

АВТОМАТИЗОВАНЕ ДЕТЕКТУВАННЯ ТА МОНІТОРИНГ ПЕРЕМІЩЕННЯ ОБ'ЄКТІВ НА БАЗІ БПЛА

Сьогодні розвиток технологій у галузі БПЛА дозволяє створювати різноманітні системи спостереження, що забезпечують високу точність виявлення та можливість аналізу даних у реальному часі. Наномережі, з свого боку, забезпечують працездатність у складних умовах та мають значний потенціал для збору та обробки інформації. Об'єднання цих двох технологій в структуровану систему дозволяє оптимізувати використання ресурсів та забезпечити комплексний підхід до пошукових операцій. У відповідності до поставлених завдань обираються методи та алгоритми комп'ютерного зору. Багато завдань вимагають створення власних наборів даних для якісного аналізу зображень. Наномережі є перспективним методом завдяки здатності працювати з обмеженими ресурсами, масштабованості та високій швидкості обробки даних.

Порівнявши популярний метод детектування TensorFlow SSD MobileNet v1 [1] з API Nanonets [2], можна побачити деякі важливі відмінності. Модель TensorFlow SSD MobileNet v1 – це складний навчальний алгоритм на основі MobileNet для виявлення об'єктів на відео та зображеннях. Вона вимагає значних обчислювальних ресурсів, але дозволяє гнучке налаштування. Натомість, API машинного навчання Nanonets пропонує готову до використання послугу виявлення об'єктів, яка є простішою та швидшою у впровадженні, хоча і менш гнучкою. Натомість, використовуючи API NanoNets, є можливим зробити прогноз лише за 2 кроки: отримання ключа API та прогнозування.

Обробку даних для виявлення об'єктів за допомогою попередньо навченої моделі машинного навчання виконує сервіс Nanonets. Після надсилання кадру/зображення на сервер Nanonets через HTTP POST-запит, сервер обробляє зображення моделлю та повертає координати обмежувальних прямокутників для виявлених об'єктів у форматі JSON. Код візуалізує ці прямокутники на оригінальному кадрі/зображенні. В той же час, TensorFlow SSD MobileNet v1 використовує 4 основні кроки прогнозування.

Коли дрон виявить людину, він має 4 цифри, що описують поле: верхній лівий кут (x_1, y_1) і нижній правий кут (x_2, y_2). Враховуючи це, ми також можемо легко обчислити центр прямокутника, а також його площу. Щоб обчислити площу, нам потрібно лише обчислити ширину як $(x_2 - x_1)$ і висоту як $(y_2 - y_1)$ і помножити їх. Що стосується центру, то він обчислюється просто як $(x_2 + x_1) / 2$ і $(y_2 + y_1) / 2$. Центр прямокутника може вказати нам, чи знаходиться людина в центрі зображення, чи справа чи зліва. Маючи цю інформацію, ми можемо надіслати інструкцію дрону повернутися ліворуч або праворуч відносно його вертикальної осі, щоб перевести людину в центр кадру. Площа прямокутника може дати нам приблизну інформацію про те, наскільки близько знаходиться людина. Більший прямокутник означає, що людина дуже близько, а крихітний прямокутник вказує на те, що людина далеко.



Рис. 1. Результат роботи програми

Розроблений код з використанням API Nanonets успішно виконує завдання виявлення людей для подальшого моніторингу безпілотниками. Ключовими перевагами даного рішення є: використання попередньо навченої моделі, що спрощує розгортання та зменшує витрати; масштабованість завдяки хмарному API; висока точність виявлення об'єктів за рахунок глибоких нейронних мереж. Це демонструє ефективне застосування сучасних методів машинного навчання та комп'ютерного зору для практичних завдань безпеки та моніторингу.

Список використаних джерел

1. TensorFlow SSD MobileNet v1 [Електронний ресурс] – Режим доступу: <https://kamleshs.medium.com/object-detection-using-ssd-mobilenet-with-tensorflow-1ee6e45a378b>
2. Nanonets [Електронний ресурс] – Режим доступу: <https://deepgram.com/ai-apps/nanonets>