

Laboratorium 1

Wprowadzenie do sztucznej inteligencji

Agnieszka Głuszkiewicz

1 Sieć neuronowa - cyfry z bazy MNIST

1.1 Opis zadania

Celem zadania było stworzenie i wytrenowanie sieci neuronowej rozpoznawającej cyfry ze zbioru danych MNIST zawierającego 60 000 obrazów treningowych i 10 000 obrazów testowych. Wykorzystałam w tym celu bibliotekę Keras w Pythonie. Oprócz tego należało przedstawić wartości wspomnianych w zadaniu wskaźników określających stopień poprawności rozpoznawania obrazów (*accuracy*, *loss*, *precision*, *recall*), dodatkowo przedstawiłam wyniki w formie *Confusion Matrix*.

1.2 Struktura modelu

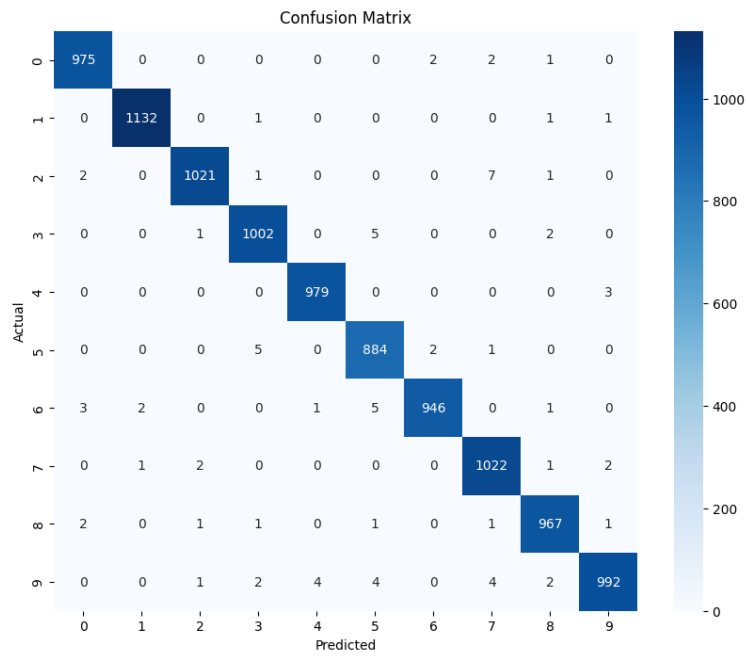
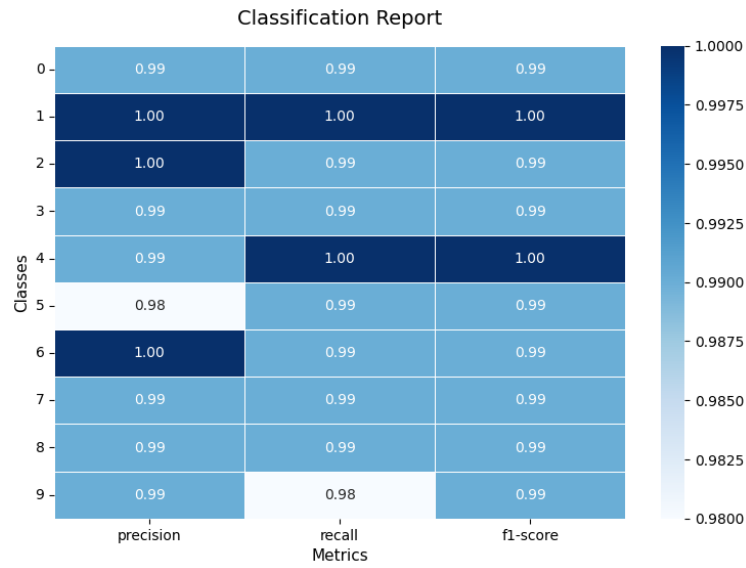
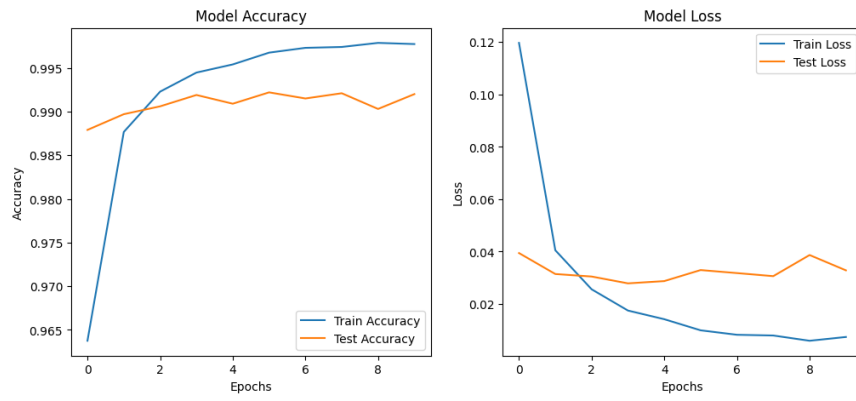
```
model = keras.Sequential([
    keras.layers.Input(shape=(28, 28, 1)),
    keras.layers.Conv2D(32, kernel_size=(3, 3), activation="relu"),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.Conv2D(128, kernel_size=(3, 3), activation="relu"),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.Flatten(),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(10, activation='softmax')
])
```

1.3 Kompilacja modelu

```
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

1.4 Wyniki

- Dokładność na zbiorze testowym: Model osiągnął dokładność 0,992 (wartość uzyskana w wyniku ewaluacji).
- Na podstawie wartości pojawiających się w *Confusion Matrix* można wysunąć hipotezę, że większość błędów pojawia się w klasach o podobnych kształtach (niektóre "3" mogą być mylone z "5", "2" z "7" itp.).
- Wniosek: Model bardzo dobrze rozpoznaje cyfry z klasycznego zbioru MNIST. Ewentualne błędy mogą wynikać z podobieństwa wizualnego niektórych cyfr lub z niewielkiej liczby trudnych przypadków w zbiorze testowym.



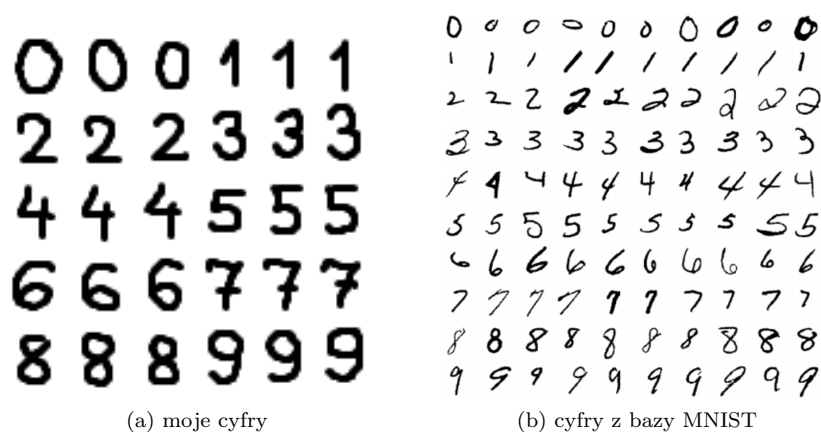
2 Sieć neuronowa - własne obrazy cyfr

2.1 Opis zadania

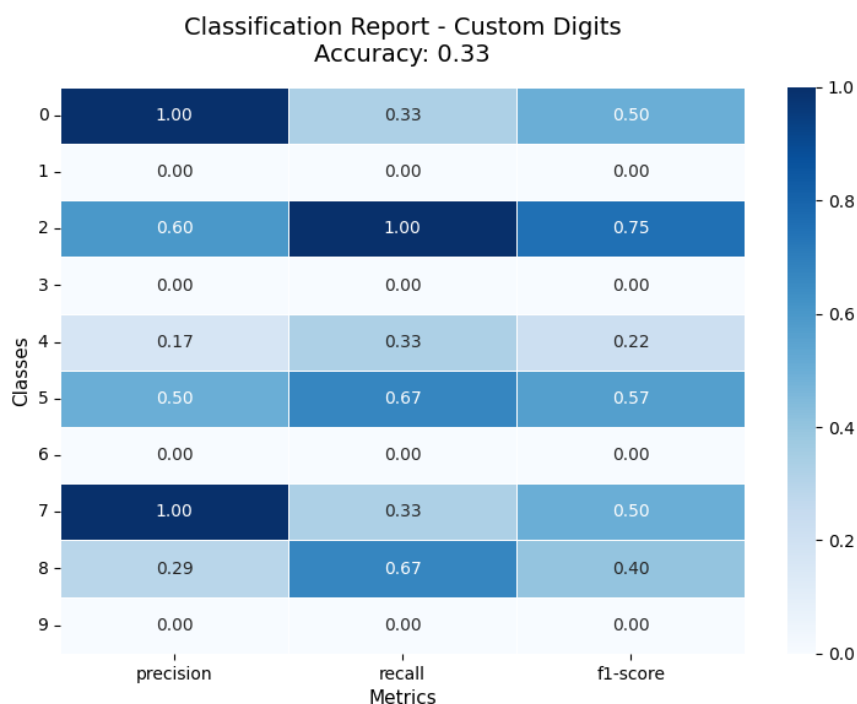
Na podstawie własnych próbek pisma (3 egzemplarze każdej cyfry w formacie 28×28 pikseli) należało utworzyć osobny zbiór testowy i sprawdzić, jak reaguje na niego sieć neuronowa stworzona w poprzednim zadaniu.

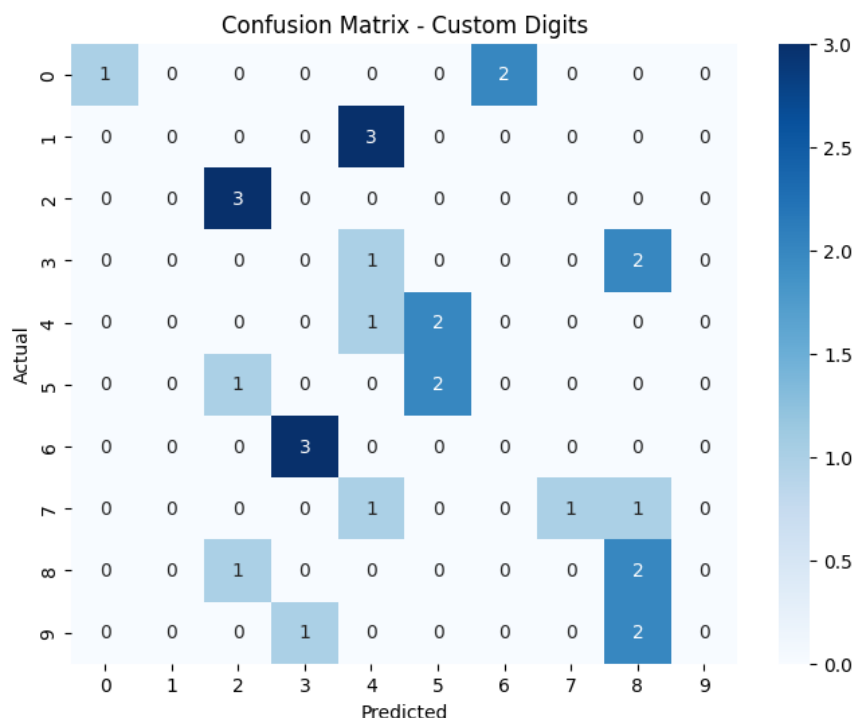
2.2 Wyniki

- Dokładność na zbiorze testowym: Model osiągnął dokładność 0,33.
- Współczynnik rozpoznawalności jest znacznie niższy niż dla zbioru MNIST.
- Może być to spowodowane faktem, że baza MNIST powstała na bazie pisma amerykańskiego, którego styl różni się od naszego (np. u mnie 7 ma belkę - w bazie nie, u mnie 9 ma zaokrąglony ogonek - w bazie nie, itd. - zobacz rysunek 1).
- Nie bez znaczenia pozostaje także jakość obrazków, które przygotowałam w Paintcie. Czynniki takie jak brak wycentrowania cyfr czy nieodpowiednia grubość czcionki mogły również wpłynąć na pogorszenie wyglądu, a tym samym - rozpoznawalności.



Rysunek 1: Różnice w wyglądzie cyfr





3 *Random Forest* - cyfry z bazy MNIST

3.1 Opis zadania

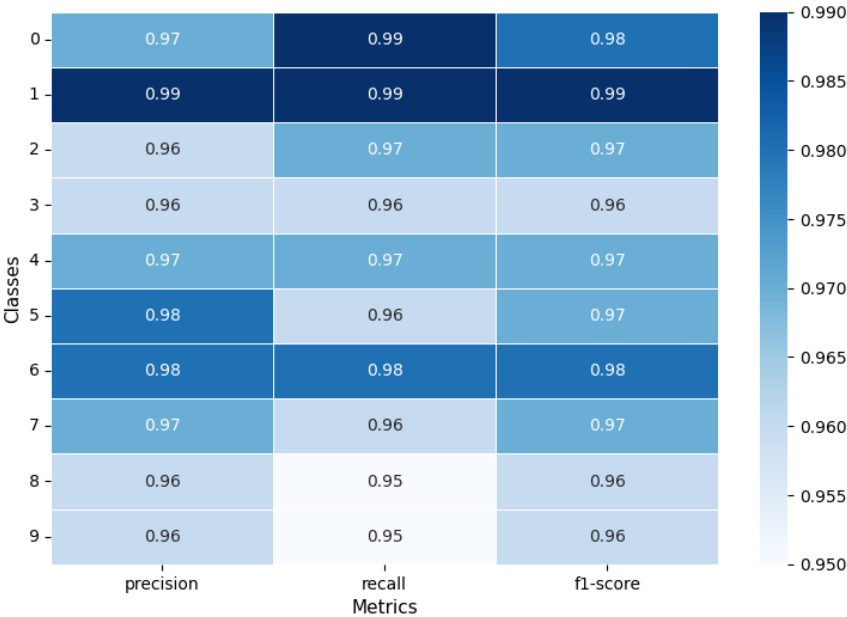
Celem eksperymentu było stworzenie klasyfikatora opartego na Random Forest, który rozpoznaje cyfry ze zbioru danych MNIST oraz sprawdzenie jego skuteczności na zbiorze testowym.

3.2 Wyniki

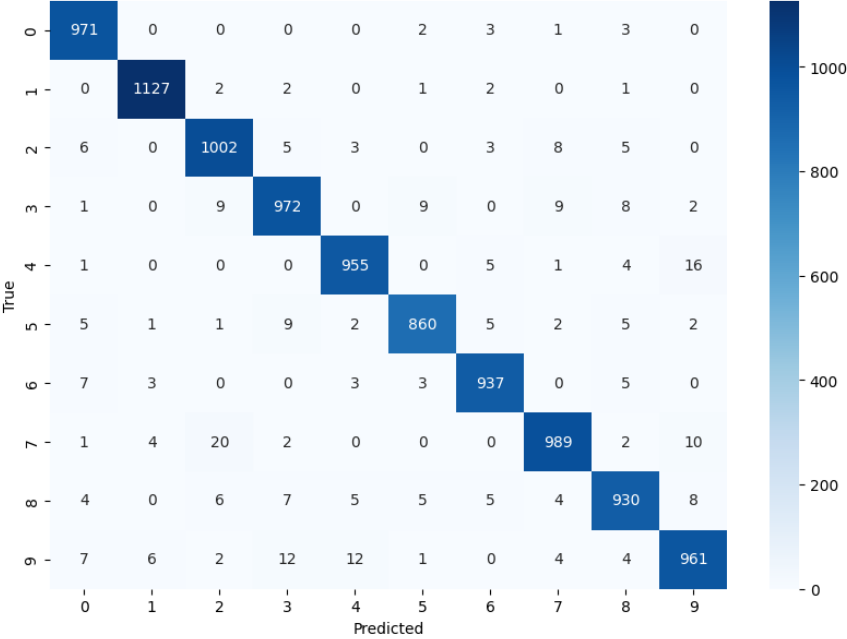
Model został oceniony na dwóch zbiorach testowych:

- Wyniki dla zbioru testowego MNIST
 - Dokładność (Accuracy): 0,97
 - Macierz błędów oraz raport klasyfikacji wskazują, że model dobrze rozpoznaje cyfry MNIST.
- Wyniki dla własnych cyfr
 - Dokładność (Accuracy): 0,03
 - Macierz błędów pokazuje, że model bardzo często myli cyfry.
 - Czulość (Recall) i precyzja (Precision) były bardzo niskie – model miał duże trudności z poprawnym rozpoznaniem cyfr.
 - Przyczyną niskiej skuteczności na własnych danych mogą być różnice w stylu pisania (patrz poprzednie zadanie) oraz bardzo mały zbiór testowy – testowanie na 30 obrazkach mogło powodować przypadkowe wyniki.

Classification Report - MNIST Digits
Accuracy: 0.97

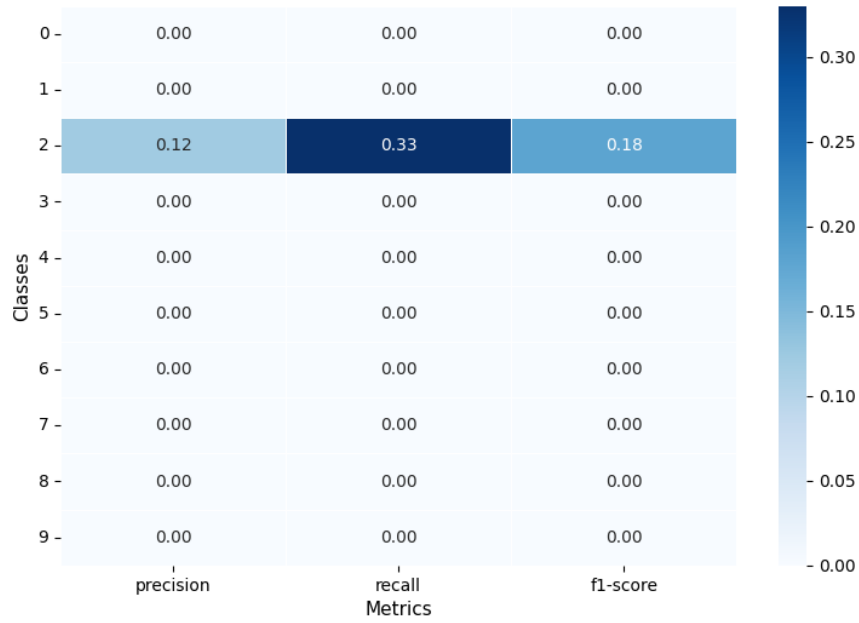


Confusion Matrix - MNIST Digits



Classification Report - Custom Digits

Accuracy: 0.03



Confusion Matrix - Custom Digits

