Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Брянский государственный инженерно-технологический университет»

Кафедра «Информационные технологии»

КУРСОВАЯ РАБОТА

по дисциплине: «Машинное обучение»

Разработка информационной системы для классификации рукописных цифр по фотографии

КР-02068025-09.03.01-2.033

Автор работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Мельников Артем Александрович

Группа: ИВТ-301 № Зачётной книжки: 21-2.035

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_ канд. экон. наук, доцент О.Д. Казаков

Нормоконтроль \_\_\_\_\_\_\_\_\_\_\_\_\_\_ канд. экон. наук, доцент О.Д. Казаков

Доступ к защите: «\_\_»\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ канд. экон. наук, доцент О.Д. Казаков

Дата защиты: «\_\_»\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_

Члены комиссии: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ канд. экон. наук, доцент О.Д. Казаков

Брянск 2023

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc153808590)

[1. ИНЖЕНЕРИЯ МАШИННОГО ОБУЧЕНИЯ 4](#_Toc153808591)

[1.1 Сверточные нейронные сети (CNN) 4](#_Toc153808592)

[1.2 Жизненный цикл проекта машинного обучения 6](#_Toc153808593)

[1.3 Что такое оценка качества моделей машинного обучения. 12](#_Toc153808594)

[2 ПОДГОТОВКА ДАННЫХ И РЕАЛИЗАЦИЯ МОДЕЛЕЙ ДЛЯ КЛАССИФИКАЦИИ ТРАНСПОРТНЫХ СРЕДСТВ ПО ИЗОБРАЖЕНИЮ 17](#_Toc153808595)

[2.1 Постановка задачи, сбор данных 17](#_Toc153808597)

[2.2 Предобработка данных 17](#_Toc153808598)

[2.3 Реализация модели машинного обучения 19](#_Toc153808599)

[3 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА ДЛЯ КЛАССИФИКАЦИИ ТРАНСПОРТНЫХ СРЕДСТВ ПО ИЗОБРАЖЕНИЮ 26](#_Toc153808600)

[3.1 Разработка чат-бота для классификации транспортных средств по изображению 27](#_Toc153808602)

[3.2 Разработка android-приложения для классификации транспортных средств по изображению 28](#_Toc153808603)

[ЗАКЛЮЧЕНИЕ 35](#_Toc153808604)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 36](#_Toc153808605)

# ВВЕДЕНИЕ

В современном мире машинное обучение является динамично развивающимся направлением технологий, оказывающее глубокое воздействие на различные области человеческой деятельности. Машинное обучение является мощным инструментом для обработки и анализа информации, открывая новые горизонты для решения сложных задач.

Одной из основных областей его применения является классификация объектов по изображению.

Тема курсовой работы заостряет своё внимание на задаче классификации транспортных средств. Распознавание транспортных средств по изображению представляет значительный интерес разных областей, включая обеспечение безопасности и эффективное управление дорожным движением

В рамках данной курсовой работы ставится перед собой цель реализации модели машинного обучения для эффективной классификации видов транспортных средств по изображениям. Для достижения этой цели определены следующие основные задачи:

* Реализация модели машинного обучения для классификации видов транспортных средств по изображениям;

Для достижения поставленной цели были выдвинуты следующие задачи:

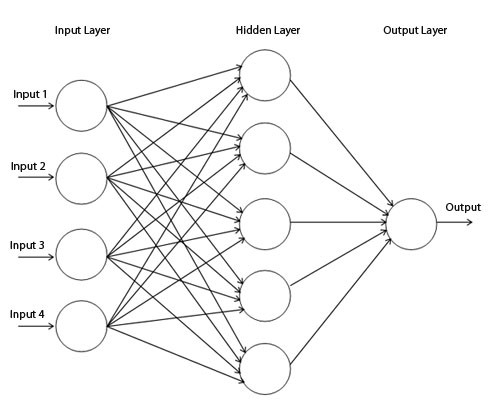
1. Изучение инженерии машинного обучения;
2. Подготовка данных и реализация модели машинного обучения;
3. Разработка программного обеспечения для предоставления модели машинного обучения конечному пользователю.

# ИНЖЕНЕРИЯ МАШИННОГО ОБУЧЕНИЯ

## Сверточные нейронные сети (CNN)

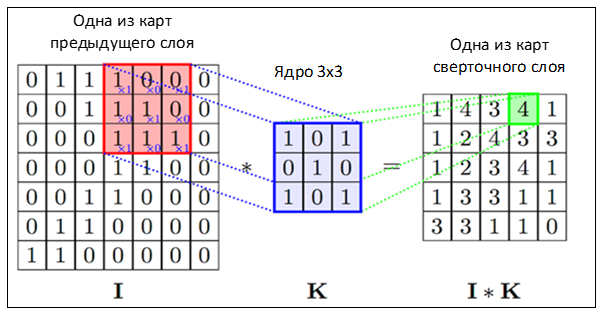
Нейронная сеть представляет собой масштабный распределенный параллельный процессор, сформированный из элементарных узлов обработки информации. Эти узлы служат для накопления опытных знаний и представления их для последующей обработки. Математическая модель нейронной сети аналогична структуре человеческого мозга, обладая способностью обработки поступающей информации из окружающей среды и настройкой синаптических весов для установления взаимосвязей между нейронами.

Пример упрощенной схемы полносвязной нейронной сети представлен на рисунке 1.1, где выделены слои входных, скрытых и выходных нейронов. Алгоритм обучения определяет последовательность настройки синаптических весов, обеспечивая необходимую структуру взаимосвязей нейронов.

  
Рисунок 1.1 – Пример схемы простой полносвязной нейронной сети

Сверточные нейронные сети (CNN) представляют собой вариацию многослойных простейших видов искусственных нейронов, биологически вдохновленных работами Хьюбела и Визела по зрительной коре кошки. Зрительная кора обладает сложной структурой, где клетки чувствительны к малым областям поля зрения, называемым рецептивным полем. Эти клетки, действуя как локальные фильтры, хорошо адаптированы к обработке пространственной локальной корреляции в естественных изображениях.

Процесс сверточных нейронных сетей заключается в определении контуров и цветового представления входного изображения через операцию свертки. В результате этой операции матрица признаков уменьшается, что является отличительной чертой данных сетей. Операция свертки осуществляется с использованием ядра, состоящего из матрицы весов, и входного изображения, представленного матрицей признаков. Процесс этой операции проиллюстрирован на рисунке 1.2.

  
Рисунок 1.2 – Процесс операции свёртки

Для изображений с одним каналом на практике используется простой процесс свертки. В случае многоканальных изображений, таких как RGB, процесс не сильно отличается. Для каждого канала используется отдельное ядро с собственными весами. Свернутые значения суммируются, добавляется скалярное смещение и получается выходной канал. Эта комбинация ядер всех каналов и одного выходного канала называется фильтром, который обрабатывает вход и создает выходной канал.

Результат для различных фильтров одинаков: каждый обрабатывает вход с уникальным набором ядер и смещением, создавая отдельный выходной канал. Эти каналы объединяются для получения общего выхода, при этом количество выходных каналов соответствует числу фильтров. Часто применяется нелинейность перед передачей входа в следующий сверточный слой, который повторяет этот процесс.

.

## Жизненный цикл проекта машинного обучения

Проекты машинного обучения (ML) имеют определенный жизненный цикл, аналогичный циклу DevOps, который определяет разработку и развертывание моделей ML. Жизненный цикл проекта машинного обучения состоит из нескольких этапов, которые обеспечивают последовательный и эффективный способ запуска ваших проектов машинного обучения в производство.

Первым этапом жизненного цикла проекта машинного обучения является систематизация проблемы и определение показателей успеха.

На этом этапе команда оценивает, может ли данная бизнес-задача быть решена с помощью ML, и определяет критерий успеха для оценки прогноза модели.

Команда состоит как минимум из специалистов по обработке и анализу данных и бизнес-эксперта (SME).

Эксперт в предметной области (SME) — это человек, обладающий специальными знаниями или навыками в определенной области или области. В контексте проектов машинного обучения малым и средним бизнесом обычно является человек, который глубоко понимает бизнес-проблему, которую пытается решить проект, и может предоставить ценную информацию и рекомендации о том, как лучше всего подойти к проблеме с помощью методов машинного обучения.

Второй этап заключается в приеме, очистке и маркировке данных. Команда оценивает, доступны ли данные, необходимые для обучения модели, и создает компоненты для приема данных из различных источников, очистки захваченных данных, возможной маркировки данных и их сохранения.

Прием данных — это процесс получения и импорта данных из различных источников в систему хранения данных для дальнейшей обработки и анализа. В контексте проекта машинного обучения прием данных — это первый шаг в подготовке данных для обучения и оценки моделей.

Он включает в себя сбор данных из различных источников, таких как файлы, базы данных или внешние API, а затем очистку и предварительную обработку данных, чтобы гарантировать, что они находятся в формате, который может использоваться алгоритмами машинного обучения.

Прием данных также включает проверку данных, чтобы гарантировать их точность и полноту.

Очистка данных – это процесс выявления и устранения или изменения неточностей и несоответствий в наборах данных, чтобы их можно было использовать для анализа. Это важный шаг в процессе подготовки данных для проектов машинного обучения, поскольку он помогает обеспечить точность, согласованность и отсутствие ошибок в данных. Очистка данных может включать в себя такие задачи, как удаление повторяющихся записей, исправление ошибок при вводе данных, удаление выбросов и заполнение отсутствующих значений.

Маркировка данных – это процесс присвоения данным соответствующих меток или аннотаций. Это часто делается вручную комментаторами-людьми, но также может быть выполнено с помощью автоматизированных методов, таких как алгоритмы машинного обучения. Метки обычно используются для классификации или распределения данных по разным классам или группам, что упрощает обучение моделей машинного обучения на данных.

Это важный шаг на этапе предварительной обработки данных проекта машинного обучения, поскольку он помогает убедиться, что данные правильно структурированы и готовы к обучению.

Третий этап — это Feature Engineering (FE), который заключается в преобразовании необработанных данных в функции, которые более актуальны для данной проблемы.

Инжиниринг признаков — это процесс создания, преобразования и выбора признаков из необработанных данных с целью повышения производительности модели машинного обучения.

Это важный шаг в процессе машинного обучения, так как он может оказать существенное влияние на окончательную производительность модели.

Разработка функций обычно включает следующие этапы:

Исследование данных. Понимание структуры, распределения и взаимосвязей необработанных данных.

Выбор функций: определение наиболее важных функций из необработанных данных, которые будут использоваться для обучения модели.

Создание функций: создание новых функций из необработанных данных путем объединения, преобразования или агрегирования существующих функций.

Масштабирование функций: нормализация или масштабирование функций, чтобы они имели схожие диапазоны, что может помочь повысить производительность некоторых моделей машинного обучения.

Кодирование функций: кодирование категориальных переменных в числовые значения, чтобы их можно было использовать в алгоритме машинного обучения.

Этот этап имеет решающее значение, поскольку он помогает улучшить производительность модели.

Четвертый этап — построение и настройка модели, когда команда начинает экспериментировать с разными моделями и разными гиперпараметрами.

Команда тестирует модель на заданном наборе данных и сравнивает результаты каждой итерации.

Затем команда определяет лучшую модель для заданных показателей успеха и сохраняет модель в реестре моделей.

Реестр моделей — это центральный репозиторий, в котором хранятся различные версии моделей машинного обучения и которые управляют ими. Он используется для отслеживания разработки, тестирования и развертывания моделей и обеспечивает способ организации, совместной работы и управления моделями.

Пятый этап — проверка модели, когда команда проверяет модель на новом наборе данных, недоступных во время обучения.

Проверка модели — это процесс оценки производительности модели на наборе данных, который она не видела во время обучения. Это делается для того, чтобы модель могла хорошо обобщать новые, невидимые данные и выявлять любые отклонения в модели.

Во время проверки прогнозы модели сравниваются с истинными значениями данных и различными показателями, такими как точность, точность, отзыв и оценка F1 используется для оценки производительности модели.

Точность – это показатель того, насколько часто модель правильно предсказывает класс заданной точки данных. Это отношение правильно предсказанных наблюдений к общему количеству наблюдений.

Точность – это показатель того, сколько положительных прогнозов, сделанных моделью, на самом деле верны. Это отношение истинных срабатываний к сумме истинных срабатываний и ложных срабатываний.

Отзыв – это показатель того, сколько реальных положительных наблюдений в данных правильно предсказано моделью. Это отношение истинных положительных результатов к сумме истинных положительных и ложных отрицательных результатов.

Оценка F1 – это показатель, сочетающий в себе точность и полноту. Это гармоническое среднее значение точности и полноты, при котором показатель F1 достигает наилучшего значения при 1 (идеальная точность и полнота), а наихудшее — при 0.

Этот этап имеет решающее значение, поскольку он определяет, достаточно ли обобщена модель для невидимых данных, или модель хорошо работает только с обучающими данными, но не с невидимыми данными, избегая переобучения.

Шестой этап — развертывание модели, когда команда выбирает модель из реестра моделей, упаковывает ее и развертывает для использования.

Затем развернутую модель можно использовать для прогнозирования или классификации новых данных. Процесс развертывания модели может также включать в себя шаги по мониторингу и обслуживанию модели, например мониторинг ее производительности и ее обновление по мере необходимости.

Здесь можно использовать традиционные процессы DevOps, чтобы сделать модель доступной как услугу.

Некоторые распространенные методы включают в себя:

Модель как услуга (MaaS). В этом методе модель развертывается как веб-служба, и клиенты могут получить доступ к модели через конечную точку API. Это позволяет делать прогнозы в реальном времени и легко интегрироваться с другими системами.

Развертывание библиотеки. В этом методе модель упаковывается в виде библиотеки, которую могут использовать другие приложения. Это полезно, когда модель необходимо интегрировать с существующими системами или когда несколько приложений должны использовать одну и ту же модель.

Контейнеризация. Этот метод включает упаковку модели и ее зависимостей в контейнер (например, Docker) и ее развертывание на платформе оркестрации контейнеров (например, Kubernetes). Это позволяет легко масштабировать и управлять моделью.

Развертывание в облаке. Многие поставщики облачных услуг предлагают платформы машинного обучения, позволяющие легко развертывать модели. Например, Google Cloud ML Engine, AWS SageMaker и Azure ML Studio<. /em> предоставляют платформу для обучения, развертывания и управления моделями.

Локальное развертывание. В этом методе модель развертывается локально, обычно на сервере. Этот метод полезен, когда важны безопасность данных и конфиденциальность.

Седьмой и последний этап — мониторинг и проверка, когда модель будет постоянно контролироваться на предмет времени отклика, точности прогнозов и того, соответствуют ли входные данные данным, на которых обучается модель.

Этот этап имеет решающее значение для обнаружения и устранения любых аномалий, таких как обнаружение выбросов, и создания предупреждений для переобучения модели с новыми наборами данных.

Это гарантирует, что модель будет продолжать работать хорошо с течением времени, а любые проблемы или изменения в данных будут обнаружены и устранены.

Жизненный цикл проекта машинного обучения является важнейшим аспектом разработки и развертывания машинного обучения, обеспечивая последовательный и эффективный способ запуска ваших проектов машинного обучения в производство. Хотя этапы могут показаться простыми, в реальном мире может быть несколько веских причин вернуться к предыдущим этапам в определенных случаях.

Каждый этап играет решающую роль в общем успехе проекта, и важно обращать внимание на детали на каждом этапе. Также важно помнить, что в реальном мире могут быть случаи, когда необходимо вернуться к предыдущим этапам для решения любых проблем, которые могут возникнуть.

Комбинируя жизненный цикл ML с MLOps, организации могут создавать более быстрые, надежные и безопасные проекты ML, упрощая внедрение моделей ML в производство и их поддержку с течением времени.

## Что такое оценка качества моделей машинного обучения.

Оценка качества моделей машинного обучения – это процесс измерения и оценки того, насколько хорошо модель способна предсказывать или классифицировать данные. Она позволяет определить, насколько точно модель может предсказывать значения целевой переменной или классифицировать объекты.

Оценка качества моделей машинного обучения является важной частью процесса разработки и применения моделей. Она позволяет оценить эффективность модели и сравнить ее с другими моделями или алгоритмами. Оценка качества моделей также помогает определить, насколько модель может быть полезной в реальных задачах и принимать решения на основе предсказаний.

Для оценки качества моделей машинного обучения используются различные метрики, такие как точность, полнота, F-мера, средняя абсолютная ошибка и другие. Эти метрики позволяют измерить различные аспекты качества модели, такие как точность предсказаний, способность модели обнаруживать и классифицировать объекты правильно, а также способность модели обобщать на новые данные.

Оценка качества моделей машинного обучения является важным этапом в разработке и применении алгоритмов искусственного интеллекта. Она позволяет определить, насколько модель может быть полезной в реальных задачах и принимать решения на основе предсказаний.

Оценка качества моделей машинного обучения имеет следующие цели:

1)Сравнение различных моделей.

Оценка качества моделей позволяет сравнить различные алгоритмы и выбрать наиболее подходящую модель для конкретной задачи. Сравнение моделей основывается на их способности делать точные предсказания, обобщать на новые данные и обнаруживать и классифицировать объекты правильно.

2)Определение оптимальных параметров модели.

Многие модели машинного обучения имеют параметры, которые можно настраивать для достижения лучшей производительности. Оценка качества моделей позволяет определить оптимальные значения этих параметров, чтобы модель работала наилучшим образом.

3)Оценка надежности модели.

Оценка качества моделей также позволяет оценить надежность модели и ее способность обобщать на новые данные. Надежная модель должна давать стабильные и точные предсказания на различных наборах данных.

4.Принятие решений на основе предсказаний

Оценка качества моделей позволяет принимать решения на основе предсказаний модели. Если модель имеет высокую точность и надежность, ее предсказания могут быть использованы для принятия решений в различных областях, таких как медицина, финансы, маркетинг и другие.

В целом, оценка качества моделей машинного обучения является неотъемлемой частью процесса разработки и применения алгоритмов искусственного интеллекта. Она помогает выбрать наиболее подходящую модель, определить оптимальные параметры, оценить надежность и принимать решения на основе предсказаний модели.

Метрики оценки качества моделей машинного обучения используются для измерения эффективности и точности моделей. Они позволяют оценить, насколько хорошо модель справляется с поставленной задачей и насколько точные предсказания она делает.

Существует множество метрик оценки качества моделей, и выбор конкретной метрики зависит от типа задачи и типа данных. Некоторые из наиболее распространенных метрик включают:

Точность (Accuracy)

Точность – это метрика, которая измеряет долю правильных предсказаний модели относительно общего числа предсказаний. Она вычисляется как отношение числа правильных предсказаний к общему числу предсказаний.

Полнота (Recall)

Полнота – это метрика, которая измеряет способность модели обнаруживать все положительные примеры. Она вычисляется как отношение числа правильно обнаруженных положительных примеров к общему числу положительных примеров.

Точность (Precision)

Точность – это метрика, которая измеряет способность модели предсказывать только правильные положительные примеры. Она вычисляется как отношение числа правильно предсказанных положительных примеров к общему числу предсказанных положительных примеров.

F-мера (F1-score)

F-мера – это гармоническое среднее между точностью и полнотой. Она представляет собой сбалансированную метрику, которая учитывает и точность, и полноту. F-мера вычисляется как среднее гармоническое между точностью и полнотой.

Средняя абсолютная ошибка (Mean Absolute Error)

Средняя абсолютная ошибка – это метрика, которая измеряет среднее абсолютное отклонение предсказанных значений от истинных значений. Она вычисляется как среднее значение абсолютных разностей между предсказанными и истинными значениями.

Средняя квадратичная ошибка (Mean Squared Error)

Средняя квадратичная ошибка – это метрика, которая измеряет среднее квадратичное отклонение предсказанных значений от истинных значений. Она вычисляется как среднее значение квадратов разностей между предсказанными и истинными значениями.

Это лишь некоторые из метрик оценки качества моделей машинного обучения. Выбор конкретной метрики зависит от задачи и типа данных, поэтому важно выбирать метрику, которая наилучшим образом отражает требования и цели проекта.

Кросс-валидация

Кросс-валидация – это метод оценки качества моделей машинного обучения, который позволяет оценить, насколько хорошо модель будет работать на новых, ранее не виденных данных. Он основан на разделении доступных данных на обучающую и тестовую выборки.

Основная идея кросс-валидации заключается в том, чтобы разделить исходные данные на K равных частей, называемых фолдами. Затем модель обучается на K-1 фолдах и тестируется на оставшемся фолде. Этот процесс повторяется K раз, каждый раз с использованием разных фолдов в качестве тестовой выборки. В конце получается средняя оценка качества модели на всех K тестовых выборках.

Кросс-валидация позволяет более надежно оценить качество модели, так как она учитывает вариативность данных и предотвращает переобучение. Она также помогает выбрать наилучшие гиперпараметры модели, такие как глубина дерева в случае деревьев решений или коэффициент регуляризации в линейной регрессии.

Существует несколько различных методов кросс-валидации, таких как K-fold, Stratified K-fold, Leave-One-Out и другие. Каждый из них имеет свои особенности и применяется в зависимости от типа данных и задачи.

Разделение выборки на обучающую и тестовую

Разделение выборки на обучающую и тестовую является важным шагом при оценке качества моделей машинного обучения. Этот процесс позволяет проверить, насколько хорошо модель обобщает данные, которые она не видела во время обучения.

Обучающая выборка – это часть данных, которая используется для обучения модели. Модель “изучает” закономерности и шаблоны в данных, чтобы научиться делать предсказания. Обучающая выборка должна быть достаточно большой и представлять разнообразные случаи, чтобы модель могла обобщить эти знания на новые данные.

Тестовая выборка – это независимая часть данных, которая не используется во время обучения модели. Она используется для оценки качества модели и проверки ее способности делать предсказания на новых данных. Тестовая выборка должна быть представительной и содержать данные, которые модель не видела во время обучения.

Разделение выборки на обучающую и тестовую может быть выполнено случайным образом или с использованием стратификации. Случайное разделение просто случайным образом распределяет данные между обучающей и тестовой выборками. Стратификация, с другой стороны, сохраняет пропорции классов или распределение целевой переменной в обучающей и тестовой выборках, что особенно полезно при работе с несбалансированными данными.

Обычно выборка разделяется на обучающую и тестовую в пропорции 70/30 или 80/20. Это означает, что 70% или 80% данных используются для обучения модели, а оставшиеся 30% или 20% – для тестирования. Однако, в зависимости от размера данных и сложности задачи, эти пропорции могут быть изменены

# 2 ПОДГОТОВКА ДАННЫХ И РЕАЛИЗАЦИЯ МОДЕЛЕЙ ДЛЯ КЛАССИФИКАЦИИ ЦВЕТОВ ПО ИЗОБРАЖЕНИЮ



## Постановка задачи, сбор данных

В рамках выполнения поставленной цели курсовой работы, основным направлением является разработка модели машинного обучения для классификации 4 видов транспортных средств по изображениям. Задача представлена в виде решения проблемы с классификации 8 разных видов транспорта.

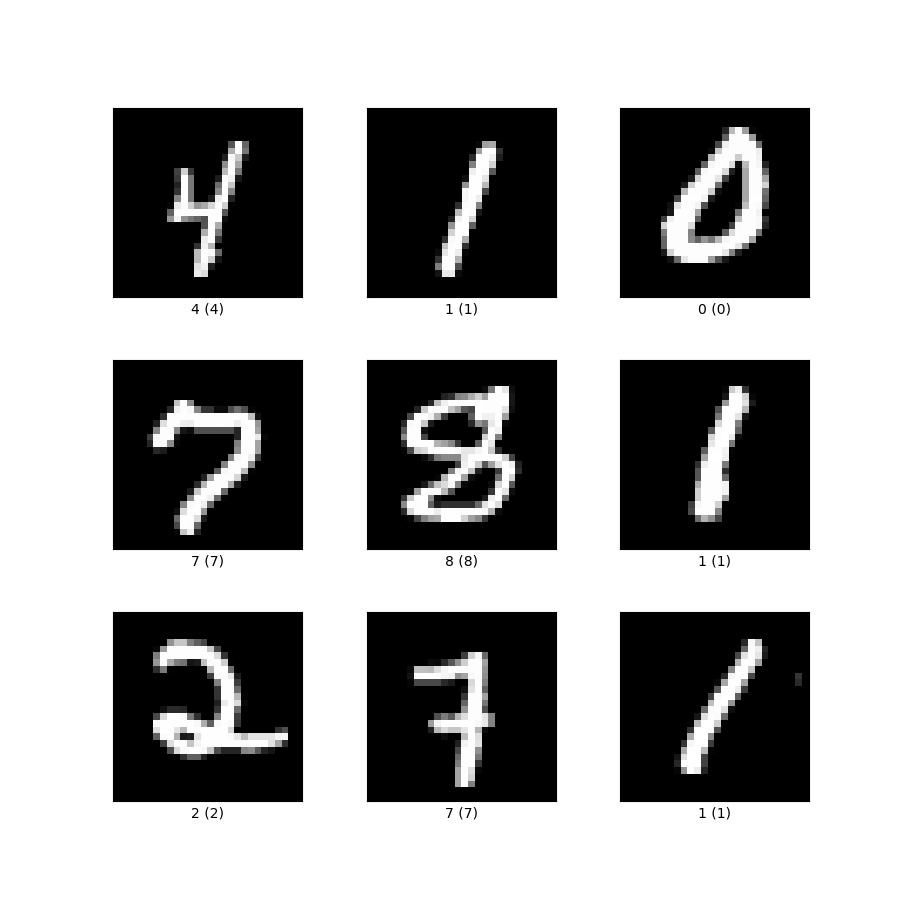
Важным этапом является разметка данных, в котором каждое изображение распределяется к определённому классу, тем самым обеспечивая максимальную точность в обучающем наборе данных.

При поиске подходящего набора данных был найден датасет на сайте Kaggle состоящий из нескольких сотен изображений, который подходит для поставленной задачи.

## Предобработка данных

В процессе предобработки данных осуществлялась подготовка найденных изображений к последующему этапу обучения модели. Изначально я разместил файл с датасетом на Google drive, для более удобного использования. После подключал Google drive в среде выполнения и предоставлял переменной путь к директории с изображениями, после чего я разбил данные на тренировочные и валидационные с заранее заданными размерами.

После завершения предобработки был получен список классов. При помощи библиотеки matplotlib через объект **plt** метод figure() я создал график размером 10 на 10 в котором вывел 9 изображений с надписями классов к которым они принадлежат, для наглядной видимости соответствия перед обучением модели, Рисунок 2.2.1

  
Рисунок 2.2.1 – График соответствия изображений к классам

Далее, для обработки данных я выполнил оптимизацию загрузки данных в модель. В первую очередь я кешировал данные, чтобы избежать повторной загрузки данных при помощи метода cache(). Потом я перемешал данные, для более эффективного обучения модели, чтобы предотвратить переобучение при помощи перемешивания порядка данных в каждой новой эпохе, это я сделал при помощи метода shuffle(1000). Чтобы ускорить процесс обучения модели благодаря загрузки следующего блока данных в память во время обработки текущего я использовал метод prefetch(buffer\_size=AUTOTUNE).

В итоге, при помощи этих действий были созданы оптимизированные наборы данных, которые готовы к эффективному использованию при обучении модели.

## Реализация модели машинного обучения

При рассмотрении основных моделей для решения задачи классификации изображений было принято решение в пользу сверточной нейронной сети (CNN) из-за ее специфических преимуществ. Эти преимущества включают в себя способность автоматически извлекать локальные шаблоны из изображений, формирование иерархии признаков, инвариантность к некоторым трансформациям, эффективное многократное использование параметров и успешное применение в задачах классификации изображений.

Сверточные нейронные сети автоматически выявляют особенности в различных частях изображения, начиная с более простых элементов и постепенно переходя к более сложным абстракциям. Это делает их идеальными для работы с визуальной информацией. Кроме того, они обладают инвариантностью к определенным трансформациям, что улучшает их устойчивость к различным условиям съемки изображений.

Выбор CNN также обоснован их способностью эффективно использовать параметры, что особенно важно при обработке изображений большого размера. Такие сети успешно применяются в задачах классификации изображений, демонстрируя выдающуюся производительность по сравнению с другими типами нейронных сетей.

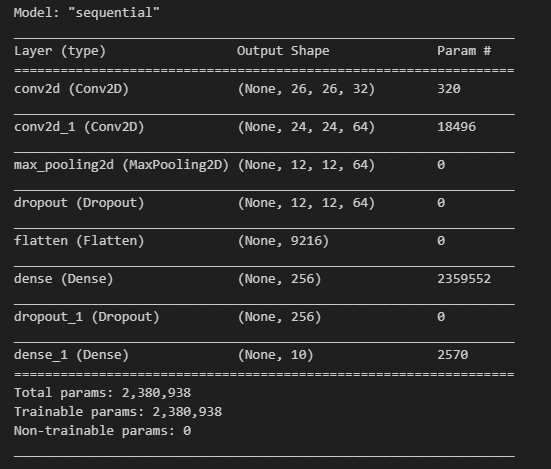
После выбора сверточной нейронной сети возник вопрос о методе обучения, и было принято решение в пользу обучения с учителем. Обучение с учителем представляет собой метод машинного обучения, при котором модель обучается на размеченных данных, содержащих пары "вход-выход". Это обеспечивает точное обучение модели с использованием явных ответов и позволяет измерять производительность на основе точных метрик.

Оптимизатор определяет метод обновления весов модели, в качестве отпимизатора был выбран Adam т.к. он является часто используемым при глубоком обучении.

После определения основных параметров последовал процесс реализации модели, который состоит из нескольких этапов:

1. Определение архитектуры модели.
2. Компиляция модели. При вызове метода компиляции указывается несколько параметров, такие как: оптимизатор: adam, функция потерь lass – она определяет какая функция будет использоваться для оценки разницы между предсказанными значениями и фактическими метками во время обучения, и указываются метрики для отслеживания в процессе обучения, здесь используется метрика accuracy.
3. Обучение модели. Метод обучения модели принимает несколько параметров: обучающий набор данных, который является тренировочным, валидационный набор данных, он используется для оценивания производительности модели в процессе обучения, и количество эпох, которое равно 10 эпохам.
4. Оценка качества обучения. Она производится при помощи метрикам качества.
5. Сохранение модели: после обучения модели, для последующего её использования. В своём случае я буду сохранять модель в формате h5 и tflite, для её использования при реализации программных продуктов.

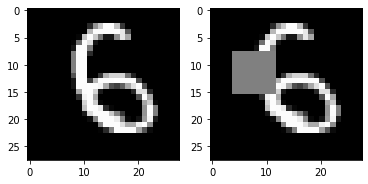
Модель имеет следующую структуру как на Рисунке 2.3.1

  
Рисунок 2.3.1 – структрура модели CNN

После обучения модели были проанализированы метрики качества, пример на рисунке 2.3.2.

  
Рисунок 2.3.2 – Метрики качества

Проанализировав их, я пришёл к выводу что модель работает плохо, и решил проблему при помощи аугментации данных, т.е. увеличением количества данных, что поможет модели обучиться более устойчиво. В методе Sequential из библиотеки Keras я вызываю нескольлко методов в которые передаю параметры для изменения имеющихся данных, например: Метод RandomFlip принимает аргумент horizontal который означает, что будет производиться переворот изображения по горизонтали, этот метод также принимает параметры указывающие на размеры входных изображений, метод RandomRotation вращает случайным образом изображение на заданный угол и метод RandomZoom случайным образом увеличивает или уменьшает изображение. После проведения этих действий, я строю график для визуального представления действий, Рисунок 2.3.3

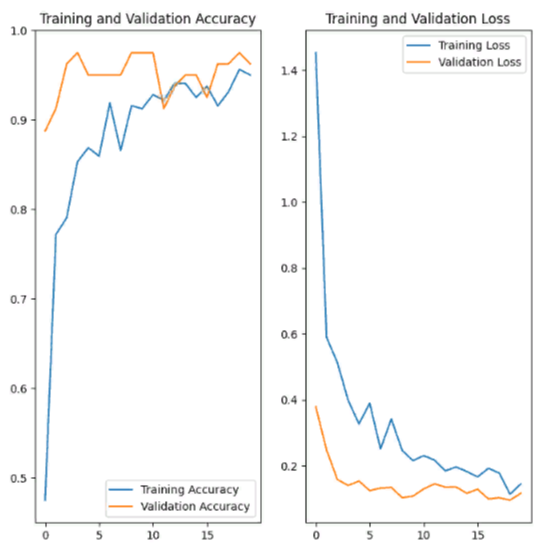
  
Рисунок 2.3.3 – изображения полученные путём аугментации

После этих действий я использую заранее обученную модель MobileNetV2, в качестве базовой, которая предоставляется TensorFlow, она принимает в качестве параметров: размеры входных изображений, параметр include\_top=False указывает на то, что верхний слой модели не должен быть включен, т.к. будет добавляться свои слои для классификации. Далее используется метод base\_model.trainable которому присвоено значение False, что означает, что веса базовой модели «замораживаются» или же не поддаются обучению.

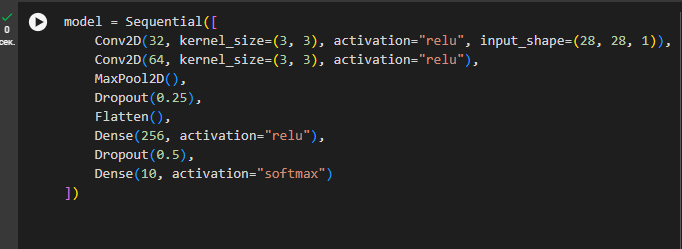
После проделанных этапов создаётся последовательная модель в которой производятся следующие действия:

1. Предоставляются данные вместе с аугментированными
2. Метод Rescaling() нормализует значение пикселей, для того чтобы они находились в диапозоне от 0 до 1.
3. Предоставляется базовая модель
4. Метод GlobalAveragePooling2D это метод, который усредняет признаки по всем пространственным измерениям и создает плоский вектор признаков
5. Метод Dropout случайным образом отключает 30% нейронов, чтобы предотвратить переобучение
6. Dense(128, activation='relu') это полносвязный слой с 128 нейронами и функцией активации
7. После идёт ещё один слой с отключением 50% нейронов
8. Выходной слой с функцией активации softmax, который предсказывает вероятность принадлежности к каждому классу layers.Dense(num\_classes, activation='softmax')

Обучение производилось на 20 эпохах. После обучения проверяю метрики качества – Рисунок 2.3.4.

  
Рисунок 2.3.4- Метрики качества обученной модели

Сравнив первую и вторую модель, я пришёл к выводу, что вторая модель выдаёт более точные ответы. Значения установленных гиперпараметров можно посмотреть на Рисунке 2.3.5.

  
Рисунок 2.3.5 – значения гиперпараметров

# 3 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА ДЛЯ КЛАССИФИКАЦИИ ЦВЕТОВ ПО ИЗОБРАЖЕНИЮ

В процессе разработки модели классификации транспортных средств по изображению ключевым этапом является создание программного продукта с простым и удобным интерфейсом, который обеспечивает простоту взаимодействия пользователей с моделью и соответствует их потребностям. Реализация такого интерфейса не только гарантирует доступность модели для широкого круга пользователей, но также придает ей практичность в повседневном использовании.

Создание программного продукта с удобным интерфейсом открывает новые перспективы для расширения использования модели в повседневной жизни. Пользователи могут применять классификацию транспортных средств не только в рамках профессиональных задач, но и в областях, где автоматизация этого процесса может повысить оперативность и точность принимаемых решений, например, в области городской мобильности или обеспечения автомобильной безопасности.

Простой и удобный интерфейс дает возможность пользователям, не обладающим техническими навыками, легко загружать изображения транспортных средств, получать точные результаты классификации и взаимодействовать с моделью в интуитивно понятном формате, без каких либо трудностей. Это особенно важно в сферах, где классификация транспортных средств играет ключевую роль, таких как транспортное планирование, автомобильная промышленность и городская инфраструктура.

Таким образом, разработка программного продукта с простым и удобным интерфейсом не только обеспечивает практическое применение модели, но и создает предпосылки для успешного внедрения ее в различные сферы, способствуя улучшению технологических решений и расширению доступности машинного обучения для широкой аудитории.

## 3.1 Разработка чат-бота для классификации рукописных цифр по изображению

Так же в процессе курсовой работы был разработан чат-бот в Telegram.

Основными преимуществами которого являются:

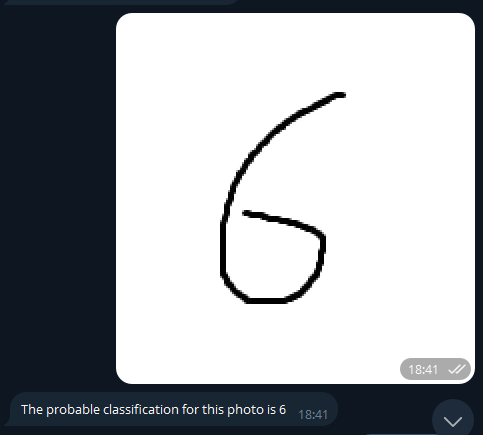
1. Доступность
2. Простота в использовании
3. Быстродействие

Принцип работы чат-бота:

1. Взаимодействие с ним начинается при помощи команды /start – Рисунок 3.2.1.

  
Рисунок 3.2.1 – Приветствие чат-бота

1. После этого сообщения мы выбираем из хранилища устройства нужное нам изображение и отправляем его.
2. После чего модель классифицирует обработанное изображение и выводит результат – Рисунок 3.2.2.

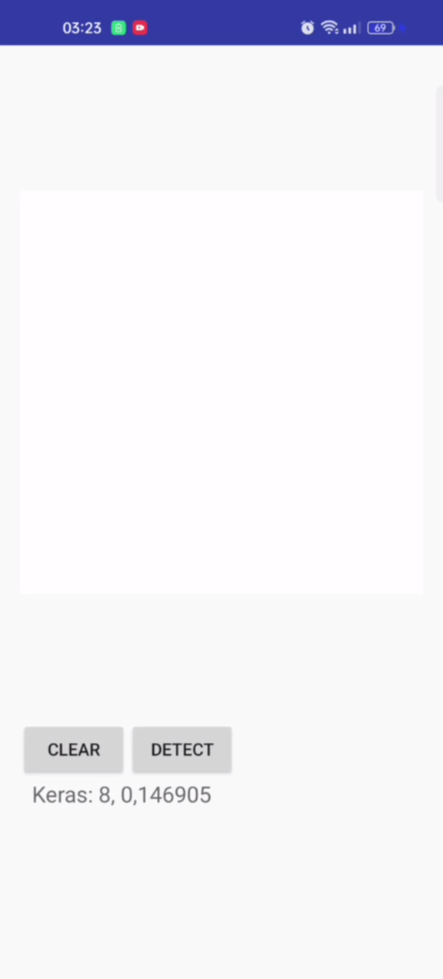
  
Рисунок 3.2.2 – Результат работы

## 3.3 Разработка android-приложения для классификации рукописных цифр по изображению

В ходе выполнения курсовой работы было предпринято решение сделать так же и android приложение для мобильного использования с возможностью использовать модель без доступа к интернету.

Принцип работы android приложения:

1. Пользователь попадает на начальный экран, где ему предлагается нарисовать интересующее его чисто– Рисунок 3.3.1

  
Рисунок 3.3.1 – Начальный экран

1. После выбора изображения, это изображение появляется у него на экране и в низу экрана пишется название класса в виде тоста, после чего пользователь может заново загрузить какое-либо изображение – Рисунок 3.3.2.

  
Рисунок 3.3.2 – результат работы андройд приложения

Но в результате реализации андройд приложения, при сохранении обученной модели в формат tflite происходило сильное падение точности угадывания.

# ЗАКЛЮЧЕНИЕ

Курсовая работа представляет собой комплексное исследование в области инженерии машинного обучения, охватывающее как теоретические аспекты, так и практическую реализацию моделей. В начале исследования были тщательно изучены основные методы, алгоритмы и техники машинного обучения, создавая теоретическую основу для последующего практического применения.

Начальным этапом работы стала разработка нескольких моделей машинного обучения. Первая модель представляет собой сверточную нейронную сеть, построенную с использованием библиотек TensorFlow и Keras. Вторая модель создавалась с использованием предобученной модели, реализованной с применением технологий TensorFlow и Keras. Две модели решали задачу классификации транспортных средств по изображению.

После обучения обеих моделей был проведён анализ метрик качества и отобрана наилучшая модель, ей оказалась вторая, с использованием предобученной моделью.

После чего было разработано несколько сервисов взаимодействия пользователя с обученной моделью. Первым сервисом является чат-бот в Telegram, который предоставляет возможность пользователю загрузить изображение и мгновенно получить результат.. Вторым сервисом является android приложение, которое имеет такой же функционал что и чат-бот, только без использования сети интернета.

В рамках данной курсовой работы не только были углублены теоретические знания в области машинного обучения, но также была закреплены практические навыки, полученные в ходе изучения дисциплины.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Kaggle: платформа для соревнований и обмена данных в области машин

ного обучения. URL: <https://www.kaggle.com/> (дата обращения 18.12.2023);

1. Youtube: серия видео уроков по использованию TensorFlow от команды Google (дата обращения 18.12.2023) ;
2. Официальная документация TensorFlow. URL: [https://www.tensorflow.org/?hl=ru/. (дата](https://www.tensorflow.org/?hl=ru/.%20(дата) обращения 18.12.2023);
3. DataCamp: сеть текстовых туториалов(дата обращения 19.12.2023);
4. NecuRU: цикл видеоуроков по разработке мобильных приложений на языках Java, Kotlin (дата обращения 15.09.2023);
5. Windertor: youtube канал посвященный углубленному изучению программной инженерии(дата обращения 9.07.2022)