

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINARSKI RAD

SUSTAV LOKALIZACIJE LETJELICE ZA INSPEKCIJU INFRASTRUKTURE

Luka Vukelić

Voditelj: izv. prof. dr. sc. Matko Orsag

Zagreb, svibanj, 2024.

Sustav lokalizacije letjelice za inspekciju infrastrukture

Luka Vukelić

Sažetak

Rad istražuje primjenu sustava lokalizacije letjelice u kontekstu inspekcije infrastrukture, s fokusom na primjenu u vertikalnom vrtlarstvu. Kroz detaljan pregled problema i radnog okruženja, analiziraju se specifičnosti *International Conference on Unmanned Aircraft Systems* (ICUAS) natjecanja za koje je razvijen simulacijski svijet u Gaza bo okruženju. Korištenjem ROS platforme i alata poput Pythona, Pynputa, OpenCV-a te PyTorch-a, razvijeno je programsko rješenje koje omogućuje lokalizaciju objekata, upravljanje letjelicom te detekciju biljaka u vrtu. Detaljno su opisani sustavi lokalizacije, proces upravljanja letjelicom te algoritmi detekcije biljaka, uz spomenuti YOLO algoritam. Nadalje, rad istražuje mogućnosti daljnjeg rada, uključujući voxelizaciju detektiranog svijeta i implementaciju planiranja trajektorije. Ovaj rad pruža temelje za razumijevanje i primjenu sustava lokalizacije letjelica u specifičnim scenarijima inspekcije infrastrukture poput vertikalnog vrtlarstva.

Ključne riječi: UAV; sustavi lokalizacije; računalni vid

UAV localization system for infrastructure inspection

Luka Vukelić

Abstract

The paper explores the application of UAV localization systems in the context of infrastructure inspection, with a focus on vertical gardening. Through a detailed overview of the problem and the working environment, the specificities of the *International Conference on Unmanned Aircraft Systems* (ICUAS) competition are analyzed, for which a simulation world is developed in the Gazebo environment. Using ROS platform and tools such as Python, Pynput, OpenCV, and PyTorch, a software solution is developed enabling object localization, drone control, and plant detection in the garden. Localization systems, drone control processes, and plant detection algorithms are described in detail, with a mention of the YOLO algorithm. Furthermore, the paper explores possibilities for further work, including voxelization of the detected world and trajectory planning implementation. This paper provides a foundation for understanding and applying UAV localization systems in specific infrastructure inspection scenarios such as vertical gardening.

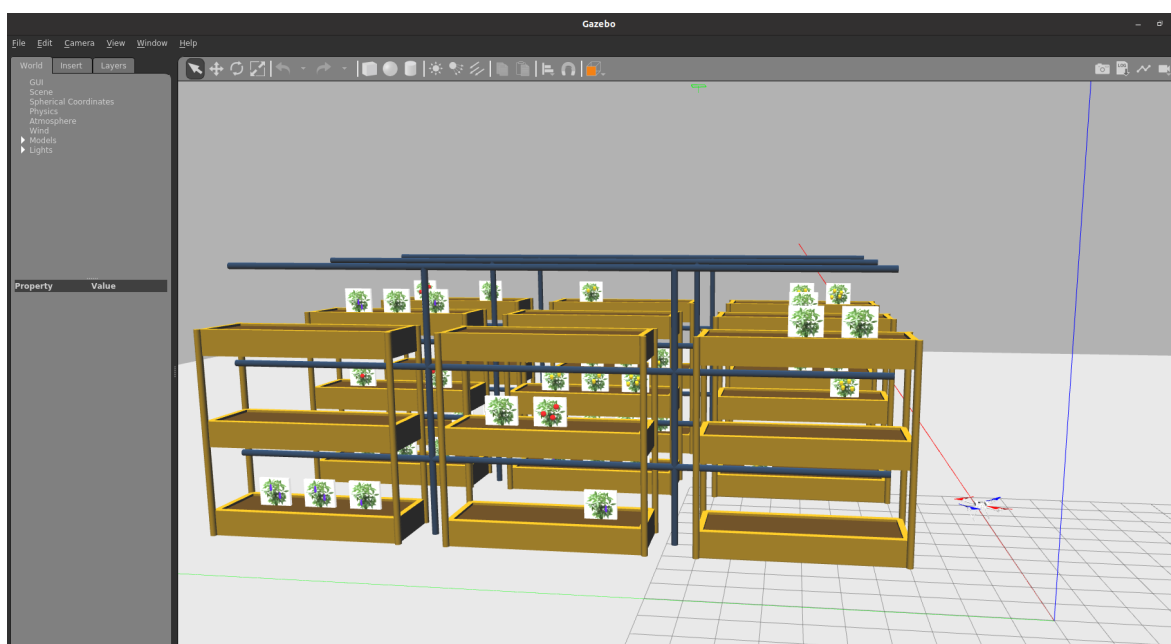
Keywords: UAV; positioning systems; computer vision

Sadržaj

Sažetak	1
Abstract	2
1. Definiranje problema i radnog okruženja	4
1.1. ICUAS 2024	4
1.2. Skup alata	5
2. Programsko rješenje	7
2.1. Vrste sustava lokalizacije	7
2.2. Upravljanje letjelicom	8
2.3. Lokalizacija koordinatnih sustava u simulaciji	9
2.4. Detekcija biljaka	10
3. Mogućnost daljnjeg rada	13
4. Literatura	14

1. Definiranje problema i radnog okruženja

1.1. ICUAS 2024



Slika 1.1. Simulacijski svijet

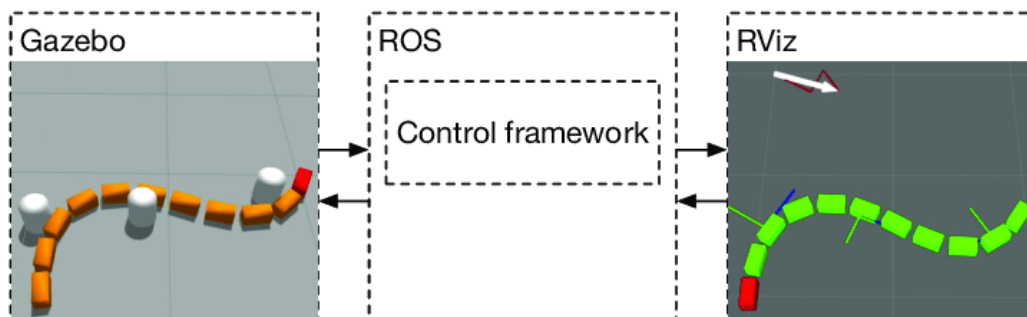
International Conference on Unmanned Aircraft Systems natjecanje je organizirano od strane nekoliko tehničkih fakulteta te podržano od strane ogranaka, koji se bave automatikom i robotikom, svjetski poznate organizacije IEEE. Na internetskim stranicama konferencije je objavljen kod na platformi GitHub koji je korišten kao predložak za problem koji se obrađuje na seminaru. Također je dostupna skripta za pokretanje sustava putem Docker okruženja, čime se pojednostavljuje proces preuzimanja već gotovih predložaka na lokalno računalo. [1]

U kodu je implementirana i izvedena simulacijska arena koja prikazuje vrtne krevete te razne vrste biljaka u njima. Također je napravljen model letjelice te mogućnosti kojima

se može upravljati njime te proširiti iste. Izazov natjecanja dijeli se na više segmenata. Sastoji se od implementiranja koda za planiranje trajektorije, lokalizaciju objekata te njihovu detekciju. Biljke se razlikuju po nuli, jednom ili više plodova koje sadrže. Fokus seminara je lokalizacija objekata. Plan je locirati te detektirati biljke na vrtu te primijeniti voxelizaciju koja bi kasnije poslužila kao pomoć za prikazivanje pozicija biljaka na 3D grafu. Ostali aspekti natjecanja bit će spomenuti te neki i površno obrađeni.

1.2. Skup alata

Simulacija zadatka napravljena je pomoću 2D/3D robotskog simulatora otvorenog koda, programa Gazebo. Simulator omogućuje realističan prikaz okruženja uključujući visokokvalitetno osvjetljenje, sjene i teksture. Može modelirati senzore koji *vide* simulirano okruženje, poput laserskih daljinomjera i kamera. Za programiranje letjelice koristi se ROS softverska platforma uz programski jezik Python koji ima pristup ROS Noetic 1 bibliotekama. Gazebo pruža ROS integraciju putem paketa poput "gazebo_ros_pkgs". Ovi paketi omogućavaju ROS čvorovima da komuniciraju s Gazebo simulatorom putem ROS tematskih poruka. ROS čvorovi i tematske poruke (eng. nodes and topics) su glavni posrednici komunikacije u ROS-u. Čvorovi se mogu pretplatiti ili slati poruke različitih tipova na određene tematske poruke. ROS čvorovi se međusobno registriraju putem ROS mastera, tematskog posrednika koji omogućava razmjenu. [2]



Slika 1.2. Odnos softverskih alata; [researchgate.net](https://www.researchgate.net)

Za pristup tipkovnici i mišu korišteni su Pynput biblioteka te PIL (*Python Imaging Library*) za obradu slika snimljenih iz kamere letjelice. Aspekt računalnog vida obrađen je s bibliotekama PyTorch i OpenCV.

Što se tiče prikaza lokacija detekcije biljaka koristi se matplotlib biblioteka, a pri tome pomaže proces transformiranja 2D mape u 3D rešetku koji može biti obrađen raznim

alatima poput *Blendera*. Pretvorba standardnog 3D prostora u diskretni set kockastih elemenata (eng. voxels) proces je *voxelizacije*.

Još jedan koristan 3D vizualizacijski alat koji će se koristiti za prikazivanje koordinatnih sustava simulacijskog svijeta je RViz.

2. Programsko rješenje

2.1. Vrste sustava lokalizacije

U kontekstu kompleksnih zadataka lokalizacije, kao što su oni u robotici ili naprednoj navigaciji, sustavi lokalizacije mogu se podijeliti na nekoliko ključnih kategorija. Prvo, imamo globalne lokalizacijske sustave koji koriste GPS (Global Positioning System) ili druge satelitske tehnologije za određivanje položaja objekata na globalnoj razini. Ovi sustavi su izuzetno korisni u vanjskim okruženjima gdje je vidljivost neba neometana. Drugo, postoje lokalni lokalizacijski sustavi koji se oslanjaju na različite senzore kao što su lidari, sonari, infracrveni senzori i kamere za mapiranje i navigaciju unutar ograničenih ili unutarnjih prostora gdje GPS nije dostupan ili je nepouzdan. Treće, relativni lokalizacijski sustavi koriste se za praćenje položaja objekta u odnosu na poznatu početnu točku ili u odnosu na druge objekte, što je često u domeni robotskih sustava koji rade u grupama. [3]

U simulaciji s kojom se seminar bavi postoji nekoliko tematskih kanala koji su relevantni za sustave lokalizacije, posebno u kontekstu navigacije, praćenja položaja i integracije senzora.

- `/odom` - *topic* obično sadrži informacije o odometriji, koje su ključne za lokalizaciju robota u odnosu na njegov početni položaj
- `/tf` i `/tf_static` - *topici* koji nose informacije o transformacijama između koordinatnih sustava
- `/red/mavros/gps...`, `/red/mavros/imu/data`, `/red/imu`, `/mavros/global_position/global`, `/mavros/global_position/local` - mogu prenositi podatke s GPS-a, inercijskih mjernih jedinica (IMU), ili drugih globalnih i lokalnih pozicijskih sustava koji pomažu

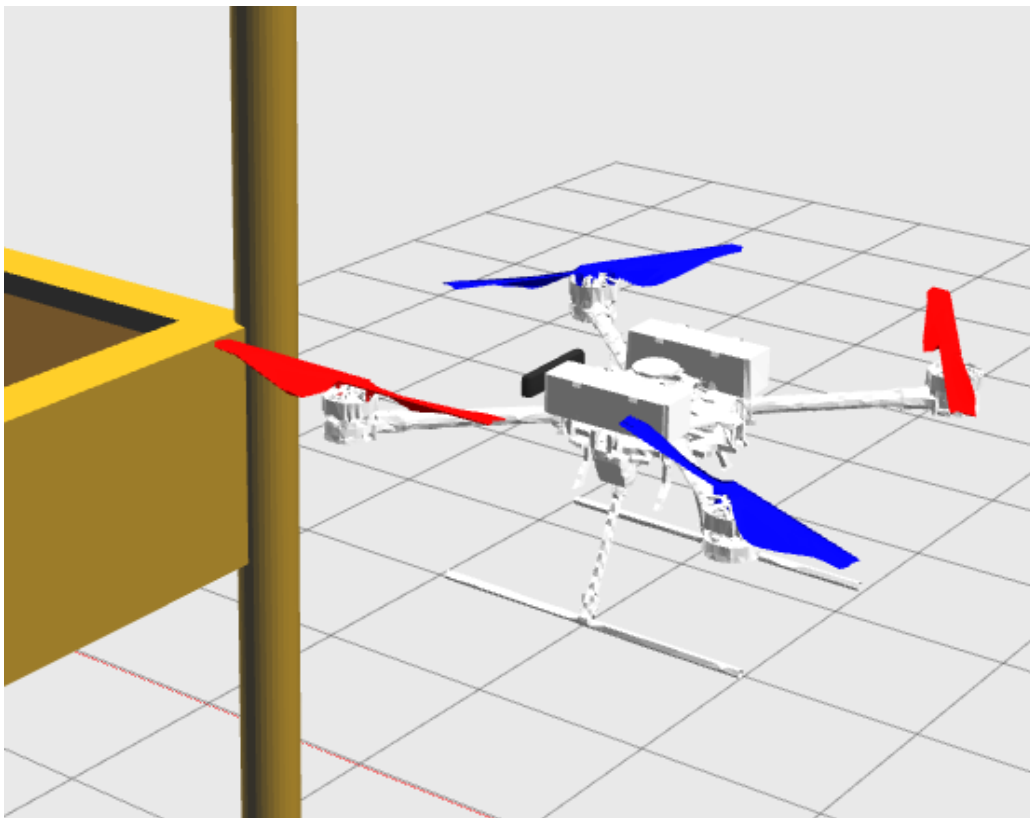
u određivanju položaja robota u globalnom prostoru ili relativno u odnosu na početnu točku

- i slični koji će biti spomenuti u nastavku

Sustavi lokalizacije su konfigurirani u simulaciji.

2.2. Upravljanje letjelicom

Korak upravljanja i planiranja trajektorije letjelice pojednostavljen je te sveden na primitivno korištenje tipkovnice.



Slika 2.1. Kretanja letjelicom

Topic /red/tracker/input_pose korišten je za slanje poruka tipa *PoseStamped* te ovisno o pritisku tipke letjelica se pomiče za fiksno određen skalar. Pozicija koju šaljemo na navedeni kanal izračunata je pomakom ovisno o trenutnoj poziciji letjelice koja je isčitana iz kanala */red/pose* metodom *rospy.wait_for_message*.

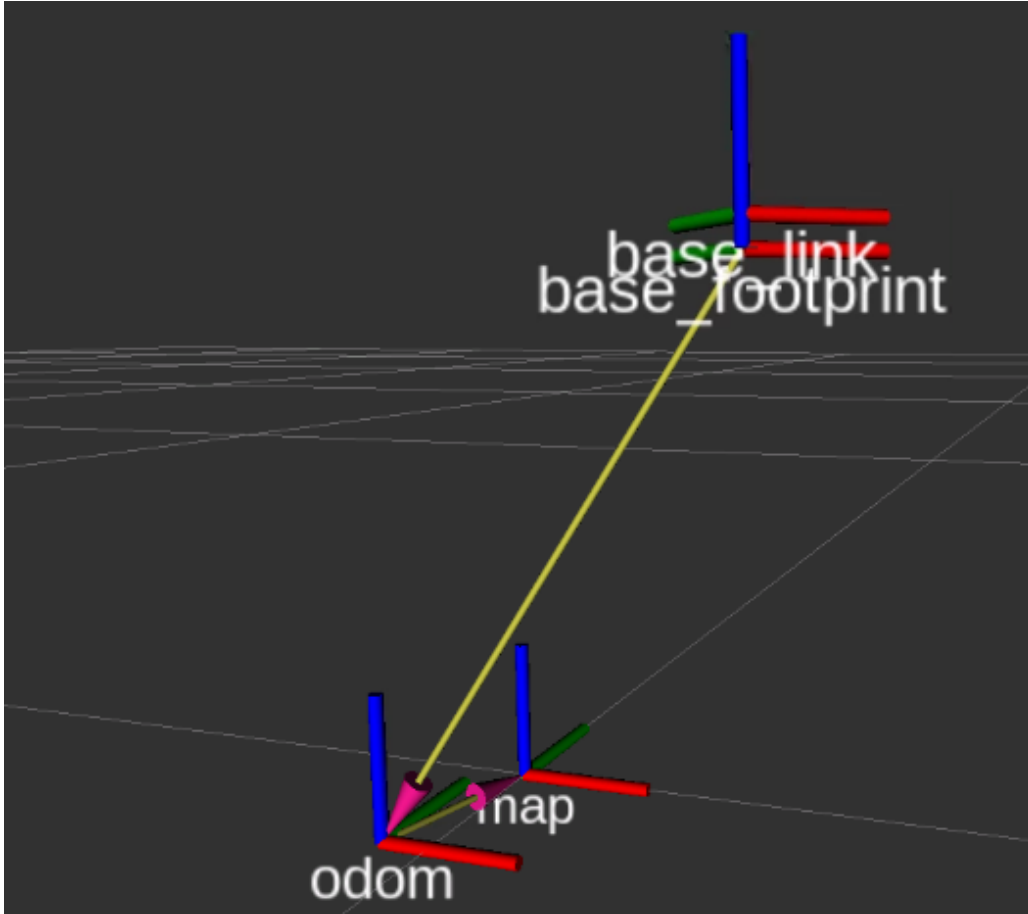
2.3. Lokalizacija koordinatnih sustava u simulaciji

Prikazivanje koordinatnih sustava iz Gazebo simulacije u drugim programima poput RViz-a, koji je dio Robot Operating System (ROS) okruženja, uključuje slanje podataka o transformacijama između različitih koordinatnih sustava u realnom vremenu. Ovo se postiže korištenjem ROS-ovog *tf* (transform) paketa, koji omogućava da se podaci o položaju i orijentaciji iz Gazebo programa efikasno integriraju u RViz.

Glavni koordinatni sustav koji se koristi kao referenca u simulaciji je k.s. *world*. Svaki detektirani objekt želimo na kraju svesti da postane lociran u navedenom sustavu kako bi nam programski zadatci bili jednostavniji. Ostali sustavi su *map*, *odom* i *base_link*. *Map* predstavlja statički k.s. u kojem se robot kreće te mu služi kao dugoročna referentna točka. *Odom* se koristi za praćenje i ažuriranje trenutne pozicije robota temeljeno na njegovim internim senzorima te je to razlog ne pouzdane lokacije. *Base_link* je referentna točka na samom robotu. Uz referentnu točku u centru robota postoje i referentne točke na svakom krilu letjelice (*/red/rotor_0*, */red/rotor_1*, */red/rotor_2*, */red/rotor_3*).

Uspoređujući izlaze tematskih kanala (*topics*) te čvorove (*nodes*) koji su na određene pretplaćeni, može se procijeniti kako je konfiguriran sustav. Na primjer, */red/odometry* je tematski kanal na koji se šalju poruke o poziciji izravno povezane na *base_link* s referencom na *world* k.s, dok */red/mavros/global_position/local* ima referencu na *map* k.s. Koristeći RViz možemo vizualizirati koordinatne sustave te njihove međusobne odnose.

Za detekciju lokacije biljke koja se nalazi u rasponu kamere letjelice postoji više mogućnosti nalaženja smjera i pozicije iz reference kamere letjelice. Kamera letjelice očitava dubinsku, s jednim kanalom, i obojano, RGB s tri kanala, sliku. S tim podacima kao ulazima može se pristupiti jednostavniji ili kompliciranijim algoritmima. U seminaru je obrađen jednostavniji algoritam iako je u 3. poglavlju opisan princip korištenja YOLO algoritma. Jednostavniji način je pozicionirati se okomito ispred biljke koju letjelica želi detektirati te pomoću biblioteke CvBridge transformirati dubinsku sliku u *numpy* polje. Iz toga se lako algebarski može doći do translacije same biljke. Koristeći dubinsku neuronsku mrežu, npr. spomenuti YOLO algoritam, letjelica bi mogla detektirati biljke iz cijelog spektra te računati i rotaciju u odnosu na samu kameru.



Slika 2.2. Odnos okvira

Pohranjena pozicija biljke se programski lako izvodi pomoću tf biblioteke, a sam algebarski prijelaz je prikazan u nastavku.

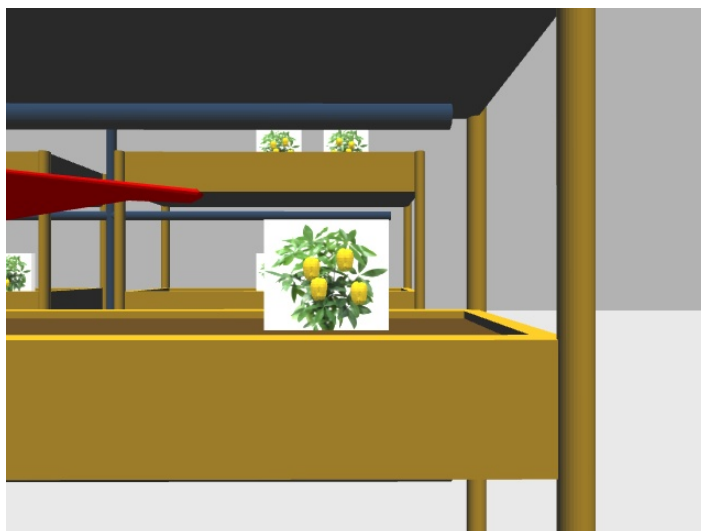
$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{p}'_h = \mathbf{T}\mathbf{p}_h$$

$$\mathbf{T}_{world}^{camera} = \mathbf{T}_{world}^{map} \cdot \mathbf{T}_{map}^{baselink} \cdot \mathbf{T}_{baselink}^{camera}$$

2.4. Detekcija biljaka

Iz razloga jednostavnosti slike, klasifikacija objekata može se izvršiti putem segmentacije temeljene na osnovnoj boji. Unatoč tome, u kodu je implementirana istrenirana verzija YOLO algoritma, verzija 7, s namjerom proširenja mogućnosti za budući rad s kompleksnijim objektima. YOLO, što znači "You Only Look Once", je popularan algo-



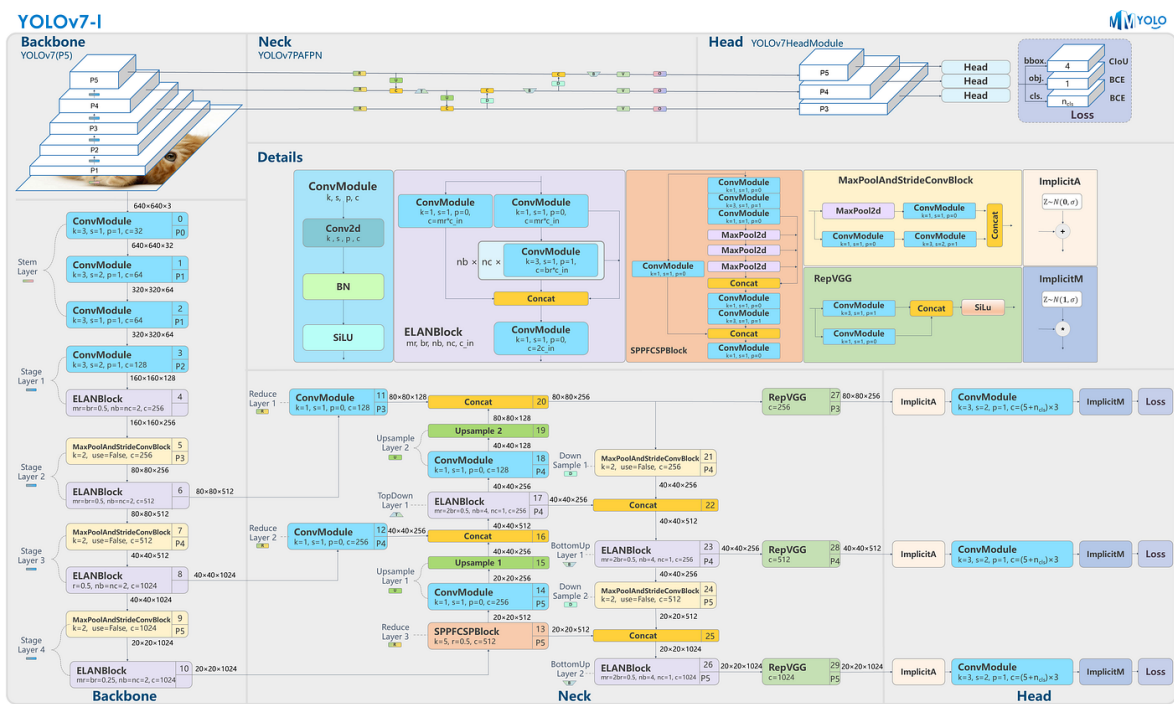
Slika 2.3. Perspektiva kamere letjelice

ritam za detekciju objekata u slikama i videozapisima. Algoritam dijeli sliku na mrežu (eng. *grid*) i za svaku ćeliju u mreži predviđa više okvira (eng. *bounding boxes*) i vjerojatnosti pripadnosti različitim klasama.

Glavna prednost YOLO algoritma je u tome što izvršava detekciju u jednom prolazu kroz neuronsku mrežu, čineći ga bržim od tradicionalnih pristupa koji koriste više koraka. Osim toga, YOLO ima tendenciju da ima bolju generalizaciju i preciznost u detekciji objekata različitih veličina i pozicija u odnosu na druge algoritme.

YOLO arhitektura obično slijedi konvolucijski model dubokog učenja, kao što su Darknet ili Tiny YOLO, sastavljen od nekoliko slojeva konvolucije, slojeva sažimanja (eng. *pooling*), slojeva normalizacije, te slojeva potpuno povezanih (eng. *fully connected*) za klasifikaciju i regresiju bounding boxova.

Razlog zbog kojeg su potrebni višestruki prolazi kod ostalih dubokih neuronskih mreža leži u raznolikosti regija koje mogu imati različite veličine i razmjere. Konvolucijske mreže zahtijevaju ulaz fiksne veličine, zbog čega nije moguće izravno proslijediti regije različitih veličina kroz takvu mrežu. Kako bi se regije prilagodile ulazu konvolucijske mreže, moraju se izdvojiti i promijeniti veličinu. Ova prilagodba omogućuje konzistentnu obradu različitih regija unutar mreže. [4]



Slika 2.4. Arhitektura YOLOv7; *medium.com*

3. Mogućnost daljnjeg rada

Nadogradnja programskog dijela rada biti će dovršiti implementaciju koda, testirati te grafički prikazati rezultate. Također fokus će biti i na poboljšanju svakog segmenta kako bi letjelica imala funkcionalnosti za kompleksnije zadatke. Voxelizacija detektiranog svijeta te prikazivanje pomoću pythonovih biblioteka sljedeći je korak, a dodatak koji nakon cijelog koda može biti implementiran za potpuno rješenje izazova samog ICUAS natjecanja je implementirati planiranje trajektorije.

1. Prvi korak u planiranju trajektorije je prikupljanje podataka o okolini. To uključuje mapiranje terena, detekciju prepreka i identifikaciju važnih objekata poput biljaka.
2. Na temelju prikupljenih podataka generira se digitalna mapa okoline. Ova mapa sadrži informacije o granicama vrta, položaju biljaka i preprekama.
3. Algoritmi planiranja putanje koriste informacije iz mape kako bi odabrali optimalnu putanju za letjelicu. Ovo uključuje izbjegavanje prepreka, minimiziranje vremena putovanja i prilagodbu putanje prema specifičnim zahtjevima misije.
4. Kada se odabere optimalna putanja, kontrola letjelice se programira kako bi slijedila tu putanju. Ovo uključuje upravljanje brzinom, smjerom i visinom leta letjelice.
5. Tijekom izvođenja misije inspekcije, sustav stalno prati položaj letjelice i okoliš kako bi mogao reagirati na neočekivane situacije ili promjene u okolini. [5]

4. Literatura

Lovro Markovic and Frano Petric and Antun Ivanovic and Jurica Goricanec and Marko Car and Matko Orsag and Stjepan Bogdan, Towards A Standardized Aerial Platform: ICUAS'22 Firefighting Competition, Journal of Intelligent & Robotic Systems. [1]

R. K. Megalingam, A. R. D. HemaTejaAnirudhBabu, G. Sriram and V. S. YashwanthAvvari, "Implementation of a Person Following Robot in ROS-gazebo platform," 2022 International Conference for Advancement in Technology (ICONAT), Goa, India, 2022, pp. 1-5, doi: 10.1109/ICONAT53423.2022.9726010. [2]

V. F. Vidal, L. M. Honório, M. F. Santos, M. F. Silva, A. S. Cerqueira and E. J. Oliveira, "UAV vision aided positioning system for location and landing," 2017 18th International Carpathian Control Conference (ICCC), Sinaia, Romania, 2017, pp. 228-233, doi: 10.1109/CarpathianCC.2017.7970402. keywords: Cameras;Drones;Aircraft;Global Positioning System;Computer vision;Portable computers;Unmanned aerial vehicles;Computer vision;Robot vision systems;Image processing;Image recognition, [3]

Vukelić, L. (2023). Prepoznavanje objekata primjenom dubokog učenja (Završni rad). Zagreb: Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva. [4]

Gasparetto, A., Boscariol, P., Lanzutti, A., Vidoni, R. (2015). Path Planning and Trajectory Planning Algorithms: A General Overview. In: Carbone, G., Gomez-Bravo, F. (eds) Motion and Operation Planning of Robotic Systems. Mechanisms and Machine Science, vol 29. Springer, Cham. <https://doi.org/10.1007/978-3-319-14705-51> [5]