

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 № 5 «Нейронные сети для обработки 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 для решения базовых задач компьютерного зрения.

Задание

1. Распределение вариантов

Формула: (Номер студента в списке) mod 6 + 1

Мой датасет по варианту:

6. Food-101 - Блюда питания (101 класс)

2. Создание архитектуры нейронной сети

Создайте и обучите сверточную нейронную сеть для решения задачи классификации изображений.

Вариант архитектуры выбирается по формуле

(Номер студента в списке) mod 8 + 1

Вариант: 2

Архитектура CNN: 3 сверточных слоя + 2 полносвязных

Регуляризация: BatchNormalization

Оптимизатор: SGD

3. Сделайте анализ графиков обучения и оценку точности на тестовой выборке.

Ход выполнения

Загрузка и разбиение датасета

```
*** Downloading from https://www.kaggle.com/api/v1/datasets/download/kmader/food41?dataset_version_number=5...
100%|██████████| 5.30G/5.30G [04:00<00:00, 23.6MB/s]Extracting files...
Path to dataset files: /root/.cache/kagglehub/datasets/kmader/food41/versions/5
images_root: /root/.cache/kagglehub/datasets/kmader/food41/versions/5/images
meta_root: /root/.cache/kagglehub/datasets/kmader/food41/versions/5/meta
num_classes: 101
Train/Val/Test: 68175 7575 25250
```

Все изображения приведены к размеру $IMG_SIZE = (160, 160)$ и нормализованы в диапазон $[0; 1]$.

Данные подаются батчами BATCH = 64 с использованием tf.data и предварительной подгрузки (prefetch) для ускорения.

Model: "functional_1"		
Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 160, 160, 3)	0
augment (Sequential)	(None, 160, 160, 3)	0
conv2d (Conv2D)	(None, 160, 160, 32)	896
batch_normalization (BatchNormalization)	(None, 160, 160, 32)	128
activation (Activation)	(None, 160, 160, 32)	0
max_pooling2d (MaxPooling2D)	(None, 80, 80, 32)	0
conv2d_1 (Conv2D)	(None, 80, 80, 64)	18,496
batch_normalization_1 (BatchNormalization)	(None, 80, 80, 64)	256
activation_1 (Activation)	(None, 80, 80, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 40, 40, 64)	0
conv2d_2 (Conv2D)	(None, 40, 40, 128)	73,856
batch_normalization_2 (BatchNormalization)	(None, 40, 40, 128)	512
activation_2 (Activation)	(None, 40, 40, 128)	0
max_pooling2d_2 (MaxPooling2D)	(None, 20, 20, 128)	0
flatten (Flatten)	(None, 51200)	0
dense (Dense)	(None, 256)	13,187,456
batch_normalization_3 (BatchNormalization)	(None, 256)	1,024
activation_3 (Activation)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32,896
batch_normalization_4 (BatchNormalization)	(None, 128)	512
activation_4 (Activation)	(None, 128)	0
dense_2 (Dense)	(None, 101)	13,029

Total params: 13,249,061 (50.54 MB)
Trainable params: 13,247,845 (50.54 MB)
Non-trainable params: 1,216 (4.75 KB)

На картинке результат создания и настройки сверточной нейронной сети.

(None, 160, 160, 3) - вход: изображение 160x160x3, а None означает “любой размер батча”.

После каждого MaxPooling2D размеры по ширине/высоте уменьшаются в 2 раза: 160-80-40-20, что и видно в Output Shape.

Flatten (None, 51200) означает, что тензор 20x20x128 разворачивается в вектор длиной $20 \times 20 \times 128 = 51200$, $20 \times 20 \times 128 = 51200$.

Обучение модели и графики

Оптимизатор: SGD (learning_rate=0.01, momentum=0.9, nesterov=True).

Функция потерь: sparse_categorical_crossentropy.

Метрика: accuracy.

Количество эпох: EPOCHS = 15



Ссылка на Google Collab:

<https://colab.research.google.com/drive/1RIB5BVTRT9s37YllIaeTCKR7TmLEdA?usp=sharing>