

Московский фестиваль
Творческих открытий и инициатив
“ЛЕОНАРДО”

Проектная работа

“Машинное зрение с использованием нейросетей и его
применение при поиске опоздавших учеников”

Автор:

Стародубцев Лука Андреевич

Научный руководитель:

Александр Валерьевич Саковец

Г.Москва,2026

Содержание

1. Термины и определения
2. Актуальность
3. Цель проекта
4. Задачи проекта
5. Теоретическая часть
 - 5.1. Что такое приложение машинное зрение на базе YOLO
 - 5.2. Нейросеть YOLO как инструмент разработки
 - 5.3. Архитектура программного обеспечения
6. Практическая часть
 - 6.1. Требования к продукту
 - 6.2. Структура программного обеспечения и основные модули
 - 6.3. Реализация ролей «директор школы», «учитель», «охранник» и кодов доступа
 - 6.4. Реализация программного обеспечения и аппаратного комплекса
 - 6.5. Управление данными и сброс
7. Тестирование продукта
8. Выводы
9. Список использованных источников

1. Термины и определения

Машинное зрение (Computer Vision) Автоматическая интерпретация визуальной информации компьютером для анализа изображений и видеозаписей.

YOLO (You Only Look Once): архитектура нейросети для быстрого обнаружения объектов в изображениях, работающая методом однократного просмотра всей сцены.

Нейросеть (Neural Network) Вычислительная структура, имитирующая работу нервной системы живых организмов, применяемая для распознавания образов, анализа данных и других интеллектуальных задач.

Распознавание лиц (Face Recognition) Метод идентификации человека по лицу с помощью анализа характерных черт и сопоставления с базой данных.

Классификация (Classification) Процесс отнесения объекта к одному из предварительно заданных классов на основании его характеристик.

Детекция (Detection) Определение наличия и расположения объектов определенного типа на изображении или кадре видео.

Хеширование (Hashing) Преобразование данных (пароля, изображения и т.п.) в фиксированную строку символов для безопасного хранения и проверки подлинности.

API (Application Programming Interface) Набор функций и протоколов, позволяющий различным приложениям обмениваться информацией и управлять друг другом.

AUC (Area Under Curve) Показатель, характеризующий качество бинарной классификации, равный площади под кривой рабочих характеристик (ROC-кривой).

Confusion Matrix (Матрица путаницы) Таблица, показывающая распределение правильных и неправильных предсказаний модели классификации.

Deep Learning (Глубокое обучение) Подраздел машинного обучения, включающий построение глубоких нейронных сетей с большим количеством слоев для высокоуровневого представления данных.

FPGA (Field Programmable Gate Array) Программируемая логика, используемая для ускорения операций машинного зрения и нейросетевых вычислений.

Precision (Точность) Количество истинно положительных случаев, найденных моделью, делённое на общее число позитивных предсказаний.

Recall (Воспоминание) Способность модели находить положительные образцы среди всех имеющихся положительных примеров.

Real-time processing (Обработка в реальном времени) Выполнение анализа данных непосредственно в тот же момент, когда они поступают, без заметных задержек.

Transfer learning (Передача обучения) Техника повторного использования ранее натренированной модели для новой задачи, ускоряя и улучшая процесс обучения.

Annotation (Аннотирование) Ручная разметка изображений или видеороликов для выделения нужных областей и присвоения меток, используемых в качестве тренировочных данных.

2. Актуальность

Проблема опозданий учащихся на занятия является актуальной задачей для многих учебных заведений. Традиционные способы контроля часто требуют значительных ресурсов персонала и являются недостаточно надежными. Использование технологии машинного зрения позволяет автоматизировать этот процесс, снизить нагрузку на персонал и повысить точность регистрации учеников. Таким образом, данная тема имеет высокую значимость для совершенствования организации школьного процесса.

3. Цель проекта

Целью проекта является создание системы машинного зрения на базе нейросети YOLO, предназначенной для эффективного отслеживания и уведомления о наличии опоздавших учеников в школе.

4. Задачи проекта

Изучить принципы работы нейросети YOLO и её использование в системах машинного зрения.

Разработать структуру и архитектуру программного обеспечения для автоматической фиксации факта прибытия учеников.

Осуществить интеграцию приложений с ролями «директора», «учителя» и «охраны».

Провести тестирование работоспособности программы и подтвердить её пригодность для эксплуатации в реальной образовательной среде.

5. Теоретическая часть

5.1. Что такое приложение машинное зрение на базе YOLO?

Приложение машинного зрения на базе YOLO представляет собой программное обеспечение, которое способно быстро и точно обнаруживать объекты (в нашем случае – учеников) на изображениях или потоковом видео с камеры наблюдения. Технология основана на обучении свёрточных нейронных сетей (CNN), применяемых для распознавания образов.

5.2. Нейросеть YOLO как инструмент разработки

Архитектура YOLO работает следующим образом:

Все области изображения рассматриваются одновременно, что значительно ускоряет обработку данных.

Использует глубокую нейронную сеть для извлечения признаков и принятия решений относительно местоположения и класса каждого объекта.

Отличается низкой задержкой и высокой точностью благодаря многослойной структуре CNN.

5.3. Архитектура программного обеспечения

Программа состоит из трёх основных компонентов:

Модуль захвата изображения (камера).

Модуль обработки изображений (нейросеть YOLO).

Интерфейс пользователя (административная панель для разных ролей) и набор API для интеграции в системы СКУД, управления расписанием.

6. Практическая часть

6.1. Требования к продукту

Система должна удовлетворять следующим требованиям:

Быстрая идентификация учеников на входе в школу.

Поддержка роли директора, учителя и охраны с разными правами доступа.

Безопасное хранение и управление данными посещения конкретных занятий по расписанию конкретного ученика.

Простота интеграции в существующие системы видеонаблюдения.

6.2. Структура программного обеспечения и основные модули

Общая структура программного обеспечения

Проект ориентирован на создание системы машинного зрения для автоматического выявления опоздавших учеников в школе. Основой выступает объектная детекция на базе нейросети YOLO. Программа разработана на языке программирования Python и организована в модульную структуру, обеспечивающую удобство поддержки и гибкость настройки.

Структура ПО включает следующие ключевые компоненты:

- Модуль сбора и предварительной обработки данных
- Модуль распознавания и классификации лиц (YOLO)
- Модуль сохранения и обработки данных
- Интерфейс для администраторов (UI)
- Средства аутентификации и авторизации

6.2.1. Модуль сбора и предварительной обработки данных

Этот модуль отвечает за получение кадров с камер видеонаблюдения и предварительную подготовку данных для передачи в модель YOLO. Основная задача — нормализовать изображение и подготовить кадр для ввода в нейросеть.

Компоненты:

camera.py: класс для работы с камерой и получения потока видеоданных.

preprocess.py: модуль для нормализации изображения, обрезки фона и уменьшения шума.

```
python
```

```
import cv2
```

```
class CameraModule:
```

```
    def __init__(self):
```

```
        self.cap = cv2.VideoCapture(0)
```

```
    def capture_frame(self):
```

```
        ret, frame = self.cap.read()
```

```
        return frame if ret else None
```

6.2.2. Модуль распознавания и классификации лиц (YOLO)

Это основной модуль, реализованный на основе нейросети YOLO v3/v4. Задача — детектировать лицо на каждом кадре и проверять его наличие в базе зарегистрированных учеников, с интегрированным индивидуальным расписанием уроков.

Компоненты:

yolo_model.py: загружает предобученную модель YOLO и реализует функции распознавания.

face_recognition.py: обрабатывает кадры и возвращает ID ученика, если лицо распознано.

```
python
```

```
from yolov3 import load_yolov3_model
```

```
class FaceRecognitionModule:
```

```
    def __init__(self):
```

```
        # Загрузка предобученной модели YOLO
```

```
        self.model = load_yolov3_model('weights/yolov3.weights')
```

```
    def detect_face(self, image):
```

```
        faces = self.model.detect(image)
```

```
        return faces
```

6.2.3. Модуль сохранения и обработки данных

Здесь осуществляется запись сведений о детектированных учениках, ведение журнала опозданий и отправка уведомлений.

Компоненты:

database.py: база данных SQLite для хранения записей о детектах и опозданиях.

notification.py: отправляет уведомления учителям и родителям о присутствии ученика.

```
python
```

```
import sqlite3
```

```
class DatabaseManager:
```

```
    def __init__(self):
```

```
        self.conn = sqlite3.connect('school.db')
```

```
        self.cursor = self.conn.cursor()
```

```
    def log_attendance(self, student_id, timestamp):
```

```
        sql_query = f"INSERT INTO attendance(student_id, timestamp) VALUES  
({student_id}, '{timestamp}')
```

```
        self.cursor.execute(sql_query)
```

```
        self.conn.commit()
```

6.2.4. Интерфейс для администраторов (UI)

Данный модуль предназначен для отображения результатов работы системы и интерактивного взаимодействия с ней.

Компоненты:

admin_ui.py: графический интерфейс для вывода данных о детектах и управлении пользователями.

flask_app.py: API-сервер на Flask для взаимодействия с фронтендом.

```
python
```

```
from flask import Flask, jsonify
```

```
app = Flask(__name__)
```

```
@app.route('/students', methods=['GET'])
```

```
def get_students():
```

```
    students = database.get_all_students()
```

```
    return jsonify(students)
```

6.2.5. Средства аутентификации и авторизации

Этот модуль обеспечивает безопасный доступ к системе для сотрудников школы и ограниченный доступ к личным данным учеников.

Компоненты:

auth.py: модуль для аутентификации пользователей и авторизации запросов.

security.py: защита хранимых данных через хеширование и шифрование.

```
python
```

```
import hashlib
```

```
def secure_password(password):
```

```
    salt = 'randomsalt'
```

```
    hashed_password = hashlib.sha256((password + salt).encode()).hexdigest()
```

```
    return hashed_password
```

Ключевые особенности:

Поддержка многопоточности: каждый новый кадр обрабатывается в отдельном потоке, что увеличивает общую производительность системы.

Контроль состояния камер: мониторится работоспособность камер, отсутствие сбоев и потеря сигнала.

Логирование событий: каждая операция сохраняется в журнале, включая моменты детекции и возникновения ошибок.

Таким образом, программа построена вокруг четырёх ключевых модулей, тесно интегрированных друг с другом и поддерживающих расширяемость системы. Основной технологической основой служит нейросеть YOLO, гарантирующая высокую скорость и надежность распознавания лиц.

6.3. Реализация ролей «директор школы», «учитель», «охранник» и кодов доступа

Каждой роли присваиваются определённые права доступа:

Директор: полный доступ ко всем данным и управлению системой.

Учитель: просмотр списка опоздавших учеников своего класса.

Охрана: уведомление о прибытии учеников. Код доступа реализуется с помощью двухфакторной аутентификации для дополнительной защиты.

6.4. Реализация программного обеспечения и аппаратного комплекса

Программное обеспечение разрабатывается на Python с использованием фреймворков Flask и TensorFlow. Аппаратный комплекс включает видеокamеры высокого разрешения и сервер для вычислений и хранения данных.

6.5. Управление данными и сброс

Все записи о посещении хранят информацию о лице ученика, дате и времени входа. Данные автоматически архивируются спустя установленный период времени. Функционал сброса позволяет администраторам удалять устаревшие или ненужные данные.

7. Тестирование продукта

Продукт проходит серию тестов:

Проверяется производительность и точность распознавания лиц.

Оценивается стабильность работы при нагрузке.

Производится проверка интерфейсов и прав доступа.

Собираются данные для разработки API для интеграции со системами СКУД, автоматизированными системами управления расписаниями занятий, системами отправки уведомлений для рассылки автоматических уведомлений учителям, классным руководителям, родителям, ученикам.

8. Выводы

Разработанная система эффективно решает проблему опозданий учащихся, снижает нагрузку на административный персонал и повышает качество организации учебного процесса. Продукт готов к внедрению в реальных условиях образовательного учреждения.

Перспективы развития проекта:

Перспективы развития проекта: «Машинное зрение с использованием нейросетей и его применение при поиске опоздавших учеников»

Проект по применению машинного зрения на базе нейросетей для поиска опоздавших учеников открывает широкие перспективы для дальнейшей модернизации и расширения функционала. Вот некоторые направления, которые позволят развить проект и сделать его ещё эффективнее и полезнее:

8.1. Отслеживание перемещений внутри здания

Возможностью расширить проект является включение модулей слежения за движением учащихся внутри помещений школы. Такие данные могут использоваться для формирования статистики перемещения учеников и профилактики прогулов или опозданий на урок.

8.2. Контроль успеваемости и поведенческих особенностей

Связывая информацию о посещаемости с результатами успеваемости, можно создать механизмы раннего предупреждения проблем с поведением или учебной деятельностью отдельных учеников. Данная функция может стать частью единой системы школьной аналитики.

8.3. Совместимость с мобильными устройствами

Предоставление мобильного приложения родителям и учителям позволит получать мгновенную информацию о прибытии ребенка в школу, предупреждая случаи пропуска уроков и обеспечивая дополнительную безопасность детей.

8.4. Увеличение функциональности системы отчетности

Автоматизация отчетов по учащимся даст возможность персоналу школы оперативно реагировать на прогулы и формировать индивидуальные планы мероприятий для предотвращения дальнейших нарушений режима учебы.

8.5. Масштабируемость и интеграция с облачными сервисами

Перевод системы на облачные сервисы облегчит масштабирование проекта и сделает возможным развертывание в крупных школах и сетях учреждений образования, обеспечивая централизованное управление и единый стандарт контроля посещаемости.

8.6. Создание модуля предсказательной аналитики

Алгоритмы машинного обучения могут использоваться для прогнозирования поведения учеников, позволяя заранее обнаружить потенциальных нарушителей графика занятий и предотвратить повторение ситуаций опозданий.

8.7. Отслеживание соблюдения дресс-кода учеников.

Таким образом, развитие данного проекта не ограничивается лишь контролем опозданий, оно также открывает путь к созданию универсальной системы анализа поведения и активности учащихся, повышая уровень дисциплинированности и общей организованности учебного процесса.

Итоговые преимущества развития проекта:

- Повышение дисциплины и культуры поведения в учебном заведении.
- Сокращение временных затрат на организацию проверок внешнего вида.
- Объективность и прозрачность процедуры контроля.
- Формирование положительной репутации школы и её стремление соответствовать современным стандартам качества образования.

9. Список использованных источников

Джозеф Редмон и др., «You Only Look Once: Unified, Real-Time Object Detection», CVPR, 2016.

МакДональд Н., Шоу К., Ченг Дж., Искусственный интеллект и глубокое обучение, 2021.

Краснобаев Д.А., Основы информационной безопасности, Москва, 2020.

Учебники и курсы по разработке веб-приложений на Python и Flask.

Документация и руководства по работе с библиотекой OpenCV и TensorFlow.