МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**
Кафедра математической кибернетики и компьютерных наук

РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ ДЛЯ РАСПОЗНАВАНИЯ ВИДОВ ПТИЦ ПО ФОТОГРАФИИ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные технологии
факультета КНиИТ
Козынченко Вячеслава Сергеевича

Научный руководитель
ст. преподаватель _____ М. И. Сафрончик

Заведующий кафедрой
к. ф.-м. н., доцент _____ С. В. Миронов

Саратов 2023

ВВЕДЕНИЕ

В настоящее время одними из важных направлений компьютерной обработки информации являются задачи создания систем распознавания и классификации образов и объектов, основывающихся на применении искусственных нейронных сетей и математических моделей автоматического распознавания образов.

Нейронные сети легко поддаются модификациям и настройке. Разнообразие архитектур нейронных сетей позволяет решить широкий спектр задач, используя различные подходы. Производительность и точность данных моделей превосходят оценку эксперта. Таким образом, используя нейронные сети можно сократить затраты на экспертов, а также увеличить скорость вычисления результата.

Нейронные сети могут использоваться в задачах повседневной жизни, например классификация объектов по изображениям. В окружающем мире человек встречает разнообразных животных. Человек не может совершенно точно помнить виды каждого животного, ему может потребоваться большое количество времени для определения их вида. В такой ситуации на помощь приходят приложения с интегрированными в них нейронными сетями, которые позволяют определить, что находится на фото.

Целью данной работы является создание клиент-серверного веб-приложения для распознавания видов птиц и вывода их краткого описания на основе классификации по заданному изображению.

Для этого были поставлены следующие задачи.

- Поиск набора данных, содержащего изображения различных птиц и названия их видов.
- Увеличение набора данных и разделение его на отдельные выборки: для обучения, валидации и тестирования.
- Подбор предобученной нейронной сети и доработка структуры модели под классификацию птиц по изображениям.
- Сбор необходимых данных о птицах: среда обитания, внешний вид, а также дополнительная информация. Представление собранных данных на русском и английском языках.
- Реализация пользовательского интерфейса.
- Интеграция полученной модели в веб-приложение.

Разработанное приложение может быть интересно широкому кругу людей, интересующихся видами и особенностями поведения птиц. Для создания приложения использовались следующие программные средства:

— предобученная свёрточная нейронная сеть `MobileNetV2`;
— веб-фреймворк `Flask`;
— предобученная нейронная сеть для обнаружения объектов `YoloV3`;
— система управления реляционными базами данных `SQLite3`.

Структура и объем работы. Для решения поставленных задач выполнена выпускная квалификационная работа, которая включает в себя введение, 5 основных глав, заключение, список использованных источников из 20 наименований и 12 приложений. Работа изложена на 76 страницах, содержит 31 рисунок и 3 таблицы.

Первая глава имеет название «Машинное обучение» и содержит информацию об основных понятиях данной сферы. Также в главе были описаны внутренние компоненты нейронной сети, которые были использованы в разработке.

Вторая глава имеет название «Клиент-серверная архитектура» и содержит информацию о работе данной архитектуры.

Третья глава имеет название «Реляционные базы данных» и содержит информацию о хранении данных и доступе к ним в таких базах.

Четвертая глава имеет название «Используемые технологии» и содержит перечисление и описание используемых в работе инструментов.

Пятая глава имеет название «Реализация веб-приложения» и содержит подробное описание процесса создания веб-приложения на базе нейронной сети.

Выпускная квалификационная работа заканчивается заключением, списком использованных источников, а также приложениями с кодом А-М.

Основное содержание работы

Объект исследования. Создание клиент-серверного веб-приложения для распознавания видов птиц и их краткому описанию на основе классификации по заданному изображению.

Машинное обучение. Машинное обучение — это раздел искусственного интеллекта, направленный на создание систем, которые обучаются на основе входных данных программы. Такие системы позволяют улучшить производительность и снизить затраты на ручную работу экспертов.

Задача классификации в машинном обучении представляет собой распознавание образов на обучающих данных для того, чтобы обнаружить один и тот же образец в новых данных. В зависимости от количества классов задачу классификации делят на бинарную и мультиклассовую.

При рассмотрении конфигураций нейронных сетей, а также алгоритмов их работы, используются данные, полученные из исследований деятельности мозга. На текущий момент такие модели являются только предположениями в силу ограниченности имеющейся информации о мозге человека, поэтому возможно создание сетей, не существующих в живой материи. Сетью в данном случае называют набор взаимосвязанных искусственных нейронов, связи которых зависят от топологии самой сети. Таким образом, нейронная сеть является попыткой моделирования деятельности человеческого мозга.

Нейрон можно представить в виде математической модели, содержащей следующие объекты:

- $X_1 \dots X_n$ — входные значения;
- Y — выходное значение;
- $W_1 \dots W_n$ — синапсы, которые также называются весами-связями;
- аксон — связь между нейроном и выходным значением, может быть представлен как связь между выходным значением одного нейрона и входным значением другого;
- вес смещения — значение S, помещенное в тело нейрона.

На выходе к полученному значению может применяться одна из следующих функций активации:

- ступенчатая функция;
- сигмоида;
- гиперболический тангенс;

— ReLu (Rectified linear out).

Таким образом, значение на выходе нейрона можно получить с помощью формулы 1.

$$Y = f(\sum_i W_i X_i + S), \tag{1}$$

где f — функция активации нейрона, W_i — веса связей, X_i — входные значения, S — вес смещения нейрона.

В нейронной сети существуют различные слои нейронов. Входным слоем называется слой, содержащий только входные значения. Выходным слоем называется слой, содержащий только выходные значения, то есть результат работы нейронной сети. Остальные слои нейронов называются скрытыми. Если каждый из нейронов одного слоя связан со всеми нейронами следующего слоя, то данные слои называются полносвязными.

В нейронных сетях, построенных для классификации неперекрёстных классов, в выходном слое обычно используется функция активации softmax. Данная функция преобразует значения, полученные в ходе работы сети, в вероятности принадлежности тому или иному классу. Сумма преобразованных значений даёт единицу, выбрав наибольшее значение из этого набора, можно определить наиболее вероятный класс. Вычисление функции активации softmax для i-го нейрона представлено формулой 2.

$$\frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}, \tag{2}$$

где z_i — значение, полученное на последнем слое нейронной сети, i — нейрон, K — количество классов.

Алгоритм процесса обучения нейронной сети можно представить следующим образом:

1. Выбор входных значений.
2. Распространение сигнала по нейронной сети.
3. Получение результата вычисления.
4. Расчет полученной ошибки.
5. Если порог допустимой ошибки пройден — завершение процесса обучения, иначе настройка весов сети, выбор новых входных значений, возвращение на пункт 2.

Для расчета полученной ошибки используется функция потерь. Чем больше значение данной функции, тем сильнее «ошибка» нейросети. Примеры функций потери — среднеквадратичная ошибка и кросс-энтропия.

При перерасчете весов необходимо найти минимум функции потерь. Для решения данной задачи используется метод градиентного спуска. Перед началом нужно задать начальную точку и длину шага градиентного спуска. Остановка производится в случае, если значение функции, представленной формулой 3 примерно равно нулю или достигнуто максимально допустимое количество итераций.

$$a_1 = a_0 - h * f'(a_0), \tag{3}$$

где a_0 — начальная точка, h — длина шага, $f'(a_0)$ — производная исходной функции в точке a_0. Если условия остановки не выполнены, то значением, соответствующим a_0 на следующего итерации становится значение a_1.

Недостатком градиентного спуска является зависимость от начальной точки и длины шага. При неправильно выбранном шаге минимум может быть найден неверно, а также может произойти «зацикливание» алгоритма. Данную проблему можно решить, выбирая на каждой итерации новое значение длины шага.

Переобучение является одной из основных проблем нейронных сетей. Оно заключается в том, что сеть вместо того, чтобы решать поставленную задачу, запоминает обучающие примеры. Таким образом, нейронная сеть будет показывать более низкую точность, если она получит на вход новые данные, отличающиеся от тех, что использовались при обучении. Одним из способов борьбы с переобучением является применение Дропаута.

Дропаут — это метод регуляризации нейронных сетей, заключающийся в исключении нейронов с некоторой вероятностью p. Данная вероятность одинакова для каждого нейрона, что предотвращает возможность слоя полагаться на другие для исправления собственных ошибок. Это предоставляет самым обученным нейронам получить больший вес в нейронной сети. Таким образом, Дропаут обучает ансамбль нейронных сетей, в состав которых входят «выжившие» нейроны, с последующим усреднением результатов.

Сверточная нейронная сеть — это алгоритм глубокого обучения, который может принимать входное изображение, назначать важность (обучаемые веса и смещения) различным аспектам на изображении и иметь возможность отличать

один объект от другого. Предварительная обработка, необходимая в сверточной нейронной сети, намного проще по сравнению с другими алгоритмами классификации.

Архитектура сверточной нейронной сети аналогична схеме подключения нейронов в человеческом мозге и была вдохновлена организацией зрительной коры. Отдельные нейроны реагируют на раздражители только в ограниченной области поля зрения, известной как рецептивное поле. Набор таких полей перекрывается, чтобы покрыть всю визуальную область.

На вход сверточной нейронной сети подается преобразованное в массив чисел изображение. В случае цветных изображений для каждого цветового канала используется собственная таблица значений.

При подготовке данных для обучения свёрточной нейронной сети, нужно определить необходимое количество изображений для каждого класса. Если данных недостаточно, то следует прибегнуть к аугментации изображений. Аугментация представляет собой увеличение обучающей выборки, путём модификаций уже существующих данных.

Transfer learning — это метод машинного обучения, который повторно использует готовую модель, разработанную для одной задачи, в качестве отправной точки новой модели для выполнения другой задачи.

Transfer learning может ускорить прогресс и повысить производительность при обучении новой модели. Данный подход используется, когда затраченное время является важным фактором разработки или ресурсы, необходимые для обучения модели, велики.

MobileNet — это сверточная архитектура нейронной сети, адаптированная под использование мобильными устройствами. Особенность данной архитектуры заключается в использовании операции свёртки с разделением по глубине, а также в отсутствии слоёв max-pooling, так как используется свёртка с шагом фильтра равным двум.

Yolo — это архитектура нейронной сети для обнаружения объектов, которая применяется один раз ко всему изображению. YoloV3 состоит из 106-ти свёрточных слоёв, а также трёх слоёв для детекции объектов разного размера.

Клиент-серверная архитектура. Клиент-серверная архитектура — это вычислительная архитектура, в которой участвуют серверы, распределяющий и контролирующий ресурсы, а также сервисы, запрашиваемые клиентом. Общение

между клиентом и сервером осуществляется с помощью протоколов передачи данных. Одним из таких протоколов является HTTP, использующий отдельные TCP\IP сессии для каждого запроса и осуществляющий доступ к ресурсам с помощью символьных строк, позволяющих идентифицировать запрашиваемый ресурс.

Реляционные базы данных. Реляционная модель данных — это теоретическая основа реляционных баз данных, представляющая собой способ структурирования данных с использованием отношений, которые представляют собой сеточные математические структуры, состоящие из столбцов и строк. Таблицы являются физическим представлением отношений в базе данных.

Преимуществами реляционных баз данных являются:

— Модель реляционной базы данных масштабируется и расширяется, предоставляя гибкую структуру для меняющихся требований и увеличения объёма данных;

— Использование ключей и строгая проверка типизации обеспечивают надёжное хранение данных в допустимых диапазонах.

— Язык структурированных запросов предоставляет возможность запросить любую информацию из таблиц и их соединений. Также возможно применение фильтров, группировок и выбор определённых столбцов для отображения релевантных данных.

Недостаточно грамотное проектирование таких баз данных может привести к усложнению отношений между данными, а также повышению затрат из-за высоких требований к защите информации.

Используемые технологии. В данной работе используются следующие технологии:

— `Jupyter Notebook`;

— `Augmentor` и `splitfolders`;

— `Tensorflow`;

— `Flask` и `Flask-Uploads`;

— `Pillow` и `cvlib`;

— `jinja2`;

— `sqlite3`.

Реализация веб-приложения. В работы был использован датасет, содержащий 525 видов птиц, взят с сайта Kaggle, который предоставляет множе-

ство наборов данных для оттачивания навыков машинного обучения и работы с нейронными сетями. Используемой архитектурой для решения задачи классификации является MobileNetV2. Для обнаружения птиц на фото используется архитектура YoloV3, предварительно обученная на датасете COCO.

Подготовка данных для обучения. Используемый датасет был предварительно разделён на выборки. При анализе этих выборок было обнаружено, что на валидационную и тестовую выборки выделено мало данных, а также выявлен недостаточный объём данных для каждого класса. Классы были объединены и разделены на выборки со следующим процентным соотношением: 70% для тренировочной и по 15% для валидационной и тестовой выборок. На тренировочной выборке были применены аугментации для увеличения количества изображения для каждого на 100 экземпляров.

Подбор параметров для составной модели. Для обучения модели классификации птиц, представленных в датасете, была взята предобученная свёрточная нейронная сеть MobileNetV2. Модель была инициализирована весами `imagenet` с указанием на «замораживание» слоёв предобученной модели, чтобы не обучать их. Также был указан гиперпараметр α со значением 1.4, так как при таком значении сеть показывала максимальный для себя результат на датасете `imagenet`.

Выходным слоем является полносвязный слой с функцией активации softmax. Для данной модели был осуществлён подбор скорости обучения. Лучшим вариантом оказалось использование значения $4e-5$, что помогло достигнуть точности 86.6% на валидационной выборке.

Лучшей моделью оказалась составная сеть со следующими слоями:

1. GlobalAveragePooling2D;
2. BatchNormalization;
3. Dense(512);
4. DropOut(0.35);
5. Dense(256);
6. DropOut(0.25).

Результаты классификации модели показывают наличие особей разного пола в датасете. Если рассматривать цвет оперения, то самцам присуща яркая окраска, а самкам более тусклая. Данное явление называется половым диморфизмом.

Также можно предположить, что в датасете присутствуют гибриды птиц. Факт скрещивания различных видов птиц в природе встречается очень часто. Чаще всего это происходит между близкородственными видами.

Модель птицы. Для работы с данными о птице, а также их локализации был создан класс `Bird`, для визуализации был реализован отдельный класс `showBird`, содержащий данные для текущего языка страницы.

Обнаружение птицы на фото. При анализе вариантов использования веб-приложения были выявлены случаи, когда пользователь загружает фото с несколькими птицами или без птиц. Для того, чтобы модель для классификации не выдавала неправильные результаты из-за невалидного поведения пользователя, был создан класс `Detector`, использующий модель YoloV3, предварительно обученную на датасете COCO, и функцию `detect_common_objects` библиотеки `cvlib`, возвращающую количество найденных птиц на изображении.

Классификация птицы на фото. Для классификации птиц на фото был создан класс `Predictor`, который при инициализации принимает модель MobileNetV2. Для загруженных изображений формата `PNG`, был реализован перевод в цветовое пространство `RGB`. Результаты классификации представляют собой вероятности принадлежности к тому или иному классу, чтобы получить наиболее вероятный класс для изображения используется метод `argmax` библиотеки `NumPy`.

Взаимодействие с базой данных. Для работы с локализацией осуществлён перенос информации из `Excel`-файла в базу данных. Для этого была выбрана однофайловая система управления базами данных `sqlite3`. Были созданы три таблицы:

- `Classes` — таблица, хранящая идентификаторы классов и их названия, под которые адаптирована модель MobileNetV2;
- `ruBirds` и `enBirds` — таблицы, хранящие информацию о птицах для русского и английского языков.

Локализация. Для работы с английским и русским языками был создан класс `Translator`, определяющий язык формы для загрузки данных, сообщений об ошибках, а также информации о классе птиц.

Локализация информации о классе птицы, полученной при обработке загруженного изображения, происходит в несколько этапов:

1. На вход функции поступает идентификатор птицы, полученный при клас-

сификации или из запроса страницы о птице.

2. С помощью идентификатора осуществляется поиск данных о птице для двух языков из соответствующих таблиц.
3. С помощью методов для извлечения данных определённого языка объекта `Bird` подготавливается необходимая локализация данных о птице для вывода.

Вёрстка веб-приложения. Вёрстка веб-приложения осуществлялась с помощью языка гипертекстовой разметки `HTML` и кода стилизации `CSS`. Дополнительно был подключен фреймворк `Bootstrap`, предоставляющий готовые стили и объекты для быстрой вёрстки веб-приложений.

Была реализована навигационная панель, содержащая выпадающий список для выбора языка страницы, а также логотип, путь до которого вычисляется с помощью функции `url_for`. Данная панель является адаптивной, поэтому при уменьшении масштаба объекты панели скрываются и появляются ниже при нажатии на специальную кнопку.

Для работы с изображениями были созданы два блока элементов: форма для загрузки изображений и блок с ошибками на форме. Форма обозначена тегом `<form>`, в атрибутах которого указан тип отправляемого запроса `POST`. Для вывода ошибок на форме выделен отдельный элемент с тегом `<span>`. Также реализована возможность просмотра загруженного фото с помощью отдельного блока элементов.

Для вывода информации о распознанной птице на фото был создан отдельный блок. С помощью тега `<p>` происходит деление информации на следующие поля: название, среда обитания, внешний вид, общая информация.

Для вывода информации о птице без подачи файла для классификации была реализована «Страница птицы». На данной странице форма загрузки изображений не отображается. Все остальные элементы выполнены аналогично.

Маршрутизация. Маршрутизация в фреймворке `Flask` определяет `URL`-адреса для доступа пользователя к методам через `HTTP`-запросы. В данной работе существуют следующие маршруты:

— главная страница;
— предпросмотр загруженного изображения;
— страница птицы;
— случайное фото для страницы птицы.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы было разработано веб-приложение с использованием фреймворка `Flask` и его дополнений. Также была обучена и внедрена в приложение модель `MobileNetV2`, обладающая точностью в 86.6%. Итоговая модель имеет следующую структуру:

1. GlobalAveragePooling2D;
2. BatchNormalization;
3. Dense(512);
4. DropOut(0.35);
5. Dense(256);
6. DropOut(0.25).

Хранение и доступ к информации были осуществлены под управлением системы управления реляционными базами данных `SQLite3`. Для повышения простоты развёртывания приложения был создан конфигурационный файл, который легко менять под необходимые требования.

Для проверки валидного ввода пользователем была использована предобученная модель `YoloV3`, которая проверяет наличие одной птицы на фото. Данное решение помогло повысить точность результатов, выводимых пользователю. Точность данной модели может быть улучшена, если её дообучить на текущем датасете.

Разработанная модель классификации птиц может быть улучшена увеличением количества изображений женских особей или разбиением видов на самцов и самок, если различия существенны, например разный цвет оперения или другие отличия внешнего вида.

Также может быть произведён переход к динамической веб-странице или мобильному приложению, что позволяет сделать небольшой объём памяти, занимаемый моделью классификации.

Может быть улучшен маршрут до страницы, содержащий информацию об одной птице, добавлением «Базы знаний». Такое улучшение может увеличить время пребывания пользователя странице, так как появляется больше доступного функционала и дополнительного объёма информации.

Таким образом, все поставленные задачи выполнены и цель работы достигнута.