Структурное подразделение “Центр детского технического творчества детей “НОВАпарк” государственного бюджетного учреждения дополнительного профессионального образования Самарской области 

“Новокуйбышевский ресурсный центр”

**Всероссийский конкурс научно-технологических проектов**

**“Большие вызовы”**

**2023-2024 учебный год**

Конкурсное направление: Беспилотный транспорт и логистические системы

**Тема: «Разработка системы управления для БПЛА на основе машинного зрения»**

Автор:

Туйзюков Артур

Класс: 10

Руководитель:

Спиридонов Алексей Андреевич,

педагог ДО ЦТТД “НОВАпарк”

г.о. Новокуйбышевск, 2024 год

# АННОТАЦИЯ

В настоящее время активное развитие получили беспилотники, а также технологии искусственного интеллекта. Комбинирование двух этих быстро развивающихся сфер и создание на основе этого нового способа управления дронами с использованием жестов на основе машинного зрения позволяет показать новые возможности для развития этих направлений и разнообразить как орудия для работы с дронами, так и расширить область применения языка python.

# КЛЮЧЕВЫЕ СЛОВА

БЕСПИЛОТНИКИ, МАШИННОЕ ЗРЕНИЕ, PYTHON, ЖЕСТЫ, УПРАВЛЕНИЕ, ДРОН.

# ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ [3]

ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ [4]

1.1 Машинное зрение и принципы его работы [4]

1.2 БПЛА и история их появления [5]

1.3 Поиск подходящих библиотек машинного зрения и БПЛА [6]

ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ [8]

2.1 Принцип работы системы управления [8]

2.2 Написание кода [8]

2.3 Итоговые жесты для управления [9]

ЗАКЛЮЧЕНИЕ [12]

ПРИЛОЖЕНИЕ 1 [13]

ПРИЛОЖЕНИЕ 2 [14]

БИБЛИОГРАФИЧЕСКИЙ СПИСОК [16]

# ВВЕДЕНИЕ

**Актуальность выбранной темы:** в настоящее время особенно стремительно развиваются технологии искусственного интеллекта и способы их применения, а также большую востребованность сейчас имеет применение БПЛА для различных целей, объединение двух этих сфер делает выбранную тему актуальной.

**Цель проекта:** создание системы управления для БПЛА на основе машинного зрения.

**Задачи проекта:**

- изучение принципов работы машинного зрения;

- изучение принципов управление БПЛА с помощью программного обеспечения;

- выбор библиотек и БПЛА, подходящих для реализации системы управления;

- создание системы управления для БПЛА на основе машинного зрения;

**Объект исследования:** БПЛА и машинное зрение.

**Предмет исследования:** управление БПЛА с использованием машинного зрения.

**Методы исследования:**

Эмпирический метод исследования – поиск литературы по данной теме.

Теоретический метод исследования – расчет способов достижения необходимой задачи.

Практико-ориентированный метод – разработка системы управления.

**Теоретическая значимость:** изучение систем управления БПЛА, принципов работы машинного зрения и возможности их объединения.

**Практическая значимость**: функционирующая система управления для БПЛА DJI Ryze Tello на основе машинного зрения с использованием библиотек opencv и mediapipe.

# ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## 1.1 Машинное зрение и принципы его работы

Машинное зрение — это технология в области искусственного интеллекта, включающая в себя получение изображений объектов реального мира, их обработка и использование полученных данных для решений разного рода прикладных задач без участия (полного или частичного) человека.

Для захвата изображения используются фоточувствительные датчики, цифровые камеры, ультрафиолетовые или инфракрасные камеры. Все эти устройства захватывают изображение и преобразуют его в цифровую информацию. Затем цифровые данные, поступающие от видеооборудования, анализируются с помощью алгоритмов обработки изображений.

Обработка и анализ изображений сосредоточены на работе с различными способами их преобразований, например, попиксельные операции увеличения контрастности, операции по выделению краёв, устранению шумов или геометрические преобразования, такие как вращение изображения. С помощью данных преобразований определяются границы объектов на фото, выявляются их признаки (размер, цвет, форма и др.). На основе полученной информации происходит анализ изображения, выявление характерных зависимостей и соответствий с ожидаемым результатом. Данные операции предполагают, что обработка/анализ изображения действуют независимо от содержания самих изображений.

После сбора и анализа информации на основе машинного зрения система по алгоритмам предпринимает какое-либо действие, ожидаемое к исполнению в соответствии с определённой ситуацией.

Машинное зрение в основном сосредотачивается на промышленном применении с использованием автономных роботов и систем визуальной проверки и измерений, требующих осуществление обработки полученных данных и программное управление роботом в реальном времени.

## 1.2 БПЛА и история их появления

БПЛА (беспилотный летательный аппарат) — [летательный аппарат](https://ru.wikipedia.org/wiki/%D0%9B%D0%B5%D1%82%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%B0%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82) без [экипажа](https://ru.wikipedia.org/wiki/%D0%AD%D0%BA%D0%B8%D0%BF%D0%B0%D0%B6) на [борту](https://ru.wikipedia.org/wiki/%D0%92%D0%BE%D0%B7%D0%B4%D1%83%D1%88%D0%BD%D0%BE%D0%B5_%D1%81%D1%83%D0%B4%D0%BD%D0%BE), управление которым осуществляется дистанционно посредством радиоэлектроники. Беспилотные летательные аппараты имеют ряд преимуществ перед классической авиацией.

Главным преимуществом является отсутствие на борту беспилотника пилота и других людей, что позволяет избежать человеческих жертв в результате чрезвычайных ситуаций. К преимуществам также относятся значительно меньшие габариты, позволяющие не только иметь большую маневренность, но и уменьшающие стоимость аппарата.

Недостатки беспилотных летательных аппаратов связаны с новизной используемых технологий. Для беспилотников не до конца решены многие вопросы связи, посадки и спасения аппарата. БПЛА имеют недостаточную гибкость применения по сравнению с классическими летательными аппаратами, а их уровень надежности уступает традиционной авиации.

История создания БПЛА начинается в воде, а не воздухе, когда в 1899 году Никола Тесла изобрел и сконструировал первый в мире радиоуправляемый кораблик, что положило начало развитию дистанционно управляемых аппаратов. Следующий беспилотник, созданный военным инженером Чарльзом Кеттерингом в 1910 году, уже стал летательным и оснащался часовым механизмом, который через определённое время сбрасывал крылья, в результате чего он пикировал на вражеские силы.

Прорывом в области БПЛА стал беспилотник многократного использования под названием DH.82B Queen Bee. Разработанный инженерами Объединённого Королевства он находился в использовании ВВС Великобритании с 1934 по 1943 год. Данный БПЛА представлял собой отреставрированную модель биплана Fairy Queen, которым управляли дистанционно. Но данному беспилотнику было суждено стать самолетом-мишенью для будущих асов и зенитчиков.

Наработки в области беспилотных летательных аппаратов не могли оставаться незамеченными Германией, СССР и США во время Второй Мировой войны, вследствие чего свет увидели самые разные прототипы беспилотников. Несмотря на бурное развитие направления дистанционно управляемых аппаратов в СССР, на данный момент ведущим лидером в разработке и применении беспилотников считается США.

## 1.3 Поиск подходящих библиотек машинного зрения и БПЛА

В настоящий момент рынок любительских БПЛА заполонен многочисленными вариациями моделей различных конструкций. В результате изучения характеристик беспилотников и возможности использования их программного обеспечения, был выбран дрон DJI Ryze Tello. Поскольку он представляет собой доступный вариант компактного безопасного беспилотника, с поддерживаемой библиотекой djitellopy, с помощью которой можно управлять БПЛА программно. Библиотека djitellopy для управления дроном имеет в своем распоряжении набор команд, одна из которых будет сообщаться беспилотнику в соответствии с распознанным машинным зрением жестом.

Лидером в подходящей для осуществления проекта сфере машинного зрения является библиотека opencv, разрабатываемая компанией Intel. Библиотека имеет имеет большой арсенал для работы с машинным зрением, она в совокупности с другой, поддерживаемой компанией Google библиотекой mediapipe, которая обучена на большом количестве изображений, способна осуществлять захват рук на изображении и анализировать положение не только руки в целом, но и отдельных ее пальцев, с помощью чего в системе будет реализовано распознавание различных жестов.

Для написания кода будет использоваться высокоуровневый язык программирования python, поскольку он активно поддерживается, предоставляет доступ к различным библиотекам, в том числе нужным нам djitellopy, opencv, mediapipe, а также является простым в использовании и освоении. В качестве среды разработки будет использоваться приложение PyCharm, позволяющее вести работу с программами, написанными на языке python.

# 

# ВЫВОДЫ ПО 1 ГЛАВЕ

В процессе изучения литературы по данной теме были изучены методы работы машинного зрения, методы дистанционного управления БПЛА. Выбраны язык программирования, среда разработки, основные библиотеки и дрон, необходимые для создания системы управления БПЛА на основе машинного зрения.

# 

# ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

## 2.1 Принципы работы системы управления

Перед использованием системы управления необходимо установить связь между компьютером и дроном посредством сети wifi. Далее создадим новый файл для исполнения python кода с расширением .py в среде разработки PyCharm и реализуем систему управления для БПЛА на основе машинного зрения.

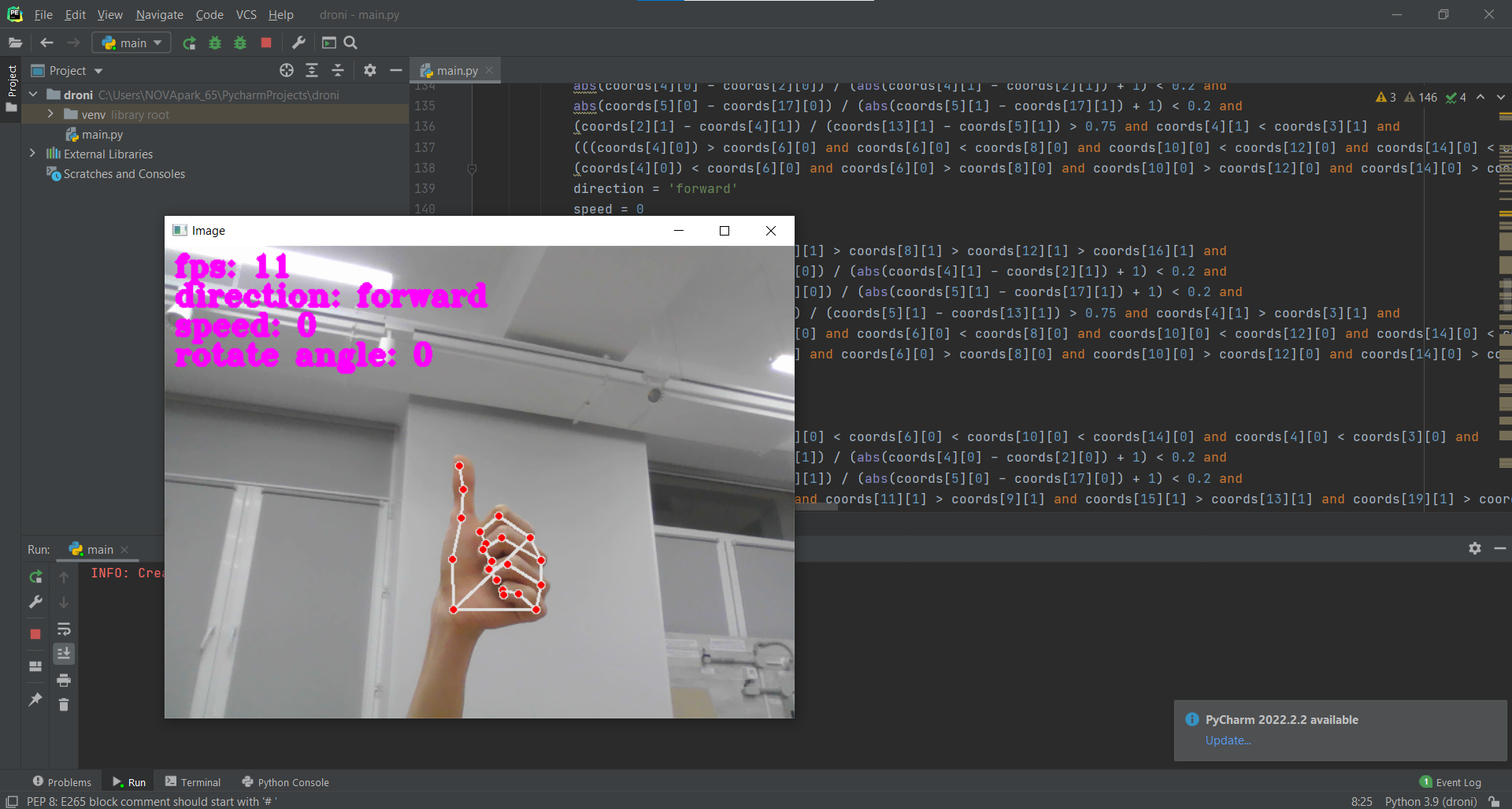
Реализацию системы управления начнем с выполнения захват камеры, необходимой для показа жестов, это может быть как встроенная камера ноутбука, так и отдельное устройство. Далее для определения жестов осуществим захват и трекинг рук с помощью технологий машинного зрения. Определить показываемый жест поможет сравнение координат опорных точек относительно друг друга. В соответствии с опознанным жестом сохраним нужную информацию в переменные и вызовем функцию для сообщения команды управления дрону, используя сохраненные данные.

## 2.2 Написание кода

В соответствии с описанным алгоритмом напишем код на языке python по следующему плану (образец кода представлен в приложении 2):

1. Подключение библиотек djitellopy, opencv, mediapipe и time (позволяет проводить операции со временем, понадобится для вычисления кадров видеопотока, обрабатываемых программой в секунду). Используемые версии библиотек и языка python смотреть в приложении 1.
2. Осуществление захвата видеокамеры посредством функции VideoCapture библиотеки opencv.
3. Выполнение первичного захвата рук в видеопотоке и получение координат опорных точек кисти, используя модуль solutions библиотеки mediapipe.
4. Анализ расположения точек относительно друг друга, их сравнение в соответствии с эмпирически установленным эталоном взаиморасположения.
5. Распознавание жеста и сохранение направления, скорости движения, угла поворота относительно первоначального положения в переменные direction, speed, rotate angle соответственно.
6. Сообщение команды дрону с использованием функций библиотеки djitellopy в соответствии с распознанным жестом и данными переменных direction, speed, rotate angle.

Для удобства использования отобразим на видеопотоке (рис. 1) количество кадров видеопотока, обрабатываемых программой в секунду (fps), текущее направление движение дрона (direction), скорость движения (speed) и угол текущего поворота относительно первоначального положения (rotate angle). Также соединим между собой опорные точки кисти, по которым производится распознавание жестов, добавим дополнительное соединение фаланг указательного и большого пальцев при определении скорости дрона, выведем дополнительный видеопоток с камеры дрона.





## 

## 2.3 Итоговые жесты для управления дроном

Итого система управления для дрона посредством распознавания жестов на основе машинного зрения имеет 12 команд, реализованных как для левой, так и для правой руки, для предотвращения случайных срабатываний распознавание жестов настроена таким образом, что необходимо полностью распрямлять пальцы и держать кисть под строго вертикальным или горизонтальным положением:

1. Взлет (take off). Внутренняя часть ладони направлена к камере, указательный и средний палец полностью расправлены и указывают вверх, остальные пальцы согнуты.
2. Посадка (land). Внешняя часть ладони направлена к камере, указательный и средний палец полностью расправлены и указывают вниз, остальные пальцы согнуты.
3. Прекращение движения (stop). Внутренняя часть ладони направлена к камере, все пальцы полностью расправлены и указывают вверх.
4. Движение вперед (forward). Внешняя часть ладони направлена к камере, большой палец полностью расправлен и указывает вверх, остальные пальцы согнуты.
5. Движение назад (backward). Внутренняя часть ладони направлена к камере, большой палец полностью расправлен и указывает вниз, остальные пальцы согнуты.
6. Движение вверх (up). Внутренняя часть ладони направлена к камере, указательный палец полностью расправлен и указывает вверх, остальные пальцы согнуты.
7. Движение вниз (down). Внешняя часть ладони направлена к камере, указательный палец полностью расправлен и указывает вниз, остальные пальцы согнуты.
8. Движение влево (left). Большой палец полностью расправлен и указывает влево, остальные пальцы согнуты.
9. Движение вправо (right). Большой палец полностью расправлен и указывает вправо, остальные пальцы согнуты.
10. Вращение по часовой стрелке (rotate clockwise). Внешняя часть ладони направлена к камере, указательный и средний пальцы указывают влево или указательный палец указывает вправо, остальные пальцы согнуты.
11. Вращение против часовой стрелки (rotate counter-clockwise). Внешняя часть ладони направлена к камере, указательный и средний пальцы указывают вправо или указательный палец указывает влево, остальные пальцы согнуты.
12. Определение скорости движения. Внутренняя часть ладони направлена к камере, указательный и большой палец полностью расправлен, меняется угол между ними, остальные пальцы согнуты.

# ВЫВОДЫ ПО 2 ГЛАВЕ

В процессе работы был написан программный код на языке программирования python для управления БПЛА на основе машинного зрения с использованием библиотек djitellopy, opencv, mediapipe, time, способный распознавать 12 различных жестов и управлять дроном в соответствии с ними.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данного проекта были приобретены навыки работы в следующих сферах: программирование на языке python с использованием библиотек машинного зрения opencv и mediapipe, дистанционное управление беспилотным летательным аппаратом DJI Ryze Tello. Была создана система управления для БПЛА на основе машинного зрения, способная управлять дроном, распознавая 12 различных жестов.

# ПРИЛОЖЕНИЕ 1

Список используемых библиотек python версии 3.11.3:

1. djitellopy (версия 2.5.0)
2. mediapipe (версия 0.10.8)
3. opencv\_contrib\_python (версия 4.8.1.78)
4. opencv\_python (версия 4.8.1.78)

# ПРИЛОЖЕНИЕ 2

Образец кода:

mpHands = mp.solutions.hands

hands = mpHands.Hands(min\_detection\_confidence=0.7, max\_num\_hands=1)

mpDraw = mp.solutions.drawing\_utils

direction = 'none'

speed = 0

max\_speed = 100

rotate\_angle = 0

max\_rotate\_angle = 30

speed\_flag = True

take\_off\_flag = False

while True:

success, image = cap.read()

image = cv.flip(image, 1)

imageRGB = cv.cvtColor(image, cv.COLOR\_BGR2RGB)

results = hands.process(imageRGB)

h, w, c = image.shape

if results.multi\_hand\_landmarks:

coords = []

for handlm in results.multi\_hand\_landmarks:

for id, lm in enumerate(handlm.landmark):

coords.append([int(lm.x \* w), int(lm.y \* h)])

mpDraw.draw\_landmarks(image, handlm, mpHands.HAND\_CONNECTIONS)

if (direction in ['rotate clockwise', 'rotate counter-clockwise'] and ((coords[8][0] > coords[6][0] and coords[6][0] < coords[5][0] and coords[12][0] > coords[10][0] and coords[10][0] < coords[9][0] and coords[16][0] > coords[14][0] and coords[14][0] < coords[13][0] and coords[20][0] > coords[18][0] and coords[18][0] < coords[17][0] and coords[4][1] > coords[3][1]) or (coords[8][0] < coords[6][0] and coords[6][0] > coords[5][0] and coords[12][0] < coords[10][0] and coords[10][0] > coords[9][0] and coords[16][0] < coords[14][0] and coords[14][0] > coords[13][0] and coords[20][0] < coords[18][0] and coords[18][0] > coords[17][0] and coords[4][1] > coords[3][1])) and abs(coords[5][0] - coords[17][0]) / (abs(coords[5][1] - coords[17][1]) + 1) < 0.2):

direction = 'none'

speed = 0

speed\_flag = True

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Глория, Буэно Гарсия. Обработка изображений с помощью OpenCV, 2016.
2. Jesse, Russell Беспилотный летательный аппарат, 2012.
3. Н. Я. Василин. Беспилотные летательные аппараты, 2012.
4. Дэвид Форсайт, Жан Понс. Компьютерное зрение. Современный подход, 2004.
5. Л. Шапиро, Дж. Стокман. Компьютерное зрение, 2006.
6. Отчет TAdviser. Компьютерное зрение: технологии, рынок, перспективы, 2018.
7. В. Б. Петушкова, С. О. Потапова. История создания беспилотных летательных аппаратов, 2018.