



AKADEMIA GÓRNICZO-HUTNICZA W KRAKOWIE

WYDZIAŁ FIZYKI I INFORMATYKI STOSOWANEJ

## Analiza Obrazów

Sprawozdanie nr 2 - Operacje morfologiczne, wyodrębnianie,  
właściwości i klasyfikacja obiektów

*Marcin Knapczyk*

12.12.2024

# 1 Operacje morfologiczne

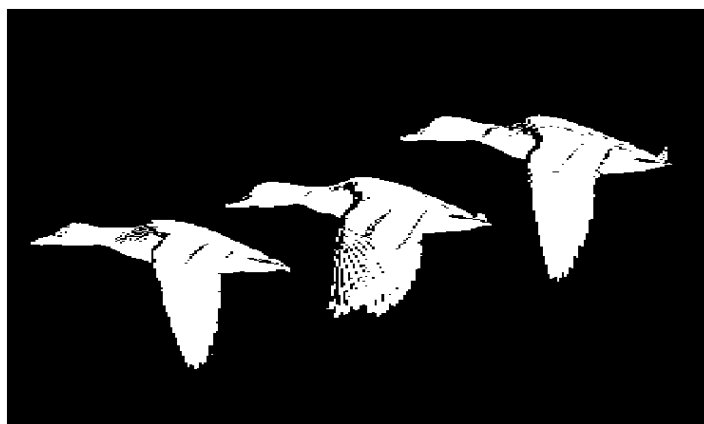
Operacje morfologiczne pozwalają na analizę obrazu pod względem kształtów/obiektów na nim się znajdujących.

Do zapoznania się z różnymi operacjami morfologicznymi posłuży obraz *kaczki*.



Rysunek 1: Oryginalny obraz *kaczki*

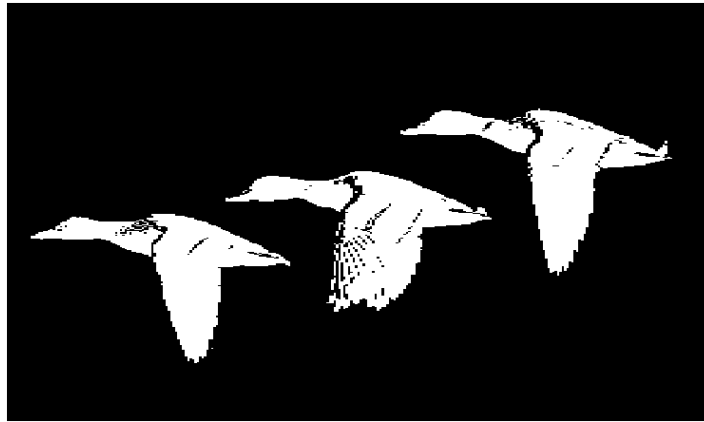
Do przeprowadzenia operacji morfologicznych w programie MATLAB służy funkcja *bwmorph*, do której jako argumenty przekazujemy zbinaryzowany obraz, typ operacji (nazwa) i ilość kroków operacji (*inf* działa tak długo, póki coś się zmienia). Do typów operacji należą *erode*, *dilate*, *open*, *close* (omówione w poprzednim sprawozdaniu), *fill*, *clean*, *remove*, *skel*, *endpoints*, *branchpoints*, *shrink*, *thin* i *thicken*.



Rysunek 2: Zbinaryzowany obraz *kaczki* (0.6)

## 1.1 Operacja fill

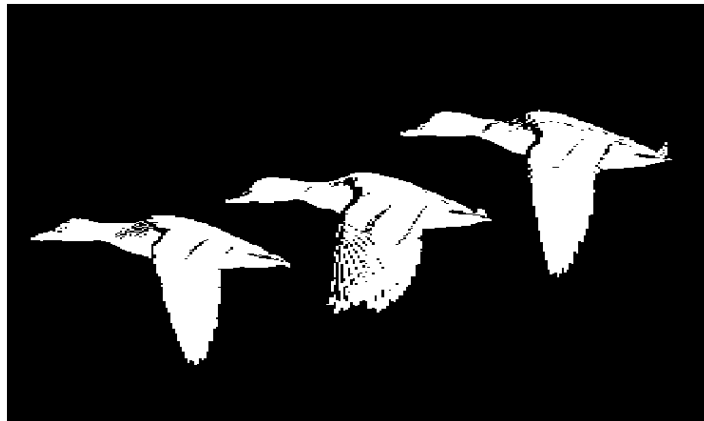
Operacja ta służy do wypełniania figury (np. czarne kropki w białej figurze).



Rysunek 3: Wypełniony zbinaryzowany obraz *kaczki*

## 1.2 Operacja clean

Operacja ta służy do usuwania szumu (np. białe kropki na czarnym tle)



Rysunek 4: Wyczyszczony zbinaryzowany obraz *kaczki*

Dla kolejnych operacji morfologicznych przygotowane zostały wyodrębnione kształty kaczek.



Rysunek 5: Zbinaryzowane kaczki po operacji domknięcia

### 1.3 Operacja remove

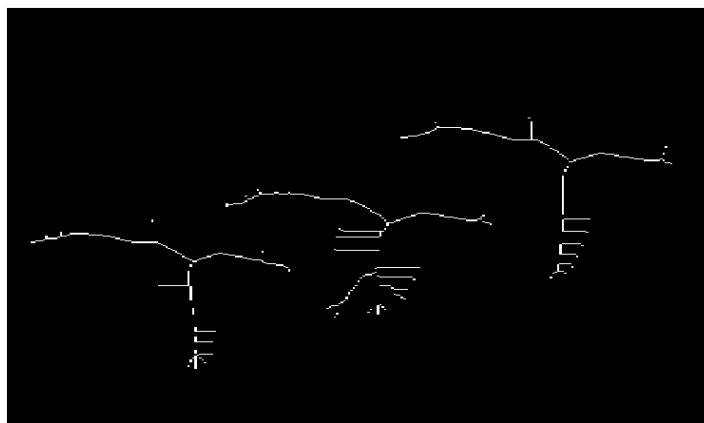
Operacja ta usuwa wnętrze obiektu.



Rysunek 6: Zbinaryzowane kaczki z usuniętym wnętrzem

### 1.4 Operacja skel

Służy do stworzenia tzw. szkieletu obiektów (zbiór punktów które są w tej samej odległości od krawędzi kształtu, po jednym pikselu na krok. Dla parametru inf otrzymamy zbiór linii).



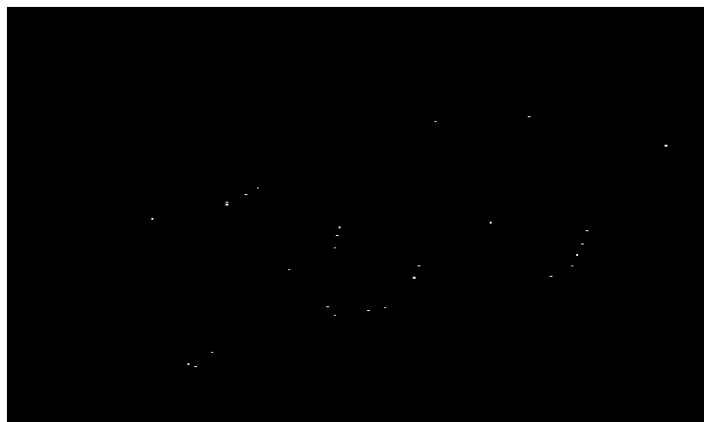
Rysunek 7: Szkielet obrazu *kaczki*



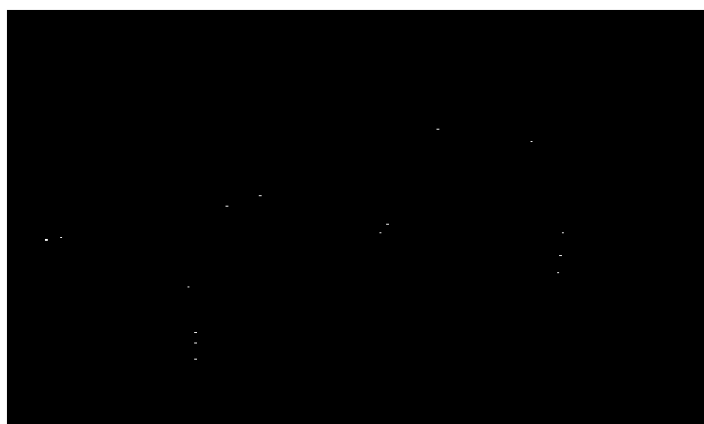
Rysunek 8: Szkielet obrazu *kaczki* nałożony na oryginalny obraz

Jak widać, szkielet całkiem dobrze opisuje oryginalne kształty kaczek.

Dla szkieletu można wyznaczyć tzw. *endpointy* (punkty końcowe linii) i *branchpointy* (punkty rozgałęzień).



Rysunek 9: Punkty końcowe szkieletów kaczek



Rysunek 10: Punkty rozgałęzień szkieletów kaczek

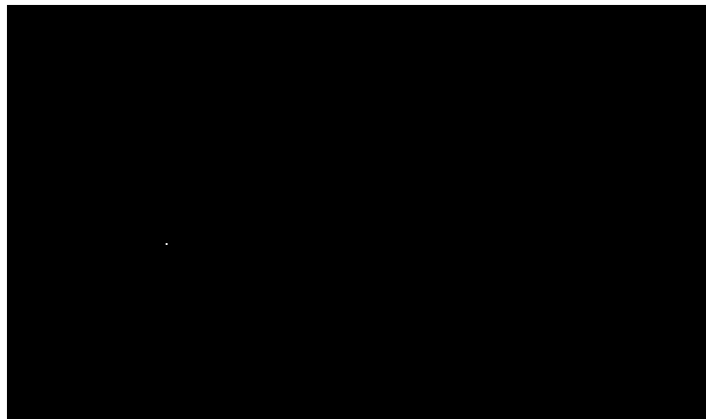


Rysunek 11: Punkty rozgałęzień nałożone na oryginalny obraz

Można zauważyć, że zbiór pojedynczych punktów charakterystycznych szkieletu przechowuje o nim dużą ilość informacji (można na ich podstawie próbować odtworzyć szkielet).

## 1.5 Operacja shrink

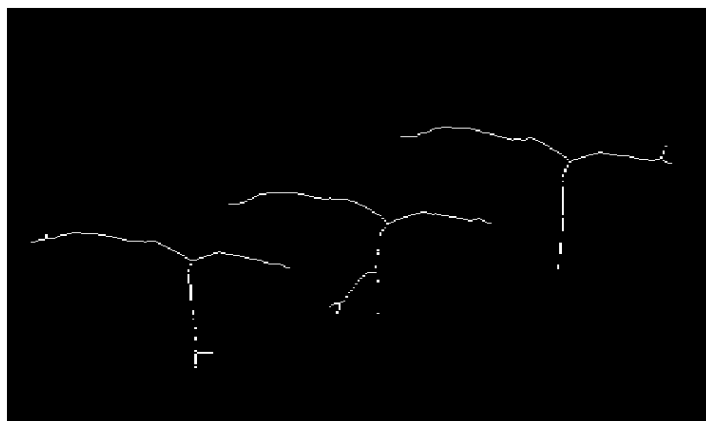
Operacja ta zmniejsza kształty. Dla nieskończonej ilości kroków (inf) dostaniemy 3 punkty dla 3 kaczek.



Rysunek 12: Efekt zmniejszenia kaczek (inf) - pozostały trzy kropki

## 1.6 Operacja thin

Operacja *thin* "wyszczupla" kształty. Dla nieskończonej ilości kroków (inf) dostaniemy linie o grubości 1.



Rysunek 13: Efekt wyszczuplania kaczek (inf)

## 1.7 Operacja thicken

Operacja ta "pogrubia" kształty, przy czym zachowana zostaje kręwdź między obiektami (w przeciwieństwie do operacji *dilate*). Dla liczby kroków inf dostajemy obraz podzielony na segmenty. W każdym segmencie znajduje się tylko jeden obiekt.



Rysunek 14: Efekt operacji *thicken*. Po lewej 30 kroków, po prawej inf kroków

Za pomocą operacji *thicken* z nieskończoną ilością kroków możemy dokonać segmentacji (segmentacja przez pogrubianie).

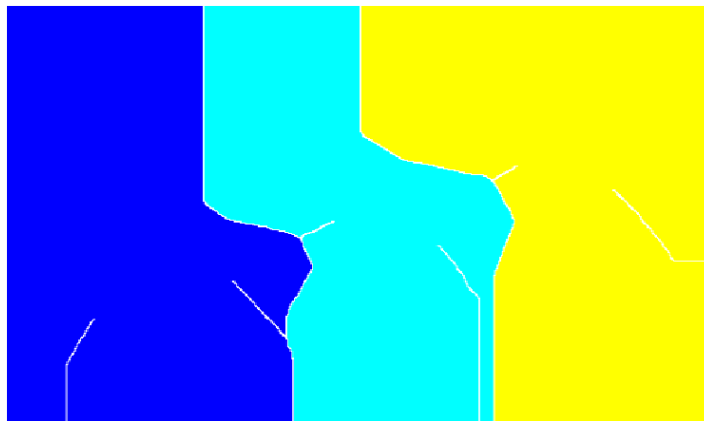
## 2 Wyodrębnianie obiektów

Do wyodrębnienia obiektów możemy posłużyć się funkcją MATLABa *bwlabel*, do której przekazujemy zbinaryzowany obraz. Każdy z pikseli (w macierzy) oddzielonych kształtów otrzyma inną wartość/label (1, 2, 3, ...).

Za pomocą funkcji *label2rgb*, do której przekazujemy macierz będącą wynikiem "labelowania", możemy uzyskać wizualne rozróżnienie obiektów (każdy otrzyma inny kolor).



Rysunek 15: Rozróżnione kaczki, dla każdego obiektu inny kolor



Rysunek 16: Rozróżnione segmenty (segmentacja przez pogrubianie), dla każdego segmentu inny kolor

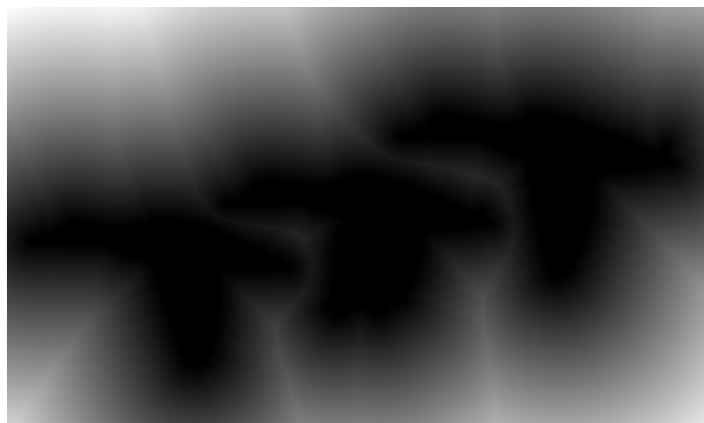
Dzięki oznaczeniom możemy wybrać konkretny obiekt.



Rysunek 17: Wyświetlenie wybranej kaczki (nr 2)

Można też obliczyć ilość obiektów na obrazie za pomocą polecenia  $m = \max(l(:))$ ;

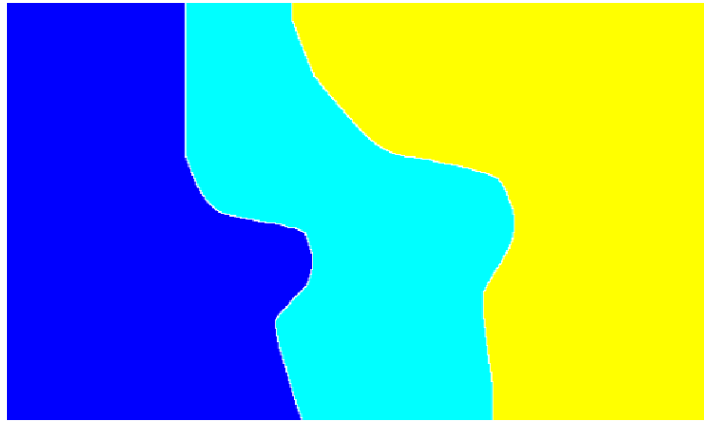
Oprócz pogrubienia, segmentacji można dokonać stosując transformatę odległościową, gdzie każdemu pikselowi przypisujemy odległość od najbliższego białego piksela (dla białego  $odl=0$ ). Służy do tego funkcja *bwdist*, do której przekazujemy zbinaryzowany obraz oraz nazwę metryki, która liczymy odległość (*euclidean*, *manhattan* (*cityblock*), *chessboard*, *quasi – euclidean*).



Rysunek 18: Wynik transformacji odległościowej dla metryki *euclidean* (kaczki czarne a tło tym jaśniejsze im dalej od kaczki)

Mając wynik transformacji odległościowej można dokonać tzw. segmentacji wododziałowej (*watershed*). Służy do tego funkcja *watershed*.





Rysunek 19: Segmentacja wododziałowa z oznaczeniem

W porównaniu do segmentacji przez pogrubienie, segmentacja wododziałowa wydaje się bardziej dokładna.

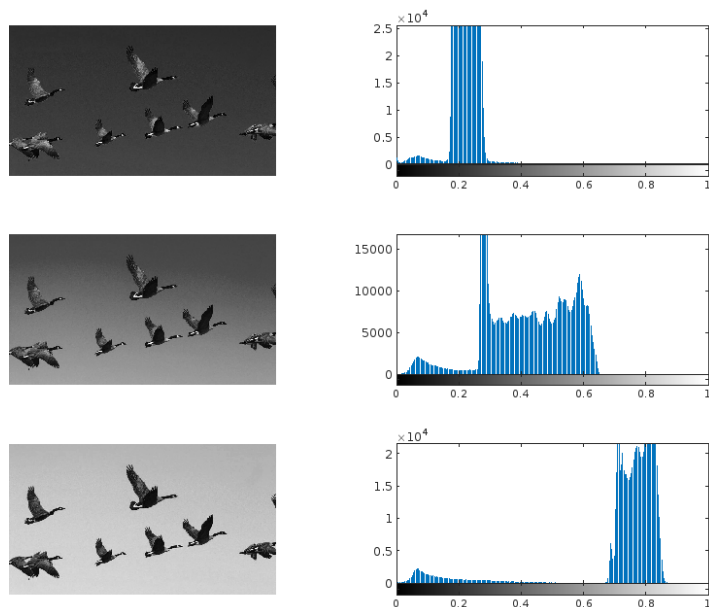
### 3 Właściwości obiektów

Do badania właściwości obiektów posłużmy przykład obrazu *ptaki*.



Rysunek 20: Oryginalny obraz *ptaki*

Binaryzacja obrazu sprawia problem przez gradient występujący w tle (niebo). Badając histogram obrazu widzimy, że gradient zawarty jest w kanale koloru zielonego.



Rysunek 21: Histogramy poszczególnych kanałów obrazu *ptaki*

Można więc zbinaryzować osobno warstwę czerwoną i niebieską i potem połączyć je logicznym operatorem *lub*.



Rysunek 22: Pomyślna binaryzacja obrazu *ptaki*, po domknięciu i otwarciu

Dalej można oznaczyć poszczególne obiekty.



Rysunek 23: Oznaczone obiekty w różnych kolorach

Szczegółowe informacje o danym obiekcie można uzyskać za pomocą polecenia *regionprops*, do którego przekazujemy macierz zawierającą obiekt. W wyniku otrzymujemy strukturę danych zawierającą informacje:

- Area - pole powierzchni (ile pikseli)
- Centroid - środek masy (fizyczny)
- Bounding Box - najmniejszy prostokąt zamykający figurę równoległy do osi układu
- Majoraxis - oś główna - największa rozpiętość obiektu
- Minoraxis - oś pod kątem prostym do osi głównej. Z obu osi można określić prostokąt okalający nierównoległy do osi
- Eccentricity - jak bardzo przesunięty jest środek masy do środka ośrodka (0-1)
- Orientation - kąt najmniejszej elipsy opisanej na figurze
- Circularity - jak bardzo figura przypomina koło
- EulerNumber – liczba Eulera (można na jej podstawie określić liczbę dziur w obiekcie)
- Image - obraz z Bounding Boxa
- FilledImage - obraz z Bounding Boxa z wypełnionymi dziurami
- Ekstrema - współrzędne punktów ekstremalnych (lewo, prawo, góra, dół)
- Solidity - stosunek pola Bounding Boxa do pola obiektu
- Perimeter (old) - obwód (stary sposób liczenia)

Poza informacjami z *regionprops* istnieją inne wartości (współczynniki), dzięki którym możemy porównywać obiekty. Algorytmy obliczające te wartości nie są wbudowane w MATLAB i należy je obliczyć za pomocą dołączonych skryptów. Do tych współczynników należą:

- wsp. BlairBliss - średnia odległości punktów od środka masy (dla koła mała)
- wsp. CircularityL - współczynnik cyrkularności
- wsp. CircularityS - współczynnik cyrkularności
- wsp. Danielsson'a - przeciętna odległość pikseli na krawędzi od środka masy
- wsp. Feret'a - stosunek wymiarów Bounding Boxa - mało przydatne
- wsp. Haralick'a - średnia odległość punktów od krawędzi (niski przy figurach porowatych)
- wsp. Malinowskiej - stosunek promienia koła o tym samym obwodzie, co figura do promienia koła o tym samym polu, co figura
- wsp. Shape

	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 24: Macierz z wyliczonymi współczynnikami dla kolejnych ptaków (kolejne wiersze). Widać, że ósmy ptak się wyróżnia

Każdy z tych współczynników pozwala na opisanie kształtu za pomocą jednej wartości. Wyliczając średnie współczynników dla wszystkich obiektów, możemy obliczyć "średni obiekt". Licząc różnicę między obiektem a średnią w wielokrotnościach odchylenia standardowego danego parametru, możemy znaleźć obiekty nietypowe.

Zakładając rozkład normalny, w odległości 1 odchylenia znajduje się 67.7%, 2 odchyleni - 95%, 3 odchyleni - 99.7% obiektów.

Aby zwiększyć poprawność odnajdywania obiektów nietypowych (outlierów), można wyłączyć po kolei każdy z obiektów z obliczania średniej i sprawdzić, czy jest outlierem (nie zaburza on wtedy statystyk). Podejście to nazywa się "leave one out".

## 4 Klasyfikacja obiektów z użyciem sieci neuronowej

Obliczone współczynniki można wykorzystać do analizowania kształtów za pomocą sieci neuronowej. Jedną z prostszych sieci jest sieć typu "feed forward". Można ją utworzyć za pomocą polecenia `nn = feedforwardnet;`.

Do uczenia i testowania sieci użyte zostaną obrazy *ptaki* i *ptaki2*.



Rysunek 25: Oryginalny obraz *ptaki2*



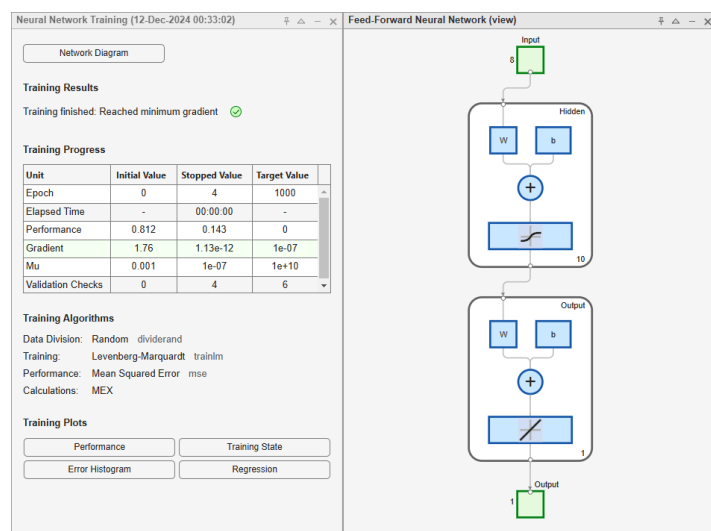
Rysunek 26: Zbinaryzowane *ptaki2*



Rysunek 27: Oznaczone *ptaki2*

Dla każdego z obiektów liczymy współczynniki (z poprzedniej sekcji).

Do trenowania sieci potrzebne są dane wejściowe i wyjściowe. Podajemy je do sieci w postaci wektora transponowanego, wykonując polecenie  $nn = \text{train}(nn, iu, ou);$ . Wejściem są obliczone współczynniki dla ptaków z obrazów *ptaki* i *ptaki2* (zostawiamy po jednym do testów). Na wyjście podajemy wartości 1 dla ptaków z pierwszego obrazu, a 0 dla ptaków z drugiego obrazu.



Rysunek 28: Wyniki trenowania sieci neuronowej i diagram architektury

Działanie sieci sprawdzamy wykorzystując dane testowe (wydzielone ze zbioru danych), np. dla pierwszego z obiektów:  $nn(M(1, :)')$ .

Wyniki trenowania są średnie (25% skuteczności), w dużej mierze przez małą liczbę danych wykorzystanych do procesu uczenia.