

Zbornik strokovne konference

# ROSUS 2024

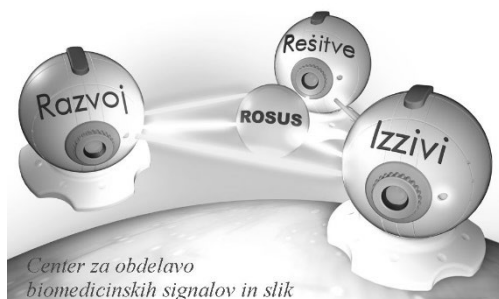
Računalniška obdelava slik in  
njena uporaba v Sloveniji 2024

Urednik:  
Božidar Potočnik



Maribor, 21. marec 2024





# **ROSUS 2024 - Računalniška obdelava slik in njena uporaba v Sloveniji 2024**

Zbornik 18. strokovne konference

Urednik  
**Božidar Potočnik**

Marec 2024

<b>Naslov</b> <i>Title</i>	<b>ROSUS 2024 - Računalniška obdelava slik in njena uporaba v Sloveniji 2024</b> <i>ROSUS 2024 - Computer image processing and its application in Slovenia 2024</i>
<b>Podnaslov</b> <i>Subtitle</i>	<b>Zbornik 18. strokovne konference</b> <i>Proceedings of the 18<sup>th</sup> Professional Conference</i>
<b>Urednik</b> <i>Editor</i>	Božidar Potočnik (Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko)
<b>Recenzija</b> <i>Review</i>	Aleš Holobar (Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko)  Simon Dobrišek (Univerza v Ljubljani, Fakulteta za elektrotehniko)
<b>Tehnični urednik</b> <i>Technical editor</i>	Jan Perša (Univerza v Mariboru, Univerzitetna založba)
<b>Oblikovanje ovitka</b> <i>Cover designer</i>	Gašper Sedej (Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko)
<b>Grafične priloge</b> <i>Graphic material</i>	Viri so lastni, razen če ni navedeno drugače. Avtorji prispevkov in Potočnik (urednik), 2024
<b>Konferenca</b> <i>Conference</i>	Strokovna konferenca ROSUS 2024: Računalniška obdelava slik in njena uporaba v Sloveniji 2024
<b>Datum in kraj konference</b> <i>Conference date &amp; location</i>	21. marec 2024, Maribor, Slovenija
<b>Programski odbor</b> <i>Program committee</i>	Aleš Holobar, Damjan Zazula, Boris Cigale, Marjan Mernik, Peter Peer, Tomaž Tomažič, Martin Šavc, Slobodan Ribarić, Vojko Flis, Saša Divjak, Stanislav Kovačič, Zdravko Kačič, Aleš Leonardiš, Simon Dobrišek, Vitomir Štruc, Franc Solina, Andrej Šoštarič, Franjo Pernuš, Jure Skvarč, Božidar Potočnik
<b>Organizacijski odbor</b> <i>Organization committee</i>	Gašper Sedej, Jurij Munda, Uroš Mlakar, Matjaž Divjak, Martin Šavc, Danilo Korže, Nina Murks, Matej Kramberger, Leon Kutoš, Aleš Holobar, Božidar Potočnik
<b>Založnik</b> <i>Published by</i>	<b>Univerza v Mariboru</b> <b>Univerzitetna založba</b> Slomškov trg 15, 2000 Maribor, Slovenija <a href="https://press.um.si">https://press.um.si</a> , <a href="mailto:zalozba@um.si">zalozba@um.si</a>
<b>Izdajatelj</b> <i>Issued by</i>	<b>Univerza v Mariboru</b> <b>Fakulteta za elektrotehniko, računalništvo in informatiko</b> Koroška cesta 46, 2000 Maribor, Slovenija <a href="https://feri.um.si">https://feri.um.si</a> , <a href="mailto:feri@um.si">feri@um.si</a>
<b>Izdaja</b> <i>Edition</i>	Prva izdaja



**Vrsta publikacije**  
*Publication type* E-knjiga

**Dostopno na**  
*Available at* <https://press.um.si/index.php/ump/catalog/book/859>

**Izdano**  
*Published* Maribor, marec 2024



© Univerza v Mariboru, Univerzitetna založba **Besedilo/ Text**  
*University of Maribor, University Press*

© Avtorji in Potočnik, 2024

To delo je objavljeno pod licenco Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 4.0 Mednarodna. / *This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License.*

Uporabnikom se dovoli reproduciranje, distribuiranje, dajanje v najem, javno priobčitev in predelavo avtorskega dela, če navedejo avtorja in širijo avtorsko delo/predelavo naprej pod istimi pogoji. Za nova dela, ki bodo nastala s predelavo, je tudi dovoljena komercialna uporaba.

Vsa gradiva tretjih oseb v tej knjigi so objavljena pod licenco Creative Commons, razen če to ni navedeno drugače. Če želite ponovno uporabiti gradivo tretjih oseb, ki ni zajeto v licenci Creative Commons, boste morali pridobiti dovoljenje neposredno od imetnika avtorskih pravic.

<https://creativecommons.org/licenses/by-sa/4.0/>

CIP - Kataložni zapis o publikaciji  
Univerzitetna knjižnica Maribor

004.932(497.4)(082)(0.034.2)

STROKOVNA konferenca ROSUS (konferenca) (18 ; 2024 ; Maribor)

ROSUS 2024 [Elektronski vir] : računalniška obdelava slik in njena uporaba v Sloveniji 2024 : zbornik 18. strokovne konference : [21. marec 2024, Maribor, Slovenija] / urednik Božidar Potočnik. - 1. izd. - E-zbornik. - Maribor : Univerza v Mariboru, Univerzitetna založba, 2024

Način dostopa (URL): <https://press.um.si/index.php/ump/catalog/book/859>

ISBN 978-961-286-837-6 (PDF)

doi: 10.18690/um.feri.1.2024

COBISS.SI-ID 188298243

**ISBN** 978-961-286-837-6 (pdf)  
978-961-286-838-3 (mehka vezava)

**DOI** <https://doi.org/10.18690/um.feri.1.2024>

**Cena**  
*Price* Brezplačni izvod

**Odgovorna oseba založnika**  
*For publisher* prof. dr. Zdravko Kačič,  
rektor Univerze v Mariboru

**Citiranje**  
*Attribution* Potočnik, B. (ur.). (2024). *ROSUS 2024 - Računalniška obdelava slik in njena uporaba v Sloveniji 2024: zbornik 18. strokovne konference*. Univerza v Mariboru, Univerzitetna založba. doi: 10.18690/um.feri.1.2024

## POKROVITELJI



**Slovenska sekcija**



## ROSUS 2024

<http://rosus.feri.um.si>

# Kazalo

	<b>Predgovor</b> <i>Foreword</i> Božidar Potočnik	1
<b>VABLJENA PREDAVANJA</b>		<b>3</b>
	<b>Inovativna vloga tehnologij prepoznavanje obraza, video analitike in senzorike v današnji praksi</b> <i>The Innovative Role of Face Recognition, Video Analytics and Sensor Technologies in Today's Practice</i> Janez Zabukovec	5
1	<b>Biometrija in varstvo osebnih podatkov: kako na tehnologijo gleda zakonodaja?</b> <i>Biometrics and Personal Data Protection: Legal Perspective</i> Ana Antuničević	9
<b>DOPOLDANSKA IN POPOLDANSKA SEKCIJA</b> Industrijske aplikacije, medicinske in biomedicinske aplikacije, študentske aplikacije, drugo		<b>23</b>
2	<b>Demonstracijska celica za prikaz globokega učenja v praktičnih aplikacijah</b> <i>Demonstration Cell for Showcasing Deep Learning in Practical Applications</i> Domen Tabernik, Peter Mlakar, Jakob Božič, Luka Čehovin Zajc, Vid Rijavec, Danijel Skočaj	25
3	<b>Gradnja konvolucijskih nevronske mreže za razvrščanje slik z uporabo evolucijskih algoritmov</b> <i>Building Convolutional Neural Networks for Image Classification Using Evolutionary Algorithms</i> Uroš Mlakar	37
4	<b>Zaznavanje podvodnih objektov z uporabo generativnih modelov</b> <i>Detection of Underwater Objects Using Generative Models</i> Sandra Rodríguez Domínguez, Janez Persš	49

5	<b>Introductory Computer Vision Teaching Materials for VET Education</b> <i>Uvodni učni materiali računalniškega vida za strokovno srednješolsko izobraževanje</i> Vasja Lev Kirn, Žiga Emeršič, Gregor Hrastnik, Nataša Meh Peer, Peter Peer	61
6	<b>Aplikacija računalniškega vida za reševanje Rubikove kocke v realnem času</b> <i>Computer Vision Application for Solving a Rubik's Cube in Real-Time</i> Jan Šuklje, Peter Peer, Bojan Klemenc	71
7	<b>Sprotna analiza slik vozil z metodami globokega učenja v ogrodju Flutter</b> <i>Computer Vision Application for Solving a Rubik's Cube in Real-Timeinstantaneous Vehicle Image Analysis With Deep Learning Methods With Flutter Framework</i> Aleksandr Shishkov, Stevanče Nikoloski	81
8	<b>Ustvarjanje ponarejenih videoposnetkov s pomočjo difuzijskih modelov</b> <i>Creating Fake Videos Using Diffusion Models</i> Bine Markelj, Peter Peer, Borut Batagelj	97
9	<b>Spektralni odziv dveh hiperspektralnih kamer za izbor končnih članov UXO</b> <i>Spectral Response of Two Hyperspectral Cameras for Uxo Endmember Selection</i> Milan Bajić, Božidar Potočnik	111

# Spoštovani!

BOŽIDAR POTOČNIK

*predsednik konference ROSUS 2024*

Po sedemnajstih konferencah ROSUS 2006–2023 s konferenco ROSUS 2024 nadaljujemo promoviranje ekonomske koristi računalniške obdelave slik in računalniškega vida na področjih industrije, biomedicine in drugih poslovnih procesov. Vezi, ki smo jih na prejšnjih konferencah stkali med raziskovalci, razvijalci, ponudniki rešitev ter uporabniki, želimo dodatno okrepiti, ob tem pa nuditi možnosti sklepanja novih sodelovanj in svetovanja pri razreševanju konkretnih poslovnih oziroma raziskovalnih problemov.

Glavni poudarek namenjamo aplikacijam s področja računalniške obdelave slik, ki so že integrirane oziroma pripravljene za integracijo v poslovne procese. Demonstrirali bomo, da avtomatska obdelava v industriji lahko zaradi svoje natančnosti in hitrosti prinaša velike ekonomske koristi, hkrati pa nakazali, da takšne aplikacije nudijo nove priložnosti za uveljavitev na trgu visokih tehnologij. Ne smemo pozabiti še na možnost diskusije ter predstavitev konkretnih problemov in potreb, ki se porajajo pri uporabnikih, s katerimi bomo računalniško obdelavo slik in njeno koristnost še bolj približali avditoriju.

Naj sklenemo uvodne misli s prisrčno zahvalo Javnemu skladu Republike Slovenije za podjetništvo, ki v okviru konference ROSUS 2024 predstavlja zanimive finančne instrumente za spodbujanje prenosa tehnoloških rešitev v podjetniško sfero. Iskreno se zahvaljujemo tudi podjetju Audax za sprejem pokroviteljstva konference. Izpostaviti želimo še medijske

pokrovitelje revijo IRT3000, revijo Svet elektronike in revijo Avtomatika + elektronika, ki so intenzivno promovirali konferenco ROSUS 2024 ter pomen strojnega vida v slovenskem prostoru.



## VABLJENA PREDAVANJA

---





# INOVATIVNA VLOGA TEHNOLOGIJ PREPOZNAVANJE OBRAZA, VIDEO ANALITIKE IN SENZORIKE V DANAŠNJI PRAKSI

JANEZ ZABUKOVEC

STELKOM d.o.o., Trzin, Slovenija  
janez.zabukovec@stelkom.si

Na svetovnem trgu je veliko število ponudnikov biometrije oz. programske opreme za prepoznavanje obraza, vendar je med njimi veliko razlik. Podjetje Stelkom ima dolgoletnega partnerja NEC Japonska, ki je eden izmed vodilnih svetovnih podjetij na tem področju. Prvi program za prepoznavo obrazov so na tržišču testirali davnega leta 1989. V kasnejših letih od 2009 do danes so bili vsako leto nagrajeni in priznani s strani NIST kot številka 1 na svetu. V nadaljevanju Vam podrobneje predstavimo njihove najnovejše dosežke. Stelkom je tudi integrator številnih partnerjev video analitike. V nadaljevanju predstavljamo uporabnost te tehnologije v današnjem življenju ter nekoliko povežemo z ostalimi senzorikami ter VMS centralnim sistemom upravljanja.

ISBN  
978-961-286-837-6

**Ključne besede:**  
biometrija,  
razpoznavanje obrazov,  
industrija,  
video analitika,  
senzorski sistemi



Univerzitetna založba  
Univerze v Mariboru

ISBN  
978-961-286-837-6

# THE INNOVATIVE ROLE OF FACE RECOGNITION, VIDEO ANALYTICS AND SENSOR TECHNOLOGIES IN TODAY'S PRACTICE

JANEZ ZABUKOVEC

STELKOM d.o.o., Trzin, Slovenia  
janez.zabukovec@stelkom.si

**Keywords:**

biometrics,  
face recognition,  
industry,  
video analytics, sensor  
systems

On the global market, there are many providers of biometrics or facial recognition software, but there are several differences between them. Stelkom company has a long-standing partner, NEC Japan, which is one of the world's leading companies in this field. The first facial recognition solution was tested on the market way back in 1989. In subsequent years, from 2009 to the present, they were awarded every year and recognized by NIST as number One in the world. We present their latest achievements in more detail in this sequel. Stelkom is also an integrator for many video analytics partners. We also present the usefulness of this technology in today's life and somewhat connect it with other sensors and the VMS central management system.



University of Maribor Press

## 1 Uvod

Pametni sistemi za video analitiko lahko prepoznajo spol, starost in določene telesne značilnosti uporabnikov infrastrukture, obenem pa prek kompleksnih algoritmov razumejo vedenjske vzorce in kretnje ter se nanje avtomatsko odzovejo.



**Slika 1: Stelkom**

Vir: splet, odprt dostop.

Za upravitelje in odločevalce lahko takšni sistemi ustvarjajo dodano vrednost tudi pri zagotavljanju optimalne uporabniške izkušnje na podlagi objektivnih podatkov: videonadzorni sistem pretvorijo v orodje za spremljanje poslovnih informacij (Business Intelligence).



# BIOMETRIJA IN VARSTVO OSEBNIH PODATKOV: KAKO NA TEHNOLOGIJO GLEDA ZAKONODAJA?

ANA ANTUNIČEVIĆ

Consilium ius d.o.o., Ljubljana, Slovenija  
ana@consilium.si

V tem prispevku se avtorica osredotoča na vzpostavitev nove zakonodaje s področja varstva osebnih podatkov tako na ravni Evropske Unije (GDPR) kot tudi na ravni Slovenije (ZVOP-2) in njen vpliv na obdelavo osebnih podatkov v okviru biometrične tehnologije.

DOI  
[https://doi.org/  
10.18690/um.feri.1.2024.1](https://doi.org/10.18690/um.feri.1.2024.1)

ISBN  
978-961-286-837-6

**Ključne besede:**  
varstvo osebnih podatkov,  
zakonodaja,  
GDPR,  
ZVOP-2,  
biometrija



Univerzitetna založba  
Univerze v Mariboru

DOI  
[https://doi.org/  
10.18690/um.feri.1.2024.1](https://doi.org/10.18690/um.feri.1.2024.1)

ISBN  
978-961-286-837-6

**Keywords:**  
personal data protection,  
legislation,  
GDPR,  
ZVOP-2,  
biometrics

# BIOMETRICS AND PERSONAL DATA PROTECTION: LEGAL PERSPECTIVE

ANA ANTUNIČEVIĆ

Consilium ius d.o.o., Ljubljana, Slovenia  
[ana@consilium.si](mailto:ana@consilium.si)

In this paper, the author focuses on the establishment of new legislation in the field of personal data protection both at the European Union level (GDPR) and at the Slovenian level (ZVOP-2) and its impact on the processing of personal data in the context of biometric technology.





## 1 Uvod

V današnjem času na praktično vseh področjih beležimo hiter razvoj digitalizacije. Ti tehnološki preboji odpirajo vrata novim načinom komunikacije, poslovanja, obdelave in shranjevanja množice podatkov, ki je bila mnogoterim še pred kratkim nepredstavljiva. Vendar pa ta digitalna revolucija prinaša tudi nove izzive; eden takšnih je tudi varovanje zasebnosti in varstvo osebnih podatkov.

Kot odgovor na večje spremembe v družbi, se je pojavila zakonodaja, ki do določene mere ureja uporabo tehnologije ter določa varnostne mehanizme, s katerimi se dviguje nivo varstva zasebnosti. Razvoj zakonodaje v takšnih trenutkih je ključen, tako z vidika zagotavljanja varnosti kot tudi razvoja tehnologije, ki jo zakonodaja ureja. Pri pripravi takšne zakonodaje pa je pomembno, da je fleksibilna in dovolj odprta, saj se mora prilagajati spreminjajočim se potrebam in tehnološkim inovacijam. Hkrati pa se je potrebno zavedati dejstva, da zakonodaja vedno nastaja kot odgovor na nastale razmere, kar neizogibno vodi v dejstvo, da po sprejemu in vpeljavi takšne zakonodaje, velikokrat dobimo občutek, da je »zastarela« še preden smo jo sploh uporabili. Okoliščine še dodatno otežuje zares bliskovit razvoj tehnologije, ki smo mu priča v zadnjem času.

V takšnih okoliščinah je nastala Splošna uredba o varstvu osebnih podatkov<sup>1</sup> (v nadaljevanju tudi samo GDPR), s katero je Evropska Unija poskusila uokviriti obdelavo osebnih podatkov. Čeprav morda GDPR na prvi vtis deluje predvsem na področju spleta in z njim povezanih velikanov kot sta Google in Meta, pa lahko ob bolj podrobnem pregledu, hitro ugotovimo, da ureditev obdelave osebnih podatkov, ki jo določa GDPR, posega tudi v poslovne procese in razvoj ter uporabo določenih tehnologij, kamor sodi tudi biometrija, o kateri bomo govorili v tem prispevku.

---

<sup>1</sup> UREDBA (EU) 2016/679 EVROPSKEGA PARLAMENTA IN SVETA z dne 27. aprila 2016 o varstvu posameznikov pri obdelavi osebnih podatkov in o prostem pretoku takih podatkov ter o razveljavitvi Direktive 95/46/ES (Splošna uredba o varstvu podatkov). Dostopno na: <https://eur-lex.europa.eu/legal-content/SL/TXT/PDF/?uri=CELEX:32016R0679>.

## 2 Pogled na obdelavo osebnih podatkov skozi določila GDPR in slovenskega Zakona o varstvu osebnih podatkov

### 2.1 Sprejem GDPR in Zakona o varstvu osebnih podatkov

Evropska unija je leta 2016 sprejela Splošno uredbo o varstvu osebnih podatkov<sup>2</sup> (GDPR), ki je nadomestila do sedaj veljavno Direktivo 95/94/ES. Po sprejemu GDPR je nastopil 2 letni *vacatio legis*<sup>3</sup>, ki je omogočal državam članicam kot tudi zavezancem in upravičencem po GDPR, čas, da se seznanijo z novo zakonodajo in jo ustrezno implementirajo. Po poteku tega obdobja, je GDPR stopila v veljavo, kar se je zgodilo 25.5.2018.

Na tej točki je pomembno poudariti, da je GDPR Uredba EU, kar pomeni, da gre za zavezujoč pravni akt, ki velja **neposredno v vseh državah članicah EU**, brez da bi bil potreben kakršenkoli prenos s sprejemom nacionalne zakonodaje. Ravno neposredna uporabljivost pa je ključna razlika med Uredbo EU (kot je GDPR) in Direktivo EU (kot je bila Direktiva 95/46/ES, ki jo nadomešča GDPR). Direktiva EU namreč določa cilje oz. začrta smernice, ki se jih morajo države članice držati, vendar pa jo mora vsaka država članica **prenesti v svoj nacionalni pravni red**. Upoštevajoč zgoraj napisano direktiva prinese manj poenoteno zakonodajo na ravni EU, saj imajo države članice pri prenosu direktive več maneverskega prostora, medtem ko pri Uredi EU **vsebinsko odstopanje ni mogoče** – države članice se morajo v celoti ravnati skladno z besedilom uredbe; ločeno pa lahko obravnavajo določene tematike, ki niso del uredbe ali pa določene tematike obravnavajo bolj ali manj rigorozno kot specifična uredba (vendar znotraj meja, ki jih začrta uredba).

Sprejem GDPR je bil torej velik preskok za Evropsko Unijo, saj je določila s področja varstva osebnih podatkov do sedaj urejala direktiva, medtem ko ima GDPR status Uredbe in s tem močno dviguje **unifikacijo prava držav članic EU na področju varstva osebnih podatkov in zasebnosti**.

---

<sup>2</sup> Kot zgoraj.

<sup>3</sup> Gre za obdobje, ki sledi sprejetju zakona (in objavi v uradnem glasilu), vendar pred uveljavitvijo zakona. To obdobje zavezancem, ki jih zakon zavezuje omogoča, da se prilagodijo na izvajanje nove zakonodaje (npr. sprejmejo nove ukrepe, prilagodijo poslovne procese ipd.).

Po sprejemu, vendar zlasti po **uveljavitvi GDPR** je sledila evforija urejanja skladnosti z GDPR v slovenskem gospodarskem prostoru. Vendar se je začetna evforija, ki so jo spremljale predvsem ogromne količine e-sporočil tipa »ostanite z nami«, kmalu polegla. GDPR pa je bila v Sloveniji razglašena za brezzobega tigra.

Kaj točno se je zgodilo?

Da bi razumeli, zakaj je GDPR v Sloveniji »obvisela v zraku« in zakaj ni bilo mogoče izreči nobene globe, si je potrebno pogledati razliko med materialnim in procesnim zakonom.

Medtem ko materialni zakon določa pravice in obveznosti v zvezi z določeno vsebino (oz. materijo), kot je npr. varstvo osebnih podatkov, pa procesni zakon ureja postopke in načine izvrševanja materialnih zakonov. Če povedano apliciramo na naš primer: GDPR je sama po sebi materialni predpis (zakon oz. uredba), zato za svojo polno uveljavitev potrebuje procesni zakon, ki ga Slovenija vse do leta 2023 ni sprejela.

S sprejemom Zakona o varstvu osebnih podatkov<sup>4</sup> (v nadaljevanju tudi samo ZVOP-2) je Slovenija dobila procesni del kot tudi nekatere materialne dopolnitve, ki podrobneje opredeljujejo določila GDPR. Eno izmed takšnih področij je tudi področje biometrije, o katerem bomo pisali v tem prispevku.

## 2.2 Osnove varstva osebnih podatkov po GDPR in ZVOP-2

Preden se spustimo v podrobno obravnavo biometrije in biometričnih osebnih podatkov, se je potrebno seznaniti z osnovami varstva osebnih podatkov po veljavni zakonodaji.

Ključen pojem, ki nikakor ni tako enoznačen kot zveni, je namreč **osebni podatek**. GDPR določa, da osebni podatek: *»pomeni katero koli informacijo v zvezi z določenim ali določljivim posameznikom (v nadaljnjem besedilu: posameznik, na katerega se nanašajo osebni podatki); določljiv posameznik je tisti, ki ga je mogoče neposredno ali posredno določiti, zlasti z navedbo identifikatorja, kot je ime, identifikacijska številka, podatki o lokaciji, spletni*

---

<sup>4</sup> Zakon o varstvu osebnih podatkov, UL RS; št. 003-02-22024-4. Dostopno na: <https://www.uradni-list.si/glasilo-uradni-list-rs/vsebina/2022-01-4187?sop=2022-01-4187>.

*identifikator, ali z navedbo enega ali več dejavnikov, ki so značilni za fizično, fiziološko, genetsko, duševno, gospodarsko, kulturno ali družbeno identiteto tega posameznika»<sup>5</sup>.*

Hitro vidimo, da je definicija **splošna in predvsem zelo široka**. Če jo na kratko povzamemo, lahko zaključimo, da je osebni podatek katerakoli informacija (ali sklop informacij), na podlagi katere je posameznik **določljiv**. Ne gre torej za standard **določenosti**; tj. da je posameznik v celoti identificiran (tipičen primer določenosti je npr. kombinacija imena, priimeka in naslova, ki velikokrat privedejo do določenosti posameznika), temveč **določljivosti**, kar pomeni, da na skupek različnih informacij **določi posameznika**.

Informacijski pooblaščenec je v svoji infografiki pripravil določene tipske kategorije osebnih podatkov, in sicer osebne podatke uvršča v naslednje kategorije:

- **Identifikacijski podatki** (ime in priimek, EMŠO, davčna številka, IP naslov ipd.),
- **Kontaktne podatki** (telefonska številka, e-mail, naslov ipd.),
- **Transakcijski podatki** (podatki o nakupih, zdravstvenih storitvah, podatki o uporabi spleta, lokacijski podatki ipd.) in
- **Statusni podatki** (podatki o zdravju, premoženju, željah, preferencah, izobrazbi, spolu, poklicu ipd.).<sup>6</sup>

Prav tako je zelo širok pojem **obdelave osebnih podatkov**, ki po GDPR vključuje katerakoli operacijo na osebnih podatkih, kot so: hramba, posredovanje, spreminjanje, uničenje in izbris.<sup>7</sup>

---

<sup>5</sup> Člen 4, I. odstavek, UREDBA (EU) 2016/679 EVROPSKEGA PARLAMENTA IN SVETA z dne 27. aprila 2016 o varstvu posameznikov pri obdelavi osebnih podatkov in o prostem pretoku takih podatkov ter o razveljavitvi Direktive 95/46/ES (Splošna uredba o varstvu podatkov. Dostopno na: <https://eur-lex.europa.eu/legal-content/SL/TXT/PDF/?uri=CELEX:32016R0679>

<sup>6</sup> Povzeto po Infografiki Informacijskega pooblaščenca, ki je dostopna na: [https://www.ip-rs.si/fileadmin/user\\_upload/Pdf/infografike/Infografika%20-%20osebni%20podatki2.pdf](https://www.ip-rs.si/fileadmin/user_upload/Pdf/infografike/Infografika%20-%20osebni%20podatki2.pdf)

<sup>7</sup> Člen 4, II. odstavek, UREDBA (EU) 2016/679 EVROPSKEGA PARLAMENTA IN SVETA z dne 27. aprila 2016 o varstvu posameznikov pri obdelavi osebnih podatkov in o prostem pretoku takih podatkov ter o razveljavitvi Direktive 95/46/ES (Splošna uredba o varstvu podatkov. Dostopno na: <https://eur-lex.europa.eu/legal-content/SL/TXT/PDF/?uri=CELEX:32016R0679>

### 2.3 Zakonitost obdelave osebnih podatkov

Na podlagi GDPR je obdelava osebnih podatkov zakonita, kadar poteka na **ustrezni pravni podlagi** in za vnaprej opredeljen, jasno določen, zakonit in javno objavljen **namen**.

Pravne podlage, na katerih je možna obdelava osebnih podatkov, so določene v 6. členu GDPR in so taksativno naštet, kar pomeni, da smo dolžni pravno podlago izbrati v okviru tega člena.

GDPR določa naslednje pravne podlage:

- Zakon,
- Soglasje,
- Pogodbena razmerje,
- Zaščita življenjskih interesov,
- Zaščita javnega interesa (izvajanje javne oblasti) in
- Zakoniti interesi.<sup>8</sup>

Preden se v praksi izvede obdelavo osebnih podatkov, je torej potrebno opredeliti: (1) zakaj bomo to obdelavo izvajali (**namen obdelave**) in (2) na kateri pravni podlagi se bo takšna obdelava izvajala (**določitev pravne podlage**).

Pri sami izvedbi je potrebno paziti še na obveznost informiranja posameznikov (13. in 14. člen GDPR) in spoštovanje načel GDPR (5. člen GDPR), kjer je potrebno izpostaviti zlasti načelo transparentnosti in načelo minimizacije podatkov; slednje je ključno pri določanju obsega osebnih podatkov, ki jih potrebujemo, da dosežemo namen, ki smo si ga zadali.

### 3 Ureditev biometrije v Sloveniji

V tem prispevku se bomo osredotočili na ureditev obdelave biometričnih osebnih podatkov, skladno z ZVOP-2, ki velja v Sloveniji.

---

<sup>8</sup> Člen 6, UREDBA (EU) 2016/679 EVROPSKEGA PARLAMENTA IN SVETA z dne 27. aprila 2016 o varstvu posameznikov pri obdelavi osebnih podatkov in o prostem pretoku takih podatkov ter o razveljavitvi Direktive 95/46/ES (Splošna uredba o varstvu podatkov. Dostopno na: <https://eur-lex.europa.eu/legal-content/SL/TXT/PDF/?uri=CELEX:32016R0679>.

Področje biometrije, v okviru GDPR velja za posebnost, saj GDPR v 9. členu določa, da smejo države članice uvesti dodatne pogoje in omejitve, kadar gre za obdelavo biometričnih podatkov<sup>9</sup>. Kot odgovor na to določilo, se je slovenski zakonodajalec odločil za bolj podrobno opredelitev obdelave biometričnih podatkov znotraj ZVOP-2, ki si jo bomo pogledali v tem delu prispevka.

Ločeni ureditvi biometrije je posvečeno 4. poglavje ZVOP-2 z naslovom »Obdelava osebnih podatkov z uporabo biometrije in genskih podatkov«<sup>10</sup>. Preden pa se posvetimo vsebini poglavja, je potrebno določiti kaj biometrija sploh je in zakaj jo obravnavamo v okviru varstva osebnih podatkov.

### 3.1 O biometriji

»Biometrija je eden izmed načinov ugotavljanja oz. preverjanja identitete«, povzame definicijo biometrije Informacijski pooblaščenec in nadaljuje, da gre pri biometriji za preverjanje identitete na podlagi neke vedenjske oz. telesne značilnosti, ki je lastna samo tej osebi<sup>11</sup>. Pri biometričnih sistemih ločimo verifikacijo in identifikacijo.

Pri verifikaciji gre za sistem preverja ali je oseba zares tista, za katero se izdaja. Gre torej za preverbo biometričnega vzorca z vzorcem, ki se nahaja v bazi.<sup>12</sup> Pri identifikaciji pa gre za prepoznavo posameznika, na podlagi podatkovne baze, ki je v sistemu.<sup>13</sup>

Ne glede na to ali gre za verifikacijo ali identifikacijo pa za obdelavo osebnih podatkov potrebujemo biometrične značilnosti, kot so: obraz, prstni odtis, DNA, očesna mrežnica, hoja, glas in drugi<sup>14</sup>. Te biometrične značilnosti predstavljajo tudi biometrične osebne podatke, kadar govorimo o obdelavi osebnih podatkov.

---

<sup>9</sup> Člen 9, IV. Odstavek, UREDBA (EU) 2016/679 EVROPSKEGA PARLAMENTA IN SVETA z dne 27. aprila 2016 o varstvu posameznikov pri obdelavi osebnih podatkov in o prostem pretoku takih podatkov ter o razveljavitvi Direktive 95/46/ES (Splošna uredba o varstvu podatkov). Dostopno na: <https://eur-lex.europa.eu/legal-content/SL/TXT/PDF/?uri=CELEX:32016R0679>.

<sup>10</sup> 4. poglavje, <sup>10</sup> Zakon o varstvu osebnih podatkov, UL RS; št. 003-02-22024-4. Dostopno na: <https://www.uradni-list.si/glasilo-uradni-list-rs/vsebina/2022-01-4187?sop=2022-01-4187>.

<sup>11</sup> Povzeto po: <https://www.ip-rs.si/varstvo-osebnih-podatkov/obveznosti-upravljavcev/prijava-biometrijskih-ukrepov/>.

<sup>12</sup> MATJAŠEC, Matjaž, 2018, Metode biometričnega overjanja: sistematičen pregled literature [na spletu]. Magistrsko delo. Maribor: Univerza v Mariboru. [Dostopano 19 februar 2024]. Pridobljeno s: <https://dk.um.si/IzpisGradiva.php?lang=slv&id=70945>, stran 6.

<sup>13</sup> Kot zgoraj.

<sup>14</sup> MATJAŠEC, Matjaž, 2018, Metode biometričnega overjanja: sistematičen pregled literature [na spletu]. Magistrsko delo. Maribor: Univerza v Mariboru. [Dostopano 19 februar 2024]. Pridobljeno s: <https://dk.um.si/IzpisGradiva.php?lang=slv&id=70945>, stran 8.

Kljub temu, da biometrične značilnosti temeljijo na unikatnih značilnostih, ki so vezane na posameznika, pa je potrebno opozoriti, da biometrija ni 100% natančna, saj se pri delovanju zanaša na določeno mero verjetnosti (medtem ko na drugi strani uporaba gesla zagotavlja 100% učinkovitost, saj je geslo lahko le pravilno ali napačno – ne more pa biti npr. pravilno s 96% verjetnostjo).<sup>15</sup>

Preden se torej odločimo za uporabo biometričnih ukrepov, je torej smiselno, da dobro premislimo, ali je tak ukrep res najprimernejši.

### 3.2 Ureditev biometrije v ZVOP-2

Tako kot je bilo predstavljeno zgoraj, se tudi ZVOP-2 opira na dva načina prepoznave biometrijskih značilnosti in sicer na **identifikacijo** in **avtentikacijo**. ZVOP-2 ureja biometrične ukrepe v členih 81 – 84 pri čemer v 81. členu najdemo splošne prepovedi, kot npr. prepoved povezovanja zbirk biometričnih osebnih podatkov z drugimi zbirkami<sup>16</sup>. V nadaljevanju pa ZVOP-2 predpisuje način uporabe biometričnih ukrepov, pri čemer deli uporabo biometrije na zasebni in javni sektor.

#### 3.2.1 Biometrični osebni podatki

Ker ta prispevek temelji na biometričnih osebnih podatkih, si bomo ogledali še ali gre pri biometričnih podatkih vedno tudi za osebne podatke ali obstajajo kakšne izjeme, ko lahko biometrične podatke uporabljamo, ne da bi šlo za obdelavo osebnih podatkov.

Pogosto je v okviru uporabe biometrije omenjeno, da »ne gre za obdelavo osebnih podatkov«, saj se za prepoznavo uporabljajo zgolj »predloge« (tj. zapis vzorca v digitalni obliki) in ne izvirni osebni podatek (npr. prstni odtis).

Tako na evropski kot na slovenski ravni je sprejeto stališče, da digitalni zapis biometričnih značilnosti (t.i. »*template*« ali predloga) predstavlja **osebni podatek** na enak način kot surovi zapis biometrične značilnosti.

---

<sup>15</sup> Smernice španskega nadzornega organa za varstvo osebnih podatkov; 14 misunderstandings with regard to biometric identification and authentication, dostopno na <https://www.aepd.es/documento/nota-equivocos-biometria-en.pdf>.

<sup>16</sup> Člen 81, III. Odstavek, Zakon o varstvu osebnih podatkov, UL RS; št. 003-02-22024-4. Dostopno na: <https://www.uradni-list.si/glasilo-uradni-list-rs/vsebina/2022-01-4187?sop=2022-01-4187>.



Takšno stališče je sprejeto na podlagi (zgoraj opredeljenega) koncepta osebnega podatka. Videli smo, da je neka informacija osebni podatek, kadar je na podlagi te informacije posameznik **določljiv** (in ne nujno določen). Prav tako smo v tem prispevku pisali, da je namen biometrije **prepoznavna posameznika** glede na njegove distinktivne značilnosti. Upoštevajoč navedeno, bo obdelava podatkov z biometričnimi podatki **vedno predstavljala obdelavo osebnih podatkov**, saj je njihov **prvotni namen** prepoznavna (in s tem tudi določljivost) posameznika. Glede na navedeno, je potrebno za uporabo biometrije spoštovati določila GDPR in ZVOP-2.

### 3.2.2 Ureditev v zasebnem sektorju

Uporabo v **zasebnem sektorju** določa 83. člen, ki pravi, da je obdelava biometričnih podatkov dovoljena le, *»kadar je to **nujno potrebno** za opravljanje dejavnosti, za varnost ljudi, varnost premoženja, varovanje tajnih podatkov ali varovanje poslovnih skrivnosti»*<sup>17</sup>.

Dodatno je v zasebnem sektorju uporaba **biometrije** dopustna tudi *»zaradi varstva točnosti identitete svojih strank»*<sup>18</sup>. Tudi v tem primeru zakon določa omejitve, na podlagi katerih je takšna raba mogoča, kadar je to nujno potrebno zaradi opravljanja dejavnost oz. varnosti (kot opredeljeno zgoraj), kadar takšno obdelavo posebej določa pogodba ali pa kadar je bila pridobljena privolitev stranke.

Zgornja dva odstavka opredeljujeta **pravno podlago** in **namene obdelave**, ki jih je potrebno določiti pri uporabi biometrije.

Kot vidimo, je nabor pravnih podlag ozek, saj ZVOP-2 določa, da je obdelava biometričnih osebnih podatkov dovoljena samo na podlagi zakona. Nekoliko širši nabor je zgolj v primeru, da gre za točnost identitete **strank**, kjer lahko upravljavec osebnih podatkov izbira med pravnimi podlagami: zakon, pogodbeno razmerje in privolitev posamične stranke.

---

<sup>17</sup> Člen 83, I. odstavek, Zakon o varstvu osebnih podatkov, UL RS; št. 003-02-22024-4. Dostopno na: <https://www.uradni-list.si/glasilo-uradni-list-rs/vsebina/2022-01-4187?sop=2022-01-4187>.

<sup>18</sup> Člen 83, II. odstavek, Zakon o varstvu osebnih podatkov, UL RS; št. 003-02-22024-4. Dostopno na: <https://www.uradni-list.si/glasilo-uradni-list-rs/vsebina/2022-01-4187?sop=2022-01-4187>.

Prav tako je **ozek in vnaprej opredeljen** tudi nabor namenov obdelave, ki jim lahko služijo biometrični ukrepi; kot je to razvidno iz zgornjih odstavkov, gre za namene: varovanja (ljudi, premoženja) ali pa varstva točnosti identitete stranke.

V zasebnem sektorju je omogočena tudi raba biometričnih ukrepov, skladno s 3. odstavkom, 83. člena ZVOP-2. V tem primeru gre za **izjemo**, ki omogoča uporabo biometričnih osebnih podatkov za druge namene, v kolikor takšna obdelava poteka **pod izključnim nadzorom stranke** ali **pod njeno izključno oblastjo** in omogoča stranki, da izrecno dovoli obdelavo teh podatkov. Prav tako mora biti takšna obdelava **potrjena** s strani nadzornega organa za potrjevanje, o čemer bomo pisali v ločenem poglavju.

Najbolj zanimiva pojma v zgornjem odstavku sta gotovo »izključni nadzor« in »izključna oblast«, ki ju bo morala napolniti pravna praksa. Do sedaj pa se lahko opremo na mnenje Informacijskega pooblaščenca, ki pravi, da bo potrebno oba pojma razlagati »v smislu spodbujanja načela vgrajenega in privzetega varstva osebnih podatkov<sup>19</sup>«, ki ga uvaja GDPR.

Zgornje načelo je potrebno razlagati na način, da je potrebno za doseg ciljev upoštevati metode, ki čim manj posegajo v zasebnost posameznika in spodbujajo visok nivo varstva osebnih podatkov. Informacijski pooblaščenec navaja, da je eden takih ukrepov tudi »*preprečitev hrambe podatkov v centraliziranih zbirkah pri upravljavcu, temveč omogočanje hrambe na medijih ali napravah, ki so pod nadzorom posameznika (npr. mobilni telefon, USB ključki ipd.)*<sup>20</sup>«.

V primeru, da se upravljavec odloči za tak pristop, mu za vpeljavo **ni potrebno** pridobiti pozitivne odločbe Informacijskega pooblaščenca (opis tega postopka se nahaja spodaj), kar nekoliko olajša postopek sprejemanja biometričnih ukrepov v organizaciji.

---

<sup>19</sup> Mnenje Informacijskega pooblaščenca; št. 07121-1/2024/128, z dne 5.2.2024, dostopno na: <https://www.ip-rs.si/mnenja-zvop-2/potrjevanje-obdelav-biometri%C4%8Dnih-op-1707898494>.

<sup>20</sup> Kot zgoraj.

### 3.2.3 Postopek uvedbe biometrije v zasebnem sektorju

Postopek, kako se uvede biometrične ukrepe, je prav tako opredeljen v ZVOP-2, in sicer od 4. do 8. odstavka 83. člena. Upravljavec je dolžan še pred začetkom obdelave biometričnih osebnih podatkov o tem obvestiti zaposlene in z zaposlenimi izvesti predhodno posvetovanje o sorazmernosti obdelave.

Naslednji korak je posredovanje opisa nameravanih obdelav in razlogov za uvedbo Informacijskemu pooblaščenca. Pri Informacijskem pooblaščenca so pripravili obrazec, ki ga lahko uporabijo upravljavci. Po prejemu vseh informacij, bo Informacijski pooblaščenec **v roku 2 mesecev** odločil o dovoljenosti uporabe biometrije. Upravljavec sme vpeljati biometrične ukrepe šele **po prejemu pozitivne odločbe s strani Informacijskega pooblaščenca**.

### 3.2.4 Biometrija v javnem sektorju

Podobno kot v zasebnem sektorju se tudi biometrija v javnem sektorju lahko določi z zakonom, in sicer »če je to nujno potrebno za varnost ljudi, varnost premoženja, varovanje tajnih podatkov, identifikacijo pogrešanih ali umrlih posameznikov ali za varovanje poslovnih skrivnosti, teh namenov pa ni mogoče doseči z milejšimi sredstvi«<sup>21</sup>.

Prav tako je v javnem sektorju omogočena tudi izjema, ki dopušča obdelavo biometričnih osebnih podatkov, »kadar so dejanja obdelave akreditirana in kadar se obdelava izvaja pod izključnim nadzorom ali izključno oblastjo posameznika ter mu omogoča, da izrecno dovoli obdelavo teh podatkov«<sup>22</sup>.

Prav tako ZVOP-2 določa nekaj dodatnih izjem, ki pa so namenjene izpolnjevanju mednarodnih obveznosti ali pa identifikaciji v okviru postopka izdaje sredstev elektronske identifikacije.

---

<sup>21</sup> Člen 82, I. odstavek, Zakon o varstvu osebnih podatkov, UL RS; št. 003-02-22024-4. Dostopno na: <https://www.uradni-list.si/glasilo-uradni-list-rs/vsebina/2022-01-4187?sop=2022-01-4187>.

<sup>22</sup> Člen 82, II. odstavek, Zakon o varstvu osebnih podatkov, UL RS; št. 003-02-22024-4. Dostopno na: <https://www.uradni-list.si/glasilo-uradni-list-rs/vsebina/2022-01-4187?sop=2022-01-4187>.

### 3.3 Akreditacija

Kot je bilo že omenjeno v zgornjih poglavjih, je kot pogoj za uporabo biometrije omenjena tudi akreditacija posamične biometrijske rešitve. Skladno z ZVOP-2 bi morala slovenska akreditacija pričeti z delovanjem 1.1.2024, vendar do danes akreditacija še ni bila vzpostavljena. Vprašanja, kot so: katero telo bo skrbelo za akreditacijo, kakšen bo postopek in nenazadnje tudi kakšna bo cena takšne akreditacije, tako v času priprave tega prispevka ostajajo neodgovorjena.

V vmesnem obdobju pa se je pri uporabi biometričnih rešitev potrebno opreti na 2. odstavek 121. člena, ki vzpostavlja domnevo, da so vsa »dejanja obdelave upravljavcev in obdelovalcev, ki morajo po določbah tega zakona za dejanja obdelave pridobiti certifikat, ta skladna z merili iz mehanizma potrjevanja«<sup>23</sup>.

## 4 Zaključno

Obdelava osebnih podatkov postaja čedalje bolj kompleksna tema, kjer je prisotno vsakodnevno tehtanje med zmožnostjo tehnologije in posegom v zasebnost posameznika. GDPR, kot temeljni predpis s področja varstva osebnih podatkov, postavlja pravni okvir, znotraj katerega morajo upravljavci in obdelovalci vzpostaviti ustrezne mehanizme in varovalke, ki bodo omogočali varno obdelavo osebnih podatkov.

V tem prispevku je bila podrobneje predstavljena tudi ureditev biometrije v slovenski zakonodaji. V tem primeru je Slovenija specifična v primerjavi z drugimi državami članicami, naša ureditev pa močno odstopa tudi od ureditve GDPR, ki specifičnih določb o biometriji nima.

Čeprav je v primerjavi s prejšnjim (in sedaj neveljavnim) Zakonom o varstvu podatkov (ZVOP-1), sedaj uporaba biometrije širša, še posebej za zasebni sektor, pa lahko trdimo, da je manevrski prostor za biometrične rešitve v slovenskem prostoru relativno ozek.

---

<sup>23</sup> Člen 121, II. Odstavek, Zakon o varstvu osebnih podatkov, UL RS; št. 003-02-22024-4. Dostopno na: <https://www.uradni-list.si/glasilo-uradni-list-rs/vsebina/2022-01-4187?sop=2022-01-4187>.

Na tej točki je potrebno izpostaviti izjemo, ki jo omogoča ZVOP-2, kjer se lahko upravljavci, ki želijo uporabiti biometrijo, izognejo potrjevanju takšne rešitve pri Informacijskem pooblaščenca, v kolikor jim uspe zagotoviti, da so vsa dejanja obdelave osebnih podatkov pod izključnim nadzorom ali izključno oblastjo posameznika, čigar podatke se obdeluje. Tukaj v ospredje prihajata dve načeli, ki ju je vzpostavila GDPR, in sicer t.i. »privacy by design« in »privacy by default«, ki naj upravljavcem služita kot smernici pri vzpostavitvi biometričnih ukrepov brez odločbe Informacijskega pooblaščenca.

Svoj košček v biometričnem mozaiku pa bo dodala tudi akreditacija, ki bo določala, kako široko se bodo odprla vrata za uporabo biometrije in nenazadnje tudi kako visok bo prag (tako z vidika tehnologije kot tudi z vidika stroškov postopka akreditacije), ki ga bo potrebno prestopiti za rabo takšne tehnologije.

V luči povedanega torej lahko zaključimo, da ostaja uporaba biometrije v Sloveniji visoko regulirana, vendar pa je, v primerjavi s prej veljavnim zakonom, prišlo do vidnih premikov k dostopnosti tovrstne tehnologije.

#### **Viri in literatura**

- UREDBA (EU) 2016/679 EVROPSKEGA PARLAMENTA IN SVETA z dne 27. aprila 2016 o varstvu posameznikov pri obdelavi osebnih podatkov in o prostem pretoku takih podatkov ter o razveljavitvi Direktive 95/46/ES (Splošna uredba o varstvu podatkov. Dostopno na: <https://eur-lex.europa.eu/legal-content/SL/TXT/PDF/?uri=CELEX:32016R0679>;
- Zakon o varstvu osebnih podatkov, UL RS; št. 003-02-22024-4. Dostopno na: <https://www.uradni-list.si/glasilo-uradni-list-rs/vsebina/2022-01-4187?sop=2022-01-4187>;
- Infografika Informacijskega pooblaščenca, ki je dostopna na: [https://www.ip-rs.si/fileadmin/user\\_upload/Pdf/infografike/Infografika%20-%20osebni%20podatki2.pdf](https://www.ip-rs.si/fileadmin/user_upload/Pdf/infografike/Infografika%20-%20osebni%20podatki2.pdf);
- MATJAŠEC, Matjaž, 2018, Metode biometričnega overjanja: sistematičen pregled literature [na spletu]. Magistrsko delo. Maribor: Univerza v Mariboru. [Dostopano 19 februar 2024]. Pridobljeno s: <https://dk.um.si/IzpisGradiva.php?lang=slv&cid=70945>;
- Smernice španskega nadzornega organa za varstvo osebnih podatkov; 14 misunderstandings with regard to biometric identification and authentication, dostopno na: <https://www.aepd.es/documento/nota-equivocos-biometria-en.pdf>.
- Mnenje Informacijskega pooblaščenca; št. 07121-1/2024/128, z dne 5.2.2024, dostopno na: <https://www.ip-rs.si/mnenja-zvop-2/potrjevanje-obdelav-biometri%C4%8Dnih-op-1707898494>.

# **DOPOLDANSKA IN POPOLDANSKA SEKCIJA**

**Industrijske aplikacije**  
**Medicinske in biomedicinske aplikacije**  
**Študentske aplikacije**  
**Drugo**

---





# DEMONSTRACIJSKA CELICA ZA PRIKAZ GLOBOKEGA UČENJA V PRAKTIČNIH APLIKACIJAH

DOMEN TABERNIK, PETER MLAKAR, JAKOB BOŽIČ,  
LUKA ČEHOVIN ZAJC, VID RIJAVEC, DANIJEL SKOČAJ

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana, Slovenija  
domen.tabernik@fri.uni-lj.si, pm4824@student.uni-lj.si, jakob.bozic@gmail.com,  
luka.cehovin@fri.uni-lj.si, vid.rjavec@fri.uni-lj.si, daniel.skocaj@fri.uni-lj.si

V zadnjih letih so metode globokega učenja postale ključno orodje za reševanje raznolikih praktičnih izzivov. Kljub temu pa potencial takih metod pogosto ostaja slabo razumljiv širši javnosti zaradi pogostega ločevanja razvoja in demonstracije algoritmov od dejanskih praktičnih problemov, ki jih algoritmi naslavlajo. V tem članku predstavljamo demonstracijsko celico, ki združuje strojno in programsko opremo ter algoritme globokega učenja, omogočajoč enostavno prikazovanje delovanja teh metod v različnih aplikativnih domenah. Celica vključuje kamere, grafični vmesnik in pet demonstracijskih programov, ki demonstrirajo klasifikacijo lesenih desk, detekcijo površinskih anomalij, štetje polipov, detekcijo prometnih znakov in detekcijo vogalov tekstilnih izdelkov. Implementiran modularni pristop omogoča enostavno integracijo različnih algoritmov globokega učenja. Sistem omogoča boljše razumevanje in uporabo teh metod v praktičnih scenarijih ter prispeva k razvoju inovativnih rešitev na področju globokega učenja.

DOI  
[https://doi.org/  
10.18690/um.feri.1.2024.2](https://doi.org/10.18690/um.feri.1.2024.2)

ISBN  
978-961-286-837-6

**Ključne besede:**  
demonstracijska celica,  
globoko učenje,  
integracija algoritmov,  
praktične aplikacije,  
klasifikacija,  
detekcija anomalij,  
štetje polipov,  
prometni znaki,  
detekcija vogalov tekstilnih  
izdelkov

DOI  
[https://doi.org/  
10.18690/um.feri.1.2024.2](https://doi.org/10.18690/um.feri.1.2024.2)

ISBN  
978-961-286-837-6

**Keywords:**

demonstration cell,  
deep learning,  
algorithm integration,  
practical applications,  
classification,  
anomaly detection,  
polyp counting,  
traffic signs,  
grasp point detection

# DEMONSTRATION CELL FOR SHOWCASING DEEP LEARNING IN PRACTICAL APPLICATIONS

DOMEN TABERNIK, PETER MLAKAR, JAKOB BOŽIČ,  
LUKA ČEHOVIN ZAJC, VID RIJAVEC, DANIJEL SKOČAJ  
University of Ljubljana, Faculty of Computer and Information Science, Ljubljana,  
Slovenia  
[domen.tabernik@fri.uni-lj.si](mailto:domen.tabernik@fri.uni-lj.si), [pm4824@student.uni-lj.si](mailto:pm4824@student.uni-lj.si), [jakob.bozic@gmail.com](mailto:jakob.bozic@gmail.com),  
[luka.cehovin@fri.uni-lj.si](mailto:luka.cehovin@fri.uni-lj.si), [vid.rjavec@fri.uni-lj.si](mailto:vid.rjavec@fri.uni-lj.si), [danijel.skocaj@fri.uni-lj.si](mailto:danijel.skocaj@fri.uni-lj.si)

In recent years, deep learning methods have become a crucial tool for solving diverse practical challenges. However, their potential often remains poorly understood by a broader audience due to a separation between development and demonstration of algorithms and actual problems being solved by those algorithms. In this article, we introduce an innovative demonstration cell that combines hardware, software, and deep learning algorithms, enabling easy showcasing of these methods in various application domains. The cell includes cameras, a graphical interface, and five demonstration programs that demonstrate the classification of wooden boards, detection of surface anomalies, counting of polyps, recognition of traffic signs, and detection of corners on towels and cloths. The implemented modular approach allows for the straightforward integration of different deep learning algorithms. The system enhances understanding and application of these methods in practical scenarios, contributing to the development of innovative solutions in the field of deep learning.



## 1 Uvod

Metode globokega učenja so v zadnjih letih postale izredno močno orodje za reševanje raznolikih praktičnih problemov. Navkljub temu pa je njihova demonstracija širši publiki pogosto težavna, saj se razvoj in demonstracija algoritmov vršita povsem ločeno od končnih sistemov, kjer taki algoritmi dejansko rešujejo praktične probleme. Zaradi tega širša publika pogosto spregleda potencial takih algoritmov za uporabo v aplikativnih domenah.

V ta namen smo ustvarili demonstracijsko celico s katero lahko enostavno prikažemo delovanje najnovejših metod globokega učenja v kontekstu različnih praktičnih problemov. Celica sestoji iz ohišja, kamer, strojne opreme, programske opreme za njen nadzor ter iz vrsto demonstracijskih programov z algoritmi globokega učenja. Glavni cilj pri zasnovi celice je bila enostavna uporaba na eni strani ter na drugi strani enostavna integracija različnih vrst algoritmov globokega učenja, ki so pogosto razviti za raziskovalne namene ter zahtevajo specifično programsko okolje za pravilno delovanje. V ta namen je bilo razvito posebno programsko ogrodje skupaj s petimi različnimi demonstracijskimi programi, kjer vsak rešuje specifičen praktičen problem ter jih je mogoče demonstrirati na realnih predmetih postavljenimi v celico.

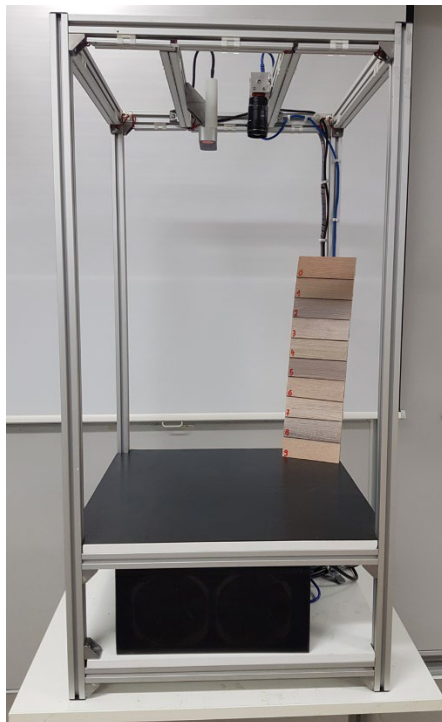
## 2 Opis sistema

Demonstracijska celica je sestavljena iz ohišja, strojne opreme ter namensko ustvarjenje programske opreme za poganjanje poljubnih metod globokega učenja v demonstracijskem načinu.

### 2.1 Ohišje in strojna oprema

Celotna demonstracijska celica obsega 60 x 60 x 120 cm veliko ogrodje sestavljeno iz ALU profila, ki je razdeljeno na dva prekata: i) glavni demonstracijski prekat ter ii) manjši prekat višine 20 cm za strojno opremo na dnu celice. Na vrhu celice se nahajajo nosilci za montažo senzorjev, ki so usmerjeni navzdol, ter 6 programsko nadzorovanih LED luči. Celica vsebuje 12MP kamero *Allied Vision Alvium 1800 U-1240c* za zajem slik ter *Kinect Azure* za zajem slik z globinsko informacijo, sočasno pa omogoča enostavno montažo dodatnih kamer z nosilci pritrjenimi na ALU profile. Prekat za strojno opremo vsebuje modul za napajanje in nadzor luči ter majhen

glavni računalnik, sestavljen iz procesorja Intel Core i5-11600KF 3.90GHz, 16 GB pomnilnika ter grafične enote NVIDIA GeForce RTX 3060 z 12GB pomnilnika. Celica vsebuje še 24" LCD zaslon na dotik za upravljanje in vizualizacijo demonstracijskih programov. Celotna demonstracijska celica je prikazana na Sliki 1.



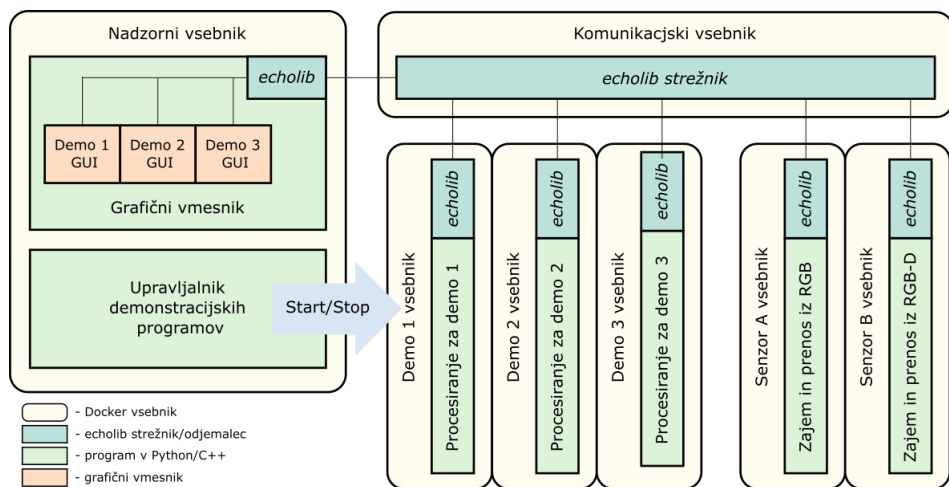
**Slika 1: Demonstracijska celica.**

Vir: lasten.

## 2.2 Programska oprema

Pri razvoju programske opreme smo zasledovali dva pomembna cilja: i) enostavno uporabo z zaslonom na dotik, ter ii) enostavno integracijo različnih vrst demonstracijskih programov. V ta namen smo razvili namenski grafični vmesnik enostaven za upravljanje na dotik, ter celotno programsko okolje zasnovali modularno na osnovi sistema vsebnikov. Sistem je tako razdeljen na: i) grafični vmesnik z modulom za upravljanje z demonstracijskimi programi, ii) na podsistem za zajem in prenos slik, iii) na posamezne demonstracijske programe, ter iv) na komunikacijski podsistem. Arhitektura programske celice je prikazana na Sliki 2.

Vsak od štirih podsistemov je implementiran v svojem vsebniku, s čimer povsem zamejimo odvisnosti od knjižnic na vsak podsistem. Tak pristop primarno razrešuje problem odvisnosti knjižnic pri različnih demonstracijskih programih, kjer je lahko vsak program implementiran na različen način in z uporabo različnih programskih jezikov in orodij. Na primer, prvi demonstracijski program je lahko implementiran v programskem jeziku Python z ogrodjem PyTorch 1.9, ki zahteva knjižnico CUDA 10, medtem ko je drugi program lahko implementiran v jeziku C++ z ogrodjem TensorFlow v2, ki zahteva knjižnico CUDA 11. Ker je v našem sistemu vsak demonstracijski program implementiran v svojem vsebniku, pa tako povsem izločimo potrebo po sočasni prisotnosti med seboj nekompatibilnih knjižnic. V praksi implementiramo vse podsisteme z orodjem Docker.



Slika 2: Programska arhitektura demonstracijske celice.

Vir: lasten.

Podsistem za zajem in prenos slik je prav tako implementiran v ločenem vsebniku Docker. Tako omogočimo enostavno uporabo različnih tipov kamer, ki zahtevajo vsak svoje gonilnike ter ogrodja. Za demonstracijsko celico smo implementirali dva vsebnika: i) vsebnik za prenos slik iz senzorja *Allied Vision Avium 1800*, ter ii) vsebnik za prenos slik iz senzorja *Kinect Azure*. Prvi omogoča zajem RGB slik, medtem ko drugi omogoča zajem RGB-D slik ter dodatno tudi informacijo iz inercialnega senzorja.

Kamere, demonstracijski programi ter podsistem za upravljanje demonstracijskih programov z grafičnim vmesnikom med seboj komunicirajo preko dodatnega komunikacijskega podsistema. Komunikacija je mogoča na podlagi vzorca objavi-naroči (ang. *publish-subscribe*), podobno kot pri sistemu ROS. Komunikacijski podsistem smo implementirali v ločenem Docker vsebniku z odprtokodno knjižnico *echolib*<sup>1</sup>, ki je prenosljiva ter lahka implementacija protokola objavi-naroči. Posameznimi podsistemi tako komunicirajo med seboj preko različnih tem (npr. tema za prenos slik, tema za prenos rezultatov za prikaz, tema za nadzor demonstracijskega programa preko grafičnega vmesnika, itd.). Tak način komuniciranja omogoča tudi sočasen prenos slik iz več kamer, kjer lahko vsaka kamera prenaša slike na svoji temi, nato pa se demonstracijski program odloči katero kamero bo uporabil.

Pomemben del predstavlja tudi podsistem z grafičnim vmesnikom ter modulom za upravljanje z demonstracijskimi programi. Grafični vmesnik smo zasnovali na podlagi odprtokodne knjižnice<sup>2</sup>, ki omogoča uporabo ogrodja OpenGL preko programskega jezika Python. Sam grafični vmesnik je relativno preprost ter primarno omogoča preklapljanje med posameznimi demonstracijskimi programi. Preklapljanje med demonstracijskimi programi nadzira modul za upravljanje z demonstracijskimi programi, ki je implementiran v programskem jeziku Python. Na vsako zahtevo po prikazu specifičnega demonstracijskega programa tako modul požene in ustavi posamezne vsebnike z demonstracijskimi programi.

## 2.2 Implementacija demonstracijskega programa

Posamezen demonstracijski program ter njegov grafični vmesnik sta implementirana ločeno od preostalih podsistemov. S tem omogočimo enostavno dodajanje novih algoritmov. Glavni del programa predstavlja aplikacija, ki se avtomatsko požene ob zagonu vsebnika, ter se mora preko komunikacijskega podsistema *echolib* povezati s kamerami ter z grafičnim vmesnikom. V vsebniku se tako izvede priklop na zajem in prenos slik iz specifičnega senzorja ter ustrezna obdelava slik, kar se nato pošlje v glavni grafični vmesnik za izris na zaslon. Del tega procesa je tudi prikaz slike ter rezultatov detekcij, ki se izvede že znotraj demonstracijskega vsebnika ter tako omogoča enostavno integracijo različnih obdelav in prikazov rezultatov.

---

<sup>1</sup> <https://github.com/vicoslab/echolib>

<sup>2</sup> [https://github.com/petermlakar/opengl\\_gui](https://github.com/petermlakar/opengl_gui)

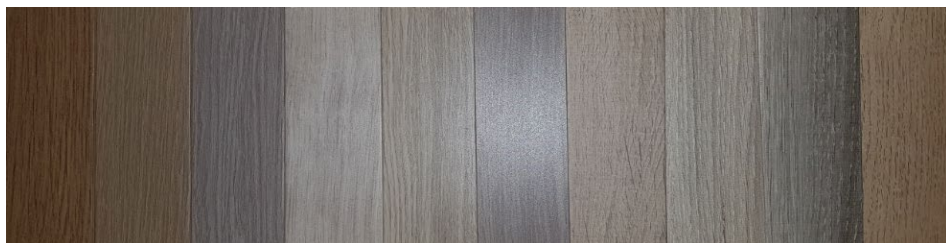
Del demonstracijskega programa je tudi grafični vmesnik za nadzor posameznih funkcij procesiranja (npr. gumb za zagon detekcije, nastavitve upragovanja, itd.), ki pa ni implementiran znotraj demonstracijskega vsebnika, ampak se dinamično vključi v glavni grafični vmesnik ob zagonu programa. Zaradi tega poteka komunikacija med grafičnim vmesnikom za demonstracijski program ter njegovim zajemom in procesiranjem slik preko komunikacijskega podsistema *echolib*.

### 3 Demonstracijski programi

Predstavljen sistem je bil zasnovan za enostavno integracijo različnih demonstracijskih programov za prikaz delovanja algoritmov globokega učenja. Implementirali smo pet spodaj opisanih programov.

#### 3.1 Klasifikacija lesenih desk

Demonstracijski program predstavlja praktičen primer klasifikacije lesenih desk v kakovostne razrede, ki je bil predstavljen v (Muhovič, Tabernik, & Skočaj, 2020). Za model smo izbrali nevronska mrežo EfficientNet-B4, ki smo jo naučili za razlikovanje med 10 različnimi tipi lesenih površin (primeri na Sliki 3). Za učenje smo uporabili 290 slik zajetih na istem demonstracijskem sistemu (29 slik za vsako kategorijo) ter učili model za 200 epoh z optimizatorjem Adam ter funkcijo izgube binarne prečne entropije.



**Slika 3: Deset kategorij kakovosti lesa.**

Vir: lasten.

Pri napovedovanju (ter tudi pri učenju) se iz slike predhodno izreže posamezno desko z enostavnim upragovanjem (ozadje je vedno črne barve) ter poravna in poveča na velikost 1512 x 536 pikslov. Grafični vmesnik demonstracijskega programa vsebuje gumb za izvedbo klasifikacije, ter prikaz rezultata v obliki segmentacije posamezne deske ter številko razpoznanega razreda deske (Slika 5).

Klasifikacijski program je implementiran v programskem okolju Python z ogrodjem PyTorch v1.7.1 ter knjižnico CUDA v11.2 in CuDNN v8 na osnovi sistema Ubuntu 18.04.

### 3.2 Detekcija površinskih anomalij

Za demonstracijo nenadzorovanih metod detekcije površinskih anomalij, kjer model učimo na izključno dobrih primerih, smo izbrali metodo DREAM (Zavrtanik, Kristan, & Skočaj, 2021), ter jo aplicirali na detekcijo anomalij na ploščicah. Metoda DRAEM, predstavlja rekonstrukcijski tip nenadzorovanih metod učenja normalnega izgleda, kjer se z arhitekturo kodirnik-dekodirnik izvede rekonstrukcijo slike, nato pa z dodatnim diskriminativnim kodirnik-dekodirnikom, naučenim na sintetičnih napakah, poskuša detektirati odstopanja od normalnega izgleda, t.j., anomalije. Za učenje smo pripravili 300 slik ploščic brez napak, ter model učili 50 epoch z optimizatorjem Adam.



Slika 4: Detekcija anomalij na ploščicah. Levo: dober primer, desno: anomalija (verjetnost anomalije v oklepajih).

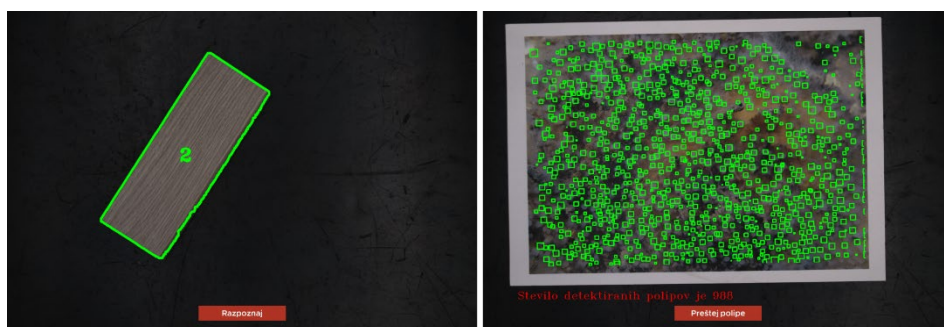
Vir: lasten.

Pri napovedovanju (ter tudi pri učenju) se iz slike predhodno izreže posamezno ploščico z enostavnim upragovanjem ter poravna in poveča na velikost 480 x 480 pikslov. Grafični vmesnik demonstracijskega programa vsebuje gumb za izvedbo detekcije, ter prikaz rezultata v obliki segmentacije posamezne ploščice ter indikator prisotnosti anomalije v obliki barve segmentacije, kot je prikazano na Sliki 4 (rdeča – anomalija, zelena – brez anomalije). Detektor anomalij je implementiran v programskem okolju Python z ogrodjem PyTorch v1.7.1 ter knjižnico CUDA v11.2 in CuDNN v8 na osnovi sistema Ubuntu 18.04.



### 3.3 Štetje polipov

Za demonstracijski program smo implementirali tudi model PoCo (Zavrtanik, Vodopivec, & Kristan, 2020) za štetje polipov. Metoda PoCo temelji na arhitekturi U-Net, ter je sestavljena iz kodirnika in dekodirnika, ter vmesnimi povezavami. Metoda šteje polipe na podlagi segmentacije, kjer se okoli polipov segmentira krog v odvisnosti od velikosti polipov, nato pa se z pomočjo transformacije razdalje detektira in prešteje prekrivajoče se polipe z uporabo konsenza razdalij (ang. *distance consensus points*). Model je bil naučen na 37 slikah s 32.685 označenimi polipi.



Slika 5: Demonstracija klasifikacije lesa (levo) ter štetja polipov (desno)

Vir: lasten.

Ker se objekti naravno nahajajo le v morju, le teh ni mogoče fizično prikazati na demonstratorju, zato za demonstracijo delovanja prikažemo natisnjeno sliko s zajetimi polipi. Pri napovedovanju se zato iz slike odstrani ozadje z enostavnim upragovanjem ter sliko s polipi poravna in poveča na velikost 2080 x 1470 pikslov. Grafični vmesnik demonstracijskega programa vsebuje gumb za izvedbo štetja, ter prikaz rezultata v obliki številke detektiranih polipov (Slika 5). Program za štetje polipov je implementiran v programskem okolju Python z ogrodjem TensorFlow v2.6.2 ter knjižnico CUDA v11.2 in CuDNN v8 na osnovi sistema Ubuntu 18.04.

### 3.4 Detekcija prometnih znakov

Implementirali smo tudi demonstracijski program za prikaz algoritma detekcije prometnih znakov predstavljenim v (Tabernik & Skočaj, 2019). Detektor sloni na metodi Faster/Mask R-CNN (He, Gkioxari, Dollár, & Girshick, 2017), naučeni za

detekcijo 200 različnih kategorij prometnih znakov. Model je bil naučen na podatkovni množici s preko 5.000 slik zajetih na slovenskih cestah, ki vsebuje preko 10.000 označenih prometnih znakov. Za hrbtenico smo vzeli arhitekturo ResNet50, ki omogoča izvajanje v realnem času. Model je bil učen 95 epoh, kjer so bile slike zmanjšane, tako da je bila najmanjša stranica velika 840 pikslov.



Slika 6: Detekcija prometnih znakov.

Vir: lasten.

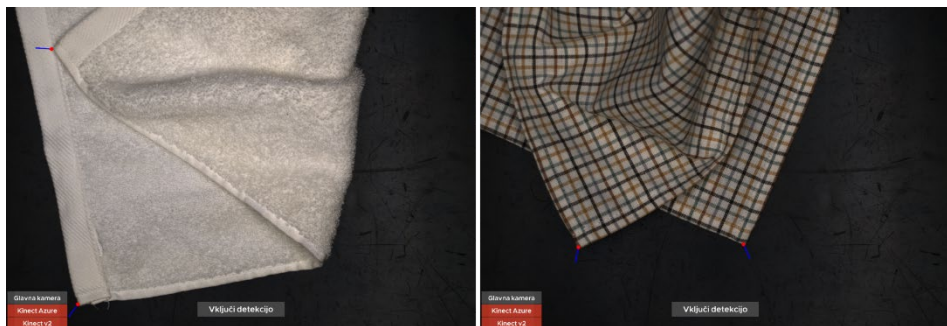
Grafični vmesnik demonstracijskega programa vsebuje gumb za izvedbo detekcije, ter prikaz rezultata v obliki očrtanih pravokotnikov okoli prometnih znakov. Dodatno izrišemo tudi zaznano kategorijo v obliki slike tipičnega predstavnika te kategorije, kot je to prikazano na Sliki 6. Program napovedovanja je implementiran v programskem okolju Python z ogrodjem Detectron in Caffe2 ter knjižnico CUDA v11.1 in CuDNN v8 na osnovi sistema Ubuntu 16.04.

### 3.5 Detekcija vogalov brisač in krp

Implementirali smo tudi demonstracijski program za detekcijo vogalov brisač in krp, ki se lahko uporablja kot detektor točk prijema za robotske aplikacije. Za detekcijo kotnih robov smo aplicirali model CeDiRNet (Tabernik, Muhovič, & Skočaj, 2023), ki sloni na dvostopenjski arhitekturi z regresijo smernih vektorjev na prvi stopnji, ter lokalizacijo točk na drugi stopnji. Metoda je bila naučena na preko 5.000 učnih slikah s prikazanimi brisačami v raznolikih konfiguracijah ter svetlobnimi pogoji.

Naučena metoda vsebuje tudi možnost uporabe globinske informacije zato za ta demonstracijski program omogočamo uporabo tako navadne RGB kamere (*Allied Vision*) kot tudi RGB-D kamere (*Kinect Azure*). V ta namen grafični vmesnik poleg

gumba za zagon detekcije vsebuje tudi opcijo za izbiro vira kamere. Rezultat detekcije prikažemo v obliki detektiranih točk, dodatno pa prikažemo tudi smer, ki določa potencialen kot pristopa za robotsko roko, kot je to prikazano na Sliki 7. Detektor vogalov je implementiran v programskem okolju Python z ogrodjem PyTorch v1.13.1 ter knjižnico CUDA v11.7 in CuDNN v8 na osnovi sistema Ubuntu 20.04.



**Slika 7: Detekcija vogalov na brisači.**

Vir: lasten.

## 4 Zaključek

V članku smo predstavili demonstracijsko celico, ki omogoča preprost prikaz delovanja metod globokega učenja v različnih praktičnih aplikacijah. Celica, ki združuje strojno in programsko opremo, ponuja modularno okolje za enostavno integracijo različnih algoritmov globokega učenja. Z izvedbo petih demonstracijskih programov, ki segajo od klasifikacije do detekcije, demonstracijska celica ponuja vpogled v uporabo globokega učenja v resničnih scenarijih. Razviti sistem prispeva k širšemu razumevanju potenciala in praktične uporabe metod globokega učenja, kar odpira vrata inovativnim rešitvam v različnih aplikacijskih domenah. S tem članek spodbuja nadaljnje raziskave in razvoj za praktično demonstracijo metod globokega učenja v aplikativnih domenah.

## Zahvala

To delo je bilo delno podprto s strani raziskovalnih projektov ARIS J2-3169 (MV4.0) in J2-4457 (RTFM), kot tudi s strani raziskovalnega programa P2-0214.

**Viri in literatura**

- He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. (str. 2961–2969). International Conference on Computer Vision.
- Muhovič, J. N., Tabernik, D., & Skočaj, D. (2020). O klasifikaciji slik v ne-enolično določljive razrede. (str. 355-358). Zbornik devetindvajsete mednarodne Elektrotehniške in računalniške konference ERK 2020.
- Tabernik, D., & Skočaj, D. (2019). Deep Learning for Large-Scale Traffic-Sign Detection and Recognition. *IEEE Transactions on Intelligent Transportation Systems*.
- Tabernik, D., Muhovič, J. N., & Skočaj, D. (2023). Lokalizacija in ocenjevanje lege predmeta v treh prostostnih. *Zbornik dvaintridesete mednarodne Elektrotehniške in računalniške konference ERK 2023*.
- Zavrtanik, V., Kristan, M., & Skočaj, D. (2021). DRAEM -- A discriminatively trained reconstruction embedding for surface anomaly detection. *International Conference on Computer Vision*.
- Zavrtanik, V., Vodopivec, M., & Kristan, M. (2020). A segmentation-based approach for polyp counting in the wild. *Engineering Applications of Artificial Intelligence*.

# GRADNJA KONVOLUCIJSKIH NEVRONSKIH MREŽ ZA RAZVRŠČANJE SLIK Z UPORABO EVOLUCIJSKIH ALGORITMOV

UROŠ MLAKAR

Univerza v Mariboru, Fakulteta za elektrotehniko računalništvo in informatiko, Maribor,  
Slovenija  
uros.mlakar@um.si

Globoke nevronske mreže so bile uspešno aplicirane že na mnogih področjih obdelave slik. Uspešnost mreže je vedno pogojena z njeno arhitekturo, ki pogosto zahteva ročno oblikovanje strokovnjaka z bogatim strokovnim znanjem. Takšen pristop je lahko v realnem svetu zamuden, morda tudi neizvedljiv, predvsem zaradi primanjkljaja izkušenj načrtovalcev oz. njihovega znanja. V tem članku smo predstavili postopek avtomatskega iskanja topologije globoke nevronske mreže v aplikaciji razvrščanja slik. Iskanje ustrezne topologije smo preslikali v optimizacijski problem, ki ga rešujemo z algoritmom diferencialne evolucije. Algoritem smo testirali nad podatkovnima zbirkama CIFAR10 in AffectNet. Dobljeni rezultati so obetavni in odpirajo novo mlado razsikalno področje načrtovanja globokih mrež brez predhodnega ekspertnega znanja.

DOI  
[https://doi.org/  
10.18690/um.feri.1.2024.3](https://doi.org/10.18690/um.feri.1.2024.3)

ISBN  
978-961-286-837-6

**Ključne besede:**  
evolucijski algoritem,  
globoka mreže,  
razvrščanje slik,  
generiranje globokih mrež,  
optimizacija

DOI  
[https://doi.org/  
10.18690/um.feri.1.2024.3](https://doi.org/10.18690/um.feri.1.2024.3)

ISBN  
978-961-286-837-6

**Keywords:**  
evolutionary algorithm,  
deep networks,  
image classification,  
generating deep networks,  
optimization

# BUILDING CONVOLUTIONAL NEURAL NETWORKS FOR IMAGE CLASSIFICATION USING EVOLUTIONARY ALGORITHMS

UROŠ MLAKAR

University of Maribor, Faculty of Electrical Engineering and Computer Science,  
Maribor, Slovenia  
[uros.mlakar@um.si](mailto:uros.mlakar@um.si)

Deep neural networks have already been successfully applied in many areas of image processing. The success of a network is always conditioned by its architecture, which requires manual design with expert knowledge. In the real world, such an approach can be time-consuming, perhaps even unfeasible, mainly due to the lack of experience of users or of their knowledge. In this paper, we presented the process of automatically finding the topology of a deep network in an image classification application. We mapped the search for a suitable topology into an optimization problem, which is solved using the differential evolution algorithm. We tested the algorithm on the CIFAR10 and AffectNet datasets. The obtained results are promising and open up a new young research field of designing deep networks without prior expert knowledge.



## 1 Uvod

Globoke nevronske mreže (angl. Deep Neural Networks - DNN), ki so temelj globokega učenja (cite), so se v zadnjih letih izkazale uspešne pri različnem naboru aplikacij, kjer so največ uporabljene pri razvrščanju slik (He, 2016) (Huang, 2017), procesiranju naravnega jezika, prepoznavanju govora (Zhang, 2017) in drugih. Njihova pomembna lastnost je sposobnost prepoznavanja in učenja pomembnih lastnosti vhodnih podatkov, brez predhodnega ročnega oblikovanja značilnic. V splošnem uspešnost DNN pogojujeta dva vidika in sicer zasnova arhitekture ter pripadajoče uteži. Šele ko sta oba hkrati v optimalnem stanju, lahko pričakujemo obetavno delovanje DNN. Optimalne uteži se pogosto pridobijo s postopkom učenja: z uporabo zvezne funkcije izgube za merjenje razlik med dejanskim in želenim izhodom, nato pa se za minimiziranje izgube pogosto uporabljajo algoritmi, ki temeljijo na gradientnem spustu. Ko je izpolnjen pogoj za zaključek učenja, ki je običajno število učnih iteracij, lahko algoritem pogosto najde dober nabor uteži (LeCun, 2015). Takšen postopek je v praksi zelo priljubljen predvsem zaradi učinkovitosti in se ga največ uporablja za optimizacijo uteži, čeprav gre zgolj za algoritme lokalnega iskanja. Po drugi strani pa iskanje optimalnih arhitektur ni mogoče neposredno formulirati z zvezno funkcijo, ne obstaja pa tudi eksplicitna funkcija za merjenje procesa iskanja optimalne arhitekture.

Obetavne arhitekture, ki dosegajo dobre rezultate večinoma razvijejo strokovnjaki z obširnimi domenskim strokovnim znanjem. Primer takšnih arhitektur so npr. VGG (Zisserman, 2015), ResNet (Sun, 2015) in DenseNet (Weinberger, 2018). Vse omenjene modele konvolucijskih nevronske mreže (angl. Convolutional Neural Networks - CNN) so ročno zasnovali raziskovalci z bogatim znanjem s področja nevronske mreže in obdelave slik. Pogosto se dogaja, da v praksi večina končnih uporabnikov nima takšnega znanja. Poleg tega so arhitekture DNN/CNN pogosto odvisne tudi od reševanega problema. Če se porazdelitev vhodnih podatkov spremeni, je treba arhitekturo ustrezno spremeniti. Proces iskanja arhitektur globokih mrež (angl. Neural Architecture Search - NAS), katerega cilj je avtomatizacija načrtovanja arhitekture globokih nevronske mreže, je opredeljen kot obetaven način za reševanje zgoraj navedenih izzivov. Matematično lahko NAS modeliramo kot optimizacijski problem:

$$\operatorname{argmin}_{(A \in \mathcal{A})} = \Gamma(A, D_{učna}, D_{uspešnost}) \quad (1)$$

V enačbi (1) je  $\Lambda$  iskalni prostor vseh potencialnih arhitektur,  $F(\cdot)$  pa meri uspešnost posamezne arhitekture, ki je naučena z učno množico  $D_{train}$ .

Načeloma je NAS težek optimizacijski problem, postavljen pred več izzivov, kot so kompleksne omejitve, diskretne predstavitve, dvonivojske strukture, računske zahtevnosti in številna nasprotujoča si merila. Algoritmi NAS se nanašajo na optimizacijske algoritme, ki so posebej zasnovani za učinkovito in uspešno reševanje problema, ki ga predstavlja enačba (1).

NAS algoritme lahko glede na uporabljen optimizacijski algoritem delimo v 3 kategorije:

- Algoritmi, ki temeljijo na učenju z okrepitevijo (angl. Reinforcement Learning -LR) (Kaelbling, 1996),
- Algoritmi, ki temeljijo na gradientu in
- Evolucijski algoritmi (Back, 1997).

Algoritmi, ki uporabljajo RL, pogosto zahtevajo ogromno grafičnih kartic že za preprostejše probleme razvrščanja slik. Algoritmi, ki temeljijo na gradientu, so sicer bolj učinkoviti od algoritmov, ki temeljijo na RL, vendar pogosto najdejo slabše arhitekture zaradi neustreznega razmerja skaliranja za optimizacijski algoritem. Dodatno je pri načrtovanju teh algoritmov potrebno že precej ekspertnega znanja za učinkovito reševanje problema. V tretjo skupino spadajo pa evolucijski algoritmi, ki temeljijo na populaciji rešitev in simulirajo razvoj vrste ali obnašanje populacije v naravi. Sposobni so poiskati rešitev problema tudi kadar matematična oblika kriterijske funkcije ni na voljo. Ta skupina algoritmov ja za problem NAS sicer zelo zanimiva, vendar je podobno kot pristop z RL precej računsko požrešna.

V zadnjih nekaj letih se v literaturi pojavlja kar nekaj pristopov načrtovanja arhitektur globokih mrež z uporabo evolucijskih algoritmov na različnih področjih, kar nakazuje na očitno pomanjkanje ekspertnega znanja. V tem članku bomo predstavili algoritem diferencialne evolucije (angl. Differential Evolution - DE), ki spada v skupino evolucijskih algoritmov. Uporabili ga bomo za iskanje optimalne arhitekture CNN za izbran problem razvrščanja slik. Pri tem ne bomo uporabljali domenskega ekspertnega znanja.



## 2 Diferencialna evolucija

Diferencialna evolucija (DE) je stohastični populacijski algoritem, ki se uporablja za reševanje zveznih in tudi diskretnih problemov (Mlakar, 2017) (Brest, 2006). Navkljub svoji enostavnosti je izredno učinkovit pri reševanju realnih problemov. Temelji na matematičnem modelu, ki uporablja razlike vektorjev. DE algoritem razvija in spreminja populacijo vektorjev skozi generacije, pri čemer gre vsak vektor skozi niz evolucijskih operatorjev, vključno z mutacijo, križanjem in selekcijo. Populacijo v okviru algoritma DE lahko matematično zapišemo kot (Mlakar U. , 2019):

$$x_i^{(g)} = \left( x_{(i,1)}^{(g)}, \dots, x_{(i,D)}^{(g)} \right), \text{ za } i = 1, \dots, Np, \quad (2)$$

kjer je  $D$  dimenzija reševanega problema,  $Np$  je pa velikost populacije. Algoritem DE v prvem koraku izvede operator mutacije, nato pa mutiran vektor križa. Predlaganih je bilo že precej različnih mutacij, vendar se v algoritmu DE najpogosteje še vedno uporablja metoda "rand/1", ki jo matematično definiramo kot (Mlakar U. , 2019):

$$v_i^{(g)} = x_{r_1}^{(g)} + F \left( x_{r_2}^{(g)} - x_{r_3}^{(g)} \right). \quad (3)$$

V enačbi (3) so  $r_1, r_2$  in  $r_3$  naključno generirane celoštevilske vrednosti na intervalu

$[1, Np]$ ,  $F$  je skalirni faktor na intervalu  $[0,1]$ ,  $v_i^{(g)}$  pa je mutiran vektor. Dobljen mutiran vektor  $v_i^{(g)}$  je v drugem koraku križan z originalnim vektorjem  $x_i^{(g)}$ , kjer kot rezultat dobimo poskusni vektor. Križanje matematično zapišemo kot (Mlakar U. , 2019):

$$u_{i,j}^{(g)} = \begin{cases} v_{i,j}^{(g)}, & \text{čr } \text{rand}(0,1) \leq Cr \text{ ali } j = j_{rand}, \\ x_{i,j}^{(g)}, & \text{sicer.} \end{cases} \quad (4)$$

V enačbi (4) je verjetnost križanja  $C_r$  definirana na intervalu  $[0,1]$ . Dodatni pogoj  $j = j_{rand}$ , kjer je  $j_{rand} = rand(0, D)$ , zagotavlja, da bo poskusni vektor  $u_i^{(g)}$  od originalnega različen vsaj v enem elementu. S tem preprečimo izdelavo enakih posameznikov in posledično ne izgublamo časa z ocenjevanjem rešitve, ki smo jo že ocenili. V zadnjem koraku evucijskega procesa poskusni vektor tekmuje z originalnim v postopku selekcije. Vektor, katerega funkcija uspešnosti je boljša, je izbran za preživetje v naslednjo generacijo. Matematično operator selekcije zapišemo kot (Mlakar U., 2019):

$$x_i^{(g+1)} = \begin{cases} u_i^{(g)}, & \text{čef } (u_i^{(g)}) \leq f(x_i^{(g)}), \\ x_i^{(g)}, & \text{sicer.} \end{cases} \quad (4)$$

### 3 Algoritem diferencialne evolucije za reševanje problema NAS

Največji izziv pri uporabi evucijskih algoritmov za reševanje problem kot je NAS, je učinkovita in smiselna predstavitev posameznika v populaciji. V literaturi so pojavljata predvsem dva načina, in sicer predstavitev na podlagi plasti, ter predstavitev na podlagi blokov. Pri prvem so osnovni elementi v kodirnem prostoru kodiranja osnovne plasti, kot so npr. kovolucijske ali polno povezane. Pri tem načinu se pojavlja težava ogromnega iskalnega prostora, saj poskušamo zakodirati veliko količino informacij. Skoraj zagotovo bom s tem kodiranjem porabili tudi več časa, da najdemo primerno rešitev, predvsem zato, ker je težje sestaviti dobro delujočo DNN/CNN zgolj z osnovnimi sloji. S takim kodiranjem zagotovo ne moremo poiskati oz. najti mreže, ki je po topologiji podobna Resnet-u. Pri drugem načinu kodiranja, pa so osnovni gradniki v iskalnem prostoru bloki, ki so se že izkazali kot primerni v drugih tipih mrež. S tem načinom kodiranja sicer lahko pohitrimo postopek iskanja optimalne mreže, vendar potrebujemo že nekaj ekspertnega znanja, saj moramo izbrati takšne bloke, ki bodo ustrezali reševanemu problemu. V tem delu smo izbrali srednjo pot, kjer smo kombinirali obe predstavitvi.

Vsaka rešitev v populaciji algoritma DE predstavlja globoko arhitekturo, ki jo je potrebno ustrezno preslikati iz iskalnega prostora v prostor rešitev. To preslikavo naredimo z ustreznimi preslikovalnimi funkcijami, s katerimi zagotovimo pravilnost zapisa arhitekture in jo s tem pripravimo na učenje. Delovanje preslikovalnih funkcij je deterministično, saj se določena vrednost v določenih mejah iz iskalnega prostora

vedno preslika v enak sloj. Vsak sloj nosi s seboj v prostoru rešitev tudi dodatne informacije, ki so potrebne za pravilno preslikavo. Globina arhitekture mreže je odvisna od dimezije rešitve v iskalnem prostoru.

Vsaka dimenzija torej predstavlja gradnik mreže, ki je lahko osnovni sloj ali pa sestavljen blok. V tem delu smo uporabili naslednje osnovne sloje:

- Konvolucijski sloj,
- Polno povezan sloj (angl. Fully-Connected layer) in
- Združevalni sloj (angl. Pooling layer),
- Izpustni sloj (angl. Dropout layer) ter

naslednje bloke:

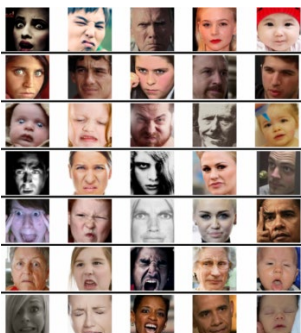
- Rezidualni blok (angl. Resnet Block), kot je uporabljen v ResNet arhitekturi in
- Gost blok (angl. Dense Block), kot je uporabljen v DenseNet arhitekturi.

Konvolucijski sloj ima v prostoru rešitev dodano število uporabljenih filtrov, polno povezan sloj nosi informacije o številu nevronov ki jih vsebuje, združevalni sloj pa definira še tip združevanja (maksimalno ali povprečno). Izpustni sloj nosi informacije o odstotku povezav, ki jih bo ignoriral. Oba uporabljena bloka sta zaradi manjšanja velikosti iskalnega prostora in posledično računske zahtevnosti nespremenljiva.

## 4 Eksperimentalno okolje

V tem poglavju bomo opisali eksperimentalno okolje, v katerem so tekli vsi eksperimenti v okviru tega dela. Za testiranje predlaganega algoritma DE za iskanje optimalnih globokih arhitektur, smo uporabili podatkovni zbirki CIFAR10 (Hinton, 2009) in AffectNet (Mollahosseini, 2019). Podatkovna zbirka CIFAR10 se pogosto pojavlja pri validaciji rezultatov globokih mrež, ne glede na izbran pristop gradnje mreže. Vsebuje 60000 slik, ki jih je možno razvrstiti v 10 razredov. Vsaka slika je velikosti 32 x 32 slikovnih elementov v barvnem prostoru RGB. Na drugi strani pa podatkovna zbirka Affectnet vsebuje približno 440000 slik, ki vsebuje slike obraznih

izrazov, ki jih je možno razvrstiti v 8 razredov. V tej podatkovni zbirki so slike velike 96 x 96 slikovnih elementov, tudi v barvnem prostoru RGB.



Slika 1: Vzorci slik iz podatkovne zbirke AffectNet.

Vir: lasten

Za potrebe validacije predlaganega algoritma na problemu NAS smo podatkovno zbirko razdeliti na učno in testno množico. CIFAR10 ima sicer ti množici že strogo definirani, kjer se v učni množici nahaja 50000 slik, v testni pa 10000. Pri podatkovni zbirki AffectNet smo delitev morali narediti sami. Razdelitev smo opravili v razmerju 90 % za učno množico in 10 % za testno. Razdelitev smo naredili na nivoju števila obraznih izrazov. Dodatno smo pri delitvi zagotovili, da se slike, ki so bile izbrane za učno množico, niso pojavile hkrati še v testni. Za potrebe učenja in zagotavljanja generalizacije naučene mreže, smo dodatno iz učne množice izvzeli 20 % podatkov, ki smo jih uporabili za validacijsko množico. Ta se je uporabljala za ovrednotenje kriterijske funkcije, ko je arhitektura končala z učenjem na učni množici po določenem številu epoch. Ko je algoritem zaključil z iskanjem, smo najboljšo najdeno arhitekturo testirali na testni množici. Uspešnost razvrščanja nad testno množico je tudi končni rezultat najdene mreže.

Parametre algoritma, kot so velikost populacije, število generacij, začetna dimenzija problema, število epoch in velikost paketa smo nastavljali eksperimentalno. Hkrati smo spreminjali tudi dovoljene gradnike v arhitekturah. Za učni algoritem smo izbrali algoritem Adam s privzetimi nastavitvami v knjižnici Keras. Ker gre za računsko težek problem, smo algoritem, napisan v programskem jeziku python, prilagodili za izvajanje na superračunalniku Vega, kjer smo poskrbeli za sočasno ovrednotenje celotne populacije na trenutno razpoložljivih virih.

## 5 Rezultati

V tem poglavju bomo predstavili rezultate eksperimentalnega dela. Cilj eskperimentov je bil pokazati, da je možno z algoritmom DE najti smiselne arhitekture za izbran problem razvrščanja slik, brez posredovanja ekspertnega znanja iskalnemu algoritmu. Dodatno smo želeli še preveriti, kako velikost učne množice vpliva na uspešnost iskanja optimalne arhitekture. Ker spada algoritem DE med stohastične algoritme, smo ga za vsako kombinacijo parametrov zagnali 5 krat. V Tabeli 1 so zbrani rezultati za podatkovno zbirko CIFAR10, v Tabeli 2 pa rezultati za AffectNet. Stolpec gradniki v obeh tabelah opisuje dovoljene sloje in bloke v zagonu algoritma. V zadnjih dveh stolpcih je zapisana povprečna napaka razvrščanja in standardni odklon petih zagonov nad validacijsko in testno množico. Naj še poudarimo, da se je algoritem pri določenih kombinacijah parametrov ujel v lokalni optimum, zato teh rezultatov ne poročamo.

**Tabela 1: Rezultati algoritma DE nad podatkovno zbirko CIFAR10.**

Gradniki <sup>1</sup>	Vel. Pop.	Gen.	% učne množice	Dim	#Epoh	Vel. paketa	Nap. na val.	Nap. na test.
P,C,S,D,F,B	20	20	0,4	30	100	128	20,99 (3,3)	7,44 (2,3)
P,C,S,D,F,B	20	20	0,4	40	100	128	24,75 (2,4)	9,56 (1,4)
P,C,D,F,B	20	20	0,4	40	50	64	19,99 (3,6)	10,53 (2,7)
P,C,S,D,F,B	20	20	0,4	30	100	128	21,79 (5,5)	9,35 (3,3)
P,C,D,F,B	20	20	0,8	40	100	128	15,2 (4,3)	9,04 (2,6)
P,C,D,F,B	20	20	0,8	50	100	128	17,4 (6,1)	10,79 (1,1)
P,C,D,F,B	20	20	0,8	60	100	128	15,32 (2,1)	9,46 (0,8)
P,C,D,F,B	20	20	0,8	70	100	128	17,44 (3,8)	11,61 (2,2)
P,C,S,D,F,B	20	20	0,8	20	100	128	19,8 (3,9)	9,64 (1,6)
P,C,S,D,F,B	20	20	1	30	10	128	38,59(5,8)	14,63 (3,4)
<b>P,S,F,B</b>	<b>20</b>	<b>30</b>	<b>1</b>	<b>10</b>	<b>50</b>	<b>64</b>	<b>12,44 (1,6)</b>	<b>8,3 (0,4)</b>
P,S,F,B	20	40	1	10	10	128	25,9 (3,1)	10,57 (2,2)
P,S,F,B	20	40	1	10	10	64	20,93 (3,4)	9,29 (1,1)
P,S,F,B	20	40	1	15	10	128	27,34 (1,3)	12,62 (2,2)
P,S,F,B	20	40	1	15	30	128	16,25 (4,3)	10,25 (2,5)
P,S,F,B	20	40	1	15	30	256	24,27 (3,9)	11,94 (3,1)

<sup>1</sup> P-Združevalni sloj; C-Konvolucijski sloj; F-Polno povezan sloj; D-Izpustni sloj; S-Rezidualni blok; B-Gost blok

Tabela 2: Rezultati algoritma DE nad podatkovno zbirko AffectNet.

Gradniki	Vel. Pop.	Gen.	% učne množice	Dim	#Epoh	Vel. paketa	Nap. na val.	Nap. na test.
P,S,F,C,D	20	20	0,1	10	20	128	33,02 (2,4)	52,35 (3,3)
P,S,F,B	20	20	0,18	5	10	256	34,10 (3,8)	56,68 (4,1)
<b>P,S,F,B</b>	<b>20</b>	<b>20</b>	<b>0,19</b>	<b>10</b>	<b>10</b>	<b>128</b>	<b>29,18 (5,2)</b>	<b>50,55 (2,4)</b>

Iz Tabele 1 lahko razberemo, da dobimo najboljši rezultat, ko se uporabljamo pretežno enostavne sloje in celotno podatkovno zbirko v procesu učenja. Tukaj smo dosegli najnižjo napako pri 8,3 %. V tem primeru je imel algoritem tudi več časa za samo optimizacijo s 600 ovrednotenji kriterijske funkcije, kar pa sicer v splošnem ne zagotavlja boljših rezultatov. Ugotovili smo tudi, da velikost učne množice nima večjega vpliva na rezultate, saj so napake razvrščanja pri vseh variantah velikosti zelo podobne ( $< 2\%$ ), se pa računska zahtevnost zagotovo zmanjša. V literaturi najboljše metode iz družin evolucijskih algoritmov na podatkovni zbirki CIFAR10 dosegajo napake manjše tudi od 5 %, kjer pa z določenimi omejitvami algoritma manjšajo iskalni prostor. S tem seveda tudi vnašajo ekspertno znanje v reševanje problema, čemu smo mi skušali izogniti.

Na podatkovni zbirki AffectNet smo dosegli najnižjo napako razpoznave pri 50,55 %, na približno 19 % učnih podatkov. Pri večjih velikostih učne množice, se je algoritem velikokrat ujel v lokalni optimum, prav tako se je večala računska zahtevnost. V literaturi se za to zbirko rezultati najboljših metod gibljejo pod 40 % napake, kjer so pa izbrane arhitekture vedno ročno načrtovane (Zhang, 2017). Glede na velikost uporabljene učne množice, je naš rezultat na tej podatkovni zbirki sprejemljiv.

## 6 Zaključek

V članku smo predstavili algoritem diferencialne evolucije za iskanje optimalnih topologij globokih nevronske mreže na problemu razvrščanja slik. Algoritem smo preizkusili nad dvema podatkovnima zbirkama za razvrščanje slik. Rezultati so pokazali, da je možno z uporabo evolucijskega algoritma razviti oz. poiskati globoko

mrežo, ki dosega primerljive rezultate s stanjem tehnike brez uporabe poglobljenega ekspertnega znanja.

V prihodnosti bomo algoritem preizkusili še na drugih podatkovnih zbirkah, ga skušali izboljšati v smislu računske zahtevnosti, kjer bi lahko uporabili prenosno učenje. Med drugim bomo še poskušali optimizirati parametre učnega algoritma, ki ima velik vpliv na potek učenja.

### **Viri in literatura**

- Back, T. a. (1997). Handbook of evolutionary computation. Release.
- Brest, J. a. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE transactions on evolutionary computation*, 646-657.
- He, K. a. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, (str. 770-778).
- Hinton, A. K. (2009). Learning multiple layers of features from tiny images. Toronto: Toronto, ON, Canada. Pridobljeno iz <http://www.cs.toronto.edu/~kriz/cifar.html>
- Huang, G. a. (2017). Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, (str. 4700-4708).
- Kaelbling, L. P. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 237-285.
- LeCun, Y. a. (2015). Deep learning. *Nature*, 436-444.
- Mlakar, U. a. (2017). Multi-objective differential evolution for feature selection in facial expression recognition systems. *Expert Systems with Applications*, 129-137.
- Mollahosseini, A. a. (2019). AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild. *IEEE Transactions on Affective Computing*, 18-31.
- Sun, K. H. (2015). Deep Residual Learning for Image Recognition. *arXiv*.
- Weinberger, G. H. (2018). Densely Connected Convolutional Networks. *arXiv*.
- Zhang, Y. a. (2017). Very deep convolutional networks for end-to-end speech recognition. *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, (str. 4845-4849).
- Zisserman, K. S. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv*, 2015.

### **O avtorju**

Uroš Mlakar je leta 2019 doktoriral na Univerzi v Mariboru. Trenutno je izvoljen v naziv docent za področje računalništvo na Fakulteti za elektrotehniko in računalništvo v laboratoriju za sistemsko programsko opremo. Sodeloval je že pri pripravi več kot 30 znanstvenih revijalnih in konferenčnih člankov. Njegovi raziskovalni interesi vključujejo evolucijsko računanje, podatkovno rudarjenje in obdelavo slik. Med drugim je tudi recenzent pri mnogih priznanih mednarodnih revijah.





# ZAZNAVANJE PODVODNIH OBJEKTOV Z UPORABO GENERATIVNIH MODELOV

SANDRA RODRÍGUEZ DOMÍNGUEZ,<sup>1,2</sup> JANEZ PERŠ<sup>2</sup>

<sup>1</sup> Universidad Politécnica de Madrid, Madrid, Španija  
sandra.rodriguez.dominguez@alumnos.upm.es

<sup>2</sup> Univerza v Ljubljani, Fakulteta za elektrotehniko, Ljubljana, Slovenija  
sandra.rodriguez.dominguez@alumnos.upm.es, janez.pers@fe.uni-lj.si

V podvodnih okoljih predstavljajo spremenljiva osvetlitev, motnost vode in biološka raznolikost občutne ovire, zaradi katerih tradicionalne metode računalniškega vida ne delujejo dobro. Tudi učljive metode delujejo le, če uporabimo dovolj raznoliko zbirko podatkov, ki vsebuje vso pričakovano variabilnost podvodnega sveta. Zaradi narave samega podvodnega okolja pa je to lahko težavno, drago ali celo nemogoče, vsekakor pa zahteva veliko delovnih ur za označevanje objektov v učni množici. Ta problem smo naslovili z razvojem nove metodologije, ki na podlagi izjemno majhnega nabora sintetično generiranih slik objektov (10 v našem primeru) in večjega nabora ozadij brez objektov zanimanja (nekaj 100 slik) izdelava učno bazo poljubne velikosti, primerno za učenje globokih metod zaznavanja objektov, ki ne zahteva nobenega ročnega označevanja. V našem primeru smo metodologijo uporabili za detekcijo ribe *Acanthurus leucosternon*, katere podobo za učenje smo generirali s pomočjo orodij DALL-E in Stable Diffusion. Naučen model smo preizkusili na realnih posnetkih tropskih koralnih grebenov z algoritmom zaznavanja objektov YoloV8, pri čemer dosežemo  $F1=0.6$ , ne da bi algoritem videl eno samo realistično sliko objekta v času učenja.

DOI  
[https://doi.org/  
10.18690/um.feri.1.2024.4](https://doi.org/10.18690/um.feri.1.2024.4)

ISBN  
978-961-286-837-6

**Ključne besede:**  
globoke nevronske mreže,  
detekcija objektov,  
augmentacija,  
generativni modeli,  
podvodni posnetki



Univerzitetna založba  
Univerze v Mariboru

DOI  
[https://doi.org/  
10.18690/um.feri.1.2024.4](https://doi.org/10.18690/um.feri.1.2024.4)

ISBN  
978-961-286-837-6

**Keywords:**  
deep neural networks,  
object detection,  
augmentation,  
generative models,  
underwater images

# DETECTION OF UNDERWATER OBJECTS USING GENERATIVE MODELS

SANDRA RODRÍGUEZ DOMÍNGUEZ,<sup>1,2</sup> JANEZ PERSŠ<sup>2</sup>

<sup>1</sup> Universidad Politécnica de Madrid, Madrid, Spain  
[sandra.rodriguez.dominguez@alumnos.upm.es](mailto:sandra.rodriguez.dominguez@alumnos.upm.es)

<sup>2</sup> University of v Ljubljana, Faculty of Electrical Engineering, Ljubljana, Slovenia  
[sandra.rodriguez.dominguez@alumnos.upm.es](mailto:sandra.rodriguez.dominguez@alumnos.upm.es), [janez.pers@fe.uni-lj.si](mailto:janez.pers@fe.uni-lj.si)

In underwater environments, variable lighting, water turbidity, and biodiversity present significant obstacles that cause traditional computer vision methods to perform poorly. Even learning-based methods only work if one uses a sufficiently diverse dataset that contains all the expected variability of the underwater world. However, due to the nature of the underwater environment itself, this can be difficult, expensive or even impossible, and it certainly requires many man-hours to annotate objects in the training dataset. We addressed this problem by developing a new methodology that, based on an extremely small set of synthetically generated object images (10 in our case) and a larger, diverse set of backgrounds without objects of interest (a few 100 images), produces a training dataset of arbitrary size, suitable for training deep object detection methods, without the need for any manual annotation. In our case, we used the methodology to detect the fish *Acanthurus leucosternon*, whose training images were generated using DALL-E and Stable Diffusion tools. We tested the learned training on real images of tropical coral reefs with the YoloV8 object detection algorithm, achieving  $F1=0.6$  without the algorithm seeing a single realistic image of the object during learning.



## 1 Uvod

Podvodna okolja, bogata z biotsko raznovrstnostjo in kompleksnostjo, predstavljajo pomembne izzive za zaznavanje in identifikacijo objektov. Metode računalniškega vida pod vodo se uporabljajo v različne namene, vključno z raziskovanjem morskih virov (Han, 2020), podvodno navigacijo (Xie, 2018), podvodnim videonadzorom (Shkurti, 2012), ocenjevanjem populacij morskih vrst, preučevanjem ekosistemov, ohranjanjem morskih vrst, ribolovom, odkrivanjem neeksplodiranih ubojnih sredstev pod vodo in podvodno arheologijo.

Zaradi pomanjkanja osvetlitve, motnosti vode in raznolikosti oblik, barv ter velikosti morskih organizmov tudi moderni algoritmi globokega učenja odpovedo, razen če jim damo na voljo zelo raznoliko učno bazo slik, ki vsebuje vse možne variacije v podvodnem okolju. Po drugi strani pa je podvodno okolje v primerjavi s kopnim bistveno bolj nevarno tako za ljudi kot za robote (Aldhaheri, 2022). Zajem raznolikih podatkov, ki bi omogočali obsežno učenje globokih modelov je torej drag, dolgotrajen in nevaren, če pa gre za redko videne živalske ali rastlinske vrste pa sploh nemogoč.

V tem članku smo se problema lotili z uporabo generativnih modelov za izdelavo slik na podlagi tekstovnih opisov, kot jih poznamo iz orodij DALL-E (Ramesh, 2021) in Stable Diffusion. Osnovna ideja našega pristopa je v tem, da uporabimo majhno bazo segmentiranih slik pridobljenih z enim od teh orodij, potem pa izvedemo ekstremno augmentacijo: slike objektov transformiramo tako geometrijsko kot barvno, jih postavimo na naključno mesto v naključno izbrani sliki ozadja brez objektov, in jim dodamo šum. Posebnost našega pristopa je, da zaradi specifik podvodnega okolja izvedemo *prenos svetlobnega vira* iz slike ozadja na objekt, kar zahteva obsežno, vendar avtomatsko predobdelavo slik ozadja, s katero ocenimo parametre osvetlitve.

## 2 Sorodna dela

V razvoju algoritmov za detekcijo objektov so konvolucijske nevronske mreže (CNN) pokazale izjemno učinkovitost pri prepoznavanju in lokalizaciji objektov na slikah. Uporaba nevronske mreže se je razširila tudi na podvodne aplikacije (Villon, 2016; Yang, 2020) z uporabo mrež Yolo in Faster RCNN. Avtorji so se ukvarjali

tudi s prilagoditvijo algoritmov na podvodno okolje, vključno z uporabo generativnih modelov in tehnik obnavljanja slik (Wang, 2020; Chen, 2020; Liu, 2020).

### 3 Metodologija

Predlagana metoda sloni na naslednjih komponentah:

- Model globoke nevronske mreže za detekcijo ali segmentacijo objektov. V našem primeru smo uporabili mrežo YoloV8<sup>1</sup>.
- Večje število (npr. nekaj 100) slik ozadja, ki pa morajo vsebovati vso pričakovano raznolikost morskega okolja. Anotacije niso potrebne, z izjemo potrditve, da na slikah ni objektov, ki jih hočemo detektirati.
- Manjše število (npr. 10) slik objekta, pridobljenih z generativnimi slikovnimi modeli (v našem primeru DALL-E 2 in Stable Diffusion), ter segmentacijske maske za vsako sliko. Ker je število slik majhno, je potreben vložek dela za pridobitev mask zanemarljiv.

#### 3.1 Ocena svetlobnega vira slik ozadja

Ocena svetlobnega vira (angl. illuminant) je korak, pri katerem ocenimo, *kakšne transformacije* kanalov RGB slike so potrebne, da iz slike raznolikih, živih barv (slika  $A$ ), dobimo sliko, ki je barvno in svetlostno čimbolj podobna sliki ozadja ( $B$ ), ki jo imamo pred sabo. Z drugimi besedami, oceniti želimo parametre funkcije, ki nam iz slike  $A$  naredi sliko, ki je naj čimbolj barvno podobna  $B$ :

$$B \approx f(A; \theta_1, \theta_2, \dots, \theta_{12}) \quad (1)$$

Določitev parametrov  $\theta_1 \dots \theta_{12}$  je optimizacijski problem, pri katerem optimiziramo kriterijsko funkcijo podobnosti med sliko  $f(A; \theta_1, \theta_2, \dots, \theta_{12})$  in sliko  $B$ .

$$\theta_1, \theta_2, \dots, \theta_{12} = \underset{\theta_1, \theta_2, \dots, \theta_{12}}{\operatorname{argmin}} d(B, f(A; \theta_1, \theta_2, \dots, \theta_{12})) \quad (2)$$

---

<sup>1</sup> <https://docs.ultralytics.com/>

pri čemer so parametri  $\theta_1 \dots \theta_{12}$  parametri splošne linearne preslikave za vsakega od kanalov slike  $A$ :

$$B_{\text{red}} = \theta_3 + \frac{\theta_4 - \theta_3}{\theta_2 - \theta_1} (A_{\text{red}} - \theta_1) \quad B_{\text{green}} = \theta_5 + \frac{\theta_6 - \theta_5}{\theta_8 - \theta_7} (A_{\text{green}} - \theta_7) \quad (3)$$

$$B_{\text{blue}} = \theta_9 + \frac{\theta_{10} - \theta_9}{\theta_{12} - \theta_{11}} (A_{\text{blue}} - \theta_{11})$$

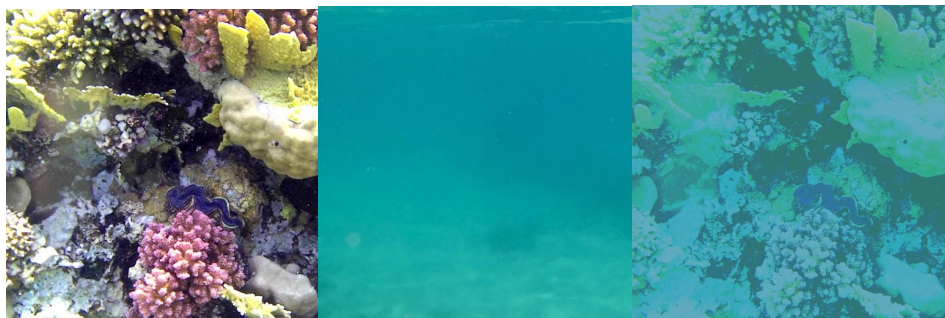
Razdalja  $d$  je kar razdalja Bhattacharaya med barvnima, 3D histogramoma slik  $A$  in  $B$ :

$$d_{\text{Bhattacharyya}}(h_A, h_B) = -\ln\left(\sum_i \sqrt{h_A(i) \cdot h_B(i)}\right) \quad (4)$$

Optimizacijski postopek izvedemo za vsako sliko ozadja posebej, pri čemer za sliko  $A$  uporabimo eno sliko ozadja, ki je lepih, živih barv. V trenutni implementaciji uporabimo optimizacijo z metodo simpleksov, ki ne potrebuje gradientov kriterijske funkcije (Lagarias, 1998). Postopek je časovno potraten, vendar ga je treba za vsako sliko ozadja izvesti le enkrat, shranjene parametre  $\theta_1 \dots \theta_{12}$  pa lahko seveda uporabimo večkrat

### 3.2 Prenos svetlobnega vira

Ocenjene parametre  $\theta_1, \theta_2, \dots, \theta_{12}$  shranimo in jih lahko kadarkoli uporabimo za to, da na poljubno sliko živih barv *prenesemo svetlobni vir* z uporabo enačb (2).



Slika 1: Učinek prenosa svetlobnega vira

Vir: lasten

Primeri slik  $A$ ,  $B$  ter transformirane slike  $f(A; \theta_1, \theta_2, \dots, \theta_{12})$  so prikazani na sliki 1. Od leve proti desni vidimo vhodno sliko živih barv, sliko ozadja, kjer je osvetlitev močno v prid modrozeleni barvi in na skrajni desni rezultat po prenosu osvetlitve iz druge slike na prvo sliko.

### 3.3 Generiranje slik objektov

Za generiranje slik objektov smo uporabili orodji DALL-E (verzija 2) in Stable Diffusion. Optimalne poizvedbe se razlikujejo od orodja do orodja. DALL-E 2 deluje dobro s prostim tekstom, Stable diffusion pa potrebuje ključne besede. Za naš primer ribe *Acanthurus leucosternon* dobimo dobre rezultate orodja DALL-E 2 s preprosto poizvedbo »*powder blue tang on black background*«, ki nam da štiri slike ribe na črnem ozadju. Slike je potrebno segmentirati od črnega ozadja, kar lahko izvedemo relativno enostavno s poljubnim risarskim programom, ki omogoča izrezovanje poligonov. Ugotovili smo, da se avtomatska segmentacija (ozadje je črno) ne splača.



Slika 2: Naravna slika ribe *Acanthurus leucosternon* in trije generirani primerki.

Vir: lasten

Slika 2 prikazuje primer naravne slike ribe in tri sintetično generirane primerke. Po vrsti od leve zgoraj desno in navzdol: naravna slika, rezultat DALL-E 2, rezultat Stable Diffusion, rezultat DALL-E 3 (Microsoft Bing image creator).

### 3.4 Generiranje učne množice

Vsako sliko iz učne baze generiramo po naslednjem postopku:

1. Naključno izberemo sliko ozadja in preberemo shranjene parametre osvetlitve  $\theta_1, \theta_2, \dots, \theta_{12}$
2. Naključno izberemo sliko objekta, preberemo tudi binarno segmentacijsko masko.
3. Sliko objekta zmanjšamo na velikost, ki smo jo določili naključno.
4. Na sliko objekta prenesemo osvetlitev ozadja po formulah (3)
5. Z enakomerno verjetnostjo izberemo kot rotacije v območju od  $-45^\circ$  do  $45^\circ$  in sliko objekta ter masko rotiramo.
6. Z verjetnostjo  $1/8$  sliko objekta prezrcalimo po višini in z verjetnostjo  $1/2$  po širini.
7. Naključno z verjetnostjo  $1/2$  izberemo ali bomo sliko »stisnili« po višini, enako po širini. Koeficient zmanjšanja širine ali višine izberemo naključno med  $0.2$  in  $0.8$ .
8. Naključno izberemo amplitudo naključnega belega šuma in ga prištejemo sliki objekta.
9. Z Gaussovimi filtrom zgladimo prej binarno sliko maske in z glajeno masko uteženo sliko objekta prilepimo na naključno izbrano lokacijo v sliki ozadja
10. Iz transformacij, ki smo jih izbrali v prejšnjih točkah generiramo sintetično anotacijo objekta na sliki.

## 4 Eksperimenti in rezultati

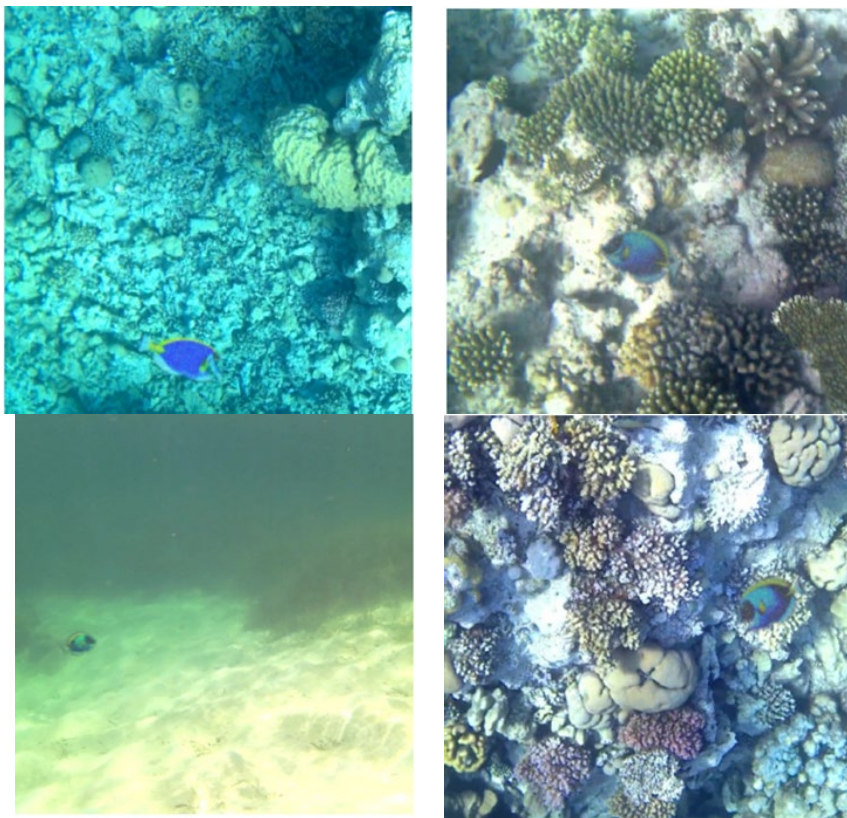
Za eksperiment smo uporabili zasebno podatkovno zbirko s posnetki tropskih morij iz treh različnih lokacij (Rdeče morje, Karibi, Indijski ocean).

#### 4.1 Testna množica

V zbirki smo označili odseke, kjer ni bilo vidne ribe *Acanthurus leucosternon* in tiste odseke, kjer je bila riba prisotna. Na ta način smo pridobili veliko množico slik ozadja ter primerno veliko testno množico, ki smo jo ročno anotirali.

#### 4.2 Učna množica

Po postopku iz poglavja 3.4 smo generirali učno množico 10.000 učnih slik s pripadajočimi anotacijami in jo razdelili na 6000 učnih slik in 4000 validacijskih slik. Edini označeni objekt na slikah je bila riba *Acanthurus leucosternon*. Ni nas motilo, če so bile na slikah ozadja tudi ribe drugih vrst. Nekaj učnih slik prikazuje slika 3.



Slika 3: Štiri od slik iz sintetično generirane učne baze

Vir: lasten

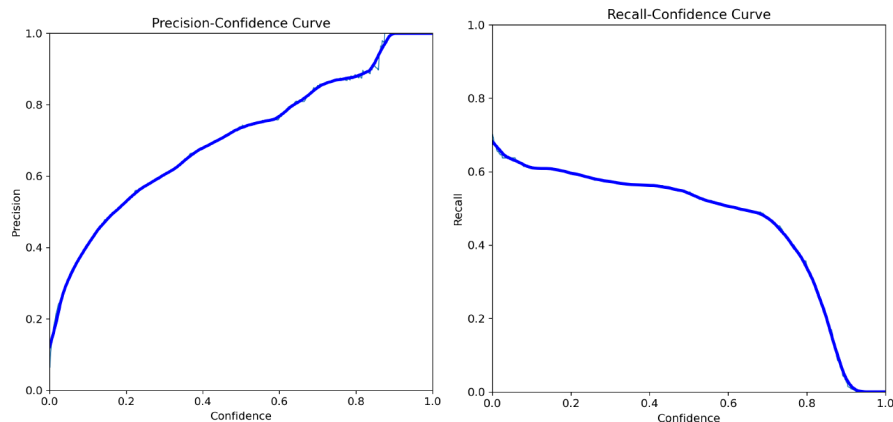


### 4.3 Učenje detektorja objektov

Uporabili smo že prednaučeno arhitekturo YoloV8x, ki smo jo na učni bazi samo doučili z enorazrednimi anotacijami (razred »riba«, ki je predstavljal ribo *Acanthurus leucosternon*). Uporabili smo privzete parametre učenja.

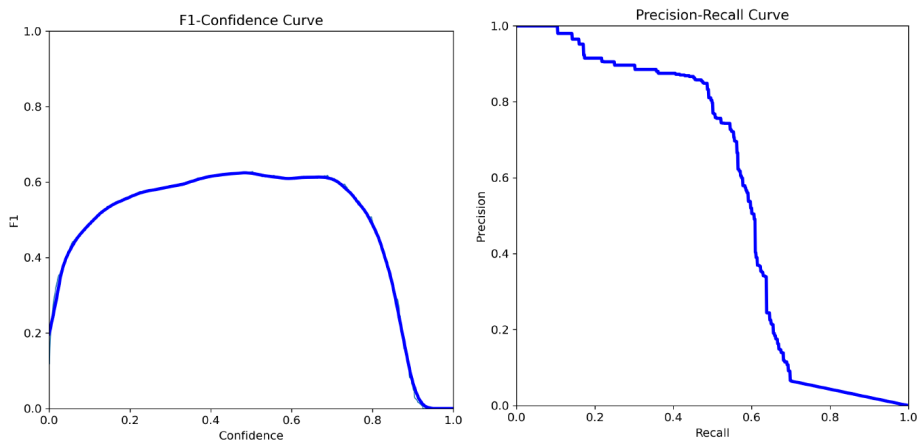
### 4.4 Rezultati

Uporabili smo knjižnico Ultralytics YoloV8 in vanjo vgrajena orodja za analizo uspešnosti detektorja objektov. Opazovali smo natančnost (ang. precision), priklic (ang. recall) in mero F1. Mere so odvisne od praga zaupanja (angl. confidence threshold), ki se giblje med 0 in 1, zato jih prikazujemo v obliki grafikonov (sliki 4 in 5). Vse mere so bile izračunane na podlagi predpisanega praga Jaccardovega indeksa 0.5 (ang. Intersection over Union, IoU). Tako učenje kot testiranje smo izvajali na slikah velikosti 512x512 slikovnih elementov. Na večini od 1464 testnih slik ni bilo rib, tako da je bilo v testni množici 345 rib.



Slika 4: Rezultati: natančnost (precision) in priklic (recall) na realni testni množici, prikazana glede na prag zaupanja.

Vir: lasten



Slika 5: Rezultati: mera F1 in krivulja natančnost/priklíc na realni testni množici

Vir: lasten

## 5 Zaključek

Pokazali smo, da so generativni slikovni modeli, tudi v obliki splošno in brezplačno dostopnih orodij, presenetljivo koristen pripomoček za generiranje velikih količin učnih podatkov. Naši rezultati na sicer omejenem problemu detekcije tropske ribe v okolju koralnih grebenov kažejo, da je mogoče doseči mero F1 ki presega 0.6 (ob priklícú in natančnosti 0.6), kar je izrazito dober rezultat, če upoštevamo, da algoritem med učenjem ni videl nobene realistične slike opazovane ribe, testiran pa je bil samo na realističnih slikah.

### Viri in literatura

- Han, M., Lyu, Z., Qiu, T., & Xu, M. (2020). A review on intelligence dehazing and color restoration for underwater images. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50, 1820-1832, doi: 10.1109/TSMC.2017.2788902.
- Xie, K., Pan, W., & Xu, S. (2018). An underwater image enhancement algorithm for environment recognition and robot navigation. *Robotics*, 7, 14. doi: doi.org/10.3390/robotics7010014
- Shkurti, F., Xu, A., Meghjani, M., Higuera, J. C. G., Girdhar, Y., Giguère, P., et al. (2012). Multi-domain monitoring of marine environments using a heterogeneous robot team. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*, 1747-1753. doi: 10.1109/IROS.2012.6385685.
- Aldhaferi, S., De Masi, G., Pairet, È., & Ardón, P. (2022). Underwater Robot Manipulation: Advances, Challenges and Prospective Ventures. *OCEANS 2022 – Chennai* doi:10.1109/OCEANSChennai45887.2022.9775489

- Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., & Sutskever, I. (2021). Zero-Shot Text-to-Image Generation. arXiv:2102.12092 [cs.CV]. doi: 10.48550/arXiv.2102.12092
- Yang, H., Liu, P., Hu, Y., & Fu, J. (2020). Underwater object recognition based on yolov3. ICUS 2021, doi: 10.1109/ICUS52573.2021.9641489
- Wang, Z., Liu, C., Wang, S., Tang, T., Tao, Y., Yang, C., Li, H., Liu, X., & Fan, X. (2020). UDD: An underwater open-sea farm object detection dataset for underwater robot picking. 2020.
- Chen, X., Lu, Y., Wu, Z., Yu, J., & Wen, L. (2020). Reveal of Domain Effect: How Visual Restoration Contributes to Object Detection in Aquatic Scenes. arXiv:2003.01913 [cs.CV]. doi: 10.48550/arXiv.2003.01913
- Liu, H., Song, P., & Ding, R. (2020). Towards domain generalization in underwater object detection. 2020. ICIP 2020. doi: 10.1109/ICIP40778.2020.9191364.
- Lagarias, J. C., Reeds, J. A., Wright, M. H., & Wright, P. E. (1998). Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. *SIAM Journal of Optimization*, 9(1), 112-147.



# INTRODUCTORY COMPUTER VISION TEACHING MATERIALS FOR VET EDUCATION

VASJA LEV KIRN,<sup>1</sup> ŽIGA EMERŠIČ,<sup>1</sup> GREGOR HRASTNIK,<sup>1</sup>  
NATAŠA MEH PEER,<sup>2</sup> PETER PEER<sup>1</sup>

<sup>1</sup> University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia

vk3843@student.uni-lj.si, ziga.emersic@fri.uni-lj.si, gregor.hrastnik@scv.si,  
peter.peer@fri.uni-lj.si

<sup>2</sup> School center Velenje, Velenje, Slovenia  
natasa.mehpeer@scv.si

Rapidly advancing development of artificial intelligence technologies, including deep learning techniques in the field of computer vision, has encouraged the need for early education about artificial intelligence in schools. This paper briefly describes the development of a computer vision curriculum, part of the AIM@VET (Artificial Intelligence Modules for Vocational Education and Training) EU project, targeting VET high-school students. The introductory materials presented in this paper are structured in three main teaching units (TUs), covering object detection and image segmentation. Each TU consists of eight tasks and a final assignment, totaling approximately 10 hours of classroom work. The course material, prepared in both traditional learning materials and in Python notebooks, combines theoretical concepts with practical coding exercises, with separate teacher and student versions. Materials rely on interactive tools and open-source libraries such as OpenCV, facilitating hands-on learning and immediate application of computer vision concepts.

DOI  
[https://doi.org/  
10.18690/um.feri.1.2024.5](https://doi.org/10.18690/um.feri.1.2024.5)

ISBN  
978-961-286-837-6

**Keywords:**  
computer vision,  
Python notebook,  
object detection,  
teaching material,  
segmentation



University of Maribor Press

DOI  
[https://doi.org/  
10.18690/um.feri.1.2024.5](https://doi.org/10.18690/um.feri.1.2024.5)

ISBN  
978-961-286-837-6

# UVODNI UČNI MATERIALI RAČUNALNIŠKEGA VIDA ZA STROKOVNO SREDNJEŠOLSKO IZOBRAŽEVANJE

VASJA LEV KIRN,<sup>1</sup> ŽIGA EMERŠIČ,<sup>1</sup> GREGOR HRASTNIK,<sup>1</sup>  
NATAŠA MEH PEER,<sup>2</sup> PETER PEER<sup>1</sup>

<sup>1</sup> Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana, Slovenija  
vk3843@student.uni-lj.si, ziga.emersic@fri.uni-lj.si, gregor.hrastnik@scv.si,  
peter.peer@fri.uni-lj.si

<sup>2</sup> Šolski center Velenje, Velenje, Slovenija  
natasa.mehpeer@scv.si

**Ključne besede:**  
računalniški vid,  
Python notebook,  
detekcija objektov,  
učna gradiva,  
segmentacija

Hitro napredujoč razvoj tehnologij umetne inteligence, vključno s tehnikami globokega učenja na področju računalniškega vida, je spodbudil potrebo po zgodnjem izobraževanju o umetni inteligenci v šolah. Ta članek na kratko opisuje razvoj učnega načrta za računalniški vid, ki je del projekta EU AIM@VET (Artificial Intelligence Modules for Vocational Education and Training), namenjen srednješolcem poklicnega izobraževanja in usposabljanja. Uvodna gradiva, predstavljena v tem dokumentu, so strukturirana v treh glavnih učnih enotah, ki pokrivajo zaznavo objektov in segmentacijo slik. Vsak TU je sestavljen iz osmih nalog in zaključne naloge v skupnem okvirnem obsegu 10 ur izvajanja v učilnici. Gradivo je pripravljeno v tradicionalnih učnih gradivih in v zvezkih Python, združuje teoretične koncepte s praktičnimi vajami kodiranja z ločenimi različicami za učitelje in učence. Gradivo temelji na interaktivnih orodjih in odprtokodnih knjižnicah, kot je OpenCV, ki olajšajo praktično učenje in takojšnjo uporabo konceptov računalniškega vida.

## 1 Introduction

The field of computer vision (CV) has rapidly evolved from its inception in the mid-20th century, undergoing significant advancements in the late 20th and early 21st centuries. Initially relying on handcrafted techniques, Computer Vision (CV) has witnessed a transformative leap with the introduction of deep learning methods, leading to fully automated and efficient approaches (IBM, 2023). CV now plays a crucial role in various real-world applications, from aiding medical image interpretation to enhancing surveillance systems and contributing to autonomous vehicles' success (Jahne, 2000, Janai, 2020). As digitalization becomes more prevalent, applications like consumer-grade facial detection are already integrated into daily life.

Recognizing the growing importance of CV education, this paper addresses the development and implementation of a CV course within the AIM@VET project, aimed primarily at 16 to 20-year-old high-school students. Development was also a part of the diploma thesis (Kirm Lev, 2024). The CV course comprises of three modules, focusing on fundamental CV concepts through practical tasks, as already presented in (Emeršič, 2023). Here, we specifically present three teaching units (TUs) within the second module, covering Object Detection and Segmentation.

Key objectives of this work include responding to the need for CV education, targeting a specific age group, and developing TUs with tasks and assignments for effective learning. The coursework is presented digitally through Python notebooks, offering an interactive learning experience.

AIM@VET project, funded by the European Union aims to prepare the next generation for the increasing relevance of AI in the social and economic aspects of the future. It consists of the following groups and contents: (I) Spanish group: University of Coruña and Integrated Professional Training Center Rodolfo Ucha Piñeiro, focused on the robotics module, (II) Portuguese group: University of Minho and Caldas das Taipas Secondary School, focused on the ambient intelligence module, (III) Slovenian group: University of Ljubljana (UL) and School center Velenje (SCV), focused on the computer vision module.

## 1.1 Instructional Approaches to Computer Vision & Development

Traditional teaching methods in various fields, including computer science, often rely on textbooks for theoretical knowledge. However, practical understanding of programming and implementation, especially in areas like object detection and image segmentation, is essential in computer vision (CV). This course adopts an interactive approach using Python notebooks, consisting of eight tasks and a two-part assignment in each teaching unit (TU) to cover specific CV topics.

Introducing CV to high-school students is crucial due to its rapid growth, and providing hands-on experiences with visible outcomes, such as detecting edges in an image, can stimulate interest. The use of Python notebooks offers a visual and practical learning experience, enhancing student engagement and fostering further exploration in computer science (AIM@VET, 2023).

While CV is traditionally a higher-level topic in university studies, presenting it effectively can serve as an entry point for high-school students in their programming journey. Real-world applications like facial recognition make CV intriguing and relatable. The AIM@VET project aligns with modern teaching tools, utilizing Python Notebooks, like other online programming courses and initiatives like the AI+ project (AI Plus, 2023). AIM@VET builds on the insights gained from AI+, focusing on specialized AI education for secondary school students. The project continues the project-based learning approach with a hands-on emphasis and real-world tasks (Emeršič, 2023, AIM@VET, 2023).

## 2 Created Materials

This work encompasses the development of three Teaching Units (TUs), each consisting of 8 tasks and a final two-part assignment, totaling 30 hours of course content. Tasks, designed for both teachers and students as Python notebooks, progressively build on each other within a TU and logically connect between TUs, ensuring a comprehensive understanding of CV by the course's end.

Each task has two versions – one for students without exercise answers and less pre-written code, and one for teachers containing solutions. Corresponding course documents for each TU, initially in MS Word and later converted to LaTeX, provide a standardized structure for easy access. These documents outline the course plan,



background knowledge, and practical exercises, aiding teachers in guiding students through the material.

The first TU focuses on object detection fundamentals, covering basic image manipulation and introducing hand-crafted algorithms, laying the groundwork for further TUs, as shown in the excerpt from the materials, shown in Figure 1 and Figure 2. Figure one show a section from the PDF materials provided to the students, and Figure 2 shows an example of a Python code that students need to improve and run. Teaching unit 1 consist of two activities:

- Activity 1 Preparation for object detection: Learn to make simple object detection using templates, Bounding Box Techniques and Perform face detection, Image processing necessary for detection, Feature Extraction, and Image Pyramids.
- Activity 2 Object detection: Object Detection using Sliding Windows, Object Detection using Selective Search, Real-Life Examples of Object Detection, Evaluating Object Detection Algorithms, Assignment: Creating Simple Object Detection Algorithm



Figure 3: Bounding boxes around detected faces.

### Required Knowledge for Task 3: Image Preprocessing for Object Detection: Image processing necessary for detection

As we have already learned in the past lessons of this course, Object Detection is a complicated task. We want to create an automatic detection of objects of interest within an image and also localize them, by drawing a bounding box around them. These two things need to be implemented well and work successfully in order for visual processing to be effective in real-world scenarios. There are different techniques used in object detection, but before we input our image into our algorithm, we need to pre-process it in order to receive the best result.

Pre-processing an image for object detection means adjusting its properties in such a way that the objects we wish to detect and localize will be more easily detected and that the noise in our image (which results in false positives of detection) will be reduced. Techniques for pre-processing images include:

- Resizing the image
- Cropping a certain region of the image
- Applying color conversion on the image

#### Applying color conversion to an image

The first technique of pre-processing that we will explore is applying color conversion to an image. This means changing the relative spectrum of colors used to represent our image, which is then displayed as different values of colored pixels on our screen. In the example below we load the example image "Lenna"

**Figure 1: Excerpt from the materials covering the required knowledge needed for the successful completion of the tasks in the TU.**

Source: own

TU2 explores deep learning-based object detection, emphasizing modern approaches. It guides students through configuring and training the YOLOv5 deep learning model. TU2 consist of three activities, which contain:

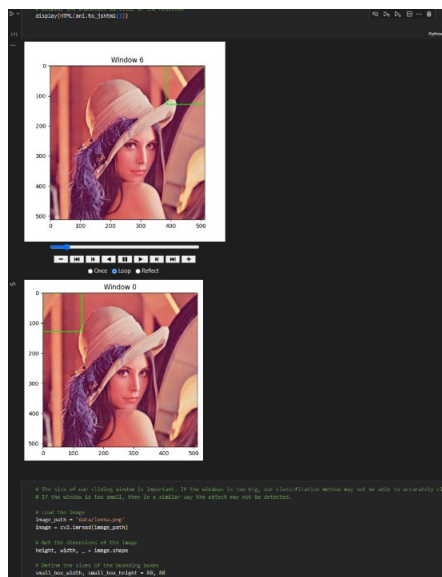
- Activity 1 Introduction into deep learning: Introduction to Deep Learning-Based Object Detection, Popular Object Detection Models, Image processing necessary for detection, Setting up Environment and Dataset Preparation.
- Activity 2 Training detection model: Configuring the configuration files of YOLOv5 and understanding its architecture, Training the model using the custom dataset, Exploring the essential hyperparameters and optimizing them for improved model performance.
- Activity 3 Evaluating and improving detection: Testing the trained model on test image set and evaluating its performance, Assignment: Deep Learning Object Detection Algorithm.

TU3 shifts to image segmentation, covering fundamentals, techniques, and notable algorithms. It consists of three activities:

- Activity 1 Introduction into image segmentation: Introduction to image segmentation, Overview of image segmentation techniques, Image pre-processing for image segmentation.
- Activity 2 Implementing image segmentation algorithms: Image segmentation using thresholding, Image segmentation using region growing, Clustering for image segmentation, Image segmentation using watershed.
- Activity 3 Evaluating and improving segmentation: Evaluating image segmentation algorithms, Assignment: Image segmentation algorithms.

### **3 Evaluation and Adaptation of Course Materials**

Teaching materials underwent adjustments during the creation of TUs 1, 2, and 3 for the second learning module. Feedback from a focus group and project coordination prompted structural changes.



**Figure 2: Teaching material example**

Source: own

Initially, tasks in each Teaching Unit (TU) were outlined, adhering to the project's requirements. Ongoing collaboration with SCV allows for feedback and adjustments, presenting a snapshot of the current course version in this paper.

In response to initial feedback from SCV, the course structure was revised, materials simplified, TU shortened. The last two tasks were replaced by a comprehensive two-part assignment, fostering a more cohesive learning experience. Students sequentially complete tasks, building foundational knowledge, and then apply it independently in the final assignment, enhancing practical understanding.

The first adjustment involved creating two versions of each task: one for teachers with complete examples and solutions, and one for students with omitted solutions. This adaptation fosters critical thinking in students while aiding teachers in guiding them through tasks.

The second adjustment separated theoretical content from Python notebooks, relocating it to a standardized course document. This change facilitated a more collaborative classroom environment, allowing teachers to effectively convey theoretical knowledge.

The third adjustment reduced tasks in each TU from 10 to 8, with a final assignment covering two lessons. This shift aimed to focus on a smaller set of topics, ensuring a more cohesive learning experience.

Feedback from a focus group suggested simplifying content for high school students, leading to adjustments in task difficulty and the incorporation of simple exercises. Further feedback influenced changes to TU2, emphasizing step-by-step training of the YOLOv5 model, making the content more engaging and practical.

After materials were used at SCV, additional corrections were made to the materials, forming the final versions of the TUs. However, the ongoing development incorporates received feedback, aiming to create a more accessible and engaging course for both students and teachers.

## 4 Conclusion

This paper presents computer vision teaching materials, designed for students that are new to programming, and it introduces them fundamentals of AI techniques in computer vision. The course employs Python notebooks, emphasizing interactivity and engagement, allowing students to learn by doing and teachers to guide effectively.

Aligned with AIM@VET's goal of early AI education, this course targets young students, fostering interest through interactive learning. The adaptable structure facilitates future additions and revisions. Ongoing collaboration with SCV students ensures active adjustments, making the course responsive to student needs.

The development is ongoing until 2025, combining meaningful challenges and positive student feedback, promising a valuable resource for AI education. The course's expansion, coupled with AIM@VET's broader efforts, aims to equip students with essential AI knowledge, supporting their future studies and work.

## References

IBM. "What is computer vision? " Pridobljeno iz: <https://www.ibm.com/topics/computer-vision> (2023).

- Jahne, B., & Haußecker, H. "Computer vision and applications". Academic Press, 2000, pp. 609–660. (2000)
- Janai, J, et al. "Computer vision for autonomous vehicles: Problems, datasets and state of the art". *Foundations and Trends® in Computer Graphics and Vision* 12.1–3 (2020), pp. 1–53. (2020)
- Kirm Lev, V. "Teaching Units for a Computer Vision Course", Diplomsko delo, Pridobljeno iz: <https://repositorij.uni-lj.si/IzpisGradiva.php?id=153473>. (2024).
- Emeršič, Ž., Hrastnik, G, Meh, N., Peer, P. "Adapting VET Education to Labor Market Needs with Focus on Artificial Intelligence and Computer Vision", ROSUS 2023 (2023)
- AIM@VET. "Guidelines". Pridobljeno iz: <https://aim4vet.udc.es/backgroud/> (2023).
- AI Plus. "AI Plus". Pridobljeno iz: <https://aipudc.azurewebsites.net/> (2023).



# APLIKACIJA RAČUNALNIŠKEGA VIDA ZA REŠEVANJE RUBIKOVE KOCKE V REALNEM ČASU

JAN ŠUKLJE, PETER PEER, BOJAN KLEMENC

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana, Slovenija  
sukljejan@gmail.com, peter.peer@fri.uni-lj.si, bojan.klemenc@fri.uni-lj.si

Rubikova kocka je ena najbolj znanih igráč, tako za mlade kot tudi za starejše ljudi. Za začetnike je precej velik zalogaj, zato smo se odločili narediti aplikacijo, ki jo novincem pomaga rešiti. Primarno namen same aplikacije ni učenje reševanja Rubikove kocke, saj se uporabnik z njeno uporabo ne uči, temveč samo dela gibe, ki jih aplikacija pokaže. Njen cilj je pomagati, saj če se nekomu zaplete pri reševanju kocke, si lahko z aplikacijo pomaga in kocko vrne v prvotno rešeno stanje. Kar izstopa pri tej aplikaciji v primerjavi z drugimi orodji za reševanje Rubikove kocke, je uporaba kamere kot ključnega elementa. Namesto omejevanja uporabnika na 2D ali 3D grafični prikaz kocke, ki je lahko precej zamudna, ta aplikacija izkoristi kamero, ki omogoča uporabo toka videa kot podlago uporabniškega vmesnika. Uporabnik v realnem času vidi navodila za reševanje Rubikove kocke, kar olajša in pospeši celoten proces reševanja.

DOI

[https://doi.org/  
10.18690/um.feri.1.2024.6](https://doi.org/10.18690/um.feri.1.2024.6)

ISBN

978-961-286-837-6

## Ključne besede:

računalniški vid,  
Rubikova kocka,  
obogatena resničnost,  
Python,  
OpenCV

## Prispevek temelji na:

Šuklje, J. (2022). *Aplikacija računalniškega vida za reševanje Rubikove kocke v realnem času*: diplomsko delo, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana.



Univerzitetna založba  
Univerze v Mariboru

DOI  
[https://doi.org/  
10.18690/um.feri.1.2024.6](https://doi.org/10.18690/um.feri.1.2024.6)

ISBN  
978-961-286-837-6

**Keywords:**

computer vision,  
Rubik's cube,  
augmented reality,  
Python,  
OpenCV

**The proceedings is based on:**

Šuklje, J. (2022). *Aplikacija računalniškega vida za reševanje Rubikove kocke v realnem času*. bachelor's thesis, University of Ljubljana, Faculty of Computer and Information Science. Ljubljana.

# COMPUTER VISION APPLICATION FOR SOLVING A RUBIK'S CUBE IN REAL-TIME

JAN ŠUKLJE, PETER PEER, BOJAN KLEMENC

<sup>1</sup> University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia  
sukljejan@gmail.com, peter.peer@fri.uni-lj.si, bojan.klemenc@fri.uni-lj.si

The Rubik's cube is one of the most well-known toys, both for young and older people. For beginners, it can be quite challenging, so we decided to create an application that helps them solve it. The main purpose of the application is not to teach solving the Rubik's cube, as users don't learn from it but only perform the moves shown by the application. Its goal is to assist if the user gets stuck while solving the cube – the user can use the application to return it to the original solved state. What stands out in this application compared to other tools for solving the Rubik's Cube is the use of the camera as a key element. Instead of limiting the user to a 2D or 3D graphical representation of the cube, which can be quite time-consuming, this application utilizes the camera, allowing video stream to be the user interface. The user receives instructions for solving the Rubik's cube in real time, which makes the entire process of solving this challenging game easier and faster.





## 1 Uvod

Rubikova kocka že dolgo časa navdušuje kot igrača, ki testira naš um in zabava ljudi vseh generacij. Medtem ko nekateri uživajo v zapletenih obratih in vrtenju ploskev kocke, se drugi borijo s prepoznavanjem njenih kompleksnosti. Zavedajoč se potrebe po pomoči pri reševanju te ikonične uganke, smo razvili aplikacijo v Pythonu, ki izkorišča tehnike računalniškega vida in knjižnico *OpenCV*, katere prvo javno različico je podjetje *Intel* dalo v uporabo pod licenco BSD (sedaj *Apache*) leta 2000 (Kaehler & Bradski, 2016). Aplikacija uporabnikom zagotovi intuitivno in interaktivno rešitev tako, da uporablja kamero za vodenje skozi postopek reševanja.

Namen prispevka je predstaviti razvoj in funkcionalnost inovativnega reševalca Rubikove kocke. Predstavili bomo problemsko domeno, postopek analize vizualnih podatkov v realnem času, ki vključuje zajemanje videa s kamero, zaznavanje kocke in njenih barv ter izračun rešitev Rubikove kocke, kakor tudi implementacijo uporabniškega vmesnika s pomočjo prikaza nad videom. Osredotočili se bomo na tehnične vidike aplikacije, kot so orodja, ki smo jih uporabili za obdelavo vizualnih podatkov, pridobljenih s kamero. Poleg tega bomo predstavili različne funkcije, vgrajene v aplikacijo, ki uporabnikom pomagajo pri procesu reševanja.

## 2 Pregled področja

### 2.1 Rubikova kocka

Rubikova kocka je priljubljena 3D uganke, sestavljena iz manjših kock, ki so različne barve. Lahko jo manipuliramo s premikanjem njenih ploskev v različnih smereh. Ko so vsi kvadrati na vseh posameznih ploskvah enake barve, pomeni da je kocka rešena.

Za opisovanje premikov Rubikove kocke smo uporabili najpogosteje uporabljeno notacijo (Ferenc, 2023), v kateri se posameznim stranem Rubikove kocke priredi specifična črka.

## 2.1 Sorodna dela

Priljubljenost Rubikove kocke je privedla do tega, da jo ljudje rešujejo na različne načine. Medtem ko so nekateri stremeli k hitrim rešitvam, so drugi iskali izzive v nekonvencionalnih metodah, ki presegajo standardne pristope. Eden od pristopov, ki ga je uporabilo podjetje OpenAI, je uporaba robotske roke za reševanje kocke (Akkaya, et al., 2019). S tem so demonstrirali prenosljivost računalniškega modela, treniranega samo v simulaciji, na fizično manipulacijo. Spet drug pristop, ki temelji na globokem spodbujevalnem učenju in iskanju, prikaže algoritem, ki lahko reši kocko v različnih okoljih brez zanašanja na človeško znanje (Agostinelli, McAleer, Shmakov, & Baldi, 2019). Delo, ki je najbolj podobno naši implementaciji, tudi uporablja kamero za prikaz uporabniškega vmesnika, namenjeno pa je le prikazovanju napredka reševanja Rubikove kocke in ne samemu reševanju (Ajisaka, et al., 2020). Podoben študentski projekt je tudi diplomska naloga Simona Gerholda (Gerhold, 2014), ki se osredotoča na razvoj interaktivne in vizualizacijo Rubikove kocke.

## 3 Zajem videa

Za zajemanje videa smo uporabili mobilni telefon, na katerem je bila naložena aplikacija IP Webcam<sup>1</sup>. Nato smo dodali URL toka videa kot parameter naši aplikaciji za reševanje Rubikove kocke. Za zajem slik iz toka videa smo uporabili knjižnico OpenCV in implementirali razred VideoCaptureAsync, ki deluje asinhrono, kar pripomore k hitrosti delovanja. Ta razred omogoča učinkovito branje slik iz videoposnetka, kar izboljša obdelavo videoposnetkov.

Tolerance za oddaljenost in orientacijo kocke so velike, zato ni potrebno natančno nastavljanje stojala za mobilni telefon.

## 4 Zaznavanje Rubikove kocke

Za zaznavanje Rubikove kocke smo uporabili knjižnico OpenCV in Python. S pomočjo knjižnice OpenCV smo obdelali vhodno sliko, izluščili ključne informacije in se osredotočili na prepoznavo devetih posameznih kvadratov na kocki. Zatem

---

<sup>1</sup> [https://play.google.com/store/apps/details?id=com.pas.webcam&hl=en\\_US](https://play.google.com/store/apps/details?id=com.pas.webcam&hl=en_US)

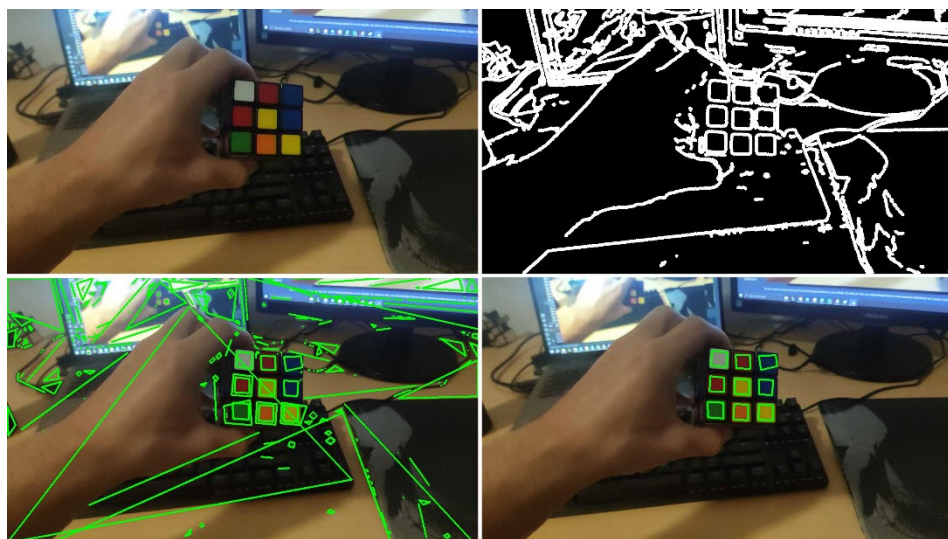
smo poskušali ustvariti zunanji pravokotnik, ki popolnoma obdaja te kvadrate in pove, ali je bila kocka uspešno zaznana. Če je kocka uspešno zaznana, na naslednji sliki video toka uporabimo masko za pospešeno zaznavanje.

#### 4.1 Zaznavanje barvnih kvadratov

Za zanesljivo identifikacijo kvadratov Rubikove kocke je uporabljen niz postopkov, ki vključujejo Gaussovo zameglitev, Cannyjev detektor robov, iskanje kontur, optimizacijo kontur s funkcijo `approxPolyDP` ter filtriranje kontur glede na število točk, površino, konveksnost in kotne lastnosti.

Gaussova zameglitev se uporablja za zmanjšanje šuma in pridobivanje bolj konsistentnih podob kvadratov. Cannyjev detektor robov omogoča natančno identifikacijo robov v sliki. Iskanje kontur se izvaja s funkcijo `cv2.findContours()`, nato pa se uporablja funkcija `approxPolyDP` za optimizacijo kontur z manjšim številom mejnih točk.

Filtriranje kontur se izvaja glede na število točk, površino, konveksnost in kotne lastnosti. Prekrivanje med kvadrati se odstrani, zagotavljajoč, da ostanejo samo relevantni kvadrati za nadaljnjo analizo.



Slika 1: Prikaz postopka zaznavanja barvnih kvadratov

Vir: lasten

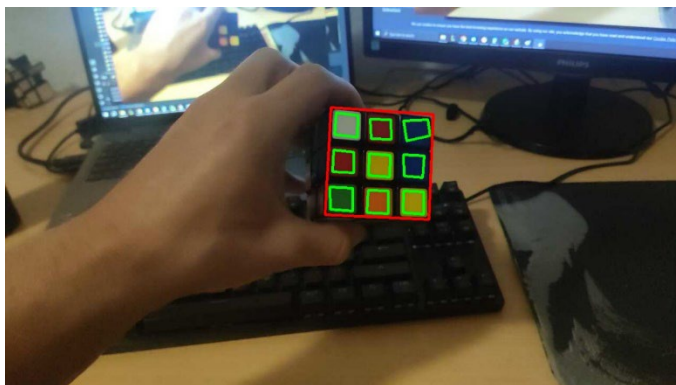
Slika 1 prikazuje začetno sliko, identifikacijo robov z Cannyjevim detektorjem, nato zaznane robove po optimizaciji kontur in na koncu vseh devet zaznanih kvadratov po končni filtraciji. Ta celoten postopek omogoča zanesljivo zaznavanje barvnih kvadratov Rubikove kocke, bolj podroben opis lahko najdete v (Šuklje, 2023).

## 4.2 Nepravilno število kvadratov

Po izvedbi filtriranja se lahko pojavi situacija, kjer imamo lahko več ali manj kot devet kvadratov, ki so ključni za prepoznavo Rubikove kocke. Če ostane manj kot devet takšnih kvadratov, postopek prekinemo in čakamo na naslednjo sliko. To se običajno zgodi, ko na sliki ni Rubikove kocke ali je prikazan le del nje. Nasprotno, če imamo več kot devet kvadratov, najprej identificiramo srednji kvadrat, obdan s tremi ali več sosednjimi kvadrati na vseh štirih straneh. Nato znotraj slike poiščemo osem najbližjih kvadratov temu srednjemu kvadratu, da pridemo do končnih devetih kvadratov. To se zgodi v primerih, ko je ozadje Rubikove kocke polno drugih kvadratnih objektov, ki so tudi zaznani kot kvadrati.

## 4.3 Indikator celotne Rubikove kocke

Da bi se prepričali, ali vsi kvadrati tvorijo eno skupno kocko, uporabimo dodaten postopek, kjer jih poskušamo očrtati s še enim kvadratom, primer je razviden na sliki 2. Ogljšča tega novega kvadrata določimo glede na položaj manjših štirih kvadratov, ki se nahajajo v ogliščih zaznane kocke.



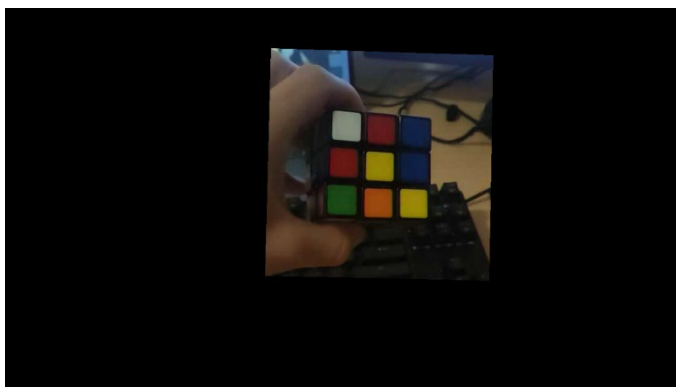
**Slika 2: Očrtanje devetih kvadratov**

Vir: lasten

Če ima kvadrat z rdečim robom dovolj podobne dolžine vseh štirih stranic ter notranje kote, ki so dovolj podobni pravim kotom, smo lahko skoraj prepričani, da smo zaznali Rubikovo kocko.

#### 4.4 Maskiranje nadaljnjih zaznavanj

Če uspešno zaznamo Rubikovo kocko na eni sliki, lahko pričakujemo, da se bo na naslednji sliki nahajala na približno istem mestu. To omogoča optimizacijo obdelave slik z izbrisom ali zakrivanjem delov slike, ki so dovolj oddaljeni od zaznane kocke. Uporaba strategije maskiranja zmanjša obremenitev računalniških virov, še posebej, če se uporabi natančen Cannyjev detektor robov. Med zajemanjem slik je pomembno upoštevati hitrost premikanja kocke in razdaljo le-te od kamere. Primer maskiranja je prikazan na sliki 3.



Slika 3: Maskiranje izvorne slike

Vir: lasten

## 5 Zaznavanje barv

Zaznavanje barv je ključno pri reševanju Rubikove kocke, zato je pomembno, da to opravimo natančno in zanesljivo v različnih svetlobnih pogojih. Analiza vsakega kvadrata se izvaja s preverbo, ki ugotovi prevladujočo barvo. V knjižnici OpenCV je uporabljen barvni prostor BGR, vendar za olajšano analizo barv uporabljamo barvni prostor HSV, v katerega prevladujočo barvo tudi pretvorimo. Barvni prostor HSV omogoča ločevanje odtenkov, nasičenosti in svetlosti, kar olajša obdelavo in primerjavo barv med seboj.

S poskušanjem smo prepoznavali, katerim barvam pripadajo vrednosti v modelu HSV. S postopnim testiranjem smo izboljševali pretvorbo in zagotovili, da ni bila preveč občutljiva na različne svetlobne pogoje. Trenutno delovanje sistema za prepoznavanje barv poteka zelo dobro, izjema so zelo slabi svetlobni pogoji.

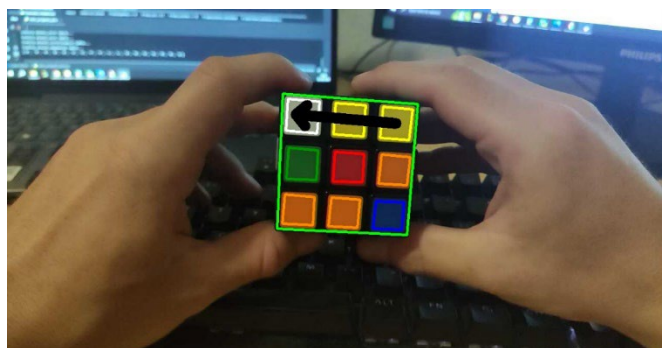
## 6 Algoritem za reševanje Rubikove kocke

Algoritem Kociemba z dvema fazama (Kociemba, 1992) se izkaže kot izjemno zmogljiva in učinkovita metoda za reševanje zapletene uganke, kot je Rubikova kocka. To inovativno metodo je razvil Herbert Kociemba v zgodnjih 2000-ih letih. Metoda se je hitro uveljavila kot ena izmed najbolj naprednih tehnik za premagovanje izziva sestavljanja te priljubljene ugankarske igrače.

Ena izmed najbolj opaznih prednosti tega algoritma je sposobnost, da kocko reši v manj potezah v primerjavi z mnogimi drugimi metodami. Omogoča hitro reševanje celo najzahtevnejših konfiguracij kocke v razumnem časovnem okviru.

## 7 Uporabniški vmesnik

Uporabnik na začetku sledi navodilom, ki mu povedo, kako in v katerem vrstnem redu mora prvotno zaznati vse strani Rubikove kocke. S tem pridobimo vse kvadrate in njihove barve, ki jih potrebujemo za iskanje gibov, ki vodijo do rešene kocke. Ko je to urejeno, se uporabniku na videu začnejo prikazovati navodila, kako Rubikovo kocko premikati, da bo prišel do končnega rešenega stanja.



Slika 4: Prikaz puščice za potreben premik

Vir: lasten

Ker smo želeli narediti kar se da enostaven uporabniški vmesnik, smo se odločili, da bo prikaz potrebnih gibov prikazan samo s puščicami na sami kocki v realnem času (slika 4). Tako si lahko uporabnik zlahka predstavlja, kaj mora narediti.

Po vsakem gibu, ki ga naredi uporabnik, se preveri, ali je bil gib pravilno narejen. Če je bil napačen, se celotno zaznavanje prekine, uporabnik pa mora zopet začeti znova in še enkrat pokazati vse strani Rubikove kocke.

Za boljšo predstavo o delovanju je primer uporabe objavljen na omrežju Youtube (Šuklje, 2023)(Šuklje, 2023).

## **8 Rezultati**

Število potez, potrebnih za rešitev Rubikove kocke, je ključna mera, ki je odvisna od učinkovitosti algoritma in od tega, kako dobro je kocka premešana. Algoritem išče rešitve v največ dvajsetih potezah, kar se je izkazalo kot optimalno, saj se rešitev običajno najde hitro. Hitrost in odzivnost aplikacije sta zadovoljivi, brez zaznavnih kjučnih zakasnitev pri prikazu slike. Teste smo izvedli na namiznem računalniku, ki ima procesor AMD Ryzen 7 2700 in na prenosniku z procesorjem AMD Ryzen 5 5500U, v obeh primerih je aplikacija delovala brez problemov, razlike v odzivnosti med njima nismo opazili. Primerjava s hitrostjo prikaza surovega videa kaže minimalno razliko v zakasnitvi, kar kaže na dobro optimizacijo aplikacije. Ker aplikacija omogoča samodejno zaznavanje barv in prikaz navodil na videu kocke, zmanjšuje tveganje za napake, saj uporabniku ni potrebno ročno vnašati vseh 54 barv kvadratov, med reševanjem same kocke pa napačno izveden gib pomeni prekinitev delovanja.

Aplikacijo so preizkusile štiri odrasle osebe in tudi dva otroka stara sedem let, rezultati pa so pokazali, da je aplikacija uporabna tako za starejše kot tudi za mlajše uporabnike. Vsi odrasli so uspešno rešili kocko v nekaj minutah, medtem ko so otroci potrebovali malo dlje, saj so si sprva težko predstavljali katero stran kocke gledajo, ampak ko so to ugotovili, jim je kocko uspelo rešiti.

## 9 Zaključek

Najzahtevnejši del pri razvoju aplikacije je bilo razvijanje samega sistema za zaznavanje Rubikove kocke. Opravili smo številne iteracije, pri čemer smo vsakič uporabili drugačen pristop k prepoznavanju kocke. Stalnica pri tem je bila uporaba knjižnice OpenCV. Uporaba aplikacije je preprosta in hitrejša v primerjavi z drugimi podobnimi rešitvami, kar smo potrdili z uspešnim preizkusom med družinskimi člani in prijatelji, ki niso znali sestaviti Rubikove kocke. S pomočjo aplikacije so vsi uspeli rešiti Rubikovo kocko v le nekaj minutah.

Med razvojem in testiranjem aplikacije smo zaznali nekaj točk, kjer bi lahko izvedli izboljšave. Ključna izboljšava bi bila povečanje natančnosti pri prepoznavanju Rubikove kocke na kamerah nižje kakovosti. Testiranje z vgrajeno kamero v prenosniku je razkrilo, da se detekcija poslabša v slabših razmerah zaradi omejenega kontrasta slike in večjega šuma. Dodatna izboljšava v prihodnosti bi omogočila uporabnikom, da Rubikovo kocko zaznajo in identificirajo v poljubnem vrstnem redu med prvo fazo zajemanja vseh strani kocke.

### Literatura

- Agostinelli, F., McAleer, S., Shmakov, A., & Baldi, P. (2019). Solving the Rubik's cube with deep reinforcement learning and search. *Nature Machine Intelligence*, 1, 356–363.
- Ajisaka, S., Hara, S., Matsuchi, M., Luo, S., Yoshida, S., Xie, H., & Miyata, K. (2020). Learning rubik's cube through user operation history. *Nicograph International (NicoInt)*, str. 43–46.
- Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., and others. (2019). Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*.
- Ferenc, D. (2023). *Rubik's cube notation*. Pridobljeno 1.2.2024: <https://ruwix.com/the-rubiks-cube/notation/>
- Gerhold, S. (2014). Razvoj interaktivne Rubikove kocke: diplomsko delo, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko. Ljubljana.
- Kaehler, A., & Bradski, G. (2016). *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. O'Reilly Media, Inc.
- Kociemba, H. (1992). *Rubik's Cube Twophase eSolver*. Pridobljeno 1.2.2024: <https://github.com/hkociemba/RubiksCube-TwophaseSolver>
- Šuklje, J. (2023). Aplikacija računalniškega vida za reševanje Rubikove kocke v realnem času: diplomsko delo, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko. Ljubljana.
- Šuklje, J. (2023). *Computer vision application for solving a Rubik's cube in real-time*. 1.2.2024: <https://www.youtube.com/watch?v=qAvpx65JOVI>



# SPROTNA ANALIZA SLIK VOZIL Z METODAMI GLOBOKEGA UČENJA V OGRODJU FLUTTER

ALEKSANDR SHISHKOV,<sup>1,3</sup> STEVANČE NIKOLOSKI<sup>2,3</sup>

<sup>1</sup> Univerza v Mariboru Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor, Slovenija

aleksandr.shishkov@student.um.si

<sup>2</sup> Univerza v Novem mestu Fakulteta za ekonomijo in informatiko, Novo mesto, Slovenija

stevance.nikoloski@uni-nm.si

<sup>3</sup> Result d.o.o., Ljubljana, Slovenia

aleksandr.shishkov@student.um.si, stevance.nikoloski@uni-nm.si

V članku raziskujemo integracijo modela MobileNetV3 v ogrodju Flutter, osredotočajoč se na napredno klasifikacijo slik avtomobilov. Preučujemo večplasten pristop, ki vključuje uporabo raznolikih podatkovnih zbirk, fino prilagajanje modela ter njegovo brezhibno implementacijo v mobilno aplikacijo. S poudarkom na izboljšanju uporabniške izkušnje smo ustvarili tri specializirane modele z visoko stopnjo natančnosti (97%), ki prepoznajo ustrezne slike, klasificirajo tip slike (vozilo, armaturna plošča ali dokument) ter določajo stran avtomobila (spredaj, levo, desno, zadaj). Rezultati kažejo izjemno hitrost in odzivnost aplikacije, pri čemer MobileNetV3 zagotavlja natančno klasifikacijo v le 60 ms, kar prispeva k izjemni učinkovitosti celotnega sistema.

DOI

[https://doi.org/  
10.18690/um.feri.1.2024.7](https://doi.org/10.18690/um.feri.1.2024.7)

ISBN

978-961-286-837-6

**Ključne besede:**

MobileNetV3,  
Flutter,  
klasifikacija slik,  
fino prilagajanje,  
globoko učenje



Univerzitetna založba  
Univerze v Mariboru

DOI  
[https://doi.org/  
10.18690/um.feri.1.2024.7](https://doi.org/10.18690/um.feri.1.2024.7)

ISBN  
978-961-286-837-6

# INSTANTANEOUS VEHICLE IMAGE ANALYSIS WITH DEEP LEARNING METHODS WITH FLUTTER FRAMEWORK

ALEKSANDR SHISHKOV,<sup>1,3</sup> STEVANČE NIKOLOSKI<sup>2,3</sup>

<sup>1</sup> University of Maribor Faculty of Electrical Engineering and Computer Science,  
Maribor, Slovenia  
[aleksandr.shishkov@result.si](mailto:aleksandr.shishkov@result.si)

<sup>2</sup> University of Novo mesto Faculty of Economics and Informatics, Novo mesto,  
Slovenia  
[stevance.nikoloski@uni-nm.si](mailto:stevance.nikoloski@uni-nm.si)

<sup>3</sup> Result d.o.o., Ljubljana, Slovenia  
[aleksandr.shishkov@result.si](mailto:aleksandr.shishkov@result.si), [stevance.nikoloski@uni-nm.si](mailto:stevance.nikoloski@uni-nm.si)

**Keywords:**  
MobileNetV3,  
Flutter,  
image classification,  
fine-tuning,  
deep learning

The article explores the integration of the MobileNetV3 model into the Flutter framework, focusing on advanced image classification of vehicles. We examine a multi-faceted approach that includes the use of diverse datasets, fine-tuning the model, and seamlessly implementing it into a mobile application. With an emphasis on enhancing the user experience, we have created three specialized models with a high level of accuracy (97%), capable of recognizing appropriate images, classifying the type of image (vehicle, dashboard, or document), and determining the side of the vehicle (front, left, right, rear). The results demonstrate exceptional speed and responsiveness of the application, with MobileNetV3 providing precise classification in just 60 ms, contributing to the overall efficiency of the system.



## 1 Uvod

Umetna inteligenca in globoko učenje sta revolucionirala našo sposobnost optimizacije in avtomatizacije raznovrstnih nalog. Eden od zanimivih izzivov na tem področju je klasifikacija slik. Naloga strojnega učenja, ki je temu namenjena se imenuje večciljna klasifikacija (angl. Multi-label classification), pri čemer za vsak nov klasifikacijski objekt, model vrača vrednosti med 0 in 1 za vsako posamezno kategorijo, odvisno od izračunane verjetnosti, da je v novem klasifikacijskem objektu prepoznana posamezna kategorija (de Carvalho & Freitas, 2009). V tem članku se bomo posvetili uporabi večciljnega klasifikacijskega modela MobileNetV3 (Qian et al., 2021). Osredotočili se bomo na njegovo uporabo pri validaciji ustreznih slik, klasifikaciji v treh različnih kategorijah, kot so armaturna plošča, vozilo ali dokument ter klasifikaciji strani avtomobilov, pri čemer bomo vodili skozi postopek integracije naučenega modela v aplikacijo, zgrajeno na ogrodju Flutter (Singh & Bhadani, 2020).

MobileNetV3 predstavlja vrhunski klasifikacijski model naučen s konvolucijskimi nevronskimi mrežami (CNN). Medtem ko obstaja več algoritmov strojnega učenja za klasifikacijo slik, kot so logistično regresijo, odločitvena drevesa, naključni gozdovi ipd., smo se odločili za uporabo CNN, saj raziskava v področju strojnega učenja modele CNN izloči kot najbolj zanesljivi, saj pričakujemo visoko stopnjo natančnosti modela za klasifikacijo slik (Wang et al., 2021).

Podatki predstavljajo temeljno komponento sodobnega sveta. V našem članku bomo podrobno opisali postopek pridobivanja podatkov, nato pa razkrili, kako smo jih skrbno obdelali, upoštevajoč specifične potrebe našega projekta. Pri tem smo se srečali z raznolikimi izzivi, ki smo jih sistematično naslovili in premagali, kar je ključno za zagotavljanje kakovostnih rezultatov našega modela. Slednje smo dosegli, ko smo skupaj s slikami, ki smo jih pridobili preko naše aplikacije vključili še slike iz zunajnih relevantnih virov.

Pomembno je poudariti, da uspešno obvladovanje podatkovne faze prispeva k celotni zanesljivosti in natančnosti modela ter posledično izboljšuje uporabniško izkušnjo. Z našim pristopom k pridobivanju in obdelavi podatkov želimo bralcem ponuditi vpogled v kompleksnost ter pomembnost pravilnega ravnanja s podatki v procesu razvoja in izpopolnjevanja modelov umetne inteligence.

MobileNetV3, ki deluje kot naš osnovni model, gre skozi proces fino prilagajanja (fine-tuning), da bi izboljšal svojo učinkovitost pri specifičnem opravilu klasifikacije strani avtomobilov. Ko je model enkrat prilagojen, sledi ključen korak brezhibne integracije v aplikacijo Flutter (Singh, 2020). Poglobili se bomo v zapletenosti te integracije, poudarili korake, ki smo jih sprejeli, da bi model brez težav vključili v arhitekturo aplikacije.

## 2 Metodologija

Kot osnovni programski jezik smo izbrali Python, ki velja za klasično izbiro pri globokem in strojnem učenju (angl. deep learning in machine learning). Pomagala nam je knjižnica TensorFlow (Pang et al., 2020), s katero smo naučili in testirali model, obdelali podatki ter konvertirali naučeno model v format TensorFlow Lite, optimiziran za uporabo na mobilnih napravah.

### 2.1 Priprava podatkov

Na začetku našega projekta smo imeli v mislih razvoj modela za napoved strani vozila. Zato smo se odločili uporabiti nabor podatkov CompCars (Yang et al., 2015), ki predstavlja obsežno kolekcijo slik avtomobilov in njihovih delov. Naša prvotna zahteva je bila, da morajo biti slike avtomobilov v naboru podatkov označene glede na stran vozila, kot so spredaj, levo, desno ali zadaj. Na žalost smo ugotovili, da ta zbirka ne vsebuje potrebnih označb. Z namenom, da bi ohranili učinkovitost projekta, smo se soočili z izzivom, saj bi ročno označevanje kar 136 tisoč slik predstavljalo nerealno obsežno opravilo. Zato smo sprejeli odločitev, da poskusimo drug nabor podatkov, ki bi ustrezal našim zahtevam.

Posledično smo se odločili tudi klasificirati kategorijo slik, ki jih lahko uporabnik posname, in sicer ali gre za armaturno ploščo, vozilo ali dokument. Za ta namen smo uporabili slike iz produkcijskega strežnika, saj projekt že deluje približno dve leti in vsebuje zadostno količino slik. Prednost te izbiri je v tem, da posebne lastnosti in format teh slik bo povečata točnost modela, ker so podatki za treniranje in za nadaljnjo uporabo podobni. Te surove slike so za naš primer zelo ustrezne, vendar sploh niso označene in zato smo se lotili dodatnega označevanja.

Za ta namen smo razvili aplikacijo z grafičnim vmesnikom (GUI), ki omogoča ročno klasifikacijo slik. Pri razvoju smo mislili na povečanje hitrosti ročne klasifikacije, zato smo implementirali klasifikacijo z uporabo hitrih tipk (angl. hotkeys) in dinamično izbiro kategorij in map slik. Ta rešitev je poenostavila ročni postopek klasifikacije slik in zagotovila ustvarjanje prilagojenega podatkovnega nabora iz surovih slik.

Takrat smo že začeli delati z MobileNetV3, in na srečo ta model ni zahteval velikega nabora podatkov pri finem prilagajanju in prav tako ni bilo vpliva na točnosti. Zato smo označili manj slik in trenirali model na majhnem naboru podatkov, preostali slike pa smo označili že z uporabo modela.

Prav tako smo za implementacijo funkcionalnosti validacije morali pripraviti dve kategoriji v naboru podatkov: ustrezne in neustrezne slike. Kot ustrezne slike smo uporabili slike iz prejšnjega nabora podatkov, medtem ko smo za neustrezne slike uporabili nabor podatkov iz spletne strani Kaggle (Sinha, 2024), ki vsebuje več odprtodostopnih slik poljubnih kategorij.

Smo ustvarili tri vrste podatkovnih zbirk za učenje modela za klasifikacijo strani avtomobila, model za klasifikacijo tipa slik in validacijski model.

Za vsako kategorijo imamo med 3000 in 5000 slik, pri čemer pri treniranju uravnotežimo količino. Kljub velikemu številu slik lahko pri finem prilagajanju dejansko uporabimo manjše število, saj razlika v točnosti ni tako velika. Smo dosegli natančnost več kot 90% ali 95% pri 100 slikah na kategorijo.









### **2.3 Naloga strojnega učenja**

Naloga modela strojnega učenja je vključevala predvsem večciljno klasifikacijo vrste slike, tj. dodelitev slike v določeno kategorijo iz nabora različnih kategorij. Za izpolnitev te naloge smo razvili arhitekturo modela, ki temelji na MobileNetV3 in je bila fino prilagojena specifičnim zahtevam našega projekta. Enoten model je bil oblikovan za izvajanje natančnih napovedi, pri čemer smo se osredotočili na prilagajanje končnih slojev glede na značilnosti naših podatkov. S tem smo ustvarili model, ki je sposoben učinkovito klasificirati slike v različne kategorije, kar je ključno za uspešno izvajanje naše aplikacije.

### 2.3.2 Večciljna klasifikacija

Večciljna klasifikacija (angl. multi-label classification) je algoritem strojnega učenja, ki se uporablja za napoved več kot ene kategorije oz. oznake za dano vhodno instanco (de Carvalho & Freitas, 2009).

V primeru večciljne klasifikacije imamo opravka s posebnim izzivom, imenovanim binarna relevantnost (Szymański & Kajdanowicz, 2017; Binary Relevance, 2024). Ta koncept se nanaša na napovedovanje verjetnost prisotnosti posamezne kategorije na vhodni sliki, pri čemer model vrača vrednosti med 0 in 1. Kot prag napovedane vrednosti se pogosto nastavlja 0.5, da lahko določimo, ali slika spada v posamezno kategorijo ali ne. Ta pristop omogoča osredotočanje na posamezne kategorije neodvisno ena od druge, kar je ključno pri nalogah, kjer želimo ločeno obravnavati več kategorij za dano instanco. Slika 1 prikazuje nabor slik, opisanih z vizualnimi značilnostmi, izluščenimi iz slik, kot so število črt različnih tipov (ukrivljene, horizontalne, diagonalne itd.), in ciljne atribute, ki označujejo prisotnost (T) ali odsotnost (⊥) objektov različnih tipov.

Image	Descriptive attributes				Target attributes						
				...	<i>traffic light</i>	<i>car</i>	<i>truck</i>	<i>building</i>	<i>traffic sign</i>	<i>bridge</i>	<i>tree</i>
	32	3	54	...	T	T	T	T	T	⊥	T
	55	43	1	...	T	T	T	T	T	T	T
	23	4	5	...	T	T	⊥	T	T	⊥	T
	23	4	5	...	T	T	T	T	⊥	⊥	T
	21	2	4	...	T	T	⊥	⊥	T	⊥	T
...	...	...	...	...	...	...	...	...	...	...	...

Slika 1: Ilustrativni primer podatkovne zbirke z večciljno klasifikacijo.

Vir: lasten

## **2.4 Globoko učenje**

Globoko učenje (angl. deep learning) predstavlja področje strojnega učenja, ki se osredotoča na uporabo kompleksnih umetnih nevronske mreže z več plastmi, znanih kot globoke nevronske mreže (LeCun et al., 2015). Glavni cilj globokega učenja je avtomatsko pridobivanje in predstavljanje kompleksnih vzorcev v podatkih. Ključna značilnost tega pristopa je njegova sposobnost samodejnega učenja hierarhičnih in abstraktnih značilnosti iz podatkov na različnih ravneh kompleksnosti.

V primerjavi s tradicionalnimi metodami strojnega učenja, kjer so modeli običajno plitvi, globoko učenje omogoča učenje predstavitev podatkov na več ravneh. To omogoča boljše zajemanje in razumevanje hierarhične strukture podatkov. Temeljni gradniki globokih nevronske mreže vključujejo vhodni sloj, skrite sloje in izhodni sloj. Skriti sloji igrajo ključno vlogo, saj omogočajo modelu, da samodejno izpelje značilnosti iz podatkov na bolj abstraktnih ravneh, kar vodi do učinkovitega reprezentiranja kompleksnih vzorcev. Ključna pomankljivost modele globokega učenja je njihova interpretivnost. Zato so modele globokega učenja imenujejo tudi »black-box« modele. Vendar so zelo uporabni in aplikativni zaradi njihove velike natančnosti.

### **2.4.1 Konvolucijska nevronska mreža**

Konvolucijska nevronska mreža (angl. Convolutional Neural Network ali CNN) je vrsta nevronske mreže zasnovana obdelave podatkov, ki imajo strukturo mreže, kot so slike (O'Shea & Nash, 2015). CNN se pogosto uporablja v nalogah računalniškega vida, kot so klasifikacija slik, detekcija objektov in segmentacija slik. Glavna značilnost CNN je uporaba konvolucijskih plasti, ki omogočajo učinkovito prepoznavanje lokalnih vzorcev v vhodnih podatkih.

Pri tradicionalnih algoritmih strojnega učenja za večciljno klasifikacijo se pomembne značilke izbirajo v predpripravi podatkov in postanejo vhodni podatki v učnem procesu. Po drugi strani pa je ključna prednost konvolucijske nevronske mreže v tem, da namesto ročnega izbiranja značilk iz vhodnih podatkov v fazi predpriprave podatkov, model samodejno izpelje pomembne značilke s pomočjo konvolucijskih plasti, kar poteka skozi proces, imenovan inženiring značilk (ang. feature engineering), ki je vključen v samem učnem procesu. Konvolucijske plasti

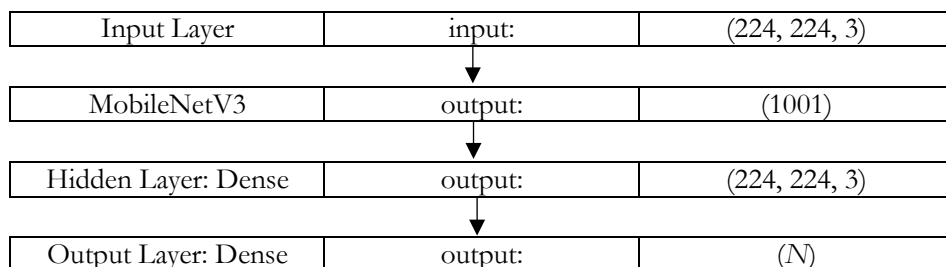
uporabljajo jedra (filtri), ki se pomikajo po vhodnih podatkih in izvajajo konvolucijo, kar omogoča prepoznavanje lokalnih vzorcev, kot so robovi, teksture in oblike (Slika 2) (Siebel & Rice, 2019).

Poleg konvolucijskih plasti CNN običajno vključuje tudi plasti združevanja (pooling), ki zmanjšajo dimenzionalnost podatkov in povečajo prejetje globalnih značilnosti. CNN ima običajno tudi polno povezane sloje, ki se uporabljajo za končno klasifikacijo ali regresijo.

Zaradi njihove uspešnosti pri obdelavi slik in drugih mrežnih podatkov so CNN postale ključna tehnologija na področju računalniškega vida.

Pri prvih poskusih smo probali zgraditi modelo z lastno napisano arhitekturo, podobno kot v dokumentaciji TensorFlow-a (TensorFlow, 2024), ki je vsebovala več slojev in je bila prazna. Po treniranju in prilagojenju parametrov modela nismo dosegli točnosti več kot 65%. Zato smo prenehali z uporabo tega modela in začeli iskati že modele naučene na velikem datasetu.

V našem primeru model za napoved več kategorij sledi naslednji arhitekturi: na vходу sprejme podatke sliki z ločljivostjo 224 na 224 pikslov, slika je v RGB formatu. Uporablja se MobileNetV3 kot osnovni model, ki je že predhodno treniran in ima lastne uteži. Sledi skrit sloj Dense, ki ga nadalje treniramo. Končno, kot izhod služi zadnji sloj Dense, ki ima  $N$  enot, kjer je  $N$  število kategorij v naboru podatkov, kot je prikazano na Slika 2.



Slika 2: Arhitektura modela za večciljno klasifikacijo.

Vir: lasten.



## **2.4.2 MobileNetV3**

MobileNetV3 je model nevronske mreže, zasnovan za klasifikacijo slik in druge naloge v področju računalniškega vida (Qian et al., 2021). Gre za tretjo iteracijo MobileNet modelov, ki so znani po svoji sposobnosti doseganja visoke natančnosti klasifikacije ob hkratni ohranitvi nizke računske zahtevnosti. Glavni cilj MobileNetV3 je izboljšati učinkovitost in natančnost predhodnih modelov.

Model uporablja arhitekturo nevronske mreže, ki vključuje konvolucijske plasti, aktivacijske funkcije, plasti normalizacije po serijah (angl. Batch Normalization), in globalno povprečenje. Značilnost MobileNetV3 je tudi uporaba blokov Inverted Residuals, ki omogočajo boljše zajemanje informacij in povečanje zmogljivosti.

MobileNetV3 je bil posebej zasnovan za mobilne naprave in aplikacije z omejenimi računskimi viri, kjer je pomembno ohraniti ravnovesje med natančnostjo in hitrostjo delovanja. Zato je priljubljen v aplikacijah, ki zahtevajo hitro obdelavo slik ob minimalni porabi energije.

## **2.5 Uporabniški primeri**

Sedaj, ko imamo arhitekturo modela na osnovi MobileNetV3, lahko te modele treniramo z različnimi podatki. V našem primeru želimo ustvariti tri različne modele, vsakega posebej treniranega na ločenih datasetih, z namenom uporabe v specifičnih situacijah.

### **2.5.1 Validacijski model**

Validacijski model se osredotoča na napovedovanje ustreznosti slike in vrača vrednosti med 0 in 1. Pri odločanju o ustreznosti uporabljamo prag, v našem primeru je to 0.5, kar predstavlja koncept binarne relevantnosti. Ta model je posebej prilagojen, kadar želimo napovedati le eno kategorijo.

### **2.5.2 Model za klasifikacijo tipa slike**

V tem primeru imamo tri kategorije, in sicer armaturno ploščo, vozilo in dokumente. Model vrne verjetnost za vsako kategorijo, kar predstavlja klasičen primer večciljni klasifikacije.

### 2.5.3 Model za klasifikacijo strani avta

V tej situaciji imamo štiri kategorije, ki predstavljajo različne strani avtomobila, spredaj, levo, desno, zadaj. Model prav tako vrne verjetnost za vsako kategorijo, kar predstavlja še en primer večciljne klasifikacije.

## 2.6 Treniranje modelov

V Jupyter notebook smo implementirali fino prilagajanje, podporo spremembam parametrov, spremembo datasetov, prikazovanje rezultatov treniranja, testiranje, shranjevanje modela in konvertacijo v format TensorFlow Lite.

Učili smo modele na lokalnih računalnikih, opremljenih z NVIDIA GPUjev in podporo za CUDA. S pravilnimi nastavitvami TensorFlow zazna GPU in trenira nanj. V tem primeru je celoten proces zelo hiter, in en model se lahko izuči v 5 do 10 minutah.

Naš pristop omogoča tudi prilagajanje parametrov, kjer lahko izbiramo med velikostjo osnovnega modela (small ali large) ter vrednostjo za depth multiplier (0.75 ali 1). Vsi naši modeli so bili trenirani z uporabo modela small in depth multiplier 0.75, saj nismo zaznali bistvenih razlik med drugimi kombinacijami. Nato sledi izbira dataseta in razdelitve, kjer smo določili 70% za učni nabor in 30% za validacijo. Število epoch smo nastavili na 10. Uporabili smo Adam optimizer in Binary Crossentropy loss za model enojne klasifikacije (validacijski model), medtem ko smo za večciljno klasifikacijo uporabili Sparse Categorical Crossentropy.

## 2.7 Postavitev modela v mobilno aplikacijo

Naša aplikacija je napisana na platformi Flutter, okviru za izdelavo večplatformskih aplikacij. Flutter ima repozitorij paketov, kje razvijalci lahko dodelijo svojo kodo. Za delo z Tensorflow modeli smo se odločili uporabiti paket tfLite\_flutter (Flutter, 2024), ki prinaša številne prednosti. Ena izmed njih je hitrost, saj paket uporablja kompilirano C kodo, kar prispeva k izjemni učinkovitosti pri obdelavi slikovnih napovedi.

Z uporabo tega paketa in osnov objektno orientiranega programiranja smo ustvarili vmesnik, kjer lahko enostavno dodamo model in ga uporabimo v aplikaciji. Ta pristop nam zelo olajša delo, saj imamo tri različne modele z različnimi kategorijami.

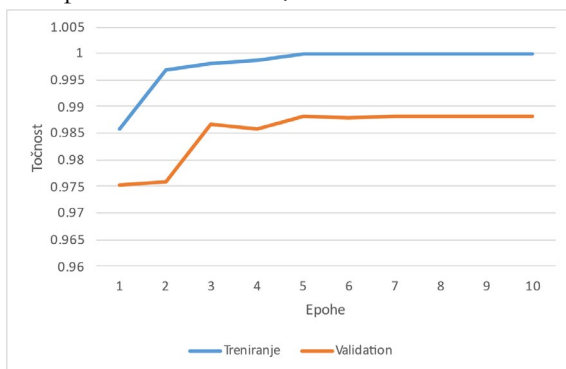
Pri izdelavi vmesnika smo pazili na pravi prenos slikovnih podatkov v model, saj mora biti enak kot pri učenju. Zato smo poudarili testiranje pretvorbe sliki v vhodne podatke. Preverjali smo, ali je pretvorjena slika enaka sliki pri učenju, če so bile originalno podane.

Model uporabimo pri fotografiranju sliki avtomobila. Ko uporabnik klikne na gumb, aplikacija posname sliko, jo spremeni v pravo velikost in konvertira v vektor pikslov. Ta vektor je vhod za naš model, ki nam vrne rezultat inference.

Ta model nam omogoča izboljšanje uporabniškega vmesnika v dveh situacijah. Prvič, validira sliko in preveri, ali je ustrezna; če ne, obvesti uporabnika. Če je slika ustrezna, jo klasificira, doda oznako in prikaže v uporabniškem vmesniku.

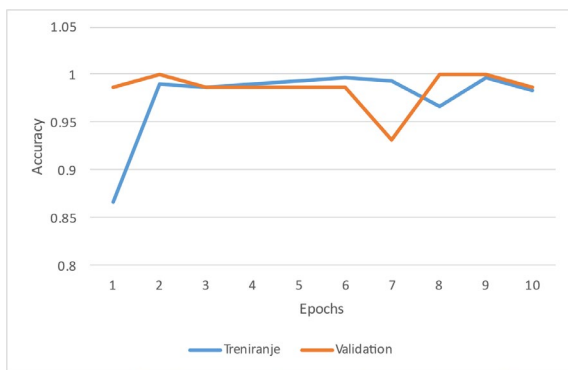
### 3 Rezultati

Po uspešnem učenju modela na osnovi MobileNetV3 in izvedbi fino prilaganja za klasifikacijo slik smo dosegli opazen napredek v več pogledih. Rezultati učenja treh modelov so ustrezno prikazani na Slika 4, Slika 5 in Slika 6.



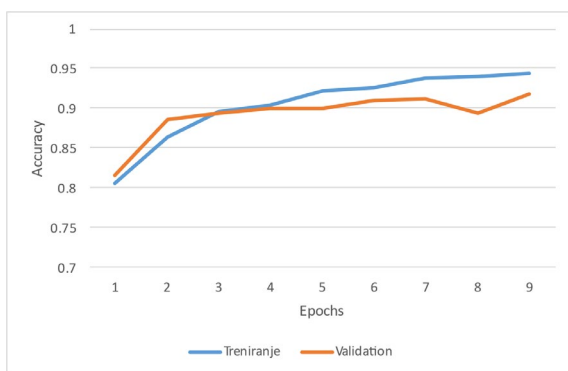
Slika 4: Grafični prikaz rezultatov validacijskega modela.

Vir: lasten.



Slika 5: Grafični prikaz rezultatov modela za klasifikacijo tipa slike.

Vir: lasten.

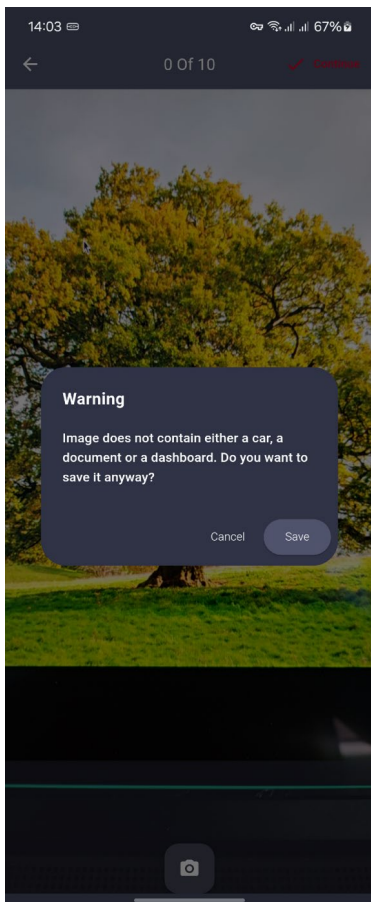


Slika 6. Grafični prikaz rezultatov modela za klasifikacijo strani vozila.

Vir: lasten.

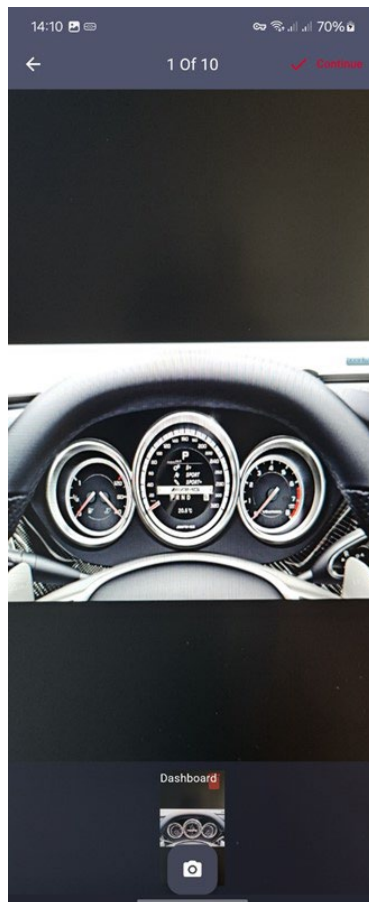
### 3.1 Implementacija modelov v mobilni aplikaciji

Dodana funkcionalnost klasifikaciji v mobilni aplikaciji je bistveno izboljšala izkušnjo uporabnika. Uporabniki lahko zdaj enostavno fotografirajo avtomobile in prejmejo takojšnjo klasifikacijo, ki vključuje informacije o ustreznosti slike (Slika 7), tipu slike (vozilo, armaturna plošča ali dokument) (Slika 8) ter strani avtomobila (spredaj, levo, desno, zadaj). Napoved ne vpliva na hitrost in odzivnost aplikaciji, ker MobileNetV3 je optimiziran za mobilni napravi in srednja napoved je dolg 60 ms.



Slika 7. Slika iz mobilne aplikaciji, primer preverjanja ustreznosti sliki.

Vir: lasten.



Slika 8. Slika iz mobilne aplikaciji, primer klasificiranja vrste sliki.

Vir: lasten.

## 4 Zaključek

V tem članku smo uspešno predstavili uporabo modela MobileNetV3 za klasifikacijo slik v mobilni aplikaciji, zgrajeni v ogrodju Flutter. Razvili smo tri različne modele za klasifikacijo avtomobilskih slik glede na ustreznost, tip slike in stran avtomobila.

Pomemben poudarek smo namenili pridobivanju in obdelavi podatkov, ki sta ključna koraka pri razvoju modelov umetne inteligence. S sistematičnim pristopom smo premagali izzive, povezane z označevanjem slik in ustvarili kakovosten podatkovni nabor za učenje.

MobileNetV3 se je izkazal kot učinkovit model za klasifikacijo slik, saj dosega visoko natančnost ob ohranjanju nizke računske zahtevnosti. Rezultati učenja modelov so bili obetavni, kar kaže na uspešnost našega pristopa.

Implementacija modelov v mobilno aplikacijo je prinesla izboljšanje uporabniške izkušnje pri klasifikaciji avtomobilskih slik v realnem času. Hitrost delovanja modela MobileNetV3 je omogočila odzivno in učinkovito uporabo v mobilni aplikaciji.

Skupno gledano smo s tem projektom potrdili potencial uporabe umetne inteligence, zlasti globokega učenja, v praksi. Kombinacija zmogljivega modela, ustrezno pripravljenih podatkov ter integracije v uporabniški vmesnik mobilne aplikacije odpira vrata za različne aplikacije v industriji, kjer je klasifikacija slik ključnega pomena.

#### Viri in literatura

- Binary Relevance (31.1.2024). Pridobljeno iz Scikit-multilearn:  
[http://scikit.ml/api/skmultilearn.problem\\_transform.br.html](http://scikit.ml/api/skmultilearn.problem_transform.br.html) (2024).
- de Carvalho, A. C. & Freitas, A. A. "A tutorial on multi-label classification techniques". *Foundations of Computational Intelligence Volume 5: Function Approximation and Classification*, (2009): 177-195.
- Flutter. (31. 1. 2024). `tflite_flutter`. Pridobljeno iz Pub.dev: [https://pub.dev/packages/tflite\\_flutter](https://pub.dev/packages/tflite_flutter) (2024)
- LeCun, Y., Benagio, Y. & Hinton, G. "Deep learning". *Nature* (2015): 436-444.
- Nikoloski, S. "Structured Output Prediction and Modeling Soil Functions". Doctoral Dissertation. Jozef Stefan International Postgraduate School, Ljubljana Slovenia (2020).
- O'Shea, K. & Nash, R. "An introduction to convolutional neural networks". *arXiv preprint arXiv:1511.08458* (2015).
- Pang, B., Nijkamp, E. & Wu, Y.N. "Deep learning with Tensorflow: A review". *Journal of Educational and Behavioral Statistics* (2020): 227-248.
- Qian, S. N. (2021). "MobileNetV3 for Image Classification". 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE) (str. 490-497). IEEE.
- Siebel, T. & Rice, C. "Digital Transformation: Survive and Thrive in an Era of Mass Extinction". New York: Rosetta Books (2019).
- Singh, A. & Bhadani, R. "Mobile Deep Learning with TensorFlow Lite, ML Kit and Flutter: Build scalable real-world projects to implement end-to-end neural networks on Android and iOS". Packt Publishing Ltd (2020).
- Sinha, A. "Image Dataset". Pridobljeno iz Kaggle:  
<https://www.kaggle.com/datasets/starktony45/image-dataset> (2024).
- Szymański, P. & Kajdanowicz, T. "A scikit-based Python environment for performing multi-label classification". *ArXiv preprint arXiv:1702.01460* (2017).
- TensorFlow. "Image classification". Pridobljeno iz TensorFlow:  
<https://www.tensorflow.org/tutorials/images/classification> (2024).

- Wang, P., Fan, E. & Wang, P. "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning". *Pattern Recognition Letters* (2021): 61-67.
- Yang, L., Luo, P., Change, L.C. & Tang, X. "A large-scale car dataset for fine-grained categorization and verification". *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015): 3973-3981.





# USTVARJANJE PONAREJENIH VIDEOPOSNETKOV S POMOČJO DIFUZIJSKIH MODELOV

BINE MARKELJ, PETER PEER, BORUT BATAGELJ

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Ljubljana, Slovenija  
bm9928@student.uni-lj.si, peter.peer@fri.uni-lj.si, borut.batagelj@fri.uni-lj.si

V članku predstavimo postopke in tehnike generiranja globoko ponarejenih videoposnetkov ali krajše globokih ponaredkov (angl. deepfakes). To so videoposnetki, pri katerih je prišlo do manipulacij s tehnikami globokega učenja. Taki videoposnetki predstavljajo velik problem pri širjenju lažnih novic, politični propagandi, uničevanju podobe posameznikov, izdelavi pornografskih vsebin, izsiljevanju itd. V članku opišemo podatkovno zbirko FaceForensics++ in predstavimo lastno metodo za potencialno izdelavo podzbirke omenjene baze z uporabo najnovejših generativnih difuzijskih modelov. Uporabljene postopke eksperimenta predstavimo in analiziramo njihovo kvaliteto in uspešnost. Komentiramo tudi smiselnost uporabe in nevarnost, ki jo predstavljajo ponarejeni videoposnetki, izdelani z difuzijskimi modeli.

DOI  
[https://doi.org/  
10.18690/um.feri.1.2024.8](https://doi.org/10.18690/um.feri.1.2024.8)

ISBN  
978-961-286-837-6

#### Ključne besede:

ponarejeni videoposnetki,  
globoki ponaredko,  
globoko učenje,  
nevronska mreža,  
difuzijski modeli,  
stabilna difuzija

#### Prispevek temelji na:

Markelj, B. (2023),  
*Ustvarjanje ponarejenih  
videoposnetkov s pomočjo  
difuzijskih modelov za  
razširitev zbirke za odkrivanje  
ponarejenih videoposnetkov*.  
diplomsko delo, Univerza  
v Ljubljani, Fakulteta za  
računalništvo in  
informatiko.



Univerzitetna založba  
Univerze v Mariboru

DOI

[https://doi.org/  
10.18690/um.feri.1.2024.8](https://doi.org/10.18690/um.feri.1.2024.8)

ISBN

978-961-286-837-6

**Keywords:**

fake video,  
deepfake,  
deep learning,  
neural network,  
diffusion models,  
stable diffusion

**The proceedings is based on:**

Markelj, B.(2023),  
*Ustvarjanje ponarejenih  
videoposnetkov s pomočjo  
difuzijskih modelov za razširitev  
zbirke za odkrivanje ponarejenih  
videoposnetkov.* bachelor's  
thesis, University of  
Ljubljani, Faculty of  
Computer Science and  
Informatics.

# CREATING FAKE VIDEOS USING DIFFUSION MODELS

BINE MARKELJ, PETER PEER, BORUT BATAGELJ

University of Ljubljana, Faculty of Computer Science and Informatics, Ljubljana,  
Slovenia

[bm9928@student.uni-lj.si](mailto:bm9928@student.uni-lj.si), [peter.peer@fri.uni-lj.si](mailto:peter.peer@fri.uni-lj.si), [borut.batagelj@fri.uni-lj.si](mailto:borut.batagelj@fri.uni-lj.si)

In the article we present techniques and procedures for generating deepfake videos. These are videos that were subjected to manipulations with deep learning techniques. Such videos represent a major problem in the spread of fake news, political propaganda, destruction of individual's public image, production of pornographic content, extortion, etc. In the article we describe deepfake database FaceForensics++. We also present our own method for potential creation of a subset of the mentioned database using the latest generative diffusion models. We describe multiple techniques, that we used and tested in our experiment and analyse their quality and success. We also comment on the utility and danger posed by fake videos generated by diffusion models.



## **1 Uvod**

V zadnjih letih vse več slišimo o umetni inteligenci in njenih hitrih in impresivnih napredkih. Orodja generativne umetne inteligence, kot so veliki jezikovni modeli (GPT-3.5 GPT-4, PaLM2), postajajo vse bolj uporabljena, kvalitetna in sofisticirana. Velik napredek generativne umetne inteligence pa je bil v zadnjem času dosežen tudi na področju generiranja slikovnih medijev.

Slike in videoposnetki predstavljajo enega izmed najbolj razširjenih in zaupanja vrednih načinov širjenja informacij. Napredki v umetni inteligenci in razvoj na področju generativnih nasprotniških mrež omogočajo izdelavo zelo prepričljivih ponarejenih videoposnetkov (angl. deepfake), ki postajajo vse boljši in jih danes s prostim očesom praktično ne moremo več ločiti od avtentičnih videoposnetkov.

Danes lahko ponarejene posnetke ustvarimo z zelo naprednimi spletnimi orodji, za katera ne potrebujemo veliko tehničnega znanja. Zato vedno pogosteje prihaja do zlorab. Ocenjujejo, da je velika večina ponarejenih videoposnetkov uporabljena za izdelavo pornografskih vsebin, lažnih novic, zlorabo identitete itd. (Markelj, 2023).

Večina trenutnih ponarejenih videoposnetkov je narejenih z uporabo globokih nevronskih mrež. Največkrat sta uporabljeni 2 tehniki: samokodirniki (angl. autoencoders) in generativne nasprotniške mreže (angl. generative adversarial network, GAN).

Velik napredek na področju generativnih nevronskih mrež predstavljajo difuzijski modeli ter njihova uporaba za generiranje slikovnih medijev. Zato se bo naš članek osredotočil na njihovo uporabnost in tveganje pri ustvarjanju ponarejenih videoposnetkov (Markelj, 2023).

Zaradi vseh nevarnosti, ki jih ti posnetki predstavljajo, se nam zdi pomembno, da se metode in podatkovne zbirke za prepoznavanje in identifikacijo ponarejenih videoposnetkov ves čas razvijajo in izboljšujejo. Zato predlagamo nov postopek uporabe modelov stabilne difuzije, ki bi lahko služil za kasnejšo izdelavo podzbirke že obstoječe podatkovne zbirke ponarejenih videoposnetkov FaceForensics++ (FF++) (Rössler et al., 2022). Med raziskavo področja difuzijskih modelov smo

odkrili več različnih tehnik in načinov za pridobitev ponarejenih videoposnetkov, ki jih v nadaljevanju tudi predstavimo in analiziramo.

## 2 Pregled področja

### 2.1 Generiranje ponarejenih videoposnetkov

Večina ponarejenih videoposnetkov je ustvarjenih z uporabo generativnih nasprotniških mrež – GAN in samokodirnikov.

Tehnike generiranja ponarejenih videoposnetkov v grobem delimo v 2 glavni skupini: zamenjava obraza ter uprizoritev obraza (Yu et al., 2020).

Pri zamenjavi obraza algoritem izvede menjavo obraza med prvotno in ciljno osebo, pri čemer ohranja obrazno mimiko in izraz. To je zelo popularna tehnika zaradi različnih dostopnih spletnih orodij, ki omogočajo hiter dostop do dobrih rezultatov.

Uprizoritev obraza pa vključuje prenos obraznih mimik in pozicije obraznih točk prvotne osebe na ciljno osebo. Program zajame mimiko, globino ter osvetlitev obraza, kar omogoča prenos obraznih mimik in značilnih točk (angl. facial landmarks).

### 2.2 Odkrivanje ponarejenih videoposnetkov

V grobem poznamo 2 načina prepoznavanja ponarejenih videoposnetkov:

1. Klasične metode, ki temeljijo na predprocesiranju vhoda, luščenje značilk (angl. feature extraction) in klasifikaciji. Algoritmi iščejo zabrisane robove, artefakte itd.
2. Metode globokega učenja, za katere so značilni sočasno učenje, luščenje značilk in njihova klasifikacija. V ta namen se uporabljajo konvolucijske nevronske mreže oz. CNN in vizualne transformatorje (Zhang et al., 2022), ki kot vhod lahko sprejmejo vizualne medije. Za zaznavo ponarejenih posnetkov je ključnega pomena upoštevanje časovne komponente (Gu et al., 2021).

### 2.3 Podatkovna zbirka ponarejenih videoposnetkov FaceForensics++

V našem članku smo se osredotočili na videoposnetke iz zbirke FaceForensics++ ali krajše FF++ (Rössler et al., 2022).

To je ena najbolj citiranih in uporabljenih zbirk na področju odkrivanja ponarejenih videoposnetkov. Pogosto je uporabljena kot merilo uspešnosti algoritmov za odkrivanje obraznih manipulacij.

Sestavljena je iz 1000 originalnih posnetkov, nad katerimi so avtorji uporabili 5 različnih tehnik manipulacije: FaceSwap, Deepfakes, Face2Face, Neural Textures in FaceShifter.

Posnetki te zbirke so nam služili kot osnova za naše delo.

## 3 Teoretično ozadje in metodologija

Zaradi napredkov v raziskovanju tehnik globokega učenja (angl. deep learning) in razvoja arhitekture konvolucijskih nevronske mreže – CNN, ki omogočajo obdelavo vizualnih podatkov, je področje ustvarjanja in raziskovanja globokih ponaredek doživelo velik razcvet in popularnost.

Najbolj uporabljani tehniki generiranja ponarejenih videoposnetkov sta:

1. Variacijski samokodirniki – VAE  
Ti združujejo glavne lastnosti klasičnih samokodirnikov ter verjetnostnih modelov za generacijo novih podatkov z vnaprej naučenimi značilnostmi (npr. obraz).
2. Generativne nasprotniške mreže – GAN  
So posebna vrsta generativnih modelov. Sestavljene so iz dveh nevronske mreže (generator in diskriminator), ki med učenjem med seboj tekmujeta in se tako hkrati izboljšujeta. Po koncu učenja generatorska mreža generira realistične slike (npr. obrazov).

Najnovejši in zato za nas najzanimivejši in aktualni generativni modeli pa so difuzijski modeli.

### 3.1 Difuzijski modeli

Difuzijski modeli so sposobni generiranja visoko kvalitetnih in realističnih slik iz šuma in vodenja z besedilom ali sliko (Rombach et al., 2022}. Za razliko od tehnologij, ki temeljijo na GAN, difuzijski modeli generirajo sliko v več korakih, pri čemer ohranjajo njene dimenzije skozi iteracije. Ti modeli se naučijo zadeti porazdelitev podatkov z obračanjem postopka večkoračnega procesa šumenja (angl. noise process). Iz šuma generirajo kompleksne in realistične vzorce in dosežajo primerljive oz. boljše rezultate kot modeli GAN in VAE.

Difuzijski modeli torej vsebujejo dva glavna procesa:

1. Difuzijski proces naprej (angl. forward diffusion process)  
To je proces, ki vhodni sliki v korakih dodaja Gaussov šum. Na koncu vrne sliko, ki je le še šum s standardno normalno razporeditvijo -- šum s povprečno vrednostjo 0 in enotsko varianco. Ker je proces dodajanja šuma vnaprej določen, se ga model ne uči, uporaben je za učenje obratnega difuzijskega procesa.
2. Obratni difuzijski proces (angl. reverse diffusion process)  
Difuzijski proces obrnemo. Začnemo s sliko šuma, ki jo v korakih rekonstruiramo nazaj v realistično sliko – odstraniti moramo šum iz difuzijskega procesa naprej. Postopek poteka v iteracijah in ga nadzoruje model nevronske mreže. Naloga mreže je, da v vsaki iteraciji napove šum, ki je bil sliki dodan pri difuzijskem procesu naprej v prejšnji iteraciji.

Oba difuzijska procesa sta še vizualno prikazana v (Raya, 2023).

Pogosta arhitekturna implementacija generativnih difuzijskih modelov je mreža U-net (Ronneberger et al., 2015), ki vsebuje več zaporednih konvolucijskih in združevalnih slojev ter sloje povečanja vzorčenja (angl. upsampling layer) in preskočne povezave (angl. skip connections).

Difuzijski modeli so zaradi visoke kvalitete generiranih slik v zadnjem času postali zelo priljubljeni. Za naše delo bomo uporabljali stabilno difuzijo, ki jo podrobneje opišemo v poglavju 3.3.

## 3.2 Modeli LoRA

Modeli LoRA oz. prilagoditveni modeli nizkega ranga (angl. low rank adaptation model) se uporabljajo za potrebe manjših prilagoditev, uglaševanja (angl. fine tuning) in domenske prilagoditve velikih modelov. Dosegajo dobre rezultate dodajanja znanja velikim modelom z relativno malim številom učnih primerov ter kratkim časom učenja. Model lahko naučimo koncepta, stila slike, predmeta ali pa osebe.

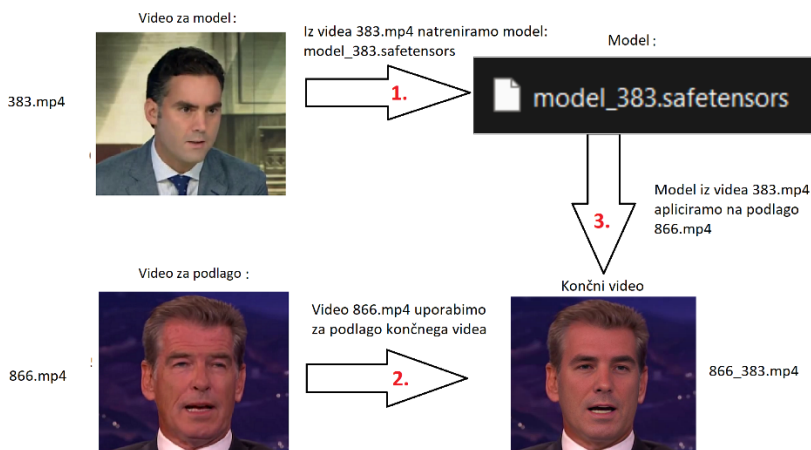
Model LoRA plastem navzkrižne pozornosti velikega modela z obtežitvijo dodaja matrike nizkega ranga. Ideja izhaja iz matrične faktorizacije na osnovi ranga, kjer matriko predstavimo kot produkt dveh manjših matrik, kar občutno zmanjša število parametrov in s tem čas učenja.

### 3.2.1 Postopek učenja identitete

V članku želimo odkriti najboljši način generiranja prepričljivih ponarejenih videoposnetkov z uporabo generativnih difuzijskih modelov. Glavni problem takih videoposnetkov, predstavlja povezanost med sličicami, ki tvorijo videoposnetek. Zato za generiranje sličic potrebujemo nekakšno podlago, ki bo služila vodenju sistema, da bodo zaporedne sličice karseda povezane. Kot podlago smo vzeli obstoječ video iz podatkovne zbirke FF++ in ga razdelili na posamezne sličice. Nato smo iz sličic drugega videa model LoRA naučili generiranja te osebe. Tako smo difuzijski proces omejili, da v celotnem posnetku ohranja identiteto iste osebe. Osnovni potek je prikazan na Sliki 2.

Model LoRA torej doseže, da difuzijski model generira sličice, na katerih je ves čas ista oseba.

Pripravo učnega seta za modele LoRA smo avtomatizirali. Razvili smo poseben program, ki iz videa avtomatsko izlušči zahtevano število sličic, ki so si med seboj čim bolj različne, in jih na koncu še dodatno obdela. Slikam v učni mapi je pred učenjem potrebno dodati istoimenske besedilne datoteke, ki opišejo, kaj je na sliki. Algoritem tako med učenjem ve, na kaj mora biti pozoren in se nauči povezav med videnim in napisanim.



**Slika 2: Struktura osnovne ideje: iz sličic videa 383.mp4 s pomočjo modela LoRA naučimo identiteto, video 866.mp4 uporabimo za podlago končnega videa. Model identitete model\_383.safetensors apliciramo na podlago. Dobimo video identitete modela 383 na videoposnetku 866.mp4, imenovan 866\_383.mp4 (spodaj desno).**

Vir: lasten.

Za LoRA učenje je pomembna tudi vnaprej predpisana struktura učne mape, množica regularizacijskih slik, število korakov učenja in optimalni parametri učenja modela, ki smo jih pridobili z daljšim eksperimentiranjem.

### 3.3 Stabilna difuzija

Model stabilne difuzije je generativni difuzijski model izdan leta 2022 (Rombach et al., 2021). Naučen je bil na več milijardah slik in na podlagi tekstovnih navodil oz. poziva (angl. prompt) in/ali slike ustvari novo realistično sliko. Stabilna difuzija ima več možnosti generiranja: besedilo v sliko (angl. txt2img), slika v sliko (angl. img2img) in slikovna vrisava (angl. image inpainting).

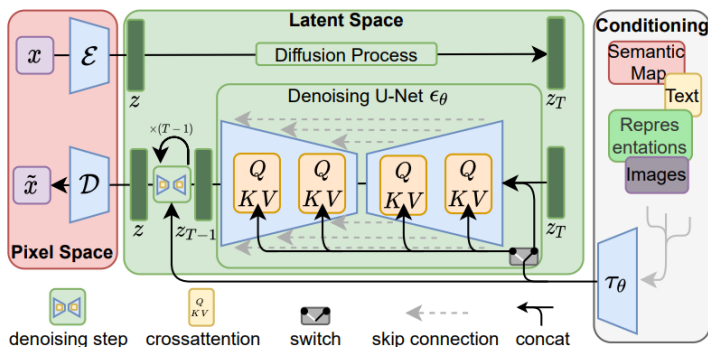
Arhitektura sistema stabilne difuzije ima 3 glavne komponente (Slika 3):

1. Kodirnik besedila, ki difuzijski proces dodatno usmerja z upoštevanjem besedilnega ukaza (zakodiranega v latentni prostor).
2. Nevronska mreža tipa U-net je glavni del sistema, ki na podlagi besedilnega navodila in latentnega šuma obrne difuzijski proces, tako da v vsakem koraku čim bolj natančno predvidi šum, ki ga mora



odstraniti za generacijo slike. Odstranjevanje šuma poteka v latentnem prostoru, kar zaradi manjših dimenzij znatno pohitri postopek generiranja.

3. Slikovni samokodirnik, ki je zadolžen za kodiranje in dekodiranje slike iz začetnega prostora v latentno predstavitev in nazaj.



Slika 3: Prikaz osnovne arhitekture stabilne difuzije, osrednji del predstavlja difuzijski proces z mrežo U-net, prehajanje med levim in srednjim delom predstavlja delovanje slikovnega kodirnika. Na desni strani so prikazani načini omejevanja nastale slike – besedilno navodilo, slika itd.

Vir: (Rombach, 2022).

### 3.3.1 Naš postopek

Stabilna difuzija nam pred generiranjem omogoča izbiro velikega števila različnih parametrov, ki vplivajo na končno generirano sliko. Parametri in njihov vpliv so opisani v Tabeli 1.

Videoposnetek generiramo kot serijo zaporednih sličic, ki jih na koncu sestavimo skupaj v zaključen videoposnetek. Največji izziv pri povezavi posebej zgeneriranih sličic v video predstavlja časovna povezanost med njimi. Slaba konsistenca med posameznimi sličicami lahko povzroči nenaravni efekt preskakovanja (angl. flickering). V ta namen smo uporabili orodja izboljšave, kot so: *EbSynth Beta*, ki šum odpravlja s stiliziranjem posnetka po vodilnih sličicah in razširitev *Abyss- $\alpha$ -LAB-Ext*, ki odstrani šum, utripanje ter normalizira barve.

Tabela 1: Glavni parametri generiranja in njihov vpliv na sliko

PARAMTER	VPLIV NA GENERIRANO SLIKO
poziv	besedilno navodilo, opis željene končne slike
negativni poziv	besedilno navodilo, stvari, ki jih ne želimo na sliki
slika	slika, po kateri želimo, naj se končna slika ravna
metoda vzorčenja	algoritem vzorčenja, uporabljen pri generaciji slike
koraki vzorčenja	število korakov, ki jih izvede izbrana metoda vzorčenja
dimenzije slike	širina in višina končne slike
število serij	koliko serij končnih slik bo zgeneriranih
velikost serije	število zgeneriranih končnih slik v eni seriji
lestvica CFG	stopnja prilaganja končne slike pozivu
stopnja odpravljanja hrupa	stopnja prilaganja končne slike vhodni
naključno seme	naključno število, ki vpliva na začetni šum

Za glavno "podlago" smo vzeli sličice obstoječega pristnega videoposnetka iz podatkovne zbirke FF++ in osebi na posnetku zamenjali obrazno identiteto z naučenim modelom LoRA. Uporaba obstoječega posnetka je nujna za generiranje realističnih videoposnetkov, saj nam le-ta služi kot nekakšen skelet in vodilo, po katerem se nov ponarejeni video ravna.

Eden najpomembnejših mehanizmov nadzora generiranja sličic je razširitev kontrolnih mrež (angl. controlnet) (Zhang et al., 2023). To so posebne nevronske mreže, ki procesu generiranja dodajo dodatne omejitve in pogoje. Uporabne so za maskiranje slik, omejitve robov, sledenje obraznim mimikam, itd. (Slika 4, levo).

Za izdelavo obraznih mask za način vrisovanja smo uporabili razširitev *batch-face-swap*, ki na posamezni sliki zazna obraz in na njegovem mestu naredi poljubno veliko masko (slika 4, desno).



Slika 4: Levo: prikaz izdelanih map kontrolnih mrež različnih predprocesorjev. Desno: prikaz obrazne maske.

Vir: lasten

## 4 Eksperiment

Najprej predstavimo eksperiment treniranja modelov LoRA. Nato opišemo tudi potek in proces eksperimentiranja generiranja ponarejenih videoposnetkov s stabilno difuzijo, ter analiziramo dobljene rezultate.

### 4.1 Priprava modelov LoRA

Za učenje uporabnega modela LoRA potrebujemo dovolj raznoliko učno množico slik, ki osebo, ki jo želimo z modelom zajeti, predstavi v več različnih pozicijah, na različnem ozadju in iz različnih zornih kotov. Ker je pri uporabi enega samega videoposnetka to lahko velika omejitev, smo v ta namen razvili program, ki avtomatsko iz videoposnetka pobere ključne sličice in uvede umetno raznolikost.

Ugotovili smo, da je optimalno število sličic za učenje identitete 25. Program najprej izbere vse ključne okvirje s pomočjo orodja *ffmpeg*. Nato naključno izbere preostale sličice, kriterij izbire je, da zajame optimalno razmerje raznolikih sličic. To doseže z obrazno analizo, kjer na vsaki sličici poišče vse glavne obrazne točke s pomočjo knjižnice *mediapipe* in klasificira ali ima oseba zaprte/odprte oči in usta. Ustrezno razmerje tako različnih slik nam kasneje omogoča uprizoritev govorjenja in raznolikih izrazov obraza. Na koncu naključnim fotografijam dodamo še odstranimo ozadje s pomočjo modela *rembg* in jih še dodatno obrežemo. S tem še povečamo raznolikost in dinamičnost učne množice.

Med procesom učenja smo vsak model unikatno poimenovali in dodali večjo regularizacijsko množico slik (okoli 200). Datoteke opisov slik smo zgenerirali z globokim modelom BLIP. Optimalne rezultate smo dobili z 9 ponovitvami učne in 1 ponovitvijo regularizacijske množice, 10 epohami učenja ter serijo velikosti 1. Celotno optimalno učenje modela tako poteka v okoli 2200 korakih.

### 4.2 Generiranje lažnih videoposnetkov

Trenutno izdelava prepričljivih videoposnetkov z difuzijo zaradi same novosti tehnologije še vedno predstavlja izziv. Hkrati naši ciljni posnetki vedno prikazujejo govorjenje, ki je za replikacijo z difuzijskimi modeli zelo zapleteno. Prav zato smo k delu pristopili zelo eksperimentalno in s pomočjo spletnih virov in različnih

razširitev in programov preizkusili in ustvarili čim več različnih pristopov. Najprej le na kratko opišemo prvotne načine, ki pa so nas končno pripeljali do našega najuspešnejšega načina, ki ga tudi podrobneje analiziramo in komentiramo.

#### 4.2.2 Predhodni eksperimenti

Sprva smo želeli z načinom `img2img` generirati celotne sličice, ne le obrazni del. Za ohranjanje premikov in poze oseb smo uporabili več kombinacij kontrolnih mrež. Rezultat smo nato dodatno izboljšali z posebno skripto, ki uporabi kot dodatno vodilo generiranja še prejšnje slike. Kasneje smo preizkusili še uporabo orodja za stiliziranja po vodilnih sličicah – *EbSynth Beta*. Za izboljšanje enakosti med sličicami smo sličice generirali v obliki mreže 25 sličic, ki so bile tako bolj enotne. Ker omenjeni eksperimenti niso dali zadovoljivih rezultatov, smo se osredotočili na način generiranja z vrisovanjem, kjer menjamo le obrazno identiteto osebe. Ta način smo združili z orodjem *EbSynth Beta*. Generirali smo vsako deseto sličico videoposnetka in jih v orodju združili s še dodatnim zamikanjem sličic in navzkrižnim bledenjem (angl. *crossfade*). Rezultati omenjenih tehnik, še niso bili dovolj prepričljivi (zabrisani robovi, nenaravno premikanje ust, degradacija barv, itd.), služili pa so nam, kot podlaga za naslednji postopek.

#### 4.2.2 Najuspešnejši postopek generiranja in analiza rezultatov

Najprej za vsako sličico ciljnega videoposnetka z razširitvijo *batch-face-swap* pridobimo obrazno masko ustrezne velikosti. Nato z vrisovanjem poiščemo optimalne nastavitve generiranja sličic in kombinacije kontrolnih mrež. Izbira pravih kontrolnih mrež in njihovih obtežitev predstavlja enega izmed ključnih korakov.

Med najpomembnejše kontrolne mreže, ki dajejo najboljše rezultate, spadajo:

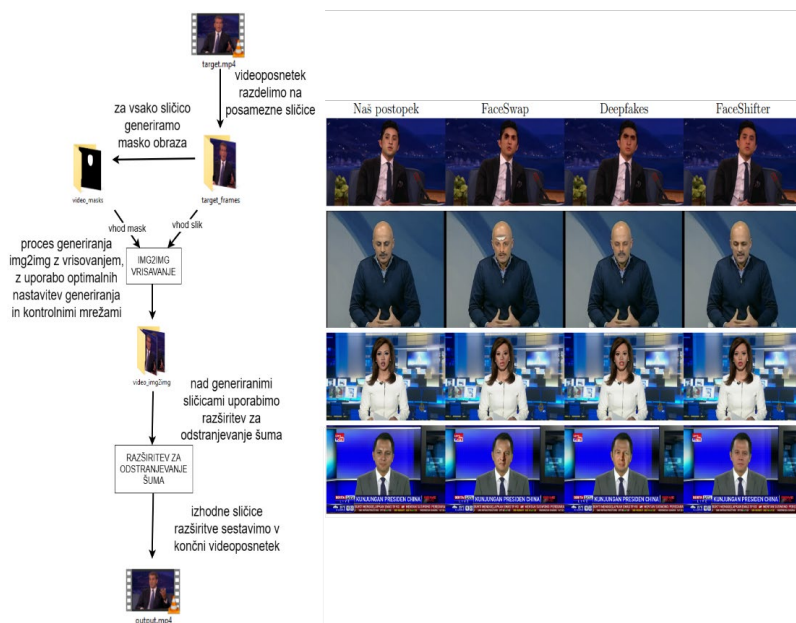
- Kontrolni mreži *OpenPose* in *MediaPipeFace*, ki skrbita za zajem ključnih obraznih točk in mimik med govorjenjem.
- Kontrolna mreža *TemporalNet*, ki je ključnega pomena za doseganje časovne konsistence. To dosega z uporabo konteksta vseh prej zgeneriranih sličic.
- Kontrolna mreža *Tile*, ki pomaga pri popraviljanju barv – izenačitev kožnega odtenka novo generiranega obraza in ciljnega obraza.
- Kontrolna mreža *Inpaint*, ki pri vrisovanju poskrbi za glajenje robov maske

Proces iskanja optimalnih parametrov je lahko zaradi raznolikih videoposnetkov lahko zelo dolgotrajen in nekoliko drugačen ob vsakem generiranju.

Enake parametre generiranja in kombinacije kontrolnih mrež nato uporabimo nad vsemi sličicami ciljnega posnetka ter tako na vsako naneseemo novo identiteto z modelom LoRA.

Končno posamezne sličice združimo v ponarejeni videoposnetek. Nato z uporabo razširitve *Abyss-LAB-Ext* še dodatno odstranimo zadnje sledove šuma, utripanja ter izvedemo še končno barvno normalizacijo. Celoten postopek prikažemo na Sliki 5.

Ponarejeni videoposnetek ustvarjen z zgoraj opisanim postopkom ima visoko časovno konsistenco, minimalno prisotnostjo utripanja in visoko kvaliteto. Smo mnenja, da proces predstavlja mejnik na področju generiranja ponarejenih posnetkov z difuzijskimi modeli in daje rezultate, ki so enakovredni oz. celo presegajo obstoječe tehnike ponarejanja videoposnetkov v bazi FF++. Vizualno primerjavo prikažemo na Sliki 5.



Slika 5: Levo: prikaz celotnega poteka generiranja ponarejenega videoposnetka. Desno: vizualna primerjava generiranih istoležnih sličic z našim postopkom in tehnikami v zbirki FF++.

Vir: lasten

## 5 Zaključek in nadaljnje delo

V članku smo pokazali več načinov generiranja prepričljivih ponarejenih videoposnetkov z difuzijo ter osvetlili teoretično ozadje. Vendar pa problematika generacije ponarejenih videoposnetkov z difuzijskimi modeli še zdaleč ni dokončno raziskana. Pokazali smo, da naša metoda daje rezultate, ki po kakovosti presegajo trenutne tehnike manipulacij. Predvidevamo pa, da bo zaradi hitrega razvoja področja prišlo do vedno boljših načinov manipulacij z difuzijo in bodo taki posnetki v bližnji prihodnosti predstavljali veliko nevarnost.

Prav zaradi hitrega napredovanja v razvoju tehnologije se nam zdi pomembno, da se čim več raziskav posveti področju odkrivanja ponarejenih videoposnetkov nastalih z novimi tehnikami difuzijskih modelov. Izdelava dobre podatkovne zbirke takih manipulacij nam lahko pomaga, da ostanemo v koraku s časom z napadalci, ki bi difuzijsko tehnologijo lahko uporabljali v zle namene. Pomemben naslednji korak bi bila avtomatizacija procesa generiranja posnetkov in izdelava celotne zbirke takih posnetkov, ki bi služili za učenje algoritmov za odkrivanje manipulacij z difuzijo.

### Viri in literatura

- MARKELJ, BINE, 2023, Ustvarjanje ponarejenih videoposnetkov s pomočjo difuzijskih modelov za razširitev zbirke za odkrivanje ponarejenih videoposnetkov [na spletu]. Diplomsko delo. Pridobljeno s: <https://repozitorij.uni-lj.si/IzpisGradiva.php?lang=slv&id=149326>
- Andreas Rössler in sod. "FaceForensics++: Learning to Detect Manipulated Facial Images". V: International Conference on Computer Vision (ICCV). 2019.
- Peipeng Yu in sod. "A Survey on Deepfake Video Detection". V: IET Biometrics 10.6 (2021), str. 607–624. doi: <https://doi.org/10.1049/bme2.12031>. eprint: <https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/bme2.12031>. url: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/bme2.12031>.
- Daichi Zhang in sod. Deepfake Video Detection with Spatiotemporal Dropout Transformer. 2022. arXiv: 2207.06612 [cs.CV].
- Zhihao Gu in sod. Spatiotemporal Inconsistency Learning for DeepFake Video Detection. 2021. arXiv: 2109.01860 [cs.CV].
- Robin Rombach in sod. High-Resolution Image Synthesis with Latent Diffusion Models. 2021. arXiv: 2112.10752 [cs.CV].
- Robin Rombach in sod. High-Resolution Image Synthesis with Latent Diffusion Models. 2022. arXiv: 2112.10752 [cs.CV].
- Gabriel Raya in Luca Ambrogioni. Diffusion Models Seminar. url: <https://diffusionmodels.nl/> (pridobljeno 25. 8. 2023).
- Olaf Ronneberger, Philipp Fischer in Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015. arXiv: 1505.04597 [cs.CV].
- Lvmin Zhang. Controlnet - Official implementation of Adding Conditional Control to Text-to-Image Diffusion Models. url: <https://github.com/lllyasviel/ControlNet> (pridobljeno 30. 8. 2023).

# SPECTRAL RESPONSE OF TWO HYPERSENSITIVE CAMERAS FOR UXO ENDMEMBER SELECTION

MILAN BAJIĆ,<sup>1</sup> BOŽIDAR POTOČNIK<sup>2</sup>

<sup>1</sup> Zagreb University of Applied Sciences, Department of IT and Computer Sciences,  
Zagreb, Croatia  
mbajic@tvz.hr

<sup>2</sup> University of Maribor, Faculty of Electrical Engineering and Computer Science,  
Institute of Computer Science, Maribor, Slovenia  
bozidar.potocnik@um.si

There is currently no recommended procedure for acquiring endmembers in hyperspectral target detection when targets are larger than a single pixel. What is the best approach when multiple cameras are available for a dataset construction? This study examines the differences between hyperspectral cameras Specim IQ and Specim Inspector V9 that recorded the same surfaces under the same lighting conditions. A white balance card and a mortar mine are considered. As calibration procedures for cameras differ, raw data without processing are compared, and the same wavelength range is chosen. Clear differences are noticed between the spectra of the two cameras. Finally, guidelines for selecting statistically reliable endmembers and constructing an endmember dataset are provided based on the obtained results.

DOI  
[https://doi.org/  
10.18690/um.feri.1.2024.9](https://doi.org/10.18690/um.feri.1.2024.9)

ISBN  
978-961-286-837-6

## Keywords:

hyperspectral camera,  
spectral response,  
unexploded ordnances  
(UXO),  
endmember selection,  
mortar mine  
hiperspektralna kamera,  
spektralni odziv,  
neeksplozivna ubojna  
sredstva (UXO),  
izbira končnih članov,  
minometna mina

DOI  
[https://doi.org/  
10.18690/um.feri.1.2024.9](https://doi.org/10.18690/um.feri.1.2024.9)

ISBN  
978-961-286-837-6

**Ključne besede:**

hiperspektralna kamera,  
spektralni odziv,  
neeksplozivna ubojna  
sredstva (UXO),  
izbira končnih članov,  
minometna mina

# SPEKTRALNI ODZIV DVEH HIPERSPEKTRALNIH KAMER ZA IZBOR KONČNIH ČLANOV UXO

MILAN BAJIĆ,<sup>1</sup> BOŽIDAR POTOČNIK<sup>2</sup>

<sup>1</sup> Tehničko Veleučilište u Zagrebu, Katedra za IT in računalništvo, Zagreb, Hrvaška  
[mbajic@tvz.hr](mailto:mbajic@tvz.hr)

<sup>2</sup> Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Inštitut  
za računalništvo, Maribor, Slovenija  
[bozidar.potocnik@um.si](mailto:bozidar.potocnik@um.si)

Trenutno ni priporočenega postopka za pridobivanje končnih članov pri hiperspektralnem zaznavanju tarč, kadar so tarče večje od enega piksla. Kateri je najboljši pristop, ko je za konstruiranje nabora podatkov na voljo več kamer? Ta študija preučuje razlike med hiperspektralnima kamerama Specim IQ in Specim Inspector V9, ki sta posneli iste površine pod enakimi svetlobnimi pogoji. Upoštevana sta bela kalibracijska kartica in minometna mina. Ker se postopki umerjanja za kameri razlikujejo, se primerjajo neobdelani surovi podatki in izbere se enako območje valovnih dolžin. Opazne so jasne razlike med spektroma obeh kamer. Nazadnje na podlagi dobljenih rezultatov podajamo smernice za izbiro statistično zanesljivih končnih članov in izdelavo podatkovne zbirke končnih članov.





## **1 Introduction**

Hyperspectral image data comprises of a set of coherent images that represent intensities across different wavelength bands. These images consist of a set of pixels, also known as voxels, containing two-dimensional spatial details (with  $m$  rows and  $n$  columns) and spectral information across  $K$  wavelengths. Typically, they are referred to by various terms such as three-dimensional hyperspectral cube (shortly, hypercube), data cube, data volume, spectral cube, or spectral volume. Such data can provide insights into a tested material's physical and chemical properties. The information includes physical and geometric aspects like size, orientation, shape, colour, and texture. An unprocessed hyperspectral image consists of interconnected sub-images, each depicting the intensity and spatial arrangement of the examined object at a specific wavelength. These individual spatial images can be extracted from the hypercube at any wavelength within the entire spectral sensitivity range of the system. Consequently, a hyperspectral image denoted as  $I(x, y, \lambda)$  can be understood either as an independent spatial image  $I(x, y)$  at each wavelength ( $\lambda$ ) or as a spectrum  $I(\lambda)$  at every pixel  $(x, y)$ . Each pixel in the hyperspectral image encapsulates a spectrum specific to its position, serving as a unique fingerprint for characterizing the composition of that pixel. Hyperspectral imaging captures spatially distributed spectral responses at the pixel level, allowing for a flexible selection of regions of interest on a target object with variable sizes and locations. For instance, extracting two pixels from distinct compositional locations in the hypercube would result in different spectral signatures, each showcasing unique fingerprints associated with their respective compositions. These resulting spectra act as fingerprints facilitating the characterization of the chemical composition at each pixel, thereby providing valuable insights into the material under examination (ElMasry & Sun, 2010). Many hyperspectral cameras are available in the market. Usually, there are no serious problems if endmembers and studied material are observed with the same camera. However, several questions arise if endmembers are collected with one camera and studied material with another.

Unexploded ordnances (UXO) are present in more than sixty countries around the world. For instance, UXO from World War I are still discovered when archaeological excavation or construction works are performed (National University of Public Service, Hungary & Ember, 2021)(Roberts & Williams, 1995). Consequences of explosions in ammunition storage areas, especially in densely

populated regions, have had a significant humanitarian impact, leading to fatalities, injuries, environmental harm, displacement of communities, and disruption of livelihoods in more than 100 countries. Accidental detonations of ammunition warehouses rank among the most powerful explosions ever documented (Ammunition Storage Area Explosions – EOD Clearance, 2021). While UXOs buried in the ground are not suitable for hyperspectral detections, those on the surface can be very well detected (Bajić et al., 2013) (Bajić & Bajić, 2021).

In the evaluation article of Behmann et al. (Behmann et al., 2018), the Specim IQ camera was compared to the Specim V10E. The conclusions from this study can help us understand what problems might arise if one camera is used to select the endmembers and another to detect the materials studied. However, the evaluation article's faulty assumption is that the IQ camera uses other principle for acquisition of images and not the 'push broom' technique. However, when we tested the IQ camera, we tried it handheld, but the obtained image (cube) was very distorted. A workflow that we were able to track on the IQ camera screen indicated that it is a line scanner in a box.

With the availability of similar two hyperspectral cameras as in Behmann's case, i.e., the Specim IQ and older Specim Inspector V9 cameras, we would like to highlight the problem of correct selection of endmembers in hyperspectral detection of targets larger than one pixel. It should be emphasized that the IQ delivers very fast and has a user-friendly workflow, which makes the limitations of technology unnoticeable. Thus, in our study, we will conduct a similar experiment as in the above evaluation article and, simultaneously, report gained experiences with both hyperspectral cameras.

## 2 Materials and methods

The first sensor used in our research is **Specim Inspector V9** (see Figure 1a). It is a hyperspectral prism-grating imaging spectrograph, which has a spectral range of 430-900 nm, with a spectral resolution of 7 nm, and possible 1024 channels. It is a very good instrument but require longer post processing for creation of hyperspectral cube. Raw data are collected in lines, Figure 1b, and, afterwards, data need to be calibrated (see Figure 1c).

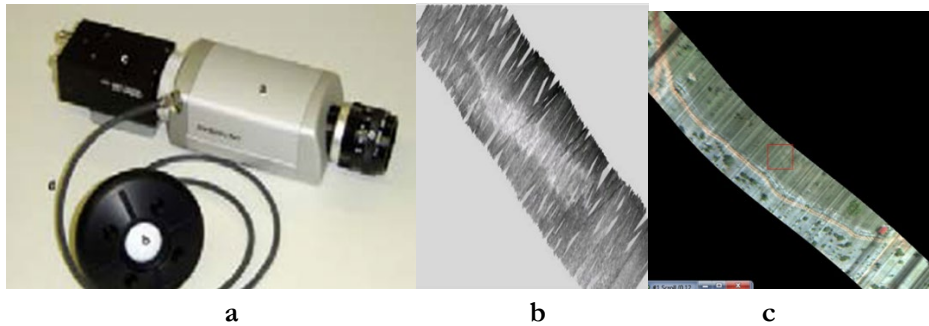


Figure 1: A hyperspectral push-broom sensor provides real hyperspectral data, while complex processing is needed to produce a calibrated hyperspectral cube. a) Hyperspectral line scanner, b) data are collected in scan lines, which need parametric geocoding, and c) hyperspectral cube needs calibration.

Source: own

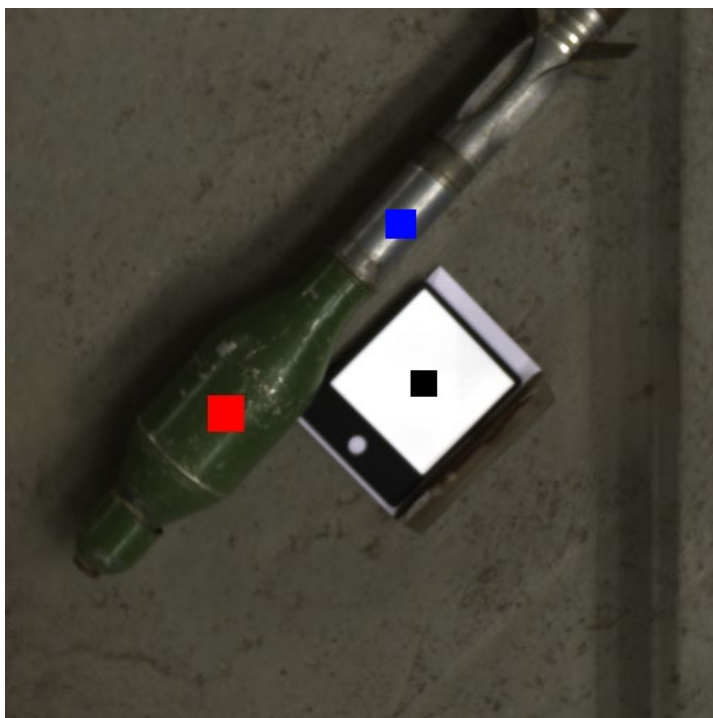
Another sensor used in this study is **Specim IQ**. It is a portable carry-on hyperspectral camera that contains features needed for hyperspectral data capturing, data processing, and result visualization. It covers wavelengths between 400 and 1000 nm. It has 204 spectral bands. Image resolution is 512 x 512 pixels, while spectral resolution is 7 nm. Data output is stored in 12-bit format. A full Field Of View (FOV) is 31 x 31 degrees, whereat FOV at 1 m is 0.55 x 0.55 m. It supports WiFi and GPS. The camera has 32 GB SD memory card.



Figure 2: Handheld hyperspectral imaging camera Specim IQ.

Source: own

In our study, the Specim Inspector V9 and Specim IQ hyperspectral cameras are compared by measuring UXO on Friday 11<sup>th</sup> of February 2024. A white balance card and a mortar mine were chosen for measurements (see Figure 3). Raw values from the Specim Inspector V9 line scanner and Specim IQ are presented in all subsequent figures. Figure 4 depicts measurements for the white balance card. On y axis, there are sensor digital output values (digital numbers) that need to be calibrated to get standardized measures of radiance or reflectance. A wavelength is denoted on x axis. It should be stressed that a wavelength range of IQ camera is cut to match a range of V9 camera. Three squares in Figure 3 denote locations, where a material was sampled (measured). Black square denotes a location on the white balance card, while the red and blue squares denote locations on a green and grey part of the mortar mine (UXO). The values shown in the graphs were determined by averaging the pixels within the square at particular wavelength.



**Figure 3: Mortar mine and white balance card. A red rectangle denotes a green part, while a blue rectangle denotes a grey part of mortar mine. Three squares (red, blue and black) denote an approximate location, where a material was measured.**

Source: own

We used white balance card delivered with Specim IQ camera to test spectral response of both sensors. Lighting conditions were the same for both cameras. Sensor output values by the same wavelengths differ, of course, because an exposure time is not the equal for both cameras (see Figure 4). Stronger amplitudes are always noticed by V9 camera. The range of wavelengths between 530 nm and 630 nm is very interesting, because the V9 output stays almost constant. Higher amplitudes are noticed for wavelengths between 730 nm and 830 nm by V9 compared to IQ camera as well.

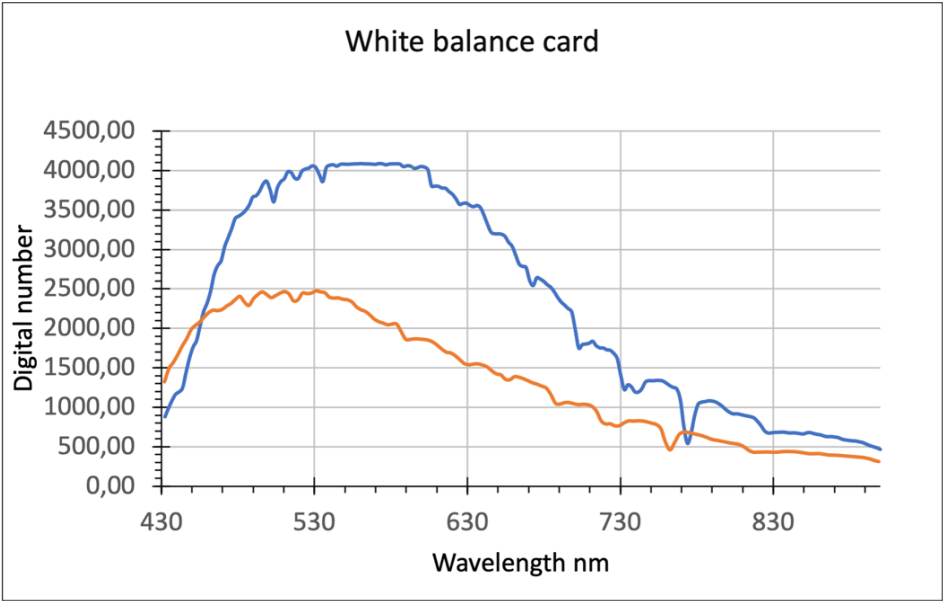


Figure 4: Uncalibrated outputs of Specim IQ (orange) and Specim Inspector V9 (blue) hyperspectral cameras by measuring the white balance card.

Source: own

We measured the mortar mine in the next experiments. First, the green part of the mortar mine was selected. This is a pretty realistic scenario, described also in (Ammunition Storage Area Explosions – EOD Clearance, 2021) about ammunition depot explosions. One can expect to find all kind of UXOs in grass or other green surroundings. Thus, it important to see whether we would be able to separate the mine from the environment characterized by vegetational jump around wavelength 700 nm. Figure 5 depicts outputs for both tested hyperspectral cameras.

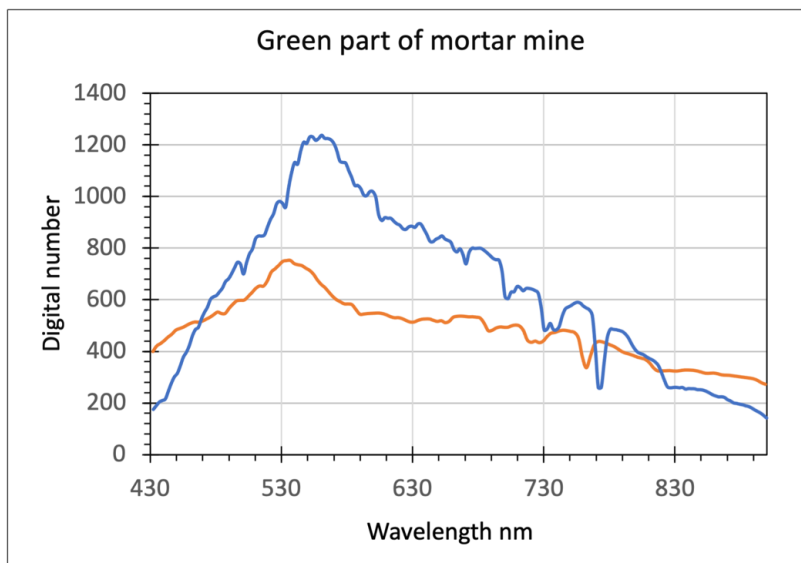


Figure 5: Uncalibrated outputs of Specim IQ (orange) and Specim Inspector V9 (blue) hyperspectral cameras by measuring the green part of the mortar mine.

Source: own



Figure 6: Uncalibrated outputs of Specim IQ (orange) and Specim Inspector V9 (blue) hyperspectral cameras by measuring the grey part of the mortar mine

Source: own

Hyperspectral sensors are passive devices, relying on spectral response rather than on chemical properties of the materials. Thus, it is important by endmembers selection to inspect also colour variations within the same object. Our test subject (mine) was new, with no traces of use or rust. Therefore, we wanted to check if sensors' responses are different for grey part of the mortar mine compared to the responses for the green part. Figure 6 depicts graphs for both studied sensors.

### **3 Discussion and conclusion**

In a fundamental work by Manolakis, (Manolakis, 2005), endmember selection is very clearly defined. If there are two spectra from the same target, one should find mean vector of the existing spectra or use other endmembers selection technique. Very important consideration is also that the number of used pixels should be larger than the number of spectral channels used, with recommendation to use from 10 up to 100 times more pixels than channels to obtain statistically reliable estimate of the covariance matrix. It is worth to mention that these suggestions are from a remote sensing domain, based on satellite imagery and a high inter-pixel variability.

With our experiments we wanted to inspect a spectral response of both sensors for the white balance card and the UXO. Results are as expected. The size of our targets is approximately 10,000 pixels when viewed from a distance of 1 metre from the ground. At the altitude of detection procedures, which is around 30 metres above the ground, the size is at least 10 pixels depending on the target. If the target is smaller than one pixel or mixed with spectra from other materials, our recommendations for endmember selection may not be applicable or may need to be adjusted. Authors in (Behmann et al., 2018) for comparison of IQ and V10E cameras used 400 pixels and, afterwards, calculated mean reflectance, standard deviation, and the mean and maximum spectral distance. To the best of our knowledge, there are no clear suggestions in the literature on how many pixels to consider by statistical analysis, taking into account a variability of the terrain and a size of the target. Especially, when performing hyperspectral detections on targets with their size on the image between 1 pixel and 10 to 100 times of number of channels. The conclusions presented in this sequel are based on our measurements and knowledge. When constructing a statistically reliable endmembers dataset, it is important to adhere to the following principles:

- Different parts of the target must be included,
- Sample area should be of sufficient size,
- When reducing data dimensionality, it is important to exercise caution when using principal component analysis for wavelength selection,
- Consider including a spectral graph of the endmembers to visually confirm similarities, even if there is a shift in wavelength.

### **Viri in literature**

- Ammunition storage area explosions – EOD clearance. (2021). United Nations Office for Disarmament Affairs.
- Bajić, M., & Bajić, M. (2021). Modeling and Simulation of Very High Spatial Resolution UXOs and Landmines in a Hyperspectral Scene for UAV Survey. *Remote Sensing*, 13(5), 837. <https://doi.org/10.3390/rs13050837>
- Bajić, M., Ivelja, T., Krtalić, A., Tomić, M., & Vuletić, D. (2013). The multisensor and hyper spectral survey of the UXO around the exploded ammunition depot, of the land mines test site vegetation. *Proceedings 10th International Symposium HUDEM*, ISSN, 9206, 91–96.
- Behmann, J., Acebron, K., Emin, D., Bennertz, S., Matsubara, S., Thomas, S., Bohnenkamp, D., Kuska, M., Jussila, J., Salo, H., Mahlein, A.-K., & Rascher, U. (2018). Specim IQ: Evaluation of a New, Miniaturized Handheld Hyperspectral Camera and Its Application for Plant Phenotyping and Disease Detection. *Sensors*, 18(2), 441. <https://doi.org/10.3390/s18020441>
- ElMasry, G., & Sun, D.-W. (2010). Principles of Hyperspectral Imaging Technology. In *Hyperspectral Imaging for Food Quality Analysis and Control* (pp. 3–43). Elsevier. <https://doi.org/10.1016/B978-0-12-374753-2.10001-2>
- Manolakis, D. G. (2005). Taxonomy of detection algorithms for hyperspectral imaging applications. *Optical Engineering*, 44(6), 066403.
- National University of Public Service, Hungary, & Ember, I. (2021). The role and the risks of explosive ordnance decontamination in Hungary. *Science & Military*, 16(1), 32–42. <https://doi.org/10.52651/sam.a.2021.1.32-42>
- Roberts, S., & Williams, J. (1995). *After the guns fall silent: The enduring legacy of landmines*. Veterans of America Foundation, Washington D.C., USA.



Tukaj smo mi. Slovenski podjetniški sklad.

Da vam svetujemo, pomagamo, da vas podpiramo.  
Da vam prisluhujemo in da vas slišimo.

| [podjetniskisklad.si](https://podjetniskisklad.si)

**Slovenski podjetniški sklad (SPS) je osrednji finančni sklad za podporo start-up in scale-up podjetij in MSP-jev z ukrepi za zmanjševanje različnih vrzeli, ki so jih podjetja deležna na trgu.**

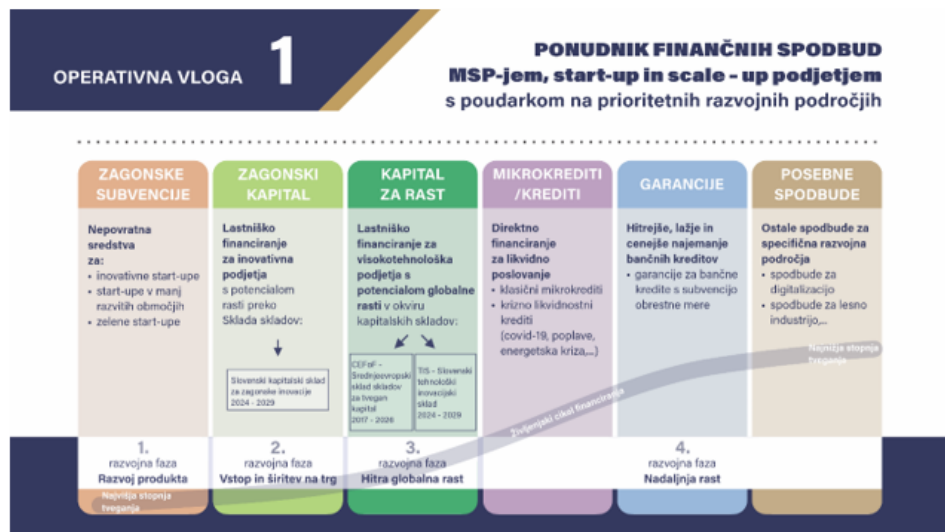
SPS tudi v letu 2024 igra pomembno vlogo v slovenskem podjetniškem okolju, saj nastopa kot:



Z različnimi spodbudami ne le pomaga MSP-jev pri tekočem poslovanju ali rasti podjetja, temveč tudi spodbuja iskanje rešitev za globalne izzive, **od podnebnih sprememb in trajnostne energije, do zdravega življenja in hrane**. Doprinaša tudi k **digitalizaciji** in **naprednim digitalnim tehnologijam** naših MSP-jev.

## V letu 2024 SPS skupno ponuja:

- o → **okoli 127 mio EUR** novih pomoči
- o → **za cca 3.000 podjetij**, ki bodo v letu 2024 lahko koristila pomoč SPS-a

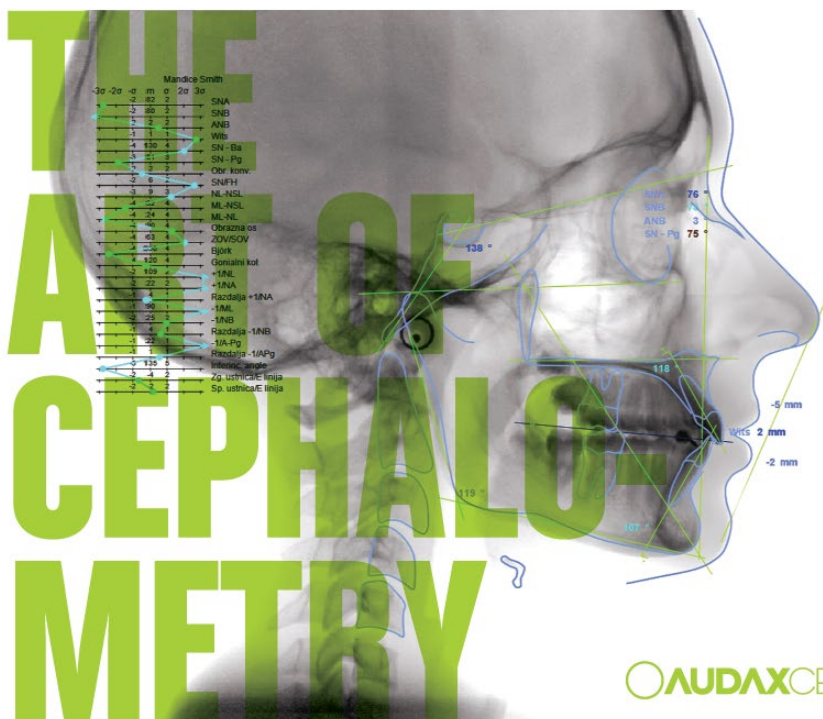


Poleg omenjenih spodbud bo SPS še naprej sklepal tudi strateška partnerstva za **koriščenje EU virov** in **strateška partnerstva s podjetniškimi mrežami v EU in mednarodnem okolju** z namenom zagotavljanja pomoči MSP-jem pri internacionalizaciji – še zlasti povezovanje, spodbujanje k prenosu znanja in inovacij na tuje trge, iskanje novih razvojnih sodelovanj in razvoj skupnih projektov na mednarodni ravni.

**Vabimo vas, da se na spletni strani [www.podjetniskisklad.si](http://www.podjetniskisklad.si) prijavite na E-Notice in si zagotovite tekoče informacije o aktualni ponudbi različnih oblik pomoči.**

Rastemo skupaj!

Slovenski podjetniški sklad



# AudaxCeph. A new paradigm of computer-based cephalometric analyses.

Use only the best-of-breed tools in your work. AudaxCeph persuades with its high-end performance, numerous features and ease of use.

The AudaxCeph software suite includes tools which are required for creating cephalometric analyses, superimposition and planning orthodontic treatment. It provides flexible and easy-to-use tools to quickly and efficiently process even the most difficult cases in orthodontics, dentistry and maxillofacial surgery. AudaxCeph offers an interface in many languages and can be easily translated to any language using Latin or Cyrillic alphabet. Technical support is available and can be provided on-line.



[www.audaxceph.com](http://www.audaxceph.com)  
[info@audaxceph.com](mailto:info@audaxceph.com)  
 AUDAX (fax) - 386 40 40 40 40  
 tel. +386 1 501 40 10 - fax. +386 1 42 47 00



# ROSUS 2024 - RAČUNALNIŠKA OBDELAVA SLIK IN NJENA UPORABA V SLOVENIJI 2024: ZBORNİK 18. STROKOVNE KONFERENCE

BOŽIDAR POTOČNIK (UR.)

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Inštitut  
za računalništvo, Maribor, Slovenija  
bozidar.potocnik@um.si

ROSUS 2024 – Računalniška obdelava slik in njena uporaba v Sloveniji 2024 je strokovna računalniška konferenca, ki jo od leta 2006 naprej vsako leto organizira Inštitut za računalništvo iz Fakultete za elektrotehniko, računalništvo in informatiko, Univerze v Mariboru. Konferenca povezuje strokovnjake in raziskovalce s področij digitalne obdelave slik in strojnega vida z uporabniki tega znanja, pri čemer uporabniki prihajajo iz raznovrstnih industrijskih okolij, biomedicine, športa, zabavnštva in sorodnih področij. Zbornik konference ROSUS 2024 združuje strokovne prispevke več avtorjev, od tega dve vabljeni predavanji ter več demonstracijskih prispevkov. Prispevki podajajo najnovejše dosežke slovenskih strokovnjakov s področij digitalne obdelave slik in strojnega vida, osvetljujejo pa tudi trende in novosti na omenjenih strokovnih področjih. Velik poudarek prispevkov je na promoviranju ekonomske koristnosti aplikacij računalniške obdelave slik in vida v slovenskem prostoru. Takšne računalniške aplikacije zaradi visoke natančnosti, robustnosti in izjemnih hitrosti pri obdelovanju informacij nudijo namreč nove priložnosti za uveljavitev na trgu visokih tehnologij.

DOI  
[https://doi.org/  
10.18690/um.feri.1.2024](https://doi.org/10.18690/um.feri.1.2024)

ISBN  
978-961-286-837-6

**Ključne besede:**  
računalniška obdelava slik,  
strojni vid,  
biomedicina,  
industrijske aplikacije,  
prenos znanja



Univerzitetna založba  
Univerze v Mariboru

DOI  
[https://doi.org/  
10.18690/um.feri.1.2024](https://doi.org/10.18690/um.feri.1.2024)

ISBN  
978-961-286-837-6

# ROSUS 2024 – COMPUTER IMAGE PROCESSING AND ITS APPLICATION IN SLOVENIA 2024: PROCEEDINGS OF THE 18<sup>TH</sup> PROFESSIONAL CONFERENCE

BOŽIDAR POTOČNIK (ED.)

University of Maribor, Faculty of Electrical Engineering and Computer Science, Institute  
of Computer Science, Maribor, Slovenia  
[bozidar.potocnik@um.si](mailto:bozidar.potocnik@um.si)

**Keywords:**

evolutionary algorithm,  
deep networks,  
image classification,  
generating deep networks,  
optimization

ROSUS 2024—Computer image processing and its application in Slovenia 2024 is a professional conference that, since 2006, has been organised every year by the Institute of Computer Science of the Faculty of Electrical Engineering and Computer Science, University of Maribor. The conference connects researchers in the fields of Image Processing and Machine Vision with users of this knowledge, whereby users are coming from diverse industrial environments, such as Biomedicine, Sport, Entertainment, and related fields. The proceedings of ROSUS 2024 combine scientific articles by dozens of authors, including two invited lectures and several demonstration articles. Contributions represent the latest achievements of Slovenian experts in the fields of Image Processing and Vision, and also highlight trends and novelties in these areas. Great emphasis is on promotion of the economic usefulness of Image Processing and Vision applications in the Slovenian region. Namely, such software, due to high precision, robustness, and exceptional speed in information processing, provides new opportunities for penetration on the high technologies market.



University of Maribor Press





Univerza v Mariboru

---

Fakulteta za elektrotehniko,  
računalništvo in informatiko

 **LSPO**

Laboratorij za sistemsko  
programsko opremo

<https://rosus.feri.um.si/rosus2024>  
[rosus.feri@um.si](mailto:rosus.feri@um.si)

