



VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
KOMPIUTERIJOS KATEDRA

**Transporto priemonių numerių atpažinimo algoritmų analizė
bei universalios atpažinimo sistemos teorija**

Magistrinis darbas

Atliko: II mag. kurso studentai

Andrius Girjotas

Tomas Kazlauskas

Darbo vadovas:

Algimantas Juozapavičius, dr.

Vilnius

2006

Turinys

TURINYS	2
SUTRUMPINIMŲ BEI VAIZDINĖS MEDŽIAGOS SARAŠAS	4
ANOTACIJA	6
ABSTRACT	7
IVADAS	8
BENDRAS NUMERIO ATPAŽINIMO MODELIS	9
VAIZDO NORMALIZACIJOS ALGORITMAI	10
TRIUKŠMŲ ELIMINAVIMO ALGORITMAI	10
SIMBOLIŲ IŠRYŠKINIMO FILTRAI	14
KRAŠTŲ IŠSKYRIMO METODAI	16
BINARIZACIJA	20
MORFOLOGINĖS OPERACIJOS	22
HISTOGRAMOS IŠLYGINIMAS	24
VAIZDO TRANSFORMACIJOS	25
NUMERIO LOKALIZACIJOS ALGORITMAI	26
HOUGH TRANSFORMACIJA	28
PROJEKcinė ANALIZĖ	30
MORFOLOGINĖ LOKALIZACIJA	32
SIMBOLIŲ SEGMENTACIJA	35
SEGMENTACIJOS ALGORITMAS	35
SIMBOLIŲ ATPAŽINIMAS	38
OCR ALGORITMAS	38
SISTEMOS REALIZACIJA	43
SISTEMOS FIZINĖ ARCHITEKTŪRA	43
SISTEMOS LOGINĖ ARCHITEKTŪRA	47
TYRIMAS	49
NUMERIO LOKALIZACIJOS TYRIMAS	49
SIMBOLIŲ SEGMENTACIJOS TYRIMAS	52
SIMBOLIŲ ATPAŽINIMO TYRIMAS	54
IŠVADOS IR REKOMENDACIJOS	57

Transporto priemonių numerių atpažinimo algoritmų analizė
bei universalios atpažinimo sistemos teorija

<u>LITERATŪROS SĄRAŠAS.....</u>	<u>58</u>
<u>PRIEDAS A. KOMPAKTINIO DISKO STRUKTŪRA IR KOMPIUTERIO SISTEMINIAI REIKALAVIMAI</u>	<u>59</u>
<u>PRIEDAS B. NUMERIO LOKALIZACIJOS TESTŲ REZULTATAI</u>	<u>60</u>
<u>PRIEDAS C. SEGMENTACIJOS TESTO REZULTATAI.....</u>	<u>62</u>
<u>PRIEDAS D. SIMBOLIŲ ATPAŽINIMO TESTO REZULTATAI.....</u>	<u>64</u>
<u>PRIEDAS E. SIMBOLIŲ ATPAŽINIMO ALGORITMO KLAIDOS.....</u>	<u>70</u>

Sutrumpinimų bei vaizdinės medžiagos sąrašas

<i>Figūra 1 Vidurkinio filtro branduolys</i>	10
<i>Figūra 2 Originalus vaizdas</i>	11
<i>Figūra 3 Vidurkinio filtro rezultatas po 2 iteracijų</i>	11
<i>Figūra 4 Medianinio filtro branduolys</i>	11
<i>Figūra 5 Originalus vaizdas</i>	12
<i>Figūra 6 Medianinio filtro rezultatas po 2 iteracijų</i>	12
<i>Figūra 7 Kuwahara filtro branduolys</i>	13
<i>Figūra 8 Originalus vaizdas</i>	13
<i>Figūra 9 Kuwahara filtro rezultatas po 2 iteracijų</i>	13
<i>Figūra 10 Kryptiniai filtrai</i>	14
<i>Figūra 11 Roberto operatoriaus branduoliai</i>	16
<i>Figūra 12 Roberto metodas. Originalas</i>	17
<i>Figūra 13 Roberto metodas. Filtruotas</i>	17
<i>Figūra 14 Sobelio operatoriaus branduoliai</i>	18
<i>Figūra 15 Sobelio metodas. Originalas</i>	19
<i>Figūra 16 Sobelio metodas. Filtruotas</i>	19
<i>Figūra 17 Statinė binarizacija. Originalas</i>	20
<i>Figūra 18 Statinė binarizacija. Binarizuotas vaizdas</i>	20
<i>Figūra 19 Adaptyvi binarizacija. Originalas</i>	21
<i>Figūra 20 Adaptyvi binarizacija. Binarizuotas vaizdas</i>	21
<i>Figūra 21 Originalus vaizdas ir po histogramos išlyginimo</i>	24
<i>Figūra 22 Hough metodas. Originalas</i>	29
<i>Figūra 23 Hough metodas. Surastos tiesės</i>	29
<i>Figūra 24 Hough metodas. Išskirtas numeris</i>	29
<i>Figūra 25 Koordinačių taškai ir kelios galimos linijos</i>	29
<i>Figūra 26 Projekcinė lokalizacija. Pradinis vaizdas</i>	32
<i>Figūra 27 Projekcinė lokalizacija. Vertikali histograma</i>	32
<i>Figūra 28 Projekcinė lokalizacija. Išskirtas numeris</i>	32
<i>Figūra 29 Morfologinė lokalizacija. MaxMinMax</i>	34
<i>Figūra 30 Projekcinė lokalizacija. Uždarinys (a)</i>	34
<i>Figūra 31 Projekcinė lokalizacija. Atvėrinys (b)</i>	34
<i>Figūra 32 Projekcinė lokalizacija. Atvėrinys (c)</i>	34
<i>Figūra 33 Projekcinė lokalizacija. Skirtumas (d)</i>	34
<i>Figūra 34 Projekcinė lokalizacija. Uždarinys (e)</i>	34
<i>Figūra 35 Rastas vaizdo segmentas bei jo skeletonas</i>	40
<i>Figūra 36 Skeletonų palyginimas su segmentu bei šablonu</i>	41
<i>Figūra 37 config.cfg informacija apie šablonus</i>	45

<i>Figūra 38</i> Lietuviškų numerių simbolių šablonas	45
<i>Figūra 39</i> Loginė sistemos architektūra	47

Anotacija

Automatinis transporto priemonių registracijos numerio atpažinimas atlieka svarbų vaidmenį daugelyje programinių paketų, taikomų tiek valstybinėse institucijose, tiek ir privačiose kompanijose, kuriuose yra naudojamos įvairios atpažinimo technologijos. Tačiau net ir dabar neįmanoma sukurti idealiai veikiančios sistemos, kuri paliektų laisvę efektyviausių algoritmų paieškai.

Šio tiriamojo darbo tikslas yra išanalizuoti alternatyvius automobilio numerio lokalizacijos ir kitų atpažinimo etapų algoritmus, jų efektyvumą bei adaptyvumą. Analizė atliekama juos realizuojant ir atliekant tyrimus su testiniais duomenimis bei iš jų gautais rezultatais.

Iš realizuotos alternatyvių atpažinimo algoritmų sistemos gauti rezultatai parodė, kad kiekviena atpažinimo proceso grandis yra jautri įvairiems faktoriams, kurių kitimas lemia tarpinių bei galutinių rezultatų variaciją.

Abstract

Automatic license plate recognition plays an important role in numerous applications and a number of techniques have been proposed for public institutions or private companies. However, even now it is impossible to design a perfect and operational recognition system. It still leaves a space for creativity and research of the most effective algorithms.

The main objective of this dissertation is to analyze alternatives of license plate localization and other stages of recognition, their efficiency and adaptability. Selected means of this research are such as implementation of algorithms, analysis of testing data and test results.

Every stage of recognition process is extremely sensitive to different factors which determinate variation of transitional and final results. This was proven by analysis of alternative algorithms functionality.

Įvadas

Elektroninių sistemų kūrimas bei žmogiškųjų resursų reikalaujančių operacijų automatizavimas yra vienos iš opiausių problemų šiuolaikinėms kompanijoms bei institucijoms. Šis procesas reikalauja nemažai resursų ir ne visada yra galimybė įsigyti produktą, tenkinantį organizacijos poreikius. Automatinės transporto priemonių numerių atpažinimo sistemos yra viena iš tų sričių, kurios iki šios dienos reikalauja tobulinimo siekiant gauti maksimalius rezultatus bei patenkinti vis didėjančius rinkos poreikius.

Šis tiriamasis darbas ir nagrinėja opiausią atpažinimo sistemų problemą – efektyvumo stoką. Šiuo metu egzistuoja begalės išnagrinėtų, pritaikytų tarpusavyje ir realizuotų įvairių atpažinimo etapų algoritmų, tačiau jų kombinacijų aibė vis dar neleidžia maksimaliai išnaudoti numerio atpažinimo srities.

Vienas iš pagrindinių darbo tikslų yra pabandyti surasti algoritmų ar jų modifikacijų kombinacijas, leidžiančias išspręsti aukščiau aprašytą problemą. Tai buvo siekiama realizuojant algoritmus ir tiriant gautus pagrindinių atpažinimo etapų rezultatus. Paraleliai su kokybės siekiu buvo bandoma rasti pakankamai adaptyvius algoritmų grandis, minimaliai jautrias įvairiems faktoriams – pradinio vaizdo kokybei, parametrų bei slenksčių parinkimui ir tarpinių duomenų iškraipymui.

Realizavus algoritmus buvo ieškoma efektyvių kombinacijų, kad būtų pasiektas užsibrėžtas tikslas. To pasekoje buvo sukurta loginė algoritmų jungimo schema, kurios pagrindu išskirti trys numerio atpažinimo vykdymo kryptis. Kiekviena iš jų buvo ištirta ir palygintos tarpusavyje – resursų poreikis, tikslumas ir adaptyvumas.

Atlikus tyrimus su realizuota sistema pastebėta, kad norint rasti efektyvią ir adaptyvią sistemą reikia atlikti dar išsamesnius tyrimus, nes ištirtų algoritmų grandžių pateikti rezultatai ne visiškai atitiko pradinis lūkesčius: nepasiektas tikėtas efektyvumas, nors pavyko pakankamai eliminuoti parametrizaciją, t.y. priartėti prie adaptyvios sistemos idėjos. Tačiau kai kurie tarpiniai algoritmai, pvz. Hough lokalizacija, parodė ypač aukštos kokybės rezultatus.

Bendras numerio atpažinimo modelis

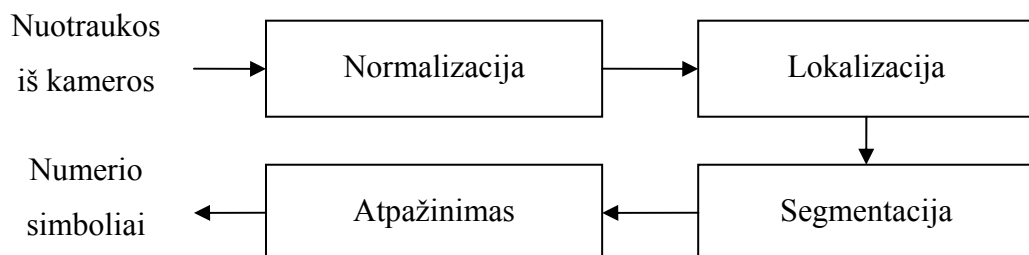
Transporto priemonių atpažinimo sistemos LPR (angl. Licence plate recognition) veikia įvairiausių algoritmų pagrindu. Tačiau, kad ir kokie bebūtų sudėtingi algoritmai, bendras sistemos veikimo modelis susideda iš dviejų dalių:

1. Numerio identifikavimas ir vietos nustatymas

- a. Vaizdo normalizacija (angl. *preprocessing*): turimas vaizdas yra apdorojamas grafiškai – sutvarkomas vaizdo šviesumas, kontrastas ir kt. paruošiamieji darbai lokalizacijai.
- b. Lokalizacija (angl. *detection*): pradiniame vaizde (nuotraukoje, video faile) yra surandamas ir išrenkamas automobilio numeris, kuris turi būti atpažintas;
- c. Transformacija (angl. *transformation*): išrinktas numeris yra paruošiamas atpažinimui – atstatomas pakreiptumas (angl. *skew*) ir atliekamos kitos transformacijos: posūkis ir skalė (angl. *rotation, scale*);
- d. Papildoma normalizacija: papildomai reikalaujami vaizdo normalizacijos algoritmai (pvz. binarizacija, jei prieš tai nebuvo atlikta)

2. Optinis simbolių atpažinimas

- a. Simbolių segmentacija (angl. *segmentation*): Išrenkami atskiri skaičiai ir raidės pagal turimą apdorotą vaizdą.
- b. Simbolių atpažinimas (angl. *recognition*): Atpažįstamas kiekvienas simbolis ir gražinamas rezultatas.



Vaizdo normalizacijos algoritmai

Triukšmų eliminavimo algoritmai

Triukšmus arba vaizdo defektus galima visiškai ar dalinai pašalinti naudojant laipsninį filtravimą, kurio vizualinis efektas yra vaizdo kontūrų ryškumo sumažinimas (angl. *blur*).

Filtravimui yra parenkamas tam tikro dydžio skanavimo plotas – branduolys, kuris leidžiamas per pradinį vaizdą kiekvienam vaizdo taškui. Branduolyje apibrėžto taškų perskaičiavimo algoritmo pagalba yra perskaičiuojamos naujos vaizdo taškų vertės.

Vidurkinis filtras (angl. *Mean smoothing*)

Šio filtro idėja yra paprasčiausiai sukeisti kiekvieno pikselio reikšmę su jo kaimynų ir jo paties reikšmių vidurkiu. Rezultate gaunamas nepriklausančių aplinkoms taškų eliminavimo efektas. Dažniausiai šiam filtrui yra naudojamas 3x3 dydžio branduolys, algoritmui nurodantis kuriuos einamo taško kaimynus reikia įtraukti į reikšmių vidurkio perskaičiavimą.

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

Figūra 1 Vidurkinio filtro branduolys

Kiekvienas branduolio taškas įgauna svertinį koeficientą, lygų

$$k = \frac{1}{N}, \text{ kur } N - \text{branduolio taškų skaičius.}$$

Žemiau pateiktame pavyzdyje yra pradinis vaizdas ir vidurkinio filtro (3x3 branduolys) rezultatas po dviejų iteracijų.



Figūra 2 Originalus vaizdas



Figūra 3 Vidurkinio filtro rezultatas po 2 iteracijų

Medianinis filtras (angl. *Median smoothing*)

Medianinis filtras, kaip ir vidurkinis, perskaičiuoja pasirinkto taško reikšmę remdamasis kaimyninių taškų reikšmėmis pagal apibrėžta branduolio algoritmą. Tačiau esminis skirtumas nuo vidurkinio yra tai, kad patenkančių taškų į medianinio filtro branduolį reikšmės yra išrūšiuojamos ir pasirinkto taško nauja vertė yra lygi išrūšiuoto reikšmių masyvo vidurinei reikšmei. Pavyzdžiui pasirinkamas branduolys, kurio dydis yra 3x3.

	123	125	126	130	140	
	122	124	126	127	135	
	118	120	150	125	134	
	119	115	119	123	133	
	111	116	110	120	130	

Figūra 4 Medianinio filtro branduolys

Išrūšiuotos į branduolį patenkančių taškų reikšmės yra 115, 119, 120, 123, 124, 125, 126, 127, 150, o masyvo vidurinė reikšmė yra lygi 124 – tai ir yra perskaičiuojamo taško reikšmė.

Tais atvejais, jei į branduolį patenkančių taškų masyvas yra lyginio dydžio (pvz. branduoliai 4x4, 6x6, 8x8 ir t.t.), tuomet naujo taško reikšmė yra dviejų vidurinių masyvo taškų reikšmių aritmetiniam vidurkiui.

Lyginiu atveju

$$N = N_{\frac{m+1}{2}}$$

Nelyginiu atveju

$$N = \frac{N_{\frac{m}{2}} + N_{\frac{m+1}{2}}}{2}$$

m – branduolio dydis



Figūra 5 Originalus vaizdas



Figūra 6 Medianinio filtro rezultatas po 2 iteracijų

Kuwahara filtras (angl. *Kuwahara smoothing*)

Kuwahara filtras yra dar vienas iš priklausančių laipsninio filtravimo grupei. Šis filtras dažnai sutinkamas kaip kraštus išsaugantis filtras, kuris išlygina vaizdo taškų kaimynines reikšmes išlaikydamas tiek vaizdo segmentų kraštų poziciją, tiek ir jų ryškumą.

Filtro branduolys yra suskaidomas į keturis atskirus regionus, kaip pateikta žemiau esančiame pavyzdyje.

a	a	ab	b	b
a	a	ab	b	b
ac	ac	$abcd$	bd	bd
c	c	cd	d	d
c	c	cd	d	d

Figūra 7 Kuwahara filtro branduolys

Kiekviename iš a , b , c ir d regionų yra suskaičiuojami taškų reikšmių vidurkiai ir intensyvumų pokyčiai (angl. *variance*). Naujai taško $abcd$ reikšmei (kuri ir yra ieškoma) yra paimamas mažiausią intensyvumų pokyčių skaičių turinčio regiono reikšmių vidurkis.



Figūra 8 Originalus vaizdas



Figūra 9 Kuwahara filtro rezultatas po 2 iteracijų

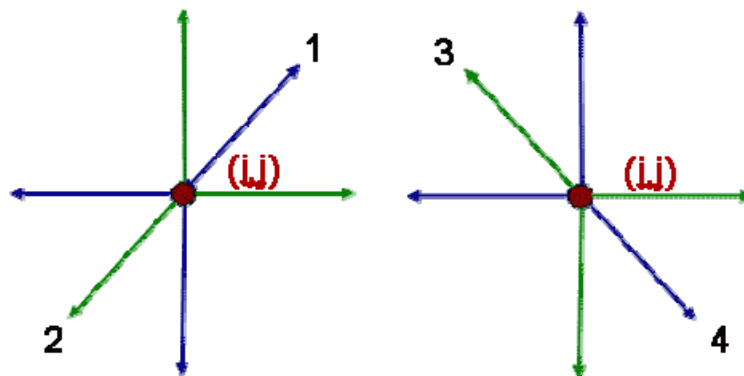
Simbolių išryškavimo filtrai

MaxMinMax filtras

MaxMinMax (MMM) – tai „žvaigždutės“ tipo netiesinis filtras, kurio vykdymo išeigoje gaunamas pakankamai švarus vaizdas su išskirtais objektais, turinčiais dideli kontrastą su fonu. Pavyzdžiui automobilio numerio lokalizacijai galima naudoti šį filtrą, nes rezultate išskiriami automobilio numerio simboliai bei kiti smulkius vaizdo elementai, kuriuos vėliau apdorojus kitais algoritmais galima surasti numerio vietą vaizde.

MMM filtro algoritmas:

- Nurodytomis keturiomis kryptimis ir pasirinkto ilgio atkarpose surandamos maksimalios vaizdo taškelių reikšmės. Slenkant kryptių šabloną per visą vaizdą, gaunamas pirmas „max“ tipo vaizdas.
- Fiksuojamas vaizdo taškas $u_{i,j}$ ir paskaičiuojami minimumai iš žaliai ir mėlynai pažymėtų pirmame punkte rastų maksimalių reikšmių keturių trejetų (Figūra 10). Slenkant per visus vaizdo taškelius suformuosime keturis „minmax“ vaizdelius v^k-u , $k=1,2,3,4$. Čia u yra pradinis vaizdas.



Figūra 10 Kryptiniai filtrai

- Surandamas „maxminmax“ vaizdas:

$$v_{i,j} = \max_k v_{i,j}^k$$

- Norint eliminuoti dėl triukšmo išryškintus vaizdo elementus, reikia surasti "maxminmax" vaizdo didžiausiąją reikšmę v_{\max} ir suformuoti galutinį "maxminmax" vaizdą, švelnaus slenksčio metodu pašalinant reikšmes, mažesnes už $v_{\max}/4$, t.y.

Transporto priemonių numerių atpažinimo algoritmų analizė
bei universalios atpažinimo sistemos teorija

$$w_{i,j} = \max \left\{ 0, \frac{v_{i,j} - v_{\max}}{4} \right\}.$$

Slenksčio reikšmė $v_{\max}/4$ yra fiksuojama apibrėžtumo dėlei. Jei vaizdo numerio simbolių kontrastas yra nedidelis (fotografuojama rūko arba nakties sąlygomis ryškių automobilio šviesų fone), tokia slenksčio reikšmė gali būti per didelė ir ją reikėtų sumažinti.

Kraštų išskyrimo metodai

Roberto kraštų išskyrimo metodas (angl. Roberts Cross Edge Detector)

Roberto operatorius atlieka paprastą, greitai paskaičiuojamą, dvimatį vaizdo gradiento paskaičiavimą. Jis išryškina aukšto erdvinio gradiento regionus, kurie dažniausiai atitinka vaizdo elementų kraštus (intensyvumų skirtumas). Kiekvienas gražinamo vaizdo pikselis atitinka paskaičiuotą absoliutų gradiento dydį tame pradinio vaizdo taške.

Šis metodas yra ypač greitas, kadangi yra dirbama tik keturių taškų ploto skaičiavimais. Dėl šios priežasties iškyla ir vienas didžiausių minusų – nemaža tikimybė gauti blogą rezultatą, jei pradiniam vaizde daug triukšmo.

Operatorius susideda iš poros 2x2 matricių - susuktų branduolių (angl. *convolution kernel*). Abu branduoliai yra identiški, vienas iš kurių yra pasuktas 90° kampu.

+1	0		0	+1
0	-1		-1	0
G_x			G_y	

Figūra 11 Roberto operatoriaus branduoliai

Branduoliai yra sudaryti taip, kad maksimaliai reaguotų į vaizdo elementų kraštus, einant per vaizdą 45° kampu – kiekvienas branduolys leidžiamas statmenomis kryptimis. Norint atlikti atskirus matavimus kiekviena kryptimi (pvz. G_x ir G_y), šie branduoliai gali būti leidžiami ir atskirai. Taigi leidžiant abu branduolius vienu metu galima paskaičiuoti gradiento dydį bei kryptį kiekviename taške. Gradiento skaičiavimas:

$$G = \sqrt{G_x^2 + G_y^2} \quad \text{arba} \quad r_{i,j} = \sqrt{(u_{-}x_{i,j})^2 + (u_{-}y_{i,j})^2}, \quad \text{kur}$$

$$u_{-}x_{i,j} = ((u_{i+1,j+1} + 2u_{i+1,j} + u_{i+1,j-1}) - (u_{i-1,j+1} + 2u_{i-1,j} + u_{i-1,j-1}))/8;$$

$$u_{-}y_{i,j} = ((u_{i+1,j+1} + 2u_{i,j+1} + u_{i-1,j+1}) - (u_{i+1,j-1} + 2u_{i,j-1} + u_{i-1,j-1}))/8$$

Transporto priemonių numerių atpažinimo algoritmų analizė
bei universalios atpažinimo sistemos teorija

Apytikslį gradientą galima skaičiuoti ir šiek tiek greitesniu būdu

$$|G| = |G_x| + |G_y|.$$

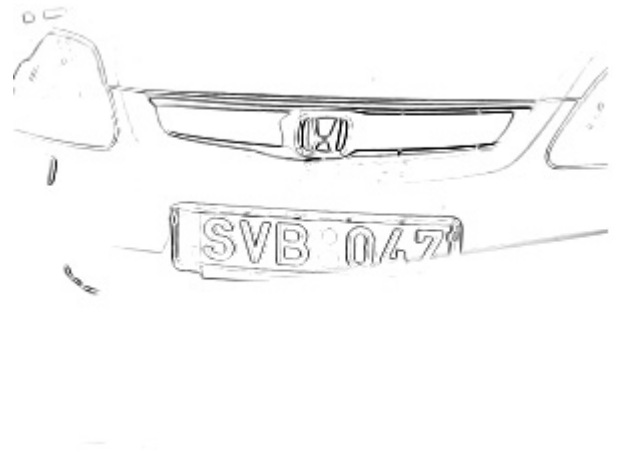
Krašto gradiento kryptis taške yra paskaičiuojama pagal

$$\theta = \arctan(G_y / G_x) - 3\pi / 4.$$

Tokiu atveju gautas 0 reiškia, kad maksimalaus kontrasto tarp juodos ir baltos srities kryptis yra iš kairės į dešinę, o kiti kampai skaičiuojami pagal laikrodžio rodyklę nuo 0.



Figūra 12 Roberto metodas. Originalas



Figūra 13 Roberto metodas. Filtruotas

Sobelio kraštų išskyrimo metodas (angl. Sobel Edge Detector)

Sobelio operatorius aproksimuoja vaizdo gradientą, skaičiuodamas suvidurkintas dalines išvestines horizontalia ir vertikalia kryptimi. Metodas yra labai panašus į Roberto operatorių, tačiau yra lėtesnis dėl didelių skaičiavimo kiekių. Tuo pačiu didesni susukti branduoliai leidžia dirbti su „triukšmingu“ pradiniu vaizdu ir gauti pakankamai gerą rezultatą.

Operatorius susideda iš poros 3x3 susuktų branduolių. Abu branduoliai yra identiški, vienas iš kurių yra pasuktas 90° kampu.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Figūra 14 Sobelio operatoriaus branduoliai

Formulių kalba gradientinis vaizdas $s_{i,j}$ apskaičiuojamas pagal tokias formules:

$$G = \sqrt{Gx^2 + Gy^2} \text{ arba } s_{i,j} = \sqrt{(u_{-}x_{i,j})^2 + (u_{-}y_{i,j})^2}, \text{ kur}$$

$$u_{-}x_{i,j} = ((u_{i+1,j+1} + 2u_{i+1,j} + u_{i+1,j-1}) - (u_{i-1,j+1} + 2u_{i-1,j} + u_{i-1,j-1}))/8;$$

$$u_{-}y_{i,j} = ((u_{i+1,j+1} + 2u_{i,j+1} + u_{i-1,j+1}) - (u_{i+1,j-1} + 2u_{i,j-1} + u_{i-1,j-1}))/8.$$

Apytikslio gradiento skaičiavimas yra kaip ir Sobelio metode:

$$|G| = |Gx| + |Gy|.$$

Krašto gradiento kryptis taške yra paskaičiuojama pagal

$$\theta = \arctan(Gy / Gx).$$

Tokiu atveju gautas 0 reiškia, kad maksimalaus kontrasto tarp juodos ir baltos srities kryptis yra iš kairės į dešinę, o kiti kampai skaičiuojami prieš laikrodžio rodyklę nuo 0.



Figūra 15 Sobelio metodas. Originalas



Figūra 16 Sobelio metodas. Filtruotas

Automobilio numerio atpažinimo atveju gautame gradientiniame vaizde išryškėja mažinos numerio simbolių kontūrai. Kartu išryškėja ir visi kiti vaizdo kontūrai, t.y. tokios vietos vaizde, kuriose vyksta spartus intensyvumo kitimas.

Binarizacija

Bendrasis binarizacijos principas yra vaizdo transformacija į dviejų spalvų intensyvumų aibę.

Binarizacija yra atliekama sudarant bendrą vaizdo histogramą ir pasirenkant arba automatiškai parenkant binarizacijos slenkstį (angl. *threshold*), o vėliau priskiriant vieną iš dviejų nustatytų spalvų intensyvumų kiekvienam taškui. Tai daroma lyginant kiekvieno vaizdo taško reikšmę su binarizacijos slenksčiu – jiems yra priskiriamos naujos reikšmės iš dviejų elementų aibės $\{a, b\}$. Dažniausiai pasitaikantis elementų parinkimas yra maksimali ir minimali vaizdo histogramos reikšmė. Pvz. jei vaizdas sudarytas iš 255 pilkumo intensyvumų, tai binarizacijos išeiiga būna juodai balta.

Statinė (globali) binarizacija

Statinė binarizacija yra pats greičiausias ir retai kada efektyvus vaizdo binarizavimo metodas. Šio metodo slenkstis yra paskaičiuojamas kaip visų vaizdo taškų reikšmių aritmetinis vidurkis ir galutiniame rezultate taškų reikšmės, esančios žemiau slenkščio, pakeičiamos į mažiausią histogramos reikšmę – *RGB #000000*, o didesnės už slenkščio reikšmę – *RGB #FFFFFF*.



Figūra 17 Statinė binarizacija. Originalas



Figūra 18 Statinė binarizacija. Binarizuotas vaizdas

Adaptyvi binarizacija

Pasirinkus statinį binarizacijos slenkstį susiduriama su pagrindine problema – informacijos praradimu – tai gali turėti ypač didelę reikšmę numerio lokalizacijai bei simbolių atpažinimui. Šios

problemos eliminavimui yra naudojamas prisitaikančio slenksčio binarizacijos metodas (angl. *adaptive thresholding*) arba kitaip tariant – adaptyvi binarizacija.

Šis metodas remiasi dinamiško arba prisitaikančio prie aplinkos slenksčio pagrindu. Vaizdas yra padalinamas į persidengiančių vaizdo dalių masyvą, kurio dydis yra lygus vaizdo taškų skaičiui, ir kiekvienai iš jų randant optimalų slenkstį. Pastarasis yra apskaičiuojamas sudarant kiekvienos dalies histogramą bei surandant jos intensyvumų vidurkį.

$$Th = \frac{1}{n} \sum_{k=1}^n P_k, \text{ kur } n - \text{vaizdo dalies dydis.}$$

Adaptyvios binarizacijos pagalba taip pat galima dalinai arba pilnai eliminuoti šešėlio problemą, kuri yra viena iš aktualiausių automobilio numerio atpažinimo sistemose. Žemiau pateiktame pavyzdyje (Figūra 19) matomas šešėlis, kertantis automobilio numerį. Prieš tai pateiktoje statinės binarizacijos išeigoje (Figūra 18) numerio dalis šešėlyje buvo visiškai užtamsinta ir šioje vietoje numerio atpažinimas nebegalimas. Tuo tarpu adaptyvios binarizacijos išeigoje (Figūra 20), šešėlis buvo eliminuotas ir po juo esanti numerio dalis išryškinta identišškai su apšviestąja dalimi.



Figūra 19 Adaptyvi binarizacija. Originalas



Figūra 20 Adaptyvi binarizacija. Binarizuotas vaizdas

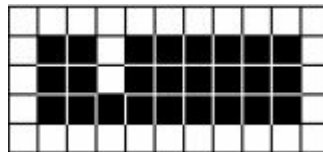
Morfologinės operacijos

Morfologinės operacijos atlieka pakankamai svarbų vaidmenį numerio atpažinimo procese. Šios operacijos yra naudojamos dviejuose etapuose, t.y. numerio lokalizacijos metu bei išskirtų numerio simbolių “šlifavime”.

Yra dvi pagrindinės morfologinės operacijos, t.y. erozija bei praplėtimas. Tačiau numerio atpažinimo procese naudojamos dvi sudėtingesnės operacijos, t.y. atvėrinys (angl. *opening*) bei uždarinys (angl. *closing*). Šios operacijos gaunamos, pritaikant dvi pagrindines operacijas (eroziją bei praplėtimą): atvėrinys gaunamas iš pradžių atlikus eroziją, o po to praplėtimą. Uždarinys gaunamas iš pradžių atliekant praplėtimą, o vėliau eroziją.

Morfologinių operacijų “veikimo principas”.

Operacijose dalyvaujantys objektai: binarinis vaizdas (sudarytas tik iš juodų bei baltų pikselių)

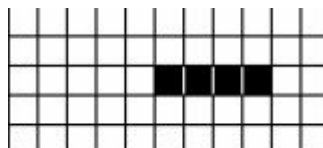


bei struktūrinis elementas su centru x:

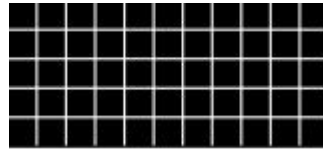


Atskirų morfologinių operacijų rezultatas:

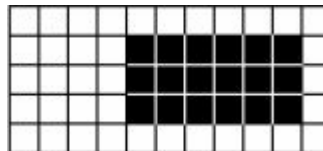
Erozijos:



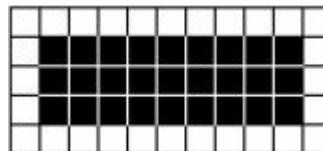
Praplėtimo:



Atvėrinio:



Uždarinio:



Morfologinės operacijos yra skirtos darbu su tam tikromis binarinio vaizdo detalėmis. Binariniame vaizde esantys objektai gali turėti: iškyšulių, plyšių, skylių, atskirus mažus objektus, kalus bei sąsiaurius. Kiekviena iš aukščiau aprašytų operacijų savaip „paveikia“ kiekvieną iš aukščiau aprašytų detalių: vienos detalės „išnyksta“, kitos – suliejamos su pagrindiniu objektu, trečios – atskiriamos nuo pagrindinio objekto ir t.t.

Kaip jau buvo minėta, aprašytos morfologinės operacijos numerio atpažinimo procese bus naudojamos numerio lokalizacijos bei išskirtų simbolių atpažinimo procesuose.

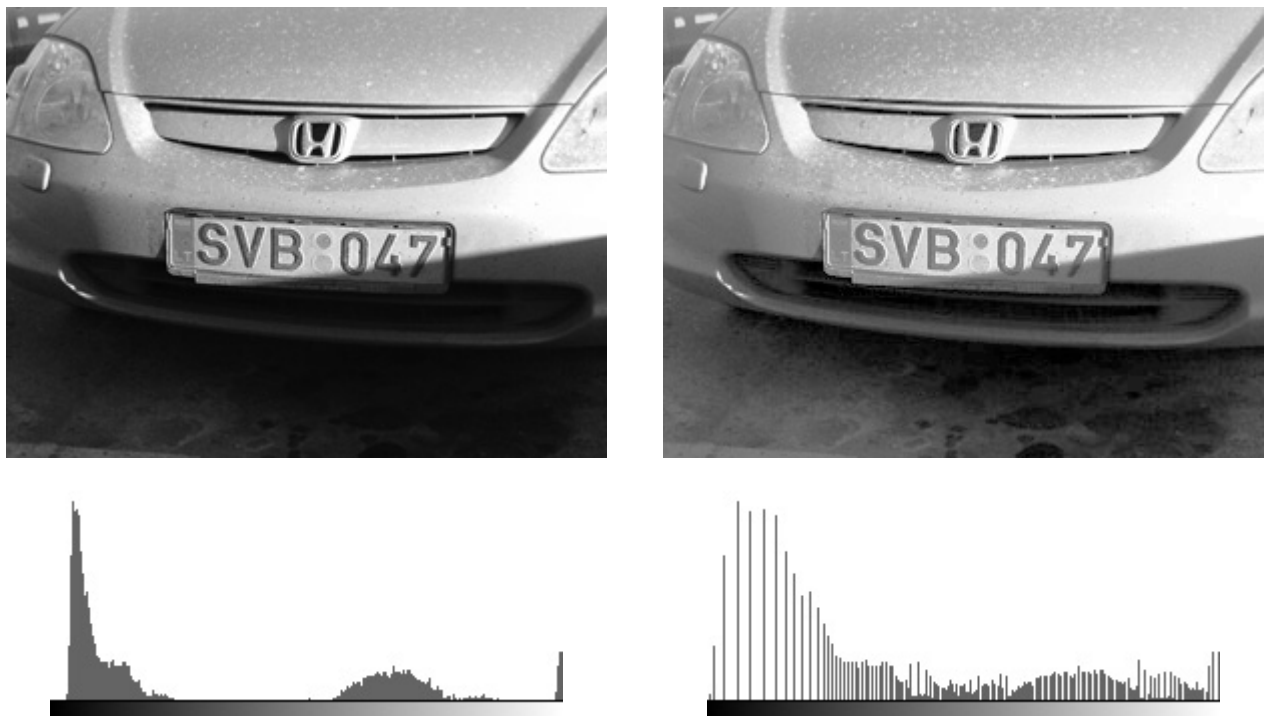
Morfologinių operacijų panaudojimas lokalizacijos procese yra labai efektyvus. Panaudojant uždarinio morfologinę operaciją (su pakankamai didelio ilgio (kvadrato formos) struktūriniu elementu, pvz.: 11×11 px) numerio lokalizacijos procesas pasidaro paprastesnis, nes „vaizdo zona“ kurioje yra numerio lentelė, „užtušuojama“. Taigi rezultate gaunamas baltas vaizdas su juoda „lentele“ jame.

Besiruošiant išskirtų numerio simbolių atpažinimo procesui, labai svarbu yra kiek galima labiau „nušlifuoti“ vaizde rastus simbolius. Dėl įvairiausių priežasčių (pvz.: dėl nekokybiško pradinio

vaizdo arba dėl atpažinimo procese dalyvavusių algoritmų „pašalinio poveikio“) rasti simboliai nebūna tokios kokybės kokios reikėtų, todėl reikalingas procesas, kurio metu išskirti simboliai būtų kiek galima labiau „nušlifuoti“ bei kiek įmanoma labiau panašūs į šablonuose apibrėžtus simbolius. Priešingu numerio atpažinimo tikimybė galėtų labai sumažėti. Šioje situacijoje labai naudinga morfologinė operacija – atvėrinys. Ji palygina rasto simbolio kontūrus, pašalina iškyšulius bei įvairius kitus pašalinius objektus, tokiu būdu padarydama simbolius daug „švaresniais“ bei labiau paruoštais finaliniam numerio analizės atpažinimo proceso žingsniui – simbolių atpažinimui.

Histogramos išlyginimas

Histogramų išlyginimas (angl. *equalization*) – viena iš pirmųjų operacijų ruošiant numerį atpažinimui. Ji nepakeičiama turint pakankamai blogos kokybės vaizdą. Blogą vaizdą gali įtakoti daugybė veiksnių, pvz.: maža nuotraukos raiška, nuotrauka daryta blogomis oro sąlygomis (esant rūkui, lyjant lietui, sningant ir t.t.). Histogramų išlyginimas sustiprina kontrastą tarp objektų, todėl šviesūs objektai pašviesinami, tamsūs – patamsinami. To pasėkoje vaizdas gaunasi daug kokybiškesnis ir labiau paruoštas tolimesniems numerio atpažinimo procesams. Žemiau pateiktas pavyzdys, puikiai atspindintis histogramos išlyginimo efektą:



Figūra 21 Originalus vaizdas ir po histogramos išlyginimo

Histogramos išlyginimą galima aprašyti šia formule:

$$O_i = \left[\sum_{j=0}^i N_j \right] \times \frac{M}{T}$$

M – maksimali intensyvumo reikšmė, kurią gali įgyti pikselis (šiam darbe ji yra 255, t.y. maksimali galima reikšmė). T – pikselių skaičius pradiniam vaizde. N – pikselių skaičius vaizde su intensyvumu j . O – nauja intensyvumo reikšmė intensyvumui j .

Atlikę skaičiavimus pagal šią formulę, randame naujas intensyvumo reikšmes. Pakeitę senus intensyvumus į naujai paskaičiuotus (kiekvienam pikseliui vaizde), gausime daug kokybiškesnį vaizdą (ypač jei pradinis vaizdas ja nepasižymi).

Vaizdo transformacijos

Posūkis

Numerio posūkio operacija neišvengiamai reikalinga, atlikus numerio lokalizacijos vaizde procesą. Retas atvejis, kai numerio lokalizacijos rezultate gaunasi visiškai horizontali „lentelė“. O pasivirę tam tikru kampu, numerio simboliai yra truputį „kitokie“ nei apibrėžti simbolių šablonuose. Taigi, akivaizdu, jog dėl šios priežasties tikėtis sėkmingo numerio atpažinimo būtų sunku. Objekto (numerio) posūkio vaizde operacija smarkiai padidina tikimybę, jog atpažinimo procesas bus sėkmingas. Ypač jei numeris vaizde pasviręs pakankamai dideliu kampu. Objekto posūkis atliekamas šios transformacijos pagalba:

$$x_2 = \cos \theta (x_1 - x_0) - \sin \theta (y_1 - y_0) + x_0$$

$$y_2 = \sin \theta (x_1 - x_0) + \cos \theta (y_1 - y_0) + y_0$$

Čia x_0, y_0 – posūkio centro koordinatės; θ – posūkio kampas; x_1, y_1 – transformuojamo taško koordinatės; x_2, y_2 – taško koordinatės po transformacijos.

Pritaikius aukščiau aprašytą transformaciją kiekvienam vaizdo pikseliui, rezultate gausime pradinį vaizdą, pasuktą kampu θ .

Numerio lokalizacijos algoritmai

Numerio lokalizacijos algoritmų įvairovė verčia juos skirstyti į funkcionalumo klases. Šio metu yra dažniausiai naudojami vaizdo apdorojimo bei neuroninių tinklų tipo algoritmai.

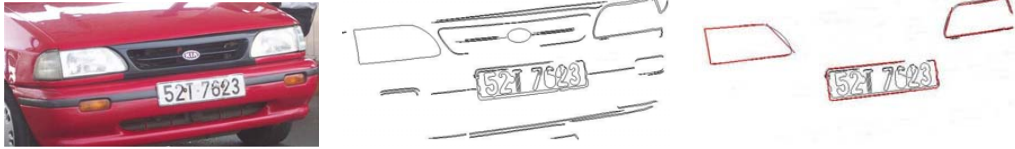
- Vaizdo apdorojimo (angl. Image processing)
Griežtai pririštas prie vaizdų apdorojimo tipas, kurio algoritmams yra suteikiamos parametru aibės, pagal kurias atliekami numerio lokalizacijos procesai.
- Savaimė besimokantys – neuroninių tinklų.
Algoritmams yra pateikiami keli testiniai vaizdai ir su minimaliu vartotojo įsikišimu yra apmokomi elgtis numatytose situacijose.

Taip pat svarbus lokalizacijos metodikos klasifikavimo aspektas yra, kad dalis algoritmų yra pritaikyti tik tam tikriems numeriams atpažinti, t.y. jie pagrįsti specifinių simbolių, tam tikrų valstybių numerių tipų ar pan. atpažinimui. Tuo tarpu kiti algoritmai nereikalauja numatytų sąlygų:

- numerio dydis, proporcijos;
- spalvos;
- valstybės identifikatoriai;
- simbolių tipai;
- simbolių išdėstymas.

Vaizdinio apdorojimo tipo algoritmai dažniausiai nereikalauja pradinių sąlygų – tai universalūs algoritmai. Todėl šiame diplominiame darbe bus nagrinėjami tik šio tipo algoritmai.

Automatinio numerio lokalizacijos problema gali būti sprendžiama remiantis numerio lentelės kraštų atpažinimo pagrindu (angl. *Boundary based*): pradinis vaizdas yra apdorojamas tam tikrais kraštų išskyrimo algoritmais, o rezultatas binarizuojamas ir pagal gautą vaizdą yra randami kraštai naudojant Hough transformaciją ar kitus panašius algoritmus. Rezultatas yra dviejų lygiagrečių linijų porų atskirtas plotas, kuris įtraukiamas į potencialių kandidatų į numerius sąrašą.



Antrasis numerio lokalizacijos būdas yra vadinamas morfologine paieška (angl. *Morphology based*). Šiuo atveju visas dėmesys sutelktas į numerio savybes – šviesumas, simetriškumas, kampai ir t.t. Remiantis šiomis savybėmis yra analizuojamas pradinis vaizdas ir lokalizuojamas numeris.

Remiantis trečiuoju, šabloninės analizės (angl. *Texture based*) metodu, automobilio numeris yra identifikuojamas, kaip objektas su skirtingais fonais ir rėmais. Tam tikras numerio tipo šablonas yra leidžiamas per pradinį vaizdą ir atitikmenys yra išrenkami, kaip potencialūs kandidatai į numerius. Specialus klasifikatorius priima sprendimą, ar tai yra tikrasis numeris, ar ne. Taip pat šis metodas yra dažnai taikomas teksto vaizde paieškai.

Hough transformacija

Hough transformacija bendroju atveju yra metodas, leidžiantis iš pradinio vaizdo išskirti pasirinktų vaizdo elementų formų savybes. Kadangi metodas reikalauja tam tikrų pradinių sąlygų savybėms aprašyti, klasikinė Hough transformacija dažniausiai yra naudojama reguliarių kreivių išrinkimui – linijos, apskritimai, elipsės ir t.t.

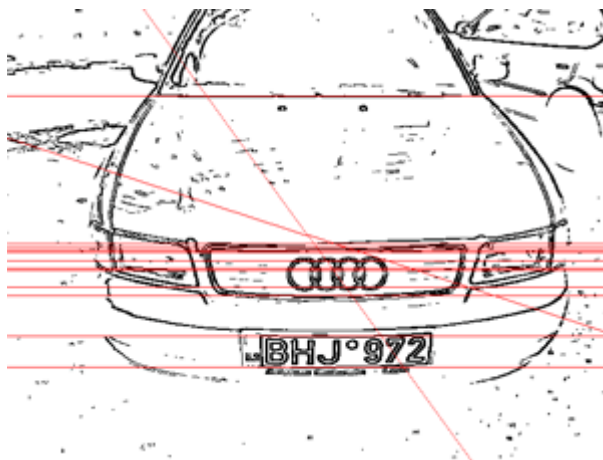
Automobilio numerio lokalizacijos atveju, iš anksto yra žinoma, kad numerio lentelė yra iškiliojo keturkampio formos (dažniausiai stačiakampio) ir naudojantis Hough transformacija yra galimybė surasti keturkampio kraštines arba konkrečiau – numerio kraštus.

Algoritmas

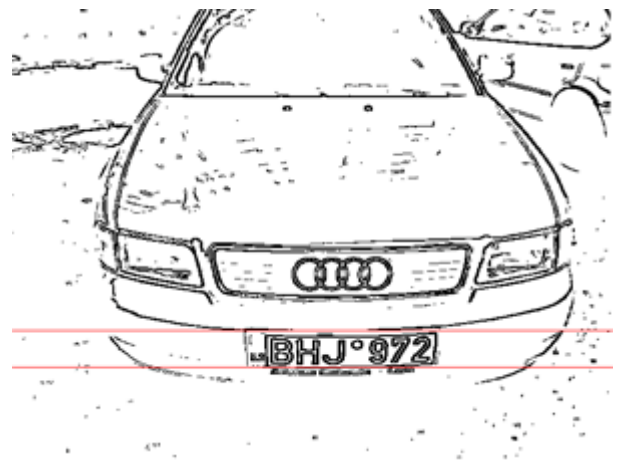
1. Pradinis vaizdas yra apdorojamas vienu iš kraštų išskyrimo algoritmu ir binarizuojamas.
2. Vaizdo taškai, patenkantys į tieses, yra randami naudojant Hough metodą.
3. Randamos poros lygiagrečių tiesių ir ieškoma tų porų susikirtimo taškų.
4. Atskirti vaizdo elementai yra traktuojami kaip potencialūs kandidatai į rastus numerius.
5. Eliminuojami kandidatai, kurie neatitinka iš anksto apibrėžtų sąlygų.



Figūra 22 Hough metodas. Originalas



Figūra 23 Hough metodas. Surastos tiesės



Figūra 24 Hough metodas. Išskirtas numeris

Transformacijos teorija

Hough transformacijos veikimo principas remiasi tuo, kad egzistuoja begalinis skaičius tiesių, kurias galima nubrėžti per erdvės taškus.



Figūra 25 Koordinatinių taškai ir kelios galimos linijos

Kiekvieną iš tų linijų galima aprašyti įvairiais būdais, tačiau patogiausia būtų jas išsireikšti per normalinę formą:

$$x \cos \theta + y \sin \theta = r,$$

kur r yra trumpiausias atstumas nuo koordinatinių pradžių taško iki tiesės, o θ – kampas tarp atstumo r ir ordinačių (x) ašies. Pagal normalinę formą, kiekvienam tiesės taškui x ir y parametrai r ir θ yra pastovūs.

Vaizdų apdorojimo atveju, kraštų segmentų taškų koordinatės (x_i, y_i) yra žinomos ir naudojamos tariamų tiesių parametrus r ir θ rasti. Apskaičiavus galimas (r, θ) reikšmes, išreiškiamas kiekvienam tiesių taškui (x_i, y_i) , Dekarto plokštumoje esančius taškus būtų galima išreikšti kreivėmis (pvz. sinusoidėmis) poliarinėje Hough plokštumoje. Ši transformacija „taškas į kreivę“ ir yra vadinama Hough transformacija.

Dekarto plokštumos taškai, kurie yra vienoje tiesėje arba labai arti jos, aiškiai yra matomi iš poliarinės Hough plokštumos, kur kelių parametru (r, θ) reikšmės artimos ir vizualiai matosi kreivių susikirtimų sancaupos.

Sekančiame etape Hough plokštuma yra peržiūrima ieškant sinusoidžių susikirtimo taškų. Iš anksto apibrėžus skaitinį parametru, ribojantį minimalų sinusoidžių susikirtimo taškų skaičių, galima surasti tam tikrą kiekį parametru (r, θ) reikšmių. Kuo didesnis parametras, tuo didesnė tikimybė surasti labiausiai tinkančias tieses.

Projekcinė analizė

Projekcinės analizės metodas yra paremtas vaizdo intensyvumų pokyčių projekcinės histogramos tyrimu.

Algoritmas

1. Vaizdo paruošimas

Šiam algoritmui iš pradžių reikia paruošti paduodamą vaizdą, apdorojant bet kuriuo kraštų radimo algoritmu. Faktiškai šio paruošimo užtektų, tačiau norint gauti tikslesnius rezultatus, po kraštų radimo naudinga apdoroti vaizdą MaxMinMax filtru, kuris užpildo mažų uždaru

kreivių ertmes. Duotu atveju yra paryškina numerio simboliai. Galutinis rezultatas yra binarizuojamas.

2. Histogramos sudarymas

Binarinis vaizdas su išryškintu numeriu yra skanuojamas vertikaliai ir horizontaliai. Skanavimo metu yra žingsniuojama po vieną vaizdo elementą, stebint jo ir sekančio eilėje kaimyninio taško vertes. Jei vertės identiškios, tikrinamas sekantis taškas. Tais atvejais, kai tikrinamojo ir kaimyninio taško vertės nesutampa, histogramos vertė yra padidinama vienetu. Tokiu būdu yra sudaromos horizontalios ir vertikalios projekcinės pokyčių histogramos.

3. Histogramų analizė

Tyrimui yra paskaičiuojamas pokyčių vidurkis ir histogramose ieškomas aukščiausių verčių sritys, esančios aukščiau vidurkio. Tokiu būdu išanalizuojamos tiek horizontali, tiek vertikali projekcija yra jų aukščiausių verčių sričių susikirtimo vietose gaunami numerio kandidatai.

Žemiau yra pateiktas pavyzdys tiriant tik horizontalios projekcijos atveju. Pateiktas pradinis vaizdas buvo apdorotas filtrais, o iš rezultato sudaryta histograma. Histogramos analizei buvo paduotas parametras, nurodantis mažiausią juodų taškų skaičių eilutėje skaičių. Galutiniame rezultate buvo rasti keturi regionai, atitinkantys reikalavimus.



Figūra 26 Projekcinė lokalizacija. Pradinis vaizdas



Figūra 27 Projekcinė lokalizacija. Vertikali histograma



Figūra 28 Projekcinė lokalizacija. Išskirtas numeris

Morfologinė lokalizacija

Šis metodas remiasi vien tik morfologinėmis operacijomis, kurios leidžia apdoroti pradinį vaizdą tiek, kad išeigoje gaunamas vaizdas su išskirtais automobilio registracijos numerio kandidatais. Tai yra Ispanijos Vigo universiteto pasiūlytas algoritmas.

Algoritmas

1. Vaizdo paruošimas

Pradiniam vaizdui yra išlyginama histograma ir leidžiamas MaxMinMax filtras, kurio pagalba išryškunami numerio simboliai. Rezultatas binarizuojamas pasirinktu algoritmu ir paduodamas morfologiniam apdorojimui.

2. Morfologinis apdorojimas

- a. Binarizuotam vaizdui vykdoma uždarinio (angl. *Closing*) operacija, naudojant branduolį, sudarytą iš horizontalios linijos, ilgesnės nei didžiausias atstumas tarp numerio simbolių;
- b. Vykdoma atvėrinio (angl. *Opening*) operacija, naudojant branduolį, sudarytą iš vertikalios linijos, trumpesnės, nei minimalus simbolio aukštis;
- c. Vykdoma identiška operacija (atvėrinio) su vertikaliu branduoliu, tik ši kartą didesniu nei maksimalus simbolio aukštis.
- d. Iš gauto rezultato [b] yra atimamas rezultatas [c], į skaičiavimą neįtraukiant fono.
- e. Vykdoma atvėrinio operacija su horizontaliu branduoliu.

3. Regionų analizė

Gautas rezultatas yra segmentuojamas po vieną elementą ir atrenkami atitinkantys numatytus numerio lenteles kriterijus – ilgio ir aukščio santykis ir pan.

Žemiau yra pateiktas vaizdas kiekviename algoritmo veikimo etape.



Figūra 29 Morfologinė lokalizacija. MaxMinMax



Figūra 30 Projekcinė lokalizacija. Uždarinys (a)



Figūra 31 Projekcinė lokalizacija. Atvėrinys (b)



Figūra 32 Projekcinė lokalizacija. Atvėrinys (c)



Figūra 33 Projekcinė lokalizacija. Skirtumas (d)



Figūra 34 Projekcinė lokalizacija. Uždarinys (e)

Simbolių segmentacija

Simbolių segmentacija yra trečiasis etapas automobilio numerio atpažinime. Kai numeris yra lokalizuojamas, t.y. yra nustatomos jo koordinatės pradiniam vaizde, ir jis yra iškerpamas, būtina atlikti mažus iškirpto vaizdo apdorojimus, nes segmentacijai reikia paduoti binarinį vaizdą – taškų masyvo reikšmės gali būti tik dviejų verčių. Kaip prie binarizacijos algoritmų aprašyta, dažniausiai pasitaikantis variantas yra juodų taškų aibė, žyminti numerio simbolius ir galimus vaizdo triukšmus baltame fone.

Segmentacijos metu yra išskiriami automobilio numerio simboliai, jei jie yra vaizde, bei vaizdo triukšmai yra atskirus vaizdo segmentus. Po vaizdo padalinimo būtina dalį segmentų eliminuoti, atmetant neatitinkančius iš anksto numatytų sąlygų. Šios sąlygos yra apibrėžiamos pagal žinomas simbolių charakteristikas.

Žemiau yra pateiktas segmentacijos algoritmas, suskaidytas į žingsnius arba etapus.

Segmentacijos algoritmas

1. Juodų objektų radimas pateiktame vaizde. Kiekvieno pikselio, patenkančio į rastą objektą, koordinatų (pradiniam vaizde) išsaugojimas.
2. Mažų objektų (susidedančių iš mažiau, nei nurodytas slenkstis, pikselių skaičiaus. Šiame darbe minėtas slenkstis yra 15 pikselių) pašalinimas.
3. Rastų segmentų „įvilkinimas“ į stačiakampius. Segmentas pilnai telpa į jį gaubiantį stačiakampį bei liečia kiekvieną iš jo kraštinių. Taip pat randamos ir maksimalios bei minimalios segmento koordinatės (viršutinis kairysis bei apatinis dešinysis minėto stačiakampio taškas).
4. Segmentų, neatitinkančių pločio ir aukščio proporcijos reikalavimo pašalinimas. Automobilio numerio simbolių aukštis yra didesnis už plotį, todėl visi segmentai, neatitinkantys šių proporcijų yra likviduojami.

5. Kelių lygių finalinė sąlyga. Po ankstesnių segmentacijos etapų likusiems segmentams taikoma sudėtingesnė sąlyga. Patikrinami visi objektai.

5.1. Kiekvienam iš objektų ieškomi du kaimynai (kairėje ir iš dešinėje). Šią sąlygą paprastai tenkina pvz.: standartinio lietuviško automobilio numerio antra raidė bei antras skaičius. Kaimynystė apibrėžiama taip: segmentas yra kaimynas, jei atstumas tarp jų (lyginant tik horizontalias koordinates) yra ne didesnis nei apibrėžta paklaida. Paklaida yra 60% einamojo segmento aukščio (tai yra visiškai pakankama paklaida, nes atstumas tarp automobilio numerio simbolių yra ženkliai mažesnis už ją. Konstantos padidinimas reikalingas tam, kad per įvairius vaizdo apdorojimo etapus, vykdomus prieš segmentaciją, galimi pikselių praradimai simbolių segmentuose, todėl atstumai tarp jų gali truputį varijuoti).

5.2. Skirtumas tarp segmentų aukščių negali būti didesnis nei nurodyta paklaida (šiai sąlygai paklaida yra 20% einamojo segmento aukščio). Aukščio kitimai tikėtini dėl galimų pikselių praradimų, aprašytų ankstesnėje sąlygoje (5.1.). Segmentų, tenkinančių aukščiau aprašytą sąlygą, turi būti ne mažiau nei 5.

5.3. Skirtumas tarp segmento vertikalųjų koordinačių minimumų (t.y. segmentų viršutinių dalių) negali būti didesnis nei apibrėžta paklaida (šiai sąlygai apibrėžta paklaida yra 60% einamojo segmento aukščio). Segmentų, tenkinančių aukščiau aprašytą sąlygą, turi būti ne mažiau nei 5.

6. Likusių segmentų bei jų sudėties išsaugojimas sekantiems darbo etapams. Iš segmentacijos pradžioje turėto vaizdo ‚iškerpamas‘ kiekvienas segmentas (praėjęs visus eliminavimo etapus). Kiekvienas iš jų išsaugomas kaip atskiras objektas (binarinis vaizdas), paruoštas lokalizacijos procesui.

7. Visi likę segmentai išrūšiuojami horizontaliai (didėjančia tvarka), t.y. pagal kiekvieno segmento minimalią horizontaliąją koordinatę. Šis veiksmas reikalingas tam, kad rasti segmentai, kuriuose yra automobilio numerio simboliai, eitu vienas paskui kitą (kaip ir originaliame vaizde), nes neišrūšiuojus, labai tikėtina, jog taip nebus. Taip yra todėl, kad

pirminiame segmentacijos etape (objektų paieškoje) pirminiai duomenys (binarinis vaizdas) apdorojami analizuojant kiekvieną vaizdo eilutę (pradedant viršutine ir baigiant apatine) bei nuosekliai einant per pikselių aibę (iš kairės į dešinę). Esant tokiam procesui bei pradiniame vaizde turint žemyn nusvirusią numerio kairiąją pusę, numerio simboliai į segmentų seką būtų patalpinti atbuline tvarka: paskutinis simbolis patektų į segmentų seką pirmas, nes is būtų aukščiausiai (jo minimali vertikali koordinatė būtų aukščiausiai kitų numerio simbolių atitinkamų koordinatė atžvilgiu) ir t.t.

Simbolių atpažinimas

Kadangi numeris sudarytas iš atskirų raidžių ir skaitmenų, pakanka atpažinti kiekvieną atskirai išskirtą simbolį. Daugiametė spausdintinių simbolių kompiuterinio atpažinimo patirtis rodo, kad, jei simbolis išskirtas teisingai, tai daugelis atpažinimo metodikų gana patikimai atpažįsta išskirtus spausdintinius simbolius. Automobilio numerio atpažinimo atveju galime laikyti, kad pateikiami atpažinimui išskirti simboliai yra spausdintiniai. Taip yra dėl to, kad valstybiniai automobilių numeriai yra standartizuoti, jie gaminami ir skirstomi centralizuotai, ir tai lemia šio uždavinio panašumą į vieno šrifto spausdintinių simbolių atpažinimo uždavinį (angl. *OCR*).

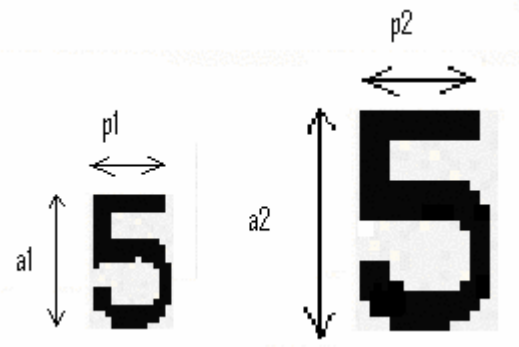
OCR algoritmas

Simbolių atpažinimo algoritmas, kaip ir daugelis prieš tai buvusių vaizdo apdorojimo algoritmų susideda iš dviejų stambių etapų – pradinio vaizdo apdorojimo arba vaizdo paruošimo sekančiam etapui bei analizinio algoritmo. Šiuo atveju vaizdo paruošimo metu vykdomas segmentų mastelių suvienodinimas bei skeletonizacija. O antrasis yra pats OCR algoritmas, analizuojantis prieš tai vykdytų operacijų rezultatus ir gražinantis analizės rezultatą.

Paruošimas atpažinimui

Mastelių suvienodinimas

Po segmentacijos rastų potencialių numerio simbolių masteliai kiekviename pavyzdyje kinta. Tai priklauso nuo nuotraukos kokybės, kaip toli nuotraukos perspektyvoje atsiduria numeris ir pan. Kadangi simbolių atpažinimo procese neišvengiamai reikalingas potencialaus automobilio numerio simbolio lyginimas su simbolių šablonais, taigi būtinas atpažįstamo objekto bei šablono simbolių mastelių suvienodinimas. Mastelį geriausia derinti prie simbolių šablonų, nes jie visi turi fiksuotą aukštį (šio darbo atveju šis aukštis yra 22 taškai). Taigi objekto mastelis bus keičiamas taip, kad jo aukštis liktų 22 taškai. Jei objekto aukštis mažesnis nei 22 taškai, tada bus atliekamas plėtimas, priešingu atveju – suspaudimas (angl. *scale*). Spaudimo arba plėtimo koeficientas paskaičiuojamas imant šablono aukščio (22 taškai) bei atpažįstamo objekto aukščio santykį. Pagal paskaičiuotą koeficientą proporcingai padidinamas objektas (tiek į aukštį, tiek ir į plotį). Suvienodinus visų išskirtų objektų mastelius, pereinama prie sekančio simbolių atpažinimo etapo.



Mastelio keitimo operacija. Rasto segmento matmenys $p_1 \times a_1$, šablono simbolio matmenys $p_2 \times a_2$. K – plėtimo/spaudimo koeficientas. $K = a_2/a_1$. Koeficientas – šablono simbolio bei rasto segmento aukščių santykis. Keičiamas rasto segmento mastelis (proporcingai aukštis ir plotis). Nauji segmento matmenys: $a_{1\text{naujasis}} = K \cdot a_1$, $p_{1\text{naujasis}} = K \cdot p_1$.

Skeletonizacija

Atlikus mastelio suvienodinimo operaciją seka skeletonizacijos operacija. Jos pagalba bus surasti kiekvieno iš segmentų bei kiekvieno simbolio (iš simbolių šablono) skeletonai. Būtent jie ir dalyvaus finaliniame simbolio atpažinimo procese, kuriame paaiškės kuriam iš simbolių (šablone) yra artimiausias tiriamas vaizdo segmentas. Skeletonizacijos procesui naudojamas *Stefanelli-Rosenfeld* skeletizacijos algoritmas.

Algoritmas

Turima tokia pikselių seka iš binarinio vaizdo:

P_7	P_8	P_9
P_6	P_1	P_2
P_5	P_4	P_3

Ši seka, tai binariniame vaizde rastas juodas taškas bei jį supantys 8 kaimyniniai pikseliai. Tarkime, kad juoda spalva identifikuojama skaičiumi 1, o balta – 0. Tada juodas pikselis originaliame vaizde bus keičiamas baltu, jei jis tenkins šias keturias sąlygas:

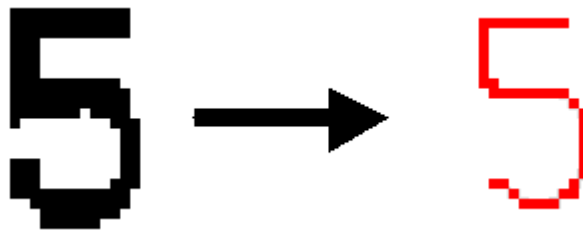
1. $Z0(P_1) = 1$
2. $2 \leq NZ(P_1) \leq 6$
3. $P_8 * P_2 * P_4 = 0$, arba $Z0(P_2) <> 1$

$$4. \quad P_6 * P_8 * P_2 = 0; \text{ arba } Z0(P_8) \triangleleft 1$$

Sąlygų paaiškinimas:

1. $Z0(P_1)$ – perėjimų skaičius nuo 0 prie 1 (sekoje $\{P_2, P_3, P_4, P_5, P_6, P_7, P_4, P_9, P_2\}$)
2. $NZ(P_1)$ – pikselio P_1 juodų kaimynų skaičius
3. Sąlyga skirta pernelyg didelės objekto erozijos išvengimui
4. Sąlyga skirta pernelyg didelės objekto erozijos išvengimui

Su šiomis keturiomis sąlygomis „perbėgama“ per visą binarinį vaizdą. Rezultate gautas vaizdas imamas kaip naujas vaizdas ir su juo atliekama ta pati operacija kaip ir pirminio vaizdo atveju (kiekvienam juodam objekto pikseliui taikomos aukščiau aprašytos keturios sąlygos). Ir taip iteruojama tol, kol „perbėgus“ per vaizdą nebeatliekamas nei vienas pakeitimas, t.y. nepašalinamas nei vienas juodas pikselis. Ir šio algoritmo rezultate gausime pradinio objekto skeletoną.



Figūra 35 Rastas vaizdo segmentas bei jo skeletonas

Simbolių atpažinimo etapas

Prieš šį etapą jau realizuota:

1. Segmentacijos metu rasti vaizdo segmentai.
2. Suvienodinti segmentų bei šablono simbolių masteliai (atliktas tempimas/spaudimas) priklausomai nuo segmentų bei šablono simbolių aukščių santykių.
3. Surasti segmentų bei šablono simbolių skeletonai.

Turint šiuos duomenis, pradedamas simbolių atpažinimo procesas. Jo eigoje:

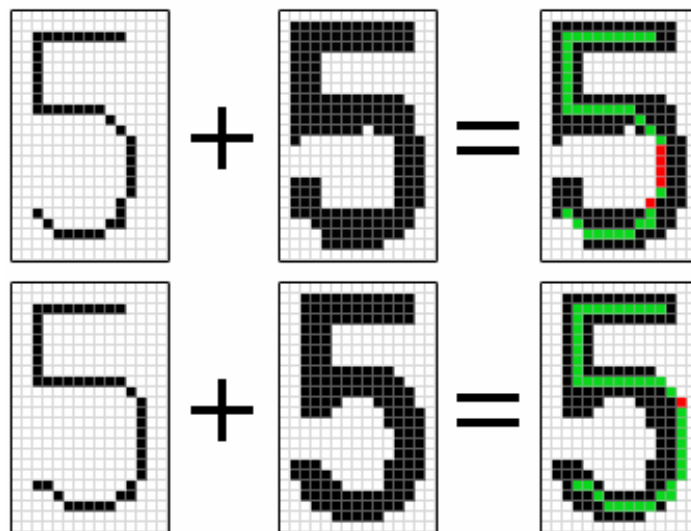
1. Kiekvieno surasto segmento skeletonas „uždedamas“ ant kiekvieno šablono simbolio. Rezultate užfiksuojamas sutapusių segmento skeletono bei šablono simbolio pikselių skaičius. Šį skaičių padalinus iš skeletoną sudarančių pikselių skaičiaus, gausime pirmą simbolio atitikimo konstantą – SAK_1 . Kuo ji artimesnė vienetui, tuo didesnė tikimybė,

kad tiriamas vaizdo segmentas atitinka tikrinamą šablono simbolį. SAK_1 , kur $0 \leq SAK_1 \leq 1$.

2. Vyksta atvirkštinis procesas nei aprašytas pirmame punkte, t.y. paskaičiavus SAK_1 , analogiškai paskaičiuojamas ir SAK_2 , tik šis atvejis yra atvirkščias pirmajam, t.y. tikrinamo šablono simbolio skeletonas „uždedamas“ ant tiriamo segmento. Sutapusių pikselių skaičius dalinamas iš skeletoną sudarančių pikselių skaičiaus ir rezultate gauname SAK_2 . Čia $0 \leq SAK_2 \leq 1$.
3. Turėdami SAK_1 bei SAK_2 skaičiuojame jų vidurkį. Vidurkis ir bus atpažinimo konstanta – AK , kur

$$AK = \frac{SAK_1 + SAK_2}{2}$$

4. Rezultate gaunama aibė konstantų, t.y. tiriamo segmento bei kiekvieno simbolio iš simbolių šablono AK . Išrūšiuojus simbolius pagal atpažinimo konstantas mažėjančia tvarka, sąrašo viršuje gausime simbolius labiausiai atitinkančius tiriamą segmentą. Einant iš sąrašo viršaus į apačią ši tikimybė mažėja.
5. Kadangi sąrašo viršuje (pirmas sąrašo elementas) yra simbolis (iš simbolių šablono) su didžiausiu AK , t.y. labiausiai atitinkantis tiriamą segmentą, tai jį, kaip labiausiai tikėtiną, ir užfiksuojame rezultate.



Figūra 36 Skeletonų palyginimas su segmentu bei šablonu

Aukščiau pateiktoje „simbolių atpažinimo“ iliustracijoje vizualizuotas atpažinimo procesas. Pirmoje iliustracijos eilutėje tiriamo segmento skeletonas uždedamas ant vieno iš tikrinamo šabloninio simbolio. Žalia spalva identifikuoja sutapusius pikselius, o raudona – nepatekusius ant simbolio pikselius. Antroje iliustracijos eilutėje vyksta atvirkštinis procesas. Tikrinamo šablono simbolio skeletonas uždedamas ant tiriamo segmento. Sutampantys pikseliai, analogiškai, išskirti žalia spalva, o nesutapimai – raudona.

Šiame pavyzdyje SAK_1 yra ~ 0.89 , o $SAK_2 \sim 0.97$, taigi $AK = (0.89 + 0.97) / 2 = 0.93$ ir jis yra tikrai aukštas, todėl labai tikėtina, kad šis simbolis atsidurs (pagal AK išrūšiuoto) sąrašo viršuje ir bus teisingai atpažintas.

Sistemos realizacija

Šiame diplominiame darbe iškeltų tikslų siekimui buvo pasirinkta JAVA programavimo kalba. Vienas iš pagrindinių šio pasirinkimų argumentų yra realizuotos sistemos nepriklausomumas nuo darbinio kompiuterio operacinės sistemos. Taip pat nemažas privalumas yra kalbos bibliotekos, leidžiančios vykdyti beveik visas grafinio vaizdų apdorojimo operacijas.

Sistemos fizinė architektūra

Realizuotą sistemą su du pagrindiniai struktūriniai elementai – paketai. Viename iš jų yra aprašytos pagalbinės sistemos klasės, vykdančios vaizdų grafinio apdorojimo operacijas, numerio šablono paruošimą atpažinimui, operacijų vykdymo laiko skaičiavimą ir kitas, reikalingas numerio paruošimui atpažinimui.

Antrąjį paketą sudaro atpažinimo algoritmų klasės, padalintos pagal atpažinimo loginius etapus, bei vartotojo sąsajos klasė.

Paketas uni_lpr

`main.class`

Sistemos startavimo klasė.

`Inteface.class`

Vartotojo sąsajos klasė, skirta minimaliai parametrizuoti sistemą. Taip pat vartotojo sąsaja yra naudojama testinių rezultatų analizei.

`logicalConnection.class`

Loginis klasių apjungimas pilnam numerio atpažinimui bei alternatyvių atpažinimo algoritmų parinkimas, aprašytas žemiau esančiame „Sistemos loginė architektūra“

`_lpreprocessing.class`

Pirmojo etapo algoritmų klasė, kurioje yra aprašyti šie išankstinio vaizdo paruošimų algoritmai:

- Triukšmų eliminavimo algoritmai
 - Vidurkinis filtras
 - Medianinis filtras

- Kuwahara filtras
- Simbolių išskyrimo metodai
 - MaxMinMax filtras
- Kraštų išskyrimo metodai
 - Sobelio kraštų išskyrimas
 - Roberto kraštų išskyrimas
 - Kryptinis kraštų išskyrimas (pagal Sobelio principą)
- Vaizdo binarizacija
 - Globali binarizacija
 - Adaptyvi binarizacija

`_2localization.class`

Antrojo etapo – numerio lentelės lokalizacijos vaizde – algoritmai:

- Hough transformacija visų vaizde esančių tiesių radimui
- Hough transformacija numerio lentelės kraštinių radimui
- Projekcinė analizė arba histograminė lokalizacija
- Morfolginė lokalizacija

`_3segmentation.class`

Simbolių segmentacijos klasė, padalinanti lokalizuotą numerį į simbolių masyvą bei, pagal iš anksto apibrėžtus numerio simbolių dydžio, išsidėstymo ir kt. standartus, eliminuojanti neatitinkančius reikalavimu. Tokiu būdu teoriškai turėtų būti gaunami skaitmenys ir raidės, sudarantys numerį.

`_4recognition.class`

Simbolių atpažinimo arba kitaip vadinama OCR klasė, kuri apdoroja gautą iš segmentacijos simbolių masyvą ir kiekvieną simbolių lygina su pateiktais šablonais. Jei simboliai atpažinti su pakankamai aukšta tikimybe, gražina tekstinę eilutę su atpažintu numeriu.

Paketas other

`ClearImage.class`

Transporto priemonių numerių atpažinimo algoritmų analizė
bei universalios atpažinimo sistemos teorija

Pagalbinė klasė, kuri apdoroja pateiktą vaizdą, išvalydama nuo niekuo nesujungtų vienetinių taškų.

`configFile.class`

Sistemos paruošimo darbui klasė, dirbanti su „`config.cfg`“ parametrų failu bei simbolių šablonais. Ši klasė yra inicijuojama sistemos startavimo metu.

- `getOption()` metodas nuskaito tam tikrą parametrų failo dalį ir grąžina parametrų masyvą;

```
# Simbolių šablonai
[templates]

template_lt.gif
ABCDEFGHIYJKLMNOPRSTUVZ0123456789

template_eu.gif
ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
```

Figūra 37 `config.cfg` informacija apie šablonus

- `getSymbols()` metodas nuskaito failus, kurių pavadinimus gauna iš `getOption()` metodo ir juos saugo atmintyje `templatesInfo` duomenų formatu.

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

Figūra 38 Lietuviškų numerių simbolių šablonas

`FileFilter.class`

`JFileChooser` naudojamas failų tipų filtras, apribojantis failų parinkimą vartotojo sąsajoje iki `*.jpg` ir `*.gif` formatų.

`Histograms.class`

Histogramų sudarymo ir atvaizdavimo klasė, naudojama vartotojo sąsajoje.

`ImageTransformations.class`

Vaizdo transformacijų klasė, leidžianti sukoti vaizdą (angl. *rotation*) bei jį didinti arba mažinti (angl. *scale*).

`MorphologicalOperations.class`

Transporto priemonių numerių atpažinimo algoritmų analizė
bei universalios atpažinimo sistemos teorija

Morfologinių operacijų (atvėrinio, uždarinio, erozijos ir praplėtimo) klasė.

`PrepareImage.class`

Konversijos klasė, kuri nuskaito paduodamą *.jpg arba *.gif failą, `PixelGrabber` pagalba jį konvertuoja į RGB masyvą ir į pilkumo intensyvumų (nuo 0 iki 255) dvimatį masyvą, kuri yra gražinamas sistemai apdorojimui ir atpažinimui.

`Template.class`

Duomenų tipas, kurio pagalba galima saugoti dvimačius šablono elementų masyvus.

`templatesInfo.class`

Duomenų tipas, leidžiantis saugoti visą reikalingą informaciją apie kiekvieną šablono simbolį:

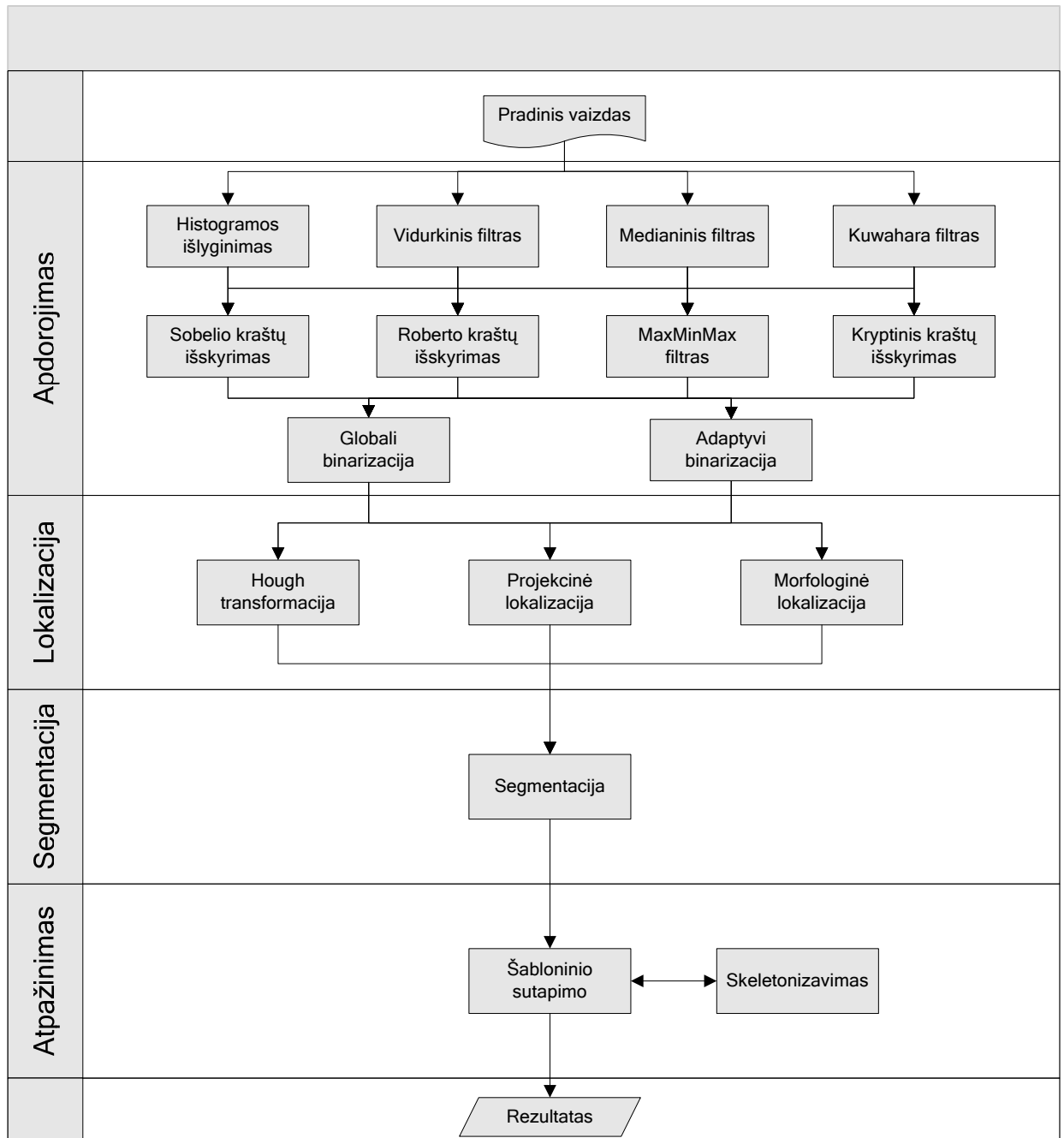
- `String` Failas – šablono failo pavadinimas;
- `String[]` Reikšmės – šablono simbolių reikšmės, kurias gražina atpažinimo metu;
- `Template[]` Simboliai – šablono simbolių taškų masyvas.

`Time.class`

Pagalbinė klasė, skirta sistemos testavimui laiko atžvilgiu.

Sistemos loginė architektūra

Sistemos loginė architektūra yra realizuota klasėje `logicalConnection.class`, kurią teoriškai galima vadinti sistemos branduoliu. Jame aprašyti visi įmanomi sąryšiai tarp alternatyvių metodų pagal bendrąjį numerio atpažinimo modelį (*vaizdo paruošimas* → *lokalizacija* → *segmentacija* → *atpažinimas*).



Figūra 39 Loginė sistemos architektūra

Iš pateiktų loginių algoritmų sąsajų galima suformuoti kelias, tarpusavyje nepriklausomas atpažinimo sistemas.

Šio diplominio darbo tyrimui buvo pasirinktos trys alternatyvos, remiantis numerio lentelės lokalizavimo algoritmų pagrindų.

Lentelė 1 Tiriama algoritmai

Pavadinimas	<i>Hough method</i>	<i>Projection method</i>	<i>Morphological method</i>
Apdorojimas	1. Histogramos išlyginimas 2. Kuwahara filtras 3. Sobelio operatorius 4. Adaptyvi binarizacija	1. Histogramos išlyginimas 2. Kryptinis kraštų išskyrimas (horizontalus) 3. Dinaminė binarizacija	1. Histogramos išlyginimas 2. MaxMinMax filtras
Lokalizacija	5. Hough lokalizacija	4. Projekcinė lokalizacija	3. Morfologinė lokalizacija
Segmentacija	6. MaxMinMax filtras 7. Dinaminė binarizacija 8. Standartinė segmentacija	5. MaxMinMax filtras 6. Dinaminė binarizacija 7. Standartinė segmentacija	4. MaxMinMax filtras 5. Dinaminė binarizacija 6. Standartinė segmentacija
Atpažinimas	9. Šabloninis sutapimas	8. Šabloninis sutapimas	7. Šabloninis sutapimas

Tyrimas

Automobilio numerio atpažinimo sistemos tyrimui naudojama 50 vidutinės kokybės ir 21 žemos kokybės skirtingų automobilių nuotraukos. Šio tyrimo tikslas yra patikrinti trijų parinktų alternatyvių atpažinimo algoritmų efektyvumą, lyginant jų galimybes teisingai ar neteisingai lokalizuoti numerį, teisingai eliminuoti netinkamus simbolius segmentacijos metu bei įvertinti simbolių atpažinimo algoritmo tikslumą.

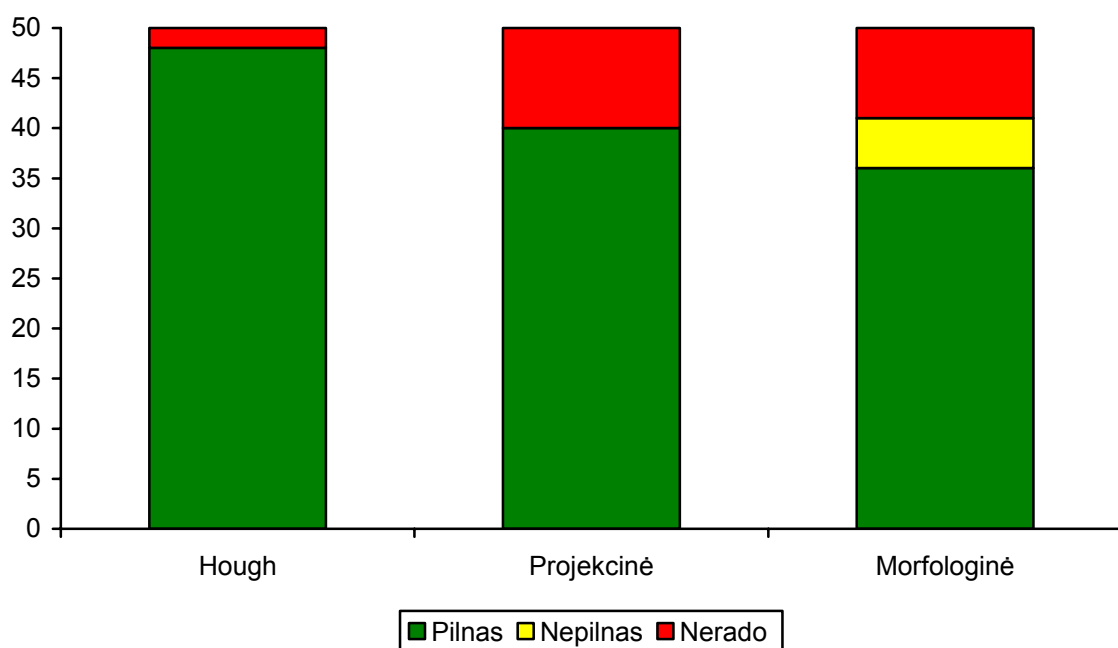
Numerio lokalizacijos tyrimas

Šio tyrimo principas yra išanalizuoti trijų alternatyvių numerio lentelės lokalizacijos algoritmų tikslumą bei veikimo laiką. Atliktų testų rezultatai yra pateikti priede B.

Lentelė 2 Vidutinės kokybės nuotraukų lokalizacijos rezultatai

Algoritmas	Hough lokalizacija	Projekcinė lokalizacija	Morfologinė lokalizacija
Rastas numeris	48	40	36
Rastas nepilnas numeris	0	0	5
Nerastas numeris	2	10	9
Vidutinis apdorojimo laikas	0,500 s	0,347 s	1,571 s

Lentelė 3 Vidutinės kokybės nuotraukų lokalizacijos rezultatų palyginimas



Iš aukščiau pateiktos lentelės akivaizdžiai matosi, kad su pateiktais testiniais duomenimis geriausiai funkcionuoja Hough lokalizacijos algoritmas. Tačiau tai yra tik vienas iš algoritmo efektyvumo rodiklių ir negalima teigti, kad jis yra geriausias.

Antrasis efektyvumo rodiklis yra algoritmo veikimo laikas. Praktiškai nuo jo priklauso ar tiriamą algoritmą galima naudoti realaus laiko atpažinimo sistemose. Iš testinių rezultatų galima teigti, kad Projekcinis ir Hough lokalizavimas tikrų šioms sistemoms, tačiau Morfologinė iš sąrašo iškrenta, nes viršiją 1 sekundės apdorojimo laiką.

Dar vienas efektyvumo rodiklis, kaip ir šio diplominio darbo tema, yra algoritmo universalumas arba kitaip tariant, parametrizavimo eliminavimas. Hough algoritmo funkcionavimui yra apibrėžti 4 pagrindiniai parametrai, nuo kurių priklauso rezultatas – tai Sobelio kraštų paryškimas esant blogai nuotraukų kokybei bei Hough transformacijos parametras, leidžiantis padidinti randamų tiesių skaičių. Kiti du yra susiję su šios transformacijos adaptavimu prie tam tikros aplinkos. Vienas iš parametru nurodo preliminarią numerio viršutinės dalies horizontalią koordinatę, o kitas – apytikslį numerio aukštį. Deja, vien dėl šių parametru Hough transformacijai galima skirti žemiausią balą už universalumą.

Projekciniam lokalizavimui yra reikalingi vėlgi du privalomi parametrai. Pirmasis, kaip ir Hough transformacijoje, yra paduodamas adaptuotam Sobelio kraštų radimo algoritmui. Antrasis parametras yra skirtas patikslinti išskirtą numerį dėl vėliau sekančio MaxMinMax algoritmo, kuris nukerpa jo žvaigždutės dydžio kraštus nuo paduodamo vaizdo. T. y. lokalizuotas numeris yra dirbtinai praplečiamas prieš MaxMinMax operaciją.

Kiek besiskiriantis savo logika Morfologinis lokalizavimas naudoja tik vieną parametru – išskirto numerio dirbtiniam praplėtimui prieš po jo sekančią MaxMinMax operaciją.

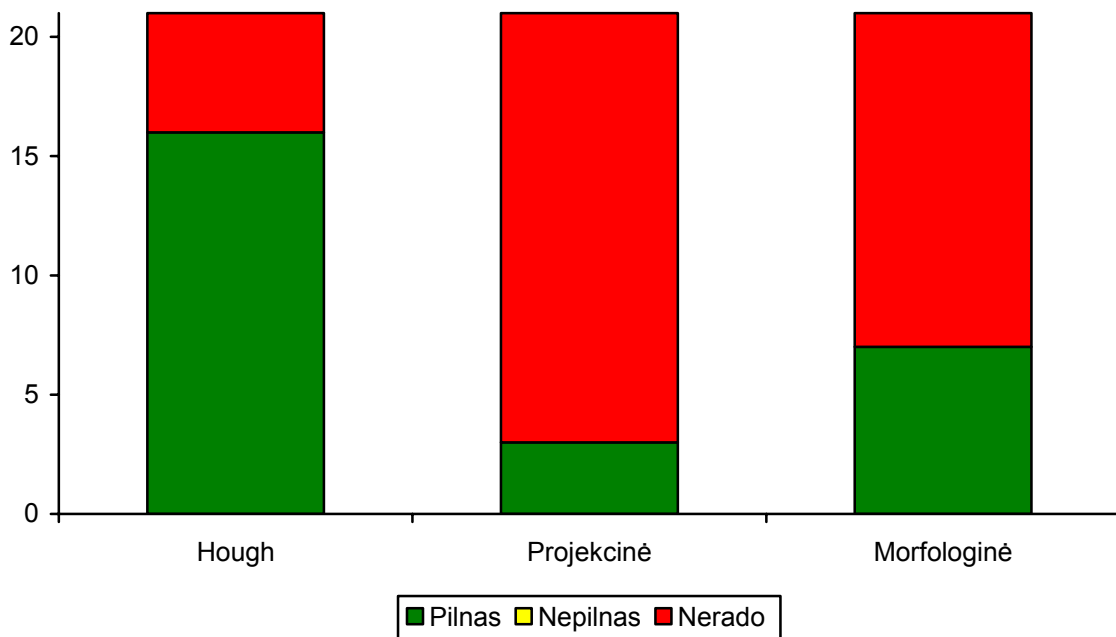
Bendrame kontekste galima matyti tokias parametru naudojimo tendencijas:

Algoritmas	Parametru skaičius
Hough	4
Projekcinis	2
Morfologinis	1

Lentelė 4 Žemos kokybės nuotraukų lokalizacijos rezultatai

Algoritmas	Hough lokalizacija	Projekcinė lokalizacija	Morfologinė lokalizacija
Rastas numeris	16	3	7
Rastas nepilnas numeris	0	0	0
Nerastas numeris	5	18	14
Vidutinis apdorojimo laikas	0,760 s	0,254 s	2,406 s

Lentelė 5 Žemos kokybės nuotraukų lokalizacijos rezultatų palyginimas



Atlikus identiškus testus su žemos kokybės nuotraukomis, rezultatai parodė, kad Projekcinė lokalizacija nėra tinkama lokalizuoti toli esančius numerius, t.y. kai automobilio numerio lentelė sudaro ypač mažą procentą viso paduodamo pradinio vaizdo. Akivaizdu, kad ir likusių dviejų algoritmų lokalizacijos tikslumas sumažėjo, tačiau panašiomis proporcijomis ir galima daryti prielaidą, kad taip ir turi būti krentant paduodamo vaizdo kokybei.

Tyrimo rezultatai

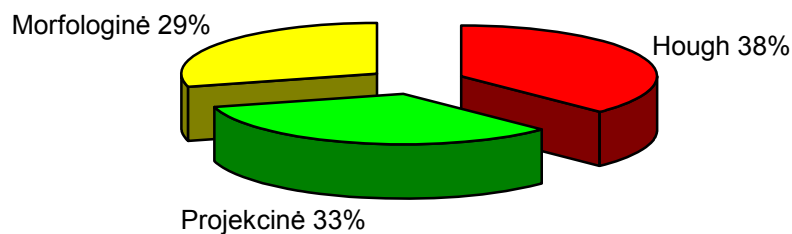
Rezultatų apžvalgai reikia pasiskaičiuoti algoritmų efektyvumo kriterijus ir išrinkti geriausią pagal bendrą įvertinimą.

Lentelė 6 Lokalizacijos algoritmų bendras įvertinimas

Kriterijus	Hough	Projekcinė	Morfologinė
Lokalizacijos tikslumas vidutinės kokybės vaizde	3	1	2
Lokalizacijos tikslumas žemos kokybės vaizde	3	2	1
Bendros vykdymo laiko sąnaudos	2	3	1
Parametrizacija arba universalumas	1	2	3

Kiekvienas iš kriterijų buvo parinktas pagal prieš tai atliktus testus ir aprašytą tiriamąją medžiagą. Algoritmai įvertinti 3 balų skalėje (1 – žemiausias balas).

Lentelė 7 Bendras lokalizacijos algoritmų įvertinimas



Pagal bendrą algoritmų įvertinimą, geriausias yra Hough algoritmas. Tačiau žvelgiant iš universalumo pusės, kas yra vienas iš esminių šio diplominio darbo klausimų, labiau vertėtų naudoti Projekcinę arba Morfologinę lokalizacijas. Toliau nagrinėjant ir lyginant pastarųjų algoritmų lokalizacijos tikslumo tiesioginę priklausomybę nuo paduodamo pradinio vaizdo kokybės galima teigti, kad mažiausiai priklausanti yra antroji – Morfologinė lokalizacija. Todėl nors ir esantis paskutinis bendrame algoritmų įvertinime bei turintis didelį atlikimo laiką, šis algoritmas turėtų būti naudojamas kuriant universalią atpažinimo sistemą.

Simbolių segmentacijos tyrimas

Tyrimui buvo išanalizuota 50 vidutinės kokybės nuotraukų, kurių numerių lentelės buvo rastos naudojant Morfologinę lokalizaciją ir apdorotos MaxMinMax filtru bei adaptyviai binarizuotos.

Galima pastebėti, kad pradinių objektų (segmentų) randama tikrai nemažai. Su jais atlikti tolimesnes operacijas, skirtas automobilio numerio simbolių atpažinimui būtų neracionalu, nes:

1. Nemažas kompiuterio operatyviosios atminties panaudojimas.
2. Nemažas procesoriaus resursų panaudojimas.

Iš pirmų dviejų teiginių išplaukia, kad būtų padidintos ir segmentacijai skirtos laiko sąnaudos, o tai taip pat labai neigiamas faktas. Šią problemą gana efektyviai išsprendžia predikcinis segmentų eliminavimas. Žvelgiant į lentelės (*Priedas C*) apačioje esančias vidurkines reikšmes, galima teigti, kad:

1. Sąlyga, pagal kurią šalinami maži objektai, sumažina segmentų skaičių $124/40 = 3,1$ karto;
2. Sąlyga, likviduojanti segmentus pagal jų aukščio ir pločio santykį (taikoma pirmos sąlygos rezultatui), sumažina segmentų skaičių 2,5 karto (pirmosios sąlygos rezultato atžvilgiu);
3. Kelių lygių finalinė sąlyga sumažina segmentų skaičių 2,94 karto (antrosios sąlygos rezultato atžvilgiu).
4. Iš viso, žvelgiant į vidurkines reikšmes, po trijų sąlyginių segmentų šalinimų, segmentų skaičius sumažėjo 22,75 karto (pradinio segmentų skaičiaus atžvilgiu)

Kyla klausimas: ar nebūtų racionaliau taikyti iš karto trečiosios, kelių lygių finalinės sąlygos? Atsakymas akivaizdus ir nenuginčijamas – ne. Vis dėlto pirma ir antra sąlygos reikalauja tik minimalių sąnaudų (operatyviosios atminties, procesoriaus bei laiko), o jų apdorojimo rezultato efektyvumas ypač didelis: segmentų skaičius sumažinamas iki 8 kartų (*Priedas C*).

Lentelė 8 Segmentacijos metu atliekamo eliminavimo etapai

Pradiniai segmentacijos duomenys:

morfologinės lokalizacijos,

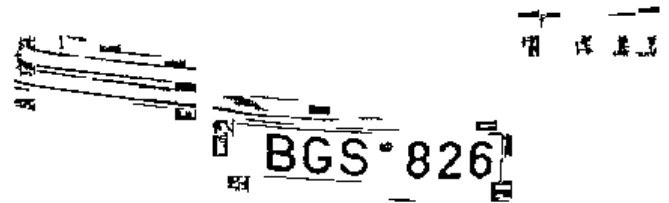
MaxMinMax filtravimo bei dinaminės
binarizacijos rezultatas.

Segmentų skaičius: 112



Pirmosios sąlygos rezultatas. Pašalinti
visi smulkūs (mažesni nei 15 pikselių)
segmentai.

Likusių segmentų skaičius: 52



Antrosios sąlygos rezultatas. Pašalinti visi segmentai (iš pirmosios sąlygos rezultato), kurių aukštis didesnis nei plotis.

Likusių segmentų skaičius: 15



Trečiosios – finalinės kelių lygių sąlygos rezultatas. Pašalinti visi (iš pirmosios sąlygos rezultato) jos netenkinę segmentai.

Likusių segmentų skaičius: 6



Simbolių atpažinimo tyrimas

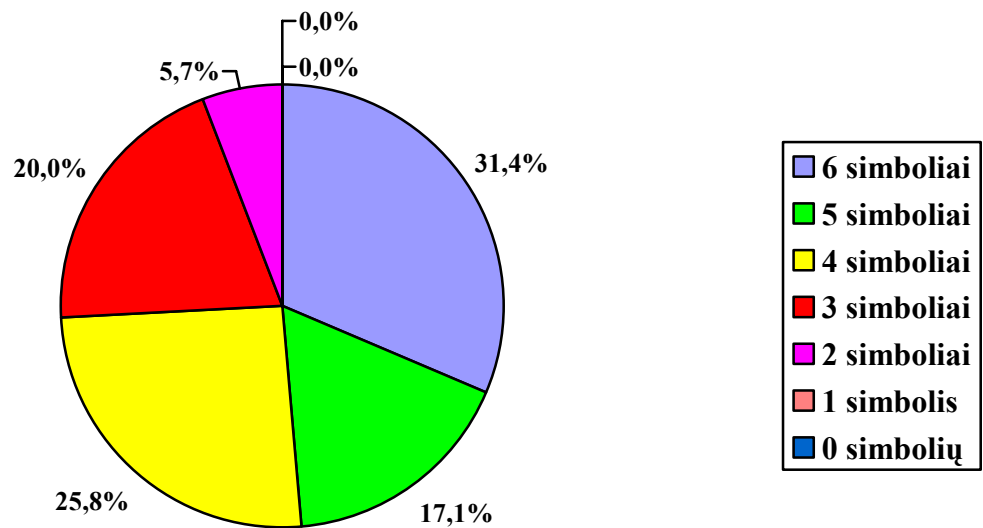
Lentelė 9 Simbolių atpažinimui paduodama informacija.

Testo rezultatų kategorija	Kiek
Nesėkmingai lokalizuoti arba nesėkmingai segmentuoti numeriai (segmentų skaičius - 0). Lokalizacija – 10, segmentacija – 3 testiniai pavyzdžiai;	13
Lokalizacijos rezultate gautas nepilnas („nukirptas“) automobilio numeris arba lokalizacija sėkminga, tačiau segmentacijos rezultate užfiksuota mažiau automobilio numerio simbolių atitinkančių segmentų, negu realiai jų yra.	2
Testiniai duomenys, kurių automobilio numerio simbolių atpažinimo procese „dalyvavo“ visi realūs numeryje esantys simboliai.	35

Tolimesnė simbolių atpažinimo (testiniams duomenims) analizė bus atliekama tik tiems numeriams, kurių visi simboliai „dalyvavo“ atpažinimo procese. Tai sudaro 70% (35/50) visų testinių duomenų.

Lentelė 10 Bendra sistemos atpažinimo kokybė

Atpažinta simbolių	Kiek numerių
6	11
5	6
4	9
3	7
2	2
1	0
0	0
<i>Viso:</i>	35



Iš aukščiau pateiktų bendros atpažinimo kokybės (*Lentelė 10*) duomenų akivaizdu, kad simbolių atpažinimo kokybė yra vidutinė, nes tik 31,4% visų pateiktų numerių buvo atpažinta kokybiškai, t.y. visi 6 simboliai. Tuo tarpu iki 3 klaidų yra beveik 62,9%. Literatūroje sutinkamas leistinas kokybės nukrypimas yra 10-15%, o tai yra 85-90% atpažinimo tikslumas.

Atpažinimo klaidų analitinėje lentelėje (Priedas E) pateikti simboliai, išrūšiuoti mažėjimo tvarka pagal jautrumą klaidoms (neatpažinimo procentinę išraišką). Iš lentelės duomenų matosi, jog 100% (testiniuose duomenyse) neatpažinti buvo tik du simboliai. Tačiau tikrai negalima teigti, jog tai tendencija. Šie simboliai buvo „sutikti“ tik po vieną kartą. „Z“ buvo supainiotas su „2“, o „T“ – su „F“, tačiau (kaip galima pastebėti simbolių atpažinimo testinių duomenų lentelėje, „T“ simbolio segmentas yra labai nekokybiškas, t.y. stipriai pažeistas ankstesniuose numerio atpažinimo sistemos etapuose, taigi nieko nuostabaus, kad jis buvo „palaikytas“ kitu simboliu).

Tikrai gana nesėkmingai buvo atpažįstamas simbolis „O“ (1 iš 7). Net keturis kartus jis buvo „palaikytas“ raide „D“ bei 2 kartus „C“. „6“ buvo painiojamas su „B“, „8“ bei „5“. „F“ numeriuose buvo užfiksuotas 3 kartus, 2 iš jų atpažintas kaip „E“. „M“ kartą atpažintas teisingai bei po kartą kaip „N“ arba „H“. Simbolis „3“ net 6 kartus iš 14 buvo atpažintas kaip „8“. „A“ 3 kartus iš 7 buvo palaikytas skaičiumi „4“.

Šie rezultatai tik patvirtina daugumos automobilių numerių atpažinimo sistemų kūrėjų bei analitikų teiginius, kad egzistuoja nemažai simbolių grupių/aibių, kurios daugiau ar mažiau, tačiau tikrai visada bus painiojamos tarpusavyje: prie tokių aibių priskiriamos tokios grupės, kaip: {A, 4}, {O, D, C}, {P, R}, {E, F}, {5, 6}, {M, H, N} ir t.t. 100% užtikrinti simbolių atpažinimo kokybės niekas negali. Yra algoritmų bei sprendimų aibės, kuriomis atpažinimo kokybę (skaičių) galima didinti/mažinti, tačiau jis niekada nesieks 100%.

Grįžtant prie atpažinimo klaidų lentelės, verta paminėti ir tuos simbolius, kurie buvo atpažįstami puikiai. Pavyzdžiui simbolis „V“ testiniuose duomenys viso buvo „sutiktas“ 19 kartų ir visus 19 kartų sėkmingai (teisingai) atpažintas. Simboliai „2“ bei „8“ testiniuose duomenyse buvo aptikti po 14 kartų ir nepadarant nei vienos klaidos atpažinti. Ne ką prasčiau sekėsi atpažinti simboli „7“ – 10 sėkmingų atpažinimų iš tik pat galimų.

Išvados ir rekomendacijos

Atlikus išsamius tyrimus su realizuotais transporto priemonių numerių atpažinimo tarpiniais algoritmais, jų rezultatais bei loginių junginių grupėmis, pastebėta, kad dalinai pavyko eliminuoti didžiausias problemas, su kuriomis susiduriama kuriant numerio atpažinimo sistemas:

- Pasiekta pakankamai aukšta lokalizacijos kokybė prie įvairių pradinio vaizdo kokybės parametrų;
- Dalinis slenksčių (angl. *threshold*) eliminavimas,
- Ypač greitas simbolių segmentavimas;
- Pasiektas didelis atpažinimo greitis, kuris leistų sistemą naudoti realaus laiko aplikacijoms.

Tačiau pasiekus tam tikrus aukštus rezultatus buvo susidurta ir su problemomis, kurios neleido pasiekti pageidaujamo galutinio rezultato. Nagrinėjant literatūrą, pastebėta, kad šios problemos yra aktualios daugeliui šios srities (numerio atpažinimo) tyrinėtojų. Viena iš didžiausių problemų – optinis simbolių atpažinimas panašių į juos aibėje. Taip pat problematiškas yra Morfologinės lokalizacijos funkcionalumas, nors teoriniam lygmenyje ji turėtų veikti nepriekaištingai. MaxMinMax filtras yra idealus variantas simbolių išryškinime, tačiau jo didžiausias minusas, kurio taip pat nepavyko eliminuoti – vaizdo kraštų nukirpimas. Tai lemia numerio esančio vaizdo šonuose eliminavimą ir tolimesnių etapų neigiamo rezultato grąžinimą. To pasekoje galutinis rezultatas taip neišvengiamai pat yra neigiamas.

Tolimesniam tyrinėjimui rekomenduojama dar giliau panagrinėti problemas, su kuriomis buvo susidurta ieškant efektyvaus ir adaptyvaus atpažinimo mechanizmo. Pirmiausia reikėtų pradėti nuo OCR (simbolių atpažinimo) technologijos – siūloma ieškoti panašių simbolių problemos eliminavimo metodų. Tokiu būdu būtų išsprendžiama viena opiausių šio tyrimo rezultatuose atsispindėjusių problemų. Taip pat būtina padidinti Morfologinės lokalizacijos efektyvumą vien dėl jos maksimalaus adaptyvumo.

Minimizavimus testų bei analizės metu išryškėjusias problemas, atsirastų galimybė praktiniam sistemos pritaikymui bei komercinio produkto sukūrimui. Tai galėtų būti šio darbo pratęsimas.

Literatūros sąrašas

- *Shyang-Lih Chang, Li-Shien Chen, Yun-Chung Chung, and Sei-Wan Chen* IEEE Automatic License Plate Recognition – 2004.
- *Vladimir Shapiro, Georgi Gluhchev* Multinational License Plate Recognition System: Segmentation and Classification – 2004.
- *Vidhya Sudhan L. Nathan, Ramkumar J., Kamakshi Priya S.* New approaches for license plate recognition system – 2004.
- *Tsang-Hong Wang, Feng-Chou Ni, Keh-Tsong Li, Yon-Ping Chen* Robust License Plate Recognition based on Dynamic Projection Warping – 2004.
- *F. Aghdasi, H. Ndungo* Automatic License Plate Recognition System – 2004.
- *Zhang Sanuyan, Zhang Mingli, Ye Xiuzi* Car plate character extraction under complicated environment – 2004.
- *Hsi-Jian Lee, Si-Yuan Chen and Shen-Zheng Wang* Extraction and Recognition of License Plates of Motorcycles and Vehicles on Highways – 2004.
- *Vivek Seetharaman, Vidhya Sudhan L. Nathan, Sathyakhala A., Sunder. P* License Plate Recognition System Using Hybrid Neural Networks – 2004.
- *Balázs Enyedi, Lajos Konyha, Csaba Szombathy, Kálmán Fazekas* Strategies For Fast License Number Localization – 2004.
- *I. Paliy, V. Turchenko, V. Koval, A. Sachenko, G. Markowsky* Approach to Recognition of License Plate Numbers Using Neural Networks – 2004.
- *Bob Fisher, Simon Perkins, Ashley Walker and Erik Wolfart* Hypermedia Image Processing Reference – 1994.
- *Yasuharu Yanamura, Masahiro Goto, Daisuke Nishiyama, Motoi Soga, Hiromasa Nakatani, Hitoshi Saji* Extraction and Tracking of the License Plate Using Hough Transform and Voted Block Matching – 2003

Priedas A. Kompaktinio disko struktūra ir kompiuterio sisteminiai reikalavimai

Transporto priemonių numerių atpažinimo sistemos realizacija kompaktiniame diske yra pateikiama kartu su diplominio darbo aprašymu.

Sistemos paleidimas

- MS Windows aplinkoje sistema paleidžiama „*run.bat*“ failo pagalba;
- UNIX/Linux aplinkoje sistema paleidžiama komandinėje eilutėje suvedus „*java -classpath "; j bcl . j ar" -Xss4096k uni _l pr . mai n*“ (be kabučių).

Kompaktinio disko struktūra

Adresas	Informacija
\\DiplominisDarbas.pdf	Šio dokumento elektroninė versija *,pdf formatu
\\DiplominisDarbas.doc	Šio dokumento elektroninė versija *,doc formatu
\\UniLPRSource	Sistemos išeities kodas
\\UniLPR	Sistemos realizacija
\\UniLPR\templates	Simbolių, naudojamų OCR, šablonai
\\UniLPR\config.cfg	Sistemos konfigūracijos failas
\\UniLPR\run.bat	Sistemos paleidimo MS Windows aplinkoje failas
\\UniLPR\uni_lpr	Sisteminis paketas „uni_lpr,,
\\UniLPR\other	Sisteminis paketas „other,,
\\UniLPR\images	Sistemos analizei naudoti testinės automobilių nuotraukos

Sisteminiai kompiuterio reikalavimai

Atpažinimo sistemos minimalūs reikalavimai:

- Windows/Linux/Unix + J2SE Runtime Environment 5.0
- CPU 500MHz, RAM 256MB

Priedas B. Numerio lokalizacijos testų rezultatai

Žemiau yra pateikti rezultatai testų, atliktų su pateiktu 50 vidutinės kokybės bei 21 žemos kokybės nuotraukų paketu, tiriant kiekvieną iš trijų lokalizacijos algoritmų. Rezultatas yra surastas arba nesurastas numeris bei kiekvieno failo apdorojimo laikas.

Lentelė 11 Vidutinės kokybės nuotraukų lokalizacija

Failas	Numeris	<i>Hough</i>		<i>Projekcinė</i>		<i>Morfologinė</i>	
100_0001.JPG	JVC 057	+	0.344 s	-	0.328 s	+	1.875 s
100_0002.JPG	MKV 289	+	0.438 s	-	0.250 s	+	1.703 s
100_0003.JPG	BFG 890	+	0.437 s	+	0.250 s	+	1.562 s
100_0004.JPG	VVJ 873	+	0.406 s	-	0.297 s	+	1.531 s
100_0005.JPG	LKC 245	+	0.485 s	+	0.250 s	+	1.579 s
100_0006.JPG	AND 824	+	0.390 s	-	0.219 s	+	1.562 s
100_0007.JPG	BGB 530	+	0.438 s	+	0.281 s	+	1.438 s
100_0008.JPG	GKL 493	+	0.547 s	-	0.375 s	+	1.593 s
100_0009.JPG	UVN 168	+	0.422 s	+	0.281 s	-	1.594 s
100_0010.JPG	VVJ 403	+	0.531 s	+	0.375 s	+	1.453 s
100_0011.JPG	ACC 646	+	0.469 s	+	0.328 s	-	1.594 s
100_0012.JPG	NVM 110	+	0.546 s	+	0.328 s	+	1.562 s
100_0013.JPG	BCN 120	+	0.422 s	+	0.454 s	-	1.516 s
100_0014.JPG	AZH 539	+	0.516 s	+	0.281 s	+	1.531 s
100_0015.JPG	RVY 891	-	0.484 s	+	0.250 s	+	1.547 s
100_0016.JPG	LVO 068	+	0.578 s	+	0.328 s	+	1.500 s
100_0017.JPG	ARF 143	+	0.610 s	+	0.250 s	+	1.594 s
100_0018.JPG	EVR 780	+	0.515 s	+	0.500 s	+	1.609 s
100_0019.JPG	CJL 726	+	0.532 s	+	0.391 s	+	1.672 s
100_0020.JPG	VKD 363	+	0.484 s	+	0.296 s	+	1.562 s
100_0021.JPG	JVM 636	+	0.500 s	+	0.313 s	+	1.547 s
100_0022.JPG	FVB 489	+	0.438 s	+	0.437 s	+	1.485 s
100_0023.JPG	HVS 928	+	0.562 s	+	0.360 s	+	1.515 s
100_0024.JPG	BBH 057	+	0.484 s	+	0.344 s	+	1.563 s
100_0025.JPG	BHL 898	+	0.579 s	+	0.421 s	-	1.593 s
100_0026.JPG	BHJ 972	+	0.421 s	+	0.250 s	+	1.579 s
100_0027.JPG	BFO 288	+	0.469 s	+	0.375 s	+	1.421 s
100_0028.JPG	BOV 342	+	0.422 s	+	0.579 s	+	1.532 s
100_0029.JPG	BCD 334	+	0.531 s	+	0.390 s	+	1.609 s
100_0030.JPG	CVO 974	+	0.485 s	+	0.297 s	+	1.484 s
100_0031.JPG	SVB 940	+	0.468 s	+	0.219 s	+	1.516 s
100_0032.JPG	AHO 931	+	0.516 s	+	0.375 s	+	1.563 s
100_0033.JPG	OKS 210	+	0.547 s	+	0.328 s	+	1.468 s
100_0034.JPG	ACS 129	+	0.531 s	+	0.344 s	+	1.563 s
100_0035.JPG	RVU 088	+	0.516 s	-	0.390 s	+	1.687 s
100_0036.JPG	BZK 111	-	0.609 s	+	0.360 s	-	1.735 s
100_0037.JPG	BGS 826	+	0.610 s	+	0.328 s	+	1.609 s
100_0038.JPG	BCE 234	+	0.484 s	+	0.422 s	+	1.641 s

Transporto priemonių numerių atpažinimo algoritmų analizė
bei universalios atpažinimo sistemos teorija

100_0039.JPG	GPG 124	+	0.500 s	-	0.312 s	+	1.562 s
100_0040.JPG	SVB 047	+	0.484 s	-	0.500 s	+	1.500 s
100_0041.JPG	AEV 989	+	0.547 s	+	0.250 s	-	1.531 s
100_0042.JPG	OVV 903	+	0.531 s	-	0.344 s	-	1.469 s
100_0043.JPG	AOM 855	+	0.454 s	+	0.406 s	-	1.516 s
100_0044.JPG	NVH 169	+	0.500 s	-	0.391 s	-	1.578 s
100_0045.JPG	TVK 835	+	0.515 s	+	0.250 s	+	1.672 s
100_0046.JPG	AOJ 689	+	0.438 s	+	0.406 s	+	1.594 s
100_0047.JPG	ACK 547	+	0.672 s	+	0.391 s	+	1.593 s
100_0048.JPG	UVC 754	+	0.546 s	+	0.265 s	+	1.625 s
100_0049.JPG	MVL 232	+	0.563 s	+	0.438 s	+	1.578 s
100_0050.JPG	UVM 911	+	0.484 s	+	0.562 s	+	1.641 s

* - morfologinės lokalizacijos atveju yra nukerpamas vienas ar keli simboliai, todėl numeris pilnai nelokalizuojamas.

Lentelė 12 Žemos kokybės nuotraukų lokalizacija

Failas	Numeris	Hough		Projekcinė		Morfologinė	
z001.jpg	KVA 353	+	0.735 s	-	0.219 s	+	2.172 s
z002.jpg	TVK 773	+	0.844 s	-	0.281 s	+	2.203 s
z003.jpg	DVV 410	-	0.765 s	-	0.344 s	-	2.344 s
z004.jpg	KKC 714	-	0.750 s	-	0.203 s	+	2.390 s
z005.jpg	LVC 584	+	0.766 s	-	0.266 s	-	2.328 s
z006.jpg	OVY 661	+	0.781 s	-	0.296 s	-	2.469 s
z007.jpg	TVD 929	+	0.766 s	-	0.188 s	-	2.453 s
z008.jpg	TVO 401	+	0.734 s	-	0.281 s	-	2.500 s
z009.jpg	PVV 629	-	0.750 s	+	0.281 s	+	2.313 s
z010.jpg	CRV 001	-	0.609 s	-	0.204 s	+	2.500 s
z011.jpg	PVV 629	+	0.641 s	+	0.265 s	-	2.265 s
z012.jpg	FPD 892	+	0.797 s	-	0.266 s	+	2.344 s
z013.jpg	TVA 690	+	0.875 s	-	0.219 s	-	2.438 s
z014.jpg	OVJ 984	+	0.687 s	-	0.265 s	-	2.500 s
z015.jpg	SVM 303	+	0.875 s	-	0.266 s	-	2.390 s
z016.jpg	TVY 124	+	0.844 s	-	0.219 s	-	2.485 s
z017.jpg	PVV 040	+	0.766 s	-	0.265 s	-	2.468 s
z018.jpg	PVZ 630	-	0.734 s	+	0.266 s	-	2.454 s
z019.jpg	PVV 040	+	0.813 s	-	0.203 s	-	2.531 s
z020.jpg	YVY 238	+	0.734 s	-	0.250 s	-	2.484 s
z021.jpg	GVF 421	+	0.703 s	-	0.281 s	+	2.485 s

Priedas C. Segmentacijos testo rezultatai

Pradiniai duomenys gauti atlikus morfologinę lokalizaciją, MaxMinMax filtravimą bei adaptyvią binarizaciją po Morfologinės lokalizacijos. Duomenys – 50 vidutinės kokybės nuotraukų.

Lentelė 13 Simbolių segmentacijos rezultatai

Failas	Pradinis	Salyga #1	Salyga #2	Salyga #3	Perteklius	Trūkumas
100_0001.JPG	35	14	10	4	0	2
100_0002.JPG	167	65	19	0	0	6
100_0003.JPG	105	40	18	7	1	0
100_0004.JPG	130	40	17	6	0	0
100_0005.JPG	110	31	14	9	3	0
100_0006.JPG	238	66	29	6	0	0
100_0007.JPG	127	44	18	9	3	0
100_0008.JPG	119	48	15	8	2	0
100_0009.JPG	-	-	-	-	-	-
100_0010.JPG	206	57	21	6	0	0
100_0011.JPG	-	-	-	-	-	-
100_0012.JPG	141	46	19	7	1	0
100_0013.JPG	-	-	-	-	-	-
100_0014.JPG	114	41	17	7	1	0
100_0015.JPG	164	37	17	7	1	0
100_0016.JPG	209	62	22	9	3	0
100_0017.JPG	144	39	16	7	1	0
100_0018.JPG	215	56	17	7	1	0
100_0019.JPG	31	18	9	6	0	0
100_0020.JPG	121	64	16	0	0	6
100_0021.JPG	38	20	11	7	1	0
100_0022.JPG	63	22	8	0	0	6
100_0023.JPG	170	52	18	8	2	0
100_0024.JPG	363	66	22	7	1	0
100_0025.JPG	-	-	-	-	-	-
100_0026.JPG	134	55	27	6	0	0
100_0027.JPG	56	29	7	6	0	0
100_0028.JPG	30	15	8	8	2	0
100_0029.JPG	102	41	14	7	1	0
100_0030.JPG	41	17	10	6	0	0
100_0031.JPG	66	35	19	6	0	0
100_0032.JPG	252	74	28	11	5	0
100_0033.JPG	77	38	20	3	0	3
100_0034.JPG	122	49	18	6	0	0
100_0035.JPG	52	22	8	6	0	0
100_0036.JPG	-	-	-	-	-	-
100_0037.JPG	112	52	15	7	1	0
100_0038.JPG	36	16	7	7	1	0
100_0039.JPG	186	42	16	7	1	0
100_0040.JPG	44	19	9	8	2	0
100_0041.JPG	-	-	-	-	-	-
100_0042.JPG	-	-	-	-	-	-
100_0043.JPG	-	-	-	-	-	-
100_0044.JPG	-	-	-	-	-	-
100_0045.JPG	129	45	11	6	0	0
100_0046.JPG	142	22	11	7	1	0

100_0047.JPG	198	56	29	7	1	0
100_0048.JPG	41	16	9	8	2	0
100_0049.JPG	89	34	16	6	0	0
100_0050.JPG	-	-	-	-	-	-
Vidurkiai:	~124	~40	~16	~5.45	~0.95	~0.60

Paaiškinimai

Failas	Testinių duomenų failų pavadinimai
Pradinis	Segmentacijos pradžioje užfiksuotas segmentų skaičius
Sąlyga #1	Segmentų skaičius po labai mažų segmentų pašalinimo
Sąlyga #2	Segmentų skaičius po segmentų, neatitinkančių pločio ir aukščio proporcijos sąlygos, pašalinimo
Sąlyga #3	Segmentų skaičius po segmentų, neatitikusių kelių lygių finalinės sąlygos (žr. 5.1.), pašalinimo
Perteklius	Segmentų (galutinėje segmentų aibėje) perteklius. Perteklius – kai segmentų daugiau nei 6 (taikoma lietuviškiems automobilių numeriams)
Trūkumas	Segmentų (galutinėje segmentų aibėje) trūkumas. Trūkumas – kai segmentų mažiau nei 6 (taikoma lietuviškiems automobilių numeriams)

Pastaba

Eilutės, kuriose visi laukeliai užpildyti brūkšniais (tokių yra 10), reiškia, kad lokalizuotame vaizde numerio nebuvo.

Priedas D. Simbolių atpažinimo testo rezultatai

Lentelė 14 Simbolių atpažinimo rezultatai

Failas	Numeris	Segmentai	Skeletas	Rezultatas	Koeficientai	+	-
100_0001.JPG	JVC 057			DS2N	D: 0.9277597402597402 S: 0.8653846153846154 2: 0.7136752136752136 N: 0.5995454545454546	0	3
100_0002.JPG	-	-	-	-	-	-	-
100_0003.JPG	BFG 890			BFG890K	B: 0.9253731343283582 F: 0.7702702702702703 G: 0.8020918785890074 8: 0.8619162640901772 9: 1.0 0: 0.8873913043478261 K: 0.4953416149068323	6	0
100_0004.JPG	VVJ 873			VVU873	V: 1.0 V: 0.8531468531468531 U: 0.6477272727272727 8: 0.9745762711864407 7: 0.7981744421906694 3: 0.8048780487804879	5	1
100_0005.JPG	LKC 245			5LLKC2452	5: 0.5357142857142857 L: 0.65 L: 0.96875 K: 0.9893617021276595 C: 0.8752252252252253 2: 0.8974358974358974 4: 0.9516129032258065 5: 0.9166666666666667 2: 0.6538461538461539	6	0
100_0006.JPG	AND 824			ANC824	A: 0.7337662337662337 N: 0.9099999999999999 C: 0.926097972972973 8: 0.9674849267872524 2: 0.9230769230769231 4: 0.6272321428571428	5	1
100_0007.JPG	BGB 530			7BGB5309A	7: 0.6180371352785146 B: 0.9477611940298507 G: 0.9891304347826086 B: 0.8529093662124162 5: 0.9 3: 0.9024390243902439 0: 1.0 9: 0.5798611111111112	6	0
100_0008.JPG	GKL 493			6GKL4B8Y	6: 0.7072368421052632 G: 0.7208777686628385 K: 0.9784921369102683 L: 0.9833333333333334 4: 0.984375 B: 0.71694160775072 8: 0.8888888888888888 Y: 0.5607142857142857	4	2
100_0009.JPG	-	-	-	-	-	-	-
100_0010.JPG	VVJ 403			VVU408	V: 0.9545454545454546 V: 1.0 U: 0.6537878787878788 4: 0.6919642857142857 0: 1.0 8: 0.8425925925925926	4	2
100_0011.JPG	-	-	-	-	-	-	-

Transporto priemonių numerių atpažinimo algoritmų analizė
bei universalios atpažinimo sistemos teorija

100_0012.JPG	NVM 110			ENVN11C	E: 0.5309311224489797 N: 0.99 V: 0.8552631578947368 N: 0.7533333333333333 I: 0.8181818181818181 I: 0.6176470588235294 C: 0.89	4	2
100_0013.JPG	-	-	-	-	-	-	-
100_0014.JPG	AZH 539			IA2M5SO	I: 0.7023809523809523 A: 0.8988413547237077 2: 0.7469911041339612 M: 0.800831443688586 5: 0.9880952380952381 S: 0.7646464646464647 O: 0.8033588435374149	2	4
100_0015.JPG	RVY 891			RVY8818	R: 0.9545454545454546 V: 0.8846153846153846 Y: 0.9821428571428572 8: 1.0 8: 0.8602375960866526 I: 1.0 8: 0.6029411764705883	5	1
100_0016.JPG	LVO 068			6LVC0B8MI	6: 0.7121212121212122 L: 0.9444444444444444 V: 0.8225108225108224 C: 0.7617354196301565 O: 0.9593877551020409 B: 0.7714873906330417 8: 0.763157894736842 M: 0.530701754385964 I: 0.6944444444444444	3	3
100_0017.JPG	ARF 143			IAHE148	I: 0.6791383219954649 A: 0.7619414483821263 H: 0.8128968253968254 E: 0.823868677905945 I: 0.9736842105263157 4: 0.8603219696969697 8: 0.8425925925925926	3	3
100_0018.JPG	ERV 780			EVR780M	E: 0.9797959183673469 V: 0.9561834561834562 R: 0.9188197767145135 7: 0.9655172413793104 8: 0.9444444444444444 0: 0.9130434782608696 M: 0.468409586056644	6	0
100_0019.JPG	CJL 726			1NL728	I: 0.6229946524064172 N: 0.5957142857142858 L: 0.7197916666666666 7: 0.9310344827586207 2: 0.7064102564102563 8: 0.9438997821350763	3	3
100_0020.JPG	-	-	-	-	-	-	-
100_0021.JPG	JVM 636			JVM5857	J: 0.6474358974358975 V: 0.9431818181818181 M: 0.856280193236715 5: 0.8893557422969187 8: 0.744973544973545 5: 0.916017316017316 7: 0.4091673675357443	3	3
100_0022.JPG	-	-	-	-	-	-	-

Transporto priemonių numerių atpažinimo algoritmų analizė
bei universalios atpažinimo sistemos teorija

100_0023.JPG	HVS 928			IHVS928P	I: 0.803921568627451 H: 0.8033333333333332 V: 0.9558823529411764 S: 0.8645833333333333 9: 0.9583333333333333 2: 0.9134615384615385 8: 0.8903356481481481 P: 0.6625258799171843	6	0
100_0024.JPG	BBH 057			BPH95ZS	B: 0.9006888633754305 P: 0.8783967391304348 H: 0.8111111111111111 9: 0.8840277777777779 5: 0.9415584415584416 Z: 0.7587587587587588 S: 0.5393939393939393	3	3
100_0025.JPG	-	-	-	-	-	-	-
100_0026.JPG	BHJ 972			QIBHU072	Q: 0.4769736842105263 I: 0.6840277777777778 B: 0.8911265589858924 H: 0.8333333333333333 U: 0.6931818181818181 O: 0.8335949764521193 7: 0.8683385579937304 2: 0.8032200357781754	4	2
100_0027.JPG	BFO 288			BED288	B: 0.9405148172182566 E: 0.8979591836734694 D: 0.8361952861952862 2: 0.8974358974358974 8: 0.9074074074074074 8: 0.9351851851851851	4	2
100_0028.JPG	BOV 342			BDV3M2N1	B: 0.8423650975889783 D: 0.9 V: 0.9857142857142858 3: 0.9003484320557491 M: 0.571200510855683 2: 0.9358974358974359 N: 0.5984848484848485 1: 0.7306501547987616	4	2
100_0029.JPG	BCD 334			MBCLB84	M: 0.488148148148148 B: 0.9112886272861047 C: 0.818448883666275 L: 0.7619047619047619 B: 0.8432835820895522 8: 0.8425925925925926 4: 0.875	3	3
100_0030.JPG	CVO 974			CVD974	C: 0.9189189189189189 V: 0.8712121212121212 D: 0.9727272727272727 9: 0.8645833333333333 7: 0.8841594827586207 4: 0.7064144736842105	5	1
100_0031.JPG	SVB 940			SVB940	S: 0.8529980657640233 V: 0.6578947368421053 B: 0.8883795309168443 9: 0.8226950354609929 4: 0.8787878787878788 0: 0.9347826086956521	6	0

Transporto priemonių numerių atpažinimo algoritmų analizė
bei universalios atpažinimo sistemos teorija

100_0032.JPG	AHO 931			FIII4NIC1II93	F: 0.6621621621621622 I: 0.6944444444444444 I: 0.8055555555555556 I: 0.6577060931899641 4: 0.6778846153846154 N: 0.7956521739130434 I: 0.6899641577060932 C: 0.86511983681795 1: 0.6092436974789917 I: 0.6979166666666666 I: 0.7647058823529411 9: 0.9895833333333333 3: 0.9506097560975609	2	4
100_0033.JPG	OKS 210			CIS	C: 0.7270752895752897 1: 0.5690537084398977 S: 0.8414031620553359	1	2
100_0034.JPG	ACS 129			4CS128	4: 0.6759868421052632 C: 0.9324324324324325 S: 0.8300395256916996 1: 0.713903743315508 2: 0.9230769230769231 8: 0.8959694989106753	4	2
100_0035.JPG	RVU 088			RVU088	R: 0.990909090909091 V: 0.7923832923832923 U: 0.8522727272727273 0: 0.9565217391304348 8: 0.932301153612629 8: 0.9444444444444444	6	0
100_0036.JPG	-	-	-	-	-	-	-
100_0037.JPG	BGS 826			BGS826F	B: 0.9109660033167496 G: 0.9252440106477373 S: 0.8894830659536542 8: 0.9344444444444444 2: 0.9487179487179487 6: 0.875 F: 0.5713682432432432	6	0
100_0038.JPG	BCE 234			BCE234A	B: 0.9029850746268657 C: 0.9459459459459459 E: 0.9466836734693878 2: 0.9743589743589743 3: 0.8780487804878049 4: 0.7703598484848485 A: 0.575098814229249	6	0
100_0039.JPG	GPG 124			4CPG124	4: 0.5905448717948718 C: 0.8524324324324324 P: 0.8695652173913043 G: 0.8854515050167224 1: 0.7058823529411764 2: 0.9230769230769231 4: 0.7748161764705883	5	1
100_0040.JPG	SVB 047			1SVB047J	1: 0.7712418300653594 S: 0.8765597147950088 V: 0.8783783783783784 B: 0.9701492537313432 0: 0.9891304347826086 4: 0.8783143939393939 7: 0.9326508620689655 J: 0.6436403508771931	6	0
100_0041.JPG	-	-	-	-	-	-	-
100_0042.JPG	-	-	-	-	-	-	-
100_0043.JPG	-	-	-	-	-	-	-
100_0044.JPG	-	-	-	-	-	-	-

100_0045.JPG	TVK 835	TVK835	TVK835	FVK88S	F: 0.6216216216216216 V: 0.956876456876457 K: 0.7905944986690328 8: 0.898635477582846 8: 0.6729900632339656 S: 0.9090909090909092	3	3
100_0046.JPG	AOJ 689	ADJ689	AOJ689	ADN6893	A: 0.8222222222222222 D: 0.8818181818181818 N: 0.6799999999999999 6: 0.9479166666666667 8: 0.9915254237288136 9: 0.8245833333333333 3: 0.5120209059233449	2	4
100_0047.JPG	ACK 547	ACK547	ACK547	4CK5M7J	4: 0.6534090909090908 C: 0.9594594594594594 K: 0.9342995169082126 5: 1.0 M: 0.537037037037037 7: 0.9310344827586208 J: 0.4758771929824561	4	2
100_0048.JPG	UVC 754	IUVC754	IUVC754	IUVC754M	I: 0.7222222222222222 U: 1.0 V: 0.8375 C: 0.9459459459459459 7: 0.8934729064039408 5: 0.923992673992674 4: 0.9671336206896552 M: 0.4861111111111111	6	0
100_0049.JPG	MVL 232	MVL232	MVL232	HVL232	H: 0.7104166666666667 V: 0.9696969696969697 L: 0.9666666666666667 2: 0.8846153846153846 3: 0.8414634146341464 2: 0.9230769230769231	5	1
100_0050.JPG	-	-	-	-	-	-	-

Paaiškinimai

Failas	Failo (iš testinių duomenų aibės) pavadinimas
Numeris	Realūs vaizde esančio automobilio numerio simboliai
Segmentai	Po segmentacijos likusių vaizdo segmentų aibė
Skeletas	Po segmentacijos likusių vaizdo segmentų aibės skeletonas
Rezultatas	Simbolių atpažinimo rezultatas
Koeficientai	Kiekvieno iš rezultate užfiksuoto simbolio atpažinimo koeficientas
+	Teisingai atpažintų simbolių skaičius
-	Neteisingai atpažintų simbolių skaičius
S	Rezultate užfiksuotas simbolis, kuris atpažintas neteisingai
S	Rezultate užfiksuotas pašalinis simbolis, kuris nepriklauso numeriui (tuo atveju, kai po segmentacijos lieka papildomų, neatitinkančių automobilio numerio simbolių, segmentų)

Brūkšniais užpildytos lentelės eilutės identifikuoja testinius pavyzdžius, kuriuose:

- nepavyko lokalizuoti numerio visiškai,
- arba pavyko lokalizuoti dalinai

Transporto priemonių numerių atpažinimo algoritmų analizė
bei universalios atpažinimo sistemos teorija

- lokalizuoti pavyko, tačiau segmentacija buvo nesėkminga – rezultate gražinta tuščia segmentų aibė.

Priedas E. Simbolių atpažinimo algoritmo klaidos

Lentelė 15 Simbolių atpažinimo klaidos

Simbolis	Viso rastas (kartų)	Neatpažintas (kartų)	Neatpažinimo (klaidos) procentas	Pakeitiniai
T	1	1	100,0%	F(1)
Z	1	1	100,0%	2(1)
O	7	6	85,71%	C(2), D(4)
6	6	4	66,70%	B(1), 8(1), 5(2)
F	3	2	66,70%	E(2)
M	3	2	66,70%	N(1), H(1)
3	14	8	57,14%	8(6), S(1), B(1)
J	7	4	57,14%	U(2), N(2)
D	2	1	50,00%	L(1)
U	2	1	50,00%	J(1)
9	11	5	45,45%	B(1), O(2), 8(2)
A	7	3	42,86%	4(3)
H	5	2	40,00%	M(1), N(1)
R	4	1	25,00%	H(1)
0	13	3	23,08%	C(1), O(1), 9(1)
C	9	2	22,22%	D(1), 1(1)
G	6	1	16,67%	C(1)
4	14	2	14,29%	M(2)
5	8	1	12,50%	S(1)
B	13	1	7,70%	P(1)
V	19	0	0,00%	
2	14	0	0,00%	
8	14	0	0,00%	
7	10	0	0,00%	
1	8	0	0,00%	
S	6	0	0,00%	
K	5	0	0,00%	
L	5	0	0,00%	
E	2	0	0,00%	
N	2	0	0,00%	
P	1	0	0,00%	
Y	1	0	0,00%	
I	0	0	0,00%	

Paiškinimai

Simbolis	Šiame stulpelyje yra visi šablone esantys simboliai (skaičiai ir raidės)
Viso rastas (kartų)	Nurodoma kiek kartų simbolis pasikartojo (testiniuose duomenyse)
Neatpažintas kartų	Nurodoma kiek kartų simbolis automobilio numeryje egzistavo, tačiau neatpažintas nebuvo

Neatpažinimo (klaidos) procentas	Procentinė simbolio neatpažinimo bei atpažinimo santykio išraiška
Kuom pakeistas	Nurodomi visi simboliai, kuriais neatpažintas simbolis buvo pakeistas (su kuriuo buvo „supainiotas“). Rodoma informacija: neteisingai atpažintas simbolis bei skliausteliuose nurodomas skaičius, kiek kartų su šis simbolis buvo neteisingai atpažintas.
