

Program

Digitalna transformacija robotiziranih tovarn prihodnosti

DIGITOP

Poročilo D6.1

Vizualna kontrola kakovosti in optimalno vodenje v
rekonfigurabilnih robotskih celicah

Projekt/RRP

Vrsta dokumenta

Verzija dokumenta

Sodelujoči partnerji

Avtorji

Datum priprave

RRp6

Poročilo o rezultatih

Končna

MAR, IJS-E1

Rok Pahič, Miha Deniša,

Ana Gabriela Kostanjevec, Janez Kunavar

7.11.2025

Kazalo

1	Uvod	3
2	Sistemska arhitektura in komunikacija	3
3	Generiranje točk pregleda	5
4	Generiranje optimalne trajektorije zajema	5
4.1	Optimizacija postavitve kamer in zaporedja pregledov	6
4.2	Optimizacija postavitve kamere glede na delovni prostor robota	7
4.3	Načrtovanje poti z uporabo OMPL	7
4.4	Časovna optimizacija trajektorije	8
5	Obdelava slik in detekcija anomalij	9
6	Načrt eksperimentalne validacije	10

1 Uvod

V okviru raziskovalno-razvojnega programa **DIGITOP** razvijamo napredne rešitve za digitalno transformacijo robotiziranih tovarn prihodnosti. Cilj projekta je razvoj rekonfigurabilnih in inteligentnih robotskih sistemov, ki omogočajo agilno avtomatizacijo proizvodnih procesov, pri čemer ima ključno vlogo vizualni nadzor kakovosti. Vizualni pregled je eden izmed najbolj zahtevnih segmentov proizvodne avtomatizacije, saj zahteva hkratno integracijo natančne mehanike, zaznavanja, umetne inteligence in optimizacijskih metod za zagotavljanje hitrega in zanesljivega odkrivanja napak.

Poročilo D6.1 predstavlja eksperimentalni razvoj in integracijo sistema za inteligentni vizualni pregled, namenjenega avtomatiziranemu pregledu površin različnih industrijskih objektov. Glavni poudarek je na celotnem poteku vizualnega pregleda – od definiranja točk pogleda, optimizacije robotskih trajektorij in postavitve kamer do obdelave slik in detekcije anomalij. Poudarek je praktičen, saj sistem temelji na implementaciji in testiranju realne pregledovalne celice, zasnovane in izvedene v podjetju Marovt v sodelovanju z IJS-E1.

Sistem omogoča avtomatizirano izvajanje pregledovalnih programov za posamezne objekte, pri čemer se robot giblje po vnaprej načrtovani trajektoriji in v vsaki točki pogleda zajame sliko s statično nameščeno kamero. Programska arhitektura temelji na okolju ROS2, ki omogoča integracijo različnih modulov – od zajema podatkov in sinhronizacije robota do uporabniškega nadzora preko spletnega grafičnega vmesnika (MarovtGUI). V sistem je vgrajena tudi podpora za učenje in uporabo nevronskega modela za detekcijo anomalij, kar omogoča nadgradnjo v smeri popolnoma samostojnega delovanja.

Poleg osnovnih funkcionalnosti poročilo vključuje tudi pregled razvitih komponent, ki omogočajo dodatno optimizacijo procesa vizualnega pregleda, kot so optimizacija postavitve kamer glede na objekt in robota, optimizacija zaporedja pregledovalnih točk na osnovi pristopov TSP, uporaba RBS in OMPL knjižnic za načrtovanje optimalnih trajektorij ter razvoj fleksibilne mehanike podpore (monopod) za pozicioniranje kamer v rekonfigurabilnih celicah.

Cilj poglavja D6.1 je pokazati, kako je mogoče z integracijo robotskega sistema, napredne programske infrastrukture in metod umetne inteligence razviti praktično izvedljivo rešitev za vizualni nadzor v industrijskih pogojih, ki je prilagodljiva različnim objektom in zahtevam pregleda.

2 Sistemska arhitektura in komunikacija

Sistem za inteligentni vizualni pregled je zasnovan modularno, s ciljem omogočiti popolno integracijo vseh procesov, potrebnih za avtomatizirano zaznavanje površinskih napak na industrijskih objektih. Celotno delovanje temelji na okolju ROS2, ki povezuje programske module, podatkovne vire, fizične naprave ter uporabniški vmesnik v enoten komunikacijski sistem.

Arhitektura je sestavljena iz treh glavnih sklopov: grafičnega vmesnika, osrednjega dela sistema in zunanje periferije. Med posameznimi sklopi poteka dvosmerna komunikacija prek podatkovne baze in standardnih industrijskih protokolov.

Zgornji sloj predstavlja grafični vmesnik (MarovtGUI), ki uporabniku omogoča generiranje pregledovalnih programov, nadzor izvajanja, učenje modelov detekcije anomalij ter pregled rezultatov. Prek komunikacijskega mostu, zasnovanega na protokolu WebSocket, je povezan z jedrom sistema, ki deluje v okolju ROS2. Vmesnik pošilja ukaze za izvedbo posameznih postopkov,

pregleda, zajete slike, učni podatki in rezultati modelov.

Spodnji sloj sestavljata fizični napravi, robot in kamera. Robot izvaja pripravljene trajektorije, medtem ko kamera sinhrono zajema slike v vnaprej določenih točkah pogleda. Kalibracija sistema zagotavlja pravilno prostorsko povezavo med koordinatnimi sistemi robota in kamere, kar omogoča natančno pozicioniranje pogleda in ponovljivost zajema.

Takšna arhitektura omogoča enostavno nadgradnjo posameznih komponent, neodvisno testiranje modulov in razširitev sistema z dodatnimi kamerami ali roboti. Modularnost in odprta komunikacijska struktura sta ključni prednosti rešitve, saj omogočata prilagoditev sistema različnim industrijskim scenarijem in zahtevam po razširjeni avtomatizaciji pregleda kakovosti.

3 Generiranje točk pregleda

Prvi korak v procesu vizualnega pregleda je generiranje točk pogleda, iz katerih kamera zajame slike tako, da se celotna površina objekta pregleda z zahtevano natančnostjo. Postopek je geometrijsko osnovan in izhaja iz znane geometrije objekta ter karakteristik uporabljenih kamer. Njegov cilj je določiti najmanjši možni nabor pogledov, s katerimi je zagotovljena popolna pokritost vseh površin objekta, ki jih je potrebno pregledati.

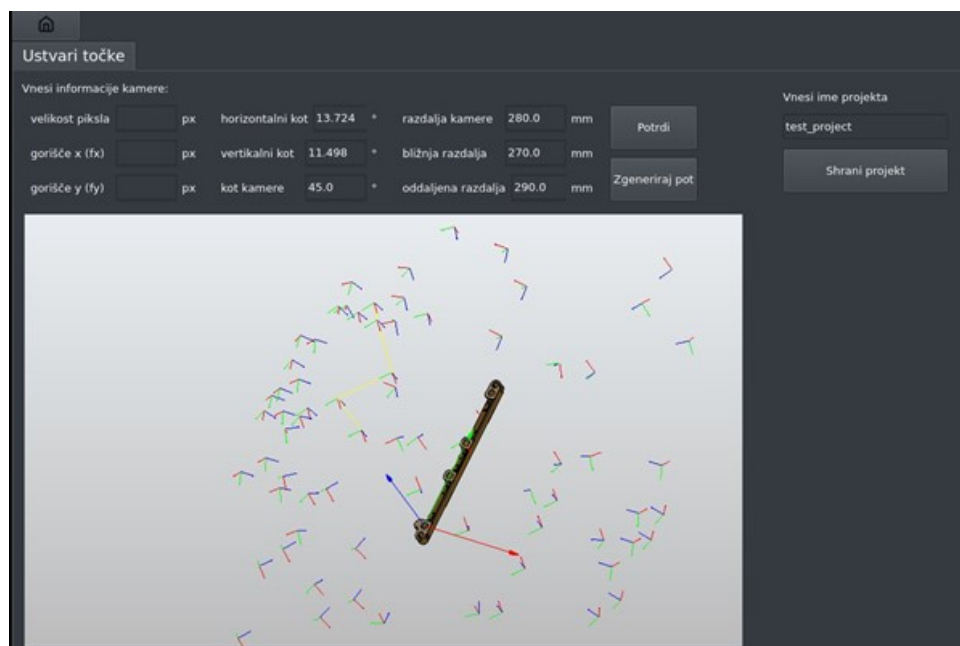
V postopku generiranja se upoštevajo parametri zajema, ki jih določi uporabnik ali so del specifikacij pregledovalnega programa. Ti vključujejo minimalno gostoto pikslov na površinsko enoto, največji dopusten kot med optično osjo kamere in normalo površine ter lastnosti kamere, kot so vidno polje, ločljivost in območje globinske ostrine. Na osnovi teh podatkov sistem izračuna ustrezne točke pogleda, ki zagotavljajo zahtevano kakovost slik in enakomerno porazdelitev zajema po celotni površini objekta.

Za vsak posamezen objekt so na voljo njegovi geometrijski podatki, pridobljeni iz CAD modela ali 3D skeniranja. Programska rešitev znotraj modula MarovtGUI uporabi te podatke za določitev vseh relevantnih površin in za izračun pripadajočih pogledov. Rezultat postopka je množica točk pregleda, ki jih sistem prikaže v grafičnem vmesniku skupaj s pripadajočimi položaji kamer. Uporabnik lahko te točke pregleda vizualno preveri in po potrebi parametrično prilagodi.

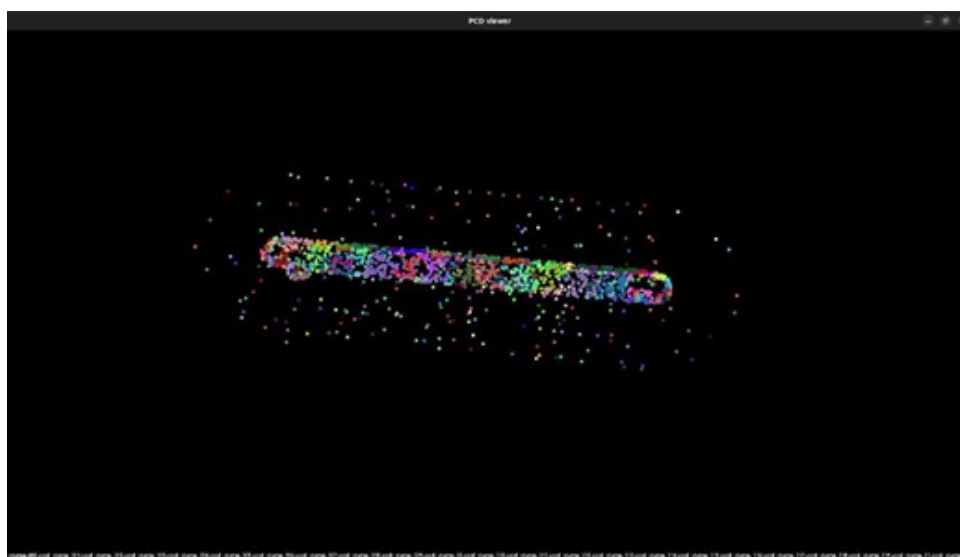
Postopek generiranja točk ne vključuje gibanja robota, temveč določa le prostorske položaje, iz katerih mora kamera zajeti slike, da se zagotovi popolna pokritost objekta. Ti geometrijsko določeni pogledi predstavljajo vhodne podatke za nadaljnje faze načrtovanja, kjer se izvede optimizacija postavitve kamer, zaporedja pregledov in trajektorij gibanja robota, kot je podrobneje opisano v naslednjem poglavju in v poročilih D2.2 ter D2.1.

4 Generiranje optimalne trajektorije zajema

Ko so določene točke pregleda, iz katerih je potrebno zbrati slike za popolno pokritost objekta, sistem v naslednjem koraku izvede generiranje optimalne trajektorije zajema. Cilj tega postopka je minimizirati skupni čas pregleda ob upoštevanju geometrije objekta, razpoložljivih kamer, omejitev gibanja robota in zahtevane natančnosti zajema. Proces generiranja trajektorije združuje tri ključne korake: optimizacijo postavitve kamer in zaporedja pregledov, načrtovanje poti brez trkov ter časovno optimizacijo izvedbe trajektorije.



(a) Modul grafičnega vmesnika za pripravo generiranja in prikaza točk ter robotske trajektorije zajema. Uporabnik nastavi parametre zajema, generira točke pregleda in preveri pokritost objekta.



(b) Rezultat generiranja točk pogleda in optimizacije njihove lokacije. Na površini objekta so označene točke zajema, ki zagotavljajo popolno pokritost glede na karakteristike kamere in zahtevane parametre zajema.

Slika 3: Prikaz postopka generiranja točk pregleda. Slika (a) prikazuje programski modul v grafičnem vmesniku, kjer uporabnik določi parametre in generira točke zajema, slika (b) pa prikazuje rezultat optimizacije, torej končno razporeditev točk pogleda na površini objekta.

4.1 Optimizacija postavitve kamer in zaporedja pregledov

Optimizacija pregleda zajema temelji na dveh medsebojno povezanih korakih: določanju optimalnega zaporedja pregledovalnih točk in optimizaciji postavitve kamer. Postopek, opisan v poročilu

D2.2, obravnava celoten proces kot enoten optimizacijski problem minimizacije skupnega časa pregleda.

Najprej se izvede optimizacija zaporedja pregledovalnih točk, kjer vsaka točka predstavlja vozlišče v grafu, povezave med njimi pa so utežene s časom prehoda robota. Problem je formuliran kot problem trgovskega potnika (TSP), ki se rešuje z metodami, kot sta 2-opt heuristika ali binarno celoštevilsko programiranje (BIP). Rezultat je zaporedje, ki minimizira skupni čas premika med posameznimi pogledi.

Ta rezultat se uporabi kot del cenilke za izvedbo optimizacije postavitve kamer. Položaji kamer se prilagodijo tako, da omogočajo zajem vseh točk z minimalnim številom sprememb položaja robota in brez nepotrebnih prekrivanj pogledov. Rezultat postopka je optimalna kombinacija razporeditve kamer in zaporedja pregledov, ki predstavlja vhod v nadaljnje načrtovanje trajektorije gibanja robota.

4.2 Optimizacija postavitve kamere glede na delovni prostor robota

Poleg optimizacije postavitve kamer glede na objekt se izvaja tudi postopek optimizacije postavitve kamere brez upoštevanja geometrije kosa. Ta pristop je namenjen določanju osnovne lege kamere v delovnem prostoru, ki zagotavlja stabilno spremljanje gibanja robota in dobro pokritost pregledovanega območja. Postopek temelji na kinematičnih značilnostih robota in njegovi manipulabilnosti.

Na osnovi intrinzičnih parametrov kamere (žariščna razdalja, velikost tipala, vidni kot) se določi ravnina pogleda, ki opisuje območje zajema kamere v prostoru. Ravnina je definirana s položajem in orientacijo kamere ter predstavlja referenčno površino, znotraj katere se ocenjuje dosegljivost robota. Postavitev koordinatnega sistema kamere in pripadajoče ravnine pogleda je prikazana na sliki 4(a).

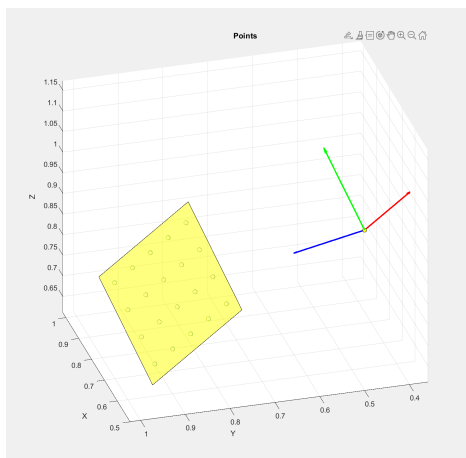
Na podlagi kinematike uporabljenega robota (v trenutnem primeru UR10e) se za posamezne točke delovnega prostora izračuna manipulabilnost, ki izraža sposobnost robota za gibanje in orientacijo. Ko se ravnina pogleda postavi na izbrano lokacijo, se oceni, kakšno manipulabilnost robot dosega v tem območju. Primer vizualizacije ravnine pogleda kamere v območju manipulabilnosti robota je prikazan na sliki 4(b). Ti podatki služijo kot osnovna cenilka za optimizacijo lege kamere, kjer se išče konfiguracija, pri kateri je manipulabilnost v vidnem polju kamere čim boljša.

V postopek se vključujejo tudi prostorske omejitve, med katerimi je ključna zahteva, da se kamera nahaja izven neposrednega delovnega območja robota oziroma čim dlje od njega, s čimer se zmanjšuje možnost trkov. V nadaljevanju se bodo v optimizacijo dodajali še dodatni pogoji in kriteriji, povezani z vidnostjo, stabilnostjo in mehanskimi omejitvami sistema.

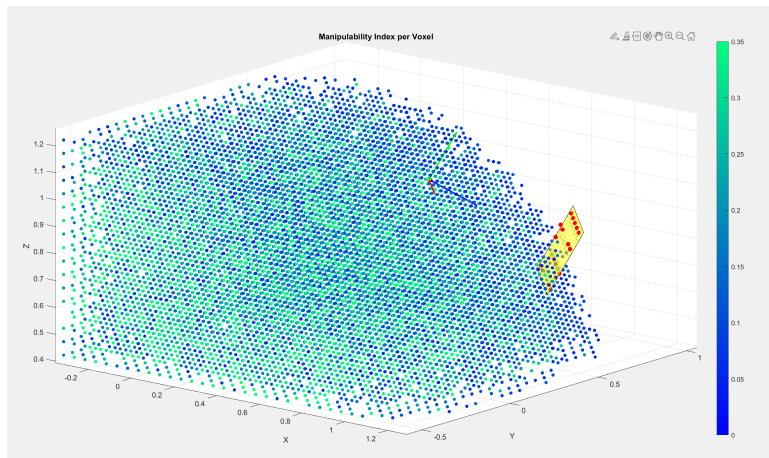
4.3 Načrtovanje poti z uporabo OMPL

Ko je zaporedje pregledovalnih točk določeno, sledi načrtovanje dejanske poti robota med njimi. Ta korak temelji na metodologiji, opisani v poročilu D2.1, kjer se za načrtovanje poti uporablja odprtokodna knjižnica OMPL (Open Motion Planning Library). Algoritmi iz knjižnice OMPL omogočajo iskanje poti v konfiguracijskem prostoru robota ob upoštevanju omejitev gibanja, meja sklepov in prisotnosti ovir v delovnem prostoru.

Za vsak prehod med dvema zaporednima točkama pregleda se ustvari pot, ki zagotavlja kolizijsko varno gibanje robota. Algoritmi, kot sta RRT (Rapidly-exploring Random Tree) in PRM



(a) Koordinatni sistem kamere in pripadajoča ravnina pogleda. Na osnovi intrinzičnih parametrov kamere se določi položaj in naklon ravnine pogleda, ki opisuje območje zajema slike v prostoru.



(b) Ravnina pogleda kamere v območju manipulabilnosti robota UR10e. Barvna lestvica prikazuje vrednosti manipulabilnosti, ki jih robot dosega na ravnini pogleda glede na trenutno lego kamere.

Slika 4: Optimizacija postavitve kamere glede na manipulabilnost robota. Slika (a) prikazuje geometrijsko definicijo koordinatnega sistema kamere in ravnine pogleda, slika (b) pa primer porazdelitve manipulabilnosti robota UR10e na ravnini pogleda.

(Probabilistic Roadmap), generirajo množico možnih poti, izmed katerih se izbere tista, ki ima najmanjšo skupno dolžino in izpolnjuje vse kinematične omejitve. Sistem tako izdelava trkov prosto pot, ki povezuje vse točke pregleda v zaporedju, določenem v prejšnji fazi.

Rezultat načrtovanja poti je sekvenca prostorskih točk in orientacij robota, ki predstavljajo izvedljivo geometrijsko trajektorijo. Ta trajektorija se nato uporabi kot osnova za časovno parametrizacijo in optimizacijo gibanja.

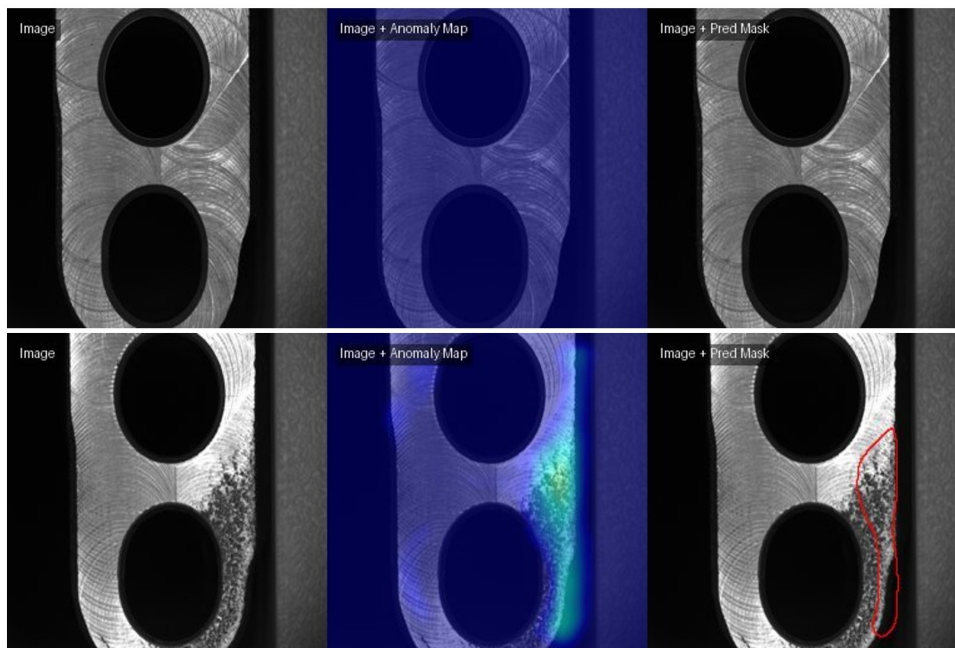
4.4 Časovna optimizacija trajektorije

Zadnji korak v postopku generiranja optimalne trajektorije zajema je časovna optimizacija gibanja robota. Na osnovi poti, dobljene v prejšnji fazi, se izvede časovna parametrizacija, pri kateri se določi, kako hitro se robot giblje med posameznimi točkami, ob upoštevanju omejitev hitrosti, pospeškov in dinamike sistema. Ta pristop je podrobno opisan v poročilu D2.1.

Za vsak segment poti se določi časovna funkcija, ki zagotavlja gladko gibanje brez sunkov in znotraj dovoljenih dinamičnih meja. Uporabljeni so trapezoidni ali polinomski hitrostni profili, ki omogočajo zvezno pospeševanje in zaviranje. V primeru zahtev po čim krajšem času cikla se izvede časovno optimalna parametrizacija, pri kateri je čas trajanja poti minimalen, vendar še vedno izvedljiv znotraj omejitev sistema.

Rezultat časovne optimizacije je končna trajektorija robota, definirana s položajem, hitrostjo in pospeškom v odvisnosti od časa. Takšna trajektorija omogoča hitro, gladko in ponovljivo gibanje robota med vsemi točkami pregleda, kar bistveno skrajša skupni čas vizualnega pregleda in poveča učinkovitost sistema.

Celoten postopek generiranja optimalne trajektorije zajema tako združuje optimizacijo zaporedja pregledovalnih točk, optimizacijo postavitve kamer, načrtovanje poti brez trkov in časovno



Slika 5: Prikaz območja objekta brez in z zaznano napako. Rezultat prikazuje detekcijo površinske anomalije na osnovi izhodov modela iz knjižnice Anomalib.

optimizacijo gibanja. Na ta način sistem zagotavlja optimalno ravnotežje med natančnostjo pregleda, hitrostjo izvedbe in mehanskimi omejitvami robotske celice.

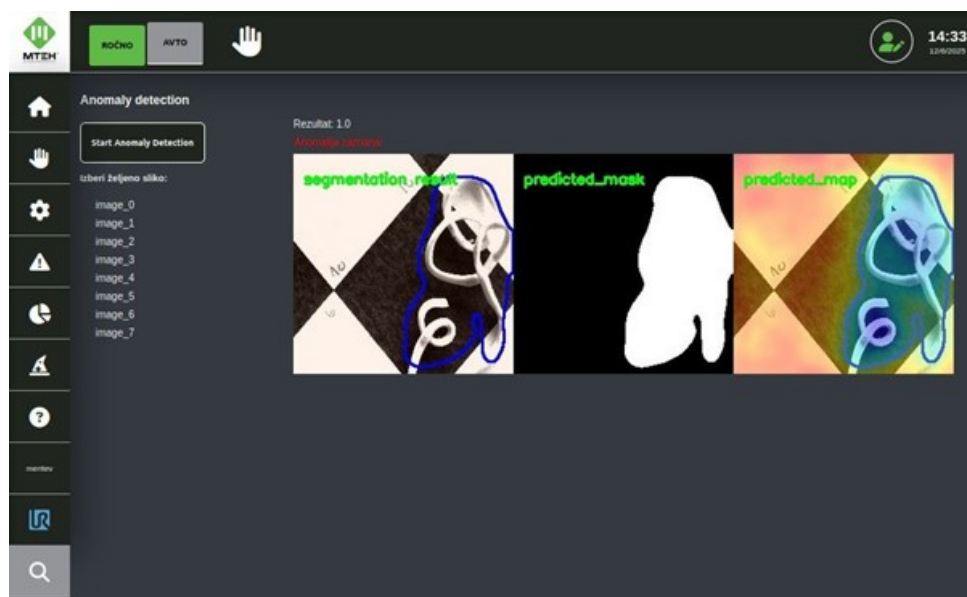
5 Obdelava slik in detekcija anomalij

Detekcija napake na površini objekta temelji na pristopu detekcije anomalije. Uporabljena je programska knjižnica Anomalib, ki združuje orodja za implementacijo več različnih globokih nevronskega modelov. Pričakovan rezultat je klasifikacija slike brez ali z anomalijo in označeno mesto zaznane anomalije. Vsak posamezen pogled kamere se obravnava s svojim modelom, tako da se za objekt z določenim številom definiranih točk zajema zbere ločene podatke in nauči ustrezno število modelov.

Pregled zbranih podatkov, njihovo morebitno označevanje in pregled zgodovine rezultatov pri uporabi modelov omogoča poseben modul v grafičnem vmesniku. Modul omogoča tudi nadzor nad procesom učenja posameznih modelov ter vizualizacijo rezultatov detekcije na slikah zajetih objektov.

Sistem trenutno temelji na uporabi modela PatchCore, vendar omogoča enostavno integracijo tudi drugih modelov iz knjižnice Anomalib. Model PatchCore za odkrivanje anomalij uporablja pomnilniške baze značilk izrezkov (ang. patch features), izluščenih iz vnaprej naučenega jedra konvolucijske nevronske mreže (CNN). Model shranjuje reprezentativne značilke izrezkov iz običajnih učnih slik in zaznava anomalije s primerjavo izrezkov testnih slik s to pomnilniško bazo.

Model uporablja iskanje najbližjih sosedov za iskanje najbolj podobnih izrezkov v pomnilniški bazi ter izračunava oceno anomalije na podlagi teh razdalj. Za učenje modela PatchCore je potrebno zbrati podatke samo od vzorcev objekta brez napake, kar olajša postopek učenja, saj so v proizvodnem procesu objekti z napako redkejši, prav tako pa se lahko pojavijo še nepredvidene



Slika 6: Modul grafičnega vmesnika za označevanje trening podatkov, trening modelov za detekcijo anomalije in pregled shranjenih rezultatov. Uporabniku omogoča vizualno označevanje vzorcev, nadzor učenja in pregled zgodovine rezultatov.

napake.

6 Načrt eksperimentalne validacije

Eksperimentalna validacija sistema za inteligentni vizualni pregled bo predstavljala naslednjo fazo razvoja v okviru projekta DIGITOP. Namen validacije je preveriti delovanje celotnega poteka vizualnega pregleda v realnem okolju in potrditi funkcionalnost posameznih modulov, razvitih v sklopu poročila D6.1.

Načrtovana validacija bo izvedena v laboratorijskem okolju in kasneje v proizvodni celici podjetja Marovt. Testni sistem bo vključeval robotski manipulator, eno ali dve industrijski kameri, računalnik z nameščenim okoljem ROS2 ter programske module za generiranje točk pogleda, optimizacijo trajektorij in detekcijo anomalij.

Cilj eksperimentalne postavitve je preizkusiti medsebojno integracijo vseh komponent sistema: zajem podatkov, sinhronizacijo gibanja robota in kamere, delovanje modula za detekcijo anomalij ter vizualizacijo rezultatov v grafičnem vmesniku. Posebna pozornost bo namenjena preverjanju natančnosti kalibracije sistema, stabilnosti gibanja robota ter času izvedbe posameznega cikla pregleda.

Na osnovi pridobljenih rezultatov bo mogoče oceniti učinkovitost razvitih algoritmov za generiranje točk, optimizacijo zaporedja pregledov in trajektorij ter delovanje metod detekcije napak. Rezultati validacije bodo vključeni v naslednje poročilo, kjer bo predstavljena celovita analiza delovanja sistema v realnem okolju in primerjava z izhodiščnimi pričakovanji.