Onderzoeksrapport Softwareselectie

Project: Lokaliseren van drones met camera's en microfoons

Opdrachtgever: Tidalis

Team: Ali, Dennis, Fabio, Tom

Datum: 30 September 2025

Versie: 0.1

Hogeschool Rotterdam
Technische Informatica

Inhoudsopgave

1 Inleiding en Context

1.1 Project Context

Dit project heeft als doel een PoC (proof of concept) systeem te ontwikkelen voor drone detectie en lokalisatie met camera's. Opdrachtgever Tidalis, specialist in maritieme situational awareness, wil hiermee zijn surveillancemogelijkheden uitbreiden.

Projectscope

- Real-time detectie van drones/objecten in 2D ruimte
- Lokalisatie met stereoscopische of IP camera setup
- Eenvoudige GUI voor visualisatie
- Proof-of-concept ontwikkeling binnen beperkte tijd

1.2 Onderzoek Scope

Dit onderzoeksrapport behandelt het besluitvormingsproces voor de selectie van technologieën en tools die gebruikt worden in het drone detectie systeem.

Doel van het Onderzoek

- Het identificeren van technologieën die aansluiten bij de projectvereisten en de huidige vaardigheden van het team
- Het vergelijken van deze technologieën op basis van objectieve criteria
- Onderbouwen van de gemaakte technische keuzes
- Het aanbevelen van een technologie-stack die de projectdoelen ondersteunt.

1.3 Belangrijke Randvoorwaarden

- **Tijdsduur:** 5 maanden (september 2025 - januari 2026)
- **Team expertise:** Java, Python, C++, C, GitHub/GitLab
- **Hardware:** Raspberry Pi 5, 2K webcams camera, beperkt budget
- **Focus:** Eenvoud en haalbaarheid

2 Requirements Analyse

2.1 Functionele Requirements

1. **Object Detectie** – Detecteren van drones/objecten in camera beeld
2. **2D Lokalisatie** – Bepalen van x,y positie in real-time
3. **GUI Visualisatie** – Tonen van drone positie in grafische interface

2.2 Niet-functionele Requirements

1. **Eenvoud** – Eenvoudige installatie, configuratie en technologieën die aansluiten bij team expertise
2. **Performance** – Werken op Raspberry Pi 5 met 4GB RAM

3. **Hardware Compatibiliteit** – Goede samenwerking tussen componenten
4. **Onderhoudbaarheid** – Duidelijke code en documentatie
5. **Budget**: Gebruik van open-source en gratis beschikbare technologieën waar mogelijk

3 Technologie Kandidaten

3.1 Computer Vision

- **OpenCV** – Rijp computer vision framework
- **MobileNet SSD** – een efficiënt en lichtgewicht ”pre-trained”model voor objectdetectie
- **YOLO(v5 of v8)** – Real-time object detectie ”pre-trained”model

3.2 GUI Frameworks

- **Tkinter** – Standaard Python GUI
- **OpenCV HighGUI** – Eenvoudige display functionaliteit
- **PyQt(v5)** – Voor rijke desktop applicaties/meer GUI functionaliteit
- **pyside6** – pyqt6 en pyside6 zijn hetzelfde in termen van functionaliteit maar pysidey heeft geen Commercial License

3.3 Andere Bibliotheeken

- **NumPy** – Coördinaten en afstand berekenen (built-in met OpenCV)
- **Matplotlib** – Visualisatie van dronepositie
- **unittest** – voor unit testen (built-in Python)
- **pytestn** – unit testen, krachtiger, gemakkelijkere syntaxis.
- **logging** – detectie events opslaan (built-in Python)
- **CSV/JSON** – droneposities bewaren (built-in Python)

4 Evaluatie Methodologie

4.1 Beoordelingscriteria

Technologieën worden beoordeeld op:

1. **Gemak van Leren/Setup** – Tijd nodig om technologie onder de knie te krijgen
2. **Functionaliteit** – Geschiktheid voor specifieke taken
3. **Performance** – Snelheid en geheugengebruik op Raspberry Pi
4. **Integratie** – Compatibiliteit met andere componenten
5. **Tijdslijn Fit** – Haalbaarheid binnen projectduur

4.2 Weegfactoren

Criteria Weging

- Gemak van Leren/configratie: **25%**
- Functionaliteit: **30%**
- Performance: **20%**
- Integratie: **15%**
- Tijdslijn Fit: **10%**

5 Vergelijkingstabel Technologieën

Tabel 1: Vergelijking Technologie Keuzes (Schaal: 1-5, waarbij 5 = beste)

Technologie	Gemak	Functionaliteit	Performance	Integratie	Totaal
Programmeertaal					
Python	5	5	4	5	**4.5**
C++	3	4	5	4	4.0
Computer Vision					
OpenCV	5	5	4	5	**4.8**
YOLO	3	5	3	4	3.8
MobileNet SSD	2	4	3	3	3.1
GUI Frameworks					
PyQt/PySide	3	5	4	5	**4.2**
Tkinter	5	3	4	4	4.0
OpenCV HighGUI	4	2	5	5	4.0

6 Discussie: Afwegingen en Risico's

6.1 Belangrijkste Afwegingen

- **Eenvoud vs Functionaliteit**: OpenCV kan drones detecteren, maar is minder betrouwbaar bij bewegende achtergrondobjecten,en heeft lagere nauwkeurigheid dan deep learning-methoden zoals YOLO of MobileNet SSD.
  OpenCV HighGUI is eenvoudig maar beperkt, PyQt is krachtig maar complex
- **Performance vs Leercurve**: YOLO biedt state-of-the-art detectie maar heeft hogere leercurve en heeft
- **Integratie**: Testen op de gekozen hardware is vereist om evaluatie mogelijk te maken

6.2 Risico's en Beperkingen

Belangrijke Risico's

- **Tijdsdruk:** 5 maanden is beperkt voor complexe computer vision
- **Hardware Limitaties:** Raspberry Pi 5 heeft beperkte rekenkracht
- **Team Expertise:** Nieuwe technologieën vereisen leerinvestering
- **Realtime Eisen:** Latentie moet acceptabel blijven

6.3 Mitigatie Strategieën

- Starten met een eenvoudige implementatie en iteratief verbeteren
- Eerst richten op de kernfunctionaliteit
- Regelmatige testen op target hardware
- Gebruikmaken van bestaande voorbeelden en documentatie

7 Uiteindelijke Keuzes

Tabel 2: Definitieve Technologie Selectie

Component	Keuze	Rechtvaardiging
Programmeertaal	Python 3.x	• Snelle prototyping mogelijk binnen 5 maanden tijdslijn • Uitstekende OpenCV bindingen beschikbaar • Uitgebreide wetenschappelijke libraries (NumPy, etc.)
Computer Vision	OpenCV 4.x	• Getest met IP-camera: werkt naar behoren • Rijp en stabiel framework met uitgebreide documentatie • Goede performance op Raspberry Pi 5 (score 4) • Ingebouwde stereo vision modules voor 2D lokalisatie • Breed scala aan features voor toekomstige uitbreiding
Object Detectie	OpenCV 4.x	• Geen training van een neuraal netwerk nodig • Draait eenvoudig op een Raspberry Pi 5 (4GB RAM) • **Tweede optie**: • MobileNet SSD: efficiënt en lichtgewicht pre-trained model • Kan worden gebruikt via OpenCV (cv3.dnn) • Geschikt voor resource-beperkte hardware (Raspberry Pi 5) • YOLO (v5/v8): optioneel voor betere detectie na basisimplementatie • Pre-trained modellen beschikbaar voor snelle integratie

Tabel 2: Definitieve Technologie Selectie (vervolg)

Component	Keuze	Rechtvaardiging
GUI Framework	Tkinter + OpenCV HighGUI	• Tkinter: standaard Python, eenvoudig te leren • OpenCV HighGUI: directe integratie met computer vision • Goede performance voor real-time display • Minimale setup vereist • **Tweede optie**: • PyQt5 • pyside6
Data Opslag	CSV/JSON	• Built-in Python ondersteuning (geen extra dependencies) • Eenvoudig voor opslaan van droneposities en detectie events • Compatibel met logging requirements • Minimale setup vereist
Testing Framework	pytest	• Krachtiger dan unittest met gemakkelijkere syntaxis • Goede ondersteuning voor unit testing • Helpt onderhoudbaarheid van code te waarborgen • Breed gebruikt in Python community
Hulpbibliotheeken	NumPy + logging	• NumPy: coördinaten en afstandsberekeningen (built-in met OpenCV) • logging: detectie events registreren (built-in Python) • Beide minimaliseren externe dependencies • Essentieel voor 2D lokalisatie functionaliteit

8 Referenties

Referenties

[1] Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media.

[2] Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. arXiv:1804.02767.

[3] Scharstein, D., & Szeliski, R. (2002). *A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms*. International Journal of Computer Vision.

[4] Raspberry Pi Foundation. (2023). *Raspberry Pi 5 Documentation*. https://www.raspberrypi.com/documentation/

[5] Tidalis. (2024). *Maritime Situational Awareness Solutions*. https://www.tidalis.com/

[6] Summerfield, M. (2007). *Rapid GUI Programming with Python and Qt*. Prentice Hall.

[7] OpenAI. (2024). *ChatGPT (GPT-4)*. Retrieved September 30, 2025, from https://chat.openai.com/

[8] Anthropic. (2025). *Claude (Claude Sonnet 4.5)*. Retrieved September 30, 2025, from https://claude.ai/

9 Wijzigingslog

Tabel 3: Document Wijzigingsgeschiedenis

Datum	Versie	Wie	Wijzigingen
26-09-2025	1.0	Ali	Eerste versie - complete technische keuze analyse
30-09-2025	0.1	Ali	Bijwerken van de vergelijkingstabel