**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

**Diplomski studij**

OBRADA SLIKE I RAČUNALNI VID

**Skeniranje studentske iskaznice**

**Mato Zečević,**

**prva godina diplmskog studija računarstva - DRD**

SADRŽAJ

[1. UVOD 3](#_Toc127282988)

[2. PREGLED PODRUČJA I PROBLEMATIKE 4](#_Toc127282989)

[3. OPIS ZADATKA I DOBIVENIH REZULTATA 5](#_Toc127282990)

[3.1. Korištene tehnologije 5](#_Toc127282991)

[3.2. Opis rješenja 5](#_Toc127282992)

[3.2.1. Učitavanje slike studentske iskaznice 5](#_Toc127282993)

[3.2.2. Prepoznavanje i oznacavanje texta 5](#_Toc127282994)

[3.2.3. Rukovanje dobivenim tekstkom 6](#_Toc127282995)

[3.2.4. Prepoznavanje lica i spremanje slike studenta 7](#_Toc127282996)

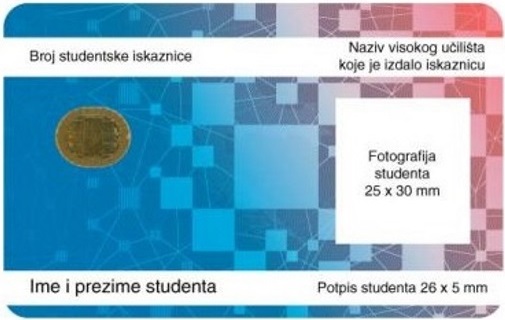
[4. ZAKLJUČAK 10](#_Toc127282997)

[5. LITERATURA 11](#_Toc127282998)

# UVOD

Cilj projektnog zadatka je izrada algoritma koji će iz fotografije studentske iskaznice pronaći i ispisati ime i prezime studenta, naziv učilišta i JMBAG. Algoritam također treba pronaći sliku studenta, izravnati je i spremiti kao sliku.

Za ovakav tip problema najčešće se koriste već postojeći algoritmi za prepoznavanje teksta unutar slika, te algoritmi za prepoznavanje lica. Međutim isti algoritmi služe samo kao alat pomoću kojeg se ostvaruju željeni ciljevi i rezultati. Na slici 1.1. možemo vidjeti izgled studentskih iskaznica koje su korištene unutar algoritma, te slici je prikazano položaj podataka na samoj iskaznici.



**Slika 1.1.** Primjer studentske iskaznice

# PREGLED PODRUČJA I PROBLEMATIKE

Algoritmi za prepoznavanje teksta unutar slika spomenuti u uvodu se nazivaju OCR [1], odnosno algoritmi za optičko prepoznavanje znakova. Optičko prepoznavanje znakova razvilo se iz potrebne slijepih osoba za korištenjem računala. U današnje vrijeme postoji čitav niz gotovih metoda i načina na koji se OCR može ostvariti.

OCR se sastoji od tri dijela: pretprocesiranje, prepoznavanje znakova i postprocesiranje. Cilj pretprocesiranja je pripremiti sliku za prepoznavanje znakova tako što će se obraditi na način da bude što čitljivija. To uključuje postupke kao što su binarizacija, uklanjanje šuma i linija, segmentacija, normalizacija i slično. Prepoznavanje znakova koristi dvije najčešće metode: prepoznavanje uzoraka i međusobne veze između slika. Postprocesiranje se odnosi na poboljšanje preciznosti OCR-a nakon prepoznavanja znakova. Primjer jednog takvog postupka je postavljanje "whitelist-a" ili liste dozvoljenih izlaza za prepoznavanje znakova. Na primjer, ako se prepoznava riječi, dozvoljeni izlazi bi mogli biti samo riječi hrvatskog jezika, a ako se prepoznavaju brojevi, dozvoljeni izlazi bi mogli biti samo znamenke od 0 do 9.

Zatim problematika algoritama za prepoznavanje lica unutar fotografija je identificiranje i označavanje ljudskih lica na fotografijama ili video zapisima. Glavne probleme uključuju točnost prepoznavanja, posebno ako su učinjene značajne promjene na licu (poput brade, naočala, itd.), ako su uvjeti svjetlosti loši ili ako se lice nalazi pod nezgodnim kutem. Također, neke tehnologije prepoznavanja lica imaju problema s razlikovanjem ljudi različitih etničkih skupina ili spola, što može dovesti do diskriminacije.

# 3. OPIS ZADATKA I DOBIVENIH REZULTATA

## 3.1. Korištene tehnologije

Tehnologije korištene u izradi algoritma su programski jezik Python te biblioteke dostupne unutar programskog jezika. Za samo prepoznavanje znakova odabran je EasyOCR [2] koji je *open source* engine te omogućuje korisnicima da izvuku tekst iz slika i skeniranih dokumenata. Dizajniran je za prepoznavanje teksta na mnoštvo jezika od kojih je jedan i hrvatski, također postoji mogućnost i izdvajana teksta iz rukom napisanih dokumenata i čak iz loše kvalitete slika. Postoji i plaćena verzija koja pruža veću preciznost, ali besplatna također ima veliku preciznost. Zatim za prepoznavanje lica korištena je *dlib*[3] biblioteka. Jedna od najpopularnijih biblioteka i alata u svijetu računalnog vida koja sadrži mnoštvo funkcionalnosti od kojih je jedna od glavnih detekcija lica koja je potrebna u projektu.

## 3.2. Opis rješenja

### 3.2.1. Učitavanje slike studentske iskaznice

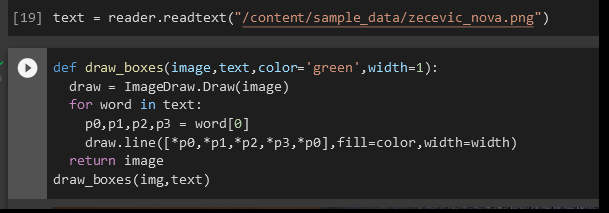
Učitavanje slike ostvareno je korištenjem biblioteke Pillow koja pruža funkcije za obradu slika, te je sam kod pomoću kojega je ostvareno sa učitavanje prikazan je na slici 3.1.



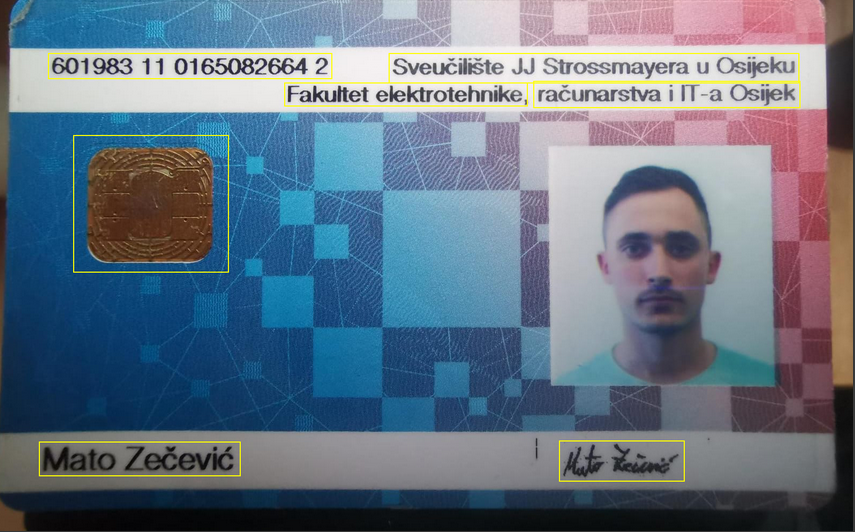
**Slika 3.1.** – Kod za učitavanje slike

### 3.2.2. Prepoznavanje i oznacavanje texta

Prepoznavanje teksta vrši se pomoću funkcije reader unutar biblioteke EasyOCR, to je glavna funkcija u biblioteci i omogućava korisnicima da prepoznaju tekst sa slika i učitavaju ga u obliku teksta. Za označavanje teksta korištena je funkcija *draw* iz biblioteke Pillow. Funkcija *draw* služi za crtanje različitih elemenata na slikama uključujući linije, kružnice, pravokutnike i slično. Funkcija *draw* radi s objektom *ImageDraw* koji se kreira korištenjem funkcije *ImageDraw.Draw()* i prima sliku kao ulazni argument. Kada se stvori objekt ImageDraw, korisnici mogu koristiti metode za crtanje elemenata na slikama. Sam kod pomoću kojega je ostvareno prepoznavanje i označavanje teksta prikazan je na slici 3.2. , a rezultat na slici 3.3.   
Na slici 3.3. također je prikazano kako je *microchip* također prepoznat kao tekst, zbog čega možemo uvidjeti nepreciznost EasyOCR-a.



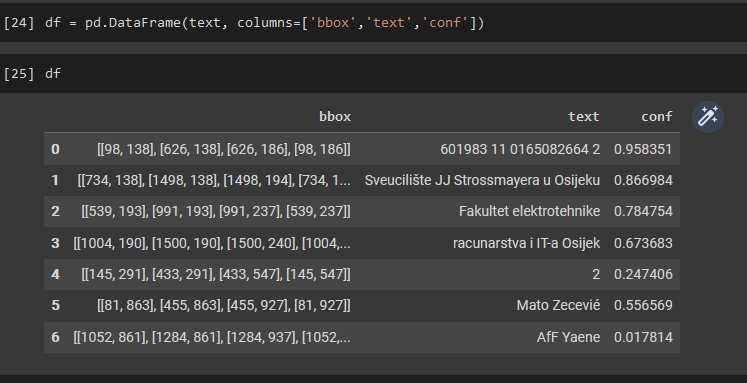
**Slika 3.2.** - Kod za prepoznavanje i označavanje teksta



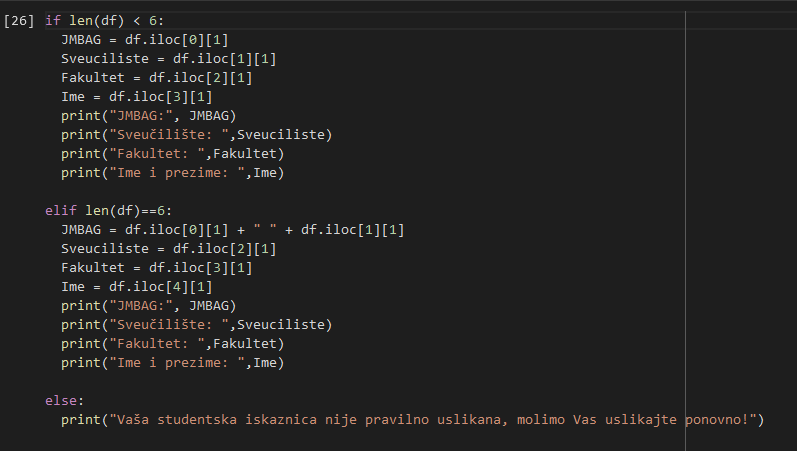
**Slika 3.3.** - Rezultat prepoznavanja i označavanja teksta

### 3.2.3. Rukovanje dobivenim tekstkom

Budući da postoje moguće pogreške u prepoznavanju teksta te provodeći algoritam na većem broju slika, ideja je bila spremiti tekst unutar tablice koju pruža biblioteka Pandas [4] sa svojom funkcijom *DataFrame.* Podaci su spremljeni u 3 stupca od kojih prvi predstavlja lokaciju unutar slike, drugi prepoznati text te treći točnost prepoznavanja. Primjer tablice je prikazan na slici 3.4. Kada su podaci spremljeni u tablicu, s podacima s podacima se rukovalo pomoću blokova naredbi prikazanih na slici 3.5.



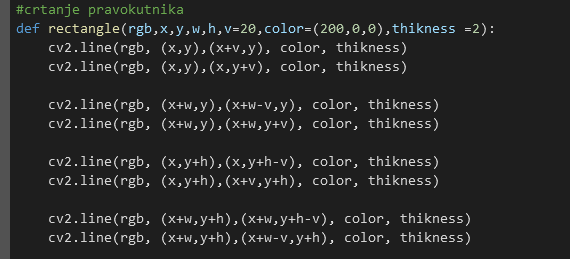
**Slika 3.4.** - Prikaz DataFrame tablice ispunjene tekstom dobivenim iz slike



**Slika 3.5.** - Prikaz bloka naredbi za rukovanje podacima

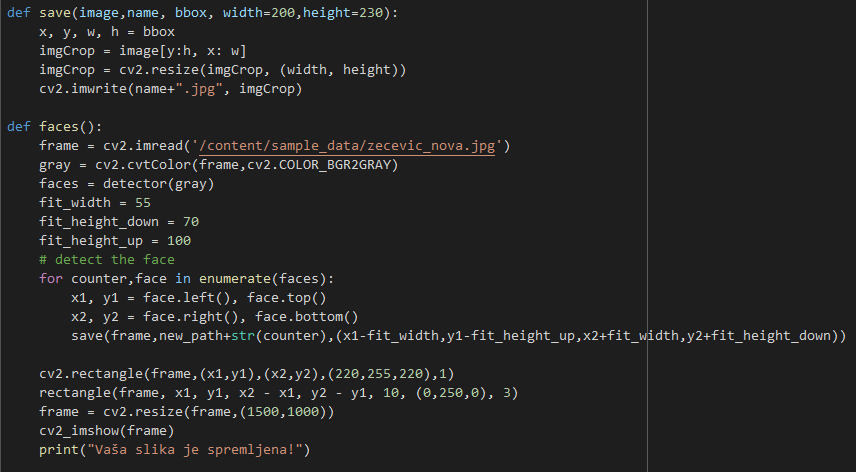
### 3.2.4. Prepoznavanje lica i spremanje slike studenta

Prepoznavanje lica studenta sa studentske iskaznice vrši se pomoću funkcije *get\_frontal\_face\_detector()* iz biblioteke *Dlib*. To je funkcija koja prima ulaznu sliku i vraća listu koordinata svakog detektiranog lica na slici. Također, dodana je funkcija pomoću koje se crta pravokutnik oko samog lica kao potvrda detekcije (slika 3.6.).



**Slika 3.6.** – Funkcija za definiranje pravokutnika

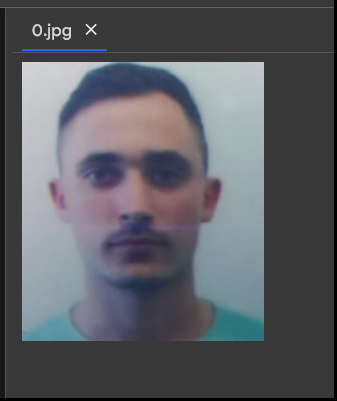
Na slici 3.7. prikazan je kod za spremanje slike studenta, korištena je funkcija *resize* pomoću koje se slika smanjuje na veličinu 200x230.



**Slika 3.7.** – Funkcija za spremanje slike studenta



**Slika 3.8.** – Rezultat funkcije *get\_frontal\_face\_detector() i* funkcije *rectangle()*



**Slika 3.9.** – Rezultat funkcije *faces()*

# 4. ZAKLJUČAK

Korištenjem biblioteka poput dlib-a i EasyOCR-a, moguće je automatski prepoznati lica i tekst iz digitalnih medija kao što su slike, u ovom slučaju slike studentske iskaznice. Međutim iako je s ove dvije biblioteke moguće napraviti izrazito učinkovite sustave prepoznavanja lica i teksta, treba imati na umu da ove tehnologije također imaju svoja ograničenja i nedostatke. Neki od nedostataka su pogreška prepoznavanja i mogućnost zloupotrebe podataka učitanih sa studentske iskaznice. Kvaliteta slika, kut pod kojim se vidi prikazani tekst ili sama fotografija i slično, mogu predstavljati veliki problem u prepoznavanju i uspješnosti algoritma.

Također, iste biblioteke zahtijevaju određenu količinu dostupnih resursa, poput procesorskog vremena i memorije te to može predstavljati problem u situacijama s ograničenim resursima.

Sam projektni zadatak predstavlja jako dobru podlogu za kreiranje klijentske aplikacije u kojoj bi se Studentske iskaznice mogle digitalizirati i koristiti pomoću *smartphone-a* ili *smartwatch-a*.

# 5. LITERATURA

[1] <https://en.wikipedia.org/wiki/Optical_character_recognition> (14.2.2023.)

[2] <https://pyimagesearch.com/2020/09/14/getting-started-with-easyocr-for-optical-character-recognition/> - EasyOCR

[3] <http://dlib.net/> - dlib biblioteka

[4] <https://pandas.pydata.org/> - Pandas  
[5] <http://stackoverflow.com/> - brojni primjeri