

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА 33

ОТЧЕТ ЗАЩИЩЕН С ОЦЕНКОЙ _____

ПРЕПОДАВАТЕЛЬ

Ассистент

должность, уч. степень, звание

Н.С. Красников

подпись, дата

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 3

**СВЕРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ В ЗАДАЧАХ РАСПОЗНАВАНИЯ
ОБРАЗОВ**

по курсу: ОСНОВЫ МАШИННОГО ОБУЧЕНИЯ

СТУДЕНТ ГР. №

3031

номер группы

подпись, дата

М.В. Вдовин

инициалы, фамилия

Санкт-Петербург

2023

1. Цель работы:

Изучение архитектуры и принципов функционирования сверточных нейронных сетей. Приобретение навыков разработки сверточных нейронных сетей в фреймворках MatLab и SuperVisely.

2. Задание:

1. Изучить архитектуру сверточной нейронной сети и этапы ее разработки в MatLab.
2. Спроектировать архитектуру сверточной нейронной сети для решения задачи многоклассовой классификации в соответствии со своим вариантом.
3. В MatLab создать сверточную нейронную сеть, реализующую функцию распознавания заданного изображения и обучающуюся с помощью алгоритма обратного распространения ошибки.
4. Обучить сверточную нейронную сеть. Целевая точность классификации должна быть не ниже 95%.
5. Протестировать нейронную сеть.
6. Оформить отчет по лабораторной работе.

3. Ход работы

3.1. Теоретические сведения

Сверточные нейронные сети (Convolutional Neural Networks или CNNs) представляют собой класс глубоких нейронных сетей, которые обладают специализированной архитектурой для обработки и анализа визуальных данных, таких как изображения. Они были впервые предложены для решения задач распознавания образов и классификации изображений, но впоследствии успешно применяются в различных задачах компьютерного зрения.

Основные компоненты сверточных нейронных сетей включают в себя:

1. Сверточные слои (Convolutional Layers).

Эти слои предназначены для обнаружения локальных паттернов в изображениях. Они используют фильтры (ядра) для сканирования изображения и извлечения признаков, таких как грани, углы и текстуры.

2. Подвыборка (Pooling Layers).

Слои подвыборки выполняют уменьшение размерности пространства признаков, сохраняя при этом ключевые информационные особенности. Это помогает уменьшить количество параметров модели и сделать ее более устойчивой к вариациям в данных.

3. Полносвязные слои (Fully Connected Layers).

После серии сверточных и подвыборочных слоев следует один или несколько полносвязных слоев, которые выполняют классификацию или регрессию на основе извлеченных признаков.

4. Функции активации.

Обычно используются функции активации, чтобы внести нелинейность в модель и обеспечить ее способность моделировать сложные зависимости.

Процесс обучения сверточной нейронной сети включает в себя подачу обучающих данных, в нашем случае изображений лиц, на вход сети, вычисление ошибки, разницы между предсказанными и фактическими значениями, и корректировку весов сети с использованием оптимизационных алгоритмов, таких как стохастический градиентный спуск.

Рассмотрим принцип работы данного вида сетей на примере распознавания лиц знаменитостей.

3.2. Практическая часть

Возьмем подготовленный датасет с лицами из первой лабораторной работы, который был сделан в SuperVisely - Рисунок 1. Используем ресурс Kaggle для реализации сверточной сети. Выберем один из видов архитектуры – MobileNet, для которой используется функция активации ReLu. После чего фотографии будут приведены в нормальный вид (нормализованы) - Рисунок 2. Затем запусти обучение сети, результаты которой можно увидеть на Рисунок 3. Протестируем полученные результаты на фотографиях - Рисунок 4. График зависимости loss-функции от количества пройденных эпох приведен на Рисунок 5.



Рисунок 1. Dataset с лицами знаменитостей



Рисунок 2. Нормализованные фотографии

```

epoch: 0%|          | 0/10 [00:00<?, ?it/s]
loss 2.198559552860535
acc 0.3573487031700288
epoch: 10%|█        | 1/10 [00:37<05:35, 37.33s/it]

Epoch 001 train_loss: 2.1986 val_loss 1.7361 train_acc 0.3573 val_acc 0.4885
loss 1.492156806528053
acc 0.5547550432276657
epoch: 20%|██       | 2/10 [01:02<04:03, 30.46s/it]

Epoch 002 train_loss: 1.4922 val_loss 1.5161 train_acc 0.5548 val_acc 0.5201
loss 0.7777887277025998
acc 0.734149855907781
epoch: 30%|███      | 3/10 [01:28<03:18, 28.31s/it]

Epoch 003 train_loss: 0.7778 val_loss 1.0378 train_acc 0.7341 val_acc 0.6839
loss 0.4302582894519014
acc 0.8602305475504323
epoch: 40%|████     | 4/10 [01:54<02:43, 27.29s/it]

Epoch 004 train_loss: 0.4303 val_loss 0.9214 train_acc 0.8602 val_acc 0.6983
loss 0.21347240137435516
acc 0.9322766570605188
epoch: 50%|█████    | 5/10 [02:19<02:13, 26.61s/it]

Epoch 005 train_loss: 0.2135 val_loss 1.2404 train_acc 0.9323 val_acc 0.7098
loss 0.17562476740962146
acc 0.9445244956772334
epoch: 60%|██████   | 6/10 [02:45<01:44, 26.11s/it]

Epoch 006 train_loss: 0.1756 val_loss 0.8700 train_acc 0.9445 val_acc 0.7960
loss 0.08469319040283174
acc 0.978386167146974
epoch: 70%|███████  | 7/10 [03:10<01:17, 25.76s/it]

Epoch 007 train_loss: 0.0847 val_loss 1.0369 train_acc 0.9784 val_acc 0.7414
loss 0.03266210458979483
acc 0.989193083573487
epoch: 80%|████████ | 8/10 [03:35<00:51, 25.56s/it]

Epoch 008 train_loss: 0.0327 val_loss 0.9702 train_acc 0.9892 val_acc 0.7759
loss 0.02797665638964548
acc 0.9942363112391931
epoch: 90%|█████████| 9/10 [04:00<00:25, 25.44s/it]

Epoch 009 train_loss: 0.0280 val_loss 0.8473 train_acc 0.9942 val_acc 0.8017
loss 0.005139682896204443
acc 0.9965590778097982
epoch: 100%|██████████| 10/10 [04:25<00:00, 26.58s/it]

Epoch 010 train_loss: 0.0051 val_loss 0.8518 train_acc 0.9986 val_acc 0.8161

```

Рисунок 3. Результаты обучения сверточной сети

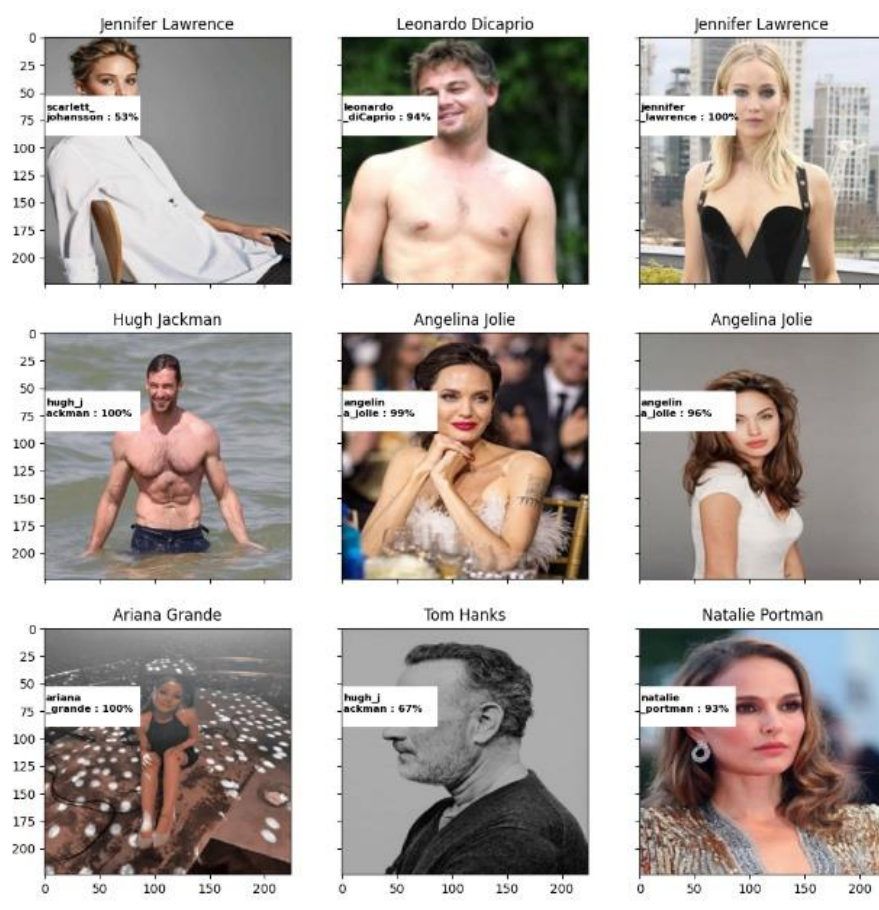


Рисунок 4. Пример работы

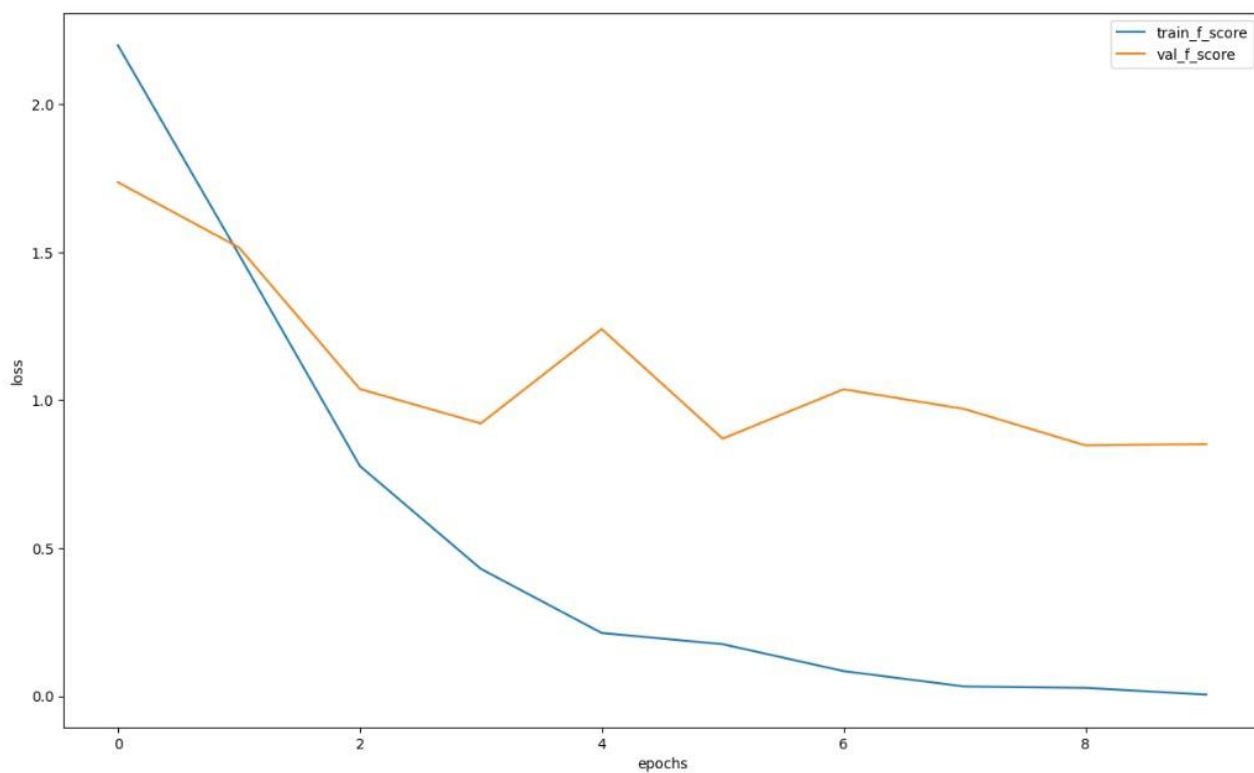
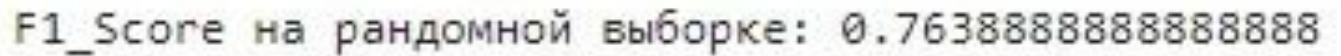


Рисунок 5. График зависимости loss-функции от количества эпох

4. Вывод

В ходе лабораторной работы были изучена архитектура и принципы работы сверточных нейронных сетей, приобретены навыки разработки сверточных нейронных сетей, с помощью ресурса Kaggle. Во время испытаний на рандомной выборке было достигнуто значение метрики Fone-score 0,764 - Рисунок 6.



F1_Score на рандомной выборке: 0.7638888888888888

Рисунок 6. F1_score на случайно выборке