МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет компьютерных технологий и прикладной математики

Кафедра вычислительных технологий

**ОТЧЕТ О ПРОХОЖДЕНИИ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ (ПРЕДДИПЛОМНАЯ ПРАКТИКА)**

по направлению подготовки

02.03.02 Фундаментальная информатика и информационные технологии

период с 29.04.2024 г. по 15.05.2024 г

Выполнил студент \_\_\_46\_ гр.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_Гиренко Д.Е.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись) (Ф.И.О. студента)*

Руководитель практики (**производственная практика (преддипломная практика)**)

к.т.н., доцент Приходько Т.А.

*(ученое звание, должность) (подпись) (Ф.И.О)*

Оценка по итогам защиты практики: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_15\_\_» \_\_\_\_мая\_\_\_\_\_\_ 2024 г.

Краснодар

2024 г.

ФГБОУ ВО «КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет компьютерных технологий и прикладной математики

Кафедра вычислительных технологий

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ, ВЫПОЛНЯЕМОЕ В ПЕРИОД**

**ПРОВЕДЕНИЯ   
ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ**

**(ПРЕДДИПЛОМНАЯ ПРАКТИКА)**

Студент Гиренко Даниил Евгеньевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(фамилия, имя, отчество полностью)*

Направление подготовки 02.03.02 Фундаментальная информатика и информационные технологии

Место прохождения практики \_\_\_ ФГБОУ ВО «Кубанский государственный университет»\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Срок прохождения практики с 29.04.2024 г. по 15.05.2024 г

Целью преддипломной практики является формирование и развитие профессиональных знаний в сфере избранной специальности, овладение необходимыми компетенциями по избранному направлению специализированной подготовки, развитие навыков самостоятельной работы, разработка и апробация оригинальных научных предложений и идей, используемых при подготовке магистерской диссертации, овладение современным инструментарием науки для поиска и интерпретации информации с целью ее использования в процессе разработки, реализации и исследования математических и информационных моделей.

Формирование компетенций, регламентируемых ФГОС ВО:

| **Код компетенции** | **Содержание компетенции (или её части)** | **Планируемые результаты при прохождении практики** |
| --- | --- | --- |
| **ОПК-1** | Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности | Научился применять фундаментальные знания, полученные в области математических и (или) естественных наук |
| **ОПК-2** | Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности | Может применять компьютерные/суперкомпьютерные методы, современное программное обеспечение для решения задач профессиональной деятельности |
| **ОПК-3** | Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям | Получил навык разработки алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей |
| **ПК-1** | Способен понимать и применять в научно-исследовательской и прикладной деятельности современный математический аппарат, основные законы естествознания, современные языки программирования и программное обеспечение; операционные системы и сетевые технологии | Умеет понимать и применять в научно-исследовательской и прикладной деятельности современный математический аппарат. |
| **ПК-2** | Способен проводить под научным руководством локальные исследования на основе существующих методов в конкретной области профессиональной деятельности | Может проводить под научным руководством локальные исследования на основе существующих методов в конкретной области профессиональной деятельности |
| **ПК-5** | Способен применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии | Умеет применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии |

Перечень вопросов (заданий, поручений) для прохождения практики:

1. Подготовка датасета и обучение нейронной сети для решения задачи детекции\_\_\_\_\_\_\_

2. Реализация программного решения проблемы получения скорости движения трамвая\_\_\_

3. Анализ методов реализации модуля принятия решения для автопилота\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ознакомлен \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*подпись студента расшифровка подписи (ФИО)*

Руководитель производственной практики

(научно-исследовательская работа)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Приходько Т. А.

**План-график выполнения работ**:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Этапы работы (виды деятельности) при прохождении практики | Сроки | Отметка руководителя практики о выполнении (подпись) |
| 1 | Инструктаж по технике безопасности, охраны труда, пожарной безопасности, а также правилами внутреннего распорядка обучающихся.  Выбор и обоснование темы исследования. | 29.04.2024 |  |
| 2 | Подготовка датасета и обучение нейронной сети для решения задачи детекции | 30.04.2024-02.05.2024 |  |
| 3 | Реализация программного решения проблемы получения скорости движения трамвая | 03.05.2024-08.05.2024 |  |
| 4 | Анализ методов реализации модуля принятия решения для автопилота | 09.05.2024-11.05.2024 |  |
| 5 | Оформление результатов проведенного исследования и их согласование с руководителем (составление отчета о прохождении учебной практики). | 12.05.2024-14.05.2024 |  |
| 6 | Защита отчета. | 15.05.2024 |  |

Ознакомлен \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_Гиренко Д. Е.\_\_\_\_\_\_\_\_

*(подпись студента) ( расшифровка подписи)*

«29» апреля 2024 г.

Руководитель производственной практики (преддипломная практика)

доцент кафедры вычислительных технологий

факультета компьютерных технологий

и прикладной математики, к.т.н. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Приходько Т.А.

*(подпись) (Ф.И.О. руководителя)*

\*Инструктаж по ознакомлению с требованиями охраны труда, техники безопасности, а также правилами внутреннего трудового распорядка.

**ОЦЕНОЧНЫЙ ЛИСТ**

**результатов прохождения   
ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ**

**(ПРЕДДИПЛОМНАЯ ПРАКТИКА)**

**по направлению подготовки**

**02.03.02 Фундаментальная информатика и информационные технологии**

Фамилия И.О студента Гиренко Д. Е.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Курс \_\_\_\_4\_\_\_\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | ОБЩАЯ ОЦЕНКА  (отмечается руководителем практики) | Оценка | | | |
| 5 | 4 | 3 | 2 |
|  | Уровень подготовленности студента к прохождению практики | + |  |  |  |
|  | Умение правильно определять и эффективно решать основные задачи | + |  |  |  |
|  | Степень самостоятельности при выполнении задания по практике | + |  |  |  |
|  | Оценка трудовой дисциплины | + |  |  |  |
|  | Соответствие программе практики работ, выполняемых студентом в ходе прохождении практики | + |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | СФОРМИРОВАННЫЕ В РЕЗУЛЬТАТЕ УЧЕБНОЙ ПРАКТИКИ КОМПЕТЕНЦИИ  (отмечается руководителем практики от университета) | Оценка | | | |
| 5 | 4 | 3 | 2 |
|  | ОПК-1 Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности | + |  |  |  |
|  | ОПК-2 Способен применять компьютерные / суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности | + |  |  |  |
|  | ОПК-3 Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям | + |  |  |  |
|  | ПК-1 Способен понимать и применять в научно-исследовательской и прикладной деятельности современный математический аппарат, основные законы естествознания, современные языки программирования и программное обеспечение; операционные системы и сетевые технологии | + |  |  |  |
|  | ПК-2 Способен проводить под научным руководством локальные исследования на основе существующих методов в конкретной области профессиональной деятельности | + |  |  |  |
|  | ПК-5 Способен применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии | + |  |  |  |

Руководитель практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ Приходько Т.А.\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись) (расшифровка подписи)*

Сведения о прохождении инструктажа по ознакомлению с требованиями охраны труда, технике безопасности, пожарной безопасности, а также правилами внутреннего трудового распорядка

Предприятие Федеральное государственное бюджетное образовательное учреждение высшего образования «Кубанский государственный университет»

Факультет компьютерных технологий и прикладной математики

Кафедра вычислительных технологий

|  |  |
| --- | --- |
| Студент | Гиренко Даниил Евгеньевич, 21 |
|  | (ФИО, возраст) |

Дата 29 апреля 2024 г.

**1.     Инструктаж по требованиям охраны труда**

|  |  |
| --- | --- |
| Провел | доцент Приходько Т. А. |
|  | (должность, ФИО сотрудника, проводившего инструктаж, подпись) |

|  |  |
| --- | --- |
| Прослушал | Гиренко Даниил Евгеньевич |
|  | (ФИО, подпись студента) |

**2.     Инструктаж по технике безопасности**

|  |  |
| --- | --- |
| Провел | доцент Приходько Т. А. |
|  | (должность, ФИО сотрудника, проводившего инструктаж, подпись) |

|  |  |
| --- | --- |
| Прослушал | Гиренко Даниил Евгеньевич |
|  | (ФИО, подпись студента) |

**3.     Инструктаж по пожарной безопасности**

|  |  |
| --- | --- |
| Провел | доцент Приходько Т. А. |
|  | (должность, ФИО сотрудника, проводившего инструктаж, подпись) |

|  |  |
| --- | --- |
| Прослушал | Гиренко Даниил Евгеньевич |
|  | (ФИО, подпись студента) |

**4. Инструктаж по правилам внутреннего трудового распорядка**

|  |  |
| --- | --- |
| Провел | доцент Приходько Т. А. |
|  | (должность, ФИО сотрудника, проводившего инструктаж, подпись) |

|  |  |
| --- | --- |
| Прослушал | Гиренко Даниил Евгеньевич |
|  | (ФИО, подпись студента) |

**Отзыв руководителя**

**О ПРОХОЖДЕНИИ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ**

**(ПРЕДДИПЛОМНАЯ ПРАКТИКА)**

|  |  |
| --- | --- |
| Студент | Гиренко Даниил Евгеньевич |
|  | *(фамилия, имя, отчество полностью)* |

Направление подготовки 02.03.02. Фундаментальная информатика и информационные технологии

За время прохождения производственной практики (Преддипломная практика) студент собрал необходимые для датасета данные, разработал метод получения скорости из симулятора, провёл анализ модулей принятия решения, и на их основе написал заключительные главы дипломной работы. Так же было разработана и протестирована первая версия автопилота симулятора трамвая.

В процессе выполнения практики были поставлены и выполнены следующие задачи:

1. Подготовлен датасет и обучена нейронная сеть для решения задачи детекции.

2. Выполнена реализация программного решения проблемы получения скорости движения трамвая.

3. Проанализированы варианты реализации модуля принятия решения для автопилота.

Были проведены все необходимые мероприятия, связанные с выделением целей и задач ПП, а также с подготовкой её аналитической части. Все пункты индивидуального плана-графика выполнены в полной мере.

В ходе работы индивидуальное задание выполнено в полном объеме, студент проявил высокий уровень самостоятельности, и творческий подход к его выполнению. Представлен оформленный текст собранного материала. Результаты практики оцениваю на оценку «\_\_отлично\_\_».

|  |  |  |
| --- | --- | --- |
| Руководитель студента |  |  |
|  | *(подпись)* | *(расшифровка подписи)* |

**СОДЕРЖАНИЕ**

[Введение 9](#_Toc164451417)

[1 Описание симулятора и методы настройки потоков данных 11](#_Toc164451418)

[2 Выбор метода детекции 15](#_Toc164451419)

[3 Теоретическое обоснование выбора объектов детекции 17](#_Toc164451420)

[Заключение 20](#_Toc164451421)

[Список использованных источников 21](#_Toc164451422)

**ВВЕДЕНИЕ**

В современном мире технологические достижения стремительно преобразуют нашу жизнь. Существенный прогресс в области нейросетей открывает перед нами возможности создания сложных алгоритмов, способных автоматизировать процессы управления транспортными средствами, включая трамваи.

Трамваи играют значительную роль в системе общественного транспорта России, обеспечивая массовый перевоз пассажиров в крупных городах. Их эффективное и безопасное функционирование не только важно для плавного движения городской жизни, но и неразрывно связано с общественным комфортом и экономическим развитием регионов.

Вместе с тем, несмотря на значимость трамвайного транспорта, существует потребность в постоянном движении вперёд, освобождая людей от рутинной работы и повышая уровень безопасности и эффективности транспортной системы.

В этом контексте разработка симулятора автопилота трамвая выступает как важное направление инноваций, направленных на улучшение жизни городского населения и оптимизацию транспортных процессов. Самое вдохновляющее то, что данная технология уже используется в экспериментальном режиме на транспорте Петербурга [1].

В рамках научно-исследовательской работы были выполнены подготовительные задачи для создание полноценного автопилота. Был выбран симулятор TramSim Vienna, метод детекции при помощи нейронной сети Yolo, а также выбраны объекты, для детекции. Теперь же предстоит реализовать все части автопилота, чтобы иметь возможность увидеть полный рабочий цикл программного кода.

Для реализации поставленной цели предполагается решить следующие задачи:

1. Подготовка датасета и обучение нейронной сети для решения задачи детекции.

2. Реализация программного решения проблемы получения скорости движения трамвая.

3. Анализ методов реализации модуля принятия решения для автопилота.

Объектом исследования в работе являются архитектура и особенности разработки автопилота трамвая для симулятора.

Предметом исследования функциональность автопилота, его автономность и его эффективность для симуляции.

Информационная база исследования включает в себя интернет-ресурсы, академические статьи и публикации по подобным темам.

В качестве методов исследования использовались анализ литературы и документации, сравнительный анализ различных подходов и практик, математическое и программное моделирование, экспериментальное тестирование различных методов и практик в разработке автопилотов.

Научная новизна работы заключается в том, что подобных шагов к автоматизации работы трамваев в мире проведено достаточно мало, так что даже разработка автопилота для симулятора позволит приобрести значительный научный опыт в этой области.

Теоретическая и практическая значимость работы состоит в обогащении теоретических знаний о разработке и предоставлении практических рекомендации по выбору оптимальных методов и практик.

1. **Формирование датасета симулятора и обучение YoLo**

Для создания симулятора работы трамвая в городском окружении было решено использовать компьютерную игру TramSim Vienna [2]. Основным критерием выбора стала разработка на игровом движке Unreal Engine, что обеспечивает высокое качество графики и приближает визуальное восприятие к реальности. Этот выбор позволяет пользователям получить максимально реалистичный опыт управления трамваем, воспроизводящий условия работы настоящего транспортного средства в городской среде.

Одним из важных преимуществ TramSim Vienna является опыт разработчиков в создании симуляторов различных транспортных средств. Их аккуратность и внимание к деталям отражаются в реализации транспортного средства в игре, а также в его отзывчивости к действиям игрока. Разработчики серии игр TramSim обладают значительным опытом, который позволяет им создавать симуляторы, максимально приближенные к реальной работе транспортных средств, что делает выбор TramSim Vienna логичным и обоснованным для целей данного исследования.



Рисунок 1 – пример графики из игры

Так как камера во время движения автопилота будет находится на носу трамвая, то и требуемый для обучения нейросети датасет тоже будет формироваться из записей с камеры в том же положении. Так как симулятор TramSim является обычной видеоигрой, то запись датасета происходила достаточно легко – с помощью наигранных часов и записи экрана. В дальнейшем из видеозаписи игрового процесса выбирались кадры, которые были достаточно богаты различными игровыми объектами.



Рисунок 2 – вид камеры на носу

В итоге из, суммарно, около часа записей игрового процесса было получено 109 различных кадров, готовых к разметке. Так как для решения задачи детекции использовалась модель YoLo, то для разметки было достаточно выделить на кадрах искомые объекты в прямоугольники, подписав каждый из них. Для удобства выполнения этого монотонного процесса было решено воспользоваться интернет-сервисом Roboflow.

Roboflow — это платформа, предназначенная для упрощения процесса подготовки данных для обучения моделей машинного обучения. Она предлагает различные инструменты и функции, которые помогают исследователям данных, разработчикам и инженерам в создании и управлении датасетами. Roboflow достаточно большой сервис, он обладает следующими возможностями:

1. Преобразование данных: позволяет быстро преобразовывать и аугментировать изображения и другие данные для обучения моделей, включая изменение размеров, повороты, изменение контрастности и многое другое.

2. Аннотирование данных: предоставляет средства для разметки изображений и других типов данных, необходимых для обучения моделей, таких как метки объектов на изображениях.

3. Интеграция с различными фреймворками: поддерживает интеграцию с популярными фреймворками машинного обучения, такими как TensorFlow, PyTorch, Keras и другими.

4. Управление датасетами: позволяет организовывать и управлять датасетами, включая загрузку, хранение и обмен данными.

5. Обучение моделей: предоставляет возможность обучения моделей машинного обучения на подготовленных данных.

6. Коллаборация: предоставляет средства для совместной работы над проектами и обмена данными и результатами между членами команды.

Roboflow помогает упростить и автоматизировать рутинные процессы, связанные с подготовкой данных для машинного обучения, что позволяет исследователям и разработчикам сосредоточиться на создании и улучшении моделей. Кроме того, благодаря возможностям подключения дополнительных людей в команду разметки и обучения моделей прямо в их сервисе, не было страха того, что исследования могут уткнуться в потолок и занять очень много времени. К счастью, опасения не подтвердились и Roboflow был использован только как сервис для хранения датасета и последующего аннотирования.

Кроме того, прямо там имеющийся датасет можно было привести в форму, необходимую для обучения YoLoV8, а также применить аугментирование, расширив имеющийся датасет до 257 фотографий. Там же датасет был разделён на тренировочный, валидирующий и тестирующий наборы в соотношении 86 к 10 к 4. Аугментирование позволило получить достаточно большой датасет для обучения, которого вполне хватило для первой версии автопилота, а разделения данных на наборы улучшило результаты обучения.

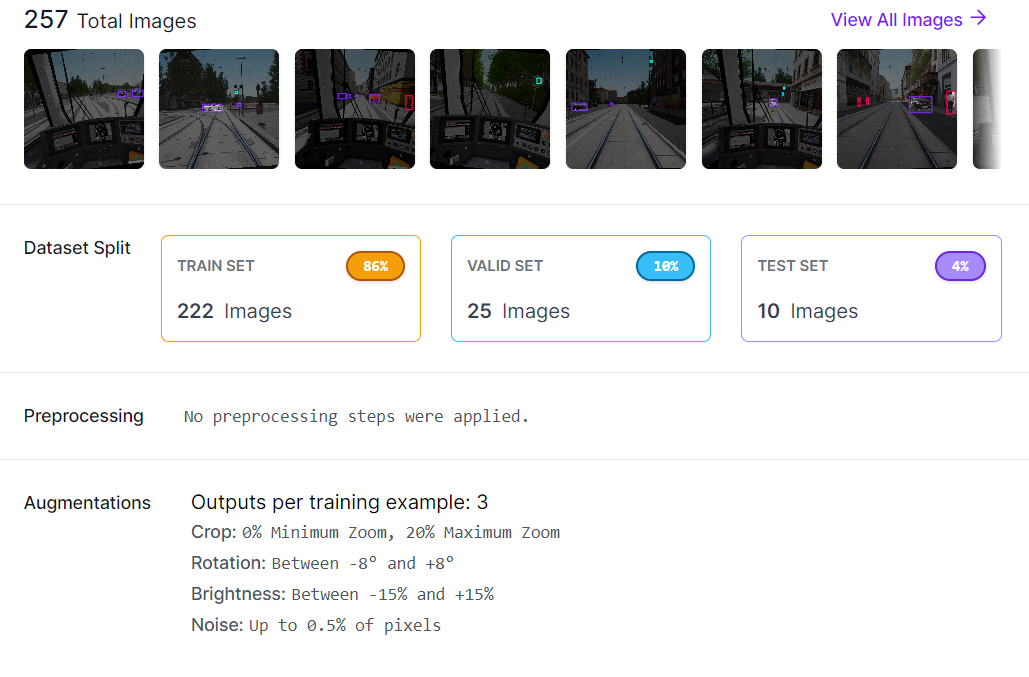


Рисунок 3 – датасет на странице Roboflow

В итоге, имеющийся датасет был использован для дообучения yolo8n, изначально обученной на стандартном датасете coco128. Версия nano была выбрана в виду того, что от модели требовалось моментально реагировать на происходящие события на дороге трамвая, точность определения объекта была на втором месте по важности [3].

Теперь требовалось обучить нейросеть, для этого использовался собственный ноутбук с видеокартой RTX 3050 Ti Laptop. Облачные сервисы не применялись в виду долгой настройки и вполне подходящего под задачу оборудования. В результате экспериментов по обучению сети, лучшей версией, согласно метрикам precision, recall и mAP50, была выбрана модель, обученная на 60 эпохах на имеющемся датасете. Именно это количество эпох оптимально для того, чтобы сеть максимально обучилась, но не имела проблем на валидирующем и тестировочном наборах. Результаты обучения представлены на графиках:

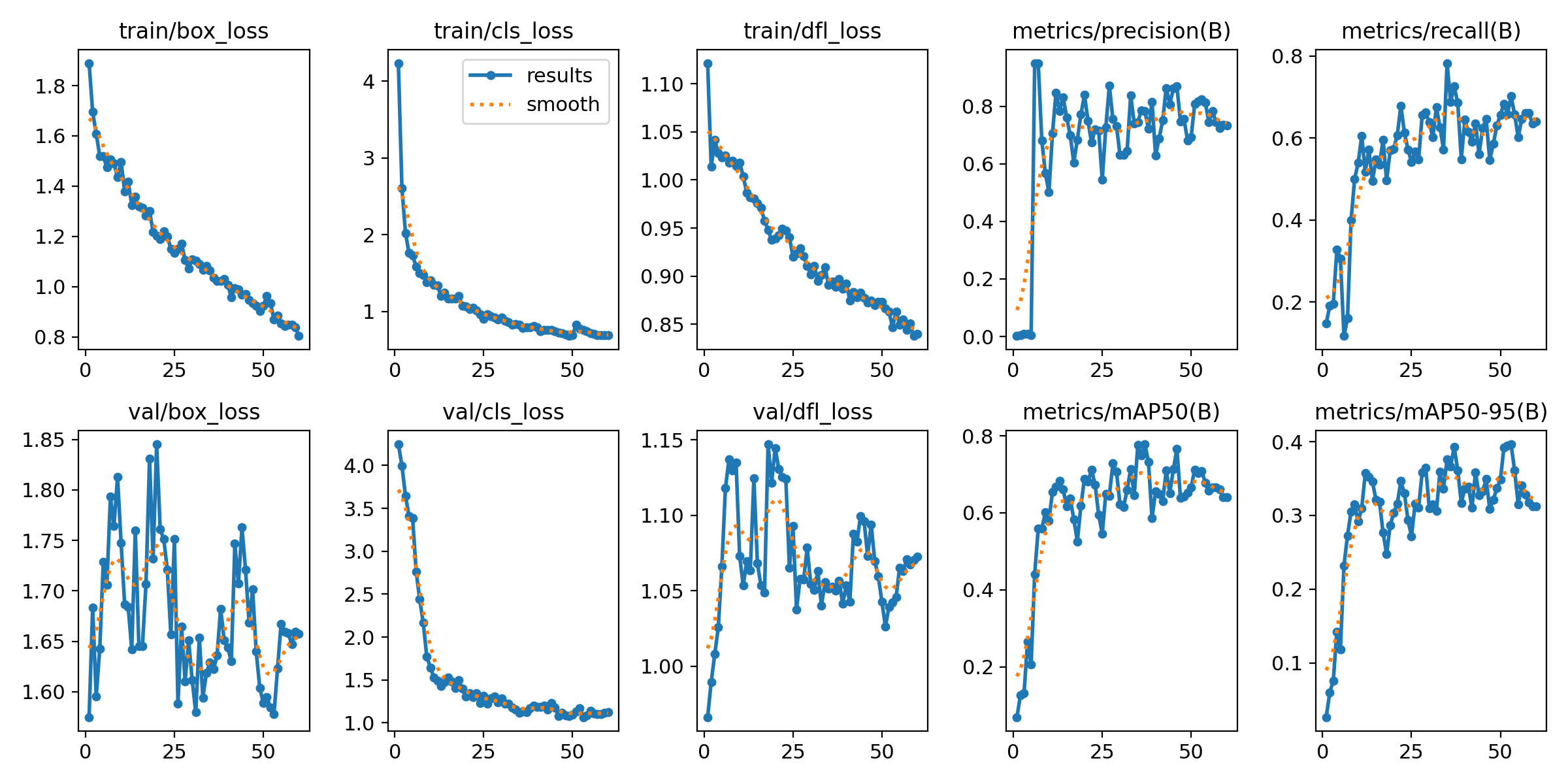


Рисунок 4 – результаты обучения YoloV8

1. **Использование Cheat Engine для получения скорости**

Для работы трамвая, а именно правильного управления им необходимо получать как минимум его скорость. Изначально в качестве решения данной задачи предполагалось каждый десятый кадр переключать вид камеры с носа в кабину, а после из кадра с кабины считывать информацию о скорости прямо со спидометра. Но, как и предполагалось, у этого метода оказалось достаточно много проблем, каждая из которых ухудшала точность измерения. Для получения хороших показаний требовалось написать ещё одну нейронную сеть, которая по спидометру будет определять его значение, ибо простейшие мультимедийные методы были далеки от идеала и требовали точной настройки.

Тогда появилась идея о том, что раз автопилот в реальных транспортных средства буквально подключается к имеющей системе и считывает оттуда данные, то почему бы по аналогии с этим не получать данные о скорости трамвая прямо из памяти симулятора [4]. Для взаимодействия с памятью процесса было решено воспользоваться Cheat Engine.

Cheat Engine — это программа с открытым исходным кодом, которая позволяет пользователям изменять параметры в компьютерных играх [5]. Она обеспечивает возможность изменения числовых значений. Пользователи могут сканировать память игры, находить адреса и значения, которые могут быть изменены, и создавать чит-таблицы для использования в игре. В основном программу применяют для редактирования числовых значений в видеоиграх для получения каких-либо нечестных бонусов. Аналогом Cheat Engine является и Art Money, однако за счёт долголетия жизни проекта в качестве программы с открытым исходным кодом, функционал Cheat Engine опережает Art Money.

Примером такого расширенного функционала являются инструменты для автоматизации определенных действий в игре через запись и воспроизведение скриптов, а также получение информации о постоянных указателях на те или иные значения в памяти. Именно этими функциями и предстояло воспользоваться, чтобы прямо с процесса работы симулятора получать информацию о скорости.

Суть операций в Cheat Engine состояла в том, чтобы из всего множества адресов памяти, которые использует процесс TramSimVienna-Win64-Shipping.exe выделить тот, который хранит в себе значение скорости трамвая, а после найти постоянный указатель среди таблицы всех указателей, который при любом запуске будет указывать именно на этот адрес. Сам адрес находился достаточно тривиально – после ускорения трамвая в игре адреса отсеивались по признаку увеличения значения, хранимого по адресу, после трамвай замедлялся и оставались лишь значения, что уменьшились с последнего отсеивания. Далее находились все указатели, которые указывали на этот адрес и, после перезапуска и повторного нахождения уже нового адреса значения скорости трамвая, из них отсеивались именно те указатели, которые теперь указывают на новый адрес. Данная процедура повторялась, пока не был найден хотя бы один такой нужный указатель.

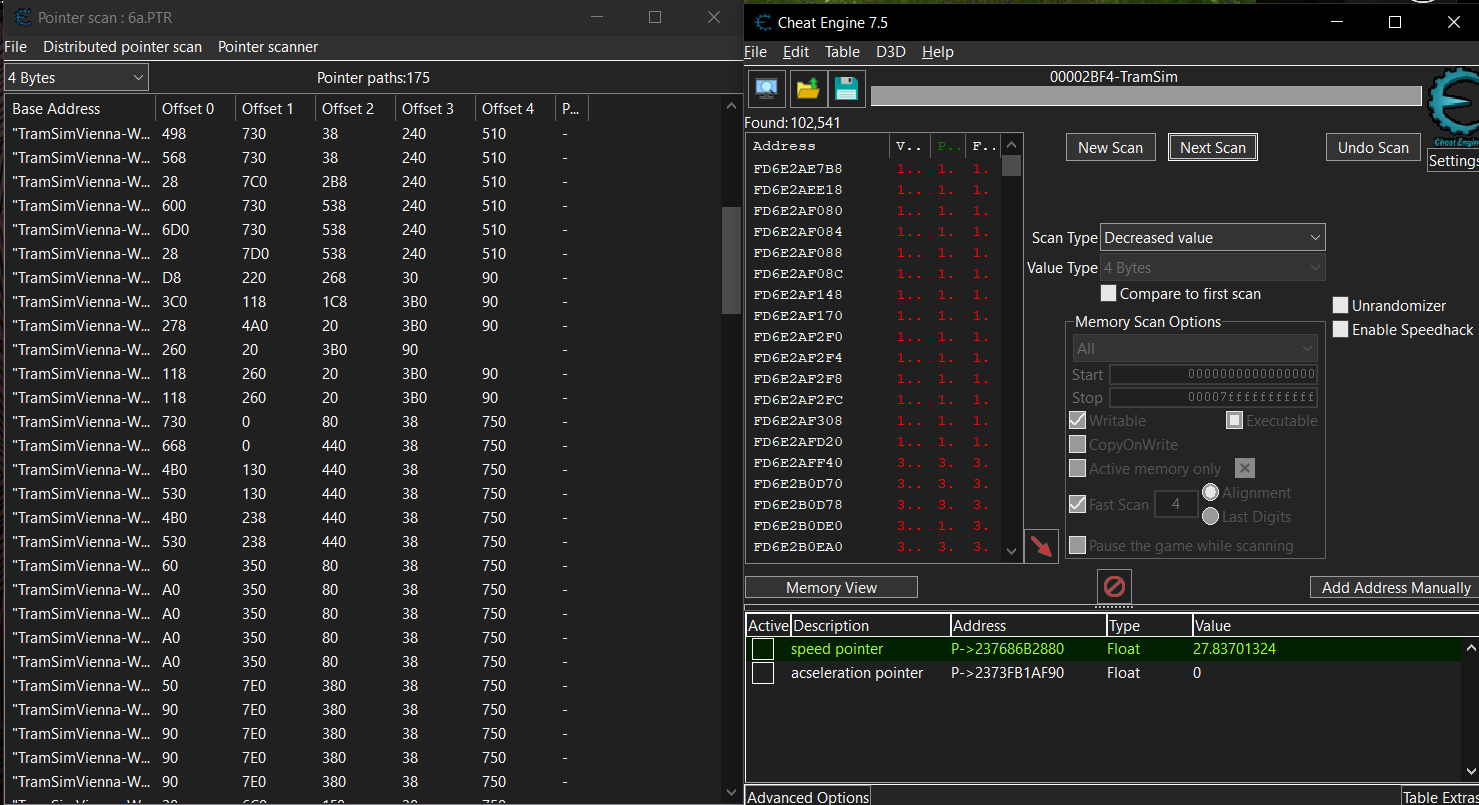


Рисунок 5 – пример работы Cheat Engine

Выделив указатель, который всегда указывает на адрес, который хранит в себе значение скорости, записываем его данные в программу на Python, которая буквально получает доступ к памяти процесса и читает хранящееся по адресу значение. Эта операция происходит моментально, так как это самая стандартная операция чтения памяти, просто она точечно настроена.

Благодаря этому автопилот, подобно физической своей версии, способен прямо из процесса работы трамвая получать его скорость и дальше использовать при принятии решений. Крайне удобно.

Аналогично скорости, было настроено получение из симулятора ускорения трамвая, для более лёгкого его управления.

1. **Анализ возможных вариантов и выбор вида модуля принятия решений**

Теперь все части для получения полной информации о мире вокруг трамвая готовы. Автопилот способен считывать значения скорости, а ускорения трамвая, а также, благодаря нейросети, определять людей, машины, светофоры и кружки для остановки. Имея всю эту информацию, автопилот вполне способен принимать нужные решения наравне с игроком в симулятор. Но для этого необходимо реализовать модуль принятия решений.

Модули принятия решений для автопилотов могут быть реализованы с использованием различных подходов и технологий. Некоторые из наиболее распространенных способов включают:

1. Правила и эвристики: подход основан на заранее заданных правилах и эвристиках, которые определяют оптимальные действия в различных ситуациях. Например, автопилот может следовать набору правил для выполнения маневров, таких как взлет, крейсерский полет и посадка.

2. Логическое программирование: метод использует логические выражения и правила для описания принятия решений. Модуль принятия решений анализирует текущую ситуацию и применяет логические правила для выбора оптимального действия. Подобные автопилоты писались на заре зарождения логического программирования для написания экспертных систем.

3. Нейросетевые модели: подход включает использование искусственных нейронных сетей для обучения модуля принятия решений на основе больших объемов данных. Нейросеть может обучаться на исторических данных о заездах и аварийных ситуациях, чтобы прогнозировать оптимальные действия.

4. Гибридные подходы: комбинация нескольких методов, например, комбинирование правил и эвристик с алгоритмами машинного обучения для достижения оптимальных результатов.

Каждый из этих подходов имеет свои преимущества и ограничения, и выбор конкретного метода зависит от требований к автопилоту.

Безусловно, эффективнее всего было бы использовать нейросеть, обученную на множестве размеченных записей управления трамваем. Однако этот метод требует очень больших усилий для создания такого датасета, подготовки правильной структуры нейросети и подготовки необходимого для обучения оборудования, что по силу только большим командам разработки. И даже так это займёт достаточно много времени.

Поэтому было решено воспользоваться способами задания механического поведения при помощи хорошо знакомого автомата состояний. Каждую итерацию автопилот будет формировать модель мира и считывать данные о скорости, после чего на основе этих данных и результатов прошлых итераций будет приниматься решение либо о переходе в новое состояние, либо продолжение выполнение поведения предыдущего. Получив итоговое состояние, автопилот выполняет некоторое действие, соответствующее состоянию, после чего начнёт новую итерацию.

Были выделены следующие состояния и реализации их поведения:

1. Движение – трамвай едет вперёд, придерживаясь некоторого интервала скорости.

2. Остановка – трамвай заметил впереди находящийся светофор или знак остановки и должен начать снижать свою скорость.

3. Ожидание – трамвай полностью остановился и сейчас ждёт разрешающий сигнал светофора.

4. Посадка – трамвай прибыл на остановку и должен произвести посадку-высадку пассажиров. Это состояние реализовано как цельный скрипт, который выполняется за определённое время.

5. Резкая остановка – трамвай заметил опасность на своём пути и вынужден быстро остановиться, чтобы избежать аварии.

Возможно, для большей точности действий в последующих версиях будут добавлены ещё несколько моделей поведения, но пока что этого набора вполне достаточно, чтобы выполнять все необходимые функции общественного транспорта. На данном этапе автопилот отрабатывает правильно все состояния и переходы, однако, так как это логическое программирование, для большей точности и аккуратности вождения требуется дальнейшая настройка параметров.

**ЗАКЛЮЧЕНИЕ**

В данной работе были реализованы ключевые модули необходимые для полной работоспособности автопилота для симулятора трамвая. При помощи Roboflow был создан необходимый для обучения нейросети датасет, на собственном устройстве обучена модель yolov8n, что в итоге позволило закрыть проблему детекции объектов при помощи изображения. Задача получения скорости была закрыта при помощи прямого получения значения скорости из памяти процесса, указатель на адрес которой был получен при помощи Cheat Engine. Среди всех рассмотренных вариантов реализации модуля принятия решения был выбран пусть и не самый эффективный, но вполне реализуемый метод логического программирования и задания правил и эвристик. На основе этих алгоритмов была разработана машина состояний, определяющая поведения трамвая при каждом обновлении.

Детекция объектов, создание модели мира, получение скорости и ускорения, принятие решения о состоянии трамвая и соответствующий состоянию вывод сигналов управления – полный цикл работы разработан, пусть и требует дальнейшей отладки и уточнений, для повышения эффективности.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Статья о результатах работы Cognitive Pilot в Питере [Электронный ресурс]. – URL: <https://www.fontanka.ru/2024/03/14/73330787/>
2. Сайт симулятора TramSim Vienna [Электронный ресурс]. – URL: <https://www.tram-sim.com/en/vienna>
3. Документация детекции для YOLOv8 [Электронный ресурс]. – URL: <https://docs.ultralytics.com/ru/tasks/detect/>
4. Статья на Хабр от Cognitive Pilot «Как наш беспилотный трамвай видит реальный город» [Электронный ресурс]. − URL: <https://habr.com/ru/companies/cognitivepilot/articles/498660/>
5. Сайт Cheat Engine [Электронный ресурс]. – URL: https://www.cheatengine.org/