МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Систем обработки информации и управления»

ОТЧЕТ

**Домашнее задание**

по дисциплине«Проектирование интеллектуальных систем»

Тема: «Web-приложение и сбор датасета»

ИСПОЛНИТЕЛЬ:                 \_\_\_

ФИО

группа ИУ5-24М               \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

" "\_\_\_\_\_\_\_\_\_\_\_2024 г.

ПРЕПОДАВАТЕЛЬ:             \_\_\_Канев А.И.\_\_\_\_\_

ФИО

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

"\_\_"\_\_\_\_\_\_\_\_\_2024г.

Москва  -  2024

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **Задание**

Необходимо создать и разметить собственный набор данных, состоящий из изображений. Набор содержит не менее 3 классов и не менее 100 экземпляров каждый. Изображения можно скачать из интернета или объединить несколько существующих датасетов. Создать web-приложение для классификации изображений полученного набора данных. Использовать аугментацию данных, регуляризацию, перенос обучения.

# **Часть 1. Сбор датасета.**

По заданию необходимо обучить нейронную сеть, способную распознавать различные виды насекомых: Helicopter, privateJet, v-22.

Для решения этой задачи необходимо найти в интернете изображения данных насекомых для формирования обучающей и тестовой выборки. Для удобства скачивания большого числа изображений воспользуемся расширением, затем вручную проверим изображения на корректность.

В результате получилось:

1. Helicopter - 100 изображения
2. privateJet - 100 изображений
3. v-22 - 100 изображений

Примеры изображений указаны на рисунках ниже:



Рисунок 1 – вертолет



Рисунок 2 – частный джет



Рисунок 3 – v-22

# **Часть 2. Обучение нейронной сети**

Посмотрим результаты обучения полносвязной нейронной сети, параметры которой заданы в блокноте. Результаты представлены в таблице 1.

Таблица 1. Результаты полносвязной нейронной сети.

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| helicopter | 1 | 0.9875 | 0.9937 |
| privateJet | 0.9875 | 0.9875 | 0.9875 |
| v-22 | 0.9877 | 1 | 0.9938 |
| Accuracy |  | | 0.9917 |
| **Test** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| helicopter | 0.7059 | 0.6000 | 0.6486 |
| privateJet | 0.8125 | 0.6500 | 0.7222 |
| v-22 | 0.6667 | 0.9000 | 0.7600 |
| Accuracy |  | | 0.7167 |

Результаты – не самые лучшие. В среднем, модель ошибается чаще, чем каждый третий раз.

Рассмотрим модель ResNet с конфигурацией из ЛР4. Будем обучать её с разморозкой всех весов. Результат представлен в таблице 2.

Таблица 2. Результаты работы сети из ЛР4.

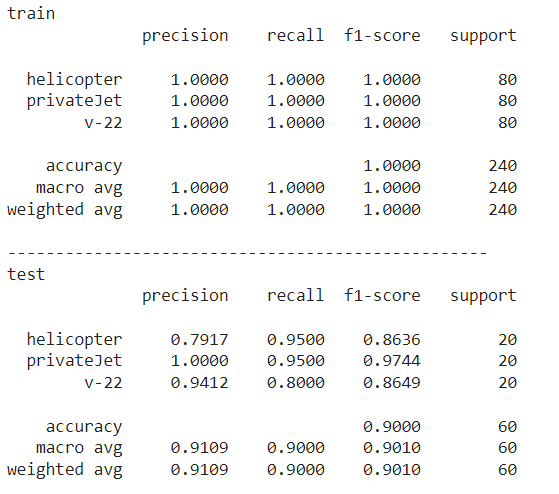
|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| helicopter | 1 | 0.9875 | 0.9937 |
| privateJet | 0.9875 | 0.9875 | 0.9875 |
| v-22 | 0.9877 | 1 | 0.9938 |
| Accuracy |  | | 0.9917 |
| **Test** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| helicopter | 0.7500 | 0.6000 | 0.6667 |
| privateJet | 0.7778 | 0.7000 | 0.7368 |
| v-22 | 0.7308 | 0.9500 | 0.8261 |
| Accuracy |  | | 0.7500 |

Результат уже, в среднем, лучше, но не сильно. Улучшилась точность при распознавании частного джета.

Как можно заметить, результат в 75 процента точности свидетельствует об улучшении распознающих способностей модели, однако показатели являются не слишком впечатляющий. Попробуем улучшить показатели модели, варьируя гиперпараметрами, а также меняя архитектуру сети. Результаты всех вариация приведены в таблице 3.

Таблица 3. Результаты.

|  |  |  |  |
| --- | --- | --- | --- |
| Конфигурация сети | Гиперпараметры | Точность | Пояснения |
| Полносвязная нейронная сеть | lr = 0.0003,  batch\_size = 32,  epochs = 60  weight\_decay=1e-5  optimizator = SGD | test = 71,1%, train = 99,1% | Полносвязная нейронная сеть |
| Resnet  Keep\_last = 65 | lr = 0.0003,  batch\_size = 32,  epochs = 60  weight\_decay=1e-5  optimizator = SGD | test = 75%, train = 100% | Размороженная mobileNet |
| Resnet  Keep\_last = 65 | lr = 0.00015,  batch\_size = 32,  epochs = 120  weight\_decay=1e-5  optimizator = SGD | test = 72%, train = 93% | Уменьшили learning rate и увеличили число эпох |
| Resnet  Keep\_last = 65 | lr = 0.0003,  batch\_size = 32,  epochs = 60  weight\_decay=1e-5  optimizator = ADAM | test = 89,5%, train = 100% | Вернулись к предыдущим параметрам, заменили метод оптимизации на ADAM |
| Resnet  Keep\_last = 65 | lr = 0.0003,  batch\_size = 64,  epochs = 120  weight\_decay=1e-5 | test = 88,3%, train = 100% | Изменили размер батча, увеличили число эпох |
| Resnet  Keep\_last = 65 | lr = 0.00015,  batch\_size = 64,  epochs = 240  weight\_decay=1e-5 | test = 90%, train = 100% | Уменьшили скорость обучения, увеличили число эпох |



Наилучший результат показала полностью размороженная модель Resnet с оптимизатором градиентного спуска Adam, гиперпараметром learning rate, равным 0.00015 и batch\_size, равным 64. В веб-приложении будем использовать эту конфигурацию.

# **Часть 3. Web-приложение.**

Выполняем всё по инструкции, заносим в Django нашу обученную модель. При правильной настройке - получаем следующий результат.

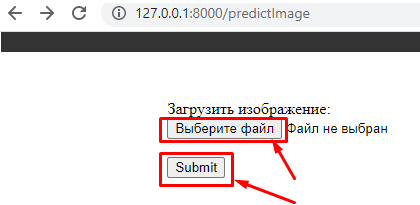


Рисунок 5 - Базовый шаблон html

Сделаем более красивое оформление, добавим вывод изображения. Результат на рисунке ниже.

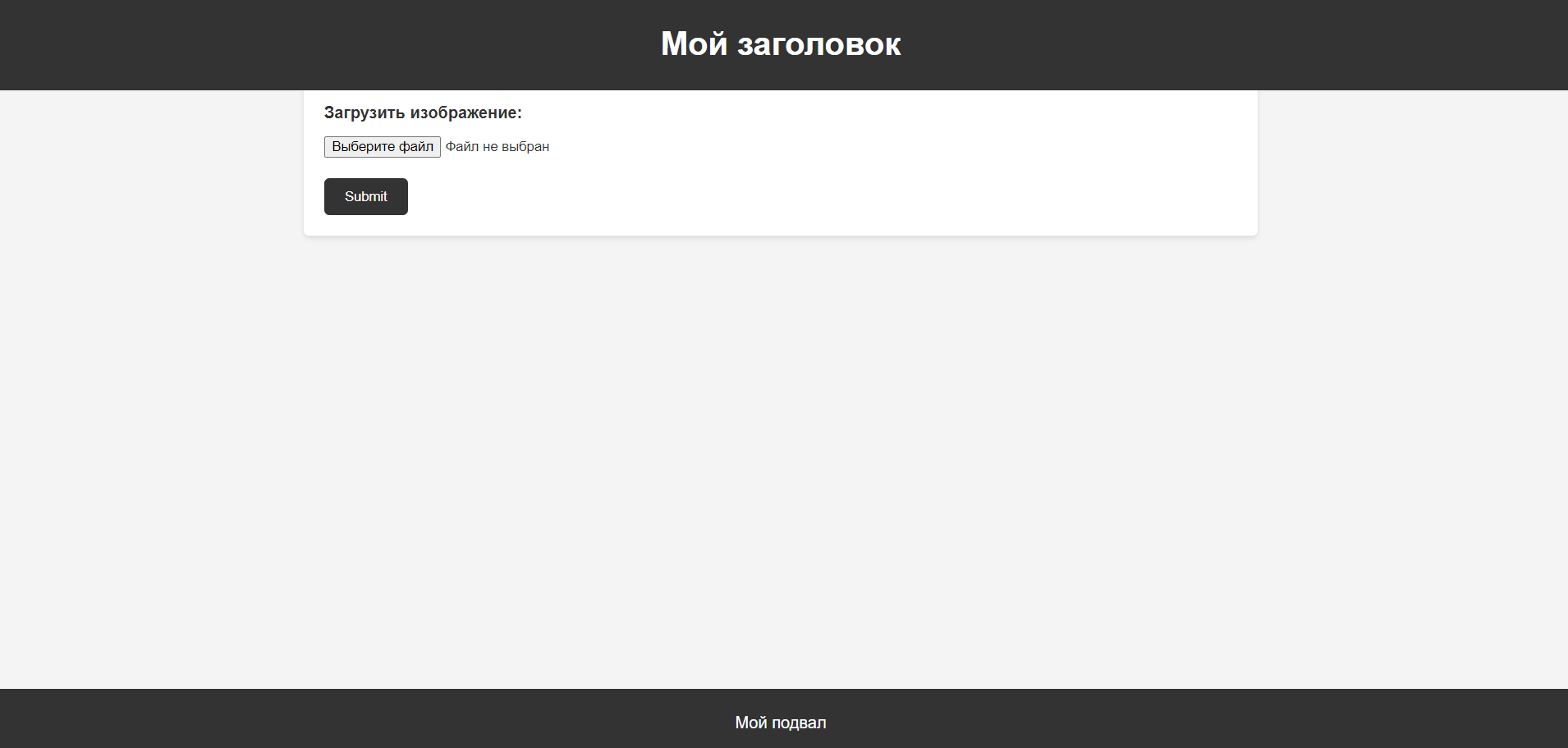


Рисунок 6 - Часть оформления. Загрузка рисунка.

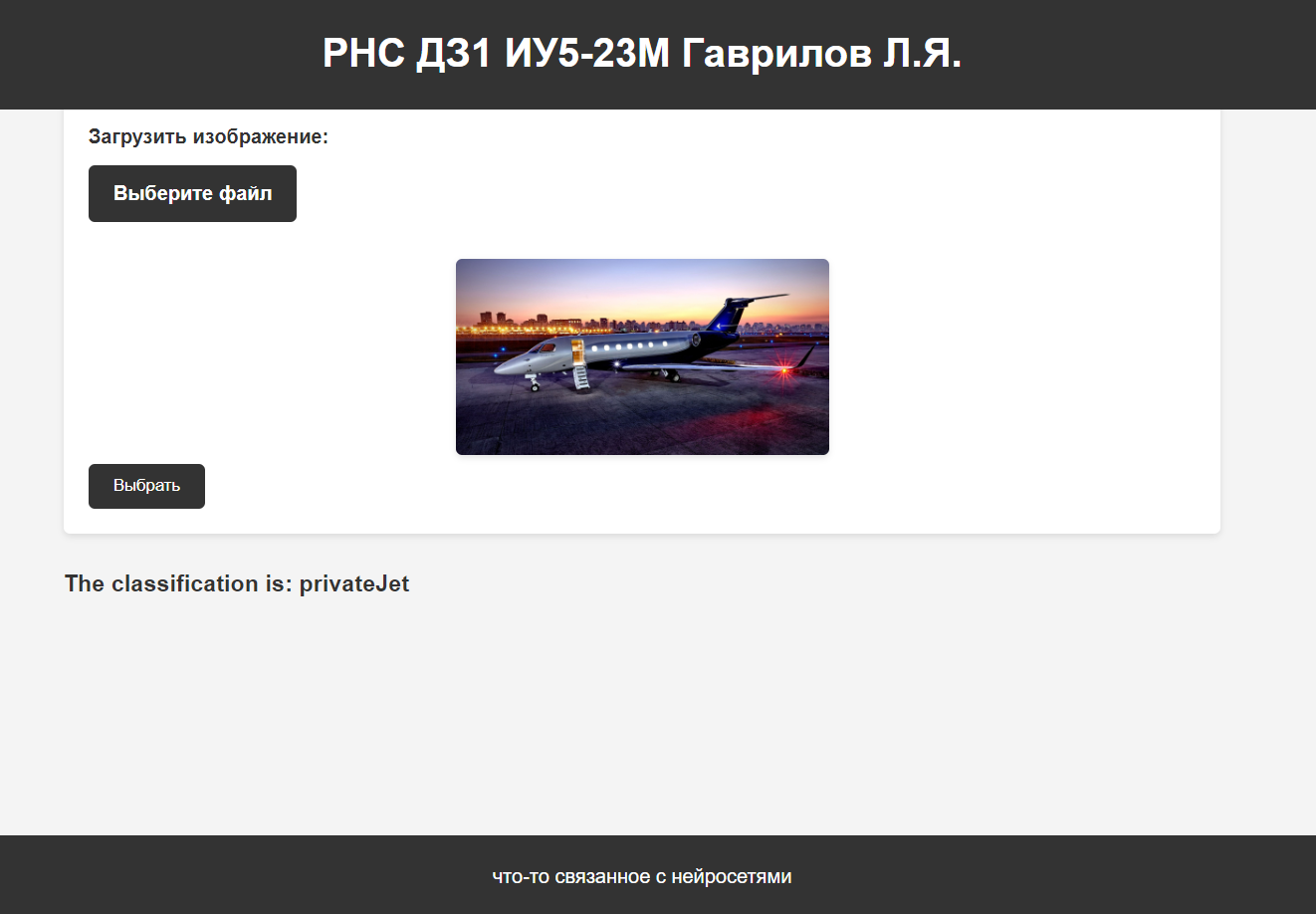


Рисунок 7 - Часть оформления. Загрузка рисунка.

**Вывод**: в рамках выполнения домашнего задания, были применены навыки обучения нейронных сетей на основе знаний, полученных с выполненных ранее лабораторных работ.

В результате домашней работы было получено веб-приложение, которым можно пользоваться для различия частного джета, v-22 и вертолета. Точность результирующей нейронной сети составляет 90%, что является хорошим результатом.

На основании представленной таблицы:

 Класс “privateJet” имеет точность 1.0000, это означает, что все прогнозируемые экземпляры “privateJet” действительно являются “privateJet”. Нет ложных срабатываний, то есть модель не ошибается, предсказывая этот класс.

С другой стороны, класс “v-22” имеет точность 0.9412, что означает, что большинство, но не все прогнозируемые экземпляры “v-22”, действительно являются “v-22”. Есть некоторое количество ложных срабатываний.

Наконец, класс “helicopter” имеет наименьшую точность 0.7917, что означает, что существует значительное количество ложных срабатываний при прогнозировании этого класса. Это может быть связано с тем, что характеристики, которые модель использует для идентификации “helicopter”, могут быть менее уникальными или более размытыми по сравнению с другими классами. Это может также указывать на то, что модели может потребоваться больше данных или более сложная архитектура для улучшения ее производительности на этом классе.

Анализируя метрики, можно сделать вывод о том, что нейронная сеть хорошо справляется с классификацией частного джета – precision и recall являются максимальными для данного класса. Это означает, что нейронная сеть редко ошибается (precision), а также что нейронная сеть хорошо распознает частного джета (recall). Нейронная сеть чаще ошибочно классифицирует вертолет и v-22, а также реже правильно классифицирует вертолет.

В ходе работы были получены навыки формирования приложений с использованием нейронных сетей, а по созданию нейронных сетей на различные тематики.

Был получен опыт сбора собственного датасета, его разметки, разбиение его на обучающую и тестовую выборку для дальнейшего обучения. Обучение проводилось с использованием разобранных ранее методов: аугментации, регуляризации, дообучения.