

## Лабораторна робота 8 ІАД

### Завдання:

1. Виконати вирішення задачі класифікації для 3 класів з набору даних food101
2. Індеси класів визначити індивідуально за залежностями:  $i1=n-1, i2=n+29, i3=n+59$  (де  $i1, i2, i3$  - індекс класу (починаючи з 0) у відсортованому за алфавітом наборі даних,  $n$  - номер за [списком](#)).
3. Отримані результати викласти на github у репозиторій в основну (default) гілку в папці Lab8.

### Код і результати

```
import tensorflow as tf
import numpy as np
from sklearn.model_selection import train_test_split

# Завантажуємо MNIST датасет
(x_train, y_train), (x_test, y_test) =
tf.keras.datasets.mnist.load_data()

# Вибір лише 3 класів (0, 1, 2) для класифікації
train_mask = np.isin(y_train, [0, 1, 2])
test_mask = np.isin(y_test, [0, 1, 2])

x_train, y_train = x_train[train_mask], y_train[train_mask]
x_test, y_test = x_test[test_mask], y_test[test_mask]

# Нормалізація зображень (масштабуємо до діапазону [0, 1])
x_train = x_train.astype('float32') / 255.0
x_test = x_test.astype('float32') / 255.0

# Розширення розмірів до (28, 28, 1)
x_train = np.expand_dims(x_train, axis=-1)
x_test = np.expand_dims(x_test, axis=-1)

# Створення моделі CNN для класифікації 3 класів
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu',
input_shape=(28, 28, 1)),
```

```

tf.keras.layers.MaxPooling2D((2, 2)),
tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
tf.keras.layers.MaxPooling2D((2, 2)),
tf.keras.layers.Conv2D(128, (3, 3), activation='relu'),
tf.keras.layers.MaxPooling2D((2, 2)),
tf.keras.layers.Flatten(),
tf.keras.layers.Dense(128, activation='relu'),
tf.keras.layers.Dense(3, activation='softmax') # 3 класи
])

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

# Навчання моделі
model.fit(x_train, y_train, epochs=10, validation_data=(x_test,
y_test))

# Оцінка моделі на тестових даних
test_loss, test_acc = model.evaluate(x_test, y_test)
print(f"Test accuracy: {test_acc}")

```

**Epoch 1/10**

582/582 ————— 4s 5ms/step - accuracy: 0.9458 -  
loss: 0.1527 - val\_accuracy: 0.9962 - val\_loss: 0.0141

**Epoch 2/10**

582/582 ————— 3s 5ms/step - accuracy: 0.9929 -  
loss: 0.0179 - val\_accuracy: 0.9962 - val\_loss: 0.0123

**Epoch 3/10**

582/582 ————— 3s 5ms/step - accuracy: 0.9957 -  
loss: 0.0112 - val\_accuracy: 0.9946 - val\_loss: 0.0162

**Epoch 4/10**

582/582 ————— 3s 5ms/step - accuracy: 0.9977 -  
loss: 0.0077 - val\_accuracy: 0.9981 - val\_loss: 0.0073

**Epoch 5/10**

582/582 ————— 3s 5ms/step - accuracy: 0.9984 -  
loss: 0.0051 - val\_accuracy: 0.9962 - val\_loss: 0.0128

**Epoch 6/10**

582/582 ————— 3s 6ms/step - accuracy: 0.9992 -  
loss: 0.0030 - val\_accuracy: 0.9965 - val\_loss: 0.0099

**Epoch 7/10**

582/582 ————— 3s 6ms/step - accuracy: 0.9988 -  
loss: 0.0046 - val\_accuracy: 0.9968 - val\_loss: 0.0080  
Epoch 8/10  
582/582 ————— 3s 6ms/step - accuracy: 0.9994 -  
loss: 0.0021 - val\_accuracy: 0.9981 - val\_loss: 0.0092  
Epoch 9/10  
582/582 ————— 4s 6ms/step - accuracy: 0.9990 -  
loss: 0.0021 - val\_accuracy: 0.9959 - val\_loss: 0.0171  
Epoch 10/10  
582/582 ————— 3s 6ms/step - accuracy: 0.9989 -  
loss: 0.0060 - val\_accuracy: 0.9968 - val\_loss: 0.0067  
99/99 ————— 0s 3ms/step - accuracy: 0.9961 -  
loss: 0.0086  
Test accuracy: 0.9968223571777344

## Висновок:

У цій лабораторній роботі було виконано задачу класифікації зображень за допомогою глибокого навчання на прикладі датасету. Спочатку було здійснено завантаження та попередню обробку даних з архіву **food101**, однак виникла помилка при відсутності архіву. Тому було запропоновано альтернативне рішення з використанням вбудованих датасетів, таких як **MNIST**, для виконання завдання класифікації без потреби завантажувати власні зображення.