

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

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

Кафедра дискретной математики и информационных технологий

**РАЗРАБОТКА ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНОЙ  
НЕЙРОННОЙ СЕТИ И МОБИЛЬНОГО ПРИЛОЖЕНИЯ НА  
БАЗЕ ОПЕРАЦИОННОЙ СИСТЕМЫ ANDROID ДЛЯ  
УЛУЧШЕНИЯ КАЧЕСТВА ФОТОГРАФИЙ**

**АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ**

студента 2 курса 271 группы  
направления 09.04.01 — Информатика и вычислительная техника  
факультета КНиИТ  
Николаева Ивана Сергеевича

Научный руководитель

к. ф.-м. н., доцент

\_\_\_\_\_

И. Д. Сагаева

Заведующий кафедрой

доцент, к. ф.-м. н.

\_\_\_\_\_

Л. Б. Тяпаев

Саратов 2024

## ВВЕДЕНИЕ

Обучение генеративно состязательной нейронной сети (GAN) для улучшения качества фотографий является одной из важных задач в области компьютерного зрения. GAN представляет собой архитектуру, состоящую из двух основных компонентов: генератора и дискриминатора. В процессе обучения эти две сети соревнуются друг с другом, чтобы достичь лучшего качества восстановленных изображений.

Важным аспектом обучения GAN является подготовка качественного датасета. Датасет должен включать в себя пары изображений, где одно изображение является оригинальным, а другое - его низкокачественной версией. Это помогает генератору научиться восстанавливать детали и улучшать качество изображений.

Помимо этого разработка мобильного приложения для взаимодействия с нейронной сетью представляет собой актуальную задачу в области информационных технологий. Мобильные устройства становятся все более востребованными среди пользователей, и создание интуитивного и функционального приложения для них становится неотъемлемой частью современного программного обеспечения.

В данной работе фокус направлен на проектирование и реализацию мобильного приложения под платформу Android, которое будет взаимодействовать с нейронной сетью. Приложение предоставит пользователям возможность выполнять POST и GET запросы к нейронной сети, обеспечивая тем самым удобный и эффективный доступ к её функционалу.

Целью данной работы является разработка и реализация модели генеративно-состязательной нейронной сети для улучшения качества изображений, а также создание инновационного мобильного приложения, способного взаимодействовать с нейронной сетью, с акцентом на простоту использования и функциональность. В рамках поставленной цели были выделены следующие задачи:

- подобрать и изучить актуальную литературу по рассматриваемой теме;
- рассмотреть инструментальные средства Python для построения нейронных сетей;
- разработать и обучить собственную генеративно-состязательную сеть для улучшения качества изображений;

- Провести анализ и изучение актуальной литературы в области мобильной разработки и взаимодействия с нейронными сетями;
- Выбрать и ознакомиться с инструментальными средствами для разработки мобильных приложений под Android;
- Разработать функциональную модель мобильного приложения;
- Реализовать мобильное приложение, обеспечивающее обмен данными с нейронной сетью через POST и GET запросы.

## КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Основное содержание работы включает в себя реферативное изложение теории генеративно-сопоставительной нейронной сети, ее обучения, а так же обзор используемых инструментов для обучения и разработки мобильного приложения под операционную систему Android и сервера на языке программирования Python с использованием микрофреймворка Flask.

Цель генератора - обмануть дискриминатор, чтобы сгенерированные им изображения были неотличимы от реальных изображений. Для достижения этой цели генератор обучается минимизировать функцию потерь, которая оценивает расхождение между сгенерированными и реальными изображениями.

**В первом разделе** приводится теоретическое изложение о том, что такое генеративно-сопоставительная нейронная сеть, из каких компонентов она состоит и как работает. Генеративно-сопоставительная нейросеть (Generative adversarial network, GAN) — архитектура, состоящая из генератора и дискриминатора, настроенных на работу друг против друга. Таким образом GAN и получила название генеративно-сопоставительная нейронная сеть.

Генеративно-сопоставительная нейронная сеть была впервые представлена в работе 2014-го года Ian Goodfellow и других исследователей университета Монреаля, включая Yoshua Bengio. Директор Facebook по исследованиям искусственного интеллекта Yann LeCun назвал сопоставительную претренировку сетей “самой интересной идеей в машинном обучении за последние 10 лет”. Потенциал Генеративно-сопоставительной нейронной сети огромен, поскольку она имитирует любое распределение данных. GAN обучают создавать структуры, невероятно сильно похожие на сущности из реального мира в области изображений, музыки, речи, прозы.

Дискриминатор - это нейронная сеть, которая принимает на вход изображение и оценивает его реалистичность. Он выполняет задачу классификации и определяет, является ли поданное изображение реальным (принадлежит исходному датасету) или сгенерированным (принадлежит генератором). Дискриминатор получает на вход как реальные изображения из обучающего датасета, так и сгенерированные изображения от генератора.

Цель дискриминатора - правильно классифицировать реальные и сгене-

рированные изображения. Дискриминатор обучается максимизировать функцию потерь, которая оценивает его способность различать эти два типа изображений.

Генератор - это нейронная сеть, которая принимает на вход случайный шум или некоторое другое представление и генерирует новые данные, которые соответствуют заданному формату или распределению. В случае обучения GAN для улучшения качества фотографий, генератор принимает на вход низкокачественное изображение (например, размытое или с низким разрешением) и старается сгенерировать высококачественное изображение, которое максимально приближено к оригинальному.

**Во втором разделе** рассказывается о принципах и этапах обучения собственной модели генеративно-состязательной нейронной сети. Принцип работы нейронной сети и ее структура взяты из нейробиологии. Сама идея заключалась в том, чтобы получить математическую модель и ее программное воплощение, которые бы имитировали деятельность человеческого мозга. Разработками в этой области ученые занимаются уже с середины прошлого века. Однако лишь в последние годы развитие нейросетей смогло достичь впечатляющих результатов.

Ни одна вычислительная система не в состоянии воплотить аналитические способности человеческого мозга. Между тем, именно эти качества необходимы программам для решения ряда сложных задач. Поэтому ключевой принцип любой нейронной сети подобен возможностям нашего головного мозга - возможность обучаться. От качества и длительности обучения нейронной сети зависит сможет ли она в сложной и неординарной ситуации определить наилучший вариант решения поставленной задачи.

Обучение генеративно состязательной нейронной сети (GAN) является итеративным процессом, в котором две модели - генератор и дискриминатор - соревнуются друг с другом. Генератор стремится создать высококачественные данные (например, изображения), которые могут обмануть дискриминатор, в то время как дискриминатор стремится различать реальные данные от сгенерированных.

Процесс обучения генеративно-состязательной нейронной сети является сложным и точным, так как он проходит множество этапов. В зависимости

от заданного датасета, а так же входных параметров (шум, веса и так далее) будет определяться итоговое качество полученной модели. Так что даже к самому простому этапу такой как подбор датасета нужно подходить правильно. Обучение генеративно-сопоставительной нейронной сети можно разбить на следующие этапы:

**В третьем разделе** проводится обзор инструментария, которые были использованы в данной работы - язык программирования Python с использованием библиотек TensorFlow, OpenCV, Flask, SQLite3; игровой движок Unreal Engine 5 для разработки мобильного приложения и его сравнительный анализ с аналогичными инструментами разработки.

Для разработки и обучения GAN нейросети был выбран язык программирования Python, который стал стандартом в области машинного обучения и глубокого обучения. Python предлагает широкий спектр библиотек и фреймворков, облегчающих разработку, обучение и развертывание нейросетей, одной из которых является TensorFlow.

Python — это высокоуровневый язык программирования общего назначения, известный своей простотой и читаемостью. Его синтаксис позволяет разработчикам писать более краткий и понятный код по сравнению с другими языками программирования, что особенно важно при работе с сложными моделями машинного обучения.

TensorFlow — это открытый фреймворк для машинного обучения, разработанный компанией Google. Он широко используется для построения и обучения нейросетевых моделей, включая GAN. TensorFlow предоставляет гибкую и масштабируемую инфраструктуру для выполнения сложных вычислений, что делает его идеальным инструментом для разработки моделей глубокого обучения.

В качестве движка мобильного приложения был выбран Unreal Engine 5 от компании Epic Games. Unreal Engine 5 (UE5) — это следующая версия UDK (Unreal Development Kit). Это новый движок, выпущенный той же компанией. Этот комплект для разработки игр поставляется с огромными возможностями для мобильных устройств, а также для разработки игр и приложений под различные платформы. UE5 включает в себя отличные графические возможности, такие как расширенное динамическое освещение и новую систему частиц, которая одновременно может обрабатывать до миллиона частиц в

сцене. Это определенно удобно для любого разработчика 3D-игр и приложений.

При разработке серверной части для обработки клиентских запросов был выбран микрофреймворк Flask. Flask, являясь легковесным фреймворком для веб-разработки на языке программирования Python, предоставляет не только базовый функционал, но и возможность расширения в соответствии с требованиями проекта. Он активно использует инструменты Werkzeug для обработки HTTP-запросов и Jinja2 для шаблонизации веб-страниц, что содействует удобному взаимодействию с клиентскими запросами и визуализации данных.

В качестве системы управления базами данных была выбрана встроенная реляционная база данных SQLite3. Этот выбор был обусловлен не только легкостью интеграции, но и многими другими преимуществами. SQLite3, будучи самодостаточной и встраиваемой СУБД, не требует отдельного сервера и хранит все данные в одном файле базы данных.

**В четвертом разделе** подробнее рассказывается об проектировании систем, таких как разработка мобильного приложения, сервера для взаимодействия с генеративно-состязательной нейронной сетью и генератором эффектов, и системы управления базами данных.

Приложение представляет собой графический интерфейс для устройств на базе Android, который позволяет пользователям загружать изображения и отправлять их на сервер для обработки. Сервер выполняет все вычисления и обработки изображений, а также другие взаимодействия, такие как регистрация пользователей и публикация результатов в ленте активности. Это решение обеспечивает стабильную и быструю работу приложения независимо от мощности мобильного устройства, а также защищает алгоритмы и данные от несанкционированного доступа.

Сервер отвечает за вычисления и обработку изображений пользователей. Это значительно упрощает модель работы приложения, поскольку сложные и ресурсоемкие операции выполняются на сервере, а не на мобильных устройствах пользователей. Сервер предоставляет пользователям несколько ключевых функций. Во-первых, позволяет обрабатывать изображения с помощью нейронной сети (GAN) и генератора эффектов. Пользователи могут загружать свои изображения для улучшения качества и добавления различ-

ных эффектов. Во-вторых, сервер осуществляет регистрацию и авторизацию пользователей, обеспечивая безопасность и управление учетными записями. В-третьих, сервер предоставляет информацию из ленты активности пользователей, которые разрешили публиковать обработанные изображения в общий доступ.

Для решения задачи хранения данных была выбрана система управления базами данных (СУБД) SQLite3, интегрированная с Python. SQLite3 позволяет с помощью языка SQL организовать полноценную базу данных, обладая при этом легкостью интеграции и простотой использования. Это делает её идеальным выбором для мобильных приложений, где важны компактность и эффективность.

**В пятом разделе** более подробно рассматривается процесс обучения собственной модели генеративно-сопоставительной нейронной сети, а также приводятся примеры ее использования.

Программа обучения собственной модели состоит из следующих шагов:

- Загрузка и предобработка датасета: Используется функция `load_dataset`, которая сначала получает список файлов изображений в заданном датасете, а затем применяет функцию `preprocess_image` к каждому файлу для чтения, обрезки, изменения размера и преобразования изображений в тензоры. Результирующий датасет разбивается на пакеты заданного размера;
- Создание модели GAN: В функции `create_gan_model` создается генератор и дискриминатор, две основные составляющие модели GAN. Генератор преобразует случайные шумовые входные данные в сгенерированные изображения, а дискриминатор классифицирует входные изображения как реальные или сгенерированные;
- Определение функций потерь: Функция `generator_loss` определяет потерю генератора, используя бинарную перекрестную энтропию между выходами генератора и целевыми значениями (все единицы). Функция `discriminator_loss` определяет потерю дискриминатора, суммируя бинарную перекрестную энтропию между реальными и сгенерированными выходами дискриминатора;
- Определение оптимизаторов: Используются два оптимизатора Adam для генератора и дискриминатора с заданной скоростью обучения. Adam



(adaptive moment estimation) - это алгоритм оптимизации, совмещающий принципы инерции MomentumSGD и адаптивного обновления параметров AdaGrad и его модификаций;

- Функция тренировки модели: Функция `train_step` выполняет один шаг обучения модели GAN. Внутри этой функции происходит прямой и обратный проходы через генератор и дискриминатор с использованием градиентного спуска. Затем вычисляются потери генератора и дискриминатора, обновляются градиенты и применяются к соответствующим оптимизаторам;
- Тренировка модели: Функция `train_model` выполняет цикл обучения модели GAN на заданном количестве эпох. В каждой эпохе обрабатываются все пакеты изображений из датасета, и для каждого пакета вызывается функция `train_step`. После каждой эпохи выводятся значения потерь генератора и дискриминатора;
- Сохранение модели: По завершении обучения модель генератора сохраняется по пути, указанном в параметре `SAVED_MODEL_PATH`.

**В шестом разделе** подробнее описывается процесс разработки генератора эффектов для картинок.

Для расширения функционала приложения, чтобы оно могло быть конкурентоспособным, на языке программирования Python версии 3.12 с использованием библиотеки компьютерного зрения OpenCV, а так же NumPy для работы с массивами больших объемов данных, были реализованы скрипты, которые генерируют определенные эффекты на картинке.

**В седьмом разделе** подробнее описывается процесс разработки мобильного приложения под операционную систему Android на движке Unreal Engine 5.

Unreal Engine 5 является мощным и универсальным инструментом для разработки мобильных приложений. Его возможности нативизации, малое потребление ресурсов и современные графические технологии делают его хорошим выбором для создания высококачественных приложений. Визуальный скриптовый язык Blueprint ускоряет процесс разработки, хотя и имеет свои ограничения, которые можно обойти с помощью языка C++. Эти особенности позволяют разработчикам создавать эффективные и производительные мобильные приложения, отвечающие современным требованиям рынка.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы был проанализирован обширный материал по генеративно-состязательной нейронным сетям, их обучению, созданию эффектов на основе библиотек компьютерного зрения, мобильной разработки, а также их интеграции между собой. В результате этого исследования было разработана генеративно-состязательная нейронная сеть и генераторы эффектов для преобразования фотографий, а также разработано мобильное приложение для операционной системы Android, которое позволяет пользователям взаимодействовать с нейронной сетью. Приложение поддерживает выполнение POST и GET запросов, обеспечивая удобный доступ к функционалу через мобильные устройства. Результаты работы были апробированы на студенческой конференции факультета. По результатам работы была подготовлена к публикации статья.

### **Основные источники информации:**

- 1 Компьютерное зрение: технологии, компании, тренды [Электронный ресурс]. — URL: <https://rdc.grfc.ru/2021/04/analytics-computer-vision/> (Дата обращения 12.05.2024). Загл. с экр. Яз. рус.
- 2 Компьютерное зрение: технологии, компании, тренды [Электронный ресурс]. — URL: <https://rdc.grfc.ru/2021/04/analytics-computer-vision/> (Дата обращения 12.05.2024). Загл. с экр. Яз. рус.
- 3 Лежнин М.С. ВИДЫ ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНЫХ СЕТЕЙ [Текст] – 2022. - № 1. – С. 178-179.
- 4 VC Unreal Engine 4 для инди и прототипирования. [Электронный ресурс]. — URL: <https://vc.ru/pixonic/51306-ue4-guide> (Дата обращения 21.12.2023). Загл. с экр. Яз. рус.
- 5 Python - язык программирования. [Электронный ресурс]. — URL: <https://cyberpunk-yazyk-programirovaniya/viewer> (Дата обращения 21.12.2023). Загл. с экр. Яз. рус.