

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**ИСПОЛЬЗОВАНИЕ КОМПЬЮТЕРНОГО ЗРЕНИЯ В МОБИЛЬНОМ
ПРИЛОЖЕНИИ ДЛЯ СОВЕРШЕНИЯ ИНТЕРНЕТ-ПОКУПОК**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные
технологии
факультета КНиИТ
Слепова Ильи Алексеевича

Научный руководитель
доцент, к. ф.-м. н.

С. В. Миронов

Заведующий кафедрой
к. ф.-м. н., доцент

С. В. Миронов

Саратов 2022

ВВЕДЕНИЕ

Удобство поиска товаров потенциальным покупателем — один из важнейших аспектов в торговле, именно поэтому многие торговые проводят цифровизацию бизнеса, предоставляя потребителю возможность заказать товары в интернете, не выходя из дома.

Зачастую поиск на подобных площадках происходит только по каким-либо фильтрам, категориям и ключевым словам, выполняя запросы в базы данных. Но, в последнее время, некоторые компании начали предоставлять возможность поиска товаров по изображениям. Более того, в некоторых случаях возможен поиск при наведении камеры мобильного устройства пользователя на интересующий объект. Примерами таких компаний являются Amazon, Google. Примером приложения для поиска товаров в реальном времени может выступать «Google Lens», которое обрабатывает большие объемы данных для обучения моделей подобного поиска с помощью облачных технологий, что делает данное решение достаточно дорогим.

В последнее время активно поднимается вопрос импортозамещения различных решений, но, к сожалению, отечественные торговые сети пока не обращают внимания на подобные способы поиска товаров, из-за чего отечественных аналогов «Google Lens» пока еще нет, но данное приложение уже доказало свою эффективность для привлечения новых покупателей, а также постепенно стирает рамки интернет-торговли. Поэтому в данной работе будет разработан аналог «Google Lens» без использования каких-либо зарубежных коммерческих технологий, таких как облачные платформы «Google Cloud Platform» или «Amazon Web Services».

Для решения этой задачи выбраны следующие инструменты, позволяющие максимально ускорить процесс детектирования и поиска объектов, а также свести количество ошибок к минимуму:

- язык программирования Python, использующийся для работы с компьютерным зрением и разработки поиска;
- предобученная на миллионах изображений модель глубокого обучения «MobileNet SSD», разработанная с использованием фреймворка «TfLite» для детектирования объектов, используя мощности устройства, на котором запущено приложение;
- «MNASNet», являющийся автоматизированным решением для построе-

ния сверточных нейронных сетей для работы с изображениями и разработанный с использованием фреймворка «PyTorch», необходимый для векторизации и классификации изображений;

- библиотека «Faiss», позволяющая крайне быстро находить нужную информацию по матрице сходства, в данном случае инструмент будет использоваться для поиска наиболее похожих изображений;
- язык программирования Kotlin, необходимый для разработки приложения под операционную систему Android.

Как описано ранее, подобные решения являются достаточно актуальными, так как сильно увеличивают количество случайных продаж, когда потребитель не знает название товара, но хотел бы его найти. Также увеличится приток новых пользователей благодаря удобству, так как достаточно только навести камеру мобильного устройства на нужный товар или воспользоваться уже готовой фотографией для поиска интересующего товара, и условной геймификации процесса поиска товаров. Еще одним вариантом использования подобных технологий являются магазины, формат которых не подразумевает наличие касс — пробные версии подобных магазинов ранее запускали Amazon и X5 Group.

Целью данной работы является разработка приложения для мобильной операционной системы Android, позволяющего детектировать объекты как на статических изображениях, так и в реальном времени, а также искать похожие товары в базе и презентовать пользователю. Для достижения поставленной цели необходимо:

- разработать модуль для детектирования объектов на статических изображениях на основе машинного обучения;
- расширить разработанный модуль для детектирования объектов в реальном времени;
- разработать модуль хранения и поиска изображений с объектами;
- разработать приложение для операционной системы Android, объединяющее ранее описанные модули.

Методологические основы использования компьютерного зрения представлены в работах Э. Д. Шакирьянова и Н. А. Похоренок.

Практическая значимость работы состоит в разработке безальтернативного на момент разработки приложения, имеющего множество путей развития, как в торговле, так и в сферах образования, медицины и других.

Структура и объём работы. Бакалаврская работа состоит из введения, 3 разделов, заключения, списка использованных источников и 6 приложений. Общий объем работы — 136 страниц, из них 72 страницы — основное содержание, включая 16 рисунков, цифровой носитель в качестве приложения, список использованных источников информации — 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Анализ предметной области» посвящен рассмотрению существующих решений, а также объясняется выбор используемых для реализации приложения технологий, в нем также представлена и подробно описана архитектура приложения, состоящая из трех основных модулей:

- модули детектирования объектов в реальном времени и на статическом изображении;
- модуль поиска объектов, состоящий из модулей векторизации изображений, их хранения и поиска;
- приложение под операционную систему Android.

Модули детектирования объектов основаны на моделях глубокого обучения «MobileNet SSD» и «MNASNet» и лучшим языком программирования в этой области на данный момент является Python, так как на его основе написано множество вспомогательных библиотек, например, Pandas или NumPy, широко используемые для обработки данных, а также PyTorch, который используется непосредственно для реализации нейронных сетей и работы с моделями машинного обучения. Стоит отметить, что также свое применение в данной области нашел и C++, но на данный момент он занимает меньшую часть рынка из-за сложных оптимизаций модели и применяется в основном для реализации моделей, требующих максимальную скорость и точность детектирования на умных устройствах.

Следующий модуль также основан на машинном обучении и реализован с помощью Python, но в нем присутствует новейший инструмент «Faiss», позволяющий делать максимально быстрый на данный момент поиск сходства объектов. Вместе с ним также используется компонент «MNASNet», который позволяет автоматически строить сверточные нейронные сети под конкретные случаи классификации объектов и на данный момент является решением с наибольшей точностью классификации и распознавания сходства между изображениями, а аналоги в лице «NASNet-A», «SqueezeNext» и многих других уступают ему по этому показателю на пять и более процентов, что является существенным отрывом в области классификации и векторизации на основе компьютерного зрения.

Последним модулем является Android приложение, для разработки которого выбран язык программирования Kotlin. Также популярными решения-

ми разработки под данную операционную систему являются игровой движок «Unity3d», реализованный на C#, а также язык программирования Java. Стоит отметить, что Kotlin основан на «JVM», который также лежит в основе Java, но на ее фоне Kotlin особенно выделяется, так как создан именно для Android разработки и имеет внутри множество оптимизаций, которые при разработке на Java продумываются и реализуются вручную. Что касается «Unity3d», его производительность весьма ограничена при интеграции с машинным обучением и дополнительной оптимизации потоков приложения, что является неплохим, но все еще отстающим от Kotlin решением для разработки приложения, рассчитанного на работу в реальном времени.

Таким образом, была исследована предметная область, а также определены инструменты для реализации практической части.

Второй раздел «Модули работы с изображениями» посвящен введению в теорию в сверточные сети и компьютерное зрение, а также описанию использованных в работе моделей глубокого обучения и реализации связанных с ними модулей.

Модуль детектирования объектов. В задачах детектирования и классификации объектов разделяют два основных типа моделей: одноэтапные и двухэтапные. Так, одноэтапная модель одновременно предсказывает ограничение области вокруг объекта и значения степеней принадлежности данного объекта к какому-либо классу, то есть классифицирует его. Двухэтапная, в свою очередь, разделяет этапы локализации и классификации объекта. Данные отличия явно определяют ниши для использования обоих типов моделей: одноэтапные работают намного быстрее, из-за чего хороши для использования в системах реального времени, например, в беспилотных автомобилях, а двухэтапные имеют намного большую точность, что позволяет использовать их, например, в медицине.

Для разрабатываемого приложения больше подходит одноэтапный детектор объектов, каким и является модель глубокого обучения «MobileNet SSD» разработанная с использованием фреймворка «TfLite». Данная модель не единственная реализованная на данный момент и помимо нее для реализации модуля также рассматривались модели «Faster R-CNN», «R-FCN», «YOLO», «FPN» и «RetinaNet». Основная проблема подобных решений — сложность реализации для работы в мобильных приложениях с использованием мощностей процессора

устройства.

Чтобы отслеживать выделение объекта, исходный функционал модели был расширен для передачи зоны выделения во внешние сервисы.

В результате была описана методология детектирования объектов, а также разработан соответствующий модуль.

Модуль поиска и хранения объектов. Сверточная сеть состоит из сверточных и полносвязных слоев, а также слоев подвыборки. Так, на первом этапе изображения попадает в сверточные слои, которые извлекают объекты из входных изображений, сохраняя при этом пространственные отношения между пикселями за счет изучения отдельных небольших областей, а не всего изображения сразу. То есть представляет изображение в виде большой матрицы, каждый элемент которой разбит на пиксели. Далее часть исходной матрицы умножается на меньшую по размеру матрицу, именуемой фильтром, благодаря чему происходит свертка исходной матрицы, а затем фильтр смещается на определенное количество пикселей и процесс повторяется. При этом, чем больше фильтров используется иначе говоря чем глубже свертка, тем точнее будет конечный результат свертки.

Вслед за операциями свертки применяются операции понижающей дискретизации, что позволяет уменьшить размерность отдельных частей матрицы, при этом сохранив значимую об объекте информацию. Происходит это за счет выбора окна с определенным количеством элементов части матрицы, которое берет максимальный элемент в каждом окне. После чего на полученные матрицы применяются операции выравнивания, переводя каждую строку матрицы в столбец.

Полученные данные переходят в полносвязный слой, именуемый многослойным перцептроном. На данном этапе каждый полученный нейрон связывается с каждым нейроном на следующем слое, а их расположение и количество нейронов определяется ранее полученными матрицами, которые, благодаря значениям весов, выбранным на прошлом этапе, производят заключительную классификацию объекта.

Алгоритм построения сверточной сети с использованием «MNASNet» делится на слои, которые сгруппированы в несколько predeterminedных блоков на основе размеров и фильтров входного изображения, и каждый блок содержит переменное количество повторяющихся схожих слоев. Для каждого блока

происходит поиск возможностей объединения для нескольких слоев, после чего повторяется один и тот же, что достигается за счет:

1. сверточных операций: обычной, глубокой и обратной свертки;
2. динамического размера ядра свертки;
3. операций сжатия полиномов;
4. операций объединения, пропуска и выявления идентичности кадров.

Для снижения нагрузки на мобильное устройство было разработано серверное приложение, которое получает изображение, а затем векторизует его с помощью «MNASNet» и ищет сходство полученного вектора с использованием инструмента «FAISS», после чего передает результат обратно в приложения.

Таким образом, были описаны теоретические аспекты используемых инструментов и сверточных сетей, а также разработана серверная часть, позволяющая искать схожие изображения.

Третий раздел Раздел «**Android приложение**» содержит описание полного цикла разработки приложения для мобильной операционной системы Android с интеграцией ранее реализованных модулей, который делится на несколько этапов: разработка интерфейса, интеграция работы с камерой и внутренней памятью, интеграция работы с модулями машинного обучения.

Разработка интерфейса. Разрабатывается интерфейс трех основных экранов приложения — домашний, а также экраны детектирования в статическом и динамическом режимах. С помощью вспомогательных объектов решается проблема утечек памяти и блокировки потоков при использовании камеры.

Стоит отметить, что при описании разработки экрана статического режима также затрагивается интеграция с внутренней памятью устройства, а при описании разработки экрана динамического режима со всеми камерами устройства для реализации элемента предпросмотра изображения в реальном времени.

В результате был разработан интерфейс приложения.

Разработка бизнес-логики. Разрабатывается бизнес-логика приложения для каждого отдельного модуля. Всего у приложения есть семь разных состояний: режим не выбран, поиск объекта на изображение, объект найден, подтверждение детектирования, а после успешного детектирования поиск и успешный поиск. Эти состояния меняются с помощью вспомогательных объектов, которые интегрированы со всеми ранее описанными модулями.

Модуль поиска интегрирован с помощью *HTTP*-запросов из Android при-

ложения, посредством которых происходит передача изображения и дополнительной информации о них.

Модуль детектирования интегрирован, как артефакт ранее описанной модели, так как выбранный для реализации модели инструмент «TfLite» позволяет ей работать непосредственно с использованием мощностей процессора мобильного устройства. Основными отличиями режимов детектирования источники получения изображения, а также количество одновременно детектируемых объектов — в динамическом режиме возможно детектировать только один объект за цикл, а в статическом несколько объектов параллельно.

Таким образом, было полностью реализовано Android приложение с интеграцией ранее описанных модулей.

ЗАКЛЮЧЕНИЕ

В процессе выполнения бакалаврской работы были выполнены следующие задачи:

- разработан модуль детектирования объектов в реальном времени, а также на статических изображениях, с использованием библиотеки «TFLite» и модели «MobileNet SSD»;
- разработан модуль хранения и поиска похожих изображений с помощью моделей «MNASNet» и поискового инструмента «Faiss»;
- разработано приложение для операционной системы Android с интеграцией ранее описанных модулей;

Данная технология может стать незаменимой в современной интернет-торговле, так как при интеграции с базой товаров интернет-магазина заметно снизит время на поиск нужного товара и за счет этого увеличит количество случайных покупок.

Разработанное приложение возможно масштабировать, интегрируя в него несколько интернет-магазинов, тем самым создавая универсальную платформу, где пользователь может выбрать товар из любого магазина по личным предпочтениям, а с добавлением функционала автоматического обнаружения и сканирования штрих-кода товаров, приложение также возможно интегрировать в физические магазины, отказавшись тем самым от касс самообслуживания.

Основные источники информации:

1. *Шакирьянов, Э. Д.* Компьютерное зрение на Python / Э. Д. Шакирьянов. — Москва: Лаборатория знаний, 2021.
2. *Похоренок, Н. А.* OpenCV и Java. Обработка изображений и компьютерное зрение / Н. А. Похоренок. — Санкт-Петербург: БХВ-Петербург, 2018.
3. *Eckel, B.* Atomic Kotlin / B. Eckel, S. Isakova — London: Mindview LLC, 2021.
4. *Laganier, R.* OpenCV 3 Computer Vision Application Programming Cookbook / R. Laganier — Birmingham: Packt Publishing, 2017.
5. MnasNet: Platform-Aware Neural Architecture Search for Mobile (Image Classification) [Электронный ресурс] — URL: <https://sh-tsang.medium.com/reading-mnasnet-platform-aware-neural-architecture-search-for-mobile-image-classification-b042aaef66f7> (Дата обращения 17.04.2022).
Загл. с экр. Яз. англ.

6. SSD MobileNetV1 architecture [Электронный ресурс] — URL: <https://iq.opengenus.org/ssd-mobilenet-v1-architecture/> (Дата обращения 07.04.2022). Загл. с экр. Яз. англ.