МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Індивідуальне завдання**

з дисципліни «**Нейронні мережі**»

**Виконав:**

студент гр. БС-03

Затуловський Г. А.

**Перевірив:**

Доцент, КБМК ФБМІ Федорін Ілля Валерійович

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2023

**Зміст**

[**Вступ 3**](#_Toc154006764)

[**1. Теоритичні відомості 4**](#_Toc154006765)

[**1.1 Загальні відомості 4**](#_Toc154006766)

[**1.2 Завдання 5**](#_Toc154006767)

[**1.3 Опис датасет 5**](#_Toc154006768)

[**1.4 Особливості датасету 6**](#_Toc154006769)

[**1.5 Опис проблематик 7**](#_Toc154006770)

[**2. Проектування 9**](#_Toc154006771)

[**2.1 Опис додатку 9**](#_Toc154006772)

[**2.2 Код програми 12**](#_Toc154006773)

[**2.3 Результати 15**](#_Toc154006774)

[**2.4 Проведення експерементів 20**](#_Toc154006775)

[**2.5 Аналіз результатів 43**](#_Toc154006776)

[**Висновок 45**](#_Toc154006777)

[**Список використаних джерел 46**](#_Toc154006778)

# Вступ

Пневмонія, яка є однією з найпоширеніших та серйозних інфекційних захворювань легенів, вимагає точної та швидкої діагностики для забезпечення ефективного лікування. У сучасній медицині обробка медичних зображень за допомогою нейронних мереж набуває все більшого значення, забезпечуючи нові можливості для автоматизації процесу діагностики.

Ця розробоча графічна робота фокусується на створенні та оптимізації нейронної мережі для бінарної класифікації рентгенівських зображень грудної клітини, розрізняючи здорових пацієнтів від тих, хто страждає від пневмонії. Завдання такого роду є критичним для раннього виявлення та лікування пневмонії, зокрема в умовах зростаючого обсягу медичних даних та потреби в ефективних та автоматизованих методах діагностики.

Процес створення та налаштування нейронної мережі для класифікації зображень рентгенівських знімків передбачає використання передових методів глибокого навчання, що дозволяє автоматично виділяти та аналізувати характеристики, властиві здоров'ю та захворюванням легень. Цей підхід розширює можливості медичної діагностики, забезпечуючи швидкий та об'єктивний аналіз рентгенівських зображень.

Метою даної графічної роботи є дослідження та реалізація нейронної мережі для бінарної класифікації, спрямованої на поліпшення точності та швидкості діагностики пневмонії на рентгенівських знімках грудної клітини. Відповідний успіх у цьому напрямку може відкрити нові можливості для покращення якості медичної практики та підвищення швидкості та доступності діагностики пацієнтів з підозрою на пневмонію.

# Теоритичні відомості

## 1.1 Загальні відомості

Пневмонія - це запалення легень, яке виникає, коли інфекція або інший подразник пошкоджує повітряні мішечки легенів (альвеоли). Альвеоли - це крихітні мішечки в легенях, де відбувається обмін кисню та вуглекислого газу. Коли альвеоли запалені, вони не можуть ефективно обмінювати гази, що може призвести до задишки, низького рівня кисню в крові та інших проблем. Пневмонія може бути викликана різними мікроорганізмами, включаючи бактерії, віруси, грибки та паразити. Інші поширені збудники пневмонії включають віруси грипу, парагрипу та риновірусу. Пневмонія може бути гострою або хронічною. Гостра пневмонія зазвичай розвивається раптово і триває кілька днів або тижнів. Хронічна пневмонія триває довше трьох місяців і може бути викликана різними факторами, включаючи інфекції, захворювання легенів та інші захворювання.

Бінарна класифікація — це один з основних типів задач у машинному навчанні, де модель навчається визначати належність об'єкта до одного з двох можливих класів. У контексті задачі діагностики пневмонії на рентгенівських знімках, це може виглядати як визнач

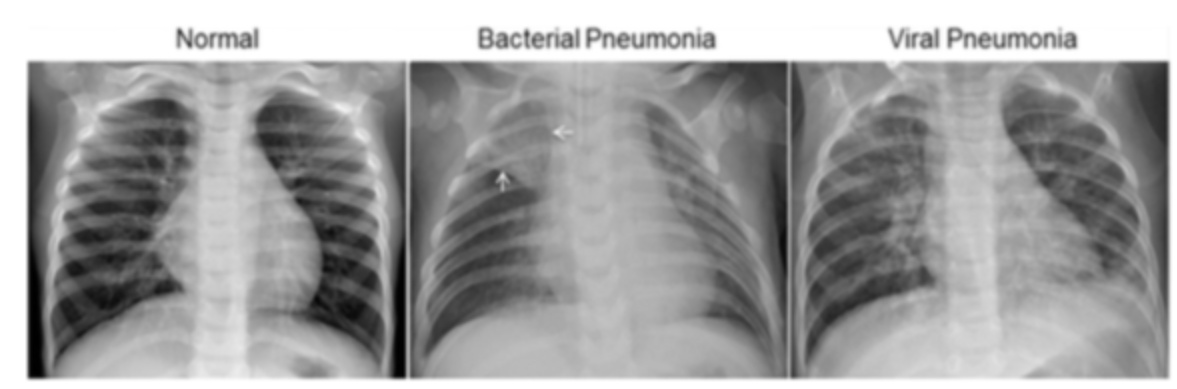
Згорткові нейронні мережі (Convolutional Neural Networks, CNN) – це клас глибоких нейронних мереж, які були спеціально розроблені для обробки даних з мультиперспективною структурою (багатовимірною, багатоканальною, тощо). У контексті зображень мультиперспективна структура може бути матрицею пікселів, де кожен піксель може бути представлений трьома каналами кольору (червоний, зелений та синій). Однак CNN не обмежені лише обробкою зображень.

## 1.2 Завдання

Створити та навчити нейрону мережу, використавши датасет Chest X-Ray Images (Pneumonia) для аналізу та вирішення задачі бінарної класифікації зображень (Yes / No pneumonia)

## 1.3 Опис датасет

Данасет організовано в 3 папки (train, test, val) та містить підпапки для кожної категорії зображень (Пневмонія/Норма). Всього є 5863 рентгенограми (JPEG) та 2 категорії (Пневмонія/Норма). Рентгенограми грудної клітини (передньо-задня проекція) були взяті зі зворотних когорт педіатричних пацієнтів віком від одного до п'яти років з Медичного центру жінок та дітей Гуанчжоу. Усі рентгенограми грудної клітини були виконані в рамках звичайної клінічної допомоги пацієнтам. Для аналізу рентгенограм грудної клітини всі рентгенограми спочатку перевірялися на контроль якості з видаленням усіх зображень низької якості або нечитабельних. Потім діагнози для зображень були оцінені двома експертами-лікарями, перш ніж бути дозволеними для навчання системи штучного інтелекту. Щоб врахувати будь-які помилки оцінювання, контрольний набір також був перевірений третім експертом.

Ілюстративні приклади рентгенівських знімків грудної клітки у пацієнтів із пневмонією:

## 1.4 Особливості датасету

Цей датасет містить зображення ОКТ (оптичної когерентної томографії) та рентгенограми грудної клітини, які описані та проаналізовані в науковій роботі "Deep learning-based classification and referral of treatable human diseases" (Класифікація та направлення лікувальних захворювань людини на основі глибокого навчання).

Вміст датасету:

* Зображення ОКТ:

Поділені на тренувальний набір та тестовий набір незалежних пацієнтів. Кожне зображення має назву у форматі (disease)-(randomized patient ID)-(image number by this patient).

Зображення групуються в 4 директорії:

CNV: Хориоїдальна неоваскуляризація.

DME: Діабетична макулярна набряклість.

DRUSEN Друзи сітківки.

NORMAL Нормальні зображення.

* Рентгенограми грудної клітини:

Опис та формат цих зображень наразі невідомий.

Датасет створили для дослідження можливості використання методів глибокого навчання для автоматичної класифікації захворювань очей та пневмонії на основі ОКТ та рентгенограм грудної клітини відповідно.

## 1.5 Опис проблематик

Датасет "Chest X-Ray Pneumonia може бути використаний для вирішення наступних проблем:

Розробка алгоритмів машинного навчання для автоматичної діагностики пневмонії:

* Данісет може служити основою для навчання моделей машинного навчання, які автоматично визначають наявність чи відсутність пневмонії на рентгенівських знімках.
* Можливо використовувати методи глибокого навчання, такі як нейронні мережі, для створення алгоритмів, які здатні точно розпізнавати ознаки пневмонії на зображеннях.

Покращення точності діагностики пневмонії:

* Використання даного датасету дозволяє оптимізувати алгоритми та моделі для забезпечення високої точності діагностики пневмонії.
* Можливо розширити набір ознак для аналізу, включаючи текстурні характеристики, форму та розмір уражених областей.

Зменшення витрат на діагностику пневмонії:

* Автоматизація діагностики за допомогою алгоритмів машинного навчання може допомогти зменшити час, необхідний для обробки рентгенівських знімків та встановлення діагнозу.
* Можливо визначити оптимальні протоколи скринінгу та використання алгоритмів для попередження затримок у діагностиці, що може сприяти швидшому та ефективнішому лікуванню.

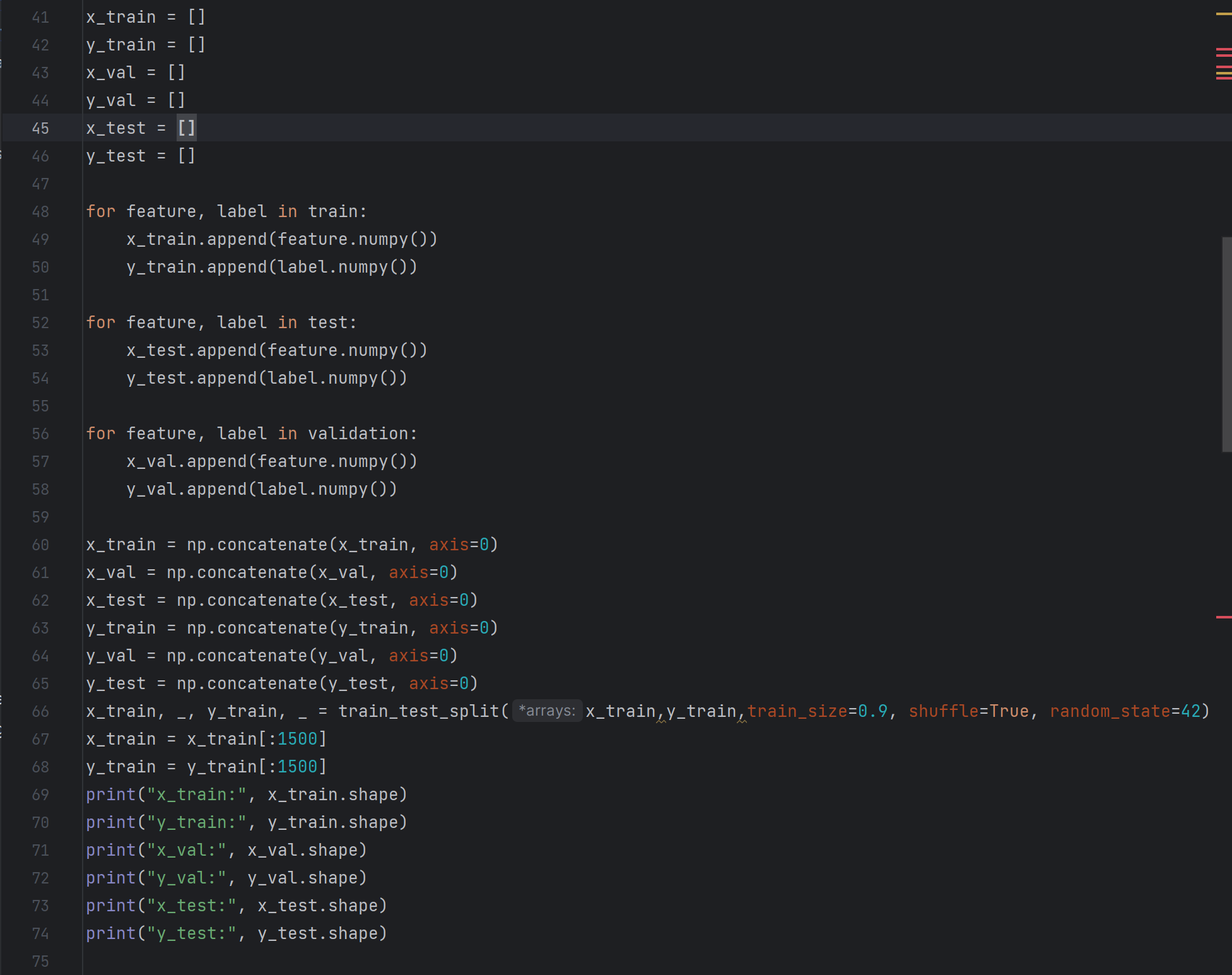
# 2. Проектування

## 2.1 Опис додатку

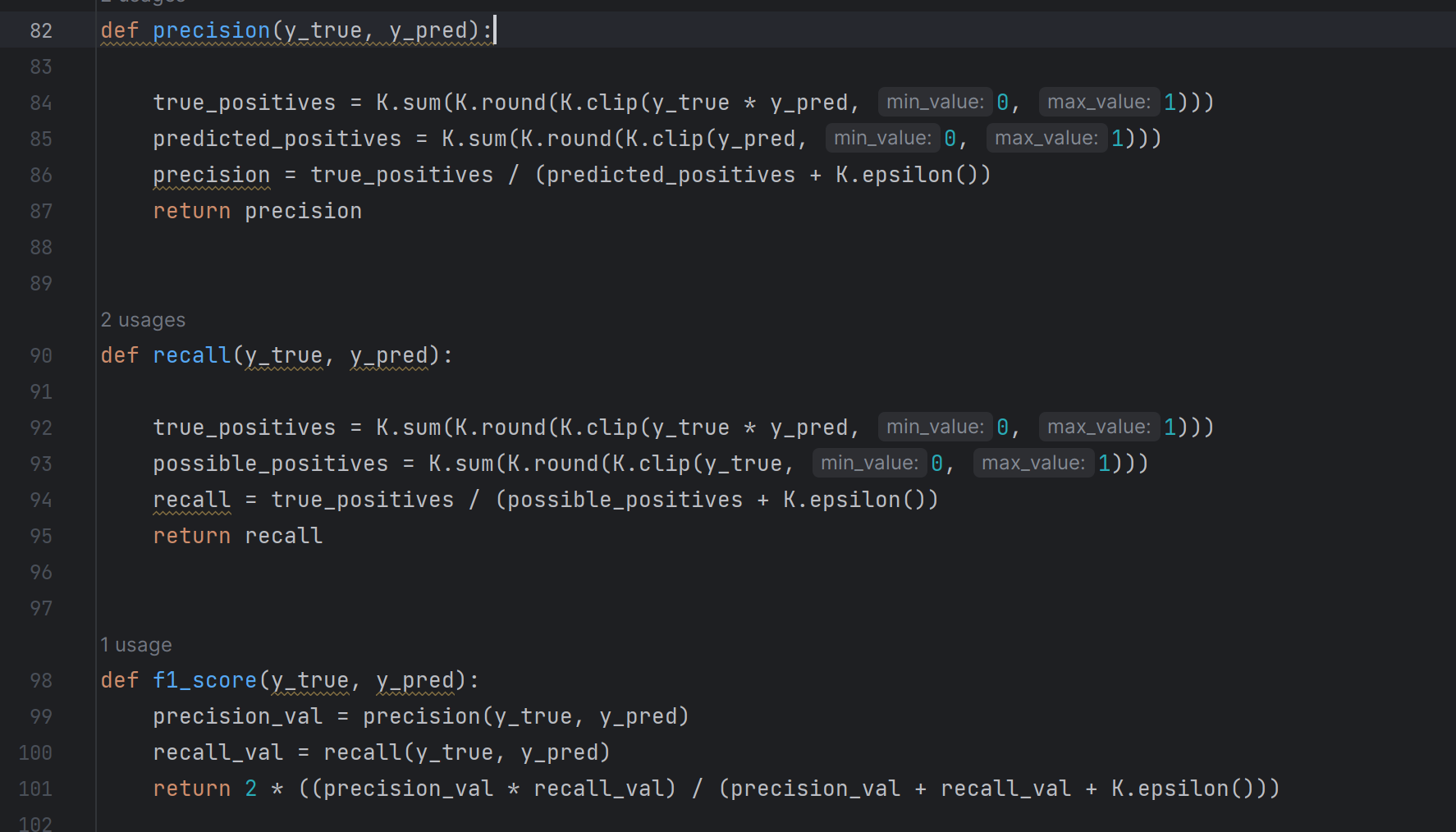
Завантаження тренувальних, валідаційних та тестових наборів даних



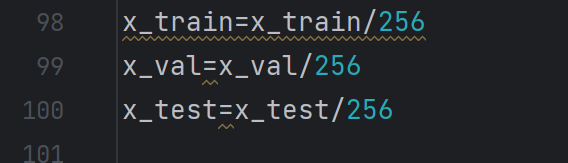
Перетворення даних в масиви:



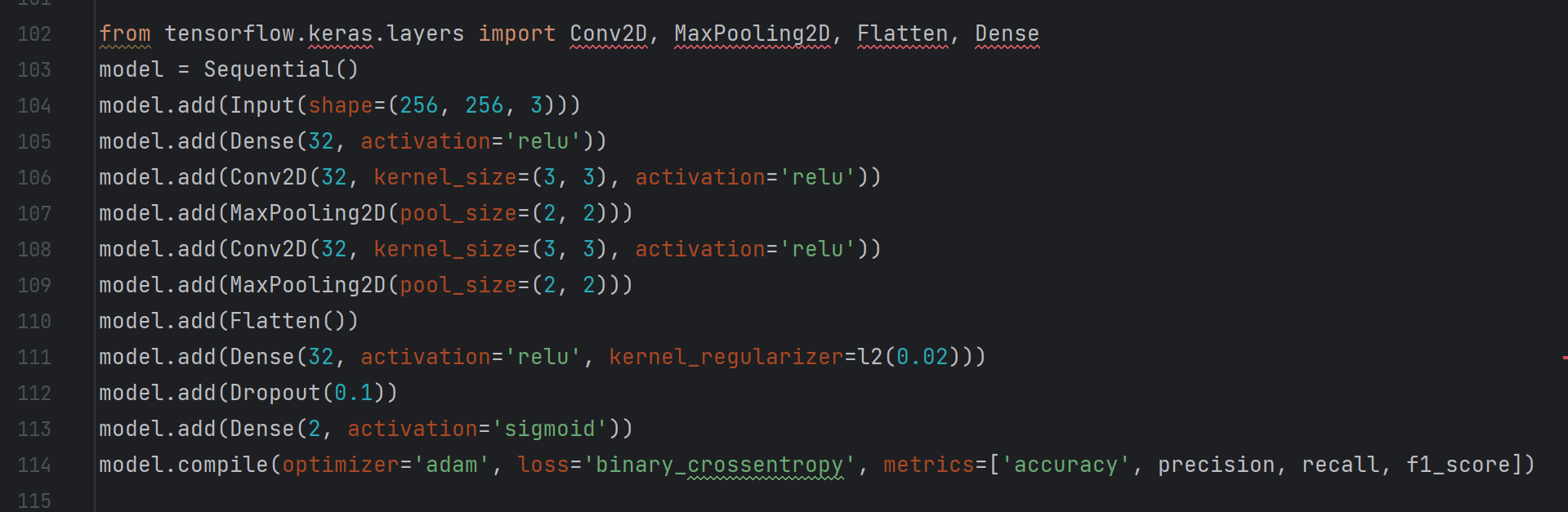
Cтворення функції метрик (F1-оцінки, точності (precision) та відгуку (recall))



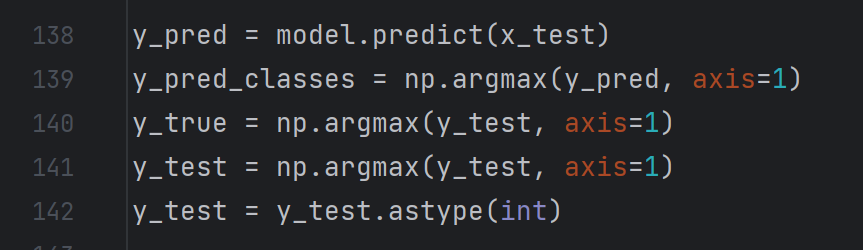
Масштабування піксельних значень для наборів даних: нормалізація



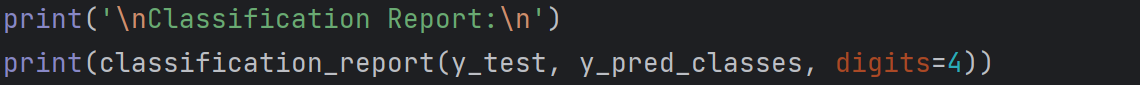
Створення моделі CNN



Передбачення моделі на тестовому наборі



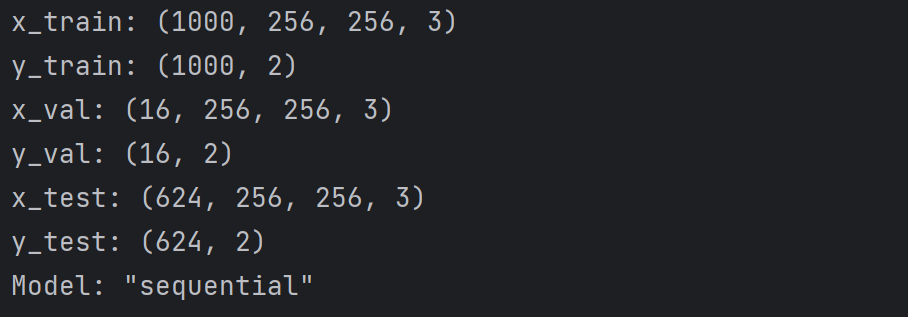
Розрахунок F1-оцінки, точності та відгуку



## 2.2 Код програми

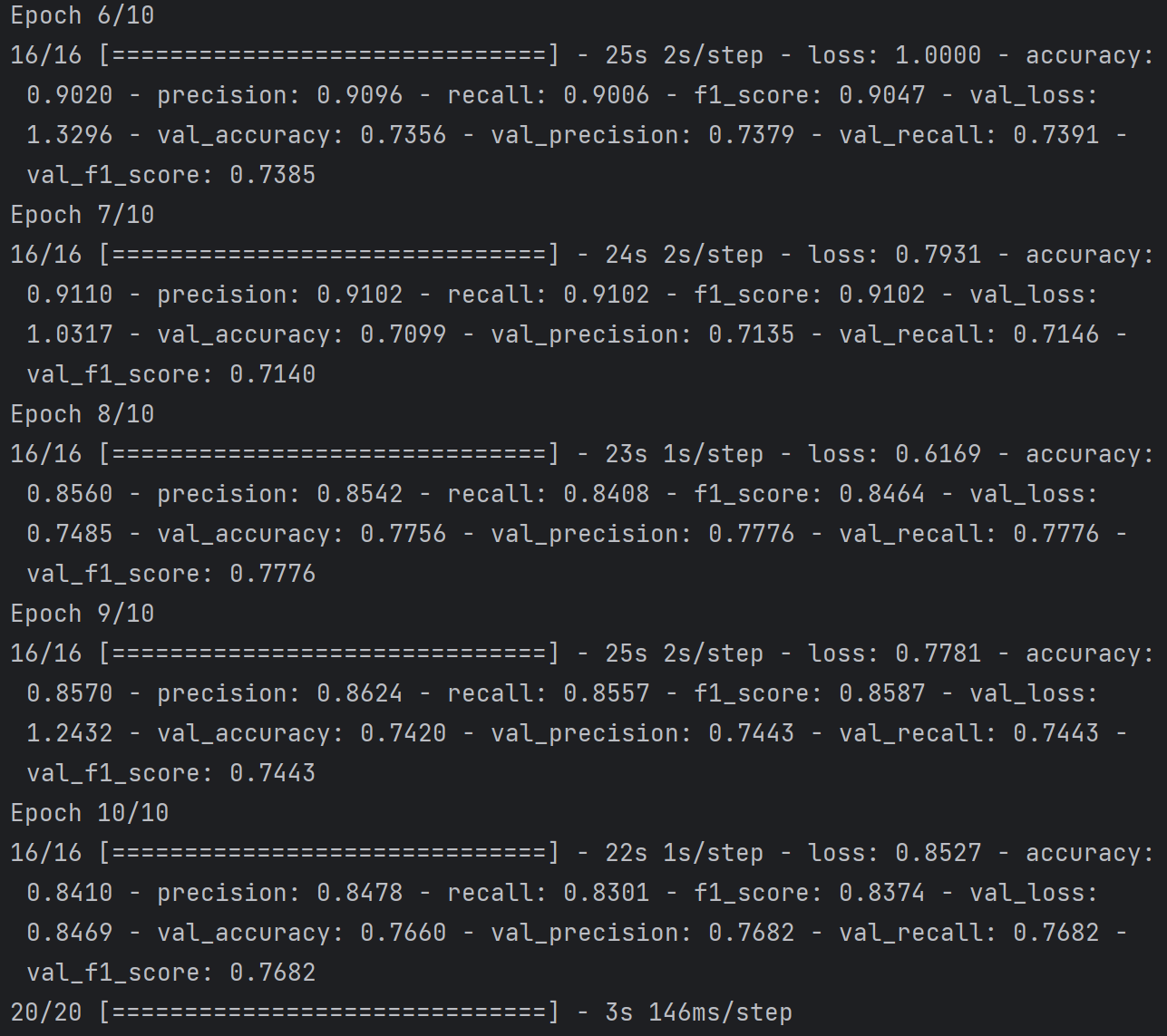
import numpy as np  
import tensorflow as tf  
from tensorflow import keras  
import matplotlib.pyplot as plt  
import seaborn as sns  
sns.set(style= "darkgrid", color\_codes = True)  
from tensorflow.keras.applications import VGG16  
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dense, Flatten, Dropout, Input  
from keras.regularizers import l2  
from tensorflow.keras.metrics import Precision, Recall, BinaryAccuracy  
from sklearn.metrics import roc\_curve, auc  
from tensorflow.keras.preprocessing.image import load\_img  
import warnings  
warnings.filterwarnings('ignore')  
from keras import backend as K  
  
img\_width, img\_height = 256, 256  
batchsize = 32  
epochs = 10  
num\_of\_class = 2  
  
train = keras.utils.image\_dataset\_from\_directory(  
 directory='chest\_xray/train',  
 labels='inferred',  
 label\_mode='categorical',  
 batch\_size=32,  
 image\_size=(256, 256))  
  
validation = keras.utils.image\_dataset\_from\_directory(  
 directory='chest\_xray/val',  
 labels='inferred',  
 label\_mode='categorical',  
 batch\_size=32,  
 image\_size=(256, 256))  
  
test = keras.utils.image\_dataset\_from\_directory(  
 directory='chest\_xray/test',  
 labels='inferred',  
 label\_mode='categorical',  
 batch\_size=32,  
 image\_size=(256, 256))  
  
  
x\_train = []  
y\_train = []  
x\_val = []  
y\_val = []  
x\_test = []  
y\_test = []  
  
for feature, label in train:  
 x\_train.append(feature.numpy())  
 y\_train.append(label.numpy())  
  
for feature, label in test:  
 x\_test.append(feature.numpy())  
 y\_test.append(label.numpy())  
  
for feature, label in validation:  
 x\_val.append(feature.numpy())  
 y\_val.append(label.numpy())  
  
# Concatenate the lists to get the full 'x' and 'y' arrays  
x\_train = np.concatenate(x\_train, axis=0)  
x\_val = np.concatenate(x\_val, axis=0)  
x\_test = np.concatenate(x\_test, axis=0)  
y\_train = np.concatenate(y\_train, axis=0)  
y\_val = np.concatenate(y\_val, axis=0)  
y\_test = np.concatenate(y\_test, axis=0)  
  
x\_train = x\_train[:1000]  
y\_train = y\_train[:1000]  
print("x\_train:", x\_train.shape)  
print("y\_train:", y\_train.shape)  
print("x\_val:", x\_val.shape)  
print("y\_val:", y\_val.shape)  
print("x\_test:", x\_test.shape)  
print("y\_test:", y\_test.shape)  
  
def precision(y\_true, y\_pred):  
  
 true\_positives = K.sum(K.round(K.clip(y\_true \* y\_pred, 0, 1)))  
 predicted\_positives = K.sum(K.round(K.clip(y\_pred, 0, 1)))  
 precision = true\_positives / (predicted\_positives + K.epsilon())  
 return precision  
  
  
def recall(y\_true, y\_pred):  
  
 true\_positives = K.sum(K.round(K.clip(y\_true \* y\_pred, 0, 1)))  
 possible\_positives = K.sum(K.round(K.clip(y\_true, 0, 1)))  
 recall = true\_positives / (possible\_positives + K.epsilon())  
 return recall  
  
  
def f1\_score(y\_true, y\_pred):  
 precision\_val = precision(y\_true, y\_pred)  
 recall\_val = recall(y\_true, y\_pred)  
 return 2 \* ((precision\_val \* recall\_val) / (precision\_val + recall\_val + K.epsilon()))  
  
# Pixel Value Scaling for Datasets: Normalizing and Standardizing the Data  
x\_train=x\_train/256  
x\_val=x\_val/256  
x\_test=x\_test/256  
  
  
model = Sequential()  
model.add(Input(shape=(256, 256, 3)))  
model.add(Dense(32, activation='relu'))  
model.add(Flatten())  
model.add(Dense(32, activation='relu', kernel\_regularizer=l2(0.02)))  
model.add(Dropout(0.1))  
model.add(Dense(2, activation='sigmoid'))  
model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy', precision, recall, f1\_score])  
model.summary()  
  
  
history = model.fit(x\_train, y\_train, batch\_size=64,epochs=10,validation\_data=(x\_test, y\_test))  
# Графік точності  
plt.plot(history.history['accuracy'])  
plt.plot(history.history['val\_accuracy'])  
plt.title('Точність моделі')  
plt.ylabel('Точність')  
plt.xlabel('Епоха')  
plt.legend(['Навчання', 'Тест'], loc='upper left')  
plt.show()  
# Графік функції втрат  
plt.plot(history.history['loss'])  
plt.plot(history.history['val\_loss'])  
plt.title('Функція втрат')  
plt.ylabel('Втрати')  
plt.xlabel('Епоха')  
plt.legend(['Навчання', 'Тест'], loc='upper left')  
plt.show()  
from sklearn.metrics import confusion\_matrix, classification\_report  
import seaborn as sns  
  
y\_pred = model.predict(x\_test)  
y\_pred\_classes = np.argmax(y\_pred, axis=1)  
  
y\_true = np.argmax(y\_test, axis=1)  
  
  
  
y\_test = np.argmax(y\_test, axis=1)  
  
  
y\_test = y\_test.astype(int)  
  
print('Predicted y: \n',y\_pred\_classes)  
print('True y: \n',y\_test)  
conf\_mtx = confusion\_matrix(y\_test, y\_pred\_classes)  
plt.figure(figsize=(10,8))  
sns.heatmap(conf\_mtx, annot=True, fmt='d', cmap='Blues')  
plt.xlabel('Predicted Label')  
plt.ylabel('True Label')  
plt.title('Confusion Matrix')  
plt.show()  
print('\nClassification Report:\n')  
print(classification\_report(y\_test, y\_pred\_classes, digits=4))

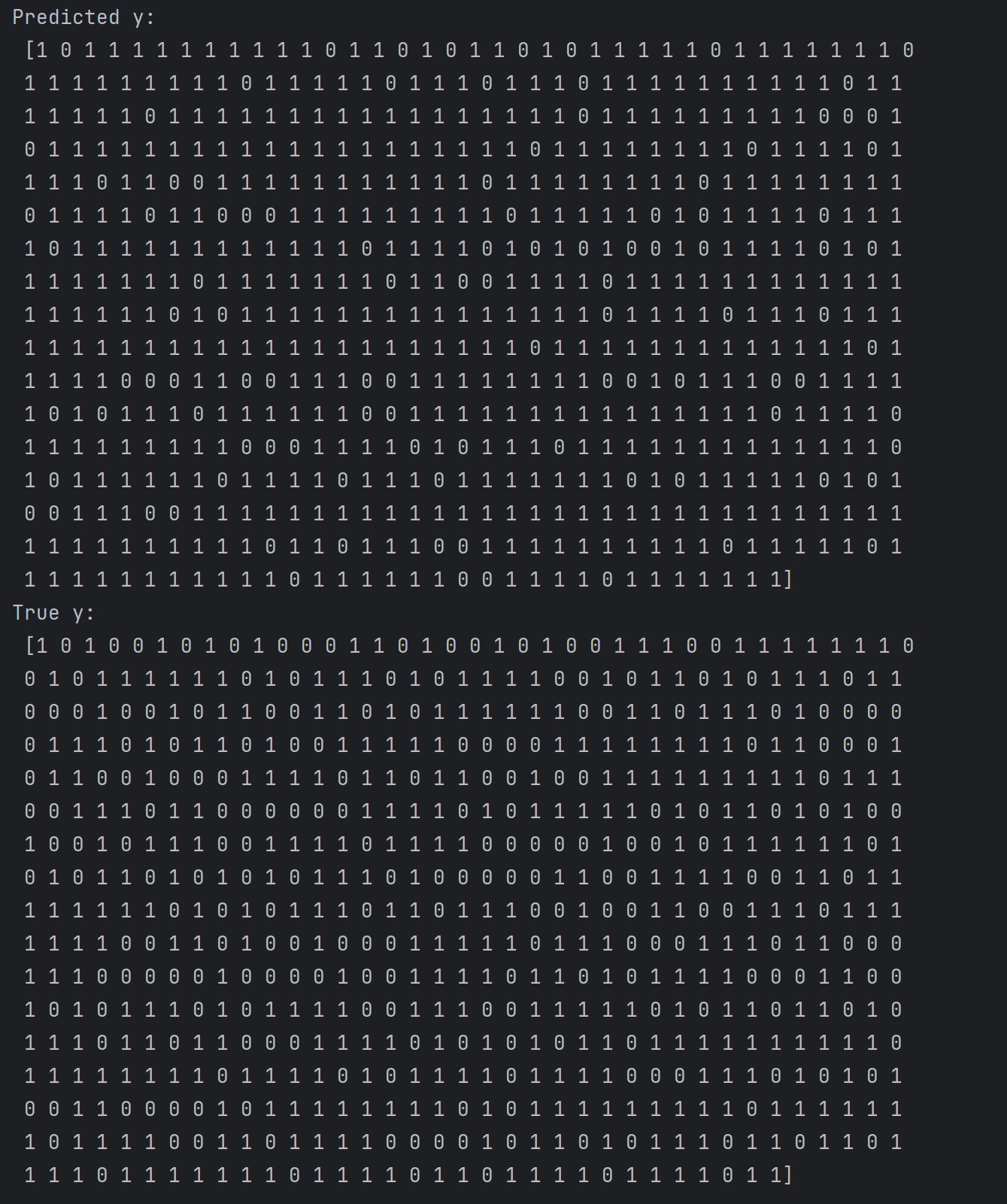
## 2.3 Результати

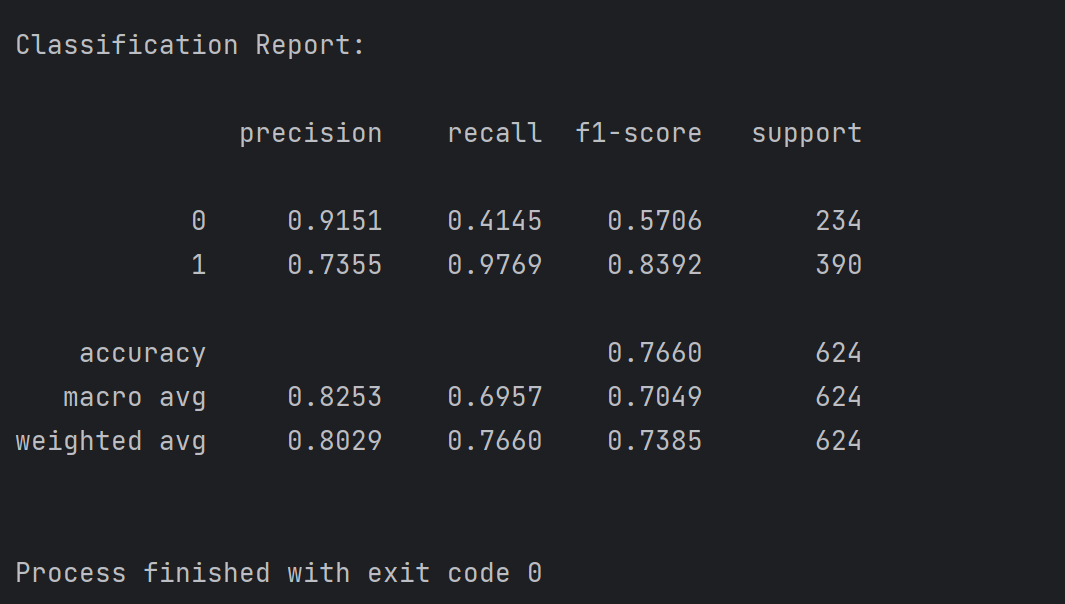


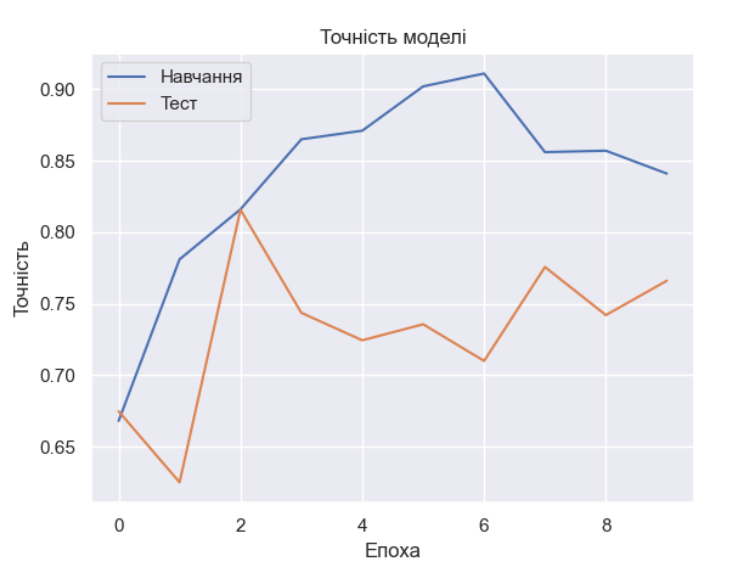


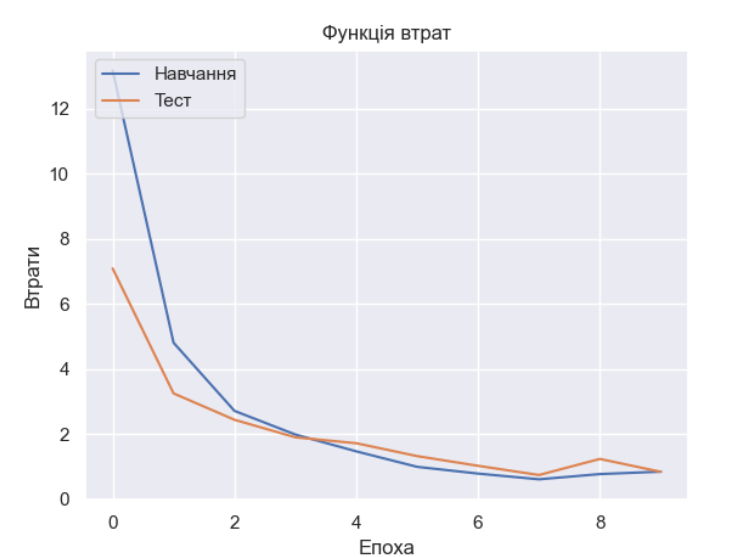


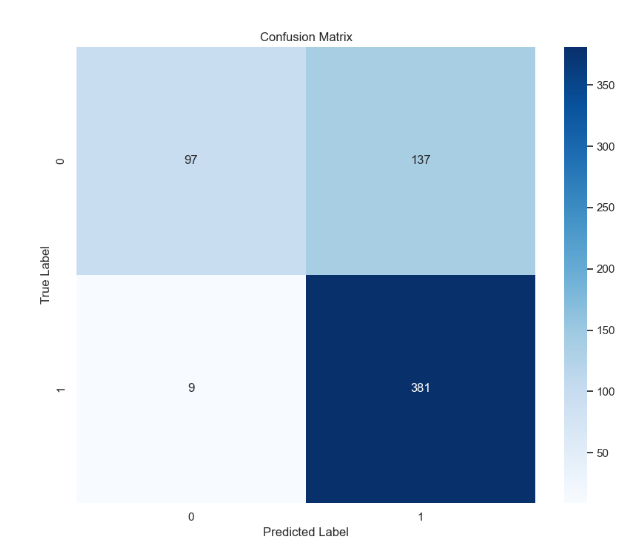






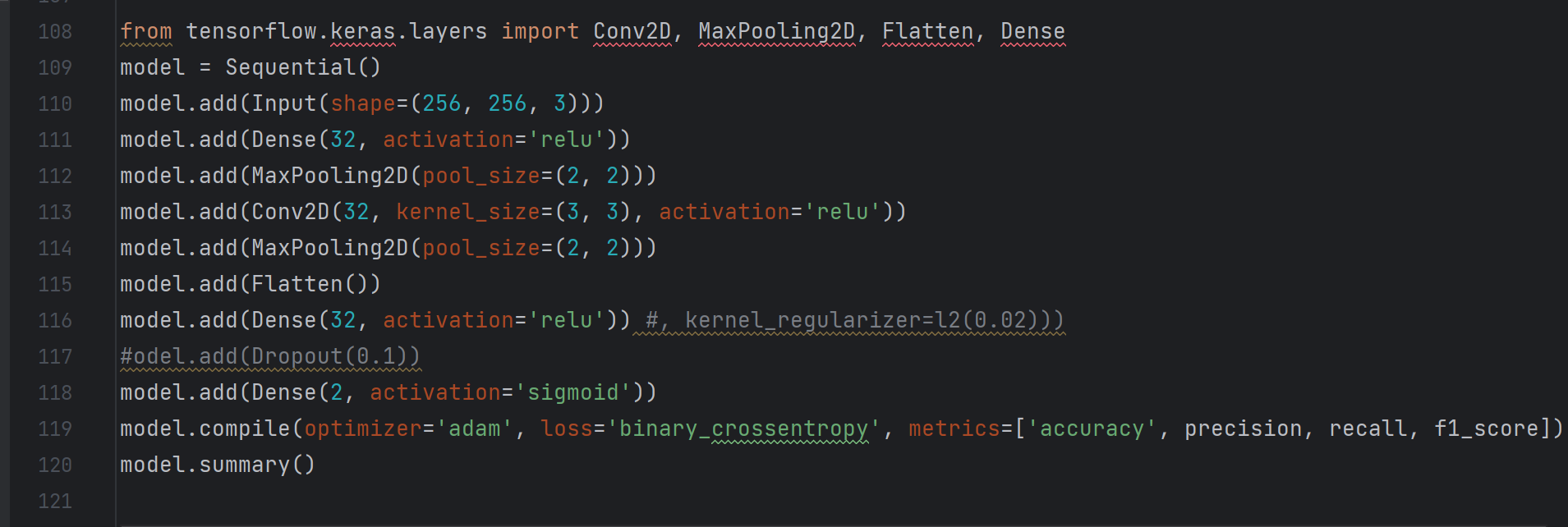




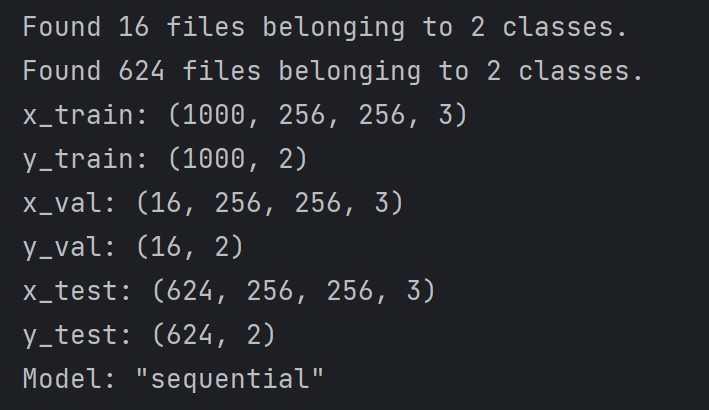


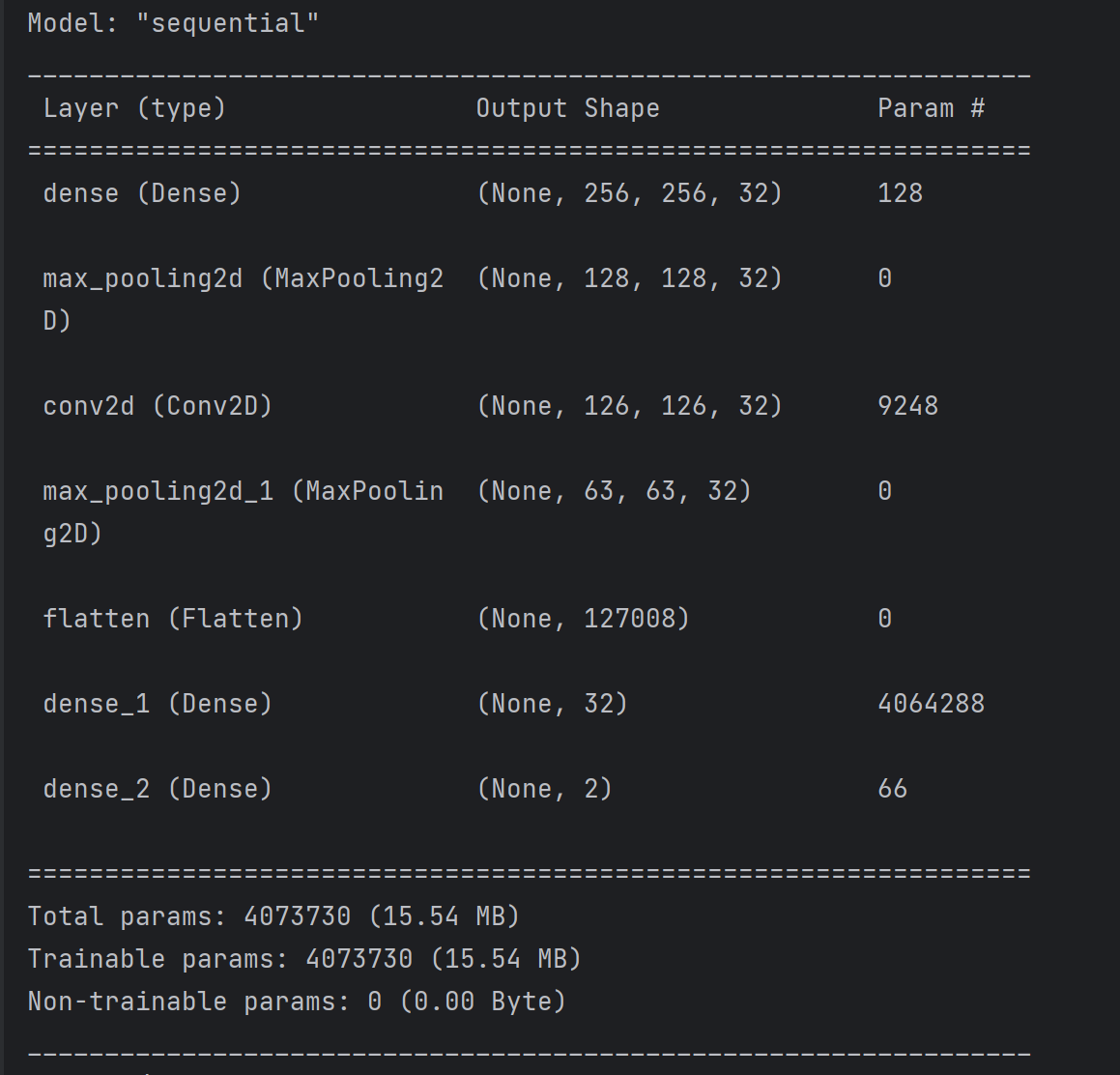
## 2.4 Проведення експерементів

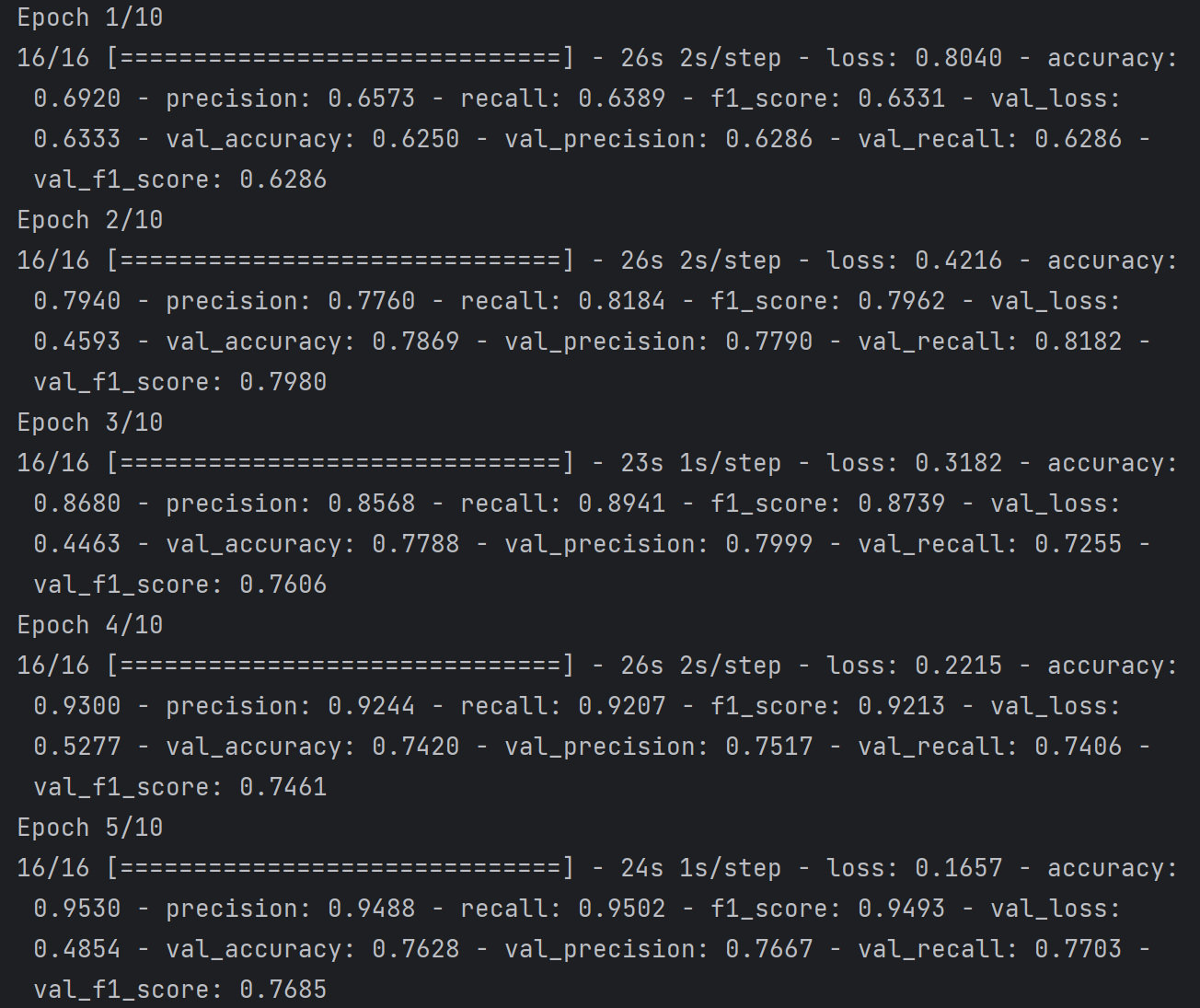
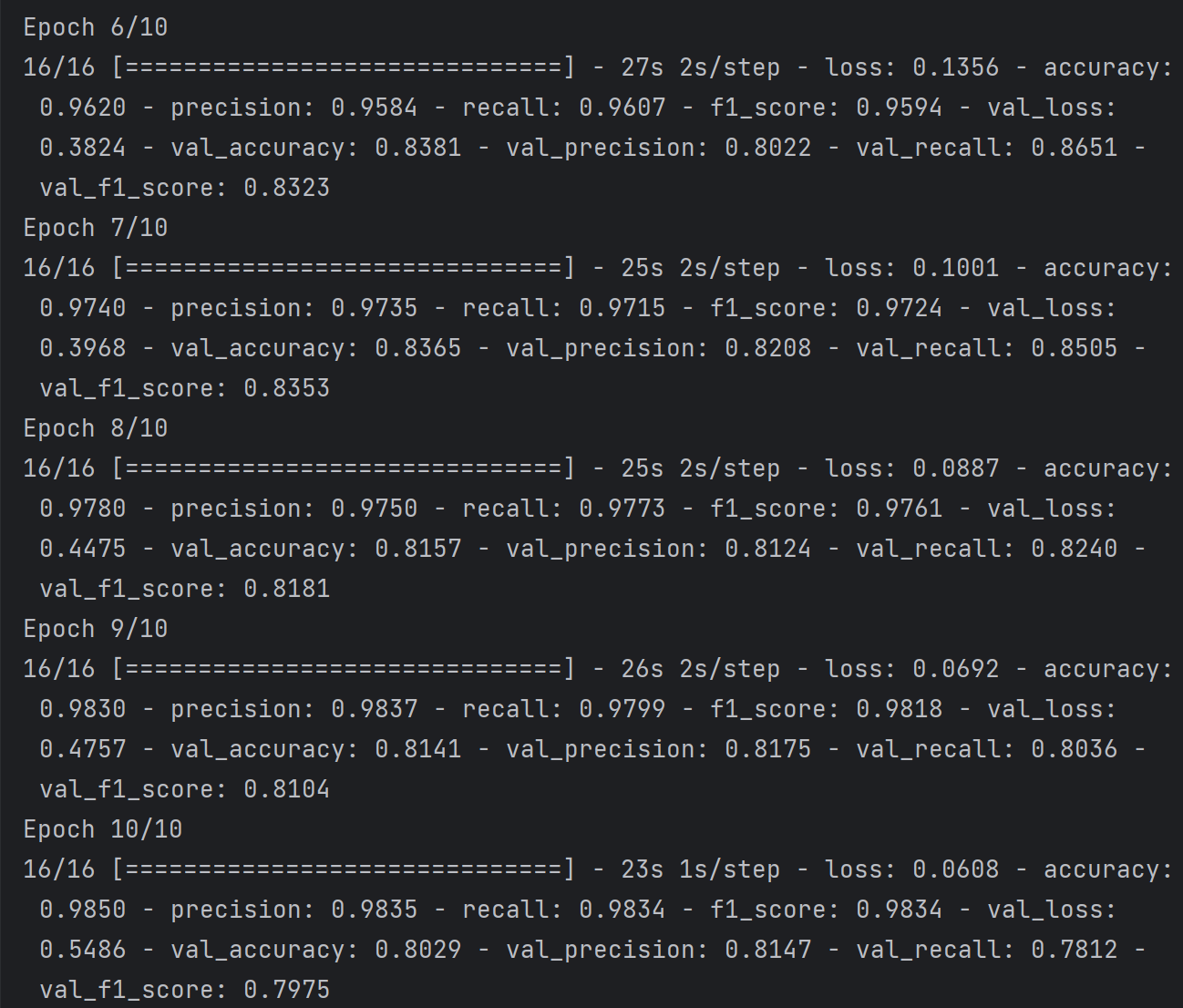
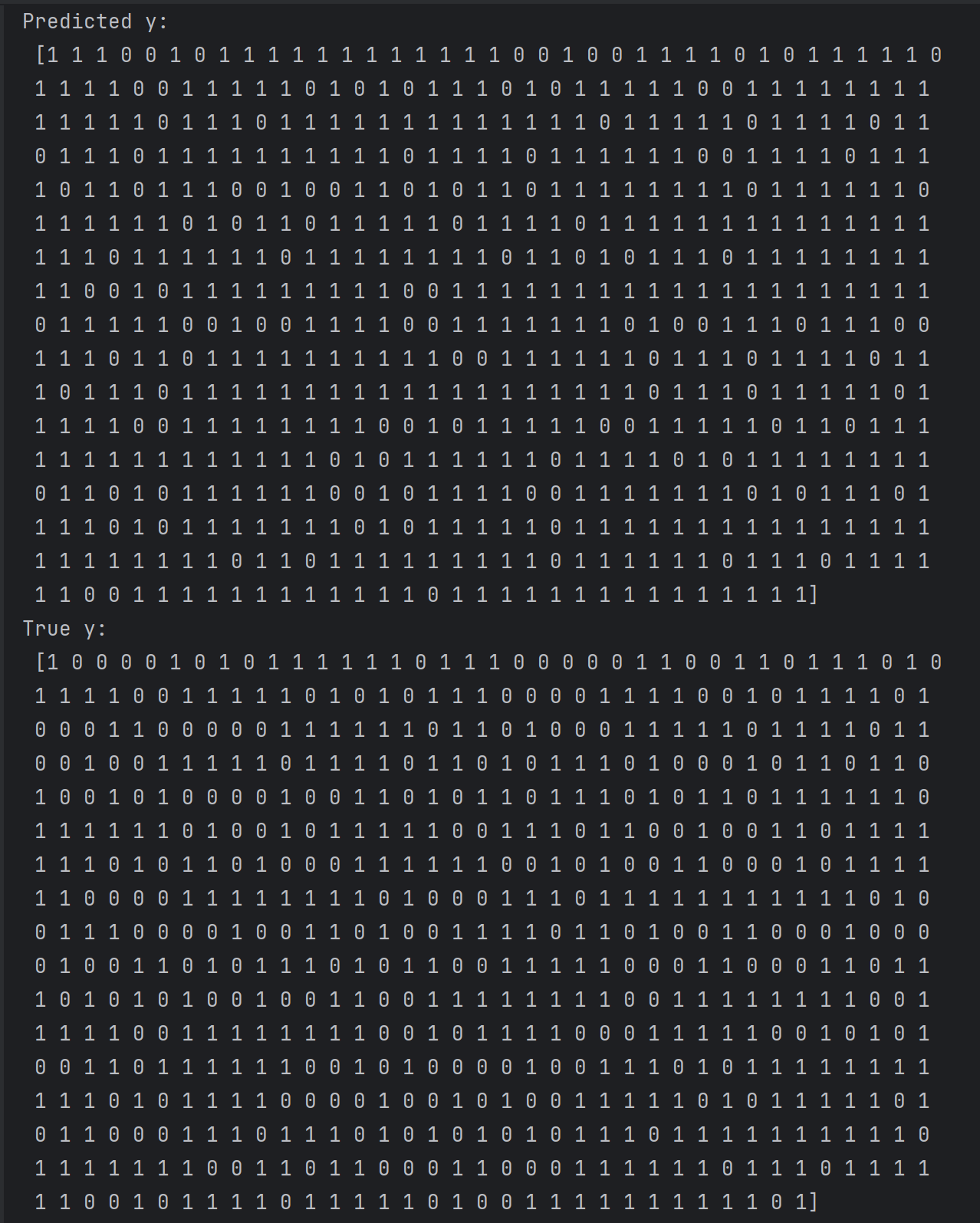
Для покращення результатів спробуємо додати до нейроної мережі різновиди пулінгу та приберемо регуляцію:

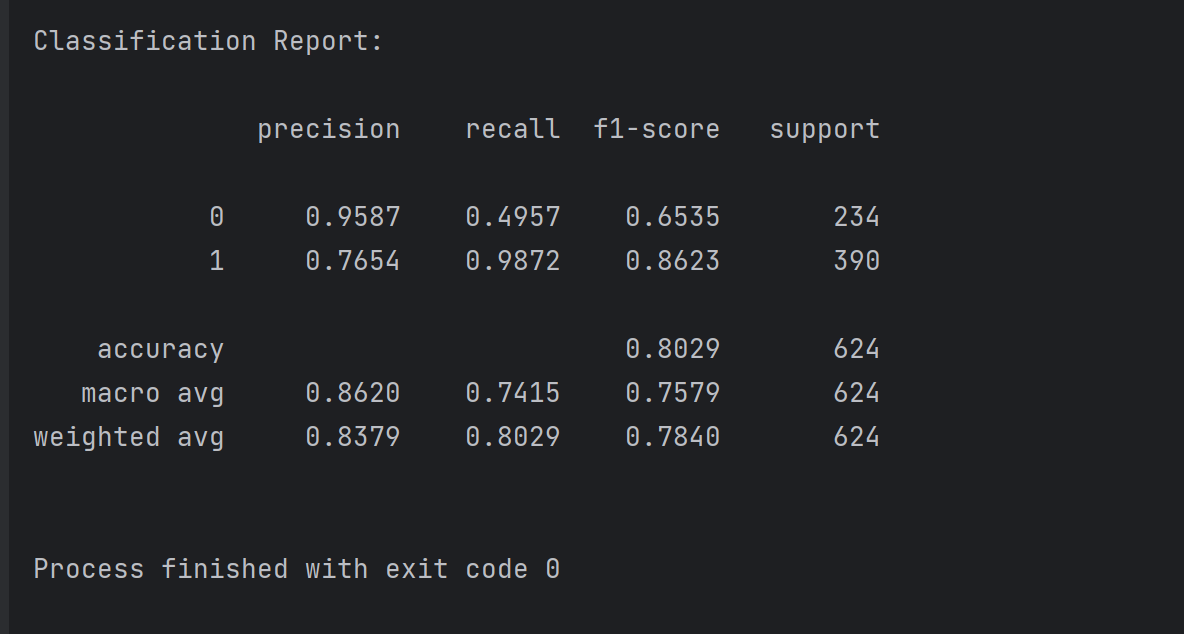


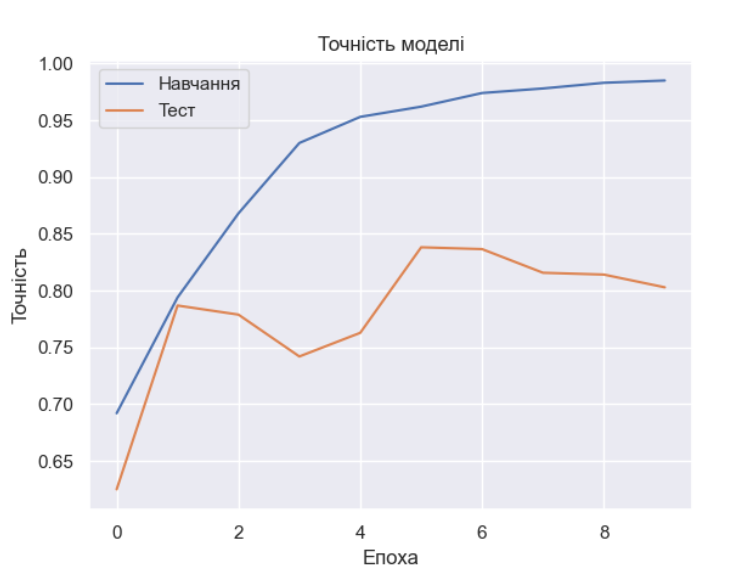
Результати:

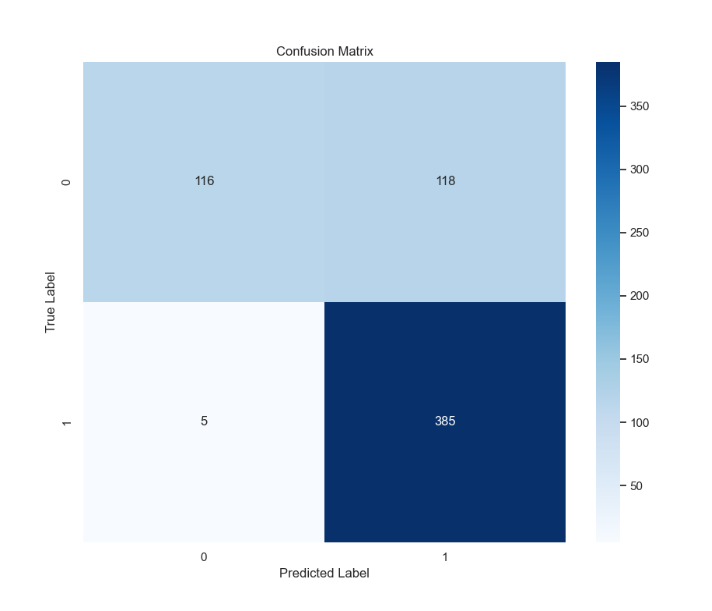




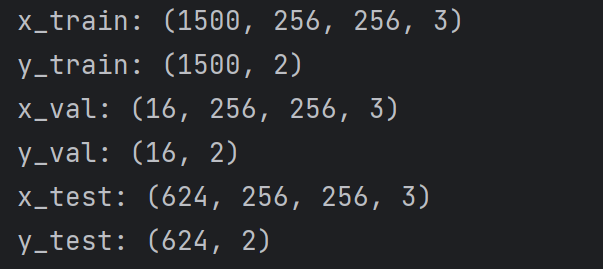


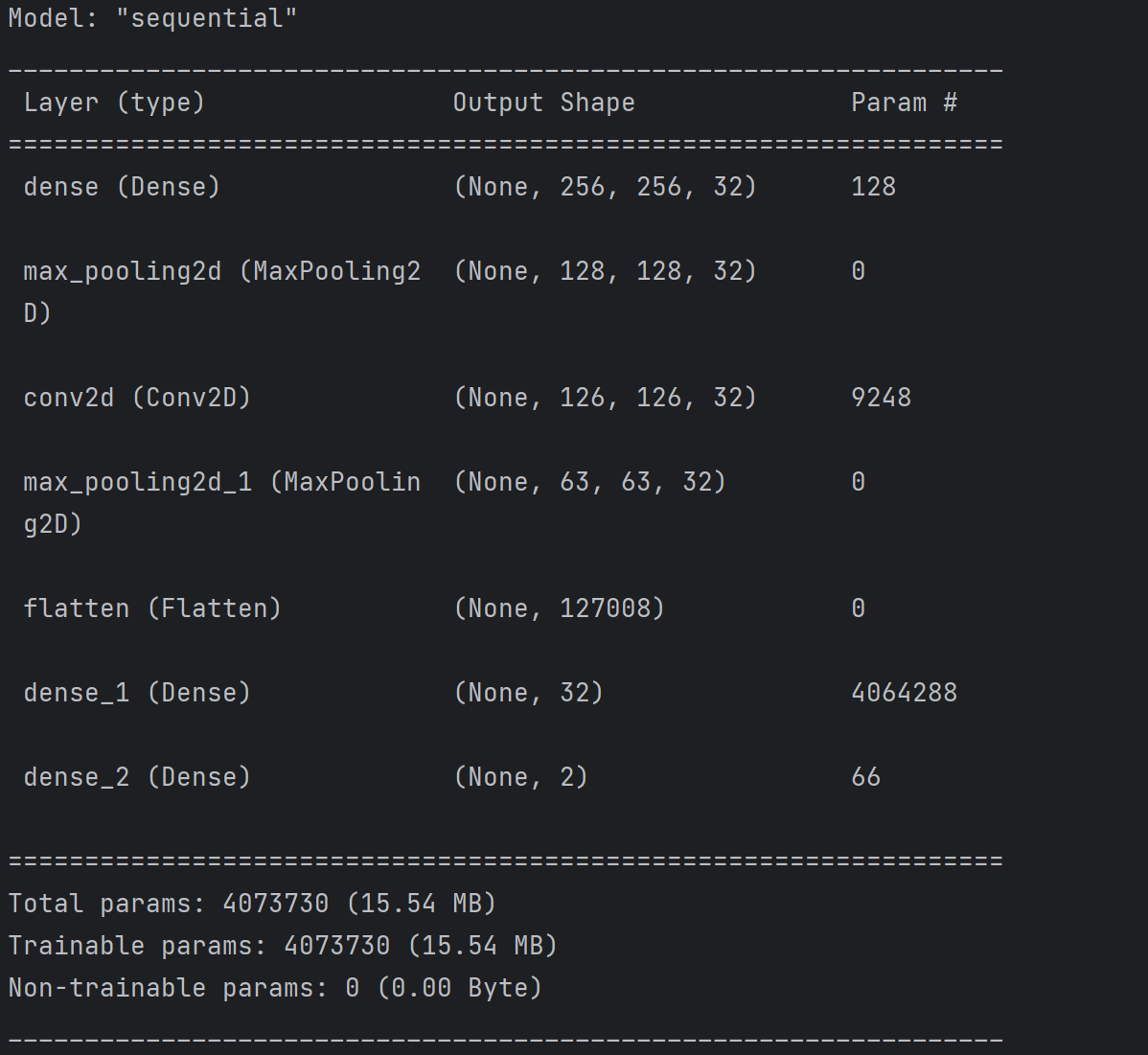


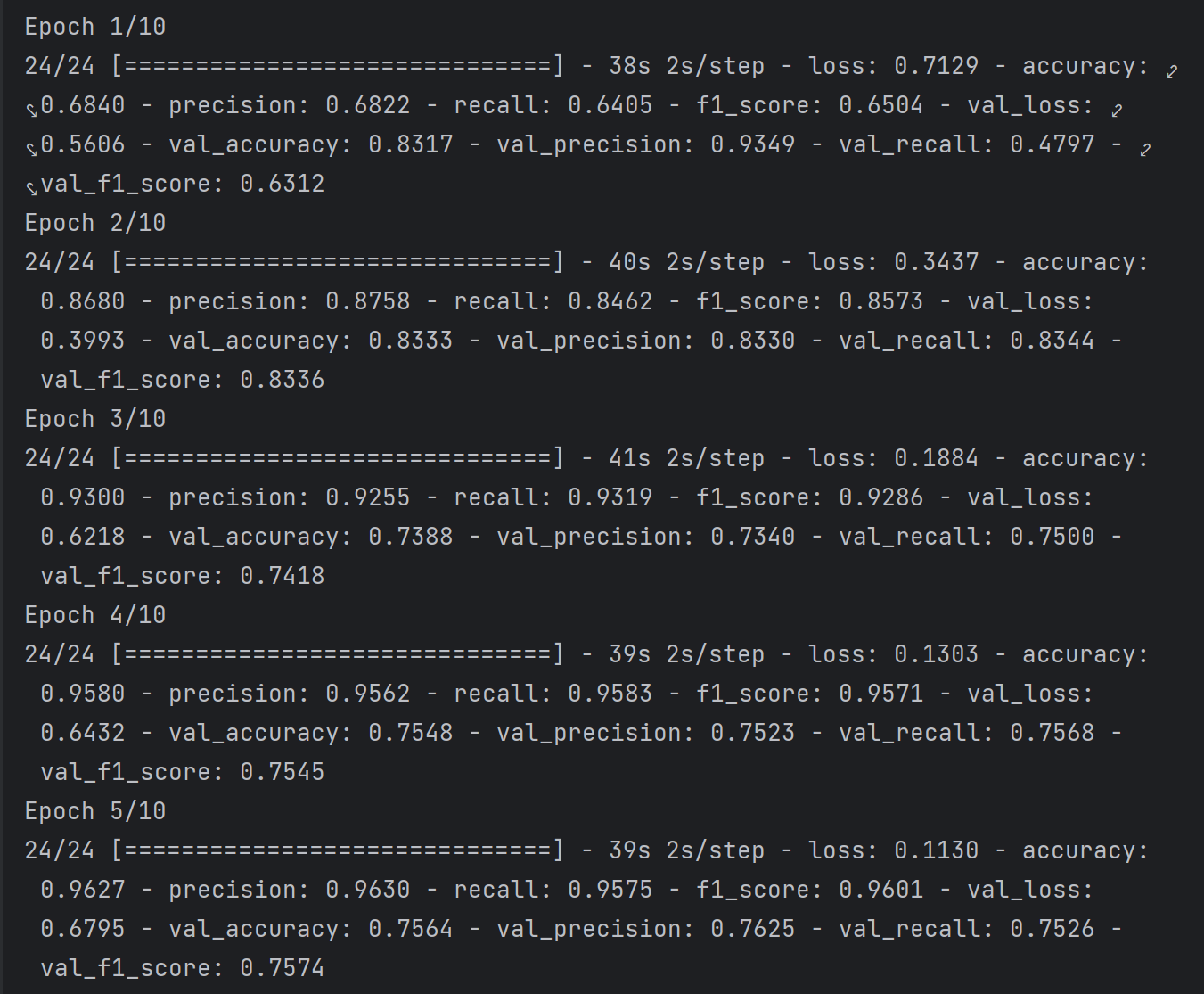


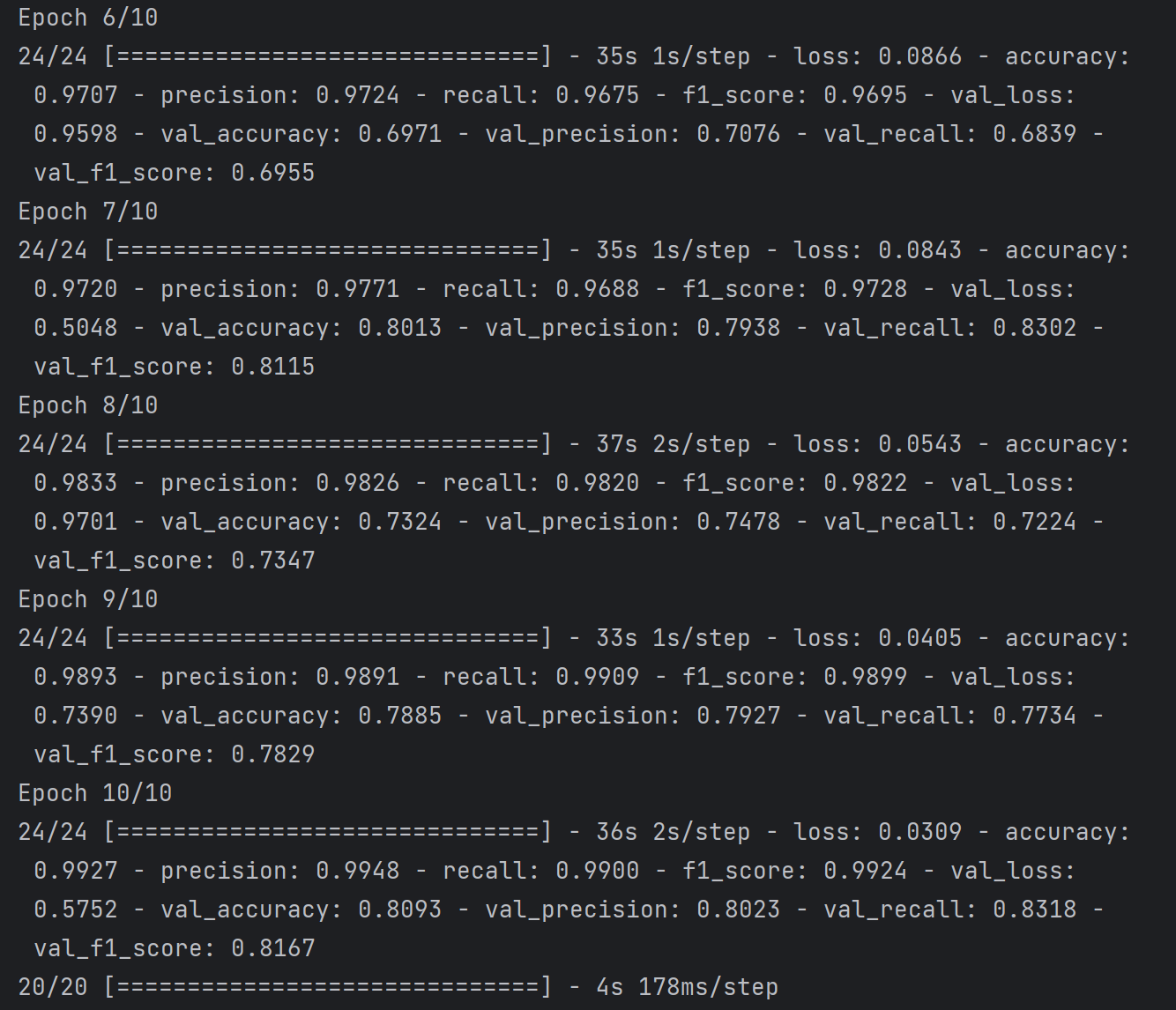
Cпробуємо збільшити кількість тренувальної збірки з 1000 до 1500:

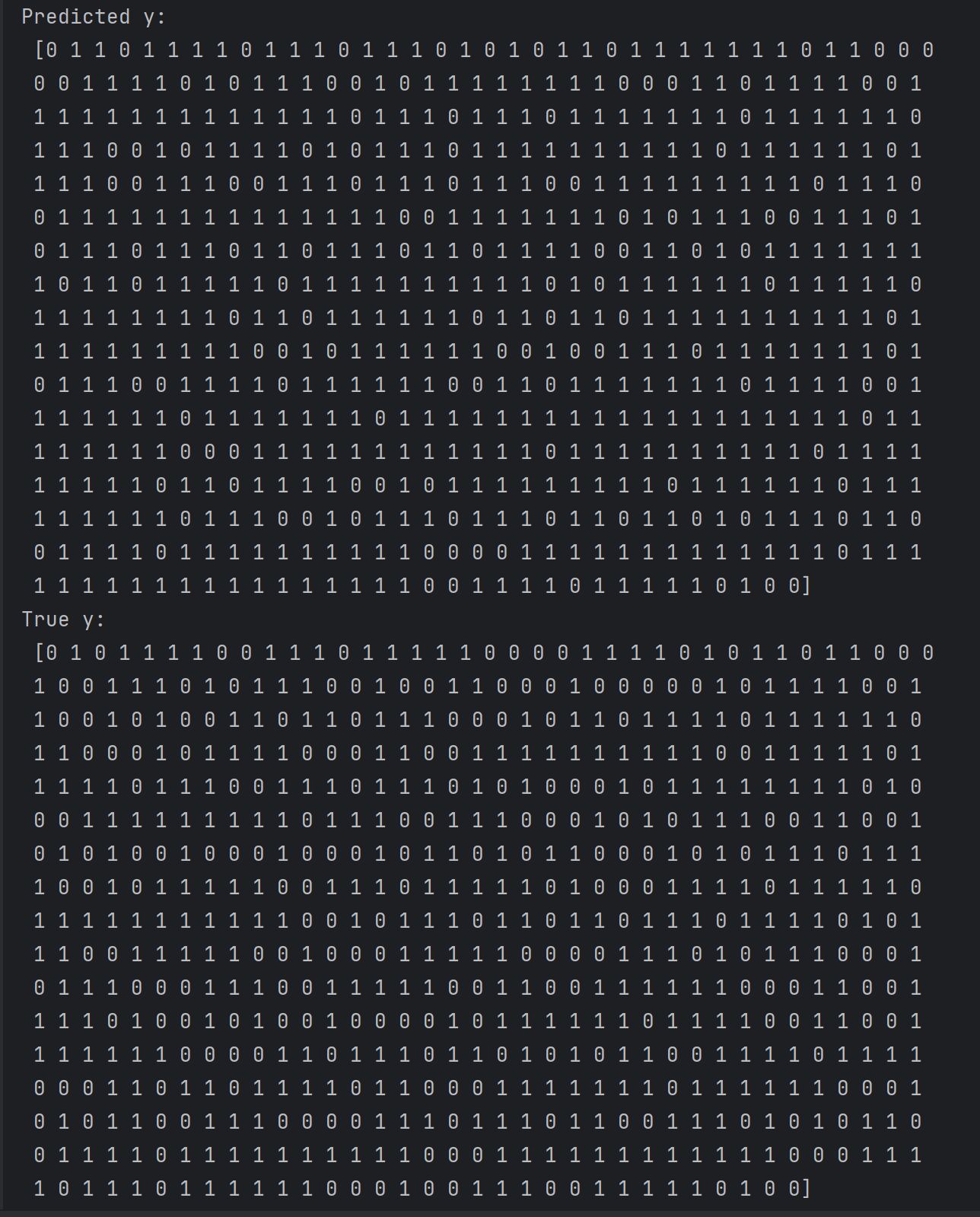
Результат:

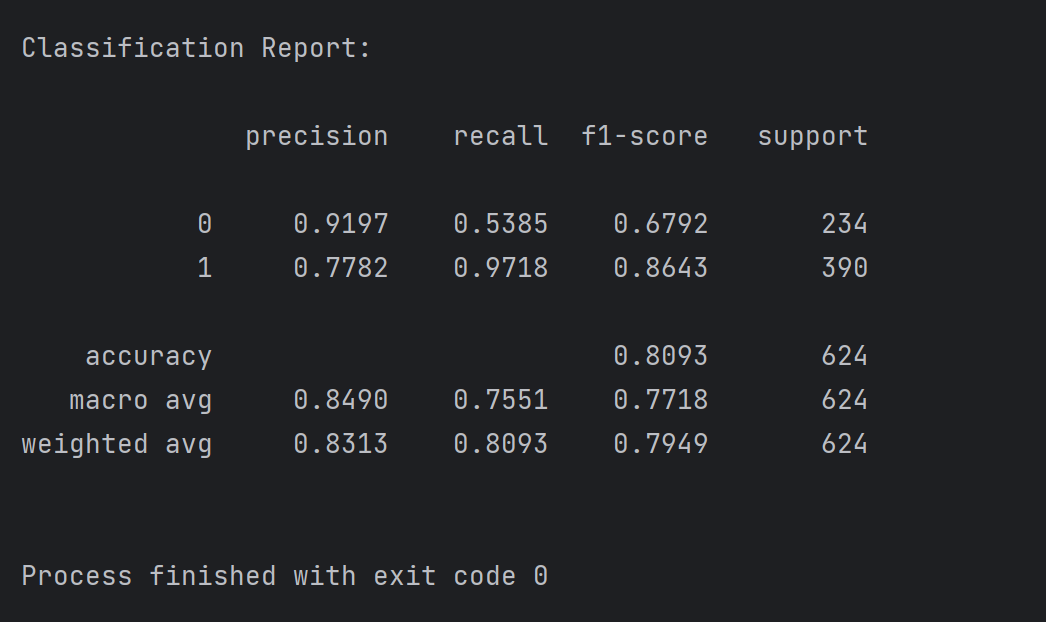


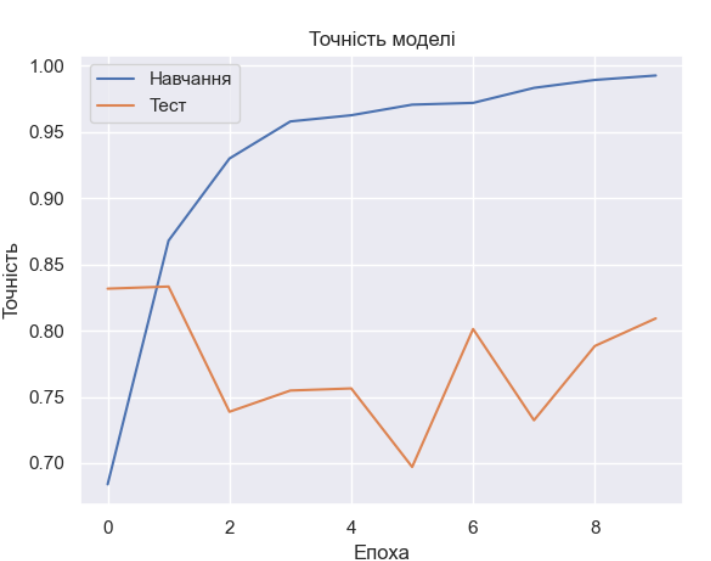


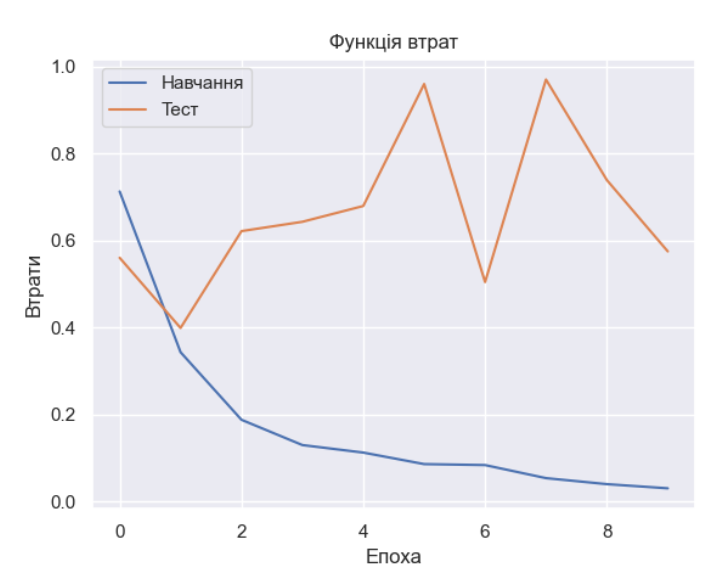


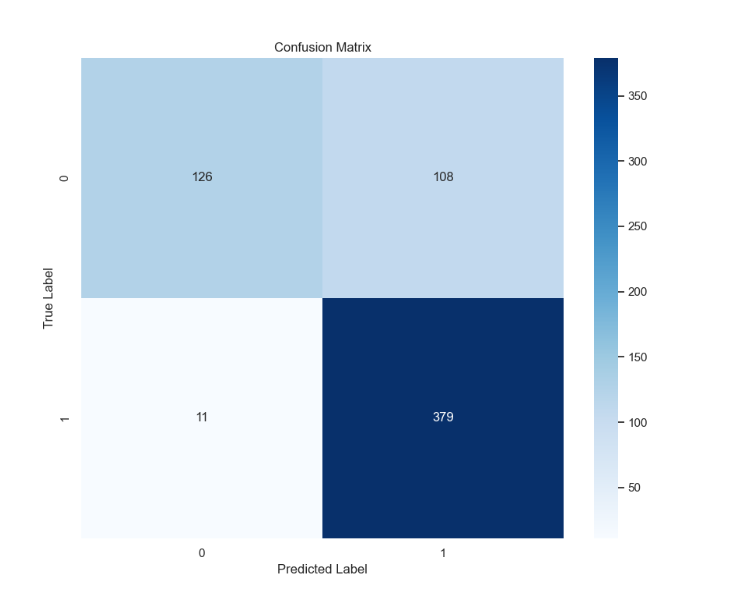








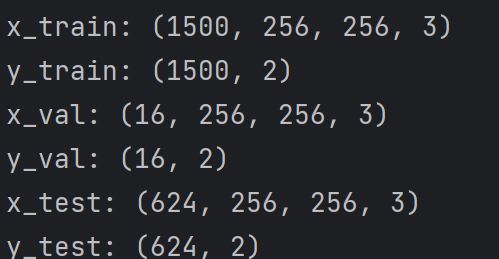
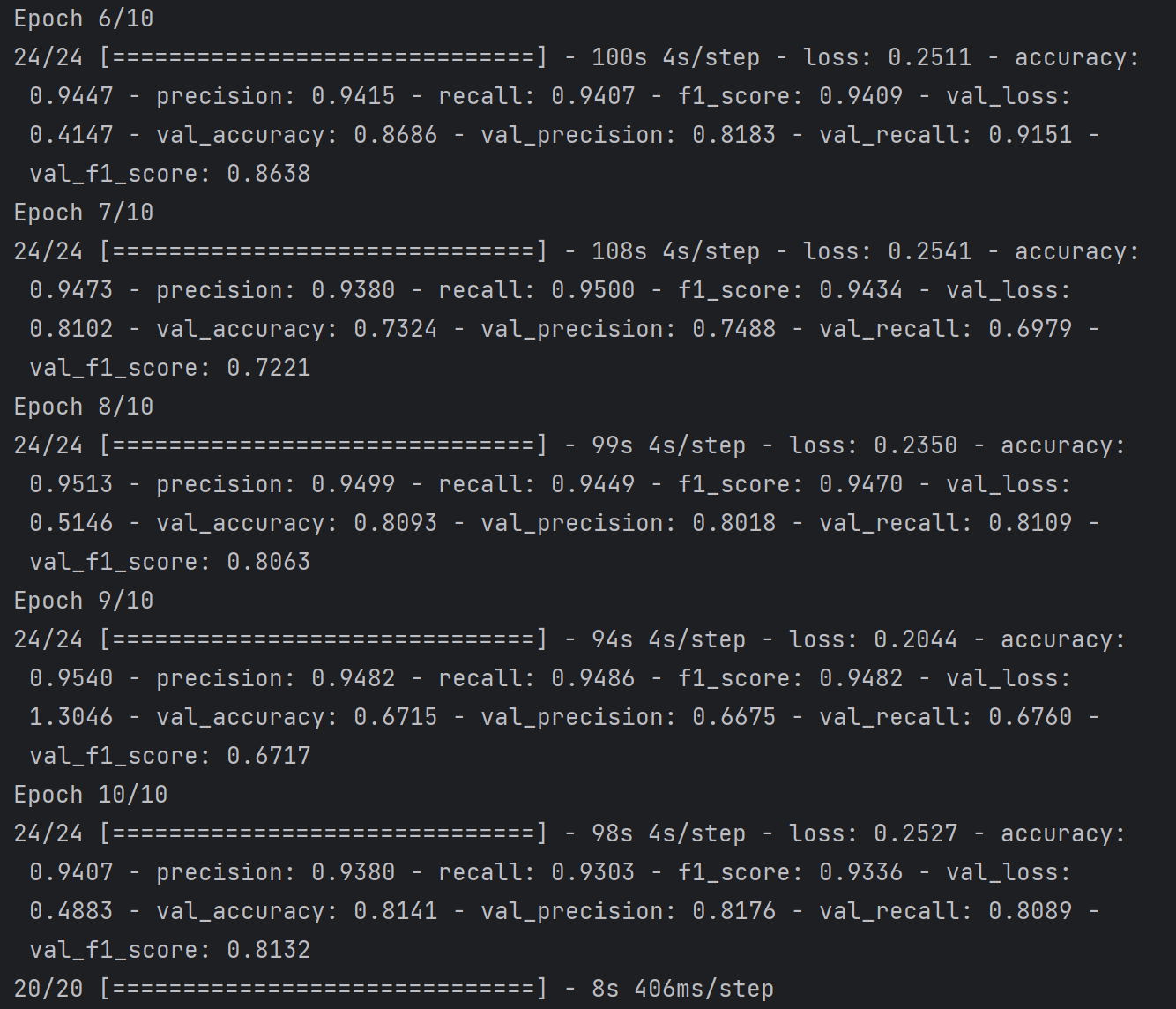
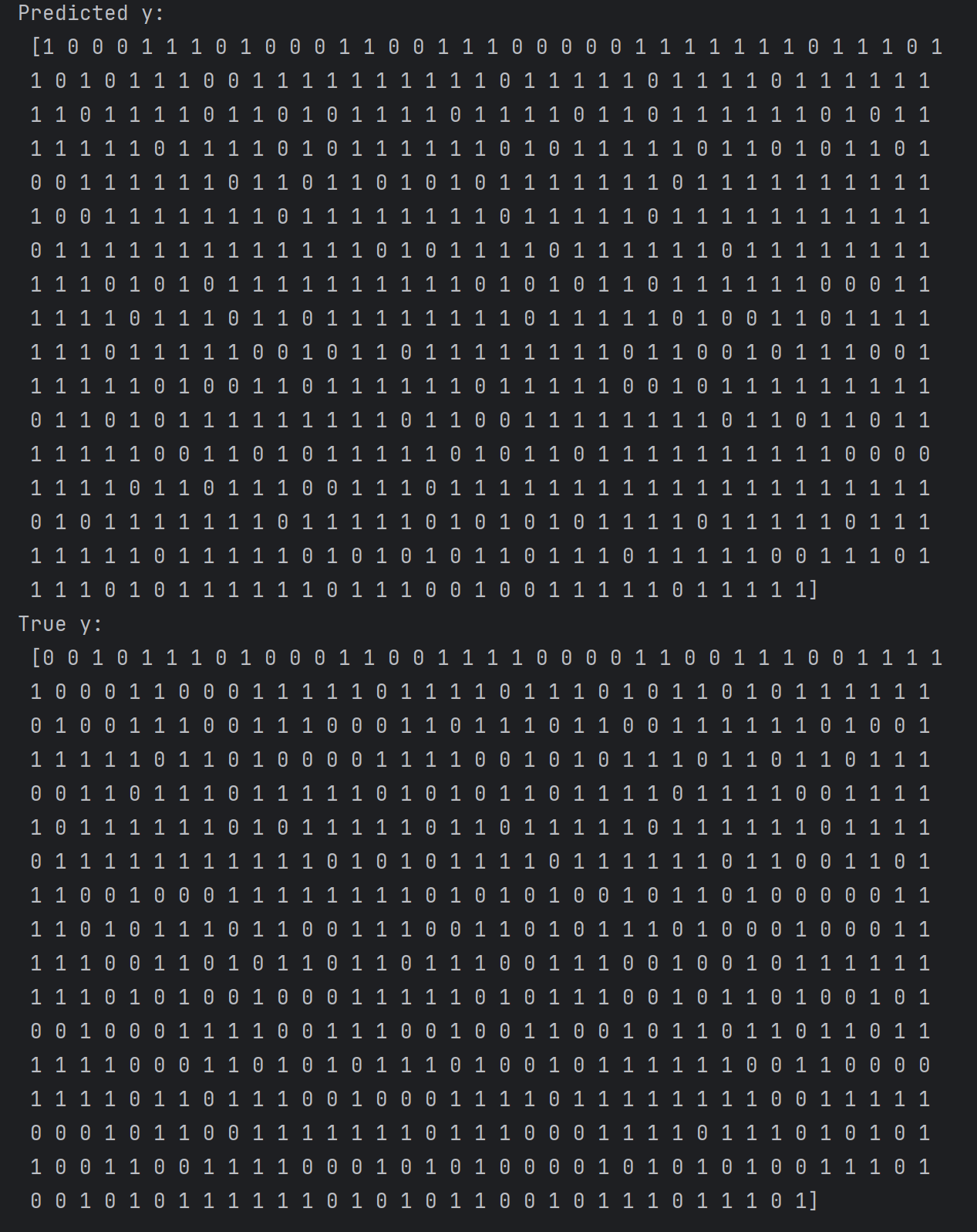
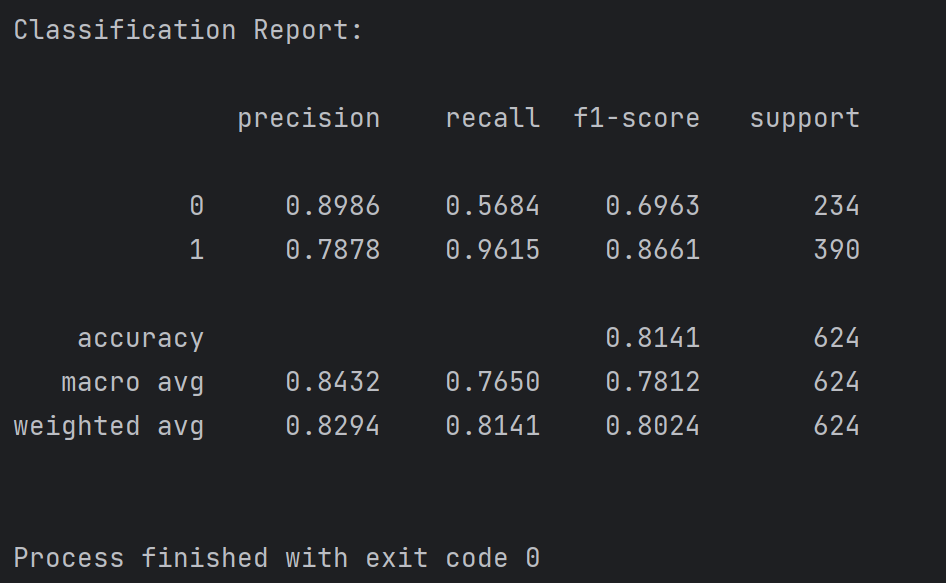


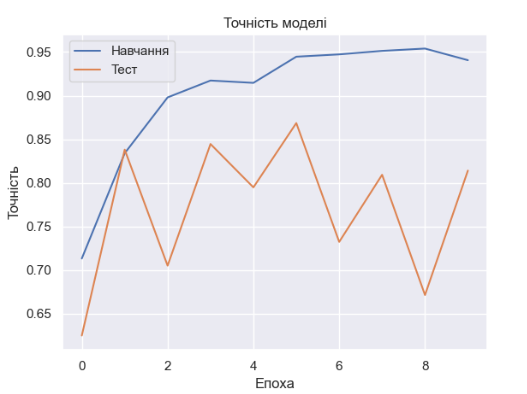
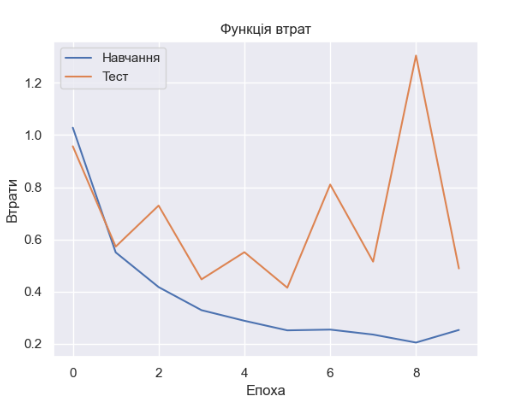


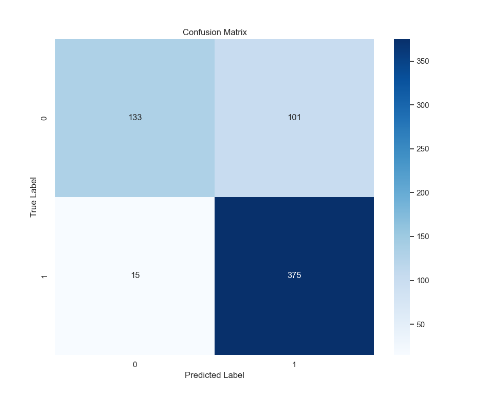
Спробуємо додати до моделі padding та спробуємо додати до нейроної мережі різновиди пулінгу, додамо регуляцію:



Результати:

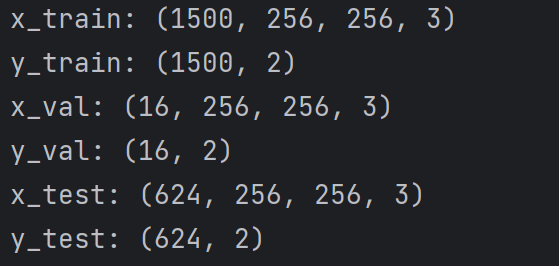
    

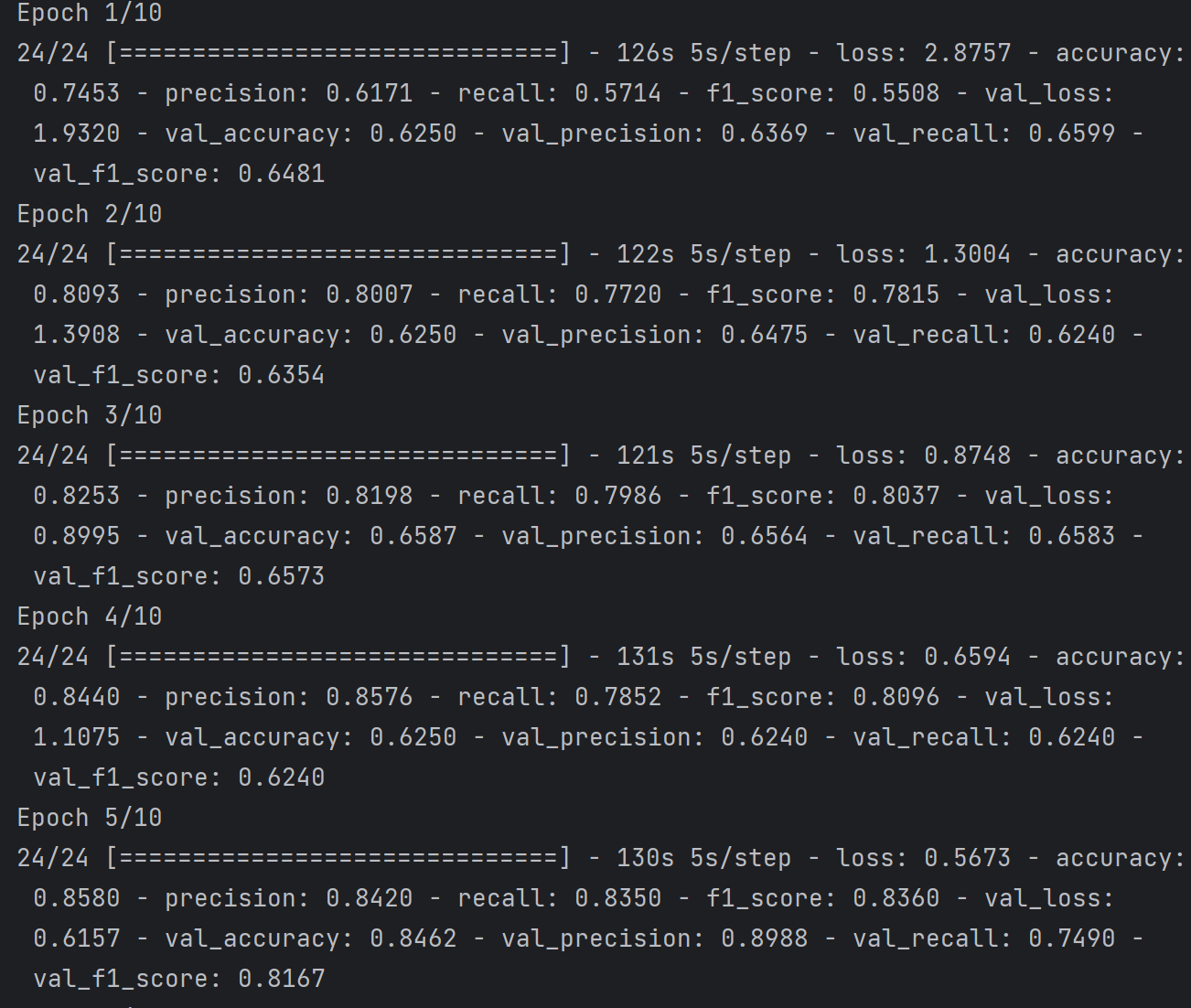
 

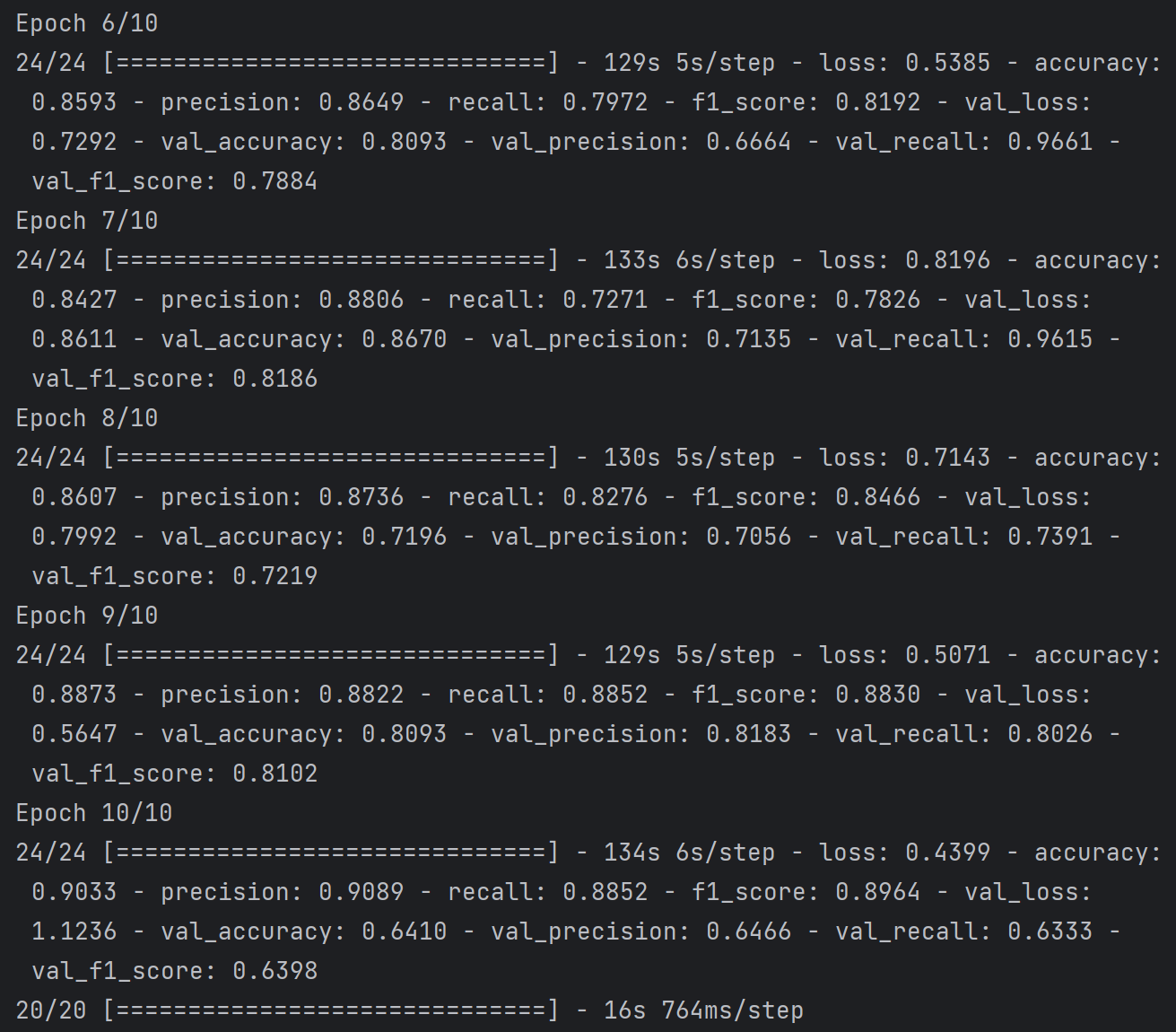


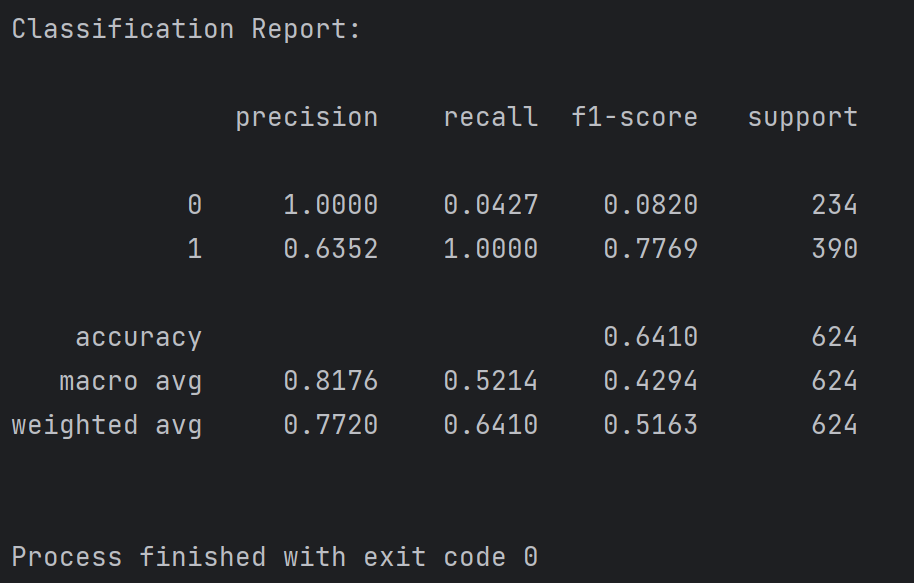
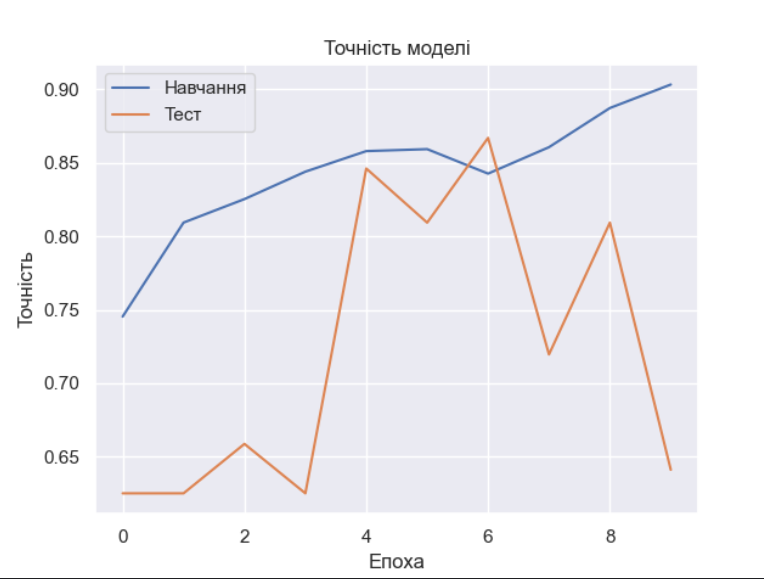
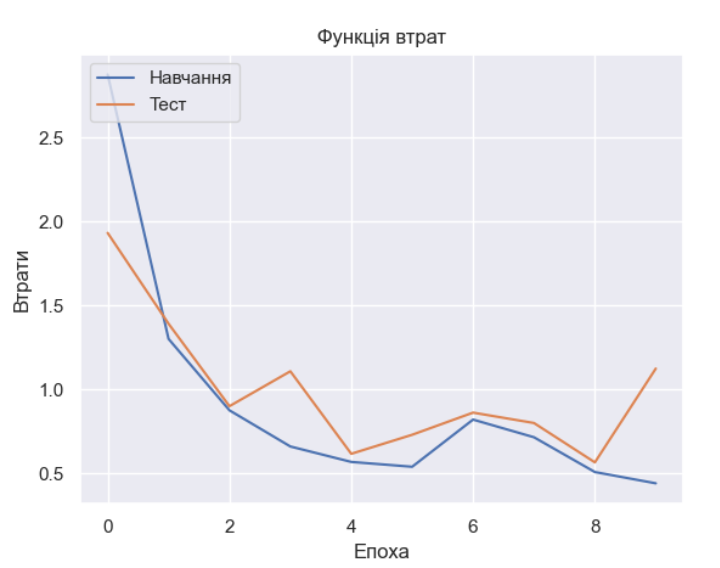
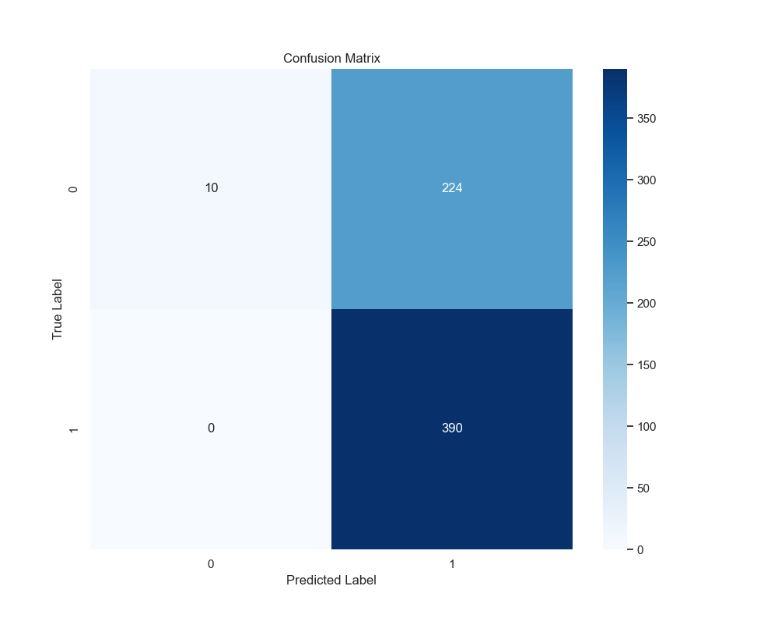
Спробуємо змінити функції активації relu на elu

Результати:







## 2.5 Аналіз результатів

Розглянувши результати можна зробити висновок, що результат навчання нейроної мережі має високу точність (Accuracy = 0.766). Також зі зміною архітектури та параметрів моделі, результат покращився. Це сталося після збільшення тренувальної збірки і додаванню до нашої моделі нейроної мережі padding, різновиди пулінгу (MaxPooling2D,Conv2D), регуляції (dropout, L2) з Accuracy = 0.766 до accuracy = 0.8141. Можна зазначити, що модель має потенціал на її вдосконалення. Для покращення нейронної мережі для бінарної класифікації рентгенівських зображень пневмонії та здорових легень можна врахувати наступні загальні рекомендації:

1. Збільшення обсягу даних: Збільште обсяг навчальних даних, особливо у випадку невеликого набору зображень, використовуючи методи аугментації даних.
2. Нормалізація даних: Виконайте нормалізацію зображень, щоб забезпечити стабільність та швидкість збіжності.
3. Оптимізація гіперпараметрів: Проведіть експерименти з різними конфігураціями архітектури, включаючи кількість шарів, розміри ядер, розміри пулінгу та кількість нейронів.
4. Використання передньо-навченої моделі: Використовуйте передньо-навчені моделі, такі як VGG16 або ResNet, які мають добре вивчені характеристики.
5. Функції активації та оптимізатор: Виберіть підходящі функції активації та оптимізатори для вашої конкретної задачі.
6. Dropout та регуляризація: Використовуйте Dropout для запобігання перенавчанню та регуляризацію для управління складністю моделі.
7. Параметри згорткових шарів: Розгляньте використання параметра padding='same' для збереження інформації по краях зображень.
8. Підбір оптимального розміру пакету: Експериментуйте із розміром пакету для знаходження оптимальної швидкості навчання та збіжності.
9. Моніторинг метрик валідації: Спостерігайте за метриками валідації, такими як точність, чутливість, специфічність, точність та F1-оцінка, для оцінки ефективності моделі.

10) Тестування на реальних даних: Важливо випробувати модель на реальних клінічних даних, щоб переконатися в її ефективності в реальних умовах.

# Висновок

Отже, під час виконання індивідального завдання ми розробили та оптимізували нейронну мережу для бінарної класифікації рентгенівських зображень грудної клітини в контексті діагностики пневмонії. Діагностика пнемонії є актуальним та перспективним напрямком у вдосконаленні медичних практик. Пневмонія, як серйозне інфекційне захворювання легень, вимагає оперативної та точної діагностики для ефективного лікування пацієнтів.

Використання нейронних мереж для обробки медичних зображень відкриває нові можливості для автоматизації діагностики, що особливо актуально в умовах зростаючого обсягу медичних даних. Зосередження на бінарній класифікації, розрізняючи здорових пацієнтів від тих, хто страждає від пневмонії, створює потужний інструмент для раннього виявлення та ефективного лікування захворювань легень.

# Список використаних джерел

1. Методи та технології обчислювального інтелекту: Практичні

роботи [Електронний ресурс] : навч. посіб. для студ. спеціальності 122

«Комп’ютерні науки» / І. В. Федорін; КПІ ім. Ігоря Сікорського. –

Електронні текстові дані (1 файл: 13,16 Мбайт). – Київ: КПІ ім. Ігоря

Сікорського, 2022. – 317 с.

1. Методи та технології обчислювального інтелекту: Навчальний

посібник [Електронний ресурс] : навч. посіб. для студ. спеціальності 122

«Комп’ютерні науки» / І. В. Федорін; КПІ ім. Ігоря Сікорського. –

Електронні текстові дані (1 файл: 15,92 Мбайт). – Київ: КПІ ім. Ігоря

Сікорського, 2022. – 314 с.

1. Chest X-Ray Images (Pneumonia) [Електронний ресурс] : - <https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia>
2. Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning - Daniel S. Kermanyc, Michael Goldbaum, Wenjia Cai, M. Anthony LewisHuimin, XiaKang Zhang [Електронний ресурс] : <http://www.cell.com/cell/fulltext/S0092-8674(18)30154-5>
3. Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification - Daniel Kermany, Kang Zhang, Michael Goldbaum [Електронний ресурс] :  <https://data.mendeley.com/datasets/rscbjbr9sj/2>
4. Convolutional Neural Networks [Електронний ресурс] : <https://www.tensorflow.org/tutorials/images/cnn>
5. Getting Started with TensorFlow [Електронний ресурс] : <https://www.tensorflow.org/tutorials/quickstart/beginner>
6. Zhang, X. A Mathematical Model of a Neuron with Synapses based on Physiology. Nat Prec (2008). https://doi.org/10.1038/npre.2008.1703.1.