



KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS

Gintarė Butkutė

**DIRBTINIO INTELEKTO METODŲ TAIKymo SKAIČIŲ
ATPAŽINIMUI GALIMYBIŲ TYRIMAS**

Baigiamasis magistro projektas

Vadovas

Doc. dr. Vidas Raudonis

KAUNAS, 2017

**KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS
AUTOMATIKOS KATEDRA**

**DIRBTINIO INTELEKTO METODŲ TAIKYMO SKAIČIŲ
ATPAŽINIMUI GALIMYBIŲ TYRIMAS**

Baigiamasis magistro projektas
Valdymo technologijos (621H66001)

Vadovas
Doc. dr. Vidas Raudonis

Recenzentas

Projektą atliko
Gintarė Butkutė

KAUNAS, 2017



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Elektros ir elektronikos

(Fakultetas)

Gintarė Butkutė

(Studento vardas, pavardė)

Valdymo technologijos,

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Dirbtinio intelekto metodų taikymo skaičių atpažinimui galimybių tyrimas“

AKADEMINIO SAŽININGUMO DEKLARACIJA

20 17 m. gegužės _____ d.
Kaunas

Patvirtinu, kad mano **Gintarės Butkutės** baigiamasis projektas tema „Dirbtinio intelekto metodų taikymo skaičių atpažinimui galimybių tyrimas“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

Butkutė, Gintarė. Dirbtinio intelekto metodų taikymo skaičių atpažinimui galimybių tyrimas. *Valdymo sistemų magistro* baigiamasis projektas / vadovas doc. dr. Vidas Raudonis; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas, Automatikos katedra.

Mokslo kryptis ir sritis: Elektros ir elektronikos inžinerija, Technologiniai mokslai

Reikšminiai žodžiai: *PCA, HOG, dirbtiniai neuroniniai tinklai, atraminių vektorių mašinos, atsitiktiniai miškai, vaizdų apdorojimo metodai.*

Kaunas, 2017. 51 p.

SANTRAUKA

Šio darbo tikslas yra ištirti požymių išskyrimo ir klasifikavimo metodų galimybes skaitmenims atpažinti bei interaktyviai nuskaityti rezultatų švieslentės duomenis.

Pirmame šio darbo skyriuje atlikta literatūros apžvalga, siekiant ištirti, kokie vaizdo analizės metodai yra naudojami, norint atpažinti skaitmenis.

Antrame skyriuje pateikiamos matematinės išraiškos, kurios padės geriau suprasti metodų procesą. Egzistuoja daug požymių ištraukimo metodų, skirtų vaizdų apdorojimui. Pateiktame darbe analizuojami du požymių išskyrimo metodai: principinių komponentių analizė (PCA) ir orientuotų gradientų histogramos (HOG). Taip pat gali būti naudojami įvairūs klasifikavimo metodai, skirti požymiams suklasifikuoti. Šiame darbe, skaitmenų atpažinimui buvo taikomi dirbtiniai neuroniniai tinklai (ANN), atraminių vektorių mašinos (SVM) ir atsitiktiniai miškai (RF).

Trečiame šio darbo skyriuje pateikti algoritmai požymių išskyrimo ir klasifikavimo metodų algoritmai, kurie parašyti su *C#* programavimo kalba *Visual Studio 2015* aplinkoje, panaudojant *OpenCV* biblioteką. Taip pat pateikti LED segmentų skaitmenų aptikimo ir atpažinimo sistemos tyrimų rezultatai. Visais paminėtais metodais išmokyta ir ištestuota sistema su 1000 skirtingų paveikslėlių. Geriausias atpažinimo tikslumas, kuris lygus 99,4%, gautas panaudojus orientuotų gradientų histogramas ir atramines vektorių mašinas.

Butkutė, Gintarė. Research of Artificial Intelligence Methods for Number Recognition Application: *Control Systems Master's thesis / supervisor doc. dr. Vidas Raudonis*. Kaunas University of Technology, Faculty of Electrical and Electronics Engineering, department of Automation.

Research area and field: Electrical and Electronics Engineering, Technological Sciences

Key words: *PCA, HOG, Artificial Neural Networks, Support Vector Machines, Random Forests, image processing methods*.

Kaunas, 2017. 51 p.

SUMMARY

The goal of this paper is to research feature extraction and classification methods possibilities for numerals recognition.

The state of the art of the number recognition methods is presented in the first chapter of the thesis.

In the second chapter, some background mathematical skills that will be required to understand the process of methods is given. There are a lot of feature extraction methods for image processing. Presented paper deals with two feature extraction methods: Principal Component Analysis (PCA) and Histogram of Oriented Gradients (HOG). Various classification methods can be used for features classification. In this paper, Artificial Neural Networks (ANN), Support Vector Machines (SVM) and Random Forests (RF) have been applied to numerals recognition.

On the third section of this paper, algorithms of feature extraction and classification methods are shown, which are written with *C#* programming language in *Visual Studio 2015*, using *OpenCV* library. The implemented algorithms for numerals of LED segments recognition are investigated and results are analyzed. All methods, which are mentioned, were trained and tested on a data set of 1000 different images. Experimental results show that the recognition accuracy is the best (99,4 %), when we use HOG with SVM.

TURINYS

SANTRUMPŲ IR ŽENKLŲ AIŠKINIMO ŽODYNAS	8
ĮVADAS	10
1 LITERATŪROS APŽVALGA.....	11
1.1 Automobilių numerių aptikimas ir atpažinimas	11
1.2 Banknoto serijos numerio aptikimas ir atpažinimas.....	13
1.3 Ranka rašytų ženklų atpažinimas	14
1.3.1 Dirbtinių neuroninių tinklų taikymas ranka rašytiems skaitmenims atpažinti.	14
1.3.2 Sąsūkinis neuroninis tinklas ranka rašytiems ženklams atpažinti.....	16
1.3.3 Atraminių vektorių mašinų taikymas ranka rašytiems skaitmenims atpažinti.	17
1.4 Interaktyvus skaitiklio rodmenų nuskaitymas	19
2 TEORINIS METODŲ PAGRINDIMAS	21
2.1 Vaizdo apdorojimo metodų analizė.....	21
2.2 Požymių išskyrimo metodas PCA	21
2.3 Požymių išskyrimo metodas HOG	23
2.3.1 Gradiento skaičiavimas	24
2.3.2 Histogramų generavimas.....	24
2.3.3 Histogramų normalizavimas	24
2.4 Požymių klasifikavimo metodas ANN	25
2.4.1 Biologinis neuroninis tinklas.....	25
2.4.2 Dirbtinis neuroninis tinklas	26
2.4.3 Vienasluoksniai ir daugiasluoksniai dirbtiniai neuroniniai tinklai.....	29
2.5 Požymių klasifikavimo metodas SVM.....	29
2.5.1 Daugialypis SVM klasifikatorius	31
2.6 Atsitiktinių miškų taikymas.....	31
3 EKSPERIMENTINIAI TYRIMAI	33
3.1 LED segmentų skaitmenų atpažinimo sistema.....	33
3.2 Duomenų bazės sudarymas	34
3.3 Požymių ištraukimo metodai	36
3.4 Požymių klasifikavimo metodai	36
3.5 LED segmentų skaitmenų atpažinimo sistemos tyrimas	38

REZULTATAI IR IŠVADOS	49
NAUDOTOS LITERATŪROS IR KITŲ INFORMACIJOS ŠALTINIŲ SĄRAŠAS	50
PRIEDAS 1. PCA PROGRAMOS FRAGMENTAS	52
PRIEDAS 2. HOG PROGRAMOS FRAGMENTAS	53
PRIEDAS 3. ANN PROGRAMOS FRAGMENTAS	54
PRIEDAS 4. SVM PROGRAMOS FRAGMENTAS	56
PRIEDAS 5. RF PROGRAMOS FRAGMENTAS	57

SANTRUMPŲ IR ŽENKLŲ AIŠKINIMO ŽODYNAS

PCA	Principinių komponentų analizė (angl. <i>Principal Component Analysis</i>)	Tai dimensijų mažinimo technika, paremta kovariacijos matrica.
HOG	Orientuotų gradientų histogramos (angl. <i>Histogram of Oriented Gradients</i>)	Tai metodas, naudojamas vaizde esantiems požymiams ištraukti. Ieškomas objektas suskirstomas į sritis, kurioms skaičiuojamos orientuotų gradientų histogramos
ANN	Dirbtinis neuroninis tinklas (angl. <i>Artificial Neural Network</i>)	Tai matematinis modelis, kuris imituoja biologinį neuroninį tinklą.
SVM	Atraminių vektorių mašinos (angl. <i>Support Vector Machines</i>)	Tai metodas, kurio ypatumas netiesinis $n - 1$ matės įeigos erdvės atvaizdavimas kitoje (galimai didesnių matavimų) erdvėje, leidžiančiai suformuoti tiesinę skiriamąją hiperplokštumą.
RF	Atsitiktiniai miškai (angl. <i>Random Forests</i>)	Metodas, kuris naudoja sprendimų medį kaip prognozavimo modelį.
KNN	K – artimiausių kaimynų (angl. <i>K – Nearest Neighbors</i>)	Tai metodas, kurio idėja yra naujo objekto palyginimas su mokymo aibės objektais, kurie yra panašūs į jį, o norint naują objektą priskirti kuriai nors klasei, yra skaičiuojami atstumai nuo to objekto iki visų mokymo aibės objektų.
MLP	Daugiasluoksniai perceptronai (angl. <i>Multilayer Perceptrons</i>)	Tai viena iš dirbtinių neuroninių tinklų struktūrų, kurią sudaro bent du sluoksniai.
LED	Diodas (angl. <i>Light Emitting Diode</i>)	Diodas, kurio sandara pritaikyta šviesai skleisti.
HD_MR	Hausdorfo atstumas skaitiklio nuskaitymui (angl. <i>The Hausdorff distance for Meter reading</i>)	Naujas skaitmenų atpažinimo algoritmas, skirtas elektros skaitiklio rodmenims interaktyviai nuskaityti.
HSV	Atspalvis, įsotinimas, vertė (angl. <i>Hue, Saturation, Value</i>)	Pirminio atvaizdo apdorojimo modelis.

CCA	Sujungtų komponentų analizė (angl. <i>Connected Component Analysis</i>)	Požymių išgavimo metodas.
DCT	Diskretinė kosinuso transformacija (angl. <i>Discrete Cosine Transform</i>)	Signalų perkėlimas į dažnio sritį.

IVADAS

Švieslentėje matomi rungtynių rezultatai, laikas, likęs iki rungtynių pabaigos (kėlinys) yra svarbus ne tik rungtyniaujančioms komandoms, bet ir lažybų punktų atstovams, žurnalistams, kurie atstovauja skirtingus televizijos kanalus bei žurnalus. Suvedinėjant rungtynių rezultatus ranka, reikia įvertinti žmogiškąjį faktorių, t. y. laikui einant, jaučiamas nuovargis, kuris lemia klaidų atsiradimą. Rezultatų galima nepamatyti dėl blogo apšvietimo ar tiesiog dėl netikėtai atsiradusios kliūtis stebėjimo erdvėje. Dėl šių priežasčių reikalinga automatinė sistema. Vis populiarėjantis vaizdų atpažinimas taikomas įvairiose srityse. Ne išimtis ir skaičių atpažinimas. Šiame darbe tiriamos dirbtinio intelekto metodų taikymo skaičiams (esantiems švieslentėje) atpažinti galimybės. Daugelis klasikinių vaizdo apdorojimo metodų analizuoja kiekvieną vaizdo elementą (piksely) ir greta esančių (kaimyninių) pikselių reikšmes. Vaizdo apdorojimo proceso metu kiekvienas pikselio įvertinimas užima didelę dalį skaičiavimo resursų, todėl požymių išskyrimo realizacija turi didelę įtaką viso proceso efektyvumui. Nagrinėjami plačiai naudojami PCA ir HOG požymių išskyrimo metodai. Atlikus požymių išskyrimą, reikia išskirtus požymius klasifikuoti. Šiam etapui realizuoti taikomi įvairūs požymių klasifikavimo metodai: dirbtiniai neuroniniai tinklai (ANN), atraminių vektorių mašinos (SVM) ir atsitiktiniai miškai (RF).

Tyrimo problemos aktualumas

Suvedinėjant rungtynių rezultatus ranka yra didelė tikimybė įvesti klaidą dėl žmogiškųjų faktorių, pavyzdžiui, nuovargio. Be to tai užtrunka nemažai laiko, todėl yra naudinga rezultatų švieslentėje esančius skaitmenis (rezultatus) nuskaityti interaktyviai. Galima rasti nemažai straipsnių apie ranka rašytų ženklų (skaičių arba raidžių), automobilių numerių apdorojimą bei aptikimą vaizduose, tačiau skaitmenims sudarytiems iš LED segmentų aptikti skirtų straipsnių yra labai mažai.

Tyrimo objektas – rezultatų švieslentėje esančių skaitmenų atpažinimas, panaudojant filmavimo kamerą ir vaizdų apdorojimo programą.

Darbo tikslas – ištirti požymių išskyrimo ir klasifikavimo metodų, tinkančių LED segmentų skaitmenims atpažinti, galimybes bei interaktyviai nuskaityti rezultatų švieslentės duomenis.

Darbo uždaviniai – atlikti požymių išskyrimo ir klasifikavimo metodų literatūros analizę, sudaryti LED segmentų skaitmenų atpažinimo sistemą bei atlikti dirbtinio intelekto metodų taikymo LED segmentų atpažinimui galimybių tyrimą.

1 LITERATŪROS APŽVALGA

Daug dėmesio skiriama ranka rašytų skaitmenų ir raidžių bei automobilių numerių atpažinimui. Taip pat pavyko rasti skaitiklio rodmenų nuskaitymo problemos sprendimą, tačiau skaitmenims, sudarytiems iš LED segmentų, atpažinti skirtų straipsnių yra labai mažai.

1.1 Automobilių numerių aptikimas ir atpažinimas

Galima rasti nemažai darbų apie automobilių numerių automatines atpažinimo sistemas. Dažniausiai automobilių numerių atpažinimas susideda iš šių dalių: automobilių numeriai nufotografuojami arba nufilmuojami; vykdomas pirminis atvaizdo apdorojimas; aptinkami kraštai; nustatoma numerių lentelė; segmentuojami skaitmenys, esantys lentelėje; atpažįstamos automobilio numerio raidės bei skaičiai [1] [2] [3].

Cosmo H. Munuo ir Michael Kisangiri darbe [4] galima rasti visus aukščiau paminėtus automobilių numerių aptikimo ir atpažinimo žingsnius.

Pirminiam atvaizdo apdorojimui dažnai naudojama pilkų atspalvių skalė [5] [6]. Pilkų atspalvių skalėje vaizdo kontrasto dažnis apskaičiuojamas kiekvienai pozicijai iš šablono sukuriant naują vaizdą, apibrėžtą slenkstine verte. Vertė, esanti aukščiau slenkstinės vertės, atvaizduojama balta spalva, o vertė, kuri yra žemiau slenkstinės vertės – juoda spalva (žr. 1.1.1 pav.). Pilkosios skalės vaizdai susideda iš skirtingų pilkų verčių diapazono, kuris kinta nuo 0 iki 255.



a)

b)

1.1.1 pav. Spalvoto atvaizdo pakeitimas pilkų atspalvių atvaizdu (a) spalvotas automobilio atvaizdas; b) pilkų atspalvių automobilio atvaizdas)[4]

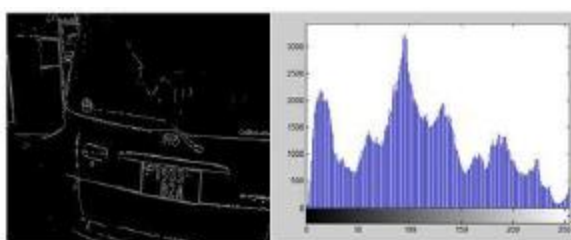
Kai atvaizdas yra pilkų atspalvių, galima jį binarizuoti [7]. Binarizuotas atvaizdas turi tik dvi reikšmes kiekvienam pikseliui, 0 ir 1, atitinkančias juodą ir baltą spalvas. Pilkosios skalės vaizdo pikselis turi aštuonis informacijos bitus, vadinasi, 256 pilkųjų verčių. Dvejetainė konversija (binarizacija) apima procesą, kurio metu pilkos skalės atvaizdas pakeičiamas binarizuotu vaizdu (žr. 1.1.2 pav.).



1.1.2 pav. Automobilio binarizuotas atvaizdas [4]

Toliau vykdomas kontūrų aptikimas. Kontūrus aptikti galima keliais būdais.

Naudojant *Sobel* kontūrų detektorių [4], apskaičiuojamas vaizdo ryškumo perėjimas kiekviename taške (žr. 1.1.3 pav.). Naudojama 3x3 matrica, dar kitaip vadinama sąsūkų branduolių pora, kuri skirta apskaičiuoti išvestines horizontalia ir vertikalia kryptimis.

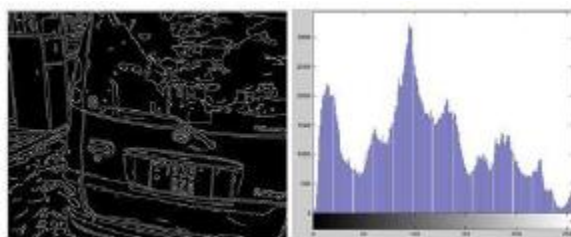


a)

b)

1.1.3 pav. Kontūrų aptikimas su *Sobel* operatoriumi (a) *Sobel* kontūrų detektorius; b) *Sobel* histograma) [4]

Naudojant *Canny* kontūrų detektorių, gaunami geresni rezultatai nei su *Sobel* operatoriumi, nes *Canny* naudoja du atmetimo slenksčius aptinkant „stiprius“ ir „silpnus“ kontūrus (žr. 1.1.4 pav.).



a)

b)

1.1.4 pav. Kontūrų aptikimas su *Canny* operatoriumi (a) *Canny* kontūrų detektorius; b) *Canny* histograma) [4]

Aptikus kontūrus, ieškoma automobilio numerio lentelė, kurioje esantys skaitmenys ir raidės yra segmentuojami. Tuomet vykdomas skaitmenų ir raidžių atpažinimas.

P. Sai Krishna požymiams išgauti naudoja CCA [8]. Panaudojus CCA, gautas 90 % atpažinimo tikslumas.

1.2 Banknoto serijos numerio aptikimas ir atpažinimas

Banknoto serijos numerį aptikti ir atpažinti panaudoti HOG požymiai ir KNN klasifikatorius. Remiantis HOG metodu, vienaženklis skaičiaus vaizdas yra padalintas į mažas susijusias stačiakampes sritis, pavadintas R – HOG blokais. Kiekvienas blokas turi savyje 2×2 celių grupę, kurios dydis yra parenkamas remiantis požymių skaičiumi, kurio pakaktų apibūdinti objektą (skaičių). Šie blokai gali būti apibūdinti trimis parametrais t. y. celių skaičiumi per bloką, pikselių skaičiumi per celę ir histogramos kanalų skaičiumi. Ramanjit Kaur darbe apie eilės numerio atpažinimą banknote (žr. 1.2.1 pav.), panaudojant HOG požymių ištraukimo metodą ir KNN klasifikatorių [9], pasirenkamas celės dydis yra 4×4 (žr. 1.2.2 pav.).

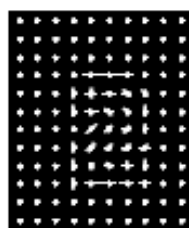


1.2.1 pav. Banknoto serijos numeris [9]

Kiekvienam pikseliui, esančiam celėje, gradiento krypčių histograma yra apskaičiuota, įvertinant gradientą ir orientacijos skaičiavimą.



a)



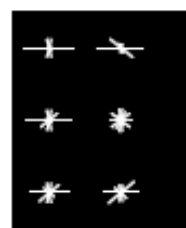
Feature length = 1764

b)



Feature length = 324

c)



Feature length = 36

d)

1.2.2 pav. HOG požymių ištraukimas su skirtingais celės dydžiais: a) dvejetainis skaitmenis „2“ vaizdas; b) celės dydis $[2 \ 2]$; c) celės dydis $[4 \ 4]$; d) celės dydis $[8 \ 8]$ [9]

Atlikus eksperimentinį tyrimą [9], gautas 81,25 % skaičių atpažinimo tikslumas ir 92,8 % raidžių atpažinimo tikslumas. Bendras atpažinimo tikslumas panaudojus HOG požymius ir KNN klasifikatorių banknoto numeriui atpažinti lygus 83,81 %.

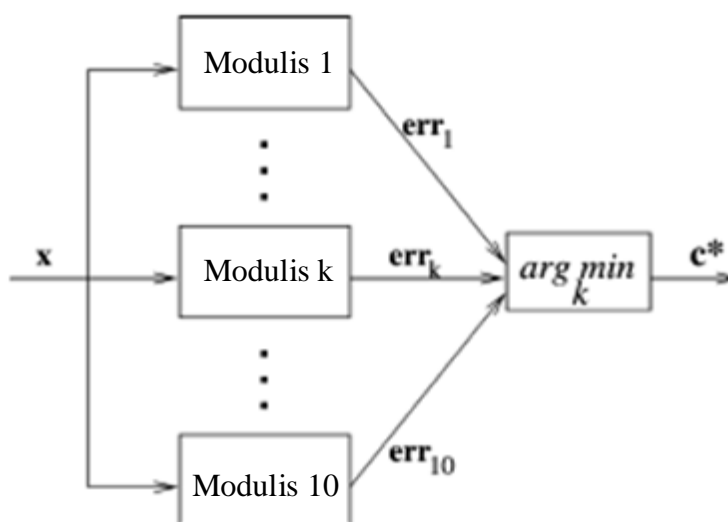
1.3 Ranka rašytų ženklų atpažinimas

Ranka rašytiems ženkliams atpažinti dažnai naudojami PCA [11] ir HOG [12] požymiai bei ANN [10] ir SVM [26] klasifikatoriai.

1.3.1 Dirbtinių neuroninių tinklų taikymas ranka rašytiems skaitmenims atpažinti

Bailing Zhang, Minyue Fu ir Hong Yan darbe [11] aprašoma modulinė klasifikavimo schema, naudojama išspręsti ranka rašytų skaitmenų atpažinimo problemą. Naudojama 10 tinklų (modulių) fiksuoti skirtingiems požymiams dešimtyje skirtingų klasių. Kiekvienas tinklas yra lokalių PCA maišytas modulis.

Vietoj vieno neuroninio tinklo klasifikatoriaus mokymo dešimčiai išvesčių (10 skirtingų skaitmenų), siūloma 10 maišytų modulių apibudinti 10 skaitmenų klasių (žr. 1.3.1.1 pav.). Mokymų metu, kiekvienas modelis priima tik savo klasės mokymo pavyzdžius.



1.3.1.1 pav. Modulinė klasifikavimo sistema [11]

Testavimo atvaizdas x yra visų 10 modulių įvestis, o svarbi problema yra apibrėžti modulio atitikimo rezultatus. Modulio mišinyje, įvertinamos visos rekonstrukcijos klaidos lyginant su pradiniais duomenimis, kurios atsiranda daugeliui modulių jungiantis į vieną. Mažiausia klaida nusako, kaip gerai šis susimaišymo modelis atitinka pradinis duomenis.

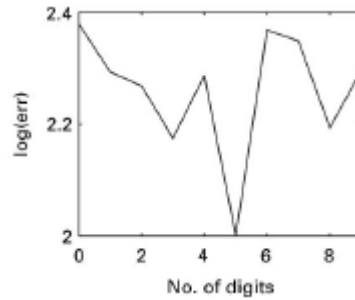
Ranka rašytų ženklų atpažinimo pavyzdys, naudojant PCA požymius ir ANN klasifikatorių, pateiktas 1.3.1.2 paveikslėlyje.



a)



b)

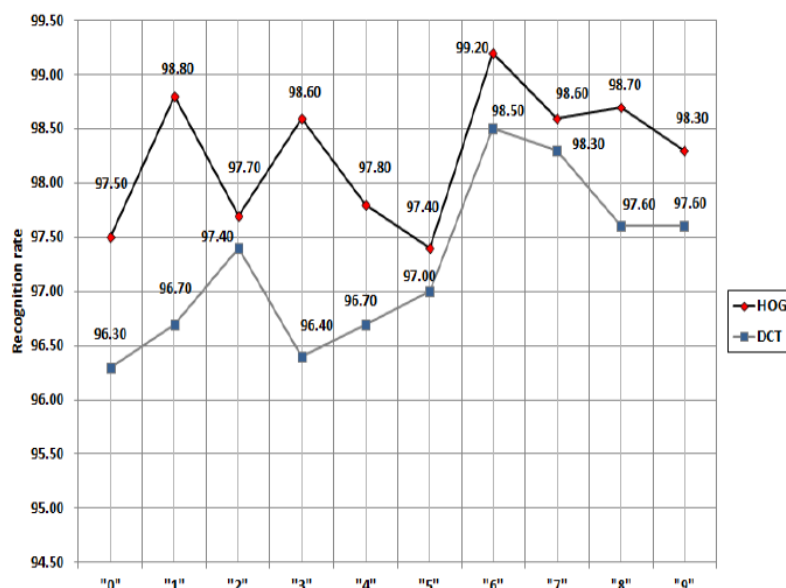


c)

1.3.1.2 pav. Ranka rašyto skaitmens atpažinimas, panaudojant PCA požymius ir ANN klasifikatorių (a) dvejetainis įvesties skaitmuo „5“; b) dešimt apdorotų šablonų; c) atkūrimo klaidos) [11]

Ranka rašytų ženklų atpažinimo tikslumas, naudojant PCA požymius ir ANN klasifikatorių, 98,6 % su mokymo imtimi ir 97,8 % su testavimo imtimi.

Ahmed Lawgali darbe apie HOG ir DCT taikymą ranka rašytiems skaitmenims atpažinti [12], atliktas tyrimas, lyginantis HOG su DCT (žr. 1.3.1.3 pav.). Nustatyta, kad HOG požymiai yra tikslesni. Orientuotų gradientų histogramoms taip pat būdingas atsparumas apšvietimo pokyčiams bei objekto pasislinkimui. Požymiai klasifikuojami su dirbtiniu neuroniniu tinklu, kuris turi 160 paslėptų sluoksnių.



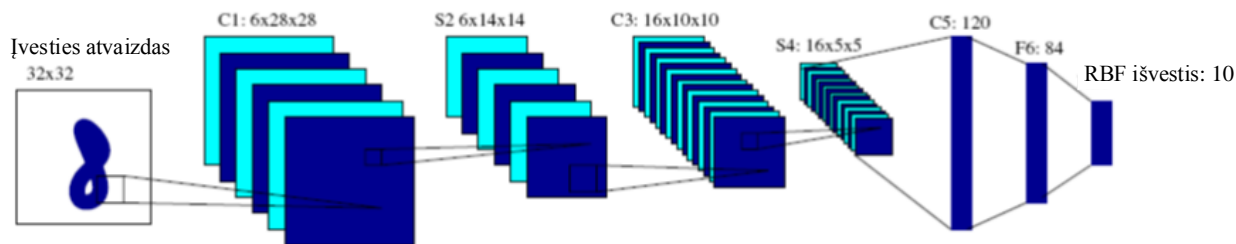
1.3.1.3 pav. Skaitmenų atpažinimo tikslumo priklausomybė nuo naudojamų požymių išskyrimo metodo [12]

Panaudojus HOG požymius gautas 98,26 % atpažinimo tikslumas, o panaudojus DCT pasiektas 97,25 % atpažinimo tikslumas (visų skaitmenų atpažinimo tikslumo vidurkis).

1.3.2 Sąsūkinis neuroninis tinklas ranka rašytiems ženklams atpažinti

Dar viena neretai naudojama neuroninio tinklo rūšis yra sąsūkinis neuroninis tinklas (angl. *Convolutional Neural Network*). Šis neuroninis tinklas turi tris pagrindinius požymius [13]:

- Lokaliųjų požymių išskyrimas. Kiekvienas iš neuronų gauna signalą iš prieš tai buvusio sluoksnio lokalaus lauko. Taip ištraukiami lokalieji požymiai (žr. 1.3.2.1 pav.). Po to, kai ištraukiamas požymis, jo buvimo vieta tampa nesvarbi, nes vieta jau yra žinoma, atsižvelgiant į kitus požymius.
- Sluoksnių suformavimas. Kiekvienas sluoksnis turi keleto plokštumų derinius (žr. 1.3.2.1 pav.). Tame pačiame sluoksnyje vienoje plokštumoje esantys neuronai turi identiškus svorių rinkinius. Taip pat neuronai yra sujungti su visomis prieš tai esančio sluoksnio sritimis. Tokia forma suteikia tinklui sudėtingą struktūrą, kuri turi nemažai privalumų.
- Požymių atrinkimas. Po kiekvieno sąsūkinio sluoksnio yra sluoksnis, kuris atlieka požymių atranką. Tokia operacija sumažina išeinančiojo signalo jautrumą nedideliems iškraipymams bei rakurso pakeitimui.



1.3.2.1 pav. Šąsūkinio neuroninio tinklo taikymas skaitmenims atpažinti [13]

Čia: $C1$, $C3$, $C5$ – šąsūkiniai sluoksniai, 5×5 – šąsūkinė matrica, $S2$, $S4$ – paslėpti sluoksniai, $F6$ – pilnai sujungtas sluoksnis.

Įvesties sluoksnyje paduodamas skaitmens vektorius. Nagrinėjami vektoriai yra centruojami paveiksle, sudarytame iš 32×32 pikselių.

Veikimo algoritmas [13]:

1. Pirmasis paslėptas sluoksnis yra šąsūkinis. Jį sudaryto 6 požymių žemėlapiai (28×28 pikseliai).

Kiekvienas požymio žemėlapiio elementas sujungiamas su įvesties sluoksniu 5×5 dalimi. Todėl kiekvienas elementas turi 16 apmokomų koeficientų ir apmokomą poslinkį.

2. Požymių atrankos sluoksnis – tai antras paslėptas sluoksnis, kurį sudaro 14×14 šeši žemėlapiai. Kiekvienas šio sluoksniuio elementas yra sujungtas su prieš tai esančio šąsūkinio sluoksniuio 2×2 dalimi, kitaip tariant, su pirmuoju paslėptuoju sluoksniu. Šio sluoksniuio uždavinys – požymių išrinkimas, todėl naudojamas vienas svorių rinkinys ir postūmio koeficientas. Šis sluoksnis turi 5880 ryšius bei 12 apmokomų parametru.

Toliau algoritmas kartojasi. Reikėtų atkreipti dėmesį į tai, kad kiekvienas individualiai pats sprendžia dėl antrojo ir trečiojo sluoksniuų požymių žemėlapių jungimo.

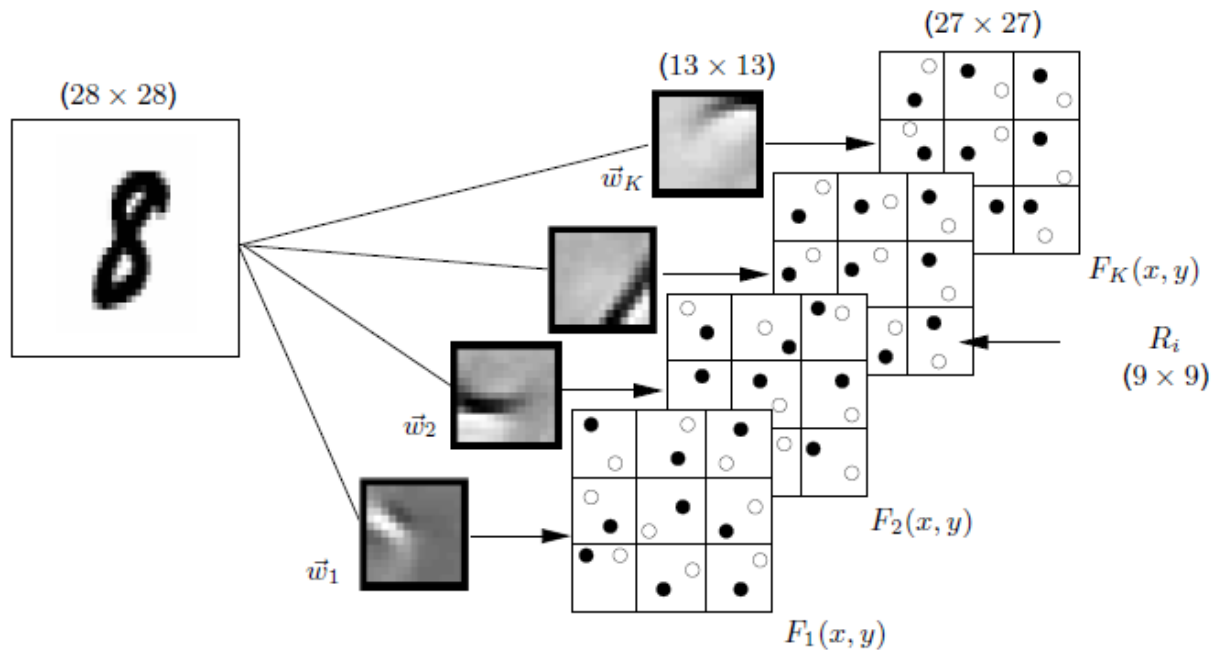
3. Išvesties sluoksnis yra paskutinis. Jo matmenys priklauso nuo uždavinio. Minėtame pavyzdyje nagrinėjami skaitmenys nuo nulio iki devynių (viso 10 skaitmenų), todėl bus 10 tinklo išvesčių.

1.3.3 Atraminų vektorių mašinu taikymas ranka rašytiems skaitmenims atpažinti

Ranka rašytiems skaitmenims atpažinti naudojami ne tik neuroniniai tinklai, bet ir atraminų vektorių mašinos [14].

Požymių išskyrimo būdo schematinis vaizdas pateiktas 1.3.3.1 paveikslėlyje. Pirmiausiai, atvaizdu koeficientai yra apskaičiuojami, naudojant PCA požymius. Koeficientai apskaičiuojami kiekvienam pikseliui ir kiekvienai pagrindinei funkcijai. Antra, kiekvieno koeficiento atvaizdas

yra padalintas į taisyklingus nepersidengiančius regionus ir kiekvienam regionui nurodomos minimalios ir maksimalios įvesties vertės. Šios vertės yra tolesnio klasifikavimo žingsnio požymiai.



1.3.3.1 pav. PCA požymių išskyrimo schema, naudojama ranka rašytiems skaitmenims atpažinti [14]

Pirmasis požymių ištraukimo etapas yra grindžiamas dydžio N^2 bazinių funkcijų w_j rinkiniais, kurie yra pritaikyti duoto atvaizdo I kiekvienam vektorių $P(x, y)$ rinkiniui.

Koeficientų atvaizdai F_j yra visų įvesties atvaizdų I bazinių funkcijų w_j koeficientai a_j . Norint gauti koeficiento atvaizdus, kurie turi tą patį dydį kaip pradinis atvaizdas I , reikia didinti I , nustatant esančių už atvaizdo pikselių vertę į fiksuotą vertę.

Paskutinis kiekvieno įvesties atvaizdo požymių vektorius, kuris yra duotas kaip įvesties klasifikatorius, susideda iš visų sričių minimalių ir maksimalių verčių, atsižvelgiant į visus koeficientų vaizdus.

Tyrimas atliktas su 60000 mokymo ir 10000 testavimo ranka rašytų skaitmenų, kurių kiekvieno dydis yra 28×28 . Gauti vaizdų atpažinimo rezultatai, naudojant PCA požymius ir atraminių vektorių mašinas, pateikti 1 lentelėje.

1 lentelė. Klaidų vidurkiai, naudojant PCA požymius skaitmenims atpažinti [14].

	Klaidų lygio vidurkis, %	Skaitmuo										Bandymo komplekto klaidų lygis, %
		0	1	2	3	4	5	6	7	8	9	
Neatpažinti duomenys	2,95 (±0,17)	3293	2293	4580	4731	4041	4223	3676	3747	5246	4930	0,80

1.4 Interaktyvus skaitiklio rodmenų nuskaitymas

Šiame darbe siūlomas naujas skaitmenų atpažinimo algoritmas, pavadintas Hausdorfo atstumu skaitiklio rodmenims (HD_MR) [14]. Kai nuskaitymas yra teisingai atpažintas, kiekvienas vienaženklis skaičius nuskaitytas yra priimtas ir kaupiamas kaip paskutinis teisingas nuskaitymas. Kai HD algoritmas negali atpažinti skaičiaus, jis naudoja paskutinį atpažintą skaičių.

Yra dvi reikalingos sąlygos: vaizdo kameros padėtis ir apšvietumas. Elektros skaitiklis yra 22 cm nuo grindų, vaizdo kamera – 29 cm nuo grindų, o atstumas tarp skaitiklio ir vaizdo kameros 5,5 cm.

Pirminis apdorojimas susideda iš skaičių vietos atvaizde nustatymo (atvaizdo vieta), atskiriant skaičius vieną nuo kito (segmentavimas) ir atvaizdų pakeitimo nespaltotu vaizdu (juoda – balta). Pirminis apdorojimas yra pagrįstas atspalvio, segmentavimo ir vertės (HSV) spalvotu modeliu. Metodas palankus dėl spalvų kombinacijos. Skaitiklyje rodomi skaičiai yra raudonos spalvos, o aplink juos – raudonos aikštelės (stačiakampis rėmelis). Pirmiausia pikseliai suklasifikuojami į raudonus ir ne raudonus. Galiausiai segmentuoti skaičiai yra binarizuojami panaudojus HSV modelį, t. y. klasifikuojami į baltus ir ne baltus pikselius.

Po segmentavimo ir perkėlimo į binarinę formą prasideda atpažinimo procesas. Procedūra prasideda nuo nepakrypusių skaičių. Naudojamas reguliarus HD, kad atpažintų skaičius ir kauptų atpažinimo rezultatus. Kiti procedūros žingsniai yra pritaikyti kiekvienam vienaženkliai skaičiui iš dešinės į kairę, kol užbaigtas skaitymas nėra atpažintas.

Pirmame žingsnyje skaitmuo atpažįstamas, jei viename kadre yra nepakreipti skaičiai.

Antrame žingsnyje procedūra nurodo slenkstį pakreiptam skaičiui, kuris turi tapti nepakreiptu. Slenkstis yra nustatytas 85% apatinėje dalyje (žr. 1.4.1 pav.). Kitaip tariant, sistema bus tik bandymas atpažinti skaičių, kai jis pasiekia 85% matomumą. Ši priemonė padidina HD kiekvieno atskiro skaičiaus atpažinimo tikslumą.

Trečiame žingsnyje sistema bando atpažinti vaizdą, kai viršijama slenkstinė riba. Taikant šią strategiją, atpažinimo laikas sumažėja.

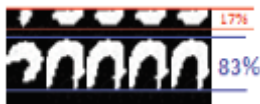
Kai visi atitinkamo skaitymo kadrai atpažįstami, ankstesnė atpažinimo vertė naudojama įvertinti faktinį skaitymo teisingumą. Nuskaitymas turi būti didesnis arba bent lygus paskutinei saugomai vertei.



a)



b)



c)

1.4.1 pav. Pakreipti skaitiklio rodmenys (a) pakreipti paskutiniai trys skaičiai; b) pakreipti paskutiniai keturi skaičiai; c) pakreipti visi skaičiai) [14]

Ekspertimentų rezultatai rodo, kad HD_MR gali pasiekti 99,9% atpažinimo tikslumą, net jei skaičiai yra pakreipti. Didžiausias atpažinimo laikas yra lygus 31 ms, taigi, siūlomas metodo tyrimų rezultatai įrodo, kad metodas yra efektyvus ir veikiantis realaus laiko sistemose.

2 TEORINIS METODŲ PAGRINDIMAS

2.1 Vaizdo apdorojimo metodų analizė

Kiekvieną vaizdą galima suskaidyti į mažesnes sudaromąsias, kurios vadinamos pikseliais. Pikselių vaizde gali būti labai daug, o kiekvieną iš jų apdorojant prarandama daug laiko, taip mažinama viso proceso sparta ir efektyvumas, todėl itin svarbus yra esminių požymių aptikimas (angl. *feature detection*). Požymių aptikimas laikomas vaizdo apdorojimo metodų pagrindiniu etapu, nuo kurio priklauso viso vaizdo apdorojimo proceso efektyvumas. Sąvoka *detektorius* skirta aprašyti požymių aptikimą, o *deskriptorius* – požymiams charakterizuoti, kuris vėliau panaudojamas objekto aptikimui ir atitikimui kituose vaizduose [15].

Yra nemažai požymių išskyrimo metodų, o pagrindiniai nagrinėjami šiame darbe:

- PCA (angl. *Principal Component Analysis*) [11];
- HOG (angl. *Histogram of Oriented Gradients*) [12].

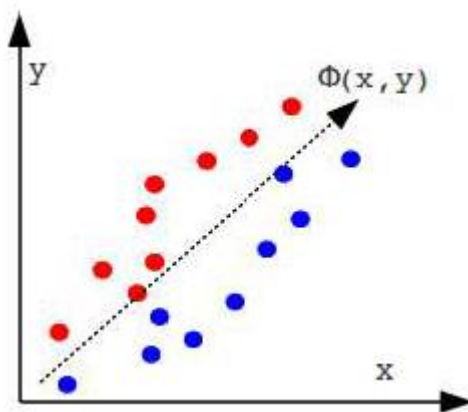
Atlikus požymių išskyrimą, reikia išskirtus požymius klasifikuoti. Šiame darbe analizuojami šie požymių klasifikavimo metodai:

- ANN – dirbtiniai neuroniniai tinklai (angl. *Artificial Neural Networks*) [10];
- SVM – atraminių vektorių mašinos (angl. *Support Vector Machines*) [26];
- RF – atsitiktiniai miškai (angl. *Random Forests*) [27].

Kiekvienas prieš tai paminėtas požymių išskyrimo bei klasifikavimo metodas plačiau analizuojamas šio skyriaus poskyriuose.

2.2 Požymių išskyrimo metodas PCA

Principinių komponentių analizė (angl. *Principal Component Analysis*) – tai vienas iš statistinių metodų, kuris dažnai naudojamas signalų ir vaizdų apdorojimui [16]. Principinių komponentių analizė skirta dimensijos mažinimui bei esminių savybių radimui, atsisakant nereikšmingų parametrų. Kitaip sakant, PCA yra matematinė procedūra, kuria siekiama sumažinti dimensiją ir išgauti principines komponentes (žr. 2.2.1 pav.) iš daugiamačių duomenų. Šios procedūros metu ieškoma kryptis, kurioje duomenų variacija yra didžiausia [17].



2.2.1 pav. Principinė komponentė [18]

Principinių komponentių analizė gali būti apibūdinama kaip tam tikrų n įvesties vektorių (kintamųjų) rinkinio, kurių ilgis K yra vienodas, transformacija $x = (x_1, x_2, \dots, x_n)^T$ [19]. Ši transformacija yra naudojama (2.2.1) formulėje:

$$y = A(x - m_x) \quad (2.2.1)$$

Ši transformacija leidžia suformuoti paprastą formulę (1), tačiau reikėtų nepamiršti, kad kiekviena x vektoriaus eilutė susideda iš K verčių, kurios priklauso nuo vienos įvesties. Vektorius m_x formulėje (2.2.1) yra visų įvesties kintamųjų vidutinės vertės (angl. *mean values*) vektorius, kuris yra išreiškiamas [19], kaip:

$$m_x = E\{x\} = \frac{1}{K} \sum_{k=1}^K x_k \quad (2.2.2)$$

Matrica A formulėje (2.2.1) yra nustatoma pagal kovariacinę matricą C_x . A matricos eilutės yra suformuotos iš C_x savųjų vektorių (angl. *eigenvectors*) e , o išdėstoma atitinkamų savųjų reikšmių mažėjančia tvarka. Tai įvertinus, C_x matrica gali būti išreikšta [19], kaip

$$C_x = E\{(x - m_x)(x - m_x)^T\} = \frac{1}{K} \sum_{k=1}^K x_k x_k^T - m_x m_x^T \quad (2.2.3)$$

Įvesties kintamųjų vektorius x yra n dimensijos, todėl akivaizdu, kad C_x dydis yra $n \times n$. Elementai $C_x(i, i)$ esantys pagrindinėje įstrižainėje [19]:

$$C_x(i, i) = E\{(x_i - m_i)^2\} \quad (2.2.4)$$

yra x nuokrypiai, o kitos vertės $C_x(i, j)$ nusako kovariaciją tarp įvesties kintamųjų x_i ir x_j [19]:

$$C_x(i, j) = E\{(x_i - m_i)(x_j - m_j)\} \quad (2.2.5)$$

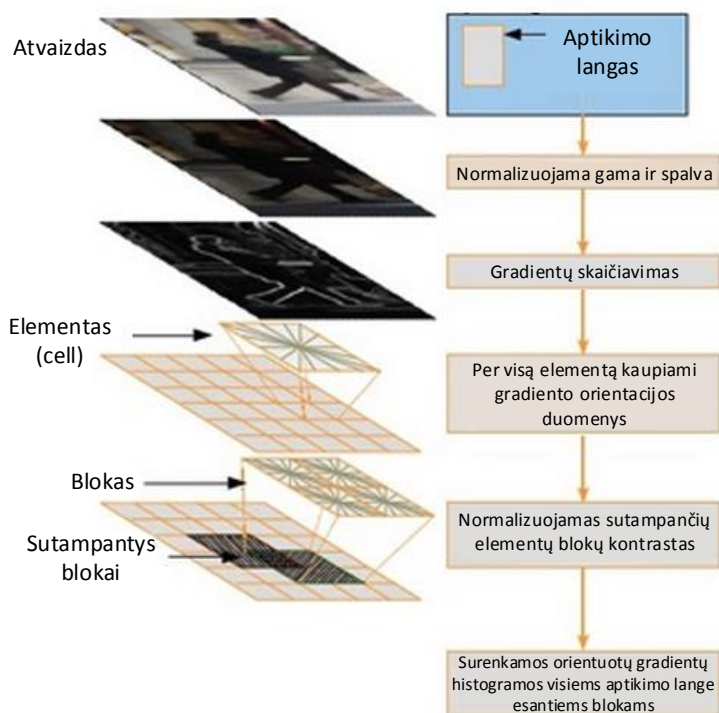
PCA inversiją galima užrašyti kaip [19]:

$$x = A^T y + m_x \quad (2.2.6)$$

PCA apdorojimo sparta ir tikslumas priklauso nuo naudojamų šablonų ir principinių komponentių skaičiaus. Metodas gana jautrus apšviestumui, nes dažniausiai šablonai yra suformuojami esant momentiniam apšviestumui. Dėl nesudėtingos realizacijos ir matematinių veiksmų, metodas yra palyginti spartus [20].

2.3 Požymių išskyrimo metodas HOG

Orientuotų gradientų histogramos deskriptoriaus realizavimas galimas dalijant atvaizdą į mažas susijusias sritis, vadinamas celėmis, ir kiekvienai celei sudarant gradientų krypčių ar kraštinių orientacijos histogramas pikseliams (žr. 2.3.1 pav.). Iš histogramų sudaromas požymių vektorius, kuris vėliau ir yra naudojamas. Norint padidinti tikslumą, gali būti normalizuotas kontrastas lokalioms histogramoms, apskaičiuojant intensyvumą didesnei atvaizdo sričiai, vadinamai bloku. Vėliau panaudojant šias vertes, normalizuojamos visos celės bloko viduje.



2.3.1 pav. HOG įvykdymo žingsniai [21]

Remiantis aukščiau minėtu paaiškinimu, HOG metodo požymių ištraukimas susideda iš daugelio orientuotų gradientų histogramų lokalizuotuose atvaizdo zonose. Šio metodo požymių ištraukimo skaičiavimas yra skirstomas į tris žingsnius:

- 1) gradiento skaičiavimą;
- 2) histogramų generavimą;
- 3) histogramų normalizavimą.

2.3.1 Gradiento skaičiavimas

Visų pirma, pirmos eilės diferencialiniai koeficientai $G_x(i,j)$ ir $G_y(i,j)$ apskaičiuojami remiantis šiomis lygtimis [22]:

$$\begin{cases} G_x(i,j) = f(i+1,j) - f(i-1,j); \\ G_y(i,j) = f(i,j+1) - f(i,j-1). \end{cases} \quad (2.3.1.1)$$

Čia $f(i,j)$ reiškia intensyvumą taške (i,j) .

Tada skaičiuojamų gradientų dydis m ir kryptis θ yra apskaičiuojama atitinkamai pagal šias išraiškas [22]:

$$m(i,j) = \sqrt{G_x(i,j)^2 + G_y(i,j)^2}; \quad (2.3.1.2)$$

$$\theta(i,j) = \arctan\left(\frac{G_x(i,j)}{G_y(i,j)}\right). \quad (2.3.1.3)$$

2.3.2 Histogramų generavimas

Po to kai yra gautos m ir θ vertės, histogramos yra generuojamos taip:

- 1) nustatoma klasė, kuriai priklauso koeficientas $\theta(i,j)$;
- 2) padidinama pirmajame punkte apskaičiuotos vertės klasė;
- 3) kartojamos aukščiau paminėtos operacijos visiems gradientams, kurie priklauso elementui.

Siekiant sumažinti triukšmų poveikį, dviejų kaimyninių klasių vertės yra padidinamos. Padidinimas n parodo a klasės indeksą, kuriam priklauso $\theta(i,j)$ ir $n+1$ gali būti klasė, kuri yra artimiausia klasei n . Padidintos vertės koeficientai m_n ir m_{n+1} yra apskaičiuojami taip [22]:

$$\begin{cases} n = \left\lceil \frac{b\theta(i,j)}{\pi} \right\rceil; \\ m_n = (1-a)m(i,j); \\ m_{n+1} = am(i,j). \end{cases} \quad (2.3.2.1)$$

Kur b parodo klasių bendrą skaičių, a yra parametras proporcingas dydžiui $m(i,j)$, kuris yra apibrėžtas kaip atstumas nuo $\theta(i,j)$ iki klasių n ir $n+1$ [22]:

$$a = \frac{b}{\pi} \left(\theta(i,j) \bmod \left(\frac{\pi}{b} \right) \right). \quad (2.3.2.2)$$

2.3.3 Histogramų normalizavimas

Galiausiai kombinuojant visas suformuotas histogramas, priklausančias blokui, sudaroma viena didelė histograma, kuri susideda iš kelių elementų. Norint sumažinti rezultatų ryškumo ir

kontrasto pakitimus, yra priimta $L1$ norma. Turint didelę jungtinę histogramą, galima normalizuoti rezultatą [22]:

$$v = \frac{V_k}{\|V_k\| + \varepsilon}. \quad (2.3.3.1)$$

Čia V_k yra vektorius, atitinkantis kombinuotą bloko histogramą, ε yra maža konstanta, v yra normalizuotas vektorius, kuris yra paskutinis HOG metodo požymis.

2.4 Požymių klasifikavimo metodas ANN

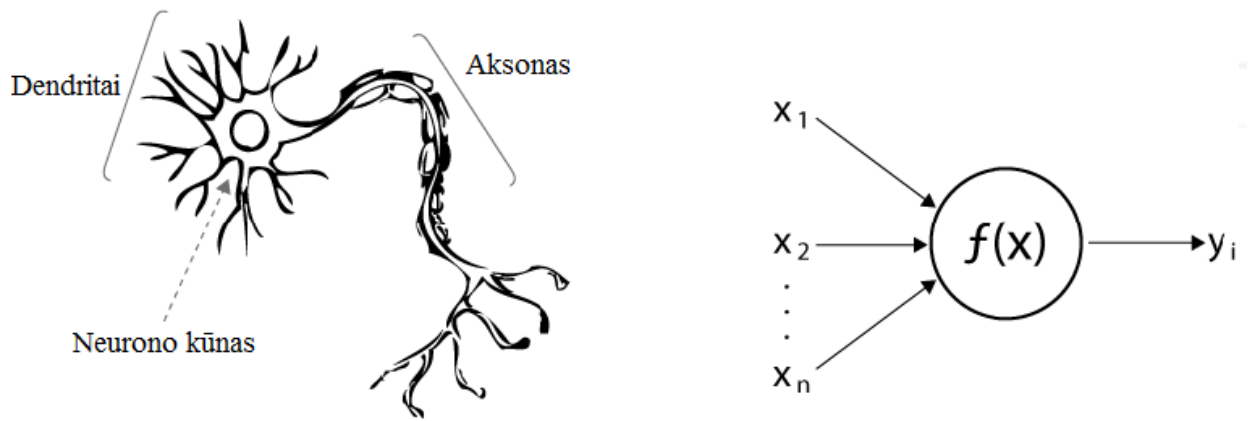
Dirbtinis neuroninis tinklas yra matematinis modelis, kuris pagrįstas žmogaus smegenų funkcionavimu. Žemiau paaiškinamas biologinis neuroninis tinklas, kurį imituoja ANN bei dirbtinio neuroninio tinklo struktūra.

2.4.1 Biologinis neuroninis tinklas

Nervų sistemą sudaro nervinis audinys, kurio pagrindinis struktūrinis ir funkcinis vienetas yra neuronas (nervinė ląstelė). Nors neuronai būna labai įvairūs morfologiškai ir funkciškai, tačiau jų sandaros schema yra panaši. Bet kuris neuronas turi šias pagrindines dalis (žr. 2.4.1.1 pav., kairėje) [23]:

- somą (kūną);
- dendritus;
- aksoną.

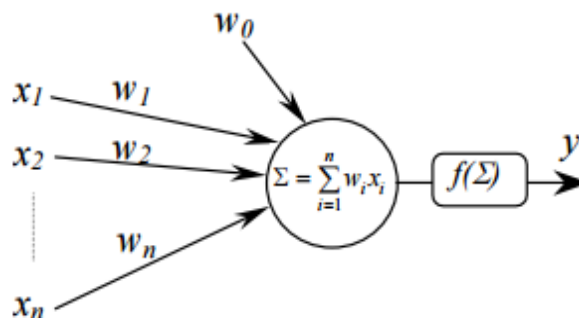
Soma – tai pagrindinė neurono kaip ląstelės dalis, kurioje išsidėsto branduolys ir kitos organelės. Nuo somos atsišakoja ilgos plonos ataugos – dendritai ir aksonai. Dendritų skaičius gali būti įvairus (dešimtys), dažnai jie būna labai šakoti, tačiau nenutolsta labai toli nuo neurono somos. Ant dendritų paprastai būna daug sinapsių (žr. 2.4.2.2 pav., kairėje), per kurias ateina informacija iš kitų neuronų (dendritai – „priimančioji“ neurono dalis). Aksonas būna tik vienas, tačiau netoli savo pabaigos dauguma aksonų šakojasi. Juo informacija yra perduodama iš neurono kitai ląstelei (pvz., kitam neuronui, raumeninei ląstelei ir kt.).



2.4.1.1 pav. Biologinis (kairėje) ir dirbtinis (dešinėje) neuronas [23]

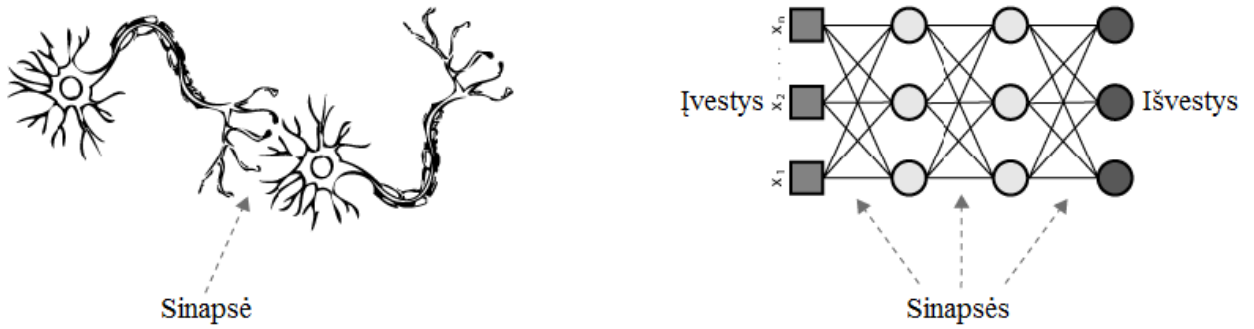
2.4.2 Dirbtinis neuroninis tinklas

Visa ANN metodologija siejama su „neuronų“ sąvoka. Neuronas gauna keletą įvesties reikšmių x (žr. 2.4.1.1 pav., dešinėje). Tai gali būti viso neuroninio tinklo įvesties reikšmės arba kitų tinklo neuronų išvestys. Kiekviena įvesties jungtis turi savo perdavimo koeficientą (svorį) w , kurie parodo biologinio neurono sinapsių efektyvumą. Kiekvienas neuronas turi savo slenksčio reikšmę. Neurono sužadavimo reikšmė formuojama skaičiuojant svorinę įvesties signalų sumą ir atimant slenksčio reikšmę. Pagal sužadavimo signalą, naudojant neurono perdavimo funkciją skaičiuojama neurono išvesties reikšmė (žr. 2.4.2.1 pav.) [24].



2.4.2.1 pav. Dirbtinio neurono sandara [24]

Dirbtinis neuroninis tinklas gali turėti vieną ar keletą paslėptų sluoksnių. Kiekvienas paslėptas sluoksnis yra sudarytas iš aktyvacijos funkcijų, kurios kontroliuoja neurono signalo perdavimą į kitą sluoksnį (žr. 2.4.2.2 pav., dešinėje). Paslėptas sluoksnis yra sudarytas iš regresijos, kuri apdoroja įvesties informaciją į netiesinius išvesties duomenis. Todėl, jei daugiau kaip vienas neuronas yra panaudotas, kad sudarytų ANN, gali būti atliekamos netiesinės koreliacijos. Dėl netiesiškumo tarp įvesties ir išvesties, kai kurie autoriai prilygina paslėptus ANN sluoksnius „juodai dėžei“.



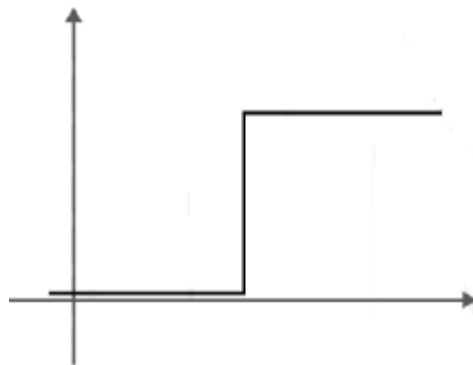
2.4.2.2 pav. Biologinės (kairėje) ir ANN (dešinėje) sinapsės [23]

Bendras ANN technikos tikslas yra pagrįstas aktyvacijos funkcijomis, kurios priima kažkokią įvestį (parametrą) ir suteikia išvestį (atsakymą). Yra kelios tipiškios aktyvacijos funkcijos, panaudotos, kad sudarytų ANN:

- Slenkstinė funkcija [23]:

$$\varphi(r) = \{1, \text{jei } n \geq 0; 0, \text{jei } n < 0\} \quad (2.4.2.1)$$

$$\varphi(r) = \begin{cases} 1, & \text{jei } v \geq 0; \\ 0, & \text{jei } v < 0. \end{cases} \quad (2.4.2.2)$$

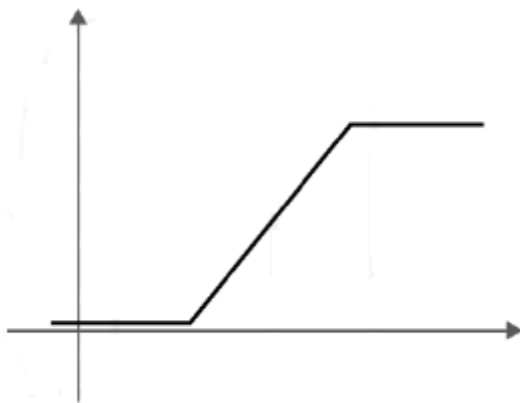


2.4.2.3 pav. Slenkstinė funkcija [23]

- Atribota tiesinė funkcija [23]:

$$\varphi(r) = \{1, \text{jei } r \geq 1/2; r, \text{jei } -1/2 < r < 1/2; 0, \text{jei } r \leq -1/2\} \quad (2.4.2.3)$$

$$\varphi(v) = \begin{cases} 1, & \text{jei } v \geq 1/2; \\ v, & \text{jei } -1/2 < r < 1/2; \\ 0, & \text{jei } v \leq -1/2. \end{cases} \quad (2.4.2.4)$$

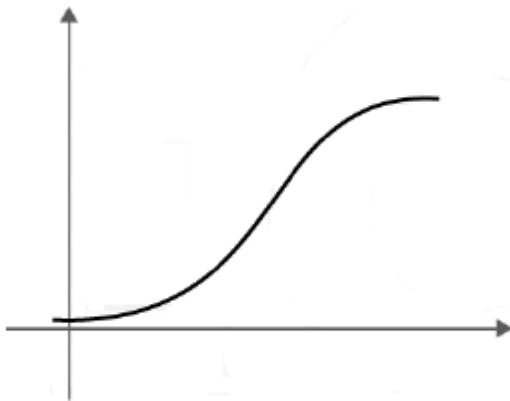


2.4.2.4 pav. Atribota tiesinē funkcija [23]

- Hiperbolinē tangento funkcija [23]:

$$\varphi(r) = \tan(n/2) = 1 - e(-n)/1 + e(-n) \quad (2.4.2.5)$$

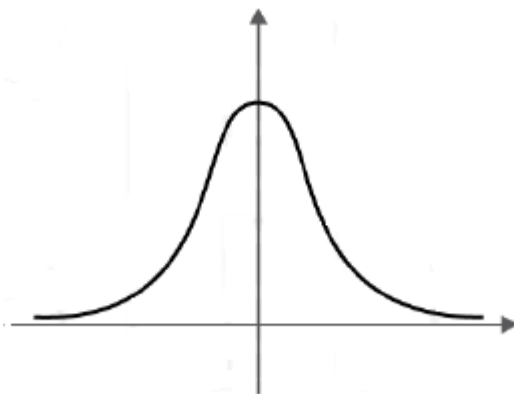
$$\varphi(v) = \tan\left(\frac{v}{2}\right) = \frac{1-e(-v)}{1+e(-v)} \quad (2.4.2.6)$$



2.4.2.5 pav. Hiperbolinē tangento funkcija [23]

- Gauso funkcija [23]:

$$\varphi(r) = e - (\varepsilon^v)^2 \quad (2.4.2.7)$$



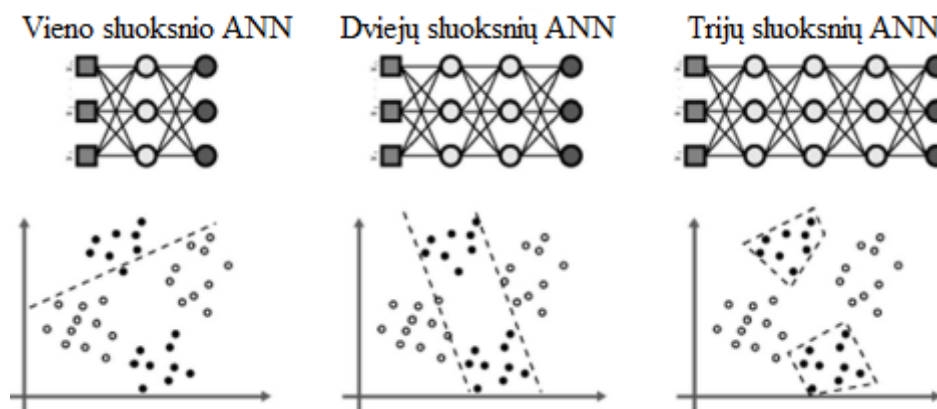
2.4.2.6 pav. Gauso funkcija [23]

2.4.3 Vienasluoksniai ir daugiasluoksniai dirbtiniai neuroniniai tinklai

Dirbtiniai neuroniniai tinklai gali būti:

- vienasluoksniai;
- daugiasluoksniai.

Vienasluoksniai dirbtiniai neuroniniai tinklai, dar kitaip vadinami perceptronais. Šie ANN turi vieną paslėptą sluoksnį. Toks ANN gali atlikti nesudėtingas matematinės operacijas.



2.4.3.1 pav. Sluoksnių skaičiaus įtaka MLP atpažinimo modeliui [23]

Daugiasluoksniai neuroniniai tinklai turi bent du paslėptus sluoksnius (žr. 2.4.3.1 pav.) ir yra naudojami siekiant suklasifikuoti daugiau skirtingų klasių. Šiame darbe naudojamas dirbtinis neuroninis tinklas su 70 paslėptų sluoksnių.

2.5 Požymių klasifikavimo metodas SVM

Atraminių vektorių mašinos (angl. *Support Vector Machines*) yra mokymo metodas, kuris buvo pradėtas naudoti 1995 metais. SVM pastaraisiais metais sukelia didelį mokslinės visuomenės dėmesį, kadangi šis metodas nesunkiai išsprendžia problemas tokiose srityse, kaip ranka rašytų skaitmenų atpažinimas [25], veido atpažinimas ir kt. Yra keletas pagrindinių formuliu, kurios apibrėžia šį metodą. Mokymo modelis aprašomas taip [26]:

$$\{(X_i, y_i)\}, i = 1, \dots, N, y_i \in \{-1, +1\}, X_i \in R^d; \quad (2.5.1)$$

Čia X_i yra mokymo bandinys, y_i yra žymė kiekvienos X_i vertės, o N – mokymo bandymų skaičius.

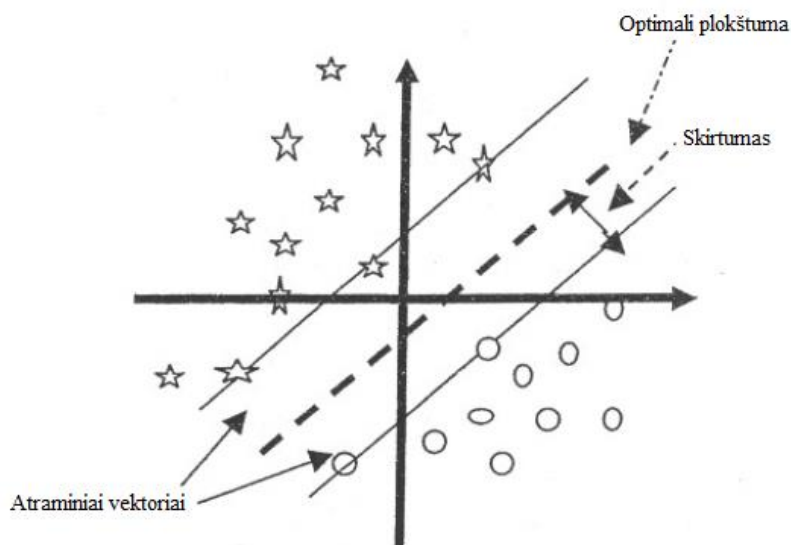
Pirmiausia atraminių vektorių mašinos, naudojant atvaizdavimą ϕ , gaunamus duomenis pažymi požymių erdvėje [26]:

$$\phi: R^d \rightarrow H. \quad (2.5.2)$$

Atvaizdavimas įgyvendinamas netiesiogiai pasinaudojant *Kernel* funkcija K . *Kernel* yra funkcijos, kurios požymių erdvėje suteikia kiekvienos vektorių poros sandaugą. Funkcijos K aprašomos taip [26]:

$$K(X_i, X_j) = \langle \phi(X_i), \phi(X_j) \rangle, \quad \forall X_i, X_j \in R^d; \quad (2.5.3)$$

Čia $\langle \phi(X_i), \phi(X_j) \rangle$ yra vidinė sandauga. Tada didelių matmenų požymių erdvėje H , reikia rasti optimalią plokštumą (žr. 2.5.1 pav.), maksimizuojant skirtumą tarp klasių, ir apribojant mokymų klaidų skaičių.



2.5.1 pav. Optimali plokštuma su maksimaliu skirtumu [26]

Sprendimo funkcija aprašoma taip [26]:

$$f(X) = \psi(\langle W, \phi(X) \rangle - b) = \psi(\sum_{i=1}^N y_i a_i \langle W, \phi(X) \rangle - b) = \psi(\sum_{i=1}^N y_i a_i K(X_i, X) - b); \quad (2.5.4)$$

kur

$$\psi = \begin{cases} 1 & \text{jei } u > 0 \\ -1 & \end{cases}; \quad (2.5.5)$$

ir

$$W = \sum_{i=1}^N y_i a_i \phi(X_i). \quad (2.5.6)$$

Jei a_i yra nelygus 0, tai atitinkamas bandinys X_i yra vadinamas atraminiu vektoriumi. Atraminų vektorių mašinos treniravimas yra koeficientų a_i radimas ($i = 1, \dots, N$), kuris gali būti pasiektas minimizuojant šią kvadratinę kaštų funkciją (angl. *cost function*):

Minimizuojama:

$$L_D(a) = \sum_{i=1}^N a_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N a_i a_j y_i y_j K(X_i X_j). \quad (2.5.7)$$

Atsižvelgiant į:

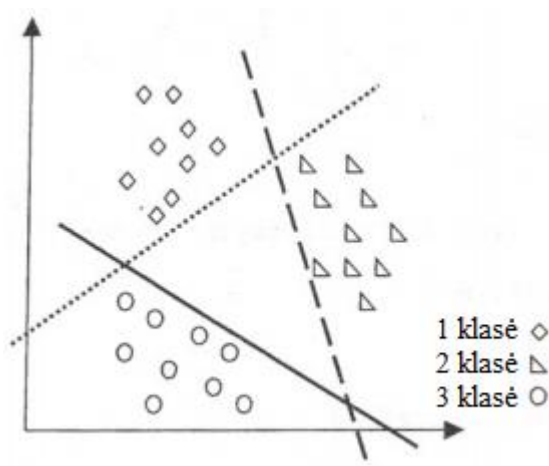
$$\begin{cases} 0 \leq a_i \leq C, i = 1, \dots, N; \\ \sum_{i=1}^N a_i y_i = 0. \end{cases} \quad (2.5.8)$$

Čia C yra parenkamas parametras. Parametras C atitinka aukštesnes vertes, priskiriamas mokymo klaidoms. Optimali plokštuma yra apibrėžiama W ir b sprendiniais.

2.5.1 Daugialypis SVM klasifikatorius

Norint atskirti keletą klasių, reikalinga panaudoti daugiau klasifikatorių ar plokštumų, kad būtų galima atskirti vieną objektą nuo kito. Tuomet naudojamas daugialypis SVM klasifikatorius.

Daugialypių SVM klasifikatorių projektavimo metodas yra vadinamas „vienas prieš kitą“ (angl. *one against the others*) (žr. 2.5.1.1 pav.).



2.5.1.1 pav. Daugialypio SVM metodo pavyzdys, skirtas spręsti trijų klasių problemą [26]

RBF *Kernel* yra panaudota bandymuose. Šis *Kernel* turi tik vieną parametą, ir tai yra labai populiaru. Jo bendroji forma yra [26]:

$$K(X_i, X_j) = \exp(-\theta \cdot \|X_i - X_j\|^2), \theta > 0, X_i, X_j \in R^d. \quad (2.5.1.1)$$

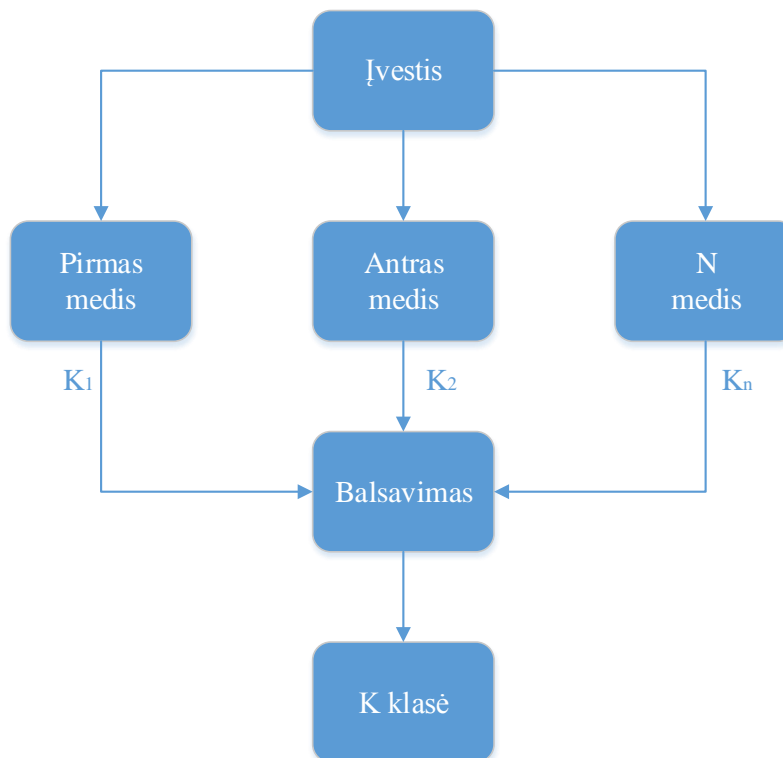
Čia X_i, X_j yra du įvesties vektoriai, B yra *Kernel* parametras.

2.6 Atsitiktinių miškų taikymas

Atsitiktinis miškas yra klasifikatorius, sudarytas iš medžio struktūros klasifikatorių rinkinio $\{h(x, \theta_k), k=1 \dots\}$, kur θ_k yra nepriklausomi, identiška pasiskirstę, atsitiktiniai medžiai ir kiekvienas medis vaizduoja įvesties x galutinę klasifikaciją (žr. 2.6.1 pav.) [27]. Paskutinė

kiekvieno medžio klasė yra sujungta ir nustatoma svorinių verčių „balsuojant“ (angl. *voted by weighted values*), kad sudarytų galutinį klasifikatorių.

Kiekvienas sprendimų medis yra apmokomas naudojant atsitiktinai iš mokymo imties atskyrus tam tikrą dalį duomenų, likę duomenys naudojami testavimui.



2.6.1 pav. Atsitiktinių miškų klasifikatorius [27]

3 EKSPERIMENTINIAI TYRIMAI

3.1 LED segmentų skaitmenų atpažinimo sistema

LED segmentų skaitmenų atpažinimo sistemą sudaro vaizdo stebėjimo kamera, kuria filmuojama rezultatų švieslentė bei kompiuterinė vaizdų apdorojimo ir atpažinimo programa (žr. 3.1.1 pav.). Požymių išskyrimo ir klasifikavimo metodų algoritmai parašyti su *C#* programavimo kalba *Visual Studio C# 2015* aplinkoje, panaudojant *OpenCV* biblioteką.



3.1.1 pav. LED segmentų skaitmenų, esančių rezultatų švieslentėje, aptikimo ir atpažinimo sistema

Filmuojamos trys skirtingos rezultatų švieslentės (žr. 3.1.2 pav.), kuriose LED segmentų skaitmenų dydžiai, spalva, ryškumas bei pakrypimo kampas yra skirtingi. Iš šių rezultatų švieslentėse esančių skaičių sudaroma duomenų bazė, kuri naudojama LED segmentų skaitmenų apdorojimui, aptikimui ir atpažinimui atlikti.



a)



b)

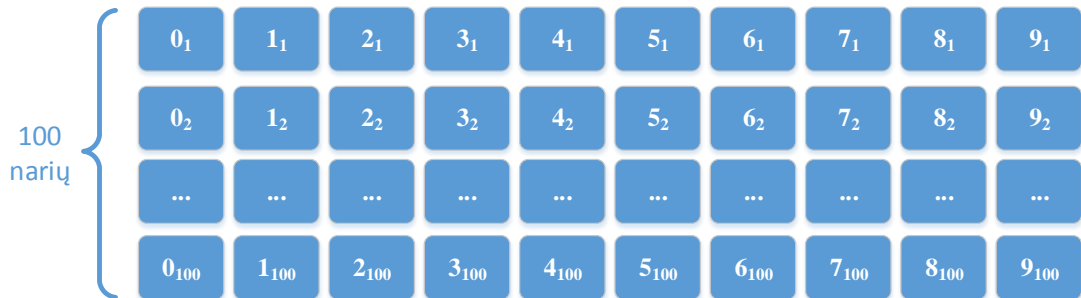


c)

3.1.2 pav. Švieslenčių pavyzdžiai

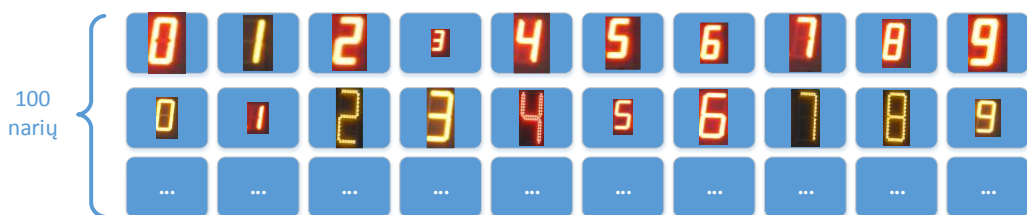
3.2 Duomenų bazės sudarymas

Mokyti ir testuoti sudaroma kiekvieno skaičiaus nuo 0 iki 9, t. y. 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, duomenų bazė. Iš viso duomenų bazėje yra 1000 narių (žr. 3.2.1 pav.), kitaip sakant kiekvieno skaičiaus 100 vienetų pavyzdžių.



3.2.1 pav. Skaitmenų duomenų bazė

Skaitmenys duomenų bazėje yra skirtingų dydžių, spalvų, ryškumo, o keli iš jų pateikti 3.2.2 paveikslėlyje.

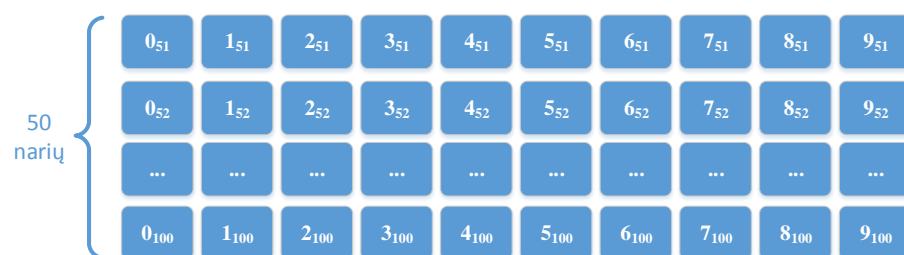


3.2.2 pav. Skaitmenų įvairovė duomenų bazėje

Puse duomenų bazės duomenų naudojami mokymui (žr. 3.2.3 pav., a)), o kita pusė – testavimui (žr. 3.2.3 pav., b)).



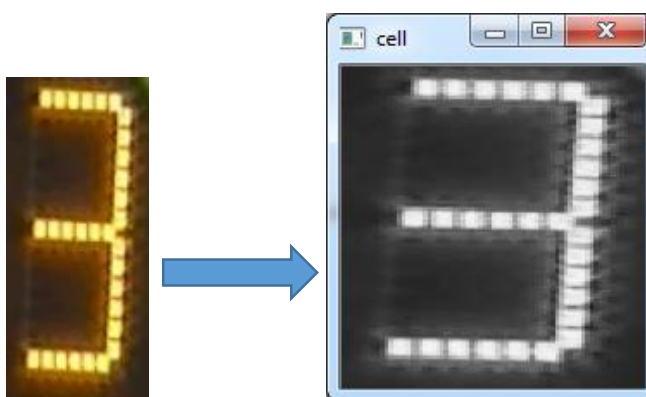
a)



b)

3.2.3 pav. Duomenų bazės resursų panaudojimas mokymui (a)) ir testavimui (b))

Duomenų bazėje skaitmenų rezoliucijos skiriasi. Norint tęsti tyrimą reikia jas suvienodinti. Naujosios rezoliucijos lygis 45x65 pikselių. Pavyzdžiui, skaičiaus „3“ rezoliucija iki pakeitimo lygi 71x161 pikselių (žr. 3.2.4 pav., kairėje pusėje), o po rezoliucijos pakeitimo – 45x65 pikselių (žr. 3.2.4 pav., dešinėje pusėje). Analogiškai pakeitimai daromi visiems duomenų bazėje esantiems nariams.



3.2.4 pav. Skaitmenų dydžio keitimas ir pervedimas į nespaltvotą vaizdą

3.3 Požymių ištraukimo metodai

Apžvelgus literatūrą, pasirinkti šie požymių išgavimo būdai: PCA (žr. [Priedas 1](#)) ir HOG (žr. [Priedas 2](#)). Principinių komponentių analizė ir orientuotų gradientų histogramos dažnai naudojamos spręsti skaitmenų atpažinimo problemą.



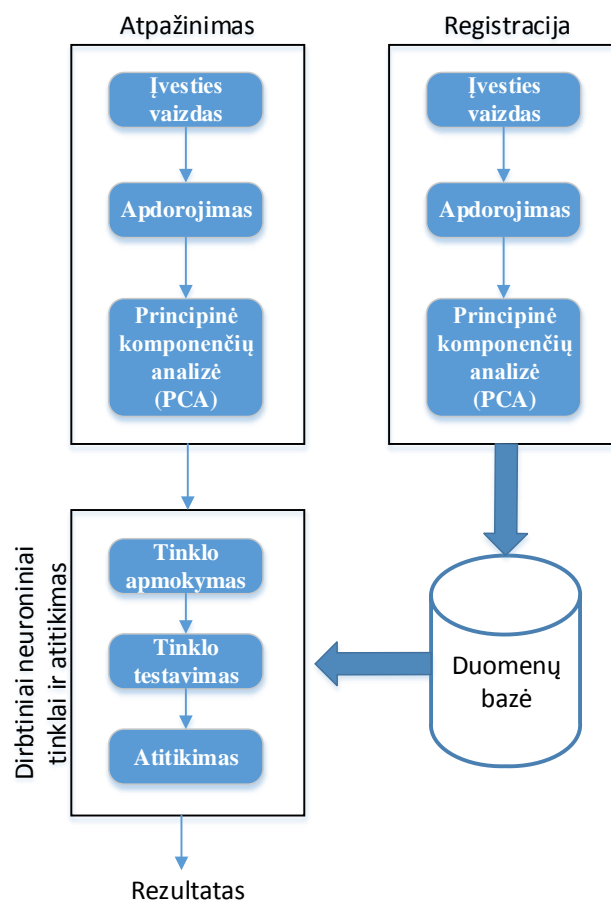
3.3.1 pav. PCA blokinė diagrama

Gautos 56 principinės komponentės.

Taip pat naudota HOG blokinė diagrama, kuri pateikta 2.3.1 paveikslėlyje.

3.4 Požymių klasifikavimo metodai

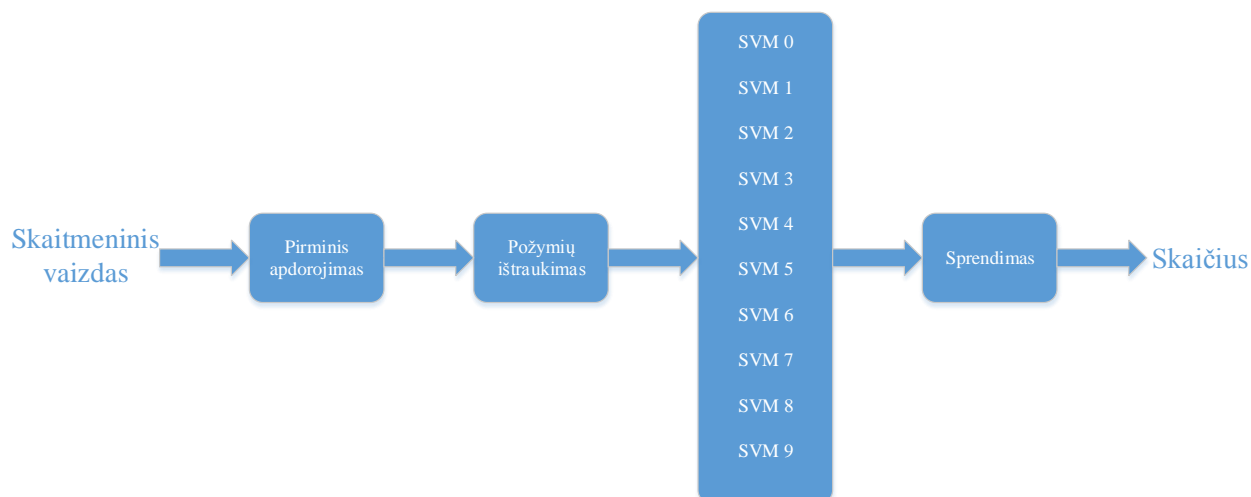
Dirbtiniai neuroniniai tinklai naudojami LED segmentų skaitmens atpažinimui, bet prieš tai vykdomas mokymas, kurio metu skaitmenų vaizdai apdorojami (pakeičiama rezoliucija), naudojant principinę komponentių analizę (arba HOG) gaunami požymiai, kurie vėliau naudojami dirbtiniame neuroniniame tinkle norint rasti skaičių atitikimus. Ši duomenų srautų diagrama pateikiama 3.4.1 paveikslėlyje.



3.4.1 pav. Skaitmenų atpažinimas panaudojant PCA požymių išskyrimo metodą ir ANN

Dirbtinio neuroninio tinklo išvestyse gaunamos tikimybės, t. y., koks skaičius yra labiausiai tikėtinas. Iš išvesčių išrenkama maksimali vertė, kuri nurodo tam tikro skaitmens atitikimą. Remiantis tuo pačiu principu randami visi tiriami LED segmentų skaitmenų atitikmenys. Programos fragmentas pateiktas [3 priede](#).

Kitas šiame darbe naudojamas požymių klasifikavimo metodas yra atraminių vektorių mašinos. Kadangi mes turime 10 vienaženklių skaičių klasių, todėl mums reikia 10 SVM klasifikatorių ar 10 plokštumų, kad atskirtume vienaženklus skaičius vieną nuo kito. Pavyzdžiui, vienas klasifikatorius atskiria visus tam tikro skaitmens pavyzdžius nuo kitų skaitmenų ir priskiria atitinkamo skaitmens grupei (žr. 3.4.2 pav.). Programos fragmentas pateikiamas [4 priede](#).

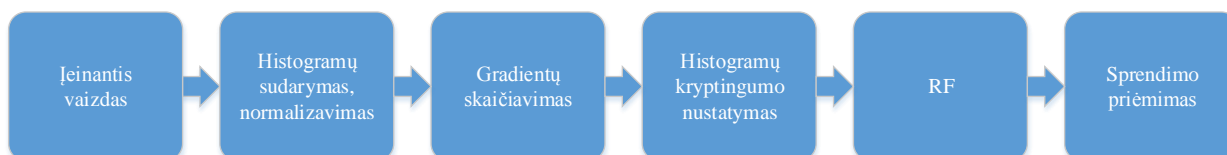


3.4.2 pav. LED segmentų skaitmenų atpažinimo funkcinė schema, panaudojant atraminių vektorių mašinas

Naudojama SVM su *Kernel* funkcija ir parenkamos tinkamos vertės parametrų, su kuriais bandoma gauti geriausią atpažinimo rodiklį kiekvienam SVM ir tuo pačiu daugialypiam SVM klasifikatoriui. Bandymuose yra naudojamos vertės 0,3 ir 10, atitinkamai koeficientams θ ir C .

Taip pat naudojami atsitiktiniai miškai (žr. 3.4.3 pav.). Naudojamas miškas turi 10 medžių.

3.4.3 paveikslėlyje pateikta orientuotų gradientų histogramų gavimo grandinė ir atsitiktinio miško (žr. [Priedas 5](#)) klasifikatoriaus panaudojimas LED segmentų skaitmenims aptikti ir atpažinti.



3.4.3 pav. HOG požymių gavimo grandinė ir atsitiktinių miškų metodo naudojimas skaitmenims atpažinti

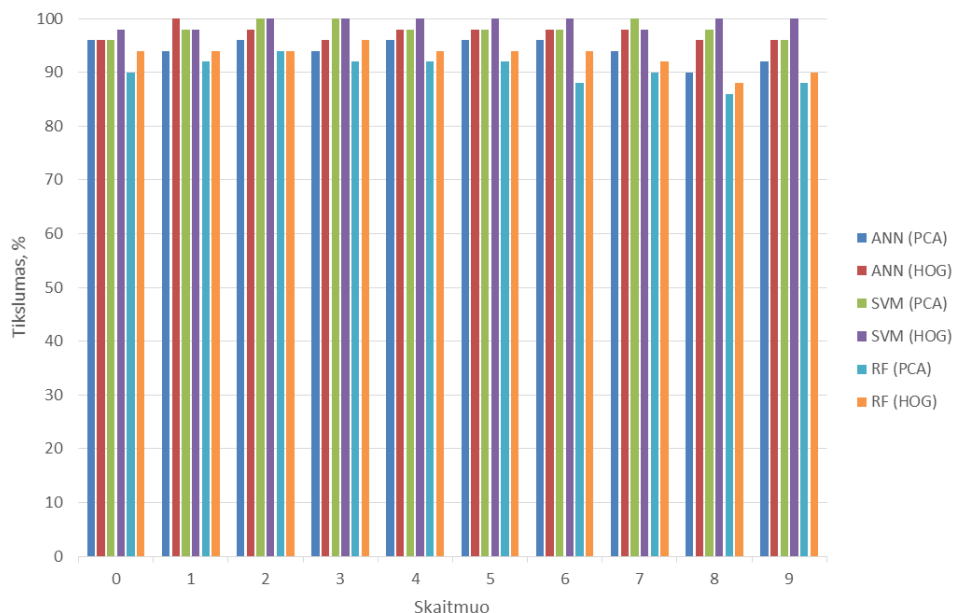
3.5 LED segmentų skaitmenų atpažinimo sistemos tyrimas

Norint išsiaiškinti sistemos patikimumą ir universalumą, reikia ją ištirti: keičiant įvairius parametrus (mokymo duomenų skaičių, paveikslėlio rezoliuciją), stebima sistemos reakcija.

Atlikus LED segmentų skaitmenų (nuo 0 iki 9) atvaizdų atpažinimo tikslumo nuo požymių išskyrimo ir klasifikavimo metodų priklausomybės tyrimą (žr. 3.5.1 pav.) paaiškėjo, kad didžiausias klasifikavimo tikslumas yra gaunamas taikant HOG požymius ir atraminių vektorių mašinų klasifikatorių (99,4 %). Mažiausias tikslumas gautas taikant PCA požymius ir atsitiktinių miškų klasifikatorių (90,4 %).

Atpažinimo tikslumas apskaičiuojamas sudauginus teisingai gautų LED segmentų skaitmenų atitikmenų kiekis n su 100 % ir šią sandaugą padalinus iš testavimo imties, šiuo atveju iš 50, nes naudojama 50 tam tikro skaitmens vaizdų pavyzdžių:

$$\text{Atpažinimo tikslumas} = \frac{n \cdot 100}{50} \quad (3.5.1)$$

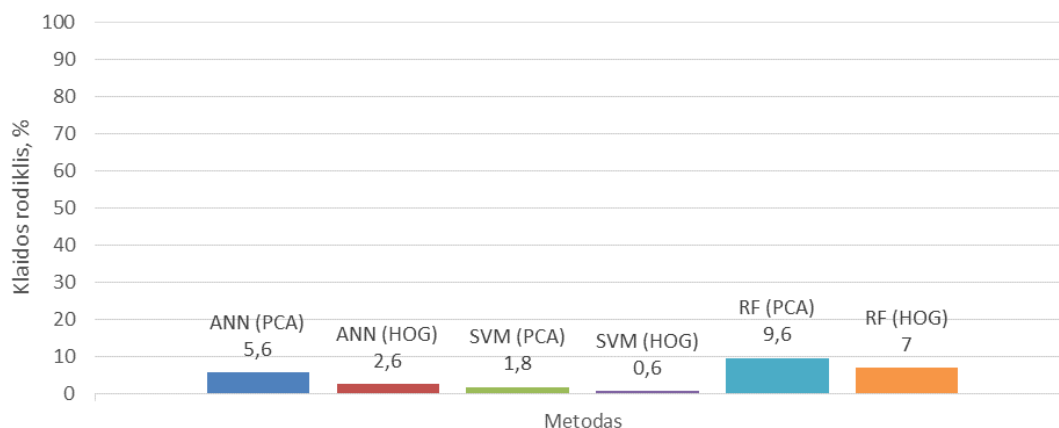


3.5.1 pav. LED segmentų skaitmenų (nuo 0 iki 9) atpažinimo tikslumo priklausomybė nuo požymių išskyrimo ir klasifikavimo metodų su mokymo imtimi

3.5.2 paveikslėlyje pateikta klaidos rodiklio priklausomybė nuo naudojamų vaizdo apdorojimo metodų. Aiškiai matyti, kad atraminių vektorių mašinų klaidos rodiklis mažiausias, t. y. panaudojus principinių komponentių analizę, gautas 1,8 % klaidos rodiklis, o panaudojus orientuotų gradientų histogramas gautas tik 0,6 % klaidos rodiklis. Didžiausias klaidos rodiklis gautas skaitmenis apdorojant su atsitiktiniais miškais (9,6 %).

Klaidos rodiklis k apskaičiuojamas iš 100 % atėmus sudaugintus teisingai gautų LED segmentų skaitmenų atitikmenų kiekis n su 100 % ir šią sandaugą padalinus iš testavimo imties. Supaprastinus lygtį, klaidos rodiklis lygus 100 % ir atpažinimo tikslumo skirtumui:

$$\text{Klaidos rodiklis} = 100 - \frac{n \cdot 100}{50} = 100 - \text{atpažinimo tikslumas} \quad (3.5.2)$$



3.5.2 pav. Klaidos rodiklio priklausomybė nuo požymių išskyrimo ir klasifikavimo metodų

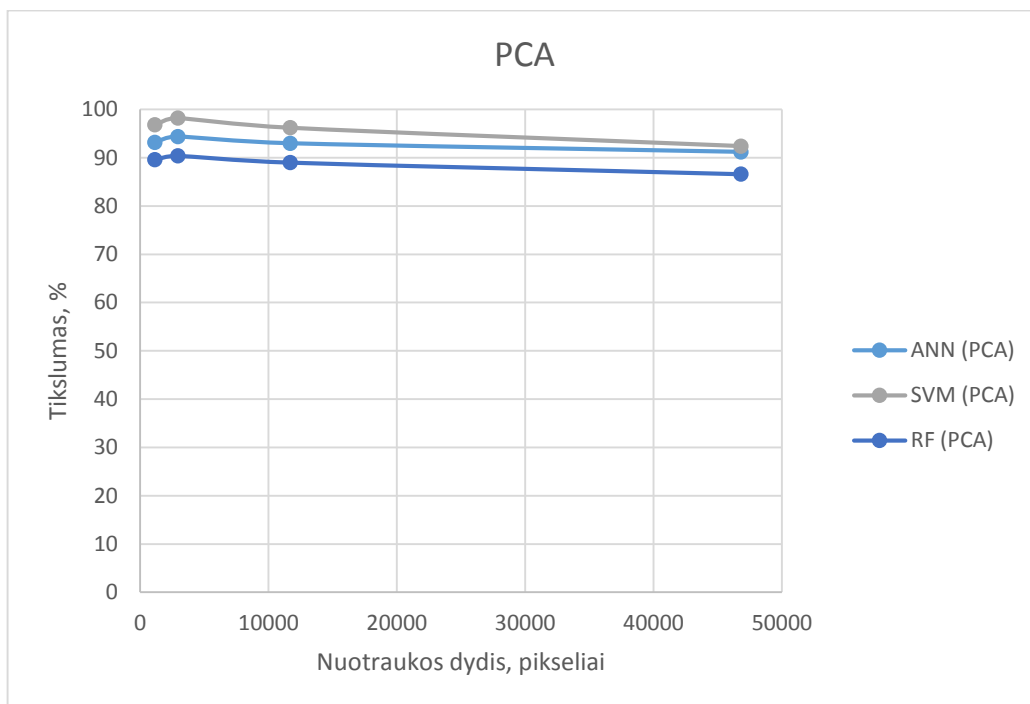
Antroje lentelėje pateikiami duomenų vidurkiai, t. y. susumuojamas kiekvienas gautas atskiras skaitmens atpažinimo tikslumas procentais, panaudojus tam tikrus metodus, ir padalinamas iš dešimties, nes tiek yra atskirų klasių.

2 lentelė. Požymių išskyrimo metodų ir klasifikatorių skaitmenims atpažinti tikslumas ir klaidų rodikliai

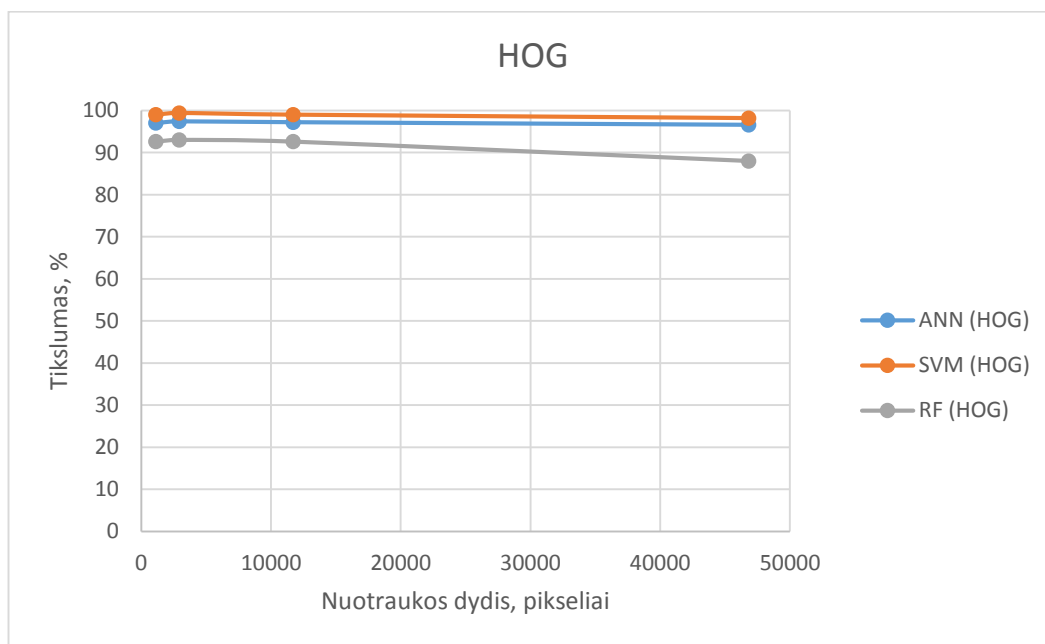
Klasifikatorius	Požymių išskyrimo metodas	Atpažinimo tikslumas	Klaidos rodiklis
ANN	PCA	94,4 %	5,6 %
	HOG	97,4 %	2,6 %
SVM	PCA	98,2 %	1,8 %
	HOG	99,4 %	0,6 %
RF	PCA	90,4 %	9,6 %
	HOG	93,0 %	7 %

Atlikus LED segmentų skaitmenų atvaizdų atpažinimo tikslumo priklausomybės nuo vaizdo rezoliucijos tyrimą, paaiškėjo, kad rezoliucijos keitimas neigiamai veikia skaitmenų atpažinimo tikslumą, kitaip tariant, mažinant arba didinant rezoliuciją, mažėja ir tikslumas. Pvz., kai rezoliucija lygi 45x65 pikselių, naudojant ANN (PCA) gautas tikslumas yra lygus 94,4 %, o sumažinus šią rezoliuciją iki 25x45 – tikslumas sumažėja 1,2 % (iki 93,2 %). Rezoliuciją padidinus iki 90x130 pikselių, atpažinimo tikslumas sumažėja iki 93 %, o prie 180x260 pikselių atpažinimo tikslumas lygus 91,2 %, naudojant PCA požymių išskyrimo metodą (žr. 3.5.3 pav.). Iš rezultatų galima daryti išvadą, kad atpažinimo tikslumas jautresnis rezoliucijos didinimui.

Didžiausias atpažinimo tikslumas keičiant rezoliuciją gautas SVM (HOG) (žr. 3.4 pav.).



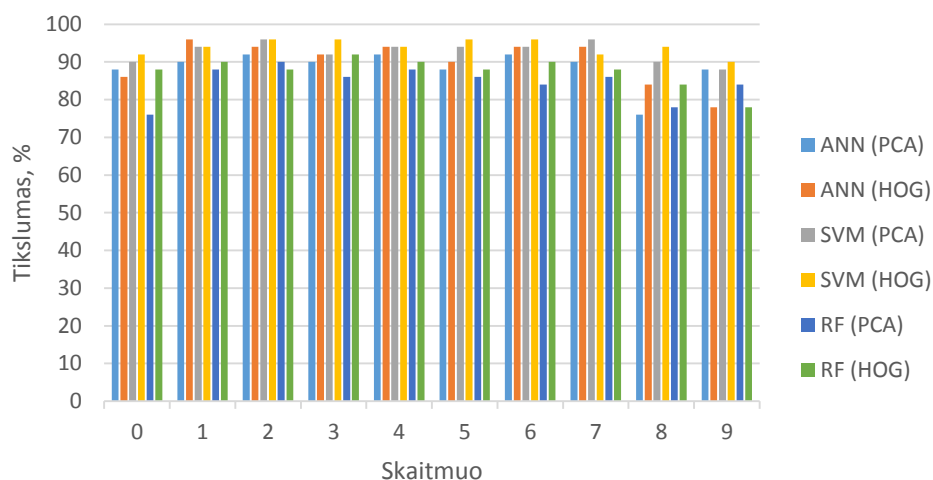
3.5.3 pav. LED segmentų skaitmenų (vidurkio) atpažinimo tikslumo priklausomybė nuo vaizdo rezoliucijos, panaudojant principinių komponentių analizę



3.5.4 pav. LED segmentų skaitmenų (vidurkio) atpažinimo tikslumo priklausomybė nuo rezoliucijos, panaudojant orientuotų gradientų histogramas

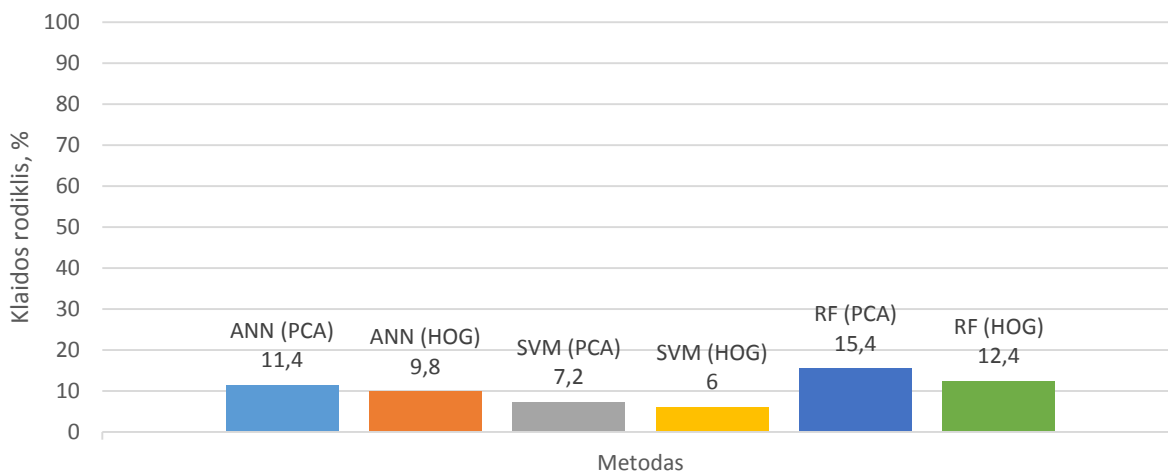
Prie duomenų bazės pridėjus „nežinomus“ LED segmentų skaitmenų pavyzdžius, t. y. duomenis ne iš 3.1.2 paveikslėlyje pateiktų švieslenčių, atliekamas sistemos testavimas. Atlikus LED segmentų skaitmenų (nuo 0 iki 9) atvaizdų atpažinimo tikslumo nuo požymių išskyrimo ir klasifikavimo metodų, kai testuojami duomenys yra nežinomi, priklausomybės tyrimą (žr. 3.5.5

pav.) paaikėjo, kad tiksliausiai duomenys atpažįstami taikant HOG požymius ir atraminių vektorių mašinų klasifikatorių (94 %). Mažiausias tikslumas gautas panaudojus RF (PCA) ir yra lygus 84,6 %.



3.5.5 pav. LED segmentų skaitmenų (nuo 0 iki 9) atpažinimo tikslumo priklausomybė nuo požymių išskyrimo ir klasifikavimo metodų, kai testavimui naudojami „nežinomi“ duomenys (testavimo imtis)

Pagal 3.5.2 formulę apskaičiuotas klaidos rodiklis naudojant testavimo imtį (žr. 3.5.6 pav.).

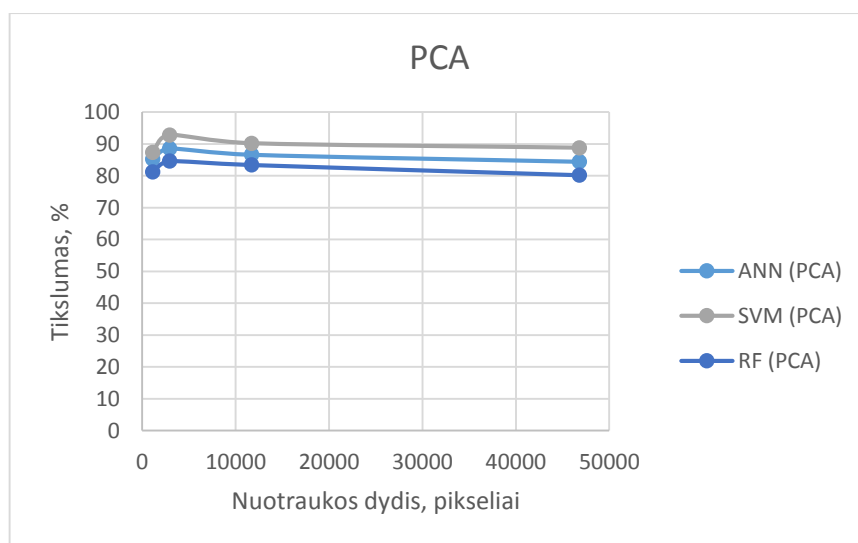


3.5.6 pav. Klaidos rodiklio priklausomybė nuo požymių išskyrimo ir klasifikavimo metodų, naudojant testavimo imtį

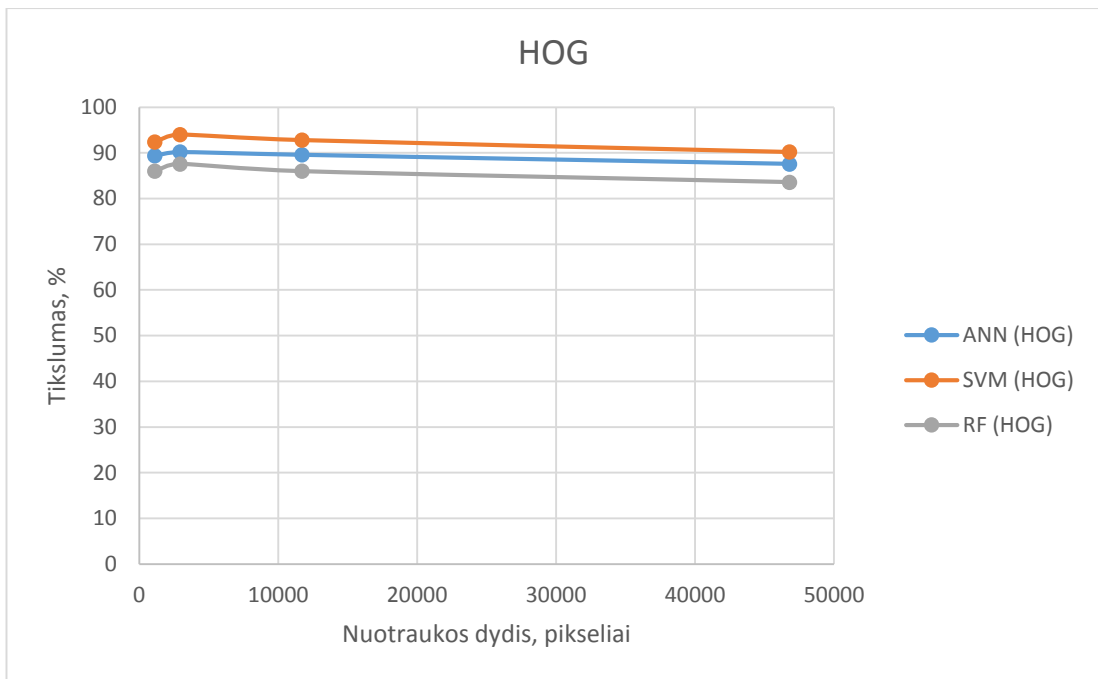
3 lentelė. Požymių išskyrimo metodų ir klasifikatorių skaitmenims atpažinti tikslumas ir klaidų rodikliai, kai naudojama testavimo imtis

Klasifikatorius	Požymių išskyrimo metodas	Atpažinimo tikslumas	Klaidos rodiklis
ANN	PCA	88,6 %	11,4 %
	HOG	90,2 %	9,8 %
SVM	PCA	92,8 %	7,2 %
	HOG	94,0 %	6,0 %
RF	PCA	84,6 %	15,4 %
	HOG	87,6 %	12,4 %

Toliau tiriama LED segmentų skaitmenų atvaizdų atpažinimo tikslumo priklausomybė nuo vaizdo rezoliucijos, kai testavimo duomenys yra „nežinomi“. Kaip ir su „žinomais“ mokymo bei testavimo duomenimis, taip ir su „nežinomais“ aiškiai matoma neigiama rezoliucijos didinimo įtaka skaitmenų atpažinimo tikslumui. Pvz., kai rezoliucija lygi 45x65 pikselių, naudojant RF (PCA) gautas tikslumas yra lygus 84,6 %, o sumažinus šią rezoliuciją iki 25x45 – tikslumas sumažėja 3,4 % (iki 81,2 %). Rezoliuciją padidinus iki 90x130 pikselių, atpažinimo tikslumas sumažėja iki 83,4 %, o prie 180x260 pikselių atpažinimo tikslumas lygus 80,2 %, naudojant PCA požymių išskyrimo metodą (žr. 3.5.7 pav.). Didžiausias atpažinimo tikslumas keičiant rezoliuciją gautas SVM (HOG) (žr. 3.5.8 pav.).

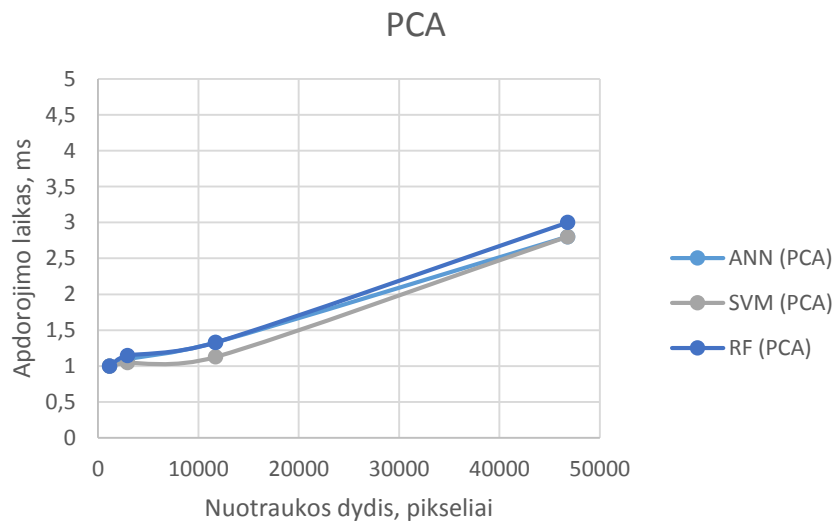


3.5.7 pav. LED segmentų skaitmenų (vidurkio) atpažinimo tikslumo priklausomybė nuo vaizdo dydžio, panaudojant principinių komponentių analizę, kai naudojama testavimo imtis

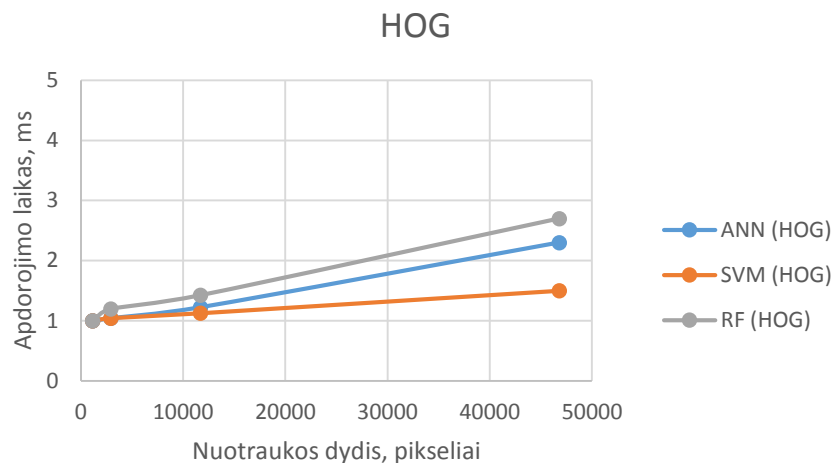


3.5.8 pav. LED segmentų skaitmenų (vidurkio) atpažinimo tikslumo priklausomybė nuo vaizdo dydžio, panaudojant orientuotų gradientų histogramas, kai naudojama testavimo imtis

Iširta kaip priklauso apdorojimo laikas (sparta) nuo vaizdo dydžio. LED segmentų skaitmenų atvaizdų rezoliucijos didinimas neigiamai veikia duomenų apdorojimo spartą. Didinant rezoliuciją taip pat ilgėja atvaizdų apdorojimo laikas (žr. 3.5.9 ir 3.5.10 pav.).



3.5.9 pav. Apdorojimo spartos priklausomybė nuo skaitmenų vaizdų rezoliucijos, naudojant principinių komponentių analizę



3.5.10 pav. Apdorojimo spartos priklausomybė nuo skaitmenų vaizdų rezoliucijos, naudojant orientuotų gradientų histogramas

Žemiau pateikiamos panašumo matricos skirtingiems klasifikatoriams. Pateiktose lentelėse pagrindinės įstrižainės skaičiai rodo teisingą tam tikro klasifikatoriaus atpažinimą. Atsiranda nemažai klaidų tarp vienaženklių skaičių: 8, 9 ir 6 ar 8 ir 3 bei 1 ir 7. Tai yra dėl šių LED segmentų skaitmenų stiliaus panašumo.

4 lentelėje pateikta panašumo matrica ANN klasifikatoriui, taikant PCA požymius. Šis dirbtinis neuroninis tinklas turi 28 įvestis, 70 paslėptų sluoksnių ir 10 išvesčių. Naudojama sigmoidinė aktyvacijos funkcija.

4 lentelė. Panašumo matrica ANN klasifikatoriui (PCA)

	0	1	2	3	4	5	6	7	8	9
0	96	0	0	0	0	2	0	0	0	0
1	0	94	0	0	0	0	0	6	0	0
2	0	0	96	0	0	0	0	0	0	0
3	0	0	4	94	0	0	0	0	6	0
4	0	0	0	0	96	0	0	0	0	0
5	0	0	0	0	0	96	0	0	0	6
6	2	0	0	0	0	2	96	0	0	0
7	0	6	0	0	0	0	0	94	0	0
8	0	0	0	4	0	0	2	0	90	2
9	2	0	0	2	4	0	2	0	4	92

5 lentelėje pateikta panašumo matrica SVM klasifikatoriui, taikant PCA požymius. Šią SVM sudaro 10 klasifikatorių bei naudojama *Kernel* funkcija.

5 lentelė. Panašumo matrica daugialypiam SVM klasifikatoriui (PCA)

	0	1	2	3	4	5	6	7	8	9
0	96	0	0	0	0	0	0	0	0	0
1	0	98	0	0	0	0	0	0	0	0
2	0	0	100	0	0	0	0	0	0	0
3	0	0	8	100	0	0	0	0	0	0
4	0	0	12	0	98	0	0	0	0	2
5	0	0	0	0	0	98	0	0	0	0
6	0	0	0	0	0	2	98	0	0	2
7	0	2	0	0	0	0	0	100	0	0
8	4	0	0	0	0	0	2	0	98	0
9	0	0	0	0	2	0	0	0	2	96

6 lentelėje pateikta panašumo matrica RF klasifikatoriui, taikant PCA požymius. Atsitiktinį mišką sudaro 10 medžių.

6 lentelė. Panašumo matrica RF klasifikatoriui (PCA)

	0	1	2	3	4	5	6	7	8	9
0	90	0	0	0	0	0	0	0	8	2
1	0	92	0	0	0	0	0	6	0	0
2	0	0	94	0	0	0	0	0	0	0
3	6	0	2	92	0	0	0	0	0	0
4	0	0	0	0	92	0	0	4	0	0
5	0	0	0	0	0	92	4	0	0	0
6	0	0	0	0	4	4	88	0	2	2
7	0	4	0	0	0	0	0	90	0	0
8	4	0	4	2	0	0	8	0	86	8
9	0	4	0	6	4	4	0	0	4	88

7 lentelėje pateikta panašumo matrica ANN klasifikatoriui, taikant HOG požymius. Šis dirbtinis neuroninis tinklas turi 28 įvestis, 70 paslėptų sluoksnių ir 10 išvesčių. Naudojama sigmoidinė aktyvacijos funkcija.

7 lentelė. Panašumo matrica ANN klasifikatoriui (HOG)

	0	1	2	3	4	5	6	7	8	9
0	96	0	0	0	0	0	0	0	2	2
1	0	100	0	0	0	0	0	2	0	0
2	0	0	98	0	0	0	0	0	0	0
3	0	0	0	96	0	0	0	0	0	0
4	0	0	0	0	98	0	0	0	0	0
5	0	0	0	0	0	98	0	0	0	2
6	0	0	2	0	0	0	98	0	2	0
7	0	0	0	0	0	0	0	98	0	0
8	4	0	0	2	0	0	2	0	96	0
9	0	0	0	2	2	2	0	0	0	96

8 lentelėje pateikta panašumo matrica SVM klasifikatoriui, taikant HOG požymius. Šią SVM sudaro 10 klasifikatorių bei naudojama *Kernel* funkcija.

8 lentelė. Panašumo matrica daugialypiam SVM klasifikatoriui (HOG)

	0	1	2	3	4	5	6	7	8	9
0	98	0	0	0	0	0	0	2	0	0
1	0	98	0	0	0	0	0	0	0	0
2	0	0	100	0	0	0	0	0	0	0
3	0	0	0	100	0	0	0	0	0	0
4	0	0	0	0	100	0	0	0	0	0
5	0	0	0	0	0	100	0	0	0	0
6	0	0	0	0	0	0	100	0	0	0
7	0	2	0	0	0	0	0	98	0	0
8	2	0	0	0	0	0	0	0	100	0
9	0	0	0	0	0	0	0	0	0	100

9 lentelėje pateikta panašumo matrica RF klasifikatoriui, taikant HOG požymius. Atsitiktinį mišką sudaro 10 medžių.

9 lentelė. Panašumo matrica RF klasifikatoriui (HOG)

	0	1	2	3	4	5	6	7	8	9
0	94	0	0	0	0	0	0	0	0	0
1	0	94	0	0	0	0	0	6	0	0
2	0	0	94	0	0	0	0	0	0	0
3	0	0	0	96	0	0	0	0	0	2
4	0	2	0	0	94	0	0	2	0	0
5	0	0	0	0	0	94	2	0	0	0
6	0	0	2	0	0	0	94	0	2	0
7	0	4	0	0	4	0	0	92	0	0
8	4	0	4	2	0	0	4	0	88	8
9	2	0	0	2	2	6	0	0	10	90

REZULTATAI IR IŠVADOS

1. Parinkti SVM θ ir C parametrai, kurie yra lygūs 0,3 ir 10, su kuriais gaunamas didžiausias atpažinimo tikslumas (99,4 %, testuojant su mokymo imtimi).
2. Atlikus LED segmentų skaitmenų (nuo 0 iki 9) atvaizdų atpažinimo tikslumo nuo požymių išskyrimo ir klasifikavimo metodų priklausomybės tyrimą naudojant mokymo imtį, gautas didžiausias atpažinimo tikslumas, kuris yra lygus 99,4 %, panaudojus atramines vektorių mašinas, prieš tai gavus orientuotų gradientų histogramas. Mažiausias tikslumas gautas panaudojus atsitiktinius miškus, prieš tai atlikus principinių komponentų analizę (90,4 %).
3. Atlikus arabiškų skaitmenų atvaizdų atpažinimo tikslumo priklausomybės nuo vaizdo rezoliucijos tyrimą, kai naudojama mokymo imtis, paaiškėjo, kad rezoliucijos keitimas neigiamai veikia skaitmenų atpažinimo tikslumą. Kai rezoliucija lygi 45x65 pikselių, naudojant PCA požymius ir ANN klasifikatorių gautas atpažinimo tikslumas yra lygus 94,4 %, o prie 180x260 pikselių atpažinimo tikslumas sumažėjo iki 91,2 %, kitaip tariant, didinant rezoliuciją, atpažinimo tikslumas mažėja.
4. Atlikus LED segmentų skaitmenų atvaizdų atpažinimo tikslumo nuo požymių išskyrimo ir klasifikavimo metodų, kai naudojama testavimo imtis, priklausomybės tyrimą, paaiškėjo, kad tiksliausiai duomenys atpažintami panaudojus atramines vektorių mašinas, prieš tai gavus orientuotų gradientų histogramas (94 %). Mažiausias tikslumas gautas panaudojus RF (PCA) ir yra lygus 84,6 %. Dėl gauto nepakankamo tikslumo, LED segmentų skaitmenų atpažinimo sistemą reikia tobulinti.
5. Panaudojus PCA požymius ir atsitiktinių miškų klasifikatorių, gautas klaidos rodiklis, kuris yra lygus 9,6 %, testuojant su mokymo imtimi, o testuojant su testavimo imtimi gautas 15,4 % klaidos rodiklis. Galima daryti išvadą, kad RF klasifikatorius yra netinkamas LED segmentų skaitmenims atpažinti dėl per didelio klaidos rodiklio.

NAUDOTOS LITERATŪROS IR KITŲ INFORMACIJOS ŠALTINIŲ SĄRAŠAS

1. RATHORE, M., KUMARI, S. Tracking Number Plate from Vehicle Using Matlab. India, 2014.
2. BAILMARE, S. H., GADICHA, A. B. A Review paper on Vehicle Number Plate Recognition (VNPR) Using Improved Character Segmentation Method. 2013.
3. MOHAMED, B. R., ABD EL KADER, H. M., RAFAAT, H. M., SHARAF, M. S. Automatic Number Plate Recognition. 2013.
4. MUNUO, C., KISANGIRI, M. Vehicle Number Plates Detection and Recognition using improved Algorithms: A Review with Tanzanian Case study. Tanzania, 2014.
5. KAR SOON, C., CHIRUNG LIN, K., YING JENG, C., SUANDI, S. Malaysian Car Number Plate Detection and Recognition System. Malaysia, 2012.
6. SAGHAEI, H. Proposal for Automatic License and Number Plate Recognition System for Vehicle Identification. Iran, 2016.
7. DANTI, A. Recognition of Vehicle Number Plates and Retrieval of Vehicle owner's Registration Details. India.
8. SAI KRISHNA, P. Automatic Number Plate Recognition by Using Matlab. Visakhapatnam, India, 2015.
9. KAUR, R. Serial Number Recognition in Banknotes Using HoG Feature Extraction and KNN Classification. India, p. 45 – 48.
10. ANDREJEVAS, A. Rašto ženklų atpažinimas naudojant neuroninius tinklus. Vilnius, 2011, p. 13 – 15.
11. ZHANG, B., FU, M., YAN, H. A nonlinear neural network model of mixture of local principal component analysis: application to handwritten digits recognition. Australia, 2006.
12. LAWGALI, A. Recognition of Handwritten Digits using Histogram of Oriented Gradients. Libya, 2016.
13. Informacija apie sąsūkinčius neuroninius tinklus ir jų taikymą skaitmenims atpažinti [žiūrėta 2017-05-15]. Prieiga per internetą: <http://yann.lecun.com/exdb/lenet/>
14. RODRIGUEZ, M., BERDUGO, G., JABRA, D., CALLE, M., JIMENO, M. HD MR: a new algorithm for number recognition in electrical meters. Colombia, 2014.
15. ARTEMČIUKAS, E. Papildytos realybės sistemų optimizavimo metodų tyrimas ir taikymas. Lietuva: mokslinė ataskaita MII-DS-09P-13-16, 2013, p. 8 – 9.
16. BARBER, D. Machine Learning and Pattern Recognition. Edinburgh, p. 2 – 4.

17. RISVIK, H. Principal Component Analysis (PCA) and NIPALS algorithm. 2007, p. 1 – 2.
18. MARQUES, I. Face Recognition Algorithms. 2010, p. 31 – 33.
19. MUDROVA, M., PROCHAZKA, A. Principal Component Analysis in Image Processing. Prague, p. 1 – 2.
20. SIMUTIS, R., VAITKUS, V., LENGVENIS, P., MASKELIŪNAS, R., RAUDONIS, V., DILIJONAS, D., ŽILINSKAS, T. Roboto ranka neįgaliesiems: QUADRIBOT. Kaunas, 2015, p. 7 – 8.
21. OŠVAT, M. Object Detection and Segmentation Using Contours. Bratislava, 2014, p. 5 – 8.
22. HAI – BO, L., XIN – RONG, Y., HONG – MEI, L., QING – HAI, D. A method for real-time implementation of HOG feature extraction. China, p. 1 – 5.
23. MALTAROLLO, V., HONORIO, K., FERREIRA DA SILVA, A. Applications of Artificial Neural Networks in Chemical. P. 204 – 206.
24. Finansinių ir fizinių procesų prognozavimo modelių tyrimas ir taikymas. Kaunas, 2009, p. 8 – 9.
25. MAJI, S., MALIK, J. Fast and Accurate Digit Classification. California, p. 1.
26. SADRI, J., SUEN, C., BUI, T. Application of Support Vector Machines for Recognition of Handwritten Arabic/Persian Digits. Canada, p. 301 – 306.
27. BORRELLI, J. Study of the Relationship of Training Set Size to Error Rate in yet Another Decision Tree and Random Forest Algorithms. Texas, 2006, p. 28 – 30.

PRIEDAS 1. PCA PROGRAMOS FRAGMENTAS

```
case 0: //PCA eigenvectors gavimas

    Mat total = new Mat(); // pav saugojimui

    for (int dir_id = 0; dir_id < 10; dir_id++)
    {
        string[] files = Directory.GetFiles(drs_img[dir_id], "*.jpg");
        int[] randomVector = new int[files.Length];
        for (int i = 0; i < randomVector.Length; i++) { randomVector[i] = i; }
        Shuffle(randomVector);
        for (int i = 0; i < nmbOfFiles[2]; i++)
        {
            string fileName = files[randomVector[i]];
            img = Cv2.ImRead(fileName, LoadMode.GrayScale);
            Mat rshp = new Mat();
            Mat tmp = new Mat();
            Cv2.Resize(img, img, new Size(45, 65));
            img.CopyTo(rshp);
            (rshp.Reshape(0, 1)).CopyTo(tmp);
            total.PushBack(tmp);
            Console.WriteLine("total: " + total.Rows.ToString());
        }
    }

    Console.WriteLine("Starttin PCA");
    Mat mean = new Mat();
    Mat eigenvectors = new Mat();
    total.ConvertTo(total, MatType.CV_32F);
    Cv2.PCACompute(total, mean, eigenvectors, 100);

    //Saugomi duomenys xml
    CvFileStorage fs = new
CvFileStorage(@"C:\Users\Gintare\Desktop\Magistrinio2dalis\PCA_ANN\usingPCA\pca_data_num.
xml", null, FileStorageMode.Append);
    fs.Write("mean", mean.ToCvMat());
    fs.Write("eigenvectors", eigenvectors.ToCvMat());
    Console.WriteLine("finish PCA");
    break;
```

PRIEDAS 2. HOG PROGRAMOS FRAGMENTAS

```
case 0: //HOG

Mat total = new Mat(); // pav. kaupimui
int imgWidth = 32;
int imgHeight = 64;
HOGDescriptor hog = new HOGDescriptor();// HOG descr
hog.BlockSize = new Size(16,32);
hog.WinSize = new Size(imgWidth, imgHeight);
int Count = 0;

for (int dir_id = 0; dir_id < 10; dir_id++)
{
    string[] files = Directory.GetFiles(drs_img[dir_id], "*.jpg");
    int[] randomVector = new int[files.Length];
    for (int i = 0; i < randomVector.Length; i++) { randomVector[i] = i;
    }
    Shuffle(randomVector);

for (int i = 0; i < nmbOfFiles[2]; i++)
{
    string fileName = files[randomVector[i]];
    img = Cv2.ImRead(fileName, LoadMode.GrayScale);
    Cv2.Resize(img, img, new Size(imgWidth, imgHeight));
    Point[] PointsHog = new Point[img.Cols * img.Rows];

    int k = 0;
for(int pointCol = 0; pointCol < img.Cols; pointCol++)
{
    for (int pointRow = 0; pointRow < img.Rows; pointRow++)
    {
        Point p1 = new Point(pointCol, pointRow);
        PointsHog[k++] = p1;
    }
}

float[] valueHog = hog.Compute(img);
Cv2.ImShow("test", img);

for (int k1 = 0; k1 < valueHog.Length; k1++)
{
    if (valueHog[k1] != 0)
    {
        Console.WriteLine("ne nulis");
    }
}

    Console.WriteLine(Count);
    Count++;
    Cv2.WaitKey(300);
}
    Console.WriteLine("finish writing number " + dir_id);
}

    Console.WriteLine("finish HOG");
    break;
}
```

PRIEDAS 3. ANN PROGRAMOS FRAGMENTAS

```
//Load ANN model

mlp.Load(@"C:\Users\Gintare\Desktop\Magistrinio2dalis\PCA_ANN\usingPCA\cellClassifier65.xml");

    //Load xml file with a PCA data
    Mat _mean = new Mat();
    Mat _eigenvectors = new Mat();
    CvMat pred_ = new CvMat(1, 10, MatrixType.F32C1);

    using (CvFileStorage fs5 = new
CvFileStorage(@"C:\Users\Gintare\Desktop\Magistrinio2dalis\PCA_ANN\usingPCA\pca_data_num.xml", null, FileStorageMode.Read))
    {
        CvFileNode param4 = fs5.GetFileNodeByName(null, "eigenvectors");
        _eigenvectors = new Mat(fs5.Read<CvMat>(param4));
        param4 = fs5.GetFileNodeByName(null, "mean");
        _mean = new Mat(fs5.Read<CvMat>(param4));
    }

    Window show = new Window("cell", WindowMode.FreeRatio);

    var csv = new StringBuilder();

    foreach (string sFile in filesP)
    {
        img = Cv2.ImRead(sFile, LoadMode.GrayScale);

        Rect[] loc = GetnLocation(img, _cellCascade1);
        double maxValue = -999;
        int id = -1;
        double[] tmpAr = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };

        foreach (Rect rect in loc)
        {
            if (rect.X > 1)
            {
                Cv2.Rectangle(img, rect, Scalar.Red, 1);
                Mat I = new Mat();
                img[rect].CopyTo(I);
                Cv2.ConvertScaleAbs(I, I);
                Cv2.Resize(I, I, new Size(45, 65));
                Mat tmp2 = new Mat();
                (I.Reshape(0, 1)).CopyTo(tmp2);
                Mat projectsOut = new Mat();
                Cv2.PCAProject(tmp2, _mean, _eigenvectors, projectsOut);

                CvMat _outData2 = (CvMat)projectsOut;
                mlp.Predict(_outData2, pred_);

                for (int ii = 0; ii < 10; ii++)
                {
                    if (maxValue < pred_[0, ii])
                    {
                        maxValue = pred_[0, ii];
                        id = ii;
                        for (int j = 0; j < 10; j++) { tmpAr[j] =
Math.Round(pred_[0, j], 4); }
                    }
                }
            }
        }
    }
    /*
```

```

        for (int i = 0; i < 10; i++)
        {
            Console.WriteLine(i.ToString() + ": " + pred_[0,
i].ToString());
        }
    }
}

var newLine =
string.Format("{0};{1};{2};{3};{4};{5};{6};{7};{8};{9}",
    tmpAr[0].ToString(),
    tmpAr[1].ToString(),
    tmpAr[2].ToString(),
    tmpAr[3].ToString(),
    tmpAr[4].ToString(),
    tmpAr[5].ToString(),
    tmpAr[6].ToString(),
    tmpAr[7].ToString(),
    tmpAr[8].ToString(),
    tmpAr[9].ToString());
csv.AppendLine(newLine);

Console.Write(".");
}

//saving results

File.WriteAllText(@"C:\Users\Gintare\Desktop\PCA_results\results.csv", csv.ToString());

show.Image = img;
Cv2.WaitKey(0);

break;

```

PRIEDAS 4. SVM PROGRAMOS FRAGMENTAS

```
// SVM
CvSVMParams param = new CvSVMParams(
    SVMType.NuSvc,
    SVMKernelType.Rbf,
    5.0, // degree
    0.2, // gamma
    0.3, // coeff0
    10.0, // c
    0.5, // nu
    0.1, // p
    null,
    criteria);
```


PRIEDAS 5. RF PROGRAMOS FRAGMENTAS

```
CvDTree Forest = new CvDTree();  
CvDTreeParams ForestParam = new CvDTreeParams();  
ForestParam.MaxDepth = 10;  
ForestParam.MinSampleCount = 10;  
ForestParam.UseSurrogates = false;  
ForestParam.CvFolds = 0;  
ForestParam.Use1seRule = false;  
ForestParam.TruncatePrunedTree = false;
```