Содержание

[1 Выбор и обоснование средств проектирования и реализации 2](#_Toc184317056)

[2 Проектирование архитектуры нейросети 3](#_Toc184317057)

[3 Проектирование набора данных 5](#_Toc184317058)

[4 Проектирование пользовательского интерфейса 6](#_Toc184317059)

1 Выбор и обоснование средств проектирования и реализации

При разработке данной программы особое внимание уделялось таким ключевым параметрам, как производительность и надежность. Эти факторы стали основой для формирования стека разработки, который был тщательно подобран для достижения оптимальных результатов.

В качестве языка программирования был выбран Python. Этот язык практически всегда используется для проектирования и обучения нейронных сетей благодаря своим высоким показателям производительности в решении задач машинного обучения и глубокого обучения. Python предлагает широкий спектр библиотек и инструментов, которые значительно упрощают процесс разработки и позволяют сосредоточиться на алгоритмах, а не на низкоуровневых аспектах программирования.

Для проектирования нейронных сетей была выбрана библиотека PyTorch. Она зарекомендовала себя как одна из самых популярных платформ в научных исследованиях благодаря своей гибкости и простоте использования.

Для создания генератора набора данных использовался язык программирования C++, так как он многократно быстрее python.

В качестве среды для разработки нейросети был выбран JupyterHub. Эта платформа позволяет создать облачный сервер для хранения данных и подключения вычислительных ресурсов, что особенно важно для работы с большими объемами информации. JupyterHub обеспечивает удобный доступ к проектам через веб-интерфейс, что делает его хорошим инструментом для совместной работы.

Для создания приложения с графическим интерфейсом, которое позволяет пользователю взаимодействовать с уже обученной моделью нейросети, использовалась среда Visual Studio Code. Этот легковесный редактор кода отличается простотой в использовании и предоставляет все необходимые инструменты для быстрой интеграции библиотек и создания проектов.

За отрисовку графики в приложении отвечает OpenGL. Этот инструмент был выбран из-за своей простоты по сравнению с Vulkan, что позволяет сосредоточиться на разработке функционала без лишних сложностей.

В качестве вычислительных ресурсов для обучения нейросети использовалась видеокарта с поддержкой технологии CUDA Nvidia RTX 3090Ti с 24 Гб видеопамятью.

2 Проектирование архитектуры нейросети

Так как необходимо решить задачу реконструкции структуры, нейросеть должна принимать на вход изображение среза реального материала. После чего модель должна построить этот материл в трехмерном пространстве.

Для решения этой задачи следует использовать генеративно-состязательную нейросеть. Архитектура нейросети представлена на рисунке 2.1.

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Автоматически созданное описание

Рисунок 2.1 – Архитектура нейросети

Процедура обучения генератора показана на рисунке 2.2.

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

Рисунок 2.2 – Процедура обучения генератора

Процедура обучения дискриминатора показана на рисунке 2.3.

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Автоматически созданное описание

Рисунок 2.3 – Процедура обучения дискриминатора

Обучение заключается в том, что на вход модели подается одно входное изображение реальной структуры вместе с несколькими срезами на определенном расстояние вглубь друг от друга.

Тогда дискриминатор будет определять поддельны ли несколько сгенерированных изображений по сравнению с одним входным. А также будет проверять насколько реально входное изображение по сравнению с целевыми.

3 Проектирование набора данных

Перед обучением нейросети необходимо составить датасет (набор данных) с учетом специфики архитектуры нейросети. На данном этапе могут возникнуть трудности, так как реальных данных обычно недостаточно, что и было в нашем случае. Поэтому были принято решение о разработке генератора данных, на основе метода вороного. Фактически необходимо повторить такой же результат генерации, как в программах рассмотренных в разделе анализа.

Архитектуру данной небольшой программы можно представить в нескольких шагах:

1. Создание трехмерного массива, который будет хранить цвет в каждой ячейке.
2. Создание точек в трехмерном пространстве со случайными координатами и некоторым цветом.
3. Генерация структуры Вороного.
   1. В трех циклах по высоте, ширине, глубине на каждой итерации рассчитывается ближайшая точка к текущей координате из тех ранее созданных точек (находится минимальное евклидово расстояние).
   2. В данной координате указывается тот цвет, который принадлежит ближайшей точки и записывается в трехмерный массив, помимо этого добавляется шум, чтобы сделать изображения более реалистичными.
4. Сохранение результата осуществляется следующим образом: составляется коллаж изображений из 2D срезов трехмерной структуры, где левая картинка — это срез на текущей итерации, а правая срез на следующей.

Таким образом с одной сгенерированной 3d структуры можно получить сотни изображений, и быстро составить датасет.

Далее формируются обучающая, тренировочная и валидационная выборка из сгенерированных 1000 изображений, каждая размером 512 на 256 пикселей, так как состоит из двух картинок. Структура датасета показана на рисунке 3.1.

Изображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Автоматически созданное описание

Рисунок 3.1 – Структура датасета

4 Проектирование пользовательского интерфейса

Интерфейс приложения состоит из нескольких блоков, все они располагаются на одной главной странице, которая открывается при запуске приложения. Иерархия экранов и элементов приложения представлена на рисунке 4.1.

Изображение выглядит как текст, диаграмма, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 4.1 – Иерархия экранов и элементов приложения

Подробное описание элементов приложения и их назначение представлено в таблице 4.1.

Таблица 4.1 – Назначение элементов приложения

|  |  |  |
| --- | --- | --- |
| № | Название | Описание |
| 1 | Панель инструментов | Панель содержащая два раздела: файл и помощь. Располагается в верхней части экрана приложения. Отвечает за сохранения и загрузку файлов. |
| 1.1 | Файл | Раздел со списком из двух функций: импорт и экспорт файлов. |
| 1.1.1 | Импорт | Кнопка. Предназначена для загрузки ранее сохраненного файла приложения. |
| 1.1.2 | Экспорт | Кнопка. Предназначена для сохранения файла текущей конфигурации приложения |
| 1.2 | Помощь | Раздел со списком из двух функций: присмотр руководства пользователя и просмотр информации о версии приложения. |

Продолжение таблицы 4.1 – Назначение элементов приложения

|  |  |  |
| --- | --- | --- |
| № | Название | Описание |
| 1.2.1 | Руководство пользователя | Кнопка. Необходима для открытия инструкций для пользователя. |
| 1.2.2 | О приложении | Кнопка. Необходима для открытия информации о текущей версии приложения. |
| 2 | Область визуализации | Область для отображения результатов моделирования программой, представляет из себя трёхмерную сцену. |
| 2.1 | Строка вывода сообщений | Текстовая строка. Предназначена для вывода сообщений о текущем прогрессе моделирования. Также в ней записывается содержание ошибок в случае их возникновения. |
| 2.2 | Сцена визуализации результатов генерации | Область для отрисовки графики. Необходима для визуализации результатов генерации. То есть для отображения в трехмерном пространстве созданного куба поликристаллического материала. |
| 3 | Панель настройки генерации | Панель содержащая два блока: выбор модели нейросети или иного метода, настройка параметров. |
| 3.1 | Выбор модели | Блок для выбора модели с помощью которой будет осуществляться моделирование. |
| 3.2 | Блок параметров | Блок с параметрами настройки нейросети или других методов, также содержит кнопки для запуска генерации. |

Дизайн приложения составлялся по аналогии с приложением Dream3D. Поэтому было выбрано синее цветовое решение. Макет экрана главной страницы показан на рисунке 4.2.

На данном экране в верхней части представлена панель инструментов с такими вспомогательными функциями как сохранение и загрузка конфигурации файла приложения, помощь. В средней части экрана находиться область, в которой визуализируются результаты моделирования. Также прогресс показывается в текстовом поле сверху. В нижней части экрана приложения находиться панель параметров настройки генерации, которая включает возможность выбора метода, загрузку как продольных так и поперечных срезов в формате изображения, либо в формате EBSD данных. Справа располагаются кнопки по запуску моделирования, а также сохранения сгенерированного результата в файл.

Изображение выглядит как снимок экрана, текст, Мультимедийное программное обеспечение, Операционная система

Автоматически созданное описание

Рисунок 4.2 – Макет экрана главной страницы