

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

09.03.01 Информатика и вычислительная техника

код и наименование направления подготовки

ОТЧЕТ

по преддипломной практике
на кафедре прикладной математики и кибернетики

Выполнил:

студент гр. ИП-713

«15» мая 2021 г.

_____ /Михеев Н.А./

оценка

Руководитель практики от университета

доцент каф. ПМиК

«15» мая 2021 г.

_____ /Ракитский А.А./

Новосибирск, 2021 г.

План-график проведения преддипломной практики

Вид практики

Тип практики: преддипломная практика

Способ проведения практики: стационарная

Форма проведения практики: дискретно по периодам проведения практик

Выдано обучающемуся Михееву Никите Алексеевичу

Фамилия Имя Отчество студента

Направление: 09.03.01 – Информатика и вычислительная техника

Код – Наименование направления

Профиль: Программное обеспечение средств вычислительной техники и автоматизированных систем

Курс 4, группа ИП-713

Тема ВКР: Разработка и исследование методов на основе машинного обучения для распознавания дорожных знаков ограничения скорости

Содержание практики:

Наименование видов деятельности	Дата (начало – окончание)
<i>Постановка задачи на практику, определение конкретной индивидуальной темы, формирование плана работ</i>	<i>01.02.21 – 13.02.21</i>
<i>Работа с библиотечными фондами структурного подразделения или предприятия, сбор и анализ материалов по теме практики</i>	<i>15.02.21 – 27.02.21</i>
<i>Выполнение работ в соответствии с составленным планом: 1. Постановка задачи 2. Анализ аналогичных средств и программного обеспечения 3. Выбор программных средств для разработки</i>	<i>01.03.21 – 17.03.21</i>
	<i>19.03.21 – 03.04.21</i>
	<i>05.04.21 – 17.04.21</i>
<i>Анализ полученных результатов и произведенной работы, составление отчета по практике</i>	<i>11.05.20 – 15.05.20</i>

Согласовано:

Руководитель практики от СибГУТИ _____ /Ракитский А.А./

Оглавление

ВВЕДЕНИЕ.....	4
1 ПОСТАНОВКА ЗАДАЧИ.....	6
1.1 Анализ аналогичных аппаратных средств и ПО	6
1.2 Основные требования к программному продукту	7
1.3 Выбор программных средств	8
1.3.1 Выбор языка программирования.....	8
1.3.2 Выбор среды разработки.....	10
1.3.3 Выбор библиотек.....	11
ЗАКЛЮЧЕНИЕ	13
СПИСОК ЛИТЕРАТУРЫ.....	14

ВВЕДЕНИЕ

В современном автомобилестроении все чаще встречаются новые технические решения, направленные на обеспечение безопасности дорожного движения. Среди них особое распространение начали получать системы распознавания дорожных знаков: они облегчают задачу водителю, позволяют ему лучше сосредоточиться на самом процессе вождения, ведь потеря концентрации и внимания может привести к опасным для здоровья и жизни последствиям.

Следует отметить, что задачи по распознаванию объектов являются достаточно сложными ввиду постоянного изменения условий (освещение, видимость, местоположение и различность окружения), усложняют задачу распознавания. Области машинного обучения и компьютерного зрения на сегодняшний день являются одними из самых развивающихся и актуальных, однако до сих пор не существует какого-то единого и оптимального подхода для решения поставленной задачи. Существует большое количество нерешенных трудностей, которые сильно снижают эффективность обнаружения:

- Низкое входное разрешение изображения;
- Сложный и неординарный фон изображений;
- Различные сложные ракурсы создающие визуальные искажения на изображениях;
- Частичное перекрытие знака на изображении листвой деревьев, электрическими проводами и иными другими объектами.

В настоящее время для распознавания дорожных знаков ограничения скорости в современных автомобилях используются закрытые, коммерческие системы, которые поставляются вместе с автомобилем как дорогостоящая опция без возможности модификации. К таким системам относятся [1]: «Speed Limit Information Function» от KIA, «Road Sign Assist» от Toyota, «Traffic Sign Recognition» от Mazda. При анализе предметной области было выявлено, что

даже указанные выше комплексы не в полной мере удовлетворяют решению поставленной задачи. Эффективность данных систем резко снижается в реальных условиях.

Исходя из вышесказанного можно сделать вывод, что тема дипломного проекта остается актуальна, особенно в развивающемся автомобилестроительном бизнесе.

Цель данной работы – разработать и исследовать надежный распознаватель знаков дорожного движения путем построения, обучения и настройки свёрточных нейронных сетей. Готовое приложение, в отличие от систем, встраиваемых в автомобили, имеет преимущества, такие как открытость, отсутствие привязки к определенному автомобильному бренду, стоимость.

1 ПОСТАНОВКА ЗАДАЧИ

1.1 Анализ аналогичных аппаратных средств и ПО

Стоит отметить, что целью работы является не копирование существующих решений в области обнаружения и классификации знаков дорожного движения, а создание своей с помощью сверточных нейросетей и компьютерного зрения.

На сегодняшний день большинство автопроизводителей имеют свои подобные системы. Но они не всегда способны быстро и точно определить дорожный знак. Некоторые из таких систем имеют дело только с ограниченным количеством знаков. Например, «Speed Limit Information Function» от Kia и «Road Sign Assist» от Toyota способны определять только знаки ограничения скорости и знак запрета обгона. Также точность перечисленных средств резко снижается при наклоне, частичном перекрытии, загрязненности или повороте знака. Существует проблема распознавания знаков в слепых зонах автомобиля, особенно когда данные знаки установлены с нарушениями дорожных норм.

На рынке также представлено множество различных «радар-детекторов», которые занимаются не только обнаружением дорожных скоростных радаров, но и имеют дополнительный функционал, предназначенный для определения знаков ограничения скорости. Однако, так как данный функционал не является основным, точность и скорость работы подобных устройств оставляет желать лучшего. По этой причине использовать такие устройства для обеспечения безопасности дорожного движения нежелательно.

Отличительная особенность разрабатываемого проекта – возможность расширения списка знаков для обнаружения и точность работы. А поскольку решение не привязано к какой-либо коммерческой системе, то при наличии необходимого оборудования есть возможность легко переносить и масштабировать программу.

1.2 Основные требования к программному продукту

Согласно выбранной теме выпускного квалификационного проекта необходимо сделать программное обеспечение для распознавания знаков ограничения скорости с помощью свёрточных нейронных сетей. Разделить данную задачу можно на два простых этапа:

- Обнаружение области расположения самого дорожного знака на изображении;
- Классификация дорожного знака по имеющимся в базе классам.

Найти и распознать знак можно будет на заранее сделанном фото, видео или в реальном времени с помощью веб-камеры. Взаимодействовать с программой можно будет с помощью оконного интерфейса.

В разрабатываемой программе должны быть реализованы следующие основные функции:

- понятный пользовательский графический интерфейс;
- переключение режима работы программы: с заранее созданными файлами или с веб-камерой в режиме «реального времени»;
- загрузка графического изображения, видеофайла и их обработка;
- получение изображения с веб-камеры и его обработка;
- другие

Для работы с программой необходимо функционирование графического интерфейса на любой современной операционной системе (например, Windows 7/10, Linux). Предпочтительной является реализация, основанная на мультиплатформенной библиотеке для разработки интерфейсов, по причине возможности охвата максимального количества устройств. Также работа нейронных сетей предполагает большое задействование вычислительных мощностей, потребуется также вычислительная машина с графическим ускорителем с большим объемом видео памяти.

1.3 Выбор программных средств

Перед началом реализации проекта необходимо определиться с языками программирования и средами разработки программного продукта, так как этот выбор может повлиять на выбор сторонних библиотек, средств разработки пользовательских интерфейсов и т.д.

Выбор программных средств должен основываться на принципах стабильной работы используемого стороннего программного обеспечения, современных их реализациях и использовании актуальных методов разработки программного обеспечения.

1.3.1 Выбор языка программирования

Выбор языка программирования определяет весь инструментарий, который будет доступен во время реализации проекта. Он включает в себя среды разработки, библиотеки, фреймворки и другие вещи, характерные для того или иного языка. Для совершения правильного выбора можно выделить следующие критерии:

- 1) Предыдущий опыт
- 2) Аппаратная платформа программы
- 3) Распространенность
- 4) Производительность
- 5) Актуальность

В настоящее время Python [2] – один из самых распространенных и часто используемых языков программирования ввиду своей простоты, удобочитаемости, обширной поддержки сторонних модулей с большим сообществом разработчиков. Также из достоинств можно выделить легкую интеграцию с веб-сервисами, динамическую типизацию и интуитивно понятные для пользователя структуры данных.

Язык С является, вероятно, самым старейшим из часто используемых языков программирования на данный момент. С++ является расширенной его

версией и их можно отнести в одну категорию. Они достаточно трудны для изучения новичкам в разработке ввиду своей сложности и избыточности. Тем не менее, оба языка широко распространены в компьютерной науке и программировании. С целью большего разнообразия для них существует большое количество компиляторов, делающих приложения легко переносимыми и хорошо портируемыми. Оба языка считаются одними из самых высокопроизводительных.

Язык программирования Java следует парадигме объектно-ориентированного программирования и подходу «Написал один раз – используешь везде». Java используется в настольных, сетевых, мобильных и корпоративных приложениях. Язык представляет из себя целую экосистему инструментария, который может понадобиться в разработке, включающую в себя:

Java Developer Kit – комплект для написания и компиляции кода на языке Java;

Java Runtime Environment – исполняющая среда программ на языке программирования Java.

Отталкиваясь от данных критериев и свойств языков программирования, выбор был сделан в пользу на Python. Он проще для изучения, чем C и C++, а написанный код выглядит более лаконично, чем объектно-ориентированный как на Java.

Также наличие большого количества сторонних библиотек (пакетов) для работы с нейронными сетями, машинным обучением и графической информацией, которые впоследствии понадобятся при разработке и тестировании системы распознавания знаков дорожного движения, стало решающим фактором при выборе языка.

1.3.2 Выбор среды разработки

Выбор среды разработки чаще всего основывается на спецификации разрабатываемого проекта. На выбор могут повлиять такие факторы, как удобство использования среды разработки, интерфейсы, дополнительные возможности при разработке, операционная система или оборудование, для которого пишется та или иная программа, и так далее.

Для разработки приложений на языке Python [3] чаще всего используется среда разработки JetBrains PyCharm. PyCharm – одна из лучших профессиональных, полнофункциональных IDE. Существует как бесплатный (open-source) версия Community, так и платный Professional вариант среды разработки. PyCharm доступен на операционных системах семейств Windows, Linux, Mac OS. Есть поддержка веб-фреймворков, таких как Django, Flask, Google App Engine и других.

Кроме PyCharm, существуют и другие среды разработки, используемые при разработке программ на языке Python.

Spyder – специальная среда разработки для научных расчетов, которая ввиду своей узкой специализации и ограниченного функционала для разработки нейронных сетей и компьютерного зрения не подходит для проекта.

Visual Studio + Python – знакомая и понятная среда разработки Visual Studio от Microsoft с плагином для разработки на языке Python. Имеет огромный функционал, но слабо подходит для машинного обучения и компьютерного зрения ввиду того, что данная среда изначально разрабатывалась под написание программ на языках C и C++.

Jupyter Notebook – действительно мощный инструмент для интерактивной разработки с большим функционалом для работы в области наук о данных и машинном обучении. Файлы исходного кода на языке Python для данного «блокнота» обладают особым расширением «.ipynb». В нем все содержимое хранится в формате JSON, для которого каждая ячейка исходного кода, вложенные изображения входы и выходы вычислений, математика и

пояснительный текст преобразуются в строки текста с некоторыми метаданными. Это необходимо для облегчения работы с системами контроля версий.

Все перечисленные среды разработки обладают меньшим функционалом, чем PyCharm, из-за чего скорость и качество работы может существенно снижаться.

Конечный выбор IDE остановился на PyCharm благодаря повсеместному использованию, знакомому и удобному интерфейсу, обширным возможностям по работе с нейронными сетями и машинному обучению, поддержке Jupyter формата «.ipynb», удобному рефакторингу кода с помощью встроенной инспекции кода (Code Inspection) для более эффективной разработки и хорошей поддержки системы контроля версий Git.

1.3.3 Выбор библиотек

Выбор библиотек, а именно включения в программу разработанного вне основного проекта программного кода, также очень важен.

Для разработки нейросети-классификатора была выбрана библиотека TensorFlow с реализацией Keras API [4]. TensorFlow – открытая программная библиотека для машинного обучения, разрабатывается и поддерживается компанией Google для решения задач построения и тренировки нейронных сетей. Применяется как для исследований, так и для разработки собственных продуктов от Google и сообществом. Основная реализация API предназначена для работы на языке Python, но также существуют и реализации для работы на C++, Java, Go, Swift и Haskell. Исходя из перечисленных качеств, альтернатив данной библиотеке просто нет.

Для быстрых векторных вычислений и более быстрых типов данных, которые будут использоваться для обучения модели, выбрана была библиотека NumPy [5]. Она обладает открытым исходным кодом, поддержкой больших многомерных массивов данных и матриц, вместе с большой

библиотекой математических функций и операций для работы с этими массивами.

Для обнаружения знаков дорожного движения была выбрана библиотека OpenCV [6]. Реализована на C++ с оберткой для языка программирования Python, также есть обертки для Java, Ruby, Lua и других языков. Насчитывает в себе более 2500 оптимизированных алгоритмов для разработки программ на основе компьютерного зрения и машинного обучения. С этой библиотекой работают почти все крупные компании, такие как Google, Microsoft, Intel, IBM, Sony.

Для создания пользовательского интерфейса существует встроенная в Python библиотека Tkinter, сторонние библиотеки PyQt5, PyGTK, PyForms и фреймворк Kivy [7]. Для создания мультиплатформенного приложения из данного набора библиотек и фреймворков подходят только Tkinter, PyGTK и Kivy. Инструментарий для создания пользовательского интерфейса должен обладать знакомым и понятным набором возможностей, быть прост в разработке.

В конечном итоге был выбран набор библиотек для создания графических интерфейсов на базе знакомой платформы Qt5 от компании Digia под названием PyQt5. PyQt5 реализован в виде набора Python-модулей. Эта библиотека имеет более 600 классов и 6000 функций и методов. Также отличительной особенностью можно назвать то, что библиотека является мультиплатформенной и работает на всех основных операционных системах, в том числе Linux, Windows и Mac OS.

ЗАКЛЮЧЕНИЕ

Целью прохождения преддипломной практики является постановка задачи, определение темы, работа с библиотечными фондами и анализ полученных знаний для выполнения практики.

В результате прохождения практики были получены знания и навыку получения и анализа информации, которые будут использованы в дальнейшем при выполнении и написании выпускной квалификационной работы.

Были проанализированы схожие по функционалу и возможностям системы, на основе этого были выделены их достоинства и недостатки, которые будут учтены при разработке выпускной квалификационной работы.

Сформированный в ходе прохождения преддипломной практики список требований к разрабатываемой программе является основным, но также может дополняться, добавляя дополнительные возможности и функции.

Также были проанализированы и выбраны основные программные средства, которые будут использоваться при разработке приложения. Их выбор напрямую влияет на функционал и структуру будущего приложения, процесс разработки, а также взаимодействию предполагаемого пользователя с будущей программой.

СПИСОК ЛИТЕРАТУРЫ

1. Как работает система распознавания дорожных знаков [Электронный ресурс]. URL: <https://techautoport.ru/sistemy-bezopasnosti/aktivnaya/sistema-raspoznavaniya-dorozhnyh-znakov.html> (Дата обращения: 20.04.2021).
2. Любанович Билл. Простой Python. Современный стиль программирования.: ООО «Питер Пресс», 2017. 480 с.
3. Выбираем самый удобный редактор кода Python [Электронный ресурс]. URL: <https://habr.com/ru/company/skillfactory/blog/521838/> (дата обращения: 20.04.2021).
4. Mattmann A. Chris. Machine Learning with TensorFlow, 2021. 456 p.
5. NumPy в Python. Часть 1. [Электронный ресурс]. URL: <https://habr.com/ru/post/352678/> (дата обращения: 20.04.2021).
6. Introduction to OpenCV. [Электронный ресурс]. URL: <https://docs.opencv.org/master/d1/dfb/intro.html> (дата обращения 20.04.2021).
7. GUI Programming in Python. [Электронный ресурс]. URL: <https://wiki.python.org/moin/GuiProgramming> (дата обращения 20.04.2021).