



FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE  
IN INFORMACIJSKE TEHNOLOGIJE

RAČUNALNIŠTVO IN INFORMATIKA

PROJEKTNI SEMINAR

# Aplikacija za prepoznavanje licitacije pri bridžu

TEHNIČNA DOKUMENTACIJA

*Avtorji:*

Dani Zupan  
Žan Peternelj  
Jani Suban

*Mentor:*

dr. Peter Rogelj

16. september 2023

# Kazalo

<b>1</b>	<b>Predstavitev</b>	<b>2</b>
<b>2</b>	<b>Analiza</b>	<b>2</b>
<b>3</b>	<b>Definicija zahtev</b>	<b>3</b>
<b>4</b>	<b>Tehnologije</b>	<b>4</b>
4.1	Izbira tehnologije za razvoj . . . . .	4
4.2	YOLO . . . . .	4
4.3	Python . . . . .	4
4.4	OpenCV . . . . .	5
<b>5</b>	<b>Generiranje učnih podatkov</b>	<b>6</b>
<b>6</b>	<b>Implementacija</b>	<b>6</b>
<b>7</b>	<b>Razdelitev dela</b>	<b>7</b>

# 1 Predstavitev

V sklopu projektnega seminarja smo si za nalogo izbrali razvoj aplikacije namenjene prepoznavi kart v fazi licitacije pri igri bridž, ki velja za najbolj igrano igro s kartami na svetu. Bridž je igra s kartami, ki jo igrajo štirje igralci, razdeljeni v dve partnerstvi. Običajno se igra s standardnim kompletom 52 kart.

Igra je sestavljena iz več faz:

- **Razdelitev kart:** Delivec premeša karte in jih razdeli vsakemu igralcu posebej, začeni z njegove leve strani. Vsak igralec prejme 13 kart.
- **Licitiranje:** Igralci izmenično dajejo ponudbe, da bi določili pogodbo za igro. Ponudba vključuje navedbo števila štihov, ki jih želi partnerstvo dobiti, in izbiro barve ali možnosti "brez trumfov". Najvišja ponudba postane pogodba in igralec, ki jo je ponudil, je izjavljalec.
- **Igra:** Partner izjavitelja, imenovan lutka, položi svoje karte na mizo z licem navzgor. Oskrbnik igra svoje karte in karte lutke, da bi izpolnil pogodbo z določenim številom štihov. Drugi partner mu to poskuša preprečiti s strateškim igranjem svojih kart.
- **Točkovanje:** Ko so odigrani vsi štiki, se rezultat izračuna glede na to, ali je izjavljalcu uspelo izpolniti pogodbo. Točke se podeljujejo za dosežene štike, bonusne točke pa je mogoče dobiti za doseganje določenih ciljev, kot sta šlam ali "igra".

V fazi licitiranja v bridžu igralci izmenično licitirajo, da bi določili pogodbo za igro. Cilji so določiti, koliko štihov želi partnerstvo dobiti (nivo), določiti barvo ali "no trump" in izbrati izjavljalca. Licitiranje vključuje sporočanje moči in razporeditve kart s posebnimi licitacijami in konvencijami. Igralec z najvišjo ponudbo postane izjavljalec in je odgovoren za izpolnitev pogodbe, njegov partner pa postane lutka. Uspešno licitiranje zahteva strateško komunikacijo med partnerjema. Rešitev smo implementirali s pomočjo tehnologije konvolucijskih nevronske mreže, ki omogoča učinkovito in zanesljivo zaznavo objektov. Ker se pri bridž licitaciji uporablja posebne igralne karte, za rešitev nismo mogli uporabiti obstoječih implementacij za zaznavo kart temveč smo model naučili z uporabo lastne učne množice. Ta je sestavljena iz slik kart zajetih v nadzorovanem okolju. Vsaka različna karta je predstavljena s svojim razredom, vsega skupno 38 razredov. Aplikacijo se bi lahko potencialno integriralo v proces bridž igre znotraj obsežnejše rešitve, ki bi nadomestila ročno beleženje potez igralcev s samodejnim beleženjem.

## 2 Analiza

Detekcija kart se lahko kot problem posploši na zaznavo objektov (ang. object detection), področje znotraj računalniškega vida in obdelave slik. Zaznava objektov obsega iskanje in klasifikacijo objekta v enega izmed množice razredov. Tehnologija, ki v zadnjem obdobju prevladuje pri zaznavi objektov so nevronske mreže. V primerjavi s tradicionalnimi tehnikami računalniškega vida in obdelave slik so nevronske mreže v prednosti zaradi višje mere prilagodljivosti in visoke natančnosti klasifikacije. Kvaliteta modela nevronske mreže je pogojena s količino in kvaliteto učnih podatkov. Pomembno je, da učni podatki obsegajo velik nabor različnih primerov in situacij kar izboljša kvaliteto zaznave v robnih primerih. V ta namen se pogosto uporablja bogatejše učnih podatkov pri čemer se poslužujemo osnovnih transformacij, kot so zrcaljenje, rotacija, izrezi in podobno, da povečamo nabor učnih podatkov. Eden izmed razlogov za prevlado uporabe nevronske mreže je tudi napredek tehnologije in posledično večja razpoložljivost računske moči ter dostopnost do podatkov. Aplikacije zaznave objektov se danes pojavljajo v številnih področjih, kot na primer na področju avtonomne vožnje za zaznavo prometnih znakov, drugih udeležencev v prometu, in ovir ter področju nadzora kvalitete za zaznavo defektnih in neustreznih izdelkov. Na področju zaznave kart že obstaja več različnih tehnik in pristopov za uspešno klasifikacijo in



Slika 1: Prikaz licitacije pri igri bridža. Vir: Romeo Varga

zaznavo kart [2, 1]. Najbolj uspešni pristopi vključujejo tehnologijo nevronske mreže. Obstajajo tudi članki, ki obravnavajo zaznavo igre bridža, vendar se osredotočajo na fazo igranja, ki za razliko od faze licitacije uporablja običajni komplet 52 igralnih kart [4]. Naš problem se zaradi tega razlikuje v naslednjih pogledih:

- Karte za licitacijo se razlikujejo od običajnih kart. Uporabljen je drugačen nabor simbolov in oblika kart ni standardna.
- Ozadja nekaterih kart so barvna, kar lahko privede do oteženega zaznavanja.

Zaradi predhodnih uspešnih implementacij na problemih iz istega področja, smo se odločili za uporabo YOLO algoritma [5].

### 3 Definicija zahtev

Na podlagi prvotnih idej in analize smo definirali glavne zahteve projekta ter tako načrtali plan razvoja aplikacije. Prepoznali smo sledeče funkcijske zahteve:

- prepoznavanje vseh licitacijskih listkov,
- prepoznavanje igralcev,
- zapis v pbn datoteko.

Aplikacije za zaznavanje klasičnega kompleta 52-ih kart sicer že obstajajo (s tem tudi nabor podatkov). Toda za našo aplikacijo pa je bilo potrebno ustvariti nov nabor podatkov, saj se klasične igralne karte razlikujejo od licitacijskih listkov.

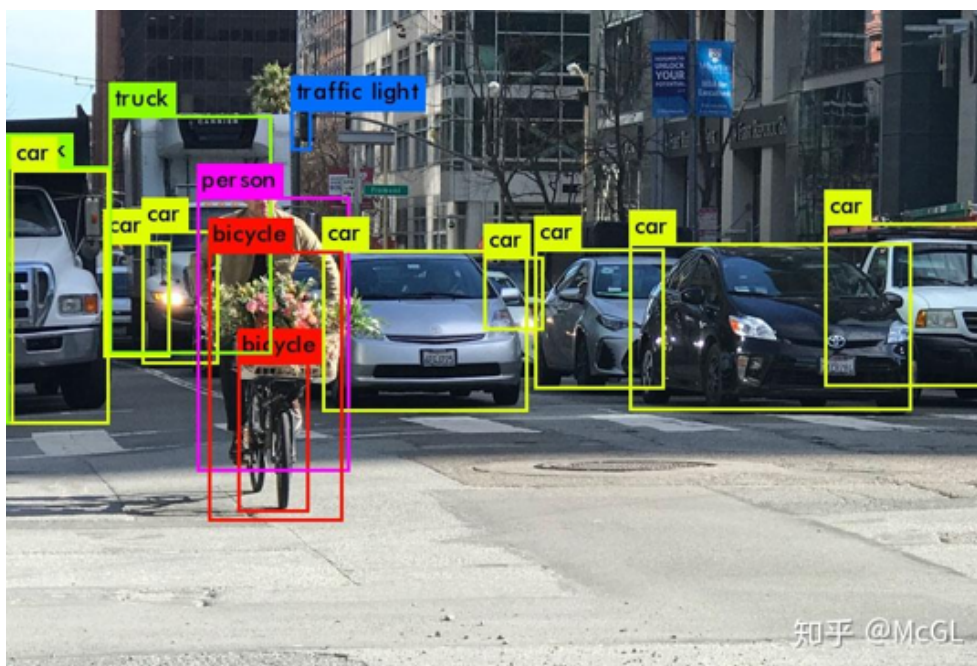
## 4 Tehnologije

### 4.1 Izbira tehnologije za razvoj

Pri analizi in načrtovanju smo se odločali med različnimi sodobnimi tehnologijami, ki se uporabljajo za zaznavanje objektov v slikah. Za pomoč pri generaciji nabora podatkov in zapisu v pbn datoteku smo uporabili programski jezik Python z več knjižnicami[3]. Za konkretno zaznavanje objektov v slikah smo uporabili algoritem YOLO[5].

### 4.2 YOLO

YOLO (You Only Look Once) je priljubljen algoritem za zaznavanje objektov v slikah in videoposnetkih. Ta algoritem je bil razvit z namenom, da doseže hitro in natančno zaznavanje objektov v realnem času. Glavna prednost YOLO je v tem, da lahko zazna več objektov naenkrat v enem prehodu skozi sliko ali videoposnetek, namesto da bi morali sliko prehajati večkrat, kar prihrani dragocen čas. YOLO uporablja konvolucijske nevronske mreže za učenje objektov iz slik in nato izvaja postopek zaznavanja na teh objektih. Ta algoritem se pogosto uporablja v različnih aplikacijah, kot so avtonomna vožnja, video nadzor, prepoznavanje obrazov, in še več, zaradi svoje sposobnosti hitrega in zanesljivega zaznavanja objektov v različnih okoljih. YOLO je odprtokoden in ima različne različice, ki jih lahko prilagodimo in uporabimo za različne potrebe v področju računalniškega vida. V našem primeru smo uporabili verzijo YOLOv8n.



Slika 2: Simulacija zaznavanja algoritma YOLO. Vir: <https://www.riset.guru.pubiway.com/yolo-object-detection-using-opencv-with-python-youtube/>

### 4.3 Python

Python je priljubljen programski jezik, ki slovi po svoji berljivi sintaksi. Ta vsestranski jezik se uporablja za razvoj številnih vrst aplikacij, vključno s spletnimi, znanstvenimi in avtomatizacijskimi. Python je interpretiran jezik, kar pomeni, da kode ni treba kompilirati, in ponuja obsežno standardno knjižnico z že napisanimi moduli za različne naloge. Zahvaljujoč široki skupnosti programerjev, ki podpirajo Python, je na voljo

bogat vir informacij in knjižnic tretjih oseb. Poleg tega je Python prenosljiv na različne operacijske sisteme, vključno s Windowsom, macOS in Linuxom. Ta jezik je še posebej priljubljen v področjih umetne inteligence, strojnega učenja in znanstvenega računanja ter ponuja orodja za enostavno branje in manipulacijo podatkov. Poleg tega spodbuja hitro razvojno iteracijo in prototipiranje aplikacij. V našem primeru smo uporabili python za generiranje učnih podatkov ter za končni zapis v pbn datoteko. Pri tem smo si pomagali s knjižnico OpenCV.

#### 4.4 OpenCV

OpenCV (cv2) je priljubljena Python knjižnica, ki se uporablja za računalniški vid in obdelavo slik. Ta knjižnica ponuja obsežen nabor funkcij in orodij za zajemanje, obdelavo in analizo slik in videoposnetkov. Uporablja se v različnih aplikacijah, vključno s strojnimi vidom, računalniškim vidom, obdelavo slik v realnem času, zaznavanjem objektov, sledenjem, prepoznavanjem obrazov, analizo gibanja in še več. OpenCV je odprtokodna in brezplačna, kar omogoča razvijalcem, raziskovalcem in inženirjem enostavno izkoriščanje njene zmogljivosti za različne projekte.

## 5 Generiranje učnih podatkov

Podatke za učenje modela smo generirali sami. V nadzorovanem okolju smo zajeli kratke, približno 10 sekund dolge videoposnetke pri čemer smo spreminjali kot osvetlitve. Pri zajemu je bila kamera pritrjena na stativ, karte so bile položene na nevtralno ozadje za osvetlitev pa smo uporabili cevne LED svetila. S spreminjanjem pozicije svetila se kontrast in osvetlitev površine karte spreminja, hkrati pa se na površini pojavijo odsevi. Namen takšnih robnih primerov je, da se model uči na primerih, ki so verjetni tudi v realnih situacijah. Bridž se namreč igra v različnih okoljih, ker osvetlitev ni popolna in se lahko pričauje tudi slabše osvetlive kart ali odseve. Zajete posnetke smo nato z uprabo *cv2* knjižnice in python skripte razrezali na sličice. Tako smo za vsako sekundo dobili 30 sličic. Iz posameznih sličic se je nato izrezalo simbol karte. V zadnji fazi pa smo iz nabora vseh izrezanih simbolov oziroma kart izbrali pet naključnih ter jih združili v kolaž, ki ponazarja igralčevo "roko" kart, ki se pojavi pri licitaciji. Primer kolaža je viden na sliki 3.



Slika 3: Primer kolaža.

Celotni potek izdelave nabora podatkov je prikazan na sliki 4. Po končanem postopku izdelave kolažov, ima vsak kolaž izdelane dve datoteki in sicer: sliko kolaža (primer Slika1.jpg) ter seznam vseh label, ki so prisotne v kolažu (primer Slika1.txt).



Slika 4: Potek izdelave nabora podatkov.

## 6 Implementacija

Zatem ko smo izdelali izdelali vse kolaže za učno in validacijsko množico, približno 10000 kolažev skupaj, so se le ti razdelili v razmerju 70:30, torej 70% kolažev je postalo učna množica in 30% pa je postalo validacijska množica. Na to smo začeli učenje nevronske mreže, s pomočjo programa v vmesniku z ukazno vrstico (ang. Command-line interface ali CLI) namenjen učenju YOLO nevronskih mrež. Točen ukaz, ki smo ga pogнали, je sledeč:

```
nice nohup yolo detect train data=coco128.yaml //  
model=./Model/best.pt epochs=10 imgsz=900 lr0=0.01 &
```

Učenje smo ponovil dvakrat na dve različnih množicah kolažev, za boljšo natančnost. Končni model smo nato uporabili v programu za prepoznavanje licitacije. Program sprejme kot parameter dva podatka in sicer sliko mize ter številko mize, na kateri igrajo igralci. Program sliko posreduje YOLO nevronska mrežo, ki vrne labele kart ter njihovo pozicijo na sliki. S temi podatki program razdeli karte na sliki pripadajočim igralcem s pomočjo sledeči pravil:

- Sever: če je  $X$  vrednost pozicije karte nižja kot 10% višine slike,
- Jug: če je  $X$  vrednost pozicije karte višje kot 90% višine slike,
- Zahod: če je  $Y$  vrednost pozicije karte nižja kot 10% širine slike,
- Vzhod: če je  $Y$  vrednost pozicije karte višja kot 90% širine slike,

pri čemer program predpostavi, da je slika mize orientirana Sever gor. Zatem program pregleda katera je licitacija igre ter zapiše vse podatke o igri v .pbn datoteko. Primer .pbn datoteke je viden na Sliki 5. Pri tem mora uporabnik spremeniti imena igralce ter lokacijo prizorišča, saj program uporablja privzete vrednosti za te podatke.

```
[Event "Projektni seminar"]
[Site "Famnit"]
[Date 2023-06-26]
[Board "3"]
[Dealer "S"]
[Declarer "W"]
[Contract "S4"]
[North "Jani"]
[East "Zan"]
[South "Dani"]
[West "Romeo"]
[Vulnerable "Both"]
[Deal "N:T932.HKQ8753.DJ.A"]
[Scoring "IMP"]
[Auction "S"]

PASS D1 X XX
H1 S1 PASS S3
PASS S4 PASS PASS
PASS
```

Slika 5: Primer pbn datoteke.

## 7 Razdelitev dela

Porazdelitev nalog:

- Žan: analiza, načrtovanje, izdelava videoposnetkov, izdelava sistema za prepoznavanje, dokumentacija
- Jani: analiza, načrtovanje, izdelava sistema za prepoznavanje, zapis v pbn file, dokumentacija
- Dani: analiza, načrtovanje, generiranje nabora podatkov, zapis v pbn file, dokumentacija

## Literatura

- [1] *Image Detection Using YOLO Algorithm and Poker Cards*. n.d. URL: <https://tooploox.com/image-detection-using-yolo-algorithm-and-poker-cards>.
- [2] *Unveiling the Cards: Real-time Poker Card Detection Using Computer Vision and YOLO*. n.d. URL: <https://medium.com/@abhijeetas8660211/unveiling-the-cards-real-time-poker-card-detection-using-computer-vision-and-yolo-6e1c22e606ed>.



- [3] *What is Python? Executive Summary*. n.d. URL: <https://www.python.org/doc/essays/blurb/>.
- [4] P. Xu, C. R. Kumar in G. Bramley-Simmons. *Bridge bidding via playing card detection using YOLO v3*. 2020. URL: <https://cs.brown.edu/research/pubs/theses/capstones/2020/xu.philip.pdf>.
- [5] *YOLO: Algorithm for Object Detection Explained*. 2023. URL: <https://www.v7labs.com/blog/yolo-object-detection>.