

# Inteligentno robotsko rukovanje i vizualna detekcija prehrambenih proizvoda

---

**Levanić, Dominik**

**Master's thesis / Diplomski rad**

**2022**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:235:227988>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-02-02**

*Repository / Repozitorij:*

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# **DIPLOMSKI RAD**

**Dominik Levanić**

Zagreb, 2022.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# DIPLOMSKI RAD

Mentor:

doc. dr. sc. Marko Švaco

Student:

Dominik Levanić

Komentor:

dr. sc. Filip Šuligoj

Zagreb, 2022.

*Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.*

*Zahvaljujem mentoru doc. dr. sc. Marku Švaci na dodijeljenoj temi i usmjeravanju te posebno komentoru dr. sc. Filipu Šuligoju na stručnim savjetima i strpljenju tijekom izrade rada.*

*Konačno, zahvaljujem obitelji i prijateljima koji su mi bili podrška tijekom cijelog studija.*

*Dominik Levanić*



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite  
Povjerenstvo za diplomske radove studija strojarstva za smjerove:  
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment,  
inženjerstvo materijala te mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa:	602-04/22-6/1
Ur. broj:	15-1703-22-

## DIPLOMSKI ZADATAK

Student: **DOMINIK LEVANIĆ** Mat. br.: 0035203743

Naslov rada na hrvatskom jeziku: **Intelligentno robotsko rukovanje i vizualna detekcija prehrambenih proizvoda**

Naslov rada na engleskom jeziku: **Intelligent robotic handling and visual detection of food products**

Opis zadatka:

Sustavi opskrbe i distribucije provode složene kombinacije aktivnosti (proizvodnja, rukovanje, skladištenje, transport, pakiranje) koje omogućavaju regionalno zadovoljavanje potreba za prehrambenim proizvodima. Zbog velike količine i varijabilnosti proizvoda distributivni su centri idealni za uvođenje fleksibilnih automatskih sustava, gdje je glavni izazov učinkovito procesiranje različitih kombinacija proizvoda bez potrebe za njihovim razdvajanjem u različite serije.

U ovom radu potrebno je razviti metode za detekciju odnosno klasifikaciju i lokalizaciju, a potom i robotsko rukovanje zadanog skupa proizvoda s radne površine što uključuje:


- Izradu skupa podataka i treniranje neuronske mreže YOLO algoritma za brzu detekciju zadanih proizvoda u 2D slici,
- Filtriranje RGB-D oblaka točaka i robusno određivanje 3D položaja (pozicija i orijentacija) detektiranog proizvoda,
- Planiranje robotskog hvatanja proizvoda na temelju njegove geometrije,
- Kalibraciju hvataljke i RGB-D vizijskog sustava kao senzora na robotskoj ruci ("in hand"),
- Razvoj korisničkog sučelja i komunikaciju između robotske ruke, računala i vizijskog sustava.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

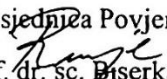
Zadatak zadan:  
11. studenog 2021.

Rok predaje rada:  
13. siječnja 2022.

Predvideni datum obrane:  
17. siječnja do 21. siječnja 2022.

Zadatak zadao:   
doc. dr. sc. Marko Švaco

Komentor:  
dr. sc. Filip Šuligoj

Predsjednica Povjerenstva:  
  
prof. dr. sc. Biserka Runje

## SADRŽAJ

POPIS SLIKA.....	II
POPIS TABLICA .....	IV
SAŽETAK.....	V
SUMMARY.....	VI
1. UVOD.....	1
1.1. Roboti.....	1
1.2. Vizijski sustavi.....	2
1.3. Problematika i pristup zadatku.....	4
2. ODABRANI POSTAV SUSTAVA ZA ROBOTSKO RUKOVANJE .....	6
2.1. Kolaborativni robot sa hvataljkom.....	7
2.2. 3D vizijski sustav .....	9
2.3. Neuronska mreža YOLO v5.....	10
2.4. Knjižnica za obradu oblaka točaka Open3D.....	11
2.5. Komunikacija između elemenata sustava .....	12
2.6. Grafičko korisničko sučelje.....	13
3. KALIBRACIJA 3D VIZIJSKOG SUSTAVA .....	14
3.1. Postavljanje hvataljke i 3D vizijskog sustava na robot.....	14
3.2. Transformacije između koordinatnih sustava .....	15
3.3. Kalibracija 3D vizijskog sustava naspram hvataljke.....	16
3.4. Analiza rezultata kalibracije.....	19
4. ROBOTSKO RUKOVANJE OBJEKTIMA .....	22
4.1. Detekcija objekata.....	22
4.1.1. Izrada i označavanje skupa slika .....	22
4.1.2. Treniranje neuronske mreže .....	26
4.1.3. Analiza rezultata treniranja.....	27
4.2. Prostorna lokalizacija objekata.....	35
4.3. Planiranje robotskog hvatanja .....	40
4.3. Komunikacija i robotsko hvatanje.....	42
4.4. Analiza rezultata robotskog rukovanja.....	46
5. EVALUACIJA SUSTAVA ZA ROBOTSKO RUKOVANJE .....	50
5.1. Evaluacija i moguće primjene sustava .....	50
5.2. Izazovi i moguća poboljšanja sustava .....	51
6. ZAKLJUČAK.....	52
7. LITERATURA .....	53

**POPIS SLIKA**

Slika 1. Industrijski robot [2].....	2
Slika 2. Vizijski sustav [3].....	3
Slika 3. <i>Eye-in-hand</i> kalibracija vizijskog sustava naspram vrha alata [5] .....	5
Slika 4. Postav sustava za robotsko rukovanje .....	6
Slika 5. Roboti Universal Robots UR3, UR5, UR10 i UR16 [6] .....	8
Slika 6. Hvataljka Robotiq 2F-140 [7] .....	8
Slika 7. Stereovizijski sustav Intel RealSense D435 [8] .....	9
Slika 8. Detekcija objekata YOLO v5 neuronskom mrežom [4].....	11
Slika 9. Obrada oblaka točaka [9] .....	12
Slika 10. <i>Three-way handshake</i> TCP protokola [10] .....	13
Slika 11. Grafičko korisničko sučelje [11] .....	13
Slika 12. Konstruirana prirubnica i adapterska ploča.....	15
Slika 13. Sklop izmjenjivača alata, hvataljke i 3D vizijskog sustava.....	15
Slika 14. Transformacija koordinatnog sustava homogenom matricom .....	16
Slika 15. Rekonstrukcija koordinatnog sustava slike šahovnice .....	17
Slika 16. Transformacije <i>eye-in-hand</i> kalibracije [12].....	18
Slika 17. Dijagram toka <i>eye-in-hand</i> kalibracije .....	19
Slika 18. Usporedba kalibriranih vektora translacije i rotacije .....	20
Slika 19. Razvijeno grafičko sučelje za kalibraciju 3D vizijskog sustava .....	21
Slika 20. Odabrane klase prehrambenih proizvoda .....	23
Slika 21. Dio skupa prikupljenih slika sa više proizvoda.....	24
Slika 22. Dio skupa prikupljenih slika bez proizvoda .....	25
Slika 23. Označavanje slike prehrambenih proizvoda.....	26
Slika 24. Odnosi osjetljivosti i preciznosti [14] .....	28
Slika 25. Treniranje neuronske mreže sa različitim stopama učenja.....	29
Slika 26. Treniranje neuronske mreže sa različitim veličinama skupova slika .....	30
Slika 27. Treniranje neuronske mreže sa različitim arhitekturama .....	31
Slika 28. Treniranje neuronske mreže sa različitim rezolucijama slika .....	32
Slika 29. Matrica konfuzije i detekcija proizvoda za početni skup rezolucije $240 \times 180$ .....	33
Slika 30. Matrica konfuzije i detekcija proizvoda za prošireni skup rezolucije $240 \times 180$ .....	33
Slika 31. Matrica konfuzije i detekcija proizvoda za prošireni skup rezolucije $480 \times 360$ .....	33
Slika 32. Matrica konfuzije i detekcija proizvoda za prošireni skup rezolucije $640 \times 480$ .....	34
Slika 33. Detekcije više prehrambenih proizvoda .....	34
Slika 34. Slika u boji i dubinska slika objekata .....	35
Slika 35. Oblak točaka u koordinatnom sustavu kamere.....	36

---

Slika 36. Oblak točaka scene u koordinatnom sustavu baze robota .....	37
Slika 37. Filtrirani oblak točaka u koordinatnom sustavu baze robota .....	38
Slika 38. Izrezivanje oblaka točaka u koordinatnom sustavu baze robota .....	39
Slika 39. Izolirani oblak točaka u koordinatnom sustavu baze robota .....	39
Slika 40. Uzorkovana konveksna ljuska oblaka točaka sa glavnim komponentama i položajima za hvatanje .....	41
Slika 41. Filtrirani oblak točaka sa vizualizacijom hvatanja proizvoda .....	42
Slika 42. Robotsko hvatanje proizvoda .....	43
Slika 43. Dijagram toka robotskog rukovanja objektima .....	44
Slika 44. Dijagram toka prostorne lokalizacije objekata .....	45
Slika 45. Robotsko hvatanje jedinog proizvoda na sceni .....	47
Slika 46. Robotsko hvatanje jednog iz grupe proizvoda na sceni .....	48
Slika 47. Razvijeno grafičko sučelje za robotsko rukovanje objektima .....	49



## **POPIS TABLICA**

Tablica 1. Tehničke specifikacije robota Universal Robots UR5 [6].....	7
Tablica 2. Tehničke specifikacije sustava Intel RealSense D435 [8].....	10

## SAŽETAK

U današnje vrijeme proizvodnja postaje sve fleksibilnija, zbog čega se kod robotskog rukovanja javlja sve veća potreba za brзом prilagodljivošću i invarijantnošću prema svojstvima proizvoda. Zadatak ovog diplomskog rada je osmisliti i integrirati sustav za inteligentno robotsko rukovanje prehrambenim proizvodima, temeljen na prikupljanju njihovih slika i oblaka točaka, detekciji na slikama neuronskom mrežom, određivanju položaja u prostoru te planiranju i realiziranju hvatanja na temelju geometrije. Korištene se slike i oblaci točaka prikupljaju 3D vizijskim sustavom Intel RealSense D435 postavljenim na kolaborativnog robota Universal Robots UR5, uz provođenje kalibracije 3D vizijskog sustava naspram robotske hvataljke. Za detekciju proizvoda na slikama koristi se neuronska mreža YOLO v5, trenirana na posebno izrađenom skupu označenih slika, dok se za određivanje položaja proizvoda u prostoru koristi knjižnica Open3D za obradu oblaka točaka. Kao temelj za planiranje hvatanja proizvoda koristi se algoritam za analizu glavnih komponenti (eng. *Principal component analysis – PCA*). Navedeni se elementi sustava povezuju grafičkim korisničkim sučeljem, dok se komunikacija između robota i računala ostvaruje TCP/IP protokolom. U radu se najprije daje uvod u navedeno područje s detaljima zadatka, nakon čega se opisuje način integracije svakog od elemenata sustava, uz zasebne analize njihovih rezultata. Zadatak se zaokružuje analizom rezultata i evaluacijom sustava u cjelini, nakon čega se izvodi odgovarajući zaključak.

Ključne riječi: robotsko rukovanje, detekcija objekata, oblak točaka, 3D vizijski sustav

## SUMMARY

Nowadays, production is becoming more flexible, which is why robotic handling is increasingly in the need of rapid adaptability and invariance to product properties. The goal of this master thesis is to design and integrate a system for intelligent robotic handling of food products, based on collecting their images and point clouds, detection in images by a neural network, determining spatial position and planning and realizing grasping based on geometry. The images and point clouds used are collected by an Intel RealSense D435 3D vision system mounted on a Universal Robots UR5 collaborative robot, with an *eye-in-hand* calibration of the 3D vision system against his gripper. The YOLO v5 neural network, trained on a specially designed set of annotated images is used to detect the product in images, while the Open3D library for point cloud processing is used to determine the spatial position of the product. The Principal component analysis (PCA) algorithm is used as the basis for planning of product grasping. The listed elements of the system are connected by a graphical user interface, while the communication between the robot and computer is realized by the TCP/IP protocol. The thesis first gives an introduction to stated area with details of the task, after which the method of integration for each element of the system is described, with separate analyses of their results. The task is completed by analyzing the results and evaluating the complete system, after which an appropriate conclusion is given.

Key words: robotic handling, object detection, point cloud, 3D vision system

## 1. UVOD

Robotika je interdisciplinarno područje koje predstavlja sinergiju strojarstva, elektrotehnike i računarstva sa čvrstom matematičkom podlogom te kao takva pruža mogućnost za integraciju u širokom rasponu teoretskih i praktičnih problema. Razvoj robotike potaknut je interesom čovjeka da olakša mukotrpane i nezanimljive poslove, a svoje mjesto u najvišoj mjeri pronalazi u industriji, u smislu daljnje automatizacije proizvodnih procesa. Roboti u industriji danas značajno utječu na održavanje ujednačene kvalitete proizvoda, efikasnog iskorištavanja materijala, sigurnosti radnika te eliminiranja ljudski grešaka, čemu je glavni razlog mnogo veća preciznost i ponovljivost robota u radu od čovjeka. [1]

Većina robota u industriji barem u nekom dijelu proizvodnog procesa treba ostvariti fizičku interakciju sa predmetom rada, zbog čega je u većini slučajeva potrebno obratiti pažnju na robotsko rukovanje. Rukovanje predstavlja veoma kompleksan zadatak kako za čovjeka, tako i za robota pa još nije definirano poopćeno rješenje problema robotskog rukovanja za različite predmete u nestrukturiranoj okolini. U kontekstu sve fleksibilnije proizvodnje, javlja se sve veća potreba za brzom prilagodljivošću i invarijantnošću robotskog rukovanja prema svojstvima predmeta. Kako bi robot mogao uspješno rukovati predmetom bez eksplicitnog programiranja, mora imati sposobnost pronalaženja predmeta u njegovoj okolini, planiranja trajektorije za njegovo hvatanje te određenu razinu koordinacije i autonomije u radnoj okolini.

### 1.1. Roboti

Pojam robota je u današnje vrijeme vrlo širok te ne postoji njegova jednoznačna definicija. Svaki pokušaj stvaranja definicije treba razmatrati u kontekstu ondašnjeg tehnološkog razvoja i problema koji zaokupljaju čovjeka. Ipak, pod pojmom robota se u današnje vrijeme najčešće misli na fizičke sustave koju su programabilni, barem u određenoj mjeri autonomni, sposobni prikupljati podatke iz svoje okoline i mijenjati ju svojim djelovanjem. Takvi se sustavi protežu od jednostavnih uređaja namijenjenih za svakodnevnu pomoć čovjeku pa sve do humanoidnih robota namijenjenih prezentiranju trenutnih znanstvenih dostignuća. Pojam robot je prvi put upotrijebio češki pisac 1920. godine te on dolazi iz slavenske riječi koja označava podčinjen posao. Prvi roboti u današnjem smislu riječi s namjenom korištenja u industriji javili su se sredinom 20. stoljeća, a od tog vremena njihov razvoj uzima sve veći zamah. Najveću primjenu doživljavaju u industriji,

gdje zbog sve utjecajnije automatizacije proizvodnih procesa postaju sve nezaobilazniji. Današnji industrijski roboti se obično sastoje od baze, robotske ruke s nekoliko zglobova te alata pričvršćenog na njezin kraj, a jedan je takav robot prikazan na slici 1. Takvi se roboti u velikoj mjeri koriste za poslove kao što pakiranje, bojanje i točkasto zavarivanje, čemu je glavni razlog mnogo veća preciznost i ponovljivost robota u radu od čovjeka. [1]

Kako bi se roboti mogli prilagođavati sve fleksibilnijoj proizvodnji, nužno ih je integrirati u sustave koji imaju veću mogućnost interakcije sa svojom okolinom. Vrlo često se koriste s vizijskim sustavima, koji ne ometaju bitno rad robota, a mogu pružiti značajnu količinu podataka o okolini.



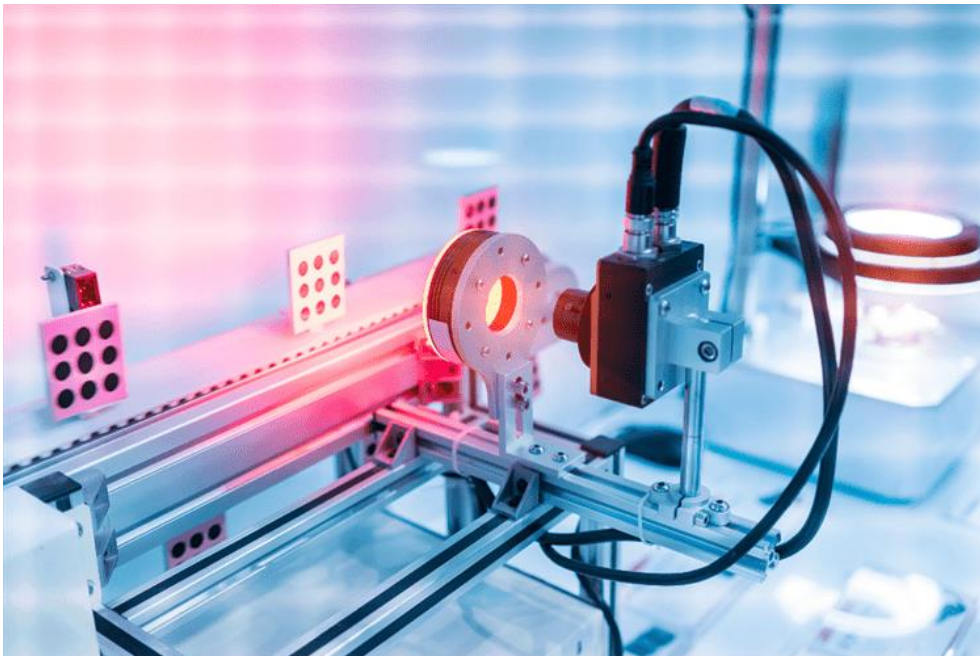
**Slika 1. Industrijski robot [2]**

## **1.2. Vizijski sustavi**

Pojam vizijskih sustava jednoznačniji je od pojma robota te označava sustav za prikupljanje podataka o okolini sastavljen od vizijskog procesora, leće i optičkog senzora kamere te opcionalnih dodatnih elemenata, pri čemu je jedan takav sustav prikazan na slici 2. Područje primjene vizijskih sustava vrlo je kompleksno zbog principa obrade podataka prikupljenih vizualnim putem, gdje je podatke potrebno smještati u kontekst pripadajuće okoline te voditi računa o mnoštvu vanjskih parametara kao što su osvjetljenje i zaklonjenost objekata. Upravo su te stavke razlog zbog kojeg

je ljudski vid za većinu kompleksnih zadataka još uvijek mnogo bolji od računalnog, a što ukazuje na visoku mogućnost razvoja vizijskih sustava uz razvoj tehnologije. [3]

U područje vizijskih sustava ulazi mnoštvo problema, koji se često sastoje od dijelova međusobno slične naravi pa se tada za njihovo rješavanje mogu koristiti zajednički dijelovi algoritama. Jedan od učestalijih problema je i detekcija objekata, koju treba jasno odijeliti od pojmova klasifikacije i lokalizacije. Klasifikacija objekata podrazumijeva određivanje klase jednog objekta koji se nalazi negdje na slici, dok se lokalizacija odnosi na određivanje vanjskih pravokutnih rubova također jednog objekta koji se nalazi na slici. Tek se objašnjavanjem prethodnih pojmova može govoriti o detekciji objekata, koja podrazumijeva spoj klasifikacije i lokalizacije više objekata na slici. Uz detekciju objekata spomenuti i vrlo sličan pojam segmentacije objekata, čija je razlika od detekcije u tome da vanjski rubovi objekata koji se određuju nisu nužno pravokutni, nego odgovaraju stvarnim konturama. Za rješavanje tih problema se zadnjih desetak godina, zbog ubrzanog napretka performansi računala, sve više primjenjuju neuronske mreže. One predstavljaju skup velikog broja parametara međusobno povezanih određenim matematičkim funkcijama, gdje se na temelju poznatih rezultata skupa podataka za treniranje optimiraju vrijednosti parametara. Navedeno se optimiranje obavlja u svrhu što bolje predikcije rezultata na još neviđenim podacima. [4]



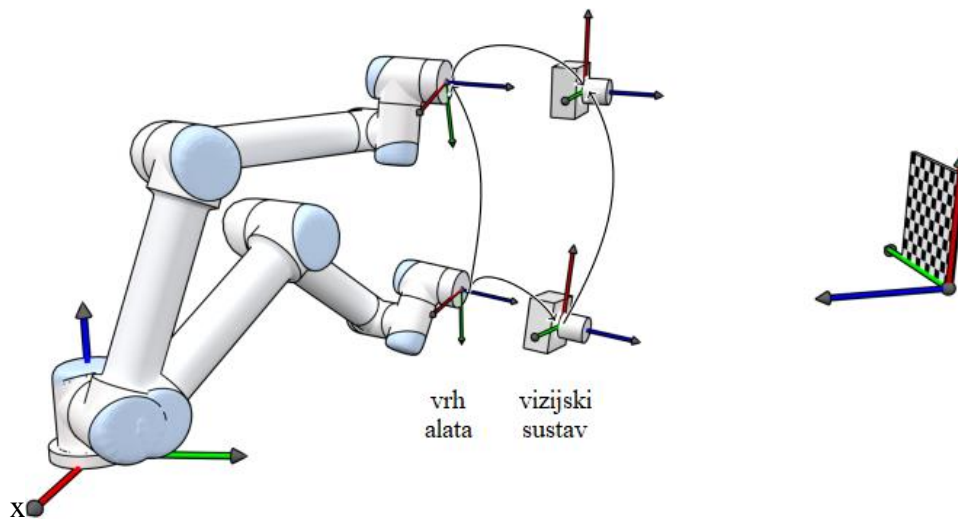
**Slika 2. Vizijski sustav [3]**

### 1.3. Problematika i pristup zadatku

Problematika zadatka ovog rada temelji se na osmišljavanju i integriranju sustava koji će bez eksplicitnog zadavanja pozicija i orijentacija predmeta moći prepoznati određeni između više drugih na sceni, uspješno ga lokalizirati, a zatim uhvatiti robotom. Općeniti problem prepoznavanja predmeta podrazumijeva određivanje klase kojoj predmet pripada, za što je potrebno posjedovati bazu svih klasa, dok je općeniti problem lokalizacije predmeta određivanje njegove pozicije i orijentacije u prostoru. Problem robotskog hvatanja, s druge strane, podrazumijeva ispravno pozicioniranje hvataljke te hvatanje predmeta bez sudaranja s bilo čime u okolini.

Osmišljeno je da se podaci o sceni koja uključuje predmete i njihovu okolinu prikupljaju u obliku oblaka točka 3D vizijskim sustavom postavljenim na robota, obrađuju na računalu u svrhu prepoznavanja i lokalizacije predmeta, a zatim koriste za njegovo hvatanje. Obrada podataka se obavlja s ciljem određivanja pozicija težišta i glavnih osi inercije predmeta, a sve pod opravdanom pretpostavkom homogenosti odabranih predmeta. Zbog te je pretpostavke pozicije težišta predmeta moguće poistovjetiti s pozicijama njihovih geometrijskih središta, a pozitivne smjerove glavnih osi inercije s vektorima njihove orijentaciju.

Kako bi se 3D vizijskim sustavom mogao dobiti vjerni prikaz cijele scene, prikupljanje podataka je potrebno obaviti iz više položaja, gdje izraz položaj zajednički označava poziciju i orijentaciju vezanu uz odgovarajući koordinatni sustav. Za prikupljanje podataka se 3D vizijski sustav mora pomicati u prostoru, a podatke o položajima predmeta je potrebno vezani uz globalni koordinatni sustav koji je statični. Kako se s robota u realnom vremenu mogu prikupljati položaji njegovog vrha alata u odnosu na bazu, logično je 3D vizijski sustav postaviti uz alat, na mjesto na kojem ne smeta njegovom radu, te kao globalni koordinatni sustav odabrati koordinatni sustav baze robota. Takvim postavljanjem 3D vizijskog sustava bilo bi vrlo teško definirati odnos njegovog koordinatnog sustava naspram koordinatnog sustava vrha alata mjerenjem pa je to potrebno napraviti analitički, što se naziva *eye-in-hand* kalibracijom i prikazano je na slici 3. Često se može naići na pojam *hand-eye* ili *hand-to-eye* kalibracija koji razmatra situaciju u kojoj je vizijski sustav postavljen nepomično u prostoru, dok je alat pomičan.



**Slika 3. Eye-in-hand kalibracija vizijskog sustava naspram vrha alata [5]**

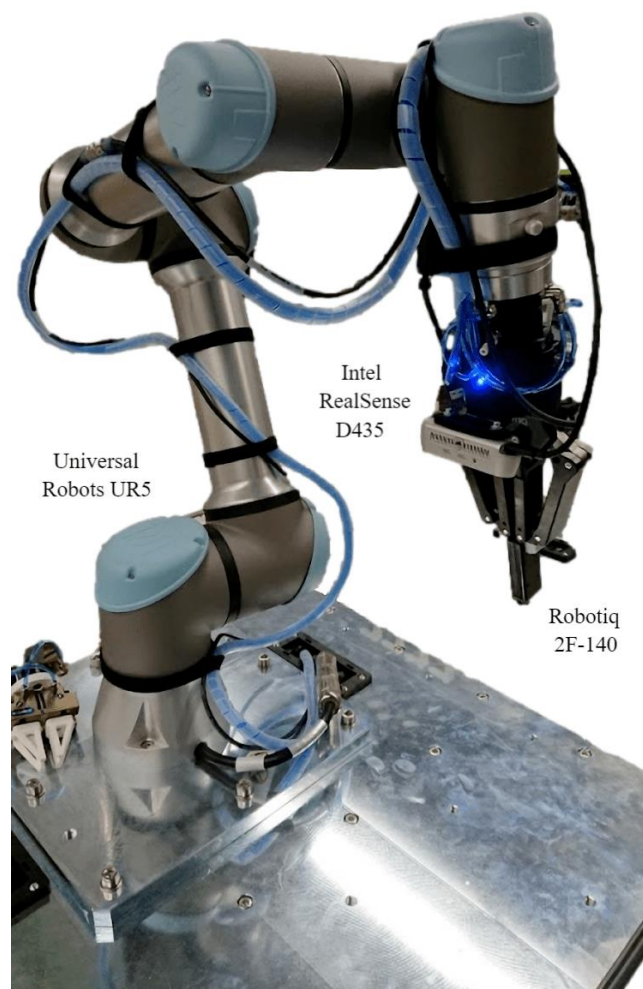
Nakon dobivanja prikaza cijele scene pomoću kalibriranog 3D vizijskog sustava, sa scene je potrebno izolirati odabrane predmete. Pošto je 3D vizijskim sustavom moguće prikupiti podatke o dubini na proizvoljnim koordinatama slike, u svrhu dobivanja prostornih koordinata rubova predmeta ima smisla koristiti detekciju predmeta sa slika, gdje će se kombinacijom različitih položaja 3D vizijskog sustava moći prikupiti vjerniji podaci o rubovima predmeta. Kako bi detekcija predmeta bila što robusnija i invarijantna s obzirom na položaje 3D vizijskog sustava iz kojih se prikupljaju slike, koristi se neuronska mreža. Bitna značajka korištenja neuronske mreže je potreba za skupom slika na kojem se mora istrenirati da bi davala vjerodostojne rezultate. Mogući izazovi opisane detekcije su uzimanje u obzir samo dijelova predmeta, pogrešna detekcija predmeta ili izostanak detekcije, što se mora uzeti u obzir kod osmišljavanja rješenja. Razmotrena je i mogućnost detekcije predmeta direktno iz oblaka točaka, no vrlo je brzo odbačena zbog slabe dostupnosti trenutnih algoritama na upotrebljivoj razini.

Planiranje hvatanja detektiranih predmeta se zatim obavlja prema njihovim pozicijama geometrijskih središta i orijentacijama, obradom njihovih oblaka točaka postupkom temeljenim na algoritmu analize glavnih komponenti. Da bi se s predmetima mogla ostvariti fizička interakcija, dobivene je podatke potrebno poslati robotu i na njemu izvršiti program kojim se on pozicionira u zadani položaj za hvatanje, hvata detektirani predmet te se miče iz položaja za hvatanje. Proces robotskog rukovanja predmetima time se zaokružuje, no radi lakšeg korištenja osmišljenog sustava, nameće se potreba za implementiranjem grafičkog sučelja na računalu. Grafičkim će sučeljem korisnik moći kalibrirati sustav za različito postavljanje 3D vizijskog sustava na robot te će moći proizvoljan broj puta obavljati cijeli proces prikupljanja i obrade podataka o sceni, planiranja hvatanja, a zatim i njegove realizacije za detektirane predmete.



## 2. ODABRANI POSTAV SUSTAVA ZA ROBOTSKO RUKOVANJE

Osmišljen sustav za robotsko rukovanje predmetima je cjelina koja je sastavljena od nekoliko glavnih elemenata, a prikazana je na slici 4. U svrhu prikupljanja podataka o sceni i hvatanje predmeta korišteni su kolaborativni robot Universal Robots UR5 sa hvataljkom Robotiq 2F-140 (opisano u potpoglavlju 2.1.) te 3D vizijski sustav Intel RealSense D435 (opisano u potpoglavlju 2.2.), za detekciju predmeta sa scene koristi se neuronska mreža YOLO v5 (opisano u potpoglavlju 2.3.), a za obradu dobivenih podataka knjižnica za obradu oblaka točaka Open3D (opisano u potpoglavlju 2.4.). Kako bi se između elemenata sustava ostvarila potrebna interakcija, elementi sustava su povezani komunikacijom preko TCP/IP protokola (opisano u potpoglavlju 2.5.) te korisničkim sučeljem (opisano u potpoglavlju 2.6.). Razvijena programska podrška se većim dijelom implementira na računalu u programskom jeziku Python te manjim na robotu u korisničkom sučelju PolyScope.



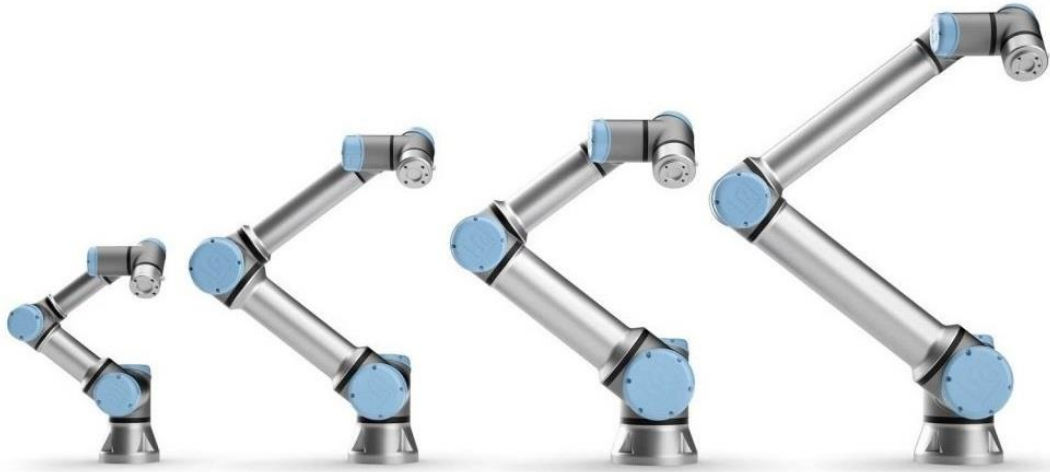
Slika 4. Postav sustava za robotsko rukovanje

## 2.1. Kolaborativni robot sa hvataljkom

Priroda ovog zadatka definira potrebu za što pristupačnijom robotom kojim može raditi u bliskom ljudskom okruženju se može jednostavno koristiti, zbog čega je odabran kolaborativni robot tvrtke Universal Robots. Ta je tvrtka osnovana s ciljem približavanja robotskih tehnologija manjim i srednjim poduzetnicima, a glavna joj je djelatnost proizvodnja kolaborativnih robota. Kolaborativni su roboti pogodni za rad u ljudskom okruženju bez sigurnosnih kaveza, a kao takvi na tržištu od navedene tvrtke postoje modeli UR3, UR5, UR10 i UR16, kojima brojevi u nazivu označavaju nosivost u kilogramima, a prikazani su na slici 5. Sva četiri modela su šesteroosna, vrlo lagana sa brzinom rotacije zglobova od  $180^\circ/\text{s}$  uz točnost ponavljanja od  $\pm 0.1$  mm te se mogu programirati preko korisničkog sučelja PolyScope pomoću privjeska za učenje (eng. *teach pendant*). Odabran je model UR5, zato što nosivošću i dimenzijama najbolje odgovara potrebama ovog zadatka, pri čemu su njegove tehničke specifikacije dane u tablici 1. [6]

**Tablica 1. Tehničke specifikacije robota Universal Robots UR5 [6]**

Universal Robots UR5	
Masa:	18,4 kg
Nosivost:	5 kg
Doseg:	850 mm
Radni opseg zglobova:	$\pm 360^\circ$
Brzina zglobova:	$180^\circ/\text{s}$
Brzina alata:	1 mm/s
Ponovljivost:	$\pm 0,1$ mm
Broj stupnjeva slobode:	6
Komunikacija:	TCP/IP 100 Mbit, Modbus TCP
Potrošnja energije:	$\sim 200$ W
Napajanje:	100 – 240 V AC sa 50 – 60 Hz
Temperaturno radno područje:	0 – 50 °C



**Slika 5. Roboti Universal Robots UR3, UR5, UR10 i UR16 [6]**

Skup predmeta kojima se rukuje odabran je tako da predmeti imaju različite dimenzije, geometriju i čvrstoću te vrlo male mase pa je potrebno je koristiti hvataljku promjenjive širine i sile hvatanja. Pošto je već odabran robot Universal Robots UR5, barem jedna dimenzija svakog predmeta je u rasponu od 20 do 100 mm, a princip rada hvataljke ne igra značajnu ulogu, odabrana je dvoprsta električna hvataljka modela Robotiq 2F-140. Navedena hvataljka ima raspon širina hvatanja od 0 do 140 mm, raspon sila hvatanja od 10 do 235 N te masu 1,025 kg, a prikazana je na slici 6. Vrlo korisna značajka te hvataljke je da automatski detektira je li uhvatila predmet te prema tome podešava širinu i silu hvatanja. [7]



**Slika 6. Hvataljka Robotiq 2F-140 [7]**

## 2.2. 3D vizijski sustav

Bitnu stavku sustava čini vizijski sustav za prikupljanje strukturiranih trodimenzionalnih podataka o sceni, što je moguće poistovjetiti s podacima u obliku oblaka točaka. Iz tog je razloga smisleno odabrati onaj vizijski sustav koji uz sliku već daje i podatke o dubini na proizvoljnim koordinatama slike. Ta je značajka upravo temelj komercijalnih 3D vizijskih sustava koji podrazumijevaju korištenje vizijskih procesora koja pretprocesiraju dohvaćene slike te dviju ili više kamera postavljenih u poznati međuođnos, iz kojih se na temelju korespondentnih točaka na slikama dobivaju podaci o njihovim udaljenostima od kamera, a što se naziva dubinskim slikama. Često koriste i kamere koje prikupljaju podatke iz infracrvenog spektra te time omogućuju dobivanje točnijih podataka pri lošem osvjetljenju. Najpoznatije komercijalne serije modela 3D vizijskih sustava su Microsoft Kinect i Intel RealSense.

Pošto je serija modela Intel RealSense dostupnija te pruža podršku za širi spektar programskih jezika, ona je i odabrana za korištenje u sustavu. Navedena je serija modela izvorno razvijena za integraciju s autonomnim sustavima i pametnim uređajima te je izašla na tržište 2015. godine, a trenutno uključuje razvojno okruženje za izradu aplikacija u desetak programskih jezika, od kojih su najčešće korišteni C++, Python i C#. Odabran je model D435, koji se sastoji od vizijskog procesora, dvije dubinske kamere za dobivanje dubinske slike, kamere koje prikuplja podatke iz infracrvenog spektra za povećanje točnosti pri lošem osvjetljenju te kamere za dobivanje slike u boji. Taj je model prikazan na slici 7, a njegove su tehničke specifikacije dane u tablici 2. [8]



**Slika 7. Stereovizijski sustav Intel RealSense D435 [8]**

**Tablica 2. Tehničke specifikacije sustava Intel RealSense D435 [8]**

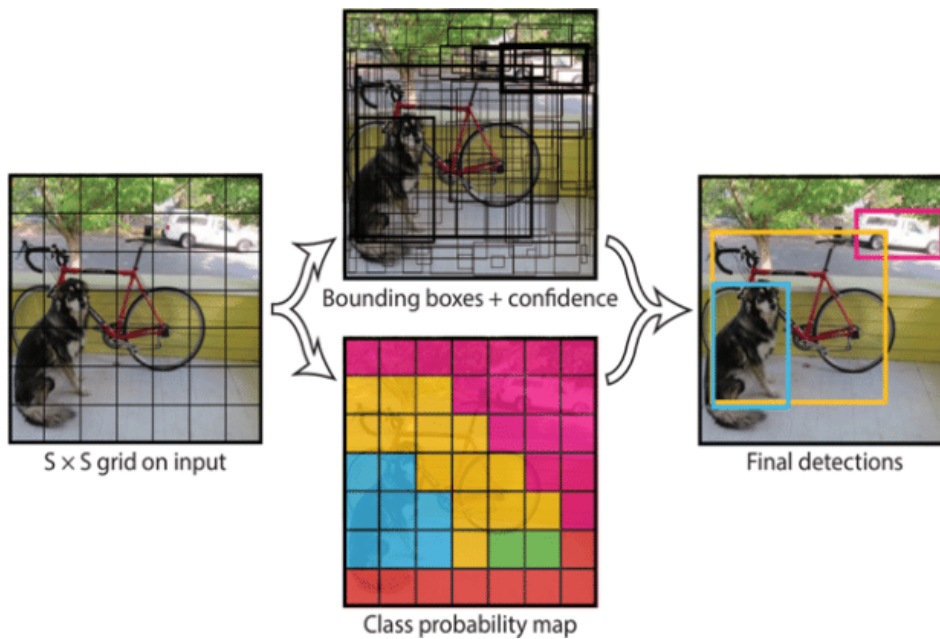
Intel RealSense D435	
Dimenzije:	50 × 25 × 25 mm
Garantirani raspon dubine:	0,3 – 3 m
Maksimalna pogreška dubine:	2% na 2 m
Vidno polje dubinske slike:	87° × 58°
Maksimalna rezolucija dubinske slike sa brojem slika u sekundi:	1280 × 720 sa 90 FPS
Vidno polje slike u boji:	69° × 42°
Maksimalna rezolucija slike u boji sa brojem slika u sekundi:	1920 × 1080 sa 60 FPS
Komunikacija:	USB-C 3.1

### 2.3. Neuronska mreža YOLO v5

Detekcija predmeta je stavka bez koje sustav ne bi bio sposoban razlučiti kojim je predmetom potrebno rukovati ni gdje se on u prostoru nalazi. Ako se vodi tim razmišljanjem, jasno je da detekcija predmeta treba biti što robusnija, dovoljno brza da sustav može funkcionirati u realnom vremenu te invarijantna s obzirom na položaje 3D vizijskog sustava iz kojih se prikupljaju slike. Zbog kompleksnosti opisanog problema, rješenje se traži u postojećoj arhitekturi koja daje pouzdane rezultate te se odabire korištenje neuronske mreže. Za obradu slika mrežama najefikasnije je korištenje onih koje se zasnivaju na principu konvolucije, što je matematička operacija za izvlačenje bitnih značajki sa slike. Postoji mnoštvo konvolucijskih neuronskih mreža koje se često značajno razlikuju po svojoj strukturi, a tako i po principu na kojem predviđaju rezultate na još neviđenim podacima.

Usporedbom pouzdanosti, brzine detekcije i jednostavnosti primjene različitih neuronskih mreža, odabir se svodi na mrežu YOLO v5. Njena najveća prednost je visoka brzina detekcije, gdje je sposobna detektirati objekte u realnom vremenu, te jednostavnost treniranja na vlastitom skupu označenih slika. Opisane se prednosti kompenziraju nešto manjom pouzdanošću od trenutno

najpouzdanijih mreža. YOLO v5 je relativno nova mreža, predstavljena 2020. godine, te se temelji na razdjeljivanju slike u područja jednakih dimenzija, određivanju onih područja na kojima bi se mogli nalaziti dijelovi objekata, obrade podataka na opisanim područjima, i konačno određivanju ukupnih područja na kojima se objekti nalaze. Princip detekcije objekata opisanom neuronskom mrežom dan je na slici 8. [4]



**Slika 8. Detekcija objekata YOLO v5 neuronskom mrežom [4]**

#### 2.4. Knjižnica za obradu oblaka točaka Open3D

Kako su podaci o sceni prikupljeni u obliku oblaka točaka, potrebno je u određenom programskom jeziku koristiti neku od knjižnica za obradu takvih podataka, pri čemu ona treba biti otvorenog koda. Raspon opisanih opcija se svodi na knjižnice PCL (eng. *Point Cloud Library*) i Open3D, koje obje sadrže mnoštvo implementiranih algoritama za obradu oblaka točaka. Iako PCL sadrži nešto širi spektar algoritama, radi jednostavnije integracije sa programskom podrškom za detekciju objekata koja je implementirana u programskom jeziku Python, odabrana je knjižnica Open3D. Algoritmi implementirani u sklopu knjižnice Open3D obuhvaćaju čitanje, zapisivanje, vizualizaciju, filtriranje, segmentaciju, registraciju te mnoge druge operacije nad oblacima točaka, a jedan primjer takvih operacija dan je na slici 9. Knjižnica je besplatna za akademske i komercijalne svrhe te je implementirana u programskim jezicima C++ i Python.

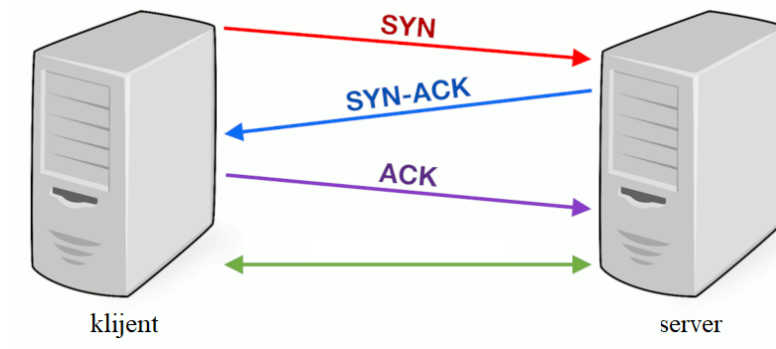




Slika 9. Obrada oblaka točaka [9]

## 2.5. Komunikacija između elemenata sustava

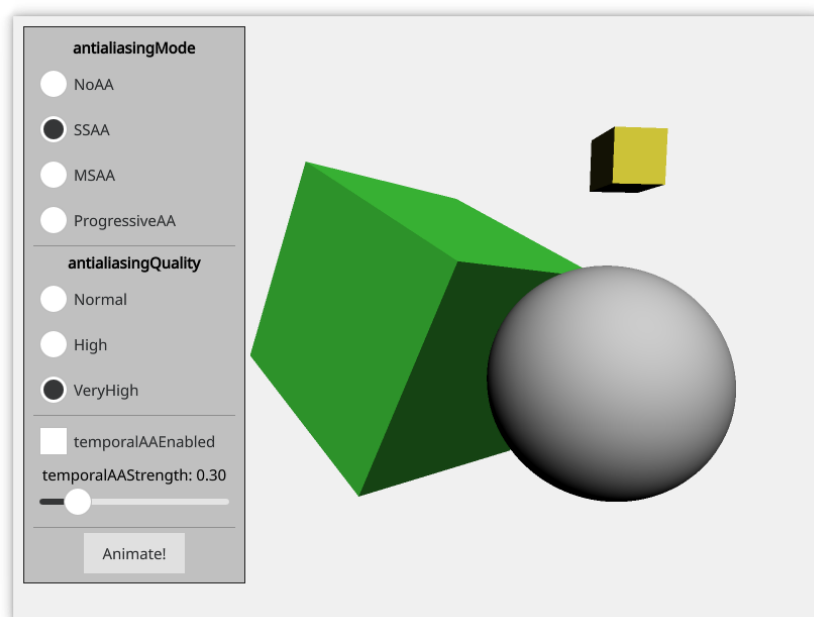
Kako se svi podaci obrađuju na računalu, između računala i robote te računala i vizijskog sustava je potrebna komunikacija. Komunikacija između računala i robota ostvaruje se putem TCP protokola (eng. *Transmission Control Protocol*), koji je jedan od osnovnih protokola unutar IP skupine, te osigurava pouzdanu izmjenu podataka između korisnika kontroliranim redoslijedom u obliku bajtova. Korisnici se nalaze ili u ulozi servera ili u ulozi klijenta, a unutar veze postoji jedan server i jedan ili više klijenata. Veza se uspostavlja tako da se pokrene server, klijenti koji se na njega žele povezati mu šalju SYN (eng. *synchronize*) segment, on za prihvaćanje odgovora sa SYN-ACK (eng. *synchronize-acknowledge*) segmentom, a klijenti opet sa ACK (eng. *acknowledge*) segmentom i tako potvrđuju povezivanje. Opisana se procedura naziva *three-way handshake* te je prikazana na slici 10, a da bi uopće bila moguća, nužno je da se server i klijenti nalaze u istoj podmreži (eng. *subnet*). Za potrebe ovog zadatka, računalo se nalazi u ulozi servera, a robot u ulozi klijenta. Prijenos podataka između računala i vizijskog sustava ostvaruje se putem USB-C-a, koji ujedno služi i za napajanje vizijskog sustava. [10]



Slika 10. *Three-way handshake* TCP protokola [10]

## 2.6. Grafičko korisničko sučelje

Priroda ovog zadatka definira potrebu za interakcijom korisnika sa sustavom, što najviše ima smisla realizirati grafičkim sučeljem. Njegova je svrha da korisniku omogući jednostavno kalibriranje sustava kod različitog postavljanja 3D vizijskog sustava na robota te jednostavno obavljanje cijelog procesa prikupljanja i obrade podataka o sceni, planiranja hvatanja, a zatim i njegove realizacije za odabrane predmete proizvoljan broj puta. Grafičko se sučelje razvija pomoću knjižnice Qt koja je besplatna za akademske i komercijalne svrhe te implementirana u programskim jezicima C++ i Python, a jedan primjer takvog grafičkog sučelja dan je na slici 11.



Slika 11. Grafičko korisničko sučelje [11]

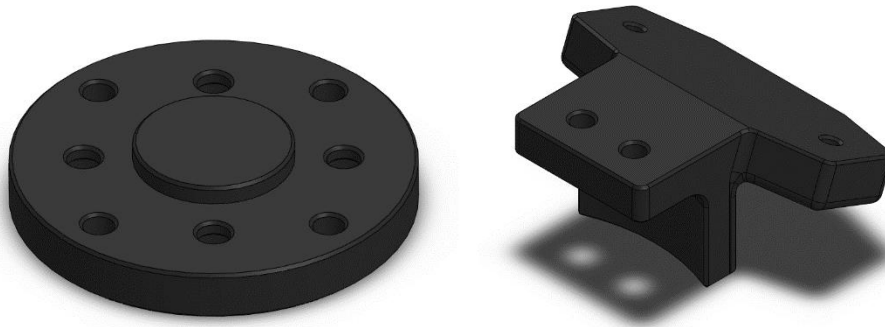


### 3. KALIBRACIJA 3D VIZIJSKOG SUSTAVA

Kod prikupljanja podataka o sceni iz više položaja, podatke o položajima predmeta je potrebno vezati uz globalni koordinatni sustav koji je statični. Pošto se za prikupljanje podataka 3D vizijski sustav mora pomicati u prostoru, potrebno je u svakom trenutku poznavati njegov položaj u odnosu na globalni koordinatni sustav. Kako se s robota u realnom vremenu mogu prikupljati podaci o položaju njegovog vrha alata u odnosu na bazu, rješenje se očituje u postavljanju 3D vizijskog sustava uz alat, na mjesto na kojem ne smeta njegovom radu. Također se kao globalni koordinatni sustav odabire koordinatni sustav baze robota. Opisanim postavljanjem 3D vizijskog sustava vrlo je teško direktno izmjeriti odnos njegovog koordinatnog sustava naspram koordinatnog sustava vrha alata pa je potrebno to napraviti analitički, što se naziva *eye-in-hand* kalibracijom. Uz navedenu, postoji i *hand-eye* ili *hand-to-eye* kalibracija koja razmatra situaciju u kojoj je vizijski sustav postavljen nepomično u prostoru, dok je alat pomičan.

#### 3.1. Postavljanje hvataljke i 3D vizijskog sustava na robot

Kao što je opisano u prethodnom poglavlju, za potrebe ovog zadatka se koriste kolaborativni robot Universal Robots UR5, hvataljka Robotiq 2F-140 i 3D vizijski sustav Intel RealSense D435, no da bi se oni mogli međusobno sklopiti, potrebno je konstruirati i izraditi određene prihvate. Kako bi se hvataljka mogla postaviti na izmjenjivač alata pričvršćen na kraj robotske ruke, konstruira se prirubnica vanjskog promjera  $\varnothing 63$  mm sa četiri prolazne rupe za navoje M5 na promjeru  $\varnothing 40$  mm te četiri navoja M6 na promjeru  $\varnothing 50$  mm. Slično opisanome, radi postavljanja 3D vizijskog sustava uz alat, konstruira se adapterska ploča s ojačanjem radijusa R37.5 mm, koje naliježe na cilindrični dio alata, s dvije prolazne rupe za navoje M3 međusobno udaljene 45 mm u svrhu pričvršćivanja na 3D vizijski sustav te dvije prolazne rupe za navoje M4 međusobno udaljene 12 mm u svrhu pričvršćivanja na alat. Ravnina adapterske ploče na koju naliježe vizijski sustav se postavlja ukoso da tijekom prikupljanja podataka o sceni alat ne bi ulazio u vidno polje vizijskog sustava. Konstruiranje prirubnice i adapterske ploče se izvodi u CAD programu SolidWorks, što je prikazano na slici 12, nakon čega se oba dijela izrađuju tehnologijom 3D printanja na printeru modela Prusa i3. Na izrađenoj prirubnici i adapterskoj ploči se urezuju potrebni navoji te se zatim svi korišteni dijelovi međusobno sklapaju, što je prikazano na slici 13.



Slika 12. Konstruirana prirubnica i adapterska ploča



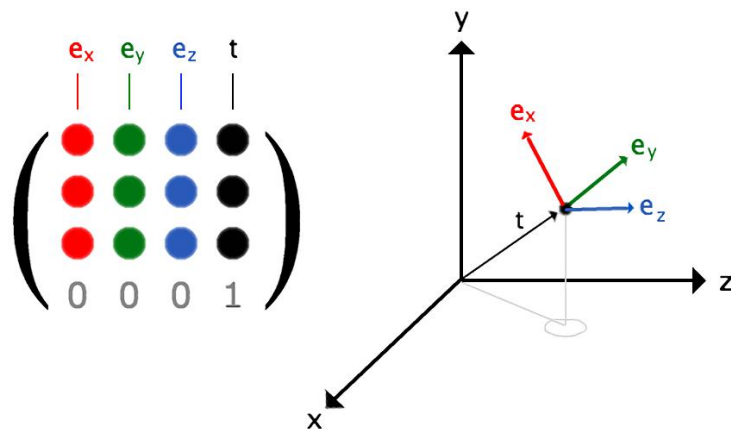
Slika 13. Sklop izmjenjivača alata, hvataljke i 3D vizijskog sustava

### 3.2. Transformacije između koordinatnih sustava

Kako bi se mogao opisati postupak *eye-in-hand* kalibracije, potrebno je najprije postaviti teoretsku podlogu za transformacije između koordinatnih sustava. Koordinatni sustavi korišteni u ovom radu vezuju se uz objekte koji se smatraju krutima i stalnih dimenzija pa su jedine radnje kojima se iz jednog koordinatnog sustava može prijeći u drugi translacija i rotacija. Pošto se podaci sa scene prikupljaju u tri dimenzije, oni se zapisuju u obliku vektora  $p$  dimenzija  $3 \times 1$ , njihova translacija  $t$  u istom tom obliku, a njihova rotacija u obliku matrice  $R$  dimenzija  $3 \times 3$ . U svrhu

zapisivanja translacije i rotacije odjednom, koristi se matrica homogene transformacije  $T$  dimenzija  $4 \times 4$ . Matricom homogene transformacije se u potpunosti može predstaviti transformacija između dva koordinatna sustava, što je prikazano na slici 14. Transformacija podataka između dva koordinatna sustava se obavlja njihovim zapisivanjem u obliku homogenih vektora  $p_H$  dimenzija  $4 \times 1$  te množenjem slijeva odgovarajućom matricom homogene transformacije. Izrazi za opisane vektore zapisuju se na sljedeći način:

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}, \quad R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \quad T = \begin{bmatrix} R & t \\ 0_{1 \times 3} & 1 \end{bmatrix}, \quad p_H = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.1)$$



Slika 14. Transformacija koordinatnog sustava homogenom matricom

### 3.3. Kalibracija 3D vizijskog sustava naspram hvataljke

*Eye-in-hand* kalibracija je postupak kojim se određuje transformacija koordinatnog sustava kamere na robotu u koordinatni sustav hvataljke. Često primjenjivan način kalibracije, a koji se koristi i u ovom radu, je kalibracija trodimenzionalnom rekonstrukcijom scene pomoću šahovnice. Takvom se kalibracijom iz slika nastoji zaključiti o trodimenzionalnim podacima sa scene, a postupak se svodi na slikanje šahovnice poznatog broja polja u recima i stupcima iz više položaja te obradi navedenih slika algoritmima računalnog vida. Za implementaciju algoritama računalnog vida se ovom radu koristi knjižnica otvorenog koda OpenCV. Šahovnica se koristi poglavito jer ima visoko izražene i ortogonalne rubove crnih i bijelih polja, a to osigurava robusnost postupka i mogućnost korištenja samo slika sivih tonova. Na svakoj od slika se određenim algoritmom

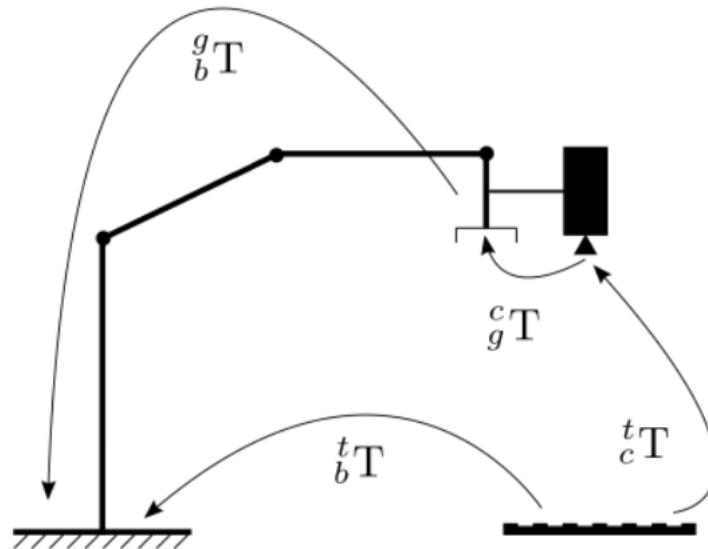
računalnog vida najprije prepoznaju rubovi i uglovi polja, za što se u ovom radu koristi funkcija `cv2.findChessboardCorners()` knjižnice OpenCV. Dobiveni su uglovi šahovnice već sortirani na određeni način te se odabire jedan ugao koji će se pratiti u svakoj slici, dok će ostali služiti samo za povećanje robusnosti postupka kalibracije.

Ideja ovog dijela postupka je postavljanje koordinatnog sustava u odabrani ugao šahovnice na slici te određivanje transformacije njegovog koordinatnog sustava u sustav kamere. Dvije osi postavljenog koordinatnog sustava čine ortogonalni rubovi šahovnice, dok se treća definira njihovim vektorskim umnoškom, što znači da je  $xy$ -ravnina tog sustava zapravo ravnina šahovnice. Za određivanje opisane transformacije, trebaju se poznavati stvarna duljina rubova šahovskih polja te intrinzični parametri kamere, što su fokalna duljina i koordinata središta slike po obje osi slike u pikselima. Opisana se transformacija, za potrebe ovog rada, dobiva korištenjem funkcije `cv2.solvePnP()`, a rezultat njenog određivanja prikazan je na slici 15.



**Slika 15. Rekonstrukcija koordinatnog sustava slike šahovnice**

Središnji dio postupka kalibracije motiviran je korištenjem transformacije koordinatnog sustava hvataljke u koordinatni sustav baze robota ( ${}^g_bT$ , gdje oznaka  $g$  predstavlja hvataljku, eng. *gripper*, a  $b$  bazu, eng. *base*) te transformacijom koordinatnog sustava odabranog ugla šahovnice u koordinatni sustav kamere ( ${}^t_cT$ , gdje oznaka  $t$  predstavlja ciljani ugao, eng. *target corner*, a  $c$  kameru, eng. *camera*) da bi se dobila transformacija koordinatnog sustava kamere u koordinatni sustav hvataljke ( ${}^c_gT$ , gdje oznaka  $c$  predstavlja kameru, eng. *camera*, a  $g$  hvataljku, eng. *gripper*). Opisani je postupak korištenja transformacija prikazan na slici 16.



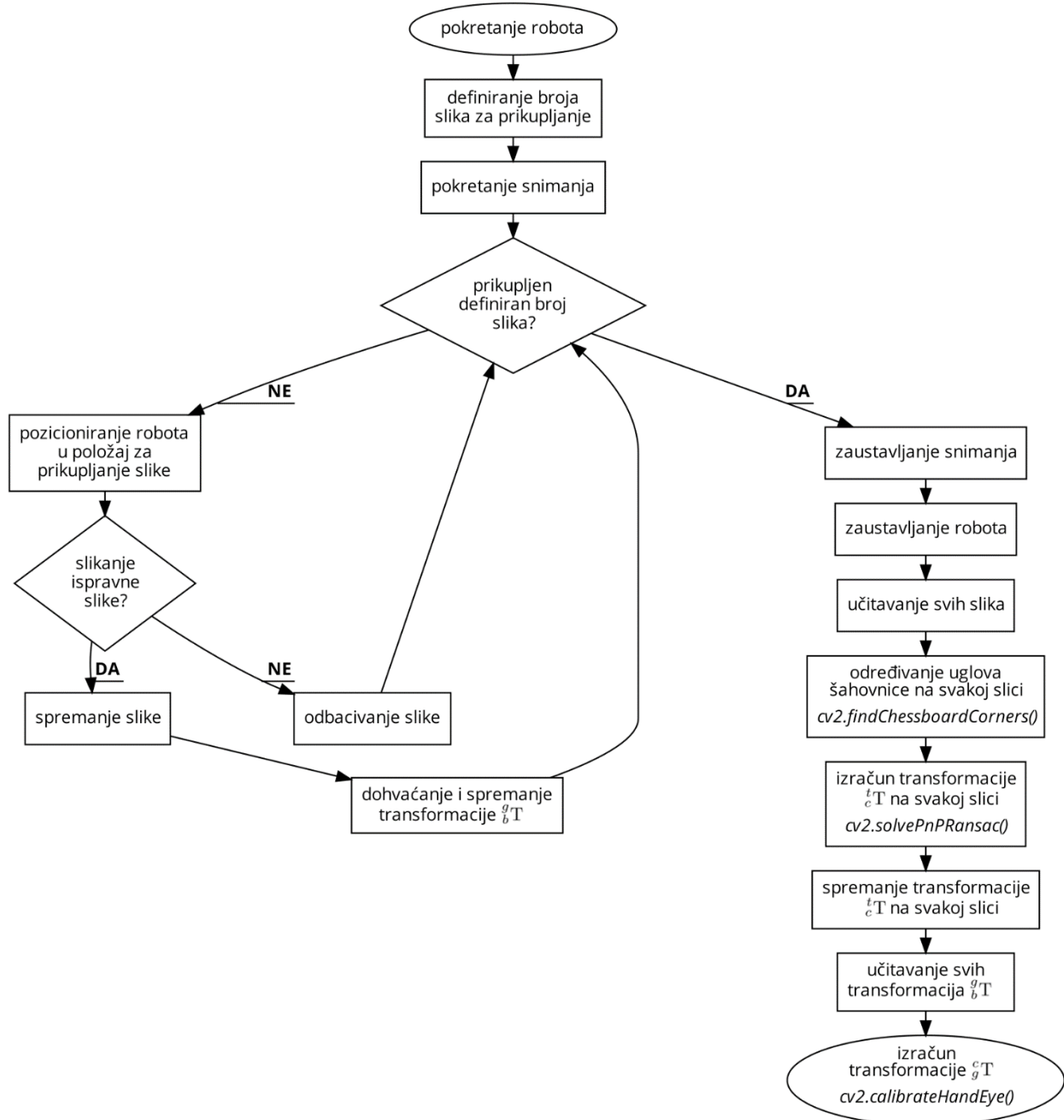
Slika 16. Transformacije *eye-in-hand* kalibracije [12]

Problem se svodi na rješavanje sljedećeg sustava jednačbi s nepoznom matricom homogene transformacije  ${}^c_g T$  za  $n$  slika, gdje su  $i$  i  $j$  proizvoljni i međusobno različiti indeksi navedenih slika:

$${}^g T_i \cdot {}^c_g T \cdot {}^t_c T_i = {}^g T_j \cdot {}^c_g T \cdot {}^t_c T_j, \quad i \neq j \quad (3.2)$$

$$({}^g T_j)^{-1} \cdot {}^g T_i \cdot {}^c_g T = {}^c_g T \cdot {}^t_c T_j \cdot ({}^t_c T_i)^{-1}, \quad i \neq j \quad (3.3)$$

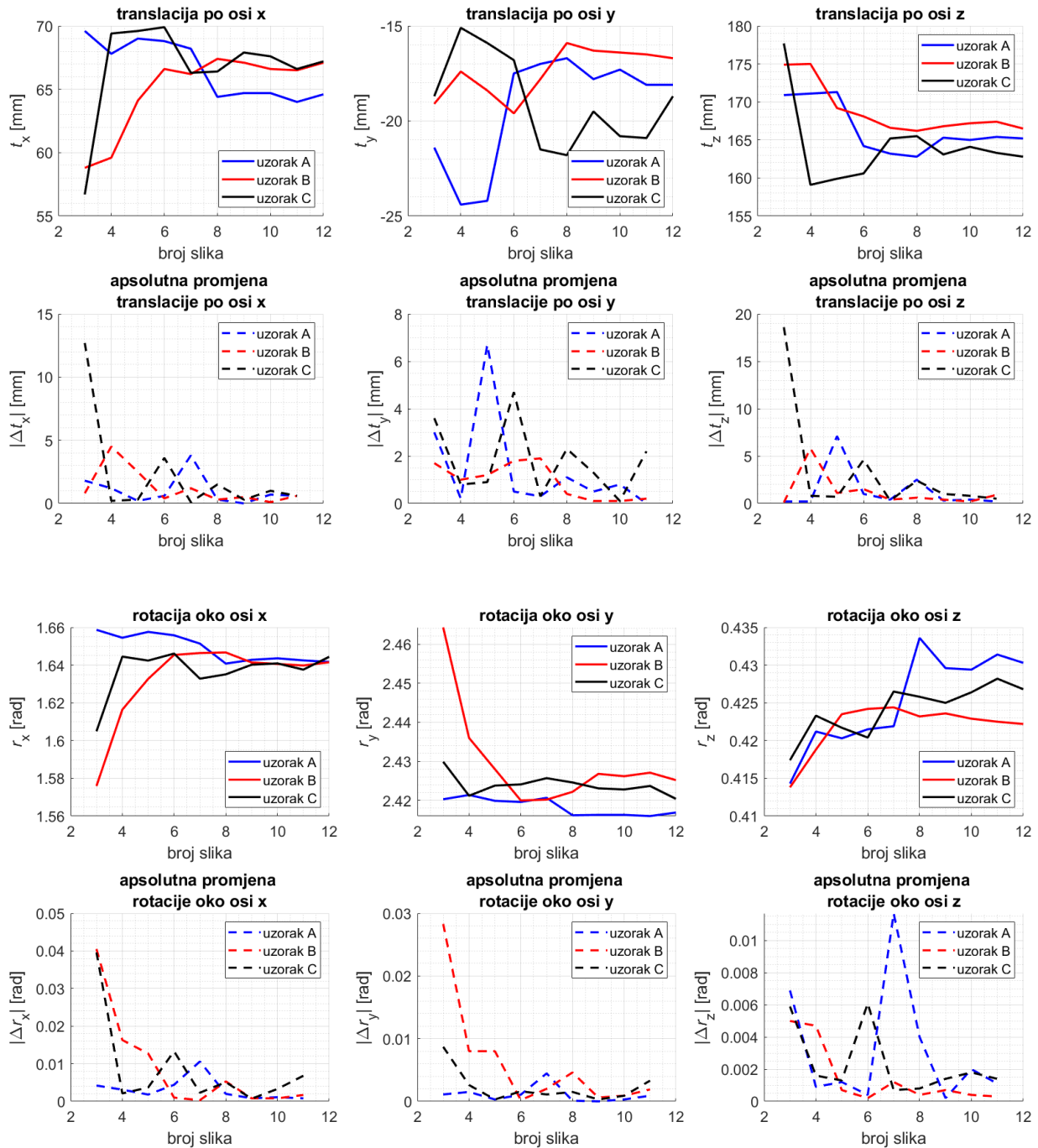
Za rješavanje ovog problema potrebne su barem dvije ispravne slike, čiji se položaj kamere ne razlikuje samo u rotaciji oko koordinatne osi okomite na ravninu šahovnice, no za postizanje veće točnosti, uputno je koristiti barem desetak slika. U ovom se radu za izračunavanje transformacije koordinatnog sustava kamere u koordinatni sustav hvataljke koristi funkcija `cv2.calibrateHandEye()`, čime se pojednostavnjuje implementacija opisanog postupka. Cjelokupni je postupak kalibracije za nepromijenjeni odnos kamere i hvataljke potrebno raditi samo jednom, nakon čega se podaci o dobivenoj transformaciji spremaju za daljnje korištenje. U nastavku je, na slici 17, prikazan dijagram toka opisanog postupka *eye-in-hand* kalibracije.

Slika 17. Dijagram toka *eye-in-hand* kalibracije

### 3.4. Analiza rezultata kalibracije

Kako bi se provjerila ispravnost razvijenog postupka kalibracije, postupak se provodi za nekoliko uzoraka s različitim brojem slika za nepromijenjeni odnos kamere i hvataljke, nakon čega se dobiveni rezultati uspoređuju. Matrice homogenih transformacija se, radi izbjegavanja uspoređivanja velikog broja njihovih elemenata, razdjeljuju na vektore translacije  $t$  i vektore

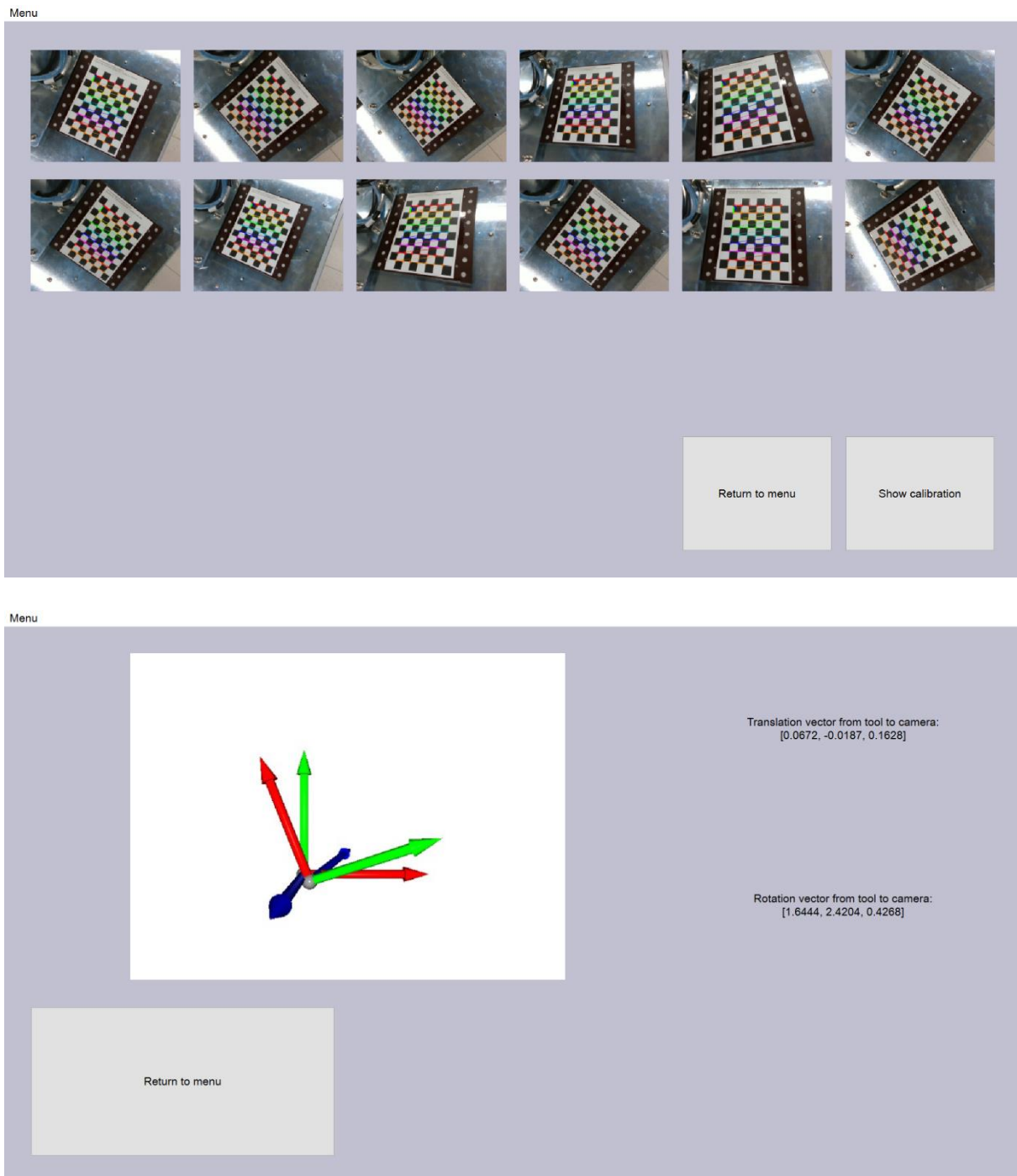
rotacije  $r$ , pri čemu vektori rotacije predstavljaju jednoznačan i kompaktan zapis matrice rotacije  $R$ . U nastavku je, na slici 18, prikazana usporedba navedenih vektora za tri uzorka s odabranim brojem slika u rasponu od tri do dvanaest.



Slika 18. Usporedba kalibriranih vektora translacije i rotacije



Analizom se rezultata utvrđuje da implementirani postupak *eye-in-hand* kalibracije daje stabilne rezultate čija točnost se, iako uz povremene šumove, općenito povećava sa brojem slika pomoću kojih se vrši kalibracija. Šumovi se uglavnom javljaju zbog ograničene ponovljivosti korištenog robota i rezolucije slika korištenog 3D vizijskog sustava. Za cjelokupni se postupak kalibracije 3D vizijskog sustava razvija i grafičko sučelje, čiji je prikaz dan na slici 19.



Slika 19. Razvijeno grafičko sučelje za kalibraciju 3D vizijskog sustava



## 4. ROBOTSKO RUKOVANJE OBJEKTIMA

Robotsko rukovanje objektima vrlo je kompleksan zadatak zbog potrebe za poznavanjem velikog broja podataka o objektima, kao što su njihova geometrija, masa, mehanička svojstva i slično, te još većeg broja podataka o njihovoj okolini. Upravo je to jedan od glavnih razloga zašto je robotsko različitim objektima u nestrukturiranoj okolini rukovanje još uvijek otvoreni problem bez poopćenog rješenja. Kako bi se, za potrebe ovog zadatka, moglo rukovati određenim prehrambenim proizvodima bez eksplicitnog zadavanja njihove pozicije i orijentacije, odlučeno je da se proizvodi najprije detektiraju na slikama neuronskom mrežom, nakon čega se odrede njihovi položaji u prostoru te se isplanira i realizira njihovo hvatanje na temelju geometrije. U ovom se radu za detekciju proizvoda na slikama odabire korištenje neuronske mreže YOLO v5, trenirane na vlastitom skupu označenih slika, koja je sposobna detektirati objekte u realnom vremenu. Kada je riječ o određivanju položaja proizvoda u prostoru, za obradu oblaka točaka se odabire korištenje knjižnice Open3D, dok se za planiranje hvatanja proizvoda odabire postupak temeljen na algoritmu analize glavnih komponenti. Komunikacija između robota, 3D vizijskog sustava i računala se ostvaruje TCP/IP protokolom, a konačno se implementira i grafičko sučelje.

### 4.1. Detekcija objekata

Detekcija proizvoda na temelju opisane neuronske mreže YOLO v5 svodi se na izradu i označavanje skupa slika proizvoda, treniranje mreže te obradu slika tom mrežom. Detekcija je potrebna kako bi sustav mogao razlučiti kojim je proizvodom potrebno rukovati te gdje se on nalazi u prostoru. Neuronska mreža treba davati vjerodostojne rezultate, zbog čega potrebno ju je istrenirati na skupu slika koji je čim sličniji slikama za detekciju, s ispravno podešenim hiperparametrima. U nastavku se opisuje cjelokupni postupak detekcije objekata za odabrani skup proizvoda i analiziraju se njegovi rezultati.

#### 4.1.1. Izrada i označavanje skupa slika

Kako bi se obradio problem što fleksibilnijeg robotskog rukovanja, za ovaj se zadatak odabire skup od dvanaest klasa prehrambenih proizvoda u rasponu dimenzija od 20 do 200 mm,

masa od 60 g do 500 g, različitih geometrijskih oblika i teksture, koji se mogu lako pronaći u trgovinama mješovitom robom. Motivacija za opisani odabir je razmatranje korištenja sustava sličnog razvijenom za mnogo veće skupove proizvoda u distributivnim centrima, uz obraćanje pozornosti na količinu uloženog vremena za prikupljanje odabranog skupa podataka. Odabiru se sljedeće klase prehrambenih proizvoda, prikazane na slici 20, s detaljnim nazivima prevedenim na engleski jezik, u svrhu izbjegavanja moguće višeznačnosti i korištenja dijakritičkih znakova:

- *Barcaffè Classic Original 225g,*
- *Franck Apple Cinnamon Tea 60g,*
- *Franck Black Currant Tea 60g,*
- *Gavrilovic Pork Liver Pate 100g,*
- *Giana Tuna Salad Mexico 185g,*
- *Kotanyi Curry Spices Mix 70g,*
- *Kotanyi Garlic Granules 70g,*
- *Nescafe Classic 225g,*
- *Podravka Luncheon Meat 150g,*
- *Solana Pag Fine Sea Salt 250g,*
- *Vindi Iso Sport Forest Strawberries 500mL,*
- *Vindi Iso Sport Lemon-Grapefruit 500mL.*



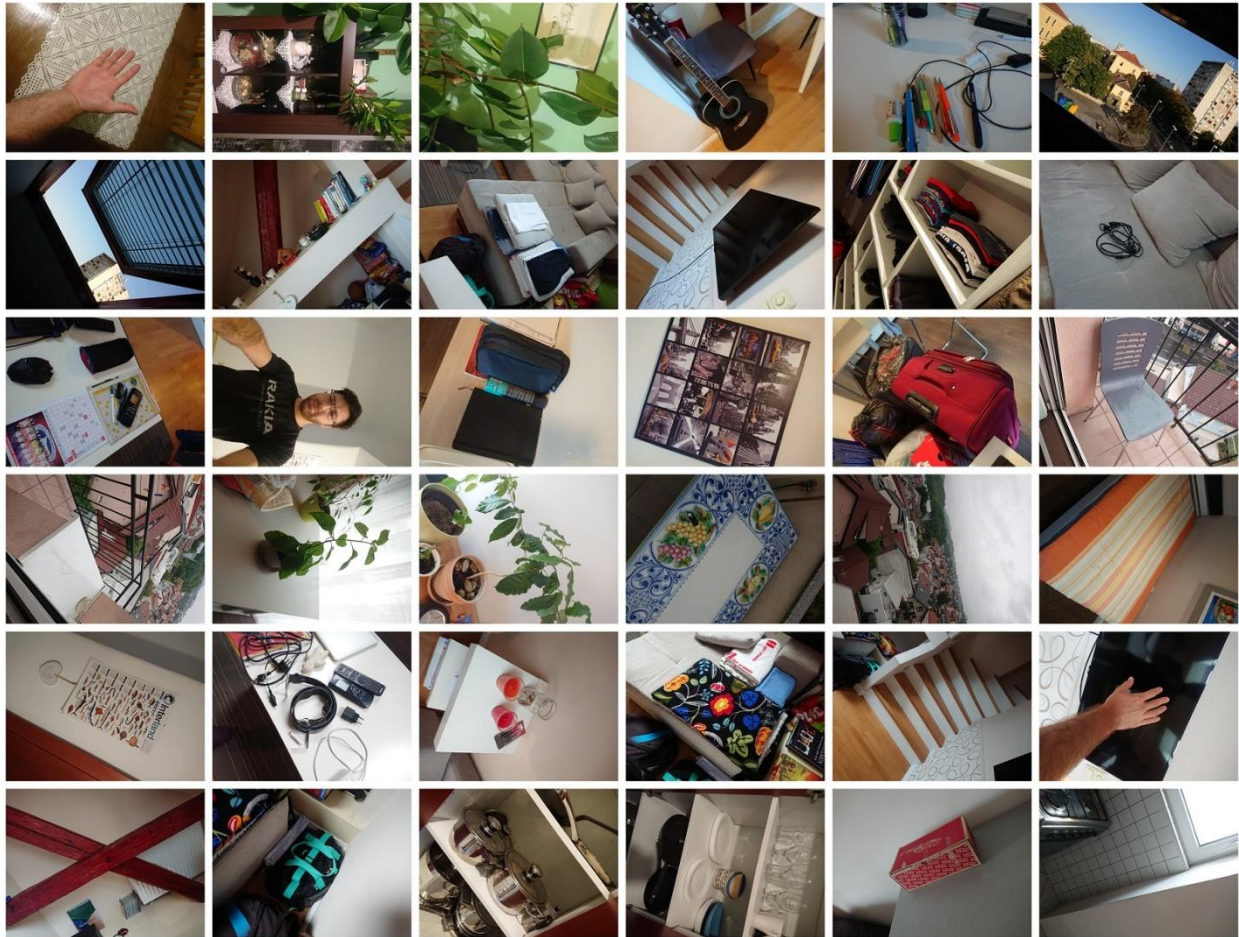
**Slika 20. Odabrane klase prehrambenih proizvoda**

Skup slika odabranih proizvoda se izrađuje tako da se slikaju u onim uvjetima u kojima bi se trebali i detektirati. Opisani postupak uključuje što različitije položaje, udaljenosti, osvjetljenje, kontrast, zamućenje i zaklonjenost proizvoda u njihovoj okolini, s time da treba voditi računa i o tome da se u slici može nalaziti i više proizvoda ili njihovih dijelova. U skup slika je također potrebno uključiti i slike na kojima se ne nalazi nijedan proizvod, kako bi udio lažnih detekcija bio što manji. Za početni je skup slika mobilnim uređajem prikupljeno 100 slika svakog proizvoda, što zajedno čini 1200 slika, 50 slika s više proizvoda te 150 slika bez proizvoda, pri čemu su sve slike rezolucije  $640 \times 480$  piksela. Iako današnji mobilni uređaji i kamere omogućuju slikanje u mnogo većoj rezoluciji, nije se kretalo u tom smjeru jer veća rezolucija zahtijeva više memorije, a time znatno duže treniranje neuronske mreže i vrijeme potrebno za detekciju, dok koristi povećanja rezolucije nakon određenih vrijednosti nisu toliko značajne. Dio skupa prikupljenih slika s više proizvoda i bez proizvoda dan je na slikama 21 i 22.



Slika 21. Dio skupa prikupljenih slika sa više proizvoda





**Slika 22. Dio skupa prikupljenih slika bez proizvoda**

Nakon što je prikupljen cijeli skup slika, na svakoj je slici potrebno označiti postojeće proizvode, tako da se odrede koordinate vrhova vanjskih pravokutnih rubova tih proizvoda. Opisani bi postupak, da se provodi bez ikakvog računalnog alata, bio izrazito vremenski zahtjevan, zbog čega je korišten besplatan program LabelImg razvijen upravo u svrhu označavanja slika. U programu LabelImg se navedene koordinate vrhova određuju intuitivnim pravokutnim obrublivanjem proizvoda klikovima miša te se spremaju u datoteke XML formata, a prilikom postupka označavanja, prikazanog na slici 23, poželjno se voditi sljedećim smjericama za općenite slučajeve: [13]

- proizvodi se označuju kao da su vidljivi u potpunosti, iako su možda dijelom zaklonjeni,
- proizvodi se označuju po svojim vanjskim rubovima ili malo izvan njih, a nikako unutar,
- proizvodi se označuju neovisno o svojoj okolini pa se tako pravokutni rubovi više proizvoda mogu djelomično preklapati.



Slika 23. Označavanje slike prehrambenih proizvoda

#### 4.1.2. Treniranje neuronske mreže

Označavanjem cijelog skupa slika stvaraju se uvjeti za treniranje neuronske mreže YOLO v5, a pošto ona ne prihvaća oznake u datotekama XML formata, oznake se pomoću jednostavne skripte najprije prebacuje u datoteke tekstualnog formata prihvatljivog za korištenu mrežu.

Postupak treniranja neuronske mreže predstavlja proces pronalaženja vrijednosti njenih parametara koji osiguravaju što točniju detekciju proizvoda na još neviđenim slikama. Prvi korak navedenog postupka predstavlja predavanje promiješanog skupa slika za treniranje u serijama (eng. *batch*) mreži s početnim vrijednostima parametara, pri čemu serije moraju biti takvih veličina da računalo ne ostane bez raspoložive radne memorije. Neuronska mreža dobivene slike skalira na zadane dimenzija te pretprocesira na zadani način, mijenjajući im orijentaciju, osvjetljenje, kontrast, zamućenje i slično, kako bi se treniranje obavljalo na što različitijem skupu slika i time se smanjila šansa za pretreniranjem. Pretreniranje mreže je pojam koji predstavlja prilagođavanje iste specifičnom skupu slika. Nakon navedenih radnji, za svaku se seriju slika prema određenom kriteriju određuje pogreška predikcije na temelju trenutnih vrijednosti parametara, a nakon učitavanja posljednje serije slika iz skupa podataka, određuje se ukupna pogreška svih serija. Ta se pogreška u sljedećem koraku predaje određenoj optimizacijskoj funkciji koja prema zadanoj stopi učenja (eng. *learning rate*) mijenja vrijednosti parametara mreže tako da pokušava

minimizirati navedenu pogrešku. Opisani koraci predstavljaju jednu epohu (eng. *epoch*) treniranja neuronske mreže, nakon kojeg slijedi proces validacije.

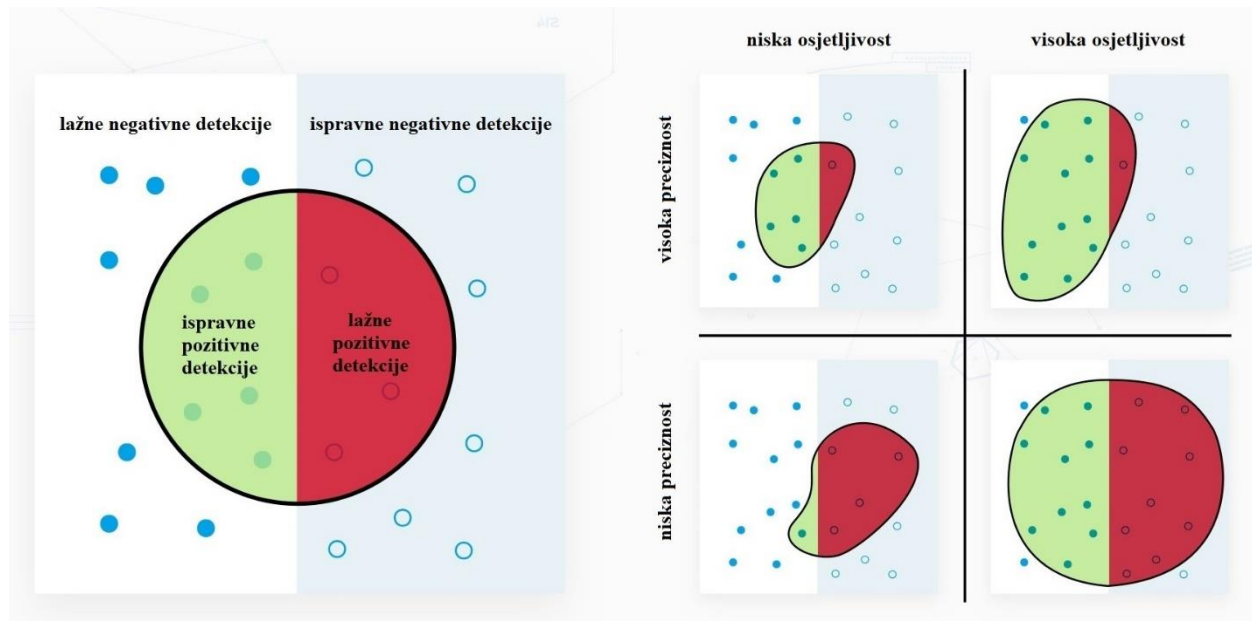
U procesu validacije se mreži s izmijenjenim vrijednostima parametara predaje određeni broj slika te se za njih određuje pogreška, kako bi se mogla pratiti njena promjena tijekom zadanog broja epoha. Korištene slike ne smiju biti sadržane u skupu slika za treniranje jer time validacija ne bi bila nepristrana, zbog čega se koristi zasebni skup slika za validaciju. Nakon završetka posljednje epohe treniranja, neuronska se mreža s konačnim vrijednostima parametara testira, tako da joj se opet predaje određeni broj slika te se za njih određuje pogreška, što se u ovom slučaju obavlja u svrhu praćenja njene promjene tijekom više postupaka treniranja. Korištene slike također ne smiju biti ni u jednom dosad spomenutom skupu slika, zbog čega se koristi zasebni skup slika za testiranje. Broj epoha treniranja se, u kombinaciji sa stopom učenja, treba odabrati dovoljno velik da pogreška treniranja bude što niža, a opet ne toliko velik da bi se mreža pretrenirala. Kako sva tri skupa slika trebaju sadržavati slike istih dimenzija i s oznakama u istom formatu, razumno je cijeli prikupljen skup razdijeliti na tri navedena skupa. Odnosi prema kojima obavlja razdjeljivanje ovise o veličini cijelog prikupljenog skupa pa se, vođenjem smjernicama za općenite slučajeve, kao skup za treniranje iskoristilo 70%, kao skup za validaciju 15%, a kao skup za testiranje 15% cijelog prikupljenog skupa slika. [13]

Treniranje neuronske mreže se, za potrebe ovog zadatka, obavlja u besplatnom *online* razvojnom okruženju Google Colab, kako bi se smanjila potreba za dugotrajnim korištenjem značajne procesorske snage, a i dalje iz održavanje slične brzine treniranja. U svrhu postizanja što boljih rezultata treniranja, potrebno je dobro podesiti iznose stopa učenja, broja epoha, dimenzija slika za treniranje te postavki pretprocesiranja, koji se zajednički nazivaju hiperparametrima mreže. Njihovo je podešavanje obavljeno na temelju iskustva i analize rezultata treniranja više mreža. Pošto su sve slike prikupljenog skupa rezolucije  $640 \times 480$  piksela, mreža će najbolje rezultate davati upravo za tu rezoluciju.

#### **4.1.3. Analiza rezultata treniranja**

Analiza rezultata treniranja obavlja se u besplatnom razvojnom okruženju Weights & Biases, pri čemu se točnost predikcije procjenjuje prema vrijednostima osjetljivosti (eng. *recall*) i preciznosti (eng. *precision*), a pretreniranje mreže odnosom vrijednosti pogreške treniranja i validacije. Osjetljivost predstavlja omjer detekcija ispravnog objekta te zbroja detekcija ispravnog

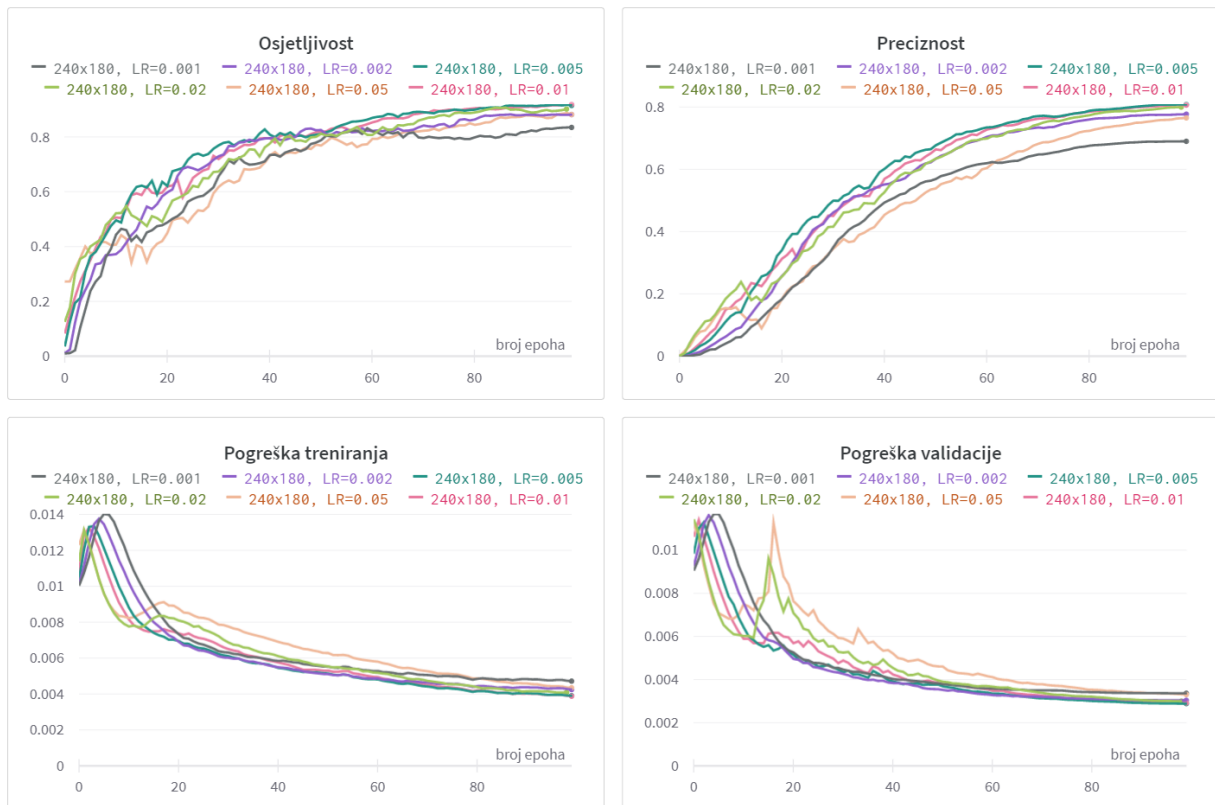
objekta i nečega što nije objekt iz prikupljenog skupa slika, dok preciznost predstavlja omjer detekcija ispravnog objekta te zbroja detekcija ispravnog i neispravnih objekata. Poželjno je da obje veličine imaju što višu vrijednost, a njihovi mogući odnosi su dani na slici 24. Pretreniranje mreže se uočava značajnim porastom pogreške validacije u odnosu na pogrešku treniranja i događa se kod prevelikog broja epoha za određeni skup slika za treniranje, zbog čega treniranje treba zaustaviti na vrijeme. [14]



Slika 24. Odnosi osjetljivosti i preciznosti [14]

Provođenjem detekcija sa svakom od istreniranih mreža danih u nastavku, uočava se da sve mreže mogu obraditi između trideset i pedeset slika u sekundi (eng. *frames per second*), što omogućuje njihovo korištenje u realnom vremenu pa se razlike u brzinama obrade slika neće posebno razmatrati. Pošto se uviđa da početni skup slika od 100 slika svakog proizvoda, što zajedno čini 1200 slika, 50 slika s više proizvoda te 150 slika bez proizvoda ne daje rezultate zadovoljavajuće točnosti predikcije, cijeli skup se učeterostručuje već opisanim postupkom izrade i označavanja. Proširenje skupa slika značajno podiže točnost predikcije, ali i zahtijeva izvjesnu količinu vremena. Kako bi se ubrzalo treniranje za različite kombinacije hiperparametara, koje se mjeri u satima, i pronašao prikladan iznos stope učenja, mreža se trenirana sa slikama rezolucije  $240 \times 180$  piksela, sa 100 epoha i s različitim stopama učenja. Rezultati tih treniranja prikazani su na slici 25, pri čemu su iznosi stopa učenja označeni veličinom LR te se radi veće vjerodostojnosti rezultata koristi proširen skup slika.





**Slika 25. Treniranje neuronske mreže sa različitim stopama učenja**

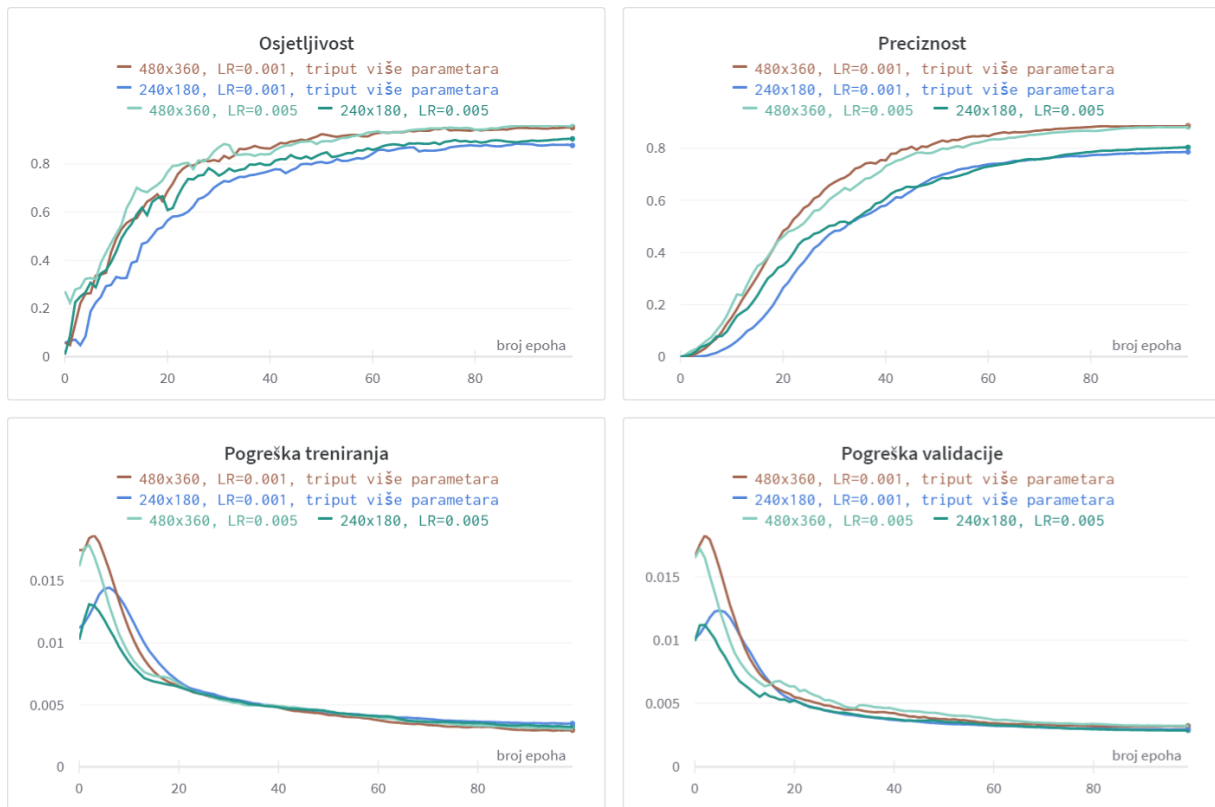
S prethodne je slike vidljivo da se najbolji rezultati treniranja postižu sa stopom učenja iznosa 0,005, pošto ostale vrijednosti stopa učenja rezultiraju i nižom osjetljivošću i nižom preciznošću. Također je vidljivo da za 100 epoha nije došlo do pretreniranja ni u jednom slučaju te da se daljnjim porastom broja epoha ne bi postigla ni značajno viša osjetljivost ni preciznost. Zbog toga će svako sljedeće treniranje izvršavati upravo s navedenim brojem epoha, što za slike rezolucije  $240 \times 180$  piksela i prošireni skup slika u razvojnom okruženju Google Colab traje oko dva sata. Kako bi se prikazao značaj veličine skupa slika za točnost predikcije, na slici 26 su prikazani rezultati treniranja s početnim i proširenim skupom slika, koji u svojem nazivu nema posebnu oznaku. Iako bi se daljnjim povećanjem skupa slika dobili još bolji rezultati, upitno je koliko bi to poboljšanje bilo značajno te se zbog izrazite vremenske zahtjevnosti postupka prikupljanja i označavanja velikog broja slika, za potrebe ovog zadatka, zadržava postojeći prošireni skup.





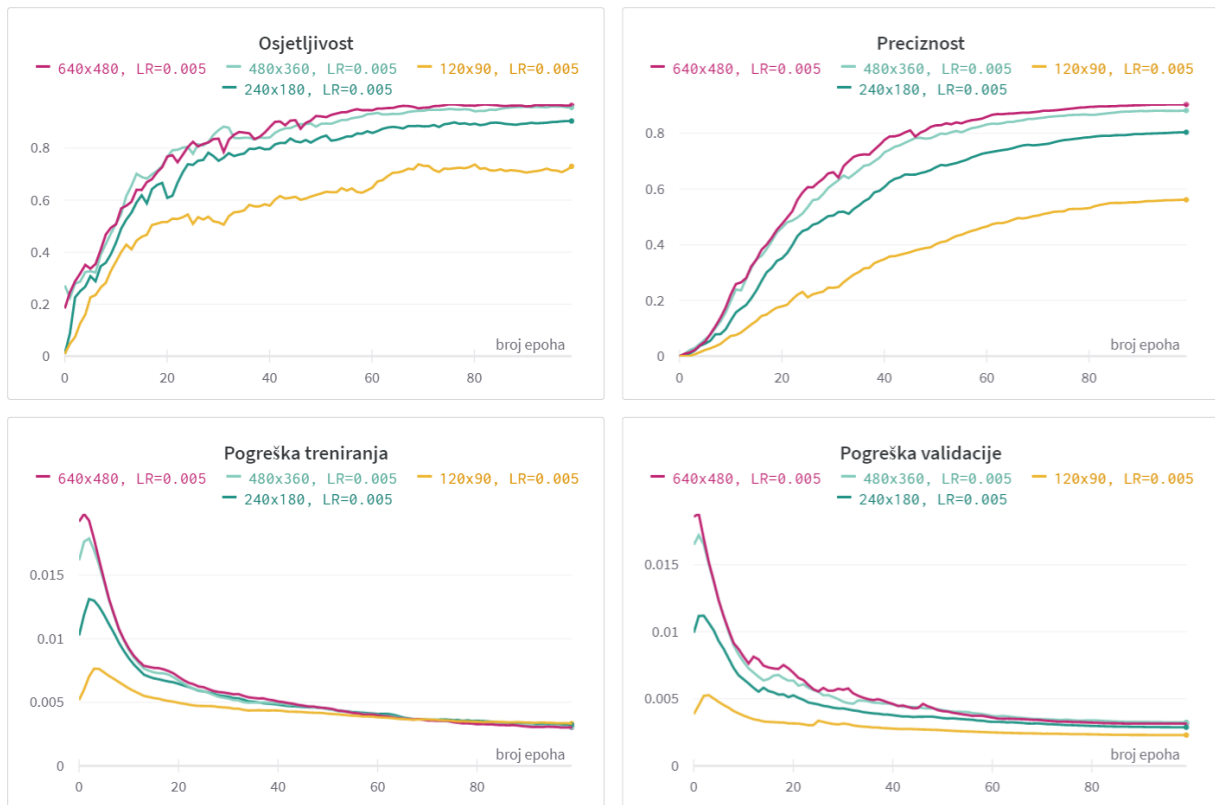
**Slika 26. Treniranje neuronske mreže sa različitim veličinama skupova slika**

Neuronsku je mrežu YOLO v5 moguće trenirati s nekoliko različitih arhitektura koje se razlikuju u broju parametara. Broj parametara je proporcionalan s točnošću predikcije, a obrnuto proporcionalan sa brzinom detekcije, pri čemu se ovdje govori o milijunima parametara. Za potrebe ovog rada, odabire se arhitektura s najmanjim brojem parametara, oko sedam milijuna te se sve prethodne slike treniranja odnose na nju. Kako bi se prikazalo da se odabirom arhitekture s većim brojem parametara ne postiže značajno veća točnost predikcije, na slici 27 su prikazani rezultati treniranja za slike različitih rezolucija i arhitekture mreže s oko sedam i oko dvadeset jedan milijuna parametara, pri čemu ona odabrana nema posebnu oznaku.



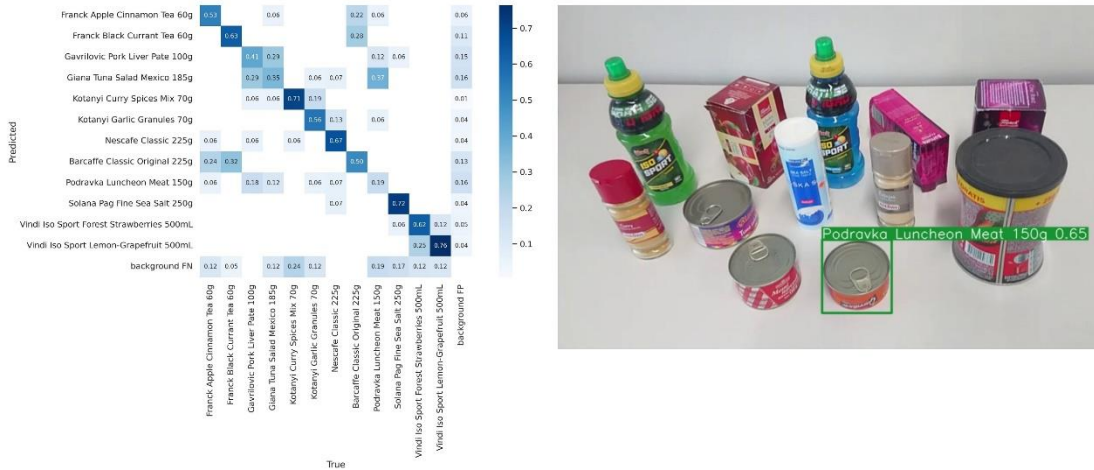
**Slika 27. Treniranje neuronske mreže sa različitim arhitekturama**

Nakon što su odabrani broj epoha, stopa učenja i arhitektura mreže, ostaje još odabrati rezolucije slika na kojima će mreža provoditi detekciju. Pošto su sve slike prikupljenog skupa rezolucije  $640 \times 480$  piksela, za očekivati je da će mreža najbolje rezultate davati upravo za tu rezoluciju. Treniranje je za različite kombinacije hiperparametara obavljano na nižim rezolucijama jer je brzina istog obrnuto proporcionalna sa brojem piksela slika pa tako dvostruko povećanje dimenzija slike uzrokuje četverostruko duže vrijeme treniranja, koje već za početni slučaj u razvojnom okruženju Google Colab iznosi oko dva sata. Kako bi se prikazao značaj rezolucije slika za točnost predikcije, na slici 28 su prikazani rezultati treniranja za slike različitih rezolucija. Daljnjim povećanjem iznad rezolucije prikupljenog skupa slika ne bi se postiglo nikakvo poboljšanje jer se ne bi mogle izdvojiti nikakve nove značajke, zbog čega se kao najviša rezolucija odabire upravo rezolucija prikupljenog skupa slika.

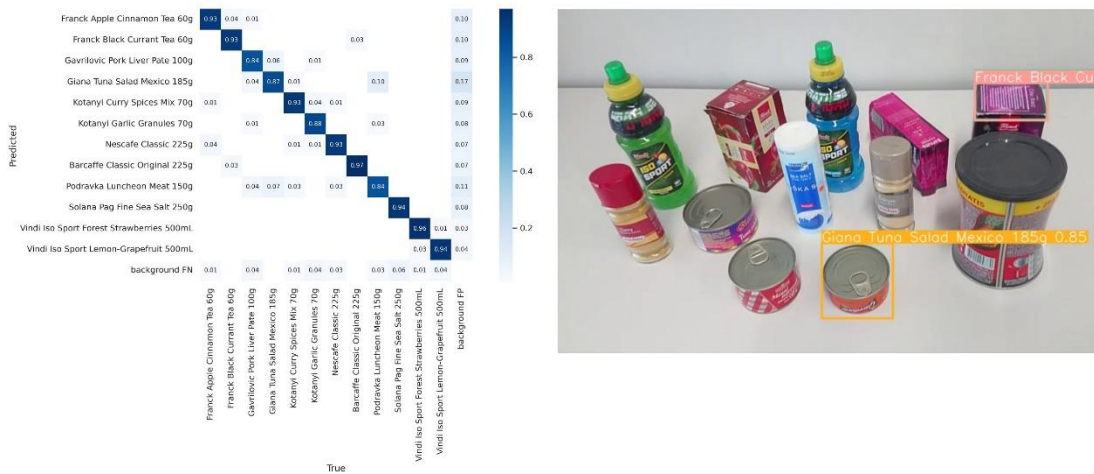


**Slika 28. Treniranje neuronske mreže sa različitim rezolucijama slika**

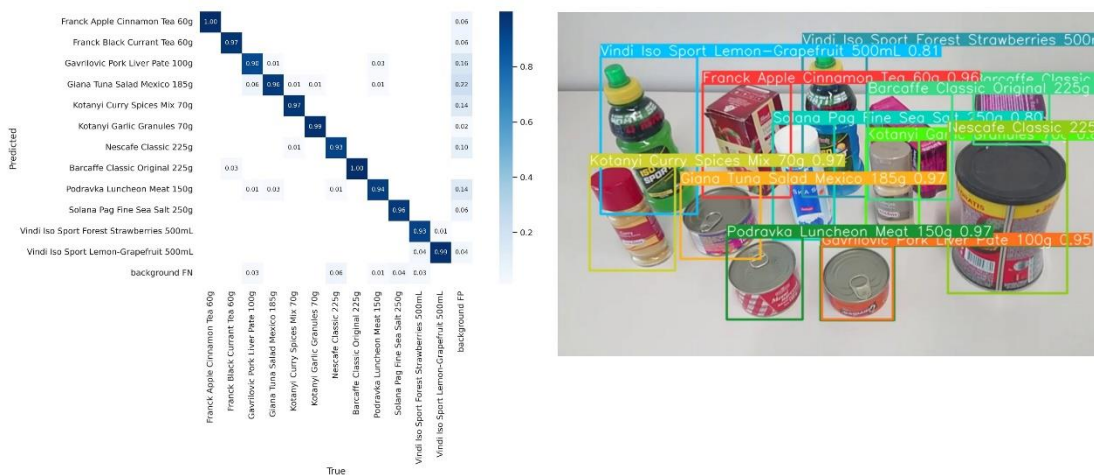
Iz dobivenih se rezultata zaključuje da je, za potrebe ovog zadatka i korištenog proširenog skupa slika, najbolje koristiti neuronsku mrežu YOLO v5 treniranu sa 100 epoha, stopom učenja iznosa 0,005, arhitekture s najmanjim brojem parametara te slika rezolucija  $640 \times 480$  piksela, pri čemu su rezultati za tu rezoluciju samo neznatno bolji od onih za rezoluciju  $480 \times 360$  piksela. Kako bi se navedeni zaključak i potvrdio, na slikama od 29 do 32 su dane matrice konfuzije i detekcije odabranih klasa prehrambenih proizvoda s minimalnom pouzdanošću od 50% redom za treniranje sa slikama početnog skupa rezolucija  $240 \times 180$  piksela te slikama proširenog skupa rezolucija  $240 \times 180$  piksela,  $480 \times 360$  piksela i  $640 \times 480$  piksela. Matrice konfuzije prikazuju iznos osjetljivosti i preciznosti pojedinog proizvoda iz prikupljenog skupa slika te tako daju detaljniji uvid u osjetljivost i preciznost proizvoda od grafova s prethodnih slika.



Slika 29. Matrica konfuzije i detekcija proizvoda za početni skup rezolucije 240 × 180

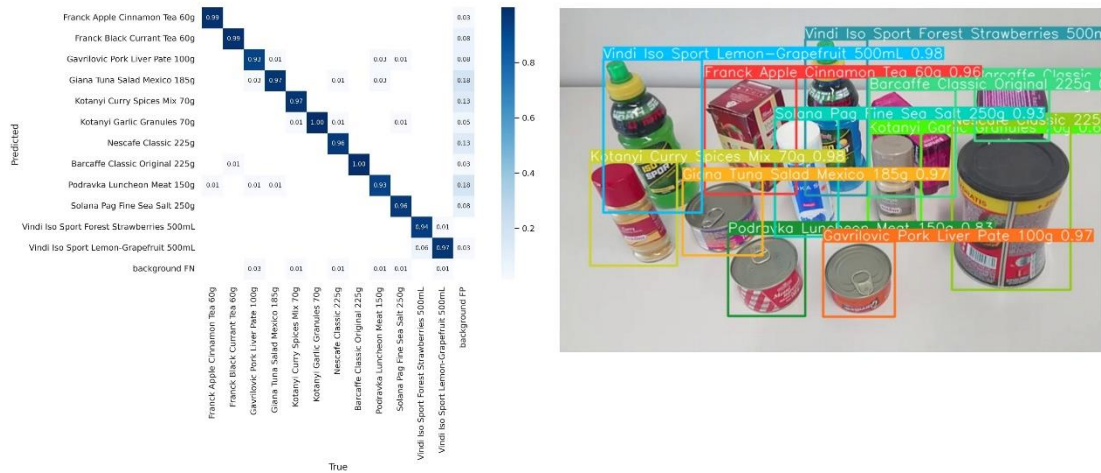


Slika 30. Matrica konfuzije i detekcija proizvoda za prošireni skup rezolucije 240 × 180



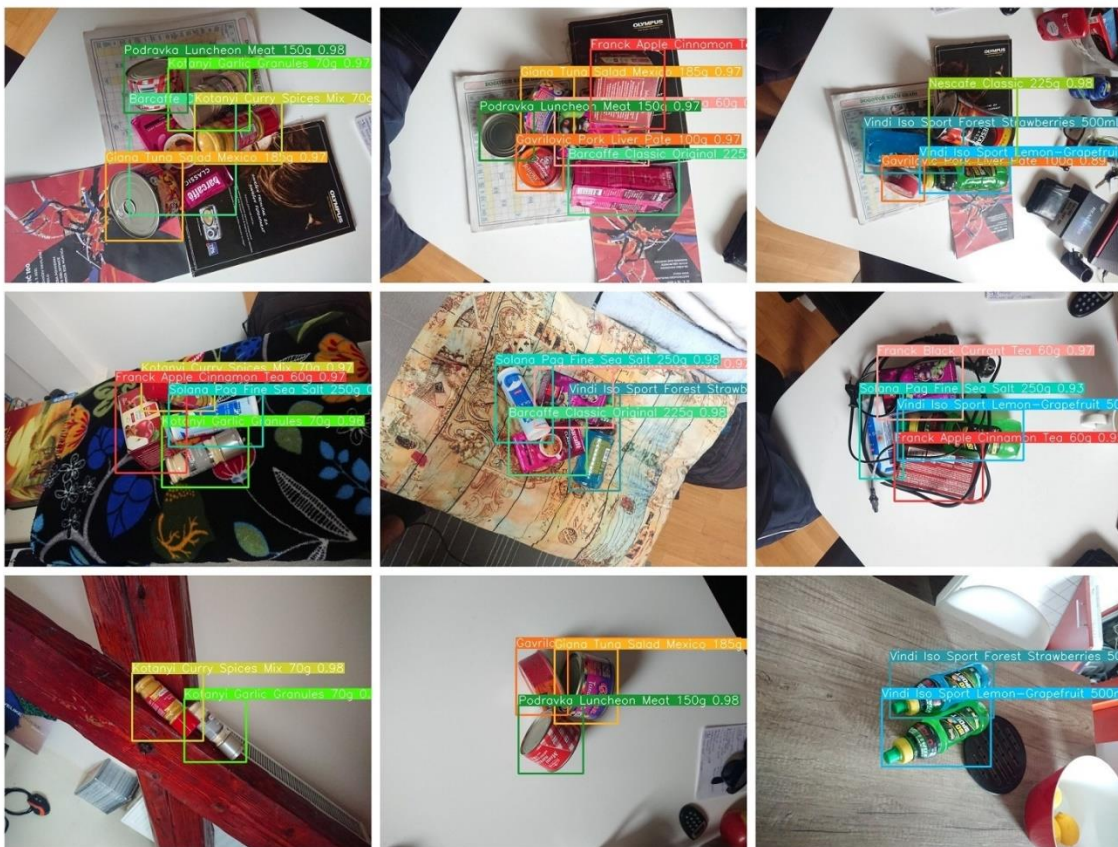
Slika 31. Matrica konfuzije i detekcija proizvoda za prošireni skup rezolucije 480 × 360





Slika 32. Matrica konfuzije i detekcija proizvoda za prošireni skup rezolucije 640 × 480

Prethodnim se slikama utvrđuje da odabrana neuronska mreža trenirana sa proširenim skupom slika rezolucija 640 × 480 piksela daje vrlo zadovoljavajuće rezultate detekcije u realnom vremenu, i to u slučajevima različitih položaja i zaklonjenosti proizvoda, što sugerira njenu visoku robusnost. Na slici 33 prikazane su detekcije više proizvoda koje potvrđuju te zaključke.



Slika 33. Detekcije više prehrambenih proizvoda

## 4.2. Prostorna lokalizacija objekata

Prostorna lokalizacija objekata je postupak kojem se može pristupiti tek kada se poznaje položaj 3D vizijskog sustava u globalnom koordinatnom sustavu, kako bi se mogao dobiti vjerni prikaz cijele scene, te kada se na sceni mogu detektirati odabrani objekti. Za potrebe je ovog zadatka prvi preduvjet ispunjen postupkom *eye-in-hand* kalibracije, a drugi treniranjem neuronske mreže YOLO v5. Postupak prostorne lokalizacije objekata se svodi na obradu oblaka točaka dobivenih kalibriranim 3D vizijskim sustavom uz izoliranje odabranih objekata neuronskom mrežom.

Kako bi se mogle obavljati razne operacije nad oblacima točaka, koje je moguće poistovjetiti sa strukturiranim trodimenzionalnim podacima, najprije je te podatke potrebno prikupiti 3D vizijskim sustavom i spremi ih u određenom formatu. Navedeni se postupak provodi prikupljanjem slika u boji i dubinskih slika objekata, prikazanih na slici 34, koje se zatim korištenjem funkcije `open3d.geometry.RGBDImage.create_from_color_and_depth()` knjižnice Open3D spremaju kao oblaci točaka u PLY formatu. Navedeni format omogućuje pohranu točaka u prostoru, njihovih boja te također njihovih normala.



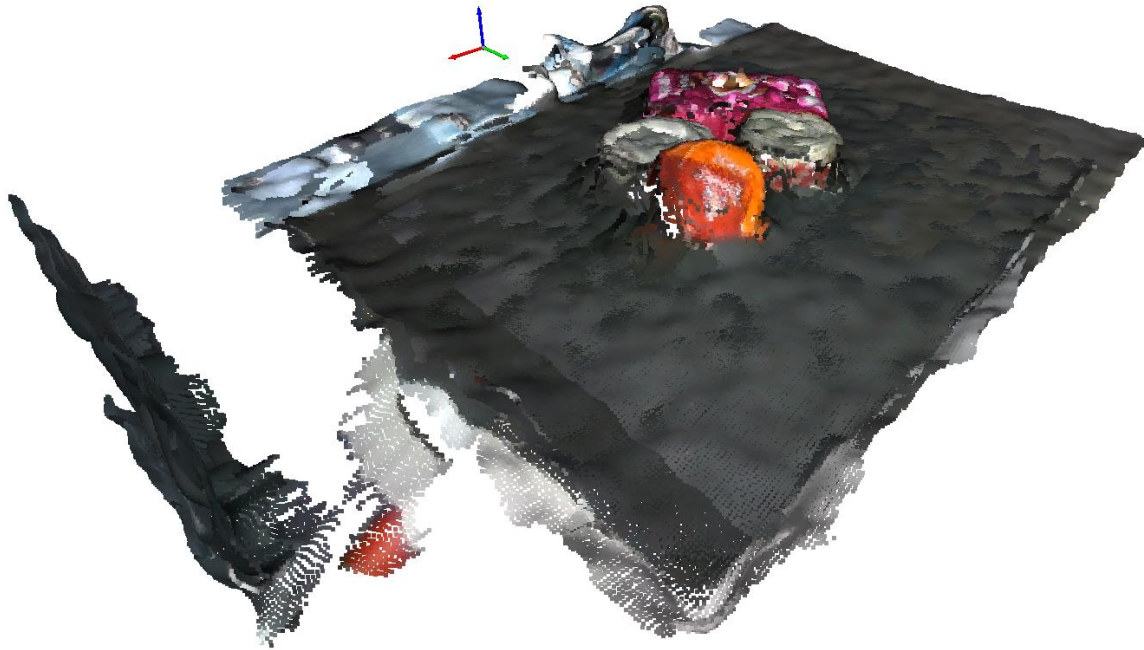
Slika 34. Slika u boji i dubinska slika objekata

Kada se iz položaja kamere u njenom lokalnom koordinatnom sustavu prikupi određen broj oblaka točaka, od kojih je jedan prikazan na slici 35, obavlja se njihovo pretprocesiranje u svrhu niže zahtjevnosti obrade i veće informativnosti podataka. Pretprocesiranje se sastoji od uzorkovanja prema uniformnoj gustoći funkcijom `open3d.geometry.PointCloud.voxel_down_sample()` te filtriranju statistički nepripadajućih točaka funkcijom `open3d.geometry.PointCloud.remove_statistical_outlier()`.



**Slika 35. Oblak točaka u koordinatnom sustavu kamere**

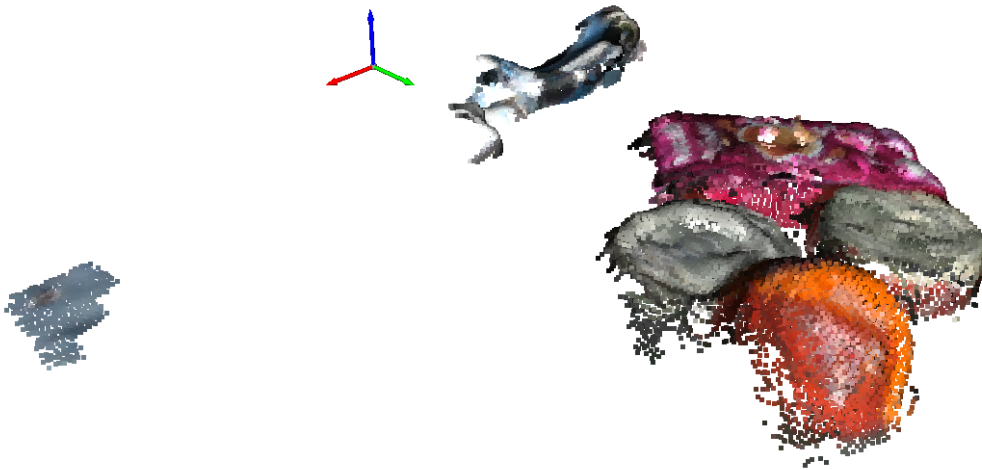
Nakon pretprocesiranja se pristupa vezivanju oblaka točaka uz zajednički globalni koordinatni sustav i njihovom spajanju u oblak točaka cijele scene. Za potrebe ovog zadatka je kao globalni koordinatni sustav odabran koordinatni sustav baze robota pa je na svaki oblak točaka potrebno primijeniti transformaciju koordinatnog sustava kamere u koordinatni sustav baze robota ( ${}^c_bT$ , gdje oznaka  $c$  predstavlja kameru, eng. *camera*, a  $b$  bazu, eng. *base*). Ta je transformacija u svakom trenutku poznata jer je sačinjena od transformacije koordinatnog sustava kamere u koordinatni sustav hvataljke ( ${}^c_gT$ ), koja je dobivena postupkom *eye-in-hand* kalibracije, te transformacije koordinatnog sustava hvataljke u koordinatni sustav baze robota ( ${}^g_bT$ ), koja se s robota može prikupljati u realnom vremenu. Po završetku vezivanja svih oblaka točaka uz globalni koordinatni sustav i spajanja u oblak točaka scene, što je prikazano na slici 36, transformacije korištenih oblaka točaka bi se još mogle fino podesiti nekim od dostupnih algoritama registracije. Uz dobru kalibraciju kamere nije potrebno raditi registraciju oblaka točaka, štoviše, ona može dovesti i do lošijih rezultata, pa se tako neće provoditi ni za potrebe ovog zadatka.



**Slika 36. Oblak točaka scene u koordinatnom sustavu baze robota**

Kako su svi objekti u ovom zadatku postavljeni iznad ishodišta koordinatnog sustava baze robota, oblaku točaka scene se algoritmom temeljenim na funkciji `open3d.geometry.PointCloud.select_by_index()` brišu sve točke ispod navedenog ishodišta. Dobiveni se oblak točaka dalje pretprocesira prema prije opisanom algoritmu, pošto se spajanjem više njegovih dijelova izgubila uniformna gustoća, nakon čega slijedi njegovo filtriranje prema algoritmu prostornog grupiranja podataka sa šumom na temelju gustoće (eng. *Density-based spatial clustering of applications with noise – DBSCAN*). Navedeni je algoritam implementiran funkcijom `open3d.geometry.PointCloud.cluster_dbscan()` te se zasniva na grupiranju usko povezanih točaka (eng. *clusters*). Uzimanjem u obzir samo grupa točaka čije su maksimalne dimenzije vanjskih rubova u zadanom rasponu te brisanjem točaka koje predstavljaju šum, dobiva se filtrirani oblak točaka scene, prikazan na slici 37, na temelju kojeg se zatim pristupa izoliranju odabranih objekata.

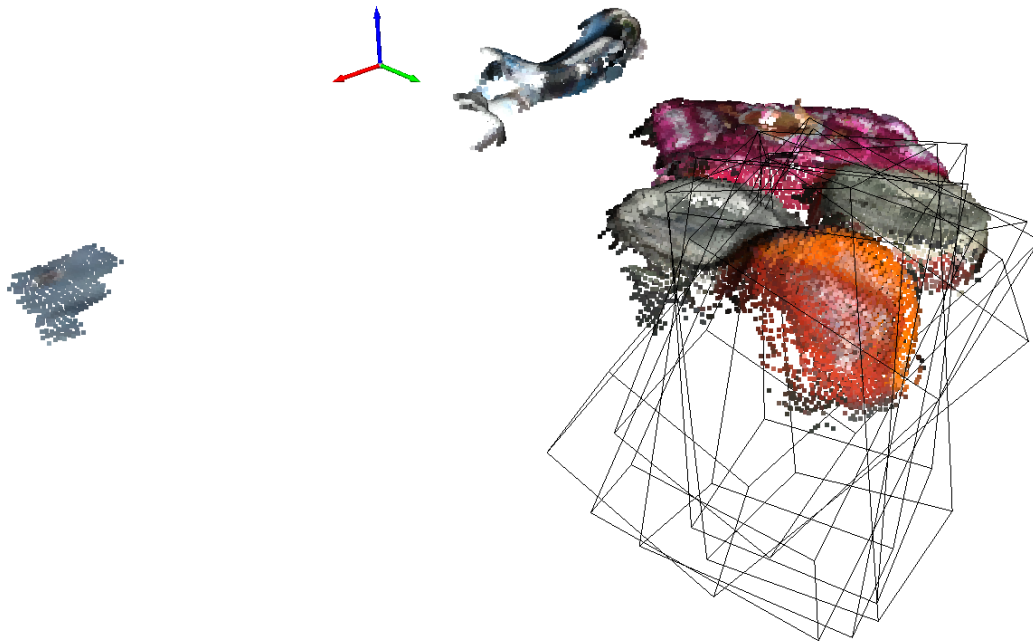




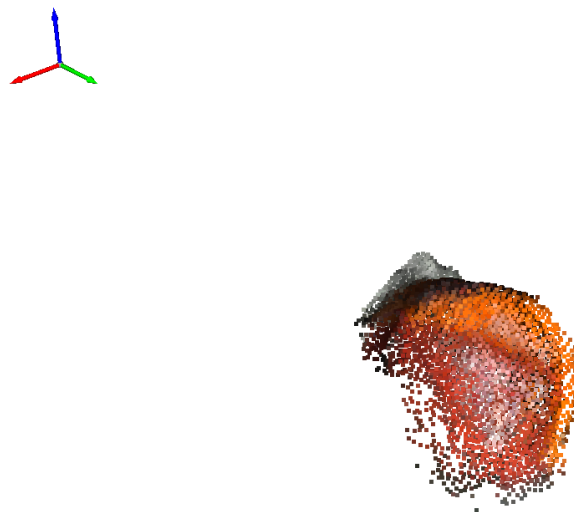
**Slika 37. Filtrirani oblak točaka u koordinatnom sustavu baze robota**

Polazište za izoliranje odabranih objekata je detektiranje svih prehrambenih proizvoda neuronskom mrežom na slikama u boji, nakon čega se odabire specifičan prehrambeni proizvod kojeg će se izolirati. Kako su detekcijom nekog proizvoda na slikama u boji određeni vrhovi njegovih vanjskih pravokutnih rubova, na istim bi se koordinatama u dubinskoj slici mogli prikupiti podaci o dubini tih vrhova u odnosu na koordinatni sustav kamere. Realnost je drukčija te kamera na nekim koordinatama slike ne može prikupiti podatke o dubini, jer na proces prikupljanja podataka utječu nepodudaranje slike u boji i dubinske slike te razni vanjski čimbenici. Iz tog se razloga podaci o dubini vrhova vanjskih pravokutnih rubova odabranog proizvoda izračunavaju kao medijan njihovih okolnih vrijednosti, čime se povećava robusnost izoliranja proizvoda.

Kako se za opisane vrhove poznaju koordinate na slici i stvarne dubine, projiciranjem tih koordinata u prostor se dobivaju stvarne trodimenzionalne koordinate navedenih vrhova. Spajanjem sva četiri vrha vanjskih pravokutnih rubova odabranog proizvoda u prostoru, tvori se dvodimenzionalni okvir koji propagacijom po osi svoje normale dobiva treću dimenziju te se može koristiti za interakciju s filtriranim oblakom točaka scene, što je prikazano na slici 38. Određivanjem opisanih trodimenzionalnih okvira za svaku detekciju odabranog proizvoda, pokušavaju se što bolje definirati njegovi vanjski prostorni rubovi. Ti se rubovi zatim koriste za brisanje svih točaka koje nisu sadržane unutar njihovog presjeka, čime se dobiva oblak točaka ugrubo izoliranog proizvoda. Kako bi se dobio oblak točaka fino izoliranog proizvoda, prikazan na slici 39, prethodno se dobiveni oblak točaka dodatno filtrira prema opisanom algoritmu prostornog grupiranja podataka sa šumom na temelju gustoće, čime se postupak prostorne lokalizacije odabranog proizvoda završava.



**Slika 38. Izrezivanje oblaka točaka u koordinatnom sustavu baze robota**



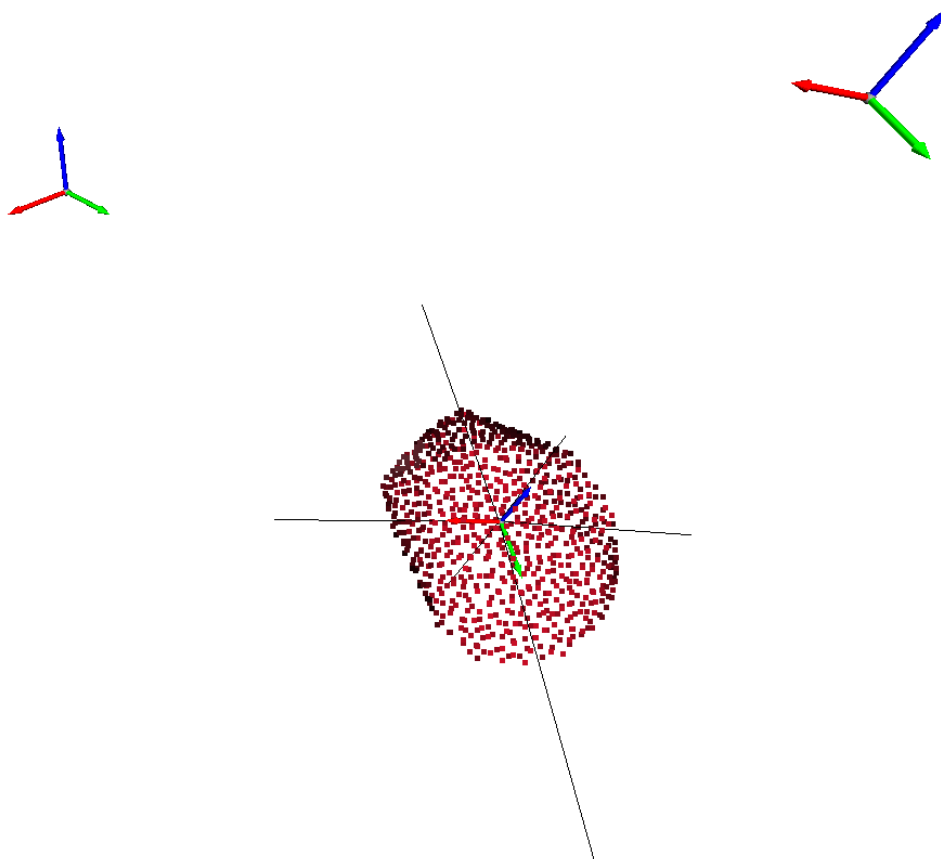
**Slika 39. Izolirani oblak točaka u koordinatnom sustavu baze robota**

### 4.3. Planiranje robotskog hvatanja

Planiranje hvatanja proizvoda svodi se na postupak obrade njihovih podataka, s ciljem dobivanja njihovih pozicija težišta i glavnih osi inercije, a sve kako bi se s proizvodima mogla obavljati fizička interakcija. Problem je moguće pojednostaviti pretpostavljanjem homogenosti proizvoda, što je za odabrani skup proizvoda u ovom zadatku te za mnoge druge realne primjene sasvim opravdana radnja. Pod navedenom se pretpostavkom pozicije težišta proizvoda poistovjećuju s pozicijama njihovih geometrijskih središta, a pozitivni smjerovi glavnih osi inercije s vektorima njihove orijentaciju. Tako postavljenom problemu može se pristupiti na razne načine, a u nastavku je, zbog svoje jednostavnosti, dan postupak temeljen na algoritmu analize glavnih komponenti (eng. *Principal component analysis – PCA*).

Određivanje pozicije geometrijskog središta i orijentacije odabranog proizvoda pripada domeni obrade trodimenzionalnih podataka, što je moguće poistovjetiti s obradom podataka u obliku oblaka točaka. Na taj se način problemu planiranja hvatanja proizvoda može pristupiti postupkom temeljenim na algoritmu analize glavnih komponenti, a upravo se on i koristi u ovom radu. Analiza glavnih komponenti je postupak kojim se podaci iz zadanog skupa preslikavaju na određeni broj linearno nezavisnih vektora tako da im se maksimizira varijabilnost. Navedeni se linearno nezavisni vektori određuju u višedimenzionalnom prostoru zadanog broja dimenzija i nazivaju se glavnim komponentama. Postupak se može shvatiti i kao redukcija dimenzija zadanog skupa podataka, tako da se dobije određeni broj glavnih komponenti koje maksimiziraju varijabilnost podataka i međusobno ne koreliraju. Glavne se komponente mogu poistovjetiti sa svojstvenim vektorima kovarijacijske matrice podataka te se definiraju tako da prva glavna komponenta sadrži najveću varijabilnost podataka u jednom smjeru, a svaka sljedeća onu preostalu u smjeru linearno nezavisnom o svim prethodnima. Analizu glavnih komponenti je tako za podatke u obliku oblaka točaka smisleno provoditi u tri dimenzije, čime glavne komponente predstavljaju ili pozitivne ili negativne smjerove vektora orijentacije proizvoda, a njihovo sjecište geometrijsko središte proizvoda. [15]

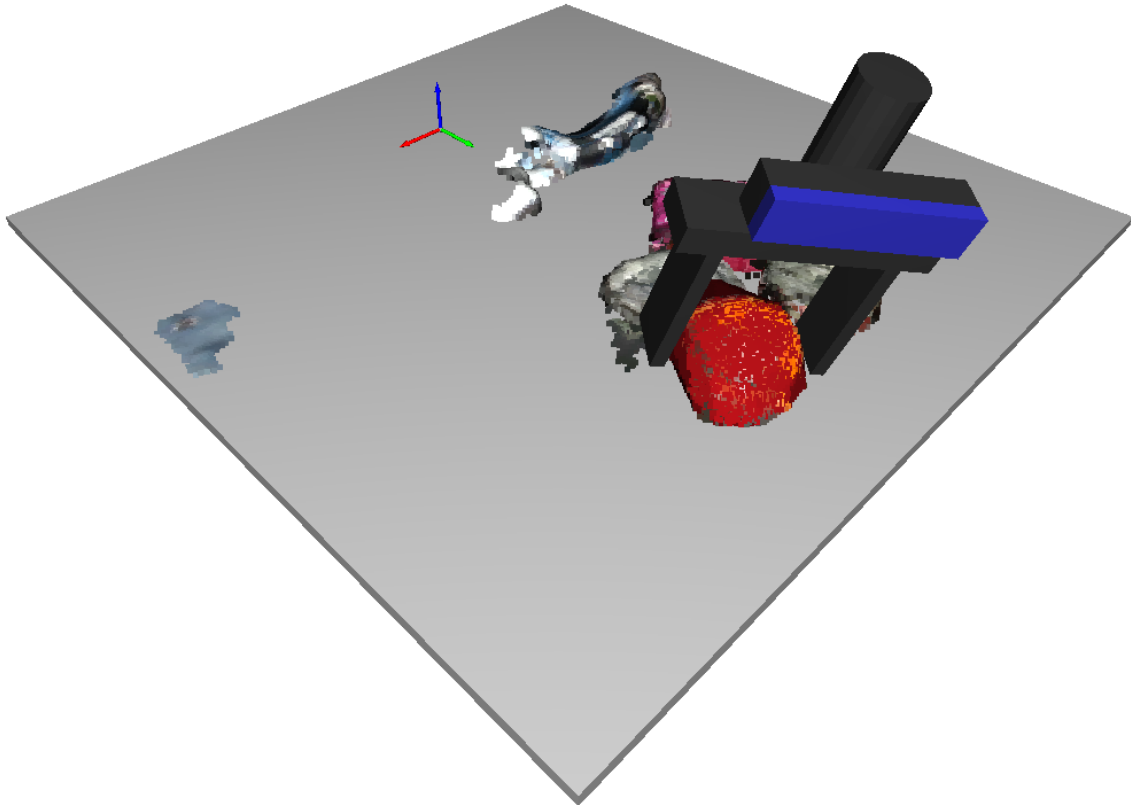
Opisani se pristup koristi i za potrebe ovog zadatka, no kako bi se uzela u obzir samo geometrija proizvoda, plašt oblaka točaka proizvoda se najprije treba uniformno uzorkovati. U tu se svrhu određuje konveksna ljuska (eng. *convex hull*) oblaka točaka, drugim riječima konveksni plašt, funkcijom `open3d.geometry.PointCloud.compute_convex_hull()` te se ista ljuska uniformno uzorkuje funkcijom `open3d.geometry.TriangleMesh.sample_points_poisson_disk()`. Uniformno uzorkovana konveksna ljuska oblaka točaka sa glavnim komponentama i položajima za hvatanje, koji će se definirati u nastavku, prikazana je na slici 40.



**Slika 40. Uzorkovana konveksna ljuska oblaka točaka sa glavnim komponentama i položajima za hvatanje**

Kako se analizom glavnih komponenti ne mogu razlikovati pozitivni i negativni smjerovi vektora orijentacije proizvoda, potrebno je odrediti one od značaja za hvatanje proizvoda, pri čemu se polazi od vektora usmjerenog prema strani prilaska hvataljke. Taj se vektor definira kao onaj koji zatvara najmanji kut s pozitivnim smjerom  $z$ -osi globalnog koordinatnog sustava, a određuje se pomoću skalarnog umnoška. Nakon prethodnog vektora, potrebno je odrediti i onaj kolinearano s osi zatvaranja hvataljke, a usmjeren prema zadanoj strani hvataljke. Eliminacijom vektora usmjerenog prema točki prilaza hvataljke i vektora suprotno usmjerenog od navedenog, za odabir ostaju četiri vektora, od kojih su po dva suprotno usmjerena. Izračunavanjem dimenzija proizvoda po osima definiranim glavnim komponentama, u užu izbor ulaze ona dva suprotno usmjerena vektora za koje je potrebna manja širina hvatanja, a od njih se konačno odabire onaj koji zatvara najmanji kut sa zadanom orijentacijom hvataljke u tom trenutku. Kako bi se definirao koordinatni sustav položaja za hvatanje proizvoda, uz dva se već odabrana vektora, treći određuje kao njihov vektorski umnožak, a ishodište se postavlja u geometrijsko središte proizvoda dobiveno analizom glavnih komponenti. U svrhu kontroliranog gibanja hvataljke do položaja za hvatanje i iz njega,

definira se i koordinatni sustav položaja prilaska, translaticiran od koordinatnog sustava položaja za hvatanje za zadanu udaljenost po vektoru usmjerenom prema strani prilaska hvataljke. Definiranjem navedenih položaja prilaska i položaja za hvatanje, završava se postupak planiranja robotskog hvatanja, a njegova je vizualizacija prikazana na slici 41.

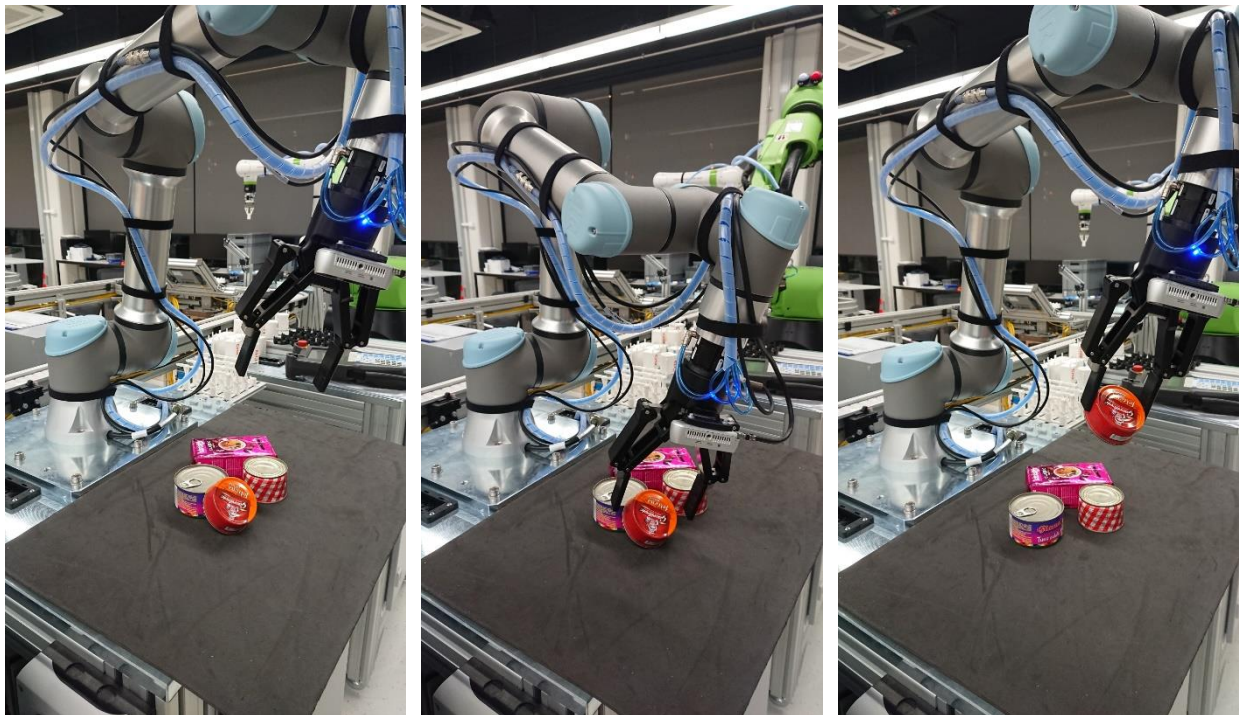


**Slika 41. Filtrirani oblak točaka sa vizualizacijom hvatanja proizvoda**

### **4.3. Komunikacija i robotsko hvatanje**

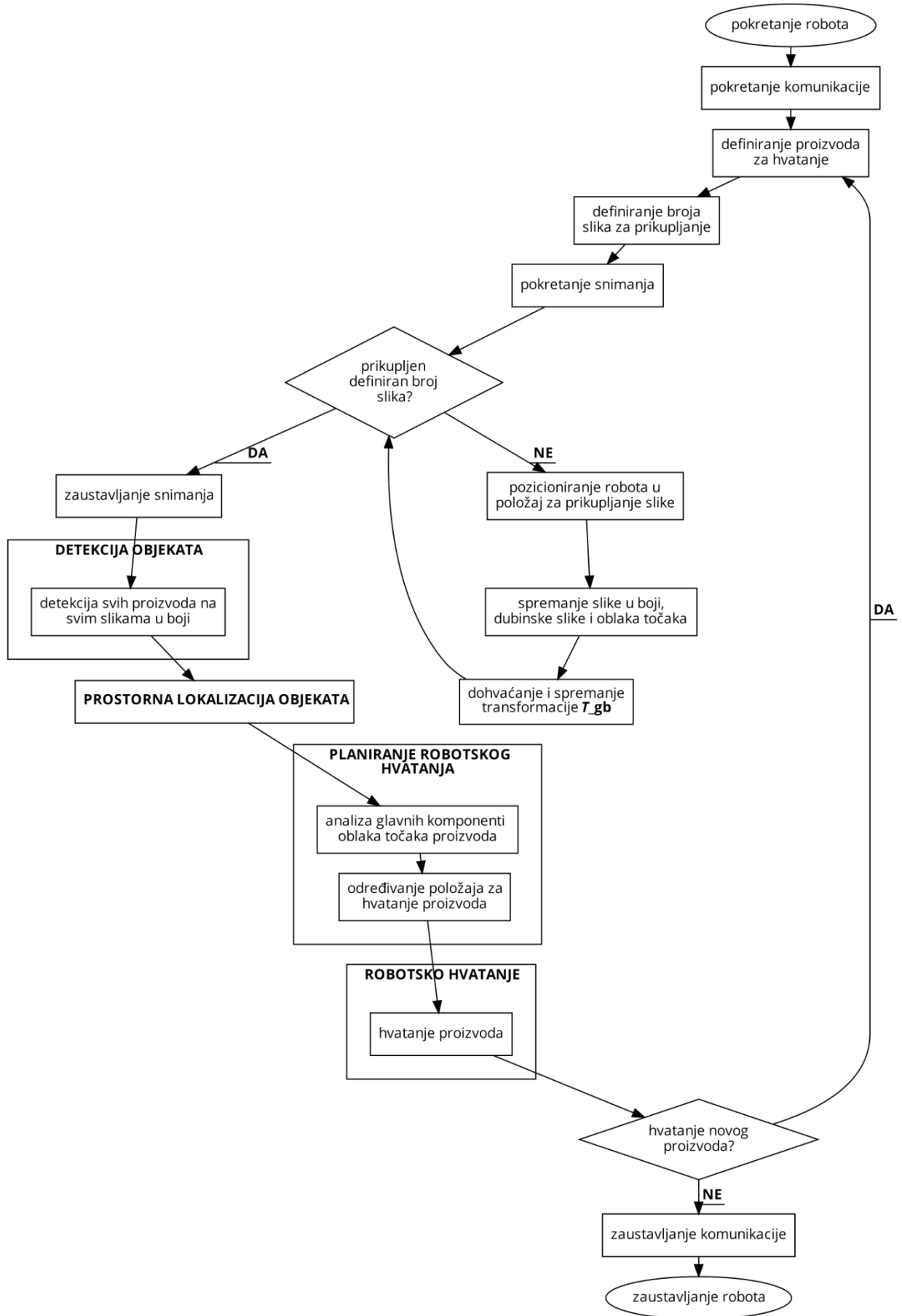
Kako bi robot mogao obavljati fizičku interakciju s odabranim proizvodima, između njega i ostalih komponenti sustava mora postojati komunikacija. Zbog osiguravanja pouzdane izmjene podataka između robota i računala kontroliranim redoslijedom u obliku bajtova, njihovu je komunikaciju prikladno ostvariti putem TCP protokola (eng. *Transmission Control Protocol*), pri čemu se računalo se nalazi u ulozi servera, a robot u ulozi klijenta. Kako bi računalo i robot uopće mogli razmjenjivati podatke, nužno je da se nalaze u istoj pod mreži (eng. *subnet*). Nakon pokretanja servera na računalu i uspostave veze, računalo robotu šalje određenu poruku te se na

robotu pokreće postupak prikupljanja podataka o sceni 3D vizijskim sustavom i pripadnih položaja vrha alata u odnosu na bazu robota. 3D vizijski sustav je s računalom spojeno putem USB-C-a, zbog čega se podaci o sceni direktno spremaju na računalo za njihovu kasniju obradu. Završetkom prikupljanja svih podataka o sceni, robot računalo odgovara definiranom porukom te se na računalo pokreće postupak detekcije svih proizvoda, obrade oblaka točaka scene, odabira proizvoda za hvatanje i planiranje njegovog hvatanja. Definiranjem položaja prilaska i položaja za hvatanje, računalo robotu opet šalje određenu poruku, nakon kojeg se robot pozicionira u položaj prilaska, zatim u položaj za hvatanje, hvata odabrani proizvod, odmiče se natrag u položaj prilaska i na kraju odlaže proizvod na predviđenu lokaciju. Nakon hvatanja proizvoda, robot konačno odgovara računalo definiranom porukom te se na računalo omogućuje ponovno izvršavanje cjelokupnog postupka robotskog hvatanja. Opisani je postupak pozicioniranje robota u navedene položaje za hvatanje odabranog proizvoda prikazan na slici 42.

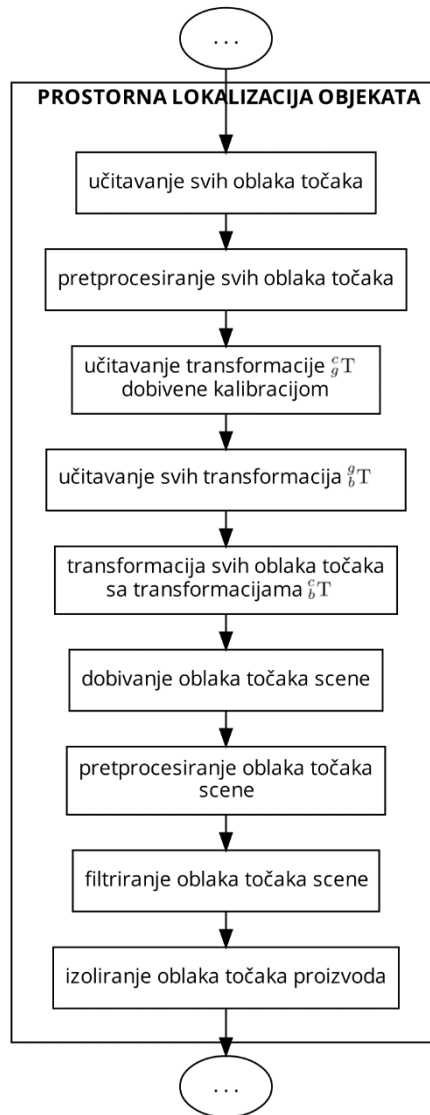


**Slika 42. Robotsko hvatanje proizvoda**

Iz prethodne je slike vidljivo da stvarni položaj za hvatanje u potpunosti odgovara onom koji je izračunat i vizualiziran, pri čemu je robot sposoban ispravno uhvatiti odabrani proizvod, uz uspješno odrađene sve prethodne korake cjelokupnog postupka robotskog hvatanja. U nastavku je, na slikama 43 i 44, prikazan dijagram toka cjelokupnog postupka robotskog rukovanja objektima.



Slika 43. Dijagram toka robotskog rukovanja objektima



Slika 44. Dijagram toka prostorne lokalizacije objekata

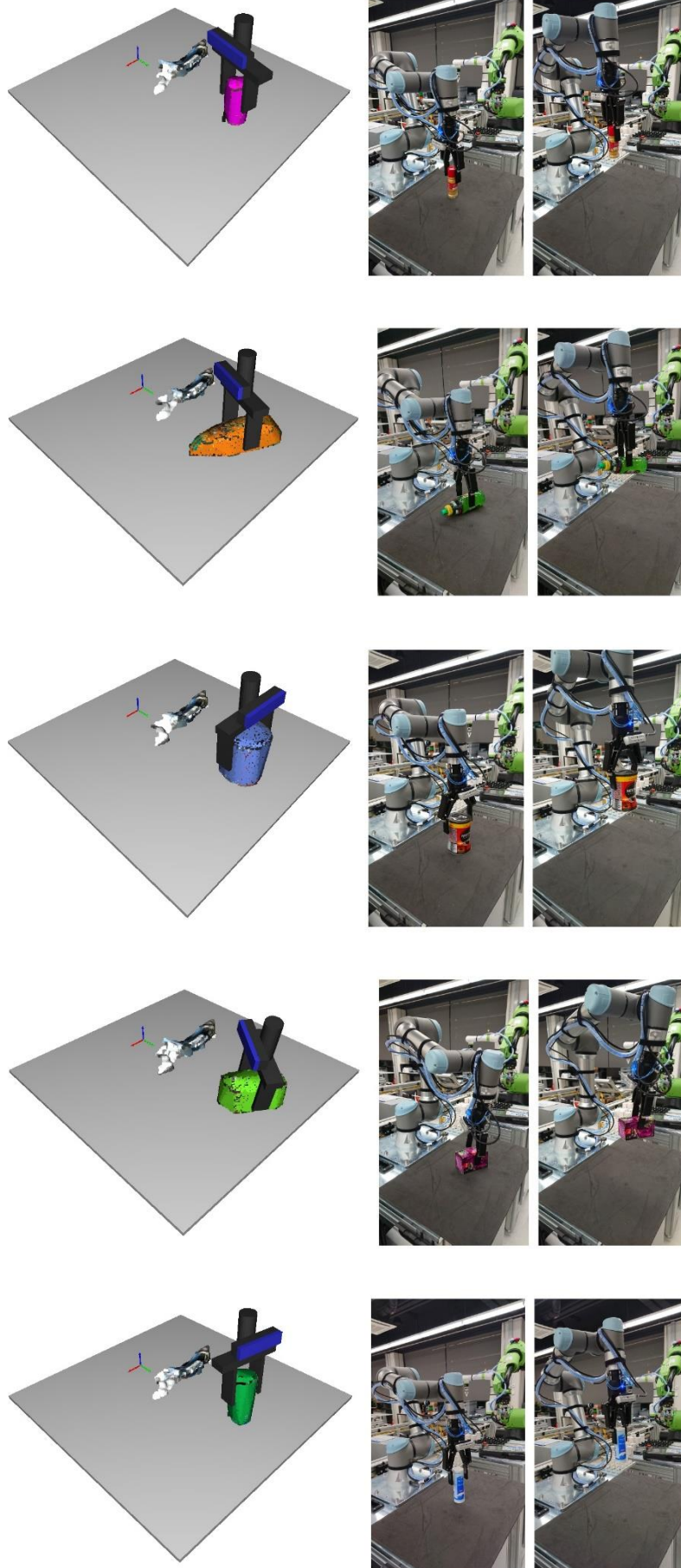


#### 4.4. Analiza rezultata robotskog rukovanja

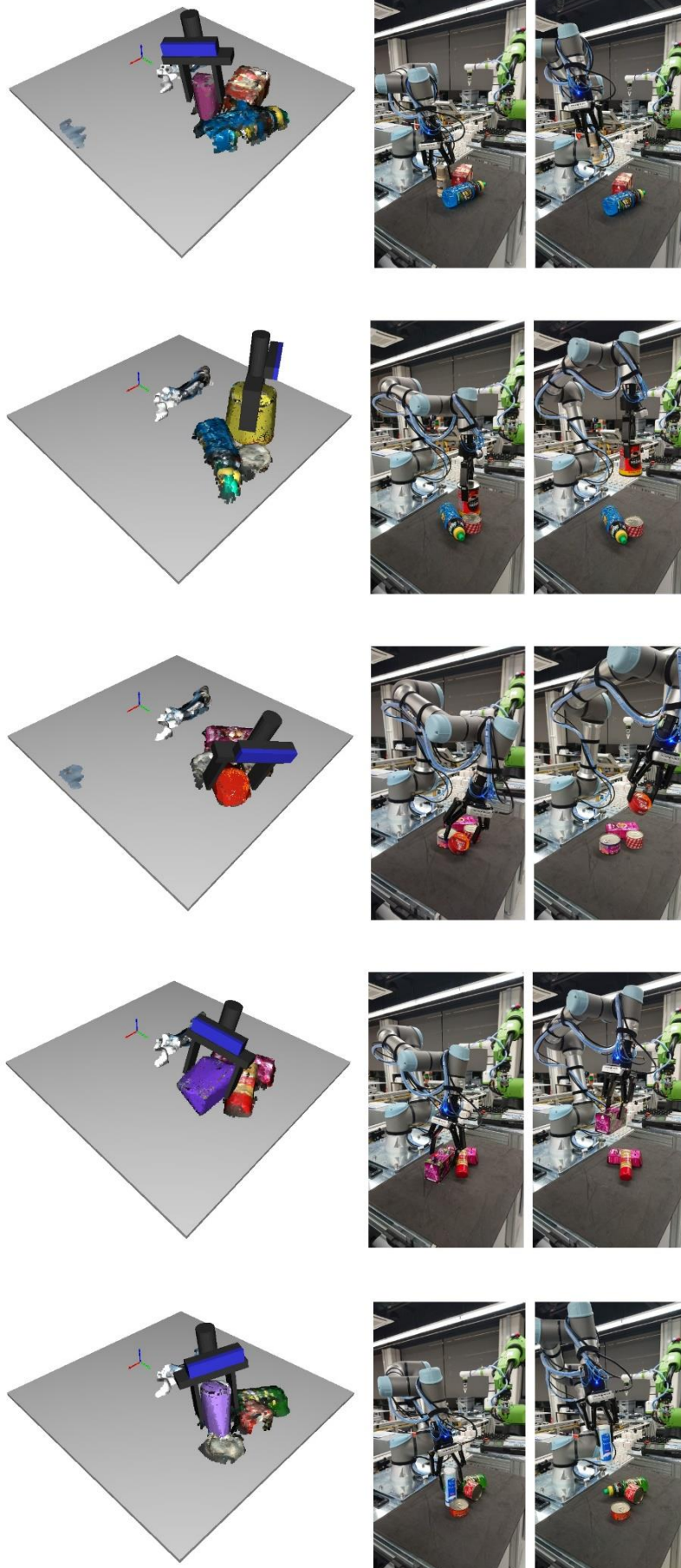
Kako bi se provjerila ispravnost rezultata dobivenih implementacijom cjelokupnog postupka robotskog rukovanja, postupak se provodi za nekoliko zasebnih te nekoliko grupiranih proizvoda u različitim položajima na sceni. U slučaju kada je odabrani proizvod jedini na sceni, razvijeni sustav daje vrlo robusne i visoko ponovljive rezultate te nema posebnih izazova koji bi se trebali detaljnije razložiti. Opisani se zaključak izvodi na temelju ponavljanja postupka rukovanja proizvodima na sceni nekoliko puta, pri čemu nema primjetnih razlika kod pozicioniranja robota u položaj za hvatanje ili nedosljednosti u ispravnosti hvatanja.

S druge strane, u slučaju kada je odabrani proizvod jedan iz grupe njih na sceni, rukovanje je tim proizvodom ostvarivo uz određena ograničenja. Prvo se ograničenje odnosi na dodirivanje više proizvoda, gdje razvijeni postupak robotskog rukovanja ne daje robusne rezultate za izoliranje proizvoda koji se u specifičnim položajima dodiruju u velikoj mjeri. Razlog opisanog nedostatka je izoliranje proizvoda samo na temelju njihovog izrezivanja u prostoru pomoću detektiranih slika, čime se ne mogu dovoljno dobro filtrirati podaci o rubnim dijelovima proizvoda. Navedeni bi se problem mogao riješiti izradom baze oblaka točaka proizvoda te njihovim izoliranjem na temelju oba razmotrena pristupa, no to uvelike prelazi okvire ovog zadatka te se zbog toga ovdje neće dalje razmatrati. Drugo se ograničenje odnosi na moguću koliziju robota pri gibanju do specifičnih položaja za hvatanje, gdje kod hvatanja odabranog proizvoda može doći do kolizije s nekim drugim proizvodom iz njegove okoline. Razlog navedenog nedostatka je planiranje hvatanja proizvoda samo na temelju njegovih podataka, a ne i podataka o njegovoj okolini, jer bi se potonjim pristupom trebalo voditi računa o velikom broju specifičnih slučajeva hvatanja. Navedeni se problem kompenzira obaveznom vizualnom kontrolom planiranog hvatanja i djelovanjem na temelju vizualne kontrole.

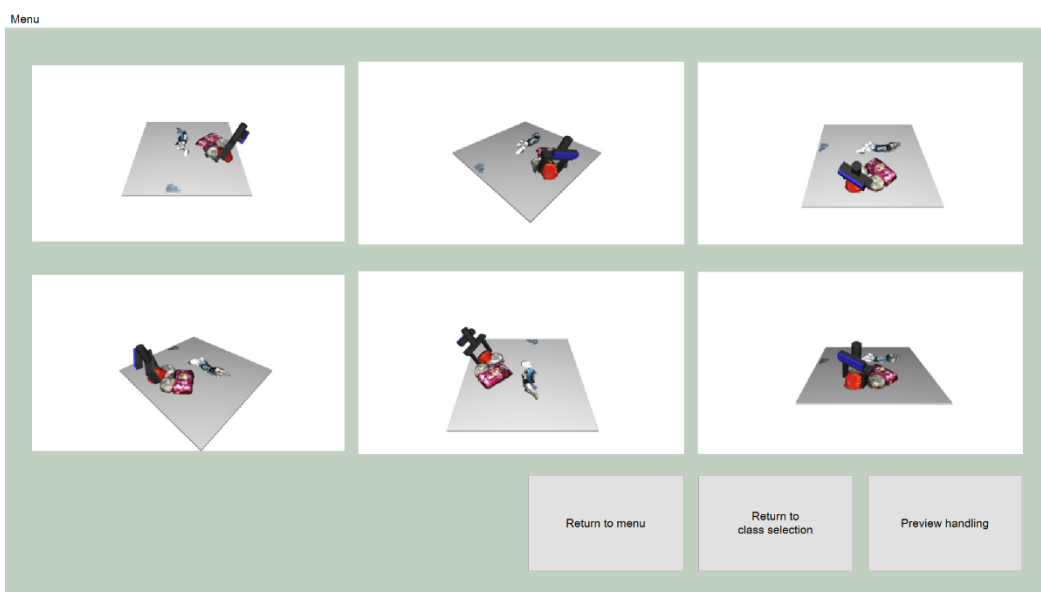
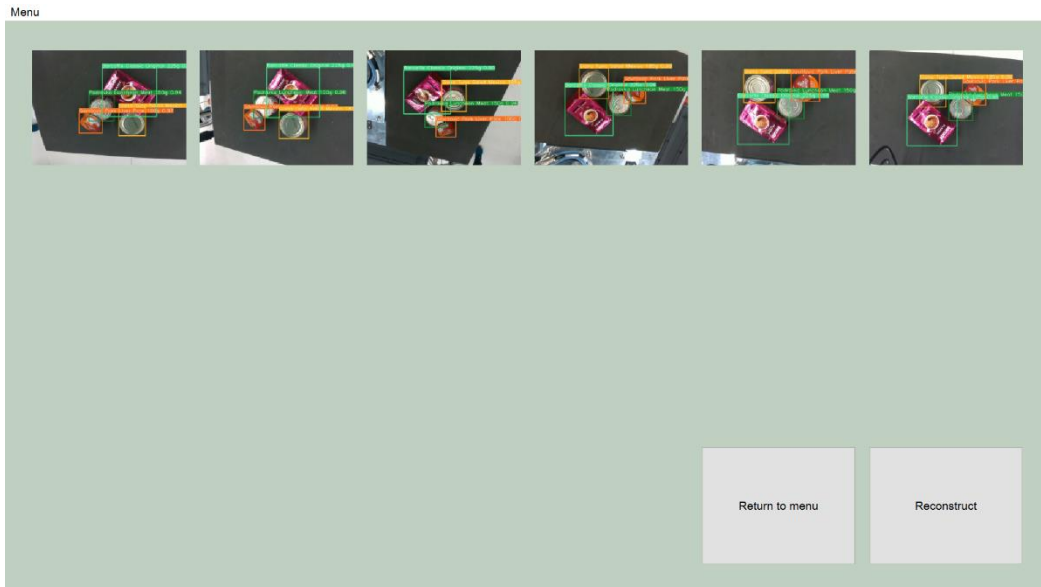
Ako se postupak robotskog rukovanja provodi za nekoliko grupiranih proizvoda na sceni te se vodi računa o opisanim ograničenjima, dobiveni rezultati postaju vrlo slični onima postupka provedenog za zasebne proizvode na sceni. Iz svega se navedenog zaključuje da je razvijeni sustav sposoban zadovoljavajuće rukovati odabranim proizvodima ako oni nisu previše zbijeni i ako se rezultati obrade na računalu vizualno kontroliraju. Kako bi se navedeni zaključak i potvrdio, na slici 45 je dan prikaz planiranog i stvarnog hvatanja jedinog, a na slici 46 jednog iz grupe proizvoda na sceni. Za cjelokupni se postupak robotskog rukovanja razvija i grafičko sučelje, čiji je prikaz dan na slici 47.



Slika 45. Robotsko hvatanje jedinog proizvoda na sceni



Slika 46. Robotsko hvatanje jednog iz grupe proizvoda na sceni



Slika 47. Razvijeno grafičko sučelje za robotsko rukovanje objektima

## 5. EVALUACIJA SUSTAVA ZA ROBOTSKO RUKOVANJE

U ovom je radu osmišljen i razvijen sustav za robotsko rukovanje predmetima između više drugih na sceni, bez eksplicitnog zadavanja njihovih pozicija i orijentacija, koji se temelji na prikupljanju slika i oblaka točaka predmeta, njihovoj detekciji na slikama neuronskom mrežom, određivanju položaja u prostoru te planiranju i realiziranju hvatanja na temelju geometrije. Kako bi se opisani postupak robotskog rukovanja uopće mogao izvesti, integriran je i postupak kalibracije korištenog 3D vizijskog sustava naspram korištene hvataljke. U nastavku je dana evaluacija razvijenog sustava, kao i njegova moguća daljnja upotreba.

Cjelovita programska podrška razvijena za potrebe zadatka te čitav skup slika izrađen i označen za potrebe detekcije, dostupni su unutar besplatnog razvojnog okruženja GitHub pod linkom <https://github.com/Doc1996/3D-object-handling>. Navedenu je programsku podršku moguće koristiti na operativnim sustavima Windows i Linux, pri čemu ona se u prvom slučaju pokreće datotekom `WINDOWS_3D_OBJECT_HANDLING_APPLICATION.bat`, a u drugom slučaju datotekom `LINUX_3D_OBJECT_HANDLING_APPLICATION.sh`.

### 5.1. Evaluacija i moguće primjene sustava

Razvijeni je sustav moguće evaluirati na temelju prethodnih rezultata analize pojedinih njegovih postupaka. Kada se govori o postupku kalibracije 3D vizijskog sustava naspram hvataljke, zaključuje se da postupak daje stabilne rezultate s točnošću koja se povećava sa brojem slika pomoću kojih se vrši kalibracija. Šumovi se uglavnom javljaju zbog ograničene ponovljivosti korištenog robota i rezolucije slika korištenog 3D vizijskog sustava. Neovisno o navedenome, zaključuje se da detekcija predmeta na slikama također daje rezultate koji su vrlo zadovoljavajući, uz to i u realnom vremenu, čime se ostvaruju preduvjeti za ispravno funkcioniranje cjelokupnog postupka robotskog rukovanja. Analizom rezultata navedenog postupka za nekoliko zasebnih te nekoliko grupiranih proizvoda u različitim položajima na sceni, zaključuje se da je zasebnim proizvodima na sceni moguće rukovati uz vrlo robusne i visoko ponovljive rezultate, dok je grupiranim proizvodima moguće zadovoljavajuće rukovati ako se oni značajno ne dodiruju i ako se rezultat planiranja hvatanja vizualno kontrolira. Navedena su ograničenja razvijenog sustava u kontekstu trenutnog, još uvijek otvorenog problema robotskog rukovanja predmetima, sasvim

smisljena te je o njima uputno razmišljati kao o posebnom izazovu, koji izlazi izvan okvira ovog zadatka.

Iako je primjena navedenog sustava izvorno vezana uz inteligentno robotsko rukovanje prehrambenim proizvodima, sustav bi se mogao primijeniti za mnogo širi asortiman proizvoda dimenzija, masa, geometrijskih oblika i mehaničkih svojstava, no treba imati na umu da bi za sve proizvode bilo potrebno imati odgovarajuću bazu podataka u svrhu detekcije. Opisani bi se sustav također mogao primijeniti samo u svojim dijelovima, kao sustav za detekciju objekata na slikama ili kao sustav za obradu oblaka točaka, no time se ne bi u potpunosti iskoristile njegove mogućnosti.

## **5.2. Izazovi i moguća poboljšanja sustava**

Kada se govori o izazovima za kvalitetnije funkcioniranje sustava, značajna je potreba za boljim izoliranjem proizvoda koji se u specifičnim položajima dodiruju u velikoj mjeri. Navedenom bi se izazovu moglo pristupiti izradom baze oblaka točaka proizvoda, pri čemu bi se izoliranje proizvoda obavljalo i na temelju navedene baze i na temelju postupka detekcije neuronskom mrežom. Drugi se značajan izazov odnosi na planiranje hvatanja proizvoda uz uzimanje u obzir podataka o njegovoj okolini, no za njega još nije definirano poopćeno rješenje za različite predmete u nestrukturiranoj okolini.

Cjelokupni je sustav osmišljen za rukovanje proizvodima unutar statične scene, zbog čega pri njegovom razvoju nisu razmatrani problemi koje se tiču dinamičnih scena. Ako se razmišlja u tom smjeru, sustav se može poboljšati mogućnošću praćenja i reakcije na različite promjene na sceni, što se može ostvariti dodatnim vizijskim sustavom postavljenim nepomično u prostoru. Navedeni bi vizijski sustav tako mogao pratiti i gibanje robota te reagirati na njegovo pozicioniranje u neadekvatne položaje, što bi značajno podiglo fleksibilnost i sigurnost sustava u cjelini.

## 6. ZAKLJUČAK

U okviru ovog diplomskog rada prikazan je razvoj sustava za robotsko rukovanje prehrambenim proizvodima, pri čemu se rješava problem prepoznavanja određenog predmeta između više drugih na sceni, njegove lokalizirati, a zatim i hvatanja robotom. Općeniti problem prepoznavanja predmeta podrazumijeva određivanje klase kojoj predmet pripada, za što je potrebno posjedovati bazu svih klasa, dok je općeniti problem lokalizacije predmeta određivanje njegove pozicije i orijentacije u prostoru. Problem robotskog hvatanja, s druge strane, podrazumijeva ispravno pozicioniranje hvataljke te hvatanje predmeta bez sudaranja s ostalim predmetima u okolini. Osmišljeno je da se podaci o sceni koja uključuje predmete i njihovu okolinu prikupljaju u obliku oblaka točka 3D vizijskim sustavom postavljenim na robota, obrađuju na računalu u svrhu prepoznavanja i lokalizacije predmeta, a zatim koriste za njegovo hvatanje.

Razvijen je sustav koji se temelji na prikupljanju slika i oblaka točaka predmeta, njihovoj detekciji na slikama neuronskom mrežom, određivanju položaja u prostoru te planiranju i realiziranju hvatanja na temelju geometrije. Navedene se slike i oblaci točaka prikupljaju 3D vizijskim sustavom Intel RealSense D435 postavljenim na kolaborativnog robota Universal Robots UR5, a kako bi se postupak robotskog rukovanja mogao ispravno izvesti, provodi se i kalibracija 3D vizijskog sustava naspram njegove hvataljke. U svrhu detekcije predmeta na slikama koristi se neuronska mreža YOLO v5 trenirana na posebno izrađenom skupu označenih slika, u svrhu određivanja položaja predmeta u prostoru knjižnica za obradu oblaka točaka Open3D, a kao za planiranje hvatanja predmeta algoritam temeljen na analizi glavnih komponenti (eng. *Principal component analysis – PCA*). Navedeni se elementi sustava povezuju grafičkim sučeljem, dok se komunikacija između robota i računala ostvaruje TCP/IP protokolom.

Analizom rezultata razvijenog sustava za robotsko rukovanje za nekoliko zasebnih te nekoliko grupiranih predmeta u različitim položajima na sceni, zaključuje se da je predmetima na sceni moguće zadovoljavajuće rukovati ako se proizvodi značajno ne dodiruju i ako se rezultat planiranja hvatanja vizualno kontrolira. Opisani je sustav moguće primijeniti za širok asortiman proizvoda različitih dimenzija, masa, geometrijskih oblika i mehaničkih svojstava, no treba imati na umu da je za sve proizvode potrebno imati odgovarajuću bazu podataka u svrhu detekcije. Značajno poboljšanje sustava, u vidu njegove fleksibilnosti i sigurnosti, moglo bi se postići integriranjem mogućnosti praćenja i reakcije na različite promjene na sceni. Sve bržim razvojem robotike i automatizacijom proizvodnih procesa, očekuje se da će problemi slični razmotrenom u ovom radu biti od sve većeg značaja, što daje poticaj daljnjem interesu za ovo područje.



## 7. LITERATURA

- [1] Craig, J. J. *Introduction to Robotics: Mechanics and Control*. 3rd ed., Pearson Education International, 2005.
- [2] „Factory automation and control: Isolation in Industrial Robot Systems.“ *Texas Instruments*, 2017, <https://training.ti.com/isolation-industrial-robot-systems>. Pristupljeno 12.1.2022.
- [3] Anwekar, V. „Vision-Guided Robotics.“ *International Society of Automation*, 2020, <https://blog.isa.org/vision-guided-robotics>. Pristupljeno 12.1.2022.
- [4] Brownlee, J. „A Gentle Introduction to Object Recognition With Deep Learning.“ *Machine Learning Mastery*, 2019, <https://machinelearningmastery.com/object-recognition-with-deep-learning/>. Pristupljeno 12.1.2022.
- [5] Myhre, T. A. „Robot camera calibration.“ *Torsteins webpage*, 2018, [https://www.torsteinmyhre.name/snippets/robcam\\_calibration.html](https://www.torsteinmyhre.name/snippets/robcam_calibration.html). Pristupljeno 12.1.2022.
- [6] „e-Series Cobots.“ *Universal Robots*, 2022, <https://www.universal-robots.com/e-series/>. Pristupljeno 12.1.2022.
- [7] „2F-85 and 2F-140 Grippers.“ *Robotiq*, 2022, <https://robotiq.com/products/2f85-140-adaptive-robot-gripper>. Pristupljeno 12.1.2022.
- [8] „Intel RealSense Depth Camera D435.“ *Intel RealSense*, 2022, <https://www.intelrealsense.com/depth-camera-d435/>. Pristupljeno 12.1.2022.
- [9] Akadas, K., Gangisetty, S. „3D Semantic Segmentation for Large-Scale Scene.“ *Computer Vision Foundation, KLE Technological University*, 2020, [https://openaccess.thecvf.com/content/ACCV2020W/MLCSA/papers/Akadas\\_3D\\_Semantic\\_Segmentation\\_for\\_Large-Scale\\_Scene\\_Understanding\\_ACCVW\\_2020\\_paper.pdf](https://openaccess.thecvf.com/content/ACCV2020W/MLCSA/papers/Akadas_3D_Semantic_Segmentation_for_Large-Scale_Scene_Understanding_ACCVW_2020_paper.pdf). Pristupljeno 12.1.2022.
- [10] Williamson, O. „Internet Protocol suite.“, *Medium*, 2019, <https://medium.com/@ohwilliamson1/internet-protocol-suite-ee585941859b>. Pristupljeno 12.1.2022.
- [11] „Qt Documentation.“, *Qt*, 2022, <https://doc.qt.io/qt-5/qtquick3d-antialiasing-example.html>. Pristupljeno 12.1.2022.

- [12] „Camera Calibration and 3D Reconstruction.“, *Open Source Computer Vision*, 2022, [https://docs.opencv.org/4.x/d9/d0c/group\\_\\_calib3d.html](https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html). Pristupljeno 12.1.2022.
- [13] Despois, J. „Stop Feeding Garbage To Your Model! – The 6 biggest mistakes with datasets and how to avoid them.“, *Hackernoon*, 2018, <https://hackernoon.com/stop-feeding-garbage-to-your-model-the-6-biggest-mistakes-with-datasets-and-how-to-avoid-them-3cb7532ad3b7?fbclid=IwAR1-5VXuYhd3QKFgryj8p9RnGTGUNmWGMWZhDWPftfaqLgU8BHUM5RHffQ8>. Pristupljeno 12.1.2022.
- [14] Tang, B. „Image and Video Understanding: A Roadmap For Implementation.“, *Medium*, 2018, <https://medium.com/stradigiai/image-and-video-understanding-a-roadmap-for-implementation-a006f6f3fe0c>. Pristupljeno 12.1.2022.
- [15] Brems, M. „A One-Stop Shop for Principal Component Analysis.“, *Towards Data Science*, 2017, <https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c>. Pristupljeno 12.1.2022.