**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Вологодский государственный университет» (ВоГУ)**

**ОТЧЕТ ПО ПРЕДИПЛОМНОЙ ПРАКТИКЕ**

**Направление подготовки: 09.03.01 «Информатика и вычислительная техника»**

**Направленность (профиль): Вычислительные машины, комплексы, системы и сети**

**Форма обучения: \_\_\_очная\_\_\_**

**Институт: Математики, естественных и компьютерных наук**

**Кафедра: Автоматики и вычислительной техники**

**Группа: 4Б09 ВМ-41**

**Студент: Пчелкина О.С.**

**Руководитель практики от университета: Кочкин Д.В.**

**Вологда**

**2023 г.**

Введение

Промышленная безопасность — состояние защищённости жизненно важных интересов личности и общества от аварий на опасных производственных объектах. Основная цель промышленной безопасности - предотвращение и/или минимизация последствий аварий на опасных производственных объектах.

На производствах достаточно часто нарушается техника безопасности, вследствие чего происходят травмы и летальные исходы. По данным статистики общее число происшествий за год снижается, но очень медленными темпами. Это все также приносит большой ущерб предприятиям. Чтобы решить эту проблему, было решено разработать такую архитектуру на базе глубоких нейронных сетей, которая в реальном времени детектирует наличие и правильность ношения средства индивидуальной защиты.

Разработка данного модуля поможет не только снизить процент травм и летальных исходов, но и автоматизировать контроль за соблюдением техники безопасности.

Объект исследования в данной работе – модуль обнаружения людей наличия и правильности ношения средства индивидуальной защиты. Предметом исследования являются сверточные нейронные сети.

Основная цель выпускной квалификационной работы – разработать модуль проверки наличия и правильности ношения средств индивидуальной защиты. для интеллектуальной системы.

Для достижения данной цели были поставлены следующие задачи:

1. Проведение анализа существующих отечественных и зарубежных аналогов.

2. Выбор технологии, языка и среды программирования для разработки.

3. Сбор и разметка выборочных данных для обучения.

4. Проектирование моделей нейронной сети.

5. Написание пользовательского интерфейса.

6. Написание API (Application Programming Interface).

7. Тестирование.

1 Анализ предметной области

1.1 Постановка задачи

Компания ООО «Малленом Системс» — ведущая российская компания в области разработки и внедрения систем видеоаналитики и промышленного контроля на основе технологий машинного зрения и искусственного интеллекта (машинное обучение, нейронные сети глубокого обучения) [1].

Основные направления деятельности компании «Малленом Системс»:

1. Визуальный контроль продукции на производстве.

2. Прослеживание продукции в алкогольной, фармацевтической и других отраслях промышленности.

3. Видеоконтроль и учет автомобильного и железнодорожного транспорта.

4. Автоматизация взвешивания автомобильного и железнодорожного транспорта.

5. Транспортное моделирование и управление дорожным движением.

6. Поставка и внедрение продукции Cognex (умные камеры, оптические датчики, сканеры кодов и маркировок) в РФ и СНГ.

Для освоения нового направления деятельности компанией «Малленом Системс» было предложено разработать модуль проверки наличия и правильности ношения средств индивидуальной защиты для интеллектуальной системы.

Описание задачи.

Разработать алгоритмы и программное обеспечение для обнаружения людей на видеоизображениях, проверки использования средств индивидуальной защиты и проверки правильности применения средств индивидуальной защиты.

Основные требования к функциональным характеристикам:

1. Анализ видеопотока в режиме реального времени.
2. Реализация событий нарушения и устранения нарушений ношения средств индивидуальной защиты.
3. Реализация API модуля.
4. Для демонстрации работы алгоритма и оценки его эффективности разработать приложение на языке C#.
5. Предусмотреть в приложении ручную загрузку видеофайла.

1.2 Выбор средств разработки

В ходе разработки потребуются:

1) работа с нейронной сетью;

2) разметка данных и ее обучение;

3) написание пользовательского интерфейса;

4) написание API.

Нейронные сети используются для решения сложных задач, которые требуют аналитических вычислений подобных тем, что делает человеческий мозг. Самыми распространенными применениями нейронных сетей является:

1. Классификация — распределение данных по параметрам. Например, на вход дается набор людей и нужно решить, кому из них давать кредит, а кому нет. Эту работу может сделать нейронная сеть, анализируя такую информацию как: возраст, платежеспособность, кредитная история и т. д.

2. Предсказание — возможность предсказывать следующий шаг. Например, рост или падение акций, основываясь на ситуации на фондовом рынке.

3. Распознавание — в настоящее время самое широкое применение нейронных сетей. Используется в Google, когда вы ищете фото или в камерах телефонов, когда оно определяет положение вашего лица и выделяет его и многое другое.

Основная задача нейронной сети в данной работе — это распознавание и классификация объектов. В связи с чем была выбрана архитектура сверточной нейронной сети. На данный момент сверточная нейронная сеть и ее модификации считаются лучшими по точности и скорости алгоритмами нахождения объектов.

В настоящее время для написания нейронных сетей больше всего подходят такие языки программирования, как Python и R.

R — интерпретируемый язык программирования, основным способом работы с которым является командный интерпретатор. Главным недостатком данного языка является то, что он сложен в обучении и достаточно медленный [8].

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен, однако стандартная библиотека включает большой объём полезных функций. Основное преимущество языка Python в том, что он является универсальным и многоцелевым языком программирования [9]. Вследствие чего для работы с нейронными сетями был выбран язык программирования Python.

Для написания интерфейса была выбрана Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств [10]. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения и веб-службы [10]. А точнее Visual Studio Community — полнофункциональная, расширяемая и бесплатная интегрированная среда разработки для создания современных приложений Android, iOS и Windows, а также веб-приложений и облачных служб. Любой индивидуальный разработчик может создавать бесплатные или платные приложения с помощью Visual Studio Community [10].

Основные ее преимущества:

1. Гибкость (создание приложений для любой платформы).

2. Производительность (конструкторы, редакторы, отладчики и профилировщики в одном инструменте).

3. Экосистема (доступ к тысячам расширений).

4. Языки (программирование на C#, Visual Basic, F#, C++, HTML, JavaScript, TypeScript, Python и других языках).

А также в этот список входит: упрощенная модульная установка, подключение к облаку, мощные инструменты программирования, интеграция с Git и многое другое. В качестве языка был выбран соответственно объектно-ориентированный язык программирования C#.

Для разработки API были выбраны технология ASP.NET Core Web API: ASP.NET Core представляет собой переписанную и расширенную версию фреймворка ASP.NET, которая работает на платформе .NET Core. ASP.NET Core Web API предоставляет средства для создания легковесных и высокопроизводительных API. Он также поддерживает различные форматы данных и предоставляет гибкую конфигурацию и маршрутизацию.