

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Baigiamasis bakalauro darbas

Simbolių atpažinimas paveikslėlių apdorojime
(Character Recognition in Picture Processing)

Atliko: 4 kurso, 1 grupės studentas

Augustas Lingys (parašas)

Darbo vadovas:

Rimantas Kybartas (parašas)

Recenzentas:

Valdas Dičiūnas (parašas)

Vilnius

2016

Turinys

Išvadas	3
1. Optinis simbolių atpažinimas.....	5
2. Atpažinimo tikslumas	6
3. OCR sudėtingo atpažinimo priežastys	7
4. „Pre-processing“ etapas ir jo metodai.....	8
4.1. Binarizacija.....	9
4.1.1. Globali binarizacija	11
4.1.2. „Otsu“ metodo binarizacija	11
4.1.3. Adaptyvi binarizacija	11
4.2. Triukšmo mažinimas	12
4.2.1. Gauso filtras	13
4.2.2. Bilateral filtras.....	14
4.2.3. Medianos filtras.....	15
5. Tyrimo metodas	16
5.1. Tyrimo duomenų aibė.....	16
5.2. Simbolių atpažinimo įrankis.....	17
5.3. Binarizacijos tyrimas	19
5.4. Triukšmo mažinimo tyrimas.....	20
5.4.1. Be papildomo triukšmo	21
5.4.2. Su papildomu triukšmu	22
Išvados	25
Summary	26
Literatūros sąrašas.....	27
Priedai	28

Išvadas

Simbolių atpažinimas dažnai laikomas jau išspręsta problema dėl labai pažangių technologijų ir pažengusių redagavimo metodų bei greito problemų sprendimo, tačiau dar kol kas neegzistuoja joks algoritmas, kuris atpažintų tekstą be klaidų bei nereikalingų papildomų vaizdo ar teksto redagavimų. Netgi patys geriausi algoritmai dar negali pasigirti 100% teisingo atpažinimo rezultatais ar tikslumu, todėl šiais laikais dirbama labiau ties sistemų kūrimo, kurios pasiekia geriausius rezultatus kuo įvairesnėse ir sudėtingesnėse situacijose, nekreipiant dėmesio į medžiagas, aplinkos veiksnius ar įvairias klaidas, kurios atsiranda dėl techninių kliūčių.

Prieš porą dešimtmečių susitelkę tyrimai ties simbolių atpažinimu (angl. Optical character recognition) buvo smarkiai apriboti faktu, kad tuo metu tinkamas nuotraukas paveikslėlių apdorojimui galėjo daryti tiksliai skaneriai. Skaneriai buvo dideli, masyvūs, ir juos nešioti iš vienos vietos į kitą buvo sudėtinga ir nepatogu, taip pat skanavimas tokių įrenginių užtrukdavo daug ilgiau negu padaryti paprastą nuotrauką fotoaparatu ar kitais įrenginiais. Todėl buvo reikalingas kuo greitesnis sprendimas skanerių problemų sprendimui. Iš pradžių buvo tobulinami patys skaneriai, nuo pačių lengviausių ar patogiausių iki greičiausių, kol buvo surastas dar optimaliausias sprendimas: programinės įrangos kūrimas, kuris turėjo visiškai panaikinti skanerių sukeltas problemas.

Nors naudojantis standartinėmis atpažinimo sistemomis ir skanerių atliktomis dokumentų ar paveikslėlių skaitmeninėmis kopijomis, įmanoma gauti labai aukštus ir užtikrintus atpažinimo rezultatus. Deja, dėl techninės įrangos bei kintančių, kitokių aplinkos sąlygų, tokių kaip tamsus apšvietimas ir dėl to nestabiliai padaryta išsiliejusi nuotrauka, ar iš nevienodo kampo padarytos nuotraukos, smarkiai pasunkina padėtį gauti panašius rezultatus su skaitmeniniais fotoaparatais darytomis nuotraukomis. Jas reikia daug labiau apdoroti nei paprastą baltą lapą su juodu šriftu, kokį skenuoja skaneris, o tai nėra stipriai efektyvu, bei gali atsirasti šiokių tokių netikslumų ar vaizdo iškreipimo po intensyvaus ir stipraus nuotraukų redagavimo.

Dažnai yra poreikis mobiliu įrenginiu apdoroti paveikslėliuką ir taip jo turinį ar kontekstą sudėti duomenis į lentelę ar gauti vientisą tekstą, kad nereikėtų rankomis perrinkti viso fragmento. Todėl, kuo geresnis paveikslėliuko apdorojimas, tuo tiksliau mes galime aptikti jame esančius simbolius ir po to atlikti dėl to mažiau veiksmų pataisant blogai nuskaitytą tekstą.

Kaip jau minėta, redaguoti nuotraukas yra būtina, kad kuo geriau atpažinimo įrankis nuskaitytų ir atpažintų simbolius bei žodžius. Iki šiol nėra padaryta daug tyrimų, kurie parodytų, kuriuo filtru redaguojant nuotraukas idant pašalinti vaizdo triukšmą yra efektyviausias nuskaitymas, ar kuris metodas yra geriausias binarizacijai atlikti, kad atpažinimo algoritmas nuskaitytų, kuo didesnę kiekį simbolių teisingai.

Todėl šio darbo tikslas yra išnagrinėti ir palyginti du paveikslėlių apdorojimo metodus: binarizaciją ir triukšmo mažinimą, kurie reikalingi simbolių atpažinimui iš fotografijų darytų įvairiomis sąlygomis, kuriose figūruoja daug spalvų bei triukšmo. Taip pat nustatyti, kuris metodas yra efektyviausias bei ištirti tokiomis sąlygomis geresnius atpažinimo rezultatus duodančius metodus. Naudojamų metodų palyginimas bus atliekamas praktiškai su pasirinkta duomenų aibe.

Pradžioje trumpai supažindinama su optinio simbolių atpažinimo, atpažinimo tikslumo bei prieš simbolių atpažinimą atliekamais veiksmais ir jo metodais. Toliau skaitytojas supažindinamas su binarizacijos bei triukšmo mažinimo metodais. Po to iškart seka praktinis eksperimentas, kuriame aprašyti prieš tai metodai palyginami. Darbo pabaigoje pateikiamos išvados bei pasiektų rezultatų santrauka.

1. Optinis simbolių atpažinimas

Teksto atpažinimas yra mašinų mokymo (angl. Machine learning) ir kompiuterių regėjimo (angl. Computer vision) uždavinys, dažniausiai vadinamas tiesiog optiniu simbolių atpažinimu. Optinis simbolių atpažinimas tai technologijos rūšis, kuri leidžia paversti paveikslėliuose esantį tekstą ar skanuotus dokumentus į redaguojamą tekstą, kurį galima panaudoti įvairiose srityse, tokiose kaip automatiniam vertime ar patogesniai duomenų perkėlimui iš fizinių dokumentų į įvairius įrenginius elektroniniu formatu.

Simbolių atpažinimo pradžia laikoma 1914 m., kai Emanuelis Goldbergas sukūrė mašiną, kuri gali skaityti simbolius iš lapo ir automatiškai juos konvertuoti į telegrafu siunčiama kodą. Šis, atrodo, primityvus būdas davė pradžią OCR kūrimui. Nors iš tikrųjų OCR buvo susidomėta tik XX amžiaus viduryje, kai pasaulyje pradėjo kurtis ir tobulėti kompiuterių industrija. Per daugiau nei pusę amžiaus padaryta begalė tyrimų ir dabar tobulais atvejais spausdintus simbolius atpažįstama net 99% tikslumu, o ranka rašytus simbolius apie 90%.

Kai žmogus skaito tekstą jo akys ir smegenys, taip pat atlieka simbolių atpažinimą jam to net nepastebint, tai išmokstamas gebėjimas žmogui. Akys atpažįsta struktūrą tarp balto ir juodo, kas ir sudaro simbolius (raidės, skaitmenys, skyrybos ženklai), o smegenys šią informaciją apdoroja ir sujungia į atskirtus tarpais ar skyrybos ženklais žodžius. Panašiai pagal šią idėją veikia ir OCR, tačiau kompiuteriu ją įgyvendinti daug sunkiau. Kompiuteris nesupranta, nei kas yra simbolis, nei kas yra tarpas, jam nuotrauka ar dokumentas tėra tik duomenų blokas, o tam kad galėtų atlikti atpažinimą jis turi vadovautis algoritmais, kurie bandys interpretuoti nuotraukoje esančius duomenis ir nustatyti, koks ten galėtų būti tekstas.

Tipinė OCR funkciją atliekanti sistema dažniausiai susideda iš 3 pagrindinių etapų:

1. Pre-processing;
2. Character Recognition;
3. Post-processing;

Pirmajame etape, pre-processing, atliekamos manipuliacijos su pačia nuotrauka, paruošiant ją kuo tinkamsnę simbolių nuskaitymui. Antras etapas (Character Recognition) susideda jau iš simbolio nuskaitymo iš nuotraukos. Post-processing, paskutinis etapas, bando apdoroti gautas simbolių sekas, lyginant su duotais žodynuose esančiais žodžiais, ar ieškant naudojamų panašių žodžių frazių, taip taisydamas nuskaitymo padarytas klaidas, ir gerindamas atpažinimo tikslumą. Šiame darbe bus dirbama prie pirmojo simbolių atpažinimo etapo – „Pre-Processing“.

2. Atpažinimo tikslumas

Simbolių atpažinimo sistemos yra skirtingos: vienos naudoja labiau pažengusias apdorojimo ir nuskaitymo technologijas ir algoritmus, kitos paprastesnius, dėl to kyla poreikis išmatuoti jų tikslumą, kad galėtume juos palyginti bei padaryti išvadas.

[Web01] Simbolių atpažinimo tikslumas nustatomas sudarant atsitiktinumo matricą (angl. „confusion matrix“), kurią sudaro 4 pagrindiniai elementai:

- True positive (TP) – atvejis, kai spėjome, kad tai yra tas simbolis ir atspėjome.
- True negative (TN) – atvejis, kai spėjome, kad tai yra ne tas simbolis ir atspėjome.
- False positive (FP) – atvejis, kai spėjome, kad tai yra tas simbolis, bet neatspėjome.
- False negative (FN) – atvejis, kai spėjome, kad tai nėra tas simbolis, bet neatspėjome.

Iš šių elementų galima nustatyti, tam tikrus santykius:

- Tikslumas (angl. „Accuracy“): bendrai, kiek atpažinimas kartų atspėja, kas tai yra ištikro.

$$\text{Tikslumas} = \frac{TP + TN}{TP + TN + FP + FN}$$

- Klaidingas klasifikavimas (angl. „Misclassification Rate“): kaip dažnai klysta atpažinimas.

$$\text{Klydimas} = \frac{FP + FN}{TP + TN + FP + FN}$$

(1 lent.) Tarkime turime 200 simbolių, atpažinimo algoritmas mano, kad 30 iš jų yra simbolis „B“ ir iš jų 25 iš tikro yra „B“, tai TP = 25, o FP = 5. Tada reiškia, likusius 170 simbolių algoritmas pripažino, kaip kitokių simbolių, tačiau iš tų 170 simbolių 10 buvo iš tikro „B“, tai TN = 160, o FN = 10.

Tokiu atveju, gauname, kad mūsų algoritmas atpažįsta simbolį „B“ tikslumu - 0,925 arba 92,5%, o suklysta tai darydamas – 0.075 arba 7.5%.

n = 200	Spėjimas: Taip	Spėjimas: Ne
Iš tikrųjų: Ne	TN = 160	FP = 5
Iš tikrųjų: Taip	FN = 10	TP = 25

1 lent. Pagal pavyzdinį uždavinį sudaryta atsitiktinumo matrica.

3. OCR sudėtingo atpažinimo priežastys

Ne visada įmanoma atpažinti vaizde esančius elementus naudojantis OCR pagalba. Atpažinimui gali trukdyti įvairūs trikdžiai, kuriuos galima būtų suskirstyti į keletą grupių pagal pobūdį. Pavyzdžiui, neteisingą atpažinimą įtakojantys aplinkos veiksniai, mechaniniai veiksniai, netinkama tekstūra ir panašiai. Surandant tam tikrą problemą, kuri trukdytų veikti OCR šimtu procentu, būtų galima greičiau surasti sprendimą tai problemai pašalinti bei efektyviau atlikti simbolių atpažinimą.

Nors žinomas raides ant popieriaus lapo OCR įrankis atpažįsta iki 99%, tačiau programoms sunku atpažinti skiriamuosius, diakritinius ženklus. Teksto atpažinimui trukdyti gali ir medžiaga, ant kurios parašytas tekstas. Tarkim suglamžytas popierius iškreipia raides, išteptas popierius taip pat gali pasunkinti OCR atpažinimą. Dažnai popieriuje esančios juodos raides gali pablukti, popierius pageltonyti nuo netinkamų laikymo sąlygų, taip pat gali būti ir netinkama teksto spalva, kuri yra sunkiai įskaitoma tokiu būdu dar labiau pasunkinant OCR nuskaitymą.

Panašios problemos gali būti ir fotografijose, įvairiose vaizduose, kur netinkama vaizdo kokybė pasunkina OCR darbą. Didelę įtaką daro ir aparato daroma nuotraukų kokybė, kurią nustato gamintojas. Tačiau ne tik fotoaparato ar kito darančio nuotrauką ar vaizdus aparato kokybė gali būti problema OCR atpažinimui. Pavyzdžiui, didelę problemą kelia vaizdo vadinamas triukšmas, kuris taip pat gali būti skirstomas į kelis pogrupius. Dažniausiai triukšmas skirstomas į vaizdo ryškumo ir spalvų triukšmą. Vaizdo ryškumas daro įtaką nuotraukos grūdėtumui. Spalvų triukšmas gali taip pat sugadinti nuotraukos kokybę, suteikdamas nuotraukai nenatūralumo bei iškreipti tam tikras vaizdo spalvas. Šis triukšmas atsiranda fotografuojant su dideliu jautrumu - ISO arba darant nuotrauką blogai apšviestoje patalpoje arba kurioje dominuoja tamsa.

Kita problema apart triukšmo gali būti ir nuotraukos padėtis. Pakreiptas vaizdas, kad ir nedideliu laipsniu, gali suklaidinti OCR atpažinimą ir dėl to atpažinimas gali būti netikslus, arba OCR programa atpažinimo gali nevykdyti iš viso. Šią problemą atsirasti skatina įvairūs žmogiškieji faktoriai, kurių, deja, panaikinti negalime, tačiau gali būti taikomas nuotraukos pakreipimas tinkamu kampu prieš atliekant OCR nuskaitymą.

Dar viena priežastis, kodėl OCR atlikti gali būti sudėtinga, tai pernelyg didelė spalvų įvairovė arba per ryškus spalvos intensyvumas, kuris trukdo taikyti slenksčio metodą. Tačiau ir pernelyg blankios spalvos gali nesuteikti intensyvaus kontrasto, todėl OCR dėl šios priežasties gali būti taip pat sunkiai įgyvendinamas.

Kiti svarbūs veiksniai, sukeliantys OCR atpažinimo klaidas ir problemas yra aplinkos veiksniai, kaip pavyzdžiui, nuotraukoje esantis fonas susilieja su pagrindiniais objektais ir dėl to OCR negali atskirti atpažinimo objekto nuo šalutinių arba fono. Trukdyti gali ir nuskaitymui reikalingiausius

objektus uždengiantys elementai, pavyzdžiui medžių šakos, lapija, įvairūs laidai ar, tarkim, lauko šviestuvai. Šios problemos yra beveik neišsprendžiamos, todėl būtina atsakingiau parinkti fotografavimo kampą ar vietą, kur įvairūs aplinkos objektai nedarytų jokios įtakos OCR nuskaitymui.

4. „Pre-processing“ etapas ir jo metodai

Šio etapo pagrindinis tikslas yra paveikslėlių apdorojimas prieš simbolių atpažinimą. Jis reikalingas tam, kad paveikslėliai būtų paruošti apdorojimui ir taip žymiai pagerintų atpažinimo rezultatus. Tai pasiekama jau turint nufotografuotą paveiksluką ir jam taikant programinius metodus, kurie padeda sumažinti paveikslėlių defektus ir padaro paveikslėlį „patrauklesniu“, tinkamesniu nuskaitymui.

Šiame etape laikomi svarbiausi metodai ir idėjos:

- Binarizacija
- Triukšmo mažinimas
- Nuotraukos pasukimas (angl. „De-skew“)
- Linijų pašalinimas
- Žodžio aptikimas
- Simbolio izoliacija arba segmentacija

Binarizacija – paverčia paveikslėlį iš spalvoto į vien baltą ir juodą. Šis procesas yra labai reikalingas, nes dauguma komercinių atpažinimo algoritmų veikia tik su binariniais paveikslėliais.

Triukšmo mažinimas naudojamas, pašalinti smulkiems defektams, t.y dulkėms, įbrėžimams, grūdėtumui, spalvų kokybės sugadinimui.

Nuotraukos pasukimas naudojamas, tada kai dokumente arba tekste esantis tekstas neina tolygiai ir horizontaliai, ir būna truputi pasuktas į kūrį nors šoną, todėl jį reikia pasukti prieš arba pagal laikrodžio rodyklę.

Linijų šalinimas – metodas atsakingas už linijų pašalinimą paveikslėlyje, jog jos nebūtų palaikytos kaip tekstas ar kitaip trukdytų simbolių atpažinimui.

Žodžio aptikimas – randamos žodžio vietos paveikslėlyje, analizuojamos simbolio formos ir atskiria žodžius, jeigu tai yra būtina.

Simbolio izoliacija arba segmentacija – iš prieš tai rasto žodžio fragmento, ištraukiami atskiri simboliai pačiam simbolių atpažinimui.

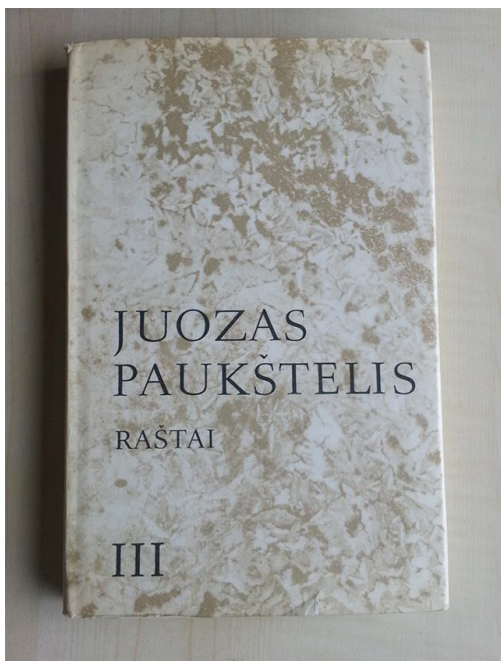
4.1.Binarizacija

Binarizacija yra pats svarbiausias „pre-processing“ metodas, kurio pagalba atskiriamas tekstas nuo fono paverčiant spalvotą ar pilką paveikslėlį į vien baltą ir juodą spalvas turintį paveikslėlį, arba kitaip tariant, į binarinį paveikslėlį (angl. „Binary Image“). Binarinis atvaizdas turi tik dvi galimas reikšmes kiekvienam atvaizdo pikseliui. Balta spalva suteikiama fonui, o juoda - tekstui.

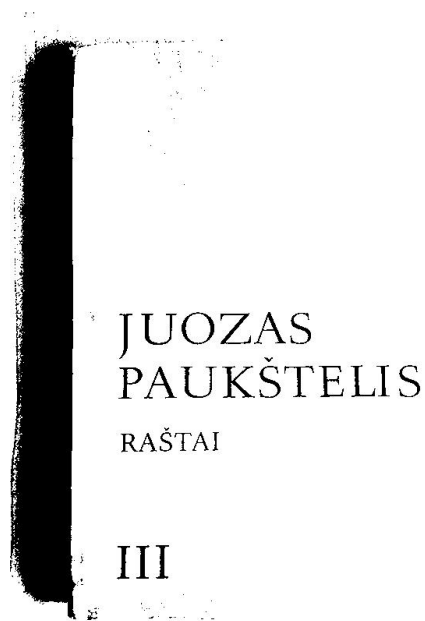
Jei paveikslėlis gan puikios kokybės, tai „binarizacijos“ metodo metu, kiekvieną juodą spalvą turinti vieta taps galimo šrifto vieta, o kitos spalvos tiesiog bus balta. Šis žingsnis svarbus, nes dauguma šiuolaikinių atpažinimo algoritmų veikia tik su binariniais atvaizdais, nes taip žymiai paprasčiau jiems atlikti šią užduotį.

Egzistuoja daug „binarizavimo“ algoritmų, tačiau daugumos pagrindinis principas panašus – parenkamas, koks nors slenkstis (angl. „Threshold“), nuo kurio žiūrima, jei jį peržengia pikselio reikšmė, tada nudažomas baltai, jeigu ne juodai.

Pateikiamas pavyzdys (**1 pav.** ir **2 pav.**), kaip pasikeičia paveikslėlis atliekant binarizaciją.

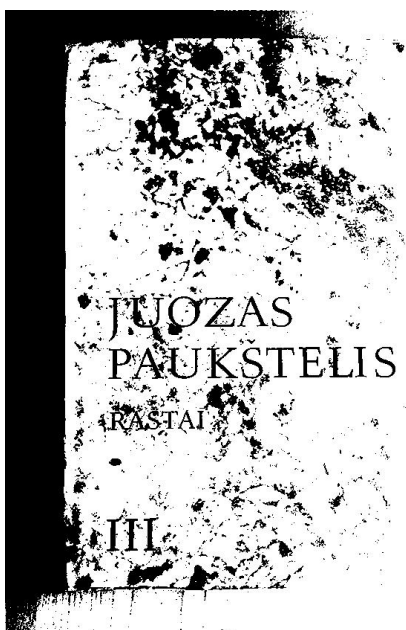


1 pav. Originalus paveikslėlis



2 pav. Paveikslėlis po „binarizacijos“ metodo („Threshold“ – 90)

Slenksčio parinkimas, svarbus veiksmas. Parinkus per mažą slenkstį, dalis teksto gali tapti baltu, taip galimai prarandami simbolių duomenys (**3 pav.**). Parinkus per didelį slenkstį, po binarizacijos proceso gali atsirasti daug pašalinių juodų pikselių, kurie gali net pabloginti simbolių atpažinimą, lyginant su originalia nuotrauka (**4 pav.**).



3 pav. Paveikslėlis po „binarizacijos“ metodo
(„Threshold“ – 40)



4 pav. Paveikslėlis po „binarizacijos“ metodo
(„Threshold“ – 140)

Pateikiama lentelė (**2 lent.**) [SCR+09], kaip slenksčio pasirinkimas gali daryti įtaką atpažinimo rezultatams.

Slenkstis	Atpažinimo tikslumas
0.50	0.99
0.55	0.99
0.60	0.99
0.65	0.98
0.70	0.96
0.75	0.91
0.80	0.86
0.85	0.82
0.90	0.76
0.95	0.67

2 lent. Atpažinimo rezultatų priklausomybė nuo slenksčio

Binarizacija labai svarbi simbolių atpažinime, ją visada apsimoka pritaikyti, be jos simbolių atpažinimas sunkiai įsivaizduojamas. Tik reikia parinkti tinkamą slenksį, ties kuriuo klasifikuojami pikseliai į juodus ir baltus, kadangi tai turi įtakos atpažinimo rezultatams. Toliau bus pateikiami ir nagrinėjami metodai binarizacijai paveikslėliuose atlikti.

4.1.1. Globali binarizacija

Globalus slenkstis naudoja tik intensyvumo tašką, kuris yra apskaičiuojamas pasiremiant statistikomis arba euristika (sprendimo metodika, kai sprendimai gaunamas bandymų ir klaidų keliu) viso paveikslėlio tam tikrais požymiais suskirstant pikselius į pirmo ir antro plano pikselius.

Šis metodas tinka labai didelio kontrasto nuotraukoms apdoroti, didžiausias šio metodo privalumas yra efektyvumas, taip pat ir skaičiavimo laikas yra labai trumpas. Kita vertus, mažo kontrasto vaizdams, šio metodo naudojimas yra netinkamas, nes gali būti neteisingai parinktas slenkstis.

Pagrindinis šio metodo minusas yra tas, kad globalaus slenksčio metodas negali atskirti tų pikselių, kurie yra to pačio pilko atspalvio, tačiau nepriklauso tai pačiai grupei (fono ar pirmo plano).

4.1.2. „Otsu“ metodo binarizacija

Konvertuoti daugiatoonį pilkų atspalvių paveikslėlį į vienspalvį yra dažna vaizdų atpažinimo užduotis. Otsu metodas, pavadintas šio metodo išradėjo Otsu Nobuyuk 大津展之 *Ōtsu Nobuyuk* vardu, yra vienas iš daugelio binarizacijos algoritmų, kuris yra labai plačiai naudojamas dėl to, jog yra paprastas ir gana veiksmingas.

[Web04] Šio metodo veikimas yra pagrįstas histograma – stulpeline diagrama, kurioje yra pavaizduota, kiek kartų reikiamo intensyvumo taškas buvo panaudotas paveikslėlyje. Otsu metodo tikslas yra surasti intensyvumo tašką, kur pirmo plano ir antro plano suma sudaro jų minimumą. Otsu metodo algoritmas išskiria dvi grupes: pirmąjį planą ir antrąjį planą. Šios dvi klasės yra taip supaprastinamos, kad kuo mažiau spalvų varijuotų pasiskirstyme.

Kaip ir minėjau, iš pat pradžių, pati histograma yra supaprastinama, kad nebūtų jokių didelių skaičių, kurie iliustruotų tam tikro vaizdo intensyvumo taško pasikartojimą. Visas taško pasikartojimas yra padalijimas iš visų taškų, sudarančių visą vaizdą, skaičiaus.

Tačiau šis metodas turi ir trūkumų. Šio metodo taikyti neįmanoma, kai apšvietimas yra nepastovus. Taip pat jis nėra labai produktyvus, priešingai nei globalaus slenksčio metodas, Otsu metodo apskaičiavimo laikas yra ganėtinai ilgas, taip pat gali atsitikti taip, kad algoritmas gali neteisingai padalinti histogramą į pogrupius ir dėl to tolesnis nuskaitymas būtų negalimas.

4.1.3. Adaptyvi binarizacija

[Fta04] Adaptyvi binarizacija arba lokalus slenkstis naudojamas tik ribojame plote arba paveikslėlio konkrečioje vietoje. Palyginus su globalaus slenksčio metodu, adaptyvi binarizacija yra

efektyvesne pašalinti triukšmui ir klaidoms, ypač kai apdorojama informacija šalia tekstų arba objektų.

Lokalaus slenksčio algoritmai skaičiuoja patį slenkstį, naudodami šalia esančių pikselių reikšmes, masyvus, arba blokus, kurio dydis yra lygus vaizdo pikselių skaičiui, iš kurių randant optimalų, svertinį vidurkį, kuris apskaičiuojamas pagal formulę:

$$Th = \frac{1}{n} \sum_{k=1}^n P_k ,$$

Kur n = vaizdo dalies dydis.

Pasinaudojant šiuo metodu, ne tik yra pašalinamas triukšmas ar klaidos, bet gali kartu pašalinti ir šešėlį, kuris trukdo OCR vykdyti atpažinimą. Taip pat kitaip nei globalaus slenksčio metodas, jis gali būti naudojamas tiek su mažu, tiek ir su didelio kontrasto vaizdais. Šis algoritmas taip pat sunaudoja mažai apimties, nes vaizdas yra skaitomas tik masyvais, lokaliomis vietomis. Tačiau kita vertus, gautas vaizdas, po adaptyvaus slenksčio panaudojimo, yra akivaizdžiai suskirstytas į blokus, o tai gali šiek tiek sutrikdyti OCR atpažinimą.

4.2. Triukšmo mažinimas

Nuotraukos, ypatingai tos, kurios padarytos pigesniu fotoaparatu, ar padidintu fotoaparato jautrumu, ar tiesiog esant mažam apšvietimui, arba net tamsai, neišvengiamai turės šią kokybės defektą, vadinamą triukšmu.

Paveikslėlių triukšmas atsiranda tada, kai fotoaparato sensorius nepajėgia išgauti pakankamai šviesos iš fotografuojamos aplinkos, todėl kai kurie nuotraukos pikseliai yra sugeneruojami atsitiktinai. Kadangi mes atliekame simbolių atpažinimą, mums svarbu gauti kuo „švaresnę“, kokybiškesnę nuotrauką.

Fotografavimo sąlygas sunku pakeisti į tinkamesnes, kai negalime pagerinti fotoaparato fotografavimo kokybės, todėl praktikoje dažniausiai naudojamas triukšmo mažinimas (angl. „Noise reduction“) programiniu būdu.

Trumpai tariant, tai nuotraukų švelninimas, kai nuotraukoje esantis pikseliai lyginami su kaimyniniais pikseliais ir taip išskiriamas pikselio vidurkis, kuris padeda sumažinti esantį triukšmą.



5 pav. Paveikslėlis prieš metodą



6 pav. Paveikslėlis po metodo



7 pav. Paveikslėlis po kelių triukšmo mažinimo metodų

Tačiau taikyti triukšmo mažinimo metodus reikia saikingai. Pritaikant jį daug kartų, nors ir dingsta didžioji dalis triukšmo, tačiau nuotrauka gali tapti neryški, o tai kenkia pagal nustatytus parametrus veikiančiam simbolių atpažinimui (**7 pav.**).

[Ema08] Darbe išbandyta keletas “pre-processing” metodų, taip pat ir mūsų aptarinėjama triukšmo mažinimo idėja. Gauti rezultatai atliekant bandymus su neryškiomis nuotraukomis pateikiami lentelėje (**3 lent.**).

“Pre-Processing” metodas	Atpažinimo rezultatas(%)
Nenaudojamas	20.97
Triukšmo mažinimas	49.04

3 lent.

Žiūrint į lentelę, matosi jog naudojant triukšmo mažinimą neryškioms nuotraukoms, atpažinimo rezultatas padidėjo netgi 30%.

Trumpai apibendrinus, triukšmo mažinimas turėtų pagerinti atpažinimą nuotraukose, kurios labai išsiliejusios ir neryškios, tačiau jeigu nuotrauka gera, ir jai bus taikomas, koks triukšmo mažinimo algoritmas, atpažinimas gali pablogėti.

Toliau bus nagrinėjami triukšmo mažinimo filtrai, kurie mažina triukšmo kiekį nuotraukose.

4.2.1. Gauso filtras

Gauso suglotninimo filtras dažniausiai taikomas vaizdo suglotninimo procese, ieškant konkrečių kontūrų. Šis vaizdo suglotninimas sumažina vaizdo triukšmus, pašalina smulkias detales, kurios būna nepageidaujamas ir trukdo atpažinti tam tikrus simbolius.

Pagal apibrėžimą, Gauso filtras yra filtras, kurio vienetinio impulso perdavimo funkcija yra Gauso funkcija (arba apytikslis skaičius jai).

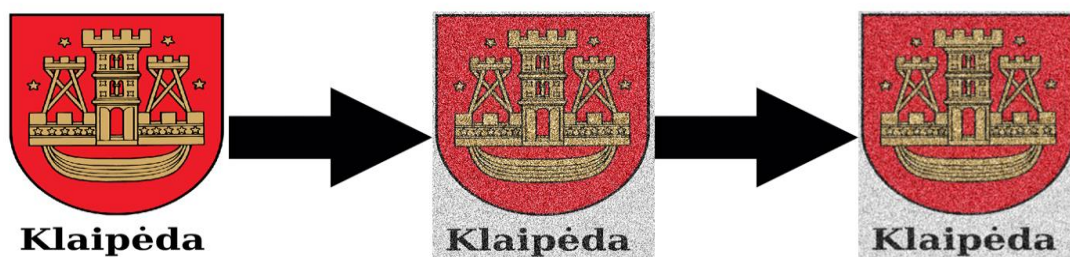
[Dav90] Fotografijos vaizdas visada yra projektuojamas plokštumoje, todėl vaizdas yra dvimatis. Dėl to filtrui privaloma naudoti izotropinį Gauso skirstinį su maksimumu, kai $x=y=0$. Toks skirstinys aprašomas dviejų argumentų lygtimi:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}},$$

kur σ yra standartinis pasiskirstymo nuokrypis.

Gauso filtro funkcija visada išveda „svertinį vidurkį“ kiekvieno pikselio esančio kraštuose su pikselių einančių link centro svertiniu vidurkiu. Dėl šios priežasties, priešingai nei kiti filtrai, Gauso filtras suteikia švelnumo bei išsaugo vaizdo kraštus.

Pateikiamas paveikslėlis (8 pav.), kuriame pirmoje vietoje originali nuotrauka. Toliau programiniu kodu pirmajai nuotraukai suteikiamas triukšmas. Paskutinė nuotrauka yra Gauso filtro rezultatas.

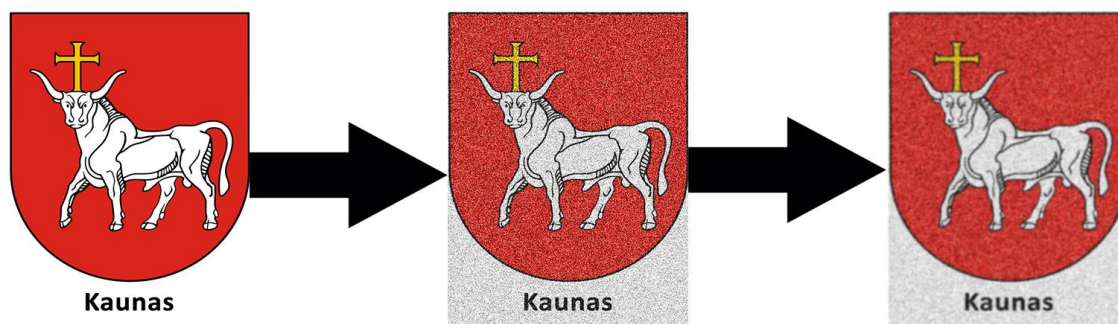


8 pav. Gauso filtro rezultatas

4.2.2. Bilateral filtras

Bilateral filtras (angl. „Bilateral Filter“) nelineinis, kraštus apsaugantis bei vaizdo triukšmą šalinantis glotninimo filtras vaizdams. [Web02] Kai Gauso filtras apskaičiuoja svertinį vidurkį pikselių reikšmių kaimynystėje link centro, todėl vidurkis mažėja einant link centrinio taško, tai bilateral filtras yra palyginus gana paprastas. Šio filtro tonų santykis kiekviename vaizdo pikselyje yra pakeičiamas tonų santykio svertiniu vidurkiu šalia esančių pikselių. Du pikseliai gali būti arti vienas kito, tai yra užimti šalia esančią erdvinę vietą, arba jie gali būti labai panašūs vienas su kitu ir turėti reikšmingą reikšmę. Šis filtras priklauso tik nuo dviejų reikšmių, kurios indikuoja dydį ir kontrastą. Todėl jį lengva naudoti ir yra greitas.

Pateikiamas paveikslėlis (9 pav.), kuriame atliekami kaip ir prieš tai aprašytuose metodo veiksmai, tik triukšmo mažinimui parinktas bilateral filtras.



9 pav. Bilateral filtro rezultatas

4.2.3. Medianos filtras

Kaip ir kiti filtrai, medianos (angl. „Median Filter“) filtras naudojamas mažinti triukšmui nuotraukoje. [Tho98] Aprašyta filtro logika paprasta. Parametro pagalba nusistatoma, kiek kaimynų algoritmas turi tikrinti, tada bėgama per kiekvieną paveikslėlio pikselį paeiliui ir žiūrima į visus jo kaimyninius pikselius, yra tikrinama ar jis ten dera. Numanoma reikšmė yra paskaičiuojama paimant visus jo kaimyninius pikselius, jie yra išrikiuojami iš eilės, didėjimo tvarka ir pasirenkama vidurinė reikšmė, matematinės medianos principu. Jeigu, skaičius lyginis, paimamos dvi vidurinės reikšmės ir padalinamos per pusę. Pavyzdys pateikiamas **4 lent.**

120	123	124	125	130
123	124	125	127	132
117	121	149	129	131
118	114	118	122	131
111	116	110	120	120

4 lent. Medianos filtro algoritmo pavyzdys

Skaičiuojama medianos filtro reikšmė viduriniam pikseliui, kurio reikšmė dabar 149 ir jo 8 kaimynų. Jo kaimynai yra 124, 125, 127, 121, 129, 114, 118, 122. Išrikiuoti didėjimo tvarka: 114, 118, 121, 122, 124, 125, 127, 129. Taigi mediana gaunasi $(122 + 124) / 2 = 123$. Tai ir būtų naujoji pikselio reikšmė po filtro.

Pateikiamas paveikslėlis (**10 pav.**), kuriame atliekami kaip ir prieš tai aprašytuose metoduose veiksmai, tik triukšmo mažinimui parinktas medianos filtras.



10 pav. Medianos filtro rezultatas

5. Tyrimo metodas

Tyrimo metu bus siekiama palyginti binarizacijos ir triukšmo mažinimo metodus, tam tikromis sąlygomis, ieškant metodo, kuris pasiektų geriausią rezultatą. Atpažinimo rezultatas bus pateikiamas tiek, kiek procentų žodžių buvo teisingai atspėta bei kiek procentaliai simbolių atpažįstama. Binarizacijos bandyme nieko papildomai nebus daroma. Tam, kad pasunkinti triukšmo mažinimo tyrimą, pridėsime jam papildomo triukšmo programiniu būdu patikrinti, kaip aprašyti prieš tai metodai susidoroja su didėjančiu triukšmu.

5.1. Tyrimo duomenų aibė

Tyrimo simboliai atpažįstami iš [Web03] „KAIST Scene Text“ duomenų aibėje esančių paveikslėlių, tarp kurių yra tekstas anglų ir korėjiečių kalbomis. „KAIST“ duomenų rinkinyje yra virš 3000 paveikslėlių, padarytų skirtingose aplinkose, įskaitant ir nuotraukas, padarytas viduje ir gatvėse, skirtingu apšvietimu. Nuotraukose fonas su tekstu varijuoja nuo knygų viršelių, stendų, reklamų, elektrinių švieslenčių iki durų iškabų. Nuotraukos darytos ir su profesionaliais fotoaparatais, ir su paprastomis išmaniųjų telefonų kameromis, tačiau visos nuotraukos sumažintos iki rezoliucijos 640x480. Pateikiama pora nuotraukų iš rinkinio (**11 pav**). Taip pat, kiekvienas iš paveikslėlių kartu turi ir savo informacinį failą. Failas yra xml struktūros, ir jame aprašomi esantys simboliai paveikslėlyje.

Pasirinkimą lėmė, tai jog fone mirga įvairiausios spalvos, dėl kurių turėtų išryškėti binarizacijos metodų efektyvumas, taip pat teksto šriftas bei spalva, kiekvienoje nuotraukoje skirtinga, o tai turėtų pasunkinti atpažinimą ir kiekvienas nuotraukai padarytas pokytis daryti įtaką atpažinimo tikslumui.

Tyrimo nagrinėsime tik anglų kalbos atpažinimą, todėl iš 3000 paveikslėlių lieka jų 400. Dėl geresnių rezultatų, 350 paveikslėlių bus paskirti atpažinimo įrankio apmokymui, o bandymai atliekami su kitais 50 paveikslėlių. Atrinktuose paveikslėliuose yra 126 žodžiai ir 681 simboliai.



11 pav. Kelios nuotraukos iš “KAIST” rinkinio

5.2.Simbolių atpažinimo įrankis

Tyrimui atlikti pasirinktas atvirojo kodo (angl. „Open Source“) simbolių atpažinimas įrankis „Tesseract“. Varikliukas buvo pradėtas kurti „Hewlett Packard“ laboratorijose 1985 m. tik asmeniniui kompanijos naudojimui. Iki 1998 m. įrankio kodas buvo smarkiai keičiamas, perrašomas kitomis programavimo kalbomis ir po to buvo apleistas. Tačiau 2005 m. šis įrankis tapo prieinamas visiems. O nuo, 2006 m. „Tesseract“ programa smarkiai susidomėjo kompanija „Google“ ir pradėjo vėl remti jo vystymą tam, kad šis įrankis būtų pritaikytas išmaniesiems telefonams.

Šio įrankio pasirinkimą lėmė tai, jog ši programa yra nemokamas ir laikoma viena geriausiai nemokama atpažįstančių simbolių programa, kokios iki šio yra sukurtos. Taip pat, viena iš šio įrankio blogybių yra ta, kad „Tesseract“ naudoja labai minimalų paveikslėlių apdorojimą. Todėl, paveikslėliai paduodami atpažinimui turi būti stipriai apdoroti prieš operaciją, kitaip atpažinimo rezultatai bus labai blogi. Kadangi mūsų tyrimas bus atliekamas su binarizacija ir triukšmo mažinimu, šios dvi apdorojimo idėjos turėtų pagerinti atpažinimo rezultatus.

„Tesseract“ neturi jokio jokios grafinės sąsajos ir yra paleidžiama iš komandinės eilutės nurodant paveiksluką kuriame reikia atpažinti simbolius, atpažinto teksto išeities failą ir pagalbinius parametrus. Pavyzdžiui: su „-l eng“ duodama užuomina įrankiui, kad paveiksluke esantis tekstas anglų kalba.

```
tesseract test.png output test -l eng
```

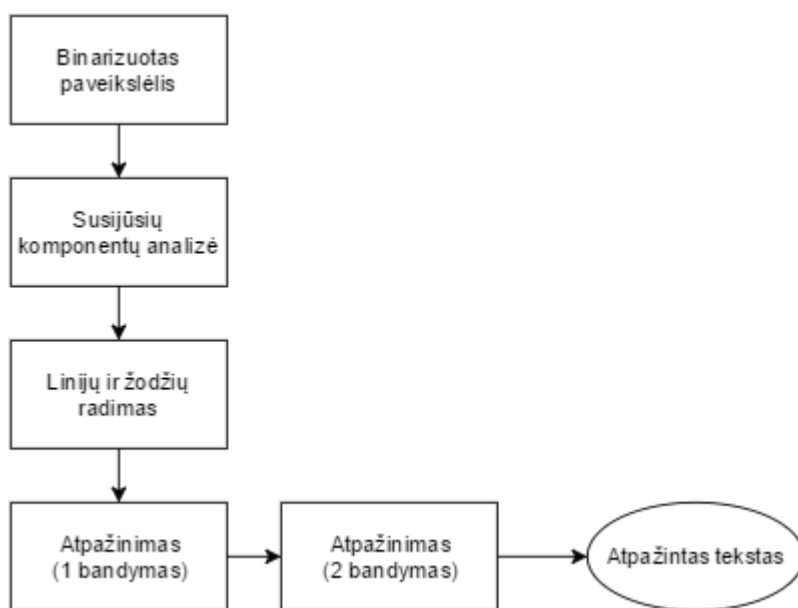
[Smi07] Šis įrankis naudoja statistinį metodą, kai išskiria manomą simbolį iš aplinkos, tada nustato simbolio apytikslus kampus bei apskaičiuoja atstumus tarp tų išskirtų kampų. „Tesseract“ atpažinimo įrankis žingsnis po žingsnio atlieka OCR veiksmus (**12 pav.**). Šie žingsniai yra:

1. Susijusių komponentų analizė: šis žingsnis yra naudojamas išskirti simbolio kontūrus. Ypatingai šio metodo nauda atsiskleidžia, tada kai vaizdas pavyzdžiui yra balto teksto su juodu fonu. „Tesseract“ tikriausiai buvo pirmasis, kuris panaudojo tokio tipo atpažinimo metodą.

2. Linijų ir žodžių radimo žingsnyje prieš tai rasti kontūrai yra surenkami kartu ir sugrupuojami į nepriklausomus fragmentus.

3. Fragmentai yra organizuojami į teksto linijas. Vėliau linijos ir teksto sritys yra analizuojamos, ir pagal analizę tekstas yra dalinamas į žodžius visais atvejais - kai tarpas tarp žodžių yra aiškus ir kai dėl tarpo buvimo dvejojama, daromas vis tiek perskyrimas.

4. Atpažinimo procesas yra atliekamas per du atpažinimo bandymus. Po pirmojo atpažinimo bandymo gauti simbolių ir žodžių atpažinimo rezultatai yra siunčiami adaptyviam klasifikatoriui, kuris pasinaudoja gautais duomenimis ir apsimoko. Po to, bandoma atpažinti tekstą antrą kartą, bet šį kartą jau naudojantis adaptyviojo klasifikatoriaus išmokta informacija. Priežastis, kodėl atpažįstama iš antrojo karto yra paprasta. Pirmojo atpažinimo rezultatai duoda mums informaciją apie teksto kontekstą. Pavyzdžiui, pirmojo bandymo pabaigoje žodžių atpažinimo rezultatai buvo daug geresni nei pradžioje, todėl ši informacija būtų apmokama klasifikatoriui, ir antra kartą jis galbūt jau žinos teisingą rezultatą ir pradžioje. Tai labai panašu į tai, kaip mes žmonės mokomės ko nors. Pirmą kartą perskaitome mokomąją medžiagą, kad sužinotume, kas yra tekste. Antra arba trečią skaitant jau suprasdami kontekstą, galime gilintis ir bandyti suprasti viską.



12 pav. „Tesseract“ vykdomi atpažinimo žingsniai

5.3. Binarizacijos tyrimas

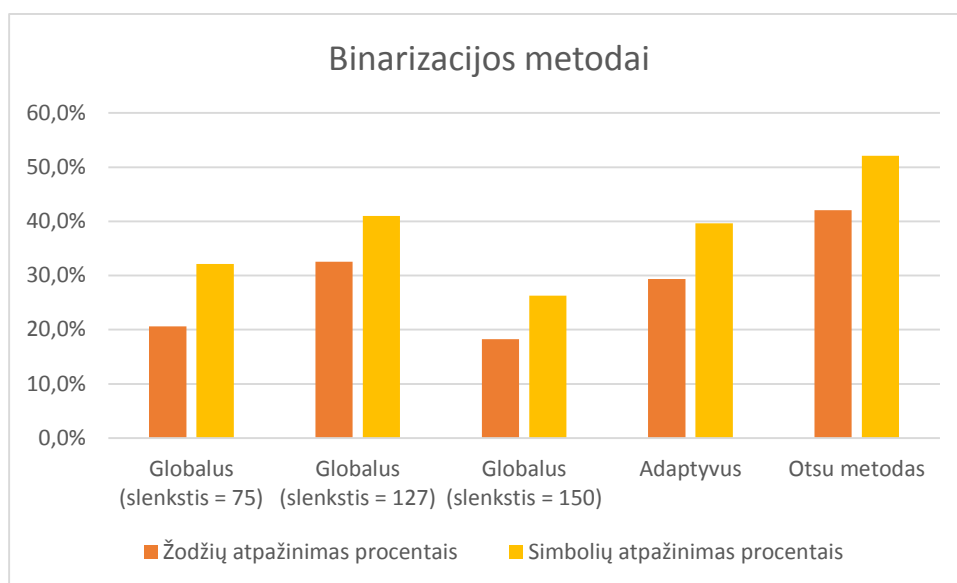
Binarizacijos tyrimui atlikti naudojau tris metodus: globalųjį, adaptyvųjį ir Otsu. Globaliajam metodui parinkau tris skirtingus slenksčius: 75, 127, 150. Nuotraukos buvo binarizuotos pasitelkiant „OpenCV“ biblioteką ir Python programavimo kalbą ir po veiksmų atliekamas atpažinimas ir gauti rezultatai pateikiami (5 lent.) ir (1 diag.).

```
import cv2
img = cv2.imread(sample.png', 0)
#Globali binarizacija
th_global_75 = cv2.threshold(img, 75, 255, cv2.THRESH_BINARY)
th_global_127 = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY)
th_global_150 = cv2.threshold(img, 150, 255, cv2.THRESH_BINARY)
#Otsu metodas
th_otsu = cv2.threshold(img, 0, 255, cv2.THRESH_BINARY+cv2.THRESH_OTSU)
#Adaptivi binarizacija
th_adaptive = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_MEAN_C,
cv2.THRESH_BINARY, 11, 5)
```

Matoma, kad globalaus metodo nuotraukos redagavimas, su slenksčiu - 75 atpažinimas yra vienas prasčiausių. Žodžius atpažino tik 26 iš 126, ir tai sudaro 20,6%, kai simbolių atpažino 219 iš 681, kurie sudaro 32,2%. Toliau parinkau tą patį metodą, tik su slenksčiu 127, kuris yra parinktas iš 255, o tai yra vidurinis variantas, todėl ir rezultatai gauti yra geresni už praeitą bei už globalaus metodo su slenksčiu 150 atpažinimą, kurio slenkstis pasirodė per didelis. Iš 126 žodžių, su slenksčiu - 127 atpažinta 32,5%, kai su - 150 atpažino tik 18,3%, kuris yra prasčiausias atpažinimo procentas iš visų bandymų. Su slenksčiu - 127 simbolių atpažino 41% palyginant su globalaus metodu, kai slenkstis buvo parinktas 150, tai yra 14,7% daugiau. Nuotraukos, redaguotos adaptyviosios binarizacijos metodu, OCR programa atpažino 37 žodžius iš 126 ir 270 simbolių iš 681. Ketvirtąjį būdą metu, naudojant Otsu metodą, buvo pasiekti geriausi rezultatai. Otsu metodu redaguojant vaizdą, buvo atpažinta 53 žodžiai, kurie sudaro 42,1%, o simbolius atpažino 355, kurie sudaro net 52,1% teisingo atpažinimo. Šis binarizacijos metodas yra efektyviausias, todėl jis yra pasirinktas taikyti kartu su tolesniais triukšmo mažinimo tyrimais.

Binarizacijos metodas	Žodžių atpažinimas		Simbolių atpažinimas	
	Atpažinti žodžiai	Procentais	Atpažinti simboliai	Procentais
Globalus (slenkstis = 75)	26	20,6%	219	32,2%
Globalus (slenkstis = 127)	41	32,5%	279	41,0%
Globalus (slenkstis = 150)	23	18,3%	179	26,3%
Adaptivus	37	29,4%	270	39,6%
Otsu metodas	53	42,1%	355	52,1%

5 lent. Binarizacijos rezultatai



1 diag. Binarizacijos rezultatų diagrama.

5.4. Triukšmo mažinimo tyrimas

Atliekant tik triukšmo mažinimo paveikslėlio apdorojimą ir bandant jį iškart atpažinti, būtų gaunami labai prasti rezultatai, binarizacija yra būtina. Todėl, pasirinkta apjungti triukšmo mažinimo metodų tyrimą su prieš tai geriausiai pasirodžiusiu binarizacijos metodu – Otsu. Kad geriau įsivaizduoti vykdomą eigą, pateikiamas paveikslėlis (13 pav.). Iš pradžių paveikslėlyje atliekamas triukšmo mažinimas, tada binarizacija ir galų gale atliekamas atpažinimas iš apdoroto paveikslėlio.



13 pav. Triukšmo mažinimo tyrime atliekami veiksmai

Filtrai buvo įgyvendinti taip pat, „OpenCV“ bibliotekos pagalba ir „Python“ programavimo kalba.

```

import cv2
img = io.imread("sample.png")
#Gauso filtras
blur_gauss = cv2.GaussianBlur(img, (5,5), 0)
#Bilateral filtras
blur_bil = cv2.bilateralFilter(img, 9, 75, 75)
#Medianos filtras
blur_median = cv2.medianBlur(img, 5)
  
```

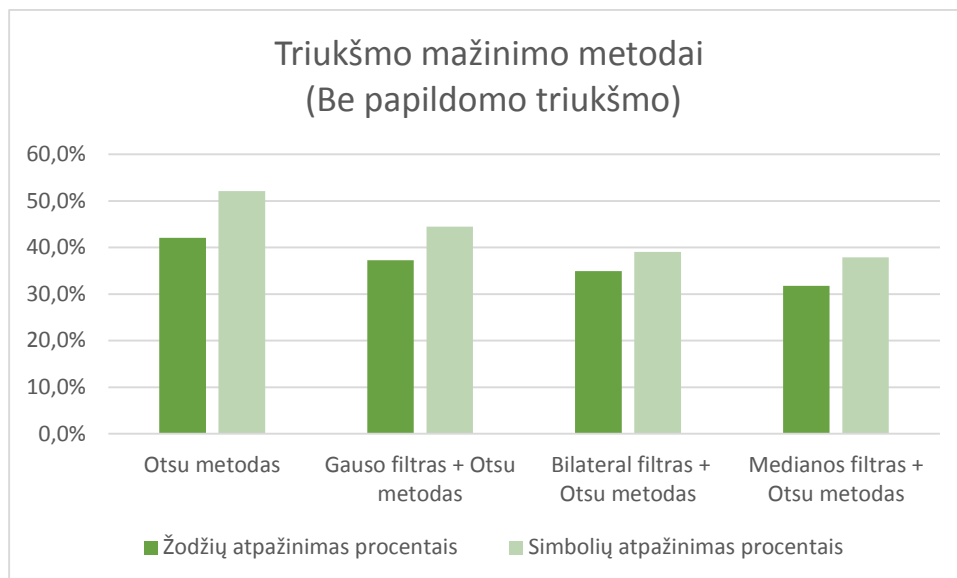
5.4.1. Be papildomo triukšmo

Nuotraukų atpažinimui be papildomo triukšmo, naudojau tris metodus gauso filtrą, bilateral filtrą bei medianos filtrą kartu su Otsu binarizacijos metodu.

Žvelgiant į **6 lent.** bei **2 diag.** matome, jog naudojant medianos filtrą su Otsu, rezultatai gavosi prasčiausi. Atpažinimo programa atpažino 40 žodžių ir 258 simbolių, kai bandymo metu naudojamas buvo bilateral filtras su Otsu metodu, programa atpažino 44 žodžius, kurie sudaro 34,9% ir 266 ženklus iš 681. Kita vertus, antru bandymu, naudojant Otsu metodą su Gauso filtru, rezultatai gavosi geresni nei pastarųjų dviejų, tačiau vis dėl to negeriausi. Šio bandymo metu parinktas metodas ir filtras atpažino 47 žodžius ir 303 simbolius. Geriausiai programa atpažino naudojant tik Otsu metodo binarizacija. Šiuo metodu redaguotas nuotraukas ir iš jų žodžių atpažino 42,1%, kuris yra 4,8% didesnis nei trečiojo bandymo rezultatai. Iš 681, jų atpažino 355, kas yra daugiau nei 50%. Tokius rezultatus lėmė tai, jog nuotraukos buvo geros kokybės, be papildomo triukšmo, todėl nenaudojant jokių filtrų, kurios nuotraukų nešvelnina ir nedarant jų blankesnėmis, tyrimas, naudojant tik Otsu binarizacija, buvo labiausiai pasisekęs, tikriausiai tai lėmė, jog duomenų aibė, kurioje atliekamas tyrimas buvo labai kokybiška.

Triukšmo mažinimo metodai	Žodžių atpažinimas		Simbolių atpažinimas	
	Atpažinti žodžiai	Procentais	Atpažinti simboliai	Procentais
Otsu metodas	53	42,1%	355	52,1%
Gauso filtras + Otsu metodas	47	37,3%	303	44,5%
Bilateral filtras + Otsu metodas	44	34,9%	266	39,1%
Medianos filtras + Otsu metodas	40	31,7%	258	37,9%

6 lent. Rezultatai, be papildomo triukšmo.



2 diag. Rezultatų diagrama, be papildomo triukšmo.

5.4.2. Su papildomu triukšmu

Šį kartą bandymą atlikau su tais pačiais filtrais ir Otsu metodu kaip ir pastarąjį tyrimą, tik nuotraukoms buvo pridėtas papildomas triukšmas per „Matlab“ su programine eilute:

```
noisyImage = imnoise(image, type, variance);
```

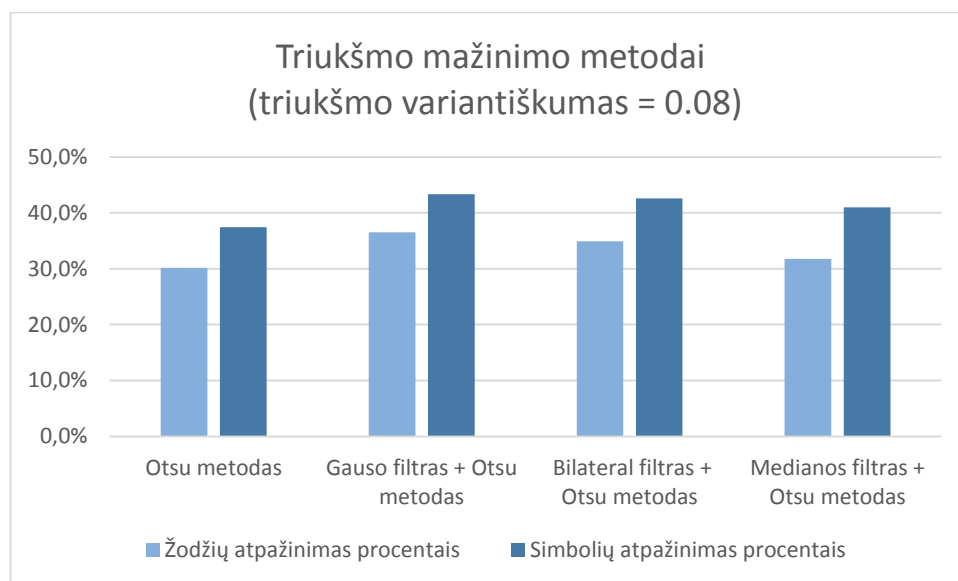
Triukšmo tipas pasirinktas – ‘**Speckle**’, dėl jo panašumo į kameros sukiamą triukšmą. Taip pat, bandymai atliekami su variantiškumu **0.08** ir **0.16**, didesnės reikšmės jau smarkiai pakeičia nuotrauką, ir pačiam skaitytojui būtų sunku atpažinti simbolius.

Kaip matoma, variantiškumo = 0.08 rezultatuose, geriausiai nuotraukas atpažino naudojant Gauso filtrą su Otsu metodu, aplenkiant prieš tai tyrimo metu geriausiai triukšmo mažinimo metodu naudotą tik Otsu metodą. Pastarasis žodžius atpažino 38, ir tai yra 8 žodžiais mažiau tuo pačiu Gauso filtru redaguojant nuotraukas. Simbolius atpažino 255, kurie sudaro 37,4% ir tai yra 5,9% mažiau nei naudojant Gauso filtrą kartu. Mažiausiai efektyvus pasirodė medianos filtras su Otsu, kurio žodžių atpažinimo procentas yra 31,7%, o simbolių atpažinimo procentas tik 41%. Bilateral filtras kita vertus yra vienas iš optimaliausių variantų, tačiau ne efektyviausiai padeda sumažinti triukšmą taip, kad OCR įrankis atpažintų maksimaliai.

Kita vertus, palyginus prieš tai buvusį bandymą, kai nuotraukos buvo perfiltruotos be triukšmo su šiuo, jau galime pamatyti atsirandančią filtrų naudą, kai nuotraukų kokybė yra suprastinama ir pridedama šiek tiek triukšmo.

Triukšmo mažinimo metodai (triukšmo variantiškumas = 0.08)	Žodžių atpažinimas		Simbolių atpažinimas	
	Atpažinti žodžiai	Procentais	Atpažinti simboliai	Procentais
Otsu metodas	38	30,2%	255	37,4%
Gauso filtras + Otsu metodas	46	36,5%	295	43,3%
Bilateral filtras + Otsu metodas	44	34,9%	290	42,6%
Medianos filtras + Otsu metodas	40	31,7%	279	41,0%

7 lent. Triukšmo mažinimo rezultatai, su pritaikytu triukšmu 0.08

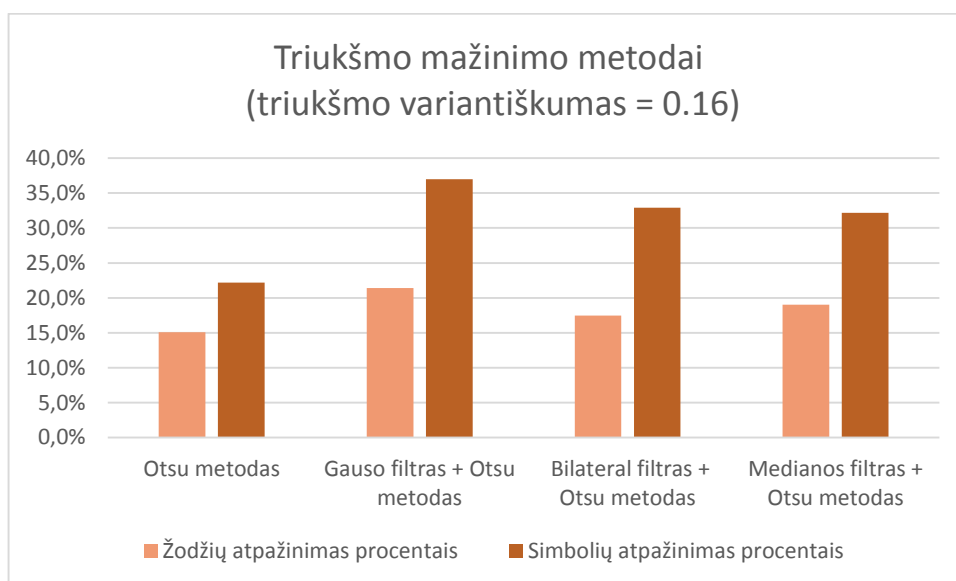


3 diag. Triukšmo mažinimo rezultatų diagrama, su pritaikytu triukšmu 0.08

Paskutiniajame bandyme, triukšmo pridėta su variantiškumu = 0.16. Nors žodžių atpažinta mažiau palyginus su prieš tai buvusiu bandymu, tačiau pastebima reali filtrų nauda. Kaip ir prieš žvelgiant į **7 lent.** ir **4 diag.** efektyviausiai nuotraukas atpažino, kai buvo filtruojama Gauso filtru ir naudojant Otsu binarizacijos metodą. Šio junginio žodžių atpažinimo taiklumo procentas yra 21,4% ir tai yra 6,3% daugiau nei naudojant tik Otsu metodą. Ženklų atpažinime, Gauso filtru redaguota nuotrauka buvo taip pat palankesnė, iš viso ženklų atpažino 37%, o tai yra 14,8% daugiau nei naudojant tik Otsu binarizaciją. Geriau nei vien naudojant Otsu binarizaciją žodžių atpažinime pasirodė Medianos filtras, o simbolių atpažinime – Bilateral filtras.

Triukšmo mažinimo metodai (triukšmo variantiškumas = 0.16)	Žodžių atpažinimas		Simbolių atpažinimas	
	Atpažinti žodžiai	Procentais	Atpažinti simboliai	Procentais
Otsu metodas	19	15,1%	151	22,2%
Gauso filtras + Otsu metodas	27	21,4%	252	37,0%
Bilateral filtras + Otsu metodas	22	17,5%	224	32,9%
Medianos filtras + Otsu metodas	24	19,0%	219	32,2%

7 lent. Triukšmo mažinimo rezultatai, su pritaikytu triukšmu 0.16



4 diag. Triukšmo mažinimo rezultatų diagrama, su pritaikytu triukšmu 0.16

Išvados

OCR nuskaitymui ir simbolių atpažinimui atlikti neigiamą įtaką gali daryti keletas veiksnių, kurie skirstomi į keletą pogrupių: aplinkos, žmogiškųjų veiksnių bei nuotraukų kokybė. Didžiausią problemą kelia nuotraukų spalvų daugiaspektriškumas, kai pernelyg daug spalvų trukdo atpažinimui bei pernelyg didelis vaizdo triukšmas, kuris gadina nuotraukos kokybę. Todėl viename iš svarbiausių OCR nuskaitymo etapų, „pre-processing“, yra taikomi tokie metodai, kaip triukšmo mažinimas ar binarizacija.

Nagrinėjami trys triukšmo mažinimo filtrai: gauso, bilateral ir medianos filtrai. Gauso filtras išveda „svertinį vidurkį“ kiekvieno pikselio esančio kraštuose su pikselių einančių link centro svertiniu vidurkiu. Dėl šios priežasties, priešingai nei kiti filtrai, gauso filtras suteikia švelnumo bei išsaugo vaizdo kraštus. Bilateral filtras pakeičia tonų santykis kiekviename vaizdo pikselyje su tonų santykio svertiniu vidurkiu šalia esančių pikselių. Medianos filtro metodu, parametro pagalba nusistatoma, kiek kaimynų algoritmas turi tikrinti, tada bėgama per kiekviena paveikslėlio pikselį paiešiai ir žiūrima į visus jo kaimyninius pikselius ir yra tikrinama ar jis ten dera.

Binarizacijos nagrinėti metodai: globalus metodas, „Otsu“ slenksčio metodas ir adaptyvioji arba, kitaip, lokaloji binarizacija. Globalios binarizacijos metu, vaizdas gaunamas bandymo ir statistikos būdais, viso paveikslėlio tam tikrais požymiais suskirstant pikselius į pirmo ir antro plano pikselius. Otsu slenksčio metu yra išvedama histograma, kuri vėliau supaprastinama tam, kad surastų intensyvumo tašką, kuris pasikartotų ir tokiu būdu sudarytų naują vaizdą pirmo ir antro plano grupėmis. Adaptvyją binarizacija naudojama tik ribojame plote, kai sudaromi blokai, kurių dydis yra lygus vaizdo pikselių skaičiui.

Iš viso padaryti keturi tyrimai: binarizacijos tyrimas naudojant globalųjį metodą su slenksčiu 75, 127 ir 150, lokalųjį metodą ir Otsu metodą; triukšmo mažinimo filtrų tyrimas, kur buvo atlikti trys tyrimai - be papildomo triukšmo nuotraukoms, bei su pridėtu papildomu triukšmu. Binarizacijos tyrimuose geriausiai pasirodė Otsu metodas, o tarp triukšmo mažinimo metodų geriausiai pasirodė – Gauso filtravimas.

Tikimasi, kad tyrimo rezultatai skaitytojų leis greičiau pasirinkti tokį binarizacijos ir triukšmo mažinimo metodą, kuris leistų efektyviau pasiekti norimus rezultatus, nesugaištant papildomo laiko tikrinant pačiam, kuriuo metodu geriausiai apdoroti nuotraukas prieš simbolių atpažinimą.

Summary

Character Recognition in Picture Processing

Optical character recognition converts physical documents and images to machine-encoded text and transmits it to electronic devices. However, a lot of interferences do not let scan images efficiently. In order to scan more effective, certain methods of editing image are invented. In this inquiry, three image noise reducing filters were explored: Gaussian, bilateral and median filters. Moreover, the investigated process of binarization also helps in optical character recognition by reducing multi-specter colors in order to make recognition faster and infallible. The methods of banalization that are analyzed here are also three: Otsu method, global and adaptive thresholding.

In the analysis of binarization with a chosen data set, it became known that Otsu method is much more efficient than the others. In the other researches, image noise reducing filters and methods are being investigated. In the research where any kind off additional voice was absent, only using Otsu method binarization was the best and the vast majority of the words and symbols were recognized. In others researches with additional noise, it was noticeable that the more image noise is added, the more use of various filters becomes efficient and their help can be seen. In this inquiry, the Gaussian filter with Otsu thresholding was the most beneficial.

Literatūros sąrašas

- [Dav90] E. Davies, „Machine Vision: Theory, Algorithms and Practicalities“, Academic Press, 1990, 42 – 43.
- [Ema08] Megan Elmore, Margaret Martonosi “A Morphological Image Preprocessing Suite for OCR on Natural Scene Images“ (http://www.cra.org/Activities/craw_archive/dmp/awards/2008/Elmore/melmore_dmp.pdf), 452 KB.
- [Fta04] Meng-Ling Feng, Yap-Peng Tan „Adaptive binarization method for document image analysis,, (<http://web.mit.edu/mfeng/www/papers/01394198.pdf>), 288 KB.
- [Smi07] Ray Smith „An Overview of the Tesseract OCR Engine,, (<http://static.googleusercontent.com/media/research.google.com/lt/pubs/archive/33418.pdf>), 164 KB.
- [SCR+09] Rodolfo P. dos Santos, Gabriela S. Clemente, Tsang Ing Ren and George D.C. Calvalcanti “Line Segmentation Based on Morphology and Histogram Projection” (<http://www.cvc.uab.es/icdar2009/papers/3725a651.pdf>), 629 KB.
- [Tho98] R. Boyle, R. Thomas, Computer Vision: A First Course, Blackwell Scientific Publications, 1988, 33 - 34.
- [Web01] Guide to confusion matrix terminology (<http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology>)
- [Web02] A Gentle Introduction to Bilateral Filtering and its Applications (<http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology>)
- [Web03] „KAIST“ Scene Text Database (http://www.iapr-tc11.org/mediawiki/index.php/KAIST_Scene_Text_Database)
- [Web04] Otsu Thresholding (<http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html>)
- [ZJK+10] Mi Zhang , Anand Joshi , Ritesh Kadmawala , Karthik Dantu , Sameera Poduri† and Gaurav S. Sukhatme “OCRdroid: A Framework to Digitize Text Using Mobile Phones” (http://www-scf.usc.edu/~mizhang/papers/mi_mobicase09.pdf), 3.17MB.

Priedai

1 priedas. Kompaktinis diskas

Prisegtas prie darbo kompaktinis diskas.