

Российская Робототехническая Олимпиада 2024

Творческая категория. Искусственный интеллект

**ОТЧЕТ по проекту «РОБОТИЗИРОВАННЫЙ КИОСК ДЛЯ СОЗДАНИЯ ОТКРЫТОК С ПРИМЕНЕНИЕМ ИИ»**

Команда «Художник»

**Выполнили:**

Дроканов Олег, 9 класс,

ГБОУ Президентский ФМЛ №239

Баранова Евгения, 8 класс,

ОДОД ГБОУ Президентский ФМЛ №239

**Руководитель:**

Иванов Василий Леонидович,

педагог дополнительного образования

ГБОУ Президентский ФМЛ №239

Санкт-Петербург

2024

­­Содержание

[1 Цель и задачи 3](#_Toc169909390)

[2 Команда 3](#_Toc169909391)

[3 Идея 5](#_Toc169909392)

[3.1 Практическая значимость 5](#_Toc169909393)

[3.2 Предпроектное исследование 5](#_Toc169909394)

[4 Этапы разработки проекта 9](#_Toc169909395)

[5 Роботизированное решение 11](#_Toc169909396)

[5.1 Проектирование и изготовление устройства 11](#_Toc169909397)

[5.2 Программное обеспечение 14](#_Toc169909398)

[5.3 Методы ИИ 15](#_Toc169909399)

[6 Результат 22](#_Toc169909400)

[7 Следующие шаги и бизнес-модель 22](#_Toc169909401)

[7.1 Актуальность услуги 23](#_Toc169909402)

[7.2 Канва бизнес-модели: 23](#_Toc169909403)

[8 Список источников 24](#_Toc169909404)

[9 Приложение 25](#_Toc169909405)

# Цель и задачи

Целью нашего проекта стало создать роботизированный киоск для создания открыток с использованием искусственного интеллекта. Для выполнения поставленной цели потребовалось выполнить следующие задачи:

1. провести предпроектное исследование
2. на основе исследования сформулировать идею проекта
3. спроектировать устройство и составить список материалов и компонентов
4. изготовить роботизированный киоск
5. разработать программное обеспечение с использованием ИИ
6. протестировать прототип в общественном месте
7. составить бизнес-план и канву бизнес-модели для выпуска реального прототипа роботизированного киоска

# Команда

Мы – команда из Центра робототехники физико-математического лицея №239, города Санкт-Петербург.

|  |  |
| --- | --- |
| Олег Дроканов | Евгения Баранова |
| Специалист по нейронным сетям, программное обеспечение | Разработка 3D-моделей, сборка плоттера |

# Идея

В современном мире ассортимент открыток в магазинах ограничивается фантазией дизайнеров и количеством наборов. Мы решили, что сделаем робототехническое устройство, которое сможет поддержать полет мысли любого человека и создать ему уникальный сувенир по мановению пальца.

В результате применения технологий искусственного интеллекта появился концепт роботизированного киоска по созданию открыток. Идея такова: человек рисует в воздухе рукой упрощённый эскиз и говорит, что он нарисовал, робот распознает движения и слова, затем нейронная сеть генерирует изображение на основе полученного наброска и темы рисунка, а плоттер рисует его на листе бумаги. В результате пользователь получает уникальную открытку на память. Такое роботизированное устройство позволяет раскрыть человеку свои эмоции через изобразительное искусство, даже если он не умеет рисовать.

# Практическая значимость

Наш робот имеет практическую значимость во многих сферах, например, развлечение посетителей в торговых центрах, в парках аттракционов, на ярмарках. Также можно его использовать в туристических целях: создавать уникальные памятные открытки, связанные с определенным местом, городом или страной. Роботизированный киоск можно установить и в учебных центрах, музеях, на выставках и конференциях, тогда он будет наглядным примером для популяризации робототехники среди детей и взрослых.

# Предпроектное исследование

Перед тем, как сформулировать окончательный вариант идеи проекта, мы изучили аналогичные устройства по двум направлениям: роботов-художников и различные цифровые киоски и автоматы представленные на рынке.

Первыми мы рассмотрели различных роботов-художников.

* + - 1. Роботы-художники, основанные на манипуляторах.

Робот-художник FRIDA[1].

Робот представляет собой манипулятор с кистью на конце. Он умеет воспроизводить картины по шаблону, анализировать эскиз и запрос пользователя. Совмещая полученные данные, робот генерирует изображение из отдельных мазков и воспроизводит его.

|  |  |
| --- | --- |
| Роборука FRIDA с ИИ создает произведения искусства в сотрудничестве с людьми  Рисунок - Робот FRIDA | https://pschaldenbrand.github.io/frida/assets/paintings/frida_bw_painting.png  Рисунок - Работа робота FRIDA |

Робот-художник TEKO[2].

Манипулятор компании TEKO анализирует полученное изображения с помощью искусственного интеллекта и воспроизводит его мазками кисти.

|  |  |
| --- | --- |
| Робот-художник для образовательных учреждений  Рисунок - Робот-художник TEKO | https://b2b.tsml.ru/upload/iblock/90e/781krwjs4ja4sd7c8sy0gshjc19vqubz.png  Рисунок - Картины робота |

* + - 1. Вторая разновидность роботов-художников – это координатные устройства

Bantam Tools NextDraw. Перьевой плоттер для письма и рисования[3]

Этот плоттер разработан для письма и рисования на различных поверхностях. Умеет писать заданный текст и рисовать заданное изображение.

|  |  |
| --- | --- |
| https://store.bantamtools.com/cdn/shop/files/1_43a09f41-c613-4f3f-acff-4dfd5776e1a2_1024x1024@2x.jpg?v=1714402527  Рисунок - Bantam Tools NextDraw | https://store.bantamtools.com/cdn/shop/files/8_82dbba11-e36b-4096-b246-50c348235383_1024x1024@2x.jpg?v=1714402522  Рисунок - Чертеж робота |

Робот-художник разработанный в МГТУ им. Н. Э. Баумана[4]

Это многофункциональный робот-художник, генерирующий свои изображения с помощью нейросети.

|  |  |
| --- | --- |
| Робот-художник  Рисунок - Робот-художник МГТУ им. Н. Э. Баумана | Иллюстрация  Рисунок - Рисунок робота-художника |

После этого, решив, что у нас будет воспроизведено координатное устройство, потому что:

Манипулятор сложно изготовить в условиях школьной лаборатории

Координатное устройство легче изготовить и программировать, его формат подходит к идее цифрового киоска и основной задумке.

Мы начали изучать разные кинематики 3D-принтеров, для того чтобы выбрать из них самую подходящую для нашей задачи. В основном мы рассматривали те, в которых печатающая головка двигалаcь по двум осям: X и Y, так как для нашей цели не нужно перемещение печатающей головки по оси Z. Были рассмотрены кинематики H-bot, Core-XY и Ultimaker.

1. Кинематика H-bot

Печатающая головка перемещается по осям за счет одного ремня и параллельной работы двух моторов.

|  |  |
| --- | --- |
| h-bot.jpg  Рисунок - H-bot | H-bot понять суть  Рисунок - Пример кинематики H-bot |

1. Кинематика Core-XY

Она очень похожа на H-bot, но имеет более сложное устройство. Она использует два ремня со сложной системой роликов. Она имеет большую точность и надежность, но сложна в реализации.

|  |  |
| --- | --- |
| h-bot.jpg  Рисунок - Core-XY | Mark Rehorst's Tech Topics: CoreXY Mechanism Layout and Belt Tensioning  Рисунок - Пример кинематики Core-XY |

1. Кинематика Ultimaker

В ней печатающая головка перемещается по направляющим валам, от качества и прямоты которых напрямую зависит качество ее передвижения. Она также имеет большую точность, но сложна в реализации и производстве.

|  |  |
| --- | --- |
| Устройство 3D принтеров, типы кинематики. Отличие дельта 3D принтера от  декартового - 3DRadar  Рисунок - Ultimaker | Обзор принтера с кинематикой от ultimaker.  Рисунок - Пример кинематики Ultimaker |

Среди всех кинематик мы выбрали H-bot, потому что она понятна, проста в реализации, имеет достаточную точность перемещений и отлично подходит для нашей задачи.

# Этапы разработки проекта

Проект был начат в сентябре 2023 года. Мы работали параллельно. Олег занимался разработкой программы для отслеживания руки, а Женя создавала 3D-модели и собирала плоттер. Поэтому наш проект можно разделить на два направления.

|  |  |  |
| --- | --- | --- |
| **Месяц** | **Конструкция и электроника** | **Программа** |
| **Сентябрь** | Поиск аналогов в Интернете | Тесты программы по распознаванию рук |
| **Октябрь** | Поиск и обработка основной 3D-модели плоттера | Обучение нейросети и встраивание её в программу |
| **Ноябрь** | Разработка модели каретки | Доработка программы рисования в воздухе |
| **Декабрь** | Разработка модели каретки | Настройка Nvidia Jetson, начало работы с выделением траектории (первичная обработка изображения) |
| **Январь** | Настройка прошивки, сборка каретки | Настройка Nvidia Jetson, выделение траектории, создание интерфейса |
| **Февраль** | Подбор микроконтроллера, тесты, доработка моделей | Перевод программы по выделению траектории на Cython, тесты Stable Diffusion |
| **Март** | Сборка всего устройства, тесты перемещения | Пробы новых методов обработки изображения, объединение всех программ |
| **Апрель** | Доработка плоттера и тестирование его работы | Создание программы по переводу траектории в команды для плоттера, тесты рисования |
| **Май** | Улучшение рисования | Доработка программы траектории |
| **Июнь** | Пробы разных вариантов маркера | Переписывание траектории на C++, добавление stdpar, улучшение интерфейса, оптимизация Stable Diffusion, другие доработки |

# Роботизированное решение

# Проектирование и изготовление устройства

Опираясь на результаты предпроектного исследования различных кинематик для нашего плоттера была взята модель 3D-принтера с кинематикой H-bot из открытого доступа**(ссылка)**. Из этой 3D-модели была удалена ось Z.

Далее нужно было создать модель собственной печатающей головки. Она состоит из двух 3мм-пластин из фанеры, скрепленных собой роликами и специальной деталью, напечатанной на 3D-принтере и крепится на конструкционный профиль 20х20. Эта деталь отвечает за крепление сервомотора. На одной из пластин находятся две направляющие MGN7C с ещё одной пластиной. В ней есть вырез для подшипника, который соединен с качалкой сервомотора. За счет этого она может двигаться вверх и вниз под действием мотора. На этой же пластине закреплен держатель маркера, напечатанный на 3D-принтере.

|  |  |
| --- | --- |
| Рисунок 15 - Каретка. Общий вид | Рисунок 16 - Каретка. Вид сверху |

Плоттер собран из алюминиевого конструкционного профиля 20х20 и скрепляющих его деталей. На двух его углах установлены металлические пластины, повышающие жесткость конструкции. На других 2 углах закреплены детали из фанеры толщиной 6мм, на которых крепятся шаговые моторы Nema17-42SHDC. Они обеспечивают высокую точность передвижения печатающей головки, а также достаточный крутящий момент.

Подвижность портала обеспечивается за счет двух фанерных деталей с роликами, толщиной 6мм.

|  |  |
| --- | --- |
| Рисунок - Плоттер. Общий вид | Рисунок - Плоттер. Вид сверху |

Все модели составляющих робототехнического устройства были объединены в одну сборку.

|  |  |
| --- | --- |
| Рисунок - Сборка. Общий вид | Рисунок - Сборка. Вид сверху |

Все этапы моделирования проводились в САПР Компас-3D. Подбор размеров всех деталей и комплектующих был произведен методами 3D-моделирования.

После того, как полная сборка была готова, мы заказали **конструкционные профили, крепеж, алюминиевые уголки, подшипники и**

А также, вырезали из фанеры на лазерном ЧПУ-станке и напечатаны на 3D-принтере остальные, нужные нам, детали.

**(красивая фотка робота)**

Для антуража и удобства перемещения робототехнического устройства нами было принято решение поместить его в старый чемодан. На его верхней крышке установлен монитор (диагональ 23.8'') с укрепленной на нем камерой Logitech HD 1080p. Внутри чемодана располагается корпус компьютера с процессором Intel Core i5-8400, 16 GB оперативной памяти и NVIDIA GeForce GTX 1070 с 8 GB видеопамяти.

**(фотка внутренностей компа)**

Провода от сервомотора проведены через шлейф внутри кабель-канала. Вместе с шаговыми моторами, они подключены к плате Arduino Mega с расширением Protoneer CNC Shield V3. Так как в дальнейшем планируется подключение большого количества датчиков, был выбран именно этот микроконтроллер. В него загружена прошивка Marlin для 3D-принтеров. Она проста в конфигурации и использовании, поэтому выбрана для проекта. Команды в формате G-code отправляются с компьютера на плату по последовательному интерфейсу.

**(схема)**

# Программное обеспечение

Сначала был протестирован код по выделению опорных точек руки.

Затем к нему было добавлено отслеживание координат указательного пальца и передвижение курсора компьютера по этим координатам. Поначалу все это тестировалось в программе Paint.

Тогда появилась идея о добавлении распознавания жестов для управления интерфейсом. Был создан датасет для обучения нейросети и подобрана оптимальная архитектура. Обученная модель была встроена в основной код.

После этого были проведены тесты генеративной нейросети Stable Diffusion. В процессе была выявлена необходимость прямой передачи нарисованного изображения в нейросеть. Так как Paint не предполагал возможности извлечения рисунка из поля рисования программным методом, было решено создать свой интерфейс рисования. Была использована библиотека Tkinter.

После того, как был написан код интерфейса, встал вопрос о вычислительных ресурсах. Генеративная нейросеть для большей скорости обработки требовала мощный графический процессор с как минимум 8 гб видеопамяти. Было два варианта решить эту проблему: либо встроить в робототехническое устройство полноценный компьютер и к нему подсоединить видеокарту достаточной мощности, либо использовать высокопроизводительный модуль NVIDIA Jetson. Первый вариант достаточно простой в реализации, так как все используемое в проекте ПО рассчитано для запуска на обычном компьютере, однако необходимые комплектующие (материнская плата, блок питания и т.п.) занимают очень много места. А второй отличается компактностью, но имеет минус в виде своей специфичной архитектуры процессора и видеомодуля, из-за чего некоторые библиотеки могут на нем не работать так же, как и на обычном ПК.

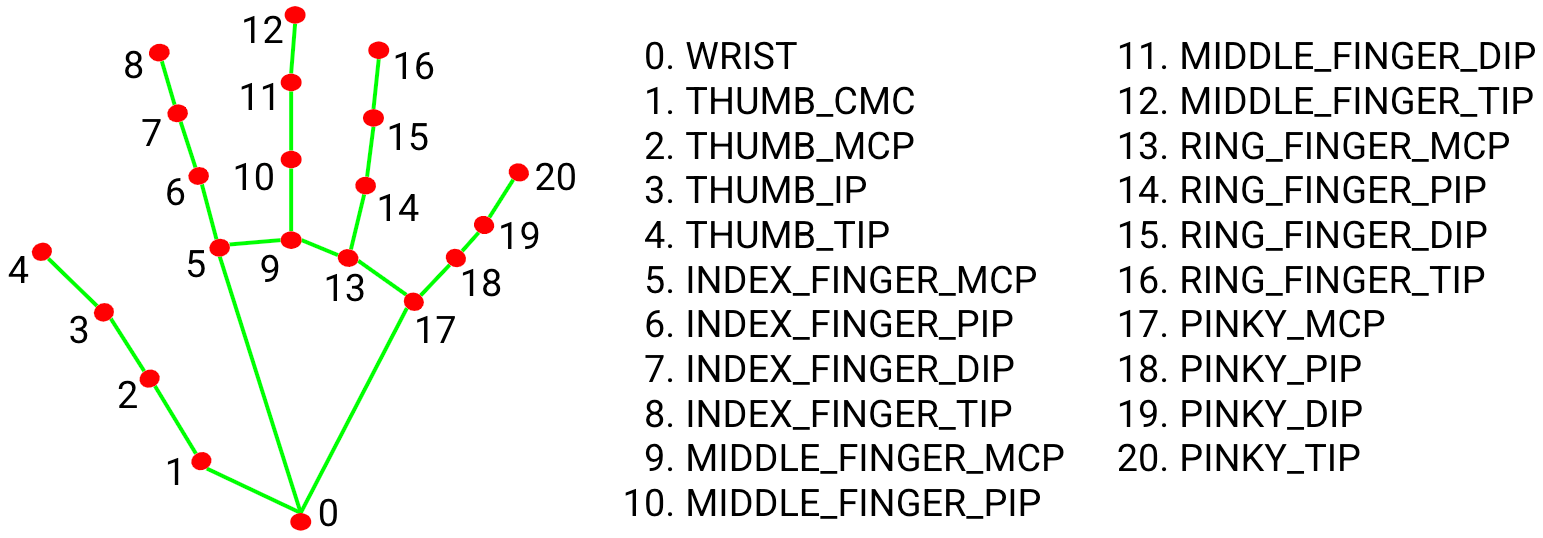
Сначала был выбран вариант с Jetson. Для тестов был взят Jetson Nano, который уже был в наличии. Было потрачено много времени на подготовку библиотек, но в результате выяснилось, что 4 гигабайт видеопамяти, которые он имеет, недостаточно для запуска Stable Diffusion даже с оптимизациями и на минимальном разрешении. Другие же модели Jetson, на которых возможно было запустить нейросеть, имели очень большую цену, поэтому мы решили перейти к другому решению.

В результате были собраны компьютер, камера и дисплей. После этого установить все необходимые библиотеки и запустить на нем весь код не составило труда.

# Методы ИИ

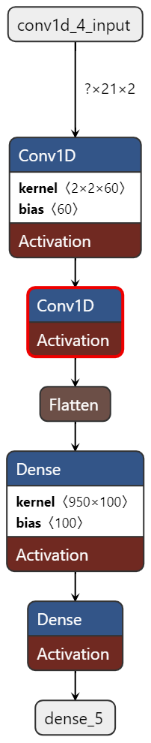
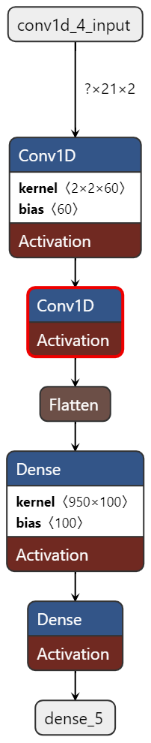
Весь код проекта написан на языке Python, за исключением программы по переводу изображения в траекторию плоттера.

Выделение ключевых точек рук реализовано при помощи библиотеки mediapipe. Далее собственная нейросеть распознает жест, который показывает рука, на основе набора из 21 координат этих точек.



В создании классификатора использовались библиотеки Tensorflow и Keras.

Рисунок - Архитектура классификатора



Датасет для неё был собран следующим образом: сначала было отснято на камеру несколько видео с показанными жестами, затем из каждого кадра были выделены ключевые точки руки и записаны в файл. Модель была обучена на 20 эпохах.

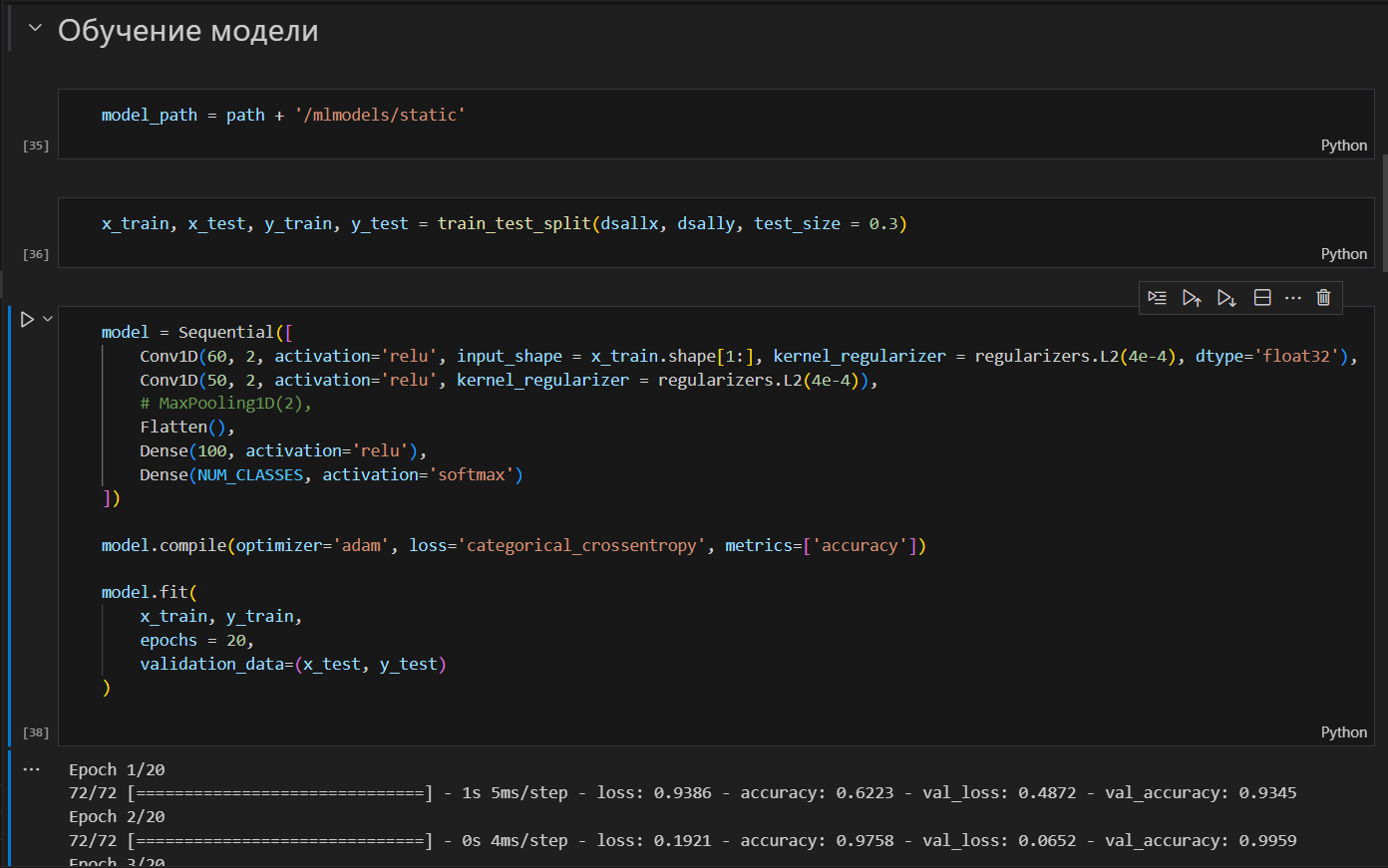


Рисунок - Модель

Для сокращения времени распознавания этой моделью она была оптимизирована при помощи TensorRT SDK.

Модель может распознавать 3 жеста: поднятый большой палец, раскрытую ладонь и поднятый указательный палец.

|  |  |  |
| --- | --- | --- |
| Рисунок - Большой палец | Рисунок - Раскрытая ладонь | Рисунок - Указательный палец |

На последовательности этих жестов реализован конечный автомат основной программы. Его последовательная схема такова:

1. Ожидание показанного большого пальца.
2. Если жест – поднятый указательный палец, то курсор перемещается в координаты подушечки этого пальца.
3. Если большой и указательный палец соединены вместе (это определяется по отношению расстояния между ними к расстоянию от запястья до кончика указательного пальца), то имитируется «перетаскивание» мышки в координаты указательного пальца.
4. Если показан второй раз большой палец – рисование прекращается, ожидается голосовой ввод подсказки к генерации.
5. Когда голосовой ввод окончен, начинается генерация изображения
6. Рисуем полученное изображение через плоттер и возвращаемся к пункту 1.

Голосовой ввод использует библиотеки google-trans и SpeechRecognition. Сначала считывается речь на русском языке, затем переводится на английский и отправляется в качестве подсказки в Stable Diffusion.

Управление генеративной нейросетью реализовано при помощи библиотеки Diffusers. Используются веса моделей Stable Diffusion 1.5 и ControlNet-Scribble, находящиеся в открытом доступе.

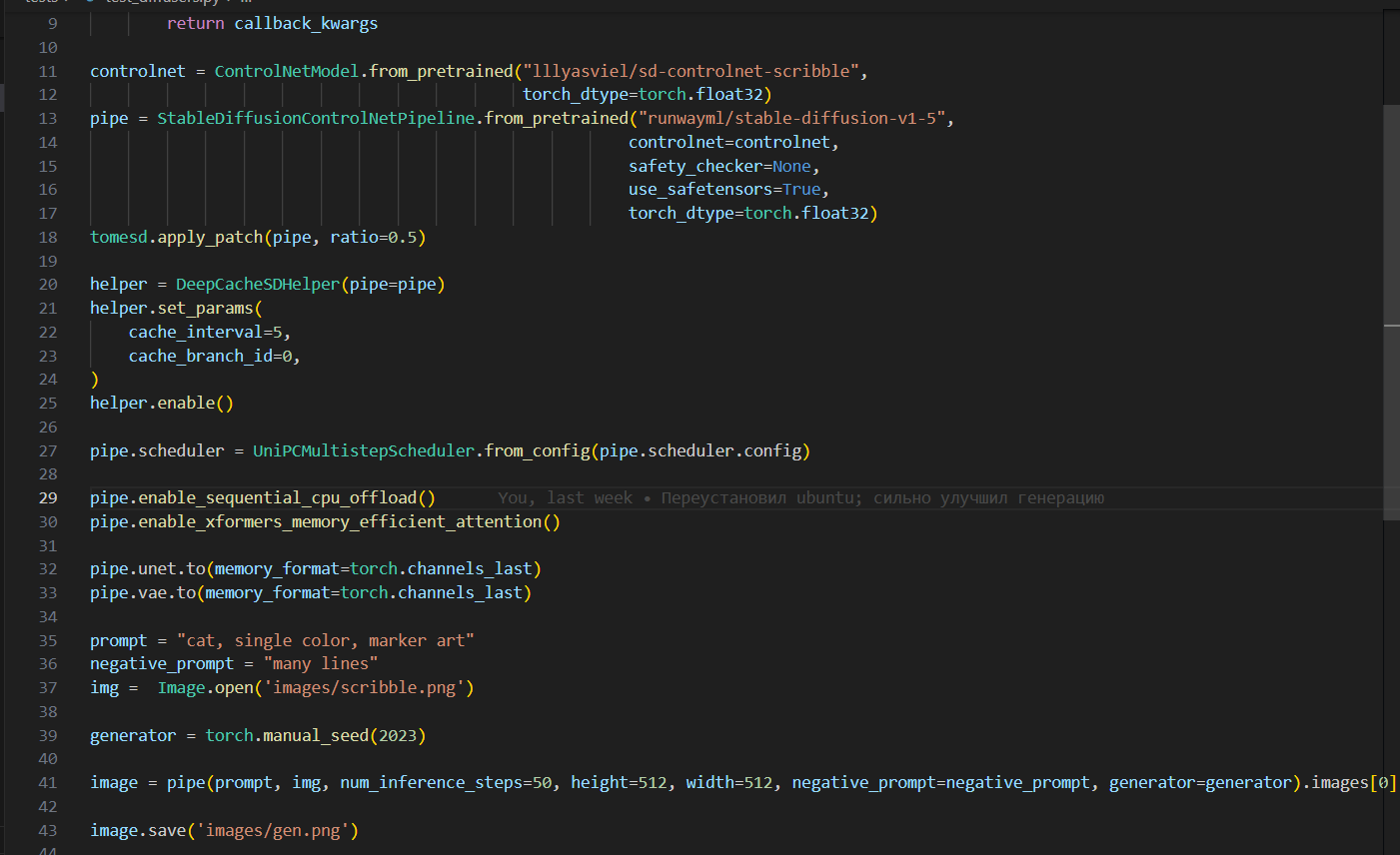


Рисунок - Управление нейросетью

При помощи библиотеки CUDA все модели запускаются на графическом процессоре, что ускоряет их работу.

Также была создана программа перевода из изображения в команды для плоттера. Принцип его работы следующий. Сначала из 3 цветовых каналов изображения выбирается тот, значения пикселей которого имеют наибольший разброс (дисперсию). Затем по этому черно-белому изображению из картинки выделяются контуры фильтром canny из библиотеки scikit-image. После этого к нему применяется бинарная диляция, то есть утолщение всех линий. Далее действует алгоритм, написанный на языке С++:

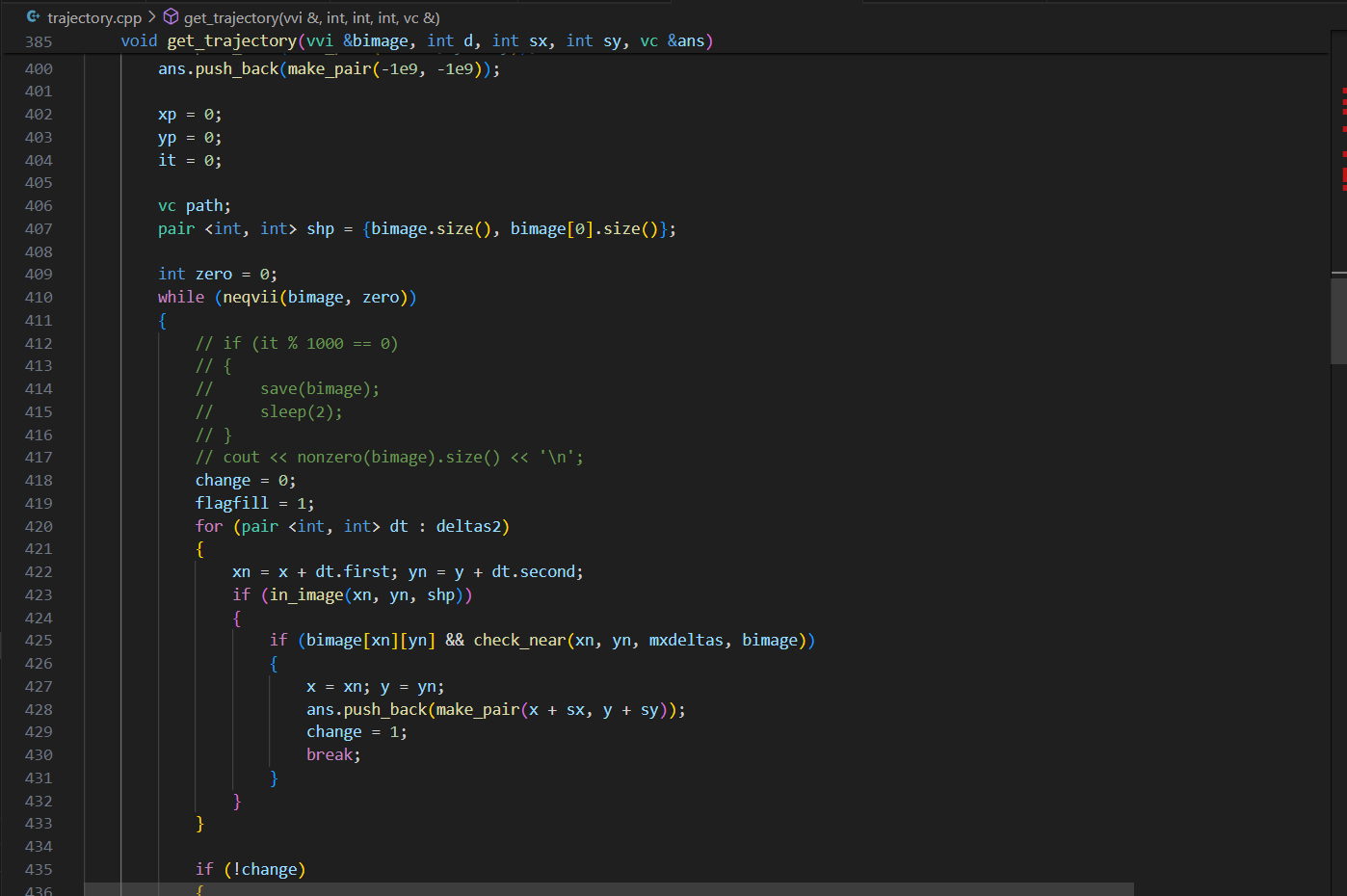
* + - 1. Выбирается случайная точка, вокруг которой в радиусе d/2 (d – величина, характеризующая диаметр кончика маркера) нет ни одной закрашенной или не входящей в рисунок точки.
      2. Вычисляются все точки, которые принадлежат пути из предыдущего пикселя в текущий (если это первая итерация цикла, то закрашиваются точки вокруг найденного пикселя), а затем для каждой из них все пиксели, принадлежащие окружности с центром в этой точке и диаметром d, закрашиваются.
      3. Координаты точки добавляются в ответ.
      4. Далее ищется точка, находящаяся на расстоянии d + 1 и удовлетворяющая условию из пункта 1.
      5. Если такой не найдено, то выбирается случайная точка изображения, удовлетворяющая этому условию.
      6. Если всё ещё не существует подходящей точки, то условие смягчается: достаточно, чтобы все пиксели в радиусе d/2 не выходили за рамки изначальной области, которую нужно закрасить.
      7. Когда получены координаты точки, переходим к пункту 2.
      8. Цикл повторяется, пока все пиксели не закрашены.

Рисунок - Перевод изображения

Так как точек обрабатывается очень много, даже C++, являющийся довольно быстрым языком программирования, справлялся недостаточно быстро. Поэтому некоторые циклы и функции были переписаны с использованием stdpar. Это функция стандарта C++17, которая позволяет запускать вычисления в параллельном режиме. А так как графический процессор в основном специализируется как раз на параллельной обработке, при помощи Nvidia HPC SDK, а в частности, компилятора nvc++, код стал запускаться на GPU. Благодаря этому производительность увеличивается в десятки раз по сравнению с чистым Python или C++.

Для реализации взаимодействия кода на двух языках программирования была использована библиотека ctypes.

Интерфейс взаимодействия с пользователем написан с использованием библиотеки tkinter. В нём есть

* окно предпросмотра изображения с камеры
* кнопка очистки поля рисования
* индикатор того, рисует пользователь или нет
* строка прогресса генерации изображения
* поле для распознанного из голоса текста
* кнопка изменения толщины и всплывающее окно с вариантами толщины линии
* кнопка для завершения генерации
* поле для рисования

|  |
| --- |
| Рисунок - Код интерфейса |
| H:\Downloads\изображение.png  Рисунок - Интерфейс |

# Результат

В результате у нас получился роботизированный киоск для создания открыток с использованием искусственного интеллекта **(общая фотка)**

# Следующие шаги и бизнес-модель

В дальнейшем в проекте планируются следующие шаги и доработки:

1. Реализация замены цвета маркера и рисование цветных изображений, а именно:
   1. спроектировать и установить в робота систему замены маркеров
   2. разработать ПО для разложения изображения на цвета и последующей обработки полученных цветовых пятен.
2. Опускание маркера на необходимое расстояние на основании показаний датчиков BL Tuch и оптического концевикаю
3. Спроектировать и разработать корпус для электронных компонентов из фанерных деталей.
4. Разобрать монитор и вмонтировать его матрицу в крышку чемодана и закрепить все комплектующие внутри корпуса
5. Реализовать обратную связь: устройство перестает рисовать, если пользователь протянул руку к листку

# Актуальность услуги

Роботизированный киоск точно не оставит равнодушным туристов и посетителей различных заведений. Людей привлекает возможность создать персональный сувенир памятную открытку и поработать в паре с искусственным интеллектом. Бумажная открытка – классический презент родственникам и детям, который легко отправить по почте или положить к книгам и документам. При этом, процесс рисования изображения роботом дарит огромное количество ярких эмоций.

# Канва бизнес-модели:

|  |  |
| --- | --- |
| **Ключевые партнеры** | Торговые центры, музеи, выставки |
| **Ключевые действия** | Рисование заданного пользователем объекта |
| **Ценностное предложение** | Развлечение и образование |
| **Взаимоотношения с клиентами** | Обратная связь с клиентами |
| **Сегменты клиентов** | Посетители центров или мероприятий |
| **Ключевые ресурсы** | Роботизированный киоск |
| **Каналы** | Сайт, приложение**,** социальные сети |
| **Потоки доходов** | Прибыль от продажи услуги |

# Список источников

1. <https://pschaldenbrand.github.io/frida/>
2. <https://b2b.tsml.ru/solutions/dopolnitelnoe-obrazovanie/robot-khudozhnik/>
3. <https://translated.turbopages.org/proxy_u/en-ru.ru.8378074e-66743484-53bb72b0-74722d776562/https/store.bantamtools.com/products/bantam-tools-nextdraw-8511>
4. <https://iz.ru/1610467/ivan-chernousov/sovremennoe-iiskusstvo-v-rf-sozdali-mnogofunktcionalnogo-robota-khudozhnika>
5. <https://3d-diy.ru/wiki/3d-printery/raznovidnosti-kinematik-fdm-3d-printerov/>
6. https://hf.ru/c/business\_avtomaty\_suvenirnykh\_monet

# Приложение



https://github.com/olegg366/Artist/