

## Detekcija registarskih tablica na vozilima pomoću OCR algoritma i detekcije ivica na slici

---

*Ovaj rad evaluira algoritme za prepoznavanje registarskih tablica upoređivanjem algoritama detekcije ivica i detekcije teksta kao i to kako oni zajedno rade. Za detekciju ivica korišćen je Kenijev detektor ivica (Canny Edge Detection) sa različitim tipovima preprocesinga za detekciju ivica dok je za detekciju teksta korišćena već istrenirana duboka neuronska mreža u ulozi algoritma za optičko prepoznavanje karaktera (OCR). Ova dva metoda su kombinovana kako bi se isekao deo slike na kojem se nalazi isključivo tablica bez ikakvog teksta sa strane radi sigurnog i tačnog čitanja registarske oznake. Za testiranje je korišćena je baza koja se sastojala od slika automobila na kojima se nalaze tablice sa područja Evrope. Uspešnost algoritma je procenjena Žakarovim indeksom sličnosti, tj. odnosom preseka i unije (IOU) labela u bazi podataka i rezultata dobijenih algoritmom. Na osnovu ovog odnosa, algoritam je na celoj bazi imao uspešnost od 62%, dok je odziv (udeo tačnih pozitiva) na test bazi iznosio 86.5%. Analizom rezultata zaključeno je da se svi rezultati sa Žakarovim indeksom većim od 70% mogu klasifikovati kao detektovane registarske oznake, u kom slučaju tačnost detekcije ovog algoritma iznosi 84%.*

---

### Uvod

Prouzrokovano ogromnim brojem vozila koja se danas kreću po celom svetu, pojavljuje se potreba da se svako vozilo može lako i efikasno prepoznati automatizovanim sistemom. Danas

postoji mnogo različitih algoritama i za čitanje i detekciju registarskih oznaka koji nalaze primenu na naplatnim rampama, autoputevima, graničnim prelazima, sistemima za kontrolu saobraćaja, sistemima za praćenje ukradenih kola itd. Skoro svaka država ima svoj sistem automatskog prepoznavanja registarskih oznaka, a neki od njih su LPR u SAD-u, CARINA u Mađarskoj, SeeCar u Izraelu itd. Postavlja se pitanje koji od ovih algoritama su najbolji u kojim primenama, to jest koliko su brzi, a koliko tačni u različitim uslovima i da li ih je moguće još poboljšati. U poslednje vreme ova tema je jako popularna i pojavljuju se radovi sa sve raznovrsnijim metodama i pristupima datom problemu. U ovom radu se kao jedna od glavnih metoda selekcije tablice koristi algoritam za optičko prepoznavanje karaktera (optical character recognition – OCR), u daljem tekstu OPK, koji u kombinaciji sa tehnikom za detekciju ivica (edge detection) određuje lokaciju registarske oznake na slikama.

### Metod

Postupak detektovanja je podeljen na nekoliko važnih algoritama koji dopunjujući jedan drugog dolaze do detekcije registarskih oznaka sa fotografija vozila (slika 1).

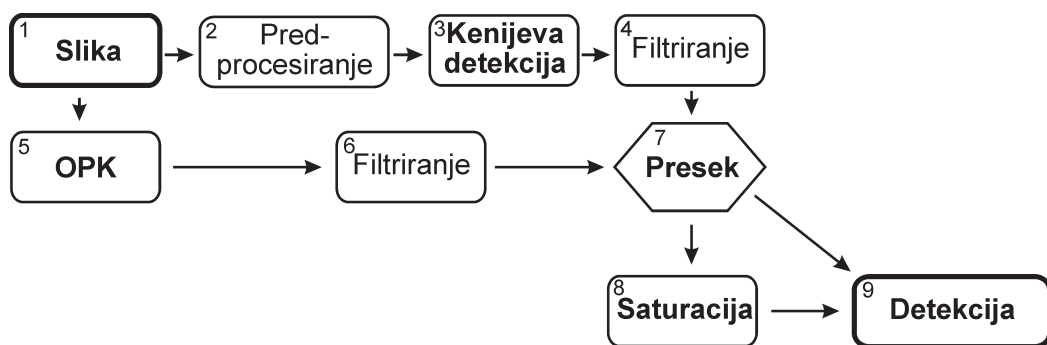
Na početku se unosi slika na kojoj se nalazi vozilo sa tablicom u prirodnom okruženju i prebacuje se u grayscale format za dalje procesiranje. Nakon pripreme, na slici se pronalaze ivice pomoću Kenijevog algoritma za detekciju ivica

---

*Dimitrije Andžić (2002), Beograd, Bulevar Mihaila Pupina 7, učenik 2. razreda Računarske gimnazije u Beogradu*

*Filip Stevanović (2002), Zemun, Užička 8, učenik 3. razreda Zemunske gimnazije*

*MENTORKA: Andrea Ćirić, studentkinja Elektrotehničkog fakulteta Univerziteta u Beogradu*



Slika 1. Blok šema rada algoritma

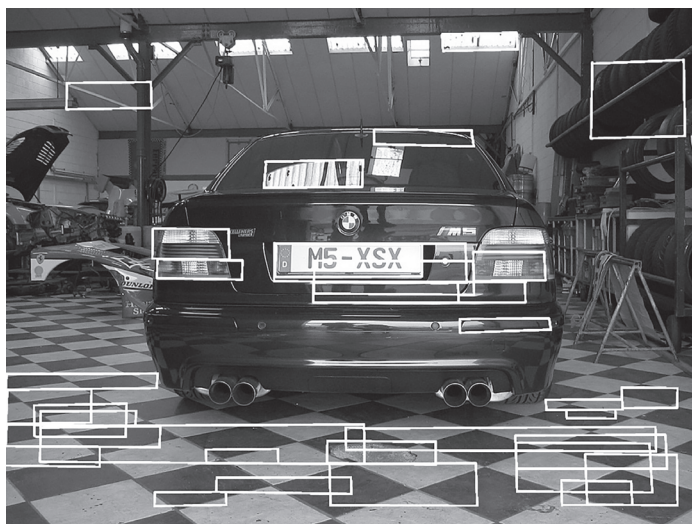
Figure 1. Block scheme of the algorithm: 1 – Image, 2 – Preprocessing, 3 – Canny edge detection, 4 – Filtering, 5 – OCR, 6 – Filtering, 7 – Intersection, 8 – Saturation, 9 – Detection.

(Canny edge detection algorithm; Sharifi *et al.* 2002), tako da nakon ove faze slika sadrži samo bele i crne piksele, gde beli pikseli predstavljaju ivice na crnoj pozadini. S obzirom da je tablica zatvorena kontura, na tako dobijenoj slici se pronalaze sve zatvorene konture koje su slične pravougaoniku, a zatim se zbog specifičnog odnosa stranica tablice, pravougaonici dodatno selektuju i dobijaju potencijalni kandidati za nju. Nakon izdvajanja kandidata, potrebna je njihova dalja selekcija kako bi se izdvojili samo pravougaonici na kojima se nalazi tablica, što se u ovom slučaju rešava algoritmom optičkog prepoznavanja karaktera (OPK) koji pronalazi karaktere na celoj slici i te površine označava pravougaonicima (White i Rohrer 1983). Pošto je standard da se na tablici nalaze karakteri, proizilazi da je kontura koja se preseca sa OPK pravougaonikom tražena tablica. Ukoliko ovaj presek ne postoji, slika se prebacuje u HSV format i za svaki kandidovani pravougaonik se izračuna prosečna saturacija, te se kandidat sa najmanjom saturacijom vraća kao detektovana oznaka (Zhai *et al.* 2010). U okviru ovih koraka algoritma postoji više različitih metoda o kojima će dalje biti reči.

## Detekcija ivica na slici i izdvajanje kontura

Prvi deo algoritma koji se izvršava na slici jeste detekcija ivica, a zatim pronalaženje kontura koje liče na pravougaonik. Projekat je rađen

u Python programskom jeziku i korišćeni su filteri i operacije sa slikama iz biblioteke OpenCV (CVisionDemy 2019). Detekcija ivica rađena je dvema metodama za koje su odvojeno mereni rezultati. Prvi metod (metod pronalaženja jačih ivica) počinje normalizacijom slike, koja se potom prebacuje u grayscale. Nakon toga se nalazi srednja vrednost intenziteta rezultirajuće slike. Zatim se primeni blur filter nad slikom, a potom i binarni trešhold gde je granica već izračunata srednja vrednost intenziteta piksela. Zatim se primenjuje Kenijev algoritam i ekstraktuju ivice. Drugi metod (metod pronalaženja svih ivica) počinje sa prebacivanjem slike u HSV format, koji se sastoji iz odvojenih kanala za nijansu boje (hue), zasićenost (saturation) i sjaj (value), od čega se ekstrahuje sjaj, na kome se registarska tablica može lakše detektovati usled njene veoma male saturacije i bele boje, što se odražava i na sam sjaj slike gde je ona potpuno crno-bela. Na value kanalu dobijene slike u HSV formatu primenjujemo *tophat* i *blackhat* filter sa kernelom  $3 \times 3$ . Nakon toga se sabiru *tophat* i sjaj slike, kako bi se dobili prošireni objekti na slici, pa se od proširenih objekata oduzme *blackhat* filter slike i dobiju se zamućeni obrisi tih objekata. Nad rezultatom se izvršava *Gausovo zamućenje* (Gaussian blur) i potom Gausov adaptivni prag (Gaussian adaptive threshold). Zatim se sa dobijene slike pronalaze ivice pomoću *Kenijevog algoritma* (Canny edge detector) i rezultujuća



Slika 2. Prikaz kandidata nakon algoritma za detekciju ivica

Figure 2. Candidates for license plate after edge detection algorithm

slika predstavlja izdvojene ivice sa početne slike. Nakon izdvajanja ivica, sa slike su ekstrahirane konture koje imaju oblik sličan pravougaoniku. Prolaskom kroz sve zatvorene konture proveravan je oblik i broj stranica te su se tako određivale konture oblika pravougaonika. Pošto evropske tablice imaju fiksni odnos između stranica ne mogu sve konture biti kandidati za tablicu. Na pronađene pravougaonike primenjivani su različiti odnosi stranica na osnovu čega su neki kandidati eliminisani. Kako je evropski standard za odnos stranica 1:5 a slike u bazi su specifične rezolucije, odlučeno je da se ispitaju eliminacioni odnosi od 1:1.5 pa do 1:5 kako bi se pronašao odnos koji nikada ne eliminiše kandidata sa tablicom i koji eliminiše obilan šum dobijen detekcijom ivica kako bi se skratilo vreme izvršenja algoritma (Al-Ghaili *et al.* 2013). Na slici 2 je prikazan primer fotografije i konture koje su na njoj pronađene. Kandidati koji su prošli eliminacioni odnos obeleženi su belim pravougaonicima dok su kandidati označeni kao šum uklonjeni sa slike.

## Optičko prepoznavanje karaktera

Nakon izdvajanja kontura detekcijom ivica na slici, dobijen je niz pravougaonika koji predstavljaju kandidate, odnosno moguće tablice, koje treba selektovati da bi se izdvojio samo jedan pravi kandidat na kome se stvarno nalazi tablica. Kako bi se ovo postiglo, koristi se op-

tičko prepoznavanje karaktera (OPK), koje treba da, u saradnji sa detekcijom ivica, izabere pravog kandidata (White i Rohrer 1983). OPK je u radu implementiran preko duboke neuronske mreže koja za ulaz prima sliku koju obrađuje i željeni koeficijent sigurnosti detekcije karaktera izražen u procentima, dok na izlaz vraća koordinate pravougaonika oko karaktera koje je pronašao (slika 3).

S obzirom na zadavanje željenog koeficijenta sigurnosti detekcije na ulazu, razdvojila su se dva metoda za korišćenje rezultata OPK. Prvi metod se zasnivao na pronalaženju fiksne vrednosti za koeficijent sigurnosti, koja bi se uvek koristila na svim slikama. Drugi metod se zasnivao na tome da se za svaku sliku koeficijent sigurnosti konstantno smanjuje, dok se ne dođe do vrednosti na kojoj OPK vraća bar jednog kandidata. Drugi metod zahteva više vremena i procesorskog rada, ali je potencijalno uspešnija varijanta za pronalaženje pravog kandidata. Upravo iz tih razloga su ova dva metoda međusobno i poređena.

## Zajednički kandidati algoritma detekcije ivica i OPK

Na osnovu rezultata detekcije ivica na slici i rezultata primene OPK algoritma, cilj je detektovati samo pravougaonik sa tablicom. Stoga se broj kandidata još treba suziti, što se postiže udruživanjem OPK algoritma i algoritma detekcije ivica. Rezultati koje vraćaju oba algoritma za

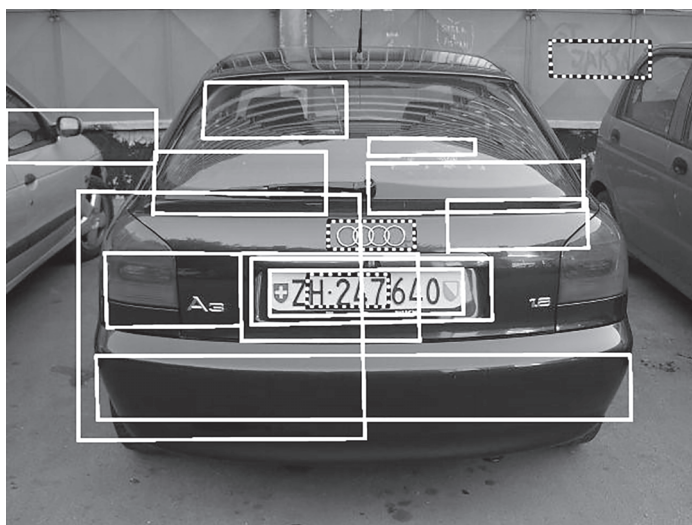


Slika 3. Rezultat rada OPK na primeru slike iz baze

Figure 3. OCR result on database picture example

istu sliku ilustrirani su na slici 4. Na slici su kandidati dobijeni detekcijom ivica i izdvajanjem kontura obeleženi pravougaonicima oivičenim punim belim linijama, a kandidati dobijeni OPK algoritmom su označeni pravougaonicima sa isprekidanim ivicama. S obzirom na to da i jedan i drugi metod vraćaju kandidate u obliku selektovanih pravougaonika, gledaju se međusobni preseki dobijenih izlaza oba algoritma, i na osnovu tih preseka se određuje pozicija tablice.

U ovom radu se kao metod udruživanja koristi bodovanje preseka dobijenih pravougaonika, tako da procenat preseka pravougaonika izdvojenog detekcijom ivica sa OPK pravougaonicima i OPK pravougaonika sa pravougaonicima izdvojenih detekcijom ivica vrede određen broj poena, koji se iteracijom kroz sve preseke dodaju na broj poena svakog kandidata. Kandidat sa najvećim skorom bi se tretirao kao tablica (Fahmy 1994). Koeficijenti za dva pomenuta preseka dobijeni su poređenjem rezultata celo-

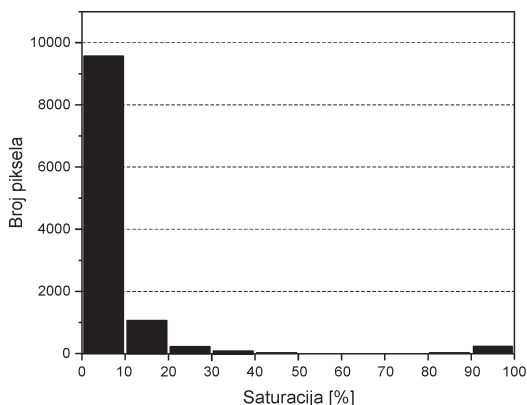


Slika 4. Presek kandidata algoritma detekcije ivica i OPK rezultata

Figure 4. Extracting the right candidate by finding intersections between edge detection candidates and OCR candidates



kupnog algoritma, kada se odvojeno koriste oba spomenuta pristupa. Pored koeficijenata za bodovanje, potrebno je definisati i način na koji se preseki kandidata pronalaze. U ovom radu su korišćena dva metoda koja su kasnije međusobno upoređena, kako bi se ispitalo koji je precizniji. Prvi metod je podrazumevao nalaženje ukupnog zbira svih preseka jednog kandidata dobijenog detekcijom ivica sa svim kandidatima OPK algoritma, dok je drugi podrazumevao izdvajanje



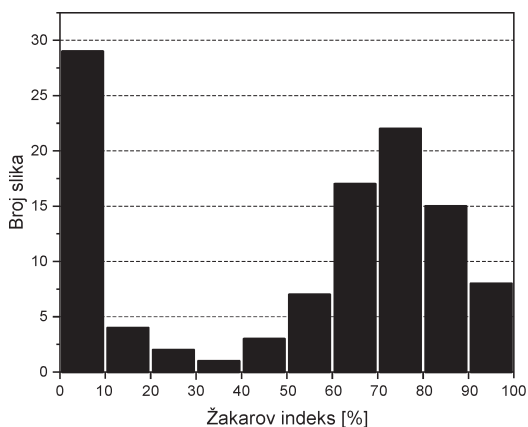
Slika 5. Primer slike na kojoj nema preseka kandidata detekcije ivica i OPK kandidata (gore) i odgovarajuća raspodela na skali saturacije od 0 do 100 procenata u regionu tablice (dole).

Figure 5. Example of image where the algorithm could not find the intersection between OCR and edge detection candidates (top), and saturation distribution of the image part where the license plate is located (bottom).

samo najveće vrednosti od svih preseka, kao broj poena za kandidata. Poređenjem ova dva metoda na analiziranim slikama, primećeno je da OPK algoritam u nekim slučajevima ne detektuje tekst na tablici u celini, već ga deli na nekoliko manjih kandidata, što je uticalo na ispravnost poređenja metoda skorovanja. Stoga je odlučeno da se svi OPK kandidati koji se nalaze na približno istoj x-koordinati i na skoro istoj y-koordinati spoje u samo jednog kandidata. Nakon što je ovaj algoritam primenjen na OPK kandidate, pristupilo se poređenju metoda bodovanja.

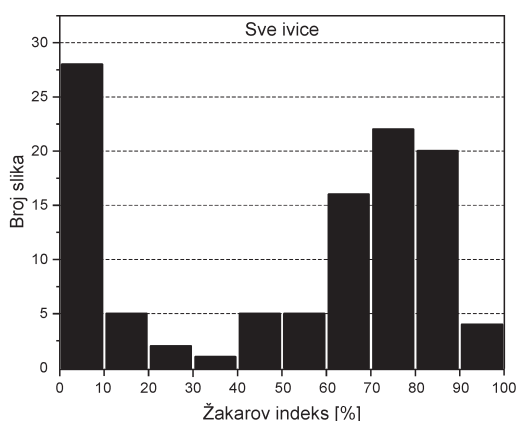
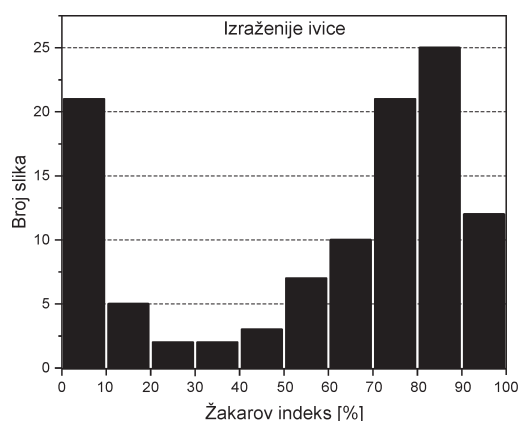
## Histogram i prosečna saturacija kandidata

U dosadašnjem toku procedura podrazumeva se da se na svakoj slici može pronaći bar jedan kandidat kao rezultat preseka OPK i Klenijevog algoritma za detekciju ivica, kako bi opisana metoda skorovanja imala smislenu ulogu u selekciji. Međutim, ipak postoje primeri slika koji ne ispunjavaju ovaj uslov (slika 5), te je broj bodova za svakog kandidata jednak nuli, pa tablicu na taj način nije moguće izdvojiti. Za ovakve slučajeve u ovom radu dalje se koristi poseban metod eliminacije (Zhai *et al.* 2010). Kako na tablici postoje bela pozadina i crni karakteri, i nijedna druga boja (grb se može zanemariti), histogram



Slika 6. Raspodela Žakarovog indeksa kada se koristi početni skelet algoritma

Figure 6. Histogram with IOU results of the first version of the algorithm



Slika 7. Rezultat primene algoritma za pronalaženje samo izraženijih ivica (levo) i algoritma za pronalaženje svih ivica (desno), i odgovarajuće raspodele Žakarovog indeksa (IOU).

Figure 7. Application of the algorithm for finding only the more pronounced edges (left) and the algorithm for finding all the edges (right), and the corresponding distribution of the IOU.

boja za region tablice treba da ima dva pika oko crne i bele boje i nisku vrednost saturacije u HSV skali. Dakle, ukoliko nakon izvršena oba osnovna algoritma ne postoji zajednički kandidat u preseku, tada se za svakog od takvih kandidata traže ove karakteristike, i za tablicu se određuje onaj kandidat koji najbliže zadovoljava navedene kriterijume za saturaciju. Primer saturacije u regionu tablice prikazan je na slici 5.

## Rezultati i diskusija

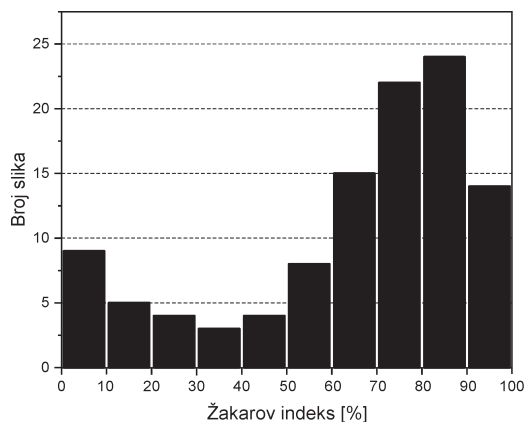
S obzirom da su za različite korake algoritma postojali različiti načini rešavanja podproblema, svaki način je bio implementiran i testiran na

trening bazi tokom nadograđivanja početnog skeleta, a samo konačan algoritam je proveren i ocenjen na test bazi, kako bi se rezultati dobijeni na trening bazi proverili i time se odredila efikasnost celog algoritma. Trening baza (Hill 2016) se sastojala od 108, dok se test baza sastojala od ukupno 100 različitih slika vozila u prirodnom okruženju pri različitim situacijama u saobraćaju, različitom intenzitetu osvetljenja okoline kao i slika različitih kategorija vozila. Slike iz test baze nađene su na internetu kao osnovni primeri za druge tipove algoritama za detekciju registarskih tablica na vozilima, nakon čega su ručno labelirane koordinate piksela gornjeg levog i donjeg desnog ćoška tablice.

Kako bi se metode međusobno poredile, na početku je napravljen skelet algoritma (prva verzija) koji se sastojao samo od kombinacije detekcije ivica i OPK algoritma koji je bio sposoban da vrati rezultat ali ne i da on bude tačan. Na njemu su menjani isključivo delovi koje je trebalo porediti kako bi se dobili validni rezultati. Na slici 6 se mogu videti rezultati očitavanja tablice početnog skeleta algoritma na trening bazi, gde Žakarov indeks (IOU) predstavlja poklapanje dobijenog rezultata sa labelom u bazi koja predstavlja koordinate gornjeg levog i donjeg desnog piksela na delu slike na kome se nalazi tablica.

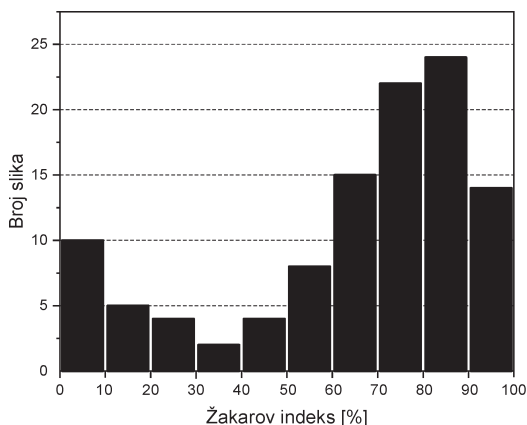
Sa ovog grafika se može zaključiti da početni skelet uspeva da u nekim slučajevima pronalazi delove tablice ali da je broj potpuno neuspešnih detekcija jako veliki. U ovom početnom skeletu nisu bile implementirane ni jedne od poređenih metoda već samo generalni koncepti te je ubacivanjem i upoređivanjem ovih metoda vrednost Žakarovog indeksa bila postepeno povećavana što se vidi na daljim graficima.

Kako se detekcija ivica na slici izvršava na početku algoritma, prvo su poredena dva algoritma specijalizovana za ovaj problem gde je svaki od njih detaljno objašnjen u metodu. Oba algoritma su implementirana na početni skelet i pušteni su da rade na trening bazi kako bi se njihovi rezultati poredili. Na slici 7 mogu se vi-



Slika 8. Raspodela Žakarovog indeksa za slike iz baze sa odnosom stranica tablice 1:1.5

Figure 8. Histogram of IOU for pictures by plate ratio of 1:1.5 between sides of candidate rectangles



Slika 9. Raspodela Žakarovog indeksa za slike iz baze sa odnosom stranica tablice 1:2

Figure 9. Histogram of IOU for every picture from the database with elimination ratio of 1:1.5 between sides of candidate rectangles

deti primeri primene metoda za pronalaženje samo izraženijih ivica (levo) i metoda za pronalaženje svih ivica (desno), sa odgovarajućim raspodelama vrednosti Žakarovog indeksa ispod.

Poređenjem dva histograma na slici 7, može se videti da je rezultat metoda pronalaženja svih ivica bolji u odnosu na metod pronalaženja jačih ivica, iz čega se može zaključiti da drugi metod u nekim slučajevima ne detektuje u potpunosti ivice tablice, pa za dalji tok algoritma ne postoji kandidat koji sadrži tablicu. Takođe, može se primetiti da se rezultat nakon uvedenih algoritama popravio u odnosu na početni skelet algoritma, te je metod pronalaženja svih ivica pripojen skeletu, i dalje poređenje je vršeno sa ovom verzijom skeleta.

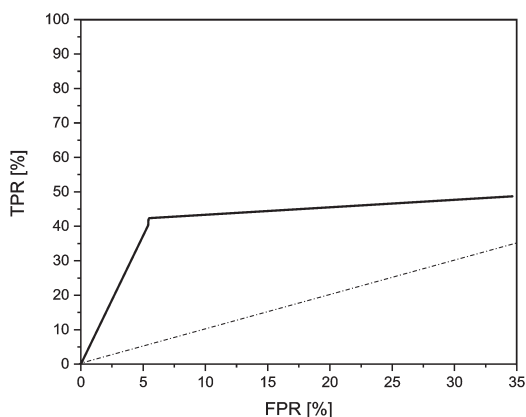
Nakon što su pronađene ivice i određene konture, usledelo je poređenje eliminacionih odnosa za stranice pravougaonih kontura. Za odnose 1:4 i 1:5 dobijen je rezultat od 45% eliminisanih kandidata na kojima se nalazila tablica (što ne bi smelo da se dešava), a za odnos 1:3 dobijena je greška ispod 20% eliminisanih tablica. Za odnose 1:1.5 i 1:2 nije eliminisan ni jedan kandidat sa tablicom, ali su rezultati uključivali određenu količinu šuma koji je smetao pri daljoj selekciji kandidata. Stoga su ova dva odnosa posebno im-

plementirala na finalnu verziju algoritma; dobijeni rezultati prikazani na slikama 8 i 9.

Sa histograma se može videti mala razlika između ova dva odnosa gde odnos 1:1.5 daje Žakarov indeks od 62% dok odnos 1:2 daje rezultat od 61.6%. Iz ovih rezultata zaključeno je da bolji rezultat ima odnos 1:1.5 te je on korišćen u konačnoj verziji algoritma.

Kako se za dobijanje novih kandidata koristio algoritam optičkog prepoznavanja karaktera na slici, bilo je potrebno odrediti željeni koeficijent sigurnosti detekcije karaktera koji se unosi u OPK algoritam, na osnovu čega on pronalazi delove slike u kojima se nalazi tekst i oviču je ga pravougaonici. Da bi se dobila optimalna vrednost koeficijenta sigurnosti OPK algoritma, urađena je ROC (receiver operating characteristic) kriva odnosa tačnih pozitiva (TPR) i lažnih pozitiva (FPR) za različite vrednosti željenog koeficijenta sigurnosti. Za svaku vrednost koeficijenta (od 0 do 1) izračunat je TPR i FPR. Potom je napravljen grafik zavisnosti TPR od FPR deparametrizacijom. Ovako dobijena ROC kriva se može videti na slici 10.

Dobijena ROC kriva ima dva linearna režima: veliki rast i mali rast. Prelomna tačka između njih se dobija za vrednost koeficijenta sigurnosti 0.8. U prvom delu možemo videti veliki porast tačnih pozitiva za mali porast lažnih

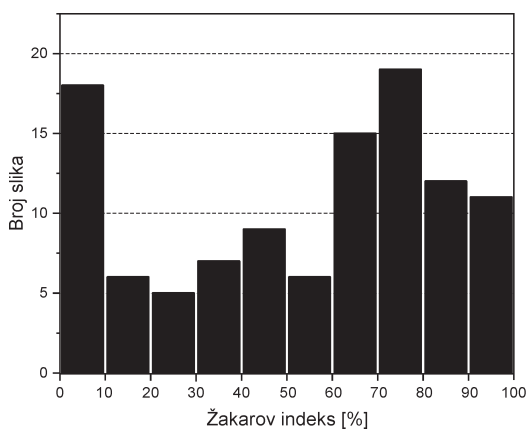


Slika 10. ROC kriva kompletnog algoritma u odnosu na koeficijent sigurnosti detekcije OPK algoritma

Figure 10. ROC curve of full algorithm in regard to the detection security coefficient of OCR algorithm

pozitiva. U drugom delu možemo videti veoma mali porast tačnih pozitiva za veoma veliki porast lažnih pozitiva, i ovo uzimamo kao loš režim za naše primene. Stoga je za optimalan koeficijent sigurnosti OPK algoritma uzeta vrednost 0.8, koja predstavlja poslednju (graničnu) tačku prvog režima. U toj tački algoritam ima najveći TPR (40%) za najmanji FPR (5%). Ovo je dobijeno isključivo korišćenjem OPK algoritma za detekciju, bez ukrštanja sa Kenijevim algoritmom, kako bi se dobili samo rezultati za OPK. Nakon analize primećeno je da metod iterativnog povećavanja koeficijenta sigurnosti OPK algoritma daje bolje rezultate za 8% u konačnom algoritmu, pa je implementiran na početni skelet i dalje se koristio u poređenjima rezultata.

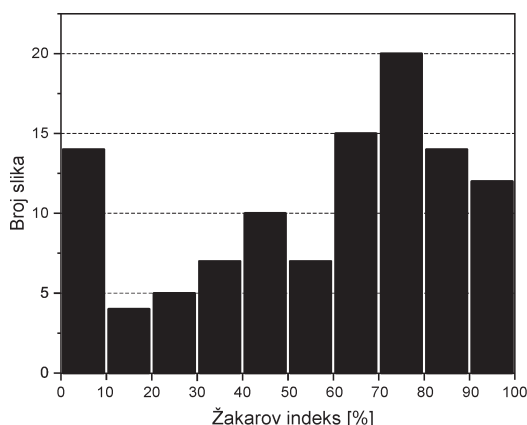
Na samom kraju algoritma, rađeno je poređenje metoda za bodovanje kandidata koji određuju koji kandidat je najverovatnije tablica. Kao i u ostalim slučajevima, oba metoda su odvojeno implementirana na skelet i međusobno upoređena. Rezultati za prvi metod, odnosno metod bodovanja zbira svih preseka OPK kandidata i kandidata dobijenih detekcijom ivica, prikazan je na slici 11, dok je rezultat za metod



Slika 11. Raspodela Žakarovog indeksa pri pronalaženju najvećeg zbira svih preseka algoritma detekcije ivica i OPK algoritma

Figure 11. Histogram of percentage of IOU per every picture from database by the using algorithm of finding the biggest sum of intersections between edge detection algorithm and OCR algorithm





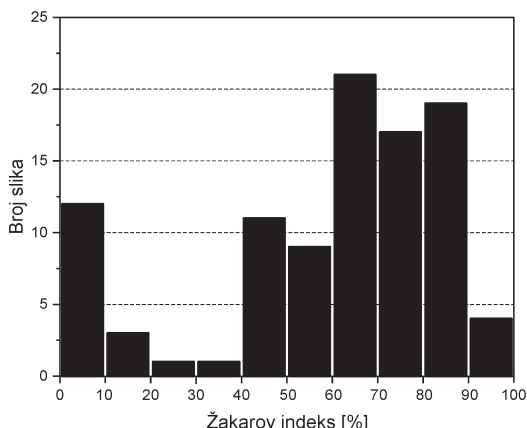
Slika 12. Raspodela Žakarovog indeksa pri pronalaženju samo jednog najvećeg preseka algoritma detekcije ivica i OPK algoritma

Figure 12. Histogram of percentage of IOU per every picture from database by using the algorithm of finding only one biggest intersection between edge detection algorithm and OCR algorithm

bodovanja samo jednog najvećeg preseka kandidata prikazan na slici 12.

Sa histograma se može zaključiti da je metod pronalaženja jednog najvećeg preseka OPK kandidata i kandidata dobijenih detekcijom ivica uspešniji. Takođe, može se primetiti i da uvođenjem sistema bodovanja, na velikom broju primera tablica se uspešnije detektuje, i konačan Žakarov indeks je viši.

Po izvršenom poređenju metoda nadogradnje skeleta algoritma, na skelet je dodat poslednji deo metoda (histogram i prosečna saturacija kandidata), nakon čega je ceo algoritam označen kao konačan rezultat ovog rada i pokrenut na posebnoj test bazi opisanoj na početku ovog odeljka, na osnovu čega je i dobijen konačan rezultat. Konačan Žakarov indeks na test bazi iznosi 62%, dok udeo tačnih pozitiva (TPR) iznosi 86.5% u odnosu na labelu, a udeo lažnih pozitiva (FPR) iznosi svega 3%. Ukoliko bi se gledalo u brojevima koliko tablica je detektovano sa svim karakteristikama na njoj, bez dodatnih karaktera sa strane na celoj test bazi, dobija se procenat uspešnosti od 84%. Raspodela Žakarovog indeksa prikazana je na slici 13.



Slika 13. Raspodela Žakarovog indeksa za sve slike iz baze obrađene konačnim algoritmom

Figure 13. Histogram of percentage of IOU on every picture processed from the database of the final algorithm

## Zaključna diskusija

Analizom dobijenih rezultata može se reći da konačan algoritam implementiran u ovom radu detektuje registarske tablice pouzdano u opštem slučaju. Takođe je utvrđeno da OPK algoritam dobro komplementira Kenijevom algoritmu detekcije ivica, i da je njihovim presekom moguće pronaći registarsku tablicu koju potom može pročitati čovek ili program namenjen tome. Ipak, uočeno je i da se algoritam slabo snalazi u specifičnim slučajevima, kada senka prekriva deo tablice, ili su tablice neuobičajenog odnosa stranica. Ovim primerima treba dodati i slike na kojima se nalazi više registarskih tablica, pošto algoritam vraća samo jednog kandidata. Predmet budućeg rada za ovaj problem bi mogla biti analiza bodovnog opsega koji se dobija prilikom bodovanja kandidata i određivanja granica u okviru kojih bi tablica mogla da se nalazi.

Tokom testiranja algoritma, primećeno je da se tehnika računanja saturacije koja se primenjuje u slučaju kada ne postoji zajednički kandidat OPK algoritma i Kenijevog algoritma

detekcije ivica, jako loše pokazala, pa se preporučuje da se umesto ovog metoda testira pouzdaniji metod, kako bi se povećala vrednost Žakarovog indeksa. Takođe, analizom neuspešnih detekcija na test bazi, uočeno je da u najvećem broju takvih slučajeva podbaci OPK algoritam, što se odražava na celokupan algoritam. Stoga se može zaključiti da bi se konačan rezultat detekcije popravio, ukoliko bi se koristio neki efikasniji i precizniji algoritam za optičko prepoznavanje karaktera.

Opšte gledano, algoritam se pokazao bolje nego što je očekivano, s obzirom na kompleksnost i zahtevnost korišćenih metoda u okviru njega. Visoka vrednost TPR-a ovog algoritma ga kategoriše kao jednog od prihvatljivih za korišćenje kao bazu u složenijim algoritmima za čitanje registarskih oznaka sa slika, što je i osnovna svrha algoritama ovog tipa.

## Literatura

- Al-Ghaili A. M., Mashohor S., Ramli A. R., Ismail A. 2013. Vertical-edge-based car-license-plate detection method. *IEEE Transactions on Vehicular Technology*, **62** (1): 26.
- Fahmy M. M. M. 1994. Computer vision application to automatic number plate recognition. *IFAC Proceedings Volumes*, **27** (12): 169.
- Hill M. 2016. Open American Licence Plate Recognition eu metadata. Michigan USA, <https://github.com/openalpr/benchmarks/tree/master/endtoend/eu>
- CVisionDemy 2019. License plate detection with OpenCV and Python 2019. CVisionDemy website: <https://cvisiondemy.com/license-plate-detection-with-opencv-and-python>
- Sharifi M., Fathy M., Mahmoudi M. T. 2002. A classified and comparative study of edge detection algorithms. U *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'02)*. IEEE, str. 117–120.
- White J. M., Rohrer G. D. 1983. Image Thresholding for Optical Character Recognition and Other

Applications Requiring Character Image Extraction. *IBM Journal of Research and Development*, **27** (4): 400.

Zhai X., Benssali F., Ramalingam S. 2010. License plate localisation based on morphological operations. U *11th International Conference on Control Automation Robotics & Vision (ICARCV 2010)*, *Proceedings*. IEEE, str. 1128-1132.

opencv.org 2015. Open Source Computer Vision Library

---

*Dimitrije Andžić and Filip Stevanović*

## Vehicle License Plate Detection Using the OCR and Edge Detection Algorithms

This paper evaluates license plate detection algorithms by comparing edge detection and text detection algorithms as well as how they work together. Canny Edge Detection methods with different types of preprocessing for edge detection were used while the already trained deep neural network in the role of Optical Character Recognition (OCR) was used for text detection. The two methods have been combined to produce a cropped portion of the image that contains exclusively the licence plate without any text on the side for safe and accurate reading of the license plate. An European car database with visible licence plates was used. The performance of the algorithm is determined by the Jaccard index (Intersection Over Union – IOU) between the labels in the database and the results returned by the algorithm. The final algorithm had a 62% IOU success rate on the whole database while the final true positive rate on the database was 86.5%. The analysis of the results concluded that all results with an IOU score greater than 70% can be classified as detected licence plates, in which case the detection accuracy of this algorithm is 84%. 