

# Integracija metoda strojnog vida i učenja za automatiziranu detekciju, lokalizaciju i verifikaciju mikroprocesorskih pločica

---

Šiktar, Luka

Master's thesis / Diplomski rad

2024

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:235:428168>

Rights / Prava: [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2024-09-18**

Repository / Repozitorij:

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



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# DIPLOMSKI RAD

Luka Šiktar

Zagreb, 2024.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# DIPLOMSKI RAD

Mentor:

Doc. dr. sc. Filip Šuligoj, mag. ing. mech.

Student:

Luka Šiktar

Zagreb, 2024.

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

Zahvaljujem se mentoru doc. dr. sc. Filipu Šuligoju na pruženoj pomoći i savjetima prilikom izrade diplomskog rada.

Zahvaljujem se svojoj obitelji, kolegama i prijateljima na podršci, korisnim savjetima i svim lijepim trenucima tijekom studiranja.

Luka Šiktar



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

Središnje povjerenstvo za završne i diplomske ispite

Povjerenstvo za diplomske ispite studija strojarstva za smjerove:

Proizvodno inženjerstvo, inženjerstvo materijala, industrijsko inženjerstvo i menadžment,  
mehatronika i robotika, autonomni sustavi i računalna inteligencija



Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 - 04 / 24 - 06 / 1	
Ur.broj: 15 - 24 -	

## DIPLOMSKI ZADATAK

Student: **Luka Šiktar**

JMBAG: 0035219285

Naslov rada na hrvatskom jeziku: **Integracija metoda strojnog vida i učenja za automatiziranu detekciju, lokalizaciju i verifikaciju mikroprocesorskih pločica**

Naslov rada na engleskom jeziku: **Integration of machine vision and learning methods for automated detection, localization, and verification of microprocessor boards**

Opis zadatka:

U kontekstu brzog razvoja tehnologije i masovne proizvodnje sofisticiranih elektroničkih komponenti, kontrola kvalitete ostaje izazovna i ekonomski zahtjevna. Posebno je strojni vid, koji koristi kamere za prikupljanje informacija, postao presudan alat u industrijskim kontrolnim procesima. U ovom radu povežujemo strojni vid i strojno učenje, konkretno neuronskih mreža, za kontrolu kvalitete kod mikroprocesorskih pločica.

Specifični zadaci rada obuhvaćaju:

1. Akviziciju slika navedenih mikroprocesorskih pločica pod različitim okolnim uvjetima koristeći industrijsku kameru.
2. Stvaranje seta podataka za treniranje neuronske mreže.
3. Primjenu YOLO (eng., You Only Look Once) neuronske mreže i algoritama strojnog vida za detekciju, lokalizaciju i verifikaciju elemenata na različitim mikroprocesorskim pločicama kao što su Arduino UNO, Jetson NX Xavier, Jetson Nano, Raspberry Pi, STM32 i UP board.
4. Analizu i evaluaciju performansi modela u različitim uvjetima, fokusirajući se na robusnost modela u kontekstu varijacija u orijentaciji, oštećenjima objekata i uvjetima osvjjetljenja.

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

Zadatak zadan:

16. studenoga 2023.

Datum predaje rada:

18. siječnja 2024.

Predviđeni datumi obrane:

22. – 26. siječnja 2024.

Zadatak zadao:

Doc.dr.sc. Filip Šuligoj

Predsjednik Povjerenstva:

Prof. dr. sc. Ivica Garašić

## SADRŽAJ

SADRŽAJ .....	I
POPIS SLIKA .....	III
POPIS TABLICA.....	V
POPIS KRATICA .....	VI
SAŽETAK.....	VII
SUMMARY .....	VIII
1. UVOD.....	1
2. Opis problema.....	2
3. Metodologija razvijenog rješenja .....	5
3.1. Konstrukcija postava za kontrolu kvalitete .....	6
3.2. Stvaranje seta podataka za trening neuronske mreže.....	6
3.3. Treniranje YOLOv8m modela konvolucijske neuronske mreže .....	6
3.4. Detekcija .....	7
3.5. Inspekcija .....	7
3.6. Optičko prepoznavanje teksta .....	7
3.7. Razvoj GUI-ja.....	8
4. You Only Look Once model.....	9
4.1. YOLOv8.....	11
4.2. Metrike YOLOv8 modela .....	12
4.3. Arhitektura YOLOv8 modela .....	14
5. Postav za kontrolu kvalitete mikroprocesorskih pločica .....	16
6. Proces izrade seta podataka za treniranje neuronske mreže .....	19
6.1. Mikroprocesorske pločice korištene za izradu seta podataka .....	19
6.1.1. Arduino Uno R3.....	19
6.1.2. Raspberry Pi 3.....	21
6.1.3. Jetson NX Xavier Development Kit .....	22
6.1.4. Jetson Nano Development Kit .....	24
6.1.5. STM32 .....	25
6.1.6. Up Board.....	26
6.2. Označavanje slika korištenjem Roboflow-a .....	28
7. Treniranje YOLOv8 modela.....	32
8. Detekcija i izdvajanje detektiranih komponenata.....	36
9. Inspekcija.....	39
10. Optičko prepoznavanje teksta.....	43
11. Implementacija sustava.....	44
12. Rezultati.....	46
12.1. Arduino Uno .....	46

---

12.2. Raspberry Pi 3 .....	49
12.3. Jetson NX Xavier .....	51
12.4. Jetson Nano .....	53
12.5. Up board.....	56
12.6. STM32 .....	58
13. ZAKLJUČAK.....	60
LITERATURA.....	61
PRILOZI .....	63

## POPIS SLIKA

Slika 1.	Automatska Optička Inspekcija [4].....	2
Slika 2.	Završni proces inspekcije u proizvodnji Raspberry Pi 3 korištenjem COGNEX sustava kamera [10].....	3
Slika 3.	Metodologija predstavljenog rješenja .....	5
Slika 4.	Primjer konvolucije s korakom pomaka 1 elementa .....	9
Slika 5.	Primjer <i>pooling</i> -a s korakom pomaka 2 elementa.....	10
Slika 6.	Detekcije korištenjem YOLO mreže[9] .....	11
Slika 7.	Prikaz objekta te primjera stvarne i detektirane ograničavajuće kućice.....	13
Slika 8.	Intersection over Union izračun .....	13
Slika 9.	Arhitektura YOLOv8 modela[15] .....	14
Slika 10.	LUCID PHX064S-CC industrijska kamera [16].....	16
Slika 11.	Edmund Optics objektiv [17] .....	17
Slika 12.	Postav industrijske kamere i osvjetljenja .....	17
Slika 13.	Arena SDK .....	18
Slika 14.	Arduino Uno.....	20
Slika 15.	Raspberry Pi 3 .....	21
Slika 16.	Jetson NX Xavier .....	23
Slika 17.	Jetson Nano .....	24
Slika 18.	STM32.....	26
Slika 19.	Up board .....	27
Slika 20.	Označena Raspberry Pi 3 pločica .....	28
Slika 21.	Primjeri označavanja komponenti na setu podataka .....	29
Slika 22.	Raspberry Pi 3 set podataka nakon augmentacije .....	31
Slika 23.	Rezultati trening procesa .....	34
Slika 24.	Detekcije u različitim orijentacijama .....	36
Slika 25.	Robusnost YOLOv8 modela na razlike unutar klase detektiranog objekta .....	36
Slika 26.	Detekcija i izdvajanje komponenti na prednjoj strani Raspberry Pi 3 mikroprocesorske pločice.....	37
Slika 27.	Detekcija i ekstrakcija komponenti na stražnjoj strani Raspberry Pi 3 mikroprocesorske pločice.....	38
Slika 28.	Arduino Uno: a) Primjeri korištenja THRESHOLD_BINARY metode, b) primjeri korištenja THRESHOLD_BINARY_INV metode .....	40
Slika 29.	Metodologija inspekcije .....	41
Slika 30.	Inspekcija na stražnjoj strani Jetson NX Xavier .....	42
Slika 31.	Elementi za ispitivanje optičkim prepoznavanjem teksta (a) STM32, b) Jetson Nano, c) i d) Jetson Xavier, e) Raspberry Pi 3, f), g) i h) Arduino Uno .....	43
Slika 32.	Rezultati optičkog prepoznavanja teksta (a) Raspberry Pi 3 model, b) STM32 model, c) Arduino Uno model.....	43
Slika 33.	GUI za kontrolu kvalitete mikroprocesorskih pločica, primjer Arduino Uno.....	44
Slika 34.	Rezultati detekcije komponenti na Arduino Uno mikroprocesorskoj pločici .....	47
Slika 35.	Rezultati inspekcije i optičkog prepoznavanja teksta komponenti na Arduino Uno mikroprocesorskoj pločici .....	48
Slika 36.	Rezultati detekcije komponenti na Raspberry Pi 3 mikroprocesorskoj pločici....	49
Slika 37.	Rezultati inspekcije i optičkog prepoznavanja teksta komponenti na Raspberry Pi 3 mikroprocesorskoj pločici .....	50
Slika 38.	Rezultati detekcije komponenti na Jetson NX Xavier mikroprocesorskoj pločici .....	51



---

Slika 39.	Rezultati inspekcije i optičkog prepoznavanja teksta na Jetson NX Xavier mikroprocesorskoj pločici .....	53
Slika 40.	Rezultati detekcije na Jetson Nano mikroprocesorskoj pločici.....	54
Slika 41.	Rezultati inspekcije i optičkog prepoznavanja teksta na Jetson Nano mikroprocesorskoj pločici .....	55
Slika 42.	Rezultati detekcije na Up Board mikroprocesorskoj pločici.....	56
Slika 43.	Rezultati inspekcije na Up Board mikroprocesorskoj pločici .....	57
Slika 44.	Rezultati detekcije na STM32 mikroprocesorskoj pločici .....	58
Slika 45.	Rezultati inspekcije i optičkog prepoznavanja teksta na STM32 mikroprocesorskoj pločici .....	59

---

**POPIS TABLICA**

Tablica 1. Označene komponente na Arduino Uno mikroprocesorskoj pločici.....	20
Tablica 2. Označene komponente na Raspberry Pi 3 mikroprocesorskoj pločici.....	22
Tablica 3. Označene komponente na Jetson NX Xavier mikroprocesorskoj pločici .....	23
Tablica 4. Označene komponente na Jetson Nano mikroprocesorskoj pločici .....	25
Tablica 5. Označene komponente na STM32 mikroprocesorskoj pločici.....	26
Tablica 6. Označene komponente na Up board mikroprocesorskoj pločici.....	27
Tablica 7. Trening hiperparametri .....	33
Tablica 8. Trening rezultati za klasu <i>40_pins</i> .....	33
Tablica 9. Trening rezultati .....	35
Tablica 10. Rezultati detekcije nad Arduino Uno mikroprocesorskoj pločici .....	47
Tablica 11. Rezultati detekcije nad Raspberry Pi 3 mikroprocesorskoj pločici.....	50
Tablica 12. Rezultati detekcije nad Jetson NX Xavier mikroprocesorskoj pločici.....	52
Tablica 13. Rezultati detekcije nad Jetson Nano mikroprocesorskoj pločici.....	54
Tablica 14. Rezultati detekcije nad Up Board mikroprocesorskoj pločici.....	57
Tablica 15. Rezultati detekcije nad STM32 mikroprocesorskoj pločici .....	59

**POPIS KRATICA**

---

<b>Kratika</b>	<b>Opis</b>
PCB	Printed Circuit Board
SBC	Single Board Computer
OCR	Optical Character Recognition
GPU	Graphical Processing Unit
AOI	Automated Optical Inspection
YOLO	You Only Look Once
COCO	Common Objects in Context
CNN	Convolutional Neural Network
R-CNN	Region-based Convolutional Neural Network
FPS	Frames Per Second
PoE	Power over Ethernet
GUI	Graphical User Interface
mAP	mean Average Precision
IoU	Intersection over Union
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
NMS	Non-Maximum Suppression
SDK	Software Development Kit

---

**SAŽETAK**

Kontrola kvalitete mikroprocesorskih pločica predstavlja krajnji korak u proizvodnom ciklusu mikroprocesora. Ona značajno utječe na vrijeme proizvodnje, kvalitetu finalnog proizvoda i njegovu konačnu cijenu. U vrijeme konstantnog tehnološkog napretka, potiče se razvoj inovativnih metoda kontrole kvalitete koje osim što osiguravaju točnost, ubrzavaju proizvodni proces, čime on postaje ekonomski isplativiji. Jedan od smjerova razvoja kontrole kvalitete je uključivanje naprednih tehnologija i alata kao što su računalni vid i duboko učenje u proces. Ovaj rad predstavlja sustav za automatiziranu detekciju, lokalizaciju, verifikaciju i inspekciju mikroprocesorskih pločica, korištenjem metoda računalnog vida i učenja. Sustav se sastoji od nekoliko segmenata: detekcija i verifikacija mikroprocesorskih pločica i njihovih komponenti korištenjem YOLOv8 konvolucijske neuronske mreže, segmentiranje i izdvajanje slika detektiranih komponenti mikroprocesora za provođenje inspekcije lemljenih spojeva i pinova korištenjem metoda adaptivnog praga (engl. adaptive threshold) i optičko prepoznavanje teksta. Sustav kombinira inovativne tehnologije kako bi omogućio točnost i brzinu kontrole kvalitete mikroprocesorskih pločica.

Cilj rada je konstruirati inovativni sustav industrijske primjene računalnog vida za adaptivnu kontrolu kvalitete mikroprocesorskih pločica. Razvoj kompletnog sustava uključuje izradu seta podataka mikroprocesorskih pločica za treniranje YOLOv8 konvolucijske neuronske mreže, trening neuronske mreže korištenjem posebnih alata te razvoj implementacije sustava za detekciju, verifikaciju, inspekciju i optičko prepoznavanje teksta. Trenirana YOLOv8 konvolucijska neuronska mreža prema *mean Average Precision* (mAP) metrici postiže rezultat od 99.53% mAP, inspeksijski sustav s adaptivnim pragom postiže preciznost (engl. precision) od 95,9%, a pokrivenost (engl. recall) iznosi 97,87%. Optičko prepoznavanje teksta postiže 95.64% točno prepoznavanje teksta.

**Ključne riječi:** Računalni vid, strojno učenje, mikroprocesorske pločice, YOLOv8, adaptivni *threshold*, optičko prepoznavanje teksta

---

**SUMMARY**

The quality control of microprocessor boards, Single Board Computers (SBCs), or Printed Circuit Boards (PCBs) is the final and crucial segment of microprocessor manufacturing. Quality control has a significant influence on final production time and product pricing. Nowadays, there is an imperative to develop solutions that speed up quality control and make it more practical and economical. This research presents automated detection, localization, verification, and inspection of microprocessor boards using computer vision and deep learning. The proposed system encompasses several modules: microprocessor board detection and component verification using YOLOv8 convolutional neural network, targeted components image extraction for solder joint quality assessment via adaptive thresholding and optical character recognition (OCR) on selected images. The main goal is to develop industrial computer vision application for adaptable quality control.

The system development involved SBCs and PCBs dataset creation for training YOLOv8 convolutional neural network, YOLOv8 train process using PyTorch, and development of final implementation that consists of solder joint inspections and OCR. The trained YOLOv8 neural network achieves Mean Average Precision (mAP) of 99.53%, the proposed inspection system scores 95,9% precision and 97,87% recall. The OCR model scores 95.64% true positive character predictions.

Key words: Computer vision, Deep learning, SBC, PCB, adaptive thresholding, YOLOv8, OCR

## 1. UVOD

Mikroprocesorske pločice, SBC-ovi (engl. Single Board Computer) i PCB-ovi (engl. Printed Circuit Board) su ključni elementi svakog elektroničkog uređaja. Razvojem tehnologije i porastom broja elektroničkih uređaja raste i potražnja za mikroprocesorskim pločicama. Tržište mikroprocesorskih pločica, posebno PCB-ova, predviđa rast s 86.18 milijardi dolara u 2023. godini na 140.74 milijardi dolara u 2032. godini [1]. Tijekom razvoja, mikroprocesorske pločice su postale sve kompaktnije i kompleksnije, a proizvodnja kompletno automatizirana i brza, dok se za kontrolu kvalitete koriste nefleksibilna i ekonomski neučinkovita rješenja. Cilj ovog istraživanja je izrada rješenja koje koristi algoritme računalnog vida, kao što su tehnike obrade slike te konvolucijske neuronske mreže za detekciju, lokalizaciju, verifikaciju i inspekciju mikroprocesorskih pločica. Primjena računalnog vida za dani problem doprinosi fleksibilnosti kontrole kvalitete kako bi se mogla primjenjivati na veći raspon modela mikroprocesorskih pločica.

Uparivanje tehnika obrade slike s konvolucijskim neuronskim mrežama dovodi do unaprjeđenja sustava kombiniranjem prednosti i uklanjanjem nedostataka navedenih metoda. Konvolucijske neuronske mreže doprinose generalizaciji prilikom detekcije, lokalizacije i verifikacije, a metode obrade slike omogućuju egzaktnu i preciznu inspekciju detektiranih i izdvojenih objekata. Razvoj kompletnog sustava se sastoji od izrade seta podataka za treniranje YOLOv8 modela, treninga samog modela te implementacije finalnog rješenja sustava, odnosno detekcije mikroprocesorskih pločica te inspekcije i optičkog prepoznavanja teksta na izdvojenim komponentama mikroprocesora. Set podataka i trening proces modela osiguravaju generalizaciju uz traženu točnost detekcije koja je jednaka točnosti detekcije dosad korištenih pristupa kao što je *template-matching* metoda [2]. Obrada slike se koristi kod inspekcije koja zahtijeva veliku preciznost i egzaktnost rezultata. Optičko prepoznavanje teksta, OCR (engl. Optical Character Recognition) također koristi neuronske mreže za prepoznavanje teksta i osiguravanje dodatne analize mikroprocesorskih pločica.

Za potrebe eksperimentalne analize i eksploatacije sustava, konstruiran je postav s industrijskom kamerom i odgovarajućim industrijskim osvjetljenjem. Predloženi sustav daje kompletno (engl. end-to-end) rješenje kontrole kvalitete mikroprocesorskih pločica koje se lako prilagođava zahtjevima proizvodnog procesa.

## 2. Opis problema

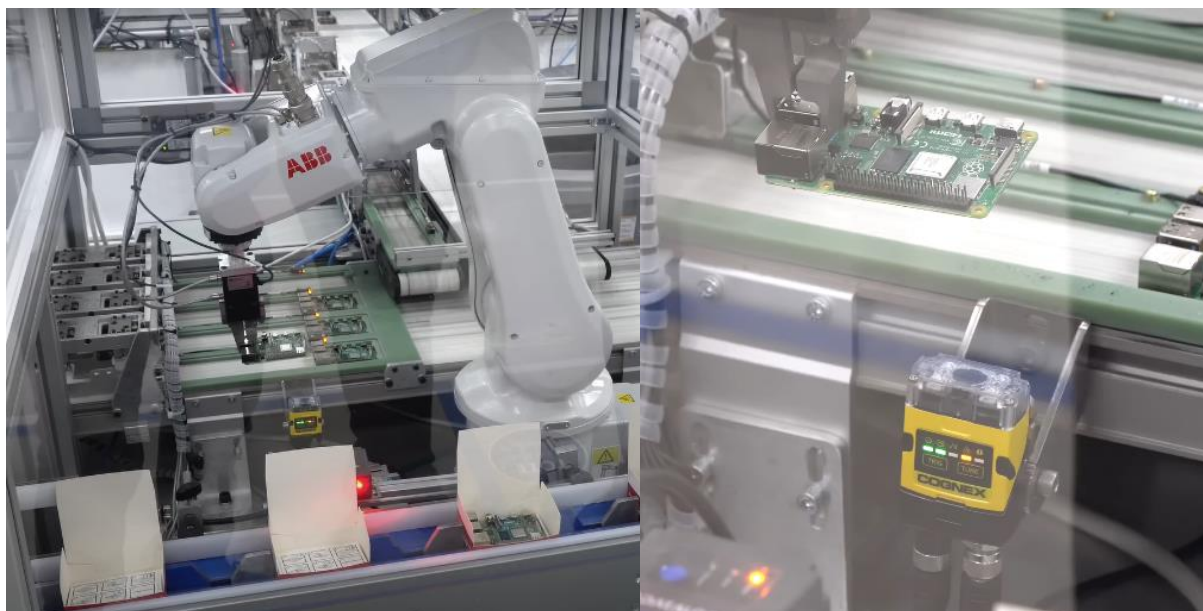
Tehnologije proizvodnje mikroprocesorskih pločica su sofisticirane i ekonomski prihvatljive za razliku od kontrole kvalitete proizvoda koja je sporija i nedovoljno precizna zbog dominantnog utjecaja ljudske radne snage u samom procesu. Proizvodnja je u potpunosti automatizirana korištenjem robotskih sustava uz specijalizirane strojeve zadužene za pojedini proces proizvodnje. Između svakog koraka proizvodnje je potrebno provesti ispitivanja radi utvrđivanja kvalitete mikroprocesorskih pločica. U industriji se u procesima kontrole kvalitete između pojedinih procesa unutar proizvodnje mikroprocesorskih pločica koriste brojne metode, a najznačajnija je Automatska Optička Inspekcija, AOI (engl. Automatic Optical Inspection) [3], koja se koristi za inspekciju kvalitete lemljenih spojeva različitih komponenti na pločici. Ova metoda je izrazito točna, ali skupa i nefleksibilna na različitosti ispitanih mikroprocesora. Metoda koristi kameru koja snima određene segmente izrađene mikroprocesorske pločice, zatim te segmente kombinira sa setom ispravnih slika te uspoređuje rezultate i detektira pogreške. Mjesta na mikroprocesorskoj pločici gdje su detektirane pogreške se podvrgavaju dodatnoj kontroli radnika, što doprinosi nesigurnosti i produljuje vrijeme ispitivanja. Navedeni nedostaci motiviraju istraživače na razvoj rješenja koja koriste metode računalnog vida i učenja prilikom kontrole kvalitete.



**Slika 1. Automatska Optička Inspekcija [4]**

Računalni vid predstavlja korištenje tehnologije, posebice kamera, za prikupljanje informacija o okolini. Spojem računalnog vida i strojnog učenja nastaje strojni vid, pomoću kojeg je moguće učenje i prepoznavanje, segmentiranje i klasificiranje informacija o okolini.

Kako bi se smanjili troškovi kontrole kvalitete AOI uređajima, moderna rješenja koriste računalni vid u obliku obrade slike u kombinaciji s naprednim algoritmima pomoću kojih se dobivaju informacije koje omogućuju automatiziranu kontrolu kvalitete. Neke od metoda su *image subtraction* i detekcija blobova u procesu određivanja oštećenja i nedostataka [5]. *Template matching* se koristi za detekciju objekata na mikroprocesorskim pločicama, u slučaju postojanja seta podataka koji se koristi kao *template* [2]. Detekcija oštećenja na mikroprocesorskim pločicama se može također provoditi korištenjem Otsu praga i Houghovih transformacija [6]. Navedene metode su točne i precizne ali nisu dovoljno fleksibilne pa se kombiniraju s umjetnom inteligencijom, točnije konvolucijskim neuronskim mrežama koje imaju sposobnost generalizacije, čime doprinose robusnosti sustava. Sposobnost generalizacije i povećanje fleksibilnosti je ključno za implementaciju inovativnih rješenja kontrole kvalitete mikroprocesorskih pločica jer omogućava jedinstveni sustav za više procesa tijekom kontrole, za razliku od trenutnih sustava sa specijaliziranim i strogo ograničenim opsegom rada. U brojnim istraživanjima se razvijaju konvolucijske neuronske mreže i setovi trening podataka za različite zadatke kontrole kvalitete mikroprocesorskih pločica. Neke od razvijenih neuronskih mreža za problem detekcije i inspekcije mikroprocesora, posebno PCB-ova i njihovih komponenti su ECLAD-Net [7], ResNeSt [8] i You Only Look Once (YOLO) [9]. Potencijal korištenja konvolucijskih neuronskih mreža u kombinaciji s računalnim vidom daje brojne mogućnosti za daljnju inspekciju mikroprocesorskih pločica ali i drugih područja.



**Slika 2. Završni proces inspekcije u proizvodnji Raspberry Pi 3 korištenjem COGNEX sustava kamera [10]**

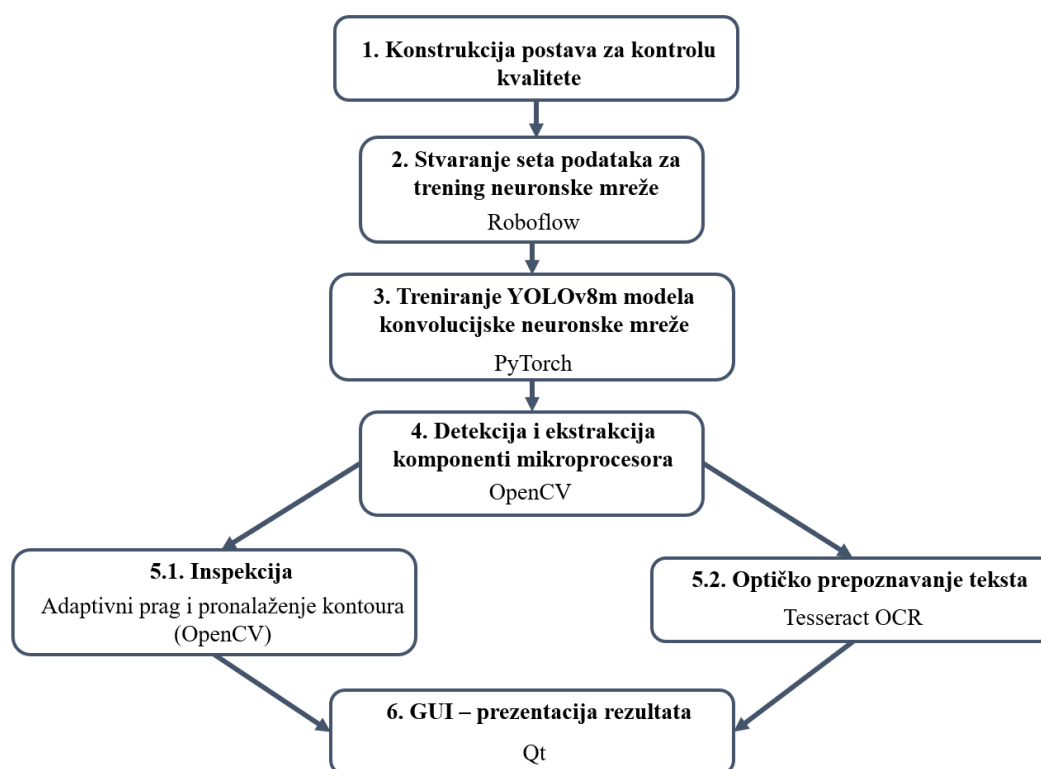


U ovom radu je razvijeno rješenje primjene strojnog vida za detekciju, lokalizaciju, verifikaciju i inspekciju mikroprocesorskih pločica, odnosno SBC-ova i PCB-ova, koje je razrađeno u pojedine zadatke.

Prije razvoja rješenja, potrebno je konstruirati postolje za kameru i industrijsko osvjetljenje kako bi se mogle snimiti slike koje će se koristiti u stvaranju seta podataka te kako bi se na samom kraju mogla provesti evaluacija sustava. Prvi zadatak je stvaranje vlastitog seta podataka koji uključuje šest najznačajnijih mikroprocesorskih pločica za industrijsku primjenu. Stvoreni set podataka se u drugom dijelu zadatka koristi za treniranje YOLOv8 modela za detekciju objekata. Detekcija objekata je neophodan proces kako bi se segmentirale i izdvojile komponente mikroprocesorske pločice, te kako bi se provela daljnja kontrola kvalitete. Treći zadatak se sastoji od razvoja rješenja inspekcije pinova i lemljenih spojeva primjenom algoritama računalnog vida zbog dokazane točnosti i preciznosti. U četvrtom zadatku se provodi optičko prepoznavanje teksta konvolucijskom neuronskom mrežom kao završna kontrola detekcije.

### 3. Metodologija razvijenog rješenja

Metodologija kontrole kvalitete mikroprocesorskih pločica se dijeli u nekoliko procesa. Za početak je potrebno konstruirati postav s kamerom i odgovarajućim industrijskim osvjetljenjem koje će se koristiti za akviziciju slike. Nad prikupljenim slikama je potrebno provesti detekciju i lokalizaciju kako bi se odredila mikroprocesorska pločica koja se kontrolira te kako bi se detektirale sve tražene komponente. Dijelovi slika s detektiranim komponentama se zatim segmentiraju i izdvajaju iz slika kako bi se na njima provela inspekcija. Inspekcija se provodi na ulazno/izlaznim komponentama, odnosno pinovima, te lemljenim spojevima, dokumentiranjem broja ispitanih elemenata te filtriranjem po veličini kako bi se analizirala njihova kvaliteta. Na komponentama koje sadrže ispisana imena SBC-ova i PCB-ova i njihovih modela i serijskih brojeva, provest će se optičko prepoznavanje teksta. Slika 3. prikazuje metodologiju predstavljenog rješenja te alate koji se koriste u svakome koraku.



Slika 3. Metodologija predstavljenog rješenja

U nastavku će biti opisan glavni zadatak svakog procesa unutar predstavljenog rješenja te njegov značaj za kompletni sustav kontrole kvalitete.

### **3.1. Konstrukcija postava za kontrolu kvalitete**

Početni korak razvoja sustava kontrole kvalitete je izrada postava za akviziciju slika u različitim okolnim uvjetima korištenjem industrijske kamere. Kompletna konstrukcija se mora sastojati od modularnog postolja na kojemu se nalaze industrijska kamera, pripadno osvjetljenje te pripadne komponente kao što su Power over Ethernet modul i napajanje za osvjetljenje. Potrebno je odabrati odgovarajuću industrijsku kameru i pripadni objektiv koji omogućuje namještanje otvora blende i fokusa. Također osvjetljenje mora imati mogućnost finog podešavanja kako bi se sustav mogao analizirati pod raznim uvjetima osvjetljenja.

### **3.2. Stvaranje seta podataka za trening neuronske mreže**

U ovom radu se koristi pre-trenirana konvolucijska neuronska mreža koju je potrebno dodatno trenirati nad specijalnim setom podataka. Specijalni set podataka se u slučaju treniranja modela za detekciju mikroprocesorskih pločica sastoji od snimljenih i označenih slika koje se dijele u set za treniranje modela i set za evaluaciju rezultata treninga. U ovom koraku je cilj snimiti slike mikroprocesorskih pločica pri različitim uvjetima osvjetljenja i orijentacije te stvoriti set podataka za trening konvolucijske neuronske mreže. Model se stvara označavanjem slika primjenom Roboflow alata te primjenom metoda za umjetno povećanje seta koja se sastoje od transformacija i manipulacija podacima. Uobičajene metode uključuju umjetno stvorene promjene u položaju objekta na slici, rotaciju slike, promjene osvjetljenja i ekspozicije te zamućenja slike.

### **3.3. Treniranje YOLOv8m modela konvolucijske neuronske mreže**

U trećem koraku je potrebno dodatno trenirati model pre-trenirane konvolucijske neuronske mreže kako bi mogao obavljati zadatak detekcije mikroprocesorskih pločica. Pre-trenirani modeli neuronskih mreža su trenirani na velikim i često primjenjivanim setovima podataka te u svojim osnovnim oblicima omogućuju detekciju. Također, modeli pružaju brojne prednosti i robusnije performanse prilikom dodatnog treniranja modela za specifične zadatke strojnog vida. Pre-trenirani model s inicijalnim učenjem na raznolikom setu podataka već ima sposobnost učenja uzoraka za slikama, koja pomažu razumijevanju sadržaja slike, za razliku od modela koji se uči iz početka. Također, pre-trenirani model omogućuje bolju generalizaciju i sprečava pretreniranje modela. Generalizacija je sposobnost detekcije i klasifikacije modela u slučajevima različitih uvjeta osvjetljenja i orijentacija objekata. Navedena prednost proizlazi iz prijenosa znanja s inicijalnog, pre-treniranog modela. Pretreniranje modela je situacija u kojoj model trenira na trening podacima i postiže sve bolje performanse detekcije ili klasifikacije nad trening podacima, ali performanse u stvarnim slučajevima padaju.

Glavni cilj dodatnog treniranja modela konvolucijske neuronske mreže u ovom slučaju je fino podešavanje parametara modela kako bi se stvorio model za obavljanje specijalnog zadatka kao što je detekcija, lokalizacija i verifikacija mikroprocesorskih pločica i njihovih komponenti. Treniranje modela odvija se korištenjem specijaliziranog PyTorch alata.

### 3.4. Detekcija

Za proces detekcije i lokalizacije mikroprocesorskih pločica i njihovih komponenti, odabrana je upotreba konvolucijske neuronske mreže. Detekcija se u slučaju neuronskih mreža odnosi na proces prepoznavanja, odnosno predviđanja klase određene mikroprocesorske pločice ili zasebnih komponenti. Detekcija se vrši prepoznavanjem karakterističnih uzoraka objekta koji su naučeni tijekom trening procesa. Lokalizacija je proces u kojem se određuje točni položaj i granice detektiranog predmeta na slici. U postupku lokalizacije se koristi regresija kako bi se utvrdile točne koordinate ograničavajućih kućica (engl. bounding boxes) koje određuju granice objekta. Cilj korištenja konvolucijskih neuronskih mreža u predstavljenom istraživanju je detektiranje traženih mikroprocesorskih pločica i njihovih komponenti u različitim uvjetima osvjetljenja, orijentacije predmeta te kvalitete slike. Na taj način se postiže robusnost prilikom detekcije. Nakon procesa detekcije i lokalizacije se provodi izdvajanje određenih detektiranih elemenata za dodatnu inspekciju.

### 3.5. Inspekcija

U sljedećem koraku je potrebno razviti algoritam inspekcije pinova i lemljenih spojeva za potrebe kontrole kvalitete i dokumentaciju o stvarnom broju ispitanih elemenata. Inspekcija se provodi nad izdvojenim segmentima slike nakon detekcije. Proces inspekcije uključuje korištenje tehnike adaptivnog praga i pronalaženja kontura za precizno određivanje vanjskih rubova segmentiranih elemenata. Tehnika adaptivnog praga omogućuje precizno i robusno određivanje kontura te osigurava prilagodljivost rješenja različitim uvjetima osvjetljenja, omogućujući dodatno fino podešavanje sustava kako bi se postigla optimalna inspekcija. Broj kontura se bilježi kako bi se izvršila brojčana analiza traženog broja kontura te provelo ispitivanje kvalitete rada same metode na testnim primjerima mikroprocesorskih pločica, te omogućila kontrola u eksploataciji.

### 3.6. Optičko prepoznavanje teksta

Optičko prepoznavanje teksta se primjenjuje na izdvojenim segmentima slike korištenjem konvolucijskih neuronskih mreža, kako bi se identificirale i interpretirale tekstualne informacije na mikroprocesorskim pločicama. Ovaj proces se koristi za verifikaciju detekcije,

pri čemu se koristi javno dostupan trenirani model mreže bez dodatnih modifikacija. Verifikacija se odnosi na proces provjere točnosti detekcije. Model optičkog prepoznavanja je treniran na raznolikom setu podataka, na više jezika, a u ovom slučaju se primjenjuje na engleskom jeziku. Optičko prepoznavanje teksta služi kao dodatni korak analize koji pruža dodatnu sigurnost i pouzdanost u postupku kontrole kvalitete prepoznavanjem i dokumentiranjem informacija o svakoj mikroprocesorskoj pločici.

### **3.7. Razvoj GUI-ja**

Važan aspekt implementacije sustava kontrole kvalitete mikroprocesorskih pločica je grafičko korisničko sučelje, odnosno GUI, koji doprinosi fleksibilnosti, efikasnosti i jednostavnosti. GUI mora biti opremljen osnovnim funkcijama koje omogućuju korisniku upravljanje sustavom te jasan pregled rezultata kontrole kvalitete.

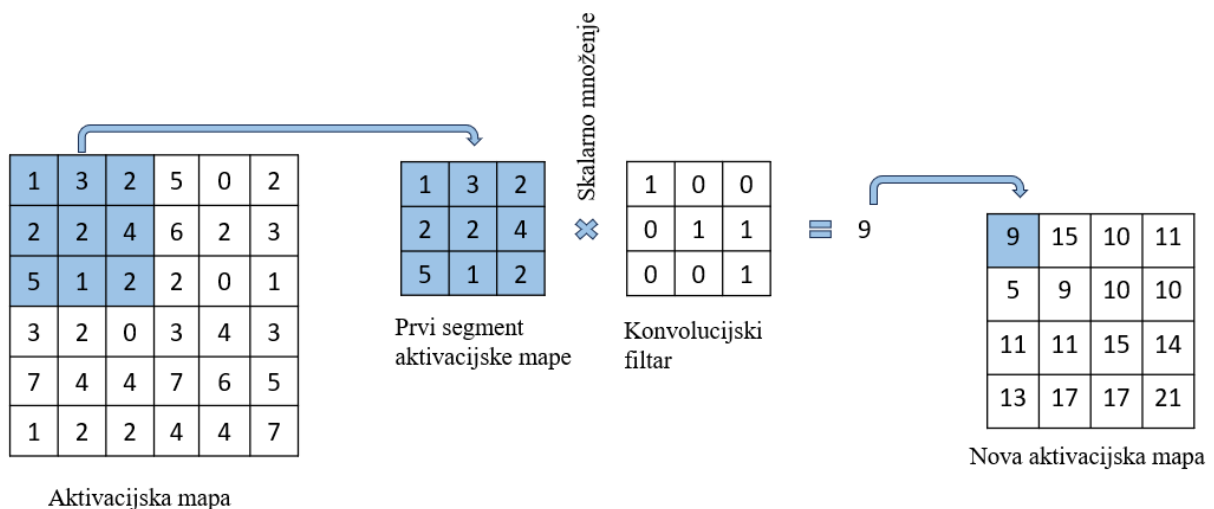
S obzirom na to da se radi o sustavu namijenjenom industrijskoj primjeni i uzimajući u obzir zahtjeve procesorskih moći kompleksnih konvolucijskih modela, odlučeno je da se sustav kontrole kvalitete izvede pomoću ulaznog signala, odnosno pritiska gumba za aktivaciju kompletnog sustava. Ovaj pristup omogućuje optimalno korištenje računalnih resursa, čime se postiže učinkovitost i smanjuju troškovi implementacije sustava. Sustav dakle omogućuje kontrolu kvalitete u realnom vremenu bez značajnog opterećenja procesorskih resursa osiguravajući primjenjivost sustava u industrijskom okruženju.

## 4. You Only Look Once model

*You Only Look Once* (YOLO) predstavlja model konvolucijske neuronske mreže koja se koristi za detekciju, klasifikaciju, segmentaciju i praćenje objekata na slikama i videozapisima. Kao i kod standardnih modela neuronskih mreža, konvolucijski model mreže uči uzorke na podacima, koji u ovom slučaju predstavljaju slike unutar trening seta podataka. YOLO konvolucijske neuronske mreže se razlikuju od standardnih neuronskih mreža po korištenju konvolucijskih, *pooling* i *fully connected*, odnosno potpuno povezanih slojeva [11]. Konvolucijski slojevi nad ulaznim podacima provode operaciju konvolucije kako bi model učio prepoznavanje uzoraka na slikama.

Kako bi slika bila razumljiva računalu potrebno ju je zapisati kao matricu dimenzija jednakih dimenzijama slike, gdje pojedini element matrice predstavlja pojedini piksel. Koordinate pojedinog piksela slike su jednake koordinatama elementa koji ga predstavlja. Svaki piksel može biti opisan s jednom vrijednosti ako se radi o jednokanalnim/crno-bijelim slikama ili s tri vrijednosti ako se radi o trokanalnim/RGB slikama. Piksel slike je transformiran u vrijednost unutar polja koja predstavlja intenzitet svjetline. U slučaju 8-bitnog zapisa slika za jednokanalnu sliku, svaki piksel može imati vrijednost od 0 za crnu boju minimalnog intenziteta ili 255 za bijelu boju maksimalnog intenziteta.

Ideja konvolucije je upotreba određenog filtra, odnosno kernela, oblika kvadrante matrice malih dimenzija na sliku. Filter se primjenjuje množenjem s određenim segmentom polja koje predstavlja sliku, element po element, kako je prikazano na slici 4. Filter je potrebno primijeniti na cijelu sliku, segment po segment, pomicanjem filtra za vrijednost koraka.

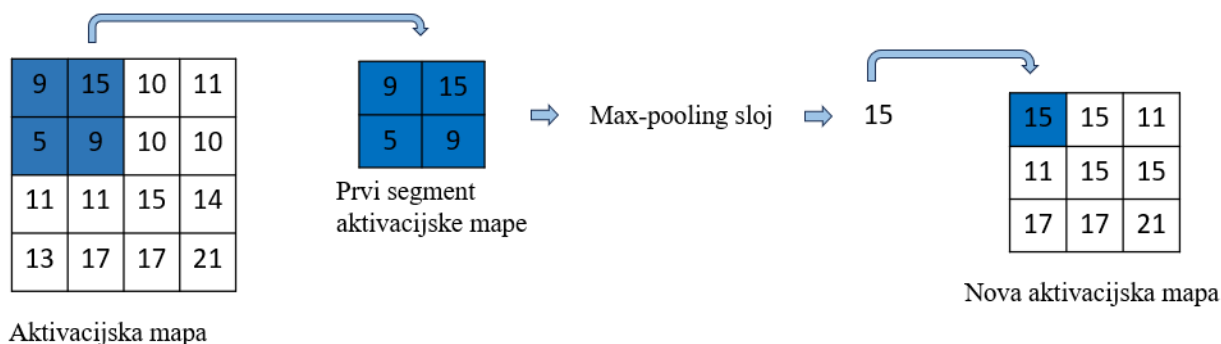


Slika 4. Primjer konvolucije s korakom pomaka 1 elementa

Rezultat konvolucije je 2D aktivacijska mapa, odnosno nova matrica koja sadrži željene uzorke i svojstva koje model uči. Aktivacijske mape se također mogu filtrirati pomoću drugih filtera. Iterativnim procesom konvolucije nad ulaznim podacima moguće je hijerarhijsko učenje uzoraka na slikama, počevši od jednostavnijih, a s rastom broja iteracija, prelazeći na sve kompleksnije uzorke.

Konvolucijski sloj se sastoji od više različitih filtera, pri čemu se koeficijenti unutar svake matrice filtera razlikuju. Tijekom treninga modela, koeficijenti se modificiraju kako bi svaki filter izdvajao specifične uzorke ili svojstva slike. Uzorci slike mogu biti linije, trokuti, kružnice ili kompleksniji oblici. Prolaskom filtera kroz cijelu aktivacijsku mapu, detektiraju se slični uzorci ili svojstva slike na različitim mjestima, odnosno konvolucijski sloj omogućuje detektiranje određenog uzorka primjenom određenog filtera.

Nakon konvolucijskog sloja dolazi *pooling* sloj koji smanjuje dimenzije 2D aktivacijske mape, a time i broj parametara modela. Najčešće se koristi *max-pooling* sloj kvadratnih dimenzija  $2 \times 2$  s korakom 2. Ovaj sloj uzima u obzir  $2 \times 2$  segment aktivacijske mape te zadržava samo maksimalnu vrijednost, provodeći isti postupak s korakom od 2 piksela dok ne filtrira cijelu aktivacijsku mapu. Konvolucijske neuronske mreže također uključuju potpuno povezane slojeve slične onima u standardnim neuronskim mrežama, a ti slojevi obično služe za klasifikaciju.

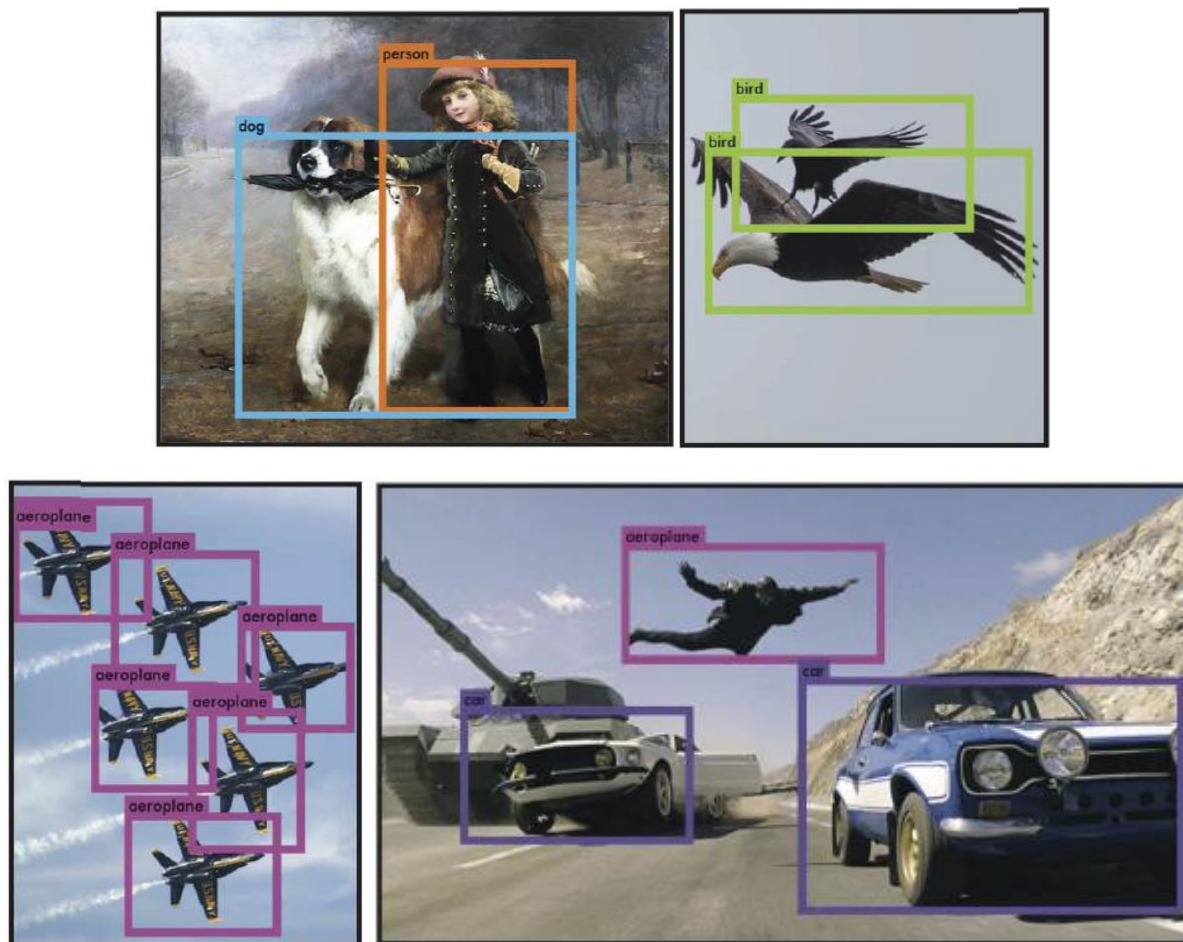


**Slika 5. Primjer *pooling*-a s korakom pomaka 2 elementa**

Standardna konvolucijska neuronska mreža ima niz konvolucijskih slojeva pri čemu početni slojevi izdvajaju jednostavna, a slojevi koji slijede izdvajaju kompleksnija svojstva sve do apstraktnih svojstava. Kompleksnost neuronske mreže definirana brojem slojeva i treniranih parametara, ovisi o faktorima kao što su broj filtera, upotreba *pooling*-a i veličina matrica filtera i *pooling*-a.

Prije pojave YOLO neuronskih mreža, za detekciju objekata koristili su se konvolucijski modeli poput R-CNN. Ti modeli stvaraju potencijalne ograničavajuće kućice i koriste

klasifikacijski model neuronske mreže nad ograničavajućim kućicama kako bi odredili klasu objekta unutar njih. Zatim se klasificirane kućice filtriraju kako bi za svaki objekt ostala po jedna kućica. Proces rada R-CNN modela je stoga spor i kompleksan. Razvojem YOLO konvolucijskih neuronskih mreža proces postaje jednostavniji jer ista konvolucijska neuronska mreža u jednom koraku određuje ograničavajuće kućice i provodi klasifikaciju objekata. To omogućava primjenu modela u realnom vremenu s visokom kvalitetom detekcije, nadmašujući kompleksnije modele i metode[9].



Slika 6. Detekcije korištenjem YOLO mreže[9]

Nadalje, prilikom detekcije korištenjem YOLO modela, ulazna slika nije podijeljena u segmente već se detekcija vrši na cijeloj slici, što dodatno ubrzava proces. YOLO modeli imaju veliku sposobnost generalizacije, što potiče primjenu modela na širok raspon problema [9].

#### 4.1. YOLOv8

YOLOv8 je novi, *state-of-the-art* konvolucijski model razvijen od strane Ultralytics-a korištenjem PyTorch alata za razvoj neuronskih mreža. Model je pre-treniran na *Common Objects in Context* (COCO) [12] setu podataka koji se koristi kao *benchmark* za treniranje i



testiranje konvolucijskih neuronskih mreža. COCO se sastoji od 330,000 slika od čega je 200,000 označenih u 80 različitih klasa. [12-13]. COCO koristi standardizirane metrike poput *mean Average Precision* (mAP) koja se koristi za procjenu kvalitete treniranja neuronskih mreža, u ovom slučaju detekcije objekata. Metrika mAP mjeri točnost i preciznost detekcije korištenjem većeg broja parametara dajući jedinstvenu ocjenu treninga koja omogućava usporedbu s drugim modelima. Metrika se temelji na *precision-recall* metrici, uspješnom prepoznavanju klasa objekata i definiranjem kvalitete ispravne procjene detekcije korištenjem *Intersection over Union* (IoU) metode[13].

#### 4.2. Metrike YOLOv8 modela

*Precision-recall* metrika definira kvalitetu rada modela korištenjem *Precision*, odnosno ocjenu preciznosti i *Recall*, ocjenu pokrivenosti. Da bi se metrike izračunale, rezultati detekcije modela moraju biti podijeljeni u četiri kategorije. U slučaju da model točno predvidi objekt, rezultat je točna pozitivna detekcija (engl. True Positive), TP. Ako model detektira objekt koji zapravo ne postoji, rezultat je pogrešna pozitivna detekcija (engl. False Positive), FP. Obratno, ako model ne detektira objekt gdje objekta niti nema, rezultat je točna negativna detekcija (engl. True Negative), TN. Na posljetku, ako model ne detektira objekt koji postoji, rezultat je pogrešna negativna detekcija (engl. False Negative), FN. Kombinacijom ovih rezultata detekcije, računaju se *Precision* i *Recall* [14]. *Precision* (1) se definira kao omjer broja točnih pozitivnih detekcija u odnosu na sve detekcije modela, dok se *Recall* (2) definira kao omjer broja točnih pozitivnih detekcija i ukupnog broja stvarnih pozitivnih detekcija.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

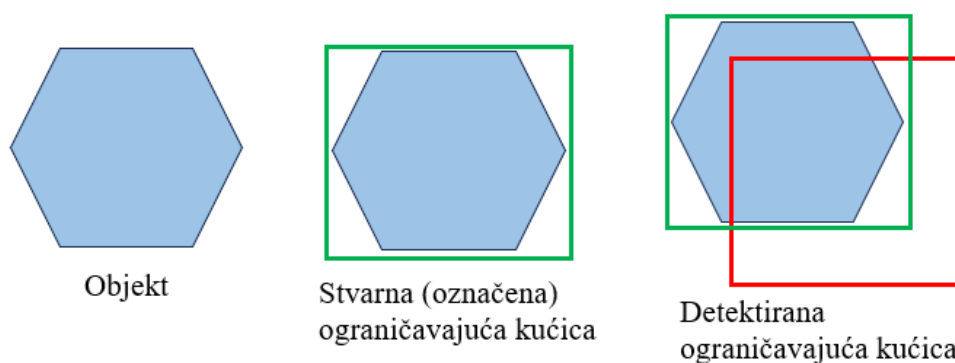
$$Recall = \frac{TP}{TP + FN} \quad (2)$$

Navedene vrijednosti su u suprotnosti, budući da omogućavanje većeg broja detekcija uzrokuje veći *recall*, ali *precision* tada pada i obrnuto. Metrika mAP predstavlja optimalni omjer danih vrijednosti i određuje performanse modela.

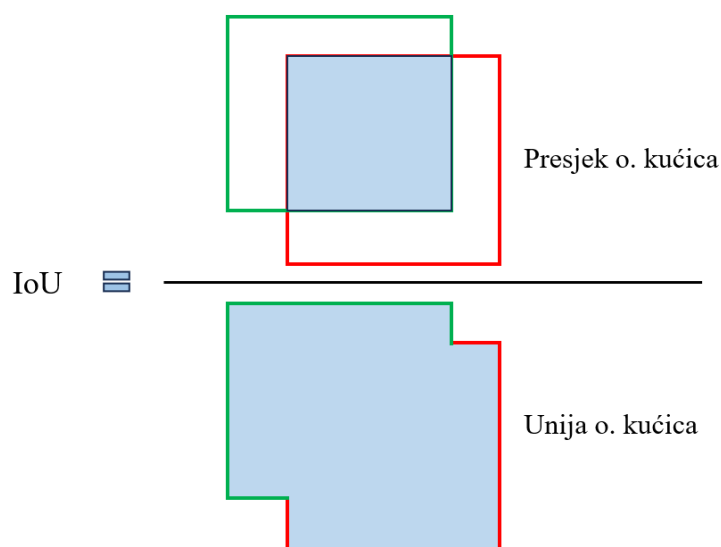
Prepoznavanje različitih klasa objekata (engl. Handling multiple object categories) se odnosi na sposobnost modela da identificira i lokalizira veći broj klasa objekata na slici. Za

svaku klasu se računa prosječna preciznost, a zatim se računa srednja vrijednost za sve klase zajedno.

Kvaliteta točne pozitivne procjene detekcije pomoću IoU metode prema izrazu (3) određuje sposobnost detekcije preciznih ograničavajućih kućica. Detekcija objekata ima zadatak lokalizirati određeni objekt postavljanjem ograničavajućih kućica oko objekta. IoU je omjer između presjeka i unije stvarnih i predviđenih ograničavajućih kućica. Definiranjem zahtjeva za veći IoU, smanjuje se broj detekcija, ali raste kvaliteta ograničavajućih kućica[12].



**Slika 7.** Prikaz objekta te primjera stvarne i detektirane ograničavajuće kućice

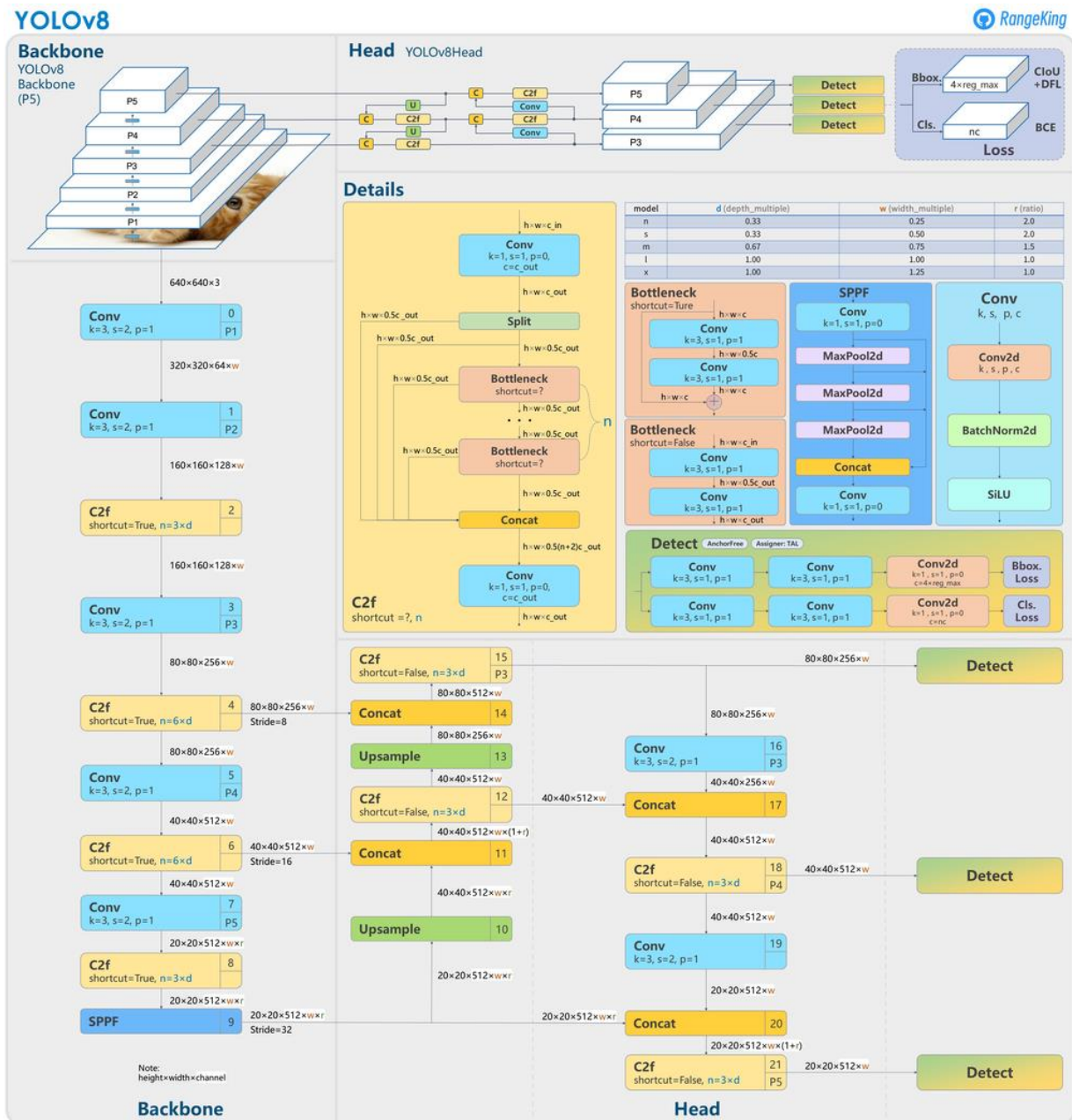


**Slika 8.** Intersection over Union izračun

$$IoU = \frac{\text{Presjek površina stvarne i detektirane ograničavajuće kućice}}{\text{Unija površina stvarne i detektirane ograničavajuće kućice}} \quad (3)$$

### 4.3. Arhitektura YOLOv8 modela

YOLOv8 model je podijeljen u tri segmenta: *backbone*, *neck* i *head*. Arhitektura modela je prikazana na slici 9.



Slika 9. Arhitektura YOLOv8 modela[15]

*Backbone* je segment modela pre-treniran na COCO setu podataka, a osnova su mu konvolucijski slojevi. Svaki konvolucijski sloj ima tri parametra: dimenziju matrice kernela/filtera koji se koristi kod konvolucije označenu s  $k$ , korak (engl. stride) pomicanja kernela kroz sliku označen sa  $s$  i ispunjenje rubova (engl. padding) koje predstavlja broj elemenata dodanih na rubove slike, označeno s  $p$ . Dimenzija kernela i korak pomicanja određuju omjer redukcije dimenzija slike, dok ispunjenje rubova omogućuje očuvanje

informacija na rubovima slike. YOLOv8 koristi novi *Backbone* segment pod nazivom “Darknet-53” koji se sastoji od 53 sloja i može klasificirati 1000 kategorija. “Darknet-53” omogućuje bržu i točniju detekciju od prethodnih *Backbone* modela kao što su ResNet ili ResNeXt. Također, model koristi veću mapu značajki te omogućuje konstrukciju kompleksnije veze među značajkama i bolje raspoznavanje uzoraka i objekata. Model također zahtijeva kraće vrijeme treniranja te je mogućnost pretreniranja manja [15].

*Neck* segment predstavljaju slojevi *Coarse to fine* (C2f) koji skupljaju i filtriraju značajke nakon prolaska aktivacijske mape kroz *Backbone*. Osim što sakupljaju značajke i uče uzorke na mapama značajki, C2f slojevi dodatno smanjuju reprezentaciju mapa značajki. *Neck* također koristi *Feature Pyramid Network* mehanizam koji varijacijama reprezentacije značajki, odnosno varijacijama vrijednosti u mapama značajki, prikuplja semantičke informacije o objektima te njihovu lokalizaciju na slici. Mehanizam omogućava prepoznavanje objekata različitih veličina korištenjem skala mapa značajki što je istovjetno stvaranju detekcija na slikama različitih veličina reprezentacije, gdje se predmet nalazi na različitim udaljenostima[15].

*Head* segment vrši detekciju i klasifikaciju objekata na slici. Radi se o više specifičnih podmreža i potpuno povezanih slojeva koji lokaliziraju i klasificiraju objekte na slici. Za razliku od prethodnih YOLO modela, *Head* segment YOLOv8 modela koristi *anchor-free* mehanizam prilikom određivanja ograničavajućih kućica. Model direktno određuje centar objekta, dimenzije ograničavajuće kućice i vjerojatnost da se unutar nje nalazi određena klasa objekta. Iako se ovo smatra inovacijom u odnosu na prethodne modele koji su koristili referentne, predefinirane setove ograničavajućih kućica, odnosno *anchor* mehanizam, sličan pristup je predstavljen u prvim iteracijama YOLO modela [12]. Ovaj direktni pristup smanjuje broj računskih operacija i broj ograničavajućih kućica koje detektiraju isti objekt. *Head* segment također koristi *Non-Maximum Suppression* (NMS) mehanizam koji filtrira ograničavajuće kućice kako bi za svaki objekt eliminirao sve kućice osim one s najvećom vjerojatnosti.

## 5. Postav za kontrolu kvalitete mikroprocesorskih pločica

Za prikupljanje slika seta podataka i provođenje eksperimenata tijekom razvoja i rada kompletnog sustava, konstruiran je postav za detekciju, lokalizaciju, verifikaciju i inspekciju mikroprocesorskih pločica koji koristi industrijsku kameru s pripadajućim industrijskim osvjetljenjem. Postav se sastoji od podloge te nosača kamere i osvjetljenja izrađenih od aluminijskih ploča i profila.

Na nosaču se nalazi *Lucid Vision Labs Phoenix PHX064S-CC* industrijska kamera. Radi se o kameri koja koristi Sony IMX178 CMOS optički senzor veličine 8.92mm (1/1.8"), rezolucije 3072\*2048 piksela, 6.4 MP i *buffer*-a od 128MB. Veličina piksela je 2.4 $\mu$ m\*2.4 $\mu$ m. Broj sličica po sekundi, odnosno FPS (engl. Frames Per Second) za maksimalnu rezoluciju iznosi 17.7. Kamera se povezuje s računalom RJ45 Ethernet kabelom kojim je i napajana *Power over Ethernet* (PoE) napajanjem. Uz kameru dolazi i pripadajući *Arena Software Development Kit* (SDK) koji omogućuje procesiranje slike, promjenu formata piksela, promjenu ekspozicije te potpunu kontrolu nad radom kamere [16].



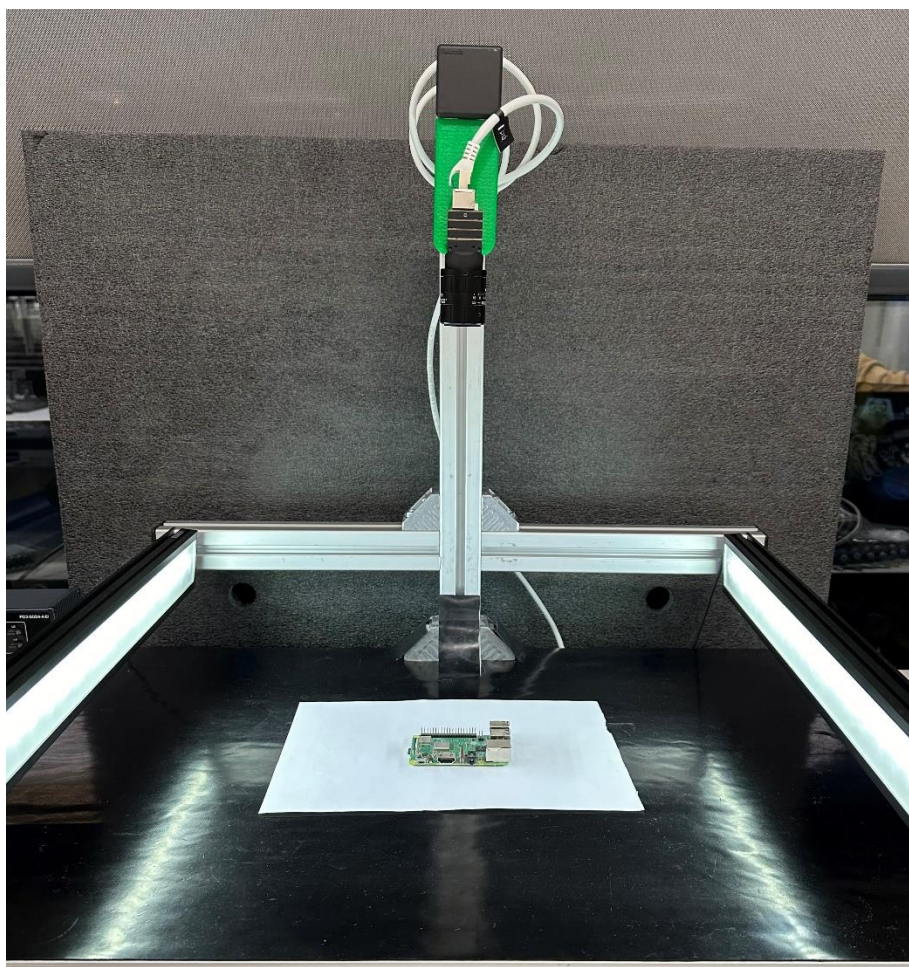
Slika 10. LUCID PHX064S-CC industrijska kamera [16]

Kamera koristi *Edmund Optics C-Mount 2/3"* 16mm f/1.6 (*C Series*) objektiv. *Edmund Optics C Series* objektivni su dizajnirani za automatiziranu industrijsku inspekciju korištenjem računalnog vida. Objektiv omogućava velike otvore leće čime se osiguravaju visoke performanse za uvjete slabog osvjetljenja. Objektiv ima leće s anti-reflektirajućim premazom, mogućnost namještanja otvora blende i fokusa. Žarišna duljina objektivna iznosi 16mm, s otvorom f/1.6. Vidno polje PHX064S-CC kameru s *C Series* objektivom iznosi 51.1mm, a pripadni kut iznosi 25.5° [17].



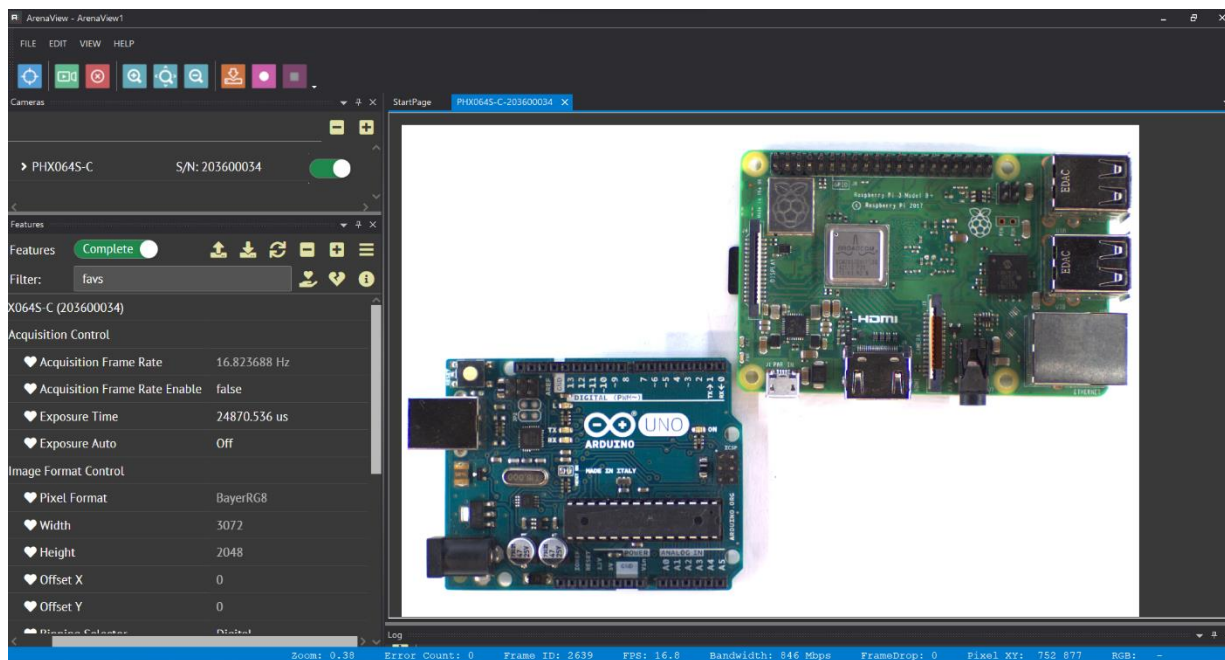
**Slika 11. Edmund Optics objektiv [17]**

Kako bi se postigli uvjeti osvjetljenja u industrijskim uvjetima rada koriste se dva HLDDL2-450X45SW-DF-W LED prilagodljiva osvjetljenja proizvođača CCS Inc.



**Slika 12. Postav industrijske kamere i osvjetljenja**

Osvjetljenje je postavljeno nisko na bočnim stranama predmeta kako bi se umanjio utjecaj refleksije na rezultate detekcije i inspekcije mikroprocesorskih pločica. Slika 13. prikazuje sliku kamere u Arena SDK. Prozor desno prikazuje vidno polje kamere konstruiranog postava, ispod su prikazani osnovni parametri kamere i statusna traka. Prozor lijevo omogućuje brzo podešavanje parametara kamere. Iako je moguće podešavanje parametara kamere u Arena SDK, parametri će se zbog jednostavnosti sustava podešavati iz skripte za akviziciju slike.



Slika 13. Arena SDK

## 6. Proces izrade seta podataka za treniranje neuronske mreže

Prvi korak, koji prethodi treniranju YOLOv8 modela konvolucijske neuronske mreže, je stvaranje velikog i preciznog seta podataka za trening, validaciju i testiranje. Trening podatci služe za učenje prepoznavanja uzoraka i svojstava na slici te čine najveći dio seta podataka. Validacijski podatci nisu dio procesa učenja modela, nego se koriste kod evaluacije modela tijekom treninga pomoću standardiziranih metrika. Test podatci također nisu uključeni u proces učenja, nego služe za evaluaciju na kraju treninga. Validacijski i test podatci obično čine 5-15% ukupnog seta podataka te su ključni za procjenu performansi modela i provjeru generalizacije na novim, modelu neviđenim podacima. Kvalitetan set podataka omogućuje treniranje modela s visokom sposobnošću generalizacije prilikom detekcije. Dodatno, precizno označavanje ograničavajućih kućica unutar seta podataka doprinosi preciznijim detekcijama i lokalizacijama tijekom testiranja i praktične primjene modela. Također, odgovarajuća veličina seta podataka ima ulogu u osiguravanju raznolikosti i smanjenju rizika od pretreniranja modela.

### 6.1. Mikroprocesorske pločice korištene za izradu seta podataka

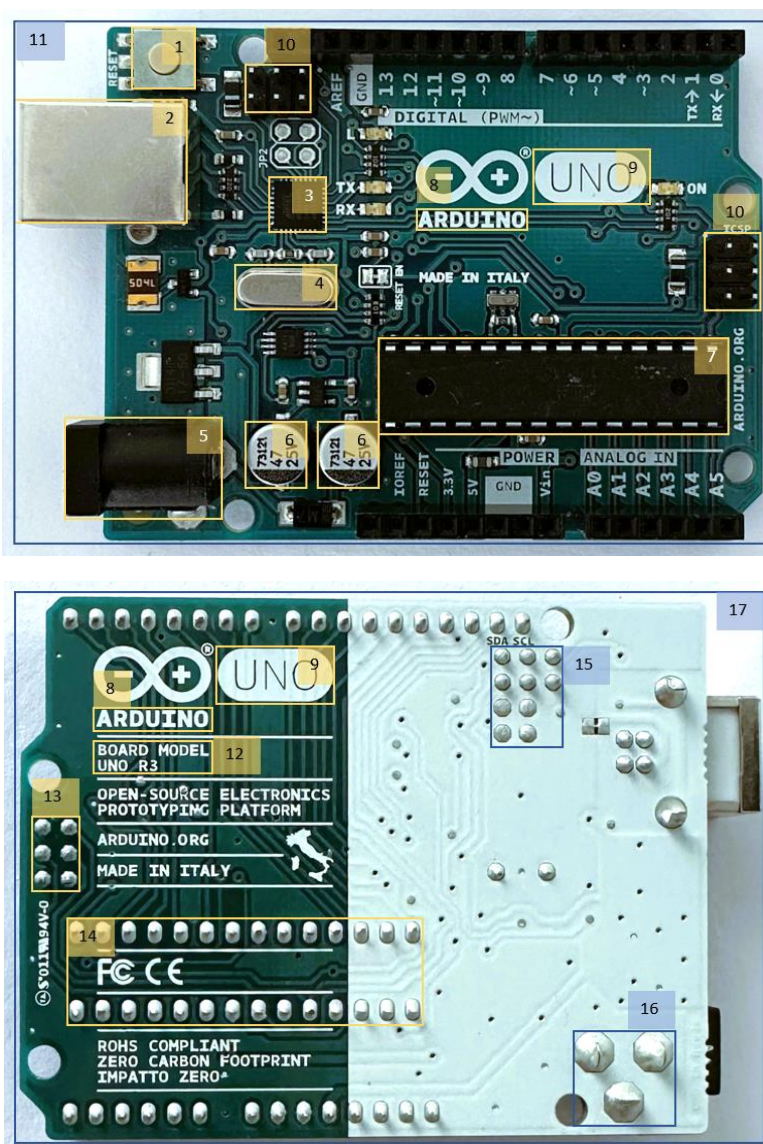
Za izradu seta podataka izabrane su mikroprocesorske pločice koje su prikazane u nastavku. Na tim pločicama su definirane komponente koje će se označavati prilikom procesa izrade seta podataka, te zatim detektirati, lokalizirati i verificirati tijekom rada modela. Glavni cilj je označiti komponente različitih veličina i oblika te komponente koje imaju istu funkciju, a različit vanjski izgled kako bi se provjerile sposobnosti trenirane neuronske mreže.

#### 6.1.1. *Arduino Uno R3*

Arduino Uno je *open-source* mikroprocesorska pločica opremljena s ATmega328P i ATmega 16U2 procesorom, 14 digitalnih I/O pinova, 6 analognih ulaza, USB konektorom, ICSP pinovima i Reset tipkom. Uno, najpoznatija mikroprocesorska pločica proizvođača Arduino.cc, koristi se za prototipiranje u edukacijskim i zadacima u industriji [18]. Na slici 14. su prikazane obje strane mikroprocesorske pločice s označenim i numeriranim elementima koji se označavaju prilikom stvaranja seta podataka.

Na prednjoj strani su označene komponente koje su standardnog oblika za sve mikroprocesorske pločice pa će se daljnja inspekcija provoditi samo na pinovima (*6\_pins*), dok su na stražnjoj strani označeni uzorci lemljenih spojeva (*Check\_pattern\_1- Check\_pattern\_4*) koji će se ispitivati. Također su na obje strane označena imena mikroprocesorske pločice (*ARDUINO, UNO\_white*) i modela (*Arduino\_UNO\_model*), koji će se koristiti za dodatnu kontrolu optičkim prepoznavanjem teksta.





Slika 14. Arduino Uno

Označena prednja i stražnja strana mikroprocesorske pločice te njegove komponente pobrojane su u tablici 1. Imena dodijeljena pojedinoj komponenti su preuzeta iz dostupnih korisničkih uputa za mikroprocesorsku pločicu [18].

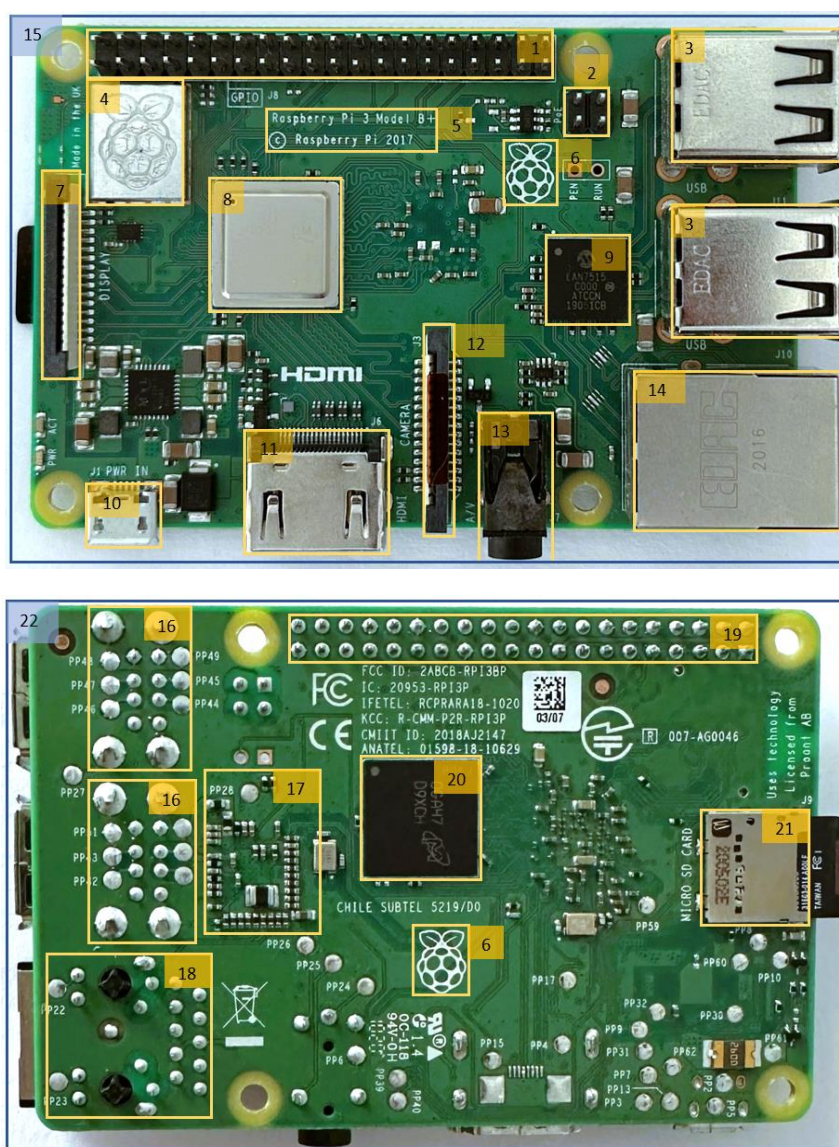
Tablica 1. Označene komponente na Arduino Uno mikroprocesorskoj pločici

Broj	Element	Broj	Element
1	Reset	10	6_pins
2	USB_Arduino	11	Arduino_front
3	USB interface	12	Arduino_UNO_model
4	Oscilator	13	Check pattern 1
5	Power	14	Check pattern 2
6	Capacitor	15	Check pattern 3
7	Arduino_CPU	16	Check pattern 4
8	ARDUINO	17	Arduino_back
9	UNO_white		

### 6.1.2. Raspberry Pi 3

Raspberry Pi 3 je mikroprocesorska pločica s Broadcom BCM2837B0 sustavom i Cortex-A53 procesorom koji radi na brzini 1.4GHz. Može se koristiti kao malo računalo za različite projekte i zadatke u komercijalnoj i industrijskoj primjeni zbog svoje fleksibilnosti korištenja različitih operacijskih sustava. Raspberry Pi ima 5GHz bežični LAN, Bluetooth, Gigabit Ethernet preko USB 2.0 i mogućnost PoE povezivanja[19].

Na prednjoj strani su označene komponente koje su standardnog oblika za sve mikroprocesorske pločice pa će se za inspekciju koristiti samo pinovi (*40\_pins*, *PoE*), dok će se na stražnjoj strani ispitivati uzorci lemljenih spojeva različitih veličina (*Check\_pattern\_1-Check\_pattern\_4*). Optičkim prepoznavanjem teksta će se kontrolirati ispisani naziv modela (*RaspberryPi\_model*) na prednjoj strani.



Slika 15. Raspberry Pi 3

Označena prednja i stražnja strana mikroprocesorske pločice te njegove komponente pobrojane su u tablici 2. Imena dodijeljena pojedinoj komponenti su preuzeta iz dostupnih korisničkih uputa za mikroprocesorsku pločicu[19].

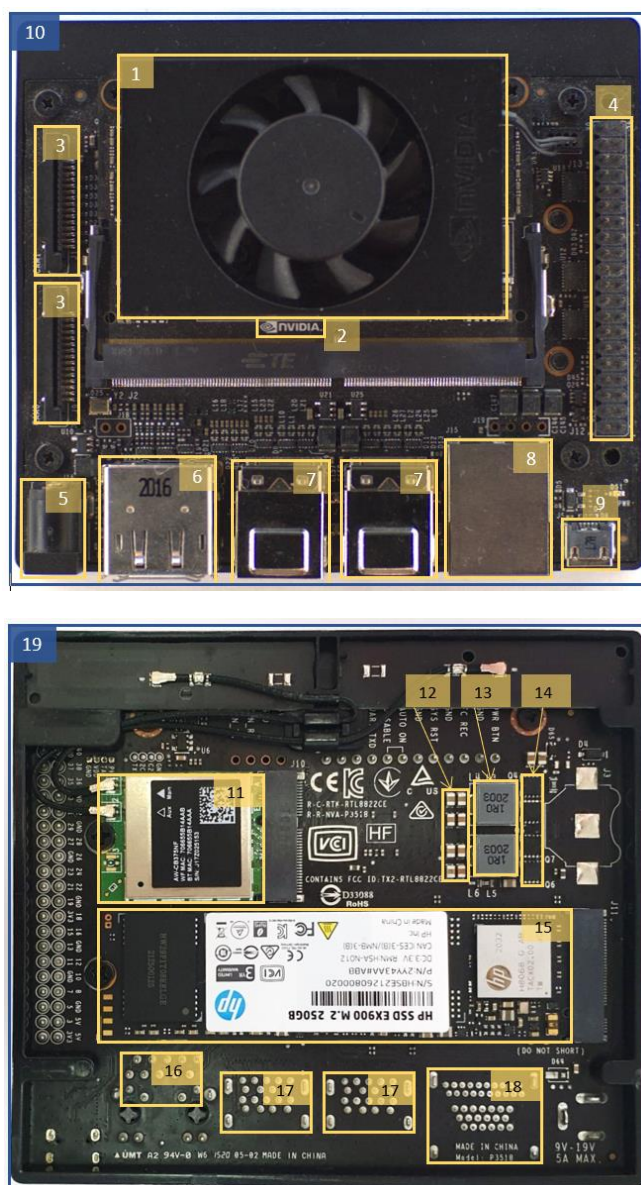
**Tablica 2. Označene komponente na Raspberry Pi 3 mikroprocesorskoj pločici**

Broj	Element	Broj	Element
1	40 pins	12	Connector_camera
2	PoE	13	A/V
3	USB	14	Ethernet
4	WIFI/Bluetooth_module	15	RaspberryPi_front
5	RaspberryPi_model	16	Check_pattern_1
6	berry	17	Check_pattern_2
7	Connector_display	18	Check_pattern_3
8	RaspberryPi_CPU	19	Check_pattern_4
9	USB interface	20	RAM
10	Connector_power	21	Micro_SD_card_port
11	Connector_HDMI	22	RaspberryPi_back

### 6.1.3. Jetson NX Xavier Development Kit

Jetson NX Xavier Development Kit je SBC kit specijaliziran za aplikacije autonomnih sustava, robotike i umjetne inteligencije, posebice vizualnu odometriju, fuziju senzorskih signala, lokalizaciju i mapiranje, detekciju prepreka i planiranje kretanja robota [20]. NX Xavier koristi osmerojezgri NVIDIA Camel CPU i Volta Tensor Core GPU. Mikroprocesor ima micro-USB, USB 3.1, Ethernet, HDMI i DC barrel jack port te MIPI CSI konektore za kamere.

Na prednjoj strani su označene komponente koje su standardnog oblika za sve mikroprocesorske pločice pa će se za inspekciju koristiti samo pinovi (*40\_pins*), dok će se na stražnjoj strani ispitivati uzorci lemljenih spojeva različitih veličina (*Check\_pattern\_1-Check\_pattern\_4* i *Resistors\_check\_pattern*). Optičkim prepoznavanjem teksta će se kontrolirati ispisani naziv proizvođača (*NVIDIA.*) na prednjoj strani.



Slika 16. Jetson NX Xavier

Označena prednja i stražnja strana mikroprocesorske pločice te njegove komponente pobrojane su u tablici 3.

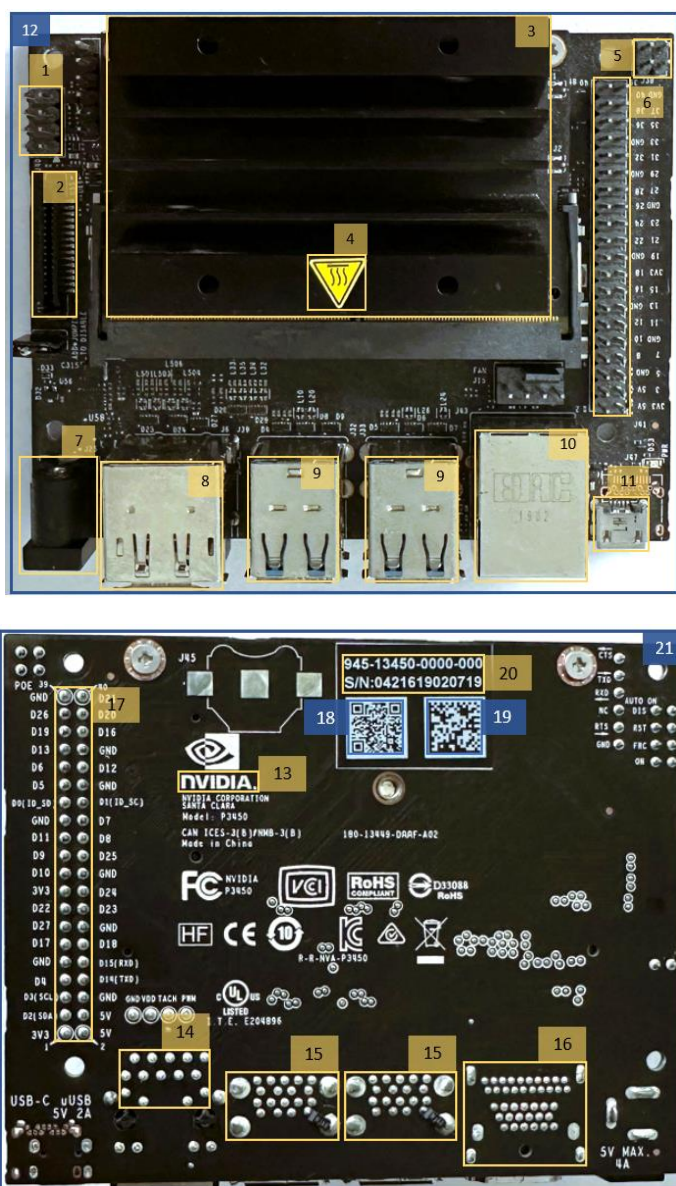
Tablica 3. Označene komponente na Jetson NX Xavier mikroprocesorskoj pločici

Broj	Element	Broj	Element
1	Ventilator	11	Wi-Fi/Bluetooth module
2	NVIDIA.	12	Resistors_check_pattern
3	Camera_connector	13	L3-l6
4	40_pins	14	Q4-Q7
5	Power	15	SSD
6	Connector_HDMI	16	Check_pattern_1
7	USB	17	Check_pattern_2
8	Ethernet	18	Check_pattern_3
9	Connector_power	19	Jetson_NX_back
10	Jetson_NX_front		

### 6.1.4. Jetson Nano Development Kit

Jetson Nano Development Kit je SBC kojeg pokreće Quad-core ARM Cortex-A57 MPCore procesor i NVIDIA Maxwell GPU. Koristi se za zadatke umjetne inteligencije poput klasifikacije slika, detekcije objekata, segmentaciju i procesiranje govora [21]. Operativni sustav koji koristi je Linux.

Na prednjoj strani su označene komponente koje su standardnog oblika za sve mikroprocesorske pločice pa će se za inspekciju koristiti samo pinovi (*40\_pins*), dok će se na stražnjoj strani ispitivati uzorci lemljenih spojeva različitih veličina (*Check\_pattern\_1-Check\_pattern\_4*). Optičkim prepoznavanjem teksta će se kontrolirati ispisani naziv proizvođača (*NVIDIA.*) te naziv modela (*Jetson\_Nano\_model*) na stražnjoj strani.



Slika 17. Jetson Nano

Označena prednja i stražnja strana mikroprocesorske pločice te njegove komponente pobrojane su u tablici 4.

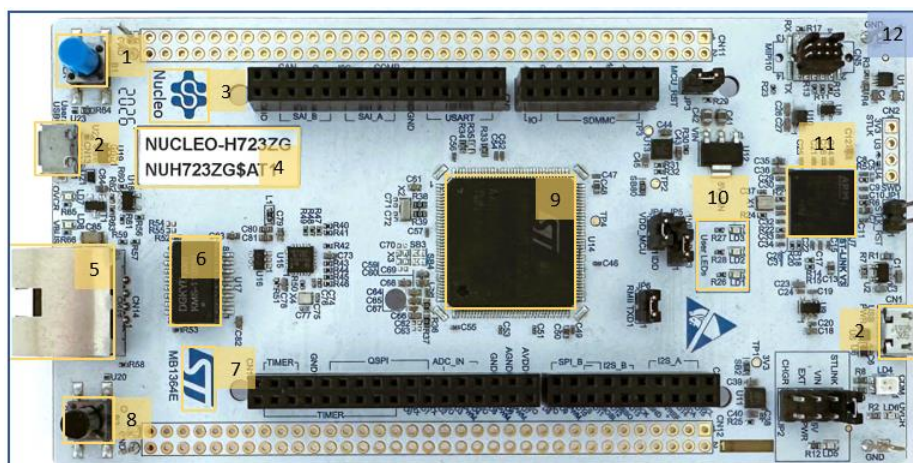
**Tablica 4. Označene komponente na Jetson Nano mikroprocesorskoj pločici**

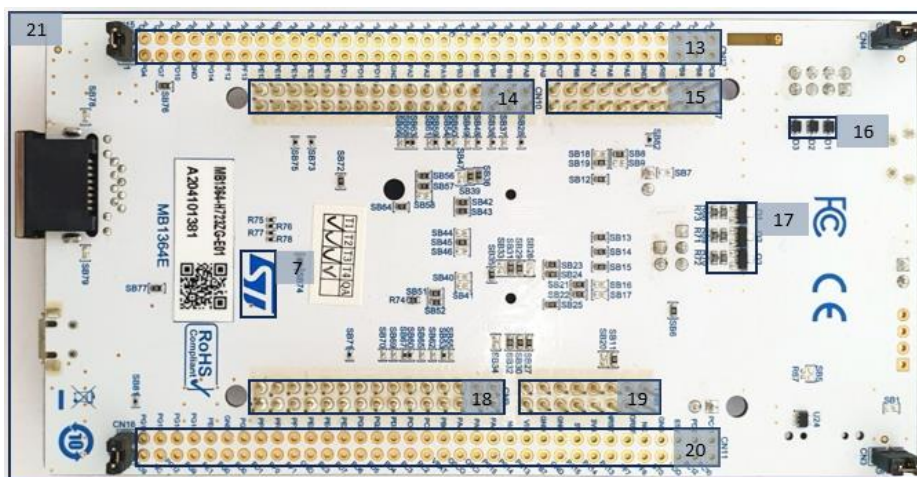
Broj	Element	Broj	Element
1	J40	12	Jetson_Nano_front
2	Connector_camera	13	NVIDIA.
3	Cooling_system	14	Check_pattern_1
4	CAUTION	15	Check_pattern_2
5	J38	16	Check_pattern_3
6	40 pins	17	Check_pattern_4
7	Power	18	QR_1
8	HDMI port	19	QR_2
9	USB port	20	Jetson_Nano_model
10	Ethernet port	21	Jetson_Nano_back
11	Connector_power		

### 6.1.5. STM32

STM32 je mikrokontroler razvijen od strane STMicroelectronics, kojeg pokreće 32-bitni ARM Cortex-M7 CPU procesor. Zbog svoje kompaktnosti i snažnih performansa, jedan je od najznačajnijih mikrokontrolera za embedded sustave i *Internet of Things* (IoT) aplikacije. Široko je prihvaćen u industriji i inženjerskim projektima [22].

U ovom slučaju se ispitivanje provodi samo na pinovima sa stražnje strane (*CN7- CN12*) dok se optičko prepoznavanje teksta provodi nad nazivom modela i serijskog broja na prednjoj strani (*STM32\_model\_name*). Optičko prepoznavanje teksta na natpisima proizvođača (*Nucleo, ST*) se ne provodi zbog drugačije orijentacije teksta.





Slika 18. STM32

Označena prednja i stražnja strana mikroprocesorske pločice te njegove komponente pobrojane su u tablici 5.

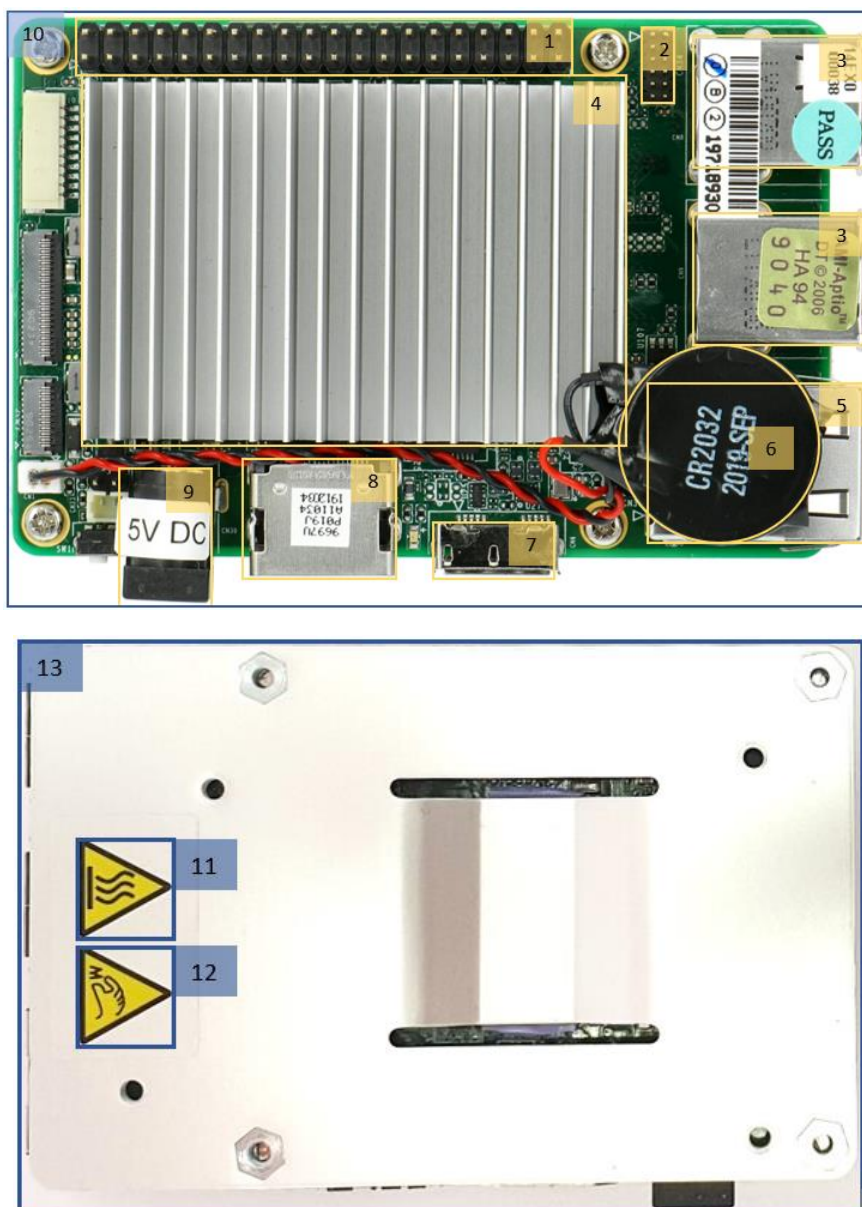
Tablica 5. Označene komponente na STM32 mikroprocesorskoj pločici

Broj	Element	Broj	Element
1	USER button	12	STM32_front
2	Connector_power	13	CN12
3	Nucleo	14	CN10
4	STM32 model	15	CN7
5	Ethernet	16	D1-D3
6	Transformer	17	Q1-Q3
7	ST	18	CN9
8	RESET button	19	CN8
9	CPU	20	CN11
10	User LEDs	21	STM32_back
11	ST V3		

### 6.1.6. Up Board

Up board koristi Intelov 64-bitni Atom x5-z8350 procesor, opremljen s Intel Gen 8 HD 400 grafičkom karticom te 4GB RAM memorije. Izlazno/ulazne komponente uključuju 5 USB 2.0 ili 3.0 konektora, Ethernet, HDMI te MIPI kamera portove[23].

Na prednjoj strani su označene komponente koje su standardnog oblika za sve mikroprocesorske pločice pa će se za inspekciju koristiti samo pinovi (*40\_pins*). Većina stražnje strane je prekrivena kućištem koje je čvrsto vezano za mikroprocesorsku pločicu što otežava dodatno rastavljanje za potrebe detaljne analize. Također prednja strana je prekrivena rebrastim sustavom za odvođenje topline koji je čvrsto vezan za CPU zbog čega nije rastavljan za ovu analizu.



Slika 19. Up board

Označena prednja i stražnja strana mikroprocesorske pločice te njegove komponente pobrojane su u tablici 6.

Tablica 6. Označene komponente na Up board mikroprocesorskoj pločici

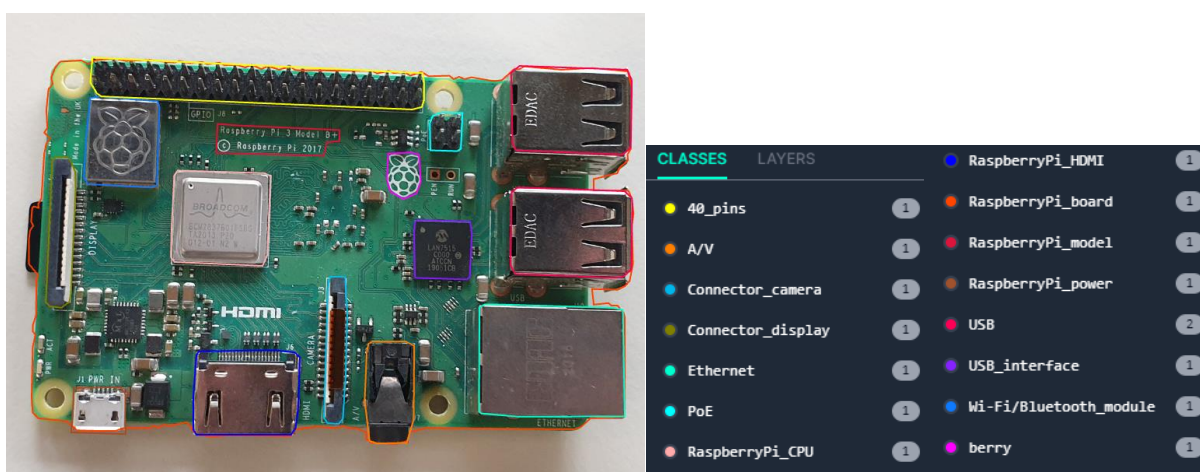
Broj	Element	Broj	Element
1	40_pins	8	HDMI port
2	CN34	9	Power
3	USB	10	Up_board_front
4	Cooling system	11	CAUTION_1
5	Ethernet (covered)	12	CAUTION_2
6	Battery	13	Up_board_back
7	Connector_power		



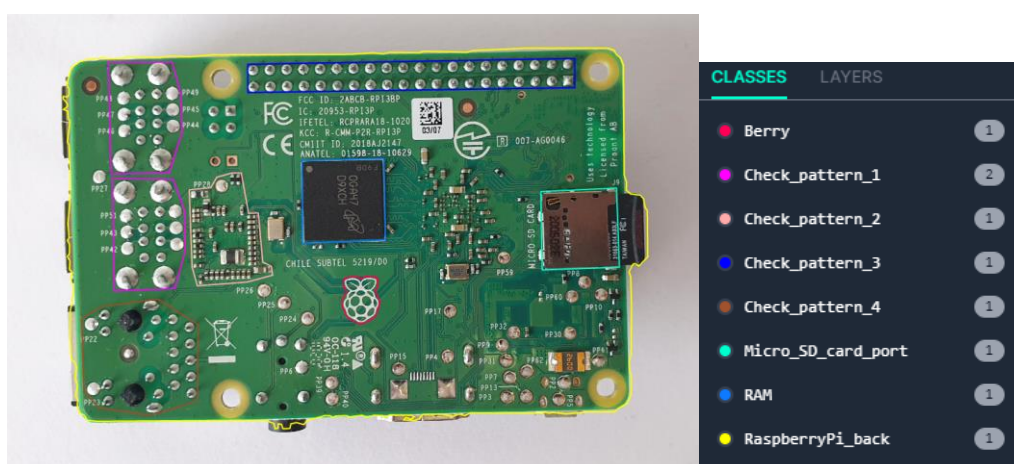
## 6.2. Označavanje slika korištenjem Roboflow-a

Proces treniranja YOLOv8 modela konvolucijske neuronske mreže počinje stvaranjem seta trening, validacijskih i testnih podataka, odnosno označavanjem prikupljenih slika. Izrada seta podataka uključuje snimanje slika i pre-procesiranje kako bi se uzele u obzir najkvalitetnije slike. Cilj je snimiti slike u uvjetima različitog osvjetljenja te iz različitih kutova i udaljenosti snimanja kako bi se osigurala raznolikost podataka. Na slici 21. je prikazano nekoliko primjera slika koje se nalaze u setu za označavanje.

Na svim slikama je potrebno označiti pločice i druge relevantne komponente te im pridijeliti odgovarajuće klase. Na slici 20. prikazan je primjer označavanja elemenata na mikroprocesorskoj pločici Raspberry Pi 3.



a) Raspberry Pi 3, prednja strana s ispisanim klasama



b) Raspberry Pi 3, stražnja strana s ispisanim klasama

### Slika 20. Označena Raspberry Pi 3 pločica

Za označavanje su odabrana I/O sučelja, na primjer *General Purpose Input/Outputs* (GPIO), priključci poput Ethernet, USB, HDMI, te elementi poput Wi-Fi, Bluetooth i USB

sučelja. Nadalje, označene su skupine, odnosno uzorci lemljenih spojeva koji povezuju pojedine elemente s pločicom, na primjer, USB priključak koji je označen na stražnjoj strani oznakom *Check\_pattern\_1* na slici 20. Označeni lemljeni spojevi i GPIO priključci će kasnije biti dodatno ispitani algoritmom adaptivnog praga kako bi se provjerilo ima li svaki element odgovarajući broj spojeva ili pinova te jesu li oni odgovarajuće veličine i kvalitete. Također se označavaju ispisani nazivi modela na pločicama za optičko prepoznavanje teksta i provjere ispravnosti rada YOLOv8 modela. U procesu segmentacije, ovisno o kvalitetama slika, kontrastima boja i uvjetima osvjetljenja, primjenjivale su se metode ručnog segmentiranja postavljanjem poligona i primjena *Segment Anything Model* (SAM) modela, treniranog na 11 milijuna slika i s 1.1 milijardom segmentacijskih zadataka [24]. Ulazni set podataka se sastoji od 1200 slika, pri čemu svaki od 6 mikroprocesora ima 200 slika, od toga 100 za prednju i 100 za stražnju stranu. Označeni set podataka ima 82 klase, 12,883 oznake, a prosječna rezolucija slika je 2560\*1440 piksela. Slika 21 prikazuje po tri označene slike za svaku stranu mikroprocesorskih pločica pri različitim orijentacijama i uvjetima osvjetljenja.

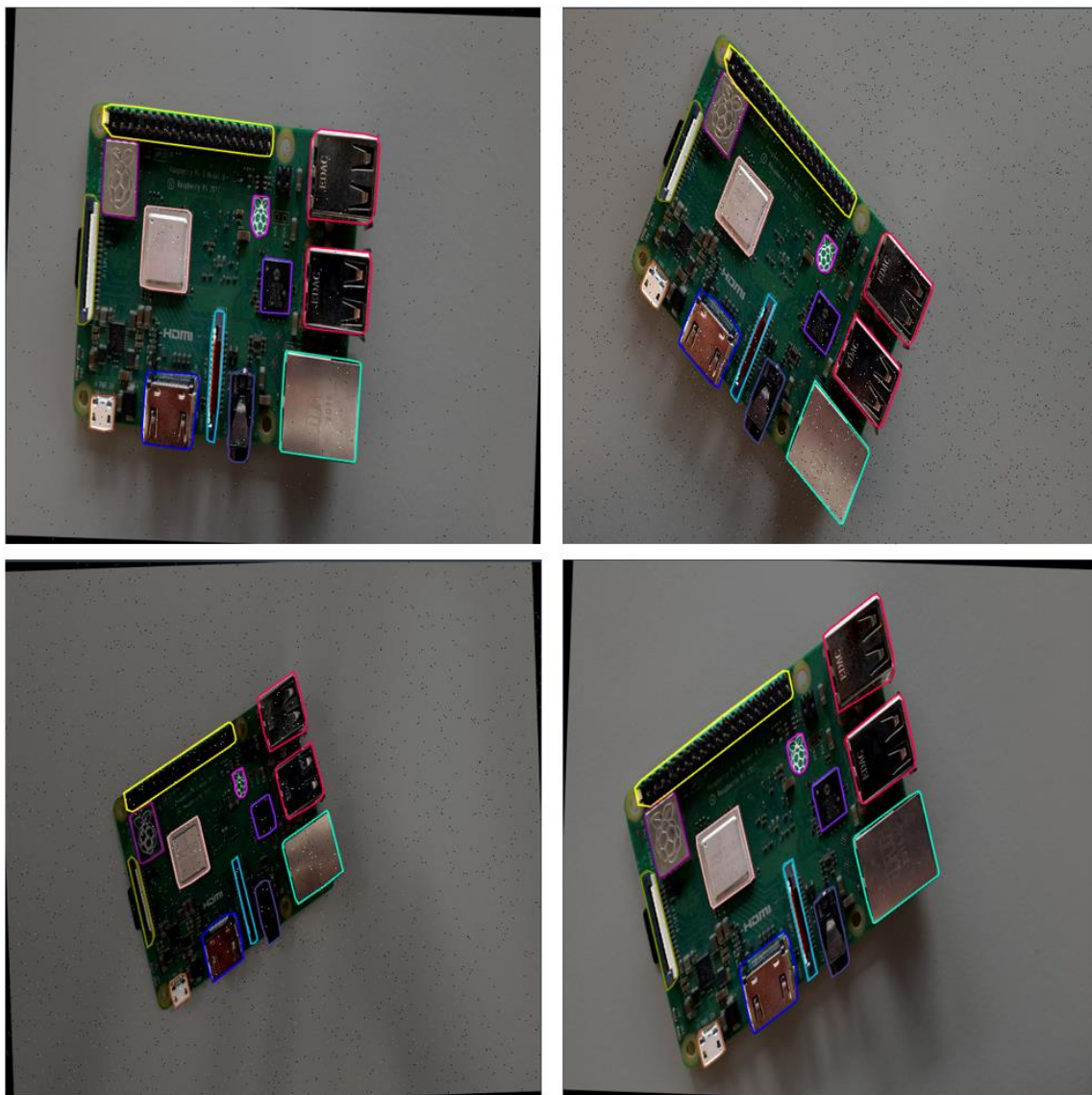


Slika 21. Primjeri označavanja komponenti na setu podataka

Prilikom generiranja seta podataka iz označenog seta slika, primjenjuje se procesiranje koje se sastoji od Auto-Orient metode, koja standardizira rotacije elemenata i raspored piksela. Zatim se primjenjuje automatska prilagodba kontrasta kako bi model bolje detektirao rubove označenih objekata. Konačno, smanjuje se rezolucija slika na 640\*640, što je optimalna veličina slika za trening YOLOv8 modela.

U nastavku se provodi augmentacija seta podataka koja uključuje rotaciju slika za  $\pm 8^\circ$  i promjene nijanse (engl. Hue) za  $\pm 6^\circ$ , zasićenja (engl. Saturation) za  $\pm 8\%$ , osvjetljenja (engl. Brightness) za raspon od -10% do 0% te ekspozicije (engl. Exposure) za  $\pm 7\%$ . Navedene vrijednosti su određene eksperimentalno. Augmentacija je tehnika koja se primjenjuje na setu trening podataka konvolucijskih neuronskih mreža kako bi se umjetno povećao set podataka. Glavni razlog za provođenje augmentacije je poticanje raznolikosti i reprezentativnosti seta za stvarne uvjete i robusnosti treniranog modela. Na taj način set podataka postaje raznovrstan, a trenirani model ima bolju generalizaciju. Osim toga, augmentacija pridonosi regularizaciji, sprječavajući pretreniranje modela. Parametri augmentacije se određuju slučajnim odabirom unutar postavljenih granica za svaku sliku u setu podataka. Na sliku se također dodaje *Salt and pepper* šum u obliku nasumično postavljenih crnih i bijelih piksela. Šum se dodaje na 0.5% piksela slike, a služi smanjenju mogućnosti pretreniranja. Konačno, na slike se primjenjuje Gaussov filtar s veličinom radijusa od 0.75 piksela kako bi se slika blago zamutila i trenirani model imao bolje generalizacijske performanse. Augmentacija se primjenjuje samo na trening i validacijskim setovima, a ne na testnom setu, osiguravajući objektivnu evaluaciju performansi modela nad novim i nepoznatim podacima. Ovaj pristup pomaže održati realističnost i robusnost modela što je bitno za kvalitetnu detekciju i klasifikaciju u realnim uvjetima eksploatacije modela. Slučajnim odabirom vrijednosti parametara moguće je multiplicirati broj slika, pa da se nakon augmentacije set sastoji od 2895 trening slika, 135 validacijskih i 100 test slika.

Slika 22 prikazuje četiri slike iz seta podataka prednje strane Raspberry Pi 3. Na slikama je moguće primijetiti nasumičnu promjenu osvjetljenja i boja, orijentacije slika te dodani šum.



Slika 22. Raspberry Pi 3 set podataka nakon augmentacije

## 7. Treniranje YOLOv8 modela

Treniranje YOLOv8 modela predstavlja iterativni proces učenja detekcije specifičnih uzoraka na slikama. Model tijekom treniranja mijenja svoje unutarnje parametre ovisno o setu trening podataka kako bi mogao precizno detektirati i lokalizirati objekte označene na slikama. Promjenom parametara tijekom treniranja smanjuje se razlika između detekcija modela i stvarnih, označenih vrijednosti. Ta razlika se može opisati funkcijama gubitaka.

Proces se sastoji od inicijalizacije pre-treniranog YOLOv8 modela sa zadanim težinama i BIAS vrijednostima, koje će se adaptirati tijekom treninga. Nakon inicijalizacije kreće iterativni postupak koji se sastoji od nekoliko ključnih koraka. Prvo se provodi unaprijedni prolaz (engl. forward pass) podataka kroz model, gdje ulazni podatci prolaze kroz konvolucijske slojeve i druge mehanizme unutar modela, te se računa izlazna vrijednost, odnosno detekcija. Ova vrijednost se zatim uspoređuje s očekivanom vrijednošću koja je dio trening seta podataka, kako bi se izračunala greška koja odražava razliku stvarne i predviđene detekcije. Zatim slijedi prolaz unatrag (engl. backpropagation) kroz strukturu modela pri čemu se računa greška detekcije u svakom sloju korištenjem funkcija gubitaka. Izračun gradijenta funkcije gubitaka određuje trend rasta ili pada greške tijekom iteracija. U procesu spuštanja gradijenta, adaptiraju se parametri modela u svakome sloju u smjeru negativnog gradijenta svakog od parametara, odnosno, u smjeru manje greške.

Treniranje se provodi pomoću Google Colaboratory alata u Python programskom jeziku. Google Colaboratory omogućuje pristup Googleovom GPU hardveru koji osigurava performanse potrebne za brzo treniranje modela. Odabrani alat za treniranje je PyTorch, koji se koristi za prototipiranje u istraživačkim radovima zbog jednostavnosti prilikom treniranja u odnosu na Keras, Tensorflow i ostale alate.

Za rješavanje postavljenog problema odabran je YOLOv8 medium model koji se sastoji od 295 slojeva, 25,902,640 parametara za učenje i 25,902,624 pripadajućih gradijenta. U nultom koraku trening procesa parametri za učenje se inicijaliziraju pre-treniranim vrijednostima modela treniranog na COCO trening setu podataka. Trening proces je podijeljen u epohe, odnosno segmente treninga u kojima kroz strukturu modela prođe cijeli set podataka. Svaka epoha se sastoji od više skupina podataka (engl. batch) u koje je podijeljen set podataka, a skupina podataka je veličine višekratnika broja 8. Nakon prolaska skupine podataka kroz model, vrši se prolaz unatrag te se adaptiraju parametri modela. Epoha završava nakon što kroz strukturu modela prođu sve skupine podataka.

Predviđeno trajanje treninga je 100 epoha, pri čemu se definira parametar strpljenja od 4 epohe. Strpljenje od 4 epohe osigurava prekidanje procesa treniranja ako nema poboljšanja u funkciji gubitaka i performansama detekcije u prethodne 4 epohe. Ova metoda se koristi kako bi se izbjeglo pretreniranje modela. Odabrana veličina skupine podataka za treniranje YOLOv8 modela iznosi 16. Brzina učenja je 0.001, a momentum pri učenju je 0.937. Kao optimizator u procesu adaptacije parametara koristi se AdamW, stohastički optimizator koji koristi gradijente prvog reda, odnosno samo prvu derivaciju funkcija gubitaka [25]. AdamW u iteracijama modificira inicijalno postavljenu brzinu učenja kako bi olakšao pronalazak globalnih minimuma funkcija gubitaka i popravio točnost detekcije. Odabrani parametri, poznati kao hiperparametri, dodatno su prikazani u tablici 7.

**Tablica 7. Trening hiperparametri**

YOLOv8m hiperparametri	
Broj epoha	100
Strpljenje	4 epohe
Batch	16
Optimizator	AdamW
Brzina učenja	0.001
Momentum prilikom učenja	0.937

Trening proces uspješno je završen nakon 51. epohe, a treniranje je prekinuto zbog postavljenog strpljenja od 4 epohe. Ukupno trajanje treninga je 3.3 sata korištenjem NVIDIA Tesla T4 grafičkog procesora s 15GB RAM memorije. Rezultati treniranja detaljno su prikazani u prilogu 1., pružajući analizu učinkovitosti modela. Tablica 8 daje primjer validacijskih rezultata za klasu *40\_pins* te detaljan uvid u sposobnosti detekcije i klasifikacije treniranog modela. Radi se o validacijskim rezultatima nakon 51. epohe.

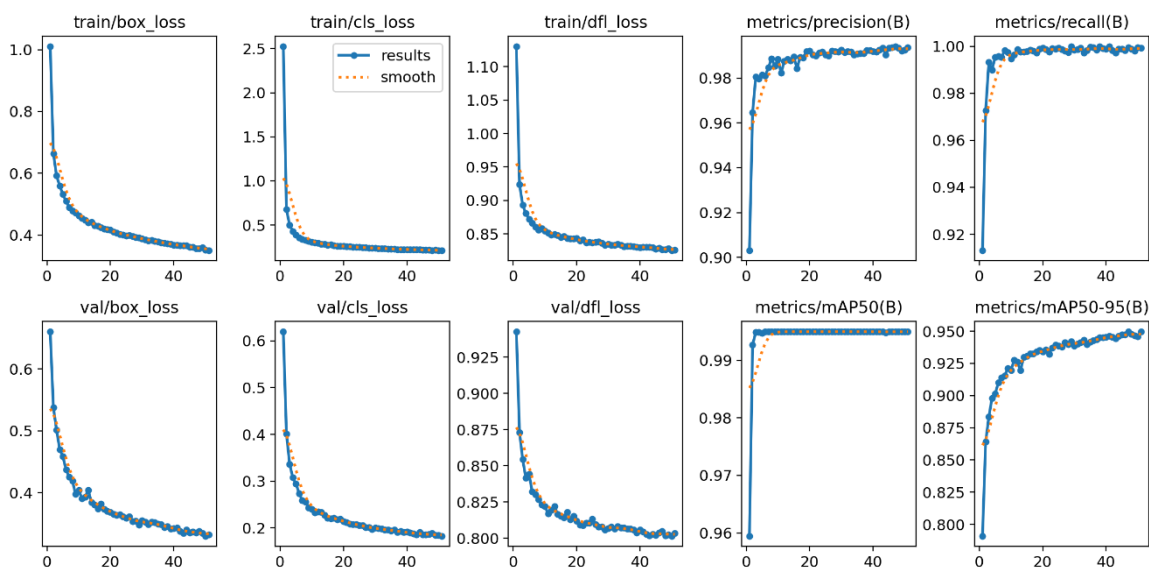
**Tablica 8. Trening rezultati za klasu *40\_pins***

Klasa	Broj slika	Broj instanci klase na slikama	Ograničavajuće kućice			
			Precision (P)	Recall (R)	mAP(50% IoU)	mAP(95% IoU)
40_pins	135	48	0.999	1	0.995	0.971

Kvaliteta detekcije se definira standardiziranim mjerama *Precision* P i *Recall* R. *Precision* modela za sve klase je iznad 96%, što znači da je većina detektiranih elemenata pravilno klasificirana. *Recall* svih modela je iznad 99%, što znači da je model sposoban prepoznati gotovo sve tražene elemente. Većina klasa ima *Recall* od 100% što znači da su detektirani svi

traženi elementi. Ovi rezultati potvrđuju sposobnost modela za detektiranje različitih objekata s visokom točnošću. *Mean Average Precision* (mAP) dodatno potvrđuje rezultate. Uzme li se IoU granica od 50% prilikom detekcije, mAP iznosi 99.5% što označava veliku točnost i sposobnost detekcije za manje precizne ograničavajuće kućice. Za vrlo precizne ograničavajuće kućice s IoU granicom od 95%, mAP iznosi 95%.

Rezultati trening procesa dodatno opisuju grafovi na slici 23. prema kojima se mogu pratiti dodatni parametri točnosti i preciznosti detekcije treniranog modela te se može pratiti trend rasta navedenih vrijednosti te trend pada gubitaka tijekom učenja. Kontrolom trendova pada funkcija gubitaka i rastom *precision*, *recall* i mAP, dolazi se to zaključka da mreža nije pretrenirana.



**Slika 23. Rezultati trening procesa**

Grafovi koji prikazuju gubitke mjerene nad trening podacima i validacijskim podacima daju uvid u proces treniranja YOLOv8 modela. Vrijednost *Bounding Box Regression loss* ili *box-loss* predstavlja preciznost određivanja koordinata ograničavajućih kućica oko objekata, uključujući središte te visinu i širinu kućica. *Box-loss* se računa metodom srednje kvadratne pogreške, a njegovo smanjenje sugerira preciznije određivanje rubova detektiranog objekta. *Classification loss* ili *cls-loss* predstavlja grešku modela zbog predviđanja pogrešne klase objekta za određenu ograničavajuću kućicu. Računa se kao omjer između vjerojatnosti detekcije određene klase objekata i stvarne oznake klase korištenjem *Cross-Entropy loss* metode. Vrijednost *Detection Focal loss* ili *dfl-loss* predstavlja gubitke uzrokovane slabije reprezentiranim klasama u setu podataka. *Dfl-loss* potiče rast točnosti detekcije slabije reprezentiranih klasa dodjeljivanjem većih težina težim klasifikacijskim primjerima. Analiza

ovih gubitaka pomaže praćenju performansi modela te pomaže određivanju segmenata gdje postoji prostor za poboljšanje detekcije.

Navedene funkcije gubitaka su sastavni dio ukupne funkcije gubitaka koja se uzima u obzir u povratnoj propagaciji, odnosno učenju modela. U trening procesu *box-loss* gubitak iznosi 36.9%, *cls-loss* iznosi 22,69%, dok je *dfl-loss* 82,87%. Viši *dfl-loss* znači i manju kvalitetu lokalizacije slabije reprezentiranih objekata. Uzrok slabije reprezentiranih objekata je u slučaju da su prekriveni ili djelomično vidljivi na trening setu podataka pa zbog toga nisu označeni. Posljedično, slabije reprezentirani objekt ima manje ograničavajućih kućica nad kojima model može učiti generaliziranu reprezentaciju objekta.

**Tablica 9. Trening rezultati**

YOLOv8m	Validacijski rezultati	Testni rezultati
Broj epoha	51	-
Box_loss	0.3505	-
Cls_loss	0.2116	-
Dfl_loss	0.8262	-
Precision	0.994	0.934
Recall	0.998	0.997
mAP (50%)	0.995	0.995
mAP (95%)	0.954	0.932

Nakon treniranja, potrebno je pretvoriti trenirani PyTorch model u *Open Neural Network Exchange* (ONNX) format. ONNX je format koji omogućuje jednostavnu razmjenu modela između različitih alata za rad s neuronskim mrežama kao što su PyTorch, Tensorflow, Keras, Matlab i ONNX Runtime. Pretvaranje modela u ONNX format zbog svoje fleksibilnosti omogućuje implementaciju modela u različitim aplikacijama.



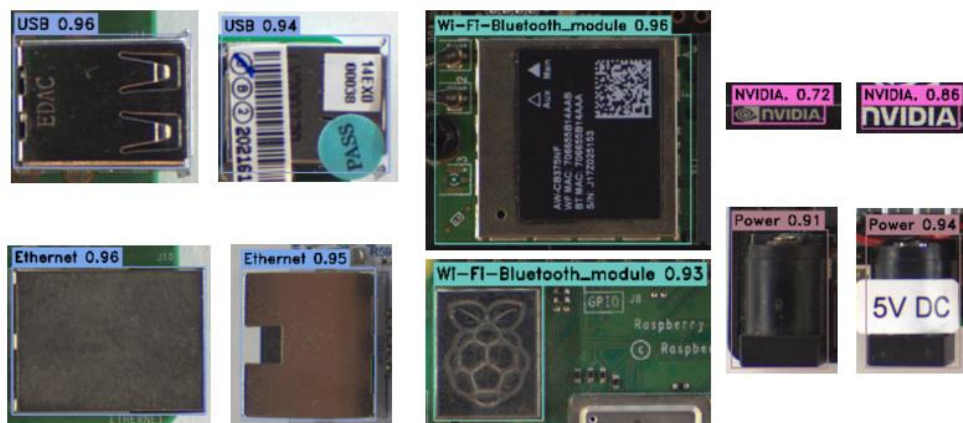
## 8. Detekcija i izdvajanje detektiranih komponenata

Prvi korak kontrole kvalitete je detekcija mikroprocesorskih pločica i označenih komponenti kako bi se kasnije mogla vršiti njihova analiza. Model je treniran za detekciju mikroprocesorskih pločica u različitim orijentacijama kao što je prikazano na slici 24, ali će se inspekcija vršiti samo u vodoravno postavljanim mikroprocesorskim pločicama za potrebe optičke kontrole teksta. Detekcije u različitim orijentacijama potvrđuju robusnost sustava.



Slika 24. Detekcije u različitim orijentacijama

Također, detekcija ima sposobnost generalizacije za elemente koji se razlikuju između procesorskih pločica, ali imaju istu ulogu pa su označeni kao isti elementi.

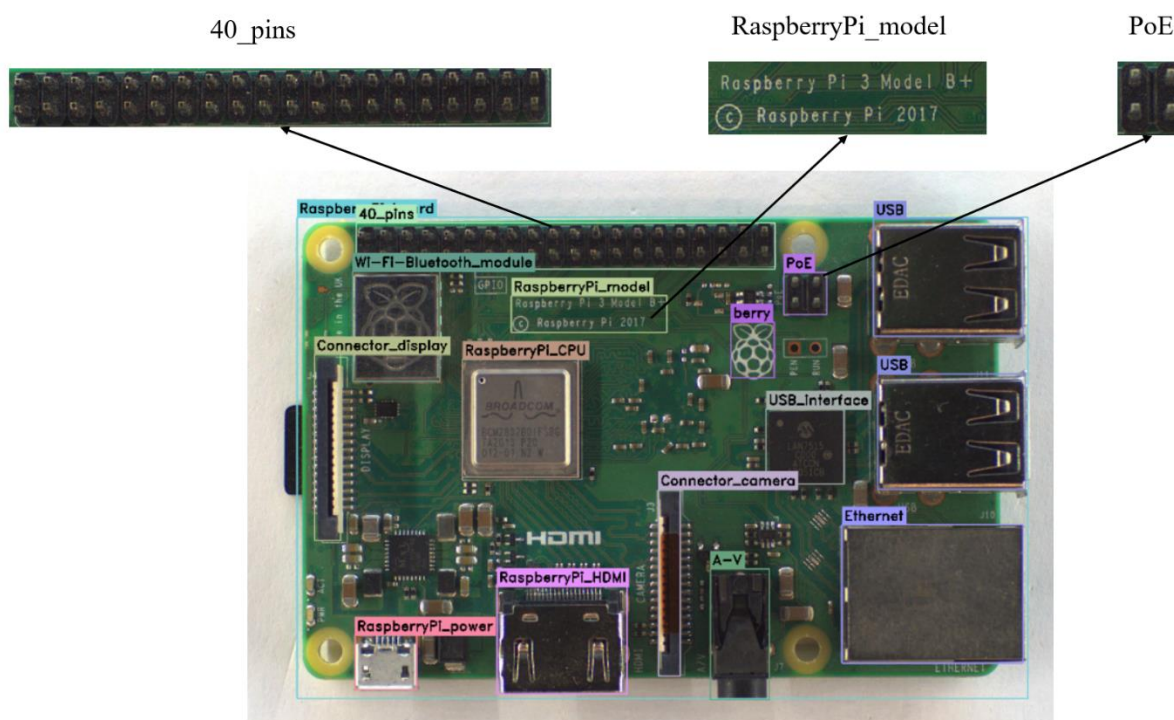


Slika 25. Robusnost YOLOv8 modela na razlike unutar klase detektiranog objekta

Vrijeme detekcije korištenjem Tesla T4 grafičkog procesora traje svega 32,2 milisekunde, odnosno 1,5 milisekundi za preprocesiranje, 19,9 za detekciju i 10,8 milisekundi za postprocesiranje i iscrtavanje ograničavajućih kućica. Dobiveni rezultati se odnose na slike trening seta podataka dimenzija 640\*640 piksela. U slučaju Phoenix PHX064S-CC industrijske

kamere koja snima rezolucijom 3072\*2048 piksela s ograničenjem brzine snimanja pri maksimalnoj rezoluciji na 17.7 FPS, te korištenjem NVIDIA GeForce GTX 1650 grafičke kartice, trajanje procesa detekcije raste na 185 milisekundi.

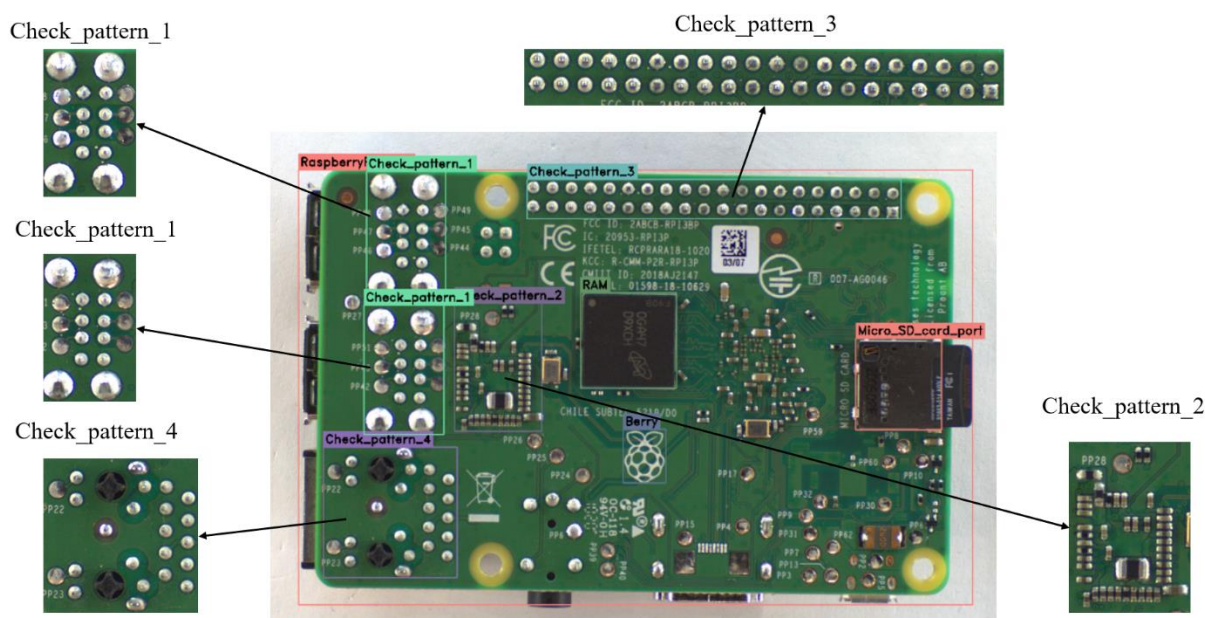
Rezultat detekcije je struktura podataka koja se sastoji od identifikacijskog broja klase `detection.class_id` i identifikacijskog broja detekcije `detection.detection_id` jer je moguće imati više detekcija iste klase na pojedinoj mikroprocesorskoj pločici. Nadalje, detekcijom se spremaju podatci o nazivu detektirane klase `detection.class_name`, vjerojatnosti točne detekcije `detection.confidence` te koordinate ograničavajuće kućice `detection.box`. Navedeni podatci se obrađuju OpenCV algoritmima kako bi se prikazali rezultati detekcije i nastavila inspekcija. Proces detekcije i izdvajanja za Raspberry Pi 3 mikroprocesorsku pločicu je prikazan na slikama 26. i 27. Prvi korak je akvizicija slike s kamere korištenjem industrijske kamere, Arena API-ja i OpenCV alata za prikaz slike. Zatim se vrši detekcija traženih komponenti YOLOv8 modelom. Slika 26. prikazuje prednju stranu s iscrtanim ograničavajućim kućicama i imenima pojedine klase iznad kućice te izdvojenim elementima za kasniju inspekciju i optičko prepoznavanje teksta.



**Slika 26. Detekcija i izdvajanje komponenti na prednjoj strani Raspberry Pi 3 mikroprocesorske pločice**

Na mikroprocesorskoj pločici su detektirani standardni elementi, poput USB i Ethernet portova, koji su zajednički za velik broj proizvođača te se zbog toga ne provodi inspekcija tih

elemenata. Umjesto toga, provest će se inspekcija njihovih spojeva sa samom pločicom. Također će se analizirati pinovi kako bi se provjerio njihov broj i ispravnost. Postupak izdvajanja koristi OpenCV funkciju `capturedImage(detection.box)` uz korištenje `clone()` funkcije za stvaranje kopije segmenta slike unutar specificiranih granica detektiranih ograničavajućih kućica. Na prednjoj strani se vrši izdvajanje elemenata poput `40_pins` i `PoE`, kao što je prikazano na detalju slike 26. Sa stražnje strane se vrši izdvajanje `Check_pattern_1` do `Check_pattern_4` elementa. `Check_pattern_1` predstavlja spojeve USB priključaka, `Check_pattern_2` predstavlja spoj USB sučelja, `Check_pattern_3` predstavlja spoj Ethernet priključka i `Check_pattern_4` predstavlja spoj GPIO/pinova. Ovi uzorci su odabrani za inspekciju zbog različitih veličina pinova i lemljenih spojeva, naglašavajući time robusnost inspekcije.



**Slika 27. Detekcija i ekstrakcija komponenti na stražnjoj strani Raspberry Pi 3 mikroprocesorske pločice**

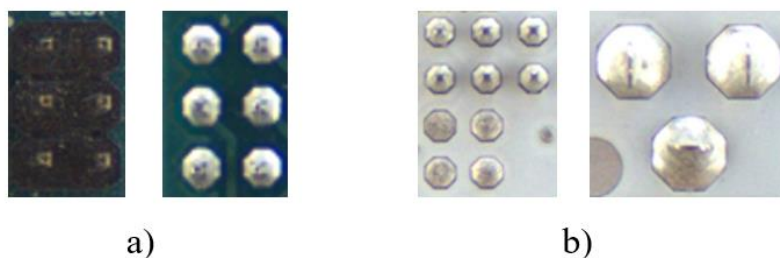
Konačno, kako bi se potvrdila točnost detekcije, koristi se optičko prepoznavanje teksta na ispisanim imenima i modelima mikroprocesorskih pločica. U slučaju Raspberry Pi 3, vrši se izdvajanje `RaspberryPi_model` komponente za daljnju analizu.

## 9. Inspekcija

Inspekcija se provodi na slikama pinova i lemljenih spojeva nakon detekcije i izdvajanja segmenata slike. Inspekcijski algoritam, razvijen korištenjem OpenCV alata, se sastoji od nekoliko koraka: transformacija trokanalne slike u jednokanalnu, primjena metode adaptivnog praga, pronalaženje kontura, filtriranje kontura i ispis ograničavajućih kućica za inspekciju.

Za provođenje inspekcije, nakon izuzimanja segmenata slike, potrebno je pojednostaviti prikaz transformacijom iz trokanalne slike u jednokanalnu, odnosno pretvorba slike u boji u crno-bijelu. Na taj način se slika zapisana u tri matrice, od koje svaka određuje intenzitet crvene, plave i zelene boje za pojedini piksel, transformira u jednu matricu u kojoj vrijednosti predstavljaju intenzitet bijele boje za svaki piksel. Na jednokanalnoj slici se može provesti segmentacija metodom adaptivnog praga.

Segmentacija korištenjem praga predstavlja segmentacijsku metodu određivanja određenog praga vrijednosti unutar raspona intenziteta piksela na slici. Određeni prag predstavlja granicu koja se koristi za usporedbu intenziteta pojedinih piksela slike, te se na temelju usporedbe provodi segmentacija slike. Raspon intenziteta piksela za 8-bitne slike je od 0 do 255. Uspoređivanjem praga vrijednosti s vrijednostima intenziteta piksela, svaki piksel mijenja svoj intenzitet ovisno o vrsti primijenjene metode praga. Metode praga koje se mogu koristiti su `THRESH_BINARY`, koji svim pikselima manje vrijednosti od praga vrijednosti mijenja intenzitet u nulu a svima iznad u maksimum. Ova metoda rezultira binarnom slikom, gdje pikseli mogu imati samo dva moguća stanja, 0 i 255. `THRESH_BINARY_INV` metoda funkcionira na isti način kao i `THRESH_BINARY` ali ona piksel manjeg intenziteta postavlja u maksimum, a onaj višeg postavlja u nulu. Nadalje, `THRESH_TOZERO` i `THRESH_TOZERO_INV` rade na isti način osim što se pikselu višeg intenziteta za standardni i nižeg intenziteta za `INV`, od vrijednosti praga ne mijenja intenzitet dok preostali pikseli poprimaju vrijednost nula. Također se koristi i `THRESH_TRUNC` koji svim pikselima iznad praga vrijednosti postavlja intenzitet jednak pragu [26]. U ovom slučaju će se koristiti `THRESH_BINARY` i `THRESH_BINARY_INV` metode. `THRESH_BINARY` metoda se koristi u slučaju boje mikroprocesorske pločice manjeg intenziteta u odnosu na pinove i lemljene spojeve, a `THRESH_BINARY_INV` se koristi ako je boja većeg intenziteta, na primjer bijela boja kod stražnje strane Arduino Uno ili STM32. Primjeri korištenja pojedine metode su prikazani na slici 28.



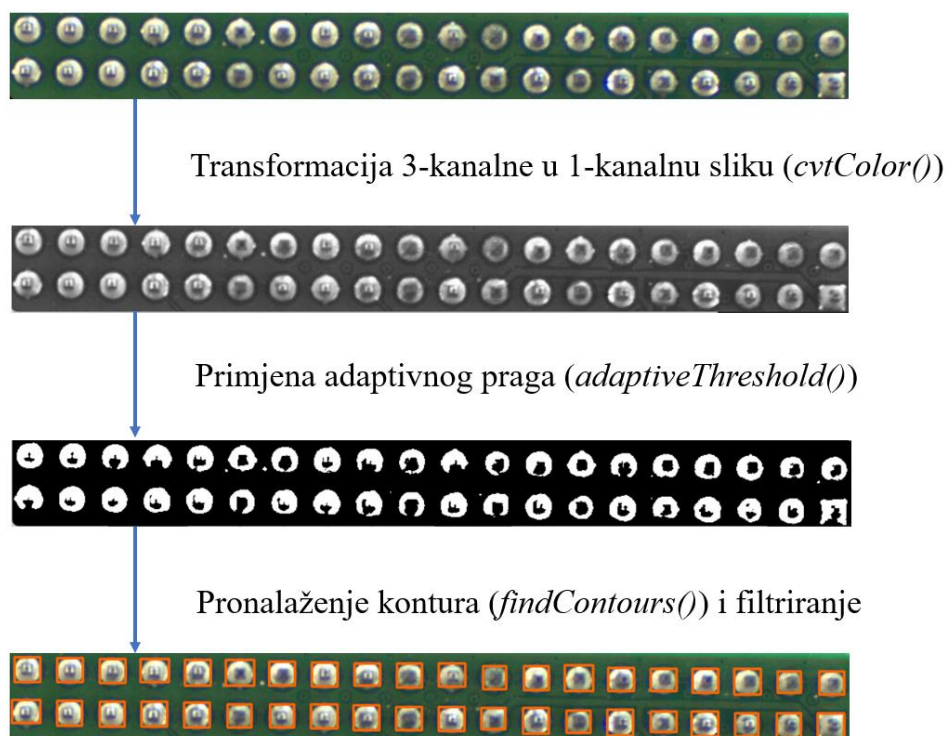
**Slika 28. Arduino Uno: a) Primjeri korištenja THRESHOLD\_BINARY metode, b) primjeri korištenja THRESHOLD\_BINARY\_INV metode**

Umjesto standardnog praga, koristi se adaptivni prag koji se računa zasebno za svaki piksel na temelju intenziteta susjednih piksela. U ovom radu se koristi algoritam sa šest različitih veličina blokova susjednih piksela, pri čemu veličina bloka određuje broj susjednih piksela koji se uzimaju u obzir prilikom računanja adaptivnog praga. Adaptivna metoda koja se koristi za računanje praga je *Adaptive threshold Gaussian C*, koja primjenjuje Gaussov princip za izračun srednje vrijednosti, dajući veću važnost pikselima bliže pikselu za koji se prag računa u odnosu one udaljenije. Različite veličine blokova se koriste kako bi se kasnijim filtriranjem pronašla najbolja veličina bloka.

OpenCV funkcija adaptivnog praga je opisana izrazom u kojem je potrebno specificirati ulaznu sliku, odnosno ulazno polje (*inputImg*), izlazno polje (*outputImg*), maksimalni intenzitet piksela (*maxIntensity*), specificiranu adaptivnu metodu za računanje praga (*adaptiveMethod*), vrstu praga koja se koristi (*thresholdType*), veličinu bloka susjednih piksela za računanje adaptivnog praga (*blockSize*) te konstantu *C*:

$$\text{adaptiveThreshold}(\text{inputImg}, \text{outputImg}, \text{maxIntensity}, \text{adaptiveMethod}, \quad (4) \\ \text{thresholdType}, \text{blockSize}, C)$$

Dodatno, eksperimentalno je dobivena konstanta *C* koja se oduzima od izračunatog praga kako bi se dodatno optimirala segmentacija.

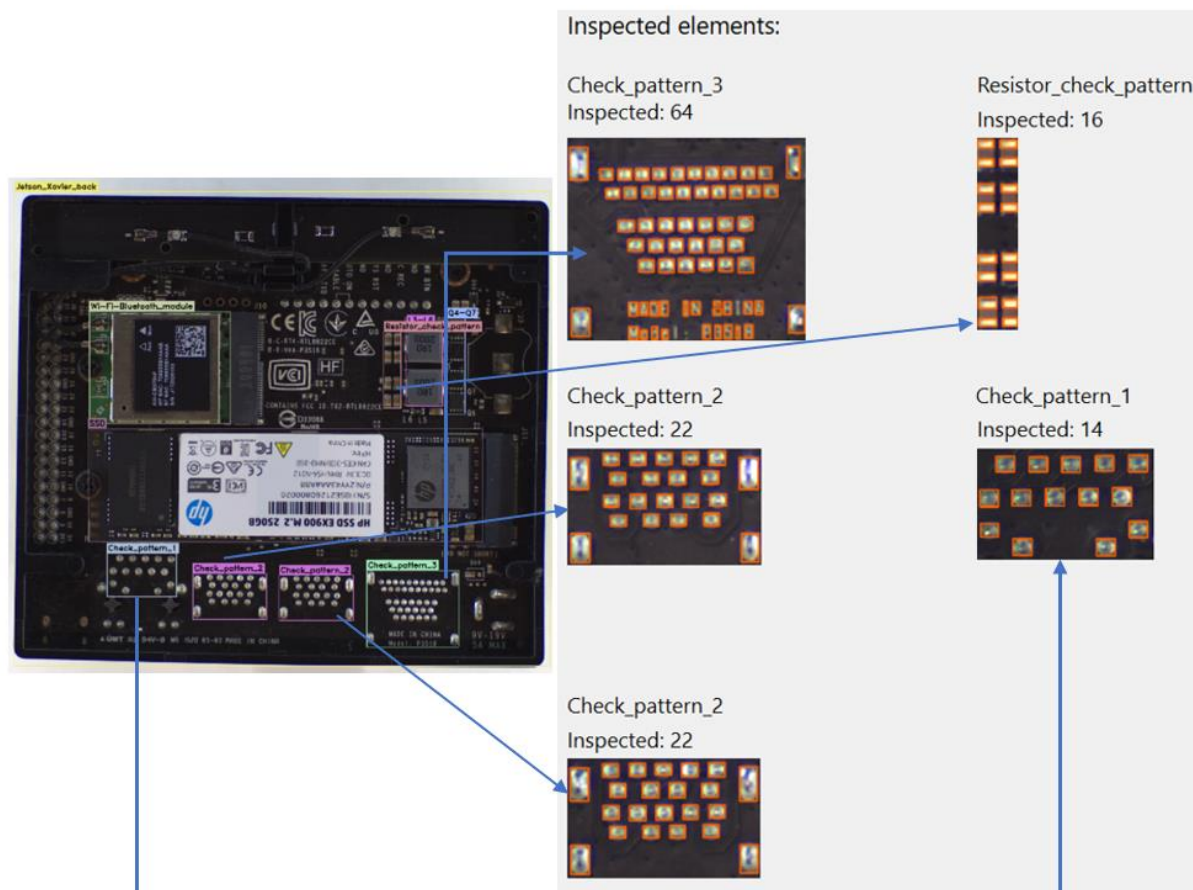


Slika 29. Metodologija inspekcije

Nakon primjene adaptivnog praga nad jednokanalnim slikama, dobiva se binarna crno-bijela slika. Pomoću OpenCV funkcije `findContours()` pronalaze sve zatvorene konture na slici. Funkcija računanjem gradijenata na slici detektira rubove pojedine konture te se detektirani rubovi spajaju u zatvorene konture ako se nalaze dovoljno blisko jedan drugome. Sve konture se spremaju u posebnu listu koja sadrži njihove koordinate. Isti proces se vrši za sve veličine blokova (`blockSize`) te se konture spremaju za naknadno filtriranje. Nakon toga se odabire ona veličina bloka koja ima najmanji broj kontura, što je eksperimentalno dokazano kao najkvalitetnija prilikom traženja pinova ili lemljenih spojeva na slici. Konture se dodatno filtriraju prema veličini, eliminirajući sve konture manje od 30 piksela za zadanu LUCID PHX064S-CC kameru. Konture manje od 30 piksela predstavljaju pogreške metode adaptivnog praga i različite nesavršenosti na mikroprocesorskim pločicama koje je potrebno ukloniti tijekom inspekcije.

Nakon što se konture filtriraju, funkcijom `BoundingRect()` se pronalaze ograničavajuće kućice za svaku preostalu konturu te se kućice spremaju u listu za daljnje filtriranje. Ograničavajuće kućice se dodatno modificiraju i filtriraju kako bi se eliminirale kućice koje se preklapaju. Ako se dvije ograničavajuće kućice preklapaju, smatra se da predstavljaju istu konturu. Stoga se te kućice spajaju u novu, jedinstvenu ograničavajuću kućicu, dok se početne, manje kućice zatim eliminiraju. Na taj način se osigurava da svaki

ispitani element bude detektiran i ispravno ispitan samo jednom. Nakon filtriranja, oko ispitanih elemenata se iscrtavaju ograničavajuće kućice, gdje svaka kućica predstavlja jedan pin ili lemljeni spoj. Iscrtane ograničavajuće kućice se zatim dokumentiraju kako bi se utvrdio točan broj pronađenih kontura. Jedan od primjera uspješne inspekcije su uzorci na stražnjoj strani Jetson NX Xavier mikroprocesorske pločice, prikazani na slici 30.

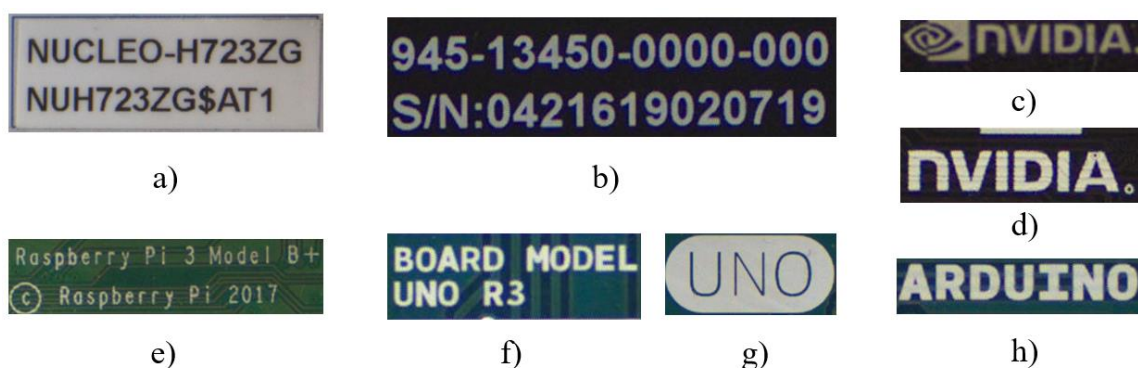


Slika 30. Inspekcija na stražnjoj strani Jetson NX Xavier

## 10. Optičko prepoznavanje teksta

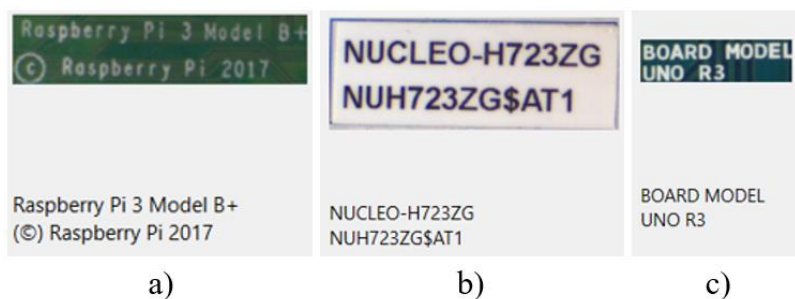
Za optičko prepoznavanje teksta na detektiranim imenima mikroprocesora, proizvođača i modelima mikroprocesorskih pločica koristi se Tesseract paket. Tesseract je open-source paket koji sadrži Tesseract OCR engine koji se koristi za optičko prepoznavanje teksta. Tesseract je razvijen u Helwett-Packard Laboratoriju u Bristolu 1994. godine. Kasnije je razvoj nastavljen pod okriljem Google-a do 2018.[27]. Tesseract OCR engine se sastoji od konvolucijskih neuronskih mreža za optičko prepoznavanje teksta. Modeli mogu prepoznati više od 100 svjetskih jezika na različitim formatima slika kao što su .png, .jepg i .tiff, te spremati podatke u formatima kao .html, .pdf, .tsv i .alto. Tesseract OCR engine je jedan od najtočnijih i najrobusnijih OCR engine-a, što su glavni razlozi za korištenje u slučaju mikroprocesorskih pločica različitog dizajna, boja te vrsta slova.

Na sljedećoj slici su prikazane komponente nad kojima se provodi optičko prepoznavanje teksta. Cilj optičkog prepoznavanja teksta je testiranje ispravnosti teksta te dodatna kontrola detekcije treniranog YOLOv8 modela.



**Slika 31.** Elementi za ispitivanje optičkim prepoznavanjem teksta (a) STM32, b) Jetson Nano, c) i d) Jetson Xavier, e) Raspberry Pi 3, f), g) i h) Arduino Uno

Slika 32 prikazuje neke od rezultata optičkog prepoznavanja teksta korištenjem Tesseract OCR engine-a.

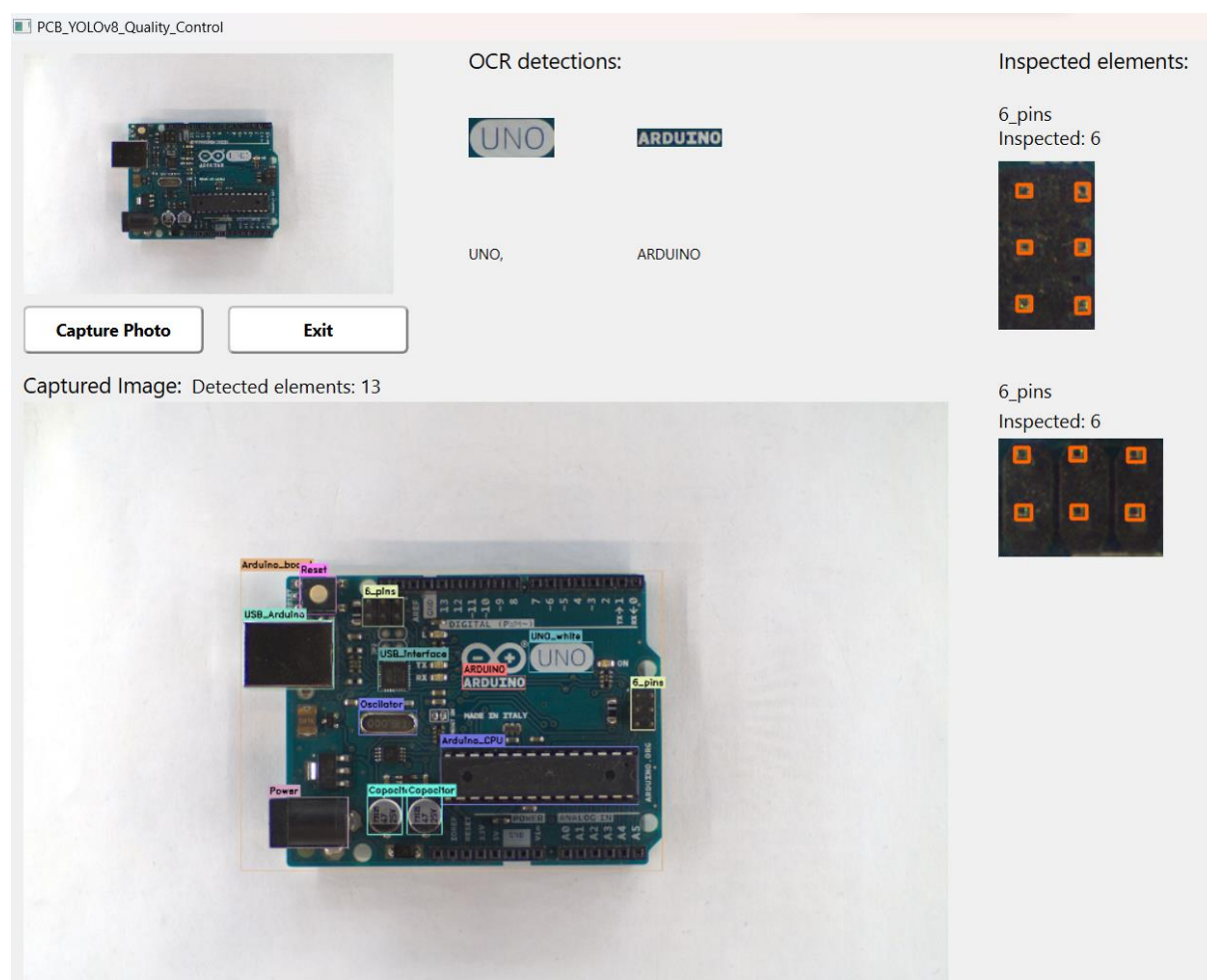


**Slika 32.** Rezultati optičkog prepoznavanja teksta (a) Raspberry Pi 3 model, b) STM32 model, c) Arduino Uno model



## 11. Implementacija sustava

Za provođenje cjelovite kontrole kvalitete mikroprocesorskih pločica, razvijeno je automatizirano rješenje u obliku grafičkog korisničkog sučelja, odnosno GUI-ja (engl. Graphical User Interface). Kompletno rješenje je razvijeno u programskom jeziku C++, koji je prikladan za industrijske primjene s robotskim proizvodnim pogonima. Rješenje se sastoji od Qt GUI-ja koji prikuplja slike s industrijske kamere te ih prikazuje kako bi se pozicionirala mikroprocesorska pločica za kontrolu kvalitete. GUI ima dva glavna gumba: *Exit* za završetak procesa kontrole kvalitete i gumb *Capture Photo* za snimanje, odnosno spremanje određene slike u vremenu kako bi se na njoj izvršila kontrola kvalitete. Sustav je realiziran s *Capture Photo* gumbom kako bi omogućio kontrolu kvalitete u realnom vremenu korištenjem velikog modela YOLOv8 neuronske mreže, čak i uz ograničene računalne resurse. Ovom metodom, resursi se optimalno koriste, samo u specifičnim trenucima tijekom kontrole kvalitete.



Slika 33. GUI za kontrolu kvalitete mikroprocesorskih pločica, primjer Arduino Uno

Nakon što se slika snimi pritiskom na *Capture Photo* gumb, provodi se detekcija, a rezultati se prikazuju na slici s detektiranim komponentama, oko kojih su iscrtane ograničavajuće kućice i nazivi pojedine komponente.

Inspekcija se provodi pomoću posebne .txt datoteke koja sadrži popis svih elemenata za inspekciju unutar zadanog seta podataka. Provjerom je li koji od detektiranih objekata na popisu komponenti za inspekciju, dolazi to izdvajanja segmenta slike i procesa inspekcije. GUI zatim ispisuje rezultate u obliku elemenata s iscrtanim ograničavajućim kućicama oko pinova ili lemljenih spojeva, te prikazuje klasu ispitane komponente i broj pronađenih kućica radi lakše kontrole.

Optička kontrola kvalitete se također provodi pomoću posebne .txt datoteke koja sadrži popis svih elemenata za optičku kontrolu unutar zadanog seta podataka. Rezultati optičke kontrole se sastoje od ispisa ispitane komponente ispod koje je ispisan rezultat optičke kontrole.

Ponovnim pritiskom na *Capture Photo* gumb, kontrola kvalitete se iznova pokreće. Proces detekcije, inspekcije i optičke kontrole kvalitete korištenjem NVIDIA GeForce GTX 1650 grafičke kartice i LUCID PHX064S-CC s najvećom kvalitetom slike traje 225 milisekundi. Kompletno rješenje je dostupno na GitHub repozitoriju<sup>1</sup>.

---

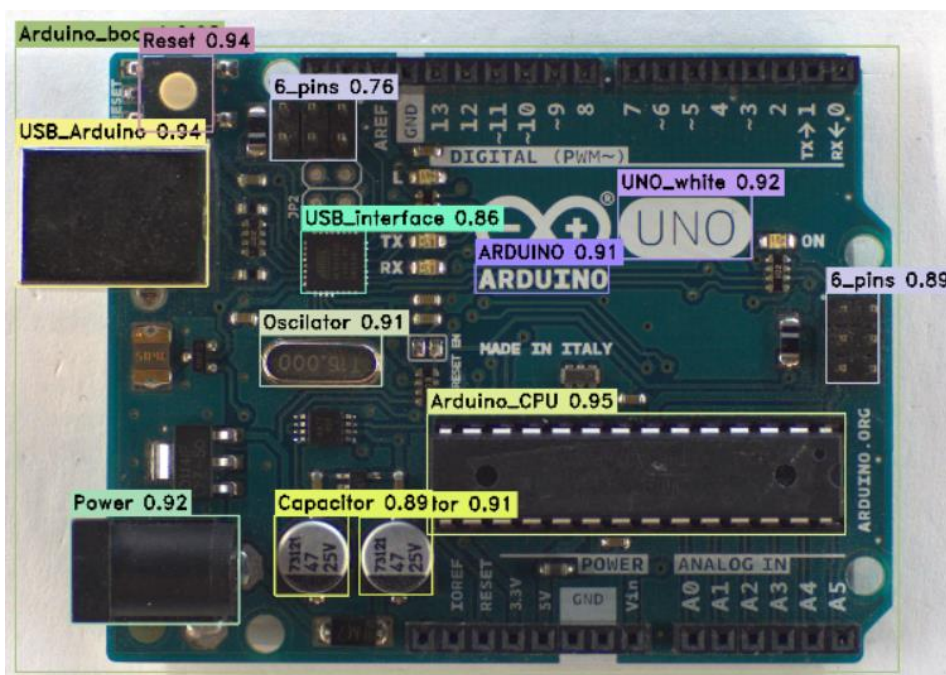
<sup>1</sup>[https://github.com/lukasiktar/PCB\\_Quality\\_Control\\_with\\_ArenaSDK.git](https://github.com/lukasiktar/PCB_Quality_Control_with_ArenaSDK.git)

## 12. Rezultati

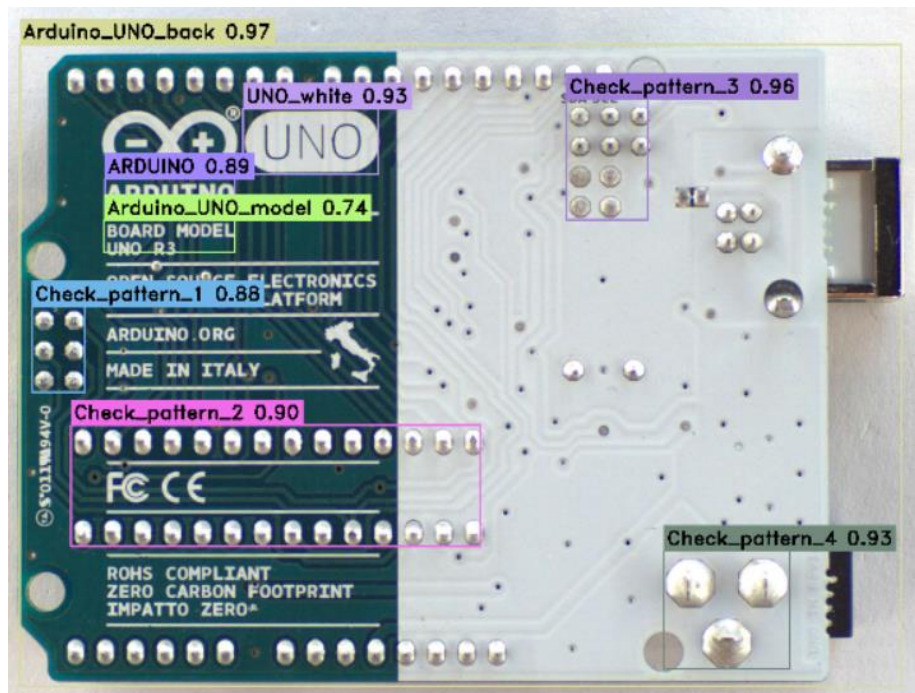
Rezultati ispitivanja sustava za kontrolu kvalitete izrade mikroprocesorskih pločica, podijeljeni su u segmente detekcije, inspekcije i optičkog prepoznavanja teksta. Rezultati detekcije predstavljaju performanse treniranog YOLOv8 modela na realnim primjerima eksploatacije u industriji. Rezultati modela su prikazani standardiziranim metrikama kao što su *precision* i *recall* u prilogu 1, a u nastavku je izračunata srednja vrijednost sigurnosti detekcije za komponente mikroprocesorskih pločica. Rezultati inspekcije su također prikazani *precision* i *recall* metrikama zasebno za svaku mikroprocesorsku pločicu. Za svaku pločicu je izračunata i točnost optičkog prepoznavanja teksta. Parametri kamere za provođenje analize sustava su postavljeni na sljedeći način: vrijeme ekspozicije je postavljeno na 17 milisekundi, piksel format je RGB8, pojačanje svjetline je 1 dB i *gamma* faktor korekcije je 0.6.

### 12.1. Arduino Uno

Rezultati detekcije nad Arduino Uno mikroprocesorskom pločicom provedene treniranim YOLOv8 modelom su prikazani na slici 34.



a) Prednja strana



b) Stražnja strana

### Slika 34. Rezultati detekcije komponenti na Arduino Uno mikroprocesorskoj pločici

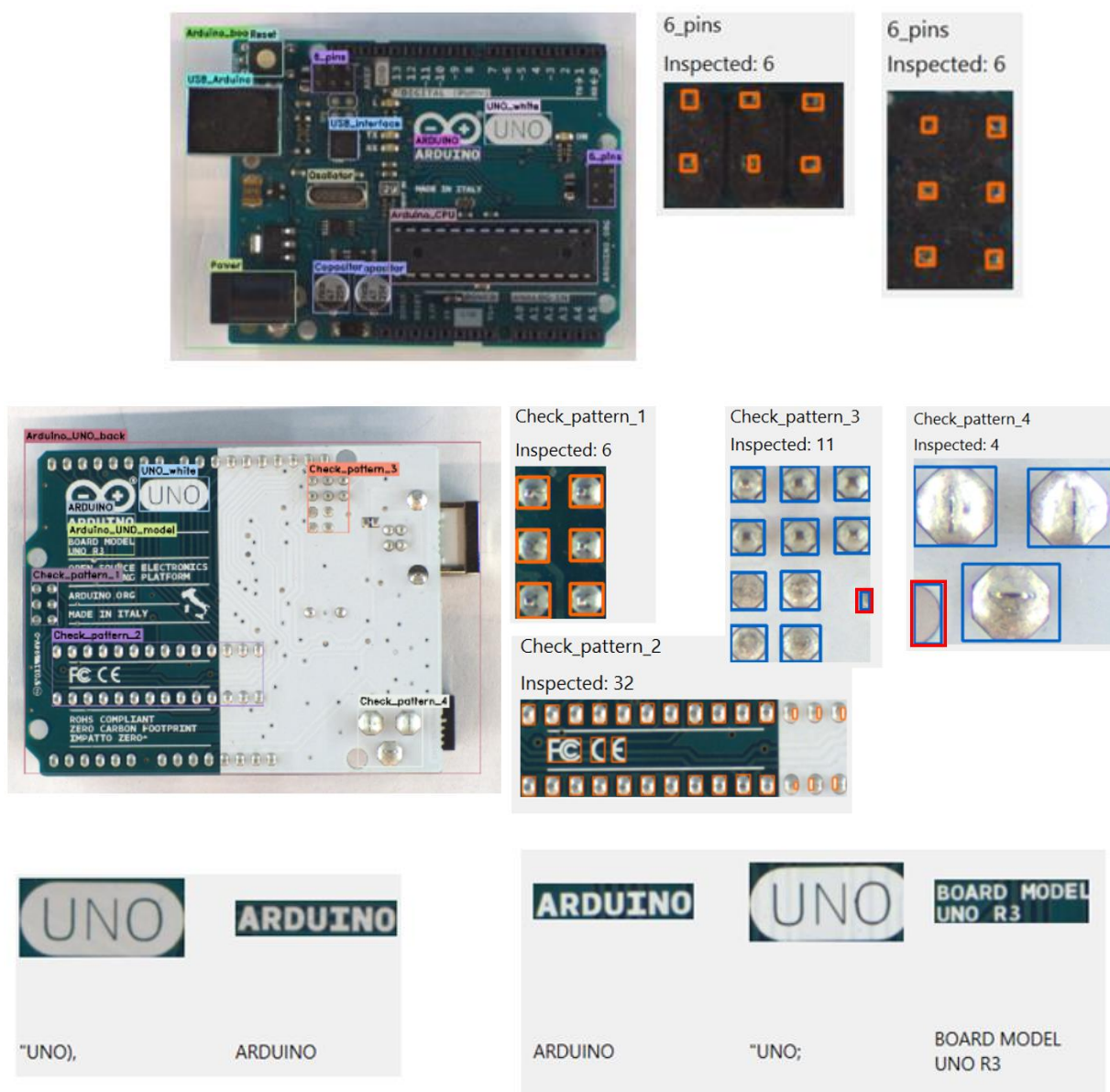
Na prednjoj strani je detektirana mikroprocesorska pločica te svih označenih 12 komponenti, a na stražnjoj strani je osim mikroprocesorske pločice detektirano i također svih označenih 7 komponenti. Provođenjem detekcije za promjenjive parametre kamere u obliku promjene vremena ekspozicije u rasponu  $\pm 10\%$ , te dodavanja pojačanja  $\pm 4\text{dB}$ , dobiveni su rezultati detekcije prikazani u tablici 10 koja prikazuje srednje vrijednosti sigurnosti detekcije treniranog YOLOv8 modela na ispitnim mikroprocesorskim pločicama. Ograničavajuće kućice su vrlo precizne te imaju visoku sigurnost pri klasifikaciji objekata.

Tablica 10. Rezultati detekcije nad Arduino Uno mikroprocesorskoj pločici

Komponenta	Sigurnost detekcije	Komponenta	Sigurnost detekcije
Arduino_board	98,77%	Arduino_UNO_back	96,85%
Reset	93,23%	Check_pattern_1	87,77%
USB_Arduino	96,91%	Check_pattern_2	91,91%
Power	91,32%	Check_pattern_3	97,63%
6_pins	87,84%	Check_pattern_4	92,44%
USB_interface	92,83%	ARDUINO	91,84%
Oscilator	95,11%	UNO_white	91,33%
Capacitor	90,84%	Arduino_UNO_model	73,65%
Arduino_CPU	98,73%	$\Sigma$	<b>92,19%</b>
ARDUINO	92,33%		
UNO_white	90,18%		

U nastavku su prikazani rezultati inspekcije na zadanim izdvojenim elementima. Provedena inspekcija je pokazala robusnost i postigla *precision* od 96,4% i *recall* od 89,8%. Na

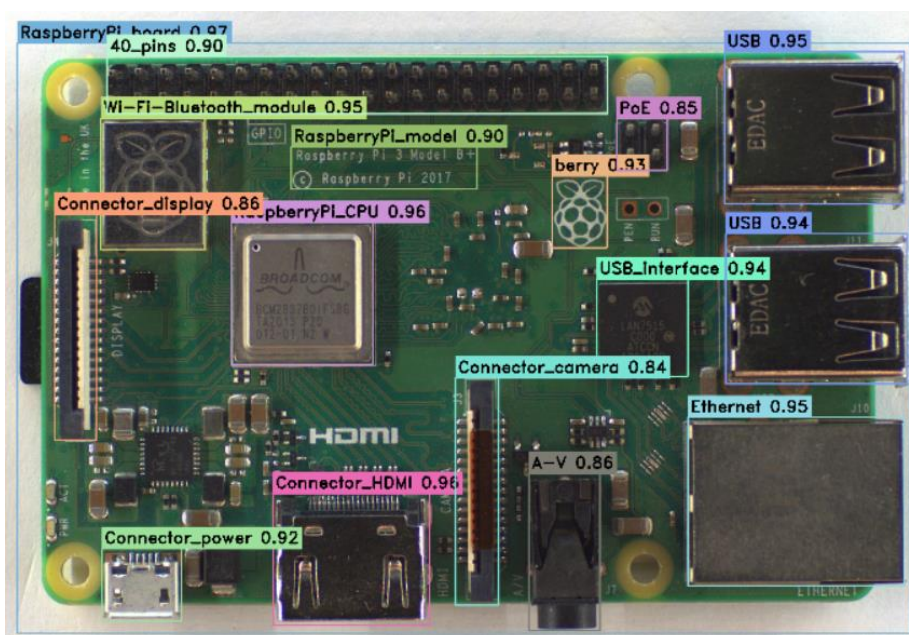
slici 35 je moguće vidjeti dvije pogrešne detekcije lemljenih spojeva označenih crvenom bojom kod *Check\_pattern\_3* i *Check\_pattern\_4* jer su nedostaci površine prepoznati kao potencijalni lemljeni spojevi. Razlog detekcije nedostataka se pripisuje pozadini svjetlijoj od lemljenih spojeva, odnosno slučaju u kojem se koristi THRESH\_BINARY\_INV, gdje se segmentiraju tamne konture na slici, a to su u ovom slučaju spojevi i nedostaci na površini. Uz rezultate inspekcije su prikazani i rezultati optičkog prepoznavanja teksta koje je pokazalo potpuno točno prepoznavanje osim u slučaju klase *Uno\_white* gdje je prilikom označavanja u procesu stvaranja trening seta uz sami tekst označena i podloga.



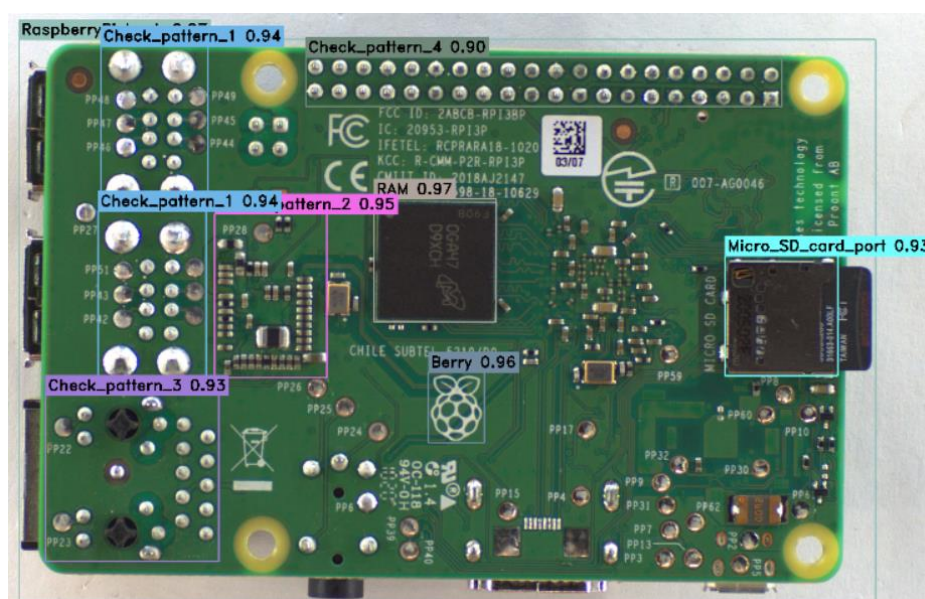
**Slika 35. Rezultati inspekcije i optičkog prepoznavanja teksta komponenti na Arduino Uno mikroprocesorskoj pločici**

## 12.2. Raspberry Pi 3

Rezultati detekcije nad Raspberry Pi 3 mikroprocesorskom pločicom provedene treniranim YOLOv8 modelom su prikazani na slici 36.



a) Prednja strana



b) Stražnja strana

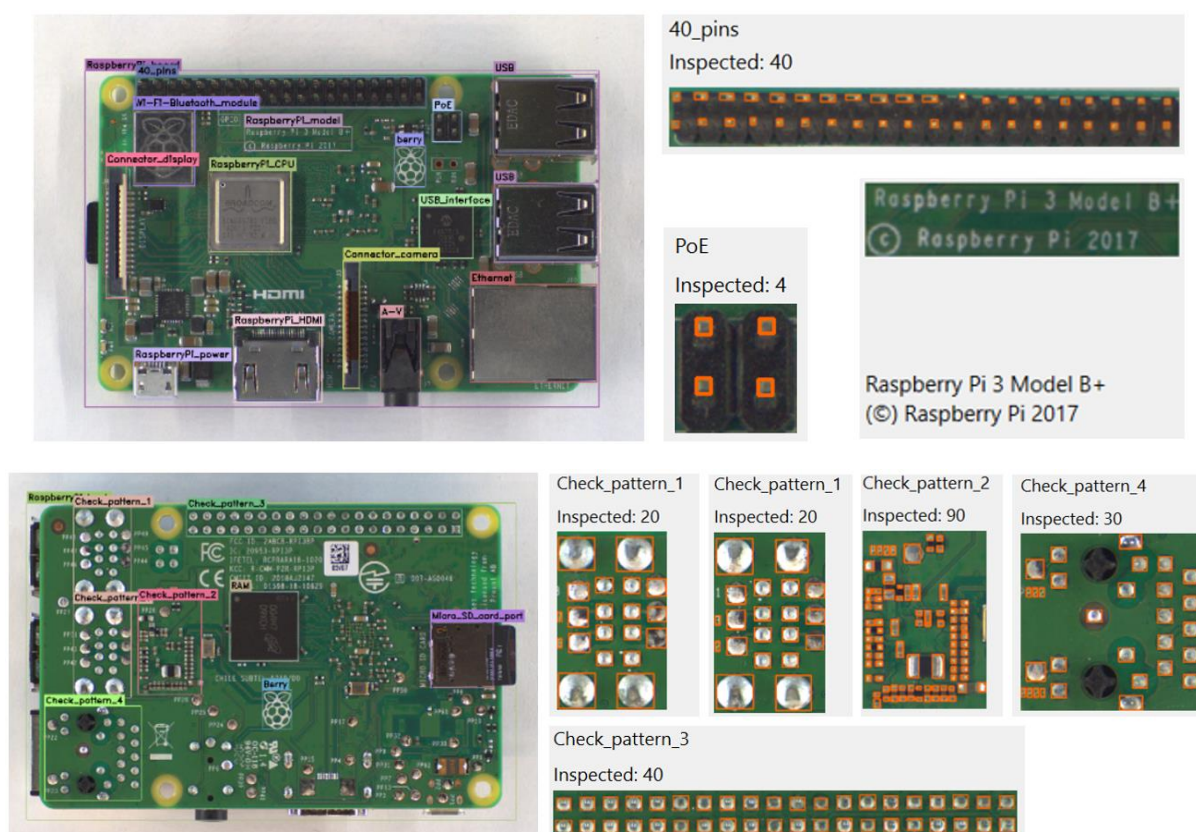
**Slika 36.** Rezultati detekcije komponenti na Raspberry Pi 3 mikroprocesorskoj pločici

Na prednjoj strani je detektirana mikroprocesorska pločica te svih označenih 15 komponenti, a na stražnjoj strani je osim mikroprocesorske pločice detektirano i također svih označenih 8 komponenti. U tablici 11. su dani rezultati detekcije provedene pri različitim parametrima kamere kako je opisano u poglavlju 12.1.

Tablica 11. Rezultati detekcije nad Raspberry Pi 3 mikroprocesorskoj pločici

Komponenta	Sigurnost detekcije	Komponenta	Sigurnost detekcije
RaspberryPi_front	97,63%	RaspberryPi_back	98,43%
40_pins	92,66%	Check_pattern_1	95,41%
PoE	84,11%	Check_pattern_2	95,83%
USB	95,98%	Check_pattern_3	93,12%
Wi-Fi/Bluetooth module	93,82%	Check_pattern_4	94,54%
RaspberryPi_model	89,77%	RAM	96,23%
Berry	94,14%	Berry	94,33%
Connector_display	87,71%	Micro_SD_Card_port	92,14%
RaspberryPi_CPU	97,92%	$\Sigma$	<b>92,96%</b>
USB_interface	94,44%		
Connector_power	90,18%		
Connector_HDMI	95,23%		
Connector_camera	83,33%		
A/V	84,42%		
Ethernet	96,67%		

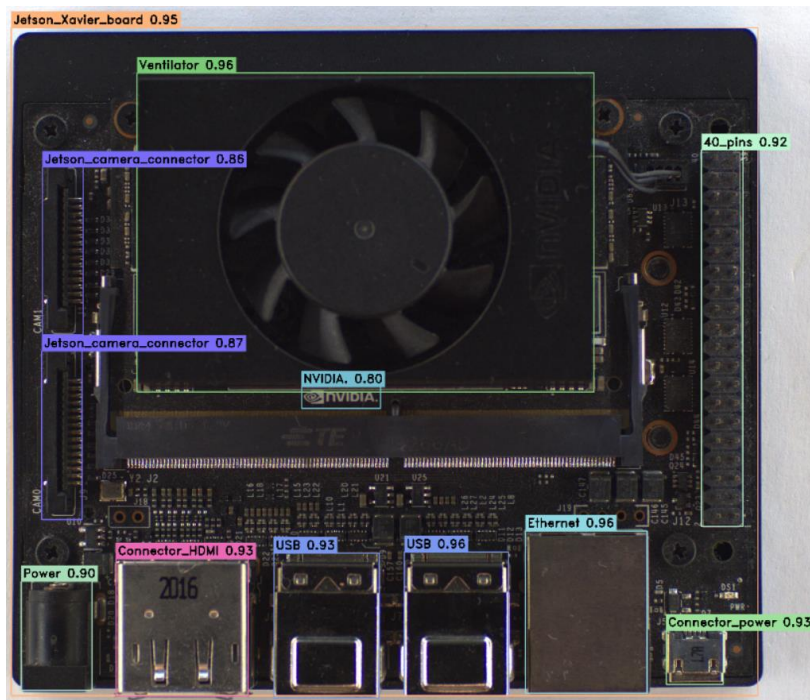
U nastavku su prikazani rezultati inspekcije na zadanim izdvojenim elementima. Provedena inspekcija je pokazala robusnost i postigla *precision* od 96.8% i *recall* od 100%.



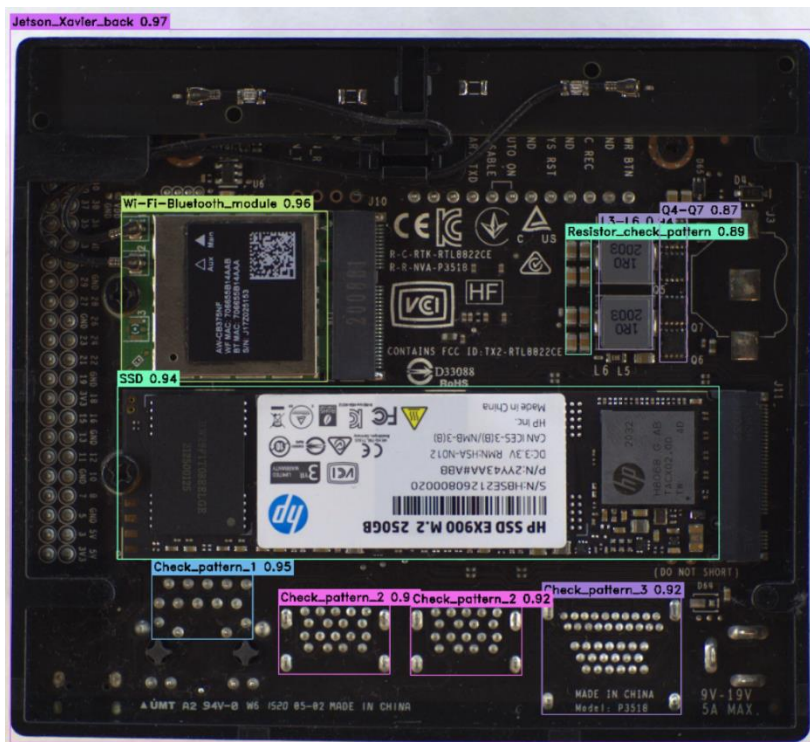
Slika 37. Rezultati inspekcije i optičkog prepoznavanja teksta komponenti na Raspberry Pi 3 mikroprocesorskoj pločici

### 12.3. Jetson NX Xavier

Rezultati detekcije nad Jetson NX Xavier mikroprocesorskom pločicom provedene treniranim YOLOv8 modelom su prikazani na slici 38.



a) Prednja strana



b) Stražnja strana

Slika 38. Rezultati detekcije komponenti na Jetson NX Xavier mikroprocesorskoj pločici

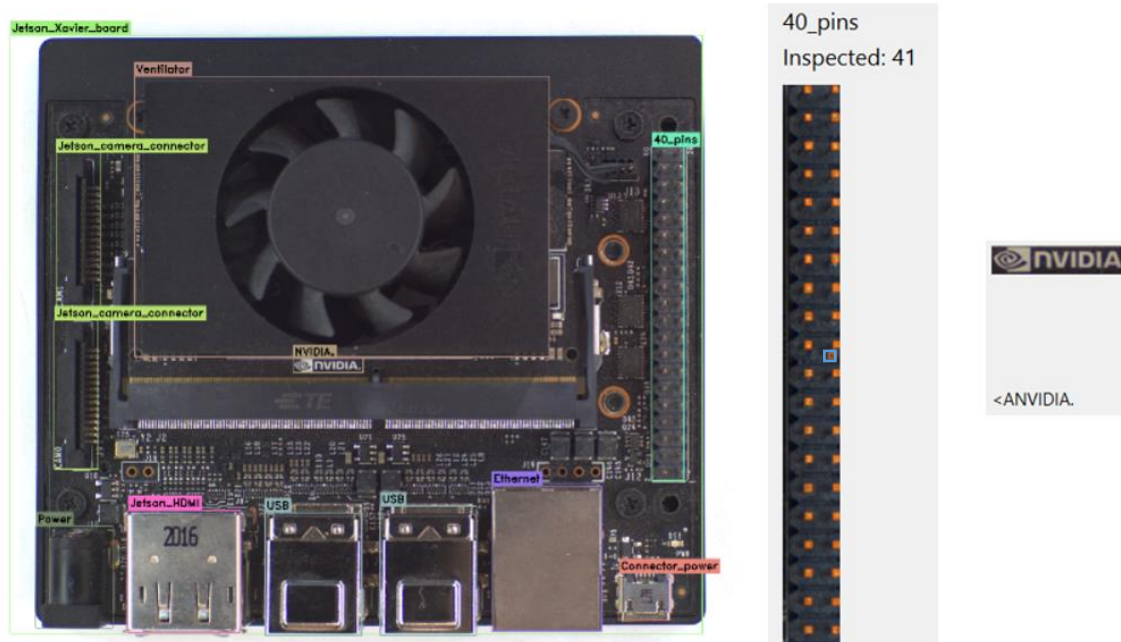


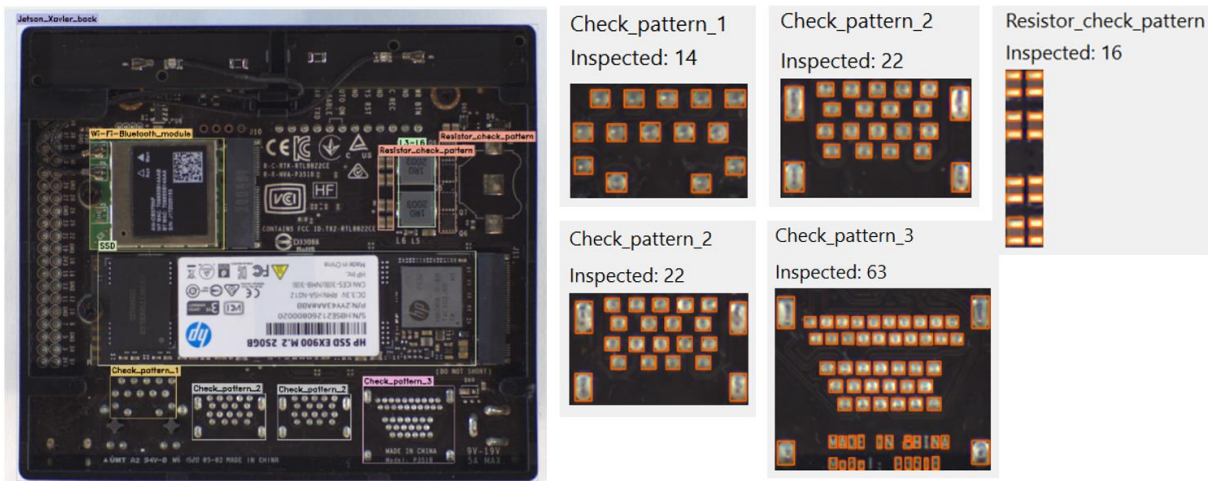
Na prednjoj strani je detektirana mikroprocesorska pločica te svih označenih 11 komponenti, a na stražnjoj strani je osim mikroprocesorske pločice detektirano i također svih označenih 9 komponenti. U tablici 12 su dani rezultati detekcije provedene pri različitim parametrima kamere kako je opisano u poglavlju 12.1.

**Tablica 12. Rezultati detekcije nad Jetson NX Xavier mikroprocesorskoj pločici**

Komponenta	Sigurnost detekcije	Komponenta	Sigurnost detekcije
Jetson_Xavier_board	94,47%	Jetson_Xavier_back	96,51%
Ventilator	95,21%	Check_pattern_1	94,97%
Jetson_camera_connector	89,44%	Check_pattern_2	93,11%
NVIDIA.	81,82%	Check_pattern_3	92,61%
40_pins	91,81%	Resistor_check_pattern	88,24%
Power	91,13%	L3-L6	91,88%
Jetson_HDMI	95,11%	Q4-Q7	86,42%
USB	93,12%	SSD	93,05%
Ethernet	95,53%	Wi-Fi/Bluetooth_module	97,54%
Connector_power	92,81%	Σ	<b>92,35%</b>

U nastavku su prikazani rezultati inspekcije na zadanim izdvojenim elementima. Provedena inspekcija je pokazala robusnost i postigla *precision* od 98,7% i *recall* od 98,1%. Na slici 39, na prednjoj strani detektirana je jedna pogrešno pozitivna kontura označena plavom kućicom, dok na stražnjoj strani nema pogrešaka. Također, označavani NVIDIA logo unosi pogrešku prilikom optičke kontrole teksta.

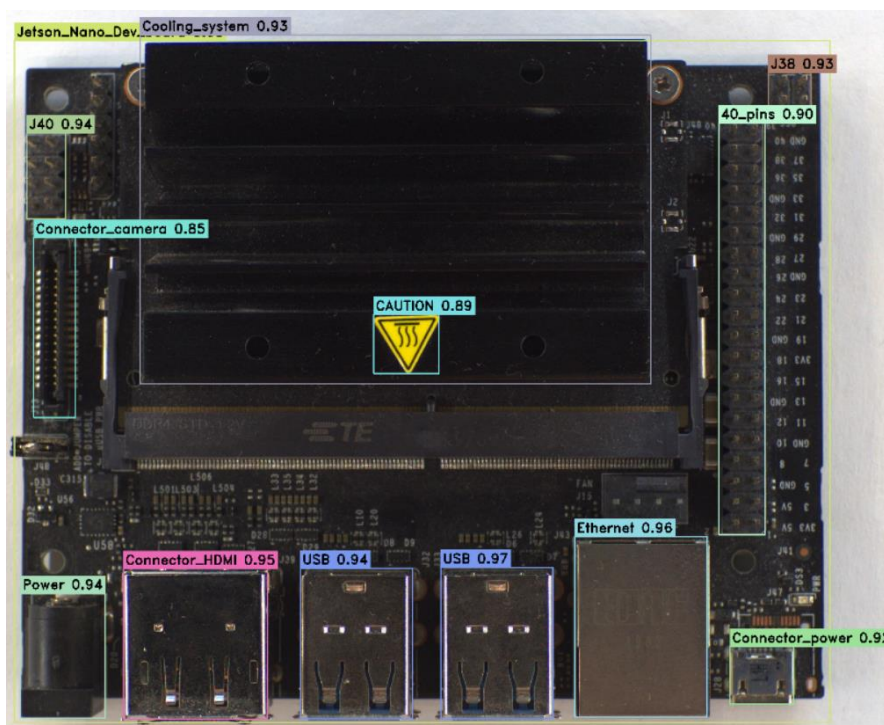




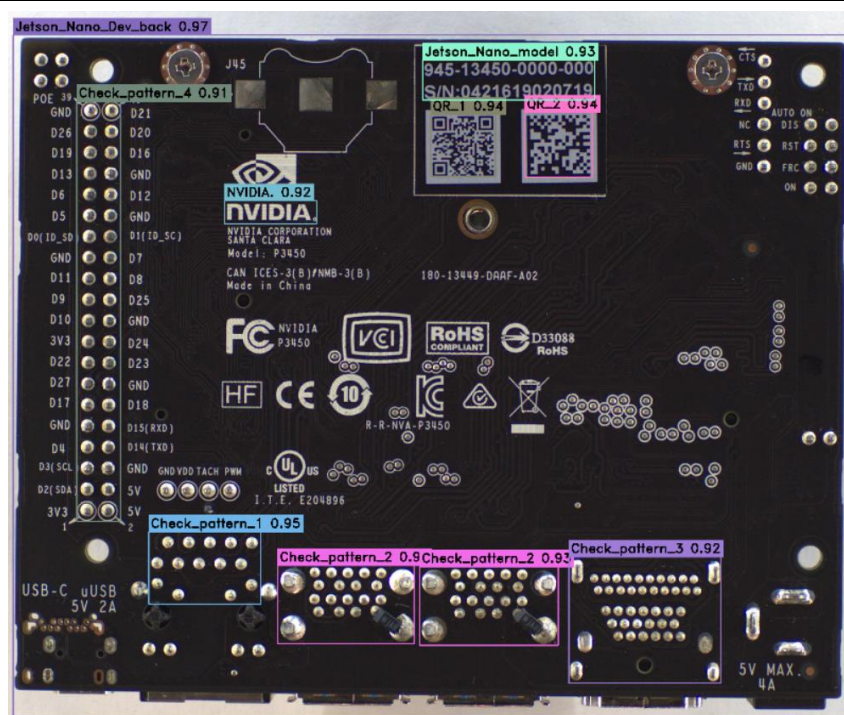
Slika 39. Rezultati inspekcije i optičkog prepoznavanja teksta na Jetson NX Xavier mikroprocesorskoj pločici

#### 12.4. Jetson Nano

Rezultati detekcije nad Jetson Nano mikroprocesorskom pločicom provedene treniranim YOLOv8 modelom su prikazani na slici 40. Na prednjoj strani je detektirana mikroprocesorska pločica te svih označenih 12 komponenti, a na stražnjoj strani je osim mikroprocesorske pločice detektirano i također svih označenih 9 komponenti.



a) Prednja strana



b) Stražnja strana

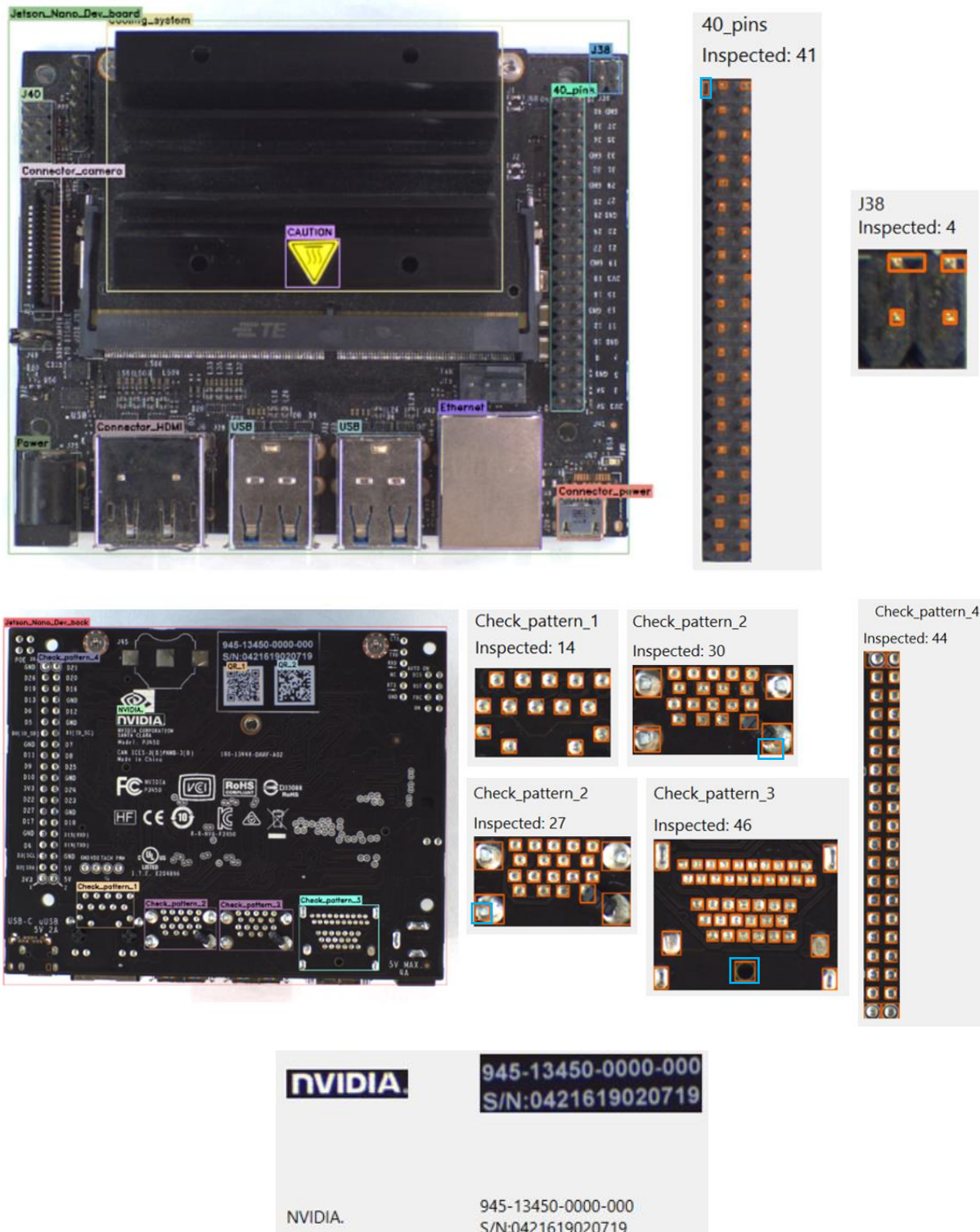
**Slika 40. Rezultati detekcije na Jetson Nano mikroprocesorskoj pločici**

U tablici 13 su dani rezultati detekcije provedene pri različitim parametrima kamere kako je opisano u poglavlju 12.1. Ograničavajuće kućice su vrlo precizne te imaju visoku sigurnost pri klasifikaciji objekata.

**Tablica 13. Rezultati detekcije nad Jetson Nano mikroprocesorskoj pločici**

Komponenta	Sigurnost detekcije	Komponenta	Sigurnost detekcije
Jetson_Nano_board	96,12%	Jetson_Nano_back	97,22%
J40	91,76%	Check_pattern_1	94,17%
J38	93,53%	Check_pattern_2	91,21%
Connector_camera	84,64%	Check_pattern_3	93,65%
40_pins	89,15%	Jetson_Nano_model	93,41%
Power	94,72%	NVIDIA.	89,41%
Jetson_HDMI	96,13%	QR_1	94,12%
USB	95,73%	QR_2	95,87%
Ethernet	96,32%	Σ	<b>93,18%</b>
Connector_power	93,19%		
CAUTION	90,11%		

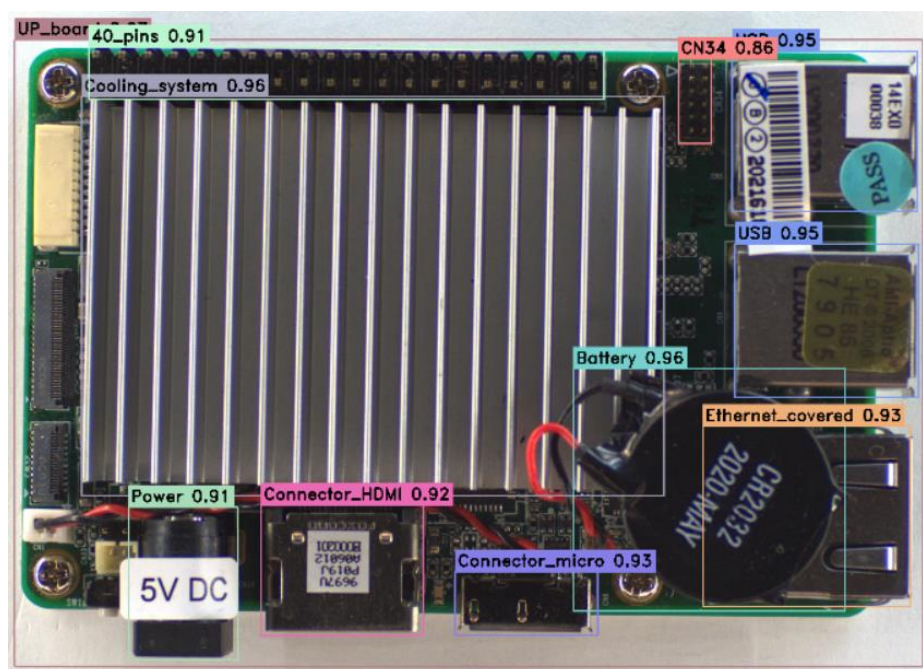
U nastavku su prikazani rezultati inspekcije na zadanim izdvojenim elementima. Provedena inspekcija je pokazala dobre performanse i postigla *precision* od 97,9% i *recall* od 100%. Na prednjoj i stražnjoj strani su četiri pogrešno pozitivne detekcije označene plavom bojom. Optičko prepoznavanje teksta na danim primjerima radi bez pogreške.



Slika 41. Rezultati inspekcije i optičkog prepoznavanja teksta na Jetson Nano mikroprocesorskoj pločici

## 12.5. Up board

Rezultati detekcije nad Up board mikroprocesorskom pločicom provedene treniranim YOLOv8 modelom su prikazani na slici 42.



a) Prednja strana



b) Stražnja strana

**Slika 42. Rezultati detekcije na Up Board mikroprocesorskoj pločici**

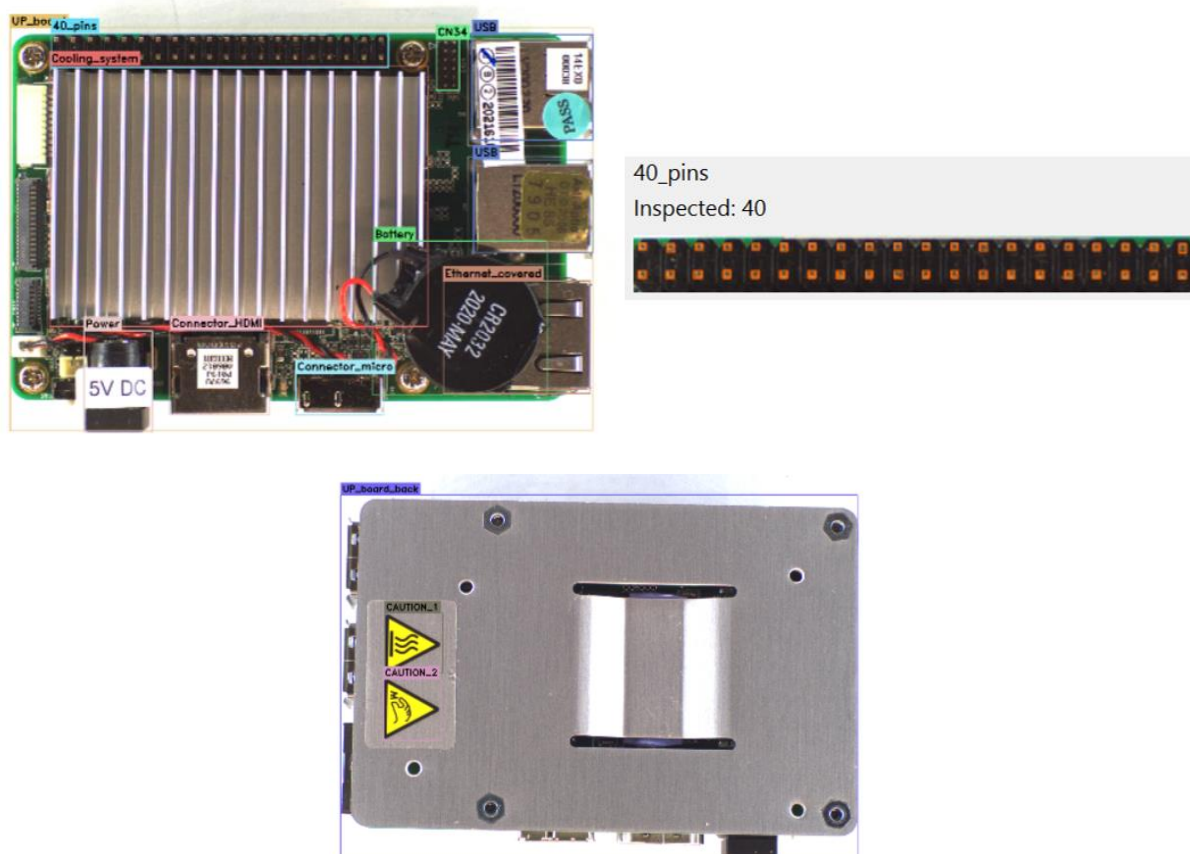
Na prednjoj strani je detektirana mikroprocesorska pločica te svih označenih 10 komponenti, a na stražnjoj strani je osim mikroprocesorske pločice detektirane označene dvije

komponente. U tablici 14. su dani rezultati detekcije provedene pri različitim parametrima kamere kako je opisano ranije.

**Tablica 14. Rezultati detekcije nad Up Board mikroprocesorskoj pločici**

Komponenta	Sigurnost detekcije	Komponenta	Sigurnost detekcije
Up_board	97,63%	Up_board_back	96,09%
40_pins	91,83%	CAUTION_1	94,11%
CN34	85,03%	CAUTION_2	94,32%
USB	85,11%	$\Sigma$	<b>92,85%</b>
Cooling_system	97,35%		
Battery	95,01%		
Power	94,96%		
Connector_HDMI	92,26%		
Ethernet_covered	92,32%		
Connector_micro	91,12%		

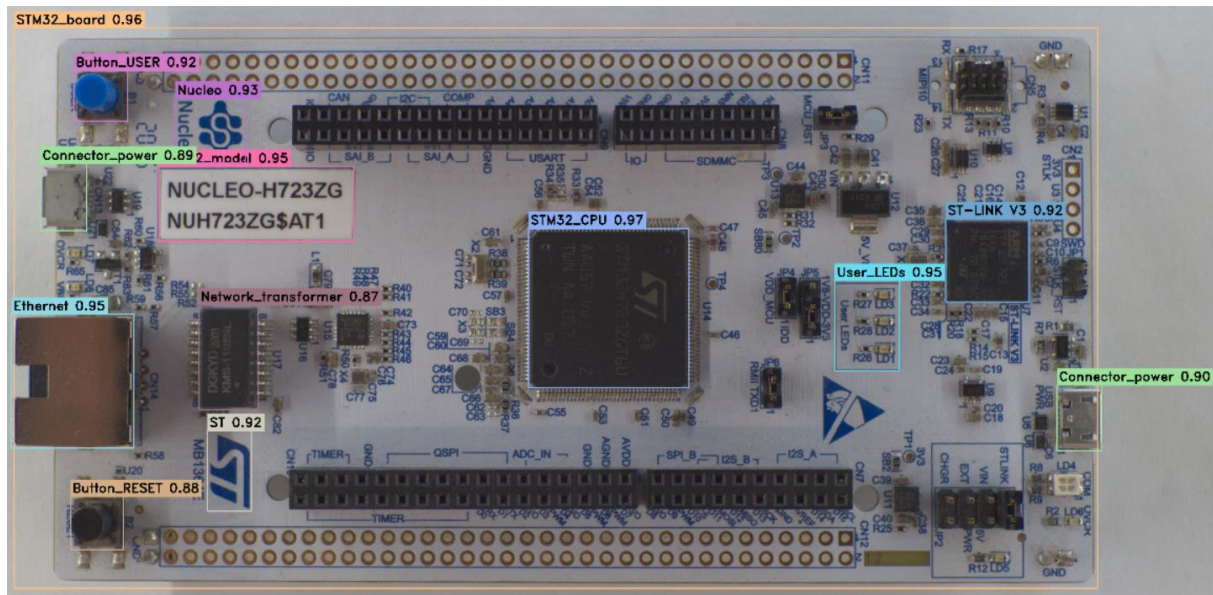
U nastavku su prikazani rezultati inspekcije na zadanim izdvojenim elementima. Provedena inspekcija je pokazala dobre performanse i postigla *precision* od 100% i *recall* od 100%.



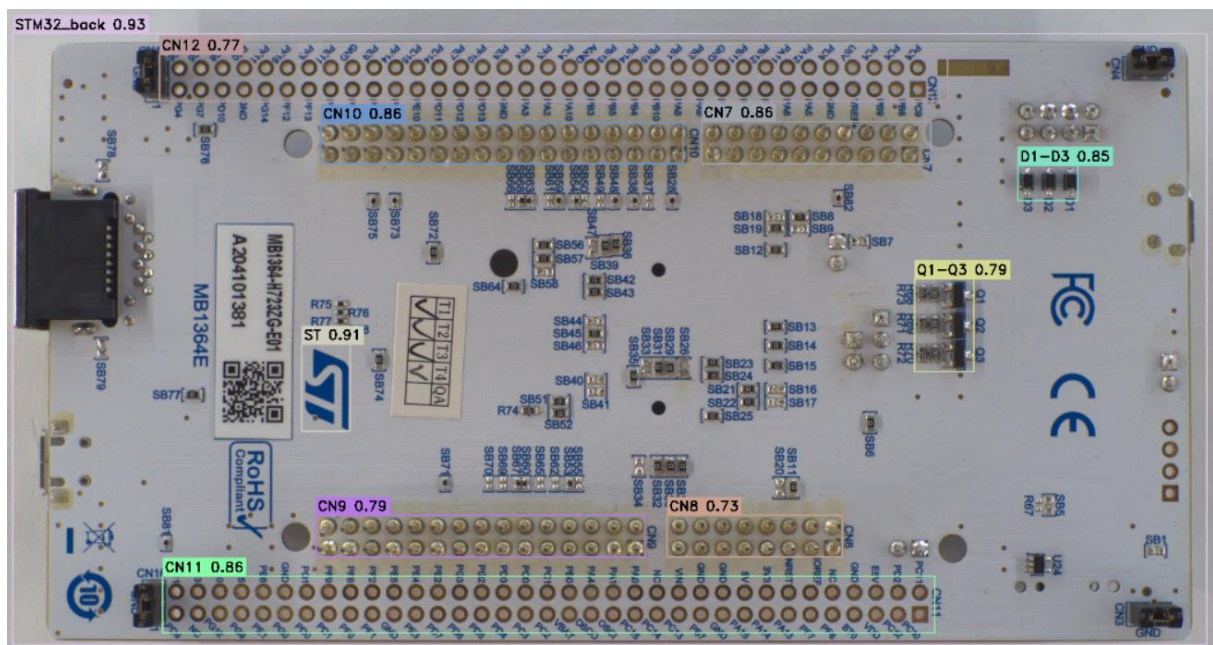
**Slika 43. Rezultati inspekcije na Up Board mikroprocesorskoj pločici**

## 12.6. STM32

Rezultati detekcije nad STM32 mikroprocesorskom pločicom provedene treniranim YOLOv8 modelom su prikazani na slici 44. Na prednjoj strani je detektirana mikroprocesorska pločica te svih označenih 12 komponenti, a na stražnjoj strani je osim mikroprocesorske pločice detektirano i također svih označenih 9 komponenti.



a) Prednja strana



b) Stražnja strana

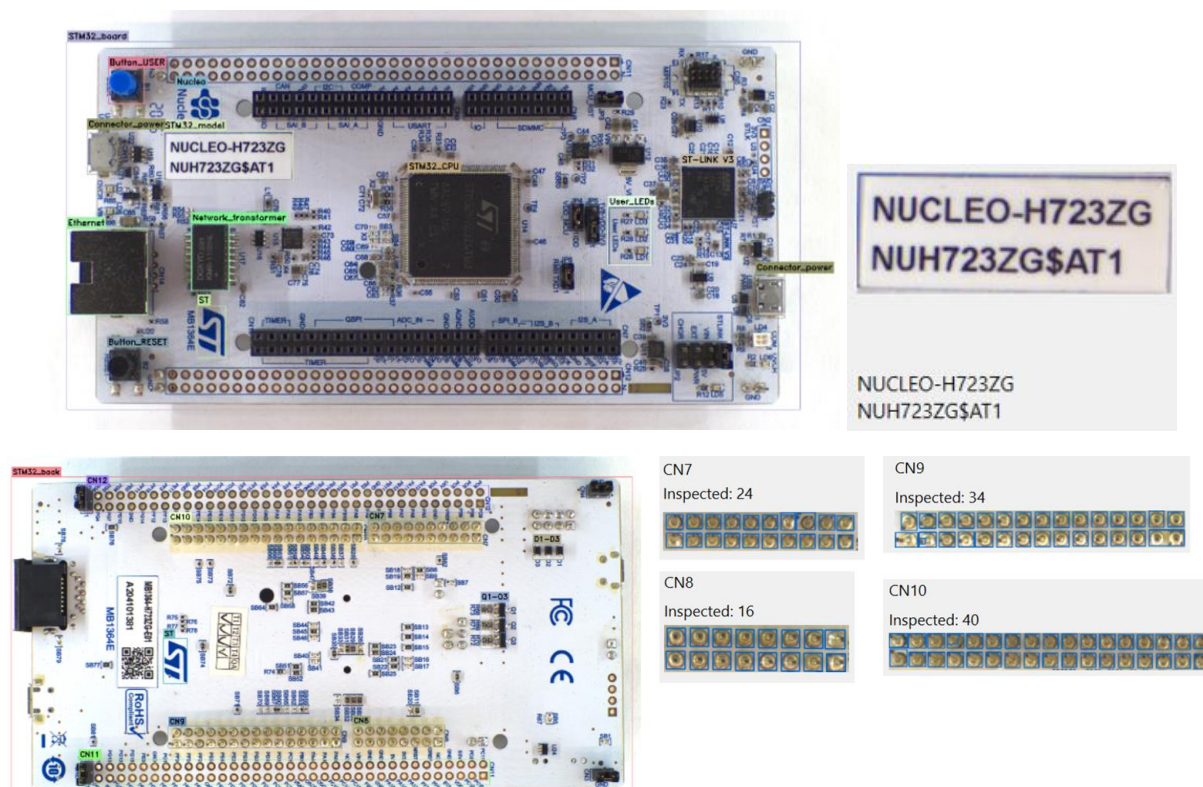
**Slika 44. Rezultati detekcije na STM32 mikroprocesorskoj pločici**

U tablici 15. su dani rezultati detekcije provedene pri različitim parametrima kamere kako je opisano ranije.

Tablica 15. Rezultati detekcije nad STM32 mikroprocesorskoj pločici

Komponenta	Sigurnost detekcije	Komponenta	Sigurnost detekcije
STM32_board	95,82%	STM32_back	94,90%
Button_USER	93,22%	CN7	83,51%
Nucleo	91,63%	CN8	72,26%
Connector_power	88,83%	CN9	87,12%
Ethernet	96,51%	CN10	89,86%
STM32_model	95,62%	CN11	88,83%
Network_transformer	90,02%	CN12	79,52%
ST	91,16%	ST	92,13%
Button_RESET	88,01%	Q1-Q3	80,11%
STM32_CPU	96,72%	D1-D3	83,97%
User_LEDs	93,12%	$\Sigma$	<b>89,38%</b>
ST-LINK V3	93,59%		

U nastavku su prikazani rezultati inspekcije na zadanim izdvojenim elementima. Provedena inspekcija je pokazala dobre performanse i postigla *precision* od 86,1% i *recall* od 99%. Smanjena preciznost je uzrokovana bliskim intenzitetima boja pinova i pozadine na jednokanalnoj slici nad kojom se provodi inspekcija. Optičko prepoznavanje teksta radi bez pogreške.



Slika 45. Rezultati inspekcije i optičkog prepoznavanja teksta na STM32 mikroprocesorskoj pločici



### 13. ZAKLJUČAK

Ovo istraživanje prezentira inovativno rješenje<sup>2</sup> automatizirane kontrole kvalitete mikroprocesorskih pločica koje kombinira različite metode računalnog vida i učenja. Sustav kontrole kvalitete koristi algoritme računalnog vida i konvolucijske neuronske mreže za provođenje detekcije, verifikacije, inspekcije i optičke kontrole teksta na najznačajnijim mikroprocesorskim pločicama za industrijsku i komercijalnu primjenu. Detekcija i verifikacija mikroprocesorskih pločica se provodi YOLOv8 konvolucijskom neuronskom mrežom za koju je proveden trening. Za potrebe treninga je stvoren set trening podataka koji se inicijalno sastoji od 1200 slika mikroprocesorskih pločica. Augmentacijom se set podataka povećava na 2895 trening, te 135 validacijskih i 100 test slika. Trenirani model postiže mAP vrijednost od 99.5% za 50% IoU i 95,4% za 95% IoU što dokazuje robusnost i vrhunske performanse detekcije traženih objekata. Prosječna sigurnost detekcije mikroprocesorskih pločica i njihovih komponenti iznosi 92,15%. Inspekcija se provodi korištenjem metoda adaptivnog praga i pronalaženja kontura pri čemu se kontrolira broj i kvaliteta pinova te lemljenih spojeva na mikroprocesorskim pločicama. Preciznost provedene inspekcije za sve mikroprocesorske pločice iznosi 95,9%, a pokrivenost (engl. recall) iznosi 97,87%. Optičko prepoznavanje teksta se provodi nad ispisanim imenima modela mikroprocesorskih pločica i imenima njihovih proizvođača. Točnost prepoznavanja teksta za sve mikroprocesorske pločice iznosi 95,63%. Navedeni procesi su povezani u cjelinu te je stvoren GUI za jednostavnu primjenu sustava. GUI omogućuje pokretanje cijelog sustava i provođenje kontrole kvalitete nad snimljenim mikroprocesorskim pločicama. Navedeno rješenje omogućuje primjenu velikih i kompleksnih modela za kontrolu kvalitete u realnom vremenu na računalima različitih procesorskih snaga. Za potrebe analize sustava i kasniju eksploataciju, konstruiran je postav s industrijskom kamerom i industrijskim osvjetljenjem.

Kompletno rješenje je odabirom tehnologija prilagođeno primjeni u robotskom okruženju proizvodnje mikroprocesorskih pločica kako bi se stvorio efikasan proces od početne „prazne“ pločice do proizvedenog mikroprocesora.

---

<sup>2</sup> [https://github.com/lukasiktar/PCB\\_Quality\\_Control\\_with\\_ArenaSDK.git](https://github.com/lukasiktar/PCB_Quality_Control_with_ArenaSDK.git)

## LITERATURA

- [1] Printed Circuit Board Market – Global Industry Analysis, Size, Share, Growth, Trends, Regional Outlook, and Forecast 2023-2032, <https://www.precedenceresearch.com/printed-circuit-board-market> , Pristup: 20.10.2023.
- [2] A.J. Crispin,V. Rankov, “Automated inspection of PCB components using a genetic algorithm template-matching approach“, Int. J. Manuf. Technol., vol.35, pp.293-300, 2007.
- [3] M. Janczki et. al., “Automatic Optical Inspection for Soldering”, Materials Science – Advanced Topics, 2013.
- [4] A. Price, “Our new 3D Automatic Optical Inspection Machine Has Arrived”, <https://www.corintech.com/news/posts/2017/may/our-new-3d-automatic-optical-inspection-machine-has-arrived/>, Pristup: 08.11.2024.
- [5] F. Raihan, W. Ce, “PCB Defect Detection Using OPENCV with Image Subtraction Method”, 2017 ICIMTech, pp.204-209, 2017.
- [6] M. Baygin et al., “Machine vision based defect detection approach using image processing”, 2017 IDAP, pp.1-5
- [7] M Sathiaselan et. al., “Why Is Deep Learning Challenging for Printed Circuit Board (PCB) Component Recognition and How Can We Address It?”, Cryptography, vol. 5, 2021.
- [8] F. Akhyar et. al., “Observation of Attention Mechanism Baseline for PCB Surface Inspection System”, 2022 APWiMob, pp.1-6, 2022.
- [9] J. Redmon et. al., “You Only Look Once: Unified, Real-Time Object Detection”, 2016 IEEE Conference on Computer Vision and Pattern Recognition”, pp.778-788, 2016.
- [10] J. Greeling, “How Raspberry Pis are made (Factory tour)“, <https://www.jeffgeerling.com/blog/2023/how-raspberry-pis-are-made-factory-tour>, Pristup:28.12.2023.
- [11] K. O'Shea, R. Nash, “An Introduction to Convolutional Neural Networks”,2015.
- [12] J.R. Terven, D.M. Cordova-Esperanza, “A Comprehensive Review Of YOLO: From YOLOv1 to YOLOv8 and Beyond, 2023.
- [13] Ultralytics YOLOv8 dokumentacija, <https://docs.ultralytics.com/> , 21.08.2023.
- [14] S. Gosh et al.,”An End-to-End Marking Recognition System for PCB Optical Inspection”, 2022.

- [15] J. Solawetz, Francesco, “What is YOLOv8? The Ultimate guide”, <https://blog.roboflow.com/whats-new-in-yolov8/>, Pristup: 21.8.2023.
- [16] Dokumentacija LUCID PHX064S, <https://thinklucid.com/product/phoenix-6-mp-imx178/>, Pristup: 2023.
- [17] Dokumentacija Edmund Optics objektiva, <https://thinklucid.com/product/edmund-optics-c-mount-2-3-16mm-f-1-6-c-series/>, Pristup: 16.12.2023.
- [18] Arduino UNO R3: Product Reference Manual, SKU:A000066
- [19] Raspberry Pi 3 B+: Product Brief
- [20] Jetson Xavier NX Development Kit – Get Started, <https://developer.nvidia.com/embedded/learn/get-started-jetson-xavier-nx-devkit>, Pristup: 16.12.2023.
- [21] Jetson Nano Development Kit, <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano-developer-kit/>, Pristup: 16.12.2023.
- [22] STM32H723ZG dokumentacija, <https://www.st.com/en/microcontrollers-microprocessors/stm32h723zg.html>, Pristup: 16.12.2023.
- [23] Up Board dokumentacija, <https://up-board.org/up/specifications/>, Pristup: 18.12.2023.
- [24] Segment Anything Model (SAM), <https://segment-anything.com/>, Pristup: 15.06.2023.
- [25] D.P Kingma i J. Lei Ba, “Adam: A method for stochastic optimization”, ICLR 2015, San Diego, 2015.
- [26] Image thresholding, OpenCV dokumentacija, [https://docs.opencv.org/3.4/d7/d4d/tutorial\\_py\\_thresholding.html](https://docs.opencv.org/3.4/d7/d4d/tutorial_py_thresholding.html), Pristup: 05.01.2024.
- [27] Tesseract OCR Github repozitorij, <https://github.com/tesseract-ocr/tesseract>, Pristup: 05.01.2024.

## PRILOZI

### I. Rezultati treninga YOLOv8 modela

Class	Images	Instances	Box(P	R	mAP50	mAP50-95)
all	135	1459	0.994	1	0.995	0.949
40 pins	135	48	0.999	1	0.995	0.971
6 pins	135	18	0.996	1	0.995	0.913
A-V	135	12	1	1	0.995	0.962
ARDUINO	135	24	0.997	1	0.995	0.913
Arduino CPU	135	9	0.984	1	0.995	0.974
Arduino UNO back	135	15	1	1	0.995	0.995
Arduino UNO model	135	15	0.993	1	0.995	0.948
Arduino board	135	9	1	1	0.995	0.995
Battery	135	11	0.988	1	0.995	0.986
Berry	135	15	0.992	1	0.995	0.942
Button RESET	135	10	0.988	1	0.995	0.915
Button USER	135	10	1	1	0.995	0.889
CAUTION	135	10	0.998	1	0.995	0.829
CAUTION 1	135	10	1	1	0.995	0.9
CAUTION 2	135	10	0.988	1	0.995	0.904
CN10	135	13	0.995	1	0.995	0.982
CN11	135	13	0.99	1	0.995	0.989
CN12	135	13	0.994	1	0.995	0.979
CN34	135	11	0.999	1	0.995	0.822
CN7	135	13	1	1	0.995	0.971
CN8	135	13	0.993	1	0.995	0.949
CN9	135	13	0.992	1	0.995	0.986
Capacitor	135	18	0.995	1	0.995	0.904
Check pattern 1	135	60	0.998	1	0.995	0.959
Check pattern 2	135	60	0.998	1	0.995	0.963
Check pattern 3	135	45	0.978	1	0.995	0.976
Check pattern 4	135	30	0.989	0.967	0.994	0.985
Connector HDMI	135	48	0.997	1	0.995	0.943
Connector camera	135	22	0.998	1	0.995	0.943
Connector display	135	12	0.993	1	0.995	0.977
Connector micro	135	11	1	1	0.995	0.918
Connector power	135	57	0.999	1	0.995	0.855
Cooling system	135	21	0.998	1	0.995	0.986
D1-D3	135	13	0.995	1	0.995	0.864
Ethernet	135	47	0.997	1	0.995	0.976
Ethernet covered	135	11	0.988	1	0.995	0.942
Graphics_card_Jetson_Xavier	135	15	0.991	1	0.995	0.99
J38	135	10	1	1	0.995	0.734
J40	135	10	1	1	0.995	0.882
Jetson Nano	135	10	0.984	1	0.995	0.995
Jetson Nano Dev board	135	10	1	1	0.995	0.995
Jetson Xavier back	135	15	1	1	0.995	0.995
Jetson Xavier board	135	15	0.99	1	0.995	0.987
Jetson camera connector	135	30	0.996	1	0.995	0.923
L3-L6	135	15	0.993	1	0.995	0.93
Micro SD card port	135	15	0.991	1	0.995	0.986
NVIDIA.	135	14	0.994	1	0.995	0.879
Network transformer	135	10	0.989	1	0.995	0.935
Nucleo	135	10	1	1	0.995	0.968
Oscillator	135	9	1	1	0.995	0.941
PoE	135	12	0.992	1	0.995	0.917
Power	135	45	0.998	1	0.995	0.937
Q1-Q3	135	13	0.993	1	0.995	0.896
Q4-Q7	135	15	0.996	1	0.995	0.898
RAM	135	15	0.99	1	0.995	0.99
RaspberryPi CPU	135	12	0.988	1	0.995	0.995

Class	Images	Instances	Box (P	R	mAP50	mAP50-95)
RaspberryPi back	135	15	1	1	0.995	0.995
RaspberryPi board	135	12	1	1	0.995	0.995
RaspberryPi model	135	12	1	1	0.995	0.995
Reset	135	9	0.988	1	0.995	0.959
Resistor check pattern	135	15	0.998	1	0.995	0.941
SSD	135	15	0.991	1	0.995	0.995
ST	135	23	0.996	1	0.995	0.906
ST-LINK V3	135	10	0.987	1	0.995	0.944
STM32 CPU	135	10	0.986	1	0.995	0.995
STM32 back	135	13	0.99	1	0.995	0.995
STM32 board	135	10	1	1	0.995	0.995
STM32 model	135	10	0.986	1	0.995	0.983
UNO white	135	24	0.995	1	0.995	0.971
UP board	135	11	0.987	1	0.995	0.995
UP board back	135	10	1	1	0.995	0.995
USB	135	96	0.999	1	0.995	0.945
USB Arduino	135	9	0.986	1	0.995	0.98
USB interface	135	21	0.996	1	0.995	0.946
User LEDs	135	10	0.987	1	0.995	0.932
Ventilator	135	15	1	1	0.995	0.989
Wi-Fi-Bluetooth module	135	27	0.995	1	0.995	0.987
berry	135	12	0.992	1	0.995	0.924