МІНІСТЕРСТВО ОСВІТИ І НАУКИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ.ІГОРЯ СІКОРСЬКОГО»

НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Лабораторна робота №3

«Прикладне програмне забезпечення навчання та тестування нейронної мережі для класифікації тестових даних. Аналіз впливу архітектури мережі та параметрів навчання на точність класифікації.»

Виконав:
Студент 3 курсу
Групи ФІ-21
Голуб Михайло

Перевірив:
Железняков. Д. О.

Київ 2025

ЗМІСТ

1. ЗАВДАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1.1. Набір даних

- CIFAR-100 або інші складні набори даних для класифікації зображень
- Можна використовувати більш прості набори даних (MNIST, CIFAR-10), але оцінка не буде знижена
- Можна використовувати набори даних для більш складного завдання (додаткові бали)

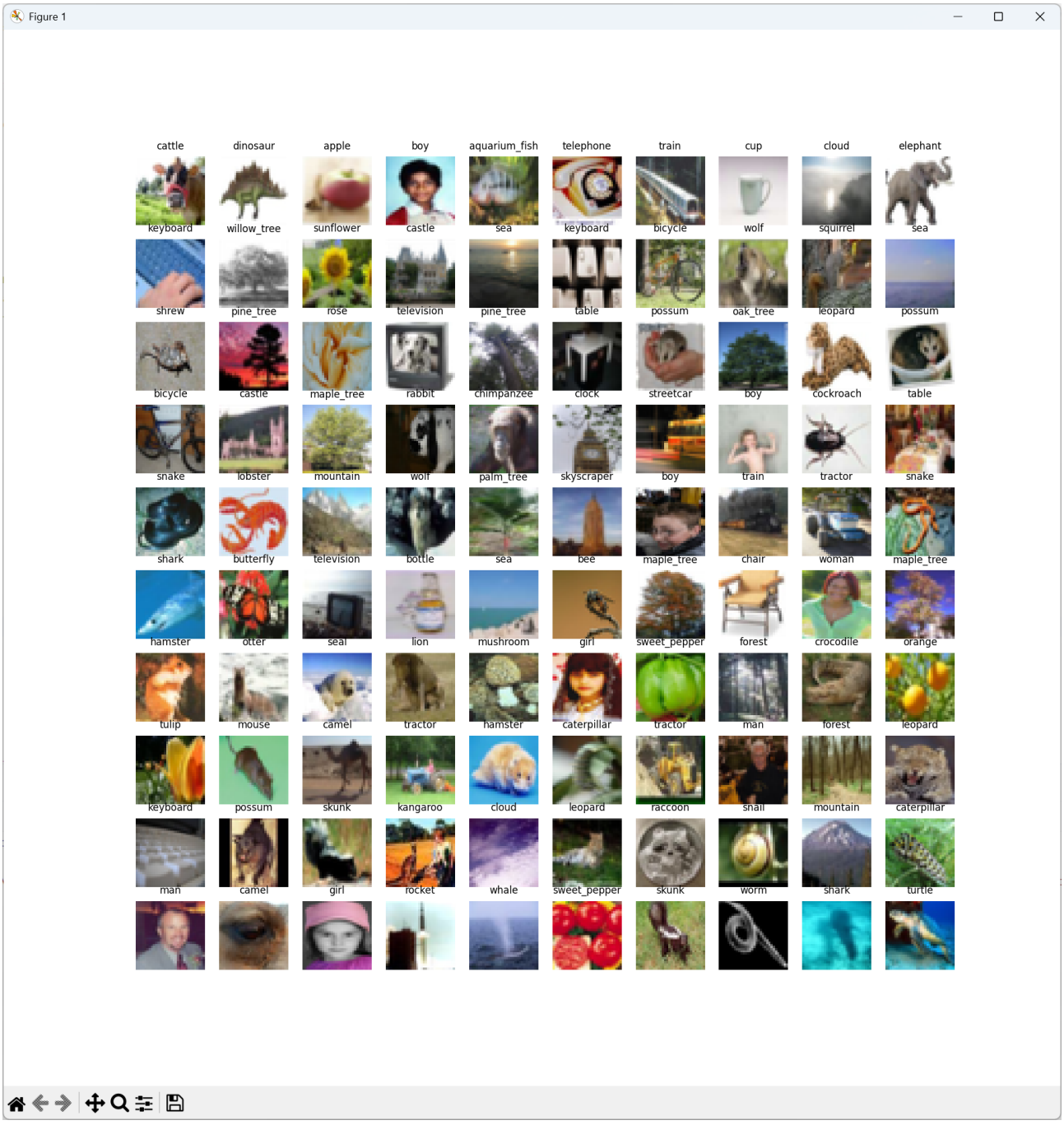
1.2. Завдання

- Ознайомитись з теоретичними відомостями до архітектур нейронних мереж (згорткові нейронні мережі, типи шарів, overfitting, ...).
- Завантажити набір даних
- За необхідності зробити попередню обробку та поділити датасет на: навчальний, валідаційний та тестовий
- Розробити архітектуру згорткової нейронної мережі для класифікації зображень
- Оцінити якість класифікації зображень (baseline)
- Провести серію експериментів (~ 4-8 експериментів) з архітектурою нейронної мережі та дослідити як впливає архітектура нейронної мережі на якість класифікації та процес навчання (loss, time). Бажано брати декілька різних параметрів. Для прикладу: збільшення кількості параметрів згорткового шару, інша функція активації, додавання Dropout, додавання нового скритого шару, тощо.
    - Зробити висновки по кожному експерименту.
- Порівняти результати. В деяких випадках доцільно показати у вигляді “Ablation Study”
    - Інколи бажано спробувати змінити один той самий параметр декілька разів (збільшити та зменшити)
    - Обрати найкращу модель та загальні висновки
- Зробити звіт
- Захистити роботу

2. ХІД РОБОТИ

2.1. Набір даних

Використовується набір CIFAR100, що містить 100 класів зображень. Сумарна кількість зображень – 60000, з них 50000 для тренування і 10000 для тестування. З 50000 тренувальних – 40000 для навчання, 10000 для передбачень і відбору.



Візуалізація класів)

2.2. Опис очікуваних характеристик

Для accuracy покладемо значення:

Accuracy (a)	Характеристичний показник
a < 0.01	Модель працює гірше за випадковий вибір
0.01< a < 0.05	Модель працює дуже погано
0.05 < a < 0.1	Модель працює погано
0.1 < a < 0.2	Модель працює неточно
0.2< a<0.3	Модель працює нормально
0.3< a<0.4	Модель працює добре
0.4 < a	Модель працює дуже добре

Для loss покладемо значення:

Loss (l)	Характеристичний показник
l > 6	Модель дуже невпевнена
4 < l < 6	Модель невпевнена
3 < l < 4	Модель дещо невпевнена
2 < l < 3	Модель нормальна
l < 2	Модель достатньо впевнена

2.3. Базова архітектура

2.3.1. Структура базової архітектури

Базова архітектура створюється наступним кодом:

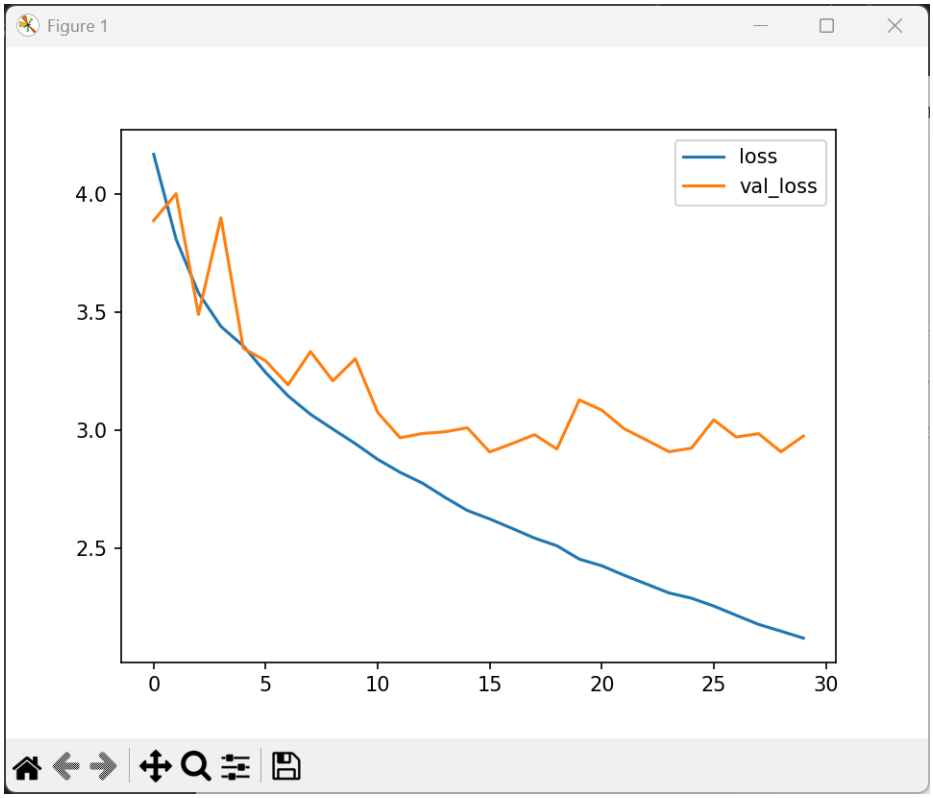
```
model = Sequential()
model.add(Input(shape=(32, 32, 3)))
model.add(Conv2D(32, (5, 5), activation='relu',padding='same'))
model.add(BatchNormalization())
model.add(Flatten())
model.add(Dense(400, activation='tanh'))
model.add(Dropout(0.3))
model.add(Dense(200, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(100, activation='softmax'))
model.compile(optimizer=Adam(learning_rate=0.001),
              loss=CategoricalCrossentropy(),
              metrics=['accuracy', TopKCategoricalAccuracy(k=2, name="Top2")])

history = model.fit(x_train, y_train_cat, batch_size=200, epochs = 30,
validation_data=(x_valid, y_valid_cat))
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 32, 32, 32)        2432

 batch_normalization (BatchN  (None, 32, 32, 32)       128
 ormalization)

 flatten (Flatten)           (None, 32768)             0

 dense (Dense)               (None, 400)               13107600

 dropout (Dropout)           (None, 400)               0

 dense_1 (Dense)             (None, 200)               80200

 dropout_1 (Dropout)         (None, 200)               0

 dense_2 (Dense)             (None, 100)               20100

=================================================================
Total params: 13,210,460
Trainable params: 13,210,396
Non-trainable params: 64
```

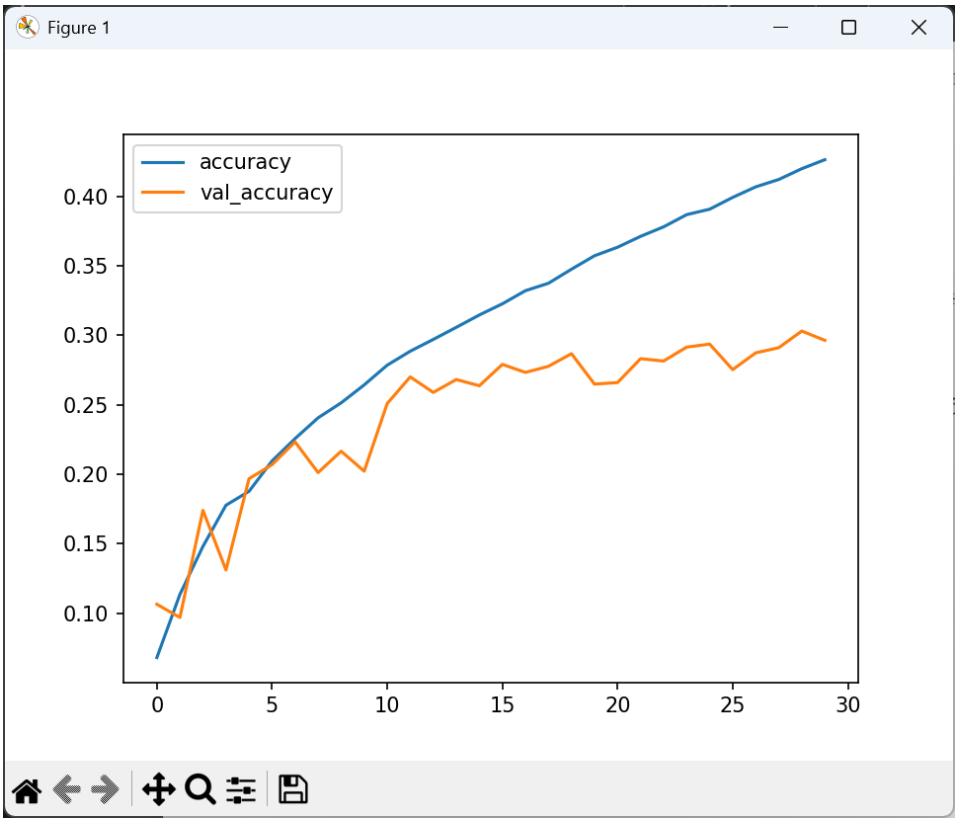
Базова архітектура. Результат виконання `model.summary()`

2.3.2. Точність і втрати базової архітектури

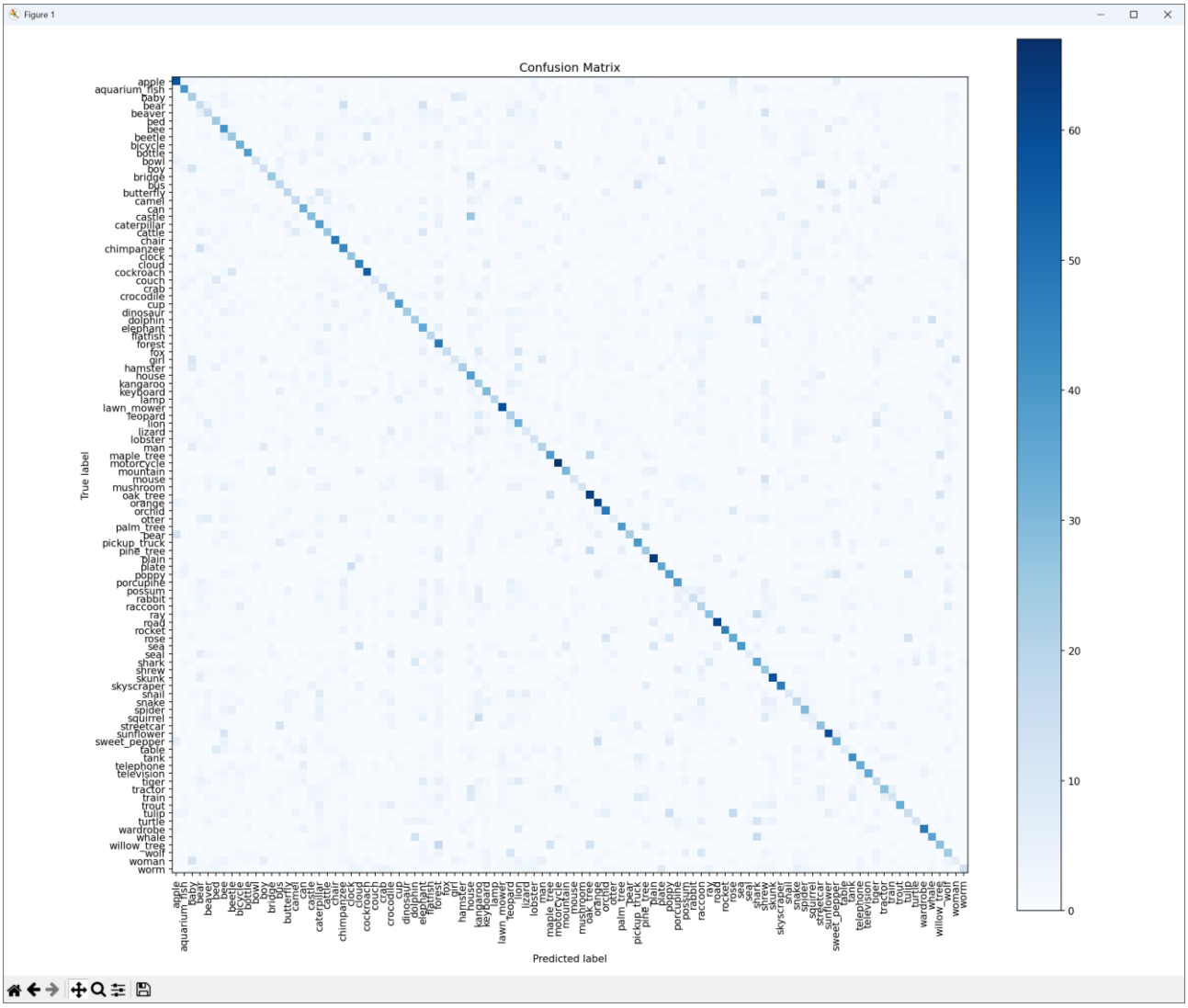
Базова архітектура перенавчається на 10-12 епосі, після чого модель дещо невпевнена і працює нормально (loss ~3, accuracy ~0.27)



[redacted] Функції втрат базової архітектури)



[redacted] Функції точності базової архітектури)



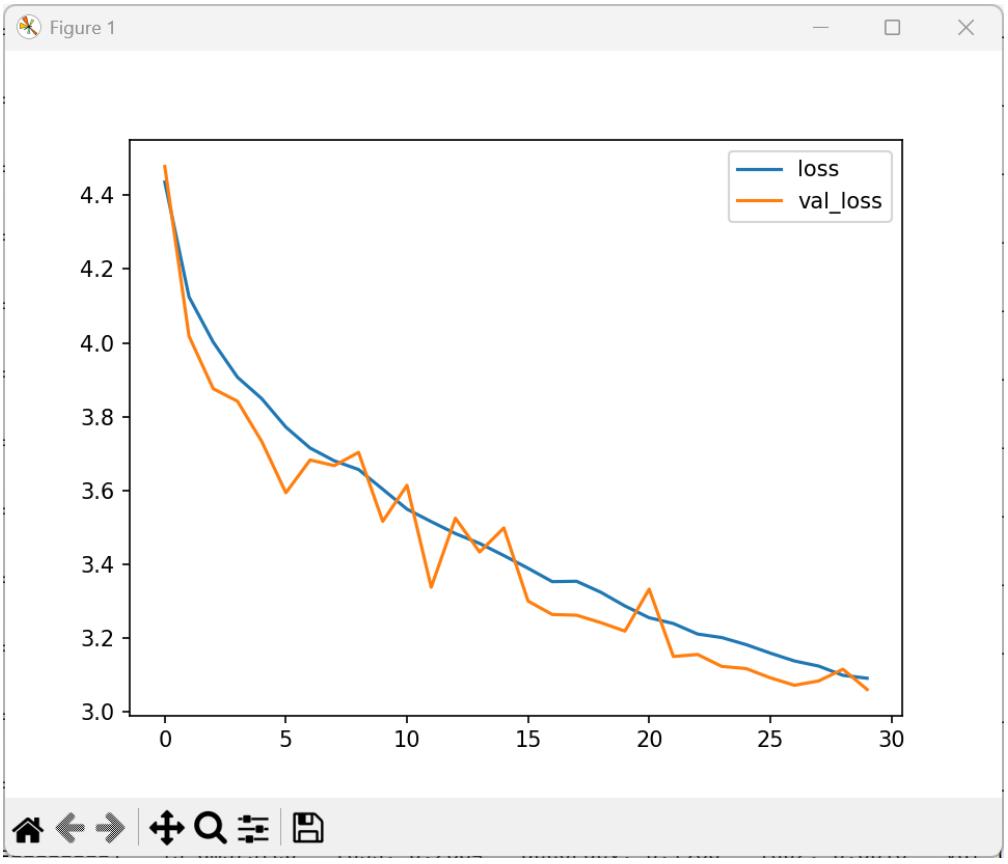
Матриця плутань базової архітектури)

З матриці плутань видно, що модель часто плутає схожі класи (дерева, дельфінів з акулами, тощо) і досить точно визначає унікальні класи (наприклад стільці).

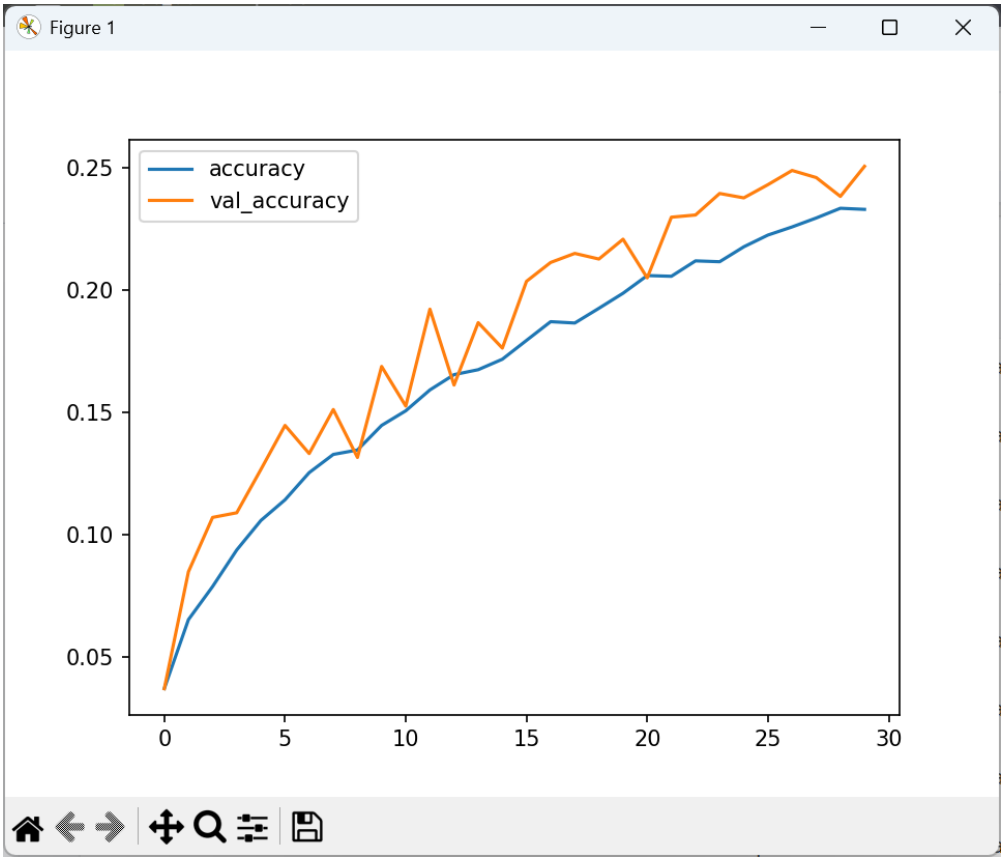
2.4. Експерименти

2.4.1. Значне збільшення Dropout

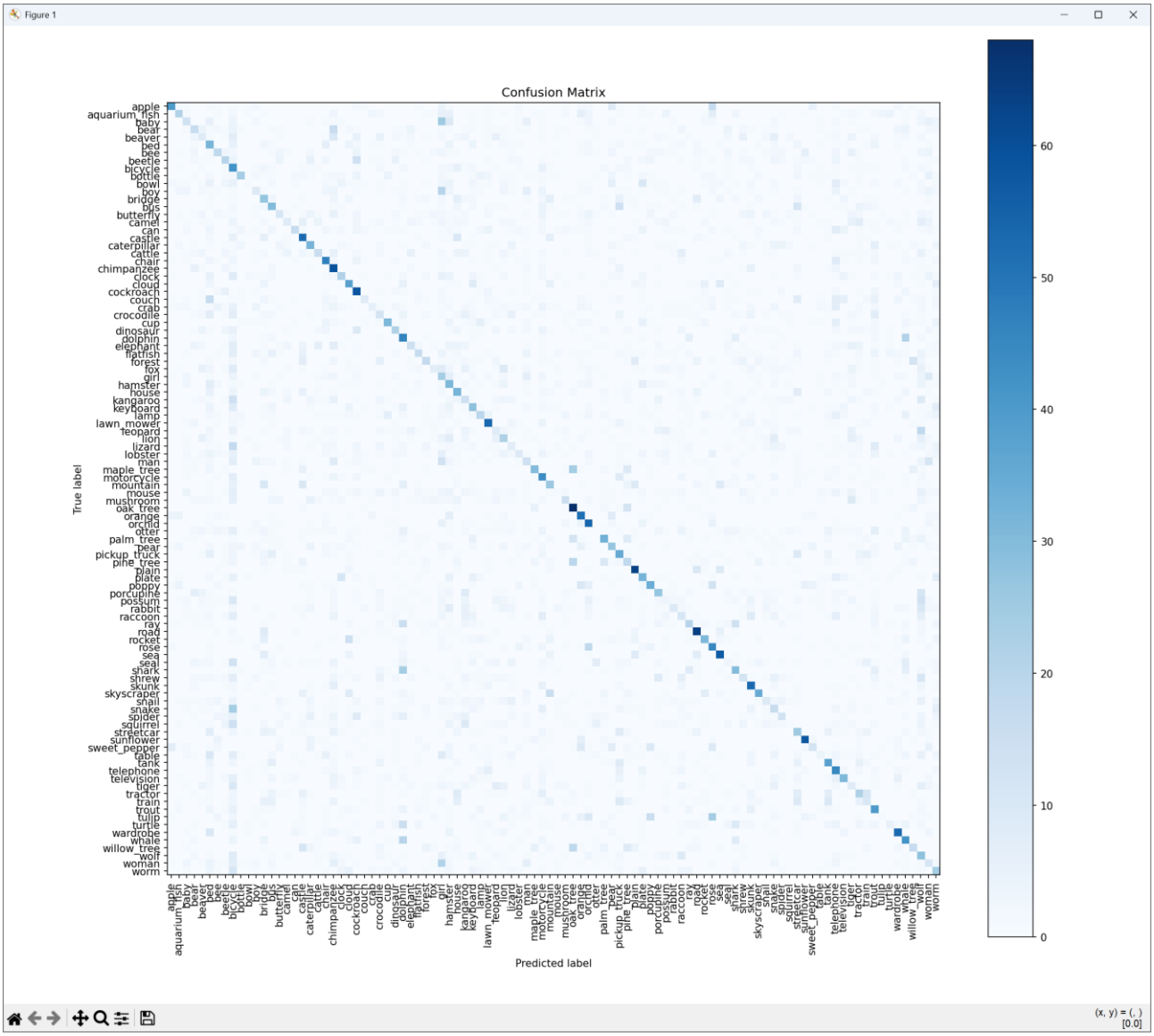
Модель перенавчається досить швидко, тож можна збільшити Dropout з 0.2 до 0.4:



[redacted] Функції втрат для першого експерименту)



[redacted] Функції точності для першого експерименту)

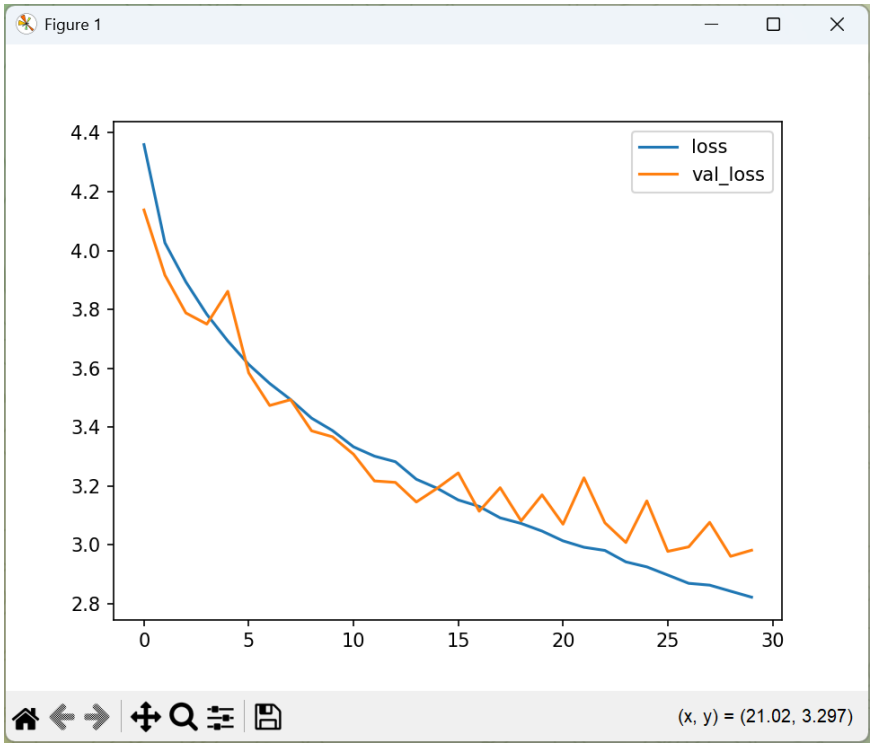


Матриця плутань для першого експерименту)

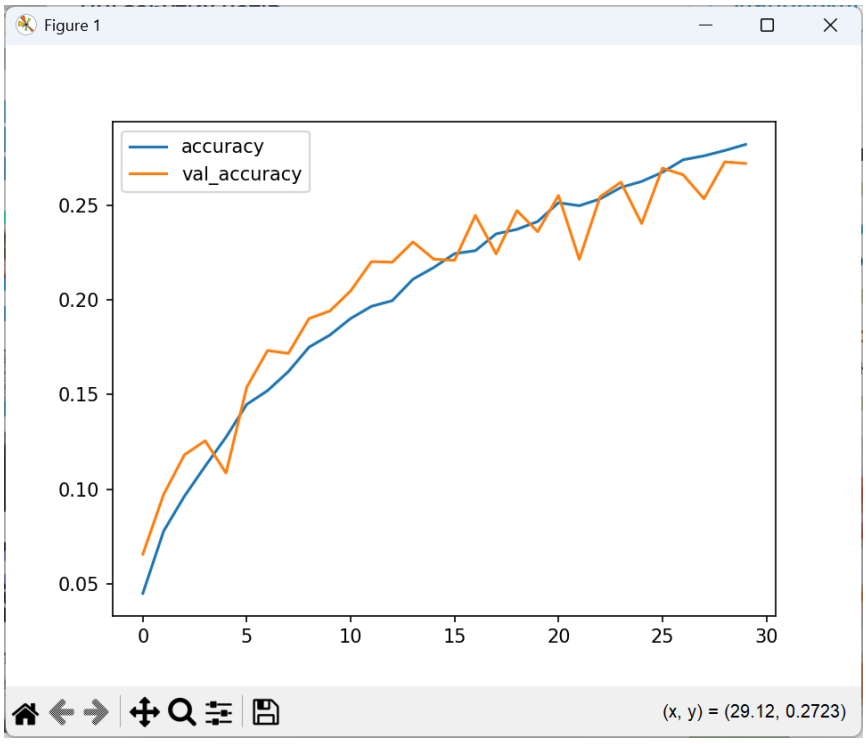
Отже, ця модель вже не перенавчається, але має схожу точність і гірші втрати.

2.4.2. Збільшення Dropout

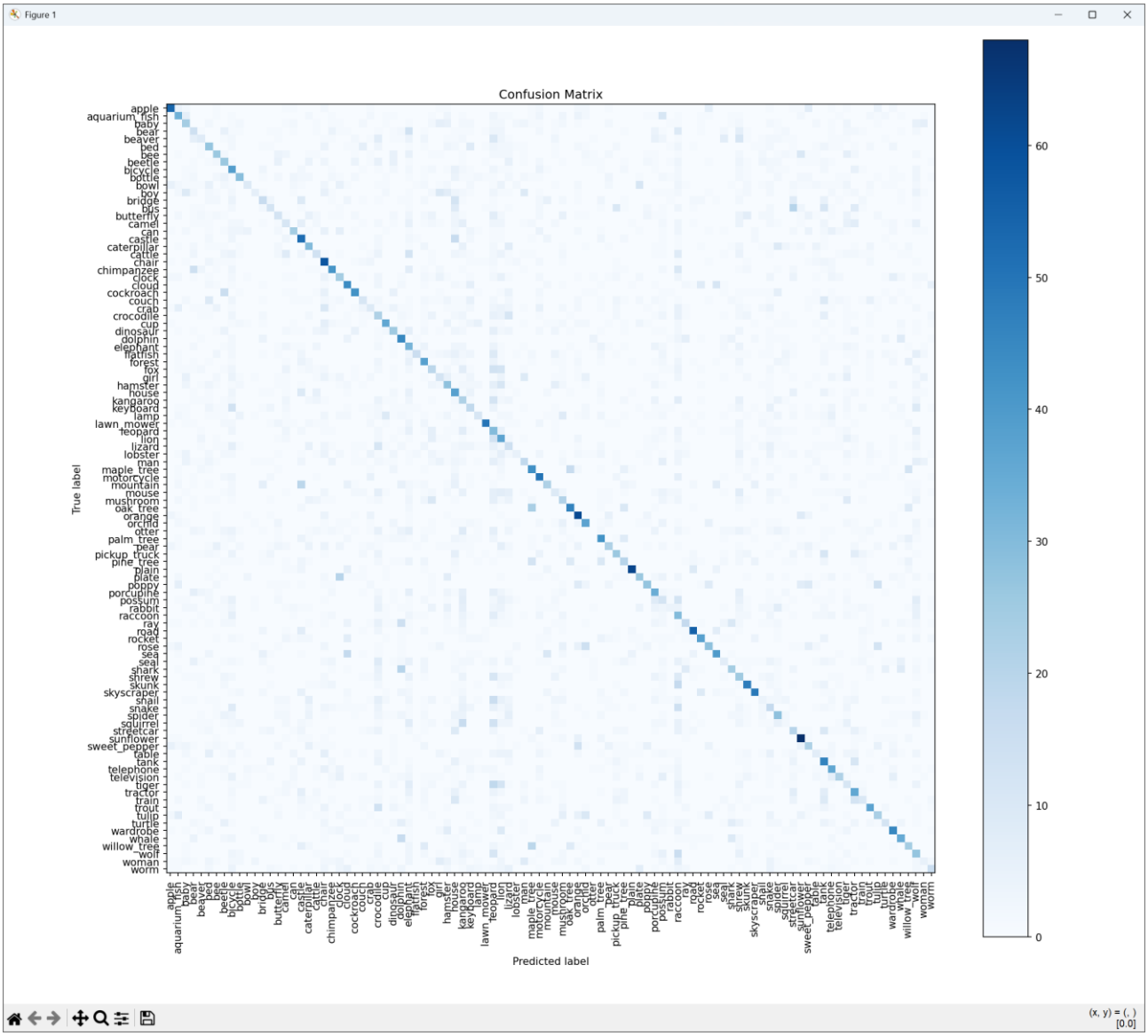
Встановлено значення dropout рівне 0.32:



[redacted] Функція втрат для другого експерименту)



[redacted] Функція точності для другого експерименту)

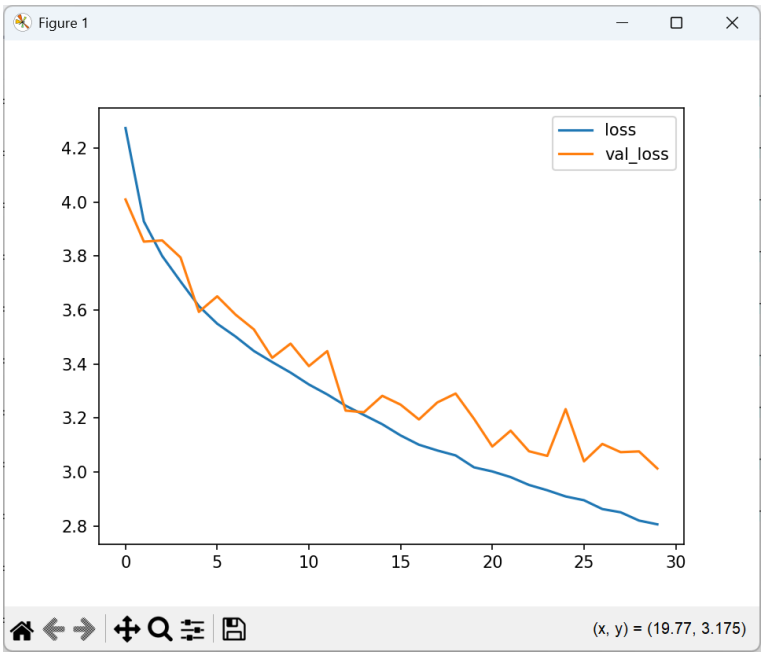


Матриця плутань для другого експерименту)

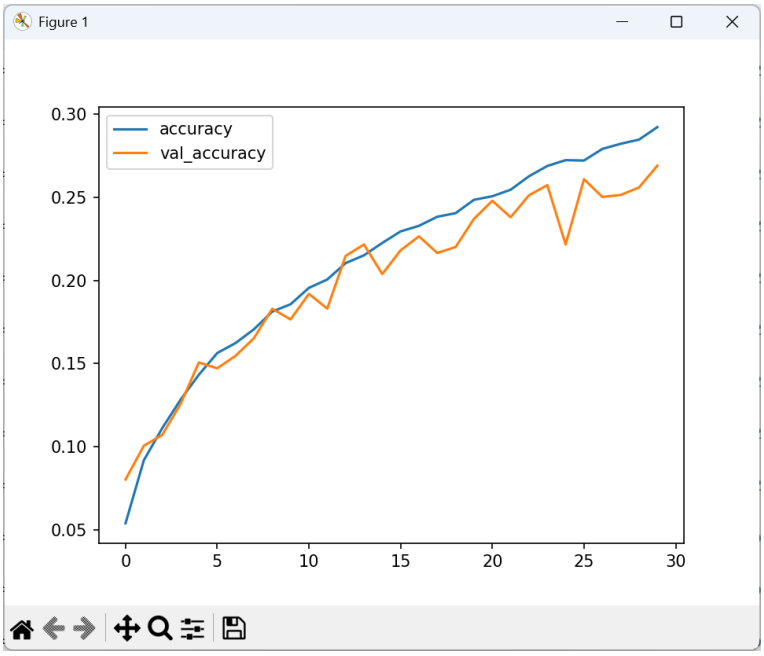
Дана модель має більшу точність і менші втрати ніж модель в першому експерименті, але меншу точність і схожі втрати ніж базова архітектура.

2.4.3. Збільшення розміру матриці Conv2D.

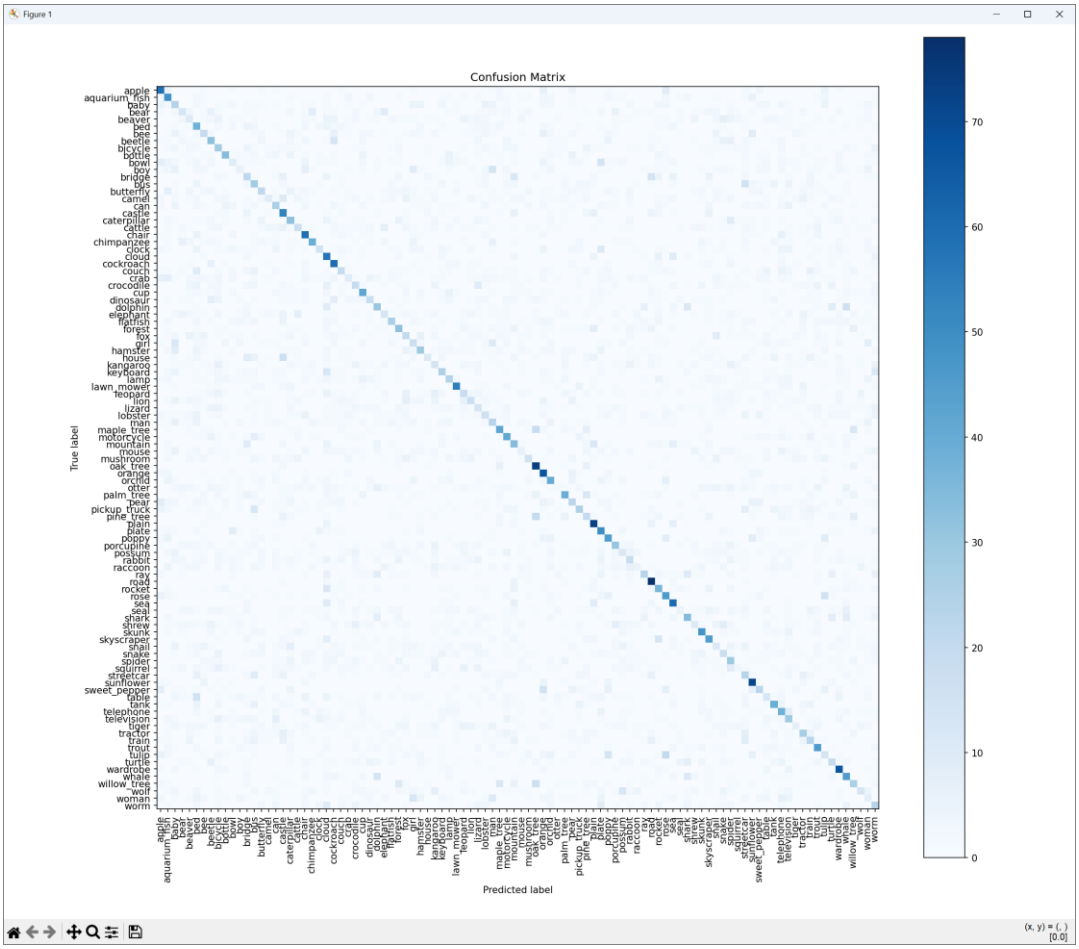
Відносно базової архітектури, збільшено розмір матриці в шарі Conv2D з 5x5 на 9x9. Dropout рівний 0.2:



Функція втрат для третього експерименту)



Функція точності для третього експерименту)



Матриця плутань для третього експеременту)