

Avtomatizirano sortiranje konektorjev Molex z uporabo strojnega vida in robota Franka Emika Panda

Rok Casar, Rok Pahič, Mitja Golob, Aleš Hace

Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru
E-pošta: rok.casar2@um.si

Automated Sorting of Molex Connectors Using Machine Vision and the Panda Robot

Abstract. This paper presents a modular robotic system for visual classification and sorting of Molex connectors using the Franka Emika Panda robot. A RGB camera captures images of moving connectors, which are classified as male or female using a ResNet18-based convolutional neural network trained on a custom dataset. A key focus of the study is to evaluate how classification accuracy depends on the size of the training dataset and to determine the threshold at which additional data no longer significantly improves model performance. Classified connectors are sorted via pick-and-place operations, with system components synchronized through SNAP7 communication between Python and a Siemens S1500 PLC. The proposed approach enables reliable and responsive operation and is applicable in both educational and industrial environments. Experiments show that with only 50 images per class, the model achieves over 94% accuracy, with diminishing improvements beyond 100 images per class.

1 Uvod

V industriji narašča potreba po avtomatizaciji prepoznavanja in sortiranja objektov, pri čemer se kot učinkovit pristop uveljavlja strojni vid z uporabo nevronske mreže. Pomembno vprašanje pri takih sistemih je, koliko učnih podatkov je potrebnih za zanesljivo klasifikacijo ter do katere mere lahko prenos učenja iz prednaučenih modelov nadomesti njihovo pomanjkanje.

Cengiz et al. [1] sta pokazala, da splošno sprejeti domnevi, da več podatkov prinese boljše rezultate in da prenos učenja (angl. transfer learning) pomaga, kadar podatkov ni dovolj, ne veljata vedno. Poudarjata, da lahko pride do točke nasičenja, kjer dodatni podatki ne prispevajo več k izboljšanju, ali celo do poslabšanja rezultatov zaradi neustreznega prenosa učenja. Podobno so Althman et al. [2] na več medicinskih podatkovnih zbirkah preučevali vpliv velikosti učnega nabora na uspešnost različnih klasifikacijskih modelov ter ugotovili, da je za uspešnost pogosto pomembnejša reprezentativnost podatkov kot njihovo absolutno število.

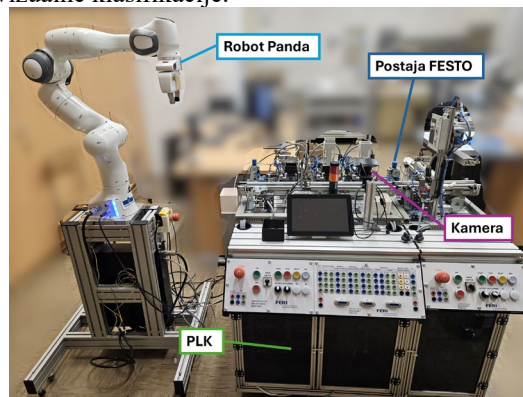
V tej raziskavi smo te ugotovitve preverili na konkretni industrijski aplikaciji, in sicer na sistemu za vizualno klasifikacijo in sortiranje konektorjev Molex, ki predstavljajo enega najpogostejše uporabljenih

standardov v avtomatizirani proizvodnji elektronskih naprav. Zaradi podobne oblike in bele barve so za strojni vid zahtevni za zanesljivo razločevanje, kar predstavlja ustrezen izziv za testiranje učinkovitosti prenosa učenja in vpliva velikosti učnega nabora. Sistem smo razvili kot avtomatizirano rešitev, ki v realnem času omogoča klasifikacijo in sortiranje konektorjev na podlagi slikovnih podatkov, pridobljenih med delovanjem transportnega sistema.

2 Uporabljena oprema

Eksperimentalna postavitve, prikazana na sliki 1, vključuje robota Franka Emika Panda, FESTO postajo s PLC-jem ter RGB kamero, ki skupaj tvorijo modularni sistem za vizualno klasifikacijo in manipulacijo industrijskih konektorjev.

- **Robot Franka Emika Panda** je 7-osni kolaborativni robot z vgrajenimi senzorji navara v vseh sklepih, kar omogoča zaznavanje sil in varno interakcijo z okoljem. Uporablja se s prijemalom, ki omogoča manipulacijo objektov do 3 kg. Zaradi odprtokodne podpore (ROS, C++, Python) je zelo primeren za raziskovalne aplikacije.
- **Postaja FESTO s PLC Siemens S1500** je didaktična delovna postaja s transportnim trakom, senzorji in aktuatorji, ki simulira industrijsko okolje. PLC skrbi za krmiljenje traku, zaznavanje objektov in izvajanje ustreznih ukrepov glede na stanje na traku.
- **RGB kamera** je komercialna spletna kamera z ločljivostjo 1080p, fiksno nameščena nad trakom pod ustreznim kotom. Omogoča neprekinjen zajem slik konektorjev v realnem času za potrebe vizualne klasifikacije.



Slika 1. Eksperimentalna postavitve - robot Franka Emika Panda in postaja FESTO.

3 Programska arhitektura sistema

3.1 Knjižnica Panda_py

Za izvajanje gibanj robota Franka Emika Panda v tej aplikaciji uporabljamo odprtokodno knjižnico Panda_py [3], ki omogoča programiranje robotskih nalog v okolju Python. Knjižnica temelji na visokonivojskem upravljanju gibanj in uporablja lastno abstrakcijo nad knjižnico libfranka, kar olajša razvoj eksperimentalnih in raziskovalnih aplikacij.

3.2 Komunikacija med Python in PLK (Snap7)

Za komunikacijo med nadzornim programom v okolju Python in programirljivim logičnim krmilnikom Siemens S1500 uporabljamo odprtokodno knjižnico Snap7. Ta omogoča branje in zapis podatkov v podatkovne bloke ter druge pomnilniške segmente PLK-ja prek TCP/IP omrežja z uporabo Siemensovega protokola S7 [4]. Takšna povezava omogoča izmenjavo signalov med PLK-jem in Python skripto ter zagotavlja sinhronizacijo med transportnim sistemom, senzorji in robotsko manipulacijo.

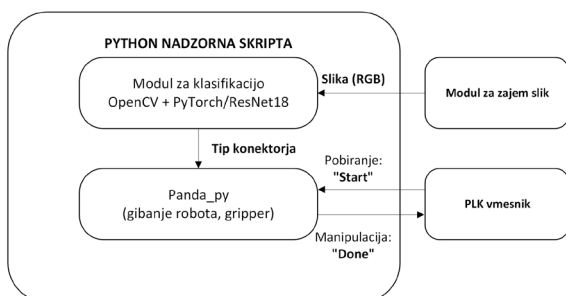
3.3 Nadzor informacijskega toka in sinhronizacija

Na sliki 2 je prikazana komunikacija med glavnimi programskimi komponentami sistema, ki omogoča usklajeno izvajanje naloge. Osrednji element predstavlja Python skripto, ki nadzoruje zaporedje procesov in vključuje dve komponenti:

- Modul za razpoznavo objekta, kjer se na podlagi slike in nevronske mreže ResNet18 izvede klasifikacija konektorja in
- komponento Panda_py, namenjeno gibanju robota in upravljanju prijemala

Zajem slike poteka v ločeni komponenti, ki posreduje slikovne podatke klasifikacijskemu modulu. Po prepoznavi tipa konektorja skripto sproži ustrezno manipulacijo robota.

Za sinhronizacijo z okoljem skripto komunicira s PLK-jem, ki na podlagi senzorjev sproži manipulacijo.



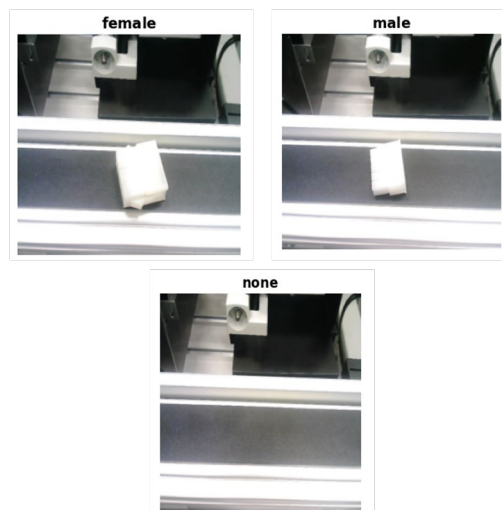
Slika 2. Shema komunikacije med programskimi komponentami.

4 Analiza učne učinkovitosti ResNet18 glede na količino vhodnih podatkov

4.1 Opis modela in pristopa

Treniranje globokih nevronske mreže iz nič zahteva obsežne podatkovne zbirke in veliko računsko moč. Učinkovitejša alternativa je prenos učenja (angl. transfer learning), kjer izkoristimo že naučeno znanje modela, ki je bil predhodno treniran na velikem naboru podatkov (npr. ImageNet), ter ga prilagodimo novemu cilju.

V našem primeru uporabljamo arhitekturo ResNet18, to je rezidualno nevronske mreže z 18 plastmi, kot osnovni model. Za nalogo klasifikacije v treh razredih (slika 3) prilagodimo le zaključne sloje modela, medtem ko preostali deli mreže ostanejo nespremenjeni. Takšen pristop omogoča prilagajanje modela na nove podatke tudi v primeru omejenega števila učnih primerov.



Slika 3. Primeri slik iz učnega nabora za tri razrede (female, male, none).

4.2 Eksperimentalna zasnova preizkusa učne učinkovitosti

Glavna hipoteza eksperimenta je, da lahko z uporabo prednaučenega modela ResNet18 dosežemo zadovoljivo natančnost klasifikacije že z razmeroma majhnim številom učnih primerov. Cilj testa je preučiti, kako se natančnost klasifikacije spreminja glede na velikost učnega nabora in določiti prag, pri katerem dodatni podatki ne prinesejo več bistvenega izboljšanja rezultatov.

4.3 Konfiguracije učnega nabora

Za potrebe eksperimenta smo pripravili več uravnoteženih podmnožic učnega nabora, ki so se razlikovale po številu učnih primerov na razred. Uporabili smo 30 različnih konfiguracij, od 10 do 300 slik na razred, s korakom po 10 slik. V vseh primerih je testni nabor ostal nespremenjen, podmnožice pa so bile naključno izbrane brez prekrivanja z validacijskim in

testnim naborom, da smo zagotovili nepristranskost. Vsako konfiguracijo učnega nabora smo eksperiment večkrat ponovili z različnimi naključnimi izbori, da smo zmanjšali vpliv posameznih kombinacij slik in pridobili bolj stabilno oceno uspešnosti. Za zagotavljanje ponovljivosti rezultatov smo pri vseh postopkih, ki vključujejo naključne izbire (npr. izbor podmnožic, razdelitev podatkov), uporabili fiksno seme naključnega generatorja.

4.4 Strategija delitve podatkov

Za vse eksperimentalne primere smo uporabili enako strategijo delitve podatkov: 70% učnih primerov, 15% validacijskih primerov in 15% testnih primerov. Takšna delitev je skladna s priporočili iz literature, kjer se pogosto uporablja kot kompromis med količino podatkov za učenje in količino, potrebno za neodvisno oceno modela [5].

4.5 Nastavitve učenja

Za vse eksperimente smo uporabili enake nastavitve učnega postopka, da bi zagotovili primerljivost rezultatov in omogočili osredotočeno analizo vpliva velikosti učnega nabora. Uporabljeni parametri so naštet v tabeli 1. Uporabili smo tudi zgodnje ustavljanje, če se natančnost na validacijskem naboru ni izboljšala pet epoh zapored. S tem se prepreči prekomerno prilagajanje modela (angl. overfitting) in skrajša čas učenja, če nadaljnje izboljšave niso več verjetne.

Tabela 1. Uporabljene nastavitve učnega postopka

Parameter	Vrednost
Batch size	8
Optimizator	Adam (začetna hitrost učenja)
Prilagojeni sloji	Zaključni sloj
Prednaučen model	ResNet18 (ImageNet)
Funkcija izgube	Cross-Entropy Loss
Strategija ustavljanja	Early stopping (patience = 5)

4.6 Evalvacija uspešnosti modela

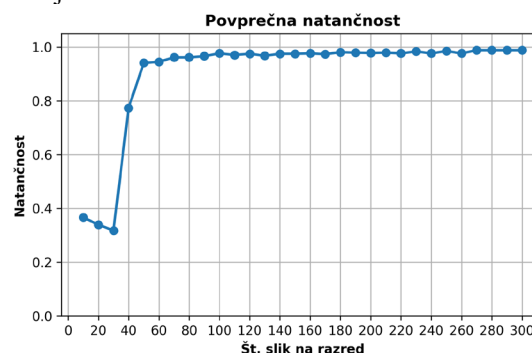
Za oceno uspešnosti klasifikacijskega modela smo uporabili natančnost (angl. accuracy), ki meri delež pravilno klasificiranih primerov glede na vse napovedi. Ta metrika je ustrezna za naš primer, saj imamo uravnotežene razrede in nas zanima celotna učinkovitost modela. Natančnost je definirana z enačbo (1) [6]:

$$\text{natančnost} = \frac{\text{pravilne napovedi}}{\text{skupno število napovedi}} \quad (1)$$

4.7 Rezultati in analiza

Analiza učne krivulje (slika 4) pokaže razvoj natančnosti glede na velikost učnega nabora. Pri zelo majhnih vzorcih (10-30 slik na razred) opazimo nizko in nestabilno natančnost (0.31-0.36), kar kaže na nezadostno količino podatkov za zanesljivo učenje. Opazimo, da je natančnost pri 10 slikah nekoliko višja kot pri 30, kar lahko pripišemo naključni variabilnosti pri zelo majhnih vzorcih.

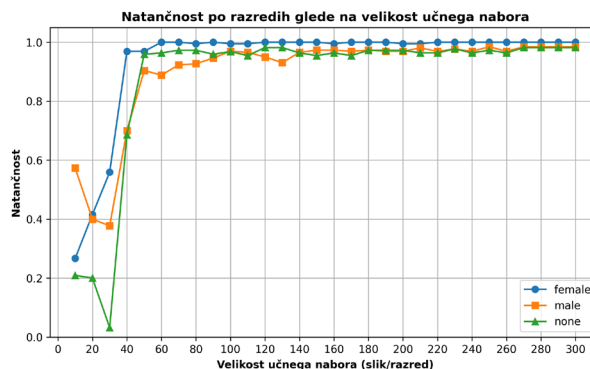
Prelom se pojavi pri 40 slikah na razred, kjer natančnost skokovito naraste na 0.77. To predstavlja prag, pri katerem postane količina podatkov zadostna, da se lahko učinkovito izrazi predhodno znanje iz prednaučenega modela ResNet18, saj so pogoji bolj primerni za stabilno prilagoditev zadnjih slojev klasifikatorja. Že pri 50 slikah na razred natančnost doseže 0,94, kar pomeni, da model pri tej količini podatkov dosega zelo visoko natančnost, ki je za večino klasifikacijskih nalog že povsem uporabna. Krivulja se pri 100 slikah na razred splošči, kar kaže na točko nasičenja.



Slika 4. Natančnost glede na velikost učnega nabora.

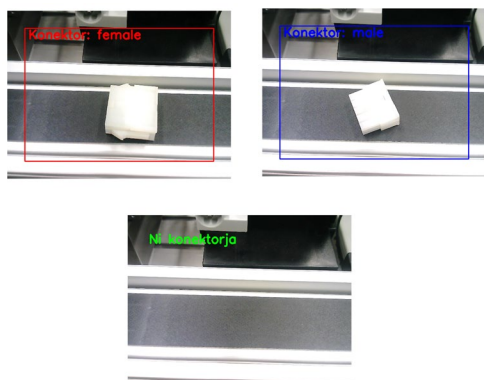
Za dopolnitev povprečne analize smo preverili tudi, kako se natančnost spreminja po posameznih razredih. Slika 5 prikazuje natančnost modela posebej za vsak razred glede na velikost učnega nabora. Pri zelo majhnih količinah podatkov (10–30 slik na razred) so med razredi opazne razlike. Najslabše je prepoznaven razred none, ki pri 30 slikah doseže le 0,032, medtem ko male doseže 0,37 in female 0,55. To kaže, da je model sprva nezanesljiv in različno uspešen pri klasifikaciji posameznih razredov.

Z večanjem količine podatkov se natančnost vseh razredov izboljšuje. Pri 50 slikah na razred vsi trije razredi dosežejo natančnost nad 0,9, kar pomeni, da model začne zanesljivo prepoznavati tudi zahtevnejše primere. Od 100 slik na razred naprej razlike med razredi postanejo zanemarljive, pri čemer natančnost v razredu female ostaja stabilno pri 1, medtem ko razreda male in none nihata med 0,95 in 0,98.



Slika 5. Natančnost po razredih glede na velikost učnega nabora.

Slika 6 prikazuje primer rezultatov klasifikacije za vse tri razrede, kot jih zazna model na vhodni sliki.



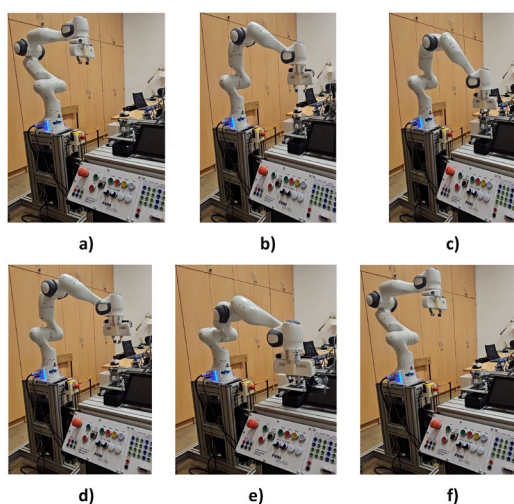
Slika 6. Primer rezultatov klasifikacije konektorjev v realnem času.

4.8 Zaporedje gibov in izvedba manipulacije

Po analizi natančnosti modela na ravni celotnega nabora in posameznih razredov smo preverili tudi, kako se klasifikacija prenese v praktično izvedbo robotske manipulacije. Ugotovljeno je bilo, da pri zadostni količini učnih primerov ($\geq 50/\text{razred}$) sistem zanesljivo ločuje razrede v realnem času, kar omogoča nadaljevanje procesa s fizično manipulacijo objekta.

Robotska manipulacija temelji na zaporedju vnaprej določenih točk v prostoru, podanih kot homogene transformacije, ki določajo lego in orientacijo vrha robota. Te točke ustrezajo posameznim fazam manipulacije (slika 7):

- (a) začetna pozicija,
- (b) premik nad konektor,
- (c) spuščanje do konektorja in zapiranje prijemale,
- (d) dvig konektorja,
- (e) premik nad ciljno mesto in odlaganje,
- (f) povratek v začetno lego.



Slika 7. Zaporedje gibov robota pri manipulaciji konektorja (a-f).

5 Zaključek

V nalogi smo predstavili sistem za vizualno klasifikacijo in sortiranje konektorjev Molex, ki združuje strojni vid, konvolucijsko nevronske mreže ResNet18 in robotsko manipulacijo. Osrednja ugotovitev je, da za zanesljivo delovanje sistema ni potrebna velika količina podatkov, model že pri 50 slikah na razred doseže natančnost nad 94%.

Analiza učne krivulje pokaže, da se natančnost pri majhnem številu primerov močno izboljšuje z dodajanjem podatkov, nato pa se okoli 100 slik na razred stabilizira. Model dobro izkoristi prednaučeno znanje, kar omogoča učinkovito učenje tudi pri omejenem številu podatkov. Ena izmed metodoloških omejitev je uporaba enake strategije delitve podatkov (70/15/15), pri kateri se z večanjem učnega nabora sorazmerno povečata tudi validacijski in testni del. To lahko vpliva na stabilnost ocene natančnosti, saj večje validacijske množice zmanjšajo variabilnost rezultatov. Analiza po posameznih razredih potrди, da se razlike med njimi z večanjem nabora postopno izničijo, kar kaže na robustno in uravnoteženo delovanje klasifikatorja.

Rezultati potrjujejo ugotovitve Cengiz et al. [1] in Althnian et al. [2], da vpliv količine podatkov na uspešnost modela ni linearen ter da kakovost učnih primerov pogosto igra pomembnejšo vlogo kot njihova številčnost. V našem primeru se je točka nasičenja pojavila že pri razmeroma nizkem številu primerov, kar kaže, da je lahko tudi majhen, a dobro izbran učni nabor dovolj za učinkovito učenje s prenosom znanja.

Literatura

- [1] A. B. Cengiz and A. S. McGough, "How much data do I need? A case study on medical data," *arXiv preprint*, arXiv:2311.15331, Nov. 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2311.15331>
- [2] A. Althnian, M. Almalki, S. Aljaam, M. Zemerly, and A. Alfakheh, "Impact of Dataset Size on Classification Performance: An Empirical Evaluation in the Medical Domain," **Applied Sciences**, vol. 11, no. 2, p. 796, Jan. 2021. [Online]. Available: <https://doi.org/10.3390/app11020796>
- [3] J. Elsner, "panda-py – Python interface for controlling Franka Emika Panda," GitHub repository, [Online]. Available: <https://github.com/JeanElsner/panda-py>. [Accessed: Jun. 30, 2025]
- [4] D. Nardella, "Snap7: An open source Siemens S7 communication suite," [Online]. Available: <https://snap7.sourceforge.net/> [Accessed: Jun. 30, 2025]
- [5] J. Brownlee, *Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python*, Machine Learning Mastery, 2020.
- [6] Evidently AI, "Classification Metrics: Accuracy, Precision, Recall," [Online]. Available: <https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall>. [Accessed: Jul. 5, 2025]