**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**УНИВЕРСИТЕТ ИНТЕРНЕТ-ПРОФЕССИЙ “НЕТОЛОГИЯ”**

**Программа обучения: “Машинное обучение: фундаментальные инструменты и практики”**

Дипломная работа на тему:

«Создание приложение для классификации грибов»

Выполнил:

студент группы AML-31

Дубинец Алексей Викторович

Г. Краснодар 2023

Оглавление

[**1.** **Введение** 3](#_Toc142030300)

[**2.** **Постановка задачи** 4](#_Toc142030301)

[**3.** **Определение названий грибов и их количества.** 5](#_Toc142030302)

[**4.** **Подготовка наборов изображений.** 5](#_Toc142030303)

[**5.** **Обработка полученных данных.** 6](#_Toc142030304)

[**6.** **Описание модели используемой нейронной сети.** 7](#_Toc142030305)

[**7.** **Обучение нейронных сетей и предотвращение переобучения** 9](#_Toc142030306)

[**8.** **Переобучение** 9](#_Toc142030307)

[**9.** **Оценка точности результатов** 11](#_Toc142030308)

[**10.** **Реализация приложения.** 12](#_Toc142030309)

[**11.** **Заключение** 13](#_Toc142030310)

[**ЛИТЕРАТУРА** 14](#_Toc142030311)

# **Введение**

Цель дипломной работы - разработка удобного и легкодоступного масштабируемого приложения, решающего задачу распознавания образов и классификации видов грибов. На основе данной классификации получение дополнительной расширенной информации об объекте классификации.

В данной работе была проведена декомпозиция поставленной задачи на более мелкие подзадачи, для которых были выбраны методы и технологии решения.

В ходе работы были задействованы инструменты веб-парсинга, преобразования и отчистки данных, а также высокоуровневые API для машинного обучения в области компьютерного зрения и высокоуровневые API для формирования модуля взаимодействия пользователя с моделью, через Telegram бота.

Система использует собранные данные видов грибов, представленных на сайте [https://wikigrib.ru/](%20https:/wikigrib.ru/%20) и проводит классификацию с ссылкой на страницу сайта содержащую подробную информацию. Данная модель может быть использовано как в исходном виде, так и в переработанном виде для реализации полноценного IT продукта и выхода с этим продуктом на рынок.

# **Постановка задачи**

Наиболее простым способом передать информацию о грибах для пользователя является загрузить ее фотографию в уже установленный и популярный мессенджер или соц.сеть. То есть для классификации грибов необходимо использовать модель для распознавания визуальных образов, дополнительно интегрированную в популярную среду.

Общая задача была декомпозирована:

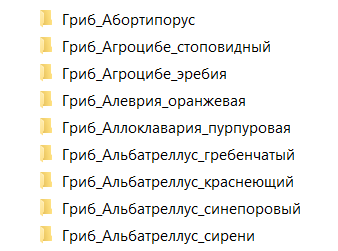
1. Необходимо получить перечень грибов, для которых будем производить классификацию. Для этого необходимо произвести веб-парсинг сайта содержащего информацию о грибах.
2. По названию грибов нужно было произвести веб-парсинг фотографий грибов. Для увеличения кол-ва фотографий, производить парсинг на русском и на английском.
3. Провести обработку полученного датасета на наличие дублей и не корректной информации. Преобразовать все фотографии к одному формату, разрешению, названию.
4. Настроить аугментацию для увеличения кол-ва обучающей выборки.
5. Загрузить предобученную модель ResNet50 и произвести дообучение на нашем dataset’e.
6. Создать Telegram-bot’а в который сможет принимать фотографии от пользователя и передавать их нашей модели для классификации. Далее выдавать ответ пользователю.

# **Определение названий грибов их количества.**

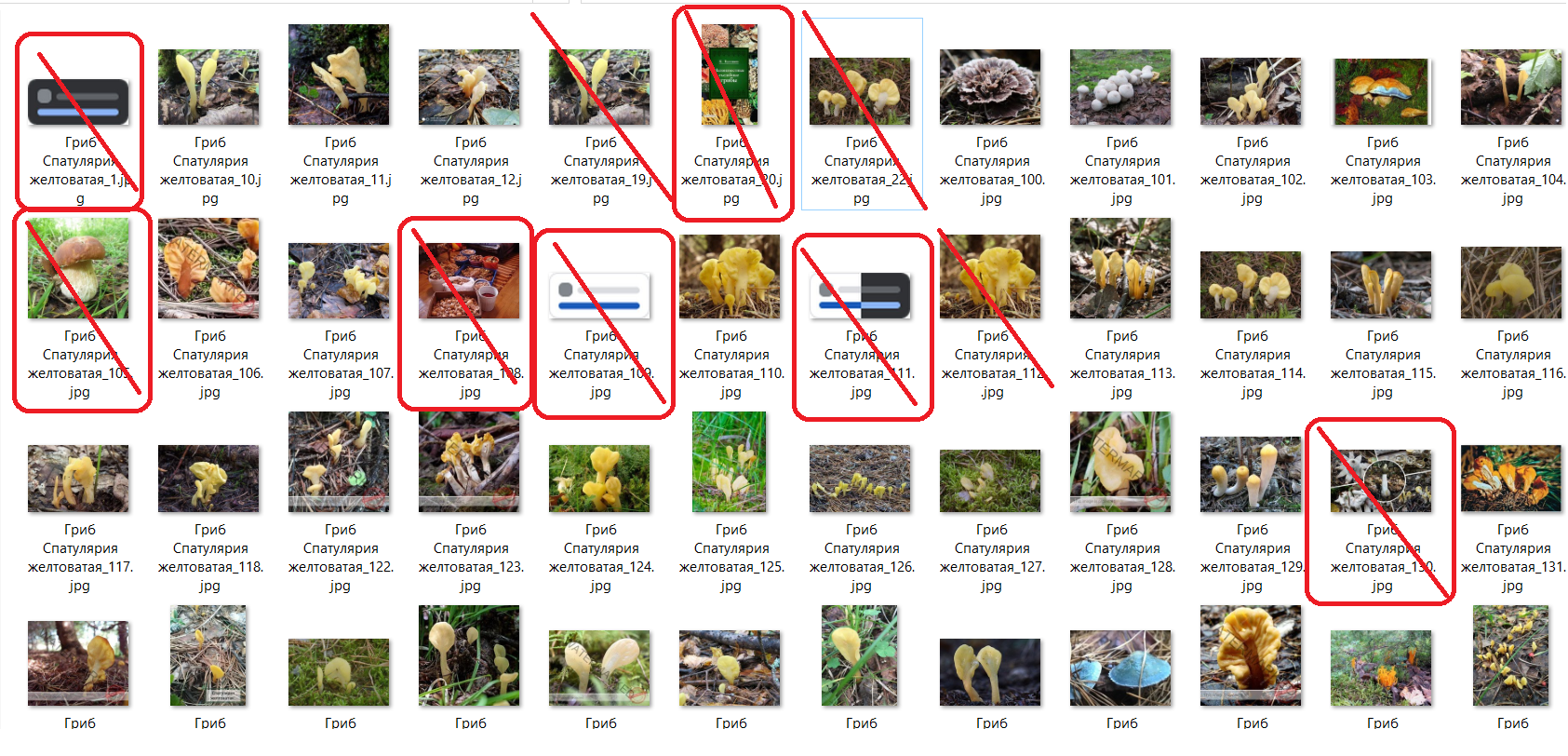
В рамках данной работы мне необходимо было определить какие грибы мы будем классифицировать. Для этого был выбран сайт <https://wikigrib.ru/> . Данный сайт имеет удобный интерфейс, на нем представлено порядка 1045 грибов на момент парсинга. Названия грибов были представлены, как на русском так и на латинском, что позволяет произвести 2 цикла парсинга фотографий. Также плюсом являлось то, что на сейте отсутствовали блокировки при частых запросах. При парсинге был сформирован dataframe который будет использоваться при дальнейшей классификации и описании грибов. Код данного блока описан в файле “Parsing\_data.ipynb”.

# **Подготовка наборов изображений.**

Далее по полученному списку названий был произведен парсинг фотографий на русском. По названию гриба производился поисковый запрос в google, ответ в виде фотографий сохранялся в папку с названием, соответствующим поисковому запросу.



Итого было получено 1045 папок содержащих фотографии грибов. В первом цикле поиска было получено 250\*1045 =261 250 фотографий. Второй цикл происходил по той же схеме, но уже на латинском. Пример реализованного кода для парсинга фотографий с помощью поисковых запросов в google находится в файле “Parsing\_data.ipynb”.

Пример полученных фотографий 

Как мы видим, данная выборка имеет большое количество дублей и информации, не относящейся к теме поискового запроса. Это требует дополнительной обработки.

# **Обработка полученных данных.**

Для подготовки полученных фотографий в качестве dataset нужно было провести дополнительную работу. Все папки на английском переименовать на русский. Фотографии из папок также переименовать на русский и совместить папки.

Полученный набор фото мы проверили на наличие дублей с помощью hash -алгоритмов библиотеки hashlib. Это позволило быстро найти и удалить дубли с совпадающим hash.

Оставшиеся фотографии мы конвертировали в один формат jpg, а также привели к одному размеру 240x240. Это необходимо чтобы упростить дальнейшую работу с ними. Оставшиеся дубли мы удалили методом по пиксельного сравнивание используя библиотеку cv2. После данной работы осталось около 130 000 фотографий. В среднем 124 фото на 1 класс. Далее мы провели работу по удалению лишней информации.

# **Описание модели используемой нейронной сети.**

В 2014 г. глубокие нейронные сети превысили уровень точности распознавания изображений в 95% – уровень распознавания человеком. Изображения были взяты из набора данных ImageNet – набора данных, составляющего более 15 миллионов изображений, принадлежащих 22-м тысячам классов [1]. Однако подобные нейронные сети достаточно большие (от нескольких десятков до нескольких сотен слоёв), и для их обучения требуются большие вычислительные ресурсы. При этом все картинки состоят из примерно одинакового набора элементарных объектов: точки, наклонные линии, контрастности и т.д. Поэтому можно не обучать каждый раз нейронную сеть с нуля, а только обучить до конца уже существующую.

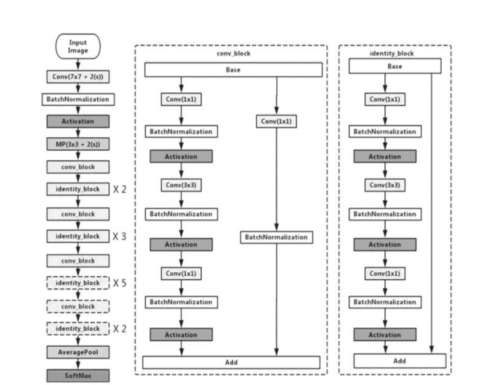
Данный подход называется “перенос обучения”. Он предлагает взять уже обученную на данных ImageNet модель нейронной сети, а затем дообучить ее полносвязные слои и откорректировать количество классов на выходе. Такая технология позволяет существенно сократить количество времени на обучение, а также, благодаря уже настроенным весам в сверточной части модели, получить в большинстве случаев большую, чем на необученной модели нейронной сети, точность распознавания изображений на Вашем наборе данных.

ResNet (от “Residual Network” – “остаточная сеть”) – серия моделей нейронных сетей разной глубины, предоставляющая возможность применить “остаточное обучение” [2]. Было замечено, что если просто добавлять в архитектуру модели больше слоев, то качество такой модели растет до некоторого предела, а затем начинает падать. Причиной возникновения такого эффекта является исчезающий градиент. Люди часто сталкиваются с этой проблемой при обучении искусственных нейронных сетей [2], включающих обратное распространение и обучение на основе градиентов. Эксперты используют градиенты для обновления весов в конкретной сети [3]. Однако иногда всё по-другому, т.к. градиент становится невероятно маленьким и почти исчезает. Это не позволяет весам изменять свои ценности, что приводит к тому, что сеть прекращает обучение, т.к. одни и те же ценности будут распространяться снова и снова без какой-либо значимой работы [4].

ResNet позволяет решить данную проблему с помощью “остаточного обучения”. После прохода каждой пары слоев свертки веса, которыми обладала модель до прохода через пару, т.е. “остатки”, передаются туда же, где находятся веса, которые модель вычислила уже после прохода. Затем она решает, какой из результатов приносит лучший результат, и использует его в дальнейшем обучении. Данное добавление промежуточных связей облегчает обучение и позволяет избежать затухания градиента.

Остаточные нейронные сети являются идеальным решением проблемы исчезающего градиента. По мере того, как мы продолжаем обучение, модель понимает концепцию сохранения полезных слоев, а не использования тех, которые не помогают.

В данной работе для распознавания изображений использовалась модель ResNet50. Архитектуры данных нейронных сетей представлены на рис.



# **Обучение нейронных сетей и предотвращение переобучения**

Для выбранной модели нейронных сетей проводился анализ наборов настраиваемых параметров, таких как:

– количество нейронов в полносвязном слое;

– оптимизатор – метод, позволяющий ускорить процесс обучения и оптимизировать функцию потерь;

– скорость обучения - параметр оптимизатора, позволяющий оценить, насколько сильно необходимо изменять веса связей между нейронами для уменьшения функции потерь;

В целях поиска, наиболее подходящих модели и набора параметров в процессе работы был проведен сравнительный анализ, результаты которого приведены в табл.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Модель | Количество нейронов | Оптимизатор | Скорость обучения | Максимум на тестовой выборке |
|
| ResNet50 | 4168 | SGD | 0.0001 | 22% |
| ResNet50 | 2084 | SGD | 0.0001 | 24% |
| ResNet50 | 2084 | Adam | 0.0003 | 24% |
| ResNet50 | 2084 | Adam | 0.0001 | 25% |
| ResNet50 | 2084 | Adam | 0.001-1.0000e-05 | 27% |

Лучшей стала модель ResNet50 с 2080 нейронами в полносвязном слое и оптимизатором Adam, скорость обучения для которого равна менялась от 0.001 до 1.0000e-05 в ходе обучения с помощью заданной callbacks функции. Максимальная точность распознавания при таких параметрах составила 27%.

# **Переобучение**

Одной из важных характеристик алгоритмов машинного обучения является обобщающая способность. Обобщающая способность – это свойство модели отражать исходные данные в требуемые результаты на всем множестве исходных данных (во всех сценариях, а не только на тренировочных примерах). Величину обобщения оценивают через обратную величину – отклонение или ошибку. Ошибка – это численно выраженная разница между ответом модели и требуемым (реальным) значением. В более общем смысле обобщающая способность – способность модели найти некое соответствие, которое будет описывать неизвестную нам и скрытую взаимосвязь входных и выходных данных [5]. Переобучение – одна из проблем глубоких нейронных сетей. Суть данной проблемы состоит в следующем: модель хорошо объясняет только примеры из обучающей выборки, адаптируясь к обучающим примерам, вместо того, чтобы учиться классифицировать примеры, не участвовавшие в обучении (теряя способность к обобщению). Нейросети регулярно сталкиваются с этой проблемой, и в последнее время были разработаны действенные методы уменьшения уровня переобучения. В данной работе использовались два основных способа уменьшить переобучение:

– dropout;

– аугментация данных;

Dropout – один из методов борьбы с переобучением модели. Суть dropout состоит в "забывании" части информации. т.е. некоторый, заданный заранее процент нейронных связей, разрывается (забывается) на выходе из текущего слоя нейронной сети [6]. Таким образом вместо того, чтобы идеально подобрать веса только лишь для обучающего набора данных, НС учится подбирать ответ для похожих данных, которые не встречались в обучающем наборе. Это может значительно улучшить точность сети для данных, которые не встречались в обучающем наборе.

Наилучшие результаты модель, показала при значении вероятности разрыва нейронных связей, равной 0.3. В среднем, переобучение сократилось на 2–4% благодаря применению dropout с оптимальным значением.

Аугментация данных. По данным Google Brain, расширение данных с помощью методов аугментации увеличивает точность модели [7]. Под аугментацией данных понимается увеличение выборки данных для обучения через модификацию существующих данных. Использование методов аугментации данных показало себя хорошо на задаче классификации изображений. Благодаря такому эффекту модель учится обобщать признаки одного класса, не сосредоточиваясь слишком сильно на отдельных изображениях, что в дальнейшем увеличивает точность распознавания на проверочном и тестовом наборах данных.

Для аугментации мы будем использовать библиотеку Keras, класс

ImageDataGenerator. Этот класс позволяет осуществить сразу несколько операции:

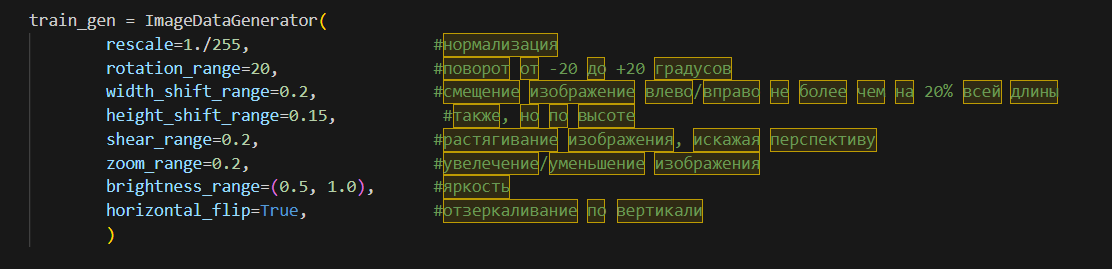
1) Изменить размер всех изображений до 240x240 пикселей;

2) Разделить данные по батчам, чтобы пропускать их поочередно, а не

все разом;

3) Наложить на данные аугментации.

Функция позволяет произвести сразу несколько видов преобразований. Описание преобразований представлены на рис.



После применения аугментации мы подаем на вход нейронной сети уже преобразованное изображение. Пример такого преобразования.



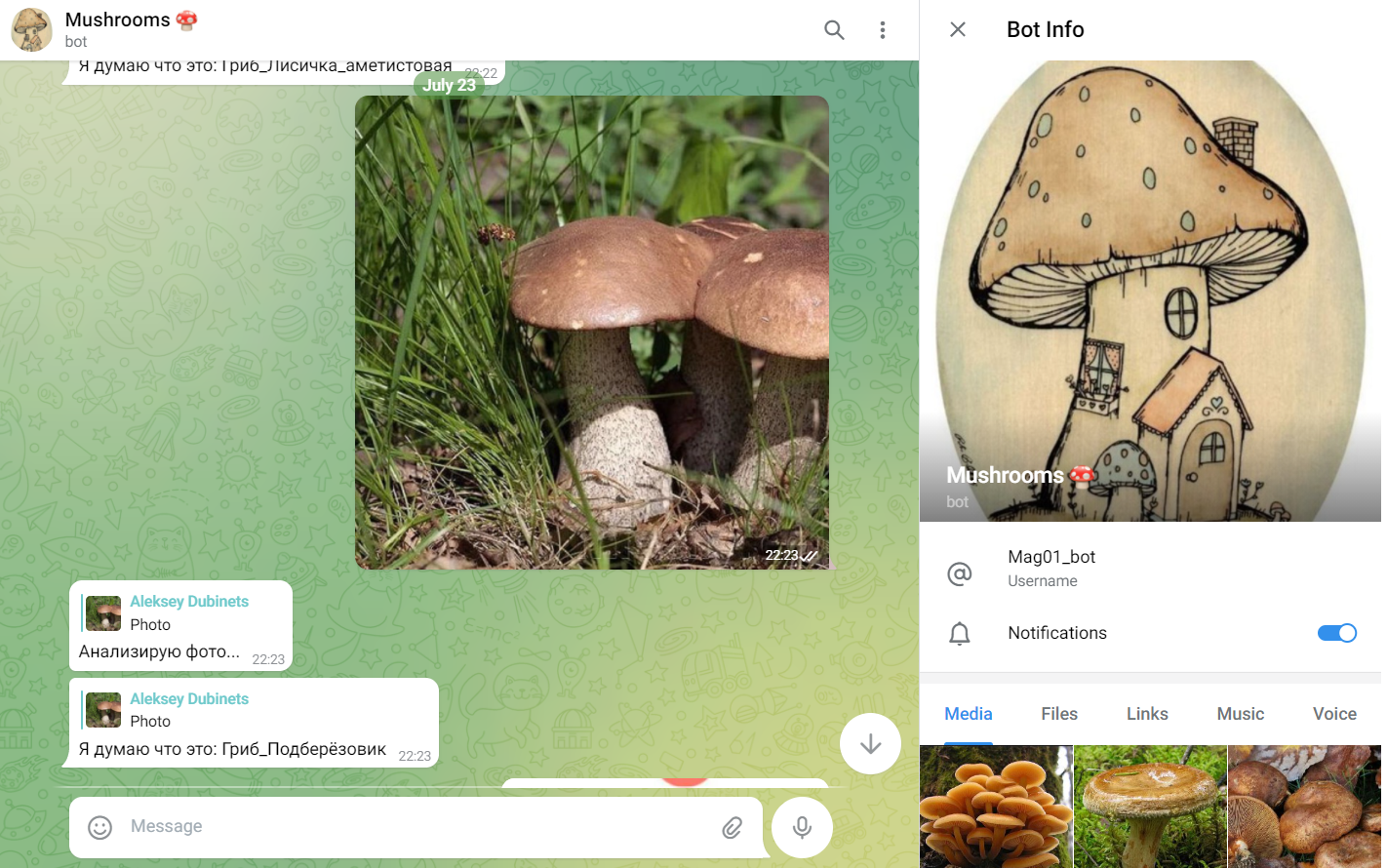
# **Оценка точности результатов**

Максимальная точность на тестовых данных по итогам работы алгоритма – 27%. Этого недостаточно для того, чтобы однозначно определить, что это за гриб, поэтому я принял решение выводить первые 10 классов с наибольшей вероятностью. Это позволит увеличить точность определения.

# **Реализация приложения.**

Представленная в работе система имеет потенциал к развитию, но и на текущий момент она может называть продуктом, который соответствует концепции MVP (Minimal Viable Product, Минимальной жизнеспособный продукт). «MVP — тестовая версия товара, услуги или сервиса с минимальным набором функций (иногда даже одной), которая несет ценность для конечного потребителя.» [12] Возможность только отправить изображение и получить класс на нём (вид гриба), и ссылку на подробное его описание, — это минимальная функциональность, которую можно назвать продуктом, в случае представления доступа сторонним пользователям. KPI такого продукта – это число проверяемых пользователями изображений, которое будет говорить о востребованности сервиса.

Мною было выбрана реализация подключение системы к сторонним площадкам через API в виде бота в мессенджере Telegram. Данная интеграция позволит использовать приложение в повседневной жизни и не требует установки дополнительного приложения (@Mag01\_bot). Данное приложение работает на моем стационарном компьютере и является демонстрационным. Для перехода на “продуктовый” режим работы, оно требует переноса в облако. Это можно сделать с помощью таких сервисов, как Heroku, например, но требует платной подписки.



# **Заключение**

Таким образом, в данной работе представлен алгоритм, благодаря которому появляется возможность классифицировать грибы по изображениям. Используемый подход позволяет существенно сузить список вероятных грибов, тем самым сократить время на поиск их в справочниках для начинающих специалистов. Удобный интерфейс делает использование модели мобильным. Пользователю не нужно иметь установленное специальное программное обеспечение и оборудование для того, чтобы произвести распознавание грибов, достаточно иметь телефон с установленным мессенджером. Также данный подход может быть использован как инструкция для реализации своего проекта по распознаванию изображений.

Данный продукт имеет потенциал для улучшения. Например, можно выделить 100 обобщенных классов которые позволят определить, например семейство гриба, а потом производить уже поиск внутри семейства, это должно увеличить точность. Дополнительная фильтрация и обогащение фотографиями также должно увеличить точность модели.

# **ЛИТЕРАТУРА**

1. ImageNet – 2021. URL – https://www.image-net.org/update-mar-11-2021.php 4. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition // arXiv. – 2015. – URL – https://arxiv.org/abs/1512.03385.

2. Проблема исчезающего градиента // Создание децентрализованных приложений на блокчейн – 2020. URL – https://clck.ru/psurd

3. Градиентный спуск: всё, что нужно знать // Neurohive – 2020. URL – https://neurohive.io/ru/osnovy-data-science/gradient-descent/

4. Грас Д. Data Science: наука о данных с нуля. – 2-е изд. Перераб. и доп. – 2020.

5. Massa E., Jonker M., Roes K., Coolen A. Correction of overfitting bias in regression models // arXiv – 2022. – URL – https://arxiv.org/abs/2204.05827

6. Hasan M., Khosravi A., Hossain I. Controlled Dropout for Uncertainty Estimation // arXiv – 2022. – URL – https://arxiv.org/abs/2205.03109

7. Xu M., Yoon S., Fuentes A., Sun Park D. A Comprehensive Survey of Image Augmentation Techniques for Deep Learning // arXiv – 2022. – URL – https://arxiv.org/abs/2205.01491

8. Pearce T., Brintrup A., Zhu J. Understanding Softmax Confidence and Uncertainty // arXiv – 2021. – https://arxiv.org/abs/2106.04972

9. Williams C. The Effect of Class Imbalance on Precision-Recall Curves // arXiv. – 2021. – V.2. – URL – https://arxiv.org/abs/2007.01905

10. Лабинцев Е. Метрики в задачах машинного обучения // Хабр. – 2017. – URL – <https://habr.com/ru/company/ods/blog/328372/>

11. Standout Examples of Minimum Viable Products [Электронный

ресурс]. - <https://myva360.com/blog/examples-of-minimum-viable-products>

12. MVP: что это такое и как работает? [Электронный ресурс]. -

https://habr.com/ru/company/productstar/blog/508892/

13. A Minimum Viable Product Is Not a Product, It's a

Processhttps://www.ycombinator.com/library/4Q-a-minimum-viable-product-is-

not-a-product-it-s-a-process