

WYDZIAŁ  
MATEMATYKI  
I FIZYKI STOSOWANEJ  
POLITECHNIKI RZESZOWSKIEJ

# Klasyfikacja Obrazów MNIST i PneumoniaMNIST przy Użyciu Sieci Neuronowych

Marcin Przybylski

29 kwietnia 2025

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>2</b>
<b>2</b>	<b>Metodologia</b>	<b>2</b>
2.1	Wczytywanie i Przygotowanie Danych . . . . .	2
2.1.1	Zbiór MNIST . . . . .	2
2.1.2	Zbiór PneumoniaMNIST . . . . .	3
2.2	Architektury Modeli . . . . .	3
2.2.1	W Pełni Połączona Sieć Neuronowa (FCN) . . . . .	3
2.2.2	Konwolucyjna Sieć Neuronowa (CNN) . . . . .	3
2.3	Proces Treningu . . . . .	3
2.4	Ewaluacja Modeli . . . . .	3
<b>3</b>	<b>Uzasadnienie Wyboru Metod i Narzędzi</b>	<b>4</b>
3.1	Wybór Metod (FCN vs. CNN) . . . . .	4
3.2	Wykorzystane Narzędzia . . . . .	4
<b>4</b>	<b>Wyniki i Interpretacja</b>	<b>4</b>
4.1	Klasyfikacja MNIST . . . . .	4
4.1.1	Model FCN . . . . .	4
4.1.2	Model CNN . . . . .	6
4.2	Klasyfikacja PneumoniaMNIST . . . . .	8
4.2.1	Model FCN . . . . .	8
4.2.2	Model CNN . . . . .	9
4.3	Interpretacja Wyników i Odpowiedzi na Pytania . . . . .	11
<b>5</b>	<b>Podsumowanie i Wnioski</b>	<b>12</b>

# 1 Wstęp

Niniejszy dokument stanowi sprawozdanie z projektu dotyczącego zastosowania głębokiego uczenia do problemów klasyfikacji obrazów. Projekt obejmował analizę dwóch różnych zbiorów danych: MNIST, zawierającego ręcznie pisane cyfry, oraz PneumoniaMNIST, składającego się z obrazów rentgenowskich klatki piersiowej do diagnozy zapalenia płuc.

Głównym celem projektu było zbudowanie, wytrenowanie i porównanie dwóch popularnych architektur sieci neuronowych – w pełni połączonej sieci neuronowej (FCN) oraz konwolucyjnej sieci neuronowej (CNN) – w kontekście tych dwóch zadań klasyfikacyjnych. Chciano w ten sposób praktycznie zademonstrować działanie tych modeli, ich proces przygotowania danych, treningu oraz oceny, a także zrozumieć przewagi i ograniczenia każdej z architektur w zależności od charakteru danych obrazowych.

Prace zostały wykonane przy użyciu języka Python oraz bibliotek TensorFlow i Keras. W sprawozdaniu przedstawiono kolejne etapy prac, uzyskane wyniki oraz wnioski płynące z przeprowadzonych eksperymentów.

Wykorzystane w projekcie notatniki Google Colab, zawierające kod i instrukcje, dostępne są pod poniższymi linkami:

- Główny notatnik roboczy: [Link do Colab](#)
- Notatnik - Rozpoznawanie cyfr MNIST: [Link do Colab](#)
- Notatnik - Klasyfikacja PneumoniaMNIST: [Link do Colab](#)

## 2 Metodologia

Prace projektowe obejmowały kilka kluczowych etapów, realizowanych niezależnie dla obu zbiorów danych (MNIST i PneumoniaMNIST), ale z wykorzystaniem podobnych technik i narzędzi.

### 2.1 Wczytywanie i Przygotowanie Danych

#### 2.1.1 Zbiór MNIST

Dane MNIST, zawierające 60 000 obrazów treningowych i 10 000 testowych cyfr (28x28 pikseli, skala szarości), zostały wczytane bezpośrednio z biblioteki Keras. Zbiór treningowy podzielono dodatkowo na właściwy zbiór treningowy (85%) i zbiór walidacyjny (15%) w celu monitorowania procesu uczenia. Przygotowanie danych obejmowało:

- **Normalizację:** Wartości pikseli (oryginalnie 0-255) przeskalowano do zakresu  $[0, 1]$  poprzez podzielenie przez 255.
- **Spłaszczenie (dla FCN):** Obrazy 28x28 pikseli zostały przekształcone w jednowymiarowe wektory o długości 784.
- **Zachowanie struktury 2D (dla CNN):** Dla sieci CNN obrazy zachowały kształt (28, 28, 1).

- **Kodowanie One-Hot Etykiet:** Etykiety klas (0-9) przekonwertowano na format "one-hot".

### 2.1.2 Zbiór PneumoniaMNIST

Dane PneumoniaMNIST (obrazy RTG klatki piersiowej, klasy: Normal vs. Pneumonia) zostały wczytane przy użyciu biblioteki 'tensorflow\_datasets'. Przygotowanie danych obejmowało:

- **Zmiana Rozmiaru i Skala Szarości:** Obrazy przeskalowano do 28x28 pikseli i przekonwertowano do skali szarości.
- **Normalizacja:** Wartości pikseli przeskalowano do zakresu [0, 1].
- **Spłaszczenie (dla FCN):** Obrazy 28x28x1 przekształcono w wektory 784.
- **Zachowanie struktury 2D (dla CNN):** Zachowano format (28, 28, 1).
- **Kodowanie One-Hot Etykiet:** Etykiety binarne (0, 1) przekonwertowano na format one-hot ([1, 0], [0, 1]).

## 2.2 Architektury Modeli

### 2.2.1 W Pełni Połączona Sieć Neuronowa (FCN)

Model FCN składał się z warstwy wejściowej (784 neurony), dwóch ukrytych warstw gęstych (Dense) po 512 neuronów z aktywacją ReLU i Dropout (0.2, 0.1/0.2) oraz warstwy wyjściowej Dense (10 neuronów dla MNIST, 2 dla PneumoniaMNIST) z aktywacją Softmax.

### 2.2.2 Konwolucyjna Sieć Neuronowa (CNN)

Model CNN składał się z dwóch bloków konwolucyjnych (Conv2D 3x3 ReLU + MaxPooling 2x2; 32 filtry w bloku 1, 64 w bloku 2; dodatkowo BatchNormalization w modelu PneumoniaMNIST) oraz części FCN (Flatten, Dense 512 ReLU, Dropout 0.2, Dense wyjściowe z Softmax).

## 2.3 Proces Treningu

Modele kompilowano z funkcją kosztu 'categorical\_crossentropy', optymalizatorem 'Adam' (lub 'SGD' dla FCN MNIST) i metryką 'accuracy'. Trening trwał określoną liczbę epok, z ustalonym 'batch\_size'. Dla CNN PneumoniaMNIST zastosowano augmentację danych. Postęp monitorowano za pomocą 'livelossplot'.

## 2.4 Ewaluacja Modeli

Wydajność oceniano na zbiorach testowych, wykorzystując metryki: dokładność (accuracy), stratę (loss) oraz analizując macierz pomyłek i przykłady błędnych klasyfikacji.

## 3 Uzasadnienie Wyboru Metod i Narzędzi

### 3.1 Wybór Metod (FCN vs. CNN)

Zastosowanie FCN jako modelu bazowego i CNN jako standardu w przetwarzaniu obrazów pozwoliło na porównanie podejść i zademonstrowanie przewagi CNN w wykorzystaniu informacji przestrzennej obrazu.

### 3.2 Wykorzystane Narzędzia

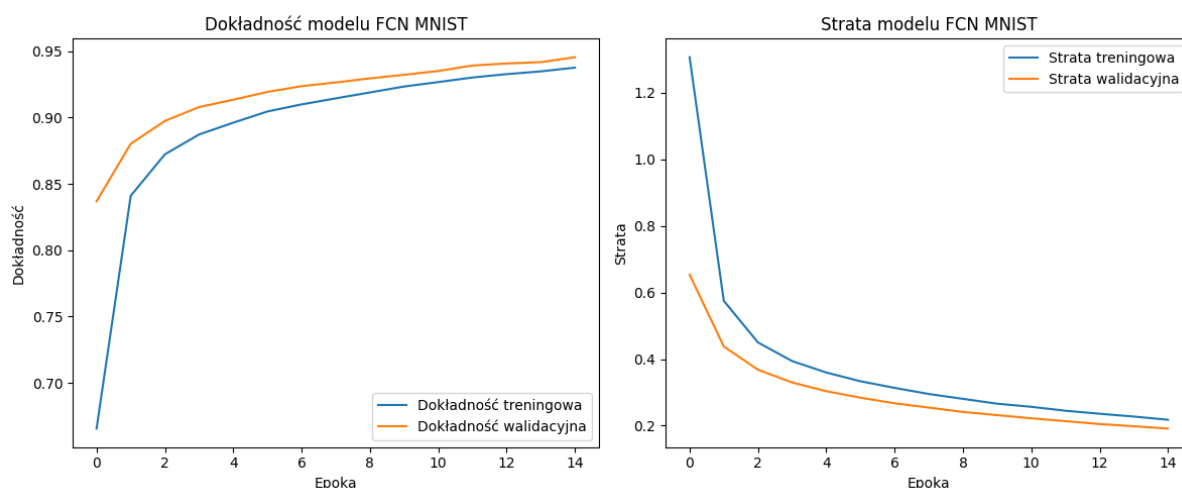
Wybrano standardowe narzędzia ekosystemu Python: TensorFlow z Keras API do budowy i treningu modeli, TensorFlow Datasets do wczytania danych PneumoniaMNIST, NumPy do operacji na danych, Matplotlib i Seaborn do wizualizacji, Scikit-learn do podziału danych i metryk, oraz LiveLossplot do monitorowania treningu. Wybór podyktowany był ich popularnością, efektywnością i dostępnością zasobów.

## 4 Wyniki i Interpretacja

### 4.1 Klasyfikacja MNIST

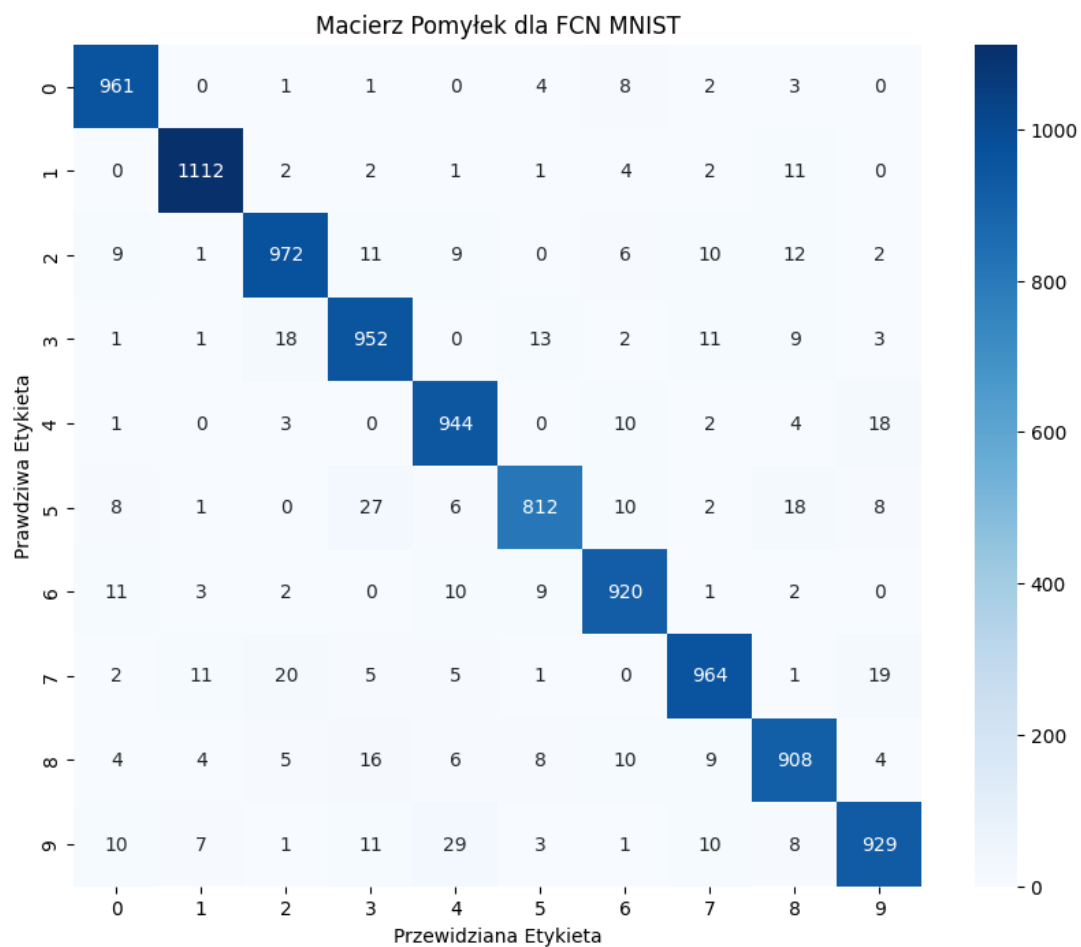
#### 4.1.1 Model FCN

Wytrenowany model FCN osiągnął na zbiorze testowym MNIST **dokładność 94.74%** (strata 0.1844). Poprawnie sklasyfikowano 9474 obrazy, błędnie 526. Krzywe uczenia (Rysunek 1) pokazują typowy przebieg nauki dla tego typu modelu.



Rysunek 1: Krzywe uczenia (dokładność i strata) dla modelu FCN na zbiorze MNIST.

Analiza macierzy pomyłek (Rysunek 2) wskazuje, że najczęstsze błędy dotyczyły mylenia cyfr wizualnie podobnych. Przykłady błędnych klasyfikacji (Rysunek 3) często obejmują obrazy niejednoznaczne.



Rysunek 2: Macierz pomyłek FCN (MNIST)

### FCN MNIST: Przykłady Niepoprawnych Predykcji

Pred: 1, Praw: 3



Pred: 9, Praw: 8



Pred: 2, Praw: 3



Pred: 3, Praw: 5



Pred: 2, Praw: 4



Pred: 2, Praw: 7



Pred: 4, Praw: 6



Pred: 2, Praw: 7



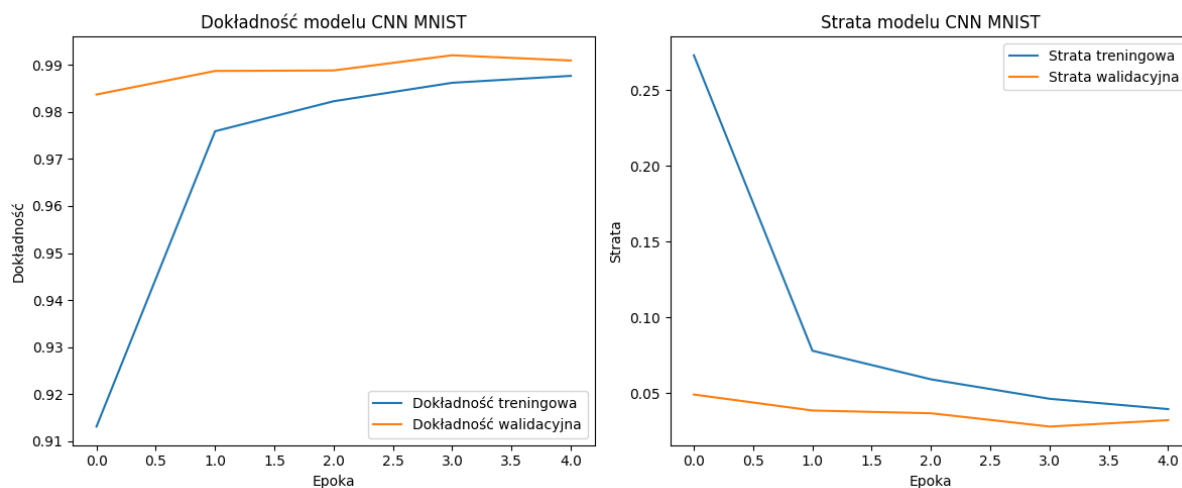
Pred: 7, Praw: 2



Rysunek 3: Przykłady błędnych klasyfikacji FCN (MNIST)

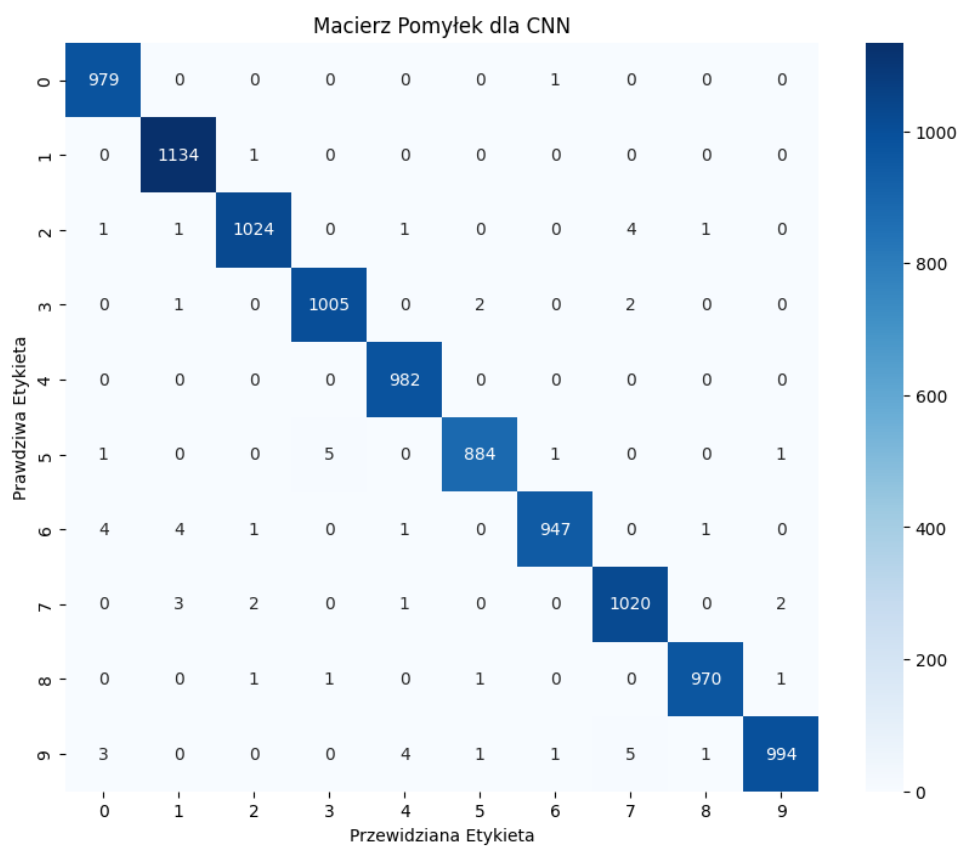
### 4.1.2 Model CNN

Model CNN wykazał się bardzo wysoką skutecznością, osiągając **dokładność 99.23%** (strata 0.0225) na zbiorze testowym MNIST. Podczas ostatniej epoki treningu dokładność na zbiorze walidacyjnym wyniosła 99.09% (strata 0.0320). Krzywe uczenia (Rysunek 4) wskazują na efektywny proces nauki.



Rysunek 4: Krzywe uczenia (dokładność i strata) dla modelu CNN na zbiorze MNIST.

Na zbiorze testowym model poprawnie sklasyfikował 9923 cyfry, popełniając 77 błędów. Macierz pomyłek (Rysunek 5) i przykłady błędów (Rysunek 6) potwierdzają wysoką jakość klasyfikacji.



Rysunek 5: Macierz pomyłek CNN (MNIST)

#### CNN: Przykłady Niepoprawnych Predykcji

Pred: 4, Praw: 6



Pred: 5, Praw: 3



Pred: 5, Praw: 9



Pred: 5, Praw: 8



Pred: 9, Praw: 8



Pred: 7, Praw: 3



Pred: 3, Praw: 5



Pred: 2, Praw: 1



Pred: 3, Praw: 8



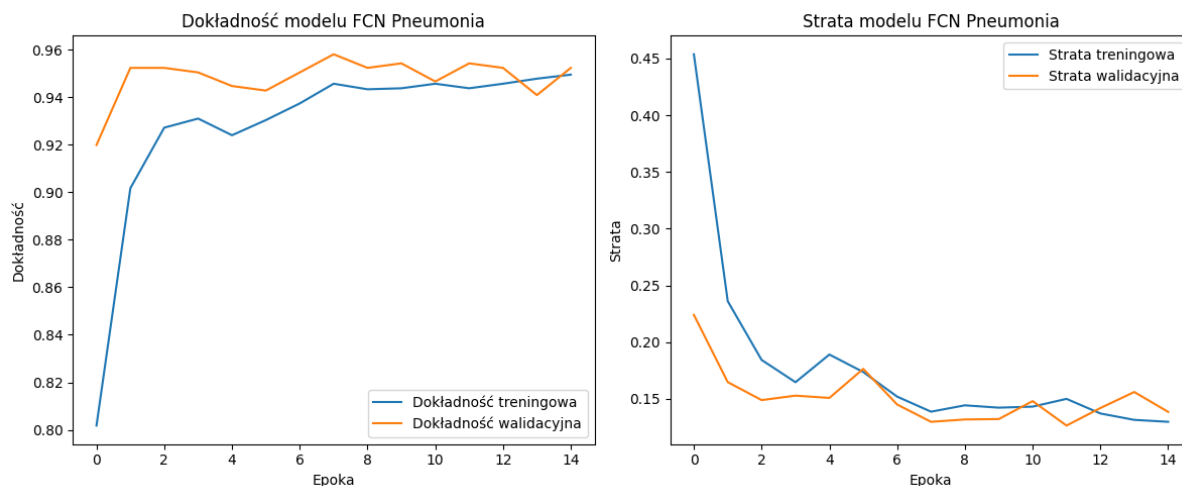
Rysunek 6: Przykłady błędnych klasyfikacji CNN (MNIST)



## 4.2 Klasyfikacja PneumoniaMNIST

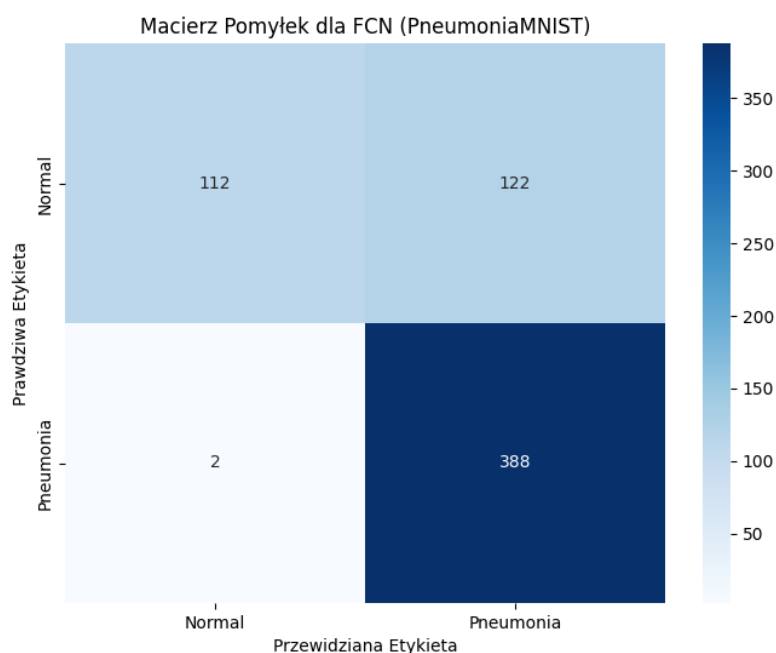
### 4.2.1 Model FCN

W zadaniu klasyfikacji binarnej obrazów RTG, model FCN osiągnął **dokładność 80.13%** (strata 0.8064) na zbiorze testowym. Poprawnie sklasyfikowano 500 obrazów, błędnie 124. Krzywe uczenia przedstawiono na Rysunku 7.



Rysunek 7: Krzywe uczenia (dokładność i strata) dla modelu FCN na zbiorze PneumoniaMNIST.

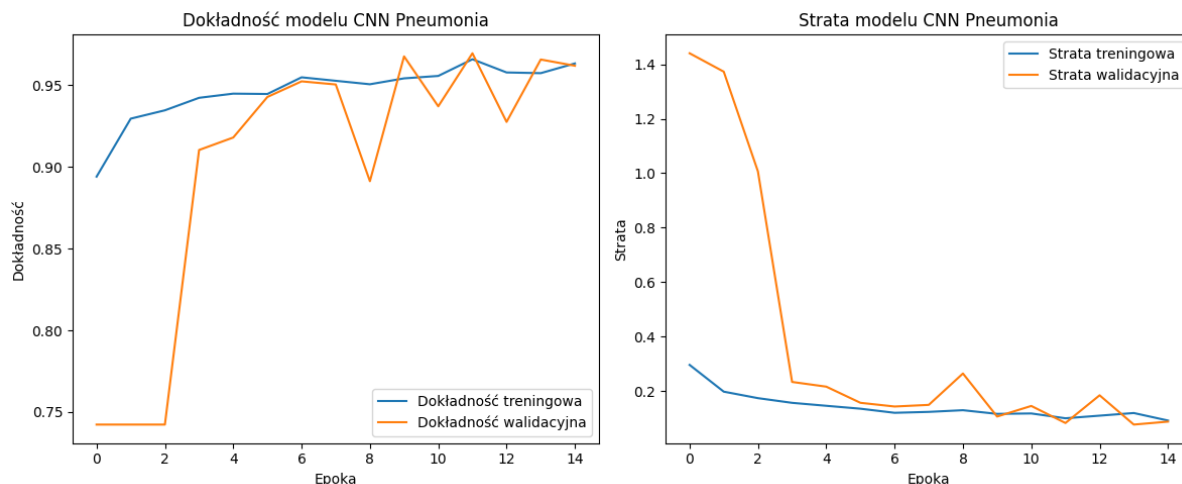
Macierz pomyłek (Rysunek 8) pozwala ocenić typy błędów. [Opisz krótko obserwacje z macierzy pomyłek dla Twoich wyników FCN Pneumonia, np. ile było FN i FP].



Rysunek 8: Macierz pomyłek FCN (PneumoniaMNIST).

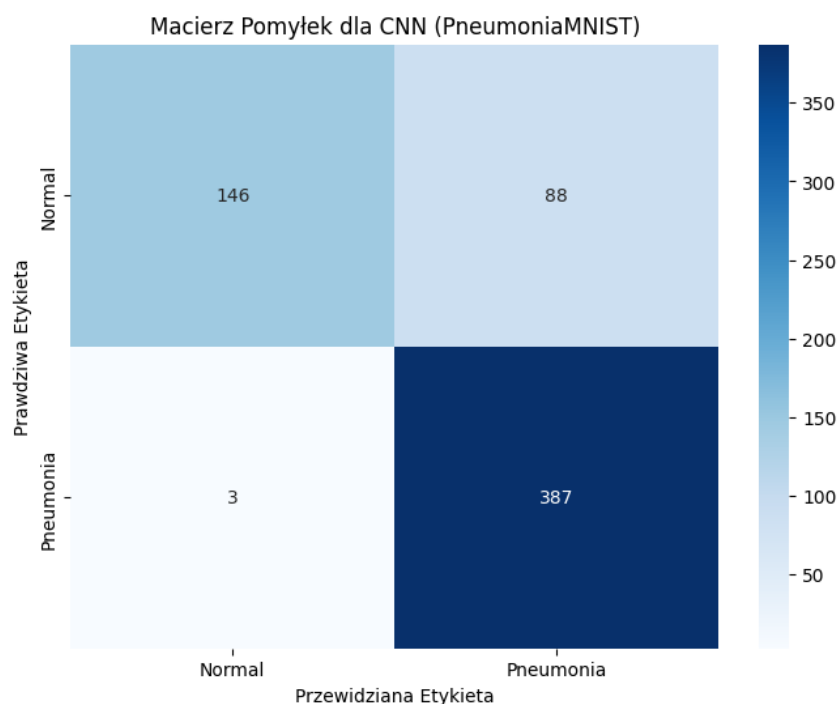
#### 4.2.2 Model CNN

Model CNN uzyskał w tym zadaniu **dokładność 86.38%** (strata 0.4459) na zbiorze testowym. W ostatniej epoce treningu dokładność walidacyjna wyniosła 96.18% (strata 0.0876). Krzywe uczenia przedstawiono na Rysunku 9.



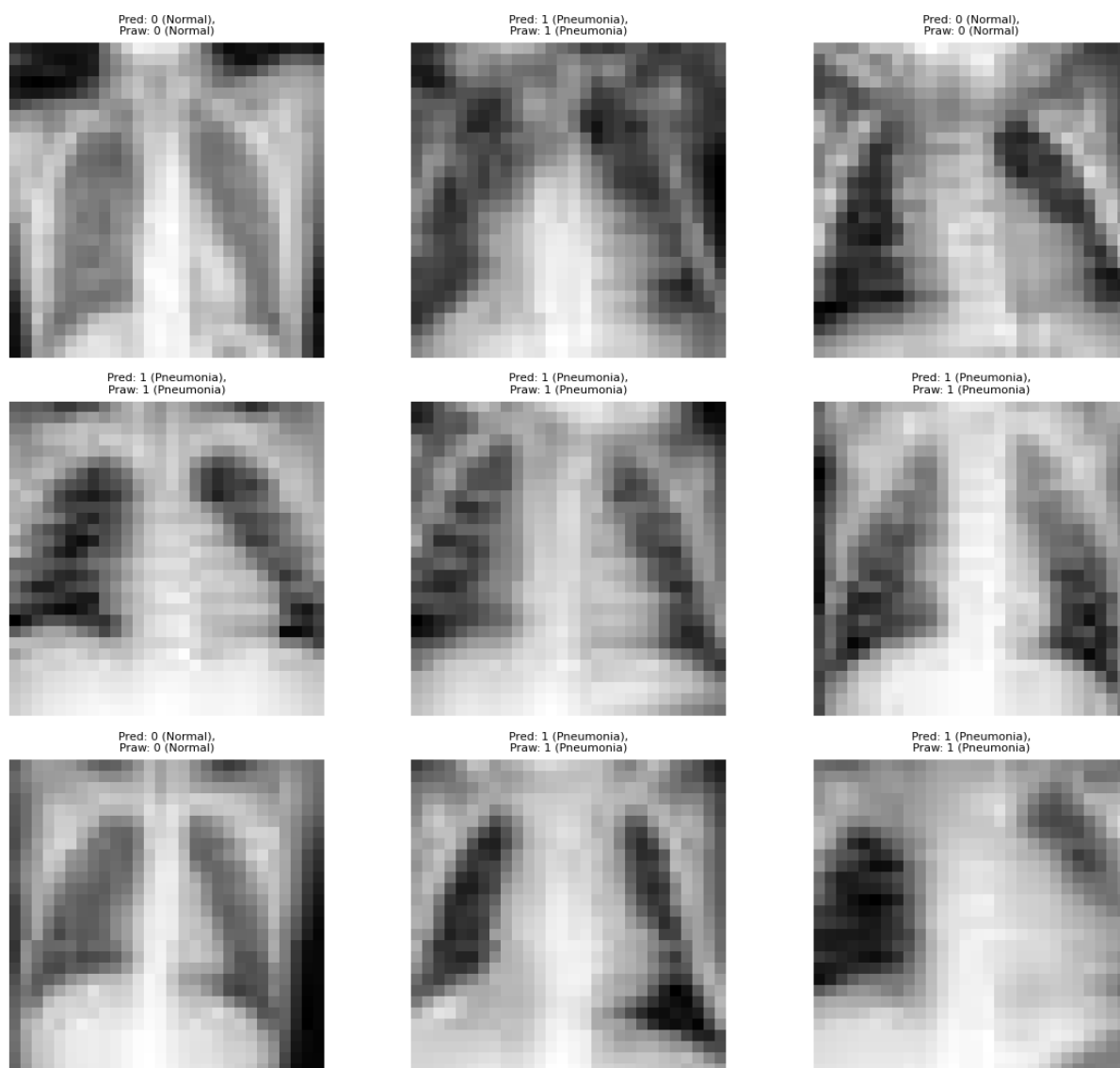
Rysunek 9: Krzywe uczenia (dokładność i strata) dla modelu CNN na zbiorze PneumoniaMNIST.

Na zbiorze testowym model poprawnie sklasyfikował 539 przypadków, myląc się w 85 przypadkach. Macierz pomyłek dla CNN (Rysunek 10) szczegółowo ilustruje te wyniki. [Opisz krótko obserwacje z macierzy, np. ile było FN i FP]. Przykłady poprawnych i błędnych klasyfikacji przedstawiono na Rysunkach 11 i 12.



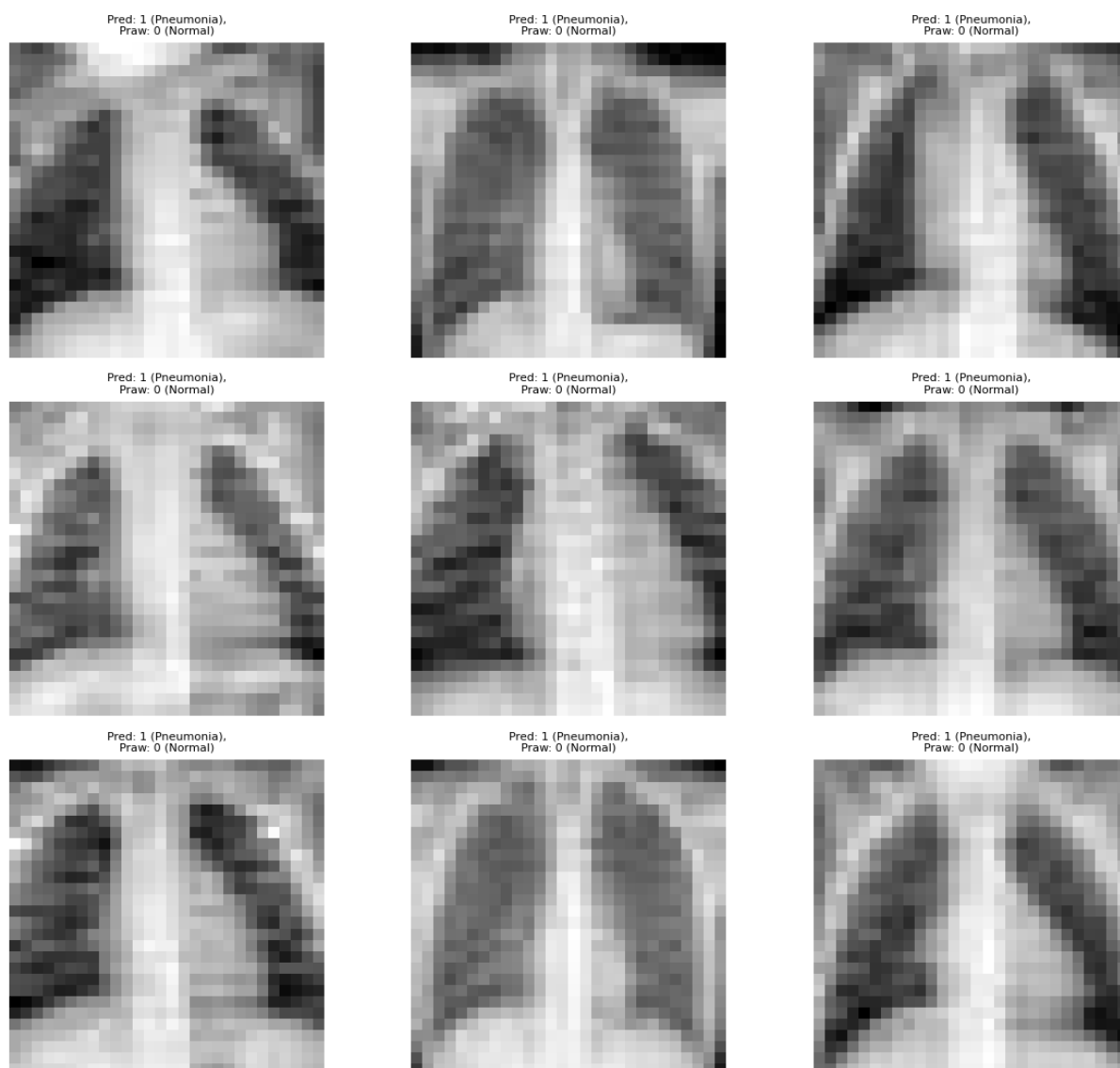
Rysunek 10: Macierz pomyłek CNN (PneumoniaMNIST)

### CNN: Przykłady Poprawnych Predykcji (0: Normal, 1: Pneumonia)



Rysunek 11: Przykłady poprawnych klasyfikacji CNN (PneumoniaMNIST)

### CNN: Przykłady Niepoprawnych Predykcji (0: Normal, 1: Pneumonia)



Rysunek 12: Przykłady błędnych klasyfikacji CNN (PneumoniaMNIST)

## 4.3 Interpretacja Wyników i Odpowiedzi na Pytania

Wyniki jednoznacznie potwierdzają, że **architektury CNN są znacznie lepiej przystosowane do zadań klasyfikacji obrazów** niż standardowe sieci FCN. Wykorzystanie cech przestrzennych przez warstwy konwolucyjne i pooling prowadzi do wyższej dokładności zarówno w przypadku rozpoznawania cyfr (99.23% vs 94.74%), jak i klasyfikacji obrazów medycznych (86.38% vs 80.13%). Różnica w dokładności jest szczególnie widoczna na bardziej złożonym zbiorze PneumoniaMNIST.

Analiza błędów pokazuje, że modele często mylą klasy wizualnie podobne (cyfry) lub przypadki graniczne (obrazy medyczne). Nawet dla dokładniejszych modeli CNN liczba błędów nie jest zerowa (77 dla MNIST, 85 dla PneumoniaMNIST).

Odpowiedzi na pytania:

- **Zbiór MNIST:** To standardowa baza danych ręcznie pisanych cyfr (0-9), 28x28 pikseli, powszechnie używana do testowania algorytmów rozpoznawania obrazów.
- **Format One-Hot:** Metoda kodowania kategorii (np. cyfr) jako wektorów binarnych, gdzie tylko jedna pozycja (odpowiadająca danej kategorii) ma wartość 1, a reszta 0. Np. cyfra '3' to '[0,0,0,1,0,0,0,0,0,0]'.
- **Neuron dla cyfry '0':** W 10-klasowym modelu MNIST z kodowaniem one-hot, neuron na pierwszej pozycji (indeks 0) odpowiada cyfrze '0'.
- **Poprawność wykresu kosztu:** Wykres ilustrujący minima lokalne i globalne funkcji kosztu jest koncepcyjnie poprawny – pokazuje wyzwanie optymalizacyjne polegające na unikaniu minimów lokalnych.
- **Funkcje kosztu (Cross-Entropy):** *Categorical* jest używana dla wielu klas ( $>2$ ) z etykietami one-hot. *Binary* jest używana dla dwóch klas, często z pojedynczym neuronem wyjściowym (sigmoid).
- **Przyspieszenie GPU:** Użycie GPU znacząco przyspiesza trening (zwłaszcza CNN), często kilkukrotnie lub kilkudziesięciokrotnie w porównaniu do CPU, ale dokładny zysk zależy od sprzętu i modelu.
- **Błędy "ludzkie":** Tak, sieci neuronowe (szczególnie FCN) mogą mylić wizualnie podobne cyfry (np. 1 i 7), co jest analogiczne do błędów ludzkich. Widać to w macierzach pomyłek.
- **Duży Batch Size (np. 10000):** Skraca czas epoki, ale może pogorszyć generalizację i finalną dokładność modelu. Dla MNIST byłby to bardzo duży batch.
- **Mały Batch Size (np. 32):** Wydłuża czas epoki, ale często poprawia generalizację (choć może zwiększyć szum w treningu). Jest to popularny wybór.
- **FCN vs CNN dla cyfr:** CNN są znacznie lepsze do rozpoznawania cyfr (i ogólnie obrazów) niż FCN, ponieważ wykorzystują informacje o przestrzennej strukturze pikseli.

## 5 Podsumowanie i Wnioski

W ramach projektu zrealizowano zadanie budowy i oceny modeli głębokiego uczenia (FCN i CNN) do klasyfikacji obrazów z dwóch różnych zbiorów danych: MNIST i PneumoniaMNIST. Przeprowadzono pełen cykl pracy: od przygotowania danych, przez definicję i trening modeli, po ewaluację i interpretację wyników.

Główne wnioski płynące z projektu to:

1. **Skuteczność CNN:** Konwolucyjne sieci neuronowe wykazały znaczną przewagę nad w pełni połączonymi sieciami w obu zadaniach klasyfikacji obrazów, osiągając wyższą dokładność (99.23% dla MNIST i 86.38% dla PneumoniaMNIST). Potwierdza to ich adekwatność do analizy danych wizualnych.
2. **Znaczenie Przetwarzania Danych:** Odpowiednie przygotowanie danych (normalizacja, formatowanie, kodowanie etykiet) jest kluczowe. Augmentacja danych może poprawić generalizację (stosowana dla CNN PneumoniaMNIST).

3. **Interpretowalność Modeli:** Analiza wyników (metryki, macierze pomyłek, przykłady błędów) dostarcza cennych informacji o wydajności i ograniczeniach modeli.
4. **Praktyczne Zastosowanie Narzędzi:** Projekt zademonstrował praktyczne wykorzystanie popularnych bibliotek (TensorFlow, Keras, Scikit-learn, Matplotlib, Seaborn, TFDS) w realizacji zadań uczenia maszynowego.