

Министерство цифрового развития
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«Сибирский государственный университет телекоммуникаций и
информатики»
(СибГУТИ)

Кафедра прикладной математики и кибернетики

Отчёт

по лабораторной работе № 6 «Нейронные сети с использованием Pytorch для
классификации изображений»

Выполнил:

студент группы ИП-312

Прозоренко К.В

Работу проверил: старший преподаватель
кафедры ПМиК
Дементьева К.И.

Новосибирск 2025 г.

Цель: Освоить на практике принципы построения, обучения и оценки нейронных сетей с использованием Pytorch.

Задание

Выберите один набор данных из предложенных:

1. Cats-vs-Dogs
2. Pizza or Not Pizza?
3. Broken Eggs
4. Muffin vs chihuahua
5. HORSES-CAMELS
6. Mask_Detection
7. Happy-Sad Peoples

Я выбрал 1 датасет

Задание 1:

1. Загрузите датасет с правильными преобразованиями
2. Выведите информацию о размерах тренировочной и тестовой выборок
3. Визуализируйте 10 случайных изображений с подписями

Задание 2.

Создайте сверточную нейронную сеть для классификации изображений в соответствии с вашим вариантом. Обучите её и выполните предсказание на нескольких изображениях не из набора данных.

Задание 3.

Проанализируйте ошибки на процессе обучения и тестирования модели.

Ход работы

Загрузил датасет, выполнил преобразования, получил информацию о выборках и вывел 10 случайных изображений с подписями.

Для обучения применялись преобразования: изменение размера, аугментации (например, повороты/отражения), перевод в тензор и нормализация; для теста — только “честные” преобразования без аугментаций.

В датасетах Cats vs Dogs встречаются повреждённые/обрезанные изображения, из-за которых DataLoader может падать при чтении. Для решения использован параметр `is_valid_file` в `ImageFolder`, который позволяет отфильтровать некорректные файлы на этапе построения датасета.



Построена CNN-модель из нескольких сверточных блоков (Conv + активация + pooling) и полносвязной “головы” для классификации на 2 класса. Модель обучалась на GPU (CUDA) с функцией потерь для классификации и оптимизатором (например, Adam), при этом считались loss и accuracy на train и test по эпохам.

```
Device: cuda
Epoch 1/5 | train loss=0.8095 acc=0.6686 | test loss=0.5623 acc=0.7002
Epoch 2/5 | train loss=0.5117 acc=0.7473 | test loss=0.4617 acc=0.7804
Epoch 3/5 | train loss=0.4617 acc=0.7818 | test loss=0.5946 acc=0.6870
Epoch 4/5 | train loss=0.4184 acc=0.8095 | test loss=0.4349 acc=0.7976
Epoch 5/5 | train loss=0.3822 acc=0.8291 | test loss=0.4092 acc=0.8064
```

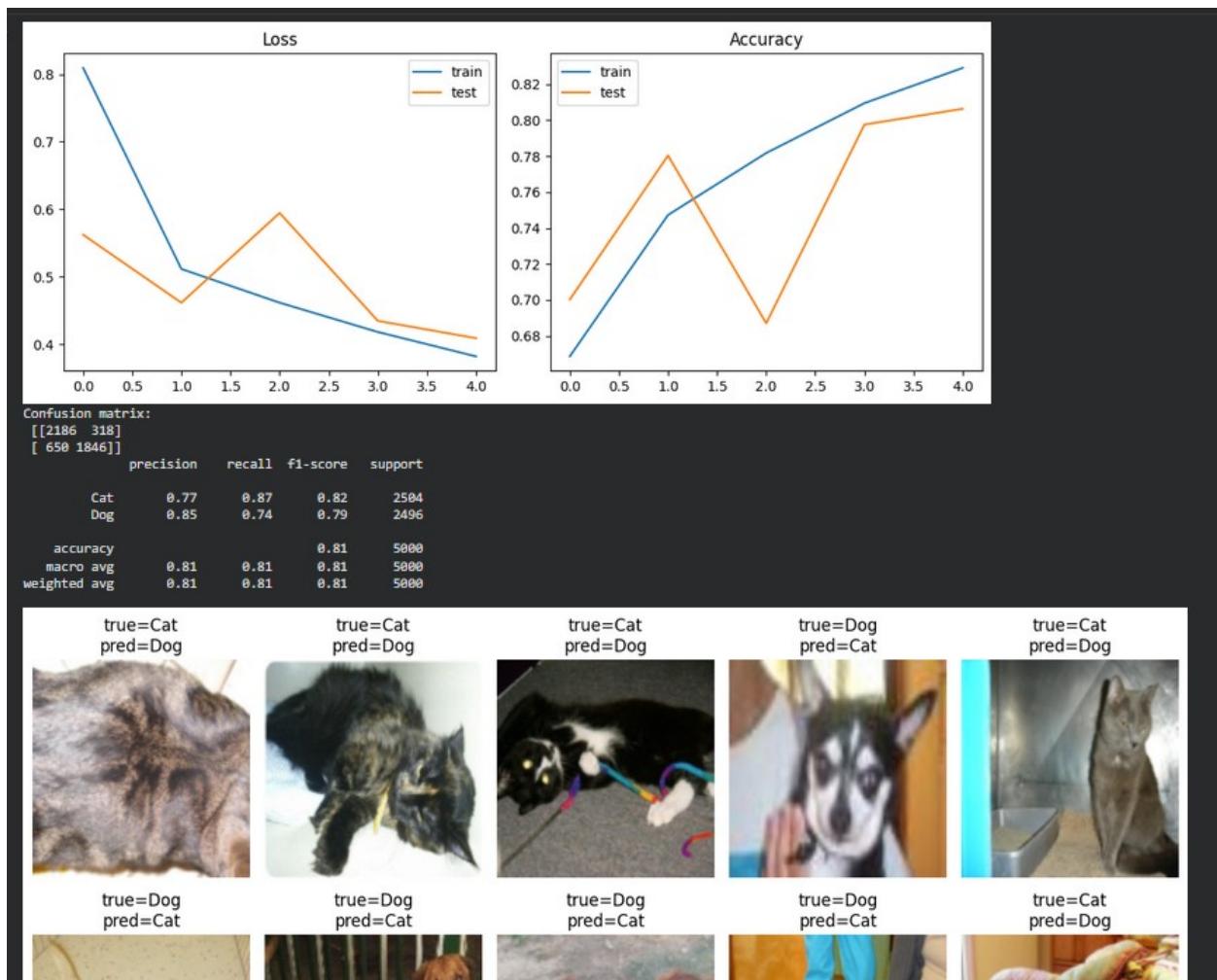
Выполнено предсказание на нескольких изображениях, не входящих в датасет, с выводом предсказанного класса и уверенности (softmax).

```
** dog.jpg -> Dog conf= 0.8883  
cat.jpg -> Cat conf= 0.7379
```

Построены матрица ошибок и отчёт с precision/recall/F1 для каждого класса, что позволяет понять, какие ошибки преобладают (например, “собаку приняли за кошку” и наоборот).

Показаны изображения, на которых модель ошиблась, с true/pred подписями, чтобы визуально выделить причины ошибок (размытие, плохой свет, необычный ракурс, частичная видимость объекта).

По результатам тестирования получена accuracy около 0.81, также видна разница в качестве по классам (по метрикам и матрице ошибок).



Ссылка на Google Collab:

https://colab.research.google.com/drive/1_2dFdNYPMKmi4gO7psNUmaXZNZUKSsN?usp=sharing

