

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Iva Barišić, Tin Petric

## **Klasifikacija ispita metodama strojnog učenja**

Zagreb, 2021.

*Ovaj rad izrađen je u Zavodu za komunikacijske i svemirske tehnologije pod vodstvom doc. dr. sc. Daria Bojanjca i predan je na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2020./2021.*

## Sadržaj

1. Uvod .....	1
2. Metode .....	4
2.1. Segmentacija obrasca .....	4
2.1.1. Reorijentacija obrazaca.....	4
2.1.2. Segmentacija interesnih područja .....	7
2.1.3. Morfološka obrada slika .....	10
2.2. Klasifikacija rukom pisanih znakova .....	13
2.2.1. Modeli neuronskih mreža .....	14
2.2.2. Treniranje i validacija mreža .....	15
2.3. Klasifikacija zacrnjenih kružića .....	16
3. Rezultati.....	19
3.1. Klasifikacija studenta .....	19
4. Zaključak i daljnji rad.....	23
Literatura .....	24
Sažetak.....	25
Summary.....	26

# 1. Uvod

Računalni vid područje je umjetne inteligencije koje se bavi pomaganjem računalima u razumijevanju i numeričkom opisivanju sadržaja digitalnih slika i videa. Problem računalnog vida čini se jednostavan jer ljudski mozak vrlo lako percipira i razlikuje stvari oko sebe, ali taj je problem dugo bio nemoguć za računala. Problemi poput prepoznavanja rukom pisanog teksta bili su vremenski zahtjevni i neučinkoviti jer se prepoznavanje teksta svodilo na tradicionalni OCR (Optical Character Recognition), odnosno računalo je nastojalo pohraniti sve varijacije fontova i na taj način prepoznavati tekst.

Napretkom umjetne inteligencije i neuronskih mreža danas se OCR široko koristi kao metoda unosa podataka u različitim područjima. Neke od njih su pretvorba teksta u govor kao pomoć osobama s poteškoćama u vidu, unos podataka za dokumente, automatsko prepoznavanje registracijskih pločica, prepoznavanje prometnih znakova.

Ideja ovog rada bila je kreiranje alata za automatsko unošenje podataka sa ispitnih obrazaca na Fakultetu elektrotehnike i računarstva. S obzirom na velik broj studenata na pojedinim kolegijima na FER – u, profesorima i asistentima bi se uvelike olakšao posao unosa rezultata. Korištenjem ovog alata vrijeme unošenja rezultata znatno bi se smanjilo te bi i studentima smanjilo vrijeme čekanja na rezultate. Daljnjim korištenjem i unapređivanjem ovog modela broj pogrešaka kod unosa mogao bi biti manji nego kod ručnog unosa.

Za izradu alata bili su potrebni rukom ispunjeni ispitni obrasci koji su dizajnirani u suradnji s mentorom (Slika 1.1). Ispitne obrasce popunjavali su studenti na kolegiju Linearna algebra generacije 2020/2021. te su potpisali privolu kako bismo njihove podatke smjeli koristiti u razvoju ovog alata. Za izradu ovog rada prikupljeno je oko 300 ispunjenih obrazaca.

Ime i prezime:

JMBAG:

Broj zadatka:

Broj bodova:  0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15

- CILJ:** ispitivanje izvedivosti strojnoga čitanja rukopisa u svrhu automatskoga povezivanja potpisnoga studenta s brojem bodova dodijeljenih mu na pojedinom zadatku pismenoga ispita.
- KAKO ISPUNITI OBRAZAC:** molimo vas da upišete svoje ime i prezime odvojeno jednom praznom kućicom, JMBAG te da kao broj zadatka upišete zadnje dvije znamenke svoga JMBAG-a, a kao broj bodova zacrnite kružić koji se nalazi uz broj koji dobijete kao ostatak pri dijeljenju broja koji čine zadnje dvije znamenke JMBAG-a brojem 16. Taj isti broj upišite i u kućice s desne strane.
- ZA ŠTO ĆE SE KORISTITI VAŠ OBRAZAC:** skenirani obrasci koristit će se, obrađivati i pohranjivati u skladu s gore navedenim ciljem, a po potrebi i u druge svrhe vezane za strojno čitanje rukopisa te po potrebi za izgradnju baze rukom pisanih znakova koja bi se koristila za treniranje neuronske mreže za segmentaciju i klasifikaciju. Podatke će koristiti članovi istraživačkog tima te podatci neće biti ustupljeni drugima. Zaprimiti podatci čuvat će se do kraja istraživanja.

**PRIVOLA ZA OBRADU OSOBNIH PODATKA**

Sudjelovanjem u istraživanju, u svojstvu ispitanika, dajem privolu za obradu mojih osobnih podataka koji se nalaze na ovom obrascu u svrhe opisane u trećoj točki. Voditelj obrade je Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, Unska 3, Zagreb. Obavješten sam da imam pravo uvida u to kako se koriste moji podatci i pravo povući ovu privolu u bilo kojem trenutku. Obavješten sam da se, ako smatram da je došlo do povrede mojih prava u vezi s obradom mojih osobnih podataka, mogu javiti na adresu elektroničke pošte [sluzbenik.osobnipodaci@fer.hr](mailto:sluzbenik.osobnipodaci@fer.hr) ili mogu podnijeti prigovor Agenciji za zaštitu osobnih podataka, Sečska cesta 136, Zagreb, na adresu elektroničke pošte [azop@azop.hr](mailto:azop@azop.hr).

Potpisom potvrđujem da sam pročitao sve gore navedeno, da se s time slažem te da pristajem sudjelovati u istraživanju.

---

Slika 1.1: Ispitni obrazac

Ispitni obrazac sastoji se od prostora za upis imena i prezimena duljine 31 znakova. Duljina tog segmenta odabrana je zbrajanjem maksimalne duljine imena i maksimalne duljine prezimena studenata na FER – u. Ispitni obrazac sadrži i prostor za unos JMBAG – a, broja zadatka te prostor za broj bodova i kružiće za zacrnjivanje broja bodova koje ispunjava profesor/asistent nakon ispravljanja ispita. Radi lakše orijentacije obrazaca dodane su crne točke kao granice promatranog područja.

Ispunjeni obrasci skenirani su u Centru informacijske potpore FER-a, a podaci sa svakog obrazaca ručno su upisani u Excel dokument (Slika 1.2).

	A	B	C	D	E	F	G	H	I
1	oznaka	ime1	ime2	prezime1	prezime2	jmbag	brzad	brbod	brbod_broj
2	CIPsharp_20210326_110903_0001	DAMIR		ZADRO		0036512345	45	13	13
3	CIPsharp_20210326_110903_0002	IGOR		RADIŠNI		0036523232	32	0	00
4	CIPsharp_20210326_110903_0003	ALGORITAM		RADI		0036523212	12	12	12
5	CIPsharp_20210326_110903_0004	PETAR		PETROVIĆ		0036419237	37	5	05
6	CIPsharp_20210326_110903_0005	MARKO		MARKOVIĆ		0036423791	91	11	11
7	CIPsharp_20210326_110903_0006	IVICA		ĐAKIĆ		0036321113	13	13	13
8	CIPsharp_20210326_110903_0007	PIŠEM		PISMO		0036442310	10	10	10
9	CIPsharp_20210326_110903_0008	NISAM		NORMALAN		0036219234	34	2	02
10	CIPsharp_20210326_110903_0009	AMERIČKI		PREDSJEDNIK		0036467890	90	10	10
11	CIPsharp_20210326_110903_0010	KOLINDA		MILANOVIĆ		0036423704	04	4	04
12	CIPsharp_20210326_110903_0011	ALEKSANDAR		PETROVIĆ	MALI	0036400009	09	9	9
13	CIPsharp_20210326_110903_0012	ZVONIMIR		HRABAR		0036412003	03	3	3
14	CIPsharp_20210326_110903_0013	FRANO		FRANIĆ		7321043999	10	12	
15	CIPsharp_20210326_110903_0014	PERO		PERIĆ		7894321056	10	15	
16	CIPsharp_20210326_110903_0015	JURAJ		MARUŠIĆ		9999999999	99	9	09
17	CIPsharp_20210326_110903_0016	JURAJ		ČMARUŠIĆ		0123456783	10	0	00
18	CIPsharp_20210326_110903_0017	EVITA		PERON		1234567891	10	14	14
19	CIPsharp_20210326_110903_0018	EVITA		ŠOPP		0036513375	55	15	15
20	CIPsharp_20210326_110903_0019	IVAN		IVKOVIĆ		0036515710	10	10	10
21	CIPsharp_20210326_110903_0020	PETAR		PETROVIĆ		0036515550	50	2	02
22	CIPsharp_20210326_110903_0021	LUKA		MODRIĆ		0034199218	18	2	02
23	CIPsharp_20210326_110903_0022	STJEPAN		MESIĆA		0034181294	94	14	14
24	CIPsharp_20210326_110903_0023	JANICA		KOSTELIĆ		0036518820	20	4	04

Slika 1.2 Izgled Excel dokumenta

Prvi korak kod izlučivanja podataka sa ispitnog obrasca je segmentacija slike. Segmentacija slike je traženje regija interesa na kojima se nalaze pojedini podaci te podjela tih regija na atomarne vrijednosti. Kasnije se te vrijednosti koriste za treniranje neuronske mreže u obliku sličica.

Nakon segmentacije, konstruira se model neuronske mreže i zatim trenira kako bi se odredio student te broj dobivenih bodova za određeni zadatak. Prepoznavanje studenta je olakšano zbog unaprijed poznatog popisa studenata na pojedinom kolegiju pa je i uz pogreške klasifikacije znakova moguće klasificirati studenta.

## 2. Metode

Ovaj rad izrađen je u programskom jeziku Python verzije 3.7 uz korištenje raznih biblioteka za rad sa strojnim učenjem i obradom slika. Korištena je biblioteka OpenCv za procesiranje slika koja implementira brojne algoritme računalnog vida. Za prepoznavanje slova i njihovo klasificiranje korišteni su Keras[7] i Tensorflow[8] koji podržavaju rad s neuronskim mrežama.

### 2.1. Segmentacija obrasca

#### 2.1.1. Reorijentacija obrazaca

Kako bi segmentacija obrazaca bila uspješna važno je da su obrasci jednako orijentirani. S obzirom na to da su moguće greške kod skeniranja velikog broja obrazaca potrebno je napraviti automatsku reorijentaciju obrazaca. Slika 2.1 pokazuje primjer „dobro“ skeniranog obrasca, a Slika 2.2 primjer „loše“ skeniranog obrasca.

Ine i prezime: A U E L S A N D A R P E T C I V C W A L I

JMBAG: C C 2 4 4 C C C C T

Broj zadatka: C 1

Broj bodova: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 [ 7 ]

1. CILJ: Ispitivanje i provedba strojnoga čitanja rukopisa u svrhu automatizirane provedbe potpisanooga studenta s brojem bodova dodijeljenih mu na pojedinom zadatku pismenoga ispita.

2. KAKO ISPUNITI OBRZAC: molimo vas da upišete svoje ime i prezime odvojeno jednom praznom kvadratom, JMBAG te da kao broj zadatka upišete zadnje dvije znamenke svoga JMBAG-a, a kao broj bodova nacrtate krugić koji se nalazi na broj koji dobijete kao ostatak pri dijeljenju broja koji čine zadnje dvije znamenke JMBAG-a brojem 10. Taj isti broj upišite i u kvadrat s desne strane.

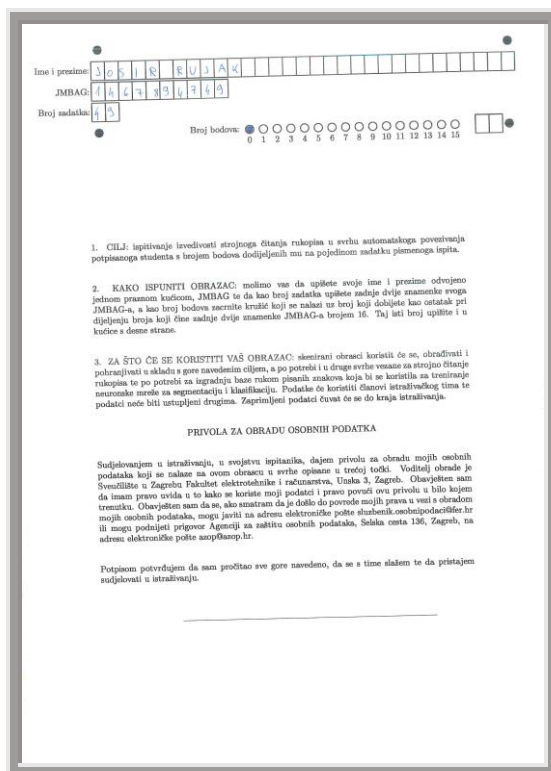
3. ZA ŠTO ĆE SE KORISTITI VAŠ OBRZAC: skenirani obrasci koristiće se za obavljanje i pohranjivanje u bazu i gore navedenim ciljem, a po potrebi i u druge svrhe vezane sa strojno čitanje rukopisa te po potrebi za isradnju baze rukom pisanih znakova koja će se koristiti za treniranje neuronske mreže za segmentaciju i klasifikaciju. Podatke će koristiti članovi istraživačkog tima te podatci neće biti ustupljeni drugima. Zaprimljeni podatci čuvat će se do kraja istraživanja.

PRIVOLA ZA OBRADU OSOBNIH PODATKA

Sudjelovnjem u istraživanju, u svojstvu ispitanika, dajem pristanak na obradu mojih osobnih podataka koji se nalaze na ovom obrascu u svrhe opisane u trećoj točki. Voditelj obrade je Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, Ustaka 3, Zagreb. Obavještjen sam da imamo pravo videti u to kako se koriste moji podatci i pravo povući ova pristanak u bilo kojem trenutku. Obavještjen sam da se, ako smatram da je došlo do povrede mojih prava u vezi s obradom mojih osobnih podataka, mogu javiti na adresu elektroničke pošte: [slazbenik.oo@mpodocifer.hr](mailto:slazbenik.oo@mpodocifer.hr) ili mogu podnijeti prigovor Agenciji za zaštitu osobnih podataka, Šetka cesta 130, Zagreb, na adresu elektroničke pošte: [uao@azop.hr](mailto:uao@azop.hr).

Potpisano potvrđujem da sam pročitao sve gore navedeno, da se s time slažem te da pristajem sudjelovati u istraživanju.

Slika 2.1 Primjer „dobro“ skeniranog obrasca



Slika 2.2 Primjer „loše“ skeniranog obrasca

Kod reorijentacije korišten je ORB (Oriented FAST and rotated BRIEF) iz Pythonove OpenCV biblioteke. ORB je alat koji služi za detekciju ključnih točaka i ekstrakciju deskriptora iz slike. ORB prvo koristi FAST za detekciju ključnih točaka te zatim primjenjuje Harrisov kutni detektor kako bi izdvojio N najboljih točaka među njima[11]. Harrisov kutni detektor je afino-invarijantan, tj. pronalazi slična područja na slikama povezana jednostavnim geometrijskim transformacijama (translacija, rotacija, skaliranje i smik)[6].

Odabran je jedan dobro skeniran obrazac kao referentan te je reorijentacija ostalih rađena prema njemu. Kreiran je objekt `orb` nad kojim se poziva funkcija `detectAndCompute` koja kao argument prima sliku i vraća ključne točke i deskriptore za tu sliku (Kôd 2.1). Stvoren je objekt `matcher` tipa `BRUTEFORCE_HAMMING` pomoću OpenCV funkcije `DescriptorMatcher_create`. `Matcher` je apstraktna klasa koja služi za uparivanje deskriptora ključnih točaka dvije slike. Nad njime je pozvana metoda `match` koja svaki deskriptor značajke prve slike spaja sa svakom značajkom druge slike i računa Hammingovu udaljenost za svaki par deskriptora [11]. Zatim se deskriptori sortiraju uzlazno da bi najbolji parovi deskriptora (s najmanjom Hammingovom udaljenosti) bili na početku liste `matches` koju kasnije koristimo pri reorijentiranju obrasca.



```

orb = cv2.ORB_create(MAX_FEATURES)
keypoints1, descriptors1 = orb.detectAndCompute(im1Gray,
None)
keypoints2, descriptors2 = orb.detectAndCompute(im2Gray,
None)
matcher =
cv2.DescriptorMatcher_create(cv2.DEScriptorMatcher_BRUTEFORC
E_HAMMING)

matches = matcher.match(descriptors1, descriptors2, None)
matches = sorted(matches, key=lambda x:x.distance)
numGoodMatches = int(len(matches) * GOOD_MATCH_PERCENT)
matches = matches[:numGoodMatches]

```

### Kôd 2.1 Generiranje ključnih točaka i deskriptora

Pronalaze se lokacije najboljih parova i pomoću funkcije `findHomography` pronalazi se transformacija uparenih ključnih točaka. Funkcija `warpPerspective` provodi dobivenu transformaciju na danu sliku koja se sprema za daljnju segmentaciju.

```

points1 = np.zeros((len(matches), 2), dtype=np.float32)
points2 = np.zeros((len(matches), 2), dtype=np.float32)
for j, match in enumerate(matches):
    points1[j, :] = keypoints1[match.queryIdx].pt
    points2[j, :] = keypoints2[match.trainIdx].pt
h, mask = cv2.findHomography(points1, points2, cv2.RANSAC)
height, width = imReference.shape[:2]
imReg = cv2.warpPerspective(im, h, (width, height))

```

### Kôd 2.2 Provođenje transformacije slike

Rezultat reorijentacije vidljiv je na Slika 2.3.



Slika 2.3 Rezultat obavljanja reorijentacije

## 2.1.2. Segmentacija interesnih područja

Nakon obavljene reorijentacije svi obrasci su približno jednako orijentirani što znači da se prostori u kojima se nalaze podatci studenata nalaze na istom mjestu, odnosno istim koordinatama. Interesna područja su područja slike koje želimo izdvojiti za daljnju obradu.

Grupe podataka interesnih područja su:

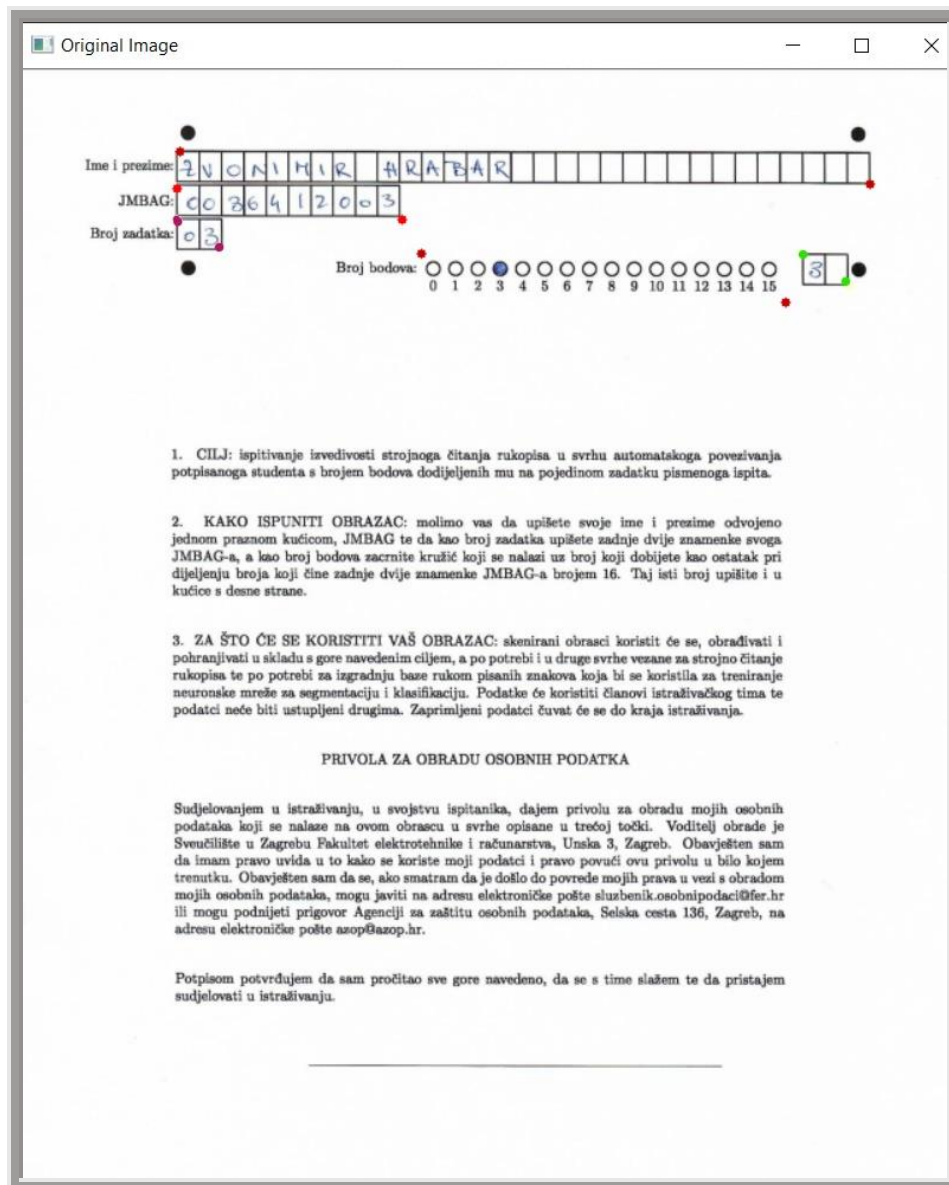
- Ime i prezime
- JMBAG
- Broj zadatka
- Broj bodova u obliku kružića za zacrnjivanje
- Broj bodova u pisanom obliku

Određivanje koordinata interesnih područja provedeno je python skriptom pokrenutom nad referentnim obrascem.

Skripta sadrži `MouseEvent` funkciju koja se poziva kod lijevog klika mišem čime se određuju koordinate točke. Regije interesa odabiru se lijevim klikom mišem u gornji lijevi kut i donji desni kut polja za unos podataka pojedine grupe. Kod svakog klika mišem označena točka oboji se u nasumičnu boju kako bismo lakše vidjeli tražene rubove. Nakon odabira jedne regije interesa upisuje se tip podataka (text ili number) te ime grupe podataka. Funkcija koja se poziva kod svakog klika mišem prikazana je u Kôd 2.3. Koordinate prvog klika su  $(x_1, y_1)$ , koordinate drugog klika su  $(x_2, y_2)$ , time je regija interesa određena  $[x_1 : x_2, y_1 : y_2]$ . Pet takvih parova koordinata određuje područja slike koja će se segmentirati. Slika 2.4 prikazuje odabrane regije interesa na obrascu.

```
def mousePoints(event, x, y, flags, params):
    global counter, point1, point2, counter2, circles, myColor
    if event == cv2.EVENT_LBUTTONDOWN:
        if counter==0:
            point1=int(x//scale),int(y//scale);
            counter +=1
            myColor =
(random.randint(0,2)*200,random.randint(0,2)*200,random.randi
nt(0,2)*200 )
        elif counter ==1:
            point2=int(x//scale),int(y//scale)
            type = input('Enter Type')
            name = input ('Enter Name ')
            myPoints.append([point1,point2,type,name])
            counter=0
        circles.append([x,y,myColor])
        counter2 += 1
```

Kôd 2.3 Funkcija za određivanje koordinata interesnih regija



Slika 2.4 Prikaz odabranih regija na obrascu

Svaki obrazac segmentiran je na pet područja naredbom oblika:

```
imgCrop=img[y1:y2, x1:x2]
```

Gdje je `img` originalna slika, a `x1,y1` i `x2,y2` su koordinate područja.

Svako područje osim područja za upisivanje broja bodova u kružiće segmentirano je na određeni broj dijelova tako da svaki novonastali dio predstavlja jedan kvadratić, odnosno jedan znak. Takvo segmentiranje obavljeno je pomoću Numpy funkcije `hsplit` vertikalno s obzirom na broj kućica:

- Ime i prezime sastoji se od 31 kućice – broj dobiven tako da smo uzeli najdulje ime i najdulje prezime studenata na FER-u

- JMBAG se sastoj od 10 kućica
- Broj zadatka sastoji se od 2 kućice
- Broj bodova sastoji se od 2 kućice

Kako bi bilo moguće podijeliti sliku funkcijom `hsplit`, širina slike mora biti višekratnik broja kvadratića regije interesa. Kod segmentacije imena i prezimena širina slike mora biti višekratnik broja 31 pa slici mijenjamo veličinu na najbliži višekratnik broja 31, a to je 899. Kod segmentacije JMBAG – a širina slike mora biti višekratnik broja 10 pa slici mijenjamo veličinu na 300. Kod segmentacije broja zadatka i broja bodova širina slike mora biti višekratnik broja 2 pa slici mijenjamo veličinu na 60. Primjer segmentacije jednog obrasca prikazan je na slikama: Slika 2.5, Slika 2.6, Slika 2.7, Slika 2.8, Slika 2.9.



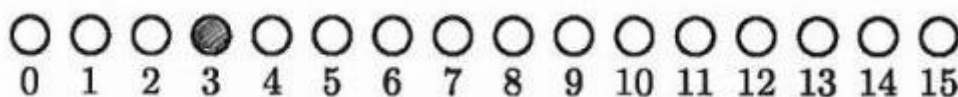
Slika 2.5 Primjer segmentacije područja za ime i prezime (bez dodatnih praznih kvadratića)



Slika 2.6 Primjer segmentacije područja za JMBAG



Slika 2.7 Primjer segmentacije područja za broj zadatka



Slika 2.8 Primjer segmentacije područja za unos broja bodova u kružice



Slika 2.9 Primjer segmentacije područja za broj bodova

### 2.1.3. Morfološka obrada slika

Vidljivo je da su na segmentiranim slikama ostale vertikalne i horizontalne linije kvadratića. Potrebno je dodatno obraditi segmente da bi bili spremni za klasificiranje. Uklanjanje nepotrebnih linija ostvareno je morfološkim transformacijama (Kôd 2.4)[14].

Digitalna morfologija je obrada slike na temelju unaprijed definiranih elemenata zvanih kerneli. Odabirom veličine i oblika kernela možemo konstruirati morfološku operaciju. Dvije korištene morfološke operacije su dilatacija i erozija. Erozija uklanja vanjske granice objekta na slici. Prolazi kroz sliku u obliku 2D konvolucijske matrice. Svaki piksel u originalnoj slici koji je ili 1 ili 0 bit će 1 samo ako su svi pikseli u kernelu 1, inače je 0. To znači da će pikseli blizu granice objekta na slici biti pretvoreni u 0 i debljina objekta će se smanjiti. Dilatacija je proces suprotan eroziji što znači da se piksel postavlja na vrijednost 1 ako je barem jedan piksel u kernelu 1[13]. Tim procesom povećava se širina objekta. Korištena je funkcija `threshold` koja obavlja transformaciju slike prema određenom pragu. Prag je vrijednost piksela koji se koristi u operaciji tako da se za svaki piksel slike provjerava odnos vrijednosti piksela i vrijednosti praga, ako je vrijednost piksela manja od vrijednosti praga, piksel se postavlja na nulu, inače se postavlja na maksimalnu vrijednost. Korištenjem opcije `TRESH_OTSU` nije potrebno određivati vrijednost praga, nego se ona automatski izračunava[10]. Funkcijom `getStructuringElement` kreiramo horizontalni kernel dužine 25 i debljine 1 piksel i vertikalni kernel dužine 1 i debljine 25 piksela.

Funkcija `cv2.morphologyEx` prima 4 parametra. Slika na kojoj je primijenjen `threshold`, `cv2.MORPH_OPEN`, `kernel` i broj iteracija. `cv2.MORPH_OPEN` predstavlja opciju rada funkcije `cv2.morphologyEx` te se primjenjuje erozija praćena dilatacijom što je tipično kod uklanjanja šuma na slici. Erozija uklanja bijeli šum, ali i smanjuje objekt. Tada primjenom dilatacije povećavamo objekt, ali se šum neće vratiti.

```
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
thresh = cv2.threshold(gray, 0, 255,
cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)[1]
horizontal_kernel =
cv2.getStructuringElement(cv2.MORPH_RECT, (25,1))
vertical_kernel=
cv2.getStructuringElement(cv2.MORPH_RECT, (1,25))
detected_lines1 = cv2.morphologyEx(thresh,
cv2.MORPH_OPEN, horizontal_kernel, iterations=2)
detected_lines2 = cv2.morphologyEx(thresh,
cv2.MORPH_OPEN, vertical_kernel, iterations=2)
```

#### Kôd 2.4 Morfološke transformacije

Sljedeći korak je pronaći konture u slici. Konture su kontinuirano povezane točke uz granicu objekta na slici koje imaju istu boju ili intenzitet. Potrebno je prije pronalaska kontura nad slikom primijeniti `threshold`. U OpenCV – u pronalazak konture jednak je pronalasku bijelog objekta na crnoj podlozi. Konture se pronalaze pomoću funkcije `cv2.findContours`. Funkcija prima 3 vrijednosti: slika nad kojom smo proveli morfološku transformaciju, `cv2.RETR_EXTERNAL`, `cv2.CHAIN_APPROX_SIMPLE`. Opcija `cv2.RETR_EXTERNAL` vraća samo ekstremne vanjske konture. Opcija `cv2.CHAIN_APPROX_SIMPLE` vraća samo krajnje točke koje su potrebne za iscrtavanje linije konture. Nakon toga pronađene konture iscrtavaju se bijelom bojom pomoću funkcije `cv2.drawContours`.

```

cnts2 = cv2.findContours(detected_lines2, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
cnts2 = cnts2[0] if len(cnts2) == 2 else cnts2[1]
for c in cnts2:
    cv2.drawContours(image, [c], -1, (255,255,255), 2)

cnts1 = cv2.findContours(detected_lines1,
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
cnts1 = cnts1[0] if len(cnts1) == 2 else cnts1[1]
for c in cnts1:
    cv2.drawContours(image, [c], -1, (255,255,255), 2)

```

#### Kôd 2.5 Pronalazak i iscrtavanje kontura

Potrebno je napraviti još jednu morfološku transformaciju koja popravlja mogući gubitak podataka prethodnom transformacijom. Kreira se novi strukturni element pomoću funkcije: `cv2.getStructuringElement(cv2.MORPH_RECT, (1, 6))`. Zatim nad slikom pozivamo funkciju `cv2.morphologyEx` s novo izračunatim strukturnim elementom i opcijom rada `cv2.MORPH_CLOSE`. `cv2.MORPH_CLOSE` je proces suprotan `cv2.MORPH_OPEN`-u koji prvo provodi dilataciju, a zatim eroziju te je koristan u uklanjanju malih rupica unutra objekta ili malih crnih točaka na objektu (Slika 2.10).



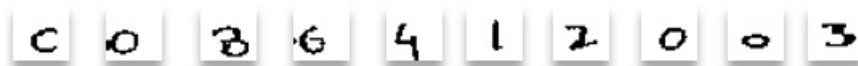
Slika 2.10 Uklanjanje nepravilnosti MORPH\_CLOSE operacijom [3]

Slike je potrebno izoštriti kako bi uzorci bili što jasniji. Za izoštravanje slike koristi se tehnika Unsharp masking (USM). USM koristi zamagljenu (mutnu) negativnu sliku te na temelju nje kreira masku originalne slike. Maska kombinirana sa originalnom pozitivnom slikom daje sliku koja je manje zamagljena od početne. U terminima obrade signala USM je linearni ili nelinearni filter koji pojačava visokofrekventne komponente signala. Funkcija `cv2.GaussianBlur` služi za zamagljivanje slike i uklanjanje Gaussovog šuma iz slike. Izoštrenu sliku nadalje računamo pomoću linearne kombinacije originalne slike i dobivene zamagljene slike.

Rezultati nakon morfološke obrade slika za već prikazani obrazac su:



Slika 2.11 Morfološki obrađeni segmenti imena i prezimena



Slika 2.12 Morfološki obrađeni segmenti JMBAG-a



Slika 2.13 Morfološki obrađeni segmenti broja zadatka



Slika 2.14 Morfološki obrađeni segmenti broja bodova na zadatku

## 2.2. Klasifikacija rukom pisanih znakova

Nakon segmentacije obrazaca potrebno je konstruirati neuronsku mrežu te klasificirane podatke podijeliti u skupove za treniranje i validaciju. Nakon formiranja skupova, neuronske mreže se treniraju i ispituju kako bi se dobila preciznost modela te kako bi se,



uz pomoć poznatih podataka o studentima, prepoznati podaci na obrascu pridružili studentima koji su ispunili obrazac. Postoje tri vrste učenja/treniranja neuronskih mreža:

- Nadzirano
- Nenadzirano
- Podržano

U ovome radu korišteno je nadzirano učenje. Nadzirano učenje je učenje mreže nad podacima za koje već znamo željeni izlaz, odnosno mreža kao parametre za treniranje prima parove [ulaz, željeni izlaz] [5].

### **2.2.1. Modeli neuronskih mreža**

Umjetna neuronska mreža je skup međusobno povezanih jednostavnih elemenata koji imitiraju biološke neurone. Sastoji se od ulaznog sloja, skrivenih slojeva i izlaznog sloja neurona. Ulazni sloj prima podatke koje treba klasificirati, izlazni sloj daje rezultat klasifikacije ulaznog podatka, a skriveni slojevi omogućavaju neuronskoj mreži provođenje transformacija nad težinama neurona[2].

Konstruirane su dvije neuronske mreže, jedna mreža za klasificiranje rukom pisanih slova (Kôd 2.6) i druga mreža za klasificiranje rukom pisanih brojeva(Kôd 2.7). Obje mreže imaju 784 neurona u ulaznom `Flatten` sloju, što predstavlja izravnjanje slike veličine 28x28 piksela u jednodimenzionalno polje od 784 piksela. `Dense` sloj je regularni sloj za duboko učenje te je najkorišteniji sloj kod neuronskih mreža. Kao argument sloju, uz broj neurona, predajemo i aktivacijsku funkciju. Aktivacijska funkcija definira na koji način se težinska suma ulaza pretvara u izlaz nekog neurona. Obje mreže sadrže dva skrivena sloja od 1024 neurona sa ispravljenom linearnom aktivacijskom funkcijom ReLu (Reactified Linear unit). ReLu je funkcija koja propušta ulaz direktno na izlaz ako je on pozitivan, inače je izlaz nula. ReLu je postala uobičajena aktivacijska funkcija za mnoge neuronske mreže jer su takvi modeli jednostavniji za treniranje i često daje bolje rezultate [9]. Model neuronske mreže za klasificiranje rukom pisanih brojeva ima jedanaest neurona u izlaznom sloju; deset neurona za znamenke od 0 do 9 i jedanaesti neuron za klasificiranje praznog kvadratića. Model neuronske mreže za klasificiranje rukom pisanih slova ima 24 neurona u izlaznom sloju. Zbog neravnomjernog broja podataka slična slova klasificirana su istim neuronom (C, Ć i Č kao C, D i Đ kao D, S i Š kao S, Z i Ž kao Z) što je ukupno dvadeset i

jedan neuron za velika tiskana slova hrvatske abecede, jedan neuron za prazan kvadratić i jedan neuron za povlaku (-). Obje mreže u izlaznom sloju koriste aktivacijsku funkciju `softmax` koja daje distribuciju za  $n$  različitih klasa, odnosno vjerojatnost svake klase iz intervala  $[0,1]$  tako da je suma svih vjerojatnosti klasa jednaka jedan [4].

```
model = tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(1024, activation=tf.nn.relu),
    tf.keras.layers.Dense(1024, activation=tf.nn.relu),
    tf.keras.layers.Dense(24, activation=tf.nn.softmax)
])
```

Kôd 2.6 Model neuronske mreže za klasificiranje slova

```
model = tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(1024, activation=tf.nn.relu),
    tf.keras.layers.Dense(1024, activation=tf.nn.relu),
    tf.keras.layers.Dense(11, activation=tf.nn.softmax)
])
```

Kôd 2.7 Model neuronske mreže za klasificiranje brojeva

## 2.2.2. Treniranje i validacija mreža

Na početku je izdvojeno 15% posto obrazaca koji služe za završno testiranje. Svaki ostali obrazac segmentiran je i transformiran na način opisan ranije. Svakom segmentu određena je pripadnost tipu podatka (broj ili slovo) te mu je pridružena oznaka zapisana u Excel dokumentu.

Slike segmenata učitane su u liste `slike_brojevi` i `slike_slova`, a željeni izlazi istim redoslijedom učitani su u liste `oznake_brojevi` i `oznake_slova`. Primjer učitavanja skupova brojeva prikazan je u Kôd 2.8. Takvi skupovi podataka podijeljeni su na skup za učenje koji sadrži 80% podataka i skup za validaciju koji sadrži 20% podataka (Kôd 2.9). Iako neuronske mreže trenirane na balansiranom skupu podataka daju bolje rezultate, naši skupovi podataka nisu balansirani radi malog broja istih. Ipak, s tim ciljem eliminiran je veliki dio slika koje predstavljaju prazne kvadratiće jer su one sačinjavale 90% podataka.

```

img = cv2.imread(path + str(i))[:, :, 0]
img = cv2.resize(img, (28, 28))
img = np.invert(np.array([img]))
img = img / 255
slike_brojevi.append(img)
oznake_brojevi.append(broj)

```

#### Kôd 2.8 Učitavanje slika i oznaka brojeva

```

number=math.floor(len(oznake_brojevi)*0.80)
x_train = np.array(slike_brojevi[:number])
y_train = np.array(oznake_brojevi[:number])

x_test = np.array(slike_brojevi[number +
1:len(oznake_brojevi) - 1])
y_test = np.array(oznake_brojevi[number +
1:len(oznake_brojevi) - 1])

```

#### Kôd 2.9 Podjela skupova na skup za učenje i skup za validaciju

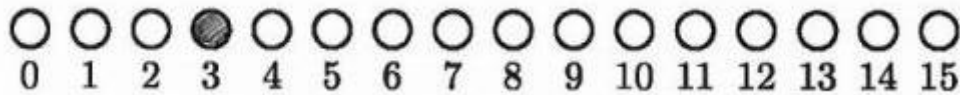
Učenje mreže provođeno je nad skupovima za učenje u 10 epoha za svaki model, a rezultati validacije mreža su sljedeći:

- 75% točno klasificiranih podataka skupa za validaciju na modelu za prepoznavanje brojeva
- 60% točno klasificiranih podataka skupa za validaciju na modelu za prepoznavanje slova

Rezultati modela su očekivani zbog malog broja podataka za učenje.

## 2.3. Klasifikacija zacrnjenih kružića

Segment obrasca s kružićima gdje zacrnjeni kružić predstavlja broj bodova dobivenih na zadatku (Slika 2.15) služi kao provjera u slučaju loše klasifikacije brojeva u kvadratićima za broj bodova.



Slika 2.15 Primjer segmenta sa zacrnjenim kružićem

Ekstrakcija broja iz ovakvog segmenta provedena je uz pomoć OpenCV metode `HoughCircles` koja služi za detekciju krugova na slici. Metoda `HoughCircles` prima kao parametre sliku, metodu detekcije (trenutno implementirana samo prikazana metoda `HOUGH_GRADIENT`), inverzni faktor rezolucije slike, minimalnu udaljenost između dva kruga, gornji prag za unutarnji Canny detektor rubova, prag za detekciju centra kruga te minimalni i maksimalni radijus kruga [12]. Kako bi se odredio broj koji je zacrnjen, potrebno je širinu slike podijeliti sa 16 što daje približnu širinu jednog kruga. Nakon detekcije zacrnjenog kruga, njegova x koordinata dijeli se sa približnom širinom jednog kruga. Ukoliko je dobiveni broj u intervalu npr. [12,81, 13,80] zacrnjena vrijednost je 13. Korištenje ove metode pokazalo se 100% uspješno.

```
def readButton(img):  
  
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)  
  
    blurred = cv2.medianBlur(gray, 25)  
    minDist = 250  
    param1 = 20  
    param2 = 10  
    minRadius = 1  
    maxRadius = 25  
    circles = cv2.HoughCircles(blurred, cv2.HOUGH_GRADIENT,  
2, minDist, param1=param1, param2=param2,  
minRadius=minRadius, maxRadius=maxRadius)  
    size = img.shape  
    xvalue= size[1]/16  
    result=None  
  
    if circles is not None:  
        circles = np.uint16(np.around(circles))  
        for i in circles[0,:]:
```

```
cv2.circle(img, (i[0], i[1]), i[2], (0, 255, 0),  
2)  
  
value = i[0]/ xvalue  
if value - floor(value) > 0.8:  
    result=ceil(value)  
else:  
    result=floor(i[0]/ xvalue)  
return result
```

Kôd 2.10 Detekcija krugova i određivanje zacrnjene vrijednosti

## 3. Rezultati

### 3.1. Klasifikacija studenta

Svaki ispitni obrazac pokazuje koliko je bodova na nekom zadatku ispita dobio neki student. S obzirom na rezultate validacije neuronskih mreža, potrebno je implementirati metodu pomoću koje bismo pouzdano mogli povezati studenta iz baze podataka sa osvojenim bodovima.

Rezultat izvođenja programa prije krajnje klasifikacije studenta za obrazac (Slika 3.1) je:

```
Rezultat klasifikacije: {'ime': 'MARIARO CUTOK',  
'jmbag': '0062245534', 'brzad': '34', 'brbod_broj': '09', 'brbod': 2}
```

Ime i prezime: M A R I A R O C U T O K

JMBAG: 0 0 6 2 2 4 5 5 3 4

Broj zadataka: 3 4

Broj bodova: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 2

1. CILJ: ispitivanje izvedivosti strojnoga čitanja rukopisa u svrhu automatskoga povezivanja potpisanoga studenta s brojem bodova dodijeljenih mu na pojedinom zadatku pismenoga ispita.
2. KAKO ISPUNITI OBRAZAC: molimo vas da upišete svoje ime i prezime odvojeno jednom praznom kućicom, JMBAG te da kao broj zadataka upišete zadnje dvije znamenke svoga JMBAG-a, a kao broj bodova sacrnite kružić koji se nalazi uz broj koji dobijete kao ostatak pri dijeljenju broja koji čine zadnje dvije znamenke JMBAG-a brojem 16. Taj isti broj upišite i u kućice s desne strane.
3. ZA ŠTO ĆE SE KORISTITI VAŠ OBRAZAC: skenirani obrasci koristit će se, obrađivati i pohranjivati u skladu s gore navedenim ciljem, a po potrebi i u druge svrhe vezane za strojno čitanje rukopisa te po potrebi za izgradnju baze rukom pisanih znakova koja bi se koristila za treniranje neuronske mreže za segmentaciju i klasifikaciju. Podatke će koristiti članovi istraživačkog tima te podatci neće biti ustupljeni drugima. Zaprimiti podatci čuvat će se do kraja istraživanja.

PRIVOLA ZA OBRADU OSOBNIH PODATKA

Sudjelovanjem u istraživanju, u svojstvu ispitanika, dajem privolu za obradu mojih osobnih podataka koji se nalaze na ovom obrascu u svrhe opisane u trećoj točki. Voditelj obrade je Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, Unska 3, Zagreb. Obavješten sam da imam pravo uvida u to kako se koriste moji podatci i pravo povući ovu privolu u bilo kojem trenutku. Obavješten sam da se, ako smatram da je došlo do povrede mojih prava u vezi s obradom mojih osobnih podataka, mogu javiti na adresu elektroničke pošte [sluzbenik.osobnipodaci@fer.hr](mailto:sluzbenik.osobnipodaci@fer.hr) ili mogu podnijeti prigovor Agenciji za zaštitu osobnih podataka, Selaska cesta 136, Zagreb, na adresu elektroničke pošte [azop@azop.hr](mailto:azop@azop.hr).

Potpisom potvrđujem da sam pročitao sve gore navedeno, da se s time slažem te da pristajem sudjelovati u istraživanju.

Slika 3.1 Primjer obrasca za prikaz konačnih rezultata

Prema dobivenim podacima rađena je klasifikacija studenata koristeći popis imena studenata i njihovih JMBAG-ova. Na početku se iz imena i prezimena studenta uklanjaju prazni kvadratići nakon završetka prezimena s pretpostavkom da ih neuronska mreža vrlo dobro klasificira. S obzirom na broj znakova u imenu i prezimenu, iz baze studenata selektiraju se mogući studenti koji imaju jednak broj znakova u imenu i prezimenu. Koristeći Levenshtein-ovu udaljenost izračunate su udaljenosti dobivenog imena i imena svakog preostalog studenta. Levenshtein-ova udaljenost je mjera sličnosti između dvije „riječi“ koja se računa kao broj potrebnih brisanja znaka, zamjene znaka ili dodavanja znaka da bi se jedna riječ transformirala u drugu[15]. Studenti s najmanjom Levenshtein-ovom udaljenošću izabrani su za daljnje izvođenje programa. Nad tim studentima izračunata je Levenshtein-ova udaljenost njihovih JMBAG-ova i JMBAG-a dobivenog klasifikacijom rukom pisanih brojeva ('0062245534'). Student/i s minimalnom udaljenošću JMBAG-ova je/su krajnji rezultat klasifikacije. S obzirom na veličinu baze studenata, koja sadrži oko 300 njih, pomoću ovakvog algoritma uvijek je uspješno određen samo jedan student. Ovakvim pristupom postignuta je stopostotna točnost klasifikacije studenta, odnosno određivanje studenta kojemu pripada ispitni obrazac. Rezultat klasifikacije studenta s prikazanog obrasca je:

```
Najvjerojatniji student:[{'ime': 'MARINKO ĆUTUK', 'jmbag': '0062245834'}]
```

Rezultati ekstrakcije podataka s petnaest ispitnih obrazaca prikazani su u nastavku, na način da „Rezultat klasifikacije“ predstavlja podatke klasificirane neuronskim mrežama, a „Najvjerojatniji student i njegovi podaci“ predstavlja nađenog studenta u bazi podataka prema rezultatu klasifikacije.

```
Rezultat klasifikacije: {'ime': 'AARAO OAAKNVIC', 'jmbag': '0036115315', 'brzad': '13', 'brbod_broj': '13', 'brbod': 15}
```

```
Najvjerojatniji student i njegovi podaci: [{'ime': 'MARKO MARKOVIĆ', 'jmbag': '0036515515', 'brzad': '15', 'brbod_broj': '15', 'brbod': '15'}]
```

```
Rezultat klasifikacije: {'ime': 'EILVP JAEAENOA', 'jmbag': '0036518112', 'brzad': '05', 'brbod_broj': '12', 'brbod': 12}
```

```
Najvjerojatniji student i njegovi podaci: [{'ime': 'FILIP GREBENAR', 'jmbag': '0036515712', 'brzad': '12', 'brbod_broj': '12', 'brbod': '12'}]
```

```
Rezultat klasifikacije: {'ime': 'ANTOMIO RRMDERAB', 'jmbag': '0093610294', 'brzad': '01', 'brbod_broj': '11', 'brbod': 15}
```

Najvjerojatniji student i njegovi podaci: [{'ime': 'ANTONIO BANDERAS',  
'jmbag': '0098610031', 'brzad': '31', 'brbod\_broj': '15', 'brbod': '15'}]

Rezultat klasifikacije: {'ime': 'OAROU ORNOIC',  
'jmbag': '0036866831', 'brzad': '51', 'brbod\_broj': '16', 'brbod': 15}

Najvjerojatniji student i njegovi podaci: [{'ime': 'JAKOV DRUMIĆ',  
'jmbag': '0036515631', 'brzad': '31', 'brbod\_broj': '15', 'brbod': '15'}]

Rezultat klasifikacije: {'ime': 'MCLE AOTJC',  
'jmbag': '0055516187', 'brzad': '63', 'brbod\_broj': '00', 'brbod': 9}

Najvjerojatniji student i njegovi podaci: [{'ime': 'MILE ANTIĆ',  
'jmbag': '0036516873', 'brzad': '73', 'brbod\_broj': '09', 'brbod': '9'}]

Rezultat klasifikacije: {'ime': 'NIROKINA RRDIMIC',  
'jmbag': '0026522125', 'brzad': '25', 'brbod\_broj': '69', 'brbod': 9}

Najvjerojatniji student i njegovi podaci: [{'ime': 'NIKOLINA BUDIMIR',  
'jmbag': '0036522425', 'brzad': '25', 'brbod\_broj': '09', 'brbod': '9'}]

Rezultat klasifikacije: {'ime': 'SRIJTINA ZALA',  
'jmbag': '0051518321', 'brzad': '89', 'brbod\_broj': '05', 'brbod': 5}

Najvjerojatniji student i njegovi podaci: [{'ime': 'KRISTINA ČULE',  
'jmbag': '0036515321', 'brzad': '21', 'brbod\_broj': '05', 'brbod': '5'}]

Rezultat klasifikacije: {'ime': 'OORILSKU NAMUKAKIC',  
'jmbag': '0066513233', 'brzad': '53', 'brbod\_broj': '3', 'brbod': 5}

Najvjerojatniji student i njegovi podaci: [{'ime': 'TOMISLAV REZUNOVIĆ',  
'jmbag': '0036517253', 'brzad': '53', 'brbod\_broj': '5', 'brbod': '5'}]

Rezultat klasifikacije: {'ime': 'IVO AUDRIC',  
'jmbag': '0363537633', 'brzad': '62', 'brbod\_broj': '09', 'brbod': 8}

Najvjerojatniji student i njegovi podaci: [{'ime': 'IVO ANDRIĆ',  
'jmbag': '0362577455', 'brzad': '47', 'brbod\_broj': '08', 'brbod': '8'}]

Rezultat klasifikacije: {'ime': 'IIRKO MILILNVIC',  
'jmbag': '0267667215', 'brzad': '16', 'brbod\_broj': '00', 'brbod': 0}

Najvjerojatniji student i njegovi podaci: [{'ime': 'MIRKO MILIČEVIĆ',  
'jmbag': '1248567210', 'brzad': '16', 'brbod\_broj': '00', 'brbod': '0'}]

Rezultat klasifikacije: {'ime': 'ANA AARIE',  
'jmbag': '0012345669', 'brzad': '63', 'brbod\_broj': '00', 'brbod': 0}



Najvjerojatniji student i njegovi podaci: [{'ime': 'ANA MARIĆ', 'jmbag': '0012345649', 'brzad': '49', 'brbod\_broj': '00', 'brbod': '0'}]

Rezultat klasifikacije: {'ime': 'IVAN IVANKOVIC', 'jmbag': '6381613425', 'brzad': '58', 'brbod\_broj': '12', 'brbod': 12}

Najvjerojatniji student i njegovi podaci: [{'ime': 'IVAN IVANKOVIĆ', 'jmbag': '4281213428', 'brzad': '28', 'brbod\_broj': '12', 'brbod': '12'}]

Rezultat klasifikacije: {'ime': 'NAAIJAT SVCAC', 'jmbag': '0463133110', 'brzad': '21', 'brbod\_broj': '11', 'brbod': 11}

Najvjerojatniji student i njegovi podaci: [{'ime': 'MARIJAN ŽUTIĆ', 'jmbag': '6263123891', 'brzad': '91', 'brbod\_broj': '11', 'brbod': '11'}]

Rezultat klasifikacije: {'ime': 'EELRAM EADDM', 'jmbag': '0036326919', 'brzad': '03', 'brbod\_broj': '09', 'brbod': 9}

Najvjerojatniji student i njegovi podaci: [{'ime': 'BEDRAN VADUN', 'jmbag': '0036476910', 'brzad': '03', 'brbod\_broj': '09', 'brbod': '9'}]

Rezultat klasifikacije: {'ime': 'IOZL MDLOB', 'jmbag': '8036181323', 'brzad': '89', 'brbod\_broj': '0', 'brbod': 5}

Najvjerojatniji student i njegovi podaci: [{'ime': 'IGOR GOLUB', 'jmbag': '2456781385', 'brzad': '85', 'brbod\_broj': '5', 'brbod': '5'}]

## 4. Zaključak i daljnji rad

U ovom radu opisan je postupak ekstrakcije podataka s ručno ispunjenih obrazaca. Prepoznavanje rukom pisanih znakova danas je dobro znan i rješiv problem, a bitan dio uspješnog rješenja čine dobri podaci. Segmentacija obrazaca i dobra priprema podataka čini najveći dio ovog programskog rješenja i pokazala se kao vrlo složen problem. Iako postoje pouzdani i balansirani skupovi podataka kao što je MNIST [1], model neuronske mreže treniran nad tim podacima pokazao se nezadovoljavajućim kod klasificiranja naših segmenata zbog razlika u podacima. Radi povećanja točnosti klasifikacije, konstruirani su skupovi za treniranje neuronskih mreža iz prikupljenih obrazaca. Modeli neuronskih mreža trenirani nad relativno malom skupu naših podataka (ukupno oko 3000) dali su za 10% bolje rezultate nego modeli neuronskih mreža trenirani na MNIST skupu podataka (ukupno 60000) zbog kompatibilnosti podataka. Točnost naših modela je 70% za brojeve i 65% za slova i isključivo takvom klasifikacijom ne bi bilo moguće odrediti studenta kojemu pripada ispitni obrazac, ali uz dostupnu bazu studenta klasifikacija se pokazala 100% točna, odnosno svaki je obrazac uspješno pridružen točnom i samo jednom studentu.

Primjenom OCR-a kod ispitnih obrazaca našeg fakulteta, asistentima i profesorima olakšao bi se i ubrzao proces unošenja bodova u sustav, ali i studentima bi se smanjilo vrijeme čekanja na rezultate. Daljnjim radom na ovom programskom rješenju te prikupljanjem ispunjenih obrazaca, baza rukom pisanih znakova bi se obogatila te bi se i preciznost modela neuronskih mreža povećala. S dovoljno velikom bazom bilo bi moguće postići preciznost od preko 90% u prepoznavanju znakova te bi takvo programsko rješenje bilo pouzdano u stvarnom životu, prvenstveno kod ispita iz matematike i srodnih kolegija, a kasnije razradom univerzalnih obrazaca i kod ostalih kolegija na FER – u. Ukoliko bi se postigla velika preciznost ove neuronske mreže broj pogrešaka kod unosa podataka velikog broja studenata bio bi manji od ručnog unosa.

## Literatura

- [1] MNIST Database URL: <http://yann.lecun.com/exdb/mnist/>
- [2] Bojana Dalbelo Bašić, Marko Čupić, Jan Šnajder, Umjetne neuronske mreže, materijali predmeta Umjetna inteligencija, 2020.  
URL:  
[https://www.fer.unizg.hr/download/repository/UI\\_12\\_UmjetneNeuronskeMreze\[1\].pdf](https://www.fer.unizg.hr/download/repository/UI_12_UmjetneNeuronskeMreze[1].pdf)
- [3] OpenCV Documentation, URL: <https://opencv.org/>
- [4] Softmax aktivacijska funkcija: [https://en.wikipedia.org/wiki/Softmax\\_function](https://en.wikipedia.org/wiki/Softmax_function)
- [5] Bojana Dalbelo Bašić, Marko Čupić, Jan Šnajder, Umjetne neuronske mreže, Zagreb, svibanj 2008.
- [6] Tuytelaars, T. i Mikolajczyk, K. Local Invariant Feature Detectors: A Survey. Foundations and Trends in Computer Graphics and Vision, 3 (2007)
- [7] Keras Documentation, URL: <https://keras.io/>
- [8] Tensorflow Documentation, URL [https://www.tensorflow.org/api\\_docs/](https://www.tensorflow.org/api_docs/)
- [9] ReLu aktivacijska funkcija, URL: <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-network>
- [10] Thresholding, URL:  
[https://docs.opencv.org/master/d7/d4d/tutorial\\_py\\_thresholding.html](https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html)
- [11] ORB Documentation, URL:  
[https://docs.opencv.org/3.4/db/d95/classcv\\_1\\_1ORB.html](https://docs.opencv.org/3.4/db/d95/classcv_1_1ORB.html)
- [12] Hough Circles Documentation, URL:  
[https://docs.opencv.org/4.5.2/da/d53/tutorial\\_py\\_houghcircles.html](https://docs.opencv.org/4.5.2/da/d53/tutorial_py_houghcircles.html)
- [13] Eroding and Dilating Documentation, URL:  
[https://docs.opencv.org/3.4/db/df6/tutorial\\_erosion\\_dilatation.html](https://docs.opencv.org/3.4/db/df6/tutorial_erosion_dilatation.html)
- [14] Removing horizontal lines in image, Stackoverflow rasprava,  
URL:<https://stackoverflow.com/questions/46274961/removing-horizontal-lines-in-image-opencv-python-matplotlib>
- [15] Levenshtein distance, URL:  
<http://people.cs.pitt.edu/~kirk/cs1501/Pruhs/Spring2006/assignments/editdistance/Levenshtein%20Distance.htm>

## Sažetak

Naslov : Klasifikacija ispita metodama strojnog učenja

Autori: Iva Barišić, Tin Petric

U ovom radu opisana je ekstrakcija podataka i prepoznavanje rukom pisanih znakova na primjeru ispitnih obrazaca. Objasnjena je priprema ispitnih obrazaca za segmentaciju, segmentacija ispitnih obrazaca te procesiranje slika i detekcija elemenata na slici. Korištene su neuronske mreže trenirane na našem skupu podataka za prepoznavanje rukom pisanih znakova. U radu se opisuje prepoznavanje imena i JMBAG-a studenta iz obrasca uz pomoć baze podataka studenata na temelju prepoznatih znakova iz ispitnog obrasca. Programsko rješenje razvijeno je u programskom jeziku Python. Korišten je OpenCV za obradu slika te Keras i Tensorflow za treniranje neuronskih mreža.

Ključne riječi: OCR, segmentacija, klasifikacija, reorijentacija, neuronske mreže, ispitni obrazac, Python, OpenCv, Tensorflow

# Summary

Title: Exam classification by machine learning methods

Authors: Iva Barišić, Tin Petric

In this paper we present a method for handwritten character recognition using neural networks applicable on exam forms. The preparation of exam forms for segmentation, segmentation of exam forms, image processing and detection of elements in the image are all explained in this paper. Neural networks trained on our data set were used to recognize handwritten characters. The paper describes detecting the student's name and JMBAG from the form with the help of the database of students based on the recognized characters from the exam forms. This system was developed using python programming language. We have made use of OpenCV for Image processing and have used Keras and Tensorflow for training the neural network.

Keywords: OCR, segmentation, classification, reorientation, neural networks, exam form, Python, OpenCv, Tensorflow