

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра « Прикладная математика и информатика»
(наименование)

01.03.02. Прикладная математика и информатика
(код и наименование направления подготовки)

Компьютерные технологии и математическое моделирование
(направленность (профиль))

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Применение методов машинного обучения для преобразования видео с
изображением шкалы прибора в значение измеряемой величины

Обучающийся В.С. Якимов

(Инициалы Фамилия)

(личная подпись)

Руководитель С.В. Митин

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2023

Аннотация

Темой выпускной квалификационной работы является применение методов машинного обучения для преобразования видео с изображением шкалы прибора в значение измеряемой величины.

Объект исследования: метод преобразования видео с изображением шкалы прибора в значение измеряемой величины.

Цель выпускной квалификационной работы: провести исследование методов машинного обучения и разработать проект для преобразования видео с изображением шкалы прибора в значение измеряемой величины.

Методология проведения исследования составляют текущие разработки в сфере машинного обучения для распознавания изображений.

Область применения разработки: автоматическая поверка измерительной техники.

Abstract

The theme of the final qualification work is the use of machine learning methods for converting video with the image of the instrument scale into the value of the measured value.

Object of research: a method of converting video with the image of the scale of the device into the value of the measured value.

The purpose of the final qualification work: development of a project for converting a video with an image of the scale of the device into the value of the measured value using machine learning methods.

The methodology for conducting the study constitutes the current developments in the field of machine learning for image recognition.

Scope of development: automatic verification of measuring equipment.

Оглавление

Введение.....	5
Глава 1. Теоретические основы и методы машинного обучения.....	9
1.1 Машинное обучение. Понятие и сущность.....	9
1.2 Основные принципы машинного обучения.	11
1.3 Машинное обучение в решении практических задач	13
1.4 Типы машинного обучения.....	15
Глава 2. Выбор и обоснование методов машинного обучения	19
2.1. Анализ достоинств и недостатков существующих методов машинного обучения	19
2.2. Выбор и обоснование применяемых методов.....	25
Глава 3. Экспериментальная часть	29
3.1 Выбор среды разработки и языка программирования	29
3.2 Практическая реализация проекта	31
Заключение	41
Список используемой литературы	43
Приложение А исходный код программы.....	46

Введение

Благодаря новым вычислительным технологиям, машинное обучение сегодня не похоже на машинное обучение прошлого. Оно было основано на распознавании образов и теории о том, что компьютеры могут учиться, не будучи запрограммированными на выполнение конкретных задач; исследователи, интересующиеся искусственным интеллектом, хотели посмотреть, могут ли компьютеры учиться на данных. Итеративный аспект машинного обучения важен, потому что, поскольку модели подвергаются воздействию новых данных, они способны самостоятельно адаптироваться. Они извлекают уроки из предыдущих вычислений для получения надежных, воспроизводимых решений и результатов. Это наука, которая не нова, но набрала новый импульс.

Машинное обучение адаптируется в соответствии с использованием необходимых данных для получения опыта. Эти данные могут быть получены в следствии применения методов линейной алгебры, статистики, оптимизации и вычислительных методов компьютерных систем для вывода взаимосвязей и принятия решений. Машинное обучение играет ключевую роль в современных технологиях и предлагает широкий спектр возможностей для автоматизации и оптимизации различных отраслей. Одним из таких примеров является распознавание видео. Системы распознавания видео основаны на анализе и интерпретации визуальных данных, что позволяет компьютерам распознавать и классифицировать объекты, сцены, лица и другие элементы на видеозаписях.

Применение машинного обучения в распознавании видео имеет огромный потенциал и обеспечивает множество преимуществ. Например, в области транспорта машинное обучение может использоваться для автоматического обнаружения и классификации дорожных знаков, пешеходов или определения состояния дорожного покрытия. Это помогает

улучшить безопасность на дорогах и оптимизировать управление транспортным потоком.

В развлекательной отрасли машинное обучение применяется для рекомендаций контента, фильтрации нежелательного контента и анализа поведения пользователей. Например, платформы стриминга видео используют алгоритмы машинного обучения для предложения персонализированных рекомендаций фильмов и сериалов, основываясь на предпочтениях и предыдущем поведении зрителя.

В розничной отрасли машинное обучение применяется для анализа данных покупателей, прогнозирования спроса, оптимизации ценообразования и предотвращения мошенничества. Магазины могут использовать модели машинного обучения для предсказания покупательских предпочтений и предлагать персонализированные предложения или рекламу.

В энергетической отрасли машинное обучение может быть применено для оптимизации процессов генерации, передачи и потребления энергии. Алгоритмы машинного обучения могут анализировать данные о потреблении энергии и предсказывать будущий спрос, что позволяет эффективно планировать производство и распределение энергии.

Возвращаясь к конкретному примеру, применение методов машинного обучения для преобразования видео с изображением шкалы прибора в значение измеряемой величины имеет большое значение для автоматизации обработки измерительных данных. Это позволяет устранить необходимость ручного считывания и интерпретации показаний приборов, что может быть трудоемким, ошибочным и требующим значительных затрат времени. Автоматизация этого процесса с помощью машинного обучения повышает точность и эффективность измерений, а также снижает вероятность ошибок.

В целом, машинное обучение является мощным инструментом, который продолжает развиваться и находить все большее применение в различных сферах деятельности. Благодаря нему мы можем создавать интеллектуальные системы, способные анализировать данные, обучаться на

основе опыта и делать автоматизированные решения, что приводит к повышению производительности, оптимизации процессов и улучшению качества жизни.

Примеры машинного обучения в настоящее время широко распространены и, как ожидается, еще больше повлияют на транспортную, развлекательную, розничную и энергетическую отрасли.

Одним из таких примеров является распознавание видео.

В настоящее время системы автоматизации активно внедряются в технологические процессы различных предприятий. Зачастую производственные цехи и управляющие пультами оборудованы разнообразными стрелочными датчиками (манометрами, вольтметрами, термометрами и т.д.), которые позволяют обслуживающему персоналу или оператору следить за состоянием технологического процесса. Часто имеет место ситуация, когда вмешательство в технологический процесс с целью его полноценной автоматизации нежелательно, невозможно или экономически неоправданно, но, тем не менее, дублирование или замена оператора автоматикой необходимы.

Таким образом актуальность выпускной квалификационной работы заключается в применении методов машинного обучения для преобразования видео с изображением шкалы прибора в значение измеряемой величины с целью автоматизации обработки измерительных данных.

Объектом выпускной квалификационной работы является метод преобразования видео с использованием машинного обучения.

Целью выпускной квалификационной работы является разработка проекта с применением методов машинного обучения для преобразования видео с изображением шкалы прибора в значение измеряемой величины.

Для реализации поставленной цели в работе решаются следующие задачи:

- Исследовать понятие и сущность машинного обучения;

- Провести анализ применения машинного обучения в решении практических задач;
- Выбрать и обосновать выбор методов машинного обучения для преобразования видео с изображением шкалы прибора в значение измеряемой величины;
- Провести анализ достоинств и недостатков существующих методов машинного обучения;
- Выбрать и обосновать выбор среды разработки и языка программирования для реализации проекта;
- Практическая реализация проекта по преобразованию видео с изображением шкалы прибора в значение измеряемой величины.

Глава 1. Теоретические основы и методы машинного обучения

1.1 Машинное обучение. Понятие и сущность

В современном мире, где данные играют все более важную роль, методы машинного обучения становятся все более востребованными. Эти методы позволяют извлекать ценную информацию из огромных объемов данных, что позволяет сделать быстрые и точные прогнозы и принимать решения на основе данных. В данной работе рассматривается понятие и сущность методов машинного обучения.

Методы машинного обучения представляют собой обширное поле исследований и практического применения, которое продолжает развиваться с постоянными новыми открытиями и улучшениями. Они находят применение во множестве областей, включая бизнес, финансы, медицину, науку, технологии, искусство и многие другие.

В основе методов машинного обучения лежат математические и статистические модели, которые позволяют алгоритмам автоматически обучаться на основе данных и делать предсказания или принимать решения на их основе. Одним из ключевых аспектов машинного обучения является способность моделей обобщать и находить общие закономерности в данных, что позволяет им работать с новыми, ранее неизвестными данными. Методы машинного обучения являются совокупностью алгоритмов и методов, которые позволяют автоматически находить закономерности в данных. Они основаны на математических и статистических методах и позволяют строить модели, которые могут быть использованы для прогнозирования или классификации.

Одним из основных принципов методов машинного обучения является обучение на основе данных. Для этого используются различные алгоритмы, которые обрабатывают данные и выявляют закономерности. Например, для классификации данных используются алгоритмы, которые выделяют группы

данных с определенными характеристиками. Для прогнозирования используются алгоритмы, которые определяют, как изменится одна переменная при изменении другой.

Сущность методов машинного обучения заключается в поиске закономерностей в данных и использовании этих закономерностей для решения задач. Они используются для решения различных задач, таких как классификация, регрессия и кластеризация. Они могут быть применены к различным типам данных, таким как текстовые, числовые, изображения и видео.

Существует несколько основных типов методов машинного обучения. Одним из них является наблюдаемое обучение (supervised learning), которое используется для решения задач классификации и регрессии. В наблюдаемом обучении модель обучается на основе размеченных данных, где каждому примеру данных соответствует известная метка или значение. Например, при обучении модели для определения, является ли электронное письмо спамом или не спамом, модели предоставляются размеченные письма с указанием их класса (спам или не спам). Затем модель обучается на этих данных, чтобы научиться классифицировать новые, неизвестные письма.

Еще одним типом методов машинного обучения является необратимое обучение (unsupervised learning). В необратимом обучении модель обучается на неразмеченных данных, где нет заранее известных меток или значений. Целью является обнаружение скрытых структур или паттернов в данных. Например, при кластеризации данных модель может выявить группы объектов, которые имеют схожие характеристики или свойства. Это позволяет проводить сегментацию данных и делать выводы о сходстве или различии между наборами данных [16].

Другим распространенным подходом является усиление (reinforcement learning), которое используется для обучения агентов взаимодействовать с окружающей средой и принимать оптимальные действия для достижения поставленных целей. В усилении агент получает обратную связь или награду

за свои действия, что позволяет ему корректировать свое поведение и улучшать свою стратегию [6].

Методы машинного обучения могут быть реализованы с использованием различных алгоритмов, таких как метод опорных векторов (Support Vector Machines), нейронные сети (Neural Networks), и многое другое. Каждый алгоритм имеет свои особенности, преимущества и ограничения, и выбор конкретного алгоритма зависит от задачи, доступных данных и требуемой точности.

Одним из ключевых аспектов успешного применения методов машинного обучения является обработка и предварительная обработка данных. Некорректные или неполные данные могут привести к неверным результатам и плохой производительности модели. Поэтому важно проводить анализ данных, очистку данных, масштабирование и отбор признаков для получения оптимальных результатов.

В заключение, методы машинного обучения играют все более важную роль в современном мире, где данные являются ключевым ресурсом. Они позволяют извлекать ценную информацию из больших объемов данных и делать быстрые и точные прогнозы. Непрерывные исследования в области методов машинного обучения способствуют развитию новых алгоритмов и моделей, что открывает новые возможности для решения сложных задач и достижения новых достижений в различных областях.

1.2 Основные принципы машинного обучения

Машинное обучение - это метод построения алгоритмов, которые могут извлекать знания из данных, не требуя явного программирования. Оно основано на использовании статистических и математических методов для обнаружения закономерностей в данных.

Одним из ключевых принципов машинного обучения является использование обучающих данных. Обучающие данные представляют собой

наборы данных, которые содержат информацию об измеряемой величине (целевой переменной) и связанных с ней параметрах (признаках). Например, в задаче распознавания изображений, обучающие данные будут содержать изображения различных объектов и соответствующие им метки с указанием класса объекта.

Преобразование шкалы с видео в значение измеряемой величины является одним из примеров применения машинного обучения. Например, в производственных процессах может использоваться вольтметр для измерения напряжения на каком-либо устройстве. Однако, визуальное представление шкалы на вольтметре может быть неудобным для чтения и использования в автоматических системах контроля и управления.

Для преобразования шкалы с видео в значение измеряемой величины необходимо собрать обучающий набор данных. Этот набор данных должен содержать видеозаписи с отображением отклонения стрелки вольтметра и соответствующие значения напряжения, полученные с помощью других методов измерения. Набор данных может включать различные значения напряжения и разные положения стрелки на шкале вольтметра.

Затем необходимо обучить модель машинного обучения на этом наборе данных. Модель может быть обучена, например, на основе сверточной нейронной сети (Convolutional Neural Network, CNN), которая специально разработана для работы с визуальными данными, такими как изображения и видео. Сверточные нейронные сети могут автоматически извлекать признаки из видео и использовать их для предсказания значения напряжения на основе положения стрелки на шкале.

После обучения модели ее можно использовать для предсказания значения напряжения на основе новых видеозаписей, где неизвестно искомое значение. Это позволяет автоматизировать процесс измерения напряжения и упростить его интеграцию в другие системы.

Машинное обучение имеет широкий спектр применений в различных областях, включая медицину, финансы, транспорт, рекламу, робототехнику и

многое другое. Оно позволяет решать сложные задачи, которые традиционно требовали большого объема ручного программирования или человеческого экспертного знания. С развитием технологий и доступностью больших объемов данных, машинное обучение продолжает прогрессировать и находить все новые применения, помогая нам понять мир и принимать более эффективные решения [15].

Таким образом, машинное обучение - мощный метод, который позволяет извлекать знания из данных без необходимости явного программирования. Оно основано на использовании статистических и математических методов для обнаружения закономерностей в данных. Применение машинного обучения включает создание моделей на основе обучающих данных, которые содержат информацию о целевой переменной и связанных с ней параметрах. Модель может быть обучена на основе этих данных, чтобы предсказывать значения для новых наблюдений. Машинное обучение находит широкое применение в различных областях, помогая решать сложные задачи и принимать эффективные решения. С развитием технологий и доступностью больших объемов данных, машинное обучение продолжает прогрессировать и находить все новые применения, что помогает нам лучше понимать мир вокруг нас.

1.3 Машинное обучение в решении практических задач

Машинное обучение приобрело большую известность в последние годы из-за его способности применяться во многих отраслях промышленности для эффективного и быстрого решения сложных проблем. Вопреки тому, что можно было бы ожидать, случаи использования машинного обучения не так уж сложны.

В отличие от традиционного программирования, где разработчики создают явные инструкции для компьютера, машинное обучение позволяет компьютеру самостоятельно "учиться" на основе данных. Вместо того чтобы

явно программировать алгоритмы для выполнения определенных задач, машинное обучение использует данные для обучения моделей и создания алгоритмов, которые могут делать предсказания и принимать решения на основе этих данных.

Одной из ключевых особенностей машинного обучения является его способность находить закономерности и паттерны в больших объемах данных. Модели машинного обучения могут анализировать миллионы и даже миллиарды данных точнее и быстрее, чем человек, и выявлять скрытые связи и структуры, которые могут быть незаметны для человеческого восприятия. Это делает их незаменимыми инструментами для анализа и обработки данных в различных областях, таких как медицина, финансы, маркетинг, транспорт и многие другие.

Алгоритмы машинного обучения могут быть применены к различным типам данных, включая числовые значения, тексты, звуки, изображения и видео. Например, в задачах обработки изображений, модели машинного обучения могут обучаться распознавать объекты, лица или даже классифицировать изображения по содержанию. В области обработки естественного языка модели могут обучаться понимать и генерировать тексты, а также выполнять задачи автоматического перевода, анализа тональности и классификации текстовых документов.

Одной из наиболее популярных областей машинного обучения является глубокое обучение (deep learning), которое основано на использовании нейронных сетей с множеством слоев для обработки данных. Глубокое обучение демонстрирует впечатляющие результаты в области компьютерного зрения, распознавания речи и других сложных задач, требующих анализа больших объемов данных.

Машинное обучение стало неотъемлемой частью современного мира, применяясь в различных сферах промышленности и науки. Оно отличается от традиционного программирования тем, что позволяет компьютерам самостоятельно обучаться на основе данных. Эта способность машинного

обучения находить закономерности и паттерны в огромных объемах информации делает его неоценимым инструментом для анализа и обработки данных.

Алгоритмы машинного обучения могут обрабатывать разнообразные типы данных, включая числовые значения, тексты, звуки, изображения и видео. В задачах обработки изображений, они способны распознавать объекты и лица, а также классифицировать изображения. В области обработки естественного языка модели могут понимать и генерировать тексты, выполнять автоматический перевод и анализ тональности текстов.

Одной из самых мощных техник машинного обучения является глубокое обучение, основанное на нейронных сетях с множеством слоев. Глубокое обучение продемонстрировало впечатляющие результаты в области компьютерного зрения, распознавания речи и других сложных задач, где требуется обработка больших объемов данных [8].

Машинное обучение продолжает развиваться и находить все большее применение в различных отраслях, от медицины и финансов до маркетинга и транспорта. Его способность эффективно решать сложные проблемы и делать точные предсказания делает его неотъемлемым инструментом для современного мира. В будущем можно ожидать дальнейшего прогресса и новых инноваций в области машинного обучения.

1.4 Типы машинного обучения

Сегодня алгоритмы машинного обучения в основном обучаются с использованием трех основных методов.

- Контролируемое обучение:

Один из самых элементарных типов машинного обучения, контролируемое обучение, это тот, где данные помечены, чтобы сообщить машине о точных шаблонах, которые она должна искать. Хотя данные должны быть точно маркированы для работы этого метода, контролируемое

обучение является убедительным и обеспечивает отличные результаты при использовании в правильных обстоятельствах.

Например, у популярного сервиса потокового видео Netflix при нажатии на кнопку pause, алгоритм машинного обучения находит похожие видео на основе предпочтений подписчика. Алгоритм машинного обучения здесь снабжен небольшим обучающим набором данных для работы, который является меньшей частью большего набора данных. Он служит для того, чтобы дать алгоритму представление о проблеме, решении и различных точках данных, с которыми нужно иметь дело.

Обучающий набор данных здесь также очень похож на конечный набор данных по своим характеристикам и предлагает алгоритм с помеченными параметрами, необходимыми для задачи. Затем алгоритм машинного обучения находит связи между заданными параметрами, устанавливая причинно-следственную связь между переменными в наборе данных.

- Неконтролируемое обучение:

Неконтролируемое обучение, как следует из названия, не имеет меток данных. Машина ищет шаблоны случайным образом. Это означает, что нет человеческого труда, необходимого для того, чтобы сделать набор данных машиночитаемым. Это позволяет программе работать с гораздо большими наборами данных. По сравнению с контролируемым обучением, неконтролируемое обучение не очень популярно из-за меньшего количества приложений в повседневной жизни.

Поскольку неконтролируемое обучение не имеет ярлыков для работы, оно создает скрытые структуры. Отношения между точками данных затем воспринимаются алгоритмом случайным образом или абстрактно, без каких-либо входных данных, требуемых от людей.

Вместо конкретной, определенной и заданной постановки задачи неконтролируемые алгоритмы обучения могут адаптироваться к данным, динамически изменяя скрытые структуры.

- Обучение с подкреплением:

Обучение с подкреплением в первую очередь описывает класс задач машинного обучения, в которых агент работает в среде без фиксированного набора обучающих данных.

Обучение с подкреплением имеет алгоритм машинного обучения, который совершенствуется сам по себе. Он обычно учится методом проб и ошибок для достижения четкой цели.

В этом алгоритме машинного обучения благоприятные результаты усиливаются или поощряются, тогда как неблагоприятные результаты не поощряются.

Приложений машинного обучения много, включая внешние (ориентированные на клиента) приложения, такие как рекомендации продуктов, обслуживание клиентов и прогнозы спроса, а также внутренние, чтобы помочь предприятиям улучшить продукты или ускорить ручные и трудоемкие процессы.

Достижения в области глубокого обучения стимулировали быстрый прогресс в методах распознавания изображений и видео за последние несколько лет. Они используются для нескольких областей, включая обнаружение объектов, распознавание лиц, обнаружение текста, визуальный поиск, обнаружение логотипов и ориентиров, а также композицию изображений.

Поскольку машины хорошо обрабатывают изображения, алгоритмы машинного обучения могут обучать структуры глубокого обучения распознавать и классифицировать изображения в наборе данных с гораздо большей точностью, чем люди.

Подобно распознаванию изображений, такие компании, как eBay, Amazon и Facebook используют машинное обучение для распознавания видео, где видео разбиваются кадр за кадром и классифицируются как отдельные цифровые изображения.

Алгоритмы машинного обучения, такие как контролируемое обучение, неконтролируемое обучение и обучение с подкреплением, играют важную

роль в различных областях, от потокового видео и рекомендаций продуктов до обработки изображений и видео. Контролируемое обучение позволяет точно помечать данные и находить связи между параметрами, что полезно для точных задач. Неконтролируемое обучение обрабатывает большие объемы данных, находит скрытые структуры и адаптируется к изменениям. Обучение с подкреплением основано на пробах и ошибках, где благоприятные результаты поощряются, а неблагоприятные не получают подкрепления. Применение этих алгоритмов в различных областях, таких как рекомендации, обслуживание клиентов и распознавание изображений, демонстрирует их значимость и прогресс в области машинного обучения. Благодаря достижениям в глубоком обучении, обработке изображений и видео стала более точной и эффективной, и это привело к широкому использованию этих методов в различных отраслях. Машинное обучение продолжает эволюционировать и играть ключевую роль в решении сложных задач, обработке больших объемов данных и автоматизации процессов. Это обеспечивает новые возможности и улучшает эффективность в различных областях, от науки и технологий до бизнеса и повседневной жизни. Применение алгоритмов машинного обучения способствует инновациям, повышению точности и улучшению пользовательских опытов.

Глава 2. Выбор и обоснование методов машинного обучения

2.1. Анализ достоинств и недостатков существующих методов машинного обучения

В данном исследовании рассмотрены и проанализированы достоинства и недостатки различных существующих методов машинного обучения. Машинное обучение является одной из наиболее быстро развивающихся областей компьютерной науки, и его применение в реальном мире становится все более распространенным.

Метод линейной регрессии:

Описание: Линейная регрессия является одним из самых простых и популярных методов машинного обучения для решения задачи регрессии. В контексте преобразования видео с изображением шкалы вольтметра, мы можем использовать линейную регрессию для построения модели, которая будет предсказывать значение измеряемой величины на основе позиции стрелки на шкале вольтметра [11].

Перед использованием линейной регрессии необходимо выполнить следующие шаги:

Предварительная обработка изображения: Применяются методы предварительной обработки изображений, доступные в OpenCV, чтобы улучшить качество изображения и выделить необходимые детали, такие как шкала вольтметра и положение стрелки.

Извлечение признаков: Используя функции OpenCV для обработки изображений, извлекаются признаки, необходимые для прогнозирования значения измеряемой величины. Например, можно выделить контуры шкалы вольтметра и определить угол между стрелкой и нулевым делением [2].

Подготовка обучающих данных: Собирается набор обучающих данных, содержащий изображения шкалы вольтметра и соответствующие им известные значения измеряемой величины. Набор данных должен быть

достаточно разнообразным, чтобы модель могла обобщить и прогнозировать значения для новых изображений [3].

Обучение модели: Применяется метод линейной регрессии, доступный в OpenCV или используется библиотека машинного обучения, которая поддерживает линейную регрессию, чтобы обучить модель на подготовленных данных. Модель будет настраивать коэффициенты линейной функции, чтобы минимизировать среднеквадратичную ошибку между предсказанными значениями и реальными значениями измеряемой величины [18].

Тестирование модели: Оценивается производительность модели на тестовом наборе данных. Измеряются метрики, такие как средняя абсолютная ошибка или коэффициент детерминации, чтобы оценить точность модели. При необходимости можно внести корректировки в предварительную обработку изображений или добавить дополнительные признаки для улучшения результатов [12].

Преимущества:

- Простота и понятность модели;
- Высокая интерпретируемость результатов.

Недостатки:

- Линейная регрессия может быть недостаточно гибкой для моделирования сложных зависимостей между изображением шкалы вольтметра и измеряемой величиной;
- Ограниченная способность обработки шума и вариаций в условиях освещения и позиционирования.

Метод сверточных нейронных сетей (Convolutional Neural Networks, CNN):

Описание: Сверточные нейронные сети являются мощным инструментом для анализа изображений и обработки видео. В контексте преобразования видео с изображением шкалы вольтметра, мы можем использовать CNN для автоматического извлечения признаков из

изображений и построения модели, которая будет предсказывать значение измеряемой величины [1].

Для применения CNN для преобразования видео с изображением шкалы вольтметра, можно выполнить следующие шаги:

Подготовка данных: Подготавливается обучающий набор данных, содержащий изображения шкалы вольтметра и соответствующие им известные значения измеряемой величины. Дополнительно, можно использовать аугментацию данных, чтобы увеличить разнообразие обучающего набора и улучшить обобщающую способность модели [19].

Создание архитектуры сети: Определяется архитектуру сверточной нейронной сети, которая будет использоваться для обработки изображений шкалы вольтметра. Архитектура может включать в себя сверточные слои, пулинг, слои нормализации и полносвязные слои для классификации.

Обучение модели: Используется обучающий набор данных для обучения сверточной нейронной сети. Процесс обучения включает в себя передачу изображений через сеть, вычисление потерь и обновление весов сети с помощью алгоритма обратного распространения ошибки [13].

Оценка производительности модели: Используется тестовый набор данных для оценки производительности модели. Измеряются метрики, такие как средняя абсолютная ошибка или коэффициент детерминации, чтобы оценить точность модели. При необходимости можно внести корректировки в архитектуру сети или параметры обучения для улучшения результатов.

Преимущества:

- Способность автоматического извлечения признаков из изображений;
- Гибкость в обработке сложных зависимостей между изображением шкалы вольтметра и измеряемой величиной.

Недостатки:

- Большое количество обучаемых параметров может потребовать большой объем данных для обучения;

- Возможна сложность интерпретации результатов из-за глубокой структуры сети и нелинейных преобразований.

Метод опорных векторов (Support Vector Machines, SVM):

Описание: Метод опорных векторов является одним из популярных алгоритмов машинного обучения, который может быть применен для задач классификации и регрессии. В контексте преобразования видео с изображением шкалы вольтметра, мы можем использовать SVM для решения задачи регрессии и построения модели, которая будет предсказывать значение измеряемой величины на основе позиции стрелки на шкале вольтметра [4].

Применение метода опорных векторов может включать следующие шаги:

Подготовка обучающих данных: Собирается набор обучающих данных, содержащий изображения шкалы вольтметра и соответствующие значения измеряемой величины. Каждому изображению должна соответствовать точка на шкале с известным значением. Опционально, можно провести предварительную обработку изображений, чтобы улучшить качество и выделить нужные детали [5].

Извлечение признаков: Используется функция OpenCV для извлечения признаков из изображений шкалы вольтметра. Например, можно выделить контуры шкалы и стрелки, а также определить угол между нулевым делением и позицией стрелки.

Обучение модели: Применяется метод опорных векторов, доступный в OpenCV или используется библиотека машинного обучения, поддерживающую SVM, чтобы обучить модель на подготовленных данных. Модель будет находить оптимальную гиперплоскость, которая максимально разделяет данные, и использовать ее для предсказания значений измеряемой величины [20].

Тестирование модели: Оценивается производительность модели на тестовом наборе данных. Измеряются метрики, такие как средняя

абсолютная ошибка или коэффициент детерминации, чтобы оценить точность модели. При необходимости можно внести корректировки в предварительную обработку изображений или параметры SVM для улучшения результатов [17].

Преимущества:

- Эффективность работы с высокоразмерными пространствами признаков.

Недостатки:

- Чувствительность к выбору параметров, таких как ядро и коэффициент регуляризации;
- Возможна медленная работа при большом объеме данных.

Метод обработки изображений OpenCV:

OpenCV (Open Source Computer Vision Library) - это библиотека с открытым исходным кодом, которая предоставляет множество функций и алгоритмов для обработки изображений и компьютерного зрения. В контексте преобразования видео с изображением шкалы вольтметра, OpenCV может быть использован для выполнения различных операций обработки изображений, таких как фильтрация, сегментация, детектирование объектов и извлечение признаков.

Использование OpenCV для обработки изображений и преобразования видео с изображением шкалы вольтметра может включать следующие шаги:

Загрузка и предварительная обработка изображений: Используется функция OpenCV для загрузки видеофайла или захвата потока видео с камеры. Применяется предварительная обработка изображений, такая как фильтрация шума, улучшение контраста или преобразование цветового пространства, чтобы улучшить качество изображений и выделить нужные детали, такие как шкала вольтметра и положение стрелки.

Извлечение признаков и анализ изображений: Используются функции OpenCV для извлечения признаков из изображений шкалы вольтметра.

Например, можно использовать алгоритмы детектирования контуров или дескрипторы особых точек для определения положения шкалы и стрелки.

Анализ и интерпретация результатов: Анализируются извлеченные признаки и используются их для прогнозирования значения измеряемой величины.

Преимущества:

- Обширный набор функций и алгоритмов для обработки изображений и компьютерного зрения;
- Простота использования и наличие документации и примеров кода.

Недостатки:

- Некоторые операции обработки изображений могут быть вычислительно сложными и требовать высоких вычислительных ресурсов;
- Ограниченная способность обработки сложных зависимостей и изменений в условиях освещения или позиционирования объектов.

Метод сверточных нейронных сетей (Convolutional Neural Networks, CNN) является мощным инструментом для анализа изображений и обработки видео. Он позволяет автоматически извлекать признаки из изображений шкалы вольтметра и строить модель для предсказания измеряемой величины. Этот метод включает подготовку данных, создание архитектуры сети, обучение модели и оценку ее производительности. Преимуществами CNN являются способность автоматического извлечения признаков и гибкость в обработке сложных зависимостей. Однако, недостатками являются необходимость большого объема данных для обучения и сложность интерпретации результатов из-за глубокой структуры сети.

Метод опорных векторов (Support Vector Machines, SVM) является популярным алгоритмом машинного обучения, который также может применяться для решения задачи предсказания измеряемой величины на основе позиции стрелки на шкале вольтметра. Процесс применения SVM включает подготовку обучающих данных, извлечение признаков, обучение модели и тестирование ее производительности. Одним из преимуществ SVM

является его эффективность в работе с высокоразмерными пространствами признаков. Однако, недостатками метода являются чувствительность к выбору параметров и возможная медленная работа при большом объеме данных.

OpenCV (Open Source Computer Vision Library) представляет собой библиотеку с открытым исходным кодом, которая обладает обширным набором функций и алгоритмов для обработки изображений и компьютерного зрения. В контексте преобразования видео с изображением шкалы вольтметра, OpenCV может быть использован для выполнения различных операций обработки изображений, таких как фильтрация, сегментация, детектирование объектов и извлечение признаков. Применение OpenCV включает загрузку и предварительную обработку изображений, извлечение признаков, анализ изображений и интерпретацию результатов. Преимуществами OpenCV являются его обширный функционал, простота использования и наличие документации и примеров кода. Однако, недостатками являются вычислительная сложность некоторых операций и ограниченная способность обработки сложных зависимостей [7].

В целом, методы сверточных нейронных сетей, опорных векторов и обработки изображений с использованием OpenCV представляют различные подходы к анализу изображений и обработке видео с изображением шкалы вольтметра. Каждый из этих методов имеет свои преимущества и недостатки, и выбор конкретного метода зависит от требований задачи и доступных ресурсов.

2.2. Выбор и обоснование применяемых методов

В данной выпускной квалификационной работе был проведен анализ и сравнение методов машинного обучения для обработки изображений и компьютерного зрения в контексте считывания показаний с вольтметра. В результате исследования было установлено, что использование метода

обработки изображений OpenCV и метода компьютерного зрения имеет неоспоримые преимущества по сравнению с другими методами.

Для метода на основе OpenCV были применены различные техники обработки изображений, такие как фильтрация, преобразования цветового пространства и пороговая обработка. Эти методы позволили достичь точной сегментации и обнаружения объектов на изображении, что важно для извлечения показаний с вольтметра. Библиотека OpenCV также предоставила удобный и гибкий интерфейс для реализации алгоритмов обработки изображений. Она предлагает множество функций и алгоритмов, которые позволяют эффективно работать с видео и изображениями. Вот подробное описание преимуществ OpenCV при преобразовании видео с вольтметра в значение измеряемой величины:

Обнаружение и извлечение объектов: OpenCV предлагает различные методы для обнаружения и извлечения объектов на видео. Вы можете использовать алгоритмы машинного обучения, такие как каскады Хаара или нейронные сети, чтобы обнаружить вольтметр на видео и выделить его изображение.

Операции с изображениями: OpenCV предоставляет широкий набор функций для манипуляций с изображениями. Вы можете изменять размеры изображения, изменять его яркость и контрастность, применять фильтры для удаления шума или улучшения качества изображения. Это полезно при обработке видео с вольтметра, чтобы улучшить его читаемость и точность измерений.

Анализ и обработка видео: OpenCV позволяет анализировать видео и применять различные алгоритмы для обработки каждого кадра. Например, вы можете использовать алгоритмы фильтрации или сегментации для извлечения значений изображаемой величины с вольтметра.

Калибровка и коррекция искажений: Если видео с вольтметра содержит искажения, OpenCV предоставляет функции для калибровки и коррекции искажений. Вы можете использовать методы геометрической коррекции,

такие как удаление радиальных и тангенциальных искажений, чтобы получить более точные значения измеряемой величины.

Определение границ и контуров: OpenCV имеет мощные инструменты для обнаружения границ и контуров на изображениях. При обработке видео с вольтметра это может быть полезно для выделения шкалы и цифр на вольтметре. Вы можете использовать алгоритмы детектирования границ, такие как оператор Собеля или оператор Кэнни, чтобы обнаружить и извлечь контуры вольтметра.

Интеграция с другими библиотеками и инструментами: OpenCV обладает хорошей интеграцией с другими библиотеками и инструментами, что позволяет использовать ее в различных приложениях. Вы можете комбинировать OpenCV с библиотеками машинного обучения, такими как TensorFlow или PyTorch, чтобы улучшить точность извлечения данных с вольтметра. Также OpenCV имеет связи с языками программирования, такими как Python, C++, Java и другими, что обеспечивает удобный интерфейс для разработки и интеграции существующих систем.

Прежде всего, метод с использованием библиотеки OpenCV обладает широким спектром возможностей для обработки изображений, таких как изменение размера, поворот, смещение и т.д. Кроме того, он обеспечивает точную сегментацию изображений и обнаружение объектов на них, что делает его идеальным выбором для считывания показаний с вольтметра.

С другой стороны, метод компьютерного зрения позволяет использовать нейронные сети для классификации и распознавания образов на изображениях, что позволяет достичь еще более высокой точности и скорости обработки. Таким образом, использование метода компьютерного зрения дополнительно повышает эффективность и надежность процесса считывания показаний с вольтметра.

В результате данного исследования было установлено, что методы обработки изображений с использованием OpenCV и компьютерного зрения являются наиболее эффективными для считывания показаний с вольтметра.

При использовании OpenCV были применены различные техники обработки изображений, такие как фильтрация, преобразования цветового пространства и пороговая обработка, что позволило точно выделить и извлечь объекты на изображении. Библиотека OpenCV предоставила удобный и гибкий интерфейс для реализации алгоритмов обработки изображений и обладает множеством функций и алгоритмов, способных эффективно работать с видео и изображениями.

Кроме того, метод компьютерного зрения, основанный на нейронных сетях, позволяет классифицировать и распознавать образы на изображениях с высокой точностью и скоростью. Это повышает эффективность и надежность процесса считывания показаний с вольтметра.

Таким образом, сочетание методов обработки изображений с использованием OpenCV и метода компьютерного зрения является оптимальным подходом для решения задачи считывания показаний с вольтметра. Эти методы предоставляют широкий набор инструментов, позволяющих обнаруживать объекты, проводить операции с изображениями, анализировать и обрабатывать видео, а также корректировать искажения и определять границы и контуры на изображениях. Их интеграция с другими библиотеками и инструментами обеспечивает удобный интерфейс и возможность улучшения точности и скорости обработки данных. Все это делает данное решение идеальным выбором для считывания показаний с вольтметра и открывает перспективы для его применения в различных областях, связанных с обработкой изображений и компьютерным зрением.

Глава 3. Экспериментальная часть

3.1 Выбор среды разработки и языка программирования

Выбор среды разработки и языка программирования Python для работы с методами машинного обучения, такими как обработка изображений с использованием библиотеки OpenCV, обладает рядом преимуществ и позволяет эффективно решать задачи компьютерного зрения, включая определение измеряемых величин на видео.

Среда разработки и язык программирования Python являются незаменимыми инструментами при работе с методами машинного обучения, такими как обработка изображений с помощью библиотеки OpenCV. В данной работе рассматривается применение метода компьютерного зрения для определения измеряемой величины на видео с вольтметра. В качестве среды разработки был выбран PyCharm, а в качестве языка программирования - Python.

Одним из главных преимуществ выбора PyCharm в качестве среды разработки является его мощный набор инструментов, специально разработанных для работы с Python. PyCharm обладает широким функционалом, включая поддержку статического анализа кода, автодополнение, отладку и другие возможности, которые значительно упрощают разработку и отладку программ. Более того, PyCharm предоставляет удобный интерфейс для управления проектами и интеграцию с различными системами контроля версий, что способствует удобной организации рабочего процесса.

Язык программирования Python является отличным выбором для решения задач, связанных с обработкой данных и машинным обучением. Он обладает простым и понятным синтаксисом, что упрощает разработку и поддержку кода. Python имеет обширную экосистему библиотек для научных вычислений, включая библиотеки для обработки изображений и

компьютерного зрения, такие как OpenCV. Благодаря этому, разработчики могут быстро и эффективно реализовывать сложные алгоритмы и модели машинного обучения без необходимости писать большое количество кода с нуля.

Библиотека OpenCV является одной из наиболее популярных и мощных библиотек для компьютерного зрения и обработки изображений. Она предоставляет богатый набор функций, позволяющих выполнять различные операции с изображениями, включая фильтрацию, детектирование объектов, распознавание образов и другие. Благодаря OpenCV можно эффективно обрабатывать видеопоток и извлекать нужную информацию, такую как измеряемая величина на видео с вольтметра.

Одним из главных преимуществ использования OpenCV является возможность обнаружения и извлечения объектов на изображениях, что позволяет автоматизировать процесс обработки видео и уменьшить количество ошибок при определении измеряемой величины.

При выборе среды разработки и языка программирования Python для определения измеряемой величины на видео с помощью OpenCV имеется ряд преимуществ.

Во-первых, Python является одним из самых популярных языков программирования в мире и имеет огромное сообщество разработчиков, что позволяет быстро находить решения проблем и получать поддержку при необходимости. Кроме того, в Python существует множество библиотек для научных вычислений и обработки данных, в том числе и для компьютерного зрения, что делает его отличным выбором для данной задачи.

Во-вторых, OpenCV является одной из наиболее популярных библиотек для компьютерного зрения и обработки изображений. Она предоставляет широкий спектр функций для обработки изображений, включая функции для фильтрации, детектирования объектов, распознавания образов и многое другое.

Выбор Python и OpenCV также обеспечивает легкую интеграцию алгоритмов машинного обучения, таких как нейронные сети и метод опорных векторов (SVM), в проект. Это позволяет улучшить точность определения измеряемой величины на видео и обеспечить более точные результаты. Python также поддерживает различные стили программирования, включая функциональное и объектно-ориентированное программирование, что позволяет выбрать наиболее подходящий под конкретную задачу подход к разработке.

В целом, выбор среды разработки PyCharm и языка программирования Python для работы с методами машинного обучения и библиотекой OpenCV является логичным и эффективным решением. Эти инструменты предоставляют широкий функционал, удобный интерфейс и обширную экосистему библиотек, что позволяет разработчикам быстро и эффективно решать задачи компьютерного зрения и достигать точных результатов при определении измеряемых величин на видео.

3.2 Практическая реализация проекта

Создадим в PyCharm решение для проекта. Для реализации поставленной цели в решении требуется реализовать следующие методы:

- Получение видео;
- Обработка изображений шкалы прибора;
- Преобразование шкалы прибора в численные значения.

Для записи и воспроизведения видео с файла воспользуемся библиотекой OpenCV. В программе нужно извлечь кадры (изображения) из видеофайла или камеры и показать их в непрерывном цикле.

Для воспроизведения видеофайла объект VideoCapture должен быть создан путем передачи действительного местоположения в видеофайл. Затем объект VideoCapture должен считываться кадр за кадром. Наконец, эти кадры должны отображаться в окне последовательно.

Полный код программы представлен в приложении А.

На рисунке 1 показано как включаются файлы и пространства имен, используемые в нашей программе:

```
1  import numpy as np
2  import cv2
3  import math
4
```

Рисунок 1 – включение файлов и пространства имен

На рисунке 2 представлено как открыть видео с компьютера, а результат сохранить отдельным файлом:

```
5  webcam = cv2.VideoCapture('video.mp4')
6  fourcc = cv2.VideoWriter_fourcc(*'XVID')
7  out = cv2.VideoWriter('result.mp4', fourcc, 20.0, (260, 161))
```

Рисунок 2 – открытие видео с компьютера

Это один из немногих конструкторов, доступных в классе VideoCapture. Этот конструктор откроет видеофайл и инициализирует объект VideoCapture для чтения видеопотока из указанного файла.

На рисунке 3 представлен запуск бесконечного цикла «while», внутри которого происходит чтение очередного кадра из видеофайла webcam.read(). Полученный кадр сохраняется в переменную «imageFrame1».

Получаем кадры размером «w», «h» и количество цветовых каналов «c». Кадр изменяется в размере с помощью функции cv2.resize() на 700 пикселей в ширину и 500 пикселей в высоту.

```
11  _, imageFrame1 = webcam.read()
12  (w, h, c) = imageFrame1.shape
13  imageFrame = cv2.resize(imageFrame1, (700, 500))
14  hsvFrame = cv2.cvtColor(imageFrame, cv2.COLOR_BGR2HSV)
```

Рисунок 3 – запуск бесконечного цикла «while»

Переводим цветовую модель из BGR в HSV с помощью функции `cv2.cvtColor()`. Задаем пороговые значения для цвета вольтметра в формате HSV и создаем бинарную маску «mask» для выделения области вольтметра на изображении.

На рисунке 4 показана операция дилатации с помощью функции `cv2.dilate()` для замыкания дырок внутри области вольтметра, а так же операцию побитового AND между кадром «imageFrame» и маской «mask» с помощью функции `cv2.bitwise_and()`, чтобы получить изображение только области вольтметра. Результат сохраняется в переменную «res».

Производится поиск контуров в маске «mask» с помощью функции `cv2.findContours()`. Контуром в данном случае является граница области вольтметра.

Функция дилатации - это операция над изображением, которая позволяет увеличить размер и связность белых областей на изображении.

Для выполнения дилатации на изображении используется небольшой фильтр (структурный элемент), который перемещается по изображению. Для каждого пикселя на изображении фильтр проверяет, есть ли соседние пиксели, которые также являются белыми. Если есть, то значение текущего пикселя также устанавливается равным белому цвету.

Функция побитового AND (AND-операция) над двумя кадрами - это операция, которая выполняет логическое умножение между соответствующими пикселями двух изображений (кадров) с использованием побитовой операции AND.

Побитовый AND будет работать следующим образом:

- Каждый пиксель первого изображения сравнивается с соответствующим пикселем второго изображения;
- Если оба пикселя равны единице, то соответствующий пиксель в выходном изображении (результатирующем кадре) будет равен единице, в противном случае - нулю.

```

20     kernel = np.ones((5, 5), "uint8")
21
22     mask = cv2.dilate(mask, kernel)
23     res = cv2.bitwise_and(imageFrame, imageFrame, mask=mask)
24     cv2.imshow('2 - Mask', mask)
25
26     # поиск контура вольтметра
27     contours, hierarchy = cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

```

Рисунок 4 – функции дилатации и побитового AND

На рисунке 5 представлен блок кода в котором осуществляется поиск контуров, которые соответствуют шкале вольтметра. Для этого была использована функция `cv2.findContours`, которая возвращает список контуров и иерархию. Каждый контур представляет собой замкнутую кривую, которая образуется вокруг границы объекта.

Затем происходит перебор всех контуров в цикле `for c in contours:`. Если контур имеет высоту «h» больше или равную 2 и ширину «w» больше 2, то он считается действительным контуром. Затем на изображении `current` (кадр с изображением вольтметра) находится прямоугольник, который охватывает этот контур, используя функцию `cv2.boundingRect`. Этот прямоугольник затем используется для выделения только области изображения, соответствующей шкале вольтметра.

```

30     for pic, contour in enumerate(contours):
31         area = cv2.contourArea(contour)
32         if (area > 20000 and area < 60000):
33             x, y, w, h = cv2.boundingRect(contour)
34             print(x, y) #203 124
35             print(w, h) #270 188
36             imageFrame = cv2.rectangle(imageFrame, (x, y), (x + w, y + h), (255, 0, 0), 1)
37             cv2.putText(imageFrame, "VOLTMETER", (x, y), cv2.FONT_HERSHEY_SIMPLEX, 1.0, (255, 0, 0))
38             current = imageFrame[124 + 20:124 + 188 - 7, 203 + 5:203 + 270 - 5]
39             current = imageFrame[124 + 20:y + h - 7, x + 5:x + w - 5]
40             cv2.imshow("3 - Only Voltmeter", current)

```

Рисунок 5 – поиск контуров

На рисунке 6 представлено создание маски для окружности, используемой для обнаружения значений на шкале вольтметра. Для этого

создается черное изображение `imageEllipse` размером (`height-27, width-10, 3`) (размер окна эллипса), затем на этом изображении рисуются две дуги окружности, соответствующие шкале вольтметра. Маска для окружности создается из черно-белого изображения `imageEllipse2`, полученного путем применения пороговой фильтрации и инвертирования порогового изображения `imageEllipse1`.

```
42 # создание окна эллипса
43 imageEllipse = np.zeros((188 - 27, 270 - 10, 3), np.uint8)
44 imageEllipse[:] = (255, 255, 255)
45 WHITE = (255, 255, 255)
46 BLACK = (0, 0, 0)
47 height, width = imageEllipse.shape[0:2]
```

Рисунок 6 - создание окна эллипса

На рисунке 7 показано как создаются две дуги на черно-белом изображении с помощью функции `cv2.ellipse()`.

Первая дуга имеет начальный угол 205 градусов, конечный угол 270 градусов, радиусы осей 95 и 90 пикселей соответственно, толщину линии 2 пикселя, и находится на расстоянии 80 пикселей от верхнего края изображения с центром в середине изображения.

Вторая дуга имеет начальный угол 270 градусов, конечный угол 340 градусов, радиусы осей 85 и 90 пикселей соответственно, толщину линии 2 пикселя, и также находится на расстоянии 80 пикселей от верхнего края изображения с центром в середине изображения.

Затем изображение с дугами преобразуется в черно-белое изображение с помощью `cv2.cvtColor()` и пороговой обработки с помощью `cv2.threshold()`, чтобы получить инвертированное черно-белое изображение с областями, закрашенными черным цветом.

Изображение `imageEllipse` преобразуем в черно-белое изображение с помощью `cv2.cvtColor()`. Затем с помощью `cv2.threshold()` создаем инвертированное черно-белое изображение. Отображается созданное

изображение `imageEllipse2` и исходное черно-белое изображение `imageEllipse1`.

```
49 # дуга 205-270 градусов (чтобы не налезла на шкалу)
50 radius1_1 = 95
51 radius1_2 = 90
52 center1 = (int(width / 2), int(height / 2 + 27) + 80)
53 axes1 = (radius1_1, radius1_2)
54 angle1 = 0
55 startAngle1 = 205
56 endAngle1 = 270
57 thickness1 = 2
58
59 # дуга 270-340 градусов
60 radius2_1 = 85
61 radius2_2 = 90
62 center2 = (int(width / 2), int(height / 2 + 27) + 80)
63 axes2 = (radius2_1, radius2_2)
64 angle2 = 0
65 startAngle2 = 270
66 endAngle2 = 340
67 thickness2 = 2 # When thickness == -1 -> Fill shape
68
69 cv2.ellipse(imageEllipse, center1, axes1, angle1, startAngle1, endAngle1, BLACK, thickness1)
70 cv2.ellipse(imageEllipse, center2, axes2, angle2, startAngle2, endAngle2, BLACK, thickness2)
```

Рисунок 7 – создание дуг

На рисунке 8 представлен блок кода по созданию маски «mask» и «mask1» из черно-белых изображений `imageEllipse2` и `imageEllipse1` соответственно. На исходном изображении `current` применяем размытие Гаусса с помощью `cv2.GaussianBlur()`, а затем применяем фильтр Canny для выделения контуров на изображении с помощью `cv2.Canny()`. Полученный контур на изображении объединяем с маской `mask` с помощью `cv2.bitwise_and()`.

Также применяется операция объединения изображения `current` с маской `mask1` с помощью `cv2.bitwise_and()` и результат сохраняется в `res1`.

```

78     mask = np.array(imageEllipse2)
79     mask1 = np.array(imageEllipse1)
80     img = current.copy()
81     img = cv2.GaussianBlur(img, (3, 3), 0)
82     edge = cv2.Canny(img, 35, 15)
83     cv2.imshow('5 - Canny', edge)
84     res1 = cv2.bitwise_and(current, current, mask=mask1)
85
86     res = cv2.bitwise_and(edge, edge, mask=mask)
87     cv2.imshow('7 - Res', res)
88
89     contours, hierarchy = cv2.findContours(res.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

```

Рисунок 8 – создание масок «mask» и «mask1»

В этом блоке происходит обработка контуров объектов, найденных на изображении. Для каждого контура выполняются следующие действия:

- определяются координаты ограничивающего прямоугольника с помощью функции `cv2.boundingRect()`;
- рисуется прямоугольник на изображении с помощью функции `cv2.rectangle()`;
- вычисляется угол поворота стрелки вольтметра по формуле $\text{atan}((y - (\text{height}/2 + 27 + 120)) / (x - (\text{int}(\text{width}/2))))$, где x , y - координаты центра ограничивающего прямоугольника, width , height - ширина и высота изображения, соответственно;
- вычисляется значение напряжения на основе угла поворота стрелки вольтметра;
- отображается значение напряжения на изображении с помощью функции `cv2.putText()`;
- результат обработки каждого кадра сохраняется в выходной файл при помощи объекта `out.write()`.

Также в блоке выполняется проверка на нажатие клавиши 'q' для выхода из цикла. По окончании работы удаляются все окна с помощью `cv2.destroyAllWindows()`.

```

89 contours, hierarchy = cv2.findContours(res.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
90 for c in contours:
91     if (h >= 2 and w > 2):
92         x, y, w, h = cv2.boundingRect(c)
93         cv2.rectangle(imageFrame, (x, y), (x + w, y + h), (0, 0, 255), 3)
94         theta = 180 / math.pi * math.atan((y - (height / 2 + 27 + 120)) / (x - (int(width / 2))))
95         theta0 = 45
96         if theta < theta0:
97             volt = 0
98         if volt < 0:
99             volt = 0
100         volt_prev = volt
101         volt = (theta - theta0) / 0.4
102         delta = volt - volt_prev
103         if delta <= 2 and volt > 0:
104             tec_volt = int(volt)
105             print(tec_volt)
106             cv2.putText(res1, str(tec_volt), (20, 20), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (255, 0, 0), 2)
107         cv2.imshow('6 - Voltmeter+ellipse', res1)
108         out.write(res1)
109         if cv2.waitKey(10) & 0xFF == ord('q'):
110             break
111 cv2.destroyAllWindows()
112 webcam.release()
113 out.release()

```

Рисунок 9 – обработка контуров и вычисление значения

На рисунках 10-12 представлен результат выполнения программы:



Рисунок 10 – результат выполнения программы



Рисунок 11 - результат выполнения программы



Рисунок 12 - результат выполнения программы

Данный код выполняет поиск контура на видео и создание окна внутри этого контура с определенной геометрией (эллиптическим окном).

Для этого используется библиотека OpenCV, которая позволяет работать с изображениями и видео.

Сначала видео загружается через объект VideoCapture из указанного файла. Затем кадр считывается, изменяется размер и преобразуется в цветовое пространство HSV. Задаются нижний и верхний пороги цвета, и изображение бинаризуется, чтобы оставить только пиксели, соответствующие выбранному цвету. Затем на изображении выполняется дилатация, чтобы увеличить размер контуров.

Затем на изображении ищутся контуры объектов с помощью функции findContours. Контур объекта выбирается на основе его площади. Контуры с площадью меньше 20000 или больше 60000 игнорируются.

Далее создается эллиптическое окно внутри контура, которое представляет собой дуги двух окружностей с различными радиусами и углами поворота.

Для создания эллиптического окна с помощью дуг используются функции ellipse и cv2.ellipse. После этого эллиптическое окно конвертируется в изображение в оттенках серого и затем в маску, чтобы удалить пиксели, не попадающие в окно.

Затем выполняется операция Canny, чтобы найти границы изображения. Полученная маска сглаживается с помощью фильтра Гаусса, и затем изображение обрезается с помощью ранее созданной маски окна.

Наконец, на изображении находятся контуры с помощью функции findContours, и они выводятся на изображение с помощью функции drawContours.

Причина использования данного метода заключается в том, что он позволяет автоматически создавать эллиптическое окно внутри контура объекта.

Заключение

В результате выполнения выпускной квалификационной работы было разработано приложение, способное преобразовывать видео с изображением шкалы прибора в значение измеряемой величины. Данное приложение было разработано с использованием языка программирования Python, а программный код был написан в среде разработки PyCharm.

Выполненная выпускная квалификационная работа была сделана с учетом ГОСТ и методических указаний [9], [10], [14].

Использование Python и среды разработки PyCharm обеспечивает гибкость и удобство разработки приложения. Python является мощным языком программирования с богатой экосистемой библиотек, которые предоставляют инструменты для обработки видео и изображений. PyCharm, в свою очередь, предлагает удобную среду разработки с возможностью отладки и автоматического завершения кода, что упрощает процесс создания и сопровождения приложения.

Результаты выполненной работы позволяют сделать вывод о положительном решении задачи по преобразованию видео с изображением шкалы прибора в значение измеряемой величины. Это означает, что приложение успешно выполняет необходимые операции и предоставляет точные результаты.

Рекомендуется применять данную разработку в метрологических центрах с целью автоматизации обработки данных, полученных с измерительных приборов. Приложение может значительно упростить процесс анализа видео с изображением шкалы прибора и позволит получать значения измеряемой величины без необходимости вручную считывать показания. Таким образом, повышается эффективность работы метрологических центров и минимизируется вероятность ошибок при обработке данных.

В целом, разработанное приложение представляет собой ценный инструмент для автоматизации обработки видео с изображением шкалы

прибора и может быть успешно применено в различных областях, где требуется точное измерение значений на основе видеоданных. Это приложение открывает новые возможности для улучшения процесса измерений и обработки данных, а также способствует повышению надежности и эффективности работы.

Таким образом, разработанное приложение, созданное с использованием языка программирования Python и среды разработки PyCharm, представляет собой ценный инструмент для автоматизации обработки видео с изображением шкалы прибора. Оно успешно выполняет операции по преобразованию видео в значения измеряемых величин, что значительно упрощает процесс анализа и позволяет получать точные результаты. Рекомендуется применять данную разработку в областях, где требуется надежная и эффективная обработка данных на основе видео.

Список используемой литературы

1. Convolutional Neural Networks (CNN) with TensorFlow Tutorial. [Электронный ресурс]. URL: [<https://www.datacamp.com/tutorial/cnn-tensorflow-python>]
2. OpenCV Python Tutorial. [Электронный ресурс]. URL: [<https://www.geeksforgeeks.org/opencv-python-tutorial/>]
3. OpenCV Tutorial: A Guide to Learn OpenCV [Электронный ресурс]. URL: [<https://pyimagesearch.com/2018/07/19/opencv-tutorial-a-guide-to-learn-opencv/>]
4. SVM Machine Learning Tutorial – What is the Support Vector Machine Algorithm, Explained with Code Examples [Электронный ресурс]. URL: [<https://www.freecodecamp.org/news/svm-machine-learning-tutorial-what-is-the-support-vector-machine-algorithm-explained-with-code-examples/>]
5. Z. Gavrilov. SVM Tutorial [Электронный ресурс]. URL: [<https://web.mit.edu/zoya/www/SVM.pdf>]
6. В.В.Вьюгин Математические методы теории машинного обучения и прогнозирования. [Электронный ресурс]. URL: [<http://iitp.ru/upload/publications/6256/vyugin1.pdf>]
7. Введение в OpenCV. Учебное пособие [Электронный ресурс]. URL: [<https://dmkpress.com/files/PDF/978-5-97060-471-7.pdf>]
8. Волегов Д.Б. Считывание показаний стрелочного прибора при помощи web-камеры. / Д.Б. Волегов, Д.В. Юрин и др. // Московский физико-технический институт (государственный университет) – М.: ФГУП НПП ОПТЭКС, 2008 [Электронный ресурс]. URL: [https://www.graphicon.ru/html/2008/proceedings/Russian/SR4/Paper_1.pdf]
9. ГОСТ 2.105 – 95. Общие требования к текстовым документам [Текст]. – М.: Изд-во стандартов, 1996. – 29 с. – (Единая система конструкторской документации).

10. ГОСТ 7.1-2003. Библиографическая запись. Библиографическое описание. Общие требования и правила составления. – М.: Изд-во стандартов, 2004. – 48 с.

11. Искусственный интеллект: линейная регрессия и методы её обучения. Е. Коршунова [Электронный ресурс]. URL: [https://3dnews.ru/1054963/lineynaya-regressiya-i-metodi-eyo-obucheniya]

12. Как работает линейная регрессия [Электронный ресурс]. URL: [https://pro.arcgis.com/ru/pro-app/latest/tool-reference/geoai/how-linear-regression-works.htm#:~:text=Линейная%20регрессия%20-%20это%20контролируемый%20метод,корреляцию%20зависимых%20переменных%20с%20независимыми.]

13. Машинное обучение и искусственные нейронные сети. [Электронный ресурс]. URL: [https://docs.yandex.ru/docs/view?url=ya-browser%3A%2F%2F4DT1uXEPRrJRXlUFoewruMwROvrud9ZT222GXiQmU0CG71bXl42UU8A6TVA9xeDWz6LeAe3luYN2PP_MEt8FFPkWmU97OfETapi4xQuOEgbM2ucCKWRM162_bgKN_GFy_QK4XLSlKI5WR4ULOZLIWg%3D%3D%3Fsign%3D3Zu5aie2j6kFX9wUHkTO9hMKZUQgUKHCg0x2JA4CKYw%3D&name=10P-MOandINS.pptx]

14. Мкртычев, С. В. Прикладная информатика. Бакалаврская работа : электрон. учеб.-метод. пособие / С. В. Мкртычев, О. М. Гущина, А. В. Очеповский ; Тольяттинский государственный университет. – Тольятти : Изд-во ТГУ, 2019. – ISBN 978-5-8259-1386-5. [Электронный ресурс]. URL: [https://dspace.tltsu.ru/bitstream/123456789/8868/1/Мкртычев%201-57-17-ei-Z.pdf]

15. Овсянников А.В. Система автоматического контроля и управления на основе распознавания показаний на средствах визуализации аналоговых приборов. // Тезисы доклада на международной научнотехнической конференции «Автоматический контроль и автоматизация производственных процессов». – Минск: БГТУ, 2008. С.56-63 [Электронный ресурс]. URL:

[https://elib.bsu.by/bitstream/123456789/8236/1/Овсянников_Система%20автоматического%20контроля.pdf]

16. Поташев А.В. Интеграция математического моделирования и инновационных подходов к обучению в образовании монография/ Поташев А.В., Поташева Е.В., Сулейманова Д.Ю. – Электрон. текстовые данные – М.: Русайнс, 2015. – 96 с. [Электронный ресурс]. URL: [https://www.bgunb.ru/links/el_research/er_nauka/naukamolod/potashev_av.pdf]

17. С. Николенко. Метод опорных векторов. Презентация. [Электронный ресурс]. URL: [<https://logic.pdmi.ras.ru/~sergey/teaching/mlspsu19/06-svm.pdf>]

18. Учебный курс по Линейной регрессии. [Электронный ресурс]. URL: [<https://sysblok.ru/glossary/что-такое-линейная-регрессия/>]

19. Ф.М. Гафаров, А.Ф. Галимянов. Искусственные нейронные сети и их приложения. Учебное пособие. [Электронный ресурс]. URL: [https://kpfu.ru/staff_files/F1493580427/NejronGafGal.pdf]

20. Ю. Лифшиц. Метод опорных векторов. Лекция №7. [Электронный ресурс]. URL: [<https://yury.name/internet/07ianote.pdf>]

Приложение А
Исходный код программы

```
import numpy as np
import cv2
import math

webcam = cv2.VideoCapture('video.mp4')
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('result.mp4', fourcc, 20.0, (260, 161))
while (1):
    __, imageFrame1 = webcam.read()
    (w, h, c) = imageFrame1.shape
    imageFrame = cv2.resize(imageFrame1, (700, 500))
    #imageFrame=cv2.rotate(imageFrame2,cv2.ROTATE_90_CLOCKWISE)
    hsvFrame = cv2.cvtColor(imageFrame, cv2.COLOR_BGR2HSV)

    lower = np.array([61,0,193], np.uint8)
    upper = np.array([255, 255, 255], np.uint8)
    mask = cv2.inRange(hsvFrame, lower, upper)
    cv2.imshow('1 - HSV', hsvFrame)

    kernal = np.ones((5, 5), "uint8")

    mask = cv2.dilate(mask, kernal)
    res = cv2.bitwise_and(imageFrame, imageFrame, mask=mask)
    cv2.imshow('2 - Mask', mask)

# поиск контура вольтметра
```

```

contours, hierarchy = cv2.findContours(mask, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
for pic, contour in enumerate(contours):
    area = cv2.contourArea(contour)
    if (area > 20000 and area<60000):
        x, y, w, h = cv2.boundingRect(contour)
        print(x, y)
        print(w,h)
        imageFrame = cv2.rectangle(imageFrame, (x, y), (x + w, y + h), (255, 0, 0),
1)
        cv2.putText(imageFrame, "VOLTMETER", (x, y),
cv2.FONT_HERSHEY_SIMPLEX, 1.0, (255, 0, 0))
        current=imageFrame[124+20:124+188-7, 203+5:203+270-5]
        current = imageFrame[ + 20:y + h - 7, x + 5:x + w - 5]
        cv2.imwrite("volt_current.jpg", current)
        cv2.imshow("3 - Only Voltmeter", current)

# создание окна эллипса
imageEllipse = np.zeros((188-27, 270-10, 3), np.uint8)
imageEllipse[:] = (255,255,255)
WHITE = (255, 255, 255)
BLACK = (0, 0, 0)
height, width = imageEllipse.shape[0:2]

# дуга 205-270 градусов
radius1_1 = 95
radius1_2 = 90
center1 = (int(width/2), int(height/2+27)+80)
axes1 = (radius1_1, radius1_2)
angle1 = 0

```

```

startAngle1 = 205
endAngle1 = 270
thickness1 = 2

# дуга 270-340 градусов
radius2_1 = 85
radius2_2 = 90
center2 = (int(width / 2), int(height / 2 + 27) + 80)
axes2 = (radius2_1, radius2_2)
angle2 = 0
startAngle2 = 270
endAngle2 = 340
thickness2 = 2 # When thickness == -1 -> Fill shape

cv2.ellipse(imageEllipse, center1, axes1, angle1, startAngle1, endAngle1,
BLACK, thickness1)
cv2.ellipse(imageEllipse, center2, axes2, angle2, startAngle2, endAngle2,
BLACK, thickness2)

imageEllipse1 = cv2.cvtColor(imageEllipse, cv2.COLOR_BGR2GRAY)

imageEllipse2 = cv2.threshold(imageEllipse1, 0, 255,
cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)[1]
cv2.imshow('4 - ellipse mask', imageEllipse2)
cv2.imshow('ellipse1', imageEllipse1)

mask = np.array(imageEllipse2)
mask1 = np.array(imageEllipse1)
img = current.copy()
img = cv2.GaussianBlur(img, (3, 3), 0)

```



```

edge = cv2.Canny(img, 35, 15)
cv2.imshow('5 - Canny', edge)
res1 = cv2.bitwise_and(current, current, mask=mask1)

res = cv2.bitwise_and(edge, edge, mask=mask)
cv2.imshow('7 - Res', res)

contours, hierarchy = cv2.findContours(res.copy(), cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
for c in contours:
    if (h >= 2 and w > 2):
        x, y, w, h = cv2.boundingRect(c)
        cv2.rectangle(imageFrame, (x, y), (x + w, y + h), (0, 0, 255), 3)
        theta = 180 / math.pi * math.atan((y - (height / 2 + 27 + 120)) / (x -
(int(width / 2))))
        theta0 = 45
        if theta < theta0:
            volt = 0
        if volt < 0:
            volt = 0
        volt_prev = volt
        volt = (theta - theta0) / 0.4
        delta = volt - volt_prev
        if delta <= 2 and volt > 0:
            tec_volt = int(volt)
            print(tec_volt)
            cv2.putText(res1, str(tec_volt), (20, 20),
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (255, 0, 0), 2)
        cv2.imshow('6 - Voltmeter+ellipse', res1)
out.write(res1)

```

```
if cv2.waitKey(10) & 0xFF == ord('q'):
    break
cv2.destroyAllWindows()
webcam.release()
out.release()
```