



Государственное бюджетное общеобразовательное учреждение города Москвы «Школа №2070».
Федеральное государственное бюджетное образовательное учреждение высшего образования
«МИРЭА - Российский технологический университет» Детский технопарк «Альтаир»

Тема:

Автоматизированная система Дронов беспилотной полиции Senlife

Фильчагин Савелий Алексеевич
ученик 10 «Б» класса ГБОУ Школа
2070

Руководитель:
Преподаватель Детского технопарка «Альтаир» РТУ МИРЭА
Петров Даниил Русланович

Почта для связи:
filchaginwork@gmail.com
Детский технопарк «Альтаир»



Актуальность проекта:

Мой проект – это беспилотные автоматизированные дроны с машинным обучением, которые могут быстро находить людей и безопасно раскрывать преступления, реагировать на аварии в современном мире.



Рис. 5 Дрон DJI, пример существующих моделей



Рис. 4 Использование дронов полицией

Детский технопарк «Альтаир»

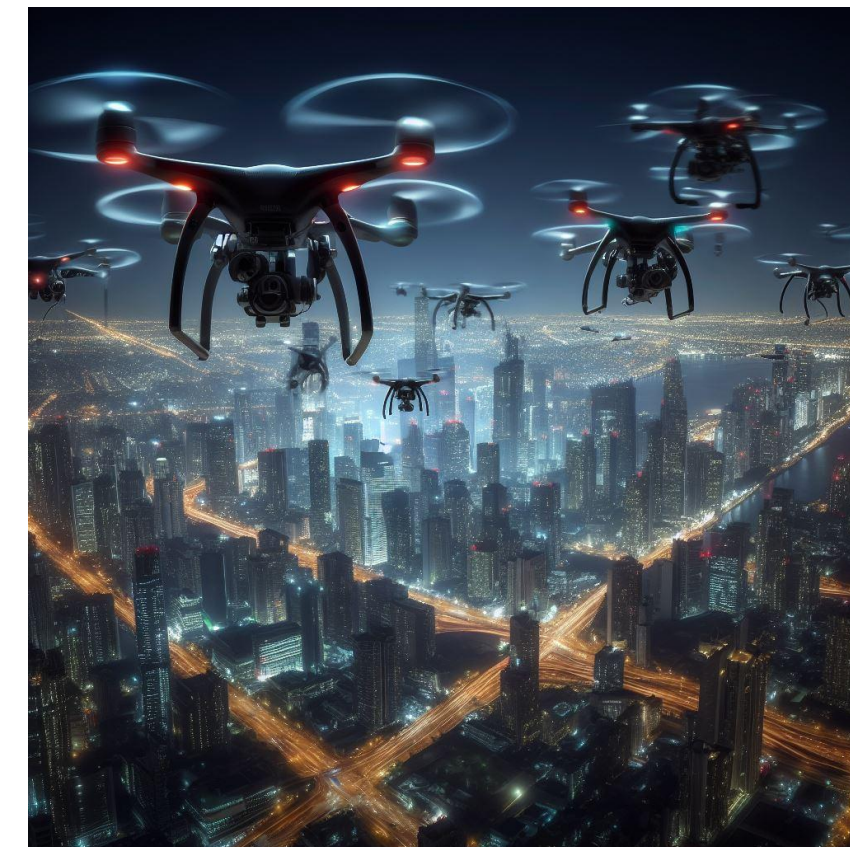


Рис. 6 Пример использование групп дронов



Цель работы:

Разработка прототипа системы беспилотных дронов, способных автоматически обнаруживать людей, помогать проводить анализ инцидентов, местности с помощью машинного обучения и способных применяться в системе беспилотной полиции.



Задачи работы:

- Разработка модели дрона, включая подготовку и сборку микросхем, плат, датчиков, а также их взаимное подключение.
- Написание модели машинного обучения для автоматического обнаружения людей.
- Разработка веб-сайта с сервером для хранения, передачи и анализа данных, полученных с дрона, в реальном времени.
- Проведение тестирования всей системы с целью обеспечения ее эффективной работы и соответствия поставленным задачам.



Этапы проекта:

1 стадия:(Модель)

- Построение корпуса частями и скрепление.
- Печать модели.
- Закупка материала.
- Подключение и соединение моторчиков и лучей
- Подключение камеры и “гимбла” (подвес для камеры) также подключение датчиков.
- Установка микросхем в корпус.

2 стадия:(Разработка)

- Изучение структуры Машинного обучения.
- Написание кода для Машинного обучения.
- Создание системы машинного обучения для анализа информации на сервере.
- Написание сайта.

3 стадия:(Тесты)

- Тестирование всех систем



Диаграмма Ганта

Задачи	Сентябрь	Октябрь	Ноябрь	Декабрь	Январь	Февраль
Разработка идеи	■	■				
Проектирование дрона		■	■	■		
Разработка машинного обучения			■	■	■	■
Разработка сайта			■	■	■	■
Тестирование					■	■



Реализация: Стадия 1

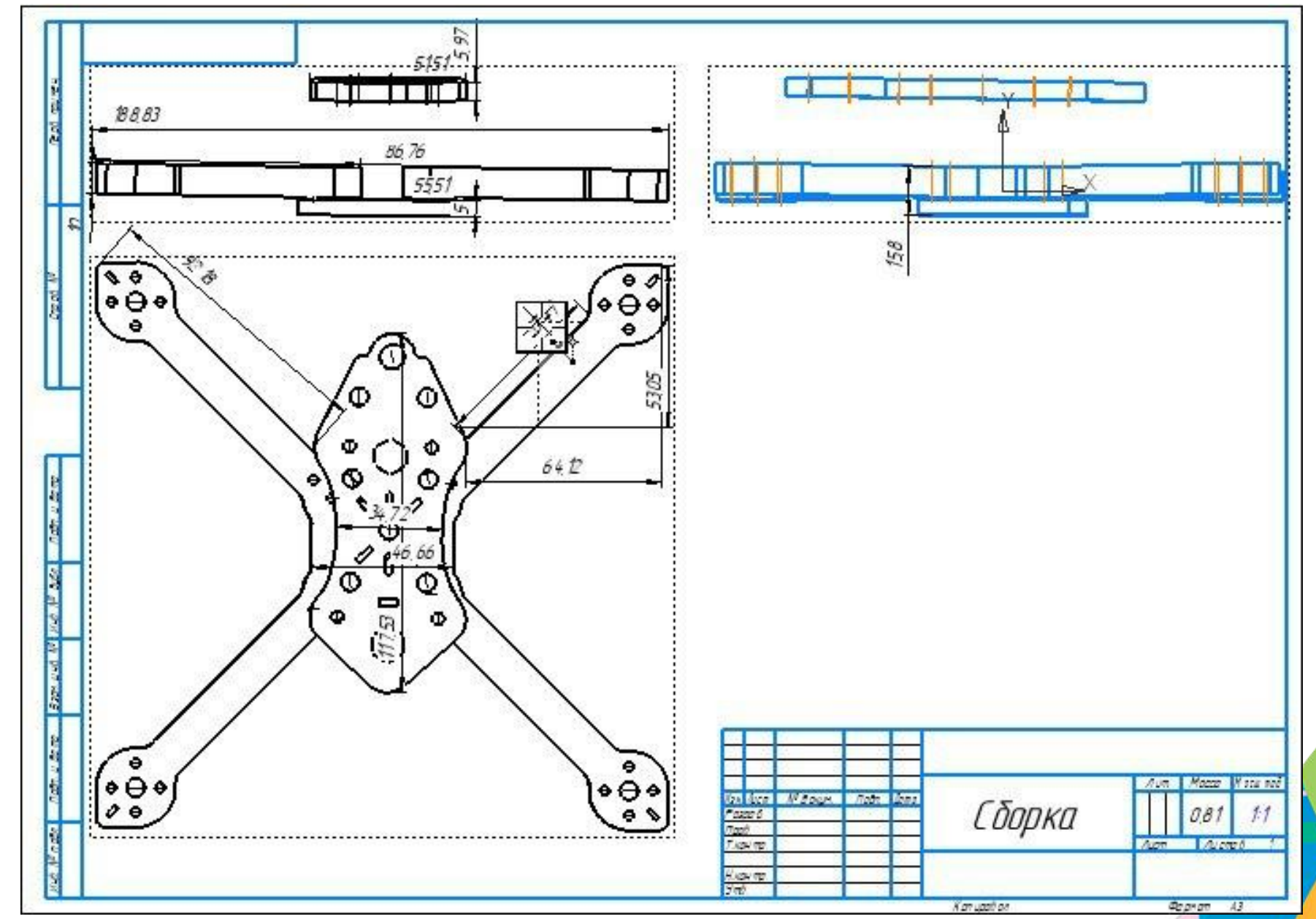
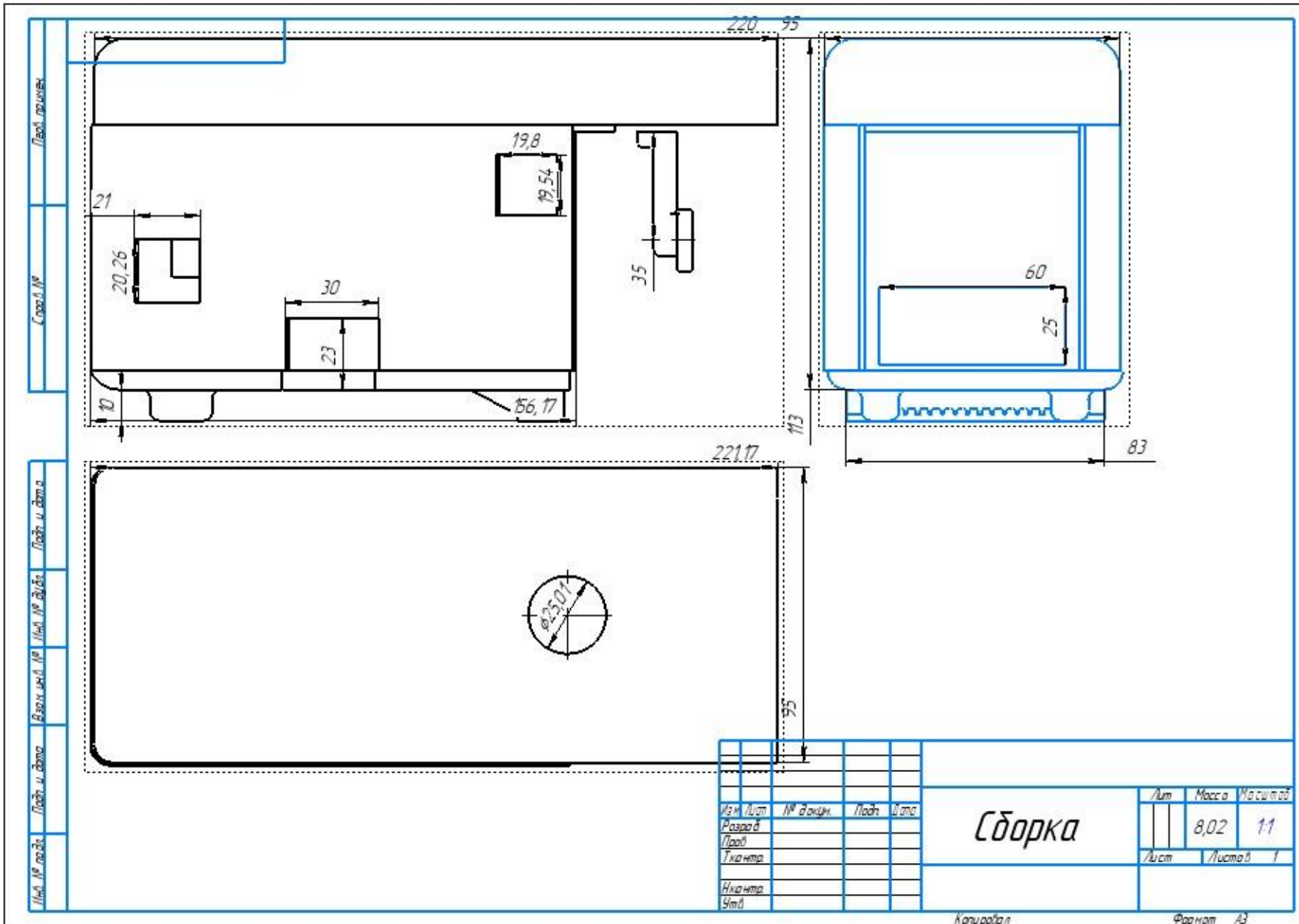


Рис. 2 Чертежи разработанных моделей дронов



Реализация: Стадия 1

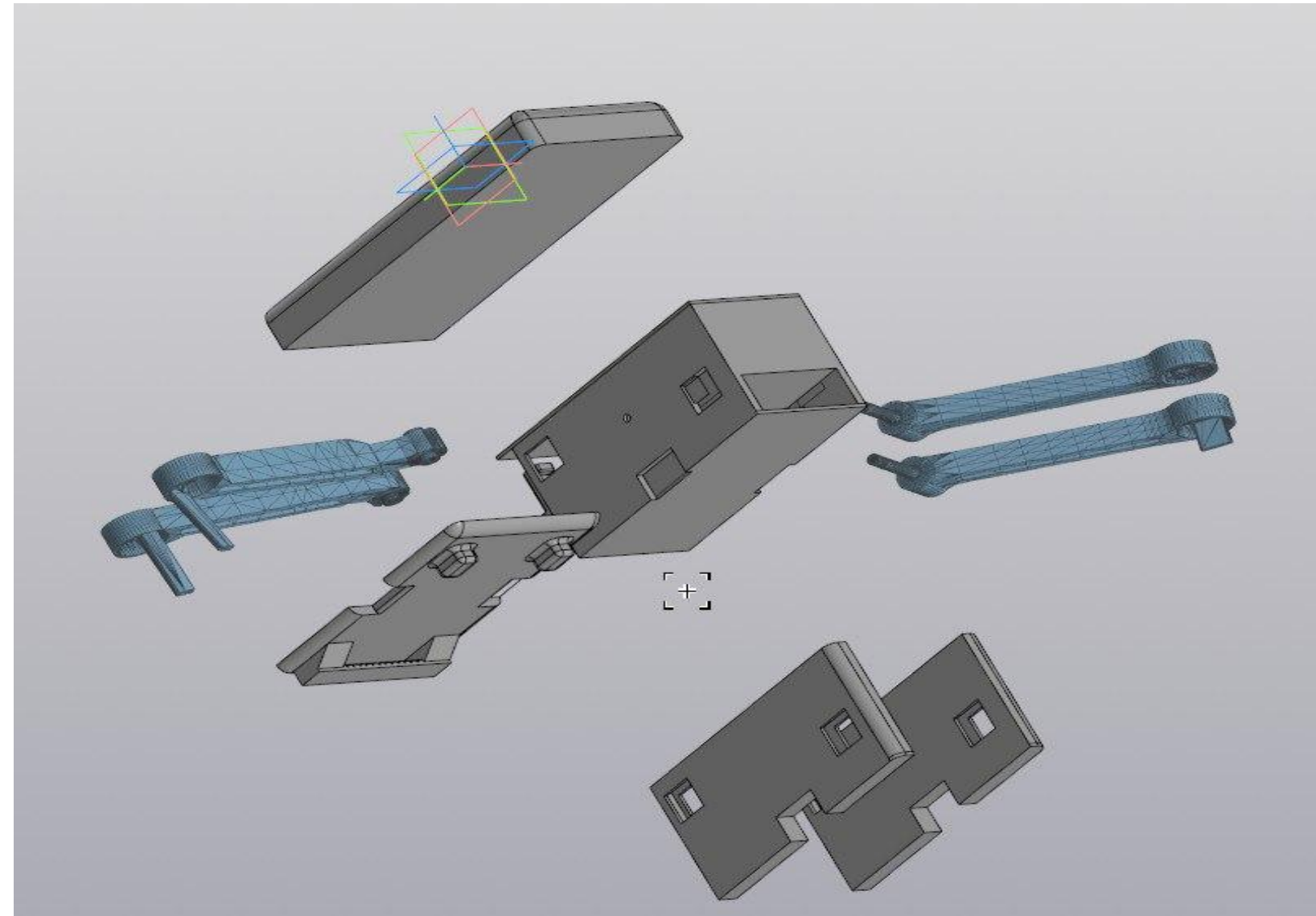
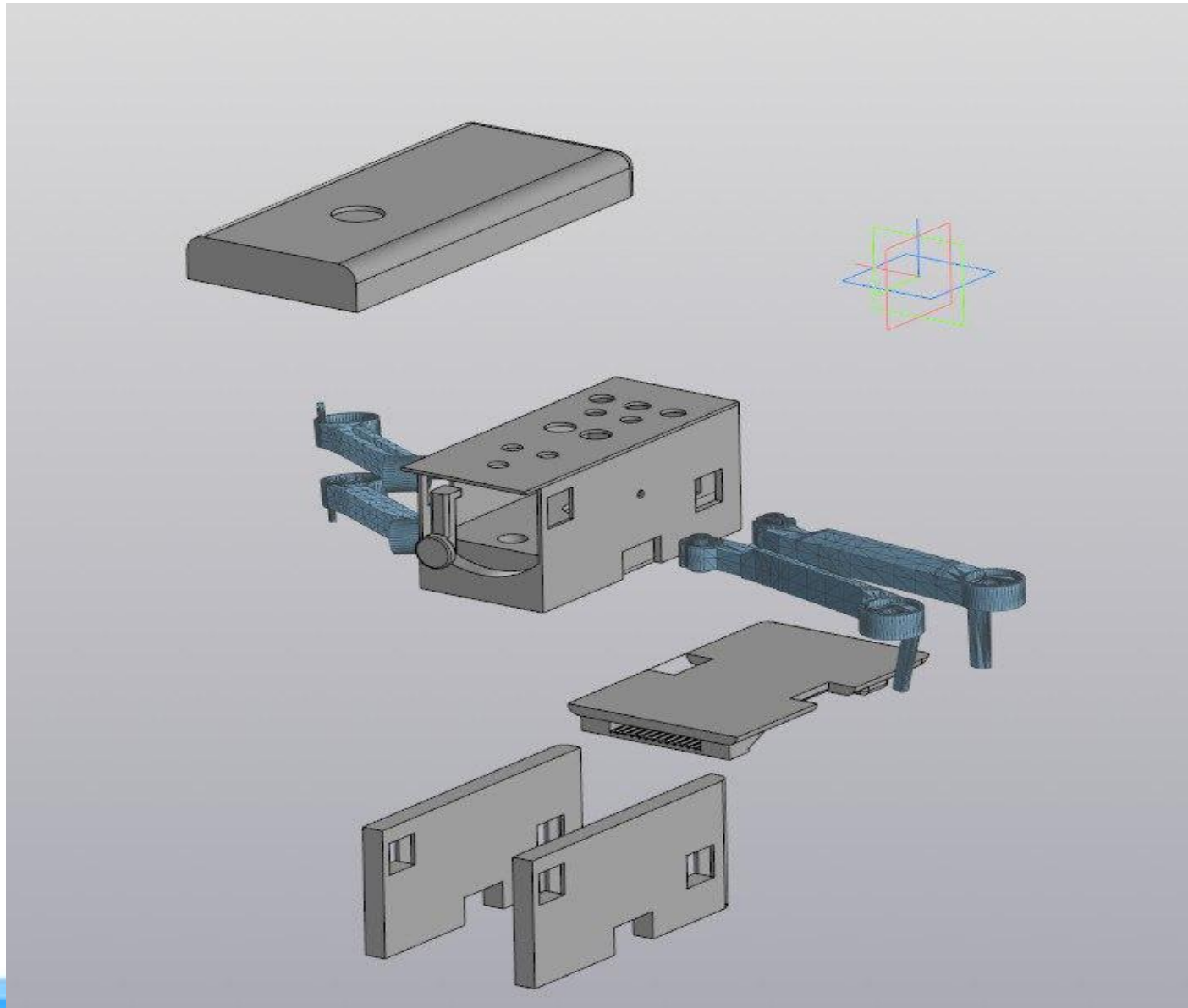


Рис. 3 Основная модель дрона



Реализация: Стадия 1

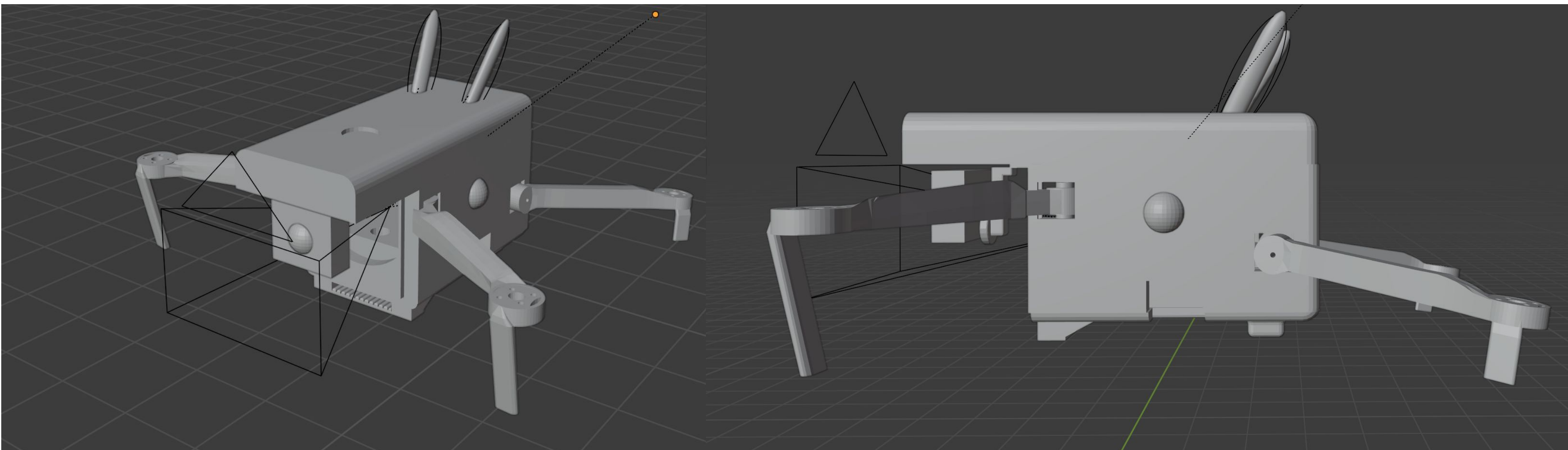


Рис. 5 Основная сборка модель дрона

Детский технопарк «Альтаир»



Реализация: Стадия 2

- Начальный этап построения сайта

Select Categories

Task: Патрулирование ▾	Main Category: Люди ▾	Second Category: ▾	Third Category: ▾	Fourth Category: ▾
Submit				



Реализация: Стадия 2

- Написание эскиза модели машинного обучения
- Написание эскиза модели сайта
- Настройка сервера

```
video.release()

"time" не определено
# обнаружение объектов на изображении и добавление обрамления (квадратика)
CodiumAI: Options | Test this function
def detect_objects_and_draw_boxes(image):
    input_image = cv2.resize(image, (224, 224)) # Примерный размер для MobileNetV2
    input_image = tf.keras.applications.mobilenet_v2.preprocess_input(input_image)
    predictions = model.predict(np.expand_dims(input_image, axis=0))

    # Предположим, что model.predict возвращает индекс класса (с наибольшей вероятностью)
    predicted_class = np.argmax(predictions)

    # Допустим, что объект с предсказанным классом имеет ограничивающий прямоугольник (x, y, w, h)
    x, y, w, h = 10, 10, 50, 50 # Замените эту информацию реальными данными

    # Рисуем обрамление (квадратик) вокруг объекта
    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)

# Проходим по всем фотографиям и добавляем обрамление
for image in images:
    detect_objects_and_draw_boxes(image)

# Отображение изображений с обрамлением
for image in images:
    cv2.imshow("Image with Boxes", image)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

# Обработка видео с обрамлением
for video in videos:
    while True:
        ret, frame = video.read()
        if not ret:
            break

        # Вызываем функцию для обнаружения объектов и добавления обрамления
        detect_objects_and_draw_boxes(frame)

        # Отображаем видео с обрамлением
        cv2.imshow("Video with Boxes", frame)

        # Добавляем небольшую задержку (например, 0.1 секунды)
```

```
# Инициализация переменных ID
(constant) ID_HUMAN: None

ID_HUMAN = None #1 только информация поступает с сайта либо human/animals
ID_NATION = None
ID_EYECOLOR = None
ID_ANIMAL = None #1 только информация поступает с сайта либо human/animals

Codium: Refactor | Explain | Generate Docstring | X
@app.route('/update_ids', methods=['POST'])
def update_ids():
    global ID_HUMAN, ID_NATION, ID_EYECOLOR, ID_ANIMAL

    # Получение новых ID из POST-запроса
    data = request.get_json()
    ID_MODE = data.get('ID_MODE')
    ID_TASK = data.get('ID_TASK')
    ID_HUMAN = data.get('ID_HUMAN') #1 только информация поступает с сайта либо human/animals
    ID_NATION = data.get('ID_NATION')
    ID_EYECOLOR = data.get('ID_EYECOLOR')
    ID_ANIMAL = data.get('ID_ANIMAL') #1 только информация поступает с сайта либо human/animals
    return "ID успешно обновлены!"

Codium: Explain
class YOLO:
    Codium: Refactor | Explain | Generate Docstring | X
    def __init__(self, weights, cfg):
        self.net = cv2.dnn.readNet(weights, cfg)
        self.layer_names = self.net.getLayerNames()
        self.output_layers = [self.layer_names[i[0] - 1] for i in self.net.getUnconnectedOutLayers()]

    Codium: Refactor | Explain | Generate Docstring | X
    def detect(self, frame):
```

```
if parameters['sub_category'] == ID_NATION:
    yolo_model = NationYOLO()
elif parameters['sub_category'] == ID_EYECOLOR:
    yolo_model = EyeColorYOLO(parameters['sub_sub_category'])
else:
    yolo_model = HumanYOLO(parameters['sub_category'])
elif parameters['main_category'] == ID_ANIMAL:
    yolo_model = AnimalYOLO(parameters['sub_category'])

cap = cv2.VideoCapture(0)
ret, frame = cap.read()
if not ret:
    return "Не удалось захватить кадр. Проверьте подключение камеры."

outs = yolo_model.detect(frame)

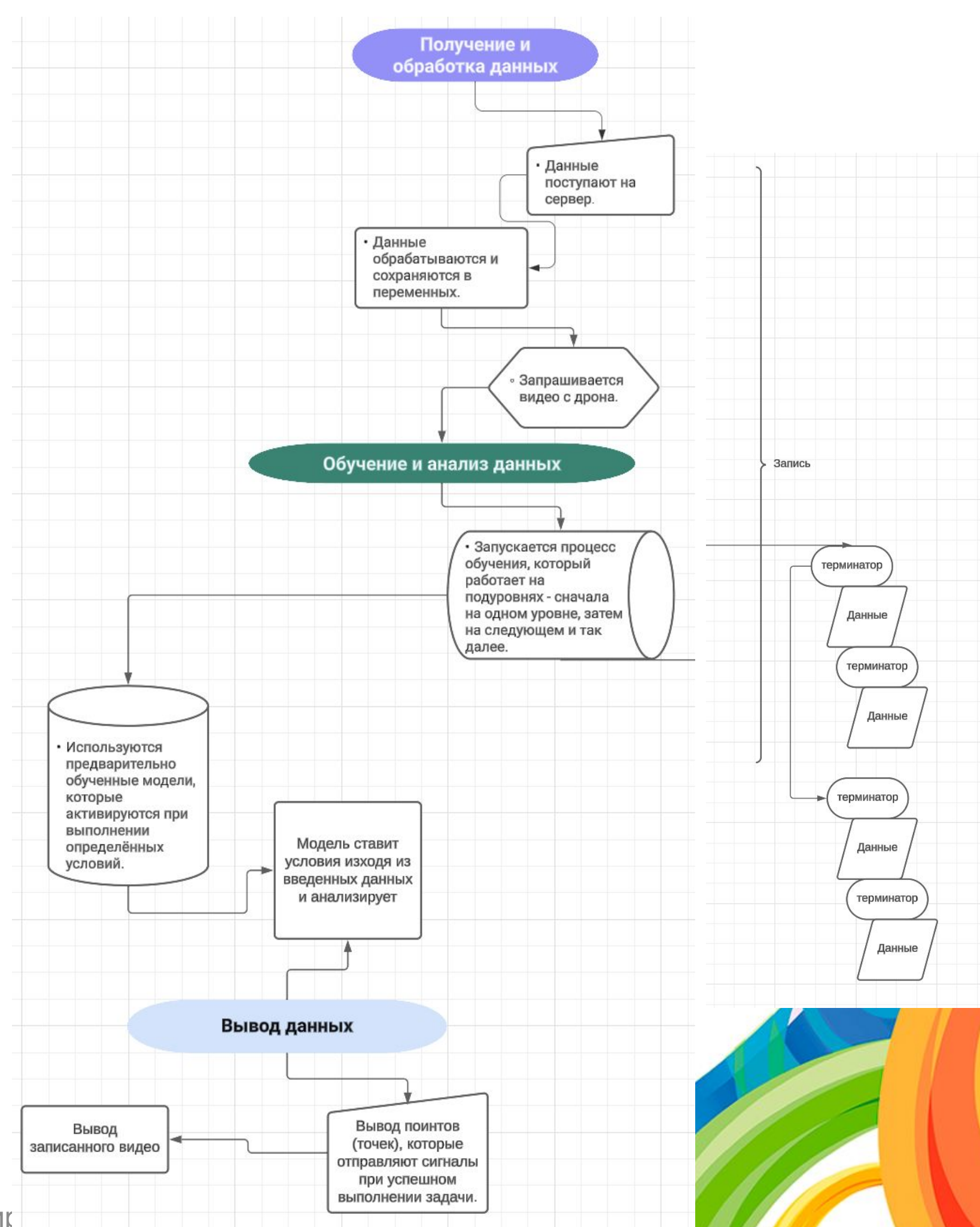
print("Код работает успешно! Конец программы!")

results = []
flag = False
for out in outs:
    for detection in out:
        scores = detection[5:]
        class_id = np.argmax(scores)
        confidence = scores[class_id]
        if confidence > 0.5:
```



Реализация: Стадия 2

- Схема алгоритма работы кода





Библиотеки:

- OpenCV
- tensorflow
- фреймворк YOLO
- Ultralytics
- Keras





Прогнозируемые результаты:

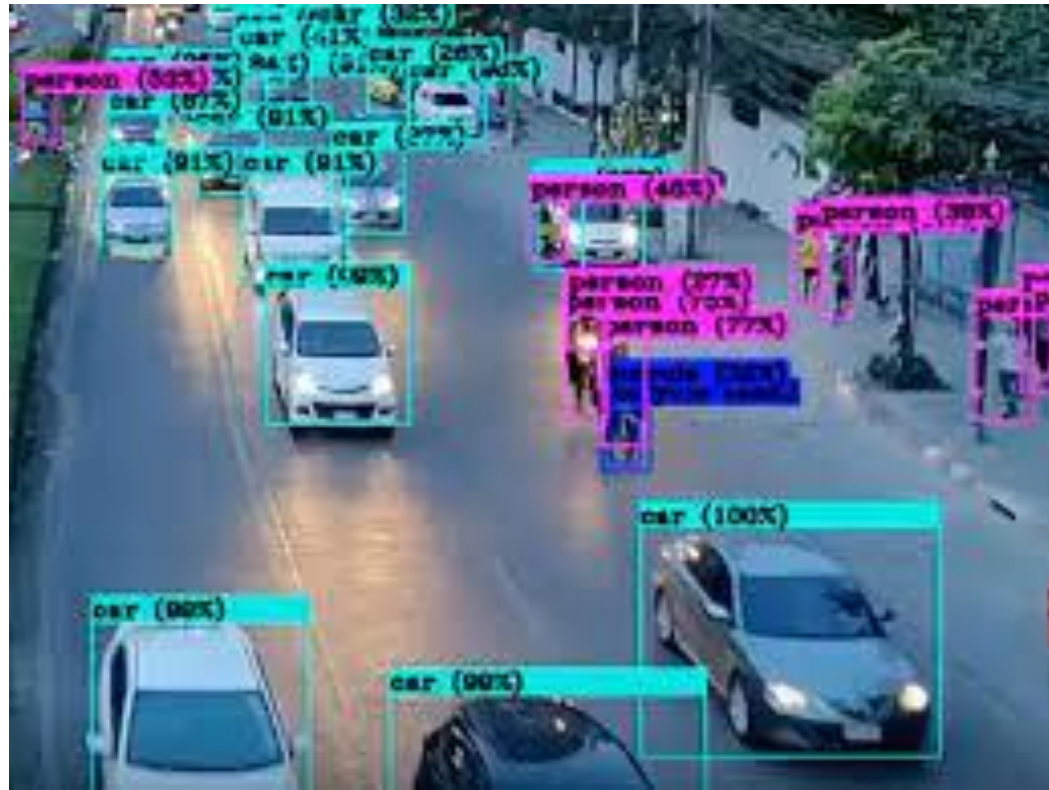


Рис. 7 Пример анализа видео и фотографий



Рис. 7 Пример анализа видео и фотографий



Рис. 8 Пример анализа видео и фотографий



Тестирование: Стадия 3

- Из-за технических сложностей, было принято решение создать тестовый Дрон frv.

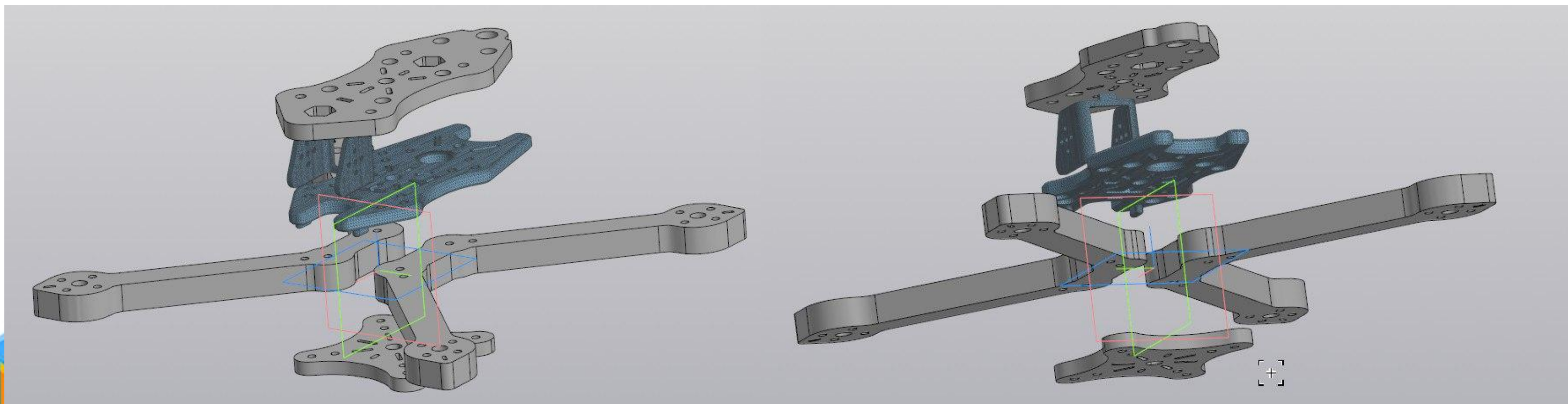


Рис. 3 Вторая frv модель дрона



Дальнейшие развитие проекта:

- Интеграция очков Goggles для удаленного доступа к видеопотоку дронов, предоставляя информацию врачам и полиции о пострадавших.
- Возможность решения вопросов при дорожно-транспортных происшествиях и других инцидентах.
- Создание 3D-изображений местности для более точной навигации и анализа.
- Разработка возможности обнаружения сигналов спасателей с использованием звуковых датчиков, также подключение к системе SOS полиции.
- Создание 3-х видов дронов



Рис. 10 Очки Goggles

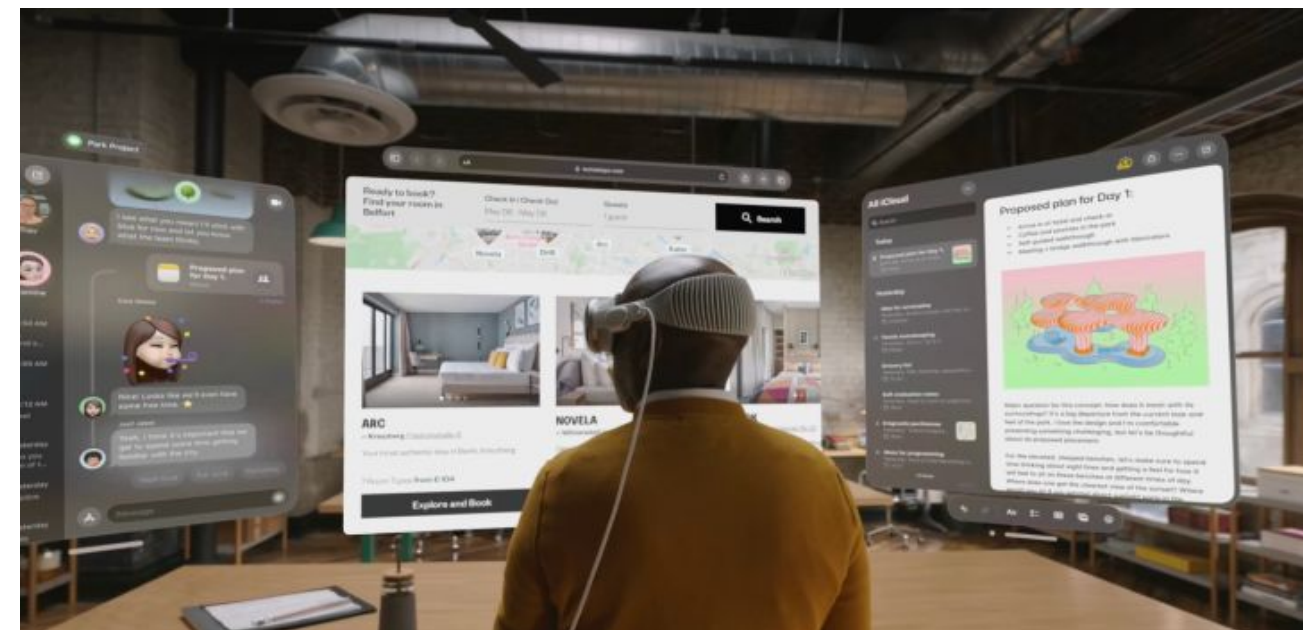


Рис. 9 Пример использования очков



Вывод:

- Создан прототип системы дронов Sendlife обладающий возможностью полета в труднодоступных местах, возможностями беспилотной полиции.
- Получены навыки в разработке системы определения человека и построение системы автоматизированной беспилотной полиции.
- Получен опыт работы с моделями в среде “Компас 3Д”, с кодом в программе “VS Code” и других системах .





Список литературы:

Книги:

- "Практическое машинное обучение с использованием Scikit-Learn, Keras и TensorFlow" от Аурелиана Жерона
- "Распознавание образов и машинное обучение" от Кристофера М. Бишопа
- "Элементы статистического обучения" от Тренора Хэсти, Роберта Тибшерани и Джерома Фридмана
- "Глубокое обучение" от Иэна Гудфеллоу, Йошуа Бенжио и Аарона Курвиля
- "Python и машинное обучение" от Себастьяна Рашки

Статьи о машинном обучении:

- [Машинное обучение: базовые понятия](<https://habr.com/ru/company/ods/blog/322626/>)
- [Как работает машинное обучение: введение](<https://habr.com/ru/company/skillfactory/blog/499340/>)
- [Глубокое обучение: основы и применение](<https://habr.com/ru/company/it-grad/blog/340508/>)
- [Машинное обучение в бизнесе: как применять для решения задач](<https://habr.com/ru/post/416905/>)
- [Как выбрать алгоритм машинного обучения: руководство](<https://habr.com/ru/company/otus/blog/348260/>)