ВСЕРОССИЙСКИЙ КОНКУРС НАУЧНО-ТЕХНОЛОГИЧЕСКИХ ПРОЕКТОВ «БОЛЬШИЕ ВЫЗОВЫ 2024/2025»

НАПРАВЛЕНИЕ «Большие данные, искусственный интеллект, финансовые технологии и машинное обучение»

«Создание высокоточной модели для распознавания жестов рук через веб-камеру и управления курсором мышки на экране»

научно-исследовательский проект

Автор работы:
Парьев Максим Алексеевич,
Г. Новосибирск
МАОУ Лицей №9, 7И

Руководитель:
Ильина Наталья Анатольевна,
учитель информатики,
МАОУ Лицей №9

Новосибирск, 2025

Оглавление

Введение

В последние годы наблюдается стремительный рост интереса к технологиям, связанным с распознаванием жестов и взаимодействием человека с компьютером. Одним из наиболее перспективных направлений является использование жестов рук для управления различными устройствами и приложениями. Это открывает новые горизонты для создания интуитивно понятных интерфейсов и улучшения взаимодействия пользователя с цифровыми системами.

Данный проект направлен на исследование возможности создания высокоточной модели для распознавания жестов рук с использованием веб-камеры. Я ставлю перед собой несколько ключевых целей, которые помогут не только разработать функциональное решение, но и заложить основу для дальнейших исследований и практического применения технологии.

Кроме того, в рамках проекта я исследую возможности его дальнейшего развития и применения в реальной жизни. Потенциальные области применения включают в себя управление мультимедийными системами, игры, а также помощь людям с ограниченными возможностями.

Таким образом, мой проект представляет собой многообещающее направление в области распознавания жестов, и я уверен, что его результаты могут внести значительный вклад в развитие технологий взаимодействия человека и машины.

- **В ходе работы над проектом будет проверяться следующая гипотеза**:
  - Можно обучить модель ИИ для распознавания жестов рук и написать программу, которая сможет интерпретировать результаты работы модели и управлять курсором мыши на экране ПК.
- **Проект будет решать следующую проблему**:
  - В современном мире все чаще требуется управление элементами виртуального мира без физического контакта. Использование традиционных устройств ввода, таких как мыши и клавиатуры, может быть неудобным и ограничивающим, особенно для людей с ограниченными возможностями. Технологии, основанные на обработке видео и распознавании жестов, позволяют пользователям взаимодействовать с компьютерами и другими устройствами более естественным и интуитивным способом, что открывает новые возможности для обучения, игры и работы. Это не только упрощает использование технологий, но и способствует созданию более инклюзивной и доступной среды для всех пользователей.

Актуальность

1. **Упрощение взаимодействия**:

   Использование жестов для управления курсором может значительно упростить работу с устройствами, особенно в ситуациях, когда традиционные устройства ввода, такие как мыши или клавиатуры, не удобны или недоступны.

2. **Технологические инновации**:

   Проект стимулирует развитие технологий распознавания жестов, что может привести к улучшению существующих систем и созданию новых, более сложных интерфейсов.

3. **Виртуальная и дополненная реальность**:

   Управление курсором жестами имеет особое значение в контексте VR и AR, где традиционное управление не всегда эффективно. Это открывает новые возможности для взаимодействия с виртуальными мирами.

Цели проекта

- Провести исследование возможности создания высокоточной модели для распознавания жестов рук через веб-камеру;
- Выбрать подходящую для проекта модель;
- Обучить модель на уникальном датасете, состоящем из фото рук с разными жестами;
- Разработать программу управления курсором мышки с помощью рук на языке Python;
- Исследовать возможность развития проекта и применения его в реальной жизни.

Задачи проекта

- **Сбор и подготовка данных**:
  - Собрать датасет из множество самодельных фото;
  - Разметить датасет;
  - Подготовить датасет к обучению;
- **Обучение модели**:
  - Обучить модель YOLO на подготовленных данных, оптимизируя параметры модели для достижения наилучшей точности;
- **Тестирование модели:**
  - Протестировать модели и узнать ее точность.
- **Разработка программы для управления мышкой;**
- **Оптимизация производительности**.

Теоретическая часть

Существующие технологии для работы с моделями искусственного интеллекта, направленные на распознавание объектов на фотографиях, охватывают несколько ключевых областей и методов. Вот основные из них:

Глубокое обучение (Deep Learning)

Глубокое обучение является основным подходом для задач распознавания объектов. Оно использует многослойные нейронные сети, которые могут автоматически извлекать признаки из изображений. Основные архитектуры, используемые для распознавания объектов, включают:

- **Сверточные нейронные сети (Convolutional Neural Networks, CNN)**: Специально разработаны для обработки изображений. CNN используют свертки для выделения признаков и имеют слои подвыборки для уменьшения размерности.
- **Модели трансформеров**: Новые архитектуры, такие как Vision Transformers (ViT), применяются для задач компьютерного зрения, включая распознавание объектов.

Алгоритмы и архитектуры для распознавания объектов

Существует множество алгоритмов и архитектур, специально разработанных для распознавания объектов:

- **YOLO (You Only Look Once)**: Алгоритм, который выполняет детекцию объектов в реальном времени, обрабатывая изображение за один проход через сеть.
- **Faster R-CNN**: Комбинирует региональные предложения и сверточные нейронные сети для более точного распознавания объектов.
- **SSD (Single Shot MultiBox Detector)**: Модель, которая также выполняет распознавание объектов за один проход, обеспечивая баланс между скоростью и точностью.
- **RetinaNet**: Использует фокусировку на сложных примерах для улучшения точности распознавания объектов, особенно в задачах с несбалансированными классами.

Обработка естественного языка (NLP) для аннотации изображений

Для улучшения распознавания объектов можно использовать сочетание ИИ для обработки изображений и текстовой информации. Например, модели могут генерировать описания изображений, что помогает в задачах аннотирования и поиска.

Компьютерное зрение (Computer Vision)

Компьютерное зрение включает в себя различные методы и технологии для анализа изображений и видео. К ним относятся:

- **Обнаружение границ**: Использование фильтров (например, Canny) для выделения границ объектов на изображении.
- **Сегментация изображений**: Разделение изображения на разные сегменты для более точного определения границ объектов.
- **Оптическое распознавание символов (OCR)**: Используется для распознавания текста на изображениях, что может быть полезно в приложениях, связанных с обработкой документов.

Фреймворки и инструменты

Существует множество фреймворков и библиотек, которые упрощают разработку и развертывание моделей для распознавания объектов:

- **TensorFlow и Keras**: Предоставляют инструменты для создания и обучения нейронных сетей, включая предобученные модели для распознавания объектов.
- **PyTorch**: Популярная библиотека для глубокого обучения, которая поддерживает множество архитектур для компьютерного зрения.
- **OpenCV**: Библиотека для компьютерного зрения, которая предлагает инструменты для обработки изображений и видео, включая функции для распознавания объектов.
- **Detectron2**: Платформа от Facebook AI Research для разработки и обучения моделей для распознавания объектов и сегментации изображений.

Облачные платформы и API

Многие облачные платформы предлагают готовые решения и API для распознавания объектов:

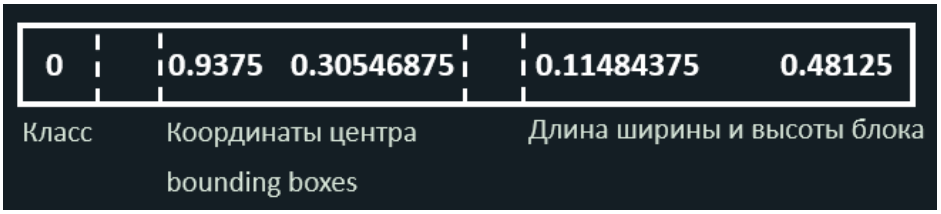
- **Google Cloud Vision**: Обеспечивает API для анализа изображений, включая распознавание объектов и текстов.
- **Amazon Rekognition**: Сервис от AWS, который позволяет распознавать объекты, сцены и текст на изображениях.
- **Microsoft Azure Computer Vision**: Предоставляет API для анализа изображений, включая распознавание объектов и описание содержимого.

Практическая часть

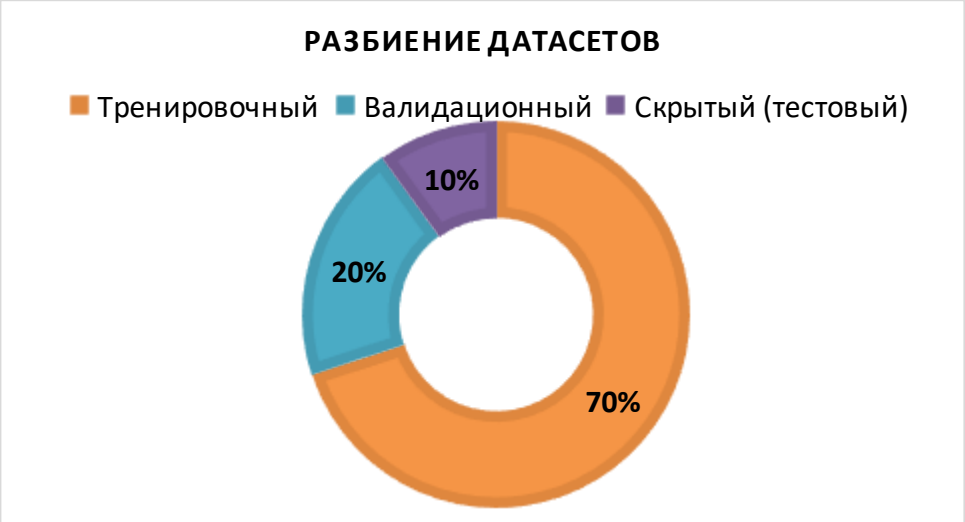
Сбор и подготовка данных

Подготовка данных состояла из следующих этапов:

1. Подготовка большого количества изображений (фото) с рукой;
2. Разметка всех изображений в программе LabelImg.
3. Формирование XML файл для каждого фото с координатами прямоугольника с рукой.
4. Преобразование координат на Python, их нормализация и подготовка TXT файлов с координатами



5. Разбиение датасета на части:



6. Пример размеченного файла

7. Принцип разметки:

Выбор модели для обучения

Рассматривалось несколько вариантов модели для обучения.

Требования к модели:

- Быстрое обучение для оперативных проверок изменений разметки датасетов и их корректировки;
- Быстрая обработка картинок для того, чтобы обеспечить минимальные задержки в обработке при захвате видео и определения координаты мышки.

Варианты создания модели:

- Самостоятельное создание модели и ее обучение
- Использование готовой модели YOLO/ ROBOFLOW

В конечном итоге, я остановился на модели YOLO, развернутой на локальном компьютере, т.к. она обеспечила и быструю обучаемость, и минимальное количество ошибок, и быструю обработку видео.

Обучение модели

Обучение модели состояло из следующих этапов:

1. Выбор оптимального количества эпох

   Выбрано 400 эпох – это оптимальное количество.

   Меньше – будет недостаточная точность

   Больше – может возникнуть эффект «переобучения модели»

2. Выбор координат разметки

   При первых попытка разметки на «левый верхний + правый нижний угол» точность была небольшая (около 40%)

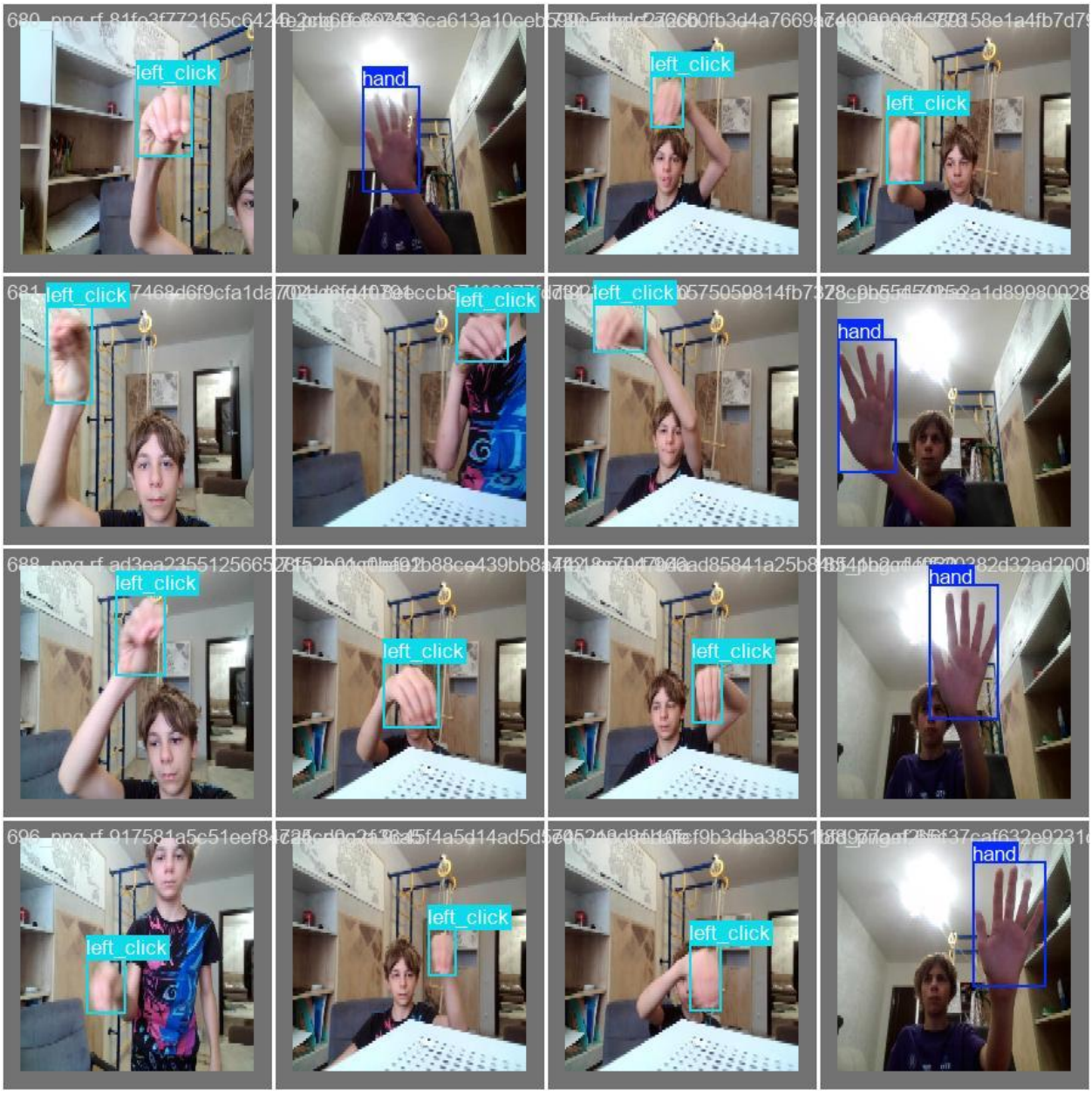
   При изменений координат на «левый нижний + правый верхний угол» точность возросла до >98%

3. Запуск обучения

   Для запуска обучения была создана программа на Python, обучения было с использованием библиотеки yolo

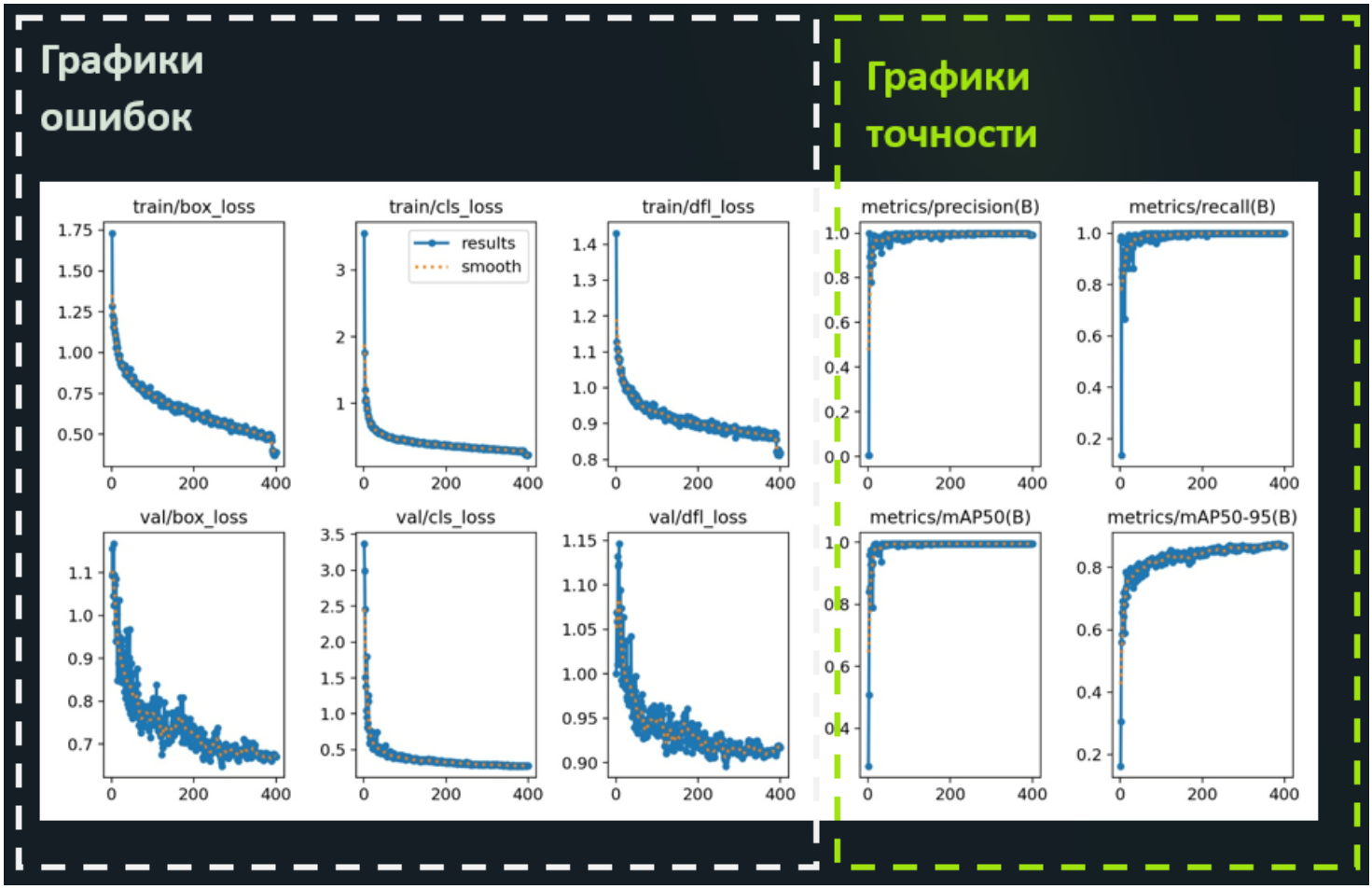
Размеченные изображения на тестовой выборке:

На примере видно, как модель определяет на изображении нужные классы – hand (для перемещения мыши) и left_click (для нажатия и удержания левой кнопки мыши).



Графики результатов обучения:

(видно, как во время обучения падает ошибка и возрастает точность модели)



Остальные результаты обучения модели приложены в Приложении 1.

Преобразование видео

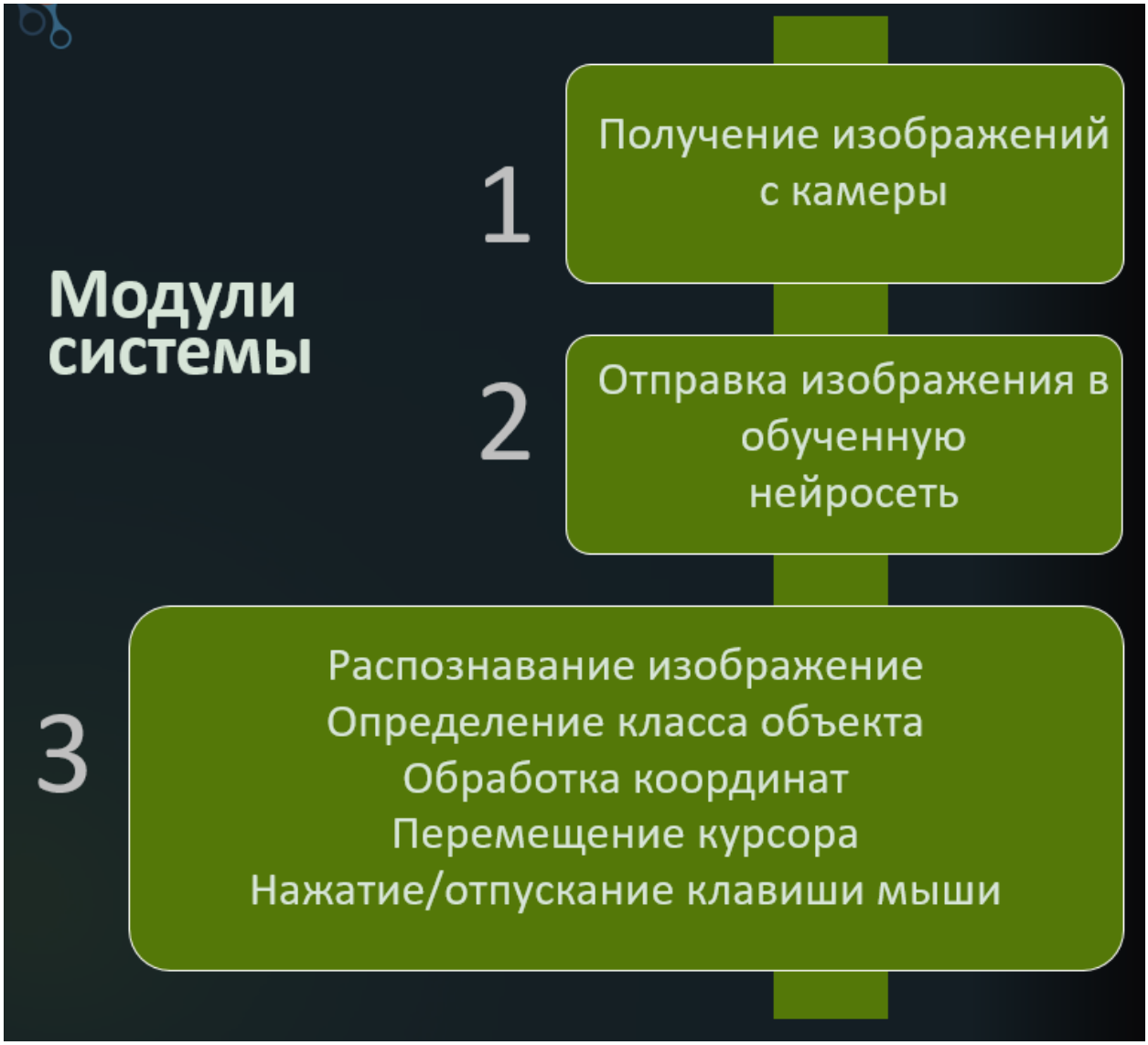
Для обработки изображений в модели нужно было разбить видео с камеры на отдельные кадры. Для этого выполнил следующие действия:

1. Написал код на Python для разбиения видео на отдельные кадры-изображения
2. Для работы с видео использовал библиотеку cv2
3. В программе запущен бесконечный цикл для считывания видео и кадрирования

Сборка проекта

Когда отдельные модуль проекта были готовы, я собрал их в единую программу. Модули системы являются последовательно вызываемыми, независимыми сервисами.

Могут дорабатываться автономно друг от друга, что позволит в будущем менять модель, менять обработку изображений и т.д. без влияния на остальные элементы программы.



Запуск и проверка

После завершения всех сборок, я запустил проект и проверил как он работает. Для фиксации результата я снял видео тестов проекта.

На видео видно, как программа распознает на кадрах видео руку, обводит ее в рамку и указывает найденный класс (hand). При появлении на изображении сомкнутой руки, модель ее также распознает и относит к другому классу (left_click).

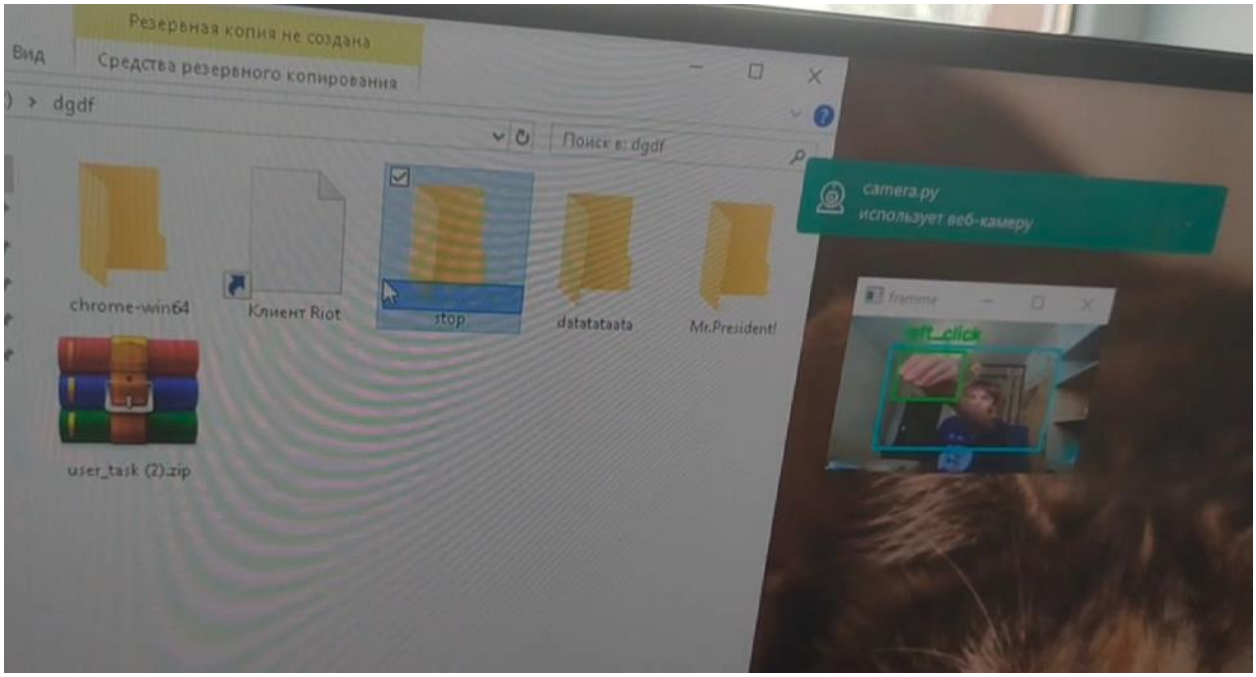
Далее рассчитываются координаты центра прямоугольника, в который вписана рука и туда перемещается курсор мыши.

Также видно, что задержки видео практически нет, курсор следует за рукой, даже при быстром движении руки.

Помимо этого, в нужный момент происходит нажатие на левую кнопку мыши и удержание ее в таком положении, пока рука на экране сомкнута. Такое состояние позволяет удерживать длительно удерживать объект на экране и перемещать курсор вместе с объектом.

В примере я показываю, как собираю объекты-файлы в папку с помощью своей программы, показываю пример выделения объектов и их перемещение.

Стоп-кадр видео (полное видео работы размещено на диске https://disk.yandex.ru/i/gT2zj9MRpGJMcQ):



Планы по развитию проекта

1. Оформить проект на GitHub и выложить свой код;
2. Создать свою модель и обучить ее по примеру YOLO;
3. Подумать над применимостью проекта в реальной жизни.

Выводы

1. Цели и задачи проекта достигнуты. Исследования в рамках проекта завершились положительно – получилось создать высокоточную модели для распознавания жестов рук через веб-камеру и управления курсором мышки с на экране;
2. Значительно увеличились знания по моделям, принципам их обучения и использования в Python;
3. Есть четкое понимание развития проекта в будущем.

Источники

1. **Ultralytics YOLO Docs**

   https://docs.ultralytics.com/ru/yolov5/tutorials/train_custom_data/#22-create-labels

2. **HABR**

   https://habr.com/ru/articles/821971/

3. **GitHub**

   https://github.com/boppreh/mouse#api

4. **YOUTUBE**

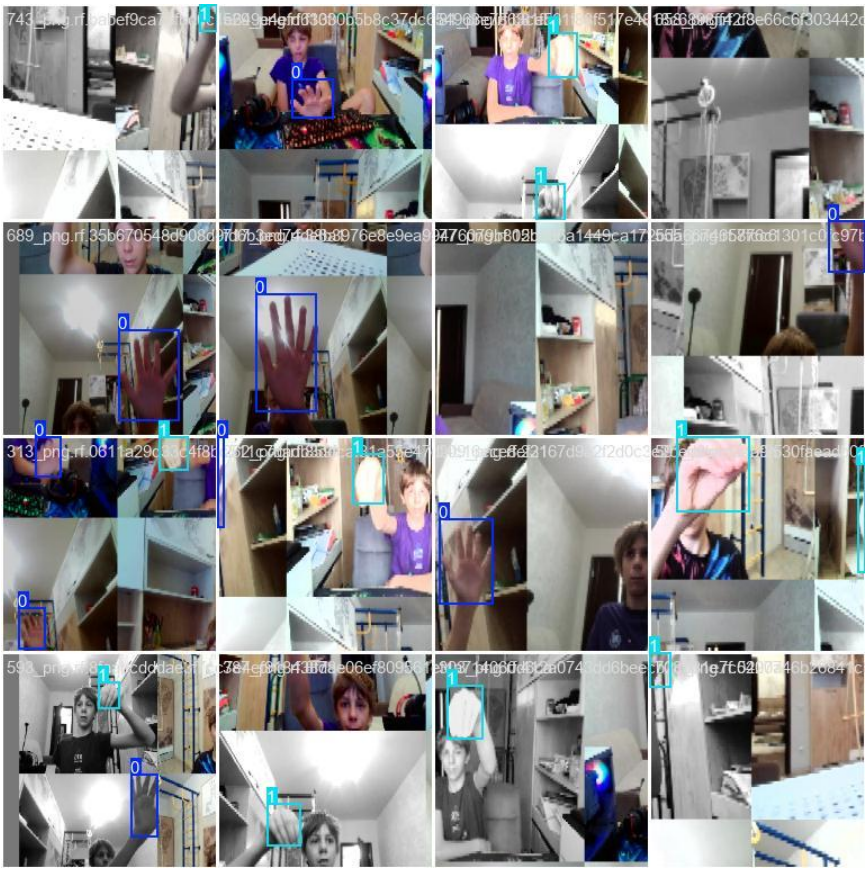
   https://www.youtube.com/

5. **Алгоритмы и методы обучения нейросетей**

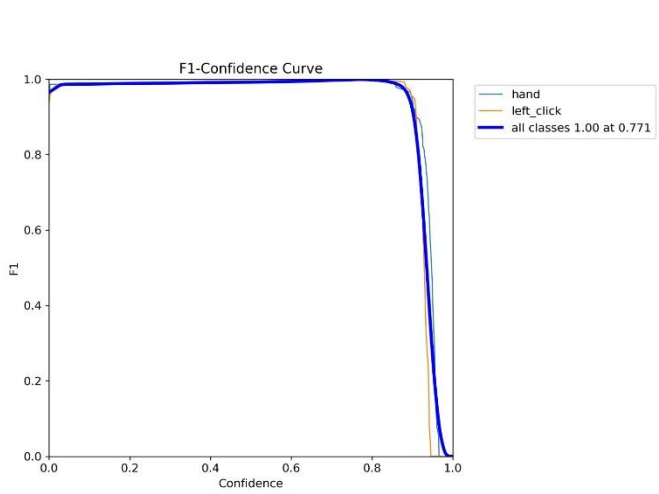
   https://data-light.ru/blog/obucheniye-nejrosetej

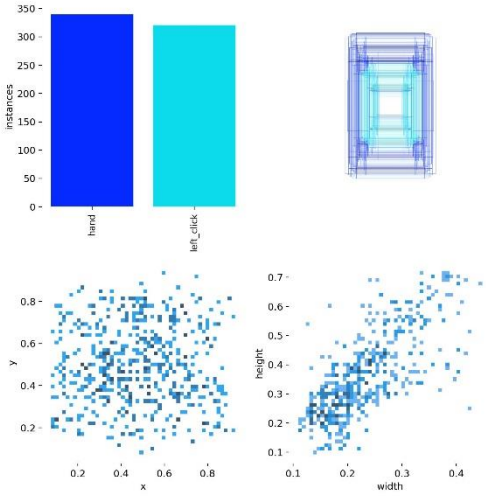
Приложение 1.

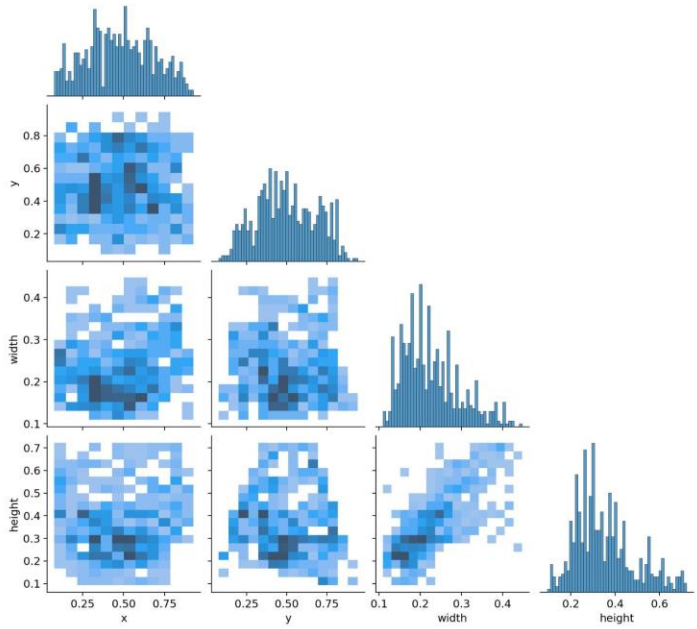
Изображения с разметками классов:

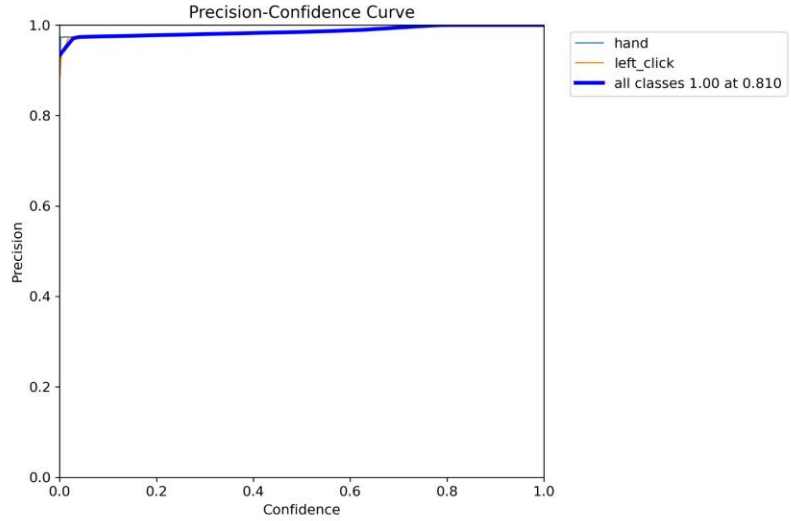


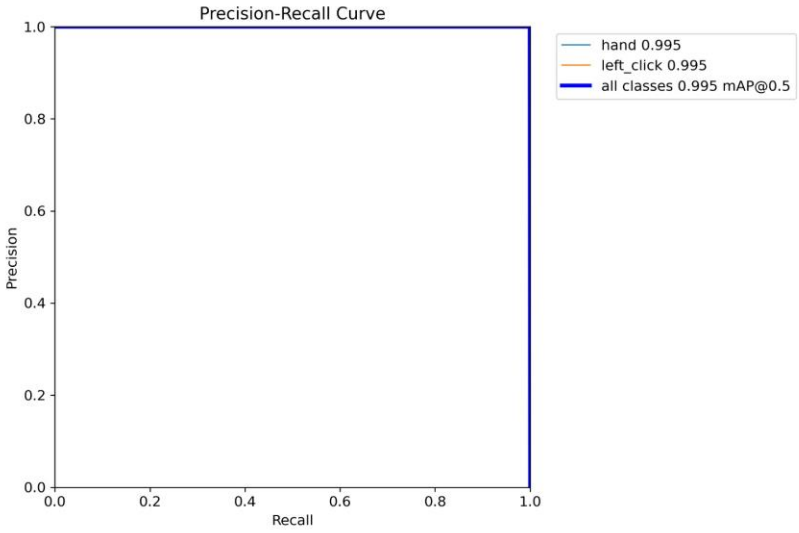
Графики с результатами обучения модели:

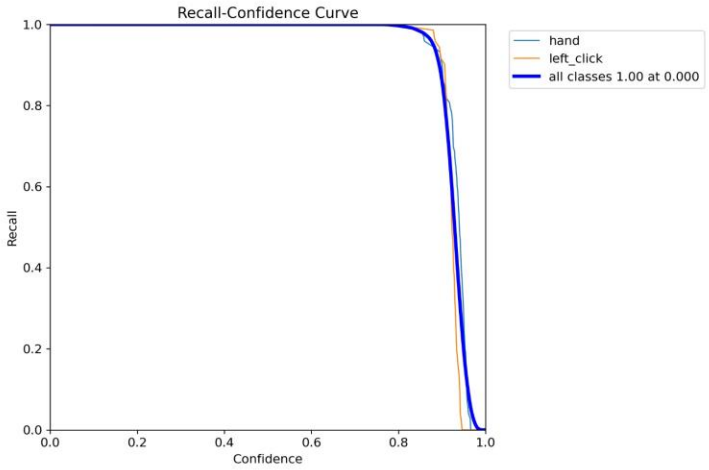


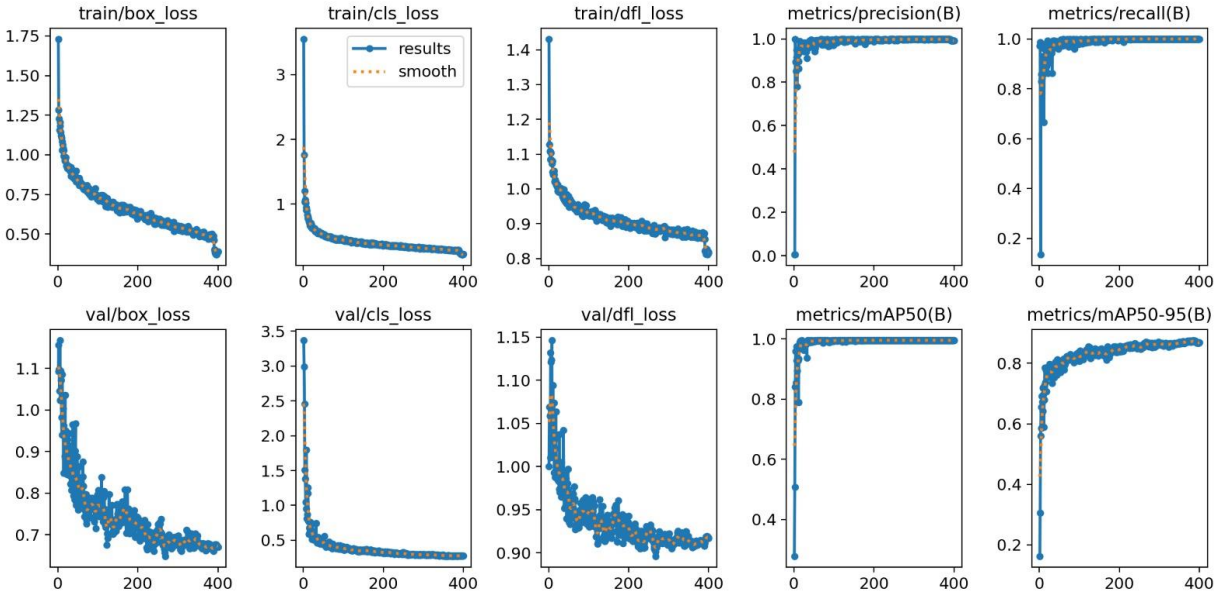


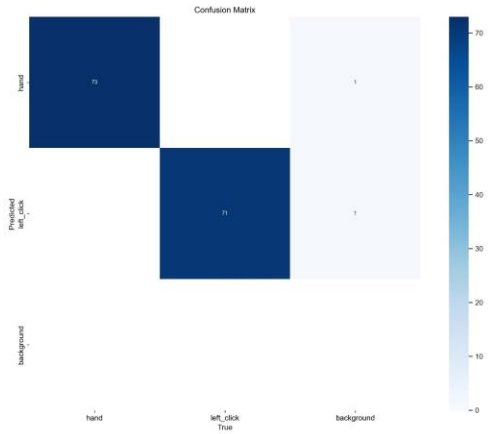


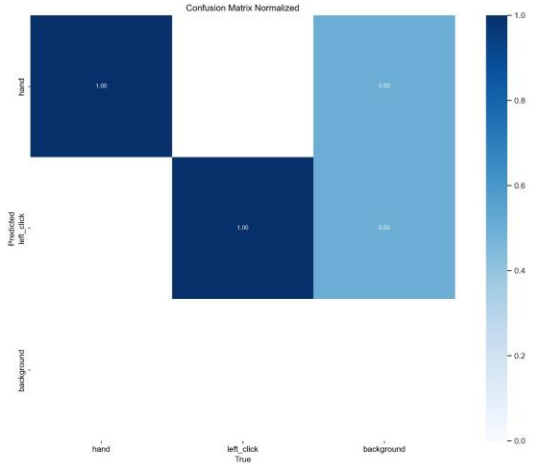



Precision-Recall Curve
hand 0.995
left_click 0.995
all classes 0.995 mAP@0.5
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


Recall-Confidence Curve
hand
left_click
all classes 1.00 at 0.000
Recall
Confidence
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


train/box_loss
train/cls_loss
results
smooth
train/dfl_loss
metrics/precision(B)
metrics/recall(B)
val/box_loss
val/cls_loss
val/dfl_loss
metrics/mAP50(B)
metrics/mAP50-95(B)


Confusion Matrix
hand
left_click
background
True


Confusion Matrix Normalized
hand
left_click
background
True