



ПрофМир

Институт непрерывного образования
и профессионального развития

ПРОГРАММА
ПОВЫШЕНИЯ КВАЛИФИКАЦИИ

«Python-разработка для БПЛА: паттерны проектирования, API-интерфейсы и библиотеки для оптимизации решений»

Преподаватель курса
Ушаков Евгений Юрьевич

Выполнил
Денисов Никита Владимирович

№ потока
БПЛА(144)_23-1.1

Москва 2023 г.

СОДЕЙСТВИЕ | Федеральный
ЗАНЯТОСТИ | проект

Проект по теме
“Поисковая операция по нахождению автомобилей
на заданном маршруте”



Шаг 1

Создайте класс для беспилотного летательного аппарата, который будет содержать основные характеристики, такие как высота, скорость, угол наклона, координаты, дистанция и т.д.

```
class Kopter:
    """Базовый класс квадрокоптера:
    x, y - начальные координаты
    h - начальная высота
    yaw - начальный угол рысканья
    v - начальная скорость"""

    def __init__(self, x: int, y: int, h=10, yaw=0, v=0):
        self.drone= System()
        async def init():
            await self.drone.connect(system_address="udp://:14540")
            # дождитесь подключения дрона
            print("Waiting for drone to connect...")
            async for state in self.drone.core.connection_state():
                if state.is_connected:
                    print("--Connected to drone!")
                    break
        asyncio.run(init())
```

War 1

```
self.x = x  
self.y = y  
self.h = h  
self.yaw = yaw  
self.v = v
```

```
def get_xy(self) -> tuple:  
    return self.x, self.y
```

```
def set_xy(self, x: int, y: int):  
    self.x = x  
    self.y = y
```

```
def get_h(self) -> float:  
    return self.h
```

```
def set_h(self, h: float):  
    self.h = h
```

```
def get_yaw(self) -> float:  
    return self.yaw
```

```
def set_yaw(self, yaw: float):  
    self.yaw = yaw
```

```
def get_v(self) -> int:  
    return self.v
```

```
def set_v(self, v: int):  
    self.v = v
```

Шаг 2

Напишите функцию для управления беспилотным летательным аппаратом. Функция должна использовать библиотеку для управления полетом, такую как Drone Kit, и добавлять функциональность для получения и отправки данных, управления двигателем и т.д.

Взлет

```
def takeoff(self):
    for health in self.drone.telemetry.health():
        if health.is_global_position_ok and health.is_home_position_ok:
            print("-- Global position estimate OK")
            break

    print("-- Arming")
    self.drone.action.arm()

    print("-- Setting initial setpoint")
    self.drone.offboard.set_position_ned(PositionNedYaw(0.0, 0.0, 0.0, 0.0))

    print("-- Starting offboard")
    try:
        self.drone.offboard.start()
    except OffboardError as error:
        print(f"Starting off board mode failed \ with error code:
              {error._result.result}")
        print("-- Disarming")

    self.drone.action.disarm()
    return
```

Шаг 2 (Функция посадки)

```
# посадка
def land(self):
    print "--Landing"
    self.drone.action.land()

    for state in self.drone.telemetry.landed_state():
        if state == LandedState.ON_GROUND:
            break

    print "--Stoppingoffboard")

    try:
        self.drone.offboard.stop()
    except Exception as error:
        print(f"Stoppingoffboardmodefailedwitherror: {error}")

    print "--Disarming")
    self.drone.action.disarm()
```

Шаг 2 (Функция полета к точке)

```
# полет к точке
```

```
def go2xy(self, x, y, h, yaw, vx, vy, vz):  
    self.drone.offboard.set_position_velocity_ned(  
        PositionNedYaw(*[x, y, h], yaw_deg=yaw),  
        VelocityNedYaw(*[vx, vy, vz], yaw_deg=yaw)  
    )
```

Шаг 3

Разработайте сценарий для тестирования беспилотного летательного аппарата. Например, можно создать сценарий для облета заданных объектов, съемки видео с высоты и т.д. Включите в сценарий проверку на ошибки и обработку исключительных ситуаций.

Постановка задачи:

- *Облет территории по точкам с заданными координатами, поиск объектов (автотранспорта) по траектории полета и их распознавание.*
- *Для достижения более хорошего результата будут использованы алгоритмы машинного обучения.*
- *Детектирование и распознавание объектов выполняется с помощью нейронной сети на базе классификаторов каскада Хаара.*

Шаг 4

Используйте библиотеку OpenCV для обработки изображений. Например, реализуйте распознавание объектов на изображениях, используя нейронные сети.

```
if waypoints[point][1] == key_point[0] and waypoints[point][2] == key_point[1]:
    # съемка ключевой точки
    img = cv2.imread("cars4.jpg")

    # обработка изображения
    img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    xml_data = cv2.CascadeClassifier("cars.xml")
    detecting= xml_data.detectMultiScale(img_gray, minSize=(10,10), scaleFactor=1.05)
    detect= len(detecting)
    if detect!= 0:
        for(a, b, widht, height) in detecting:
            cv2.rectangle(img_gray, (a, b), (a + height, b + widht), (0, 255, 0), 3)

    cv2.imshow('avto', img_gray)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

Шаг 4 (итог работы)



Рис. 1 Исходная фотография

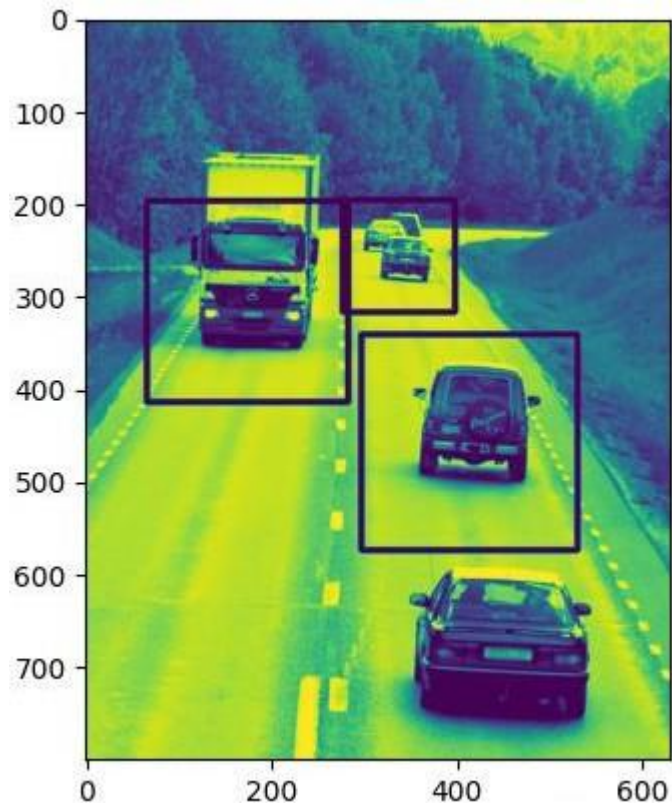


Рис.2. Обнаруженные объекты

Шаг 5

Разработайте алгоритм управления беспилотным летательным аппаратом с помощью машинного обучения. Например, можно использовать алгоритмы глубокого обучения для обучения аппарата выполнять определенные действия в зависимости от условий и среды.

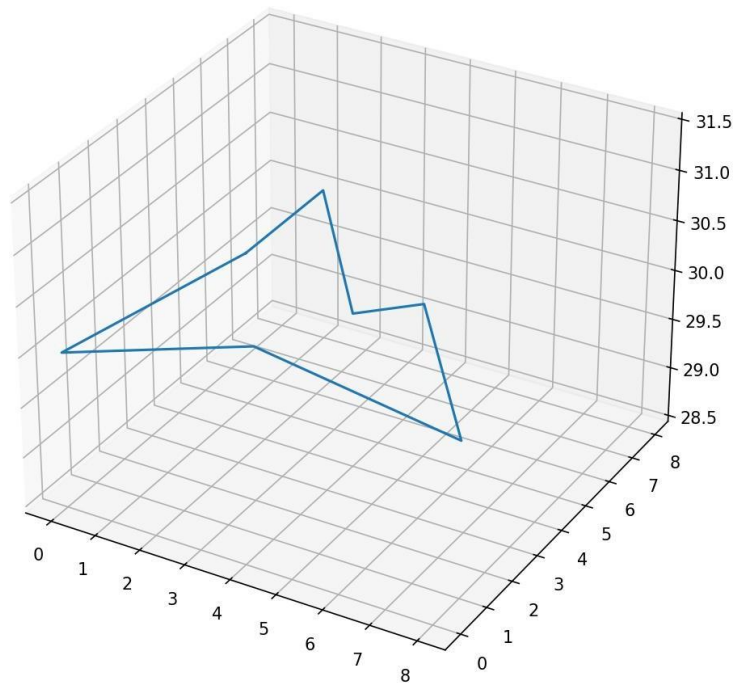
```
# оптимизируем маршрут
# фитнес-функция оптимизации
fitness_func = mlrose.TravellingSales(coords=points)

# Проблема: минимизация/максимизация
problem = mlrose.TSPOpt(length=len(points), fitness_fn=fitness_func, maximize=False)

# запуск генетического алгоритма для решения задачи
best_path, best_lenght = mlrose.genetic_alg(problem=problem, random_state=3)

print(best_path, best_lenght)
best_path = list(best_path) * 2
best_path = best_path[best_path.index(0)+1:best_path.index(0)+len(points)+1]
```

Шаг 5 (Оптимизированный маршрут облета)





ПрофМир

Институт непрерывного образования
и профессионального развития

Благодарю за внимание!

**СОДЕЙСТВИЕ
ЗАНЯТОСТИ** | Федеральный
проект



info@eduom.ru