

Autonomes Personenfolgesystem mit einer Drohne (DrohneLab)

Quang Tuan Anh Nguyen, Yohanny Carolina Camacaro Querales

KI-Projekt • Studiengang Informatik • Fachbereich Informatik und Medien • 15.01.2025

Aufgabenstellung

Ziel dieser Arbeit ist die Entwicklung eines autonomen Personenfolgesystems für eine Drohne des Typs Ryze Tello.

Die Echtzeit-Bildverarbeitung erfolgt auf einem externen Rechner, während das Tello SDK zur Kommunikation und Steuerung der Drohne eingesetzt wird [1]. Die Drohne soll eine erkannte Person selbstständig verfolgen und dabei Abstand, Ausrichtung und Stabilität im Flug beibehalten.

Mithilfe der On-Board-Kamera werden kontinuierlich Videodaten aufgenommen und an einen stationären Rechner übertragen. Dort erfolgt die Bildverarbeitung sowie die Berechnung der Steuerbefehle, welche anschließend in Echtzeit an die Drohne zurückgesendet werden. Zusätzlich soll eine sichere Fehlerbehandlung implementiert werden, falls die Zielperson verloren geht oder keine eindeutige Erkennung möglich ist.

Vorbereitung

Für die Personenerkennung wurde ein vortrainiertes Deep-Learning-Modell zur Objekterkennung eingesetzt (z. B. MobileNet-SSD) [3]. Das Modell ist in der Lage, Personen in Kamerabildern zu detektieren und deren Position sowie Größe im Bild zu bestimmen.

Die Bildverarbeitung und Steuerlogik wurden in Python implementiert. Für die Kommunikation mit der Drohne wurde die Bibliothek djitellopy verwendet. Zur Visualisierung und Debugging-Zwecken wurde eine grafische Oberfläche mit OpenCV [2] und Pygame realisiert.

Konzept

Das System basiert auf einem geschlossenen Regelkreis:

1. Aufnahme von Kamerabildern durch die Drohne
2. Übertragung der Bilddaten an den Rechner über WLAN
3. Detektion der Zielperson mittels neuronalen Netzes [3,4]
4. Berechnung der Positionsabweichung der Person relativ zur Bildmitte
5. Ableitung von Steuerbefehlen (Yaw, Vorwärts/Rückwärts, Auf/Ab)
6. Rückübertragung der Befehle an die Drohne

Drohne hält während der Verarbeitung eine stabile Schwebeposition. Zur Reduktion von unruhigen Bewegungen werden Totzonen (Dead Zones) sowie Geschwindigkeitsbegrenzungen verwendet.

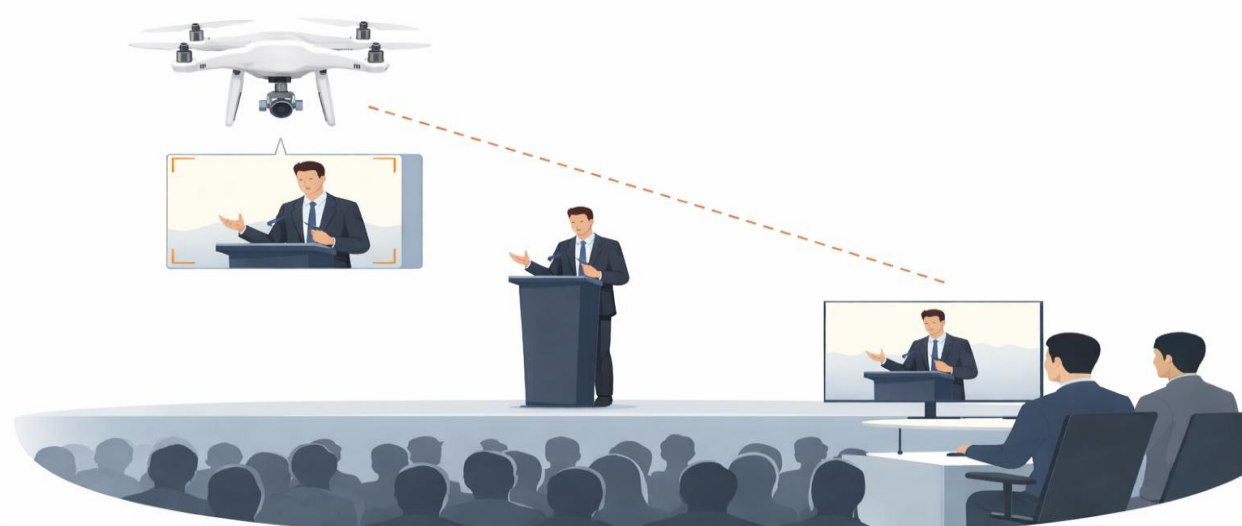


Abb. 1: Das Konzept des Systems in den Medien: Eine Drohne erkennt den Redner automatisch und fliegt um ihn herum, um Videoaufnahmen zu machen

Autonomer Verfolgungsmodus

Im autonomen Modus folgt die Drohne einer einzelnen erkannten Person.

Dabei werden folgende Kriterien berücksichtigt:

- horizontale Abweichung → Drehung (Yaw)
- vertikale Abweichung → Höhenanpassung
- Erkannte Personengröße → Distanzregelung

Falls die Zielperson nicht mehr erkannt wird, wechselt das System automatisch in einen sicheren Such- oder Hover-Modus. Wird nach einer definierten Zeit kein Ziel gefunden, landet die Drohne selbstständig.

Ergebnisse

Zur Evaluation des Systems wurden mehrere Testflüge in Innenräumen durchgeführt.

Die Drohne konnte die Zielperson in den meisten Fällen stabil verfolgen und ihre Position zuverlässig anpassen.

Probleme traten hauptsächlich bei:

- schnellen Richtungswechseln der Person
- unzureichender Beleuchtung
- teilweise verdeckter Personenerkennung

auf.

Fazit

Die Arbeit zeigt, dass eine kostengünstige Consumer-Drohne mithilfe von Deep-Learning-basierter Bildverarbeitung autonom einer Person folgen kann. Die Ergebnisse belegen die Praxistauglichkeit des Ansatzes für einfache Indoor-Szenarien.

Verbesserungspotenzial besteht insbesondere in:

- robusterer Zielverfolgung bei Okklusion
- effizienterer Fehlerbehandlung
- Erweiterung auf mehrere Zielpersonen

Quellen

[1] Ryze Technology. (2018). *Tello SDK Documentation*.

Abgerufen am **01.01.2026**, von

<https://www.ryzerobotics.com/tello>

[2] Bradski, G. (2000). *The OpenCV Library*. Dr. Dobb's Journal of Software Tools.

OpenCV Documentation, abgerufen am **01.01.2026**, von

<https://opencv.org>

[3] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016).

SSD: Single Shot MultiBox Detector.

In: Proceedings of the European Conference on Computer Vision (ECCV). Springer.

[4] TensorFlow Team. (2023). *TensorFlow Object Detection API*.

Abgerufen am **01.01.2026**, von

<https://github.com/tensorflow/models>