



VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
KOMPIUTERIJOS KATEDRA

Magistro baigiamasis darbas

**Triukšmo šalinimas duomenyse naudojant genetinius
algoritmus**

Atliko:

Rokas Kasinskas

parašas

Vadovas:

doc. dr. Tadas Meškauskas

Vilnius
2017

Turinys

Santrauka	4
Summary	5
Iyadas	6
1. Skaitmeniniai rastriniai vaizdai	8
1.1. Triukšmų tipai rastriniuose vaizduose	8
1.1.1. Gauso baltasis triukšmas	8
1.1.2. Druskos ir pipirų triukšmas	9
1.1.3. Kontrasto sumažinimas	9
1.1.4. Ryškumo sumažinimas	10
1.2. Rastrinių vaizdų filtrai	11
1.2.1. Slenkstinis filtras	11
1.2.2. Gauso filtras	11
1.2.3. Medianos filtras	11
1.2.4. Maksimumo ir minimumo filtrai	11
1.2.5. Wiener dekonvoliucijos filtravimo algoritmas	12
1.2.6. Morfologiniai filtrai	12
1.2.7. Chambolle filtravimo algoritmas	13
1.2.8. Nugludinimo kaukės filtras	13
1.2.9. Gauso - Laplaso filtras	13
1.2.10. Furje filtras	14
1.3. Vaizdų kokybės metrika	14
1.3.1. MSE įvertis	14
1.3.2. OCR tikslumas	15
2. Dirbtiniai neuroniniai tinkai	16
2.1. Aktyvacijos funkcija	17
2.2. Apmokymo algoritmai	18
2.2.1. Žingsninis klaidos sklidimo atgal algoritmas	18
2.2.2. Paketinis klaidos sklidimo atgal algoritmas	18
2.2.3. QuickProp algoritmas	18
2.2.4. RProp algoritmas	19
2.2.5. SARProp algoritmas	19
3. Genetiniai algoritmai	20
3.1. Atrankos metodai	20
3.2. Kryžminimo metodai	21
4. Susiję darbai	23
4.1. DNT ir GA taikymas vaizdų filtravime	23
4.2. Automatinio filtravimo OCR metodai	24
5. Metodas ir jo realizavimas	26
5.1. Filtravimo metodo apžvalga	26

5.2. Metodo realizavimas	26
5.3. Paskirstytasis skaičiavimas	28
6. Tyrimas ir jo rezultatų analizė	30
6.1. Triukšmų nustatymas vaizde naudojant DNT	30
6.1.1. Optimalių DNT parametrų nustatymas	30
6.1.2. Kiti regresiniai algoritmai	34
6.2. Sudėtinių filtrų gavimas naudojant GA	35
6.2.1. Optimalių GA parametrų nustatymas	35
6.2.2. Sudėtinių filtrų gavimo rezultatai	38
6.3. Automatinis vaizdų filtravimas	40
6.4. Realių duomenų automatinis filtravimas	43
6.5. Metodo palyginimas su kitais automatiniais vaizdo filtravimo įrankiais	44
Išvados ir rekomendacijos	46
Ateities tyrimų gairės	47
Literatūros šaltiniai	48

Santrauka

Šiam darbe nagrinėjamas rastrinių vaizdų filtravimas naudojant genetinius algoritmus bei neuroninius tinklus optinio simbolių atpažinimo dalykinėje srityje. Šio darbo tikslas - įgyvendinti automatinį triukšmų filtravimo metodą, siekiant pagerinti optinio simbolių atpažinimo tikslumą bei įvertinti gautus rezultatus. Pagal šį metodą, naudojant neuroninius tinklus, yra įvertinami triukšmai investiniame vaizde, bei pagal šiuos rezultatus parenkamas ir pritaikomas optimalus filtras, kuris yra gautas naudojant genetinį algoritmą. Šiame darbe yra apžvelgiami triukšmai ir naudoti filtrai, genetiniai ir neuroninių tinklų algoritmai, aprašomas pasiūlytas filtravimo metodas ir jo realizacija, nustatomi optimalūs algoritmų parametrai bei įvertinami gauti rezultatai ir metodo naudingumas. Gauti tyrimo rezultatai parodo, kad šiuo metodu pagrįstas rastrinio vaizdo filtravimas pagerina vaizdo kokybę bei padidina optinio simbolių atpažinimo tikslumą, naudojant metodą sintetiniams vaizdams filtruoti. Norint pasiekti panašius rezultatus su realiais vaizdais, metodą reikėtų tobulinti.

Raktiniai žodžiai: rastriniai vaizdai, filtravimas, genetiniai algoritmai, neuroniniai tinklai, triukšmai, optinis simbolių atpažinimas, filtrai.

Summary

OCR (optical character recognition) - is raster image digitalizing method which is widely used nowadays and definitely will be used in the future. There are still many documents, books, articles and other papers which is still not digitalized this way. Digitalizing and using OCR improves documents saving, storing, using, searching, processing capabilities. However, various noise types can prevent accurate optical recognition. These noises may be consequence of scanning, processing tasks or defects in the document itself. These noises must be removed to get satisfactory optical character recognition results. One of the ways to achieve that is by using genetic algorithms. Genetic algorithm in this case can be used in various ways: creation of new filters, combining filters, or by using direct image transformation. However, the main problems which may occur when using these methods for filtering images in the practice are these: the original image without noises is needed or the type and level of noise must be known.

The automated method of image filtering by using genetic algorithms is suggested in this thesis. The main goal of this thesis is to implement suggested method and evaluate filtering results by applying this method. The suggested denoising method has these steps. At first noises types and levels are evaluated in the input image by artificial neural network. According to these results, corresponding filter is applied, which had been constructed by using genetic algorithm. These filters consists of some other simple filters, and vary from each other by different level of filter application and the order of application.

This thesis consists of these parts. At first used filters and noises are defined. The second part describes genetic and artificial neural network algorithms. Related works are reviewed in the following part. The suggested method and its implementation is described in the next part. Finally, results of the experiment are stated and evaluated. The results show that filtering according to this method decreases level of noise in the image and increases optical recognition accuracy in synthetic images. However, to get similar results with real images, method must be improved.

Keywords: artificial neural networks, genetic algorithms, images, filters, noises, optical symbol recognition.

Ivyadas

Optinis simbolių atpažinimas (OCR, *angl.* optical symbol recognition) - tai būdas iš skaitmeninio rastrinio vaizdo, kuriame pavaizduotas tekstas, išgauti simbolių reikšmes. Optinis simbolių atpažinimas, kaip spausdintinių tekstų skaitmenizavimo būdas, yra plačiai taikomas jau kurį laiką ir bus taikomas ateityje, kadangi yra dar daug knygų, leidinių, įvairių dokumentų kuriuos bus siekiama skaitmenizuoti. Dokumento skaitmenizavimas naudojant optinį simbolių atpažinimą labai palengvina dokumento saugojimą, apdorojimą, naudojimą, paiešką jame.

Teisingam simbolių nuskaitymui gali trukdyti įvairūs su tekstinio vaizdo kokybe susiję veiksniai, kurie gali būti vadinami triukšmais. Triukšmai gali būti kelių tipų - tai su pačiu skanuojamu objektu susiję triukšmai (įvairūs defektai ir nusidėvėjimas), bei nuskaitymo, perdavimo, glaudinimo, apdorojimo metu atsirandantys triukšmai. Siekiant tikslaus optinio simbolių atpažinimo, šie triukšmai turi būti pašalinti arba kiek įmanoma sumažinti. Vienas iš būdų to pasiekti yra naudoti genetinius algoritmus. Genetinius algoritmus vaizdų filtravime galima naudoti keliais būdais: kurti naujus filtrus (derinti kelis paprastesnius filtrus [25] [6]) arba naudoti tiesioginei vaizdo transformacijai [26]. Tačiau yra keletas problemų, apsunkinančių genetinių algoritmų naudojimą praktikoje. Dažniausiai pasitaikančios problemos: filtravimui tokiu būdu reikalingas originalus vaizdas be triukšmų [26] arba turi būti žinomas triukšmo tipas arba lygis [6]. Be to, genetiniam algoritmui realizuoti reikalingi dideli skaičiavimų resursai, ir jei genetinis algoritmas naudojamas vaizdo filtravimui tiesiogiai, toks filtravimas yra itin nepraktiškas [15] [11].

Šiame darbe aprašomas metodas, kuriuo yra siekiama apeiti pagrindines kliūtis genetinių algoritmų praktiniam taikymui šioje srityje. Taigi šio darbo užduotis: įgyvendinti automatinį triukšmų šalinimo metodą, siekiant pagerinti tekstinio vaizdo kokybę ir padidinti optinio simbolių atpažinimo tikslumą, bei įvertinti gautus rezultatus. Šiame darbe pasiūlytas tekstinių vaizdų filtravimo metodas, kuriam įgyvendinti panaudojami genetinis algoritmas ir dirbtinis neuroninis tinklas. Pagal šį metodą triukšmas yra pašalinamas automatiškai: pagal įvestinio vaizdo triukšmo apskaičiavimo rezultatus, kurie gaunami naudojant dirbtinių neuroninių tinklų skaičiavimo modelį, yra pritaikomas optimalus sudėtinis filtras, kuris yra gautas naudojant genetinį algoritmą konkrečiai tam triukšmo tipui bei lygiui.

Šio darbo pradžioje yra apibūdinami tyrime naudojami triukšmai ir filtrai. Vėliau apibūdinami genetiniai bei neuroninių tinklų algoritmai, jų savybės ir parametrai. Kitoje dalyje apibūdinamas metodas bei aprašoma jo realizacija. Toliau aprašomas optimalių genetinio algoritmo bei dirbtinių neuroninių tinklų optimalių parametrų parinkimas. Kitoje dalyje atliekamas išsamus metodo tyrimas, įvertinami metodo rezultatai su sintetiniais bei realiais vaizdais, įvertinamas pasiūlyto metodo naudingumas. Galiausiai metodas palyginamas su kitais prieinamais automatinio filtravimo įrankiais.

Įgyvendinus šį metodą, gauti rezultatai parodo, kad šis metodas padeda sumažinti triukšmų lygį rastriniame tekstiniam vaizde bei pagerinti optinio simbolių atpažinimo tikslumą, naudojant jį su sintetiniais vaizdais. Naudojant metodą realioms vaizdams, nėra pasiekiami tokie geri rezultatai, tačiau nepaisant to įrodyta, kad šis metodas gali konkuruoti su tai pačiai taikymo sričiai skirtais automatinio filtravimo įrankiais. Kadangi ne visais atvejais OCR tikslumą pavyko pagerinti, norint toliau plėtoti šį metodą, reikėtų jį labiau pritaikyti realioms vaizdams bei pagerinti triukšmų įvertinimo vaizde rezultatus.

Ilustracijos bei teksto fragmentai panaudoti iš autoriaus ankstesnio darbo „Mokslo tiriamasis darbas: Triukšmo šalinimas duomenyse naudojant genetinius algoritmus“: Rastrinių vaizdų bei triukšmų apibūdinimas, slenkstinio, Gauso, medianos, minimumo, maksimumo filtrų apibūdinimas

mas, vaizdų kokybės metrikos aprašymas (1 skyrius); DNT aprašymas ir pagrindinės sąvokos (2 skyrius); GA aprašymas bei pagrindinės sąvokos (3 skyrius); DNT ir GA taikymo vaizdų filtravime aprašymas (4 skyrius); Filtravimo metodo apžvalga ir realizavimas (5 skyrius); DNT optimalios struktūros bei lango dydžio įvertinimas, kitų regresinių algoritmų įvertinimas (6 skyrius); taip pat atskiri fragmentai panaudoti įvade ir santraukoje.

1. Skaitmeniniai rastriniai vaizdai

Skaitmeniniai rastriniai vaizdai (*angl.* digital raster images, toliau - vaizdai) - tai vaizdinės informacijos užkodavimo būdas, kai vaizdas yra sudarytas iš vaizdo taškų (pikselių). Šie vaizdai kompiuteryje užkoduojami kaip dvimatis skaičių masyvas (arba matrica). Dvimačio masyvo atveju vaizdo taško koordinatės atitinka masyvo eilės numerius. Reikšmė atitinkamame masyvo elemente reiškia spalvą. Šiame darbe nagrinėjami tik nespaltoti vaizdai, taigi pikselio spalvą atitinka viena reikšmė.

1.1. Triukšmų tipai rastriniuose vaizduose

Triukšmas apskritai apibrėžiamas kaip nepageidaujama signalo dalis, kuri trukdo apdoroti naudingą signalo dalį. Taigi triukšmas rastrinių vaizdų kontekste - įvairūs veiksniai ir defektai, trukdantys išvelgti originalų vaizdą, naudingą vaizdo dalį. Triukšmai gali būti kelių tipų ir rūšių, o jų atsiradimo priežastys skirtingos. Tekstinio vaizdo atveju triukšmai iškreipia objektų (raidžių) kontūrus, pakeičia ryškumo savybes. Šiame poskyryje apžvelgiami pagrindiniai rastrinių vaizdų triukšmai ir jų sintetinis modeliavimas. Kiekvienas šių triukšmų gali būti apibūdinamas triukšmo lygiu.[8].

1.1.1. Gauso baltasis triukšmas

Gauso triukšmas - tai triukšmas turintis normalinį pasiskirstymą. Baltojo Gauso triukšmo vidurkis yra lygus 0. Toliau aprašomas ir naudojamas tik Gauso baltasis triukšmas, kuris vadinamas tiesiog Gauso triukšmu. Rastrinių vaizdų atveju jis pasireiškia kaip spalvos skirtumas tarp triukšmo paveikto ir nepaveikto vaizdo. Vizualiai - tai grūdėtas vaizdas, ir praktikoje astrianda dėl vaizdo fiksavimo sensoriaus jautrumo svyravimo. Šiuo triukšmo tikimybės tankis [7]:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \quad (1.1)$$

Triukšmo lygis gali būti apibūdinamas kaip vidutinis kvadratinis nuokrypis σ .



1 pav. Baltuoju Gauso triukšmu paveikto vaizdo pavyzdys (kai $\sigma=4200$).

1.1.2. Druskos ir pipirų triukšmas

Druskos ir pipirų triukšmas - tai vaizde atsitiktinėse vietose esantys juodi ir balti taškai [27]. Triukšmo lygis atitinka tokių taškų skaičiaus ir viso vaizdo taškų skaičiaus santykį:

$$s = \frac{B + W}{MN} \quad (1.2)$$

Čia: B,W - juodų ir baltų taškų kiekis; M,N - vaizdo dydis pikseliais. Šiame darbe laikoma, kad juodų ir baltų pikselių kiekis yra lygus (B=W).



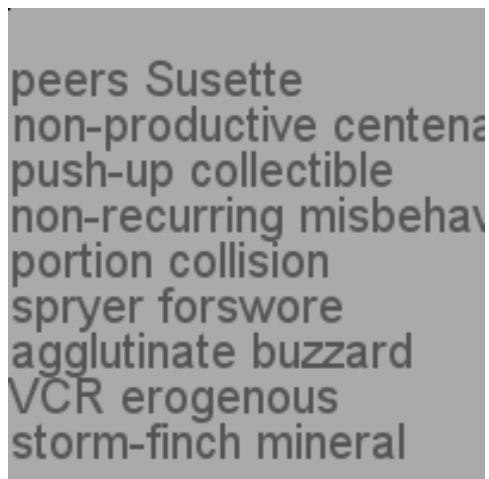
2 pav. Druskos ir pipirų triukšmu paveikto vaizdo pavyzdys (kai $B=W=0.14$).

1.1.3. Kontrasto sumažinimas

Kontrastas - tai dydis proporcingas skirtumui tarp vaizdo taško spalvos maksimalios ir minimalios reikšmės visame vaizde (skirtumas tarp objekto ir fono). Triukšmas šiuo atveju laikomas kontrasto sumažinimu. Kontrasto sumažinimas gali būti apibūdinamas kaip operacija su kiekvienu vaizdo tašku:

$$y_{ij} = x_{ij}(1 - L) + L \frac{x_{max} - x_{min}}{2} \quad (1.3)$$

Čia: i, j - vaizdo taško koordinatės; L - kontrasto sumažinimo lygis; x_{min} , x_{max} - maksimali ir minimali vaizdo spalvos reikšmės vaizde, x - filtruojamo vaizdo taškas.



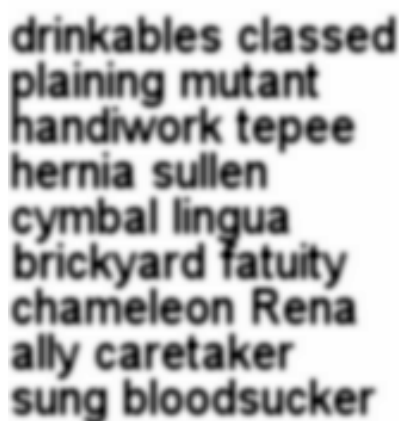
3 pav. Vaizdo, kuriam sumažintas kontrastas, pavyzdys (kai $L=0.665$).

1.1.4. Ryškumo sumažinimas

Ryškumas - vaizdo savybė, kuri apibūdina objektų kontūrų apibrėžtumą. Šiuo atveju triukšmas gali būti įvardinamas kaip ryškumo sumažinimas (*angl.* blur). Vaizde, kuriame yra mažas ryškumas, objektai yra išplaukę ir jų kontūrai sunkiai apibrėžiami. Ryškumo sumažinimo operacija gali būti kaip apibūdinama konvoliucijos operacija tarp vaizdo ir Gauso operatoriaus[27]:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1.4)$$

Čia σ - vidutinis kvadratinis nuokrypis; x, y - filtravimo lango dydis. Ryškumo sumažinimo lygis atitinką vidutinį kvadratinį nuokrypį σ .



4 pav. Vaizdo, kuriam sumažintas ryškumas, pavyzdys (kai $\sigma=1.26$).

1.2. Rastrinių vaizdų filtrai

Filtrai rastrinių vaizdų kontekste - tai algoritmas arba vaizdo taškų transformacija atliekama su vaizdu, siekiant kuo labiau sumažinti triukšmo poveikį ir atkurti naudingą vaizdo dalį. Toliau poskyryje apžvelgiami šiame darbe naudojami rastrinių vaizdų filtrai. Kiekvienas iš šių filtrų turi vieną arba kelis keičiamus parametrus filtro pritaikymo laipsnius. Šiais parametrais (filtrų pritaikymo laipsniais) gali būti reguliuojamas filtro poveikis filtruojamam vaizdai [27].

1.2.1. Slenkstinis filtras

Šio filtro lygis atitinka tam tikrą slenkstinę vertę. Jei vaizdo taško spalvos reikšmė didesnė už šią vertę, taškas pakeičiamas į maksimalią reikšmę (baltą spalvą), jei mažesnė - į minimalią vertę (juodą spalvą). Kai filtruojamame vaizde spalvos reikšmės sunormalizuotos į intervalą $[0,1]$, kiekvienas vaizdo taškas transformuojamas naudojant Hevisaido funkciją:

$$y_{ij} = H(t - x_{ij}) \quad (1.5)$$

Čia: i, j - vaizdo taško eilės numeris, t - slenkstis, x - filtruojamas vaizdas. Keičiamas šio filtro parametras - slenkstis t . Šis filtras labai gerai pašalina kontrasto sumažinimo triukšmą ir binarizuoja vaizdą (išvestinis vaizdas turi tik 2 reikšmes - juodą ir baltą).

1.2.2. Gauso filtras

Tai filtras, kurio veikimas pagrįstas vaizdo konvoliucijos operacija su Gauso funkcija arba branduoliu (*angl.* kernel):

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1.6)$$

Šis filtras dažnių srityje dvimatę vaizdo funkciją apriboja Gauso funkcija. Vidutinis kvadratinis nuokrypis σ šiuo atveju apibūdina filtro taikymo laipsnį. Branduolio dydis ($x = y$) dydis šiame darbe naudojamoje filtro implementacijoje nustatomas pagal σ reikšmę t.y. lango dydis lygus vidutinį kvadratinį nuokrypį padauginus iš 4 ir suapvalinus iki sveiką skaičių. Toks šių verčių santykis yra efektyviausias tokiam filtrui. Šis filtras vaizduose nugludina objektų kraštus, panaikina labai mažus objektus (vieno pikselio dydžio), taigi gali būti naudojamas druskos ir pipirų bei Gauso triukšmo paveiktiems vaizdams filtruoti.[27]

1.2.3. Medianos filtras

Medianos filtras kiekvieno pikselio reikšmę pakeičia į aplinkinių pikselio reikšmių medianos reikšmę. Nustatomas filtro parametras - lango dydis, t.y. kiek aplinkinių pikselių yra naudojama centrinio pikselio spalvos reikšmės apskaičiavimui. Šis filtras naudojamas druskos ir pipirų bei Gauso triukšmo paveiktiems vaizdams filtruoti.[27]

1.2.4. Maksimumo ir minimumo filtrai

Tai filtrai labai panašūs į medianos filtrą. Skirtumas tik tas, kad į tam tikro pikselio vertę yra įrašoma aplinkinių pikselių minimali arba maksimali reikšmė. Nustatomas filtro parametras - taip pat lango dydis. Šis filtras naudojamas druskos ir pipirų bei Gauso triukšmams sumažinti.[27]

1.2.5. Wiener dekonvoliucijos filtravimo algoritmas

Wiener filtras buvo sukurtas pagerinti vaizdo ryškumą. Triukšmo paveiktas vaizdas dažnių srityje apibūdinamas:

$$X(u, v) = G(u, v)Y(u, v) \quad (1.7)$$

Čia: G - originalaus vaizdo dažnių spektras, Y - triukšmo dažnių spektras; u, v - y ir x koordinačių spalvos signalo dažniai. Siekiant gauti originalų vaizdą, skaičiuojama [27]:

$$G(u, v) = \frac{H(u, v) * P_s(u, v)}{|H(u, v)|^2 P_s(u, v) + P_n(u, v)} \quad (1.8)$$

Čia: $*$ - konvoliucijos operacija, P_s - vaizdo galios spektras, P_n - triukšmo galios spektras, H - vaizdo taško sklidimo funkcija (*angl.* point spreading function, arba PSF). Čia PSF funkcija - Gauso funkcija (Žr. formulę 1.6). Taigi šiam filtrui galima keisti 2 PSF parametrus - vidutinį kvadratinį nuokrypį σ bei branduolio dydį ($x = y$).

Šis filtras yra taikomas vaizdo ryškumo savybėms vaizde pagerinti.

1.2.6. Morfolginiai filtrai

Tai vaizdo apdorojimo filtrai pagrįsti matematine morfologija. Šie filtrai pagrįsti objekto struktūros išskyrimu bei šių objektų transformacija. Morfolginiai filtrai yra pagrįsti šiomis pagrindėmis operacijomis (nebinariniais vaizdams) [23]:

- praplėtimas (*angl.* dilation):

$$(f \oplus B)(x) = \max\{f(y) : y \in B_x\} \quad (1.9)$$

- erozija (*angl.* erosion)

$$(f \ominus B)(x) = \min\{f(y) : y \in B_x\} \quad (1.10)$$

Čia: f - pradinis vaizdas, sudarytas iš taškų x , B - struktūrinis elementas, B_x - struktūrinis elementas su centru taške x .

Šias operacijas taikant viena po kitos, gaunami morfolginiai filtrai. Atliekant erozijos operaciją, objektai yra sumažinami, o atliekant praplėtimo operaciją - padidinami. Taikant erozijos ir praplėtimo operacijas viena po kitos, gaunama morfolginio atidarymo operacija. Taikant šias operacijas atvirkščia tvarka (pirmiau praplėtimą, o po to eroziją), gaunama morfolginio uždarymo operacija. Šios operacijos supaprastina objektų kontūrus, pašalina smulkius objektų kontūrų defektus. [23]

Eksperimentiniu būdu nustatyta, kad šiame darbe naudojamiems vaizdams geriausiai triukšmus pašalina paeiliui su vaizdu vykdomos uždarymo ir atidarymo operacijos. Šios paeiliui vykdomos operacijos toliau vadinamos tiesiog morfolginio filtru. Toks morfolginis filtras sumažina druskos ir pipirų bei Gauso triukšmą. Taip pat pastebėta, kad OCR taikymo srityje morfolginiai filtrai veikia tuo geriau, kuo didesnė yra vaizdo raiška (rezoliucija).

1.2.7. Chambolle filtravimo algoritmas

Tai vienas iš visos variacijos (*angl.* total variation) filtravimo algoritmų atmainų. Visos variacijos filtravimo algoritmai pagrįsti tuo, kad dažniausia kuo didesnis triukšmas yra vaizde, tuo yra didesnė vaizdo visa variacija. Visa variacija dvimačiam diskrečiam signalui x , kuris turi elementus x_{ij} , apibrėžiama [30]:

$$V(x) = \sum_{i,j} \sqrt{|x_{i+1,j} - x_{i,j}|^2 + |x_{i,j+1} - x_{i,j}|^2} \quad (1.11)$$

Visos variacijos filtravimo algoritmai yra skirti sumažinti visą variaciją vaizde, taip sumažinant ir triukšmo lygį vaizde. Tokie algoritmai siekia kiek įmanoma sumažinti [16]:

$$\min(E(x, y) + \lambda V(y)) \quad (1.12)$$

Čia: E - L2 atstumas norminis, λ - regularizacijos parametras, nuo kurio priklauso, kaip stipriai variacija vaizde yra sumažinama, y - nufiltruotas vaizdas, x - originalus vaizdas.

Chambolle algoritmas šios minimizacijos siekia atlikdamas operaciją su vaizdu [13]:

$$p(i, j) = \frac{p(i, j) + \tau(\nabla(\operatorname{div} p - \lambda f)(i, j))}{1 + \tau|(\nabla(\operatorname{div} p - \lambda f)(i, j))|} \quad (1.13)$$

Ši operacija kartojama tol, kol minimali vertė pagal formulę 1.12 tampa mažesnė už dydį ϵ . Taigi nustatomi algoritmo parametrai - sustojimo parametras ϵ , ir filtro pritaikymo laipsnis λ . Visiškos variacijos filtrai pasižymi tuo, kad ne tik pašalina triukšmą vaizde, tačiau ir išsaugo objektų kontūrus. Chambolle filtras pirmiausia skirtas baltajam Gauso triukšmui pašalinti, tačiau pagerina ir vaizdo ryškumo savybes.

1.2.8. Nugludinimo kaukės filtras

Nugludinimo kaukės filtras (*angl.* unsharp mask) pagrįstas triukšmo „atėmimu“ iš vaizdo. Pirmiausia įvestinis vaizdas yra dirbtinai užtriukšminamas (sumažinamas vaizdo ryškumas) ir taikant tam tikrus svorius šis užtriukšmintas vaizdas panaudojamas nufiltruotam vaizdui gauti. Nufiltruoto vaizdo kiekvienas taškas y_{ij} gaunamas pagal:

$$y_{ij} = x_{ij} + t(x_{ij} - n_{ij}) \quad (1.14)$$

Čia: x_{ij} - pradinio vaizdo taškas, n_{ij} - užtriukšmintas vaizdo taškas, t - svorio koeficientas.

Šio filtro pritaikymo laipsnis - svorio koeficientas t . Šis filtras naudojamas vaizdo ryškumo savybėms pagerinti.

1.2.9. Gauso - Laplaso filtras

Laplaso operatorius yra naudojamas objektų kontūrams aptikti. Aptikus objektų kontūrus, juos galima naudoti toliau vaizdų filtravime. Laplaso operatorius, bendrąja prasme, yra funkcijos antros eilės išvestinė. Siekiant pagerinti šio operatoriaus veikimą ir atsparumą triukšmams, jis naudojamas kartu su Gauso funkcija. Taip gaunamas Gauso - Laplaso operatorius [23]:

$$LG(x, y) = \left(\frac{x^2 + y^2 - 2\sigma^2}{2\pi\sigma^6}\right) e^{-\left(\frac{x^2 + y^2}{2\sigma^2}\right)} \quad (1.15)$$

Nufiltruotas vaizdas gaunamas atlikus konvoliucijos operaciją su vaizdu. Nustatomi filtro parametrai - vidutinis kvadratinis nuokrypis σ bei branduolio dydis ($x=y$). Gauso-Laplaso operatorius šiame darbe naudojamas dviems filtrams. Pirmiausia jis naudojamas užtriukšmintą vaizdą modifikuoti (paryškinti išskirtą objektą). Antru atveju pavaizduojami objektai, kurio kontūrai yra nustatomi. Pastaruoju atveju gaunamas binarinis vaizdas. Abu variantai teigiamai veikia vaizdo ryškumo savybes.

1.2.10. Furje filtras

Furje filtrai - tai tokie filtrai, kurie veikia vaizdo dažnių srityje. T.y. filtravimo operacijos vykdomos prieš tai atlikus vaizdo Furje transformaciją. Tokie filtrai dažniausiai naudojami vaizdui nugludinti (pašalinami aukšti dažniai) arba pagerinti vaizdo ryškumą (sustiprinami aukšti dažniai).

Šiame darbe naudojamas Furje filtras gaunamas tokia veiksmų seka[23]:

- Su vaizdu atliekama diskrečioji Furje transformacija:

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^M \sum_{y=0}^N f(x, y) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad (1.16)$$

- Gaunamas aukštų dažnių filtras:

$$h(u, v) = \begin{cases} 0, & \text{jei } \sqrt{u^2 + v^2} \leq d \\ 1, & \text{jei } \sqrt{u^2 + v^2} > d \end{cases} \quad (1.17)$$

Čia: d - filtravimo slenkstis.

- Su gautu reikšmių masyvų h nufiltruojamas vaizdo dažninis atsakas F .
- Su gautu signalu atliekama atvirkštinė Furje transformacija:

$$f(x, y) = \frac{1}{MN} \sum_{x=0}^M \sum_{y=0}^N F(u, v) e^{j2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad (1.18)$$

- Prie filtruojamo vaizdo pridedamas gautas signalas, taip išryškinant aukšto dažnio sritis (objektų kontūrus).

Toks filtras išryškina aukšto dažnio sritis vaizde (pagerina vaizdo ryškumo savybes). Jo nustatomas parametras - slenkstis d .

1.3. Vaizdų kokybės metrika

1.3.1. MSE įvertis

MSE (*angl.* mean square error) arba vidutinės kvadratinės paklaidos įvertis apskaičiuoja skirtumą tarp dviejų vaizdų (pvz. užtriukšminto ir originalaus vaizdo). Šiame darbe laikoma, kad užtriukšminto vaizdo MSE įvertis yra MSE funkcija tarp užtriukšminto ir originalaus vaizdo. Identiškai apibūdinamas ir nufiltruoto vaizdo MSE įvertis. Taigi MSE įvertis apibūdina vaizdo kokybę ir triukšmo lygį vaizde. Jis apskaičiuojamas taip:

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (O(i, j) - K(i, j))^2 \quad (1.19)$$

Čia: M, N - vaizdo dydis; O,J - lyginami vaizdai

Jei reikšmės, apibūdinančios pikselio spalvą, yra sunormalizuotos į intervalą [0,1], tuomet ši funkcija taip pat gali įgyti reikšmes tik intervale [0,1]. Nulinė šios funkcijos reikšmė reiškia, kad du lyginami vaizdai yra identiški, 1 - vaizdai yra visiškai skirtingi. Pavyzdžiui, MSE funkcijos rezultatas tarp visiškai juodo ir visiškai balto vaizdų bus lygus 1. Šio darbo kontekste, jei vaizdo MSE įvertis yra lygus 0, tai vaizdas yra visiškai nepaveiktas triukšmo.[27]

1.3.2. OCR tikslumas

Tai dydis nusakantis kokia dalis simbolių yra nuskaityta teisingai. Tai dydis atvirkščiai proporcingas skirtumui tarp originalaus teksto vaizde, bei teksto, gauto atlikus OCR operaciją su vaizdu. Skirtumui tarp dviejų teksto įrašų yra naudojamas Levenšteino atstumas. Jis nusako kiek operacijų (pakeitimo, ištrynimo, įterpimo) reikia atlikti su pirmu iš lyginamų įrašų, kad būtų gautas antrąjį įrašą. Kadangi šis atstumas gali priklausyti nuo įrašų ilgių, OCR tikslumas yra santykinis:

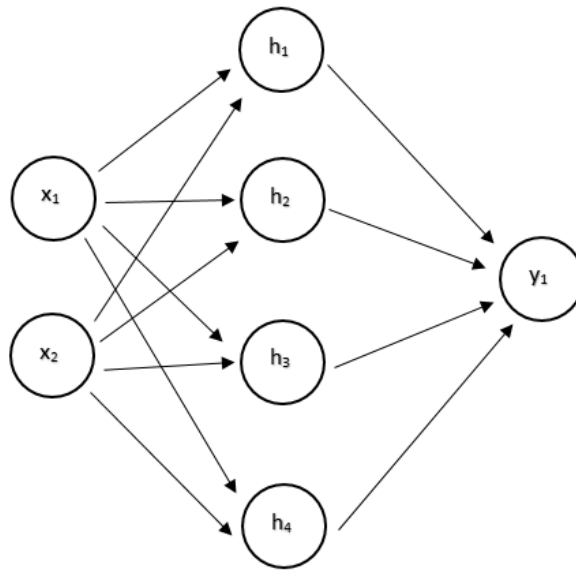
$$OCR = 1 - \frac{2(LCS(a, b))}{(L_a + L_b)} \quad (1.20)$$

Čia: LCS - Levenšteino atstumo skaičiavimo funkcija; a, b - lyginami įrašai; L_a, L_b - lyginamų įrašų ilgiai.

Šiame darbe vienas įrašas a - bus visada tikrasis tekstas vaizde, o įrašas b - tekstas gautas atlikus OCR operaciją su vertinamu vaizdu. Taigi vaizdo OCR tikslumas šiame darbe - dydis, apibūdinantis, kokia dalis simbolių yra atpažįstama teisingai atlikus OCR. Kai OCR lygus 1 - tekstas nuskaitomas visiškai teisingai, kai OCR lygus 0 - nuskaitytas tekstas ir originalus tekstas neturi bendrų simbolių. OCR tikslumo įvertis taip pat apibūdina OCR skirto vaizdo kokybę.

2. Dirbtiniai neuroniniai tinkai

Dirbtiniai neuroniniai tinkai (*angl.* artificial neural networks, toliau - DNT) - tai skaičiavimo metodas, naudojamas sudėtingų funkcijų aproksimacijai, spręsti klasifikavimo, grupavimo, klasterizavimo, sekų ieškojimo, atpažinimo, regresinius ir panašius uždavinius. Metodo veikimas pagrįstas smegenų veikla (biologiniais neuroniniais tinklais). Neuroninis tinklas sudarytas iš neuronų, kurie keičiasi informacija tarp vienas kito. DNT sudaromas iš kelių sluoksnių: įėjimo, vieno ar keleto paslėptųjų sluoksnių, bei išėjimo sluoksnių. DNT metodas pagrįstas "juodosios dėžės" principu. Įvedant duomenys DNT įvestyje, gaunami atitinkami reikalingi duomenys DNT išvestyje. Prieš taip naudojant DNT pirmiausia jis yra apmokomas, t.y. parenkami svoriai tarp neuronų ryšių. Paprasto neuroninio tinklo struktūra pavaizduota 5 pav.[28].



5 pav. DNT pavyzdys, kai įėjimo sluoksnį sudaro 2 neuronai, paslėptąjį - 4, o išėjimo - 1.

Neurono išėjimo vertę y nusakanti funkcija[28] :

$$y = \phi\left(\sum_{j=0}^n w_j x_j\right) \quad (2.1)$$

Čia: ϕ - aktyvacijos funkcija, n - įėjimų skaičius neurone, w - įėjimų svoriai.

DNT apmokymo metu yra parenkami svoriai w_j . Apmokymo su mokytoju atveju (*angl.* supervised learning) DNT yra apmokomas apmokymo duomenų rinkiniu, kurį sudaro įvesties duomenys (vektoriai) bei norimi gauti išvesties duomenys (vektoriai). DNT apmokomas iteraciniu būdu, koreguojant svorius tarp neuronų ryšių, mažinant kainos funkcija (*angl.* cost function). Tokių iteracijų skaičius yra vadinamas apmokymo epochų skaičiumi. Apmokymo iteracija kartojama tol, kol kainos funkcija tampa reikiamai maža arba pasiekiamas iš anksto nustatytas epochų skaičius.[28]

Egzistuoja ir apmokymo be mokytojo metodas, kai yra naudojami tik įvesties vektoriai, o išvesties vertės parenkamos kiek įmanoma labiau sumažinant kainos funkciją. [28]

MSE (vidutinė kvadratinė paklaida) yra dažniausiai naudojama kainos funkcija. Ji apibūdinama taip [28]:

$$MSE = \frac{1}{n} \sum_{i=1}^n (X_i - Y_i)^2 \quad (2.2)$$

Čia: X - apskaičiuota (*angl.* predicted) i -ojo vektoriaus vertė, Y - tikroji i -ojo vektoriaus vertė, n - vektorių skaičius.

Jei duomenys yra sunormalizuoti į intervalą $[0,1]$, tuomet ir ši funkcija gali įgyti reikšmes intervale $[0,1]$. DNT rezultatas įvertinti yra naudojama MSE reikšmė, gauta praleidus testinį duomenų rinkinį. Testinio duomenų rinkinio struktūra tokia pati kaip apmokymo duomenų rinkinio. Kuo ši MSE reikšmė mažesnė, tuo DNT skaičiavimo kokybė yra geresnė. Jei ši reikšmė pasiekia 0, laikoma kad DNT sistema pasiekė globalų minimumą. [28]

Kiekvienam skirtingam uždaviniui spręsti yra svarbu parinkti optimalią DNT struktūrą. Optimali struktūra priklauso nuo sprendžiamo uždavinio specifikos. Galima keisti paslėptųjų sluoksnių skaičių, bei neuronų kiekį juose, o neuronų kiekis išoriniuose sluoksniuose yra nulemtas naudojamų duomenų savybėmis.

2.1. Aktyvacijos funkcija

DNT realizacijoje svarbus parametras yra neurono aktyvacijos funkcija. Aktyvacijos funkcijos parinkimas priklauso nuo sprendžiamo uždavinio specifikos. Tai funkcija ϕ (2.1) kuri konvertuoja neurono įvestį į išvestį. Šiame darbe naudojamos aktyvacijos funkcijos [1]:

- Tiesinė funkcija:

$$\phi = xs \quad (2.3)$$

- Sigmoidinė funkcija:

$$\phi = \frac{1}{1 + e^{-2sx}} \quad (2.4)$$

Šiame darbe taip pat naudojama Sigmoidinė laiptuota (*angl.* stepwise) funkcija, kuri DNT apmokymo metu yra greitesnė už standartinę, tačiau jos tikslumas yra mažesnis.

- Gauso funkcija:

$$\phi = e^{-x^2 s^2} \quad (2.5)$$

Šiame darbe taip pat naudojama Gauso laiptuota funkcija, kuri DNT apmokymo metu yra greitesnė už standartinę, tačiau jos tikslumas yra mažesnis.

- Elioto funkcija:

$$\phi = \frac{\frac{xs}{2}}{1 + |xs|} + 0.5 \quad (2.6)$$

- Sinuso funkcija:

$$\phi = \frac{\sin(xs)}{2} + 0.5 \quad (2.7)$$

- Kosinuso funkcija:

$$\phi = \frac{\cos(xs)}{2} + 0.5 \quad (2.8)$$

Čia: x - neurono įvestis, s - statusas (atskirai nustatomas DNT parametras).

Pagal uždavinio specifiką ir kodavimą, naudojamos tik tokios funkcijos, kurių reikšmių sritis yra intervale $[0,1]$. Taigi visos šios aukščiau išvardintos funkcijos yra apribotos $0 \leq \phi \leq 1$.

2.2. Apmokymo algoritmai

2.2.1. Žingsninis klaidos sklidimo atgal algoritmas

Klaidos sklidimo atgal (*angl.* error backpropagation) algoritmas - tai dažniausiai naudojamas apmokymo su mokytoju algoritmas. Taikant šį algoritmą, pradžioje jungčių svoriai parenkami atsitiktinai. Vėliau apskaičiuojama gauto neuroninio tinklo kainos funkcija. Tuomet pakoreguojami neuroninio tinklo svoriai (dydžiu Δw_j) taip, kad kainos funkcija sumažėtų (pradedant nuo išvestinio sluoksnio). T.y. kiekvienam neuronui skaičiuojama[28]:

$$\Delta w_j = -\alpha \frac{\delta E}{\delta w_j} \quad (2.9)$$

Čia: α - mokymosi greitis (*angl.* learning rate).

Šioje formulėje mokymosi greitis gali būti ir nepastovus, kai naudojamas mokymosi inercijos parametras m (*angl.* learning momentum). Tokiu atveju mokymosi greitis yra padidinamas $1 + m$ karto, jei svorio koregavimo dydis turi tą patį ženklą negu prieš tai buvusioje apmokymo epochoje, arba mažinamas, jei svorio koregavimo dydis turi priešingą ženklą. Toks mokymosi greičio kitimas gali padėti išvengti sistemos konvergavimo į lokalų minimumą. Jei šis parametras pasirenkamas per didelis, tai DNT sistema gali "prašokti" globalų minimumą, tapti nestabilia ir neapsimokyti.

Žingsninis (*angl.* incremental) klaidos sklidimo atgal algoritmas pasižymi tuo, kad kiekvienas iš apmokymo duomenų yra apmokomas atskirai. T.y. visi DNT svoriai yra pakoreguojami pagal vieną apmokymo duomenų elementą. Vėliau procesas kartojamas su sekančiu duomenų elementu. Taigi per vieną epochą DNT svoriai pakoreguojami daug kartų. Po kiekvienos epochos yra įvertinama kainos funkcija. Jei ji nepasiekia reikiamos reikšmės, svoriai koreguojami iš naujo tokiu pat būdu.

2.2.2. Paketinis klaidos sklidimo atgal algoritmas

Paketinis (*angl.* batch) klaidos sklidimo atgal algoritmas - tai savo veikimo principu labai panašus algoritmas į žingsninį klaidos sklidimo atgal algoritmą. Vienintelis skirtumas yra tas, kad vienos epochos metu DNT jungčių svoriai yra pakoreguojami atsižvelgiant į visą apmokymo duomenų rinkinį. Taigi per vieną epochą DNT jungčių svoriai yra koreguojami tik vieną kartą.

2.2.3. QuickProp algoritmas

Tai yra modifikuota klaidos sklidimo atgal algoritmo versija. Šiam algoritmui nereikia parinkti mokymosi greičio bei mokymosi inercijos parametru, kadangi dydis, kuriuo pakoreguojamas jungties svoris, yra parenkamas atsižvelgus į prieš tai buvusios apmokymo iteracijos svorio koregavimo

dydį [18]:

$$\Delta w_j(k) = \frac{\frac{\delta E(k)}{\delta w(k)}}{\frac{\delta E(k-1)}{\delta w(k-1)} - \frac{\delta E(k)}{\delta w(k)}} \Delta w(k-1) \quad (2.10)$$

Čia: k - apmokymo epochos numeris.

Parinkti mokymosi greičio ir mokymosi inercijos parametrus kai kurioms sprendžiamoms problemoms gali būti sudėtingas uždavinys, kuriam spręsti yra keletas netrivialių metodų [35]. Taigi šis algoritmas skirtas būtent tokiems atvejams. Taip pat daugeliu atveju šis algoritmas yra spartesnis nei standartinis klaidos sklidimo atgal algoritmas. Šis algoritmas veikia paketiniu apmokymo principu. [18]

2.2.4. RProp algoritmas

Šis algoritmas panašus į QuickProp algoritmą, kadangi jame taip pat nereikia nustatyti mokymosi greičio parametro. Svorio kooregavimo dydis gaunamas pagal [29]:

$$\Delta w_j(k) = \begin{cases} \eta^+ \Delta w_j(k-1), & \text{jei } \frac{\delta E(k-1)}{\delta w_j(k-1)} \frac{\delta E(k)}{\delta w_j(k)} > 0 \\ \eta^- \Delta w_j(k-1), & \text{jei } \frac{\delta E(k-1)}{\delta w_j(k-1)} \frac{\delta E(k)}{\delta w_j(k)} < 0 \\ \Delta w_j(k-1), & \text{kitais atvejais} \end{cases} \quad (2.11)$$

Parametrai parenkami taip, kad $0 < \eta^- < 1 < \eta^+$.

Šis algoritmas laikomas vienu greičiausiu DNT apmokymo algoritmu. Jis taip pat paremtas paketiniu apmokymo principu.

2.2.5. SARProp algoritmas

Tai patobulintas RProp algoritmo variantas. Čia parametrai η^- ir η^+ yra pokereguojami atsitiktinai pridėjus triukšmą (temperatūrą). Toks šių parametru koregavimas sumažina tikimybę DNT sistemai konverguoti į kainos funkcijos lokalaus minimumo būseną. Taip pat dydis $\frac{\delta E(k)}{\delta w_j(k)}$ formulėje 2.11 yra pakeičiamas į:

$$\frac{\delta E(k)}{\delta w_j(k)} = \frac{\delta E(k)}{\delta w_j(k)} - \frac{0.01 w_j 2^{-Tk}}{1 + w_j^2} \quad (2.12)$$

Čia: T - temperatūra (triukšmas).

Toks pakeitimas padeda sumažinti svorio koregavimo dydį didėjant epochų skaičiui ir leidžia DNT sistemai kuo arčiau priartėti prie globalaus minimumo.[34]

3. Genetiniai algoritmai

Genetinis algoritmas - tai skaičiavimų modelis, paremtas evoliucijos teorija. Jis imituoja evoliucijos procesus: paveldėjimą, mutaciją, natūraliąją atranką ir rekombinaciją. Genetiniai algoritmai dažniausiai naudojami sudėtingiems optimizavimo ir paieškos uždaviniams spręsti. Genetinis algoritmas labai tinkamas, kai sprendinys priklauso nuo labai daug faktorių, tarp kurių ryšiai yra sudėtingi arba nežinomi.[22]

Genetinis algoritmas inicializuojamas atsitiktinai sugeneravus tam tikrą kiekį individų. Šis kiekis yra vadinamas populiacijos dydžiu. Individas - tai tam tikru būdu užkoduotas nagrinėjamo uždavinio sprendinys (chromosoma). Pavyzdžiui, tai gali būti tam tikromis reikšmėmis apribotas apibrėžto ilgio sveikųjų skaičių masyvas. Populiacijos individams, pagal populiacijos atrankos metodą, yra apskaičiuojama sprendinio tinkamumo funkcija (*angl.* fitness function), kuri parenkama pagal uždavinio specifiką. Ją galima siekti didinti arba mažinti - tai vėlgi priklauso nuo sprendžiamo uždavinio. Iš pirmosios populiacijos yra suformuojama nauja individų populiacija (epocha arba karta). Naujoji populiacija gali būti suformuojama kelių būdų kombinacija: mutavimu (kai dalis chromosomų yra atsitiktinai pakeičiama pagal tam tikrą taisyklę), rekombinacija (kai yra sukryžminamos kelios chromosomos), elitizmu (dalis geriausių individų patenka į kitą populiaciją nepakeisti). Genetinis algoritmas nutraukiamas kai iki reikiamo dydžio sumažinama (arba padidinama) sprendinio tinkamumo funkcija arba kai pasiekiamas iš anksto numatytas epochų skaičius.[22]

Pagrindinė GA problema yra ankstyvas ir lokalus sprendinio konvergavimas. Norint to išvengti yra siekiama palaikyti genotipo įvairovę. To dažniausiai siekiama taikant dideles mutacijos ir kryžminimo tikimybes. Tačiau optimalūs parametrai yra skirtingi daugeliui uždavinių ir priklauso nuo tiriamos srities specifikos.

Pagrindiniai genetinio algoritmo parametrai yra tokie [19] [21]:

- Populiacijos dydis - tai individų skaičius kiekvienoje populiacijoje;
- Kryžminimo tikimybė - tikimybė sukryžminti 2 individus;
- Mutacijos tikimybė - tikimybė mutuoti individui;
- Individuali mutavimo tikimybė - tai nepriklausoma tikimybė mutuoti atskiram vienam individo elementui;
- Sprendinio tinkamumo funkcija - funkcija, pagal kurią vertinamas kiekvieno individo tinkamumas;
- Epochų skaičius - iteracijų skaičius, nusakantis kiek kartų yra vykdomas naujos populiacijos formavimas.

3.1. Atrankos metodai

Yra žinomi keli individų atrankos į sekančią populiaciją būdai[21] [19]:

- **Ruletinė atranka** - tai atrankos būdas, kai tikimybė individui patekti į kitą populiaciją yra

proporcinga to individo sprendinio tinkamumo funkcijos reikšmės ir visų individų sprendinio tinkamumo funkcijos reikšmių sumos santykiui:

$$p(i) = \frac{f(i)}{\sum_{j=1}^N f(j)} \quad (3.1)$$

Čia: p - tikimybė individui būti atrinktam į kitą populiaciją, i - individo eilės numeris, N - populiacijos dydis, f - sprendinio tinkamumo funkcija.

- **Turnyrinė atranka** - tai atrankos būdas, kai atsitiktinai pasirenkami individai (jų kiekis vadinamas turnyro dydžiu) ir iš jų atrenkamas vienas, kurio sprendinio tinkamumo funkcijos reikšmė yra geriausia. Procesas kartojamas tol, kol gaunamas reikiamas individų skaičius.
- **Blogiausiųjų atranka** - atrankos metodas, kai atrenkami patys blogiausieji individai pagal sprendinio tinkamumo funkciją iš visos populiacijos ir perduodami į kitą populiaciją. Šis atrankos būdas gali būti naudingas tada, kai uždaviniui sprendiniai linkę anksti konverguoti į lokalų minimumą ir optimizuojami parametrai nekoreliuoja.
- **Geriausiųjų atranka** - atrankos metodas, kai atrenkami patys geriausieji individai pagal sprendinio tinkamumo funkciją iš visos populiacijos ir perduodami į kitą populiaciją. Naudojant šį atrankos būdą, didesnė tikimybė sprendiniui konverguoti į lokalų minimumą, tačiau neprarandami geriausieji individai, gauti pirmose genetinio algoritmo epochose.
- **Dvigubo turnyro atranka** - tai atrankos būdas panašus į turnyro atranką. Vienintelis skirtumas, kad naudojami du turnyrai - pirmasis turnyras naudojamas atrinkti kandidatus į antrąjį turnyrą. Be abejo, tokia turnyrinė atranka reikalauja daugiau skaičiavimo resursų.
- **Atsitiktinė atranka** - tai atrankos būdas, kai į kitą populiaciją perkeliama tiesiog atsitiktinai parinkti individai. Šis atrankos būdas naudoja mažiausiai skaičiavimo resursų, tačiau didelė tikimybė, kad potencialiai geriausieji individai nėra perkeliama į kitą populiaciją.

3.2. Kryžminimo metodai

Kryžminimo metodas - tai būdas iš 2 parinktų individų (tėvų), gauti naujus individus (vaikus). Vaikų chromosomas sudaro iš tėvų paveldėtos chromosomų dalys. Šiame darbe toliau naudojami tokie kryžminimo metodai [19] [21]:

- **Vieno taško metodas** - tai kryžminimo metodas, kai 2 tėvų individai yra išskaidomi atsitiktinėje chromosomos vietoje į 2 dalis, ir naujas vaiko individas gauna po vieną dalį iš kiekvieno individo.
- **Dviejų taškų metodas** - tai kryžminimo metodas, kai 2 individai yra kryžminami dviejuose taškuose. Atsitiktinai parenkami 2 taškai, per kuriuos individas yra išskaidomas į tris dalis. Vaiko individas šiuo atveju turi 2 dalis iš vieno tėvo individo, ir 1 dalį iš kito.
- **Dviejų taškų surikiuojantis metodas** - tai kryžminimo metodas panašus į dviejų taškų kryžminimo metodą. Skirtumas tik tas, kad gautos dalys iš abiejų tėvų sudedamos į vaikus didėjimo tvarka. Tokiu būdu susidaręs vaiko individas gali turėti ir tik vieno tėvo individo

chromosomos dalis. Kadangi tėvų chromosomos yra skaidomos atsitiktinėse vietose, paveldimo chromosomų ilgiai nėra vienodo ilgio, todėl dažniausiai paveldėtos chromosomų dalys yra karpomos.

- **Dalinai sutampančiųjų metodas** - tai kryžminimo metodas, kai iš dviejų individų gaunami taip pat du individai. Abiejuose tėvų individuose atsitiktinai parenkama dalis, kuri perduodama abiem vaikų individams. Likusios nepanaudotos chromosomų dalys yra sukeičiamos tarp individų. Taip gaunami 2 individai, kurių chromosomos yra dalinai sutampančios.

4. Susiję darbai

4.1. DNT ir GA taikymas vaizdų filtravime

Mokslinėje literatūroje galima rasti nemažai aprašytų metodų, naudojančių genetinius algoritmus vaizdų filtravimui. Genetinis algoritmas pirmiausia vaizdų filtravime gali būti panaudojamas, kuriant naujus sudėtinius filtrus iš paprastesnių filtrų. Būtent taip filtravimo problema buvo spęsta darbe [6]. Šiame darbe pirmiausia buvo užtriukšminami vaizdai, kuriuose, naudojant 19 įvairių filtrų ir vaizdo keitimo metodų, buvo bandoma pašalinti triukšmą. Genetinių algoritmų pagalba, naudojant užtriukšmintus ir originalius vaizdus, buvo parenkami įvairūs filtrų parametrai, vaizdų transformacijos metodai bei jų veikimo laipsniai. Geriausius rezultatus parodžiusios parametru kombinacijos buvo naudojamos naujų vaizdų filtravimui. Anot autorių, rezultatai buvo labai geri. Šio metodo trūkumas – iš anksto žinomi triukšmų tipai ir jų veikimo stiprumas. Taip pat šiame darbe nėra nagrinėjama efektyvumo ir skaičiavimo laiko problema.

Panašiai problema sprendžiama ir darbe [15], kuriami genetiniu algoritmu parenkami trijų filtrų parametrai, veikimo sekos bei veikimo laipsniai. Čia egzistuoja ta pati problema - iš anksto žinomi triukšmų tipai ir jų veikimo stiprumas. Pastarajame darbe taip pat pažymima, kad filtravimas genetiniu algoritmu yra žymiai daugiau skaičiavimo resursų reikalaujantis metodas, nei kiti įprasti filtravimo metodai.

Kitas darbas [11] nagrinėja problemą panašiai kaip ir pastarieji darbai. Tačiau pagrindinis skirtumas yra toks, kad čia didelis dėmesys kreipiamas į efektyvumą ir skaičiavimo laiko optimizavimą. Tam naudojami lygiagretūs paskirstytieji genetiniai algoritmai, kurie iš dalies sprendžia efektyvumo skaičiavimo resursų atžvilgiu problemą, taip pat kai kuriais atvejais yra tikslesni, patikimesni bei lankstesni. Čia naudojamas hibridinis filtras sudarytas iš dviejų netiesinių filtrų: medianinio filtro bei variacijos metodo. Anot autorių, jiems pavyko pasiekti labai gerą filtravimo kokybę ir skaičiavimo laiko santykį. Tačiau šiame darbe buvo nagrinėjamas tik vieno tipo, druskos ir pipirų, triukšmas. Kita vertus triukšmo lygis nebuvo iš anksto žinomas.

Išsamus genetinių algoritmų tyrimas atliekamas darbe [33]. Čia naudojami 3 vaizdo transformacijos operacijos. Triukšmo tipas yra naudojamas tas pats, tačiau triukšmo dispersija yra nustatoma pagal *CImg* išorinės bibliotekos algoritmą. Ši triukšmo dispersija ir yra naudojama sprendinio tinkamumo funkcijoje. Didelis dėmesys skiriamas pačio genetinio algoritmo optimizavimui bei optimalių reikšmių parinkimui (populiacijos dydis, epochų skaičius ir pan). Gauti geriausi rezultatai palyginami su populiariais vaizdų triukšmo šalinimo metodais bei filtrais, naudojant PSNR (didžiausio triukšmo ir signalo santykio metodą).

Darbe [26] triukšmai nespaltuose vaizduose buvo šalinami naudojant tik pikselio spalvos transformaciją. Čia buvo tiriamas MRI vaizduose gan sunkiai pašalinamas Rician triukšmas, kuris pasireiškia srityse, kuriose yra mažas signalo ir triukšmo santykis. Anot autorių, tyrimas parodo, kad genetiniai algoritmai gali pašalinti ypač stiprų triukšmą. Šiame darbe naudojamame metode būtinas originalus vaizdas be triukšmų.

Moksliniame straipsnyje [25] buvo nagrinėjamas genetiniu algoritmų taikymas vaizdų filtravime OCR srityje. Dalis vaizdų iš ICDAR 2003 tekstinių vaizdų duomenų bazės buvo suklasifikuoti į kategorijas pagal vaizdo degradaciją. Ši klasifikacija buvo vykdoma vizualiai rankiniu būdu. Kiekvienai vaizdo degradacijos klasei buvo parenkama optimali filtrų kombinacija. Šis optimalus filtras buvo panaudojamas tos pačios degradacijos klasės filtravimui. Anot autorių, rezultatai pakankamai geri ir įrodo, kad tas pats filtras gali nufiltruoti vaizdus su ta pačia vaizdo degradacija.

Dirbtiniai neuroniniai tinklai vaizdų apdorojime gali būti naudojami įvairiais būdais [17]: vaiz-

dų klasifikacijai, objektų atpažinimui, savybių išskirimui, segmentacijai ir netgi pačiam vaizdo filtravimui. Straipsnyje [10] buvo naudojamas daugelio sluoksnių dirbtinis neuroninis tinklas, pasitelkiant didelius skaičiavimų resursus bei labai didelę (362 mln rastrinių vaizdų) aibę apmokymo duomenų (originalių ir užtriukšmintų rastrinių vaizdų). Nors ir buvo naudojami keli triukšmų tipai (juostinis triukšmas, baltasis triukšmas, JPEG artefaktai), triukšmo lygis visur buvo naudojamas tas pats. Rezultatai rodo, kad naudojant vien neuroninių tinklų algoritmą, rezultatai prilygsta vienam iš geriausių laikomam BD3M filtravimo algoritmui. Tačiau akivaizdus šio metodo trūkumas – reikalingi labai dideli skaičiavimo resursai bei didelis apmokymo duomenų kiekis.

Egzistuoja ir keletas darbų, aprašančių dirbtinių neuroninių tinklų naudojimą vaizdų klasifikacijai. Moksliniame straipsnyje [20] neuroninis tinklas naudojamas vaizdams klasifikuoti pagal tam tikrus objektų šablonus. Panašus metodas buvo naudojamas kitame moksliniame straipsnyje [14], kuriame vaizdai klasifikuojami tolesniam objektų atpažinimui palengvinti. Abiejų šių darbų rezultatai rodo, kad dirbtiniai neuroniniai tinklai puikiai atlieka savo darbą šioms problemoms spręsti. Darbe [12] buvo naudojamos palydovinės nuotraukos regionų klasifikavimui į kelius, urbanistinius regionus, laukus, miškus ir pan. Anot autorių, rezultatai labai geri ir tai patvirtina, kad DNT galima naudoti vaizdų klasifikavimo uždaviniams spręsti.

Taigi dirbtiniai neuroniniai tinklai yra plačiai taikomi duomenų filtravimo srityje, tačiau jų naudojimas triukšmų vaizduose nustatymui mokslinėje literatūroje nėra išnagrinėtas.

4.2. Automatinio filtravimo OCR metodai

Automatinio filtro, skirto pagerinti vaizdo kokybę būtent OCR sričiai, idėja taip pat nėra nauja. Darbe [36] automatiniam filtravimui yra naudojami DNT klasifikavimo algoritmas bei visų porų (*angl.* all pairs classifier) klasifikatorius. Šie klasifikavimo algoritmai skirti geriausiam filtrui parinkti pagal nustatytus vaizdo parametrus. Naudojami trys filtrai: Otsu slenkstinis filtras, Niblack slenkstinis filtras bei medianos filtras. Filtravimo procesas pradedamas vaizdo parametrų nustatymu. Vėliau pagal šiuos rezultatus, naudojant klasifikavimo algoritmus, parenkamas ir pritaikomas optimalus filtras. Klasifikatorius apmokomas įvertinus filtravimo rezultatus su visais įmanomais filtrais. Nors ir šis metodas rodo teigiamus rezultatus, pagrindinis šio metodo trūkumas anot autorių, yra tai, kad pritaikomas tik vienas filtras, taigi tokia implementuota sistema negali pašalinti stiprių triukšmų.

Moksliniame darbe [32] buvo nagrinėjamas automatinis kontrasto pagerinimo algoritmas, kuris atsižvelgia į objekto ir fono spalvų reikšmes, randa optimalų slenkstį, ir pagal jį padidina vaizdo kontrastą. Nors ir rezultatai, anot autoriaus, labai geri, ir metodas veikia su spalvotais vaizdais, tačiau šis metodas gali paveikti tik vieno tipo triukšmą.

Moksliniame straipsnyje [31] aprašytame tyrime pirmiausia vaizde yra automatiškai nustatomi vaizdo kokybės parametrai: šrifto dydis, raidžių storio rodiklis, susilieusių raidžių rodiklis, mažų dėmių rodiklis, didelių dėmių rodiklis, baltų dėmių rodiklis, nevientisų objektų rodiklis. Pagal šiuos gautus parametrus, algoritmas, paremtas sprendimų medžio veikimo principu, parenka optimalų filtrą. Filtrai sudaryti rankiniu būdu, išanalizavus testinį duomenų rinkinį. Naudojami tik morfologiniai filtrai. Rezultatai rodo, kad toks automatinis filtras pagerina OCR rezultatus. Didžiausias šio metodo trūkumas yra tai, kad taisyklės, kuriomis paremtas filtro nustatymas, yra rankinis, todėl esant sudėtingesniems, įvairesniems ir intensyvesniems triukšmams, toks būdas tampa nepraktišku.

Darbe [24] automatinio filtravimo užduočiai naudojami SVM (*angl.* support vector machine) klasifikavimo algoritmai. Čia filtravimo uždavinys skaidomas į 2 dalis: filtravimą ir gauto filtruoto

vaizdo binarizavimą. Tam pritaikomos atskiros SVM implementacijos bei kiekvienai daliai yra naudojami atskiri algoritmai. Filtravimo fazėje naudojami morfologiniai, Furje, medianos, vidurkio su svoriais filtrai, o binarizavimo - Shannon, Otsu ir Niblack slenkstiniai filtrai. Panašiai, kaip ir pastarajame straipsnyje, pirmiausia yra gaunami tam tikri vaizdo parametrai. Įdomu tai, kad filtravimo fazėje vaizdo parametrai yra gaunami dažninėje vaizdo charakteristikoje (atlikus Furje transformaciją). Pagal šiuos parametrus, naudojant klasifikatorių, gaunamas optimalus filtras. Klasifikatorius apmokomas išbandžius visus įmanomus filtrus. Nors ir metodo testavimo rezultatai nebuvo labai geri, autorius šį metodą tikisi plėtoti toliau.

5. Metodas ir jo realizavimas

5.1. Filtravimo metodo apžvalga

Šio darbo tikslui įgyvendinti naudojamas automatinis filtravimo metodas, kuriam nereikalingas originalus vaizdas (vaizdas be triukšmų). Originalūs vaizdai reikalingi tik neuroninio tinklo apmokymui bei optimalių filtrų radimui naudojant genetinį algoritmą. Šiems atvejams naudoti sintetiniai užtriukšminti vaizdai. Pagal šį metodą, filtravimas turi tokius žingsnius:

1. Naudojant apmokytą neuroninį tinklą, įvertinti triukšmą vaizde.
2. Vaizdui pritaikyti šiam triukšmo tipui bei lygiui skirtą optimalų sudėtinį filtrą, kuris gautas naudojant genetinį algoritmą

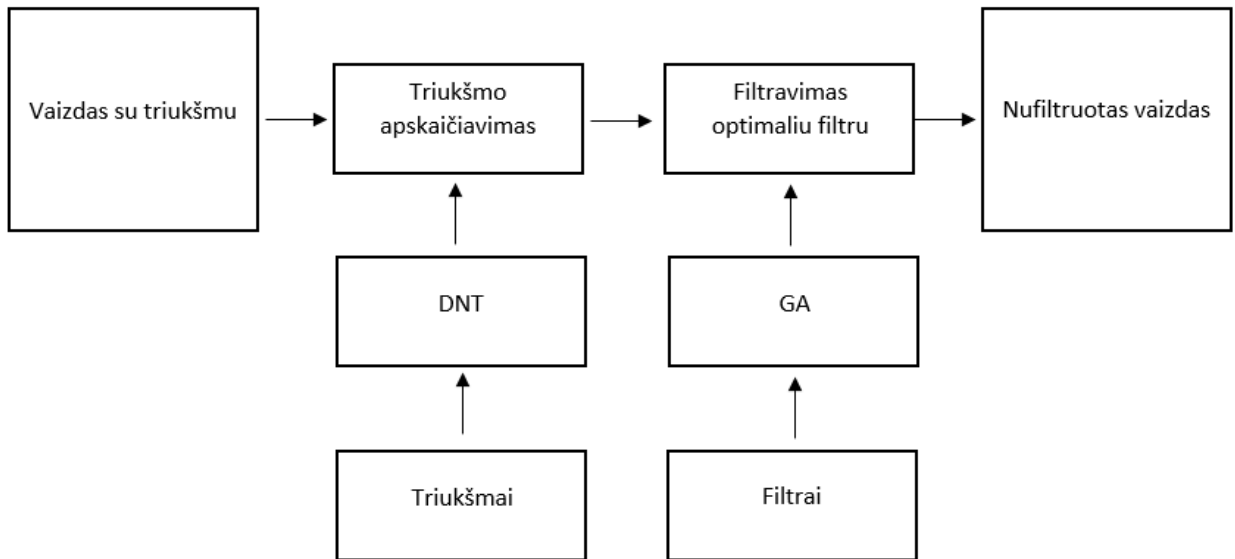
Optimalūs sudėtiniai filtrai šiuo atveju yra sukonstruoti iš kelių paprastesnių (elementariųjų) filtrų, kurie nurodyti 1.2 skyriuje. Optimalus sudėtinis filtras sudaromas derinant įvairius elementariųjų filtrų taikymo laipsnius, jų parametrus bei jų taikymo sekas.

Norint įgyvendinti šį metodą reikalingi tokie žingsniai:

1. Implementuoti 1.1 skyriuje įvardintus triukšmus;
2. Suformuoti vaizdus kurie bus naudojami eksperimente;
3. Realizuoti DNT algoritmą;
4. Nustatyti optimalius DNT parametrus;
5. Apmokyti DNT sintetiniais duomenimis;
6. Implementuoti 1.2 skyriuje įvardintus filtrus;
7. Realizuoti GA;
8. Nustatyti optimalius GA parametrus;
9. Naudojant GA, gauti optimalius filtrų parametrus kiekvienam triukšmo lygiui (sudaryti optimalių sudėtinųjų filtrų parametrų bazę);

5.2. Metodo realizavimas

Filtravimo metodas realizuotas naudojant *Python* programavimo kalba. Ši kalba buvo pasirinkta dėl to, kad jos pagrindu egzistuoja daug pagalbinių bibliotekų ir paketų, kurie palengvina nagrinėjamo metodo realizaciją. Vaizdų apdorojimui ir sukūrimui buvo naudotos *numpy*, *Scipy*, *PIL* išorinės bibliotekos. Kai kurie filtrai implementuoti su paketo *scikit-image* ir *Scipy* bibliotekų pagalba. Grafikų gavimui bei vaizdų vizualizavimui naudota *matplotlib* biblioteka. Automatinio klasifikavimo pagal triukšmus algoritmui buvo naudojamas *Fast Artificial Neural Network Library 2 (FANN2)* paketas. Kadangi *FANN* biblioteka parašyta *C* kalba, tai buvo reikalingas *fann2* jungiamasis paketas su *Python*. Kitiems klasifikavimo ir regresiniams algoritmams patikrinti bei duomenų išsaugojimui atmintyje buvo naudojama *scikit-learn* biblioteka. Genetiniam algoritmui



6 pav. Automatinio filtravimo metodo schema.

implementuoti buvo pritaikomas *DEAP evolutionary tool* paketas [19]. Optiniam simbolių atpažinimui iš vaizdų (OCR) buvo naudojama *tesseract* simbolių atpažinimo sistema bei *pytesseract* jungiamasis paketas. Kelioms funkcijoms aprašyti buvo pasitelkta internete atvirai prieinamų šaltinių pagalba [4] [2] [3] [5].

Šiame darbe neuroninio tinklo apmokymui, optimalių sudėtinių filtrų parinkimui genetiniu algoritmu ir dalyje testavimo buvo naudojami sintetiniai tekstiniai vaizdai su atsitiktinai sugeneruotais žodžiais iš anglų kalbos žodyno. Kiekviename vaizde yra 36 žodžiai (18 eilučių po 2 žodžius). Kiekvienas gautas vaizdas yra 200x200 taškų dydžio.

Triukšmų poveikio laipsniai buvo sunormalizuoti į intervalą [0,1]. Reikšmė 0 kiekvienu triukšmo atveju reiškia, kad vaizdas nėra paveiktas triukšmu. Maksimali triukšmo poveikio vertė nustatyta vizualiai. Tai maksimali vertė kai dar galima išvelgti objektus užtriukšmintame vaizde. Maksimalias triukšmų vertes atitinkantys parametrai pavaizduoti 1 lentelėje.

1 lentelė. Maksimalų naudojamą triukšmo lygį atitinkantys triukšmų parametrai.

Triukšmo tipas	Parametras	Maksimali reikšmė
Kontrasto sumažinimas	L (Žr. formulę 1.3)	0.95
Druskos ir pipirų	s (Žr. formulę 1.2)	0.2
Ryškumo sumažinimas	σ (Žr. formulę 1.4)	1.8
Gauso	σ (Žr. formulę 1.1)	6000

Kiekvienam triukšmui toliau naudojama 6 galimi triukšmo lygiai.

Panaši procedūra atliekama su filtrų parametrais (arba filtrų pritaikymo laipsniais). Jie taip pat sunormalizuojami į intervalą [0,1], kur 0 reiškia, kad filtras nėra pritaikomas, o 1 - filtro parametras yra maksimalus. Šių parametru vertės gaunamos eksperimentiniu būdu, vertinant maksimaliai triukšmo paveiktų vaizdų filtravimą. Šios vertės nurodytos 2 lentelėje.

2 lentelė. Maksimalūs naudojami filtrų parametrai.

Filtras	Parametras	Maksimali reikšmė
Slenkstinis	Slenkstis t (Žr. formulę 1.5)	1
Gauso	σ (Žr. formulę 1.6)	1.8
Medianos	Lango dydis	3
Maksimumo	Lango dydis	3
Minimumo	Lango dydis	3
Wiener	PSF brand. dydis (Žr. formules 1.7, 1.6)	8
Wiener	PSF σ (Žr. formules 1.7, 1.6)	1.8
Chambolle	Sustojimo parametras ϵ	1.0
Chambolle	λ (Žr. formulę 1.13)	0.25
Nugludavimo kaukės	t (Žr. formulę 1.14)	0.25
Gauso-Laplaso	σ (Žr. formulę 1.15)	1.0
Gauso-Laplaso	Branduolio dydis (Žr. formulę 1.15)	8
Furje filtras	Slenkstis d (Žr. formulę 1.17)	10

Morfologinis filtras neturi nustatomų parametru, todėl šioje filtro implementacijoje parametras tiesiog nurodo, ar filtras yra pritaikomas (su 50 % tikimybe).

Kiekvienam filtro parametru toliau naudojama 6 skirtingi lygiai.

5.3. Paskirstytasis skaičiavimas

Operacijos su vaizdais yra labai reiklios skaičiavimo resursams, kadangi operacijos yra atliekamos su didele slankaus kablelio skaičių matrica. Šio darbo kontekste tai ypač pasireiškia taikant genetinį algoritmą. Kadangi filtravimo operacijas, kurios daugeliu atveju nėra trivios, reikia atlikti keletą kartų daugeliui GA populiacijos individų, resursų poreikis ypač išauga. Nors ir GA nenaudojamas visas vaizdas, o tik jo dalis (langas), atliekant šiame darbe aprašytą tyrimą, nustatyta, kad optimalių sudėtinių filtrų bazės sudarymas neįmanomas su standartiniais nuosekliais programavimo metodais. Šiuo tikslu, algoritmui buvo pritaikomi paskirstytojo skaičiavimo metodai. Paskirstytasis skaičiavimas - tai skaičiavimo metodas, kai skaičiavimo uždavinys tam tikru būdu yra paskirstomas tarp skirtingų skaičiavimo mazgų, siekiant pagerinti skaičiavimo efektyvumą.

Paskirstytasis skaičiavimas buvo įgyvendinamas panaudojant MPI (*message parsing interface* protokolą. Tai protokolas aprašantis duomenų apsikeitimą tarp kelių procesų. Skirtingi procesai gali būti atliekami skirtinguose kompiuteriuose, jų procesoriuose ir pastarųjų branduoliuose. Komunikaciją tarp jų atlieka komutatoriai. Galimos dvi komunikacijos rūšys: taškinė (*angl.* point to point) ir kolektyvinė (*angl.* collective). Pirmuoju atveju komunikacija aprašoma tarp visų skaičiavimo mazgų atskirai, antruoju - komunikacija atliekama automatiškai, atskiriant pagrindinį ir šalutinius skaičiavimo mazgus. Šiame darbe buvo pritaikomas pastarasis komunikacijos būdas, kadangi jo veikimas yra greitesnis, o realizacija yra gerokai paprastesnė, ypač kai skaičiavimui naudojamų mazgų skaičius nėra iš anksto žinomas. MPI protokolui realizuoti buvo naudojamas *mpi4py* paketas.

Daugiausiai resursų reikalaujančios užduotys (pvz. optimalių sudėtinių filtrų parametru bazės sudarymas, optimalių GA parametru nustatymas) buvo atliekamos Vilniaus Universiteto, Matematikos ir informatikos fakulteto, Skaitmeninių tyrimų ir skaičiavimų centro superkompiuteryje. Reikalingų *Python* paketų įdiegimas superkompiuteryje pasiektas pasitelkiant virtualios aplinkos

paketo *virtualenv* pagalbą.

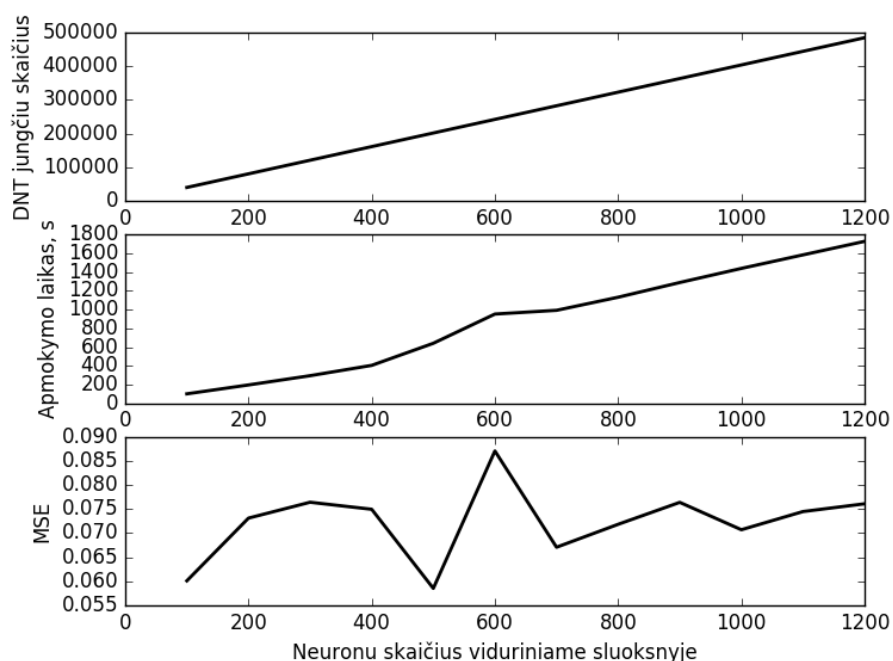
6. Tyrimas ir jo rezultatų analizė

6.1. Triukšmų nustatymas vaizde naudojant DNT

DNT panaudojimo tikslas - gauti triukšmo lygį įvestiniame vaizde. Tam reikalingas DNT apmokymas vaizdais, kurių triukšmo lygis yra žinomas. Apmokymui yra naudojami užtriukšminti atsitiktinio teksto vaizdai. Vienam vaizdui užtriukšminti yra naudojami keturi anksčiau įvardinti triukšmai, atsitiktinai parenkant jų lygį. Viso apmokymui suformuota 2000 tokių skirtingų vaizdų. Tai maksimalus kiekis duomenų, kai implementuotas DNT algoritmas gali sėkmingai būti apmokytas.

6.1.1. Optimalių DNT parametrų nustatymas

Tyrimo metu buvo naudojama 3-jų sluoksnių DNT struktūra. Kadangi DNT įėjime - vaizdo pikselio spalvos reikšmės, išėjime - kiekvieno iš triukšmų lygis, tai pirmajame sluoksnyje neuronų skaičių nulemia įvestinio vaizdo dydis, o trečiajame - triukšmų kiekis (šiuo atveju 4). Kadangi antrame sluoksnyje neuronų skaičius gali būti nustatomas laisvai, buvo tiriama DNT MSE parametro (kaip DNT apmokymo kokybės įverčio) priklausomybė nuo neuronų skaičiaus viduriniame sluoksnyje. Rezultatai pavaizduoti 7 pav.

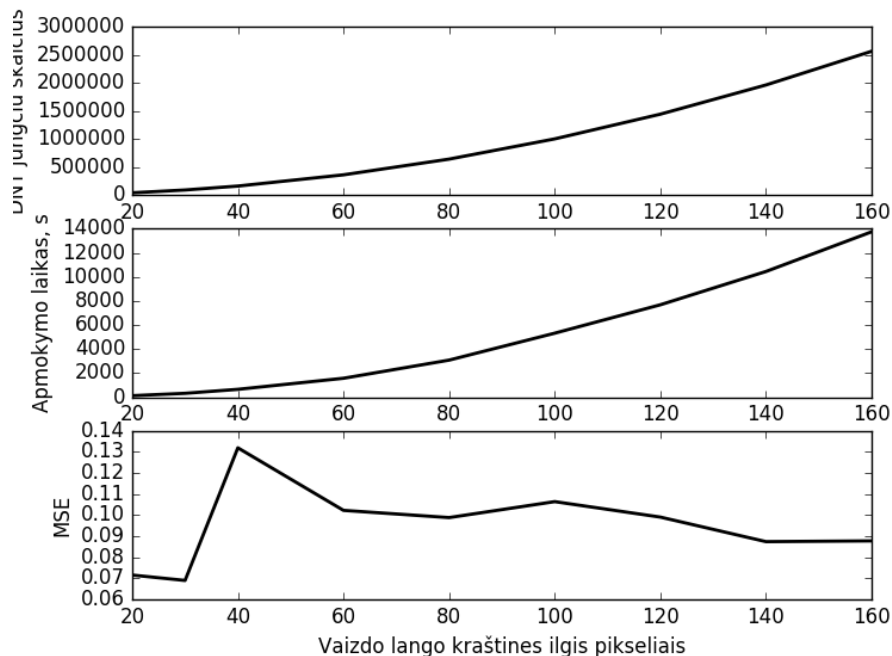


7 pav. Jungčių skaičiaus tarp neuronų (viršuje), apmokymo laiko (viduryje) bei MSE po DNT apmokymo (apačioje) priklausomybės nuo neuronų skaičiaus viduriniame sluoksnyje. Epochų skaičius - 1000, lango kraštinės dydis - 20 pikselių.

Kaip matome iš rezultatų (7 pav.), geriausi rezultatai gaunami, kai neuronų skaičius viduriniame sluoksnyje yra šiek tiek didesnis už neuronų kiekį įvestiniame sluoksnyje. Tuo tarpu apmokymo laikas didėja, didėjant neuronų skaičiui viduriniame sluoksnyje.

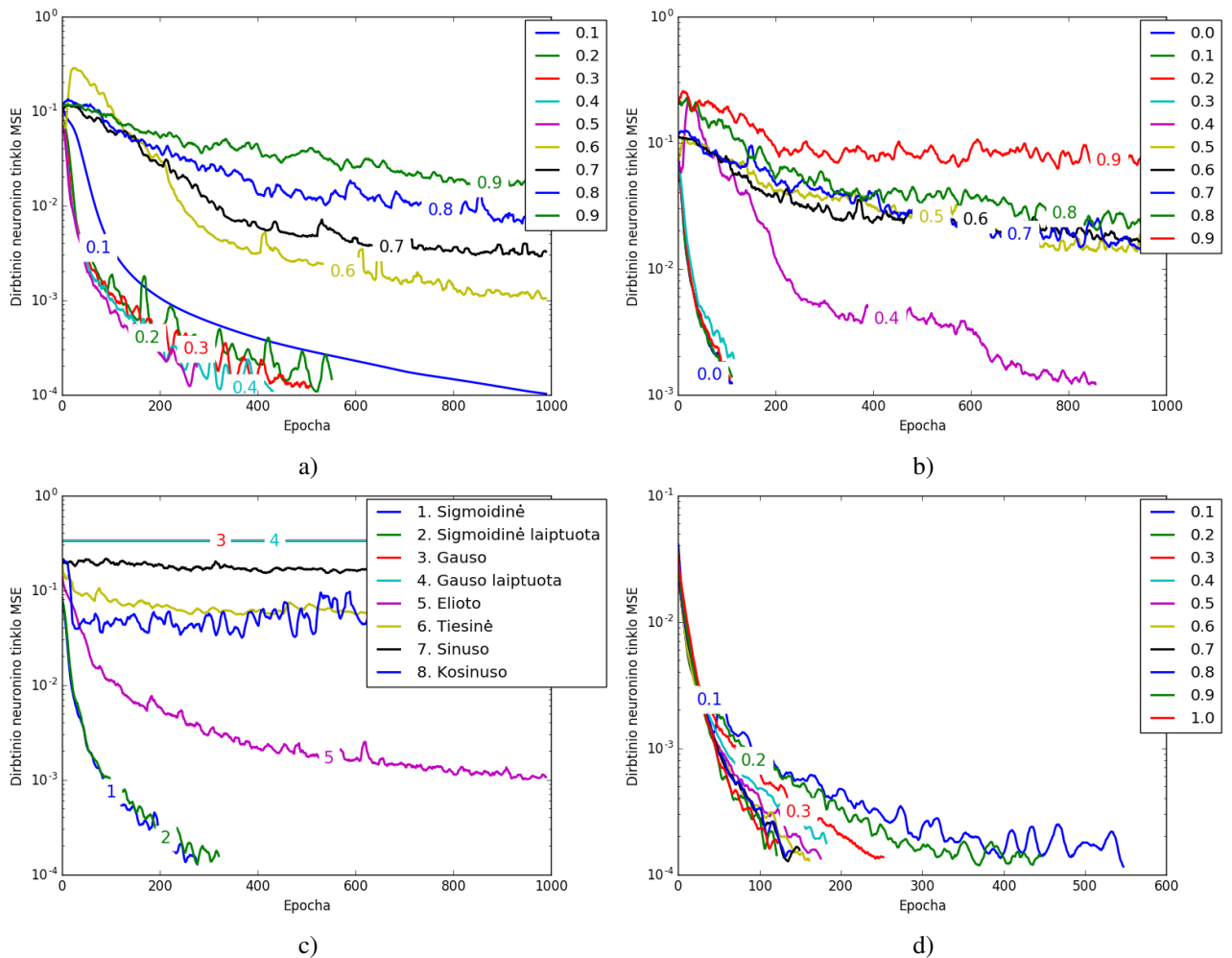
Palyginimui buvo sukonstruotas DNT su 2 ir 4 sluoksnių struktūra. Pirmuoju atveju gautas MSE - 0.1515. Tai daug prastesnis rezultatas. Naudojant keturių sluoksnių DNT, nepavyko pasiekti DNT sistemos konvergavimo.

Kadangi triukšmas sintetiniuose duomenyse yra homogeniškas (tolygus visame vaizde) ir tekstinis nespaltotas vaizdas yra pakankamai stacionarus, buvo vertinama kaip DNT apmokymo kokybė priklauso nuo vaizdo lango dydžio, kuris naudojamas DNT apmokymui. Rezultatai pavaizduoti 8 pav.



8 pav. Jungčių skaičiaus tarp neuronų (viršuje), apmokymo laiko (viduryje) bei MSE po DNT apmokymo (apačioje) priklausomybės nuo įvestinio vaizdo lango kraštinės ilgio pikseliais. Epochų skaičius - 1000, neuronų skaičius viduriniame sluoksnyje - 100.

Kaip matome iš rezultatų, visai nebūtina naudoti viso vaizdo DNT apmokyme. Rezultatai netgi geresni naudojant mažesnę lango dydį. Taip sutaupoma labai daug skaičiavimo resursų, kadangi jungčių skaičius eksponentiškai didėja didėjant lango dydžiui. Iš 7 pav. ir 8 pav. taip pat matome, kad apmokymo laikas yra proporcingas jungčių skaičiui. Taip yra todėl, kad didėjant jungčių skaičiui, atitinkamai didėja ir svorių koregavimo operacijų skaičius. Taigi skaičiavimo resursų atžvilgiu, yra tikslinga naudoti DNT struktūrą su kuo mažiau neuronų.

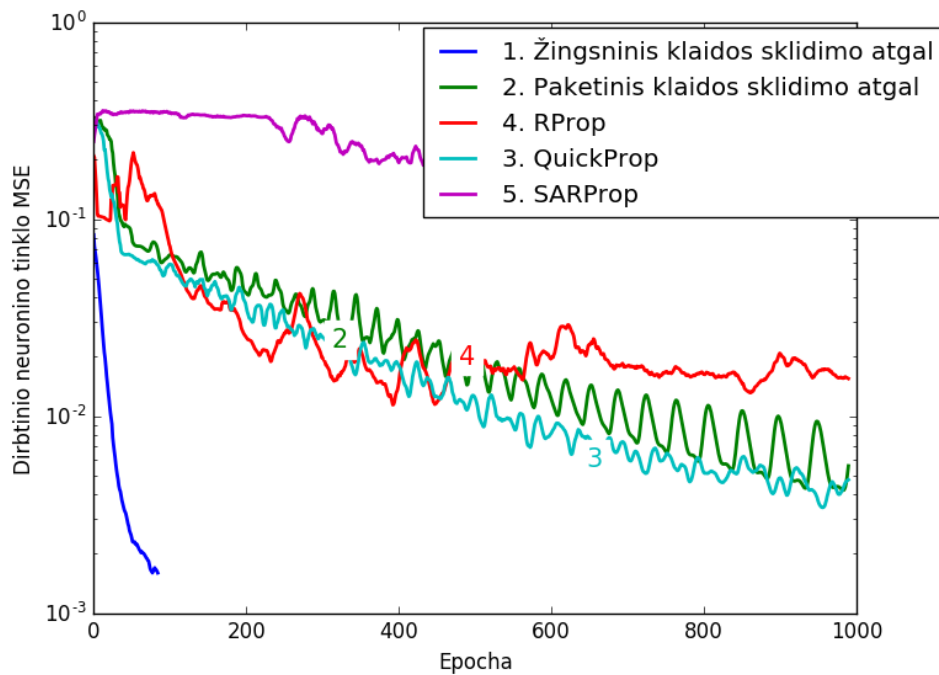


9 pav. DNT kainos funkcijos (MSE) kitimo kreivės DNT apmokymo metu priklausomybė nuo DNT parametru: a) statumo s , b) mokymosi inercijos m , c) aktyvacijos funkcijos ϕ , d) mokymosi greičio α

9 pav. pavaizduota, kaip DNT apmokymo eiga priklauso nuo kitų nustatomų DNT parametru. Iš šio pav. galima nustatyti, su kuriais parametrais DNT yra apmokomas greičiausiai ir efektyviausiai. Šios kreivės gautos, kai kiti parametrai, negu nagrinėjamas, yra optimalūs (pagal 3 lentelę). Tai pasiekama iteraciniu būdu: pirmiausia kiti parametrai, negu nagrinėjamas, yra parenkami atsitiktinai. Vėliau naudojami praeitame žingsnyje gauti geriausi parametrai. Vienos iteracijos metu taip įvertinami visi nagrinėjami parametrai. Procesas kartojamas kol nagrinėjamų parametru vertės nusistovi, t.y. kol gaunamos tos pačios parametru vertės kelis kartus iš eilės. Visi 9 pav. pavaizduoti vaizdai gauti iš paskutinės tokios iteracijos. Siekiant geriau įvertinti gautus rezultatus, ordinačių ašyje panaudota logaritminė skalė (su pagrindu 10), o kreivės nugludintos slenkančio vidurkio (*angl.* moving average) algoritmu. Pavaizduotais atvejais buvo naudojamas 100 vaizdų apmokymo duomenų rinkinys. Šiais atvejais DNT apmokymas buvo stabdomas, kai MSE vertė pasiekia 0.0001 minimalią vertę, kadangi tai yra daugiau nei pakankamas rezultatas tiksliam triukšmų atpažinimui vaizde.

Įvertinus šiuos gautus rezultatus, 9 a) pav. matome, kad DNT sistema yra efektyvesnė ir stabilesnė, kai statumo parametras nustatomas $s < 0.6$. Pagal 9 b) pav. galima nustatyti, kad nagrinėjamu atveju mokymosi inercijos parametro m taikymas nėra tikslingas. 9 c) pav. parodo, kad tik sigmo-

idinė ir Elioto aktyvacijos funkcijos yra veiksmingos šiam triukšmų nustatymo vaizde uždaviniui spręsti, o standartinė sigmoidinė funkcija čia veikia efektyviausiai. Vertinat mokymosi greičio parametro α parinkimą (9 d pav.), nustatyta, kad nuo jo DNT apmokymo efektyvumas mažai priklauso, kai jo reikšmė <0.7 .



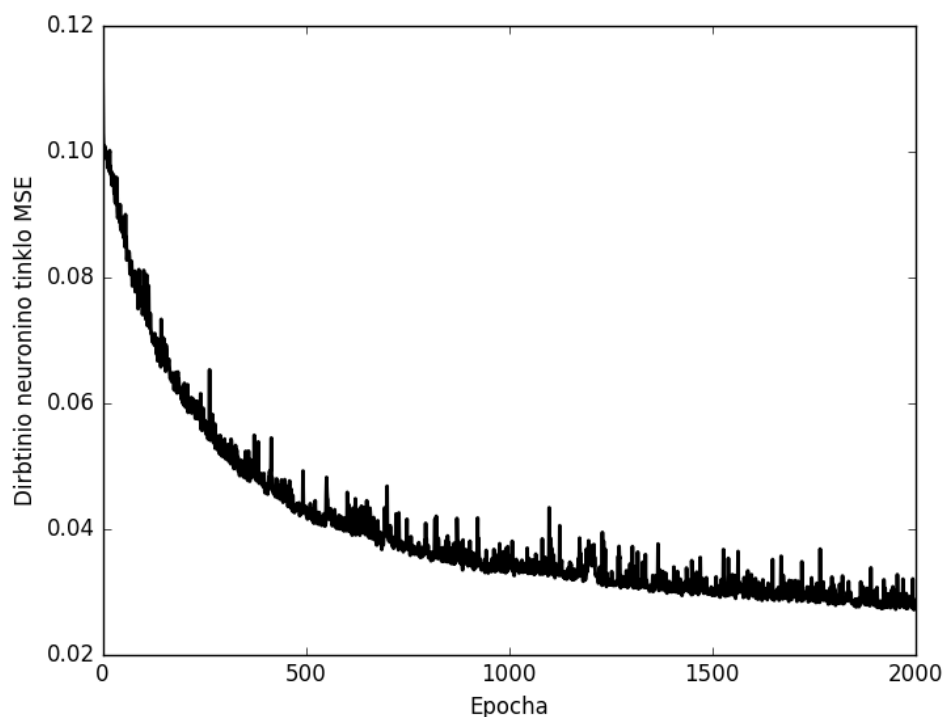
10 pav. DNT kainos funkcijos (MSE) kitimo kreivės DNT apmokymo metu priklausomybė nuo apmokymo algoritmo parinkimo.

10 pav. pavaizduota, kaip DNT apmokymo eiga priklauso nuo apmokymo algoritmo. Čia įvertinti visi 2.2 skyriuje apibūdinti apmokymo algoritmai. Šiame pav. matomas vaizdas gautas tokiu pačiu iteraciniu būdu, kaip ir 9 pav. Iš čia galima matyti, kad čia žymiai geresnius rezultatus rodo standartinis žingsninis apmokymo algoritmas. Visi kiti algoritmai nėra tokie efektyvūs. Kadangi visi kiti algoritmai yra paketiniai, galima daryti išvadą, kad šio darbo kontekste labiausiai tinkama yra žingsninė apmokymo algoritmo versija. Nors ir tokie algoritmai, kaip RProp ir Quickprop, yra laikomi ypač greitais, didelio skirtumo tarp jų ir standartinio žingsninio klaidos sklidimo atgal algoritmo nepastebėta (skaičiavimo resursų atžvilgiu).

Atsižvelgus į aptartus optimalių parametru parinkimo tyrimo rezultatus, toliau naudojami tokie DNT parametrai:

3 lentelė. Optimalūs DNT parametrai.

Parametras	Optimali reikšmė
Aktyvacijos funkcija ϕ	Sigmoidinė (Žr. formulę 2.4)
Apmokymo algoritmas	Žingsninis klaidos sklidimo atgal
Statumas s (Žr. skyrių 2.1)	0.5
Mokymosi greitis α (Žr. formulę 2.1)	0.6
Mokymosi inercija m	0



11 pav. MSE kitimas DNT apmokymo metu su optimaliais DNT parametrais.

11 pav. pavaizduota galutinio apmokymo kainos funkcijos (MSE) priklausomybė nuo epochų skaičiaus su visais gautais optimaliais DNT parametrais (nurodytais 3 lentelėje) ir galutiniu apmokymo duomenų rinkiniu (2000 vaizdų). Kaip matome, apmokymo pradžioje DNT MSE mažėja eksponentiškai. Vėliau šis mažėjimas tampa panašus į tiesinį. Nurodytų atveju MSE po paskutinės epochos - 0.027. Gautas DNT testuotas testiniais duomenimis (200 vaizdų) ir gauta MSE 0.0524.

6.1.2. Kiti regresiniai algoritmai

Palyginimui, triukšmų nustatymo vaizde uždaviniui spręsti buvo bandoma keletas kitų regresinių ir klasifikavimo algoritmų. Kaip matyti iš rezultatų (4 lentelė), naudojant DNT, rezultatai daug geresni. Taigi galima daryti išvadą, kad labiausiai tinkamas regresinis algoritmas triukšmų nustatymui vaizde yra DNT.

4 lentelė. Kitų algoritmų rezultatai su testiniais duomenimis.

Algoritmas	MSE
Sprendimų medžio klasifikavimo	0.0952
Sprendimų medžio regresinis	0.0951
k-kaimynų klasifikavimo	0.1313
k-kaimynų regresinis	0.1138
Multinominis artimiausio kaimyno	0.1807
Vienas prieš vieną klasifikavimo	0.4639
Palikimo vektoriaus klasifikavimo	0.3634

6.2. Sudėtinių filtrų gavimas naudojant GA

GA panaudojimo tikslas - parinkti optimalų sudėtinį filtrą kiekvienam triukšmo variantui. Šiame etape naudoti visi triukšmai, kurių kiekvienas turi 6 lygius. Taigi naudojant GA buvo siekiama sukurti 1296 optimalius sudėtinius filtrus.

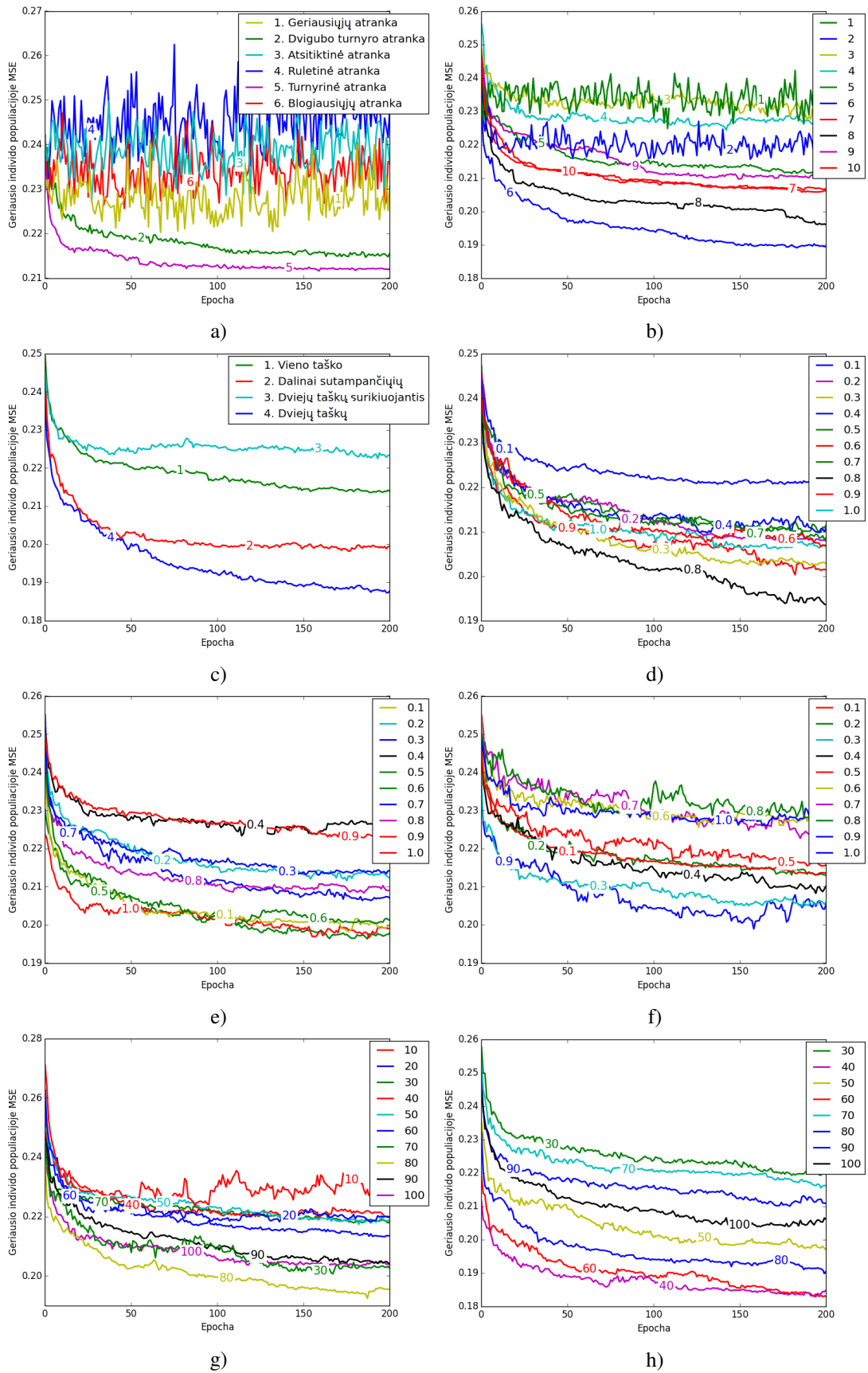
Šiame darbe nufiltruoto vaizdo MSE įvertis buvo naudojamas kaip genetinio algoritmo sprendinio tinkamumo funkcija. Šiuo atveju MSE įvertis apibūdina skirtumą tarp originalaus ir nufiltruoto vaizdo, kuris gautas panaudojus gautą sudėtinį filtrą. Genetiniu algoritmu buvo siekiama parinkti tokį filtrą, kad ši MSE vertė būtų kuo mažesnė.

6.2.1. Optimalių GA parametrų nustatymas

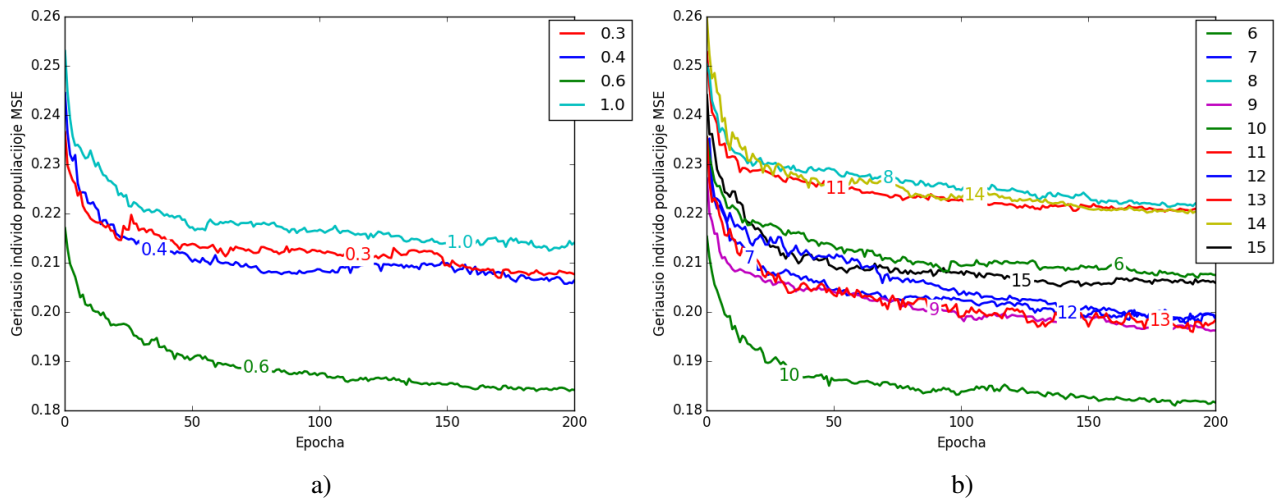
12 pav. ir 13 pav. pavaizduoti optimalių GA parametrų parinkimo tyrimo rezultatai. Čia ordinačių ašyje nurodyta reikšmė atinka geriausio populiacijos individo sprendinio tinkamumo funkcijos reikšmę, o abscisių reikšmė atitinka epochos eilės numerį. Iš tokių grafikų galima nuspręsti, kokios parametrų reikšmės leidžia gauti geriausią filtrą efektyviausiai. Šios kreivės gautos, kai kiti parametrai, negu vaizduojamas, yra optimalūs (pagal 5 lentelę). Tai pasiekama jau minėtu iteraciniu būdu: pirmiausia kiti parametrai, negu nagrinėjamas, yra parenkami atsitiktinai. Vėliau naudojami praeitame žingsnyje gauti geriausi parametrai. Vienos iteracijos metu taip įvertinami visi nagrinėjami parametrai. Procesas kartojamas kol nagrinėjamų parametrų reikšmės nusistovi t.y. kol gaunamos tos pačios parametrų reikšmės kelis kartus iš eilės. Kadangi GA veikimas ir paremtas stochastiniais (atsitiktiniais) procesais, kiekviena šiuose grafikuose nurodyta kreivė yra 10 kreivių vidurkis. Šios 10 kreivių skiriasi GA proceso metu naudojamo vaizdo triukšmo lygiais. Tai padeda išvengti atsitiktinių atvejų, kai vieno ar kito parametro reikšmė pasirodo geresnė nei visos kitos vien tik dėl atsitiktinių priežasčių.

Iš 12 a) pav. grafiko matome, kad šio darbo kontekste tik turnyro atrankos metodai yra efektyvūs. Visi kiti metodai neleidžia GA konverguoti ir pasiekti gero rezultato. Iš turnyrinės atrankos metodų akivaizdžiai geriausias yra standartinis turnyrinės atrankos metodas, kai turnyro dydis yra 6 individai (12 b) pav.). Pagal 12 c) pav., iš visų 3.2 skyriuje paminėtų kryžminimo metodų, dviejų taškų kryžminimo metodas rodo geriausius rezultatus. 12 d) pav., 12 e) pav. ir 12 f) pav. pavaizduoti tikimybinių parametrų tyrimo rezultatai. Šiuose trijuose grafikuose aiškiai neišsiskiria nė viena parametro reikšmė, todėl mažiausia MSE vertę atitinkanti parametro reikšmė ties paskutine epocha ir buvo laikoma optimalia.

Kadangi optimalių filtrų parinkimui buvo naudojama vaizdai su homogenišku triukšmu, buvo ištirta kaip GA eiga priklauso nuo procese naudojamo vaizdo lango dydžio (jo kraštinės ilgio pikseliais). Rezultatai pavaizduoti 12 h) pav. Nors čia išsiskiria 2 parametro vertės, pasirinkta mažesnė, kadangi didėjant lango dydžiui, didėja ir GA reikalingas skaičiavimo resursų poreikis.

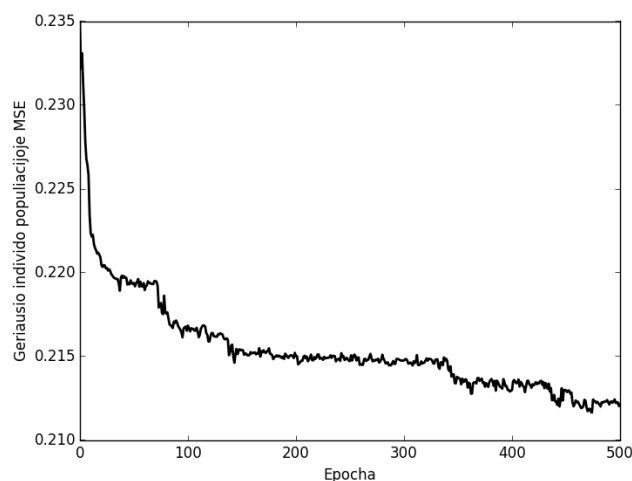


12 pav. GA eigos kreivės priklausomybė nuo GA parametru: a) atrankos metodo, b) turnyro dydžio, c) kryžminimo metodo, d) individų kryžminimo tikimybės, e) mutavimo tikimybės, f) individualios mutavimo tikimybės, g) populiacijos dydžio, h) lango dydžio.



13 pav. GA eigos kreivės priklausomybė nuo optimalaus sudėtinio filtro parametru: a) elementariojo filtro pritaikymo tikimybės, b) maksimalaus pritaikomo elementariųjų filtrų skaičiaus.

Keičiant sudėtinių filtrų užkodavimą, galima keisti tam tikrus jų parametrus: maksimalų elementariųjų filtrų pritaikymo skaičių ir filtro pritaikymo tikimybę. Šiame darbe GA individas (chromosoma) užkoduotas kaip sveikųjų skaičių masyvas, kurio elementai gali įgyti reikšmes nuo 0 iki 10. Pirmoji šio masyvo dalis apibūdina elementariųjų filtrų parametrus. Kadangi jų kiekis nėra kintamas, šios dalies dydis visada yra 14 elementų. Tolesnė dalis apibūdina filtrų taikymo tvarką, tikimybę, bei maksimalų pritaikomų filtrų kiekį. Priklausomai nuo pastarųjų dviejų parametru ši dalis kinta nuo 6 iki 90 elementų. Pagal tokį kodavimą vienas filtras gali būti panaudojamas keletą kartų viename sudėtiniame filtre. 13 pav. pavaizduota kaip GA eiga priklauso nuo šių sudėtinio filtro parametru. Čia matyti, kad optimali elementariojo filtro pritaikymo tikimybė yra 60%, o optimalus maksimalus elementariųjų filtrų pritaikymo skaičius yra 10. Taigi galima daryti išvadą, kad vienam vaizdui taikomas didelis filtrų skaičius gali ir sugadinti vaizdą, o per mažas - per mažai paveikti vaizdą ir nepašalinti triukšmų.



14 pav. GA eiga su optimaliais parametrais.

14 pav. matome GA eigą, kai visi parametrai yra optimalūs. Pagal šį grafiką buvo nuspręsta tolesniuose skaičiavimuose naudoti maksimalų 500 epochų skaičių, kadangi jau po 400 epochų randami geresni sudėtiniai filtrai savo veiksmingumu skiriasi nedaug, o didelis epochų skaičius

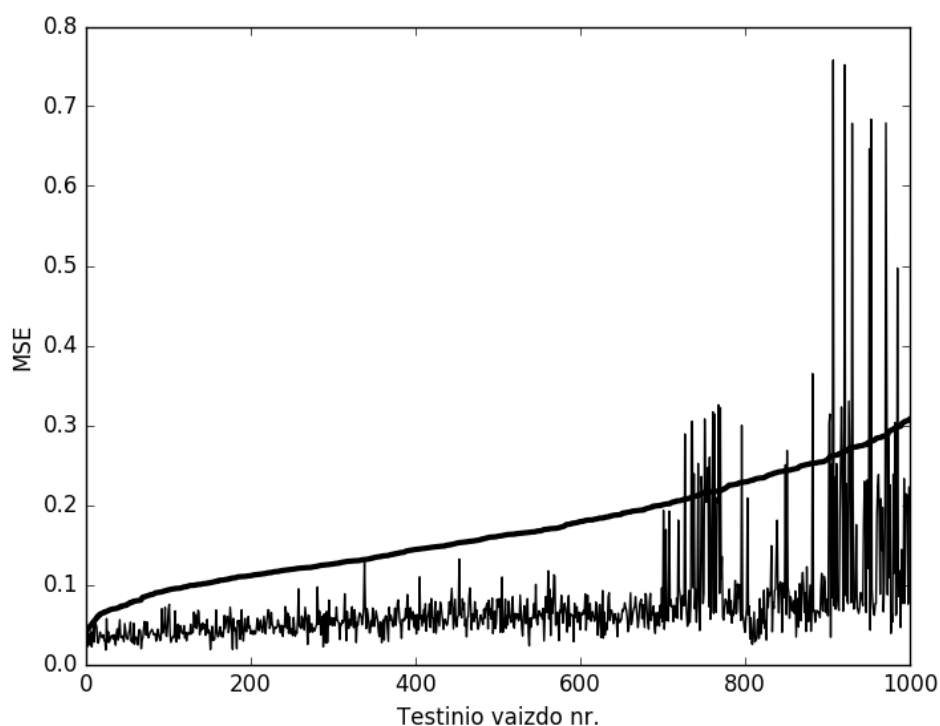
stipriai padidina skaičiavimo resursų poreikį.

Taigi įvertinus GA optimalių parametru parinkimo tyrimo rezultatus, toliau naudojami tokie GA parametrai:

5 lentelė. Optimalūs GA parametrai.

Parametras	Optimali reikšmė
Atrankos metodas	Turnyrinė atranka
Turnyro dydis	6
Kryžminimo metodas	Dviejų taškų
Kryžminimo tikimybė	0.8
Mutavimo tikimybė	0.5
Individuali mutavimo tikimybė	0.9
Populiacijos dydis	80
Lango dydis	40
Elementariojo filtro pritaikymo tikimybė	0.6
Maksimalus pritaikomų elementariųjų filtrų skaičius	10
Epochų skaičius	500

6.2.2. Sudėtinių filtrų gavimo rezultatai



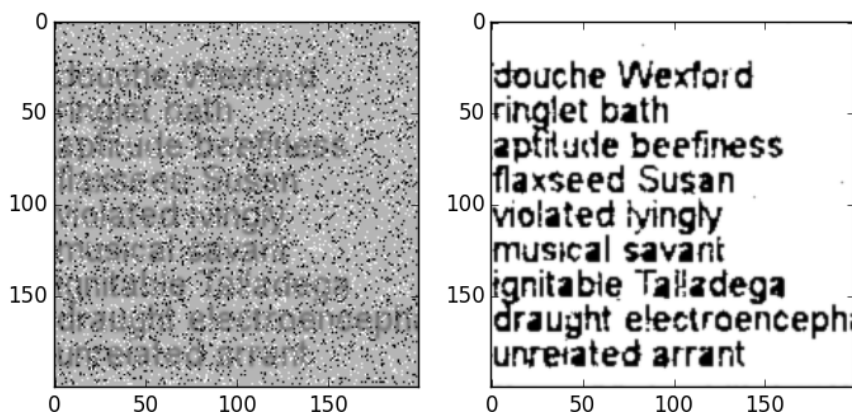
15 pav. Užtriukšminto vaizdo MSE (plačioji kreivė) bei nufiltruoto vaizdo MSE su optimaliu sudėtinu filtru (plonoji kreivė) visuose testiniuose vaizduose. Rezultatai surikiuoti pagal užtriukšminto vaizdo MSE.

Naudojant anksčiau nurodytus optimalius parametrus, GA buvo sukurti 1296 optimalūs sudėtiniai filtrai - po vieną kiekvienam įmanomam triukšmo variantui. Gauti optimalūs sudėtiniai filtrai buvo testuojami su 1000 atsitiktinai parinktų užtriukšmintų vaizdų. 15 pav. pavaizduoti šie testavimo rezultatai. Čia naudoti atsitiktiniais triukšmų lygiais užtriukšminti vaizdai, naudojant visus 4 triukšmų tipus. Filtravimas atliekamas su optimaliu tam triukšmo lygiui skirtu sudėtinio filtru t.y. triukšmo lygis bei tipas vaizde buvo žinomas. Kaip matome šiame grafike, triukšmo lygis beveik visada vaizde yra sumažinamas (96.7% visų atvejų). Tik 3.3% visų atvejų triukšmo lygis vaizde yra padidinamas. Taip pat matome, kad triukšmo lygis yra padidinamas tik vaizduose, kur triukšmo lygis yra labai stiprus, taigi tikimybė atkurti tokius vaizdus yra labai maža. Taip pat pastebėta, kad gauti sudėtiniai filtrai nėra vienodai geri ir nufiltruotų vaizdų triukšmų lygiai svyruoja.

Parametras	Vidutinė reikšmė
Užtriukšminto vaizdo MSE	0.1674
Nufiltruoto vaizdo MSE	0.0772
Originalaus vaizdo OCR tikslumas	0.8118
Užtriukšminto vaizdo OCR tikslumas	0.2256
Nufiltruoto vaizdo OCR tikslumas	0.4150

6 lentelė. Vidutinės parametru vertės, gautos atlikus GA gautų optimalių sudėtinių filtrų testavimą.

6 lentelėje pateikti agreguoti to pačio testo rezultatai. Jie rodo, kad MSE įvertį pavyko sumažinti vidutiniškai 54%, o OCR įvertį padidinti vidutiniškai 83%.



16 pav. Užtriukšmintas vaizdas (Gauso triukšmo lygis - 0, kontrasto sumažinimo lygis - 0.8, druskos ir pipirų triukšmo lygis - 0.8, vaizdo ryškumo sumažinimo lygis - 0.8) (kairėje) ir nufiltruotas vaizdas tam triukšmo lygiui skirtu optimaliu sudėtinio filtru (dešinėje).

16 pav. pavaizduoti vieno iš gautų sudėtinių filtrų filtravimo rezultatai. Šiame pav. vizualiai matome, kad genetiniu algoritmu gautas sudėtinis filtras stipriai pagerina vaizdo kokybę ir tekstas tampa daug geriau įskaitomas. Tą patį galima įvertinti iš skaitinių įverčių. Šiuo konkrečiu pavaizduotu atveju MSE įvertis po filtravimo sumažėjo nuo 0.2132 iki 0.0634, o OCR tikslumas padidėjo nuo 0 iki 0.6425. Turint omeny, kad originalaus vaizdo OCR tikslumas irgi nebuvo idealus (0.8312), tai galima laikyti geru rezultatu.

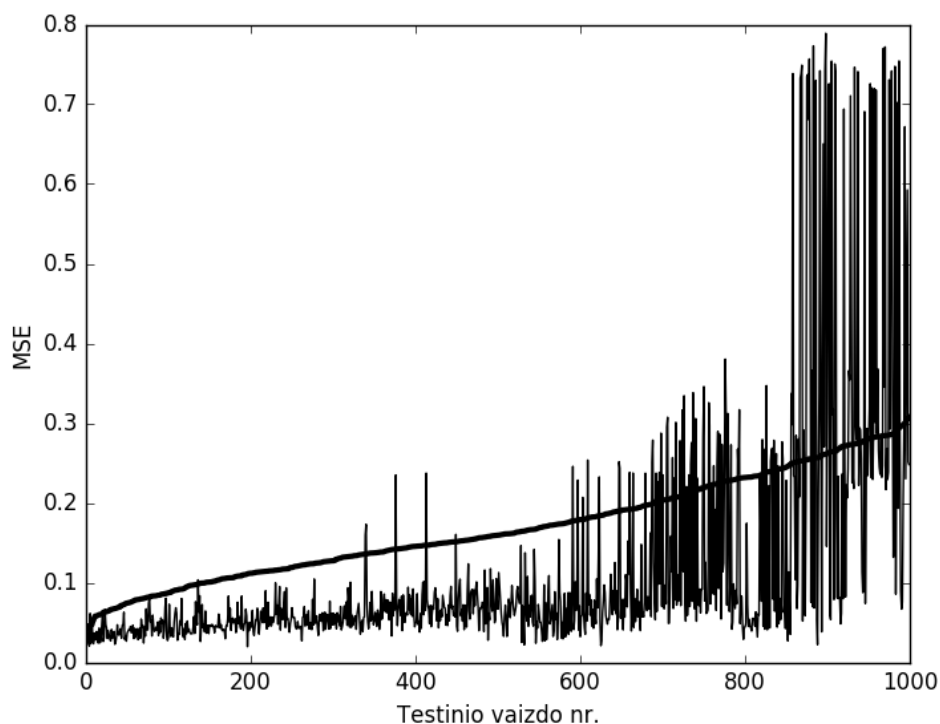
6.3. Automatinis vaizdų filtravimas

Automatinis vaizdų filtravimas buvo vykdomas panaudojant apmokytą neuroninį tinklą bei panaudojant genetiniu algoritmu gautus optimalius filtrus. Pirmiausia neuroniniu tinklu nustatomas triukšmo lygis įvestiniame vaizde, po to gautam triukšmo lygiui nufiltruoti panaudojamas optimalus sudėtinis filtras. Toks filtravimas toliau vadinamas tiesiog automatiniu filtravimo metodu. Rezultatų įvertinimui buvo panaudojama 1000 atsitiktinai sugeneruotų vaizdų. Kiekvienam iš jų atsitiktinai pritaikytas vienas iš galimų 1296 triukšmo lygių. Kiekvienam filtravimo atvejui apskaičiuoti MSE ir OCR tikslumo įverčiai prieš ir po filtravimo. Gautos vidutinės šių įverčių vertės pateiktos lentelėje:

7 lentelė. Vidutinės parametrų vertės, gautos atlikus automatinio filtravimo metodo testavimą.

Parametras	Vidutinė reikšmė
Užtriukšminto vaizdo MSE	0.1682
Nufiltruoto vaizdo MSE	0.1176
Originalaus vaizdo OCR tikslumas	0.8108
Užtriukšminto vaizdo OCR tikslumas	0.2518
Nufiltruoto vaizdo OCR tikslumas	0.3611

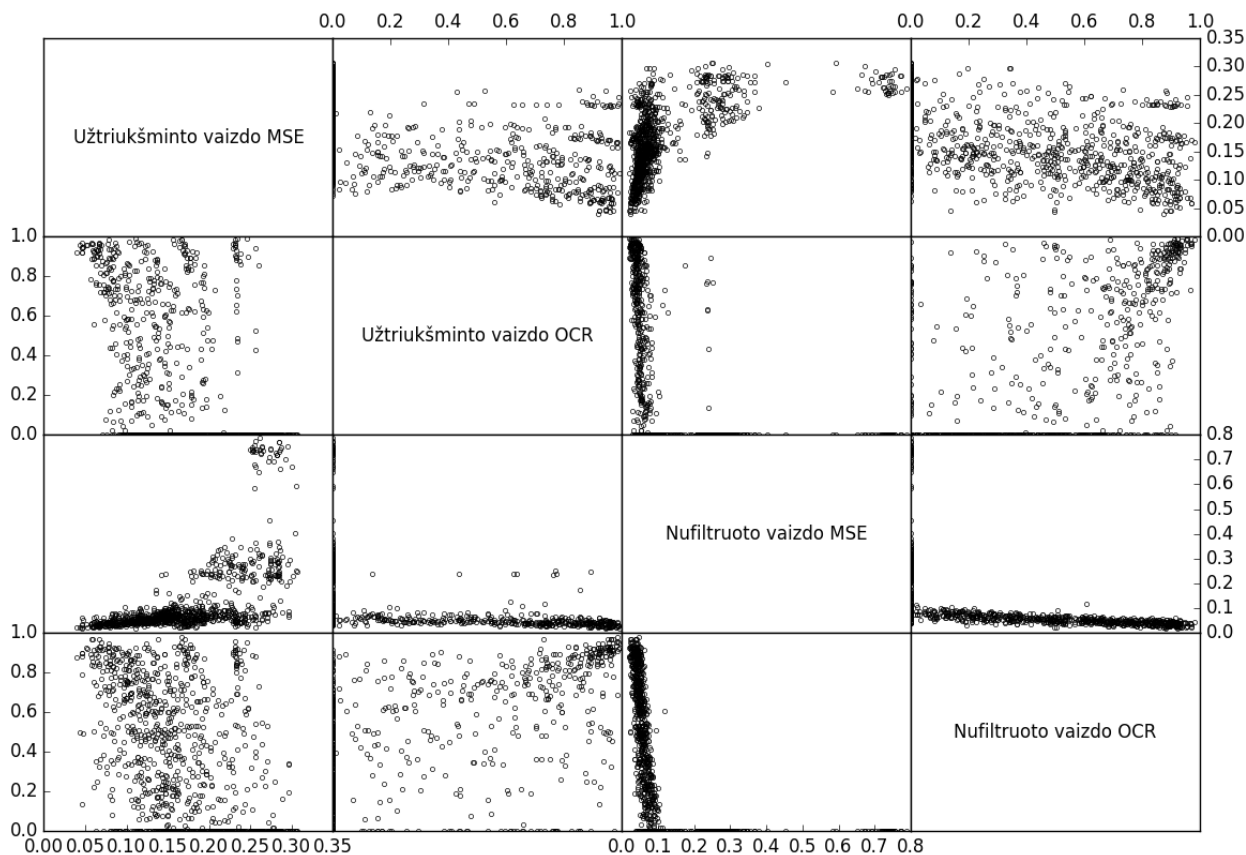
Kaip matyti iš rezultatų, šiuo automatinio filtravimo metodu pagrįstas filtravimas vidutiniškai sumažina MSE įvertį apie 30%. Tai reiškia, triukšmo lygis vaizde yra sumažinamas. Taip pat toks filtravimas pagerina OCR tikslumą vidutiniškai 43%.



17 pav. Užtriukšminto vaizdo MSE (plačioji kreivė) bei automatiniu filtravimo metodu nufiltruoto vaizdo MSE (plonoji kreivė) visuose testiniuose vaizduose. Rezultatai surikiuoti pagal užtriukšminto vaizdo MSE.

17 pav. vizualizuoti visi šio testavimo rezultatai. Kaip matome, didžiojoje dalyje (85.3%) testinių vaizdų MSE buvo sumažintas. Tačiau taip pat yra vaizdų, dažniausiai su stipriu triukšmu,

kurių vaizdo kokybė yra pabloginama. Tokių vaizdų kiekis sudaro 14.7% visų atvejų. Kadangi filtravimo rezultatai, kai triukšmo lygis yra žinomas (Žr. 15 pav. ir 6 lentelę), yra geresni, galima daryti išvadą, kad didžiąją dalį tokių vaizdų greičiausiai sudaro vaizdai, kuriuose triukšmo lygis nustatytas netiksliai ir tuomet pritaikomas ne tam triukšmo lygiui skirtas filtras.



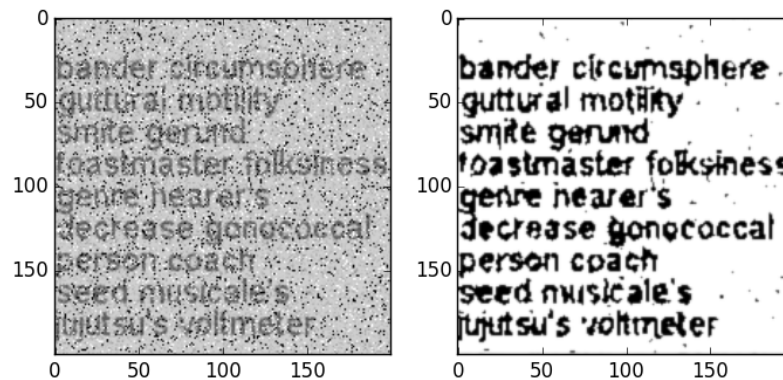
18 pav. Automatinio filtravimo metodo testavimo rezultatų (užtriukšminto vaizdo MSE, nufiltruoto vaizdo MSE, užtriukšminto vaizdo OCR tikslumo, nufiltruoto vaizdo OCR tikslumo) taškinių grafikų matrica.

18 pav. pavaizduota šio automatinio filtravimo metodo testavimo rezultatų taškinių grafikų matrica. Čia pirmiausia galima pastebėti, kaip koreliuoja MSE ir OCR įverčiai nufiltruotuose vaizduose: OCR tikslumo įvertis tiesiškai mažėja, didėjant MSE įverčiui. Kai MSE įvertis yra didesnis už 0.1, OCR tikslumas beveik visada yra lygus 0. Čia galima daryti išvadą, kad jei triukšmo lygis vaizde (MSE įvertis) yra didesnis nei 0.1, atpažinti teksto jame yra praktiškai neįmanoma. Užtriukšmintų vaizdų MSE ir OCR ryšio grafike taip pat galima išvelgti tiesinę priklausomybę, tačiau čia vertės yra gerokai daugiau pasiskirsčiusios. Dar labiau neapibrėžtą tiesinę priklausomybę galima išvelgti užtriukšminto vaizdo MSE ir nufiltruoto vaizdo OCR grafike. Čia tokios aiškios ribos nėra ir filtravimas gali sėkmingai filtruoti triukšmą ir kai MSE siekia 0.3, tačiau tokiu atveju tikimybė sėkmingai sumažinti triukšmą ir pasiekti gerą OCR įvertį gerokai sumažėja.

Taškiniame grafike, kur pavaizduotas užtriukšmintų ir nufiltruotų vaizdų OCR tikslumo ryšys, galima išvelgti, kad jei OCR tikslumas yra didelis prieš filtravimą (> 0.9), tai jis toks lieka ir po filtravimo. Visos kitos vertės (< 0.9) nekoreliuoja.

Užtriukšmintų ir nufiltruotų vaizdų MSE ryšį nurodančiame grafike, galime gauti papildomos informacijos nei pavaizduota 17 pav. Čia matome, kad vertės yra pasiskirsčiusios į tris grupes: pirmoji grupė, kur išvelgiama tiesinė priklausomybė (didėjant užtriukšminto vaizdo MSE, didėja

ir nufiltruoto vaizdo MSE); antroji grupė, kur MSE įvertis po filtravimo yra tik šiek tiek sumažinamas; trečioji grupė, kur MSE įvertis yra padidinamas. Idealiu atveju visi filtruoti vaizdai turėtų būti pirmoje grupėje. Tai greičiausiai vertės, kuriuose DNT teisingai įvertino triukšmo vertes vaizde. Antrai grupei priklauso vaizdai, kuriuose DNT įvertinti triukšmai nedaug skiriasi nuo tikrųjų verčių. Tokiems vaizdams nebuvo pritaikomas optimalus sudėtinis filtras, tačiau triukšmo lygis vaizde vis tiek yra šiek tiek sumažinamas. Trečiajai grupei greičiausiai priklauso vaizdai, kuriuose triukšmas buvo įvertintas neteisingai ir tokie vaizdai yra sugadinami.



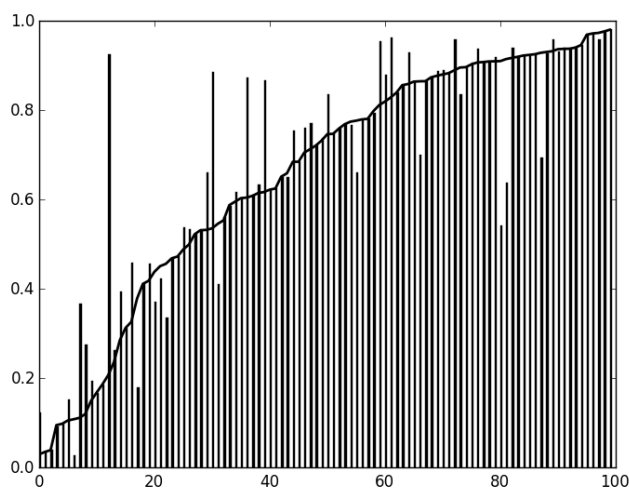
19 pav. Užtriukšmintas vaizdas (Gauso triukšmo lygis - 0.4, kontrasto sumažinimo lygis - 0.6, druskos ir pipirų triukšmo lygis - 0.6, ryškumo sumažinimo lygis - 0.6) (kairėje) ir automatinio metodo nufiltruotas vaizdas (dešinėje).

19 pav. pavaizduotas vienas iš vaizdų, nufiltruotų šiuo automatinio filtravimo metodu. Vizualiai vaizdo kokybė stipriai padidėjo ir tekstas tapo daug aiškiau įskaitomas. Gauti skaitiniai rezultatai: MSE sumažėjo nuo 0.1720 iki 0.0488, o OCR tikslumas padidėjo nuo 0.0 iki 0.6084. Šiuo atveju DNT nustatyti triukšmo lygiai šiek skiriasi nuo tikrųjų. DNT gautos vertės yra tokios: Gauso triukšmo lygis - 0.4, kontrasto sumažinimo lygis - 0.8, druskos ir pipirų triukšmo lygis - 0.2, ryškumo sumažinimo lygis - 0.6. Taigi pritaikomas sudėtinis filtras nebuvo optimalus, tačiau kadangi nustatytos ir tikrosios triukšmų vertės skiriasi nedaug, tai šis filtras pakankamai gerai filtruoja ir panašius triukšmus.

6.4. Realių duomenų automatinis filtravimas

Automatinis metodas buvo testuotas ir naudojant realius duomenis. Neuroninio tinklo apmokymas bei optimalių filtrų radimas buvo vykdomas ankstesniuose skyriuose aptartu būdu - naudojant sintetiškai sugeneruotus duomenis.

Šiam testavimui buvo naudojama 100 realių vaizdų iš IUPR (Image Understanding and Pattern Recognition) duomenų rinkinio, skirtu testuoti OCR sistemą [9]. Šie vaizdai gauti fotokamera. Didžioji dalis šių vaizdų yra prastos kokybės, yra neryškių, mažo kontrasto vaizdų. Ši duomenų rinkinį sudaro skirtingų struktūrų, fonų dokumentai, su skirtingų šriftų bei dydžių raidėmis. Taigi šis duomenų rinkinys puikiai tinka OCR bei filtravimo sistemų testavimui.



20 pav. Metodo testavimo rezultatai su realiais duomenimis. Kreivė atitinka vaizdo OCR tikslumą prieš filtravimą, vertikalūs stulpeliai - OCR tikslumą po filtravimo. Reikšmės surikiuotos pagal OCR reikšmę prieš filtravimą.

20 pav. pavaizduoti šio automatinio filtravimo metodo testavimo rezultatai su minėtais realiais duomenimis. Kaip matome, dalyje vaizdų OCR reikšmė po filtravimo yra stipriai padidinama, tačiau dalyje vaizdo kokybė yra pabloginama. Vidutiniškai OCR dydis yra padidinamas 1.7%. 58 vaizduose OCR įvertis yra padidinamas, 33 - sumažinamas, 9 - nepakitęs.

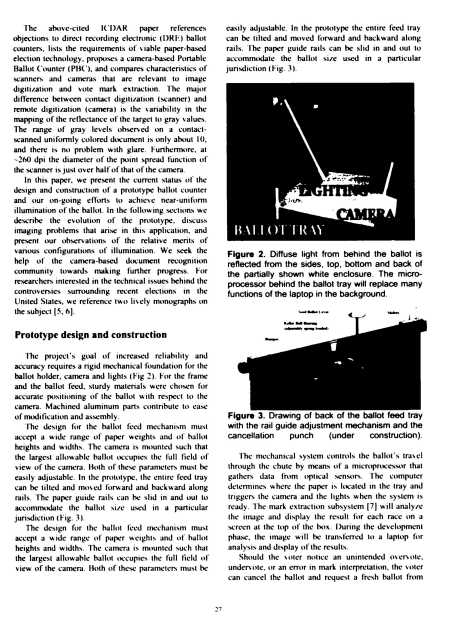
Vaizduose, kuriuose po filtravimo OCR įvertis buvo sumažinamas, greičiausiai nepavyko nustatyti triukšmo tipo bei lygio teisingai. Triukšmai tam tikromis savybėmis skiriasi realiuose ir sintetiniuose vaizduose. Todėl norint tokiu atveju pagerinti automatinio filtravimo kokybę, reikėtų įvertinti ir kitokias DNT panaudojimo implemetacijas: pvz. apmokymui naudoti realius vaizdus, dažninę vaizdo sritį arba kitus vaizdo parametrus.

Tam taip pat OCR įverčio sumažėjimui galėjo daryti įtakos ir tai, kad DNT apmokymui bei optimalių filtrų parinkimui naudojant GA buvo naudojami sintetiniai vaizdai, kurių triukšmas yra homogeniškas. Realiuose vaizduose triukšmas dažniausiai nėra homogeniškas. Implementuotu metodo atveju triukšmas nustatomas dalyje vaizdo (DNT lango dydžio srityje). Jei triukšmas kitose dalyse skiriasi, šios dalys gali būti nufiltruojamos nesėkmingai. Taigi norint pagerinti filtravimo rezultatus, reikėtų atsižvelgti į triukšmų nehomogeniškumą vaizde, segmentuojant vaizdą pagal triukšmus arba paprasčiausiai skaidant vaizdą į keletą dalių. Kiekvienai šiai daliai reikėtų nustatyti triukšmą atskirai, todėl pritaikyti ir kitą filtrą.

21 pav. pavaizduotas vienas iš automatinio filtravimo metodo realių duomenų filtravimo rezultatų. Kaip matyti iš šio pav., nufiltruotas vaizdas šiuo atveju yra binarizuotas, jo kontrastas



a)



b)

21 pav. Realus vaizdas su tekstu (kairėje) ir automatiniu filtravimo metodu nufiltruotas vaizdas (dešinėje).

žymiai geresnis, tekstas ryškesnis ir žymiai geriau įskaitomas. Šiuo konkrečiu atveju OCR įvertis filtruojant padidėjo nuo 0.6049 iki 0.8741.

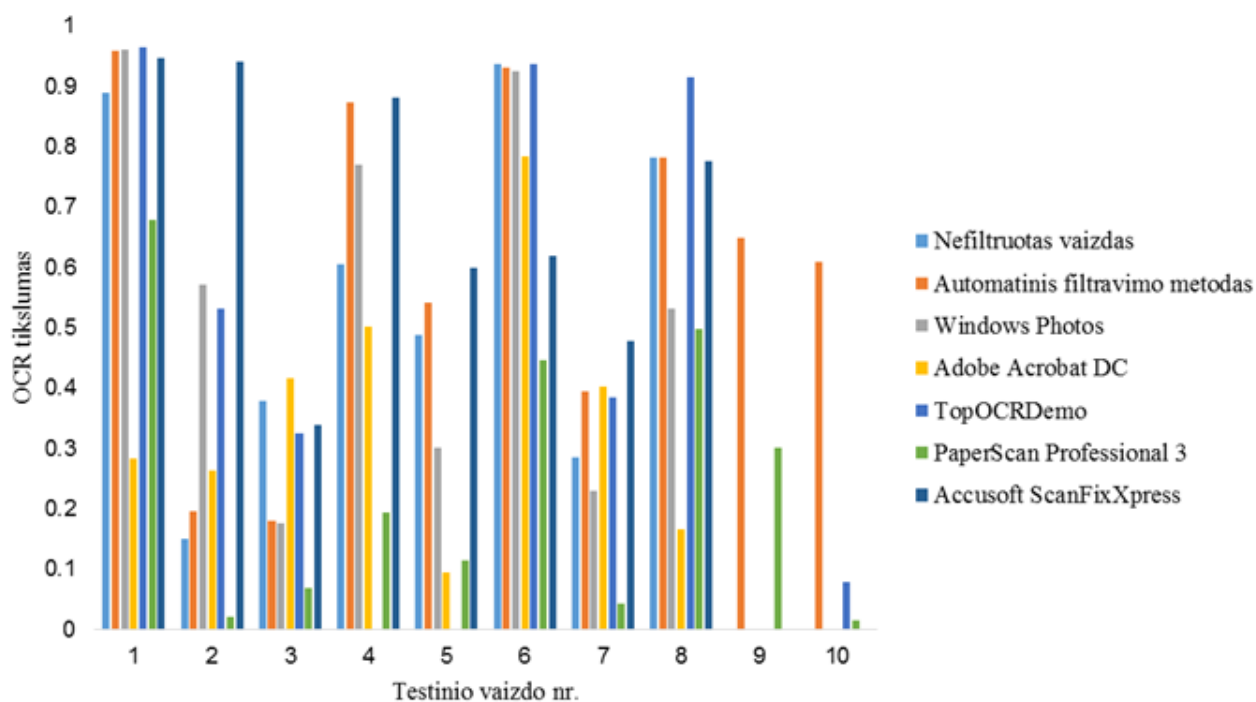
6.5. Metodo palyginimas su kitais automatiniiais vaizdo filtravimo įrankiais

Šiame darbe nagrinėtas automatinis filtravimo metodas buvo palygintas su automatiniiais vaizdų filtravimo įrankiais, kurių dauguma skirti vaizdo kokybei pagerinti prieš OCR filtravimo operaciją. Tai komerciniai ir nemokamai platinami įrankiai. Jų sąrašas bei automatiniam filtravimui naudojamos komandos apibūdintos 9 lentelėje.

8 lentelė. Automatiniai vaizdo filtravimo įrankiai.

Įrankis	Atliekamos komandos
Windows Photos	Auto Enhance (maksimalus lygis)
Adobe Acrobat DC	Optimize Scanned Image (maksimalus lygis)
TopOCRDemo	Deskew → Binarize Automatic
PaperScan Professional 3	AutoDeskew → Despeckle
Accusoft ScanFixXpress	Deskew → Binarize Automatic

22 pav. pavaizduoti automatinių filtravimo įrankių palyginimas. Čia automatino filtravimo įrankio stulpeliai atitinka vaizdų OCR tikslumo vertę po filtravimo, naudojant tą įrankį. 1 - 8 nr. pažymėti vaizdai atsitiktinai paimti iš anksčiau minėto realių duomenų rinkinio. 9 ir 10 nr. pažymėti vaizdai yra sintetiniai užtriukšminti vaizdai. Pastaruosius vaizdus, kaip ir buvo galima tikėtis, geriausiai filtruoja šiame darbe nagrinėtas automatinis filtravimo metodas, kadangi panašiais vaizdais ir buvo apmokyta visa sistema. Vertinant realių vaizdų filtravimą, rezultatai įvairūs:



22 pav. Automatinių vaizdo filtravimo įrankių palyginimo rezultatai.

dalyje vaizdų OCR reikšmė yra padidinama, dalyje sumažinama. Vienintelis įrankis Accusoft ScanFixXpress pagerina OCR įvertį visuose realiuose vaizduose.

Agreguoti rezultatai pateikti 9 lentelėje. Geriausiai šiuo atveju pasirodė Accusoft ScanFixXpress įrankis. Siekiant objektyvumo, į šias vidutines vidutines vertes neįtraukti sintetiniai užtriukšminti vaizdai. Įdomu tai, kad dalis įrankių (Adobe Acrobat DC, TopOCRDemo ir PaperScan Professional 3) netgi vidutiniškai pablogina OCR įvertį vaizde. Nors Windows Photos AutoEnhance komanda nėra skirta OCR vaizdų kokybei pagerinti, o apskritai vaizdų kokybei gerinti, šio įrankio rezultatai yra pakankamai geri. Tuo tarpu šiame darbe aptartas automatinio filtravimo metodas, naudojant GA, gali konkuruoti su kitais įrankiais ir daugelyje atveju jiems nenusileidžia.

9 lentelė. Automatinių vaizdų filtravimo įrankių palyginimo rezultatai

Vaizdų grupė	Vidutinė OCR reikšmė	OCR pokytis
Nefiltruotas vaizdas	0.533447315	-
Automatinis filtravimo metodas	0.58214759	9.12%
Windows Photos	0.561984466	5.34%
Adobe Acrobat DC	0.392301533	-26.5%
TopOCRDemo	0.448863984	-15.85%
PaperScan Professional 3	0.223933775	-58.02%
Accusoft ScanFixXpress	0.686042818	28.6%

Išvados ir rekomendacijos

Tyrimo metu buvo sėkmingai realizuotas automatinis rastrinių vaizdų filtravimo metodas, skirtas pagerinti optinio simbolių atpažinimo tikslumą vaizde. Atlikus tyrimą bei rezultatų analizę galima daryti tokias pagrindines išvadas:

- Gavus optimalius dirbtinio neuroninio tinklo parametrus, nustatyta, kad tinkamiausia aktyvacijos funkcija yra sigmoidinė (Žr. 9 c) pav.), o efektyviausias apmokymo algoritmas - žingsninis klaidos sklidimo atgal (Žr. 10 pav.). Taip pat nustatyta, kad dirbtinio neuroninio tinklo skaičiavimo kokybė beveik nepriklauso nuo apmokymui naudojamo vaizdo lango dydžio, tačiau šis dydis turi labai didelės įtakos skaičiavimo resursų poreikiams (Žr. 8 pav.). Optimali dirbtinio neuroninio tinklo struktūra yra sudaryta iš 3 sluoksnių, kai neuronų skaičius viduriniame sluoksnyje yra šiek tiek didesnis nei įvestiniame (Žr. 7).
- Gavus optimalius genetinio algoritmo parametrus, nustatyta, kad tinkamiausias yra turnyrinės atrankos metodas, o efektyviausias kryžminimo metodas - dviejų taškų (Žr. 12 pav.).
- Nustačius optimalius sudėtinio filtro parametrus, gauta, kad optimali elementariojo filtro pritaikymo tikimybė yra 60%, o tinkamiausias maksimalus pritaikomų elementariųjų filtrų kiekis yra 10 (Žr. 13 pav.).
- Testuojant optimalius sudėtinius filtrus, gautus genetiniu algoritmu, išsiaiškinta, kad tokie filtrai vidutiniškai sumažina triukšmo lygį vaizde 54% ir pagerina optinio simbolių atpažinimo tikslumą 83% (Žr. 6 lentelę ir 15 pav.). Šiuo atveju triukšmas vaizde buvo žinomas ir pritaikomas tam triukšmo tipui bei lygiui skirtas filtras. Triukšmo lygis vaizde yra sumažinamas 96.7% visų testuotų atvejų.
- Tiriant automatinį filtravimo metodą gauta, kad šiuo metodu filtruojant sintetinius vaizdus, triukšmo lygis vaizde yra sumažinamas vidutiniškai 30%, o optinio simbolių atpažinimo tikslumas yra padidinamas vidutiniškai 43% (žr. 7 lentelę ir 17 pav.). Triukšmo lygis vaizde yra sumažinamas 85.3% visų testuotų atvejų. Be to toks filtravimas vizualiai pagerina vaizdo kokybę ir tekstas tampa daug geriau įskaitomas (žr. 19 pav.).
- Vaizduose po filtravimo, optinio simbolių atpažinimo tikslumo ir MSE įverčiai koreliuoja ir yra atvirkščiai proporcingi (žr. 18 pav.).
- Tikrinant automatinį filtravimo metodą su realiais duomenimis išsiaiškinta, kad metodas pagerina optinio simbolių atpažinimo tikslumą vaizduose vidutiniškai 1.7% (žr. 20 pav.). Reikia paminėti, kad ne visuose vaizduose optinio simbolių atpažinimo tikslumas buvo pagerintas. Tokie atvejai sudarė 33% visų vaizdų.
- Palyginus automatinį filtravimo metodą su kitais automatinio filtravimo įrankiais nustatyta, kad šis metodas gali konkuruoti su jais ir daugelyje atvejų jiems nenusileidžia (žr. 22 pav. ir 9 lentelę).
- Ne visais atvejais (filtruojant tiek sintetinius, tiek realius vaizdus) nagrinėjamas automatinis filtravimo metodas pagerina optinio simbolių atpažinimo tikslumą bei vaizdo kokybę. Todėl norint pagerinti šio metodo rezultatus reikėtų pagerinti triukšmų nustatymo vaizde rodiklius bei atsižvelgti į realių vaizdų triukšmų nehomogeniškumą.

Ateities tyrimų gairės

Galimos tolesnės tyrimų kryptys šia tema:

- Pritaikyti metodą spalvotiems vaizdams;
- Patobulinti metodą realiems vaizdams, atsižvelgiant į triukšmų nehomogeniškumą;
- Pagerinti triukšmų nustatymo rodiklius, panaudojant tam tikrus vaizdo parametrus arba dažninę vaizdo charakteristiką;
- Išskaidyti filtravimo procesą į filtravimo ir binarizavimo į atskiras sistemas;
- Pritaikyti metodą kitokio tipo duomenims.

Literatūros šaltiniai

- [1] Fann2 reference manual. <http://libfann.github.io/fann/docs/files/fann-h.html>. December 2016.
- [2] Inline labels in matplotlib. <http://stackoverflow.com/questions/16992038/inline-labels-in-matplotlib>. December 2016.
- [3] Redirect stdout to a file in python. <http://stackoverflow.com/questions/4675728/redirect-stdout-to-a-file-in-python>. December 2016.
- [4] Removing high frequency contents from the image. <http://stackoverflow.com/questions/39837268/how-to-remove-high-frequency-contents-from-the-image-for-inverse-fourier-transfo>. December 2016.
- [5] Scatter plot matrix. <http://stackoverflow.com/questions/7941207/is-there-a-function-to-make-scatterplot-matrices-in-matplotlib>. December 2016.
- [6] E. R. Banks, P. Agarwal, M. McBride, and C. Owens. Evolving Image Noise Filters through Genetic Programming. *DoD High Performance Computing Modernization Program Users Group Conference*, 13:307--311, 2009.
- [7] A. K. Boyat and B. K. Joshi. A review paper: noise models in digital image processing. *Signal & Image Processing : An International Journal*, 6(2), 2015.
- [8] A. Buades, B. Coll, and J. M. Morel. On image denoising method. Technical report, CMLA (Centre de Mathematiques et de Leurs Applications), 2004.
- [9] S. S. Bukhari, F. Shafait, and T. M. Breuel. *Proceedings of the 4th international conference on Camera-Based Document Analysis and Recognition*, 2011.
- [10] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with bm3d? *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, 2012.
- [11] F. Cannavó, G. Nunnari, D. Giordano, and C. Spampinato. Variational method for image denoising by distributed genetic algorithms on grid environment. *Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2006.
- [12] M. Cetin, T. Kavzoglu, and N. Musaoglu. Classification of multi-spectral, multi-temporal and multi-sensor images using principal components analysis and artificial neural networks: Beykoz case. *International congress for photogrammetry and remote sensing*, 2004.
- [13] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20:89 -- 97, 2004.
- [14] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber. Flexible, High Performance Convolutional Neural Networks for Image Classification. *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence*, 2, 2011.

- [15] J. L. de Paiva, C. F. M. Toledo, and H. Pedrini. A hybrid genetic algorithm for image denoising. *IEEE Congress on Evolutionary Computation (CEC)*, pages 2444--2451, 2015.
- [16] J. Duran, B. Coll, and C. Sbert. Chambolle's projection algorithm for total variation denoising. *Image Processing On Line*, 3:311 -- 331, 2013.
- [17] E. Egmont-Petersen, D. de Ridder, and H. Handels. Image processing with neural networks — a review. *Pattern Recognition*, 35:2279 -- 2301, 2002.
- [18] S. E. Fahlman. An empirical study of learning speed in back-propagation networks. Technical report, Carnegie Mellon University.
- [19] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. Deap: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171 -- 2175, July 2012.
- [20] G. Giacinto and F. Roli. Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing*, 19:699, 2001.
- [21] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., 1 edition, 1989.
- [22] Alexander K. Hartmann and Heiko Reiger. *Optimization Algorithms in Physics*. Wiley-VCH, 1 edition, 2001.
- [23] A. Juozapavičius and T. Meškauskas. *Vaizdų ir signalų analizė ir apdorojimas. Mokomoji knyga*. UAB TEV, 1 edition, 2011.
- [24] A. Kieri. Context dependent thresholding and filter selection for optical character recognition. Master's thesis, Uppsala University, 2012.
- [25] H. Kohmura and T. Wakahara. Determining optimal filters for binarization of degraded characters in color using genetic algorithms. *The 18th International Conference on Pattern Recognition*, pages 3 -- 7, 2006.
- [26] D. Misra, S. Sarker, S. Dhabal, and A. Ganguly. Effect of using genetic algorithm to denoise mri images corrupted with rician noise. *IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology*, pages 146--151, 2013.
- [27] Maria Petrou and Panagiota Bosdogianni. *Image Processing, the Fundamentals*. Wiley-VCH, 1 edition, 1999.
- [28] Russell D. Reed and Robert J. Marks II. *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. A Bradford Book, 1 edition, 1999.
- [29] M. Riedmiller. Rprop - description and implementation details. Technical report, University of Karlsruhe, 1994.
- [30] I. W. Selesnick and I. Bayram. Total variation filtering, 2010.
- [31] A. Souza, M. Cheriet, S. Naoi, and C. Y. Suen. Automatic filter selection using image quality assessment. *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, 2003.

- [32] I. Ştirb. Highlight image filter significantly improves optical character recognition on text images. *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition*, 2014.
- [33] C. F. M. Toledo, L. de Oliveira, R. D. da Silva, and H. Pedrini. Image denoising based on genetic algorithm. *IEEE Congress on Evolutionary Computation*, pages 1294--1301, 2013.
- [34] N. K. Treadgold and T. D. Gedeon. The sarprop algorithm: A simulated annealing enhancement to resilient back propagation. *Proceedings International Panel Conference on Soft and Intelligent Computing*.
- [35] D. R. Wilson and T. R. Martinez. The general inefficiency of batch training for gradient descent learning. *Neural Networks*, 16:1429 -- 1451, 2003.
- [36] I. Zavorin, E. Borovikov, M. Turner, and L. Hernandez. Adaptive pre-ocr cleanup of grayscale document images. *Document Recognition and Retrieval*, 13, 2005.