

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)
Кафедра экономической математики, информатики и статистики (ЭМИС)

**МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ РАСПОЗНАВАНИЯ
АВТОМОБИЛЬНЫХ МОДЕЛЕЙ И НОМЕРОВ С
ИСПОЛЬЗОВАНИЕМ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ**

ОТЧЕТ

ПО РЕЗУЛЬТАТАМ

производственной практики: преддипломная практика

Обучающийся гр. 599-1

_____ В.С. Ревера

«___» _____ 2023 г.

Руководитель практики от

Университета:

к.ф.-м.н., доцент каф. ЭМИС

_____ Е.А. Шельмина
(оценка)

«___» _____ 2023 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)
Кафедра экономической математики, информатики и статистики (ЭМИС)

УТВЕРЖДАЮ

Зав. кафедрой ЭМИС

И.Г. Боровской

«___» _____ 2023 г.

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

на производственную практику: научно-исследовательскую работу

студенту гр. 599-1 факультета вычислительных систем

Ревера Всеволоду Сергеевичу

1. Тема практики: Мобильное приложение для распознавания автомобильных моделей и номеров с использованием нейросетевых технологий.
2. Цель практики: разработка мобильного приложения, которое сможет автоматически распознавать автомобильные модели и номерные знаки на фотографиях.
3. Сроки прохождения практики: 01.04.2023-13.05.2023.

Совместный рабочий график (план) проведения практики

№ п/п	Перечень заданий	Сроки выполнения
1	Знакомство со структурой предприятия, прохождение инструктажа	06.02.2023
2	Работа с литературными и интернет- источниками, анализ предметной области	07.02.2023-22.02.2023
3	Обзор существующих решений	23.02.2023-24.02.2023
4	Настройка среды разработки и выбор средств реализации	25.02.2023-28.02.2023
5	Сравнительный анализ архитектур нейронных сетей	01.03.2023-14.03.2023
6	Сбор и подготовка данных для обучения модели	15.03.2023-17.03.2023
7	Написание отчета	18.03.2023-26.03.2023

Дата выдачи: «01» апреля 2023 г.

Руководитель практики от университета

к.ф.-м.н., доцент каф. ЭМИС _____ Е.А. Шельмина

Задание принял к исполнению «01» апреля 2023 г.

Студент гр. 599-1 _____ В.С. Ревера

Оглавление

Введение.....	5
1 НЕЙРОННЫЕ СЕТИ	9
1.1 Искусственные нейронные сети.....	9
1.2 Описание принципов работы сверточных нейронных сетей	12
1.3 Проблемы сверточных нейронных сетей	16
1.4 Существующие решения.....	18
1.5 Выбор средств для реализации программы	22
1.6 Сравнительный анализ архитектур нейронных сетей.....	23
1.7 Модель VGG16.....	28
1.8 Технологии компьютерного зрения	32
2 МОБИЛЬНОЕ ПРИЛОЖЕНИЕ	37
2.1 Требования к системе	37
2.2 Описание набора данных	38
2.3 Показатели качества нейронной сети	40
2.4 Процесс предварительной подготовки и обработки данных	42
2.5 Создание модели классификации автомобилей	42
2.6 Создание модели распознавания автомобильных номеров.....	44
2.7 Разработка мобильного приложения	44
Заключение	45
Список использованных источников	48

Введение

Место прохождения практики - Томский государственный университет систем управления и радиоэлектроники (ТУСУР), кафедра экономической математики, информатики и статистики (ЭМИС) [1].

ТУСУР основан в 1962 году и является одним из ведущих вузов Российской Федерации. А в 2020 году вошел в топ-50 вузов в области информационных технологий [2].

Достижения ТУСУР как исследовательского лидера подтверждаются его участием в программе «Приоритет-2030», а также победой в конкурсе на получение специальной части гранта на развитие территориального и отраслевого лидерства в рамках этой программы [3]. Цель программы «Приоритет 2030» – сформировать широкую группу университетов, которые станут лидерами в создании нового научного знания, технологий и разработок для внедрения в российскую экономику и социальную сферу. Программа стратегического развития ТУСУР по «Приоритету-2030» включает в себя следующие проекты: «Микроэлектроника и системы связи нового поколения», «ИТ, безопасная цифровая среда и киберфизические системы», «Науки о космосе и инжиниринг», «Техмед» и «Управленческая и инфраструктурная трансформация».

ТУСУР является также лидером в сфере подготовки квалифицированных кадров для высокотехнологичных отраслей экономики, внедрения инновационных образовательных и исследовательских программ, прикладных разработок новой техники, аппаратуры и систем управления. Выпускники вуза создают технологии и продукты для наукоемких производств в крупнейших российских компаниях, а также становятся предпринимателями и основателями собственных фирм.

Кафедра ЭМИС занимается подготовкой программистов, обладающих углублёнными экономическими знаниями. Кафедра готовит специалистов в

области информационных систем и технологий, информатики и вычислительной техники, занимающихся исследованием, разработкой, внедрением и сопровождением информационных технологий и систем [4].

На кафедре ЭМИС студенты получают знания и компетенции, позволяющие изучать и использовать в работе самые передовые технологии в информационной сфере. Получаемые умения позволяют студентам разрабатывать архитектуры информационных систем, проводить анализ различных технологий, применять знания на практике, разрабатывать программное обеспечение.

Задачами кафедры ЭМИС в области научной деятельности являются формирование научных коллективов и выполнение научно-исследовательских и поисково-экспериментальных исследований, развитие научно-исследовательской деятельности обучающихся, а также подготовка кадров высшей квалификации. Научные исследования на кафедре выполняются по следующим направлениям:

- методы прогнозирования в экономике;
- математическое моделирование физико-химических превращений в материалах;
- искусственный интеллект и принятие решений;
- проблемы теории управления техническими системами;
- теория вероятностей и математическая статистика, и другие.

Выпускники кафедры ЭМИС благодаря полученным знаниям становятся востребованными специалистами в сфере информационных технологий (ИТ).

В последние десятилетия искусственные нейронные сети стали одной из наиболее активно развивающихся областей искусственного интеллекта. Они применяются в различных задачах, включая обработку естественного языка, распознавание речи, машинное зрение и другие.

В настоящее время технологии машинного обучения и искусственного интеллекта находят все большее применение в различных областях. Одной из таких областей являются автомобили. Исследование в этой области может иметь значительный практический интерес во многих сферах, включая автомобильную промышленность, государственные службы, системы безопасности, системы контроля за скоростью, в транспортных компаниях для отслеживания грузов и других отраслях.

В современном мире автомобили являются неотъемлемой частью нашей жизни. Каждый день мы видим множество различных моделей автомобилей на дорогах и парковках. Однако, не всегда у нас есть возможность быстро определить модель автомобиля, которую мы видим на улице. В таких случаях может пригодиться приложение, которое сможет быстро распознать модель автомобиля по фотографии.

Актуальность данной работы состоит в том, что разработка мобильного приложения с использованием нейросетевых технологий для распознавания автомобильных моделей и номеров представляет собой востребованную задачу, которая обладает большим потенциалом для применения в различных сферах, таких как автоматическое фиксирование нарушений ПДД на дорогах, контроль за автопарком компании и многие другие. Кроме того, разработка подобного приложения может значительно упростить процесс идентификации автомобилей в различных сферах жизни, сократив время и усилия, необходимые для этого.

Целью дипломной работы является разработка мобильного приложения для распознавания моделей автомобилей и номеров с использованием нейросетевых технологий. В качестве базы для распознавания моделей автомобилей будет использоваться нейросеть, обученная на наборе данных, содержащем изображения автомобилей различных марок и моделей. Для

распознавания номеров автомобилей будут применены технологии компьютерного зрения.

Объект исследования: разработка мобильного приложения для распознавания автомобильных моделей и номерных знаков с использованием нейросетевых технологий.

Предмет исследования: создание и обучение нейросетевой модели для распознавание автомобильных моделей и номерных знаков с использованием технологий компьютерного зрения, интеграция модели в мобильное приложение.

Для достижения цели необходимо решить следующие задачи:

- провести сравнительный анализ архитектур нейронных сетей методом перекрестной проверки и выбрать на его основе оптимальную модель для обучения;
- на основе выбранной архитектуры разработать и обучить нейросеть распознавать модели автомобилей;
- реализовать алгоритм распознавания автомобильных номеров на основе технологий компьютерного зрения;
- разработать мобильное приложение для Android;
- протестировать приложение и отразить результаты его эффективности.

1 НЕЙРОННЫЕ СЕТИ

1.1 Искусственные нейронные сети

Искусственные нейронные сети (ИНС) [5] – это компьютерные алгоритмы, которые используются для обработки данных, в основном в области машинного обучения и компьютерного зрения. Они состоят из множества взаимосвязанных нейронов, которые работают вместе, чтобы выполнять определенную задачу. Нейронные сети могут быть разных типов, по-разному обрабатывать входную информацию и в процессе изменения своего состояния во времени формировать совокупность выходных сигналов.

Нейронные сети возникли из исследований в области искусственного интеллекта, а именно из попыток воспроизвести способность биологических нервных систем обучаться и исправлять ошибки, моделируя низкоуровневую структуру мозга.

Структура нейронных сетей

Хотя один нейрон и способен выполнять простейшие процедуры распознавания, сила нейронных вычислений проистекает от соединений нейронов в сетях. Простейшая сеть состоит из группы нейронов, образующих слой, как показано в правой части рисунка 1.4.

Отметим, что вершины - круги слева, служат лишь для распределения входных сигналов. Они не выполняют каких-либо вычислений и поэтому не будут считаться слоем. По этой причине они обозначены кругами, чтобы отличать их от вычисляющих нейронов, обозначенных квадратами. Каждый элемент из множества входов X отдельным весом соединен с каждым искусственным нейроном. А каждый нейрон выдает взвешенную сумму входов в сеть. В искусственных и биологических сетях многие соединения могут отсутствовать, все соединения показаны в целях общности. Могут иметь место также соединения между выходами и входами элементов в слое. Удобно

считать веса элементами матрицы W . Матрица имеет n строк и m столбцов, где n – число входов, а m – число нейронов. Например, w_{23} – это вес, связывающий второй вход с третьим нейроном. Таким образом, вычисление выходного вектора Y , компонентами которого являются выходы y_i нейронов, сводится к матричному умножению $Y = XW$.

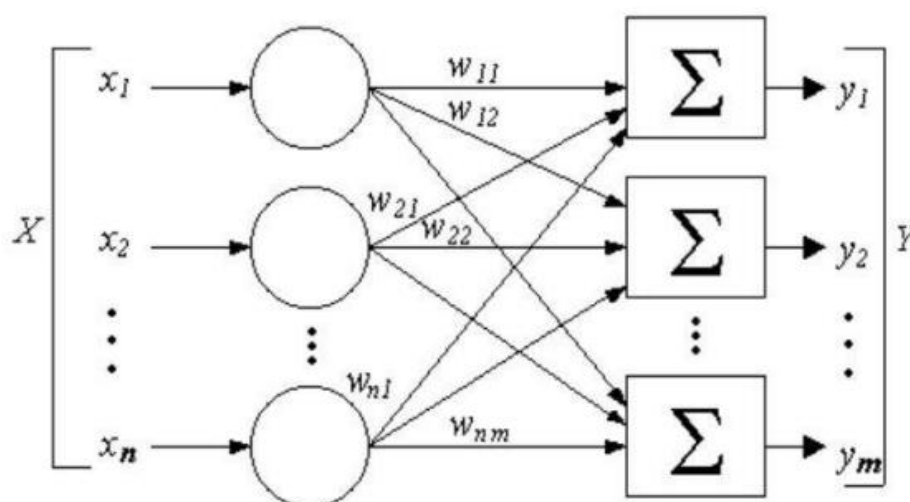


Рисунок 1.4 – Простейшая однослойная сеть

Более крупные и сложные нейронные сети обладают, как правило, и большими вычислительными возможностями. Хотя созданы сети всех конфигураций, какие только можно себе представить, послойная организация нейронов копирует слоистые структуры определенных отделов мозга. Оказалось, что такие многослойные сети обладают большими возможностями, чем однослойные, и в последние годы были разработаны многообразные алгоритмы для их обучения.

Многослойные сети могут образовываться каскадами слоев [8]. Выход одного слоя является входом для последующего слоя. Подобная сеть показана на рисунке 1.5.

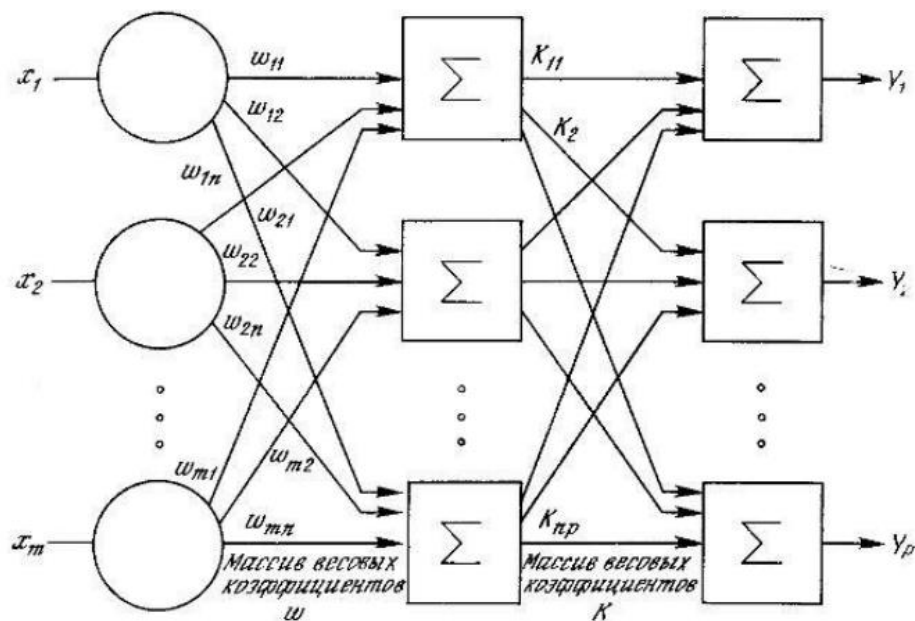


Рисунок 1.5 – Многослойная нейронная сеть

Многослойные сети не могут привести к увеличению вычислительной мощности по сравнению с однослойной сетью лишь в том случае, если активационная функция между слоями не будет линейной. Вычисление выхода слоя заключается в умножении входного вектора на первую весовую матрицу с последующим умножением (если отсутствует нелинейная активационная функция) результирующего вектора на вторую весовую матрицу. Так как умножение матриц ассоциативно, то двухслойная линейная сеть эквивалентна одному слою с весовой матрицей, равной произведению двух весовых матриц. Следовательно, любая многослойная линейная сеть может быть заменена эквивалентной однослойной сетью.

В настоящее время существует много типов нейронных сетей [9], которые включают в себя:

- полносвязные нейронные сети (Fully Connected Neural Networks - FCNN);
- рекуррентные нейронные сети (Recurrent Neural Networks - RNN);
- глубокие нейронные сети (Deep Neural Networks - DNN);
- сверточные нейронные сети (Convolutional Neural Networks - CNN).

Полносвязные нейронные сети используются для обработки данных без структуры, например текста, а рекуррентные нейронные сети используются для обработки последовательных данных, таких как речь и текст. Глубокие нейронные сети состоят из большого количества слоев и используются для решения более сложных задач, чем однослойные сети.

Однако, для задач классификации изображений, сверточные нейронные сети являются наиболее эффективными. Они обучаются выделять важные признаки изображения, позволяя точнее определять класс объекта на изображении. Также они способны распознавать образы в изображении, учитывая их форму и контекст. Кроме того, сверточные нейронные сети имеют возможность автоматически извлекать признаки изображения.

Таким образом, стоит подчеркнуть, что для распознавания моделей автомобилей на изображениях, необходимо использовать именно сверточные нейронные сети. Они способны эффективно обрабатывать и классифицировать изображения, извлекать важные признаки и обучаться на большом количестве данных.

1.2 Описание принципов работы сверточных нейронных сетей

Использование классических нейронных сетей для работы с изображениями затруднено, как правило, большой размерностью вектора входных значений нейронной сети, большим количеством нейронов в промежуточных слоях и, как следствие, большими затратами вычислительных ресурсов на обучение и вычисление сети. Сверточным нейронным сетям в меньшей степени присущи описанные выше недостатки.

Сверточные нейронные сети [10] - это тип нейронных сетей, которые используются для обработки изображений и видео. Они состоят из нескольких слоев, каждый из которых выполняет определенную функцию. Слой свертки обрабатывает изображение с использованием ядра свертки, которое

перемещается по всему изображению, вычисляя локальные свертки в каждой позиции. Затем следует слой подвыборки, который уменьшает размер изображения, удаляя избыточные данные и сохраняя только наиболее важные особенности. В конце, плотный слой используется для классификации изображений.

Таким образом, идея сверточных нейронных сетей заключается в чередовании сверточных слоев и субдискретизирующих слоев (слоев подвыборки). Структура сети – однонаправленная (без обратных связей), многослойная. Для обучения используются стандартные методы, чаще всего метод обратного распространения ошибки. Функция активации нейронов (передаточная функция) – любая, по выбору исследователя.

Архитектура сверточной нейронной сети представлена на рисунке 1.6.

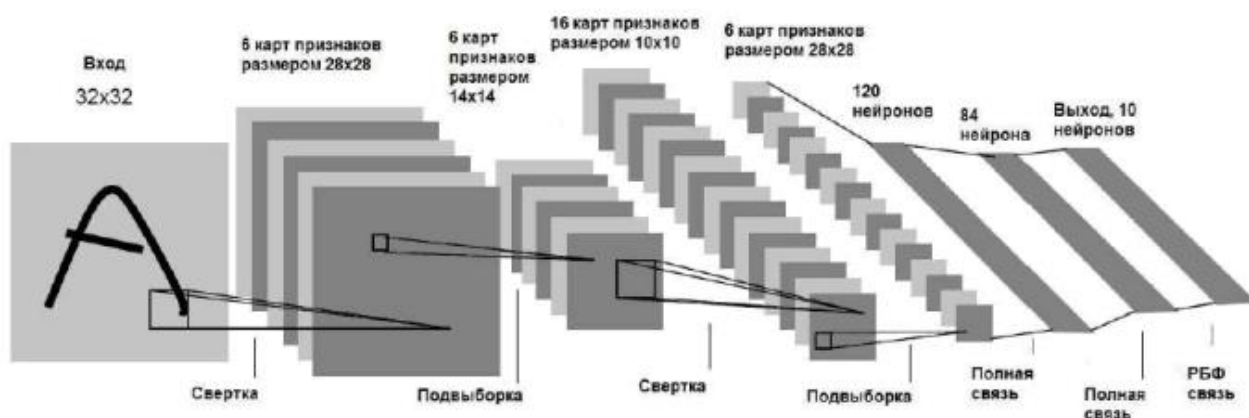


Рисунок 1.6 – Структура сверточной нейронной сети

Сверточный слой состоит из так называемых сверточных ядер [11]. Рассмотрим, что происходит на свёрточном слое на примере чёрно-белого изображения. На вход подается двумерная матрица значений размерности (n,m) . На свёрточном слое также есть рецептивное поле. Рецептивное поле можно представить как рамку размерности (k,r) , которая скользит по матрице с некоторым шагом. Сверточное ядро представляет из себя матрицу значений той же размерности. На каждом этапе выполняется операция свертки: соответствующие элементы матрицы, попавшие в рецептивное поле,

умножаются на соответствующие элементы матрицы сверточного ядра. Далее полученные числа складываются и к полученной сумме применяется функция активации. Итоговое число будет в выходной матрице, которую иногда называют картой признаков.

Иллюстрацию данного процесса можно увидеть на рисунке 1.7.

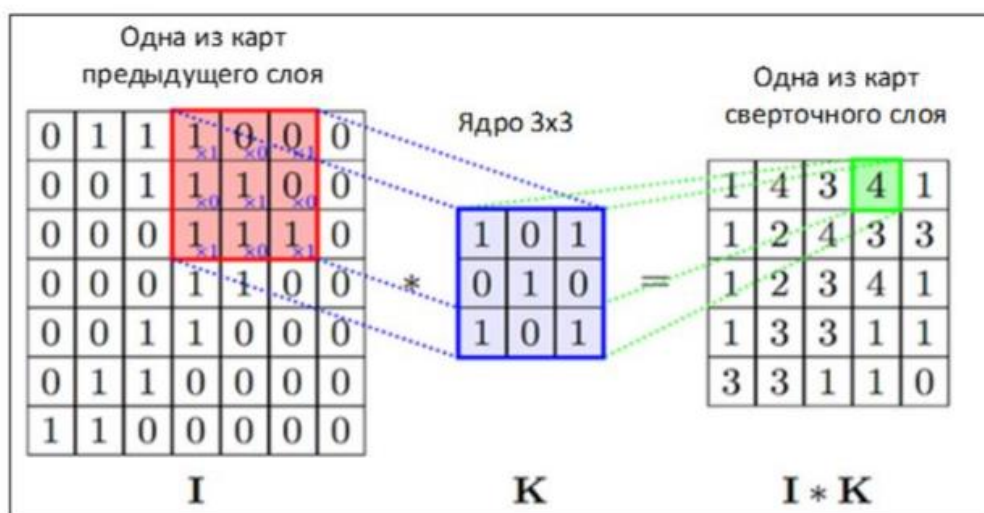


Рисунок 1.7 – Операция свертки

Редуцирующий слой необходим для уменьшения размерности карт признаков предыдущего слоя [12]. Карта признаков, поступающая на вход редуцирующего слоя, фильтруется. Фильтр чаще всего имеет размерность 2x2. Фильтрация происходит следующим образом: карта признаков делится на ячейки 2x2, и выполняется одна из операций. Самые частые – выбор максимального элемента (англ. max pooling) и усреднение (англ. average pooling). В первом случае выходные карты признаков слоя будут состоять из максимальных значений ячеек 2x2, а во втором – из средних арифметических.

Визуализацию данного процесса можно увидеть на рисунке 1.8.

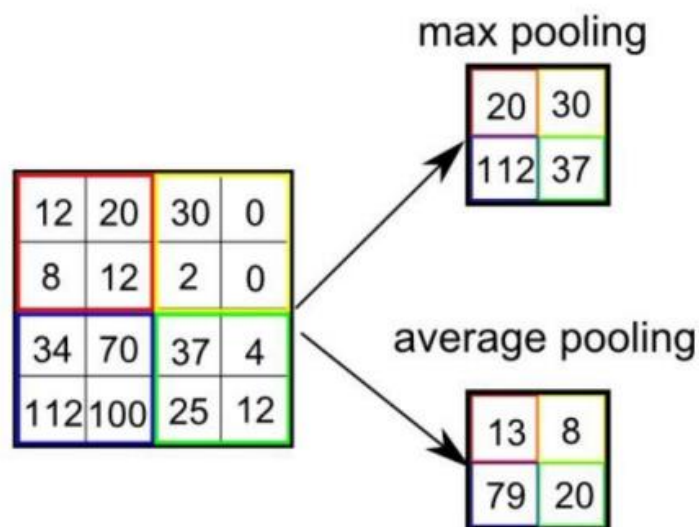


Рисунок 1.8 – Объединение по максимальному и среднему значениям

Ключевым моментом в понимании сверточных нейронных сетей является понятие т.н. «разделяемых» весов, т.е. часть нейронов некоторого рассматриваемого слоя нейронной сети может использовать одни и те же весовые коэффициенты. Нейроны, использующие одни и те же веса, объединяются в карты признаков, а каждый нейрон карты признаков связан с частью нейронов предыдущего слоя. При вычислении сети получается, что каждый нейрон выполняет свертку (операцию конволюции) некоторой области предыдущего слоя (определяемой множеством нейронов, связанных с данным нейроном).

Слои нейронной сети, построенные описанным образом, называются сверточными слоями. Помимо, сверточных слоев в сверточной нейронной сети могут быть слои субдискретизации (выполняющие функции уменьшения размерности пространства карт признаков) и полносвязные слои (выходной слой, как правило, всегда полносвязный). Все три вида слоев могут чередоваться в произвольном порядке, что позволяет составлять карты признаков из карт признаков, а это на практике означает способность распознавания сложных иерархий признаков.

Таким образом, сверточные нейронные сети состоят из слоев, редуцирующих слоев и полносвязных слоев, которые выполняют определенные функции обработки и уменьшения размерности пространства карт признаков. Это позволяет нейронной сети распознавать сложные иерархии признаков на изображениях. В целом, сверточные нейронные сети позволяют уменьшить затраты вычислительных ресурсов на обучение и вычисление сети при работе с изображениями.

1.3 Проблемы сверточных нейронных сетей

Сверточные нейронные сети — это класс нейронных сетей, которые обладают способностью распознавать образы на изображениях. Они являются одним из наиболее мощных инструментов в области обработки изображений и компьютерного зрения, и на сегодняшний день являются неотъемлемой частью многих приложений, таких как автоматическое распознавание лиц, автоматическая классификация изображений и диагностика заболеваний по медицинским изображениям.

Однако, сверточные нейронные сети также имеют ряд проблем и ограничений [13]. В данной главе мы рассмотрим некоторые из этих проблем.

Первая проблема связана с необходимостью большого количества данных для обучения. Сверточные нейронные сети имеют очень большое количество параметров, которые должны быть настроены в процессе обучения. Это означает, что для обучения такой сети требуется огромное количество размеченных изображений. В противном случае, сверточная нейронная сеть может переобучиться на имеющихся данных, что может привести к плохой обобщающей способности.

Вторая проблема связана с обработкой и распознаванием объектов разных размеров на изображении. Сверточные нейронные сети работают с

фиксированным размером изображения, и поэтому не могут эффективно обрабатывать изображения с объектами разных размеров.

Третья проблема связана с интерпретируемостью. Сверточные нейронные сети являются очень сложными и нелинейными моделями, и их внутреннее представление может быть очень сложным для понимания и интерпретации.

Для минимизации проблем, связанных с обучением сверточных нейронных сетей, можно использовать следующие методы:

1. Предобработка данных: важно подготовить данные перед обучением сети, например, выполняя их нормализацию и стандартизацию.
2. Добавление большего количества данных: чем больше данных, тем точнее будет обученная нейросеть.
3. Использование аугментации данных: это метод, который позволяет генерировать новые изображения на основе существующих, изменяя их размеры, поворачивая их и т.д. Это позволяет увеличить количество данных, что может помочь избежать переобучения модели.
4. Использование регуляризации: это метод, который позволяет ограничивать веса нейронов в сети, что также может помочь избежать переобучения.
5. Использование архитектур сетей, которые были успешно применены в аналогичных задачах: это позволяет использовать уже проверенные и оптимизированные сети, что может уменьшить количество ошибок в обучении и ускорить процесс разработки.

Таким образом, в этой главе рассмотрены проблемы, возникающие при разработке сверточных нейронных сетей, а также возможные способы их решения.

1.4 Существующие решения

Конечной целью данной дипломной работы является разработка мобильного приложения для распознавания автомобильных моделей и номеров с использованием нейросетевых технологий. Для достижения этой цели необходимо проанализировать существующие аналоги и определить их преимущества и недостатки.

Существует несколько аналогов приложений для распознавания автомобильных номеров и моделей. Некоторые из них являются коммерческими продуктами, в то время как другие доступны для бесплатного использования.

Ниже приведены наиболее популярных из них.

Automatic Number Plate Recognition (ANPR) [14] - это технология распознавания номерных знаков, которая используется в американских полицейских автомобилях и камерах на дорогах. Эта технология позволяет автоматически распознавать номера автомобилей и использовать эту информацию для обеспечения безопасности на дорогах.

На рисунке 1.9 представлен скриншот интерфейса работы ANPR.

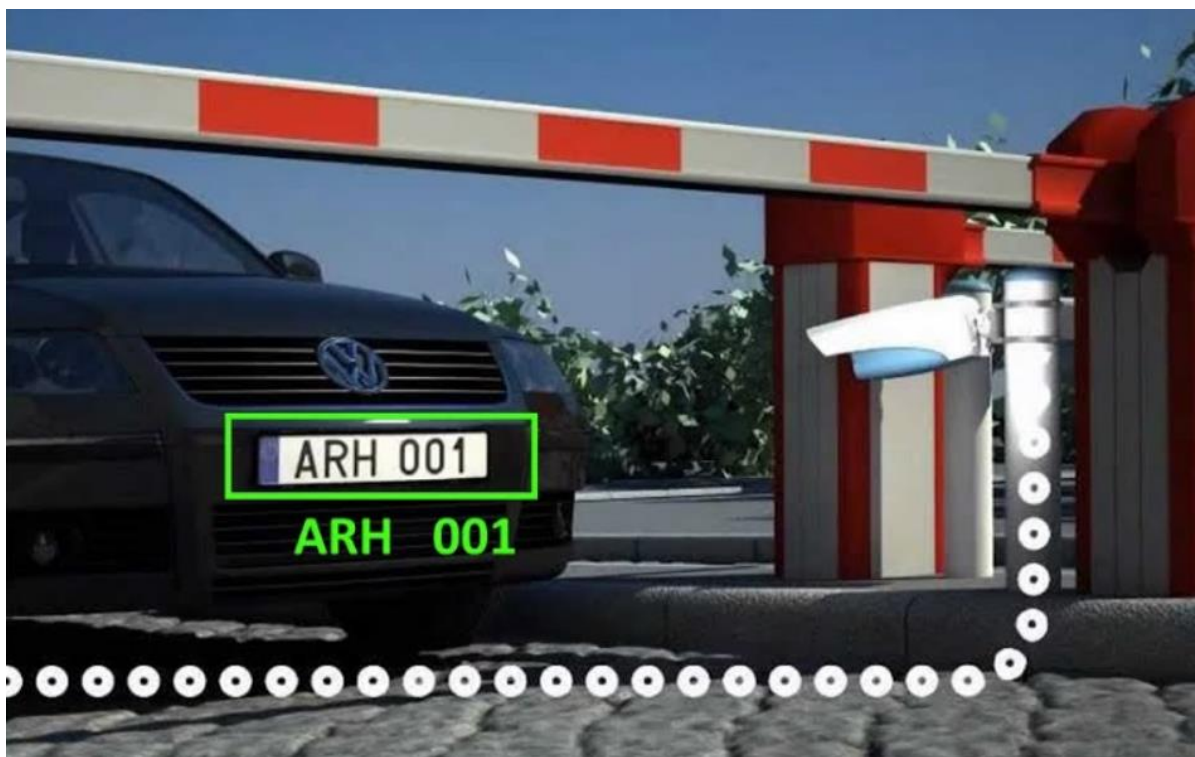


Рисунок 1.9 - Скриншот интерфейса работы ANPR

Хотя технология распознавания номерных знаков ANPR представляет собой важный инструмент для обеспечения безопасности на дорогах, она не может предоставить полную информацию о транспортных средствах. Например, ANPR не может распознать модель автомобиля и другую дополнительную информацию, которая может быть полезна для пользователей.

Мобильный Авто.ру [15] — это приложение от крупнейшего автомобильного портала в России, созданное, чтобы сделать поиск и продажу автомобиля комфортными и безопасными. Приложение имеет огромную базу объявлений о продаже автомобилей, которая позволяет пользователям быстро найти и купить автомобиль, соответствующий их потребностям и бюджету. Огромный функционал приложения включает в себя возможность просмотра технических характеристик автомобилей и отзывов пользователей, а также поиск транспортных средств по марке, модели, году выпуска, типу топлива, пробегу и многим другим параметрам.

На рисунке 1.10 представлен скриншот интерфейса работы Авто.ру.

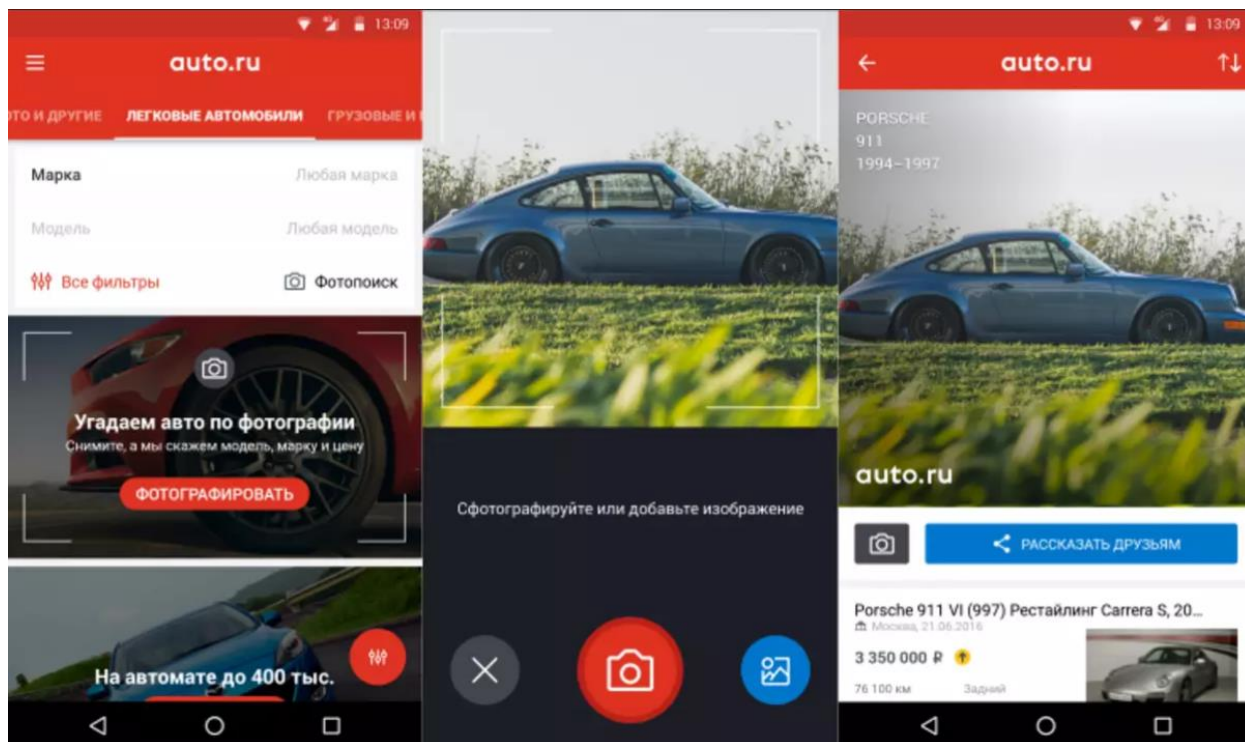


Рисунок 1.10 - Скриншот интерфейса работы Авто.ру

Одной из главных особенностей приложения Авто.ру является функция автоматического определения модели автомобиля по фотографии и предоставление пользователю объявлений с ценами на данную марку. Однако, оно не предоставляет информацию о номерах автомобилей.

Умная камера [16] — это камера в мобильном приложении Яндекс. Она распознаёт предметы, рассказывает, что видит, и советует, где купить. А ещё — переводит надписи и сканирует документы.

На рисунке 1.11 представлен скриншот интерфейса работы Умной камеры.

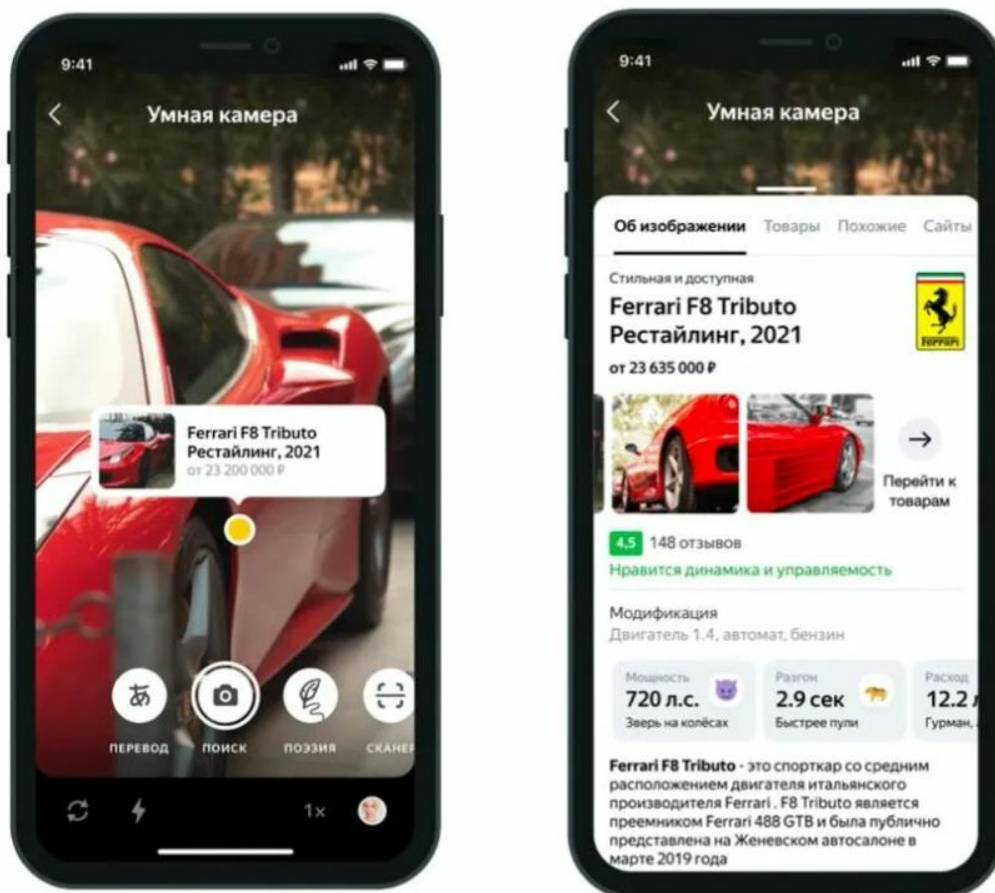


Рисунок 1.11 -Скриншот интерфейса работы Умной камеры

Одним из главных преимуществ Яндекс Камеры является возможность использования искусственного интеллекта для классификации объектов на экране в режиме реального времени. В том числе, приложение способно распознавать марку, модель и технические характеристики автомобиля, включая его мощность, скорость и расход топлива. При этом технические характеристики снабжены коротким и ёмким пояснением, например, «разгоняется, как спринтер» или «проста в управлении». Тем не менее, как и Авто.ру, умная камера не способна предоставлять информацию о номерах автомобилей.

Таким образом, можно сделать вывод, что на данный момент на рынке существуют приложения, которые предоставляют информацию о номерах автомобилей или моделях, но ни одно из них не объединяет эти два функционала в одном приложении. Поэтому разработка мобильного

приложения для распознавания автомобильных моделей и номеров является актуальной и востребованной задачей на сегодняшний день.

1.5 Выбор средств для реализации программы

Выбор средств реализации для создания приложения - важный шаг в разработке проекта. Для достижения целей дипломной работы мы будем использовать следующие средства реализации:

1. Язык программирования Python [17] - это популярный язык программирования, который широко используется в машинном обучении и разработке нейросетей. Python имеет обширную библиотеку инструментов для работы с нейросетями.

2. Фреймворк TensorFlow [18] - это открытая платформа для создания и обучения нейросетей. TensorFlow предоставляет широкий набор инструментов для разработки глубоких нейронных сетей, включая возможность создания сверточных нейронных сетей.

3. Фреймворк Keras [19] - это высокоуровневый интерфейс для работы с нейросетями, созданный для упрощения и ускорения процесса разработки глубоких нейронных сетей. Keras обеспечивает быстрое создание прототипов моделей и легко настраивается для выполнения распределенного обучения.

4. Библиотека OpenCV [20] - это библиотека компьютерного зрения с открытым исходным кодом, которая широко используется для обработки изображений и видео. OpenCV предоставляет множество инструментов для работы с изображениями, таких как сегментация изображений, детектирование объектов и многое другое.

5. Kivy [21] - это открытый фреймворк для создания кроссплатформенных мобильных приложений на языке Python. Он обладает широким набором инструментов для разработки пользовательского

интерфейса и поддерживает использование нейросетей с помощью библиотеки TensorFlow Lite.

Выбор данных средств реализации обусловлен тем, что они имеют обширные возможности и могут быть успешно использованы для решения задачи распознавания автомобильных моделей и номеров с помощью технологий глубокого обучения. Язык Python, а также фреймворки TensorFlow и Keras являются основными инструментами для разработки нейронных сетей, а OpenCV обеспечивает обработку изображений, которые будут использоваться в проекте. Kivy позволяет создавать качественные пользовательские интерфейсы и приложения на Python для операционной системы Android.

Данные средства реализации выбраны из-за их высокой популярности, обширных возможностей и поддержки сообществом разработчиков. Они обеспечивают быстрое создание прототипов и настройку моделей нейронных сетей, а также обработку изображений и видео, необходимых для реализации функционала приложения. Также эти средства реализации позволяют решать задачу на уровне высокой точности и скорости, что является важным при разработке приложений с использованием нейронных сетей.

1.6 Сравнительный анализ архитектур нейронных сетей

В данной главе рассмотрим несколько известных архитектур сверточных нейронных сетей, которые используются для задач классификации изображений. Будут проанализированы их особенности, а также сравнительно оценена точность работы на различных наборах данных.

VGG16, ResNet50 и Inception3 [22] - это три известные архитектуры сверточных нейронных сетей, которые широко применяются в задачах компьютерного зрения, включая распознавание объектов на изображении.

VGG16 (Visual Geometry Group 16) [23] - одна из самых известных сверточных нейронных сетей для классификации изображений, разработанная исследователями из Оксфордского университета в 2014 году. Архитектура VGG16 состоит из 16 слоев, включая 13 сверточных и 3 полносвязных слоя. Она имеет фиксированный размер входного изображения 224x224 пикселя и может классифицировать изображения на 1000 классов, определяемых набором данных ImageNet.

ResNet50 (Residual Network 50) [24] - еще одна популярная архитектура сверточной нейронной сети, которая была разработана в 2015 году и получила название благодаря использованию блоков со связями остатков (residual connections) в своей архитектуре. Сеть состоит из 50 слоев, включая сверточные, пулинговые и полносвязные слои. Она также использует набор данных ImageNet для классификации изображений в более чем 1000 различных классов.

Inception3 [25] - это сверточная нейронная сеть, которая разработана для более эффективного использования ресурсов и улучшения точности классификации изображений. Основным принципом Inception3 является использование различных размеров фильтров и уменьшение размерности передаваемых данных внутри сети. Это позволяет уменьшить количество параметров в сети и снизить ее вычислительную сложность.

Для выявления наиболее эффективной архитектуры нейросети проведено исследование, включающее в себя обучение каждой из моделей методом перекрестной развертки на автомобильном наборе данных.

Перекрестная развертка является методом оценки производительности моделей машинного обучения путем разделения данных на несколько подмножеств, называемых фолдами. В данном исследовании использовалась 5-фолдовая кросс-валидация, что означает, что данные были разделены на 5 равных частей. На каждой итерации обучения, один из фолдов использовался

в качестве тестового набора, а остальные четыре - в качестве обучающего набора. Процесс повторялся 5 раз, так что каждый фолд использовался в качестве тестового набора единожды.

В таблице 1.1 представлены входные данные, которые использовались для тестирования архитектур.

Таблица 1.1 – Входные данные

	VGG16	ResNet50	Inception3
Кол-во изображений	8000	8000	8000
Кол-во Fold	5	5	5
Кол-во изображений train_fold	6400	6400	6400
Кол-во изображений test_fold	1600	1600	1600
Обучено моделей	5	5	5
Кол-во эпох	30	30	30
Кол-во классов	164	164	164

Анализ производился с использованием кросс-валидации на 30 эпохах. Для каждой архитектуры были рассчитаны средние значения и стандартные отклонения метрик точности и потерь на каждой итерации кросс-валидации. Затем были сравнены результаты для каждой архитектуры и выделена наилучшая средняя точность для каждой модели на тестовых наборах.

В таблице 1.2 представлены результаты тестирования моделей методом перекрестной развертки.

Таблица 1.2 – Результаты тестирования моделей методом перекрестной развертки

Cross Validation						VGG16	ResNet50	Inception3
Модель 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	83%	85%	72%
Модель 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	80%	82%	77%
Модель 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	86%	86%	74%
Модель 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	86%	85%	79%
Модель 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	84%	84%	73%
Средняя точность						84%	84%	75%

В задаче распознавания моделей автомобилей результаты тестовой выборки являются более предпочтительными, поскольку именно по этому принципу в дальнейшем происходит классификация пользовательских изображений. Из этого следует, что модели ResNet50 и VGG16 показали себя достаточно стабильными и эффективными.

На основе анализа архитектур сформулированы критерии выбора для выявления наиболее подходящей модели в задаче классификации. Каждой из рассмотренных архитектур дана характеристика по заданным критериям.

Результат сравнения архитектур нейросетей представлен в таблице 1.3.

Таблица 1.3 – Сравнение характеристик моделей нейросетей

Критерий выбора	VGG16	ResNet50	Inception3
Точность и качество модели	Высокая точность датасетах любых размеров	Высокая точность при обучении, но средняя на тестах	Маленькая точность на средних датасетах, но может быть менее точной на крупных датасетах
Скорость обучения	Медленная скорость обучения	Быстрая скорость обучения	Быстрая скорость обучения

Сложность и глубина модели	Относительно простая модель с небольшим количеством параметров	Очень глубокая модель с большим количеством параметров	Сложная модель с несколькими входами и выходами
Наличие предобучения	Доступны предобученные веса на большом наборе данных ImageNet	Доступны предобученные веса на большом наборе данных ImageNet	Доступны предобученные веса на большом наборе данных ImageNet
Доступность библиотек и инструментов	Широко используется и легко интегрируется с различными библиотеками	Широко используется и легко интегрируется с различными библиотеками	Широко используется и легко интегрируется с различными библиотеками
Ресурсы, необходимые для работы модели	Требуется производительный GPU для эффективной работы	Требуется производительный GPU для эффективной работы	Требуется очень производительный GPU для эффективной работы

В результате сравнительного анализа архитектур сверточных нейронных сетей VGG16, ResNet50 и Inception3 можно сказать о том, что каждая из этих архитектур имеет свои особенности и предназначена для решения конкретных задач. При выборе оптимальной архитектуры для решения задачи классификации автомобилей необходимо учитывать ресурсы вычислительного оборудования, объем и качество имеющихся данных, а также требования к точности классификации.

Рассмотренные в данной главе архитектуры обладают высокой точностью при классификации изображений, включая автомобили. Тем не менее, важным при выборе конкретной архитектуры остается учет различных факторов, таких как размер модели, скорость обучения, сложность реализации и использования, а также доступность предобученных весов.

Исходя из вышеизложенного, можно сделать вывод, что для задачи распознавания моделей автомобилей на изображениях VGG16 является наиболее подходящей архитектурой из трех рассмотренных. Это связано с тем, что VGG16 может достаточно точно извлекать высокоуровневые признаки изображений, а также показывает более стабильные результаты обучения на фоне других архитектур и имеет правильную тенденцию достижения эффективности при обучении, что важно для распознавания автомобильных моделей.

1.7 Модель VGG16

VGG16 стала прорывом в области компьютерного зрения и сверточных нейронных сетей в 2014 году, так как ее высокая точность и производительность превзошли многие предыдущие модели. Она может быть использована для различных задач, таких как классификация изображений, распознавание объектов и детектирование. В настоящее время VGG16 и ее модификации продолжают использоваться во многих приложениях компьютерного зрения и машинного обучения.

Основная задача, которую выполняет VGG16, и на которую она была обучена, это классификация объектов на изображении, и чтобы она могла их классифицировать, нейронная сеть должна понимать само изображение. Это включает в себя использование необработанного изображения в качестве входных пикселей и построения внутреннего представления посредством преобразований, которые превращают необработанные пиксели изображения в сложные к пониманию функции [26]. То есть где-то между моментом, когда подаётся исходное изображение и выводится результат классификации, существует модель сложных признаков, называемая промежуточными слоями, используя которые мы можем описать содержание и стиль входного изображения.

Модель VGG16 представлена на рисунке 1.15.

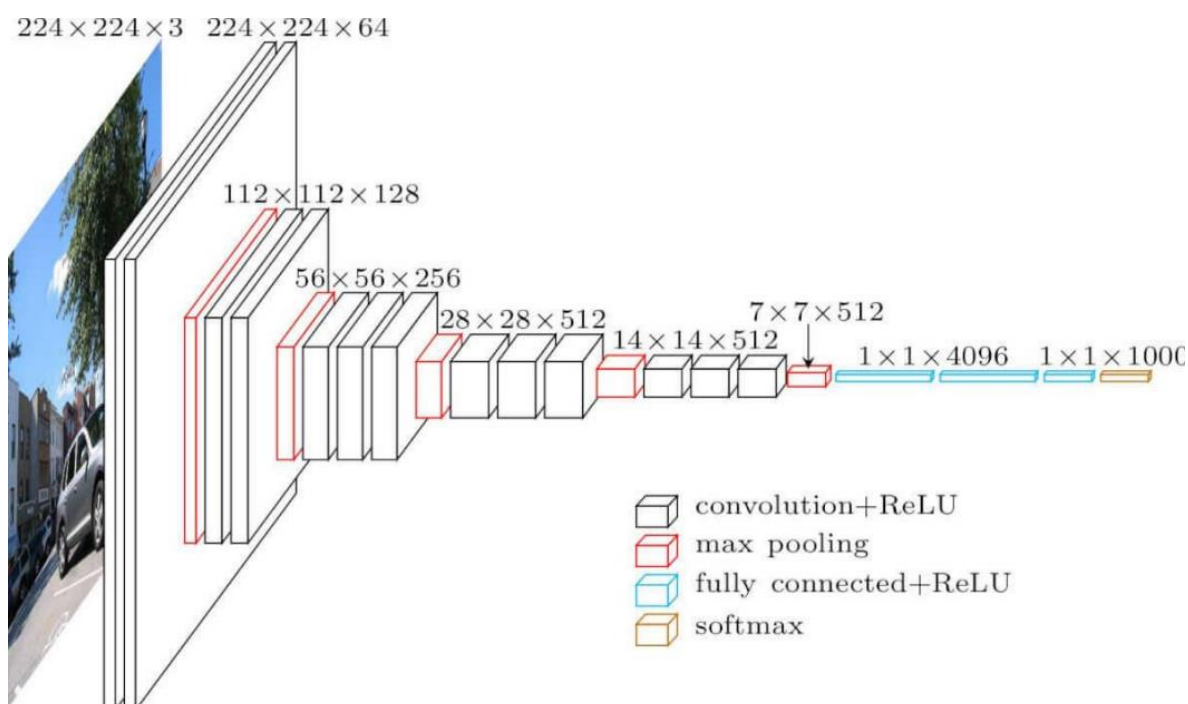


Рисунок 1.15 – Модель VGG16

Изначально VGG16 была предложена экспертами из Оксфордского университета в статье “Very Deep Convolutional Networks for Large-Scale Image Recognition”. Модель достигает точности 92.7% – топ-5, при тестировании на ImageNet в задаче распознавания объектов на изображении. Этот набор данных состоит из более чем 14 миллионов изображений [27].

VGG16 – одна из самых знаменитых моделей, отправленных на соревнование ILSVRC-2014. Она является улучшенной версией AlexNet, в которой заменены большие фильтры (размера 11 и 5 в первом и втором сверточном слое, соответственно) на несколько фильтров размера 3×3 , следующих один за другим. Сеть VGG16 обучалась на протяжении нескольких недель при использовании видеокарт NVIDIA TITAN BLACK [28].

Архитектура VGG16 представлена на рисунке 1.16.

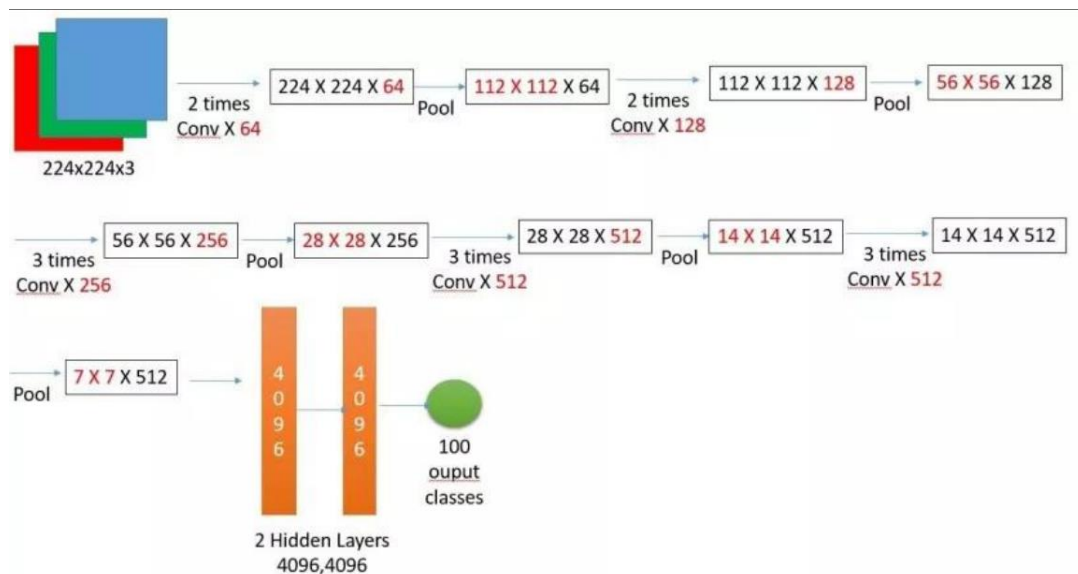


Рисунок 1.16 – Архитектура VGG16

Можно заметить, что размер входных данных нейронной сети VGG16 - 224x224x3, где 3 - это количество каналов (RGB). Размер выходных данных - 1000, что соответствует количеству классов в наборе данных ImageNet. Эти данные проходят через стек сверточных слоёв, которые реагируют на те или иные качества и характеристики изображения, с фильтрами, размер рецептивного поля которых равен 3x3 – max-pooling слои, которые отвечают за процесс простой дискретизации изображения на основе выборки. Этот процесс состоит в том, чтобы выбрать максимальное число из подрегионов изображения, отсюда и название max. Использование сверточных слоев с небольшим размером фильтра (3x3) и большим количеством фильтров (от 64 до 512) позволяет нейронной сети VGG16 извлекать более высокоуровневые признаки из изображений и достигать высокой точности классификации.

Пример данного процесса представлен на рисунке 1.17.

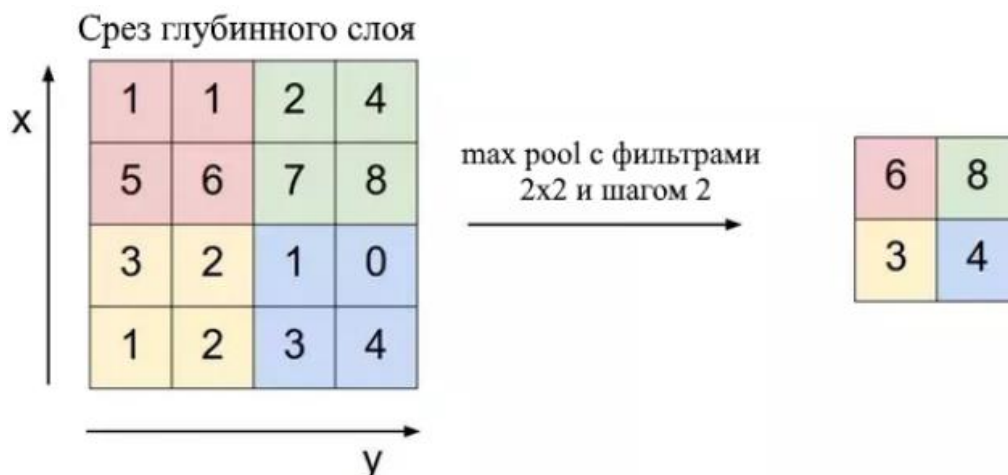


Рисунок 1.17 – Процесс свертки VGG16

После стека сверточных слоёв идут 3 полносвязных слоя [29]: первые два имеют по 4096 нейронов, третий – 1000 нейронов. Данные слои отвечают за классификацию, а третий (soft-max) слой как раз является результирующим слоем, нейроны которого и показывают отношение объекта на изображении к определенному классу. Стоит отметить, что данный слой зачастую применяется в машинном обучении, как раз для решения задач классификации, когда возможных классов больше двух.

Выбрана сверточная нейронная сеть VGG16, потому она имеет следующие преимущества:

- данная сеть уже обучена и реализована;
- высокая точность классификации, благодаря использованию большого количества сверточных слоев, что говорит о эффективности данной модели и способности выделять особенности изображения, не обращая внимания на шумы;
- хорошо исследованная и относительно простая в реализации модель по сравнению с более современными сверточными нейронными сетями, такими как ResNet или Inception;
- популярна и широко используется в различных задачах машинного обучение, поэтому отлично подходит и для обучающих целей.

Также сеть VGG16 имеет 3 серьезных недостатка:

- архитектура сети весит довольно много из-за глубины и количества полносвязных узлов;
- требует большого количества вычислительных ресурсов и времени для обучения;
- существует проблема переобучения, когда нейронная сеть начинает выделять слишком сложные и специфичные для тренировочных данных признаки, что приводит к плохой обобщающей способности на новых данных.

Несмотря на то, что у VGG16, как и у любой другой нейросети, есть свои недостатки, она остается одной из лучших моделей для решения задач классификации изображений на сегодняшний день. Ее высокая точность и способность извлекать высокоуровневые признаки из изображений делают ее популярным выбором для различных приложений в области компьютерного зрения и глубокого обучения.

Таким образом, в этом разделе была рассмотрена модель VGG16, ее архитектура, особенности, принцип работы. Были выявлены как положительные, так и отрицательные стороны данной модели. В целом, VGG16 остается одной из лучших моделей для решения задач классификации и хорошо подходит для реализации алгоритма распознавания моделей автомобилей на изображении.

1.8 Технологии компьютерного зрения

Компьютерное зрение (Computer Vision) [30] - это область искусственного интеллекта, которая занимается разработкой методов и алгоритмов для автоматического анализа и обработки изображений. В данной главе будет рассмотрено применение компьютерного зрения в задаче распознавания автомобильных номеров.

Стоит отметить, что библиотека OpenCV является одной из самых распространенных и популярных библиотек для работы с компьютерным зрением. Она предоставляет широкий спектр инструментов для работы с изображениями и видео, а также алгоритмы для обнаружения объектов, распознавания лиц и текста на изображениях.

Для распознавания символов на изображении номерного знака можно использовать архитектуру сверточной нейронной сети. Эта архитектура позволяет обучить нейронную сеть на изображениях символов и добиваться высокой точности распознавания. Этот подход является частью технологии оптического распознавания текста.

OCR (Optical Character Recognition) [31] – это процесс распознавания текста на изображении или сканированном документе. С помощью OCR можно извлекать текст из фотографий, сканов, электронных документов и других источников.

Данный подход широко используется в мобильных приложениях для распознавания текста на изображениях. Например, в приложениях для сканирования документов, распознавания номеров телефонов и адресов, распознавания номеров автомобилей и т.д.

Технология OCR основывается на анализе изображения и выделении на нем текста. Для этого применяются различные алгоритмы и методы компьютерного зрения, такие как бинаризация, фильтрация, сегментация, распознавание и т.д.

Процесс распознавания текста включает несколько этапов [32]:

1. Предобработка изображения: удаление шумов и артефактов, повышение контрастности и яркости.
2. Бинаризация изображения: преобразование изображения в черно-белый формат, где текстовые элементы – черные, а фон – белый.

3. Сегментация изображения: разделение изображения на отдельные элементы (символы, буквы, цифры).

4. Распознавание символов: определение, какие символы присутствуют на изображении.

5. Постобработка результатов: исправление ошибок, обработка и форматирование текста.

На рисунке 1.18 показан процесс распознавания текста с помощью технологии OCR.

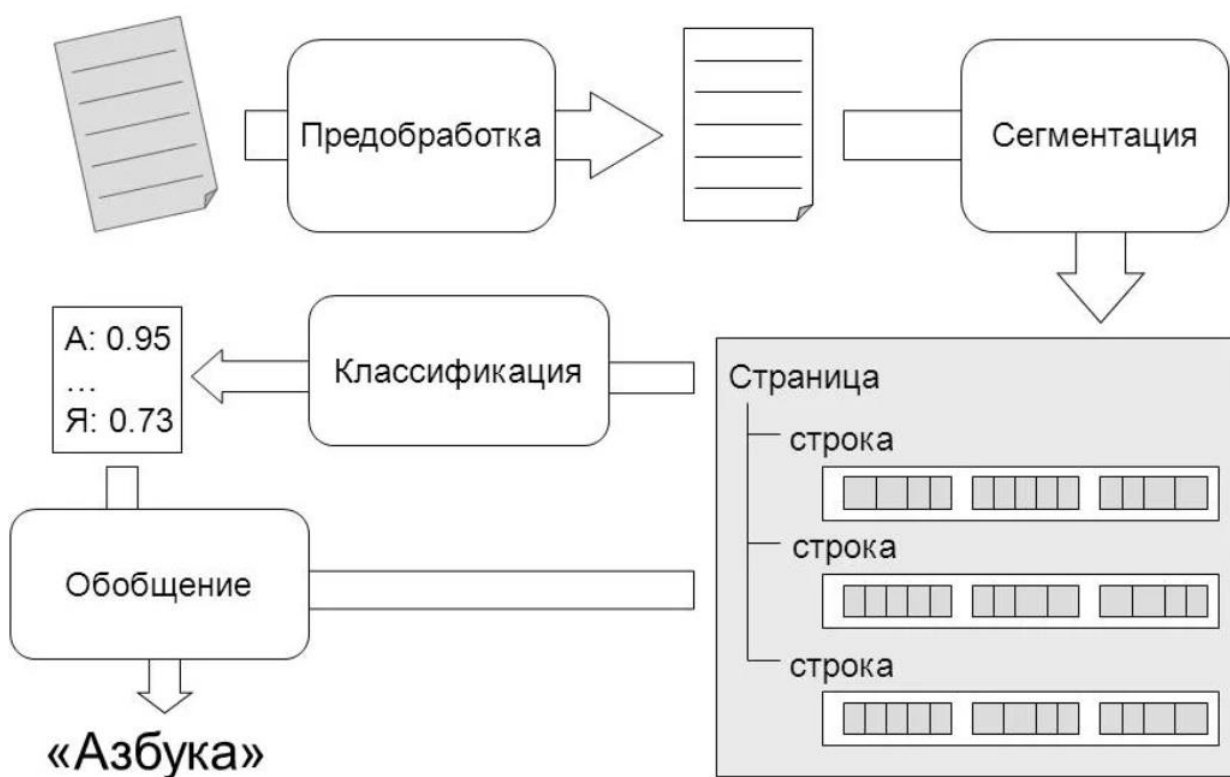


Рисунок 1.18 – Процесс распознавания текста с помощью OCR

Распознавание автомобильных номеров - это еще одно практическое применение технологии OCR. Для распознавания номеров автомобилей используются специальные алгоритмы, которые учитывают особенности шрифта и расположения символов на номерах. В некоторых случаях может потребоваться использование дополнительных методов обработки изображений, таких как фильтрация, сегментация и скелетонизация, чтобы улучшить качество распознавания.

Пример работы технологии OCR представлен на рисунке 1.19.



Рисунок 1.19 – Пример работы технологии OCR

С помощью технологии OCR можно создать мобильное приложение, которое позволит пользователям быстро и удобно получать информацию о проезжающих мимо автомобилях, в том числе и о правонарушителях на дороге. Это приложение может быть полезно для водителей, полиции, парковочных служб и других организаций.

Однако, стоит учитывать, что распознавание номеров автомобилей с помощью OCR может столкнуться с некоторыми трудностями, такими как плохое качество изображений, неравномерное освещение и различные виды искажений. Исходя из этого, при использовании данной технологии нужно тщательно подходить к выбору алгоритмов и методов обработки изображений, чтобы достичь максимальной точности распознавания.

В мобильных приложениях OCR позволяет создавать удобный и быстрый способ распознавания автомобильных номеров. Для достижения высокой точности распознавания необходимо проводить обучение нейросетей на больших объемах разнообразных данных.

В случае с распознаванием автомобильных номеров, OCR используется для распознавания текста на изображении номерного знака после его детектирования с помощью компьютерного зрения. Таким образом, OCR и компьютерное зрение совместно применяются для распознавания автомобильных номеров в мобильных приложениях.

Исходя из вышеизложенного, можно сказать, что компьютерное зрение играет важную роль в задаче распознавания автомобильных номеров. Библиотека OpenCV предоставляет широкий спектр инструментов для работы с изображениями и алгоритмов для детектирования объектов на изображениях. Для распознавания символов на изображении можно использовать нейронные сети, а также специальные библиотеки для обработки изображений с текстом, включающие технологию OCR. Комбинация этих методов позволит достигнуть высокой точности распознавания автомобильных номеров.

2 МОБИЛЬНОЕ ПРИЛОЖЕНИЕ

2.1 Требования к системе

В любом проекте, будь то разработка программного обеспечения, создание нового продукта или проведение научных исследований, важным этапом является определение требований [33]. Требования позволяют определить функциональность проекта, а также его ограничения и критерии качества.

Функциональные требования

Функциональные требования определяют функциональность программного обеспечения, т.е. описывают, какие возможности должна предоставлять разрабатываемая система. Функциональные требования включают в себя бизнес-требования и пользовательские требования. Выявлены следующие функциональные требования:

- Распознавание автомобильных номеров: приложение должно иметь возможность распознавать номера автомобилей, используя технологии машинного зрения. После распознавания номера, он должен быть отображен на экране устройства.
- Распознавание автомобильных моделей: приложение должно иметь возможность распознавать модели автомобилей, используя обученную нейронную сеть. После распознавания модели, она должна быть отображена на экране устройства.
- Поддержка файлового менеджера устройства: приложение должно иметь возможность доступа к файловому менеджеру для получения изображений автомобилей.
- Поддержка пользовательского интерфейса: приложение должно иметь интуитивно понятный пользовательский интерфейс для удобства в использовании.

Нефункциональные требования

Нефункциональные требования описывают свойства и ограничения, накладываемые на систему. Нefункциональные требования определяют бизнес-правила, системные требования и другие. Выявлены следующие нефункциональные требования:

- Точность: приложение должно обеспечивать высокую точность распознавания номеров и моделей автомобилей.
- Надежность: приложение должно быть надежным и стабильным, с минимальным количеством сбоев и ошибок.
- Поддержка разных устройств: приложение должно быть совместимо с различными устройствами, иметь адаптивный дизайн и работать на разных версиях операционных систем.

2.2 Описание набора данных

Задача распознавания образов представляет из себя правильное определение класса объекта по его изображению. Изображение представляется в виде одной (для черно-белых), либо трёх (для цветных) матриц значений яркости пикселей. Для цветных изображений каждая матрица отвечает за один из цветов RGB — красный, зеленый и синий. Чем сложнее объекты классификации, тем больше изображений должна «увидеть» сеть, чтобы корректно их классифицировать.

Совокупность таких изображений называют выборкой (датасетом). Данный набор поделен на 2 множества — обучающее и тестовое. Задача состоит в том, чтобы построить некоторую вычислительную модель, которая при помощи изображений обучающего множества научилась бы как можно точнее классифицировать изображения тестового.

Для эффективного обучения и правильной работы сети необходимы размеченные данные, содержащие большое количество примеров. В

свободном доступе существует некоторое множество таких наборов, но их разнородность и наличие ошибочных меток могут сильно повлиять на результаты работы модели. Для решения задачи определения модели автомобиля на изображении использован набор данных Stanford Cars. Источником данных выступил ресурс Kaggle, принадлежащий корпорации Google [34].

Stanford Cars Dataset [35] представляет набор данных, состоящий из изображений автомобилей разных марок и моделей. Данные в наборе были собраны с помощью интернет-поиска и охватывают 196 различных марок автомобилей и более 16 000 изображений. Каждое изображение в наборе помечено меткой, содержащей информацию о марке и модели автомобиля, а также о годе выпуска, например, 2012 Tesla Model S или 2012 BMW M3 coupe.

На рисунке 2.1 представлены изображения обучающей выборки.



Рисунок 2.1 – Изображения обучающей выборки

Набор Stanford Cars доступен для загрузки в формате MATLAB и CSV. Данные в выборке разделены на две части: тренировочный набор и тестовый набор. Тренировочный набор содержит 8 144 изображений, а тестовый набор - 8 041 изображение.

Каждое изображение в наборе данных имеет разрешение 300x300 пикселей. Изображения содержат различные особенности, такие как угол съемки, освещение и фон.

В целом, за счёт относительно небольшого количества данных для обучения, использование набора Stanford Cars в сочетании с нейросетями является оптимальным выбором, чтобы создать эффективное решение для распознавания моделей автомобилей на имеющихся в распоряжении вычислительных мощностях.

2.3 Показатели качества нейронной сети

Для оценки качества работы нейронной сети в задачах классификации и распознавания используются различные метрики. Рассмотрим основные показатели, которые позволяют определить эффективность обученной модели и сравнить ее с другими подходами.

Точность

Точность является одной из самых распространенных метрик для оценки качества классификации. Она определяется как отношение правильно классифицированных примеров ко всем примерам тестовой выборки:

Будет формула

$$\text{Accuracy} = (\text{True Positives} + \text{True Negatives}) / \text{Total Samples}$$

Эта метрика подходит для задач с сбалансированными классами, но может быть недостаточно информативной в случае сильного дисбаланса классов.

Точность, полнота и F1-мера

Точность и полнота являются важными метриками для оценки качества классификатора, особенно в случае несбалансированных классов. Точность определяет долю правильно классифицированных положительных примеров среди всех положительных примеров, предсказанных моделью. Полнота определяет долю правильно классифицированных положительных примеров

среди всех действительно положительных примеров.

Будет формула

...

F1-мера является средним гармоническим между точностью и полнотой, что позволяет учесть обе метрики одновременно.

Будет формула

...

F1-мера достигает своего максимума при полноте и точности, равных единице, и минимума при нулевых значениях этих показателей.

ROC-кривая и AUC-ROC

ROC-кривая является графическим представлением зависимости между полнотой (Recall) и специфичностью ($1 - \text{False Positive Rate}$) классификатора при различных пороговых значениях. Специфичность определяется как доля правильно классифицированных отрицательных примеров среди всех действительно отрицательных примеров.

AUC-ROC (Area Under the ROC Curve) представляет собой численное значение площади под ROC-кривой. AUC-ROC может принимать значения от 0 до 1, где значение 1 соответствует идеальному классификатору, а значение 0.5 – случайному классификатору. Метрика AUC-ROC хорошо работает для сравнения классификаторов, особенно в случае несбалансированных классов.

Таким образом, для оценки качества нейронной сети в задачах классификации и распознавания важно учитывать различные метрики, такие как точность, полнота, F1-мера, ROC-кривая и AUC-ROC. Комбинация этих показателей позволяет получить более полную картину о работе модели и

сравнить ее с другими подходами.

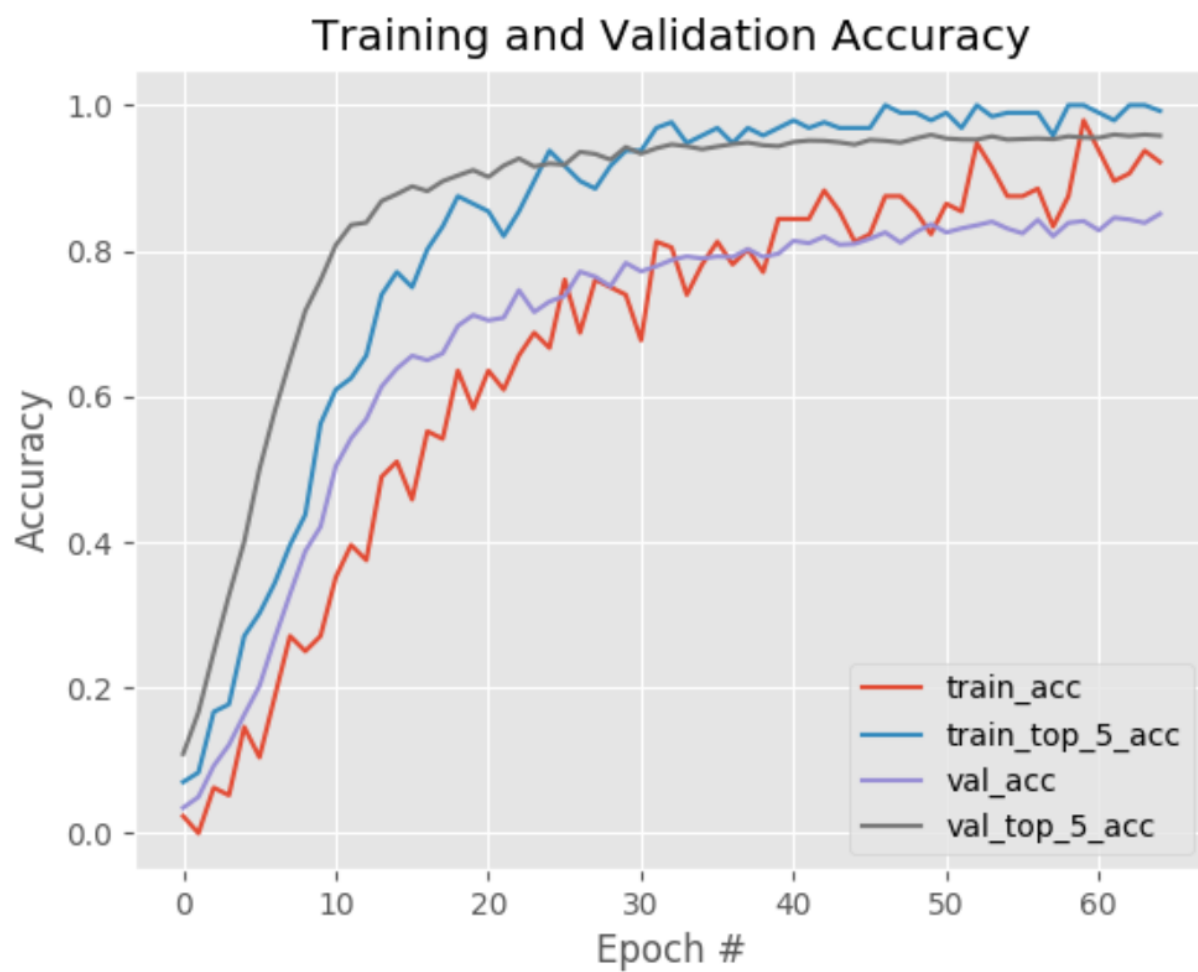
2.4 Процесс предварительной подготовки и обработки данных

Задача распознавания образов представляет из себя правильное определение класса объекта по его изображению. Изображение представляется в виде одной (для черно-белых), либо трёх (для цветных) матриц значений яркости пикселей. Для цветных изображений каждая матрица отвечает за один из цветов RGB — красный, зеленый и синий. Чем сложнее объекты классификации, тем больше изображений должна «увидеть» сеть, чтобы корректно их классифицировать.

2.5 Создание модели классификации автомобилей

Задача



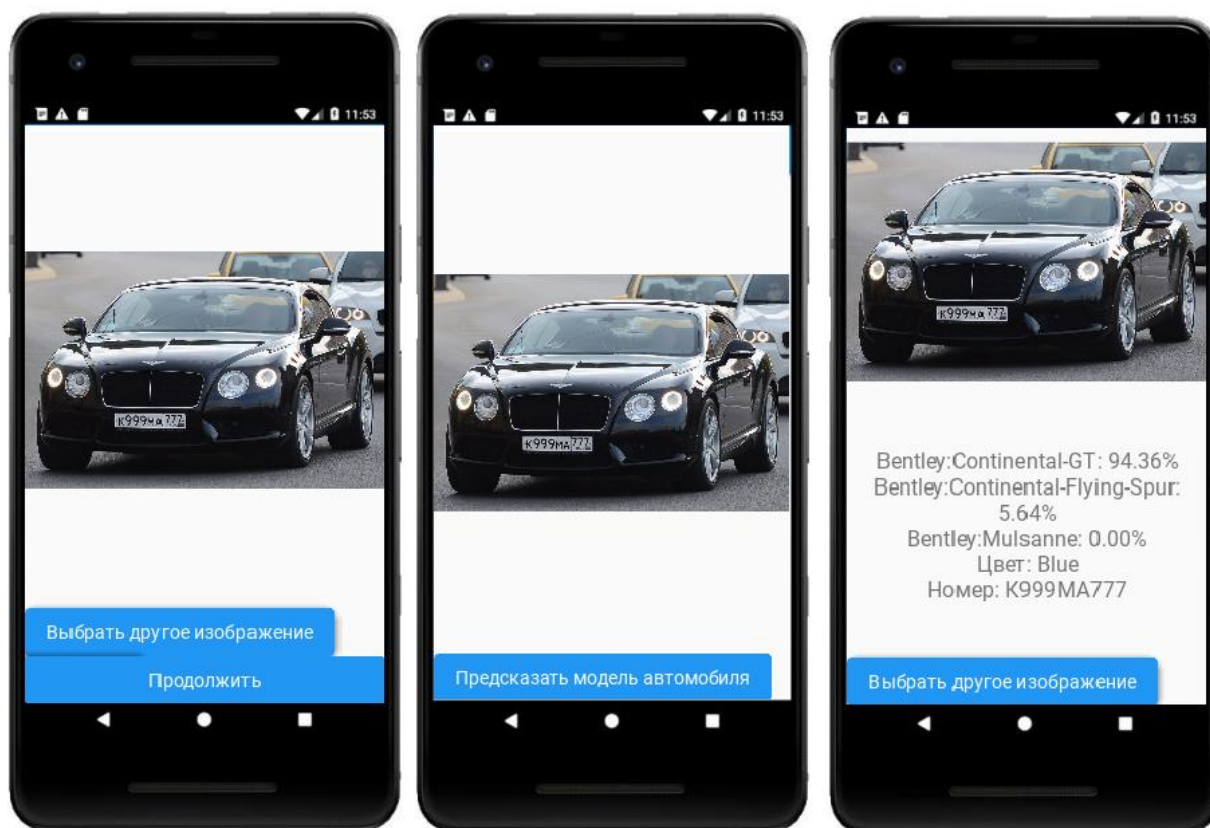


2.6 Создание модели распознавания автомобильных номеров

Задача

2.7 Разработка мобильного приложения

Задача



Заключение

В результате прохождения технологической практики можно сделать следующие выводы:

- изучена структура и деятельность предприятия и подразделения прохождения практики, а именно структура и деятельность Томского государственного университета систем управления и радиоэлектроники (ТУСУР) и кафедры экономической математики, информатики и статистики

(ЭМИС), которая занимается подготовкой специалистов, занимающихся исследованием, разработкой, внедрением и сопровождением информационных технологий и систем для различных устройств;

- выявлена проблема необходимости разработки мобильного приложения для распознавания автомобильных моделей и номеров;

- сформулирована цель технологической практики: разработка мобильного приложения для распознавания моделей автомобилей и номеров с использованием нейросетевых технологий;

- для достижения поставленной цели сформулированы задачи практики: изучение предметной области дипломной работы, изучение строения нейронных сетей, проведение анализа существующих на рынке решений, сбор и обработка наборов изображений автомобилей различных моделей и автомобильных номеров для обучения нейронной сети, рассмотрение средств разработки для создания мобильного приложения, проведение сравнительного анализа архитектур нейронных сетей и выбор на его основе нейронной сети для создания приложения;

- определен, с учетом анализа аналогов, функционал разрабатываемого мобильного приложения, выделены его функциональные и нефункциональные требования;

- рассмотрены средства разработки нейронных сетей, принято решение использовать в качестве основных средств фреймворк Kivy и язык программирования Python, в качестве вспомогательного средства – библиотеки Keras, TensorFlow и OpenCV;

- проведен сбор и обработка наборов изображений автомобилей различных моделей и автомобильных номеров, выбрать оптимальный набор данных, удовлетворяющий требованиям разработки;

- проведено исследование в области компьютерного зрения, выделены технологии и методы для реализации алгоритма распознавания автомобильных номеров;
- изучена архитектура нейросети VGG16, проанализирован и описан принцип ее работы, сделаны выводы о преимуществах и недостатках данной модели;
- проведен сравнительный анализ наиболее популярных архитектур сверточных нейронных сетей, выявлена эффективность обучения каждой из моделей, а также их преимущества и недостатки в сравнении друг с другом, на основе анализа выбрана оптимальная архитектура для разработки мобильного приложения с использованием алгоритмов распознавания моделей автомобилей и транспортных номеров.

Список использованных источников

1. Томский государственный университет систем управления и радиоэлектроники [Электронный ресурс]: официальный сайт ТУСУР. URL: <https://tusur.ru> (дата обращения: 28.03.2023).
2. Томск – город студентов [Электронный ресурс]: официальный сайт ТГАСУ. URL: <https://tsuab.ru/article/aboutTomsk> (дата обращения: 28.03.2023).
3. Томский проект "Большой университет" [Электронный ресурс]: сайт РИА Томск. URL: https://www.riatomsk.ru/article/Big_University_Tomsk (дата обращения: 28.03.2023).
4. «Приоритет-2030» [Электронный ресурс] официальный сайт Минобрнауки России. URL: <https://minobrnauki.gov.ru/action/priority2030> (дата обращения: 28.03.2023).
5. Каллан Р. Основные концепции нейронных сетей. – СПб.: Вильямс, 2003. – 288 с.
6. Осовский С. Нейронные сети для обработки информации. – М.: Финансы и статистика, 2002. — 344 с.: ил. — Пер. с польского И.Д. Рудинского.
7. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. – М.: ДМК Пресс, 2015. – 400 с.
8. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. – М.: Мир, 1992. – 184 с.
9. Умберто Микелуччи. Прикладное глубокое обучение. Подход к пониманию глубоких нейронных сетей на основе метода кейсов. // пер. с англ. – СПб.: БХВ-Петербург, 2020. – 368 с.
10. Николенко С., Кадурин Е., Архангельская А. Глубокое обучение. Погружение в мир нейронных сетей. – СПб, 2018. – 480 с.
11. Джоши, Пратик. Искусственный интеллект с примерами на Python // пер. с англ. – СПб.: ООО «Диалектика», 2019. – 448 с.

12. Анализ технологий распознавания текста на изображениях [Электронный ресурс]: Cyberleninka. URL: <https://cyberleninka.ru/article/n/analiz-tehnologiy-raspoznavaniya-teksta-iz-izobrazheniya/viewer> (дата обращения 28.03.2023).
13. Сверточные нейронные сети [Электронный ресурс]: Se.poev.info. URL: https://se.moevm.info/lib/exe/fetch.php/courses:artificial_neural_networks:lecture_3.pdf (дата обращения: 28.03.2023).
14. ANPR [Электронный ресурс]: Хабр. URL: <https://habr.com/ru/company/croc/blog/158719/> (дата обращения 28.03.2023).
15. Мобильный Авто.ру [Электронный ресурс]: Auto.ru. URL: <https://auto.ru/app/> (дата обращения 28.03.2023).
16. Умная камера [Электронный ресурс]: Yandex.ru. URL: https://yandex.ru/project/searchapp/web/t/smartcamera_general (дата обращения 28.03.2023).
17. Python [Электронный ресурс]: Python.org. URL: <https://www.python.org/> (дата обращения 28.03.2023).
18. TensorFlow [Электронный ресурс]: TensorFlow.org. URL: <https://www.tensorflow.org/> (дата обращения 28.03.2023).
19. Keras [Электронный ресурс]: Keras.io. URL: <https://keras.io/> (дата обращения 28.03.2023).
20. OpenCV [Электронный ресурс]: opencv.org. URL: <https://opencv.org/> (дата обращения 28.03.2023).
21. Kivy [Электронный ресурс]: Kivy.org. URL: <https://kivy.org/> (дата обращения 28.03.2023).
22. Селянкин В. В. Компьютерное зрение. Анализ и обработка изображений / В. В. Селянкин. – М.: Лань, 2019. – 152 с.
23. Хилл К. Научное программирование на Python / К. Хилл. – М.: ДМК-Пресс, 2021. – 646 с.

24. Абдрахманов М. И. Devpractice Team. Python. Визуализация данных / М. И. Абдрахманов. – М.: Devpractice, 2021. – 412 с
25. MachineLearning [Электронный ресурс]. – Режим доступа: <http://www.machinelearning.ru/> (дата обращения: 15.03.2023).
26. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. – М.: ДМК Пресс, 2015. – 400 с.
27. ImageNet [Электронный ресурс]: image-net.org. URL: <https://image-net.org/> (дата обращения 28.03.2023).
28. VGG16 [Электронный ресурс]: Neurohive. URL: <https://neurohive.io/ru/vidy-nejrosetej/vgg16-model/> (дата обращения 28.03.2023).
29. Применение обученной модели VGG16 [Электронный ресурс]: Хабр. URL: <https://habr.com/ru/company/skillfactory/blog/545384/> (дата обращения 28.03.2023).
30. Как устроено компьютерное зрение? [Электронный ресурс]: Хабр. URL: <https://habr.com/ru/company/droider/blog/538750/> (дата обращения 28.03.2023).
31. Деликатное введение в OCR [Электронный ресурс]: Machinelearningmastery.ru. URL: <https://machinelearningmastery.ru/a-gentle-introduction-to-ocr-ee1469a201aa/> (дата обращения 28.03.2023).
32. Технология OCR [Электронный ресурс]: Хабр. URL: <https://habr.com/ru/company/arcadia/blog/505950/> (дата обращения 28.03.2023).
33. Редько, В.Г. Эволюция, нейронные сети. Модели и концепции эволюционной кибернетики / В.Г. Редько. - М.: Ленанд, 2015. - 224 с.
34. Kaggle [Электронный ресурс]: Kaggle.com. URL: <https://www.kaggle.com/> (дата обращения 28.03.2023).
35. Stanford Cars [Электронный ресурс]: ai.stanford.edu. URL: https://ai.stanford.edu/car_dataset.html (дата обращения 28.03.2023).