|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика, искусственный интеллект и системы управления

КАФЕДРА Системы обработки информации и управления

**ОТЧЕТ**

***ПО ДОМАШНЕМУ ЗАДАНИЮ***

***НА ТЕМУ:***

**Обучение нейронной сети на собранном датасете и создание Web-приложения**

Студент \_\_ИУ5-24М\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_С.С.Винников\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_А.И.Канев\_\_**

(Подпись, дата) (И.О.Фамилия)

2024 г.

СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc169725683)

[ВВЕДЕНИЕ 3](#_Toc169725684)

[**Задание** 3](#_Toc169725685)

[**Часть 1. Сбор датасета** 4](#_Toc169725686)

[**Часть 2. Обучение нейронной сети** 6](#_Toc169725687)

[**Часть 3. Web-приложение** 9](#_Toc169725688)

[ЗАКЛЮЧЕНИЕ 12](#_Toc169725689)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 13](#_Toc169725690)

ВВЕДЕНИЕ

ResNet (residual network, “остаточная сеть”) – это нейронная сеть, в архитектуре которой была представлена концепция “остаточного обучения”, что позволило решить проблему затухающего градиента. В традиционных глубоких нейронных сетях каждый слой применяет набор преобразований к входным данным для получения выходных данных. Остаточные соединения, реализованные в ResNet, позволяют сети изучать остаточные отображения, представляющие собой различия между входными и выходными данными слоя. Остаточные соединения формируются путем добавления входных данных к выходным данным слоя, что позволяет градиентам проходить непосредственно через сеть без своего затухания.

Веб-приложение — клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными службами.

# **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.

Подробный разбор представлен в [1] и [2].

# **Часть 1. Сбор датасета**

По заданию необходимо обучить нейронную сеть, способную распознавать различные виды насекомых: осы, пчелы, шершни.

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. Пчелы - 100 изображения
2. Шершни - 110 изображений
3. Осы - 105 изображений

Изображения перепончатокрылых были выбраны, с условием, что они находятся на фоне зелени. Примеры изображений указаны на рисунках ниже:



Рисунок 1 - Пример пчелы



Рисунок 2 - Пример шершня



Рисунок 3 - Пример осы

# **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** |
| Пчелы | 0.9877 | 1.0000 | 0.9938 |
| Шершни | 1.0000 | 1.0000 | 1.0000 |
| Осы | 1.0000 | 0.9875 | 0.9937 |
| Accuracy |  | | 0.9958 |
| **Test** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Пчелы | 0.6429 | 0.4500 | 0.5294 |
| Шершни | 0.7000 | 0.6667 | 0.6829 |
| Осы | 0.5185 | 0.7000 | 0.5957 |
| Accuracy |  | | 0.6066 |

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** |
| Пчелы | 1.0000 | 1.0000 | 1.0000 |
| Шершни | 1.0000 | 1.0000 | 1.0000 |
| Осы | 1.0000 | 1.0000 | 1.0000 |
| Accuracy |  | | 1.0000 |
| **Test** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Пчелы | 0.8889 | 0.8000 | 0.8421 |
| Шершни | 0.9000 | 0.8571 | 0.8780 |
| Осы | 0.7391 | 0.8500 | 0.7907 |
| Accuracy |  | | 0.8361 |

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

Как можно заметить, результат в 83 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 = 60,1%, train = 99,6% | Полносвязная нейронная сеть |
| Resnet  Keep\_last = 65 | lr = 0.0003,  batch\_size = 32,  epochs = 60  weight\_decay=1e-5  optimizator = SGD | test = 83,6%, train = 100% | Размороженная mobileNet |
| Resnet  Keep\_last = 65 | lr = 0.00015,  batch\_size = 32,  epochs = 120  weight\_decay=1e-5  optimizator = SGD | test = 67%, train = 94% | 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 = 90,4%, 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 = 91,6%, 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.

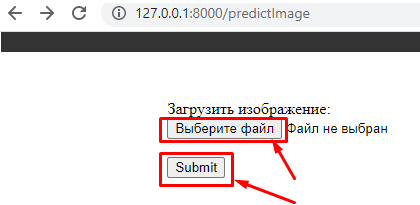


Рисунок 5 - Базовый шаблон html

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



Рисунок 6 - Часть оформления. Загрузка рисунка.

Загрузим изображения грибов, посмотрим на результаты. Выводы сети на рисунках ниже.

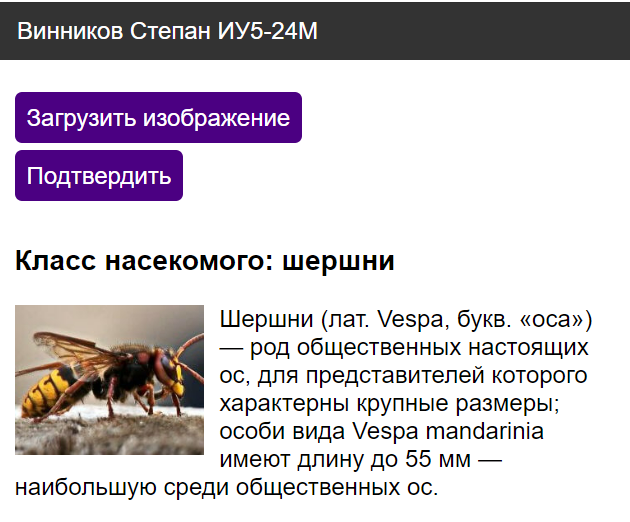


Рисунок 7. Вывод приложения при загрузке изображения с шершнем.



Рисунок 8. Вывод приложения при загрузке изображения с пчелой.

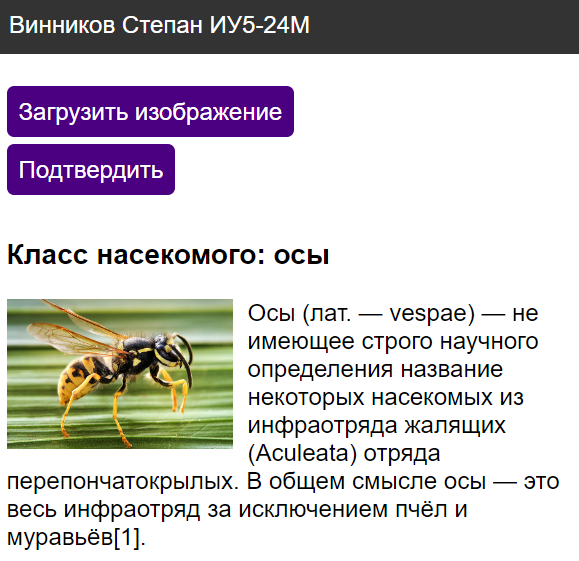


Рисунок 9. Вывод приложения при загрузке изображения с осой.

ЗАКЛЮЧЕНИЕ

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 трех видов перепончатокрылых – ос, шершней и пчёл. Точность результирующей нейронной сети составляет 91,6%, что является неплохим результатом.

Лучше всего нейронная сеть справляется с распознаванием шершней, хуже всего – с распознаванием ос. Это можно объяснить тем, что шершни имеют свои отличительные признаки – пропорции, цвета, структура головы. Пчелы так же имеют характерные признаки – пушистость, цвета. Осы сильно похожи на шершней, и нейронная сеть не смогла на 100% выделить какие-то характерные признаки ос.

Анализируя метрики, можно сделать вывод о том, что нейронная сеть хорошо справляется с классификацией шершней – precision и recall являются максимальными для данного класса. Это означает, что нейронная сеть редко путает шершней с другими видами насекомых (precision), а также что нейронная сеть хорошо распознает данный вид насекомых (recall). Нейронная сеть чаще ошибочно классифицирует пчел и шершней как ос, а также реже правильно классифицирует пчел.

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.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Методические указания к лабораторным работам. Источник: <https://github.com/iu5git/Deep-learning>
2. Материал видео-лекций по предмету “Разработка нейросетевых систем”. Источник: [Deep learning - YouTube](https://www.youtube.com/playlist?list=PLLELLTvDgUQ_d9eUj_3XVpAdGByuU37kT)