

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

**«Дальневосточный федеральный университет»**

(ДВФУ)

|  |
| --- |
| **ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)**    **Департамент математического и компьютерного моделирования** |

**ОТЧЕТ**

о прохождении производственной практики.

Научно-исследовательская практика.

направление подготовки 01.04.02 «Прикладная математика и информатика»

магистерская программа «Математические и компьютерные технологии»

|  |  |
| --- | --- |
| Оценка     отлично            Регистрационный номер \_\_\_\_\_\_\_  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025г. | Выполнил студент группы    М9123-01.04.02мкт  Панченко Никита Кириллович  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_          (подпись)  21 мая  «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025г.    Руководитель  ст. преподаватель  Малыкина И.А  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_     \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025г. |

|  |  |
| --- | --- |
|  | Практика пройдена в срок  с   «19» марта   2025 г.  по «22» мая  2025 г.  (3 з.е., рассредоточенная) |

г. Владивосток

2025

Оглавление

[Аннотация 3](#_Toc198601424)

[Введение 4](#_Toc198601425)

[1. Термины, определения и сокращения 4](#_Toc198601426)

[2. Описание предметной области 5](#_Toc198601427)

[3. Неформальная постановка задачи 5](#_Toc198601428)

[4. Обзор существующих методов решения 6](#_Toc198601429)

[Требования к окружению 8](#_Toc198601430)

[1. Требования к аппаратному обеспечению 8](#_Toc198601431)

[2. Требования к программному обеспечению 8](#_Toc198601432)

[3. Требования к пользователям 8](#_Toc198601433)

[4. Функциональные требования 9](#_Toc198601434)

[5. Требования к интерфейсу 10](#_Toc198601435)

[Проект 12](#_Toc198601436)

[1. Средства реализации 12](#_Toc198601437)

[2. Структуры данных 13](#_Toc198601438)

[3. Проект интерфейса 14](#_Toc198601439)

[Реализация и тестирования 18](#_Toc198601440)

[Заключение 21](#_Toc198601441)

[Дневник студента 22](#_Toc198601442)

# Аннотация

Тема дипломной работы: "Персональный каталогизатор-систематизатор фотографий анималистического жанра". Данная работа является продолжением ранее разработанного проекта, направленного на организацию архива фотографий животных, и фокусируется на внедрении новых функций для автоматизации обработки данных и улучшения пользовательского опыта.

Цель работы:

* Улучшение пользовательского интерфейса для более интуитивного взаимодействия с системой.
* Автоматизация обработки и систематизации медиафайлов (фото, видео, аудио).
* Внедрение новых инструментов для работы с картографическими данными.
* Разработка встроенного редактора изображений.
* Создание и интеграция нейросетевого модуля для автоматического определения видов птиц.
* Методы исследования: Анализ современных методов машинного обучения,

Результатом работы является улучшенное приложение для систематизации медиаконтента фотографов анималистов.

* Повышена эффективность работы пользователей за счет автоматизации рутинных задач.
* Улучшена точность и скорость обработки медиафайлов.
* Расширены аналитические возможности приложения благодаря новым инструментам работы с картами и нейросети.

# Введение

## Термины, определения и сокращения

Перечень обозначений и/или сокращений составляют в алфавитном порядке:

* Python – высокоуровневый язык программирования, используемый для разработки backend-части приложения.
* SQLite – встраиваемая реляционная база данных, применяемая для хранения и управления данными в проекте.
* Eel – библиотека Python для создания простых веб-интерфейсов с использованием HTML/JS.
* EXIF (Exchangeable Image File Format) – стандарт хранения метаданных в цифровых изображениях.
* API (Application Programming Interface) – интерфейс программирования приложений, использованный для интеграции с Яндекс.Картами.
* Яндекс.Карты – картографический сервис, предоставляющий API для работы с геоданными.
* Нейронная сеть – математическая модель, используемая для распознавания видов птиц на изображениях.
* Компьютерное зрение – технология, позволяющая компьютеру анализировать и интерпретировать визуальную информацию.
* Геолокация – определение географического положения объекта (в данном случае – места съёмки).
* GUI (Graphical User Interface) – графический пользовательский интерфейс приложения.
* Метаданные – структурированные данные, описывающие характеристики медиафайлов (дата, место съёмки, параметры камеры и др.).
* Бёрдвотчинг (birdwatching) – наблюдение за птицами в их естественной среде обитания.
* Кластеризация – группировка данных (например, точек на карте) по заданным признакам.

## Описание предметной области

Данная работа посвящена продолжением моей работы по созданию персонального каталогизатора-систематизатора для фотографий анималистического жанра с целью упрощения организации и поиска изображений в данной тематике. В ходе выполнения проекта были успешно решены следующие задачи: анализ существующих систем, разработка интуитивного интерфейса, создание базы данных, определение алгоритма систематизации и проведение оптимизации. Полученным результатом является персональный каталогизатор с интуитивным интерфейсом и возможностью эффективной систематизации фотографий, предназначенный как для профессиональных фотографов, так и для любителей анималистической живописи.

В процессе разработки будет внедрен функционал, упрощающий работу с фотографиями, видеозаписями и аудиозаписями. Информация о локации и дате снимка будет автоматически извлекаться из соответствующих файлов, что дополнительно улучшит пользовательский опыт. Кроме того, в рамках проекта планируется разработка нейронной сети, способной определять вид птицы по фотографии, расширяя тем самым функционал системы и обогащая ее возможности. Возможность редактировать фотографии.

## Неформальная постановка задачи

Данная работа представляет собой доработанную версию персонального каталогизатора фотографий животных с расширенным функционалом автоматизации и обработки медиа данных. В отличие от предыдущей версии, текущий проект делает акцент на следующих аспектах:

Что нужно будет сделать в новой версии:

* Автоматическое заполнение данных вместо ручного ввода
* Встроенный редактор с инструментами для первичного редактирования
* Нейросеть для определения вида птиц
* Обновленный и доработанный дизайн
* Доработанная фильтрация для поиска фотографий с помощью Яндекс.карт.

## Обзор существующих методов решения

Для разработки персонального каталогизатора необходимо проанализировать и оценить существующие решения, применяемые в данной области.

Существует множество приложений для управления фотографиями, но не все они могут быть применены для работы с фотографиями животных. Например, Adobe Lightroom прекрасно подходит для обработки фотографий в целом, но не имеет специализированных функций для работы с фотографиями, таких как классификация по виду животного, соответствующий географический регион и т.д.

С другой стороны, существуют специализированные сообщества, такие как "Bird Journal", "eBird", "Птицы Дальнего Востока", "Птицы Дальнего Востока" и т.д., которые позволяют вести наблюдения за птицами в природе и использовать специализированные классификации птиц. Однако эти приложения ориентированы на определенных пользователей и не имеют таких широких возможностей классификации и фильтрации фотографий как это требуется в данной работе. К тому же данные хранятся не у пользователя и добавляются в общий доступ.

# Требования к окружению

## Требования к аппаратному обеспечению

Для стабильной работы приложения требуется:

* Процессор: использование современных многоядерных процессоров с высокой тактовой частотой, таких как Intel Core i3, AMD Ryzen 3 или лучше.
* Оперативная память: иметь не менее 1 ГБ оперативной памяти для обеспечения плавной работы и эффективного использования приложения.
* Монитор: рекомендуется использовать монитор разрешением не менее 1280 x 720 пикселей или выше и хорошей цветовой гаммой для точного отображения изображений.
* Хранение данных: HDD или SSD с объемом данных не менее 500Мб.
* Устройства ввода: мышь и клавиатура для взаимодействия с интерфейсом.

## Требования к программному обеспечению

* Веб-браузер: для успешного запуска и работы приложения необходимо иметь установленный веб-браузер Google Chrome. Приложение может использовать функциональные возможности и API, доступные только в браузере Google Chrome.
* Операционная система: для установки и работы приложения требуется операционная система Windows 10 или более поздняя версия.

## Требования к пользователям

От пользователя требуется:

* Наличие устройства с операционной системой Windows.
* Знание основ работы с компьютером, включая работу с файловой системой и умение устанавливать программное обеспечение.
* Умение сохранять фотографии в цифровом формате и записывать звуки и видео на устройствах хранения данных.
* Желание и потребность в использовании приложения для создания и хранения своих собственных каталогов фотографий животных.
* Возможность добавлять и удалять фотографии, аудио и видео файлы, а также редактировать информацию о них.
* Умение проводить поиск по базе данных, используя различные фильтры и параметры поиска.
* Умение работать с интерфейсом приложения, включая элементы управления и различные функции.
* Желание сохранять и регулярно обновлять свои данные, используя приложение, чтобы поддерживать свои каталоги фотографий животных в актуальном состоянии.

## Функциональные требования

Приложение должно обеспечивать выполнение следующих функций:

1. Формирование в приложении классификатора, с возможностью его редактирования согласно потребностям пользователя. То есть, предоставить возможность самому пользователю создавать любую иерархическую структуру, используемую для классификации живых организмов.
2. Организация метаданных для всех имеющихся у пользователя фотографий в рамках сформированного каталога классификатора, Географическая интеграция:
3. Привязка к фотографиям дополнительных метаданных (заметки и описание снимка или его содержимого, геолокация съёмки или получения биологического экземпляра, категория качества фотографии, и др.)
4. Максимально полное автоматическое извлечение информации из EXIF файлов,
5. Работа с галереей с выборкой и сортировкой по критериям: период съемки, место съёмки, уровень организации в заданной структуре (тип, класс, отряд, семейство и т.д.), качество фотографий.
6. Работа с аудио и видео файлами с теми же функциональными возможностями, что и для фотографий,
7. Организация работы с картой, и с возможностью наглядной демонстрации географии получения снимков или образцов, изображённых на снимках,
8. Возможность максимально удобного и быстрого поиска фотографии по различным критериям и открытие папки физического расположения файла, включая облачные хранилища,
9. Создание и интеграция нейросетевого модуля для автоматического распознавания объекта, изображенного на фотографии на примере птиц.

## Требования к интерфейсу

При разработке новой версии приложения особое внимание было уделено удобству навигации, визуальной ясности и поддержке новых функций, таких как нейросетевое распознавание, встроенный редактор и карта наблюдений.

Общие требования интерфейса:

* Интуитивность — интерфейс должен быть понятным даже для пользователя без технического образования;
* Минимализм — дизайн не должен перегружать экран лишними элементами, акцент должен быть на содержимом (фотографии, карты, данные);
* Адаптивность — интерфейс должен корректно отображаться на различных разрешениях экранов и масштабах (от ноутбуков до широкоформатных мониторов);
* Отзывчивость — действия пользователя (нажатия кнопок, загрузка, фильтрация и т.п.) должны сопровождаться визуальными подтверждениями и происходить без длительных задержек;
* Унификация стиля — все элементы интерфейса (кнопки, поля, шрифты, цвета) должны быть выдержаны в едином стиле.

# Проект

## Средства реализации

Основным языком программирования остался Python благодаря своей универсальности, простоте синтаксиса и широкому набору библиотек, подходящих для задач обработки изображений, работы с нейросетями, базами данных и построения графических интерфейсов.

Для автоматизации ввода информации о фотографиях (дата, модель камеры, координаты и др.) используется модуль exifread, который позволяет извлекать метаданные из изображений. Для преобразования координат в текстовые адреса и обратно применяется библиотека geopy. Это упрощает добавление новых фотографий в систему и снижает риск ошибок, связанных с ручным вводом.

Для автоматического определения видов птиц по фотографии используется обученная модель компьютерного зрения, построенная на базе библиотеки PyTorch. Архитектура модели — efficientnet\_b0, с дообучением на специализированном датасете изображений птиц. Обработка изображений происходит локально, что обеспечивает сохранность персональных данных пользователя.

По работе с базой данных осталась библиотека SQLite однако структура таблиц была доработана для поддержки новых данных.

Графическая оболочка приложения построена с использованием библиотекиEel, обеспечивающей связку между Python и HTML/CSS/JavaScript. Интерфейс был существенно переработан: обновлены стили с использованием CSS, улучшена интерактивность с помощью JavaScript. Это позволило реализовать динамические элементы, быстрые реакции интерфейса и современные визуальные решения.

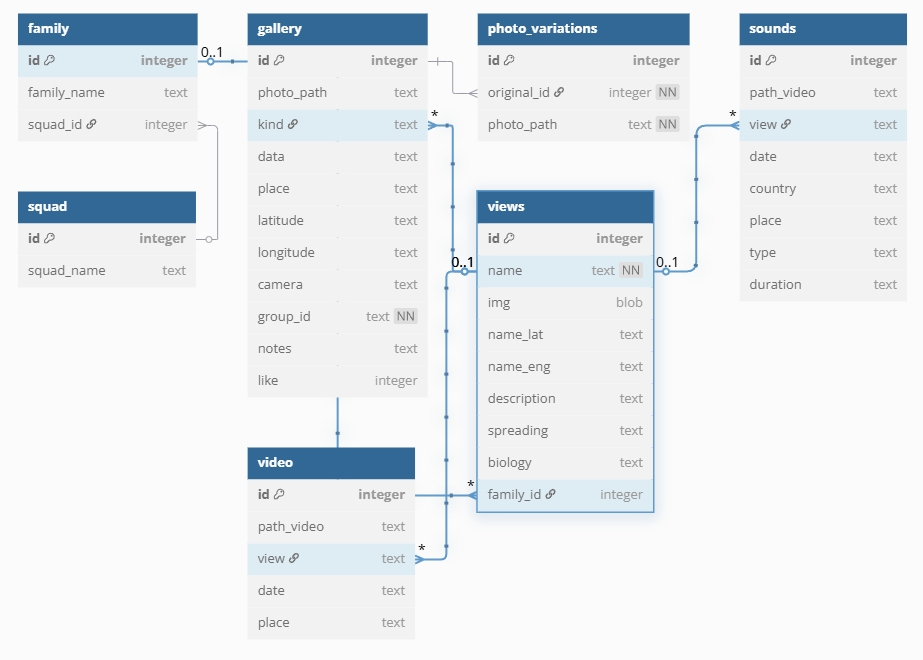
Чтобы можно было отображать местоположения фотографий используется API Яндекс.Карт. Применяются интерактивные balloons — всплывающие окна с изображением и данными, которые появляются при клике на метку. Это позволяет пользователю быстро находить нужные фотографии на карте и просматривать информацию без перехода в другую вкладку.

Редактирование фотографий реализовано средствами JavaScript с использованием элемента canvas, что позволяет пользователю производить обрезку, поворот, регулировку яркости и контрастности, а также применять фильтры. Результаты редактирования сохраняются как отдельные вариации исходной фотографии.

Чтобы запустить приложение нужно его упаковать и создать файл запуска с расширением exe для это использовалась библиотека PyInstaller.

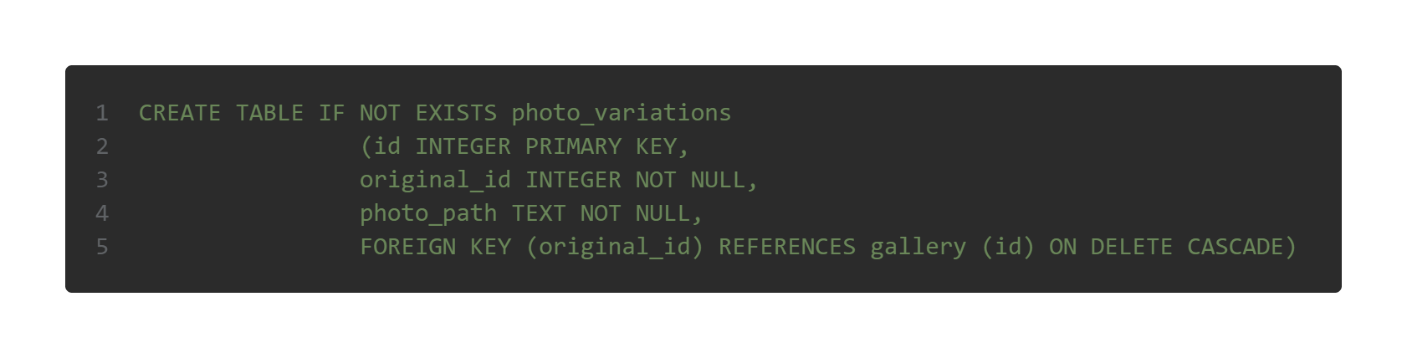
## 2. Структуры данных

Были расширены и переработаны структуры данных для поддержки новых функций. Вся информация по-прежнему хранится в локальной базе данных SQLite, что обеспечивает автономность и простоту развёртывания приложения. Обновленная схема базы данных (рисунок 1):



Рисунок

Новая таблица «photo\_variations» (рисунок 2) для хранения разных версий одной и той же фотографии – например, оригинал и отредактированные варианты.



Рисунок

Описание структуры таблицы:

Каждая запись ссылается на исходную фотографию из таблицы «gallery» через внешний ключ «original\_id». Поле «photo\_path» хранит путь к файлу с конкретной вариацией изображения.

## 3. Проект интерфейса

Цель редизайна — повысить удобство взаимодействия, адаптировать интерфейс под новые функции, а также сделать внешний вид более современным и интуитивным.

Дизайн выполнен в нейтральных, не отвлекающих тонах с акцентами на управляющих элементах. Применяется адаптивная верстка для корректного отображения на экранах с разным разрешением. Интерфейс использует нейтральные тона (таблица 1) цветов. Шрифт Roboto/Sans. Для каждой кнопки добавлены иконки для большой интуитивности.

|  |  |  |  |
| --- | --- | --- | --- |
| #3333 | #4B4B4B | #DBDBDB | #f0f0f0 |
|  |  |  |  |

Таблица

Каждый экран решает конкретную задачу — будь то загрузка фотографий, просмотр, редактирование или анализ.

Переработана система переходов между разделами, улучшены фильтры, добавлена возможность работы с вложенными элементами.

**Редактор фотографий**

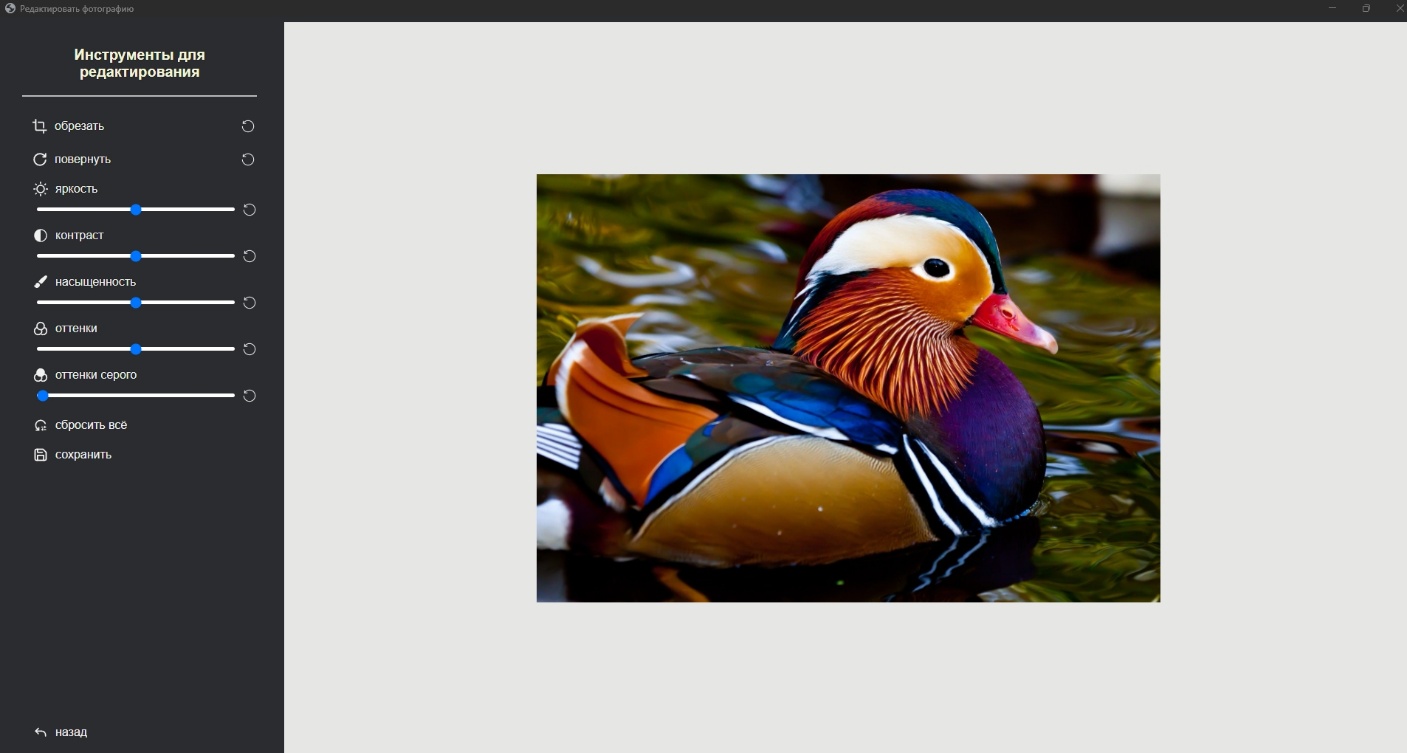
В этом экране пользователь может изменить фотографию использовав такие функции как:

* «Обрезать»
* «Повернуть»
* «Яркость»
* «Контраст»
* «Насыщенность»
* «Оттенки»
* «Оттенки серого»
* «Сбросить всё» сбрасывает примененные изменения
* «Сохранить» сохраняет изображения с приименными изменениями
* «Назад» выйти с режима редактирования

Общий стиль интерфейса (рисунок 3) с вертикальным списком инструментов. С простым шрифтом и нейтральной цветовой палитрой таблица цветов (таблица 2).

|  |  |  |
| --- | --- | --- |
| #2B2C2F | #E6E6E5 | #F1F1F1 |
|  |  |  |

Таблица

****

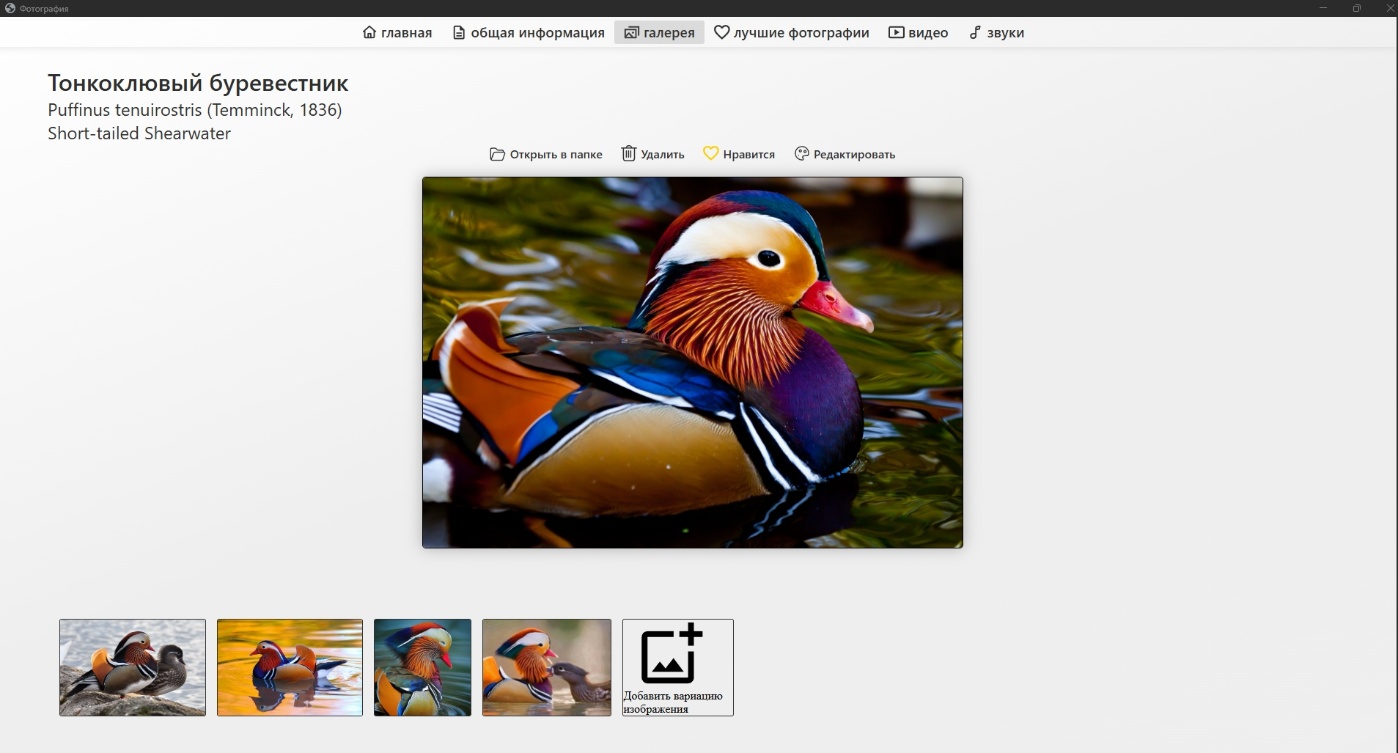
Рисунок

**Экран с фотографией**

Здесь также был изменен дизайн (рисунок 4) с использованием цветов (таблицы 1). Также добавлен новый функционал.

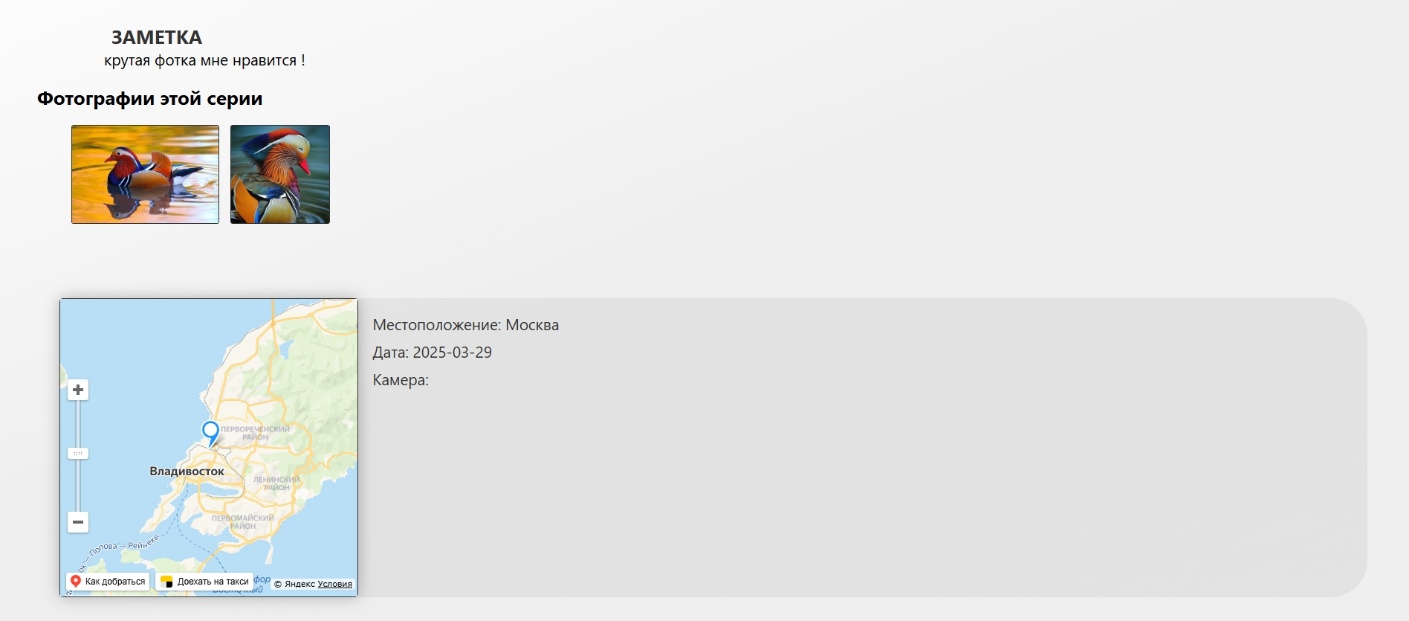
Список новых функций:

* Редактировать
* Добавить вариацию изображения
* Фотографии этой серии
* Если же путь до файла был изменен, теперь можно путь указать новый путь до файла



Рисунок

Также теперь отображаются фотографии этой же серии (рисунок 5), и пользователь может посмотреть или перейти на фотографии это же серии.

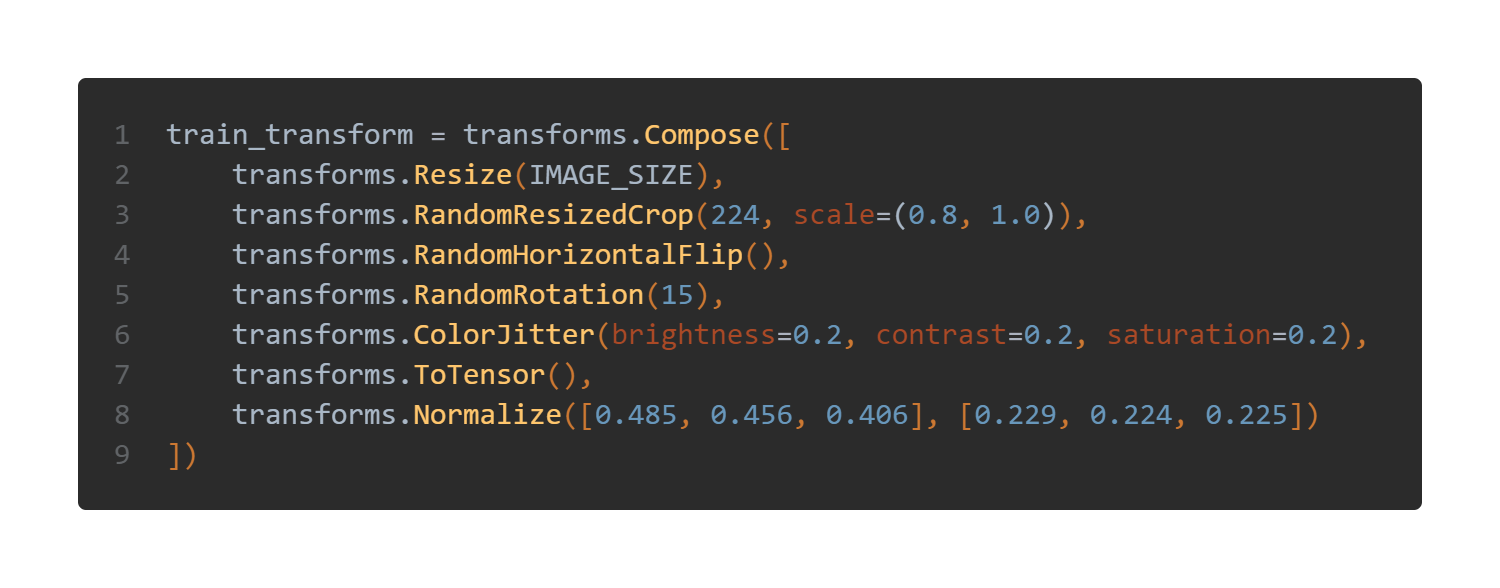


Рисунок

# Реализация и тестирования

Для автоматического распознавания видов птиц на фотографиях была создана и внедрена модель глубокого обучения, основанная на библиотеке PyTorch. В качестве архитектуры была выбрана EfficientNet-B0 — современная свёрточная нейронная сеть, оптимизированная по соотношению точности и скорости работы. Эта модель обеспечивает высокую точность при относительно небольшом размере и скорости работы, что делает её идеальным выбором для интеграции в пользовательские приложения.

Для обучения использовался датасет Birds 525 с платформы Kaggle, содержащий около более чем 500 видов птиц. Все изображения приведены к разрешению 224×224 пикселя для соответствия входным 80 000 изображений требованиям модели. Для повышения обобщающей способности сети применялись стандартные техники аугментации (рисунок 6) изображений: повороты, нормализация, масштабирование, случайные обрезки и изменение яркости.



Рисунок

В начале для того что бы подобрать оптимальные параметры и не тратить сильно много времени на обучение модели было отобрано 200 классов, максимальное число эпох было 15, но обучение заканчивалось раньше т.к использовался early stopping, learning. В зависимости от параметров в среднем обучение занимало от 25 до 35 минут. В результате точность (accuracy) на валидационной выборке удалось получить 83.5% (рисунок 7), на тестовой точность достигла чуть больше 80%.

Описание параметров модели:

* BATCH\_SIZE = 32
* EPOCHS = 15
* LEARNING\_RATE = 1e-4
* criterion = nn.CrossEntropyLoss
* optimizer = optim.Adam
* scheduler = torch.optim.lr\_scheduler.StepLR(optimizer, step\_size=5, gamma=0.5)

|  |  |
| --- | --- |
| Изображение выглядит как линия, График, текст, диаграмма  Контент, сгенерированный ИИ, может содержать ошибки. |  |

Рисунок

На проверке результатов на тестовой выборке было видно, что предсказанные классы не были предсказан не верно, но не совсем критично. Например (рисунок 8) второе изображение 2 модель не точно предсказала класс American\_crow перепутав с Common\_raven, для не разбирающего человека было бы тоже трудно отличить эти два вида если нет сильно заметных видовых различий. В общем модель показывает хорошую точность, но иногда путается в подвидах если на изображении нет заметных отличительных признаков.



Рисунок

# Заключение

В ходе разработки были реализованы значимые функциональные и технологические улучшения, направленные на повышение удобства работы пользователей и автоматизацию процессов систематизации наблюдений. Проект получил развитие по нескольким ключевым направлениям:

* внедрена нейросеть для автоматического определения вида птиц по фотографии
* реализовано автоматическое извлечение данных из метаданных изображений/видио, включая дату, координаты и модель камеры
* создан встроенный редактор изображений с возможностью базовой обработки и сохранения различных вариаций фотографий
* значительно переработан и усовершенствован пользовательский интерфейс с акцентом на простоту, адаптивность и визуальную наглядность
* интегрирована карта на базе API Яндекс.Карт с отображением медиафайлов

# ДНЕВНИК СТУДЕНТА

|  |  |  |
| --- | --- | --- |
| Дата | Краткое содержание выполняемых работ | Отметки руководителя |
| 19.03.25 | Начало работы по обновлению и доработке интерфейса |  |
| 23.03.25 | Анализ существующих решений для редактирования фотографий и видео |  |
| 10.04.25 | Разработка концепции нового интерфейса с учетом добавления функционала редактирования |  |
| 15.04.25 | Реализация базового функционала для редактирования фотографий |  |
| 25.04.25 | Интеграция автозаполнения информации по фотографиям (дата, локация) |  |
| 25.05.25 | Разработка нейросети |  |
| 05.05.25 | Тестирование и оптимизация нового функционала |  |
| 19.05.25 | Написание отчета |  |
| 21.05.25 | Сдача отчета на проверку |  |

|  |  |
| --- | --- |
| Студент: Панченко Н.К. | Изображение выглядит как зарисовка, каллиграфия, Штриховая графика, Шрифт  Автоматически созданное описаниеИзображение выглядит как зарисовка, млекопитающее, искусство, творческий подход  Автоматически созданное описание |
| Руководитель: Малыкина И.А | *подпись* |
|  | *подпись* |