

Sieci Neuronowe	
Kierunek <i>Informatyczne Systemy Automatyki</i>	Termin <i>Czwartek 17<sup>05</sup> – 18<sup>45</sup></i>
Imię, nazwisko, numer albumu <i>Piotr Brajer 272538 Jakub Golec 272553</i>	Data <i>09.12.2024</i>
Temat Porównanie sieci splotowej CNN oraz wielowarstwowej MLP , klasyfikacja zdjęć ryb	



## Sprawozdanie

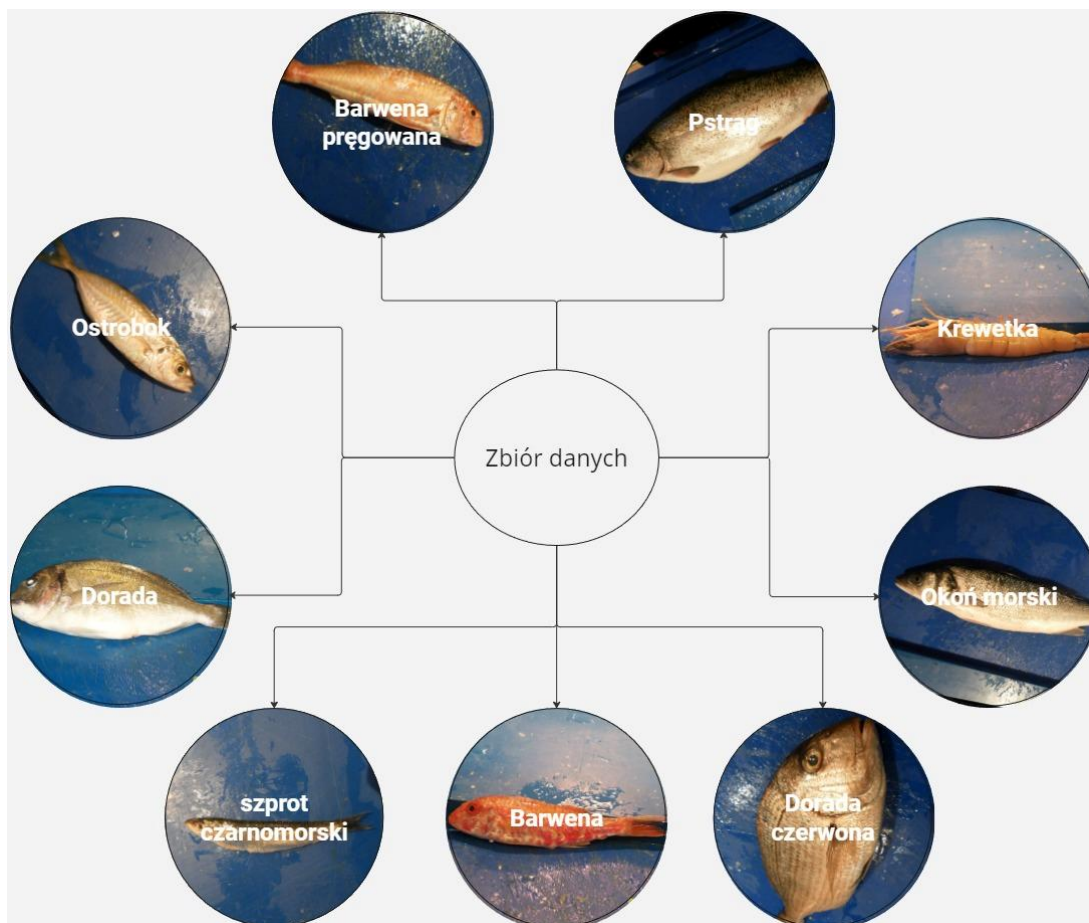
[GitHub](#)

### 1 Cel projektu i zadanie sieci neuronowej

Projekt wykonywany jest w celu nauczania się podstawowej i zaawansowanej wiedzy związanej z projektowaniem, implementowaniem oraz badaniem sieci neuronowych.

W tym sprawozdaniu będziemy rozważać wykorzystywanie sieci neuronowej do klasyfikacji obiektów znajdujących się na zdjęciach, w związku z czym zadanie sieci generowanej dla tego projektu polega na rozróżnianiu gatunków ryb za pomocą zdjęć.

### 2 Zbiór danych



## Właściwości Zbioru:

- Zbiór podzielony jest na 8 gatunków ryb i 1 skorupiak
- Każdy gatunek posiada 1000 zdjęć
- Format zdjęć to jpg
- Rozmiar zdjęć to 256x256px
- Przed rozpoczęciem nauki sieci kolejność zdjęć jest mieszana
- Na każdym zdjęciu wykonywany jest przypadkowy obrót w okresie 0-30°
- Na każdym zdjęciu wykonywany jest przypadkowy obrót pionowy
- Wszystkie zdjęcia przechodzą normalizację na podstawie zmiennych wcześniej wyliczonych przy użyciu odrębnego skryptu

## 3 Projektowanie sieci neuronowych

W celu odnalezienia najlepszego modelu dla naszego zadania wykonane będą 2 różne rodzaje sieci neuronowych.

Każdy z modeli wykonany będzie w języku Python za pomocą biblioteki pyTorch.

**Pierwsza sieć neuronowa** będzie typu wielowarstwowego perceptronu (MLP, Multilayer Perceptron)

### Struktura sieci:

1. Typ sieci:
  - Sieć w pełni połączona zbudowana z:
    - 3 ukrytych warstw
    - Aktywacji ReLu po każdej warstwie liniowej , co nadaje nieliniowości
    - Warstwy wyjściowej przystosowanej do liczby klas w zbiorze
2. Wejście sieci:
  - Wejście to obrazy o wymiarach 256x256x3, które są "spłaszczane" do jednowymiarowego wektora o długości  $256 \times 256 \times 3 = 196\,608$
3. Liczba neuronów w ukrytych warstwach:
  - Wszystkie ukryte warstwy mają tą samą ilość neuronów (w zależności od modelu liczba może być w zasięgu od 64-1024)
4. Wyjście sieci:
  - Liczba neuronów w warstwie wyjściowej odpowiada liczbie klas w zbiorze danych, a aktywacja wyjściowa jest przystosowana do funkcji kosztu

### Charakterystyka procesu treningowego:

- Model jest trenowany na GPU (jeśli dostępne), co znacznie przyspiesza obliczenia
- Optymalizator: Adam z domyślnym współczynnikiem uczenia 0.001
- Harmonogram uczenia: Redukcja współczynnika uczenia przy stagnacji poprawy (opartej na średniej stracie testowej)
- Funkcja kosztu: CrossEntropyLoss, odpowiednia dla problemów klasyfikacji wieloklasowej

**Druga sieć neuronowa** to konwolucyjna sieć neuronowa (CNN, Convolutional Neural Network)

#### **Struktura sieci:**

1. Typ sieci:
  - Sieć wielowarstwowa konwolucyjna zbudowana z:
    - Sekcji warstw konwolucyjnych i poolingowych
    - Sekcji w pełni połączonych warstw
2. Część konwolucyjna:
  - Używa sekwencji warstw Conv2d (konwolucja) i MaxPool2d (próbkiowanie maksymalne) do redukcji wymiarów przestrzennych i wyodrębnienia cech
  - W zależności od modelu od 1-3 sekcji
3. Część w pełni połączona:
  - Dane po wyjściu z warstw konwolucyjnych są spłaszczane
  - Przechodzą przez od 1 do 3 w pełni połączonych warstw (Linear) z funkcjami aktywacji ReLU każda o tej samej ilości neuronów w zależności od modelu (32-1024)
4. Wejście i wyjście sieci:
  - Wejście: Obraz 256x256x3 (wysokość, szerokość, kanały)
  - Wyjście: Wyniki klasyfikacji dla każdej z klas
5. Liczba neuronów w ukrytych warstwach:
  - Wszystkie ukryte warstwy mają tę samą ilość neuronów (w zależności od modelu liczba może być w zasięgu od 32-1024)
6. Wymiarowanie danych:
  - Wymiary danych po przejściu przez warstwy konwolucyjne:
    - Po ostatniej warstwie: 1024x4x4, gdzie 4x4 to rozmiar przestrzenny po operacjach poolingowych
    - Spłaszczone dane:  $1024 \times 4 \times 4 = 16\,384$  wejść do pierwszej w pełni połączonej warstwy

#### **Charakterystyka procesu treningowego:**

- Model jest trenowany na GPU
- Optymalizator: Adam z domyślnym współczynnikiem uczenia 0.001
- Harmonogram uczenia: Redukcja współczynnika uczenia, gdy strata walidacyjna przestaje się poprawiać (ReduceLROnPlateau)
- Funkcja kosztu: CrossEntropyLoss, odpowiednia dla problemów klasyfikacji wieloklasowej
- Zatrzymanie wczesne: Proces uczenia jest przerywany, jeśli strata walidacyjna osiągnie próg 0.15, (95% celności  $\pm 1\%$ )

## 4 Badanie sieci neuronowych

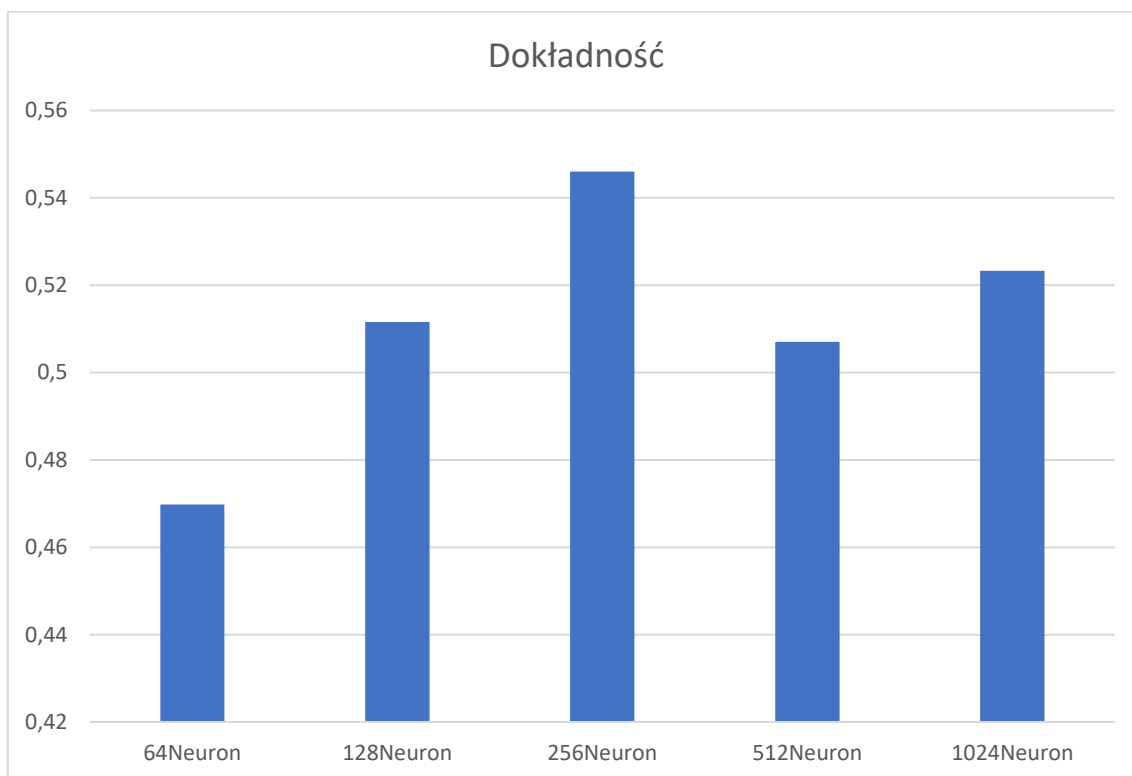
Każdy model badany będzie dla:

- Dokładności - Udział poprawnych przewidywań danej klasy w stosunku do wszystkich przykładów, które zostały przypisane do tej klasy
  - Wzór:  $Dokładność = \frac{\text{Liczba poprawnych przewidywań}}{\text{liczba wszystkich przykładów}}$
  - Dobrze oddaje efektywność modelu, gdy klasy są równomiernie zbalansowane,
  - Może być myląca w przypadku niezbalansowanych zbiorów danych, gdy większość przykładów pochodzi z jednej klasy
- Precyzji - Udział poprawnych przewidywań danej klasy w stosunku do wszystkich rzeczywistych przykładów tej klasy
  - Wzór:  $Precyzyjność = \frac{\text{Liczba prawdziwie pozytywnych (TP)}}{\text{Liczba wszystkich przewidywanych pozytywnych (TP+FP)}}$
  - Ważna w przypadkach, gdzie fałszywe alarmy (false positives) są kosztowne
- Czułości, pełności - Udział poprawnych przewidywań danej klasy w stosunku do wszystkich rzeczywistych przykładów tej klasy
  - Wzór:  $Precyzyjność = \frac{\text{Liczba prawdziwie pozytywnych (TP)}}{\text{Liczba wszystkich rzeczywistych pozytywnych (TP+FN)}}$
  - Ważna w przypadkach, gdzie pominięcie prawdziwego przypadku (false negative) jest kosztowne
- Miara F1 - Harmoniczna średnia precyzji i czułości
  - Wzór:  $F1 = \frac{Precyzja * Czułość}{Precyzja + Czułość}$
  - Miara F1 równoważy precyzję i czułość, szczególnie w przypadkach, gdy mamy niezbalansowane dane
  - Im bliższa 1, tym lepszy model w sensie ogólnej równowagi między precyzją a czułością
- Macierz konfuzji - Tabela pokazująca liczbę prawidłowych i błędnych klasyfikacji dla każdej klasy
  - Umożliwia szczegółową analizę błędów klasyfikacji:
    - TP (True Positive): Prawdziwie pozytywne - poprawnie zaklasyfikowane jako dana klasa.
    - FP (False Positive): Fałszywie pozytywne - błędnie przypisane do danej klasy.
    - FN (False Negative): Fałszywie negatywne - błędnie wykluczone z danej klasy.
    - TN (True Negative): Prawdziwie negatywne - poprawnie wykluczone z danej klasy.

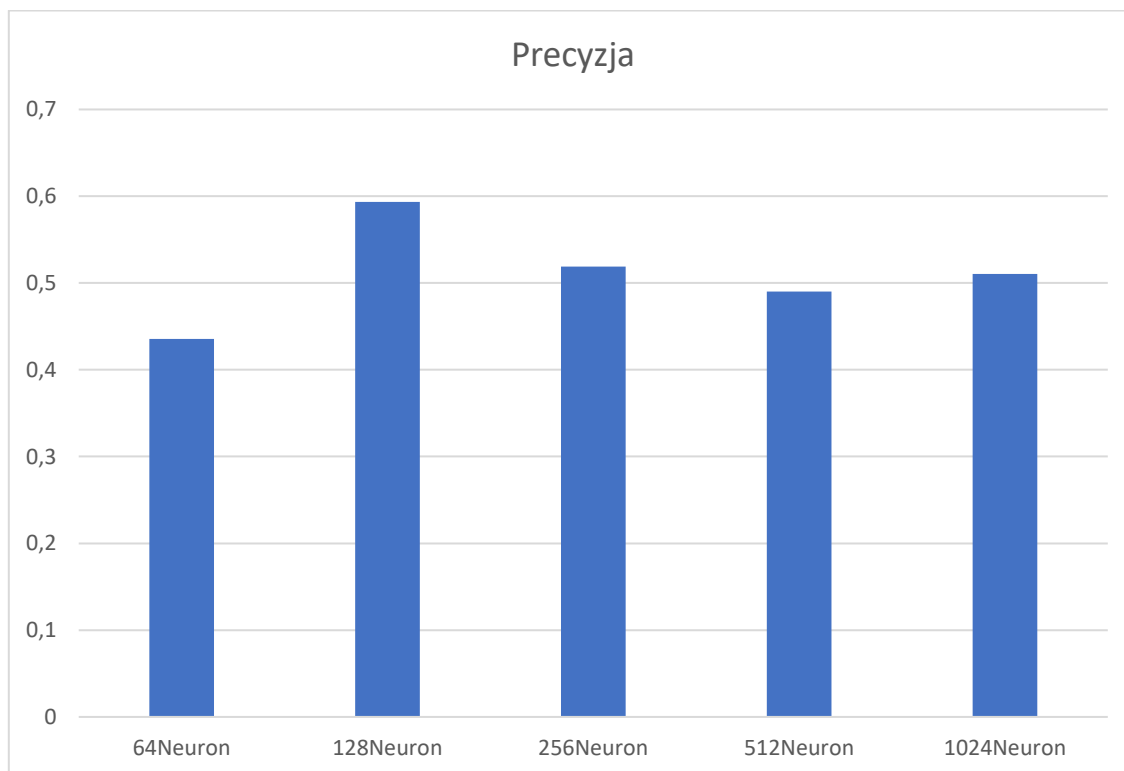
## Sieć neuronowa MLP

Każdy model nazwany będzie według ilości neuronów dla jednej sekcji w pełni połączonych warstw.

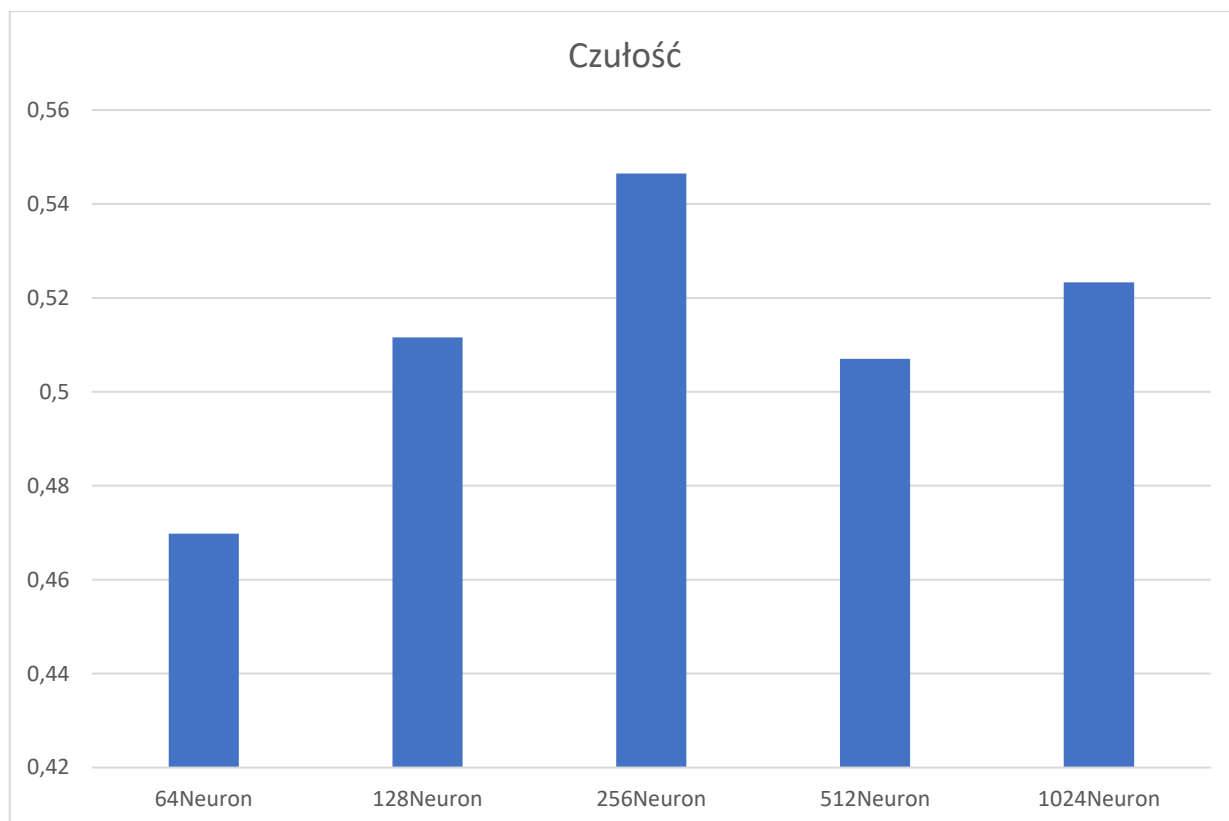
### 1. Dokładność:



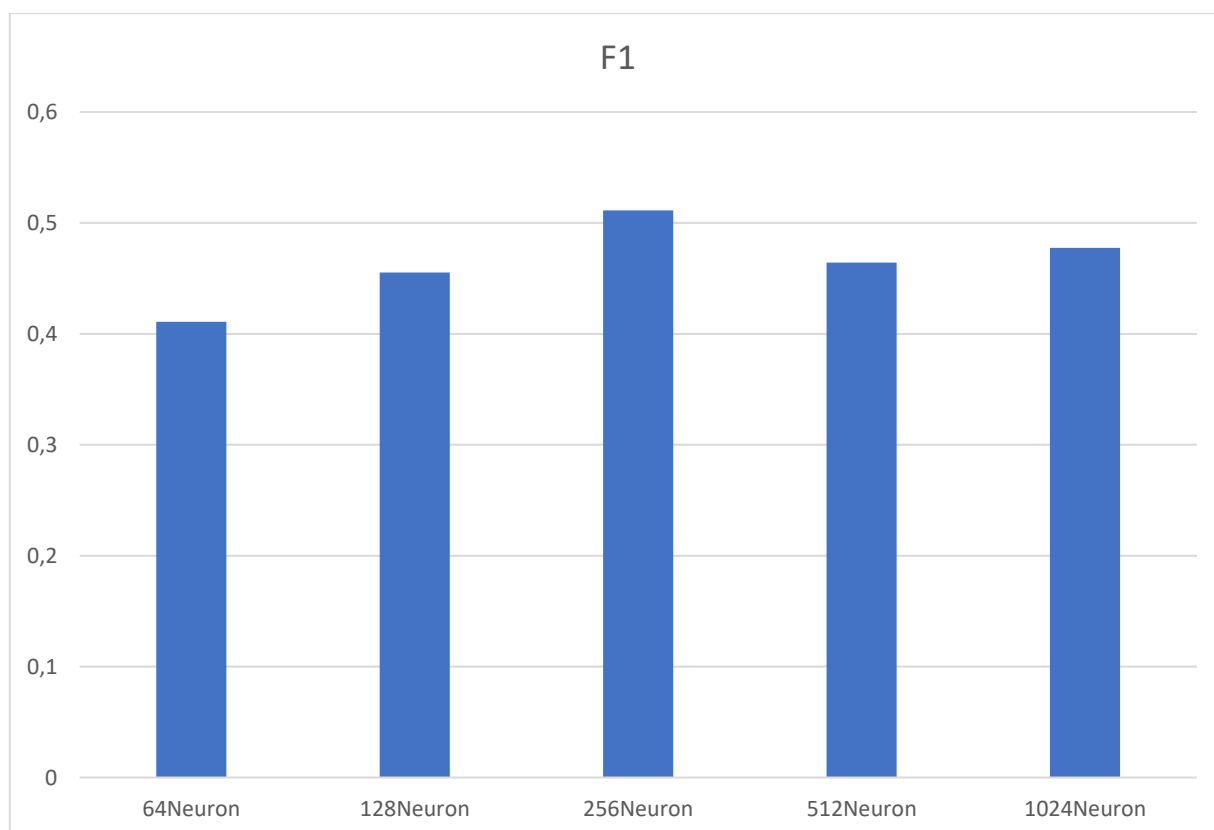
### 2. Precyzja:



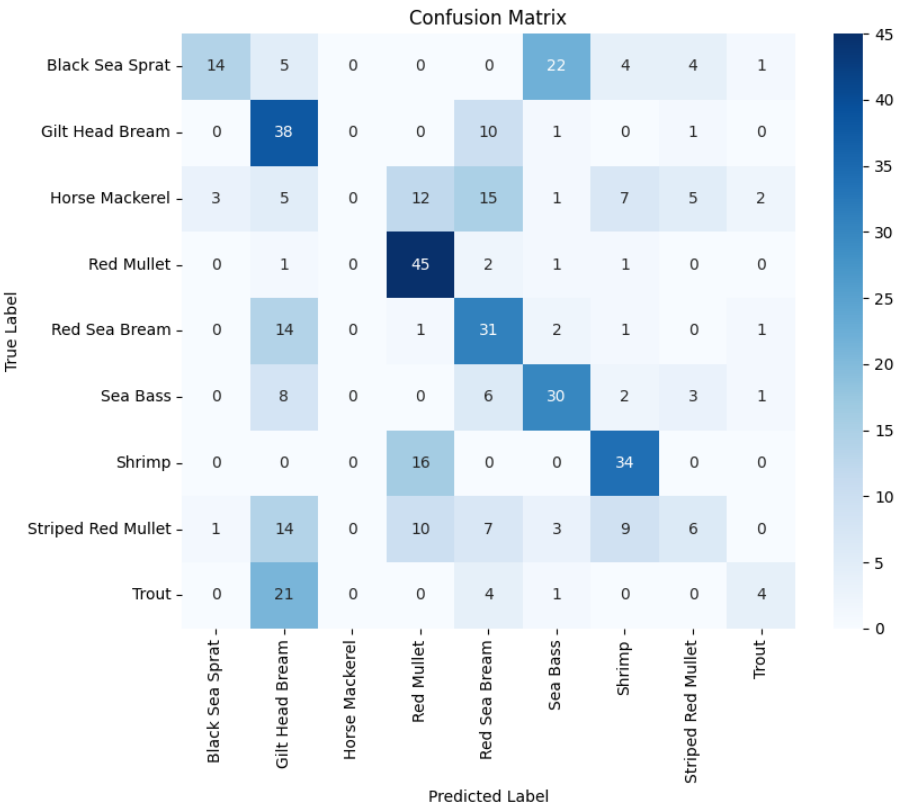
### 3. Czułość:



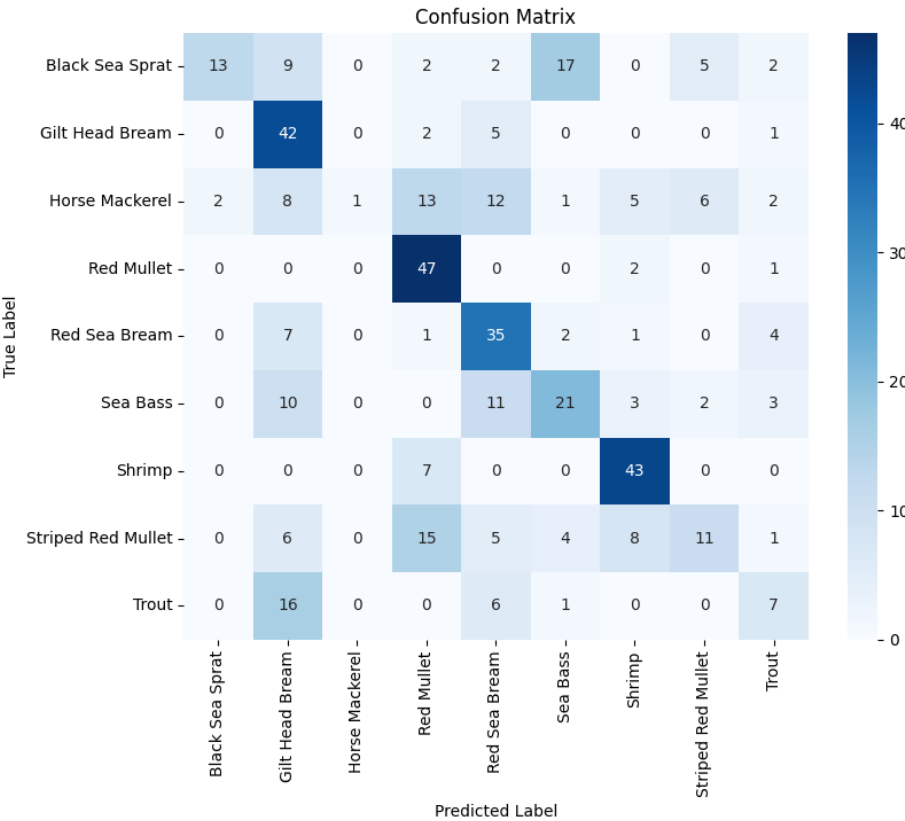
### 4. F1:



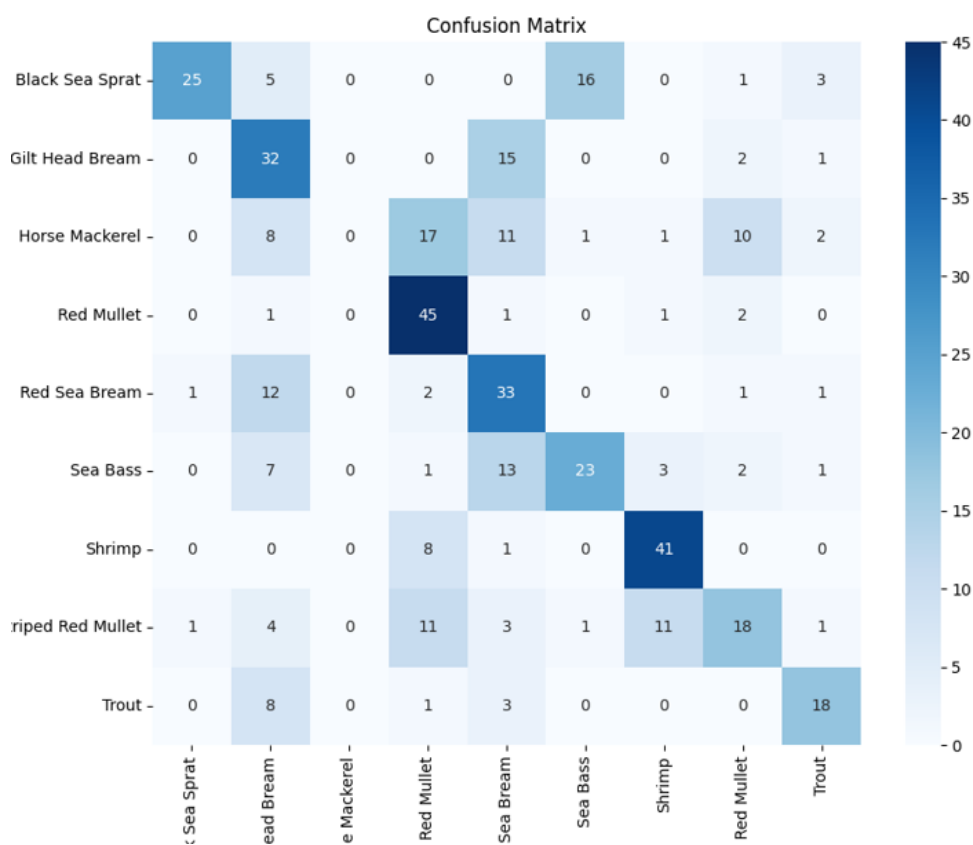
5. Macierz konfuzji:



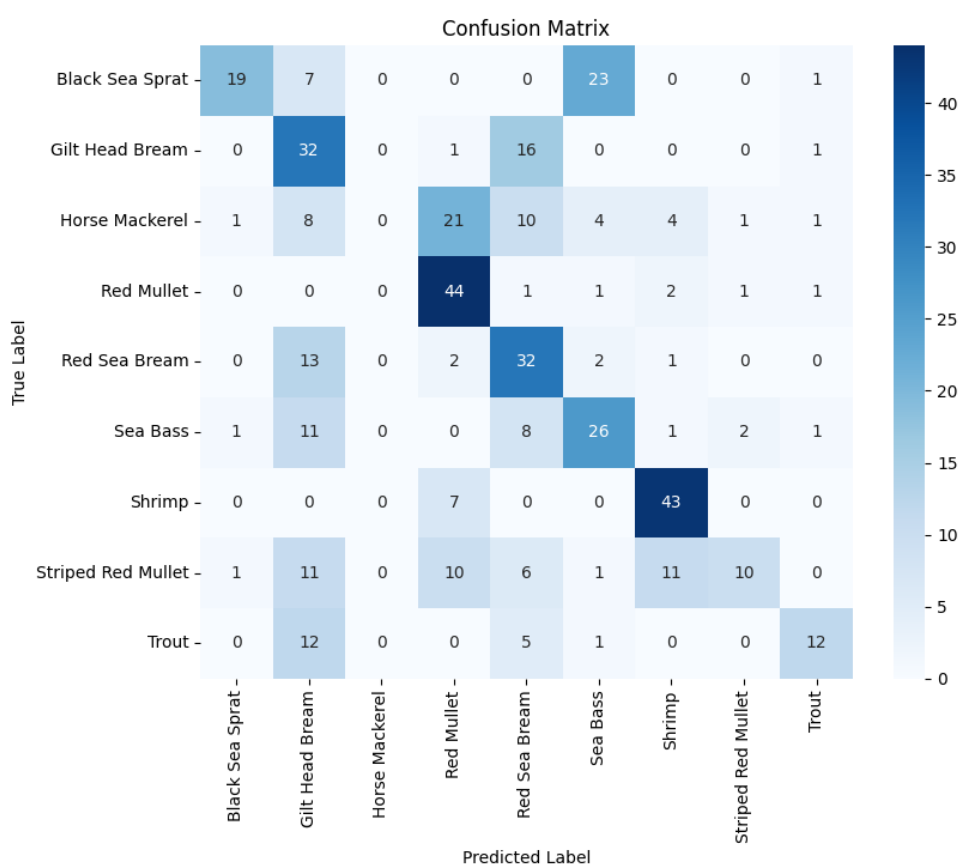
64Neuron



128Neuron

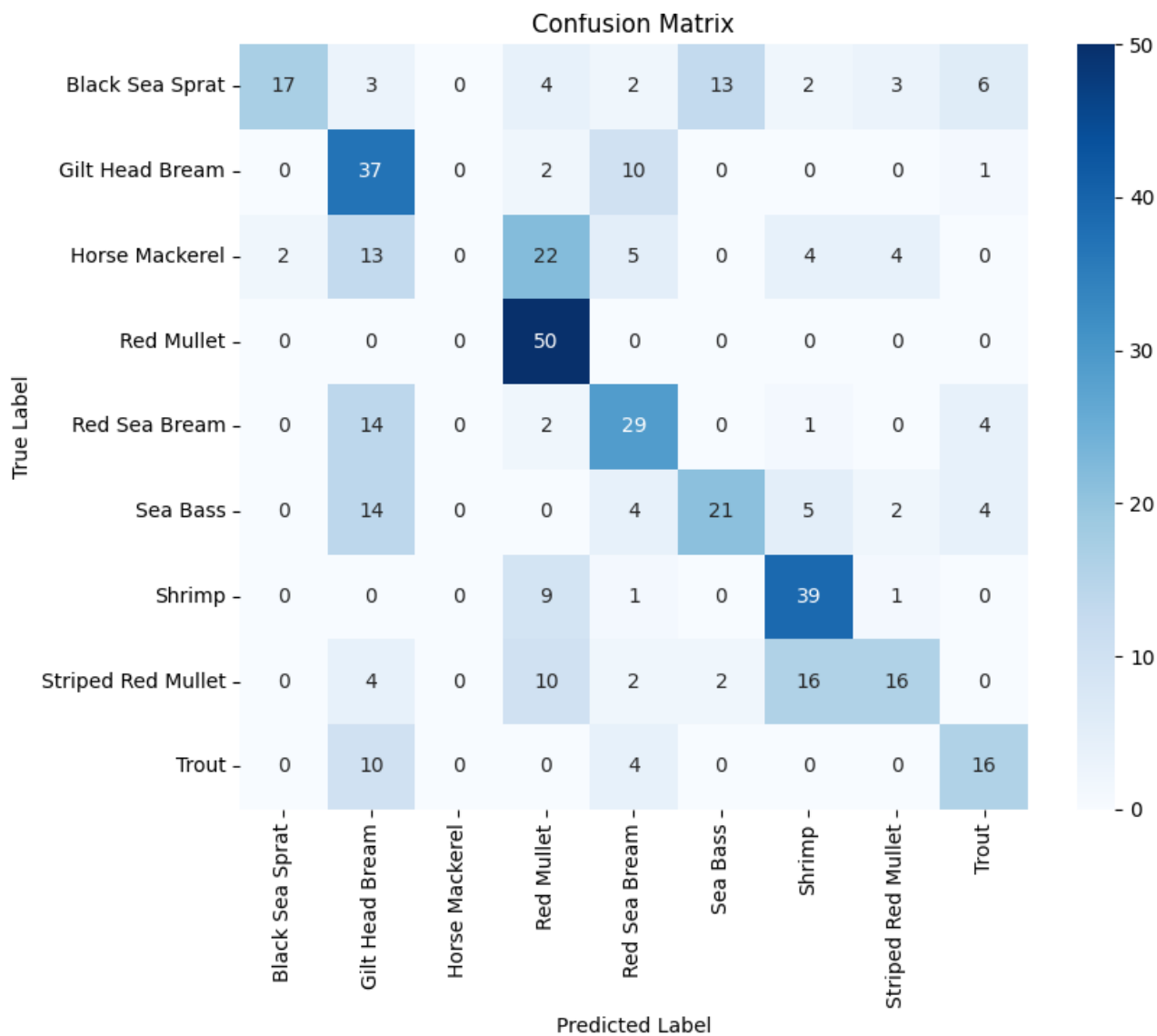


256Neuron



512Neuron



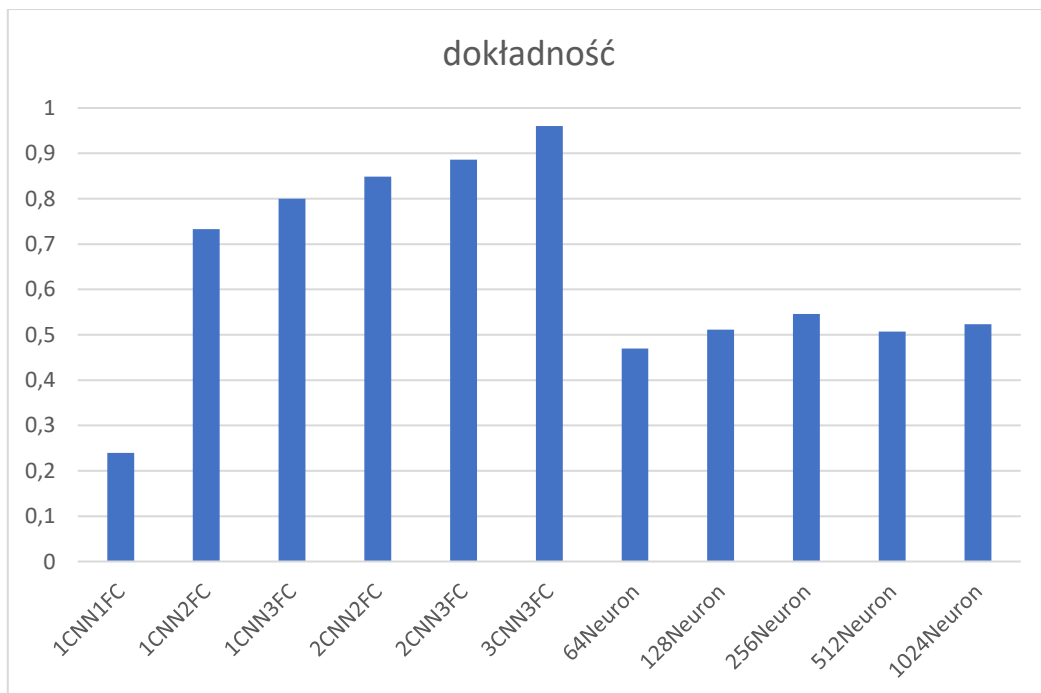
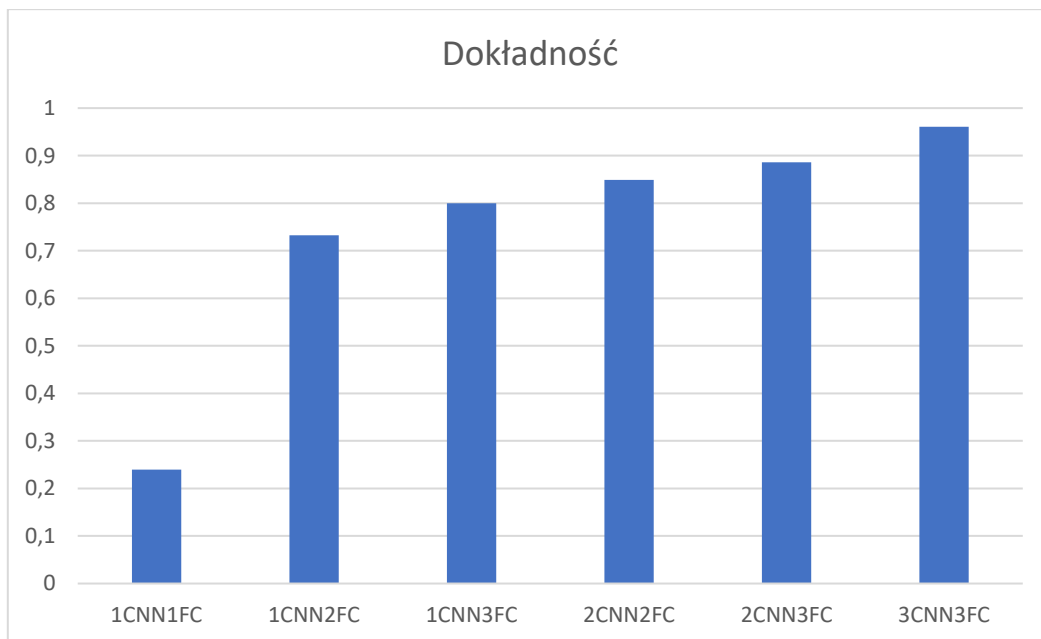


1024Neuron

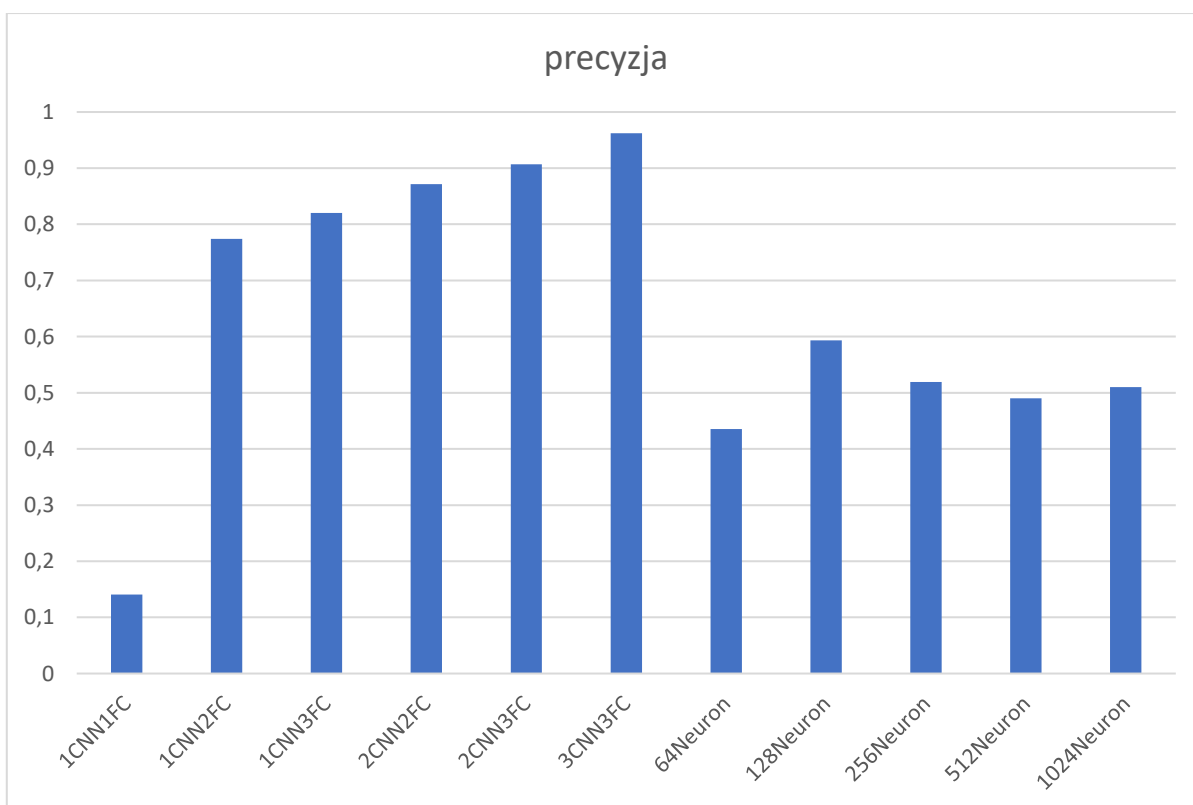
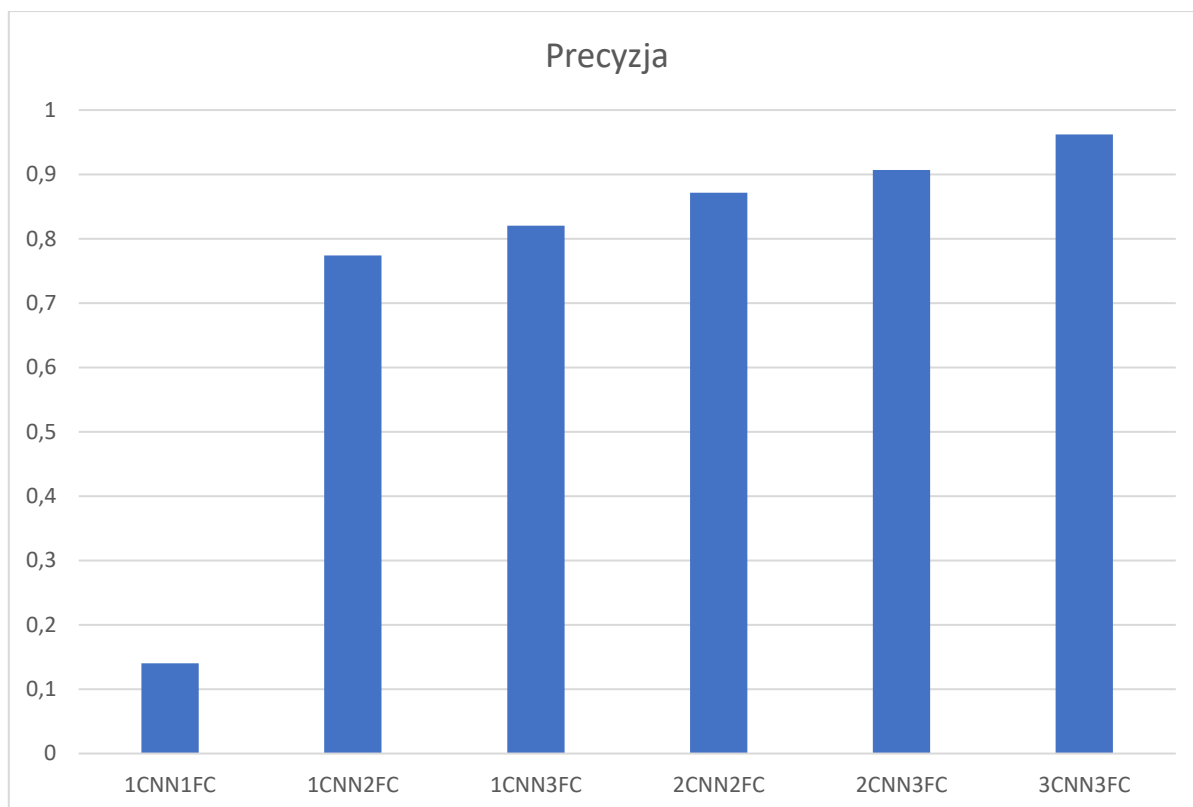
## Sieć neuronowa CNN

Każdy model nazywany będzie według ilości sekcji warstw konwolucji i poolingu oraz sekcji w pełni połączonych warstw np.: 2CNN3FC – 2 sekcje warstw konwolucji oraz 3 sekcje w pełni połączonych warstw.

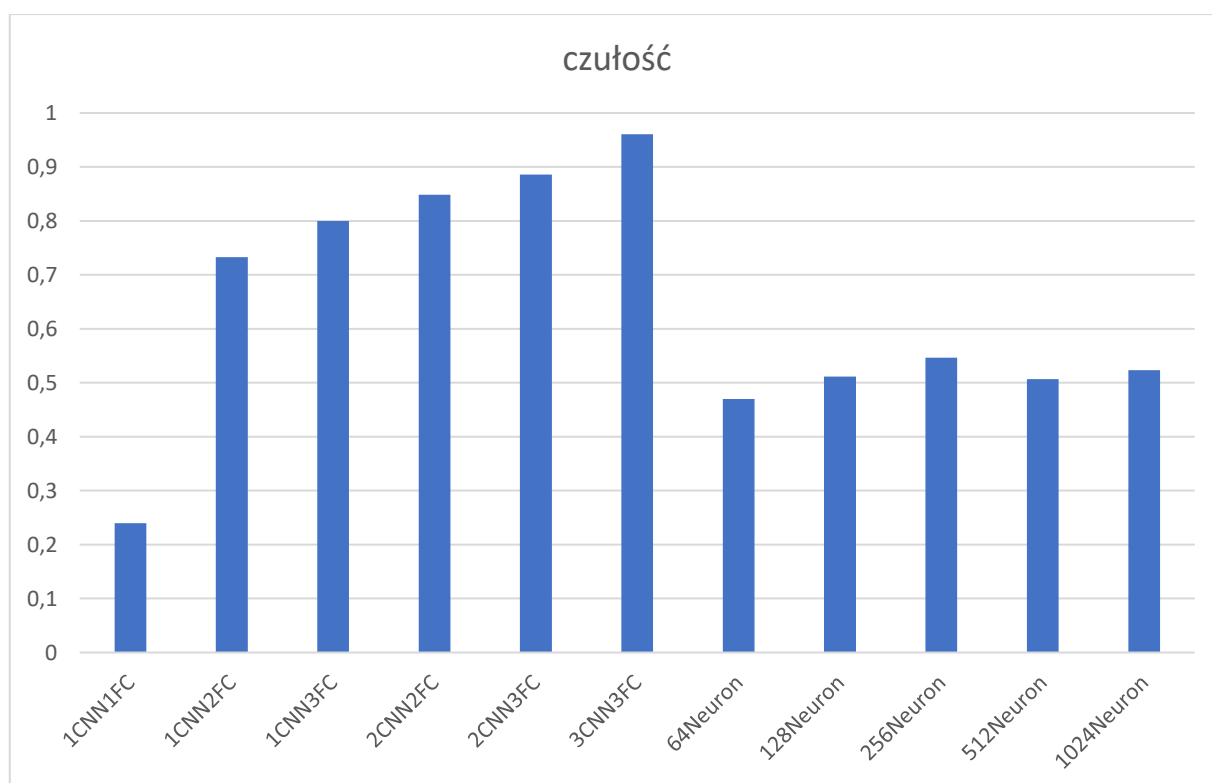
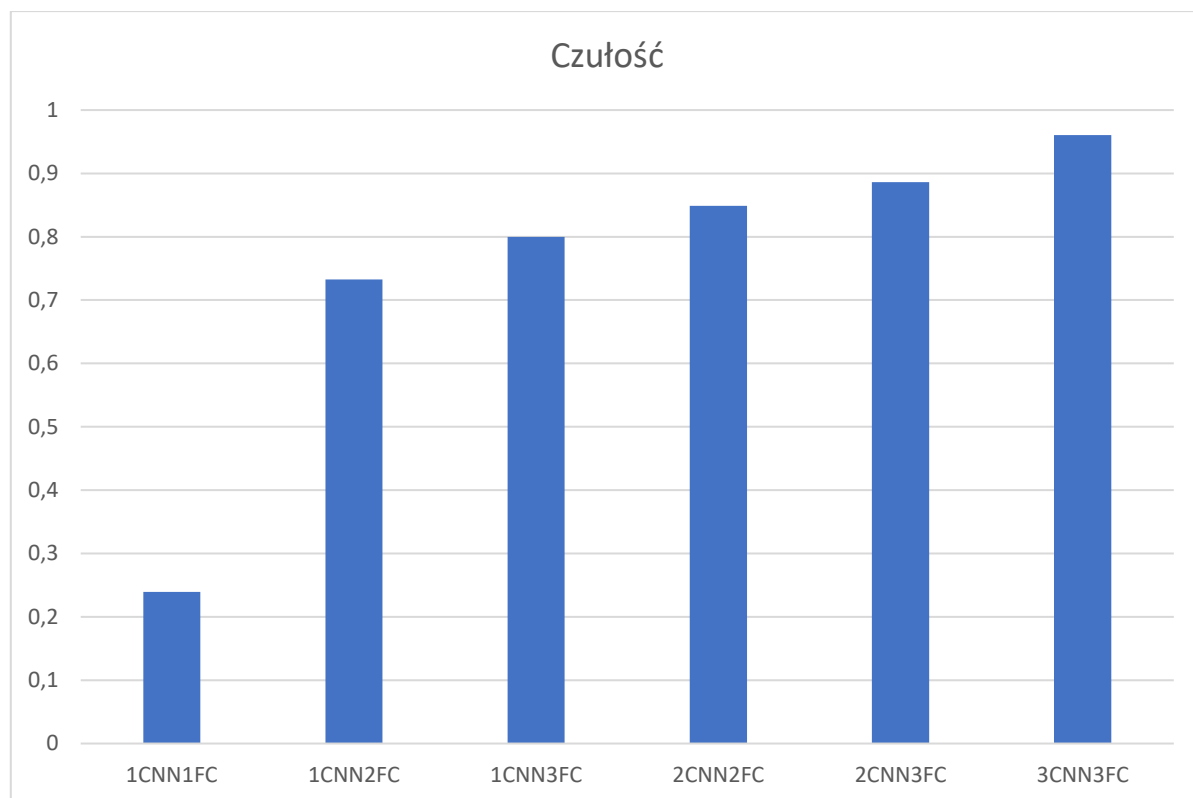
### 1. Dokładność:



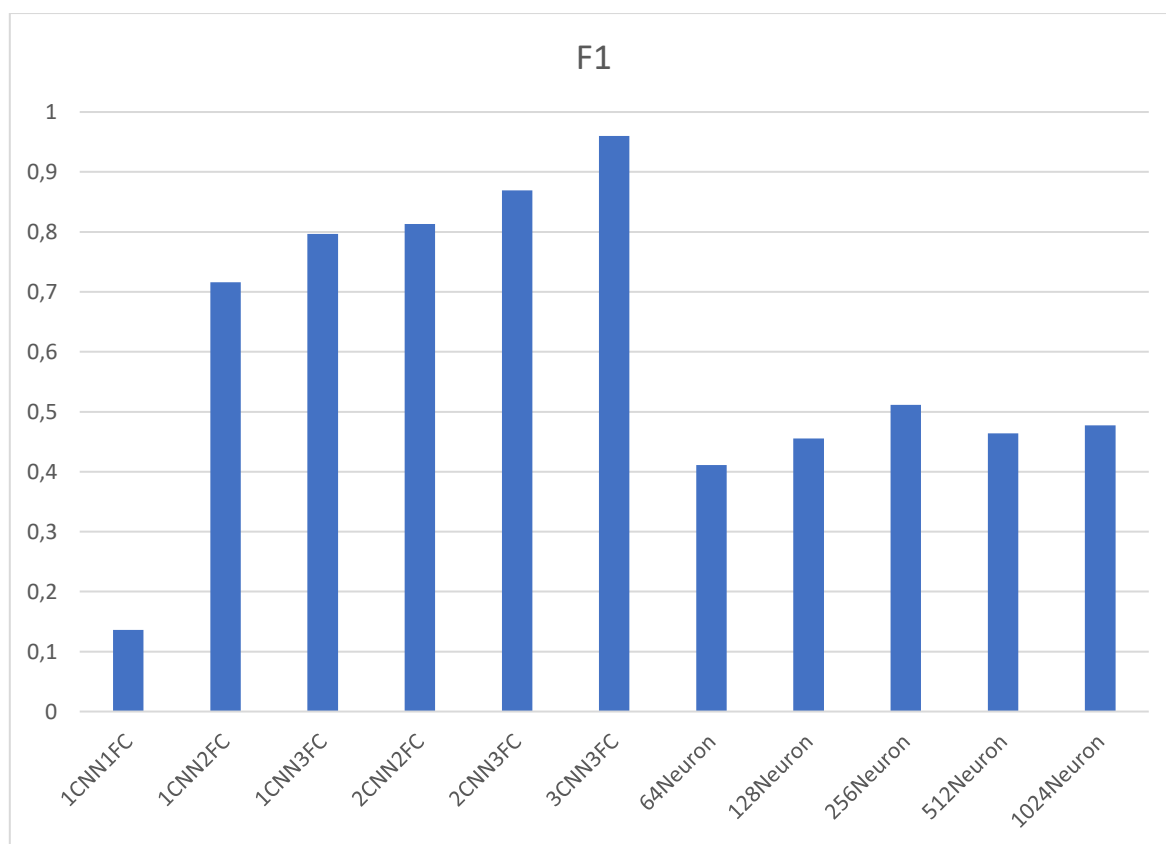
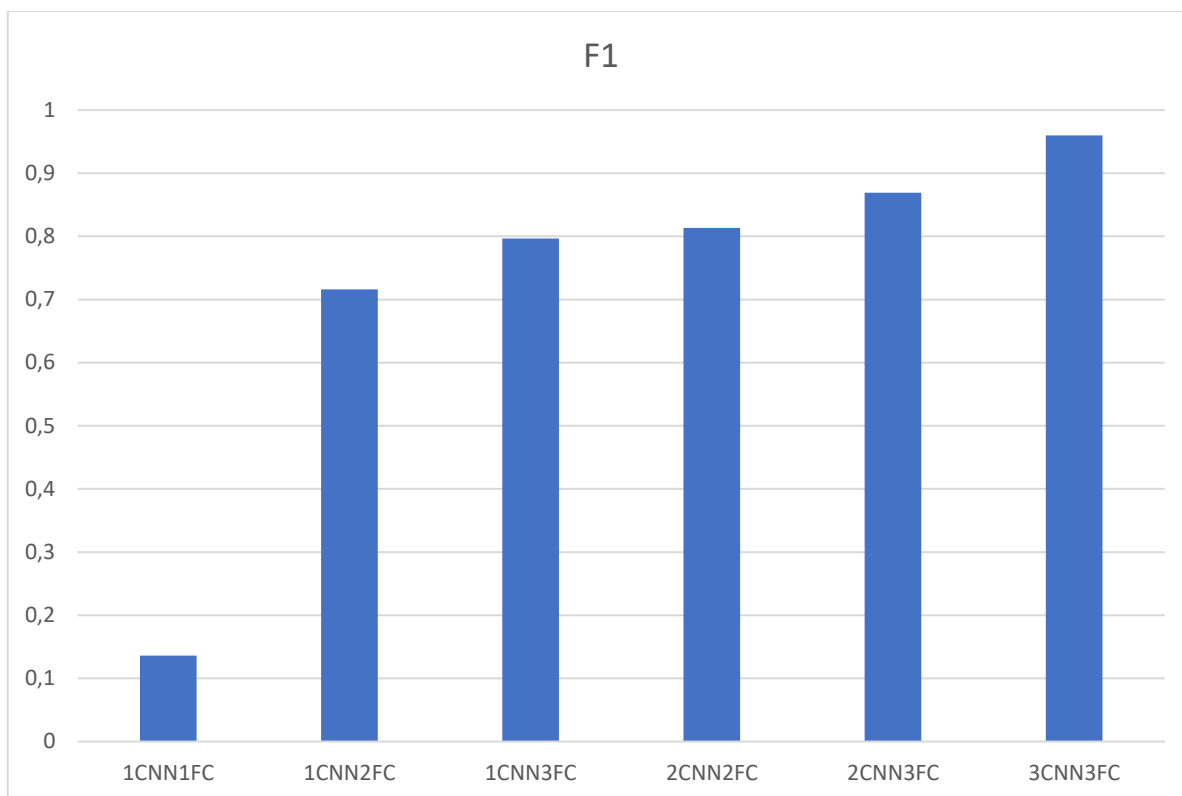
## 2. Precyzja:



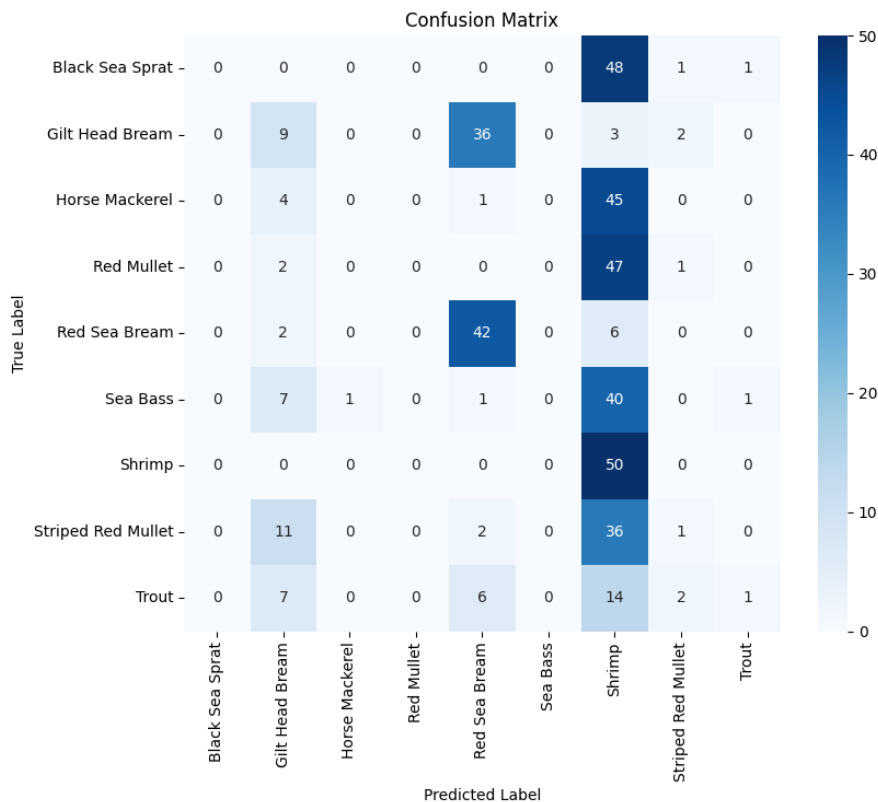
### 3. Czułość:



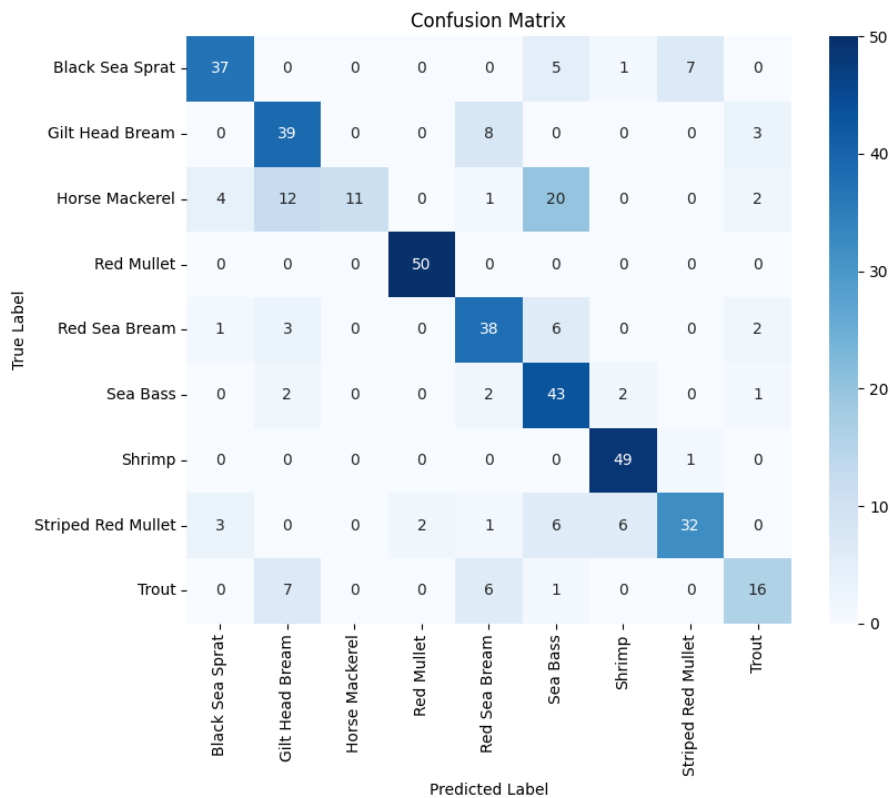
#### 4. F1:



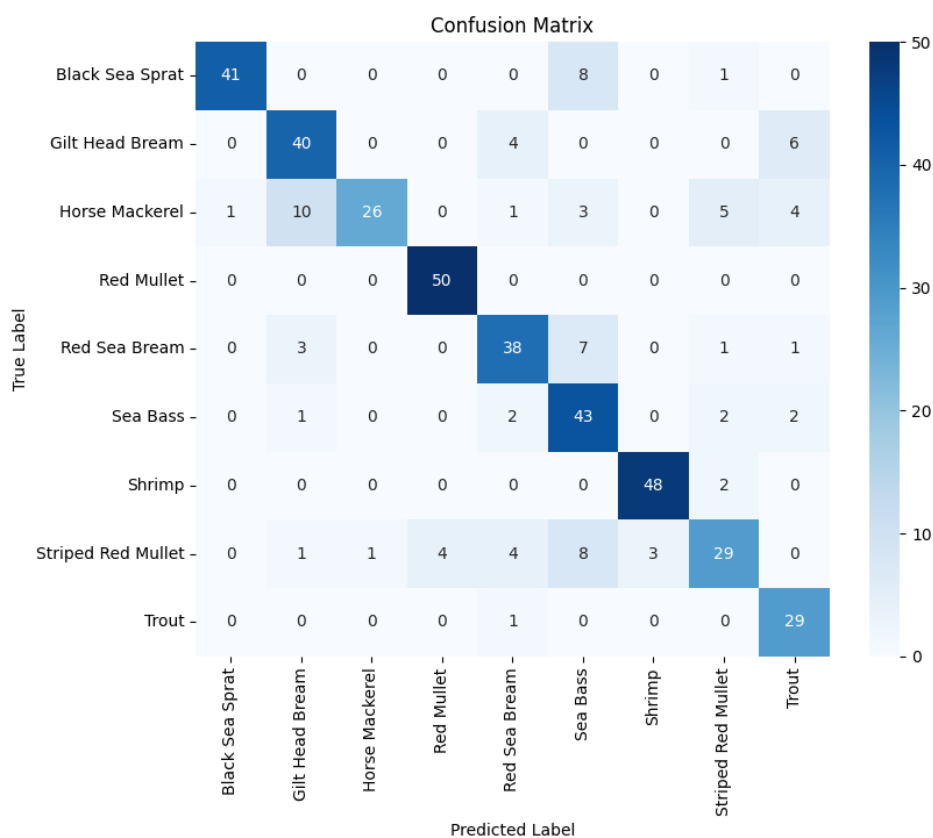
## 5. Macierz konfuzji:



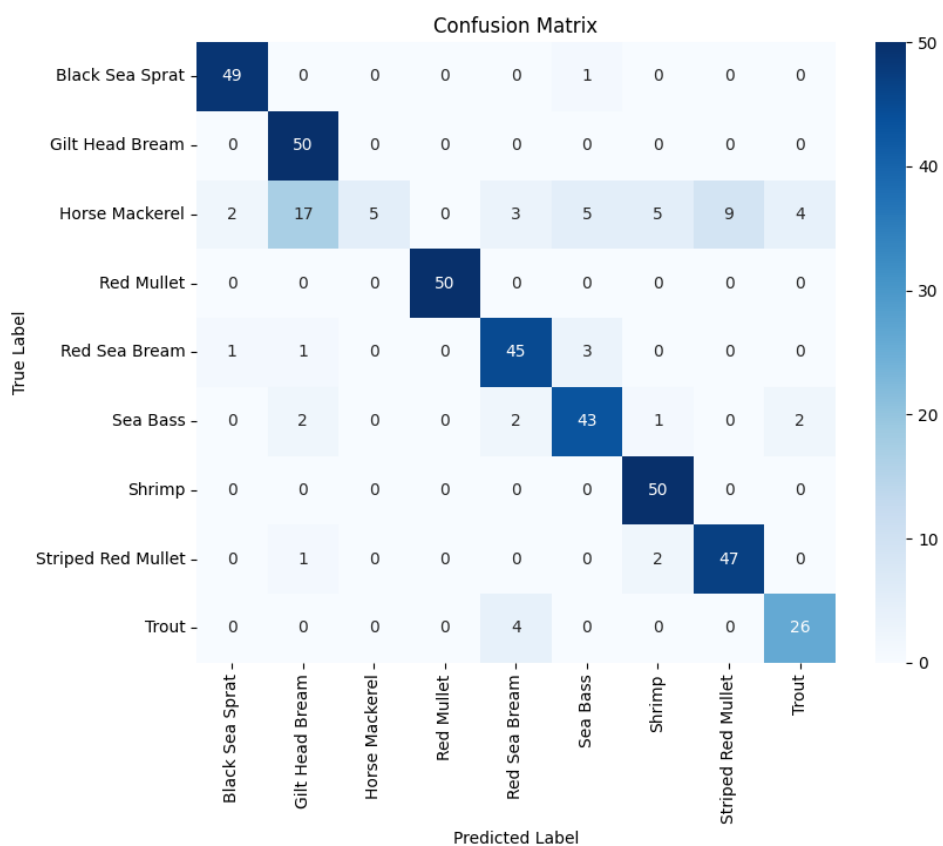
1CNN1FC



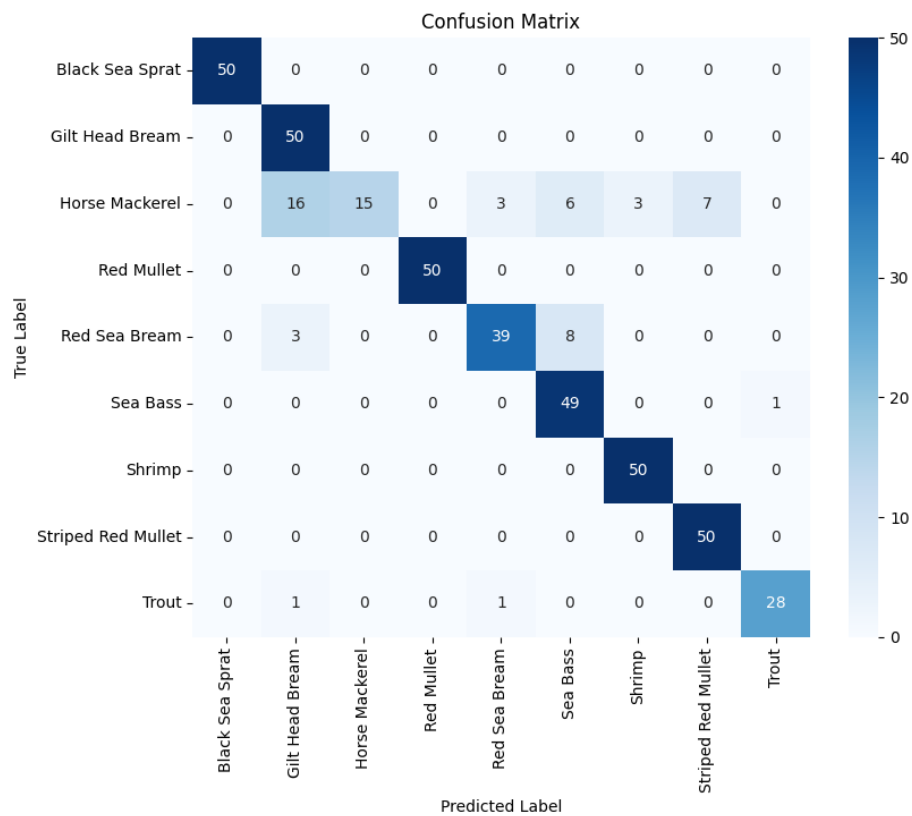
1CNN2FC



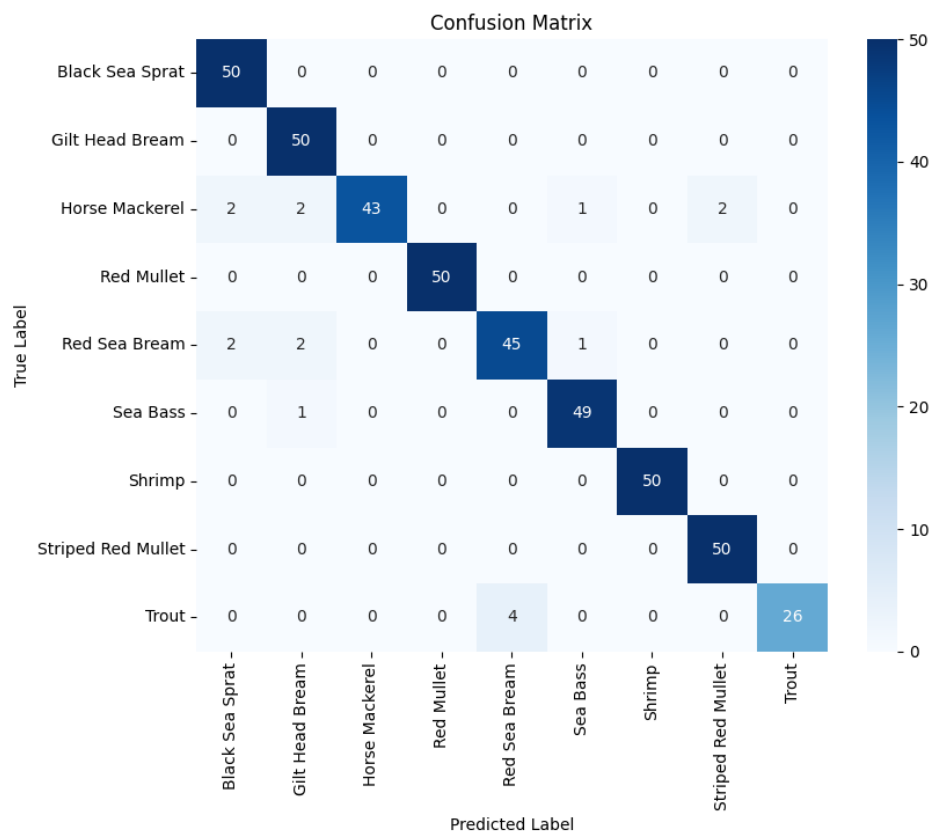
1CNN3FC



2CNN2FC



2CNN3FC



3CNN3FC



## 5 Wnioski

Porównanie sieci neuronowych typu MLP (Multi-Layer Perceptron) i CNN (Convolutional Neural Network) wskazuje na istotne różnice wynikające z ich architektury i zastosowań.

MLP to sieć w pełni połączona, gdzie każdy neuron w jednej warstwie łączy się z każdym neuronem w następnej, co sprawia, że jest uniwersalna, ale mniej efektywna w analizie danych o dużej liczbie cech, takich jak obrazy. Wymaga przekształcenia danych wejściowych na jednowymiarowe wektory, co utrudnia wykorzystanie lokalnych zależności. Z kolei CNN, dzięki operacjom splotowym, jest zoptymalizowana do przetwarzania danych przestrzennych, takich jak obrazy czy wideo, i potrafi rozpoznawać lokalne wzorce, takie jak krawędzie czy tekstury, bez konieczności spłaszczania danych.

CNN lepiej radzi sobie z analizą dużych zbiorów danych wizualnych, jest bardziej skalowalna i redukuje liczbę parametrów dzięki współdzieleniu wag. Podczas gdy MLP jest prostsze w implementacji i może być skuteczne w przypadku danych tablicowych lub o niskim wymiarze, CNN przewyższa je w zadaniach wizji komputerowej, takich jak klasyfikacja obrazów, segmentacja czy wykrywanie obiektów.