Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования «Южно-Уральский государственный университет»

(национальный исследовательский университет)

Институт естественных и точных наук

Кафедра прикладной математики и программирования

Отчет по лабораторной работе №5

по дисциплине «Современные нейросетевые технологии»

Авторы работы

Студент группы ЕТ-122

\_\_\_\_\_\_\_\_\_/ Безбородова И.В.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_2025 г.

Руководитель работы

\_\_\_\_\_\_\_\_\_\_/ Кичеев Д.М.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_2025 г.

Челябинск 2025

Задание: Цель: получить базовые навыки работы с одной из библиотек глубокого обучения (Caffe, Torch, TensorFlow или MXNet на выбор) на примере сверточных нейронных сетей.

Выполнение задания:

1. В ходе данного проекта я освоила базовые навыки работы с библиотекой глубокого обучения TensorFlow на примере построения и обучения полностью связанных нейронных сетей для задачи многоклассовой классификации.

Постановка математической задачи может быть описана на примере простой сверточной сети, с одном сверточным слоем, активацией ReLU.

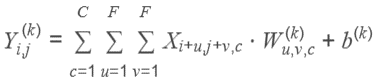
Пусть входное изображение имеет размерность H×W×C, где:

• H — высота изображения,

• W — ширина изображения,

• C — количество каналов (например, 3 для RGB).

Обозначим входное изображение как X, где X∈RH×W×C. Свертка применяется к входу X с фильтром W размером F×F×C. Пусть у нас есть K фильтров. Тогда выходная размерность карты признаков будет:



где:

• W(k) — параметры k-го фильтра,

• b (k) — смещение для k-го фильтра,

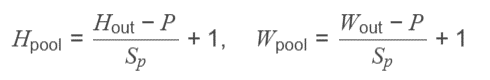
• Y (k) — выходная карта признаков для k-го фильтра. Итоговая размерность выхода после сверточного слоя:



После свертки применяется функция активации ReLU



Это обрезает все отрицательные значения в выходных данных. После слоя свертки в стандартном сверточном блоке используется слой пулинга. Пулинг уменьшает размер карты признаков, выбирая максимальное значение из локального окна размером P×P. Результирующая размерность после пулинга:



где:

• P — размер окна пулинга (например, 2×2),

• Sp — шаг пулинга. Формула для пулинга:



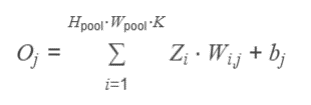
где window — область размером P×P, на которой берется максимум. После пулинга выходная размерность:



Результат пулинга (Z) преобразуется в одномерный вектор (flatten):



Данные поступают в полносвязный слой с N нейронами (где N — количество классов):

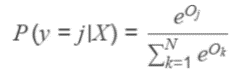


где:

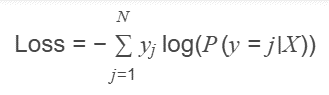
• Wi,j — веса полносвязного слоя,

• bj — смещения,

• Oj — логит для класса j. В выходном слое для многоклассовой классификации функция активации Softmax преобразует логиты Oj в вероятности:



где P(y=j∣X) — вероятность, что входное изображение принадлежит классу j. Используется кросс- энтропия в качестве функции потерь:

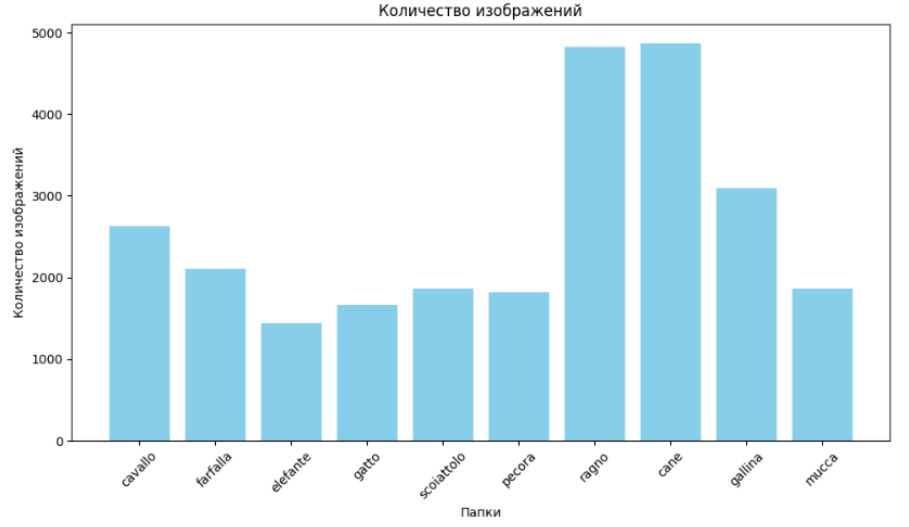


где:

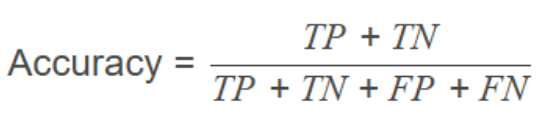
• yj — истинная метка класса (one-hot encoding),

• P(y=j∣X) — предсказанная вероятность для класса j.

Были использованы данные с kaggle (https://www.kaggle.com/datasets/alessiocorrado99/animals10/data). Данные содержат в себе 10 классов изображений животных. Каждый класс выделен в отдельную папку. Обучающий набор содержит 20947 изображений, тестовый набор - 5232 изображения. Изображения неравномерно распределены по папкам.



Изображения имеют разные параметры: минимальный размер изображения: 60x57 пикселей, максимальный размер изображения: 6720x6000 пикселей, средний размер изображения: 320.04x252.63. В связи с этим, было решено при загрузке изображений приводить их к среднему формату. 3. При тестировании моделей я использую метрику качества Accuracy, которая определяется следующим образом:



где:

- TP (True Positives) — количество истинных положительных предсказаний (правильно классифицированные объекты класса 1).

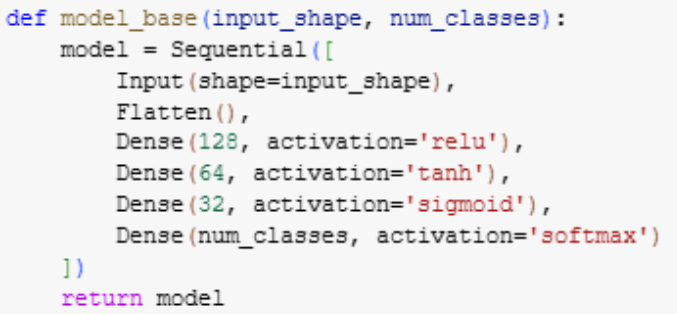
- TN (True Negatives) — количество истинных отрицательных предсказаний (правильно классифицированные объекты класса 0).

- FP (False Positives) — количество ложных положительных предсказаний (объекты класса 0, ошибочно классифицированные как класс 1).

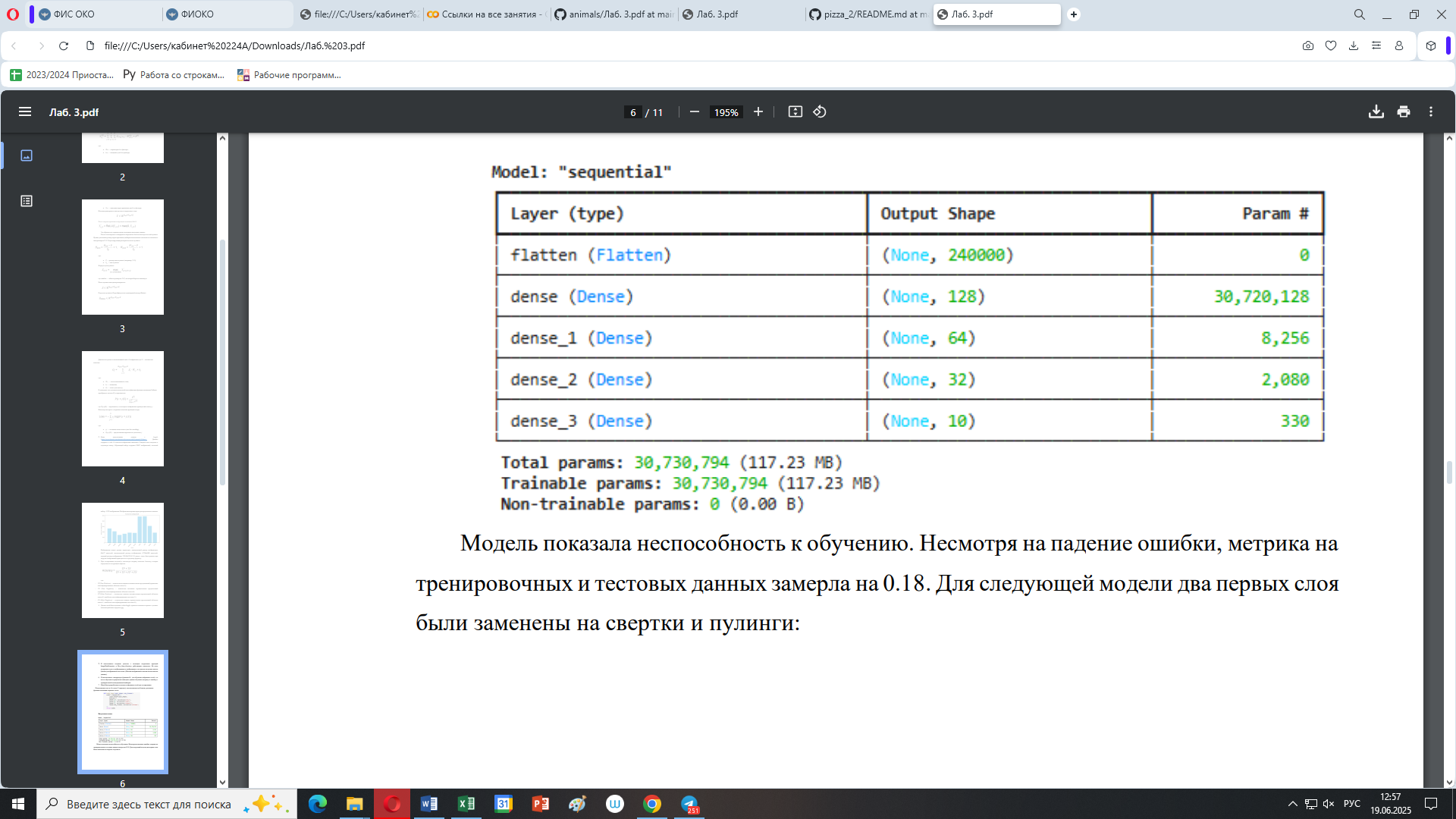
- FN (False Negatives) — количество ложных отрицательных предсказаний (объекты класса 1, ошибочно классифицированные как класс 0).

Данные мной были скачаны с сайта kaggle, хранятся локально в архиве с десятью папками файлами в формате jpg.

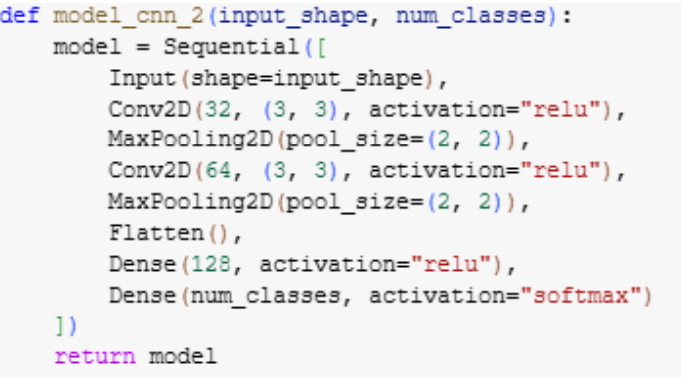
Я использовала создание датасета с помощью встроенных функций ImageDataGenerator и flow\_from\_directory, работающих совместно. На вход подавались пути к изображениям и изображения, а на выходе получили пакеты (batches) изображений и их метки. (Массив изображений и массив метов классов данных). 6. Я использовала стандартную функция fit для обучения нейронных сетей, а в после обучения я графически выводила данные обучения (метрику и ошибку) с тренировочной и валидационной выборки. 7. Мной были разработаны несколько нейронных сетей для тестирования. Полносвязная сеть из 4х слоев (3 скрытых), использовалась на базовая, различные функции активации скрытых слоев:



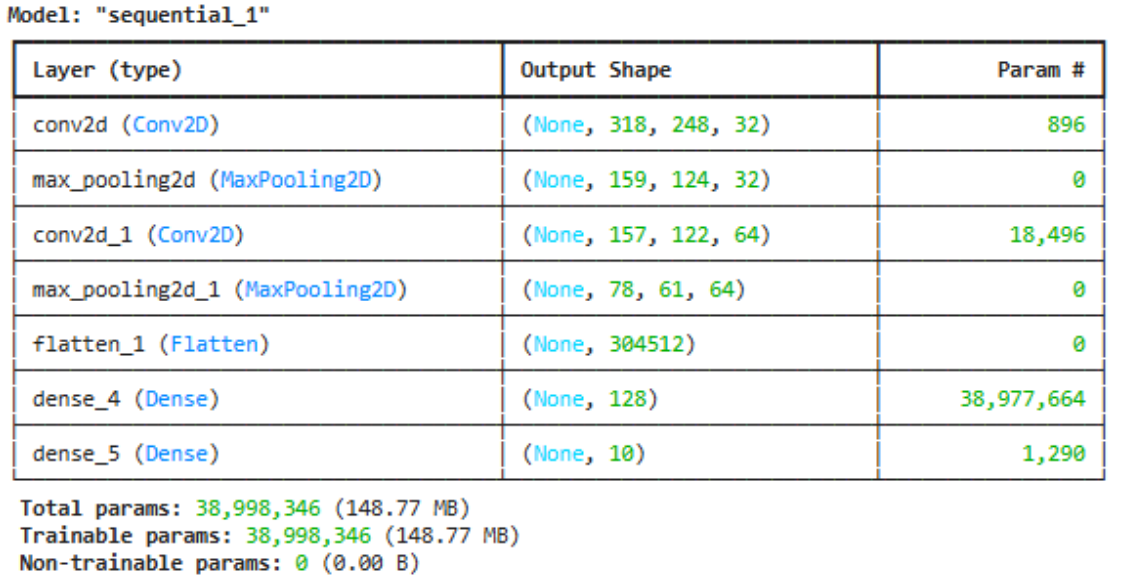
Визуальная схема:

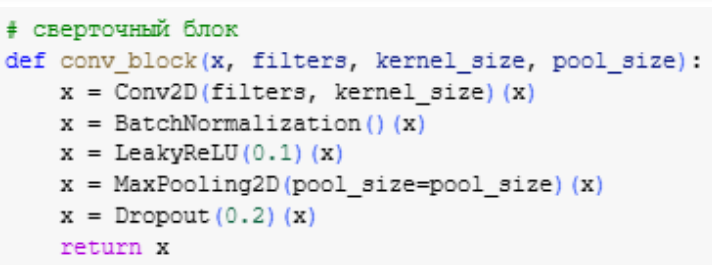


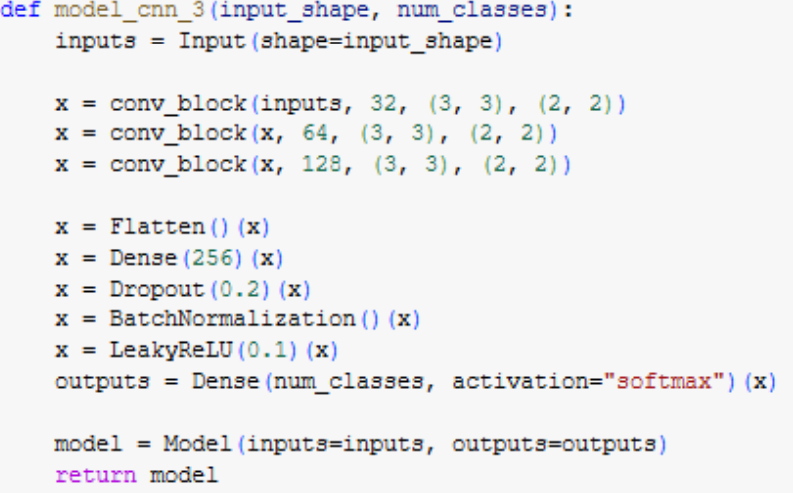
Модель показала неспособность к обучению. Несмотря на падение ошибки, метрика на тренировочных и тестовых данных замерла на 0.18. Для следующей модели два первых слоя были заменены на свертки и пулинги:

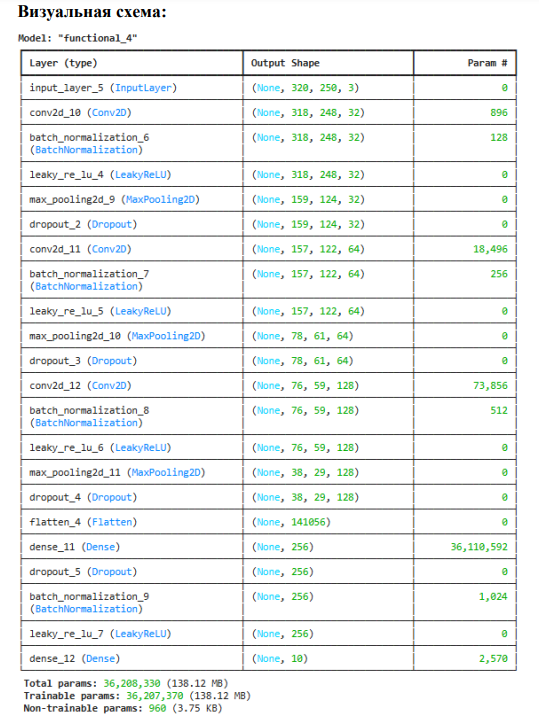


Визуальная схема:

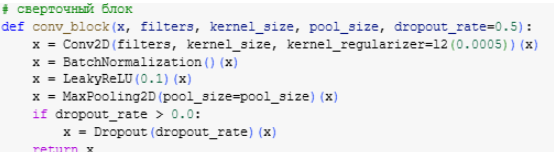


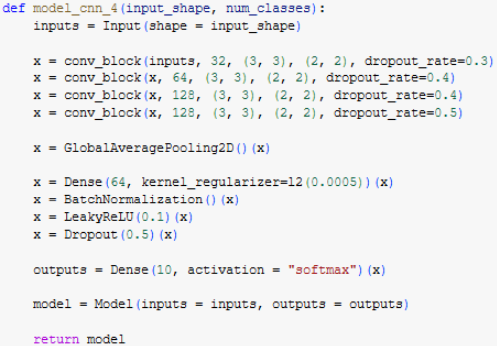
Модель показала результат лучше, чем полносвязная модель, но сильное переобучение начиная с 4 эпохи. Для следующей модели был добавлен слой свертки, а также для предотвращения переобучения добавлен обнуление нейронов (Dropout) и нормализацию данных.



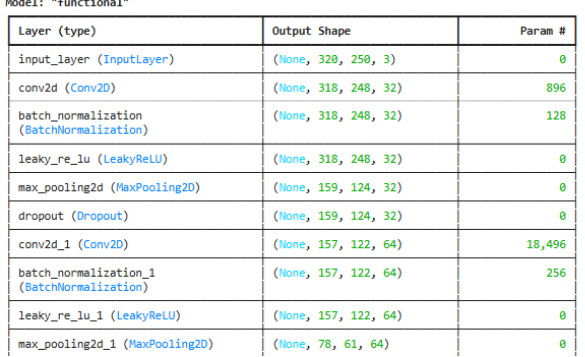


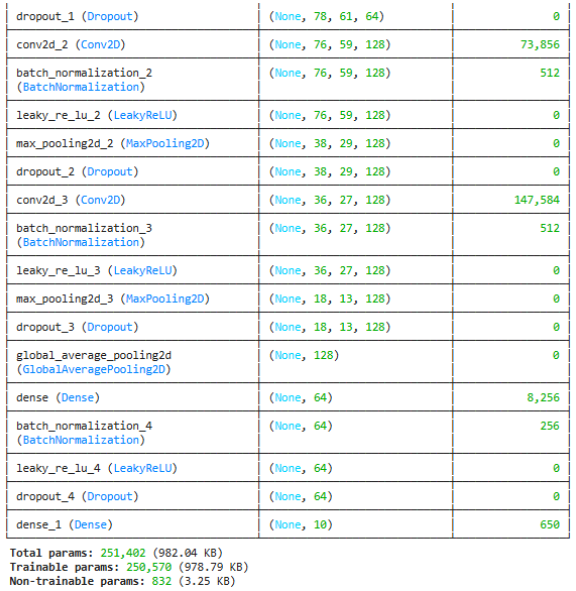
Модель уже не подсраивается под тренировочные данные, но переобучение сохраняется. Для следующей модели бы добавлен l2 регуляризацию в сверточный слой, добавим GlobalAveragePooling2D перед полносвязными слоями, сделаем разным исключение нейроннов на разных слоях (значение dropout).



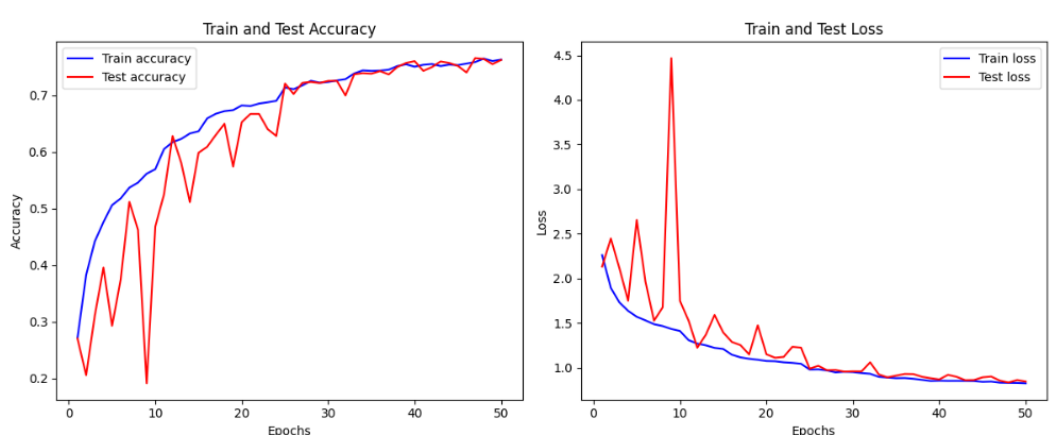


Визуальная схема:

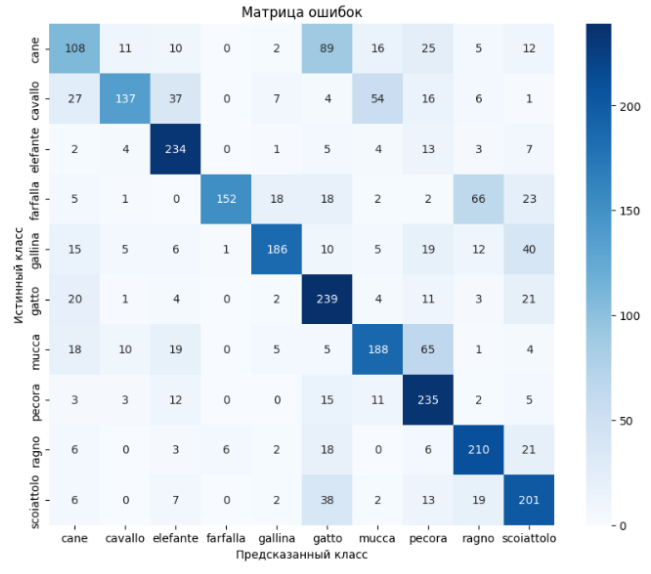




Последняя модель показала лучшие результаты в плате отсутствия переобучения. Была протестирована на большом количестве эпох. Добавлена схема уменьшения шага модели при увеличении числа эпох.



Модель неплохо справляется, достигла метрики 0.76 однако из-за неравномерности классов сильно подстраивается под мажорные классы. Модель переобучена на одинаковом количестве данных в классе, на 40 эпохах полученная метрика - 0.65.



Результаты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Модель | Время обучения | Кол-во сверт. слоев | Особенности | Accuracy |
| model\_base | 699 | 0 | Только полносвязные слои | 0.18 |
| model\_1 | 777 | 2 | Добавлены сверточные слои | 0.50 |
| model\_2 | 823 | 3 | Добавлен dropout и нормализация | 0.61 |
| model\_2 | 920 | 4 | Добавлена l2 регуляризацию, разные значения dropout на разных слоях | 0.45 |