy regionu.
Circularity	Okrągłość obiektu.
ConvexArea	Liczba pikseli w wypukłym obrazie regionu.
Eccentricity	Ekscentryczność elipsy opisującej region.
Orientation	Kąt orientacji elipsy względem osi poziomej.
Solidity	Stosunek obszaru do wypukłego obszaru regionu.

```
l = bwlabel(bim);
n = max(l(:)); % liczba obiektow
a = regionprops(l == 4, 'all'); % wlasciwosci ob. #4
```

Listing 1: Kod segmentacji i analizy obiektów

Właściwości obiektu o etykiecie 4:

- **Pole powierzchni (Area):** 16677.00
- **Centroid (środek masy):** [647.40, 234.44]
- **Obwód (Perimeter):** 862.24
- **BoundingBox:** [563.50, 120.50, 243.00, 196.00]
- **Eccentricity:** 0.75

4.4 Obliczanie współczynników geometrycznych

W analizie obiektów, oprócz standardowych właściwości, takich jak pole powierzchni, obwód czy centroid, uwzględniono dodatkowe współczynniki geometryczne. Umożliwiają one bardziej szczegółowe opisanie kształtu i struktury analizowanych obiektów. Do takich współczynników należą:

- **Blair-Bliss:** Mierzy stopień zaokrąglenia obiektu. Wartość tego współczynnika jest szczególnie użyteczna w analizie obiektów o regularnych, prawie okrągłych kształtach.
- **CircularityS:** Określa, jak blisko kształt obiektu zbliża się do pełnego koła. Wyższa wartość oznacza bardziej okrągły kształt.
- **CircularityL:** Podobnie jak CircularityS, jednak w tym przypadku używane są inne miary geometryczne, które mogą lepiej odzwierciedlać kształty bardziej eliptyczne.
- **Feret:** Mierzy największą odległość między dwoma punktami na obwodzie obiektu, co daje miarę jego rozciągłości w różnych kierunkach.
- **Danielsson:** Opisuje odległość każdego piksela w obiekcie do jego najbliższego brzegu, co może pomóc w analizie krawędziowych właściwości obiektów.
- **Haralick:** Zbiór współczynników statystycznych, które opisują teksturę obiektów na podstawie analizy macierzy współczynnika podobieństwa.
- **Malinowska:** Współczynnik, który opisuje stopień asymetrii obiektu, mierząc jego proporcje.
- **Shape:** Ogólny współczynnik kształtu, który łączy różne aspekty geometrii obiektów, takie jak długość, szerokość i skomplikowanie krawędzi.

Do obliczania tych współczynników wykorzystano funkcje zewnętrzne, które wymagają załadowania odpowiednich skryptów. Każdy współczynnik jest obliczany indywidualnie dla obiektów, a następnie zapisany w macierzy. W macierzy tej każdy wiersz odpowiada pojedynczemu obiektowi, a każda kolumna zawiera wartość danego współczynnika:

```

f = {@A05RBlairBliss, @A05RCircularityL,
    @A05RCircularityS,
    @A05RDanielsson, @A05RFeret, @A05RHaralick,
    @A05RMalinowska, @A05RShape};
fun_size = 8;
macierz_wsp_gesi = ones(n, fun_size);

for i = 1:n
    for j = 1:fun_size
        macierz_wsp_gesi(i, j) = f{j}(1 == i);
    end
end

```

Listing 2: Obliczanie współczynników geometrycznych

macierz_wsp_gesi								
8x8 double								
	1	2	3	4	5	6	7	8
1	6.7876	263.5606	148.6808	103.1748	0.4941	79.0091	0.7727	3.1423
2	5.9324	202.7634	120.5308	91.2891	0.9572	71.1760	0.6823	2.8300
3	5.3032	155.3352	92.0857	84.4064	0.9291	53.2458	0.6869	2.8455
4	6.5785	267.0620	147.9253	103.3688	0.8601	86.4291	0.8054	3.2594
5	5.6707	178.2535	104.8725	80.9426	0.8070	61.9621	0.6997	2.8890
6	6.1675	230.1380	124.7612	114.3766	0.7295	69.0464	0.8446	3.4027
7	5.8086	167.7493	106.3015	77.7273	0.4857	57.7694	0.5781	2.4903
8	3.3668	45.8366	36.2665	17.5864	2.5652	19.8241	0.2639	1.5974

Rysunek 19: Przykład macierzy współczynników geometrycznych.

4.5 Normalizacja i eliminacja odstających wartości

W celu ułatwienia porównania obiektów o różnych rozmiarach, wartości współczynników geometrycznych zostały poddane normalizacji. Znormalizowane wartości obliczono, odejmując średnią dla każdego współczynnika i dzieląc przez odchylenie standardowe. Taki proces pozwolił na uzyskanie jednolitej skali dla współczynników geometrycznych obiektów, co umożliwiło ich porównanie niezależnie od ich wielkości.

Aby usunąć obiekty z odstającymi wartościami, zastosowano filtrację, która wykrywała obiekty, których współczynniki odbiegały od średniej o więcej niż 3 odchylenia standardowe. Obiekty, które przekroczyły ten próg, zostały uznane za odstające i usunięte z dalszej analizy.

Po usunięciu obiektów z odstającymi wartościami, obraz został ponownie przetworzony, a obiekty zostały ponownie etykietowane, co umożliwiło dalszą analizę.



Rysunek 20: Obraz po usunięciu kaczek z odstającymi wartościami.

4.6 Wyniki analizy obiektów

Po usunięciu obiektów z odstającymi wartościami, uzyskano finalną segmentację obrazu, na którym obiekty zostały ponownie etykietowane. Dzięki zastosowanej metodzie, obiekty zostały skutecznie wyodrębnione, a współczynniki geometryczne stały się istotnym elementem charakterystyki każdego z obiektów.



Rysunek 21: Finalna segmentacja po usunięciu kaczek i etykietowaniu obiektów.

4.7 Wnioski

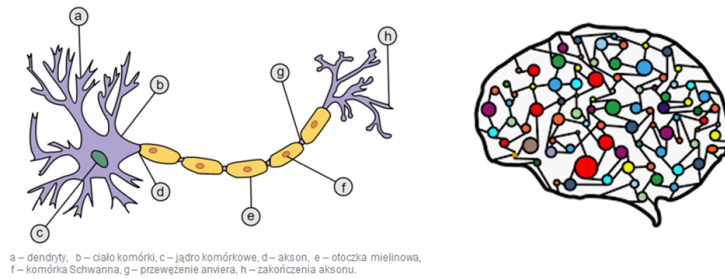
Podczas analizy obrazu przedstawiającego ptaki na tle gradientowego nieba skutecznie zastosowano zaawansowane metody binaryzacji i operacje morfologiczne, które pozwoliły na wyodrębnienie obiektów i eliminację zakłóceń. Wykorzystanie segmentacji umożliwiło precyzyjną analizę kształtów i właściwości geometrycz-

nych obiektów. Dodatkowo, normalizacja oraz filtracja pozwoliły na usunięcie odstających wartości, co poprawiło jakość wyników. Ostatecznie uzyskano dokładną segmentację obrazu z prawidłowo opisanymi obiektami, co dowodzi skuteczności zastosowanych metod.

5 Laboratorium 7

5.1 Wprowadzenie do sztucznych sieci neuronowych

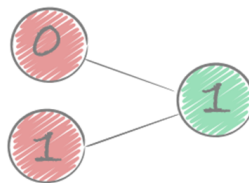
Inspiracja naturą odgrywała istotną rolę w rozwoju technologii. Przykładem tego jest proces konstruowania samolotów, które na początku czerpały wzorce z ptaków, jednak wraz z rozwojem technologii zyskały własne, unikalne rozwiązania, odchodząc od dosłownego naśladowania natury. Podobny mechanizm zaobserwowano w kontekście badań nad ludzkim mózgiem, który zainspirował powstanie **sztucznych sieci neuronowych** (ang. **Artificial Neural Networks, ANN**). Koncepcja ta została po raz pierwszy przedstawiona w roku **1943** przez neurofizjologa Warrena McCullocha oraz matematyka Waltera Pittsa, którzy zaproponowali uproszczony model matematyczny oparty na idei ludzkiego neuronu.



Rysunek 22: Ilustracja inspiracji ludzkim mózgiem w kontekście sztucznych sieci neuronowych.

5.1.1 Model matematyczny neuronu

McCulloch i Pitts zaproponowali model składający się z binarnych wejść oraz jednego wyjścia. Wyjście aktywuje się, jeśli suma wejść przekroczy zdefiniowany próg. Mimo swojej prostoty, taki układ może być podstawą do rozwiązywania problemów logicznych w kontekście sieci neuronowej, gdzie liczne połączone neurony współpracują w celu rozwiązania złożonych zagadnień.

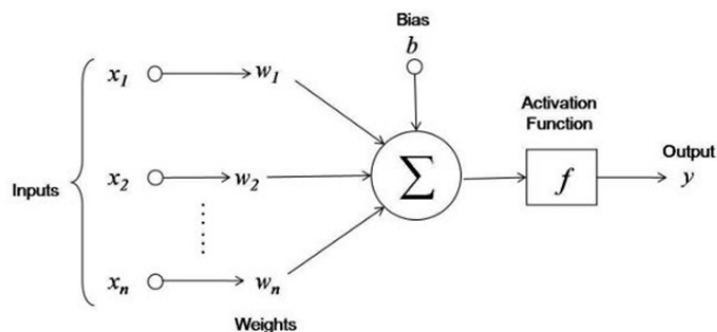


Rysunek 23: Model matematyczny neuronu zaproponowany przez McCullocha i Pittsa.

5.1.2 Perceptron

Rozwinięciem prostego modelu neuronu jest **perceptron**, zaproponowany przez Franka Rosenblatta w roku **1957**. Perceptron wprowadza kilka istotnych zmian względem pierwotnego modelu:

- Na wejściu i wyjściu mogą występować liczby rzeczywiste, które określane są jako **węzły**.
- Połączenia między węzłami są określone przez **wagi**, które są optymalizowane w trakcie procesu uczenia.
- Wynik perceptronu to suma wartości z warstw poprzednich pomnożonych przez wagi, na którą nakładana jest **funkcja aktywacji**. Funkcja ta wprowadza nieliniowość do modelu, co umożliwia wychwytywanie bardziej złożonych zależności.



Rysunek 24: Schemat perceptronu.

Perceptron stanowi podstawowy element sztucznej sieci neuronowej, której celem jest modelowanie bardziej złożonych procesów.

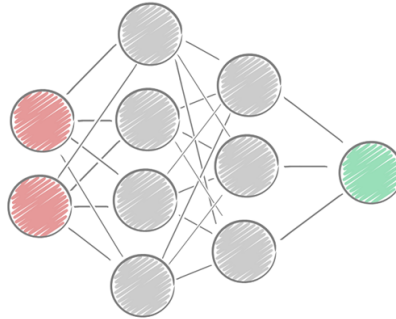
5.1.3 Sztuczna sieć neuronowa

Pojedynczy neuron, mimo swoich możliwości, jest niewystarczający do rozwiązywania złożonych problemów. Dlatego neurony łączy się w warstwy, tworząc **sztuczną sieć neuronową**. W takiej architekturze jedna warstwa przekazuje wyniki kolejnej, umożliwiając hierarchiczne przetwarzanie danych.

Przykładowa sieć neuronowa składa się z trzech rodzajów warstw:

- **Warstwa wejściowa (input layer)** – odpowiada za zbieranie danych wejściowych i przekazywanie ich dalej. Węzły warstwy wejściowej są połączone z każdym węzłem kolejnej warstwy.

- **Warstwa ukryta (hidden layer)** – odpowiada za przetwarzanie danych i poszukiwanie zależności. Może zawierać wiele warstw. Im więcej warstw ukrytych, tym bardziej złożone zależności może modelować sieć.
- **Warstwa wyjściowa (output layer)** – zwraca końcowy wynik przetwarzania.



Rysunek 25: Przykład sztucznej sieci neuronowej z warstwą wejściową, ukrytą i wyjściową.

W przypadku sieci, w której każdy węzeł warstwy jest połączony z każdym węzłem kolejnej warstwy, mówimy o **perceptronie wielowarstwowym (ang. Multi-Layer Perceptron, MLP)**. Sieci zawierające więcej niż jedną warstwę ukrytą określa się mianem **głębokich sieci neuronowych (ang. Deep Neural Networks)**, co wiąże się z koncepcją **głębokiego uczenia (ang. Deep Learning)**.

5.1.4 Zastosowanie i proces uczenia sieci neuronowych

Sztuczne sieci neuronowe doskonale sprawdzają się w rozwiązywaniu problemów o nieliniowym i złożonym charakterze, takich jak przetwarzanie:

- obrazów,
- dźwięku,
- tekstu,
- logów.

Proces uczenia sieci może przebiegać w dwóch głównych etapach:

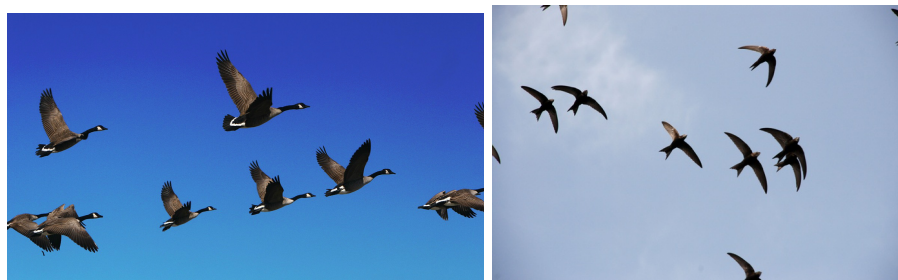
1. **Feed-forward** – przepływ danych od wejścia do wyjścia, podczas którego obliczane są wstępne wyniki na podstawie losowo zainicjalizowanych wag.
2. **Backpropagation** – proces propagacji wstecznej, podczas którego za pomocą metod opartych na gradientach i regule łańcuchowej optymalizowane są wagi połączeń między węzłami, aby minimalizować błąd predykcji.

Proces wielokrotnego uczenia wzmacnia połączenia między neuronami, umożliwiając sieci coraz lepsze rozwiązywanie złożonych problemów.

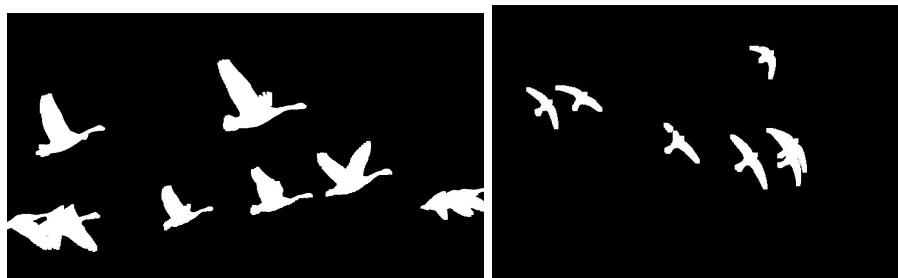
5.2 Opis eksperymentu

5.2.1 Przygotowanie danych wejściowych

Obrazy *ptaki.jpg* i *ptaki2.jpg* zostały wczytane do programu MATLAB i poddane wstępnej obróbce. Przeprowadzono segmentację na podstawie kanałów kolorów R i B, co pozwoliło na wygenerowanie binarnych masek. Następnie zastosowano filtry morfologiczne (operacja otwarcia) w celu eliminacji zakłóceń, a obiekty zostały etykietowane oraz opisane za pomocą cech, takich jak kształt i wielkość. Obrazy po binaryzacji oraz po usunięciu nietypowych obiektów zapisano do dalszej analizy.



Rysunek 26: Obrazy wejściowe: *ptaki.jpg* i *ptaki2.jpg*.



Rysunek 27: Obrazy po usunięciu nietypowych obiektów.

5.2.2 Trenowanie sieci neuronowej

Po wstępnej obróbce, przygotowano dane wejściowe do sieci neuronowej, łącząc cechy obiektów z obu obrazów. Dane wejściowe zostały znormalizowane, a etykiety przypisano na podstawie kategorii: obiekty wewnętrzne oznaczono jako ptaki, a obiekty na krawędziach jako „nie-ptaki”.

Do trenowania wykorzystano funkcję `feedforwardnet`, umożliwiającą tworzenie sieci neuronowej z pełną łącznością między warstwami. Sieć została wytrenowana za pomocą metody wstecznej propagacji błędów (*backpropagation*). Proces treningu odbywał się na danych wejściowych (`uin`) i wyjściowych (`uout`), jak przedstawiono poniżej:

```
% Przekształcanie danych do sieci neuronowej
uin = [M(1:end-2,:); M2(1:end-2,:)]';
uout = [repmat([1;0], 1, n-2), repmat([0;1], 1, n2-2)];

% Trenowanie sieci neuronowej
nm = feedforwardnet;
nm = train(nm, uin, uout);
```

Listing 3: Przekształcanie danych do sieci neuronowej

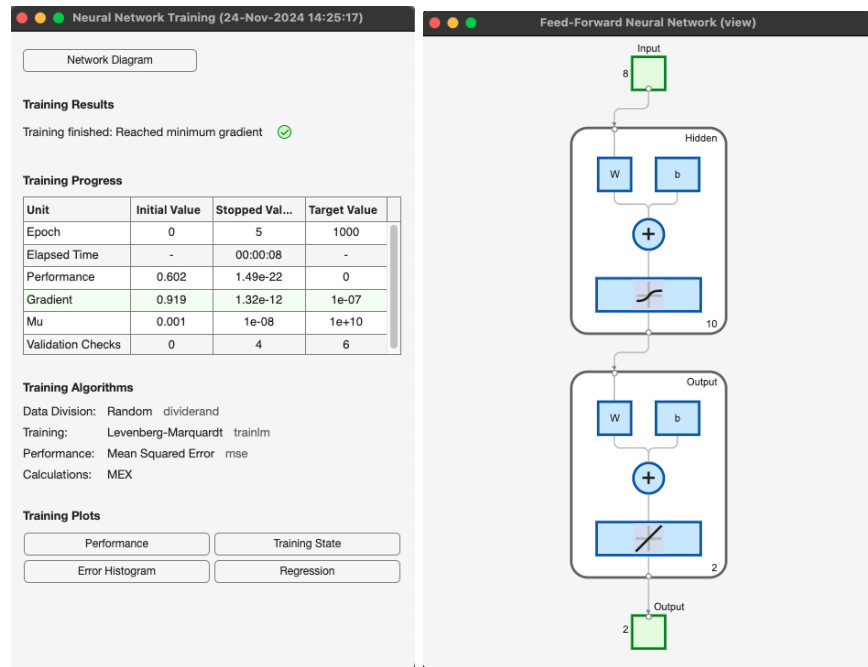
5.3 Testowanie sieci neuronowej

Sieć neuronowa została przetestowana na osobnym zestawie danych testowych (`tin`, `tout`), które nie były uwzględnione w procesie treningu. Wyniki klasyfikacji zaprezentowano w postaci wektorów, wskazujących prawdopodobieństwo przyporządkowania obiektów do danej kategorii.

Przykładowe wyniki testów:

$$\text{Output dla testu: } \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Oznacza to, że sieć poprawnie zaklasyfikowała obiekty jako ptaki (pierwsza klasa) i „nie-ptaki” (druga klasa).



Rysunek 28: Panel wyników sieci neuronowej i diagram w MATLAB.

5.4 Wnioski

Sieć neuronowa skutecznie rozpoznała obiekty na obrazach na podstawie cech obrazu. Wyniki testów pokazują, że nawet przy stosunkowo małej grupie trenin-gowej, klasyfikacja przebiega poprawnie, a sieć wykazuje zdolność do rozróżnia-nia obiektów między kategoriami.

Literatura

- [1] G. Gołaszewski, Obrazki do zajęć laboratoryjnych z przedmiotu - Analiza Obrazów: https://home.agh.edu.pl/~ggolasz/analiza_obrazow.html
- [2] M. Mamczur, Czym jest i jak się uczy sztuczna sieć neuronowa: <https://miroslawmamczur.pl/czym-jest-i-jak-sie-uczy-sztuczna-siec-neuronowa/>
- [3] Wikipedia, Rozpoznawanie obrazów: https://pl.wikipedia.org/wiki/Rozpoznawanie_obrazów