

Razpoznavanje avtomobilskih registrskih tablic z uporabo nevronske mreže

Matej Kseneman

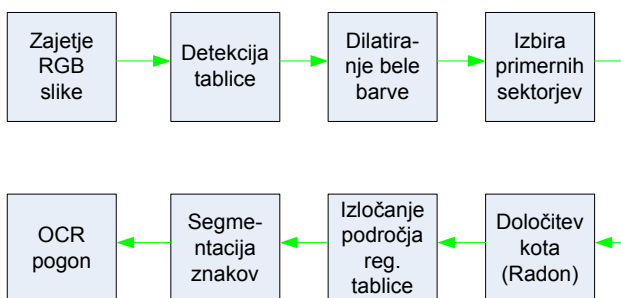
doc. dr. Peter Planinšič, mag. Tomaž Romih, doc. dr. Dušan Gleich (mentorji)
Univerza v Mariboru, Laboratorij za obdelavo signalov in daljinska vodenja, FERI
Smetanova 17, 2000 Maribor
matej.kseneman@gmail.com

Recognition of car plates using neural networks

This paper presents an algorithm for car plate recognition. The first part of the paper presents detection of the car plate using segmentation and Gaussian norm. The angle of the car plate rotation is detected and rotated. Isolated car plate is quantized and rescaled using a histogram method. In the second part of the paper the plate letters are detected using neural networks. The presented method is very efficient and provides reliable recognition.

1 Uvod

V vsakdanjem življenju se srečujemo s spremljanjem in nadzorom dogodkov. Še posebej zelo hitro napredujejo multimedijske tehnologije in z njo povezane aplikacije. V tem članku predstavljamo algoritem za razpoznavo avtomobilskih tablic [8], ki se v vsakdanjem življenju uporablja pri plačilu cestnin, v avtomobilskih garažah, nadzoru prometa, itd. Ker je aplikacija razpoznave tablic zelo razširjena smo se odločili, da podrobneje predstavimo izvedbo takšnega algoritma, kot ga npr. prikazuje slika 1.



Slika 1. Blok diagram

Prvi korak pri zaznavi registrske tablice je zajem slike, na kateri se le-ta nahaja. Rezultat

detekcije tablice je navadno beli pravokotnik z raznimi anomalijami, saj so na fotografiji prisotni še drugi slikovni elementi bele barve. Ta problem smo razrešili z uporabo 2D konvolucije in s pomočjo 2. Gaussove norme [2]. Da lahko algoritem zazna in »poveča« področje, kjer se nahaja registrska tablica, najprej izvede dilatacijo belega segmenta, saj če želimo imeti zapolnjen pravokotnik z belo barvo, moramo pokriti črne predele, ki se pojavijo na mestih črk in številok ter grbov na registrski tablici. Ko določimo ožje območje iskanja, izvedemo Radonovo transformacijo [3], s pomočjo katere rotiramo sliko tako, da je tablica natančno v horizontalnem položaju. Vzrok nagnjenosti registrske tablice so predvsem razne neravnine, ki se pojavijo na cestišču. Tako določimo manjši predel okolice registrske tablice, iz katere algoritem izvleče registrsko tablico, ki pa je na sliki že v »pravi« velikosti in jo tudi obkroži. Nato se črke na tablici prepoznajo z algoritmom za razpoznavanje črk, ki je zasnovan s pomočjo nevronske mreže.

2 Zaznavanje registrske tablice

2.1 Segmentacija slike

Segmentacija ali razčlenitev slike razdeli sliko v regije s podobnimi atributi. Največkrat se kot atribut uporabi amplituda svetlosti barve za monokromatsko sliko in barvne komponente za barvno sliko. Seveda pa so kot atributi za razčlenitev uporabni tudi robovi objektov in teksture. Moramo se zavedati, da razčlenitev ne vsebuje tudi klasifikacije posameznih segmentov, kakor tudi njihove relacije med ostalimi segmenti. Ti segmenti so le delci celotne slike. Torej si lahko razčlenitev

predstavljamo kot sestavljanke, ki jo razstavimo na posamezne koščke, ki so v našem primeru delci celotne slike [3], [7].

Ker pri sami segmentaciji ne izločimo osvetljenosti, ki je zelo odvisna od barve tablice, se poslužimo trika, pri katerem storimo podobno kot pri prejšnji metodi, t.j. delimo z vsoto trojice RGB. Algoritem deluje tako, da si izbere slikovni element, ki še ni označen (označimo ga takrat, ko ga uporabimo, ali ko se nahaja znotraj vrednosti praga nekega drugega slikovnega elementa) ter v celotni sliki išče za slikovne elemente, ki imajo razdaljo znotraj določene vnaprej podane vrednosti praga. Za izračun razdalje uporabimo enačbo (1).

$$d_{i,j} = \sqrt{(R_p - R_{i,j})^2 + (G_p - G_{i,j})^2 + (B_p - B_{i,j})^2} \quad (1)$$

Tukaj oznaka d v (1) stoji za razdaljo, indeksa i in j pa simbolično nakazujeta, da gre tukaj za matriko, in sicer i predstavlja vrstico, j pa stolpec te matrike. Indeks p nakazuje, da gre za parameter trenutno iskanega slikovnega elementa. Pri iskanju elementov s pomočjo razdalje se algoritem iskanja elementov odvija po naslednji sekvenci:

Ko zaženemo naš algoritem, so vsi slikovni elementi neoznačeni, zato izberemo kar prvi slikovni element, ki je vsebovan v sliki. Tega vstavimo v enačbo (1) in izračunamo vse razdalje glede na podani slikovni element. Da se izločijo elementi, ki spadajo k trenutnim segmentom slike, se postavi nek prag, ki v našem primeru znaša manj od vrednosti 0.1 (vrednost je manjša od ena, saj je bila slika najprej normirana na vrednosti med 0 in 1). Vse slikovne elemente, ki ustrezajo temu pogoju, označi v neki novi matriki segmentov s številom 1. Ob naslednji iteraciji, se v matriki segmentov poišče nov še neoznačen slikovni element, nakar se cikel ponovi, le da so ti slikovni elementi nato označeni s številom 2 in tako naprej. Algoritem se zaključi, ko so markirani čisto vsi slikovni elementi. Torej lahko zaključimo, da se ob izvršitvi tega algoritma pojavi kar nekaj segmentov slike. Za referenčno sliko prikazano na sliki 2 dobimo 57 segmentov, ki so prikazani na sliki 3.

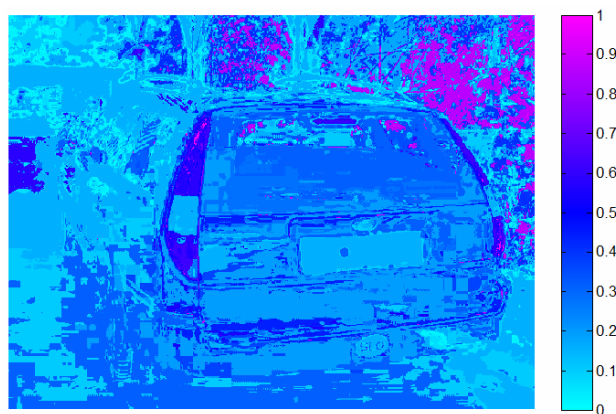
Registrsko tablico iz teh segmentov poiščemo s pomočjo 2D konvolucije oziroma korelacije, ki jo opisuje enačba (2) [5]:

$$c(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} a(k_1, k_2) b(n_1 - k_1, n_2 - k_2) \quad (2)$$

Pri tem sta a in b funkciji diskretnega argumenta n_1 in n_2 . Seveda se v našem primeru ustrezno spremenijo meje na vsotah.



Slika 2: Referenčna fotografija



Slika 3: Prikaz segmentov preko hladnih barv

Algoritem konvolucije izpeljemo z masko, ki jo predstavljamo preko celotne matrike. Maska je zgrajena tako, da v prvotni matriki velikosti 60×190 (kar je nekoliko večje kot področje registrske tablice) vrednostim -1 spremenimo predznak, in sicer v območju od 10. do 50. vrstice in od 25. do 165. stolpca (približna velikost registrske tablice). Do iskanega segmenta pridemo ob predpostavki, da se samo na enem segmentu nahaja področje, ki pripada registrski tablici in na tem segmentu dobimo maksimalno vrednost korelacijske funkcije.

Algoritem ustvari pravokotnik dimenzije 140×40. Rezultat algoritma prikazuje slika 4.



Slika 4: Iskanje registrske tablice s pomočjo 2D konvolucije

2. Gaussova norma za celotno sliko se izračuna :

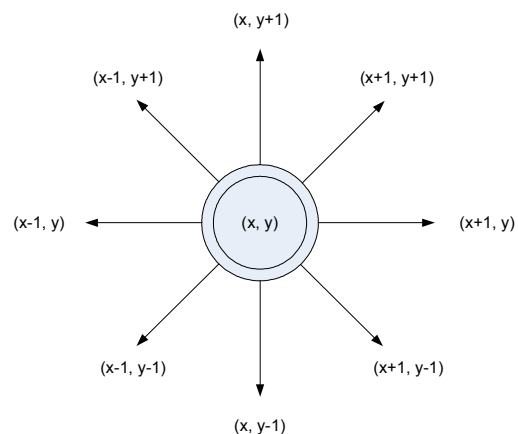
$$d_G = \sqrt{R^2 + G^2 + B^2} \quad (3)$$

Algoritem deluje na principu iskanja. Najprej išče po širini, zato potrebuje dva različna seznama, in sicer označenih in ne označenih slikovnih elementov.

Algoritem za ustvarjanje seznama neoznačenih slikovnih elementov uporablja iskanje v vseh osmih možnih smereh, torej navzgor, navzdol, levo, desno in vse 4 vmesne oz. poševne smeri (slika 5). V naslednji iteraciji se algoritem prestavi na prvi element, ki ga najde v seznamu ne označenih in ga hkrati vnese v seznam označenih. Sedaj iz tega novega mesta zopet išče v vseh osmih smereh, vendar vnese samo tiste slikovne elemente, ki niso označeni in jih še ni v seznamu neoznačenih elementov. Tukaj smo imeli nastavljen prag 10% začetnega slikovnega elementa, znotraj katerega je algoritem iskal elemente ter jih vnašal v zato primerno matriko in s tem ustvarjal masko, s pomočjo katere kasneje najdemo registrsko tablico. Algoritem se zaključi, ko označimo vse neoznačene elemente in pri tem ne najdemo novih še neoznačenih elementov.

Pri tem postopku je torej najpomembnejše najti le eno točko (slikovni element), ki se nahaja znotraj tablice, vse ostalo pa se vrši preko algoritma za iskanje pripadajočih si

elementov. Do tega slikovnega elementa lahko pridemo s pomočjo prejšnjega postopka, saj nam prejšnji postopek razpoznave registrske tablice vrne le eno točko. Paziti moramo le, da je ta točka bele barve, saj je lahko centralna barva tudi kakšne druge barve (črne). To enostavno rešimo tako, da izračunamo ali je bele barve. Če ni, se enostavno za en element premaknemo v eno izbrano stran, kjer bomo naleteli na belo barvo tablice. Rezultat zaznave prikazuje slika 6.



Slika 5: Klik, uporabljena pri 2. Gaussovi normi.



Slika 6: Rezultat uporabe 2. Gaussove norme

3 Izločitev manjšega predela z registrsko tablico

Nad sliko, ki je rezultat enega izmed prejšnjih postopkov iskanja registrske tablice, izvedemo najprej dilatacijo (razširjanje slike) tako, da se najprej pretvori v binarno sliko. Nato se izvede dilatacija s pomočjo »karo« elementa širine 5. To je priročno, saj algoritem izpusti črke in grbe, ki so na registrski tablici. S pomočjo tega postopka prekrijemo črke in grb

ter tako pridobimo zapolnjen beli pravokotnik. Rezultat je viden na sliki 7.



Slika 7: Dilatiran segment bele barve

Kot komentar navedimo, da se dobljen predel registrske tablice kar približa »ideal« dejanskega belega pravokotnika. Dilitirano sliko izboljšamo s pomočjo razčlenitve binarne slike.

4 Iskani kot zasuka tablice

Ta del algoritma poišče kot zasuka pod katerim se nahaja registrska tablica na izseku originalne slike. Tukaj predpostavimo, da so v tej sliki paralelne linije vsaj okoli tablice. Operacijo izvede na sliki, ki jo predhodno pretvori iz RGB slike v sliko s sivimi odtenki barve. Nad njimi najprej detektira robove predmetov, nakar algoritem izračuna Radonovo transformacijo [6] med koti od 0° do 179° . Vsem odšteje konstanto 25, saj s tem pridobi le najdaljše linije. Vse najdene linije nato zloži v matrični zapis, ki ga kasneje uporabi za predstavitev najdenih linij na sliki. Izmed navedenih v matriki poišče največjo in iz nje določi kot tako, da ga odšteje od vrednosti 90° . Slika 8 prikazuje vrsto najdenih linij.

V prejšnji točki algoritem izvede det kota zasuka registrske tablice, tukaj pa to vrednost uporabi, da sliko zavrti za izbrano vrednost kota. Sedaj novo, rotirano belo sliko ponovno dilatiramo podobno kot smo to že storili, torej s pomočjo objektov v obliki kare, da s tem zapolnimo večino črnih predelov znotraj registrske tablice. Algoritem ponovi iskanje pravokotnika v sliki, saj le tako lahko dobi

natančno področje registrske tablice. Povečana slika najdenega področja je vidna na sliki 9.

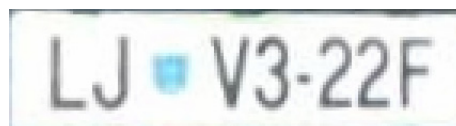


Slika 8: Prikazane linije, s pomočjo katerih algoritem določi kot zasuka



Slika 9: Izločena tablica

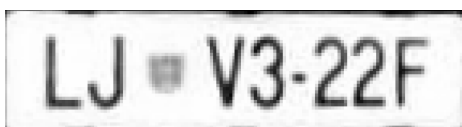
V naslednjem koraku določimo mejne točke s pomočjo kontur in na ta način določimo vsoto vrstic in stolpcev v pomanjšani sliki, saj s tem lahko natančneje določimo mejne točke iskane registrske tablice uporabljene za razpoznavo. Tako dobimo za vsako smer, torej za vrstice in stolpce, po en vektor, Za ti dve smeri algoritem posamezno izračuna skrajni dve točki, ki sta na levi in desni strani vektorja. Rezultat tega postopka je pravokotnik, s pomočjo katerega algoritem nato izvede izboljšani izsek registrske tablice, kot je prikazano na sliki 10.



Slika 10: Rezultat po obdelavi kontur

Filtriranje predela registrske tablice se izvede tako, da sliko najprej pretvorimo iz barvne slike v sliko, ki izraža svoje informacije v obliki sivih odtenkov. Nato sliko kvantiziramo in izenačimo z izračunom histograma slike. Izvede se označevanje s pomočjo limit, in sicer kar je pod spodnji_vhod enostavno označi z 0 in kar je več kot zgornji_vhod označi z 1. Intenzitetam, ki se nahajajo med navedenima mejama, priredi vrednosti med 0 in 1. Določimo še optimalen prag glede na sivinsko sliko Če je histogram slike pretežno bimodal oz. ima dva ločena hriba, se prag nastavi na vrednost na sredini. V

drugih težjih primerih, ko se hriba histograma prekrivata, bo prag minimiziral napako interpretacije slikovnih elementov ozadja kot objektov in nasprotno. Ta algoritem je poenostavitev mnogo bolj kompleksne statistične metode, ki ponuja dobre rezultate pri majhnem številu iteracij. Kvantizacija in izenačevanje sta prikazani na sliki 11.



Slika 11: Kvantizacija in izenačevanje

5 Razpoznavanje črk

Razčlenitev znakov binarne slike registrske tablice uporablja metodo izračuna histograma vsote stolpcev slike (*ang. peak-to-valley*). Metoda razčleni sliko, katere izhod sta meji vsakega znaka glede na dva podana statistična parametra, ki opisujeta širino znaka in minimalno regijo. Razčlenitev znakov prečeše celotno sliko registrske tablice od leve proti desni in izračuna vsoto stolpcev slike, katere nato primerja s statističnim parametrom širine. V primeru, da je vsota večja od praga, se odčitata dve meji tega znaka, nakar se iskanje drugega znaka prične pri desni meji pravkar najdenega elementa. Na koncu se preveri še velikost regije, katera izloči predvsem grbe in pomišljaje.

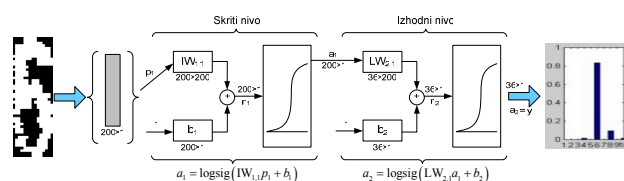
Dobljene konture, ki predstavljajo črke v registrski tablici predstavimo z znaki. Nad znaki izvedemo dilatacijo slike. Dilatacija omogoča zmanjšanje šuma slike, ki je posledica slabe kvalitete slike. Tekom te metode se poveča še razmik med znaki, kar napravi jasnejšo separacijo med znakom in ozadjem in s tem korakom olajša razpoznavanje OCR (OCR – Optical Character Recognition) pogonu.

Prilagoditev kontur registrske tablice uporablja slične metode kot predhodno opisani koraki, ki se izvajajo tako v horizontalni kot tudi v vertikalni smeri. Nato se znak skalira na izbrane dimenzije z ozirom na sklop podatkov nevronske mreže (20×10).

Jasno je, da je za uspešno razpoznavo z OCR pogonom potrebno pridobiti dobre konture znakov. Poleg tega je potrebno izvesti skaliranje, kajti velikost slike je korespondenčna z velikostjo, ki je shranjena v podatkovnem sklopu nevronske mreže. Problematika, ki tukaj nastane, je lahko posledica nepravilne izločitve slikovnih elementov tako, da se lahko stranske konture zelo popačijo.

5.1 Nevronska mreža in algoritem razpoznavanja znakov

Nevronske mreže so sestavljene iz enostavnih elementov, kateri delujejo paralelno. Ti elementi povzemajo delovanje biološkega nevrološkega sistema. Podobno kot v naravi, se tudi tukaj determinira s številom povezav med elementi. Učenje nevronske mreže za določeno aplikacijo se tako prenese na spreminjanje števila povezav (utežitev) med elementi [5].



Slika 12: Arhitektura nevronske mreže

Skalirana binarna slika velikosti 20×10 se v tem koraku pretvori v stolpec, ki vsebuje določene značilnosti slike. Vhodnim nevromom se posreduje stolpični vektor velikosti 200×1 (kolikor je tudi število nevronov na vhodnem nivoju – slika 12). Uporablja se nevronska mreža s predkrmljenjem (feed-forward), in sicer dvonivojska log-sigmoid/log-sigmoid nevronska mreža. Nevroni poskrbijo za identifikacijo znaka izmed 36 možnih. Torej na izhodu nevronske mreže dobimo izhodni vektor dimenzije 36×1 , kjer je na vsakem mestu podana verjetnost, da je primerjani znak točno določena črka abecede ali števka. Za identifikacijo iskanega znaka se enostavno vzame najverjetnejša možnost izmed vseh podanih možnosti. Nevronska mreža je sposobna razpoznavati tudi popačene vhode s šumom, kar se v naravi neprestano dogaja, in sicer se pojavi najmanj napak pri šumu s srednjo

vrednostjo 0 in standardno deviacijo 0.2 ali manj.

Nevronsko mrežo robustno na šum se zgradi tako, da se jo najprej uči na idealnih primerih, katere se nato še umetno doda šum. Učenje nevronske mreže na idealnem primeru traja vse dokler ni vrednost kvadratične vsote napake dovolj majhna. Ker se ob takšnem načinu dela lahko pojavijo problemi pri idealnih znakih na račun dobre razpoznave šumnih, je potrebno nevronsko mrežo še enkrat učiti z idealnimi primeri. Učenje nevronske mreže temelji na vračanju v primeru napake (*ang. backpropagation*), tako z adaptivnim učenjem, kot tudi z momenti.

V zadnjih nekaj letih se za razpoznavanje znakov uporablja zelo razširjena metoda nevronskih mrež, katere v nekaterih primerih pravilno rešijo problem, medtem ko ostale metode odpovejo. Problematika uporabljene metode je predvsem v tem, da OCR aplikacije temelječe na tehniki nevronskih mrež potrebujejo kar se da velik podatkovni sklop in morajo vsebovati mnogo različnih primerov posameznih znakov. Posledica nepravilne razpoznave je predvsem v majhnem številu uporabljenih znakov znotraj podatkovnega sklopa (vseh skupaj je komaj 360). Za boljše delovanje je torej nujna razširitev podatkovnega sklopa, kajti tedaj je pričakovati izboljšanje celotnega delovanja programa.

6 Poskusni rezultati

Algoritem smo testirali na 50 naključnih različno izbranih fotografij avtomobilov s slovenskimi in evropskimi registrskimi tablicami. Tabela 1 prikazuje rezultat metod za razpoznavanje tablic in razpoznavanje črk. V tabeli 1 je primerjava rezultatov teh treh metod, pri čemer moramo poudariti, da pri detekciji s pomočjo 2. Gaussove norme algoritem že ima eno točko s področja registrske tablice. Algoritem za razpoznavanje znakov na tablicah deluje zelo zanesljivo.

Tabela 1. Primerjava učinkovitosti

	Bela barva	2D konv.	2. Gaussova norma
Pravilna zaznava tablice	72%	54%	80%
Stopnja prepoznave črk	96%		

7 Zaključek

V tem članku predstavimo algoritme za določanje pozicije registrske tablice na avtomobilih in razpoznavo črk na avtomobilskih tablicah. Predstavljeni algoritmi za zaznavanje avtomobilske tablice delujejo dokaj natančno s pomočjo 2. Gaussove norme, medtem ko ima metoda z 2D konvolucijo zelo malo natančnost. Potem, ko algoritem razpozna tablico se znaki na tablicah razpoznajo z zelo veliko zanesljivostjo. Nadaljnje delo bo zajemalo implementacijo predlaganega algoritma na digitalnem signalnem procesorju in v realnih pogojih, kjer bomo poskusili zagotoviti delovanje sistema v sprotnem času.

8 Literatura

- [1] CIE (<http://www.cie.co.at/cie/>)
- [2] Useful Color Equations (<http://www.brucelindbloom.com/Math.html>)
- [3] William K. Pratt - Digital Image Processing 3Rd Edition, 2001
- [4] Color Image Segmentation Results (http://www.caip.rutgers.edu/~comanici/segm_images.html)
- [5] MATLAB help (<http://www.mathworks.com/matlabcentral/>)
- [6] Radon transform (http://en.wikipedia.org/wiki/Radon_transform)
- [7] R.C. Gonzales, R. E. Woods, S. L. Eddins, Digital Image Processing using MATLAB, Pearson Prentice Hall, 2004.
- [8] <http://www.licenseplatercognition.com/>

9 Zahvala

Članek je nastal na osnovi študentskega raziskovalnega dela v Laboratoriju za obdelavo signalov in daljinska vodenja fakultete FERi na Univerzi v Mariboru. Kot avtor bi se rad zahvalil za pomoč doc. dr. Petru Planinšiču ter mag. Tomažu Romihu in doc. dr. Dušanu Gleichu.