

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
КАФЕДРА ІНФОРМАТИКИ ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Проектування та дослідження систем із штучним інтелектом

ЗВІТ

до лабораторної роботи №4

Виконав

ІП-41мн Пашковський Євгеній Сергійович

Київ 2025

Завдання

Написати програму що реалізує згорткову нейронну мережу AlexNet для розпізнавання об'єктів з датасету ImageNet

Хід роботи

Код програми

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

```
import tensorflow as tf

(x_train, y_train), (x_test, y_test) = tf.keras.datasets.cifar10.load_data()

x_train = x_train / 255.0
x_test = x_test / 255.0

print(f'x_train shape: {x_train.shape}')
print(f'x_test shape: {x_test.shape}')

print(f'y_train shape: {y_train.shape}')
print(f'y_test shape: {y_test.shape}')

plt.figure(figsize=(10, 10))
for i in range(25):
    ax = plt.subplot(5, 5, i + 1)
    plt.imshow(x_train[i])
    plt.title(f'Label: {y_train[i]}')
    plt.axis('off')
plt.show()
```

```
import keras
from keras import models, layers

def AlexNet(input_shape=(227, 227, 3), n_classes = 200):
```

```

model = models.Sequential()

model.add(layers.Input(shape=input_shape))
model.add(layers.Resizing(227, 227))
model.add(layers.RandomColorJitter(
    value_range=(0, 1),
    brightness_factor=0.2,
    contrast_factor=0.2,
    saturation_factor=0.2,
    hue_factor=0.2
))
model.add(layers.RandomFlip(
    mode="horizontal_and_vertical"
))

    model.add(layers.Conv2D(96, kernel_size=(11, 11), strides=(4, 4),
activation='relu'))
    model.add(layers.BatchNormalization())
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    model.add(layers.Conv2D(256, kernel_size=(5, 5), activation='relu'))
    model.add(layers.BatchNormalization())
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    model.add(layers.Conv2D(384, kernel_size=(3, 3), activation='relu'))
    model.add(layers.BatchNormalization())

    model.add(layers.Conv2D(384, kernel_size=(3, 3), activation='relu'))
    model.add(layers.BatchNormalization())

    model.add(layers.Conv2D(256, kernel_size=(3, 3), activation='relu'))
    model.add(layers.BatchNormalization())
    model.add(layers.MaxPooling2D(pool_size=(3, 3), strides=(2, 2)))

    model.add(layers.GlobalAveragePooling2D())

    model.add(layers.Dense(4096, activation='relu'))
    model.add(layers.Dropout(0.5))

    model.add(layers.Dense(4096, activation='relu'))
    model.add(layers.Dropout(0.5))

```

```
model.add(layers.Dense(n_classes, activation='softmax'))
```

```
return model
```

```
from keras import optimizers
```

```
epochs = 50
```

```
initial_learning_rate = 1e-3
```

```
final_learning_rate = 1e-6
```

```
learning_rate_decay_factor = (final_learning_rate /  
initial_learning_rate) ** (1 / epochs)  
steps_per_epoch = len(x_train)
```

```
learning_rate = optimizers.schedules.ExponentialDecay(  
    initial_learning_rate=initial_learning_rate,  
    decay_steps=steps_per_epoch,  
    decay_rate=learning_rate_decay_factor  
)
```

```
model = AlexNet(input_shape=(32, 32, 3), n_classes=10)  
model.compile(optimizer=optimizers.Adam(learning_rate=learning_rate),  
loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

```
model.summary()
```

```
history = model.fit(x_train, y_train, epochs=50, validation_split=0.2)
```

```
model.evaluate(x_test, y_test)
```

```
plt.plot(history.history['loss'], label='Train Loss')  
plt.plot(history.history['val_loss'], label='Validation Loss')  
plt.xlabel('Епохи')  
plt.ylabel('Функція втрат')  
plt.title('Графік функції втрат')  
plt.legend()  
plt.grid()  
plt.show()
```

```
plt.plot(history.history['accuracy'], label='Train Accuracy')  
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
```

```
plt.xlabel('Епохи')
plt.ylabel('Точність')
plt.title('Графік точності')
plt.legend()
plt.grid()
plt.show()
```

```
predictions = model.predict(x_test).argmax(axis=1)

predictions
```

```
from sklearn.metrics import confusion_matrix

conf_matrix = confusion_matrix(y_test, predictions)
fig, ax = plt.subplots(figsize=(30, 8))
cax = ax.matshow(conf_matrix, cmap="Blues")

plt.colorbar(cax)

class_labels=range(10)

ax.set_xticks(range(len(class_labels)))
ax.set_yticks(range(len(class_labels)))
ax.set_xticklabels(class_labels, fontsize=7)
ax.set_yticklabels(class_labels, fontsize=7)

plt.title("Confusion Matrix", pad=20)
plt.xlabel("Predicted", labelpad=10)
plt.ylabel("Actual", labelpad=10)

for (i, j), value in np.ndenumerate(conf_matrix):
    ax.text(j, i, f'{value}', ha='center', va='center', color='black',
fontSize=7)

plt.tight_layout()
plt.show()
```

Отриманий результат

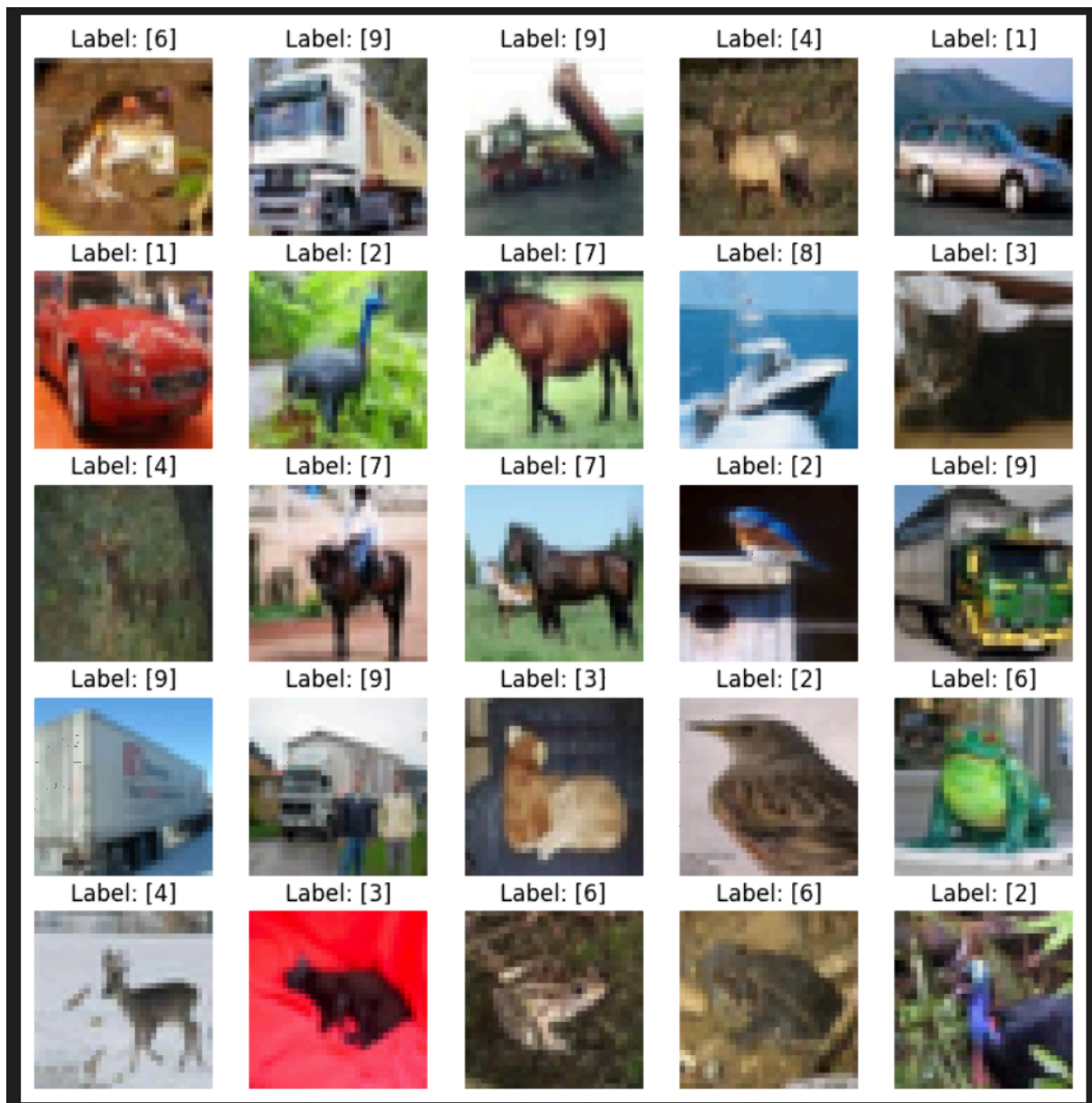


Рисунок 1 – Приклад зображень датасету

Layer (type)	Output Shape	Param #
resizing (Resizing)	(None, 227, 227, 3)	0
random_color_jitter (RandomColorJitter)	(None, 227, 227, 3)	0
random_flip (RandomFlip)	(None, 227, 227, 3)	0
conv2d (Conv2D)	(None, 55, 55, 96)	34,944
batch_normalization (BatchNormalization)	(None, 55, 55, 96)	384
max_pooling2d (MaxPooling2D)	(None, 27, 27, 96)	0
conv2d_1 (Conv2D)	(None, 23, 23, 256)	614,656
batch_normalization_1 (BatchNormalization)	(None, 23, 23, 256)	1,024
max_pooling2d_1 (MaxPooling2D)	(None, 11, 11, 256)	0
conv2d_2 (Conv2D)	(None, 9, 9, 384)	885,120
batch_normalization_2 (BatchNormalization)	(None, 9, 9, 384)	1,536
conv2d_3 (Conv2D)	(None, 7, 7, 384)	1,327,488
batch_normalization_3 (BatchNormalization)	(None, 7, 7, 384)	1,536
conv2d_4 (Conv2D)	(None, 5, 5, 256)	884,992
batch_normalization_4 (BatchNormalization)	(None, 5, 5, 256)	1,024
max_pooling2d_2 (MaxPooling2D)	(None, 2, 2, 256)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 256)	0
dense (Dense)	(None, 4096)	1,052,672
dropout (Dropout)	(None, 4096)	0
dense_1 (Dense)	(None, 4096)	16,781,312
dropout_1 (Dropout)	(None, 4096)	0
dense_2 (Dense)	(None, 10)	40,970

...

Total params: 21,627,658 (82.50 MB)

Рисунок 2 – Архітектура мережі AlexNet

```
> \n model.evaluate(x_test, y_test)\n[6]\n... 313/313 ————— 1s 4ms/step - a\n... [1.5231800079345703, 0.6886000037193298]
```

Рисунок 3 – Кінцева похибка та точність моделі

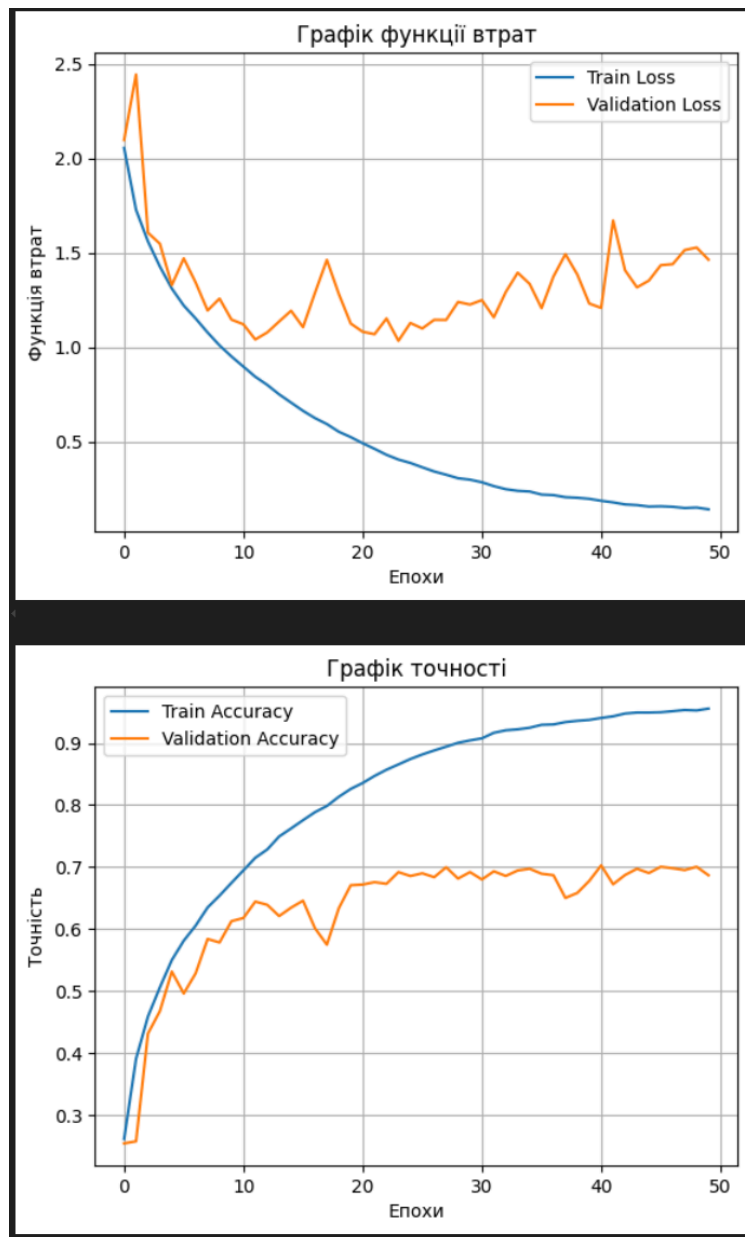


Рисунок 4 – Графіки функцій втрат та точності

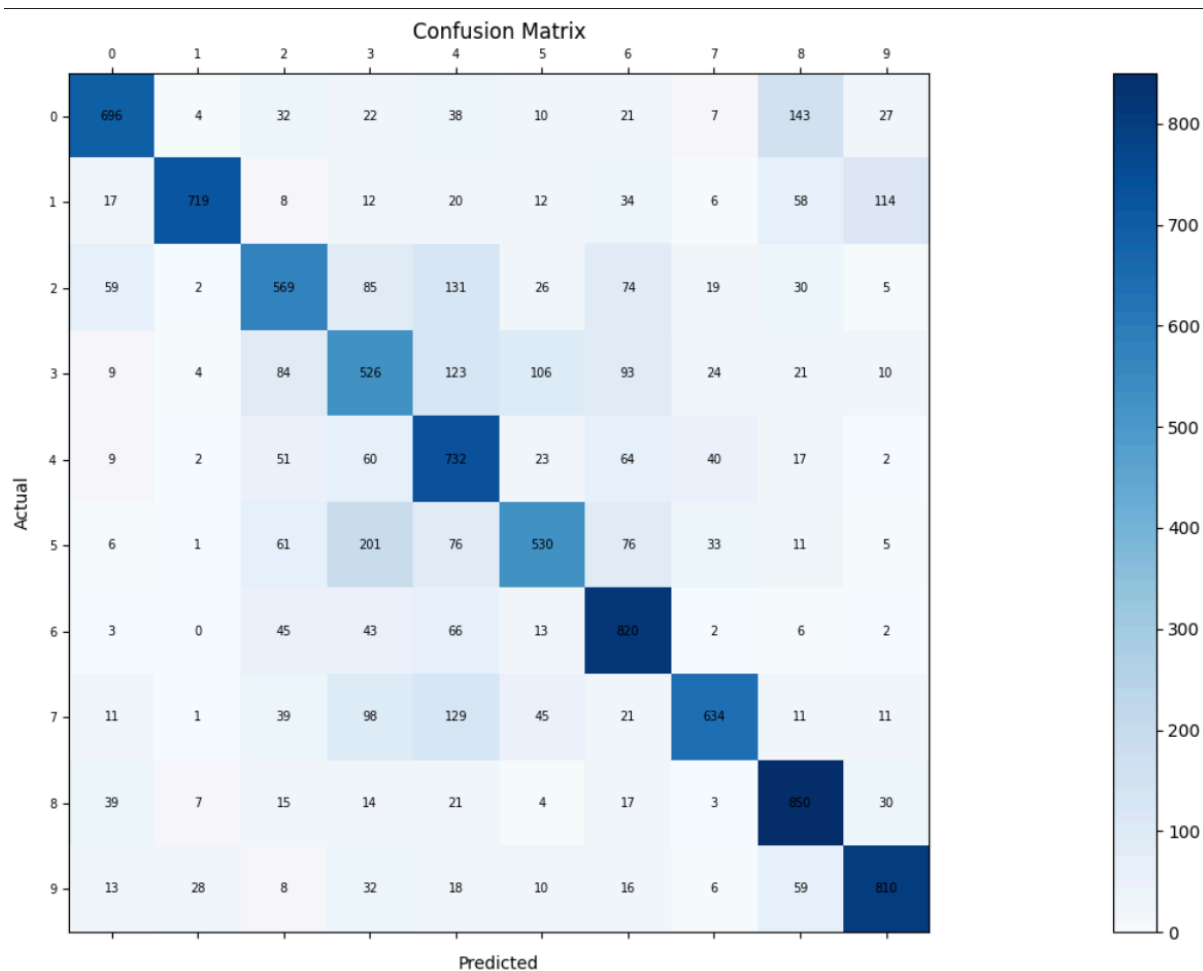


Рисунок 5 – Confusion matrix результатів від отриманої моделі

Висновки

В рамках цієї лабораторної роботи було успішно побудовано, навчено та проаналізовано модель згорткової нейронної мережі AlexNet для розпізнавання об'єктів на зображеннях.