



**KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS**

Gintas Dauskurdas

**ANT PAKUOTĖS UŽPURŠKIAMOS INFORMACIJOS
IDENTIFIKAVIMO, PANAUDOJANT SKIRTINGUS VAIZDŲ
ATPAŽINIMO METODUS, TYRIMAS**

Baigiamasis magistro projektas

Vadovas

doc. dr. Gintaras Dervinis

KAUNAS, 2017

KAUNO TECHNOLOGIJOS UNIVERSITETAS

**ELEKTROS IR ELEKTRONIKOS FAKULTETAS
AUTOMATIKOS KATEDRA**

**ANT PAKUOTĖS UŽPURŠKIAMOS INFORMACIJOS
IDENTIFIKAVIMO, PANAUDOJANT SKIRTINGUS VAIZDŲ
ATPAŽINIMO METODUS, TYRIMAS**

Baigiamasis magistro projektas
Valdymo technologijos (kodas 621H66001)

Vadovas

doc. dr. Gintaras Dervinis

Recenzentas

doc. dr. Vidas Raudonis

Projektą atliko

Gintas Dauskurdas

KAUNAS, 2017



KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS

Gintas Dauskurdas

Valdymo technologijos, 621H66001

Baigiamojo projekto

„Ant pakuotės užpurškiamos informacijos identifikavimo, panaudojant skirtingus vaizdų atpažinimo metodus, tyrimas“

AKADEMINIO SAŽININGUMO DEKLARACIJA

20 ____ m. _____ d.
Kaunas

Patvirtinu, kad mano **Ginto Dauskurdo** baigiamasis projektas tema „Ant pakuotės užpurškiamos informacijos identifikavimo, panaudojant skirtingus vaizdų atpažinimo metodus, 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)

Dauskurdas, Gintas. **Ant pakuotės užpurškiamos informacijos identifikavimo, panaudojant skirtingus vaizdų atpažinimo metodus, tyrimas.** Valdymo sistemų magistro baigiamasis projektas / vadovas doc. dr. Gintaras Dervinis; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas, Automatikos katedra.

Kaunas, 2017. 49 psl.

SANTRAUKA

Pastaruosius dešimtmečius kompiuterinės regos technologijos ir vaizdų apdorojimas susilaukė didžiulio dėmesio įvairiose mokslo ir pramonės srityse. Viena iš plačiausiai naudojamų vaizdų apdorojimų sričių yra informacijos identifikavimas. Augant pramonei ir automatizacijos lygiui, informacijos identifikavimas tampa vis svarbesnis norint užtikrinti produkcijos atsekamumą bei informacijos tikslumą. Todėl šiame darbe nagrinėjamas informacijos identifikavimo uždavinys pramonėje.

Pirmoje darbo dalyje trumpai aptariama kompiuterinė rega ir simbolių identifikavimas. Taip pat apžvelgiami k -artimiausių kaimynų KNN, atraminių vektorių mašinų SVM, dirbtinių neuroninių tinklų ANN, Naive Bajeso ir sprendimų medžio klasifikatorių metodika bei HOG, LBP, PCA, SIFT, SURF, *Haar-like* požymių deskriptorių metodika.

Antroje darbo dalyje atliekamas pramoninio rašalinio čiurkšlinio spausdintuvo užpurškiamos informacijos ant pakuotės identifikavimo tyrimas, panaudojus pirmoje dalyje aprašytą klasifikatorių ir deskriptorių kombinacijas.

Reikšminiai žodžiai: kompiuterinė rega, vaizdų apdorojimas, informacijos identifikavimas, k -artimiausi kaimynai, atraminių vektorių mašinos, dirbtiniai neuroniniai tinklai, sprendimų medis, klasifikatorius, deskriptorius.

Dauskurdas, Gintas. **Research of different image recognition methods for identification of sprayed information on the package.** Final project of control systems master degree / supervisor doc. dr. Gintaras Dervinis; Kaunas University of Technology, Faculty of Electrical and Electronics Engineering, department of Automation.

Kaunas, 2017. 49 pages.

SUMMARY

In the last decades, computer vision technology and image processing have received a lot of attention in the various scientific and industrial fields. One of the most widely used areas of image processing is the information identification. The information identification has become increasingly important in order to ensure product traceability and information accuracy while the industry and automation level are growing. Therefore, this paper explores the task of the information identification in the industry.

The first part briefly reviews the computer vision and the character identification. It also overviews the k -Nearest Neighbors KNN, the Support Vector Machines SVM, the Artificial Neural Network ANN, Naïve Bayes and Decision Tree classification methodology and HOG, LBP, PCA, SIFT, SURF, Haar-like features descriptors methodology.

In the second part the research of the industrial inkjet printer sprayed information on the package information identification is performed, using the classification and descriptor combinations, which were described in the first part of the work.

Keywords: computer vision, image processing, information identification, k -Nearest Neighbors, Support Vector Machines, Artificial Neural Network, Decision Tree, classifier, descriptor.

TURINYS

SANTRUMPŲ IR ŽENKLŲ AIŠKINIMO ŽODYNAS	8
ĮVADAS.....	9
1 APŽVALGINĖ DALIS	10
1.1 Požymių deskriptoriai.....	11
1.1.1 HOG deskriptorius	12
1.1.2 LBP deskriptorius.....	13
1.1.3 PCA deskriptorius	14
1.1.4 Pikselių vektorius	16
1.1.5 SIFT deskriptorius.....	17
1.1.6 SURF deskriptorius	17
1.1.7 <i>Haar-like</i> deskriptorius	18
1.2 Požymių klasifikatoriai.....	19
1.2.1 <i>K</i> -artimiausių kaimynų (angl. KNN) klasifikatorius	19
1.2.2 Atraminės vektorių mašinos (angl. SVM) klasifikatorius	21
1.2.3 Dirbtinių neuroninių tinklų (angl. ANN) klasifikatorius.....	23
1.2.4 Naive Bajeso klasifikatorius.....	25
1.2.5 Sprendimų medžio klasifikatorius.....	28
1.2.6 <i>Adaboost</i> algoritmas	30
2 TYRIMO DALIS	32
2.1 Pirmoji tyrimo dalis.....	35
2.2 Antroji tyrimo dalis	38
2.3 Trečioji tyrimo dalis	39
2.4 Ketvirtoji tyrimo dalis	40
IŠVADOS IR PASIŪLYMAI.....	42
LITERATŪROS SĄRAŠAS.....	43
PRIEDAI	50

PAVEIKSLŲ SĄRAŠAS

1.1 pav. Vaizdo apdorojimo žingsniai tekstui atpažinti	11
1.2 pav. LBP skaičiavimo pavyzdys: (a) nuotraukos pavyzdys, (b) dvejetainė struktūra po ribų pritaikymo, (c) svorių matrica [17]	14
1.3 pav. Duomenys ir PCA pagrindinės komponentės	15
1.4 pav. <i>Haar-like</i> deskriptoriaus savybių tipai [30]	18
1.5 pav. Klasės nustatymas artimiausių kaimynų metodu, 1 klasė – kvadratai, 2 klasė – žvaigždutės, nauji duomenys – trikampis	21
1.6 pav. SVM dviejų klasių atskyrimas hyperplokštuma [37]	22
1.7 pav. ANN modelio struktūra [4].....	24
1.8 pav. Tikimybių dažnių lentelės	26
1.9 pav. Tikimybių dažnių lentelės su apriorinėmis X ir Y tikimybėmis	27
1.10 pav. Sprendimų medžio modelis [47].....	28
1.11 pav. Sprendimų medžio pavyzdys	29
1.12 pav. Sprendimų medžio pavyzdžio sprendimas	30
2.1 pav. Produkto pakuotės nuotrauka	32
2.2 pav. Pirmojo formato informacijos šifravimas	32
2.3 pav. Antrojo formato informacijos šifravimas	33
2.4 pav. Vaizdo apdorojimo žingsniai tekstui atpažinti	35
2.5 pav. Tyrimo nuotrauka Nr. 1	36
2.6 pav. Tyrimo nuotrauka Nr. 2	36
2.7 pav. Pirmojo tyrimo rezultatai (tikslumas, %).....	37
2.8 pav. Pirmojo tyrimo rezultatai (laikas, %).....	37

LENTELIŲ SĄRAŠAS

1.1 lentelė. Duomenys ir jiems priskirtos klasės	20
1.2 lentelė. Duomenys Naive Bajeso klasifikatoriui	26
2.1 lentelė. Informacijai aprašyti naudojami simboliai	33
2.2 lentelė. Kameros specifikacijos	34
2.3 lentelė. Kompiuterio specifikacijos	34
2.4 lentelė. Keičiami klasifikatorių parametrai	36
2.5 lentelė. Pirmojo tyrimo rezultatai	36
2.6 lentelė. Pirmojo tyrimo rezultatai pagal tikslumą mažėjimo tvarka	37
2.7 lentelė. Antrojo tyrimo rezultatai (tirtos 2 nuotraukos)	38
2.8 lentelė. Antrojo tyrimo rezultatai pagal tikslumą mažėjimo tvarka (tirtos 2 nuotraukos)	38
2.9 lentelė. Antrojo tyrimo rezultatai, naudojant dirbtinių neuroninių tinklų deskriptorių	39
2.10 lentelė. Trečiojo tyrimo rezultatai (tirtos 50 nuotraukų)	39
2.11 lentelė. Mokymo duomenų bazės simboliai ir jų kiekis	40
2.12 lentelė. Ketvirtąjo tyrimo rezultatai (tirtos 50 nuotraukų)	41

SANTRUMPŲ IR ŽENKLŲ AIŠKINIMO ŽODYNAS

ANN (angl. Artificial Neural Network) – dirbtinis neuroninis tinklas.

ASCII (angl. *American Standard Code for Information Interchange*) – simbolių kodavimo standartas.

ASWF (angl. *Average Similarity – Weighted Function*) – k -artimiausių kaimynų klasifikavimo metodo tikimybės funkcijos rūšis.

DoG (angl. *Difference of Gaussians*) – Gauso skirtumas.

DVF (angl. *Discrete – Valued Function*) – k -artimiausių kaimynų klasifikavimo metodo tikimybės funkcijos rūšis.

HOG (angl. *Histogram of Oriented Gradients*) – orientuotų gradientų histogramų požymių deskriptorius.

KNN (angl. *k-nearest neighbors*) – k -artimiausių kaimynų klasifikavimo metodas.

LBP (angl. *Local Binary Pattern*) – lokalsios dvejetainės struktūros požymių deskriptorius.

MAP (angl. *Maximum a Posteriori*) – hipotezė naudojama Naive Bajeso statistikoje.

MLP (angl. *Multilayer perception*) – daugiasluoksnio perceptrono metodas.

PCA (angl. *Principal Component Analysis*) – pagrindinių komponentų metodas.

RBF (angl. *Radial Basis Function*) – SVM klasifikatoriaus branduolio funkcija.

SIFT (angl. *Scale-invariant feature transform*) – nuo mastelio nepriklausomas požymių deskriptorius.

SURF (angl. *Speeded Up Robust Features*) – pagreitintas, tvirtas požymių deskriptorius.

SVM (angl. *Support Vector Machine*) – atraminių vektorių mašinų klasifikatorius.

SWF (angl. *Similarity – Weighted Function*) – k -artimiausių kaimynų klasifikavimo metodo tikimybės funkcijos rūšis.

IVADAS

Per paskutinius dešimtmečius naujos kompiuterinės regos technologijos ir vaizdų apdorojimas tapo ypač svarbūs automatizuojant rankinius procesus daugelyje technikos sričių. Pavyzdžiui, pramonės automatizacijoje vis svarbesnę dalį užima vaizdų apdorojimo sistemos. Vaizdų apdorojimo sistemos integruojamos į kitas automatizuotas sistemas ir taip padeda įvykdyti vis sudėtingesnes užduotis pakeičiant žmogų: krovinių perkėlimas panaudojant robotą (objektų radimas ir atstumo nustatymas) [1], defektų radimas gaminiuose [2] bei informacijos identifikavimas [3].

Informacijos identifikavimo arba optinio simbolių atpažinimo (angl. *Optical character recognition*) technologijos yra smarkiai pažengusios bei plačiai naudojamos: spausdinto bei ranka rašyto teksto nuskaitymas [4], teksto nuskaitymas ir tarimas akliems [5], automobilio numerių identifikavimas [6]. Informacijos identifikavimo sistemos vis labiau pradamos naudoti pramonėje nuskaitant brūkšninius kodus bei tekstinę informaciją nuo gaminių. Nuskaityta informacija dažniausiai panaudojama dviem tikslams: gaminio atsekamumui ir informacijos tikslumo patikrinimui. Augant pramonei bei automatizacijos lygiui, atsekamumo sistemos bei tikslios informacijos užtikrinimas tampa vis svarbesni.

Informacijos identifikavimo pramonėje tema atlikta ganėtinai mažai tyrimų bei parašyta nedaug straipsnių, lyginant su kitomis sritimis. Dėl šios priežasties, darbe pasirinkta atlikti plačiai pramonėje naudojamo *Videojet* rašalinio čiurkšlinio spausdintuvo užpurškiamos informacijos identifikavimo tyrimą. Tyrimo dalyje ištirtas deskriptorių ir klasifikatorių tikslumas bei sparta identifikuojant užpurkštą informaciją ant pakuotės.

Darbo tikslas – sukurti pramoninio rašalinio čiurkšlinio spausdintuvo užpurškiamos informacijos ant pakuotės identifikavimo algoritmą ir ištirti jo veikimą, panaudojus skirtingus požymių deskriptorius bei klasifikatorius.

Darbo uždaviniai:

1. atlikti vaizdų apdorojimo literatūros analizę informacijos identifikavimo tema bei apžvelgti požymių deskriptorių ir klasifikatorių metodiką;
2. sudaryti teksto identifikavimo, naudojantis vaizdų apdorojimo metodais, algoritmą;
3. ištirti simbolių identifikavimo algoritmo tikslumą ir spartą, panaudojus skirtingų deskriptorių ir klasifikatorių kombinacijas;
4. ištirti simbolio matricos dydžio poveikį algoritmo tikslumui ir spartai;
5. ištirti dešimtį geriausių deskriptorių ir klasifikatorių kombinacijų, panaudojus 50 skirtingų užpurkštos informacijos ant pakuotės nuotraukų;
6. ištirti mokymo duomenų bazės dydžio įtaką algoritmo tikslumui ir spartai.

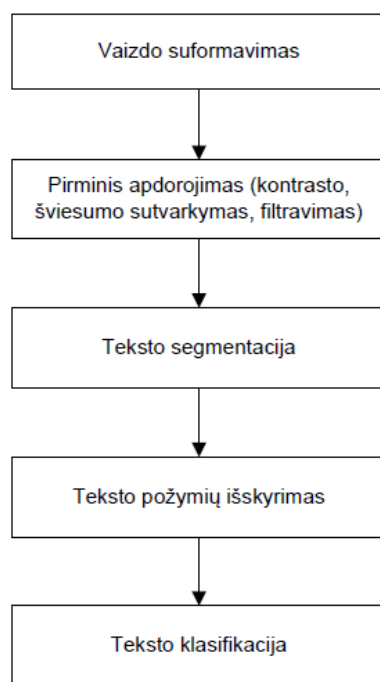
1 APŽVALGINĖ DALIS

Kompiuterinė rega yra tarpdisciplininė sritis, kurioje kuriami ir tiriama algoritmai, kurie padeda kompiuteriams suvokti aplinką iš skaitmeninių nuotraukų arba vaizdo įrašų. Kompiuterinės regos pradininku yra laikomas Laris Robertsas (Larry Roberts), kuris savo disertacijoje (1960) aprašė 3D geometrinės informacijos išskyrimo galimybes panaudojant 2D vaizdo blokus [7]. Kompiuterinė rega susilaukė ypač didelio dėmesio devintajame dešimtmetyje, kai buvo pradėta naudoti pažangesni metodai, kurie padidino algoritmų greitį. Kompiuterinę regą galima suskirstyti į du pagrindinius etapus: vaizdo suformavimą ir vaizdo apdorojimą [8].

Vaizdo suformavimo metu trijų dimensijų aplinkos vaizdas perkeliama į dviejų dimensijų matricą. Panaudojus matematinius metodus iš dviejų dimensijų matricos išgaunama informacija apie trijų dimensijų aplinką. Svarbu pabrėžti, jei vaizdui suformuoti naudojama viena kamera, tai prarandama informacija apie gylį, dėl šios priežasties neįmanoma nustatyti, koks atstumas yra iki objekto.

Vaizdo apdorojimo dalis yra dažnai vadinama vaizdo atpažinimo dalimi. Ši dalis ypač svarbi, nes nuo vaizdo atpažinimo dalyje panaudotų metodų priklauso sistemos greitaveika ir tikslumas. Dažniausiai kameros suformuota dvimatė vaizdo matrica nėra tinkama vaizdo apdorojimui, nes joje daug perdėtinos informacijos, kurią reikia nufiltruoti, taip pat dažnai pasitaiko netinkamas kontrastas ir šviesumas. Suformavus dvimatę matricą, dažnas reiškinys yra triukšmas – balti ir juodi taškai. Norint pašalinti triukšmą, būtina naudoti filtrus. Vienas iš geriausių filtrų baltų ir juodų taškų triukšmo panaikinimui yra medianos filtras [9].

Norint atpažinti tekstą, po nuotraukos filtracijos, reikia atlikti teksto segmentaciją, kurios metu iš teksto eilutės išskiriami simboliai. Tada panaudojus deskriptorių aprašomos kiekvieno simbolio vaizdo savybės. Deskriptoriaus gauti duomenys pateikiami apmokytam klasifikatoriui, kuris pagal simbolio požymius nurodo, kokiai simbolių klasei jis priklauso [10]. Vaizdo apdorojimo žingsniai tekstui atpažinti, nuo vaizdo suformavimo iki teksto klasifikavimo, pateikti 1.1 paveiksle.



1.1 pav. Vaizdo apdorojimo žingsniai tekstui atpažinti

Teksto atpažinimo algoritmo greitis ir tikslumas ypač priklauso nuo tinkamai pasirinkto požymių deskriptoriaus bei klasifikatoriaus. Todėl toliau darbe detaliau aprašyti populiariausi požymių deskriptoriai ir klasifikatoriai.

1.1 Požymių deskriptoriai

Požymių deskriptorius (angl. *Feature descriptor*) – tai algoritmas, kuris tiriamą vaizdą aprašo duomenų vektoriumi pagal vaizdinius požymius, kurie apibūdina elementariausias vaizdo charakteristikas: formą, spalvą ir tekstūrą. Gautas duomenų vektorius gali būti panaudojamas objektų radimui (figūrų, veidų atpažinimas) arba klasifikavimui (simbolių atpažinimas). Požymių deskriptoriai yra skirstomi į dvi grupes: bendros ir konkrečios srities informacijos deskriptorius. Bendros informacijos deskriptoriai dar yra vadinami globalių požymių deskriptoriais (angl. *Global feature descriptor*), o konkrečios informacijos deskriptoriai – lokalių požymių deskriptoriais (angl. *Local feature descriptor*).

Lokalių požymių deskriptoriai aprašo esmines vaizdo savybes. Šie deskriptoriai dažniausiai naudojami veidų, objektų atpažinimo problemoms spręsti. Dažniausiai naudojami lokalūs požymių deskriptoriai yra SURF, SIFT ir *Haar-like*.

Globalių požymių deskriptoriai aprašo viso vaizdo savybes ir jų vektoriai dažniausiai naudojami klasifikavimo uždaviniams spręsti. Vieni iš populiariausių globalių požymių deskriptorių algoritmų yra HOG, LBP ir PCA algoritmai. Kadangi darbo tyrimo tikslas glaudžiai susijęs su klasifikavimo uždaviniu, tai darbe detaliau bus aptarti globalių požymių deskriptoriai.

1.1.1 HOG deskriptorius

HOG (angl. *Histogram of Oriented Gradients*) orientuotų gradientų histogramų algoritmo pagrindinė idėja – lokalių gradientų intensyvumo pasiskirstymas [11]. Pirmą kartą orientuotų gradientų histogramų algoritmas pasiūlytas Navnito Dalalo (Navneet Dalal) ir Bilo Trigso (Bill Triggs) [12]. Tada HOG požymių deskriptorius buvo panaudotas atpažinti žmones nuotraukose. Tačiau po kiek laiko HOG algoritmas buvo pradėtas naudoti ir teksto simbolių atpažinimui [13]. Orientuotų gradientų histogramų požymių deskriptorius išsiskiria iš kitų požymių algoritmų tuo, kad neieško visos nuotraukos požymių, o iš karto suskirsto nuotrauką į mažesnius regionus [14]. Būtent nuo nuotraukos dalijimo į mažus regionus, kurie vadinami elementais, ir prasideda HOG algoritmo vykdymas. Tada skaičiuojamos kiekvieno elemento gradiento orientacijos histogramos. Toliau atliekamas kontrasto normalizavimas pagal rastą lokalinės histogramos amplitudę.

Visų pirma, gradiento skaičiavimo metu yra apskaičiuojamos gradiento įėjimo komponentės:

$$I_x(r, c) = I(r, c + 1) - I(r, c - 1), \quad (1)$$

$$I_y(r, c) = I(r - 1, c) - I(r + 1, c). \quad (2)$$

Apribojus gradiento kampo kitimą nuo 0 iki 180 laipsnių, gradientas perkeliamas į polinę koordinatinių sistemą ir randamas gradiento stiprumas μ bei kryptis θ :

$$\mu = \sqrt{I_x^2 + I_y^2}, \quad (3)$$

$$\theta = \frac{180}{\pi} (\tan^{-1}(I_y, I_x) \bmod \pi). \quad (4)$$

Nuotrauka dalijama į mažesnius regionus pritaikius $C \times C$ pikselių dydžio tinklelį. Kiekvienam regionui randama gradiento krypties histograma, kuri suskaidoma į B intervalus. Intervalai prasideda nuo 0 iki $B - 1$, o jų plotis apskaičiuojamas naudojant (5) formulę.

$$w = \frac{180}{B}. \quad (5)$$

Siekiant išvengti ryškumo ir kontrasto pokyčių poveikio, krypties histogramos yra normalizuojamos sujungiant langelius į didesnius blokus [12][15]:

$$b \leftarrow \frac{b}{\sqrt{\|b\|^2 + \epsilon}}. \quad (6)$$

Čia: ϵ yra labai mažas teigiamas dydis.

Šie blokai dažniausiai persidengia, kad užtikrintų, jog kiekvienas langelis patenka daugiau nei vieną kartą į galutinį požymio deskriptorių. Tokio tipo persidengimas suteikia tvirtumą nuotraukos transformacijai.

1.1.2 LBP deskriptorius

LBP (angl. *Local Binary Pattern*) yra tvirtas teksto požymių deskriptorius. Šis metodas pirmą kartą pasiūlytas Timo Ojalos (Timo Ojala) [16], kuris šį požymių deskriptorių panaudojo atpažinti tekstūros struktūras. Viena išpūdingiausių LBP metodo charakteristikų yra tvirtumas monotoniškiems pilkų atspalvių pokyčiams, kurie atsiranda keičiantis apšvietimo sąlygoms [17]. Kita labai svarbi savybė – skaičiavimų paprastumas, kuri leidžia panaudoti deskriptorių realaus laiko sistemoms. Dėl puikių savybių LBP metodas yra dažnai integruojamas į kompiuterinės regos sistemas: tekstūros struktūrų atpažinimą [18], veidų atpažinimą [19], žmonių ir judančių daiktų aptikimą bei teksto atpažinimą [20].

LBP deskriptorius taiko kiekvienam kaimyniniam pikseliui ribos metodą, kurio slenkstinė vertė lygi centrinio pikselio reikšmei. Pritaikius ribos metodą yra gaunamos dvejetainės reikšmės: 0 ir 1, kur 0 reiškia, kad kaimyninio pikselio reikšmė yra mažesnė nei centrinio pikselio, o 1 reiškia atvirkščią situaciją. Visi kaimyniniai pikseliai yra sujungiami į vektorių, kuris atitinka dvejetainį skaičių. Gautas skaičius konvertuojamas į dešimtainį skaičių, kuris nusako centrinio pikselio LBP reikšmę. LBP reikšmės radimo procesas yra aprašomas matematinėmis išraiškomis:

$$s(x) = \begin{cases} 1 & \text{jei } x \geq 0 \\ 0 & \text{jei } x < 0 \end{cases} \quad (7)$$

$$LBP = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p. \quad (8)$$

Čia: $s(x)$ yra ribos funkcija, o g_c ir g_p reiškia centrinio pikselio pilkumo atspalvio ir p kaimyno pikselio reikšmes.

Toliau p kaimyninio pikselio dvejetainė reikšmė yra padauginama iš atitinkančio svorinio koeficiento 2^p . Susumavus visas kaimyninių pikselių reikšmes yra apskaičiuojama LBP reikšmė. Šis veiksmas pavaizduotas 1.2 paveiksle, kur dvejetainė struktūra 01110101 yra paverčiama į $LBP = 1 + 4 + 16 + 32 + 64 = 117$.

6	2	7
4	5	3
5	8	9

(a)

1	0	1
0	X	0
1	1	1

(b)

1	2	4
128	X	8
64	32	16

(c)

1.2 pav. LBP skaičiavimo pavyzdys: (a) nuotraukos pavyzdys, (b) dvejetainė struktūra po ribų pritaikymo, (c) svorių matrica [17]

LBP deskriptorius negali identifikuoti informacijos pozicijos, todėl būtina padalinti visą nuotrauką į keletą blokų. Kiekvienas blokas yra apdorojamas LBP deskriptoriumi ir apskaičiuojamos kiekvieno bloko histogramos. Blokų histogramų reikšmės sujungiamos tarpusavyje į vektorius. Gautas vektorius aprašo visos nuotraukos informaciją. Norint palyginti dvi nuotraukas yra skaičiuojamas *Chi* kvadrato atstumas [20]:

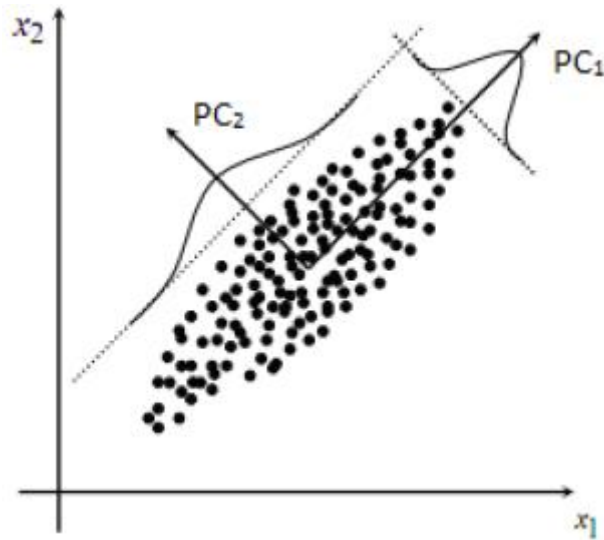
$$dist = \sum_{k=1}^N \frac{(h_i(k) - h_j(k))^2}{h_i(k) + h_j(k)}. \quad (9)$$

Yra keletas LBP metodo formų, kurios skiriasi kaimyninių taškų pozicija, orientacija ir atstumu. Populiariausi LBP metodo algoritmai yra pagrindinis LBP (angl. *Basic*), tolygus LBP (angl. *Uniform*) ir supaprastintas LBP (angl. *Simplified*).

1.1.3 PCA deskriptorius

PCA (angl. *Principal Component Analysis*) pagrindinių komponentių analizės metodas yra vienas iš klasikinės statistikos metodų. Pagrindinė šio metodo idėja yra atlikti tiesinę transformaciją ir sumažinti duomenų matmenų skaičių, atmetant dalį po transformacijos rastų komponentių, kurios turi mažiausią dispersiją [21][22]. Šis metodas yra plačiai naudojamas signalų procesuose, statistikoje ir neuroninių tinklų skaičiavimuose. PCA metodas yra dažnai vadinamas diskretine *K-L* transformacija [23], nes jis sumažina duomenų skaičių, aprašanti pavyzdžio požymius.

Visų pirma, pagrindinių komponentių analizės deskriptorius ieško pagrindinės komponentės PC_1 (žr. 1.3 pav.), kurios dispersija yra didžiausia [21]. Ši komponentė kerta duomenų centrinį tašką ir yra arčiausiai visų duomenų taškų, nes vidutinis atstumas nuo komponentės iki taškų yra minimalus. Antroji pagrindinė komponentė PC_2 taip pat kerta duomenų centrinį tašką bei yra statmena pirmajai pagrindinei komponentei.



1.3 pav. Duomenys ir PCA pagrindinės komponentės

Kiekviena pagrindinė komponentė aprašoma tikriniais vektoriais (angl. *eigenvector*) ir tikrine reikšme (angl. *eigenvalue*). Tikrinis vektorius nurodo pagrindinės komponentės kryptį, o tikrinė reikšmė nusako, kaip duomenys yra pasiskirstę šia kryptimi [24].

Duota duomenų matrica, kurios eilutės aprašo objektus, o stulpeliai – objektus nusakančius parametrus:

$$X = \{X_1, X_2, \dots, X_m\} = \{x_{ij}, i = 1, \dots, m, j = 1, \dots, n\}. \quad (10)$$

Apskaičiuojamas koreliacijos koeficientas, kuris nusako parametų x_k ir x_l tarpusavio ryšį:

$$r_{kl} = \frac{\sum_{i=1}^m (x_{ik} - \bar{x}_k)(x_{il} - \bar{x}_l)}{\sqrt{\sum_{i=1}^m (x_{ik} - \bar{x}_k)^2 \sum_{i=1}^m (x_{il} - \bar{x}_l)^2}}. \quad (11)$$

Čia: $\bar{x}_k = \frac{1}{m} \sum_{i=1}^m x_{ik}$, $\bar{x}_l = \frac{1}{m} \sum_{i=1}^m x_{il}$.

Iš apskaičiuotų koreliacijos koeficientų sudaroma koreliacinė matrica $R = \{r_{kl}, k, l = 1, \dots, n\}$.

Apskaičiuojamas kovariacijos koeficientas c_{kl} :

$$c_{kl} = \frac{1}{m-1} \sum_{i=1}^m (x_{ik} - \bar{x}_k)(x_{il} - \bar{x}_l). \quad (12)$$

Iš apskaičiuotų kovariacijos koeficientų sudaroma kovariacijos matrica $C = \{c_{kl}, k, l = 1, \dots, n\}$.

Turint kovariacijos matricą bei išsprendus (13) lygtį galima rasti tikrinius vektorius E_k ir tikrines reikšmes λ_k .

$$CE_k = \lambda_k E_k. \quad (13)$$

Tikrinės reikšmės λ_k randamos išsprendus charakteringą lygtį:

$$|C - \lambda_k I| = 0. \quad (14)$$

Čia: I – vienetinė matrica, kurios matmenys kaip matricos C .

Suradus tikrinius vektorius E_k , jie surūšiuojami tikrinių reikšmių mažėjimo tvarka ($\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_n$). Atlikus rūšiavimą, randama pagrindinių komponentių matrica $A = (E_1, E_2, \dots, E_n)$. Norint perkelti duomenis į naują ortogonalią koordinačių sistemą, kurios ašys yra pagrindinės komponentės, reikia transformuoti duomenų vektorius:

$$Y_i = (X_i - \bar{X})A. \quad (15)$$

Pagrindinių komponentių analizės algoritmą galima atlikti keturiais žingsniais [25]:

1. Uztikrinti, kad visos savybės turi vienodą vidurkį ir kintamumą.
2. Rasti kovariacijos matricą C .
3. Suskaičiuoti kovariacijos matricos C tikrinius vektorius E_k ir tikrines reikšmes λ_k .
4. Surūšiuoti tikrines reikšmes λ_k mažėjimo tvarka ir iš tikrinių vektorių E_i pagal gautą sąrašą sudaryti pagrindinių komponentių matricą A .

1.1.4 Pikselių vektorius

Vienas iš paprasčiausių būdų vaizdui aprašyti – panaudoti pikselių vektorių. Pikselių vektorius gaunamas konvertuojant vaizdo matricą į vektorių. Matricos eilutės yra sujungiamos į vektorių, kurio stulpelių skaičius lygus matricos eilučių ir stulpelių sandaugai. Toliau pateikta matricos A konvertavimo į vektorių \vec{u} pavyzdys.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix};$$

$$\vec{u} = f(A) = (a_{11} \ a_{12} \ a_{13} \ a_{21} \ a_{22} \ a_{23} \ a_{31} \ a_{32} \ a_{33}).$$

Pikselių vektorių metodas nėra pats efektyviausias naudoti, kai aprašomo vaizdo pikselių skaičius yra didelis, nes algoritmas neišskiria svarbios informacijos, o tik konvertuoja visą vaizdą į dvejetainį kodą. Tačiau, jei vaizdo pikselių skaičius mažas, galima gauti puikius rezultatus.

1.1.5 SIFT deskriptorius

SIFT (angl. *Scale-invariant feature transform*) algoritmas naudojamas lokalių savybių suradimui ir apibrėžimui nuotraukoje. SIFT deskriptorius efektyviai išnaudojamas panaudojant kelių stadijų filtravimo būdą [11]. Pirmoje stadijoje yra konstruojama mastelio erdvė. Šio proceso metu naudojami Gauso branduoliai. Gauso branduoliai naudoja Laplaso savybę, kuri yra naudinga, nes ji stabili ir duoda puikius rezultatus. Ši savybė aprašoma:

$$L(x, y, \sigma) = G(x, y, \sigma) \cdot I(x, y), \quad (16)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}. \quad (17)$$

Gauso skirtumas *DoG* (angl. *Difference of Gaussians*) yra naudojamas sukurti *DoG* piramidę. Nuotrauka yra skenuojama keičiant mastelį ir ieškant *DoG* visuose jos kaimyniniuose taškuose. Šis procesas identifikuoja minimumo ir maksimumo reikšmes. Kitas žingsnis yra briaunų reakcijų eliminavimas, kai pikų reakcijos didžiausios pagal vaizdo kraštus ir mažiausios pagal kitas kryptis. Po briaunų reakcijų eliminavimo atliekamas pagrindinių taškų orientacijos priskyrimas.

Nepaisant to, kad SIFT deskriptorius pasižymi aukštu tikslumu, tačiau jis yra ganėtinai lėtas lyginant su kitais deskriptoriais, o tai sulėtina viso algoritmo veikimą [26]. Patobulinta ir daug spartesnė SIFT deskriptoriaus forma yra SURF deskriptorius [27].

1.1.6 SURF deskriptorius

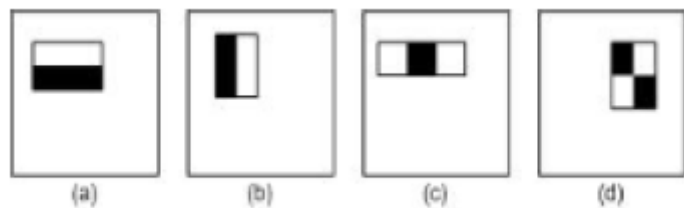
SURF (angl. *Speeded Up Robust Features*) požymių deskriptorius buvo sukurtas Herberto Bayjo [28], siekiant pagreitinti tikslumu pasižyminti SIFT deskriptorių. SURF deskriptorius naudoja *Hessian* matrica, bet tuo pačiu turi ir labai paprastą aproksimaciją. Šis metodas naudoja vientisą (angl. *integral*) nuotrauką, kad būtų sumažintas skaičiavimo laikas ir todėl jis dar yra vadinamas greitu *Hessian* požymių radimo metodu. SURF požymių deskriptorius išskiria, koduoja būdinguosius taškus neatsižvelgiant į požymio mastelį ir pasukimo kampą. Šis algoritmas dažniausiai naudojamas objektų sekimui ir jo greitaveika priklauso nuo rastų požymių skaičiaus, požymio pasukimo kampo, mastelio ir naudojamų dvimačių filtrų [29].

SURF metodas susideda iš kelių žingsnių. Pirmame žingsnyje yra fiksuojama atkuriamoji orientacija, kuri remiasi informacija iš apvalaus regiono apie dominuojančius taškus. Toliau konstruojamas kvadratinis regionas, sulygiuotas pasirinktai orientacijai, ir išskiriamas SURF požymių aprašas.

SURF požymių deskriptorius dėl *Hessian* matricos naudojimo yra daugiau nei 3 kartus greitesnis lyginant su SIFT deskriptoriumi, kuris naudoja Gauso skirtumą *DoG*. Tačiau tiek SURF, tiek SIFT deskriptorių gaunami rezultatai pasižymi panašiu tikslumu [28].

1.1.7 *Haar-like* deskriptorius

Haar-like požymių deskriptorius puikiai atpažįsta veidus, tačiau gali būti naudojamas ir informacijos laukų lokalizavimui, o tai labiausiai priklauso nuo pasirinktų mokymo pavyzdžių [6]. Yra nemažai priežasčių, kodėl naudoti *Haar-like* požymių deskriptorių yra geriau nei pikselių vektorių tiesiogiai [30]. Pagrindinė priežastis yra ta, kad savybėmis paremta sistema dirba daug greičiau lyginant su pikseliais paremta sistema. *Haar-like* deskriptorius skaičiuoja stačiakampių pikselių plotų skirtumus, kai stačiakampių pozicija ir dydis kinta lyginant su originalia nuotrauka [31]. Dažniausiai naudojami keturi *Haar-like* metodo tipai [30] (žr. 1.4 pav.). Dviejų stačiakampių (a) ir (b) savybės reikšmės yra lygios skirtumui tarp kvadratų pikselių sumų. Stačiakampiai yra vienodo dydžio, tik skiriasi padėtis.



1.4 pav. *Haar-like* deskriptoriaus savybių tipai [30]

Trijų stačiakampių (c) savybės reikšmė apskaičiuojama iš šoninių stačiakampių pikselių sumos reikšmės atimant vidurinio stačiakampio pikselių sumos reikšmę. Keturių stačiakampių (d) savybės reikšmė yra lygi įstrižainių pikselių sumų reikšmių skirtumui. Šių savybių pliusas yra tas, kad jos gali būti lengvai perskaičiuojamos išvengiant nuotraukų piramidės skaičiavimo, kas pagreitina deskriptorių.

Esminis *Haar-like* deskriptoriaus privalumas, lyginant su kitais deskriptoriais, yra nedidelis skaičiavimo sudėtingumas. *Haar-like* metodas gali apskaičiuoti požymius per pastovų laiką, nepriklausomai nuo vaizdo dydžio, kadangi deskriptorius naudoja vientisas (angl. *integral*) nuotraukas [31].

1.2 Požymių klasifikatoriai

Klasifikavimas – tai duomenų priskyrimas tam tikrai klasei pagal požymius, kai duomenų klasė nėra žinoma. Pavyzdžiui, atpažįstant simbolius, tai yra nuotraukos segmento priskyrimas tam tikrai simbolio reikšmei. Kita labai artima klasifikavimui metodo rūšis yra klasterizavimas. Klasterizavimas – tai yra duomenų suskirstymas į klases pagal panašumas. Klasifikavimą dažniausiai sudaro keli etapai. Pirmas etapas yra klasifikatoriaus apmokymas, kai pateikiami duomenys su klasėmis. Pagal duotus duomenys sudaromas klasifikatoriaus modelis. Kitas etapas – apmokyto klasifikatoriaus vertinimas, kurio metu skaičiuojamas klasifikatoriaus efektyvumas.

Kadangi klasifikavimo uždavinys yra gan dažnai taikomas, todėl sukurta labai daug klasifikavimo metodų. Pasitaiko, kad skirtingi klasifikavimo metodai yra sujungiami, kad būtų gauti geresni rezultatai. Toliau darbe bus trumpai aprašyti vieni iš populiariausių klasifikavimo metodų.

1.2.1 *K*–artimiausių kaimynų (angl. KNN) klasifikatorius

KNN (angl. *k-Nearest Neighbors*) *k*–artimiausių kaimynų klasifikatorius yra vienas iš paprasčiausių klasifikavimo algoritmų [32], kuris yra efektyvus ir neturi parametrų. Jis nereikalauja nuoseklių duomenų bei galima naudoti duomenis, kuriuose egzistuoja triukšmas [33]. Šio metodo esmė yra priskirti naujus duomenis klasei nustatant, kaip toli šie duomenys yra nutolę nuo *k* skaičiaus artimiausių kaimynų. Pagal šių artimiausių kaimynų klasę ir yra parenkama klasė naujiems duomenims.

Toliau aprašomas *k*–artimiausių kaimynų algoritmo veikimas, ieškant *X* bandinio klasės [34]. Iš viso yra *j* skaičius treniravimo duomenų klasių C_1, C_2, \dots, C_j , ir visų treniravimo duomenų rinkinių skaičius yra lygus *N*. Panaudojus treniravimo duomenis KNN klasifikatoriaus duomenų struktūros kūrimui, gautas *m* dimensijų vektorius. Toliau bandinys *X* pertvarkomas į tokią pačią struktūrą, kaip treniravimo duomenų vektorius (X_1, X_2, \dots, X_m) . Apskaičiuojami panašumai tarp treniravimo duomenų ir bandinio *X*. Panašumas tarp bandinio *X* ir treniravimo duomenų *i*–tojo bandinio $d_i (d_{i1}, d_{i2}, \dots, d_{im})$ apskaičiuojamas pagal (18) formulę.

$$SIM(X, d_i) = \frac{\sum_{j=1}^m X_j \cdot d_{ij}}{\sqrt{(\sum_{j=1}^m X_j)^2 \cdot (\sum_{j=1}^m d_{ij})^2}}. \quad (18)$$

Apskaičiavus panašumus, išrenkamas *k* skaičius treniravimo duomenų bandinių, kurių panašumai $SIM(X, d_i)$ yra didžiausi ir suformuojama *X* bandinio KNN rinkinys. Pagal išrinktus duomenis yra apskaičiuojama tikimybė, kiek *X* bandinys priklauso kiekvienai klasei. Dažniausiai

tikimybei apskaičiuoti naudojamos 3 formulės: DVF (angl. *Discrete – Valued Function*) (19), SWF (angl. *Similarity – Weighted Function*) (20) ir ASWF (angl. *Average Similarity – Weighted Function*) (21) [35].

$$P(X, d_i) = \sum_{i=1}^k y(d_i, C_j), \quad (19)$$

$$P(X, d_i) = \sum_{i=1}^k SIM(X, d_i) y(d_i, C_j), \quad (20)$$

$$P(X, d_i) = \frac{\sum_{i=1}^k SIM(X, d_i) y(d_i, C_j)}{\sum_{i=1}^k y(d_i, C_j)}. \quad (21)$$

Čia: $y(d_i, C_j)$ yra klasės priskyrimo funkcija (22).

$$y(d_i, C_j) = \begin{cases} 1, & d_i \in C_j \\ 0, & d_i \notin C_j \end{cases}. \quad (22)$$

Bandinys X priskiriamas klasei, kurios tikimybė yra didžiausia.

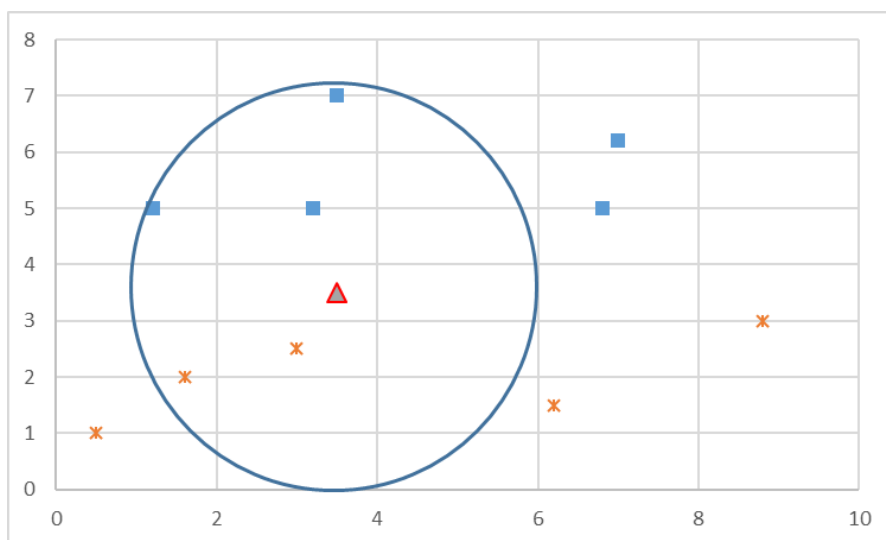
KNN klasifikavimo metodo naudojimas priklauso nuo duomenų parametrų skaičiaus. Kuo šis skaičius didesnis, tuo didesnis artimiausių kaimynų skaičius yra parenkamas. Didesnis artimiausių kaimynų skaičius užtikrina didesnę tikslumą ir sumažina bendro triukšmo įtaką. Optimalus artimiausių kaimynų skaičius didžiąjai duomenų daliai yra tarp 3 – 10. Kai nagrinėjamų duomenų parametrų skaičius nėra didelis (pavyzdžiui 2 ar 3), tai artimiausių kaimynų klasifikavimo metodo veikimą galima atvaizduoti vizualiai.

Toliau pateikiamas KNN klasifikavimo pavyzdys. Duota 10 duomenų (žr. 1.1 lentelę), kurie turi 2 parametrus ir yra priskirti vienai iš dviejų klasių (1 arba 2).

1.1 lentelė. Duomenys ir jiems priskirtos klasės

Parametras 1	Parametras 2	Klasė
1.2	5	1
3.5	7	1
0.5	1	2
1.6	2	2
6.8	5	1
3	2.5	2
8.8	3	2
6.2	1.5	2
7	6.2	1
3.2	5	1

Reikia nustatyti bandinio klasę, kurio parametrai 3,5 ir 3,5. 1.1 Lentelėje pateikti duomenys atvaizduoti dvimatėje erdvėje 1.5 paveiksle (1 klasė – kvadratai, 2 klasė – žvaigždutės, nepriskirtas duomuo – trikampis). Panaudojus artimiausių kaimynų metodą pasirinkti 5 kaimynai. 1.5 paveiksle matyti, kad iš 5 artimiausių kaimynų 3 priklauso pirmajai klasei, o 2 – antrajai klasei, todėl nauji duomenys yra priskiriami pirmajai klasei.

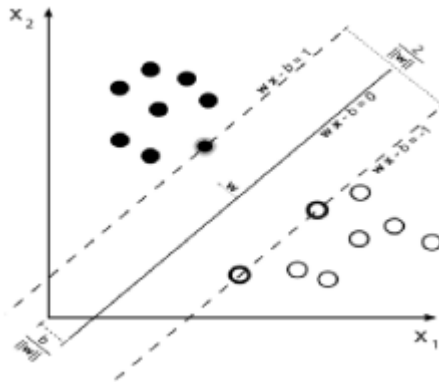


1.5 pav. Klasės nustatymas artimiausių kaimynų metodu, 1 klasė – kvadratai, 2 klasė – žvaigždutės, nauji duomenys – trikampis

Tradicinis KNN klasifikavimo algoritmas matuoja panašumą tarp teksto ir treniravimo pavyzdžių. Tai užtrunka nemažai laiko ir dėl tos priežasties pasitaiko atvejų, jog klasifikavimas netenkina laiko reikalavimų. Todėl buvo pradėta ieškoti KNN klasifikatoriaus optimizavimo metodų. Šiuos metodus galima suskirstyti į tris klases [34]: duomenų vektoriaus dydžio mažinimą; mažesnio kiekio duomenų naudojimą; algoritmo patobulinimą siekiant pagreitinti k artimiausių kaimynų radimą [36].

1.2.2 Atraminės vektorių mašinos (angl. SVM) klasifikatorius

SVM (angl. *Support Vector Machine*) klasifikavimo metodas sukurtas pagal struktūrinę rizikos mažinimo teoriją [37]. Pagrindinė SVM idėja yra sudaryti įėjimo vektorių žemėlapi panaudojant kelių dimensijų savybių erdvę ir šioje erdvėje sukonstruoti optimalią atskiriančią hyperplokštumą. SVM tikslas – maksimaliai sumažinti klasifikavimo paklaidą maksimizuojant atstumą tarp skirtingų klasių duomenų, kurie vadinami atraminiais vektoriais, hyperplokštuma. Vadinasi, geras atskyrimas laikomas tada, kai gaunama hyperplokštuma (žr. 1.6 pav.), kuri turi didžiausią atstumą iki artimiausių skirtingų klasių duomenų taškų. Kuo didesnis atstumas, tuo mažesnė klasifikavimo paklaida.



1.6 pav. SVM dviejų klasių atskyrimas hyperplokštuma [37]

Duomenų klasifikavimas yra labai dažnai naudojamas įrengimų mokymui. Duotus duomenų taškus reikia priskirti, kuriai klasei jie priklauso. Žiūrint iš atraminių vektorių mašinų (angl. SVM) metodo pusės, duomenų taškai yra laikomi p – dimensijos vektoriumi ir tikslas yra žinoti ar juos galima atskirti $p-1$ dimensijos hyperplokštuma. Tai vadinama tiesiniu klasifikavimu. Susidaro nemažas skaičius hyperplokštumų, bet pats geriausias būdas yra pasirinkti hyperplokštumą, kuri skiria dvi klases maksimaliu atstumu. Nauji duomenys yra priskiriami klasei atsižvelgiant, į kurios klasės ribas jie patenka. Tam užtenka atraminių vektorių taškų.

Atraminių vektorių mašinų klasifikavimo algoritmas yra pranašesnis lyginant su tradiciniais statistiniais apmokymo algoritmais [38]. SVM algoritmas pasižymi spartos stabilumu nepriklausomai nuo požymių vektoriaus dydžio. Du pagrindiniai SVM elementai, taikant SVM, yra matematinis programavimas ir branduolio funkcijos [39]. Parametrai yra randami išsprendžiant kvadratinę lygtį su tiesine priklausomybe ir nelygybės apribojimais. Atraminių vektorių mašinų algoritmą galima aprašyti toliau pateiktomis formulėmis. Duoti mokymų duomenys:

$$S = \{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}, x_i \in R^n, y_i \in \{-1, +1\}. \quad (23)$$

Čia: x_i – i mokymo duomenų nario savybių vektorius (n dydžio). y_i – i mokymo duomenų nario klasė. l yra mokymo duomenų narių skaičius. SVM metodas ieško optimalios hyperplokštumos (24), kuri atskiria duotas dvi klases.

$$(wx + b) = 0. \quad (24)$$

Optimali hiperplokštuma gali būti sudaroma randant kvadratinės lygties (25) maksimumą:

$$f(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i y_i \alpha_j y_j K(x_i \cdot x_j), \quad (25)$$

Jei $-\sum_{i=1}^l y_i \alpha_i = 0, 0 \leq \alpha_i \leq c, i = 1, 2, \dots, l$.

Funkcija $K(x_i, x_j)$ yra vadinama branduolio funkcija ir yra aprašoma (26) formule:

$$K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j). \quad (26)$$

SVM algoritmas turi 3 tipų branduolio funkcijas: polinomo branduolio (angl. *polynomial kernel*) (27), RBF (angl. *Radial Basis Function*) branduolio (28) ir sigmoidinio branduolio (angl. *sigmoid kernel*) (29) [40].

$$k(x, x_i) = (\gamma x \cdot x_i + r)^2, \quad (27)$$

$$k(x, x_i) = e^{-\frac{1}{2\delta^2} \|x - x_i\|^2}, \quad (28)$$

$$k(x, x_i) = \tanh[\gamma(x \cdot x_i) + r]. \quad (29)$$

Paprastai atraminių vektorių mašinų klasifikavimo algoritmas yra naudojamas klasifikuoti dvi klases, tačiau gali būti panaudotas ir kelių klasių duomenų klasifikacijai. Dažniausiai yra naudojami du būdai tai padaryti: „vienos klasės prieš visas“ metodas bei „poravimo“ metodas. Pirmo metodo metu yra ieškamos hyperplokštumos tarp vienos klasės ir visų kitų duomenų, o „poravimo“ metodo metu klasės yra suporuojamos po dvi ir išrenkama klasė iš dviejų balsavimo būdu.

1.2.3 Dirbtinių neuroninių tinklų (angl. ANN) klasifikatorius

Dirbtinio neuroninio tinklo ANN (angl. *Artificial Neural Network*) algoritmas yra vienas iš dažniausiai naudojamų algoritmų atpažįstant vaizdus [41]. Šis klasifikavimo metodas naudoja idėją, kaip sąveikauja ir funkcionuoja neuronai žmogaus smegenyse [4]. Yra žinoma, kad žmogaus smegenys veikia lygiagrečiai atpažinimui, samprotavimui ir žalos atstatymui. Kadangi ANN sugeba susidoroti su išvardintais procesais, jis gali būti naudojamas nuo paprasto pritaikymo iki sudėtingų užduočių sprendimo, pavyzdžiui, klasifikavimo problemos sprendimo. Vienas neuronas tinkle vienu metu gali susijungti su daugiau nei 10 000 kitų neuronų ir generuoti bei dalintis nauja informacija. Neuronai yra susijungę su kitais neuronais tinkle per jungtis, kurios vadinamos sinapsėmis (angl. *synapse*). Neuronų tinklas yra masiškai lygiagrečiai pasiskirstęs modelis, kuris sudarytas iš paprastų apdorojančių vienetų, kurie turi natūralią savybę kaupti eksperimentines žinias ir jas panaudoti [42]. Tai primena žmogaus smegenis pagal du

dalykus: žinios yra kaupiamos mokymosi proceso metu, o svoriniai koeficientai yra naudojami, kad šias žinias įsiminti.

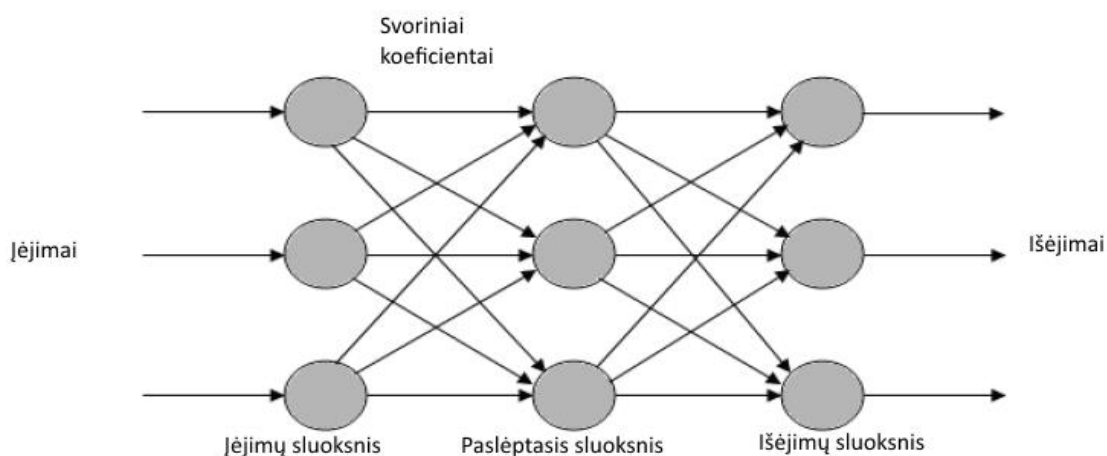
Dirbtinis neuroninis tinklas yra netiesinis matematinis modelis, kuris pasižymi paprastumu, lankstumu, dideliu apmokymo algoritmų pasirinkimu [43]. Sudėtingos netiesinės sistemos gali būti sumodeliuotos panaudojant nuspėtą koreliaciją tarp įėjimo ir išėjimo reikšmių. Vienas iš labiausiai paplitusių dirbtinių neuroninių tinklų modelių yra daugiasluoksniu perceptrono modelis MLP (angl. *Multilayer perception*). Paprastai MLP modelis susideda iš įėjimų sluoksniu, išėjimų sluoksniu ir vieno arba kelių paslėptų sluoksniu. Paslėpti sluoksniai veikia juodosios dėžės principu. Įėjimų sluoksniu perduoda signalą į išėjimo sluoksniu per paslėptus sluoksniu.

Neurono matematinis modelis yra aprašomas forma:

$$s = \omega_0 + \sum_{i=1}^p \omega_i x_i. \quad (30)$$

Čia: w_0 – pradinis svoris, w_i – svorio pokytis, o x_i – įėjimo reikšmė.

ANN modelio struktūra pavaizduota 1.7 paveiksle, kurią sudaro trys sluoksniu: įėjimo, paslėptas ir išėjimo. Kiekvienas sluoksniu susideda iš tam tikro skaičiaus neuronų ir kiekvienas neuronas turi svorinius koeficientus, kurie reikalingi tolimesniam apdorojimui.



1.7 pav. ANN modelio struktūra [4]

Norint rasti neuronų svorinius koeficientus w_i , reikia atlikti neuronų mokymą. Neurono mokymo metu į modelį paduodamas mokymo rinkinys, kurį sudaro įėjimų ir tikslų reikšmės. Tada yra skaičiuojama nuostolių funkcija c ir siekiama ją minimizuoti:

$$c = \sum_{i=1}^n (t_j - f(\omega_1 x_{j1} + \dots + \omega_p x_{jp} + \omega_0))^2. \quad (31)$$

Čia: n – mokymo duomenų skaičius, p – požymių kiekis, x_j – mokymo įėjimas, t_j – išėjimas/tikslas.

Neurono aktyvacija gali būti apskaičiuota kaip įėjimų svorinių reikšmių suma atėmus neurono slenksčio vertę. Tinklas generuoja išėjimą aktyvavimo signalui pereinant aktyvavimo arba perdavimo funkciją. Dažniausiai yra naudojama sigmoidinė (angl. *sigmoid*) aktyvavimo funkcija (32).

$$h_j = \frac{1}{1 + e^{-net_j}}. \quad (32)$$

Čia: net_j – yra tinklo įėjimas į neuroną i paslėptame sluoksnyje arba išėjimo sluoksni.

Įėjimų sluoksni apibūdina įėjimų kintamųjų reikšmes. O paslėptasis ir išėjimo sluoksni turi prijungtus neuronus prie ankstesnio sluoksni neuronų.

1.2.4 Naive Bajeso klasifikatorius

Naive Bajeso klasifikatorius (angl. *Naive Bayesian*) yra paprastas tikimybinis klasifikatorius, kuris sukurtas remiantis Naive Bajeso teorema. Šis algoritmas yra ypač tinkamas naudoti, kai duomenis aprašomų požymių skaičius yra didelis [44]. Nepaisant paprastumo, Naive Bajeso metodas dažnai pateikia geresnius rezultatus nei sudėtingi klasifikavimo metodai [45]. Bajeso teorema aprašoma formule:

$$P(Y|X) = \frac{P(X|Y) \cdot P(Y)}{P(X)}. \quad (33)$$

Čia: kur X – duomenys, o Y – klasė.

$P(Y)$ yra priorinė (angl. *prior*) klasės Y tikimybė, o $P(X)$ – priorinė duomenų X tikimybė. $P(Y|X)$ yra sąlyginė klasės Y tikimybė, kai yra žinomi X duomenys, o $P(X|Y)$ yra sąlyginė duomenų X tikimybė, kai yra žinoma, kad jie priklauso Y klasei. $P(X|Y)$ tikimybė kartais dar vadinama tikėtinumu (angl. *likelihood*).

Kai nėra galimybės apskaičiuoti priorinės duomenų X tikimybės, sąlyginė klasės tikimybė gali būti apskaičiuojama sudauginus skirtingų parametrų tikėtinumus ir priorinę klasės Y tikimybę:

$$P(Y|X) = P(X_1|Y) \cdot P(X_2|Y) \cdots P(X_n|Y) \cdot P(Y). \quad (34)$$

Apskaičiavus visas sąlygines klasių tikimybes, duomenims priskiriama klasė, kurios sąlyginė tikimybė yra didžiausia. Ši Naive Bajeso metodo savybė klasifikuoti naujus duomenis pagal labiausiai tikėtiną klasę yra paremta MAP (angl. *Maximum a Posteriori*) hipoteze [46].

Bajeso klasifikatoriaus veikimą puikiai iliustruoja pavyzdys, kurio duomenys pateikti 1.2 lentelėje. Duota 14 duomenų su 4 parametrais (oras: lietus, apsiniaukę, saulėta; temperatūra: karšta, šilta, šalta; drėgmė: aukšta, normali; vėjuota: ne, taip) ir dvi klases (eiti į lauką ir neiti į lauką).

1.2 lentelė. Duomenys Naive Bajeso klasifikatoriui

Oras	Temperatūra	Drėgmė	Vėjuota	Eiti į lauką
Lietus	Karšta	Aukšta	Ne	Ne
Lietus	Karšta	Aukšta	Taip	Ne
Apsiniaukę	Karšta	Aukšta	Ne	Taip
Saulėta	Šilta	Aukšta	Ne	Taip
Saulėta	Šalta	Normali	Ne	Taip
Saulėta	Šalta	Normali	Taip	Ne
Apsiniaukę	Šalta	Normali	Taip	Taip
Lietus	Šilta	Aukšta	Ne	Ne
Lietus	Šalta	Normali	Ne	Taip
Saulėta	Šilta	Normali	Ne	Taip
Lietus	Šilta	Normali	Taip	Taip
Apsiniaukę	Šilta	Aukšta	Taip	Taip
Apsiniaukę	Karšta	Normali	Ne	Taip
Saulėta	Šilta	Aukšta	Taip	Ne

Visų pirma, apskaičiuojamos priorinės klasių tikimybės $P(Y)$:

$$P(Y) = P(\text{Taip}) = \frac{9}{14};$$

$$P(Y) = P(\text{Ne}) = \frac{5}{14}.$$

Tada suformuojamos tikimybių dažnių lentelės (žr. 1.8 pav.):

		Eiti į lauką	
		Taip	Ne
Oras	Saulėta	3/9	2/5
	Apsiniaukę	4/9	0/5
	Lietus	2/9	3/5

		Eiti į lauką	
		Taip	Ne
Temperatūra	Karšta	2/9	2/5
	Šilta	4/9	2/5
	Šalta	3/9	1/5

		Eiti į lauką	
		Taip	Ne
Drėgmė	Aukšta	3/9	4/9
	Normali	6/9	1/5

		Eiti į lauką	
		Taip	Ne
Vėjuota	Ne	6/9	2/5
	Taip	3/9	3/5

1.8 pav. Tikimybių dažnių lentelės

Tikimybių dažnių lentelėse pateikti duomenys yra sąlyginės X duomenų tikimybės, kai žinoma klasė Y . Pavyzdžiui, kai oras buvo saulėtas, tai buvo pasirinkta išeiti į lauką 3 kartus iš 9:

$$P(X|Y) = P(\text{Saulėta}|\text{Taip}) = \frac{3}{9}.$$

Susumavus kiek iš viso kartų įvyko gautos priorinės X tikimybės $P(X)$ požymiai:

		Eiti į lauką		P(x)
		Taip	Ne	
Oras	Saulėta	3/9	2/5	5/14
	Apsinauķę	4/9	0/5	4/14
	Lietus	2/9	3/5	5/14
P(Y)		9/14	5/14	

		Eiti į lauką		P(x)
		Taip	Ne	
Temperatūra	Karšta	2/9	2/5	4/14
	Šilta	4/9	2/5	6/14
	Šalta	3/9	1/5	4/14
P(Y)		9/14	5/14	

		Eiti į lauką		P(x)
		Taip	Ne	
Drėgmė	Aukšta	3/9	4/9	7/14
	Normali	6/9	1/5	7/14
P(Y)		9/14	5/14	

		Eiti į lauką		P(x)
		Taip	Ne	
Vėjuota	Ne	6/9	2/5	8/14
	Taip	3/9	3/5	6/14
P(Y)		9/14	5/14	

1.9 pav. Tikimybių dažnių lentelės su apriorinėmis X ir Y tikimybėmis

Turint šiuos duomenis galima apskaičiuoti, kokia tikimybė išeiti į lauką, jei lauke oras saulėtas. Kadangi duotas tik vienas požymis, užtenka skaičiuoti tik vienos klasės tikimybę ir tam panaudojama (33) formulė:

$$P(\text{Taip}|\text{Saulėta}) = \frac{P(X|Y) \cdot P(Y)}{P(X)} = \frac{0,33 \cdot 0,64}{0,36} = 0,6.$$

Taigi, pagal turimus duomenis (oras saulėtas) pasirenkama klasė – išeiti į lauką.

Duoti duomenys, kuriuose pateikti visi 4 požymiai: oras – lietus, temperatūra – šilta, drėgmė – normali, vėjuota – taip. Šiai priklausomybei apskaičiuoti naudojama (34) formulė ir turi būti skaičiuojamos tikimybės abiem klasėms.

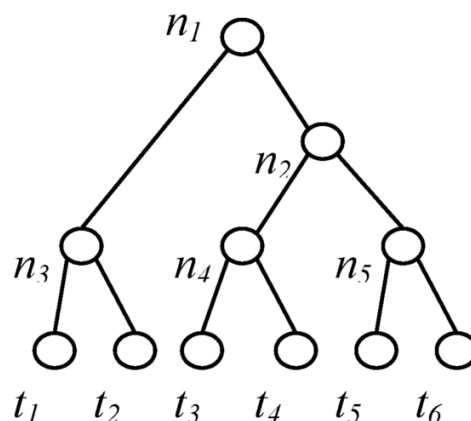
$$P(\text{Taip}|X) = P(\text{Oras} = \text{Lietus}|\text{Taip}) \cdot P(\text{Temperatūra} = \text{Šilta}|\text{Taip}) \cdot P(\text{Drėgmė} = \text{Normali}|\text{Taip}) \cdot P(\text{Vėjuota} = \text{Taip}|\text{Taip}) \cdot P(\text{Taip}) = \frac{2}{9} \cdot \frac{4}{9} \cdot \frac{6}{9} \cdot \frac{3}{9} \cdot \frac{9}{14} = 0,0141;$$

$$P(\text{Ne}|X) = P(\text{Oras} = \text{Lietus}|\text{Ne}) \cdot P(\text{Temperatūra} = \text{Šilta}|\text{Ne}) \cdot P(\text{Drėgmė} = \text{Normali}|\text{Ne}) \cdot P(\text{Vėjuota} = \text{Taip}|\text{Ne}) \cdot P(\text{Ne}) = \frac{3}{5} \cdot \frac{2}{5} \cdot \frac{1}{5} \cdot \frac{3}{5} \cdot \frac{5}{14} = 0,0103.$$

Iš rezultatų matyti, kad klasės eiti į lauką tikimybė yra didesnė, todėl pasirenkama ši klasė. Bajeso klasifikavimo metodas yra ganėtinai paprastas ir nėra iteracinis. Šis metodas pasižymi tuo, kad gali susidoroti su trūkstamais duomenimis, neįtraukiant jų į tikimybės skaičiavimą.

1.2.5 Sprendimų medžio klasifikatorius

Sprendimų medžio (angl. *Decision Tree*) klasifikatorius susideda iš pagrindinio mazgo (angl. *a root node*) n_1 , grupės vidinių mazgų (angl. *internal nodes*) n_i ir lapų mazgų (angl. *leaf nodes*) t_j (žr. 1.6 pav.). Sprendimų medžio klasifikatorius gali išspręsti sudėtingą problemą padalindamas ją į kelias paprastas klasifikavimo problemas ir pritaikydamas progresyvią klasifikavimo metodą (angl. *progressively classified method*), kad išspręstų problemą palaipsniui [47]. Sprendimų medis gali būti naudojamas tiek klasifikavimui, tiek regresijai. Klasifikavimo metu kiekvienas medžio lapas yra pažymėtas klasės žymekliu, keli lapai gali turėti tą patį klasės žymeklį [48].



1.10 pav. Sprendimų medžio modelis [47]

Vienas iš populiariausių sprendimų medžio algoritmų yra *C4.5* algoritmas, kurio veikimas prasideda nuo viršaus į apačią (žr. 1.6 pav.). *C4.5* sprendimų medžio algoritmas pasižymi didele greitaveika ir yra vienas iš greičiausių klasifikavimo metodų [49]. Algoritmo veikimo metu į pagrindinį mazgą yra nukreipiami treniravimo duomenys, pagal kuriuos sudaromi koeficientai ir jie priskiriami vidiniams mazgams, kad kiekvienas kelias nuo pagrindinio mazgo iki klasės lapo atitiktų klasifikavimo taisyklę arba vektorių. Sukurtas sprendimo medžio modelis būna per daug prisitaikęs prie treniravimo duomenų savybių ir dėl šios priežasties mažėja teisingo sprendimo priėmimo tikimybė naujiems duomenims. Dėl šios priežasties prieš naudojant sprendimų medį atliekamas jo genėjimas (angl. *pruning*), kuris skirstomas į „genėjimą po“ (angl. *post-pruning*) ir „genėjimą prieš“ (angl. *prie-pruning*). Genėjimo metu, kai kurie vidiniai mazgai yra pakeičiami lapais, jei šis sprendimas pagerina sprendimo medžio tikslumą.

C4.5 sprendimo medžio algoritmas aprašomas [50] aprioriniu klasės tikimybės įverčiu p_c , klasifikavimo taisyklės aibės informacija (anlg. *Set information*) $Info(D)$, testo informacija (angl. *test information*) $Info_{test}(D, T)$, informacijos išlošiu (angl. *information gain*) $Gain(D, T)$, dalijimo informacija (angl. *split information*) $Info_{split}(D, T)$ bei santykinu informacijos išlošiu (angl.

information gain ratio) $GainRatio(D, T)$. Šie išvardinti parametrai apskaičiuojami naudojant formules (35) – (40):

$$p_c = \frac{|D_{class(x)=c}|}{|D|}, \quad (35)$$

$$Info(D) = - \sum_{c \in classes} p_c \log_2 p_c, \quad (36)$$

$$Info_{test}(D, T) = \sum_{i=1}^m \frac{|D_i|}{|D|} Info(D), \quad (37)$$

$$Gain(D, T) = Info(D) - Info_{test}(D, T), \quad (38)$$

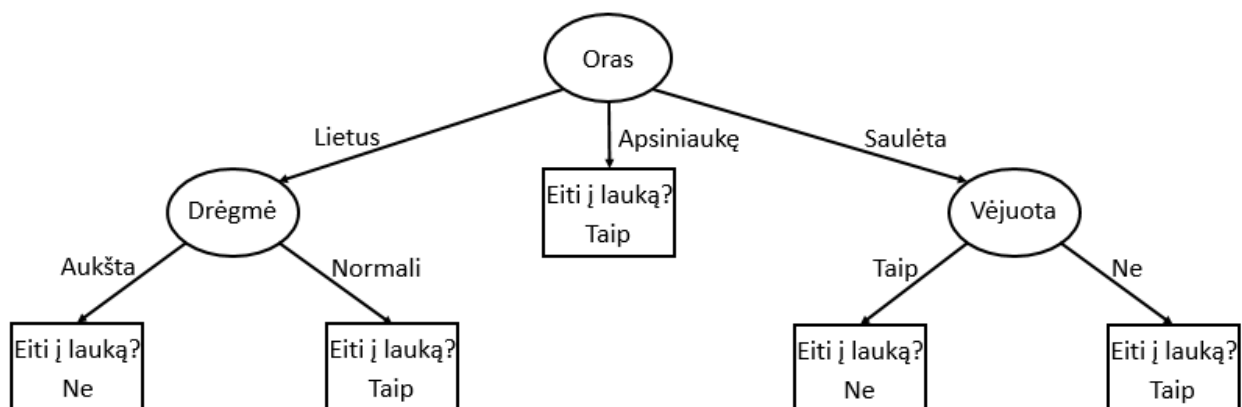
$$Info_{split}(D, T) = - \sum_{i=1}^m \frac{|D_i|}{|D|} \log_2 \left(\frac{|D_i|}{|D|} \right), \quad (39)$$

$$GainRatio(D, T) = \frac{Gain(D, T)}{Info_{split}(D, T)}, \quad (40)$$

Čia: $D_{class(x)=c}$ yra klasifikavimo taisyklės priklausančios klasei c , $c \in classes$, T – testas, m – testo baigčių skaičius.

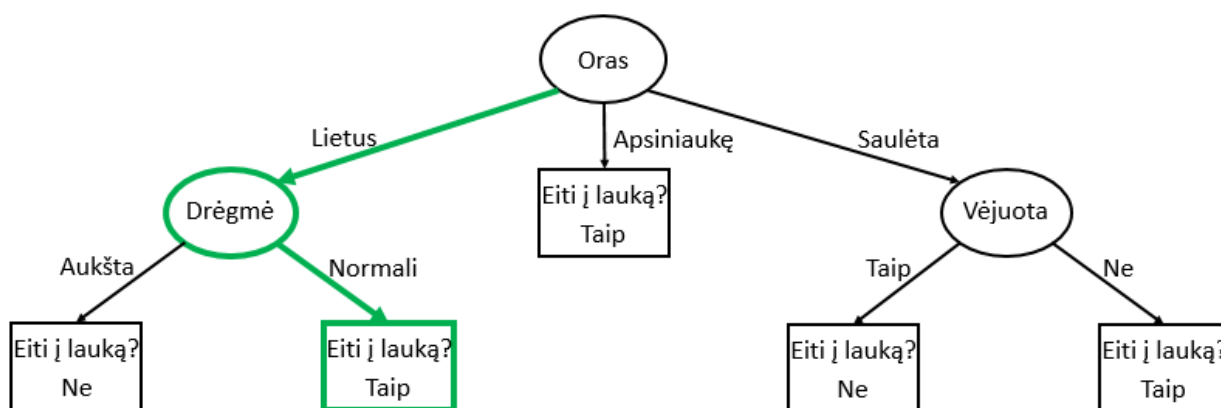
Klasifikavimo taisyklių aibės informacija $Info(D)$ nurodo informacijos kiekį, kuris reikalingas nustatyti klasifikavimo taisyklės klasę. Testo informacija $Info_{test}(D, T)$ nusako informacijos kiekį, kurio reikia klasei nustatyti po sprendimo vidinio mazgo testo atlikimo. Informacijos išlošis $Gain(D, T)$ nurodo informacijos sumažėjimą, kai pritaikomas klasifikavimo taisyklei testas T . Dalijimo informacija $Info_{split}(D, T)$ nusako informacijos kiekį, kurio reikia norint nustatyti testo baigtį. Santykinis informacijos išlošis $GainRatio(D, T)$ nurodo klasifikavimo informacijos dalį, kuri yra naudinga panaudojus pasirinktą klasifikavimo taisyklių aibę.

Norint paaiškinti sprendimų medžio klasifikatoriaus veikimą galima panaudoti 1.2 lentelės duomenis. Pagal šiuos duomenis naudojant Hunto algoritmą [51] sudarytas sprendimų medis (žr. 1.11 pav.). Sprendimų medis sprendžia problemą: „Ar eiti į lauką?“.



1.11 pav. Sprendimų medžio pavyzdys

Paimta ta pati situacija, kaip ir Naive Bajeso pavyzdyje ir pateikti visi 4 požymiai: oras – lietus, temperatūra – šilta, drėgmė – normali, vėjuota – taip. Nustatyti klasę užteko dviejų parametru: oro ir drėgmės. Nustatyta klasė yra „Taip“ (žr. 1.12 pav.)



1.12 pav. Sprendimų medžio pavyzdžio sprendimas

1.2.6 Adaboost algoritmas

Adaboost (angl. *Adaptive Boosting*) algoritmas sukurtas Yoavo Fruendo ir Roberto Schapiro [52]. Šis algoritmas naudojamas sukombinuoti silpnus klasifikatorius į stiprius ir jis gali būti taikomas su daugeliu klasifikavimo algoritmu. *Adaboost* algoritmas gali būti apibūdintas kaip pasirinkimo procesas, kuris pasirenka mažą rinkinį klasifikatorių su mažiausiomis paklaidomis ir jų svoriniais koeficientais [30]. Galutinis klasifikatorius yra laikomas stipriu, nes jis sudarytas iš kombinacijos silpnų klasifikatorių. Nors kiekvienas silpnas klasifikatorius negali suteikti gero klasifikavimo mokymo pavyzdžiams, tačiau tinkama svorinių koeficientų kombinacija su kitais klasifikatorių koeficientais gali pagerinti paskutinio klasifikavimo atlikimą.

Įėjimo apmokymo pavyzdžiai $(x_1, y_1, \dots, (x_n, y_n))$, kur $y_i = 1, 0$ atitinka gerą ir blogą pavyzdžius. Pradinis pavyzdžio svorio koeficientas $w_i = (1/2p)$ ir $(1/2q)$, kur p ir q yra skaičiai teigiamo ir neigiamo pavyzdžių.

Adaboost algoritmas:

Kartoti nuo $t = 1 \dots T$, kur T klasių skaičius:

- a) Normalizuoti svoriniai koeficientai $w_{t,i}$
- b) Apskaičiuota klasifikavimo paklaida kiekvienai savybei f :

$$C = \sum w_i \cdot |h(f, x_i) - y_i|. \quad (41)$$

- c) Pasirinkti geriausią silpną klasifikatorių $h_t(x)$ su mažiausia paklaida C_t :

$$H_t(x, f, P, \theta) = \begin{cases} 1 & \text{jei } P_t f_t(x) < P_t \theta_t \\ 0 & \text{jei } P_t = \text{poliariskumas } \{+1, -1\} \end{cases} \quad (42)$$

Atnaujinti svorinius koeficientus:

$$w_i = w_i \left(\frac{c_t}{1-c_t} \right) - |h(x_i) - y_i|. \quad (43)$$

Išėjimas – kombinuotas klasifikatorius:

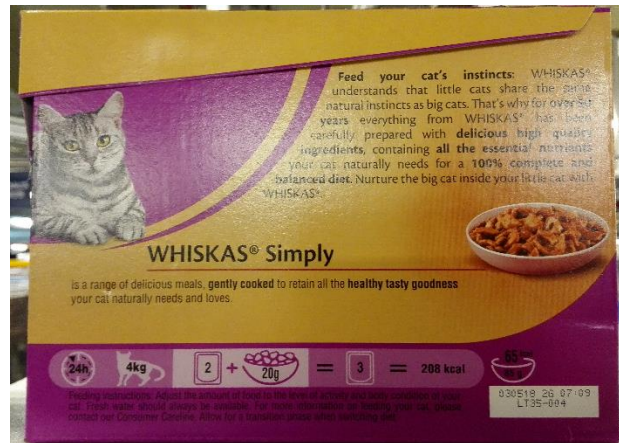
$$0,5(1 + \text{sign}[\sum \alpha_t h_t(x) - 0,5 \sum \alpha_t]), \quad (44)$$

kur $\alpha_t = \log \frac{1-c_t}{c_t}$.

Jei įėjimo vektorius apibrėžia *Adboost* klasifikatorių neigiamai, tai klasifikatorius bus ištrintas iš apmokymo pavyzdžių rinkinio ir nebepateks į sekantį sluoksnį, todėl pavyzdžių skaičius mažės didėjant sluoksnių skaičiui.

2 TYRIMO DALIS

Tyrimo dalyje aprašytas pramoninio čiurkšlinio rašalinio spausdinto *Videojet* užpurkštos informacijos ant pakuotės (žr. 2.1 pav.) identifikavimo tyrimas. Informacija spausdintuvu yra užpurškiama dešiniajame apatiniame pakuotės kampe ant tam skirto balto laukelio.



2.1 pav. Produkto pakuotės nuotrauka

Užpurškiama informacija būna dviejų formatų, priklausomai nuo gaminamo produkto formato. Pirmasis formatas sudarytas iš 22 simbolių, o antrasis iš 31 simbolio. Pirmajame informacijos formate užšifruota: galiojimo data, užpurškimo laikas, veterinarinis kodas bei gamybinės linijos numeris (žr. 2.2 pav.).



2.2 pav. Pirmojo formato informacijos šifravimas

Antrajame informacijos formate užšifruota: galiojimo data, veterinarinis kodas, gaminimo metų paskutinis skaitmuo, gaminimo metų savaitė, savaitės diena (A – G), pamainos numeris, linijos numeris bei užpurškimo laikas (žr. 2.3 pav.).



2.3 pav. Antrojo formato informacijos šifravimas

Informacijai aprašyti iš viso naudojami 22 simboliai, kurie pateikti 2.1 lentelėje. Simboliai pateikti kartu su ASCII (angl. *American Standard Code for Information Interchange*) reikšmėmis.

2.1 lentelė. Informacijai aprašyti naudojami simboliai

Nr.	Simolis	ASCII
1	–	45
2	0	48
3	1	49
4	2	50
5	3	51
6	4	52
7	5	53
8	6	54
9	7	55
10	8	56
11	9	57
12	:	58
13	A	65
14	B	66
15	C	67
16	D	68
17	E	69
18	F	70
19	G	71
20	L	76
21	R	82
22	T	84

Maksimalus produkto gamybos greitis per minutę yra 120 dėžučių. Todėl sukurtas algoritmas turi identifikuoti nuo pakuotės informaciją greičiau nei per 500 ms. Pavyzdžių nuotraukoms fotografuoti buvo panaudota mobilusis telefonas *Samsung S7*, kurio kameros specifikacijos pateiktos 2.2 lentelėje.

2.2 lentelė. Kameros specifikacijos

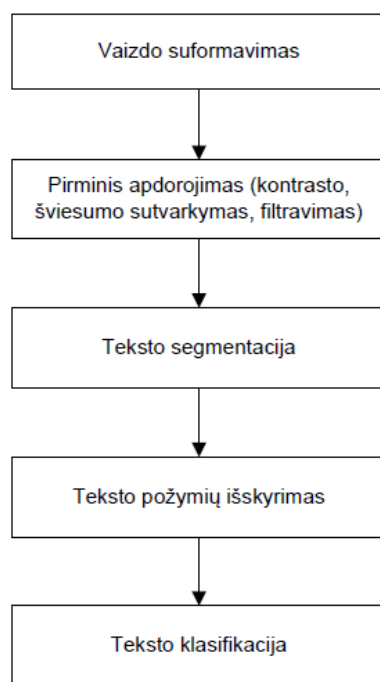
Parametras	Dydis
Megapikseliai	12
Fokusavimo nuotolis, mm	28
Diafragmos dydis	F1.7
Kameros sensoriaus dydis	1/2.5"

Tyrimo metu vaizdų apdorojimui buvo naudotas *DELL Latitude E6420* nešiojamas kompiuteris, kurio pagrindinės specifikacijos pateiktos 2.3 lentelėje. Tyrimas atliktas naudojant *MATLAB* programinę įrangą. Tyrime naudotas programos kodas pateiktas darbo prieduose „*MATLAB* programos kodas“.

2.3 lentelė. Kompiuterio specifikacijos

Parametras	Aprašymas
Procesorius	Intel® Core™ i7–2720QM
Procesoriaus dažnis, MHz	2200
Branduolių skaičius	4
Operatyvioji atmintis RAM, GB	8
Operatyviosios atminties tipas	DDR3
Operatyviosios atminties dažnis, MHz	665
Vaizdo plokštė	NVIDIA NVS 4200M
Vaizdo plokštės dažnis, MHz	740
Vaizdo plokštės atmintis, MB	512
Operacinė sistema	Windows 10 Pro 64-bit

Vaizdo apdorojimo žingsniai tekstui atpažinti, nuo vaizdo suformavimo iki teksto klasifikavimo, pateikti 2.4 paveiksle. Vaizdo suformavimo etape panaudojus kamera gaunama nuotrauka arba pirminis vaizdas, kurį reikia paruošti vaizdo apdorojimui. Antrame etape atliekamas vaizdo sutvarkymas: spalvotas vaizdas paverčiamas į pilkos spalvos vaizdą, o tada atliekamas binarizavimas (angl. *thresholding*), kurio metu pilkos spalvos vaizdas paverčiamas į juodos ir baltos spalvos (binarinį) vaizdą pritaikius slenkstinę vertę. Juodos ir baltos spalvos vaizde esantis triukšmas nufiltruojamas medianos filtru.



2.4 pav. Vaizdo apdorojimo žingsniai tekstui atpažinti

Trečiame etape atliekama sutvarkyto vaizdo teksto segmentacija, kurios metu surandami teksto simboliai. Surasto simbolio vaizdas konvertuojamas į apibrėžto dydžio vaizdą, kad suvienodinti visų simbolių dydžius ir aprašyti požymius vienodo dydžio informacija.

Ketvirtame etape deskriptorius iš konvertuoto simbolio vaizdo išskiria požymius ir juos aprašo vektoriumi. Penktame etape simbolio vektorius klasifikuojamas ir jis pagal požymius priskiriamas vienai iš simbolių klasių, kuri nusako simbolio reikšmę. Norint atlikti klasifikavimą, visų pirma, reikia paruošti klasifikatoriaus duomenų bazę. Tai atliekama identiškai, kaip ir informacijos identifikavimo metu (žr. 2.4 pav.), išskyrus teksto klasifikavimo etapu metu yra prie duomenų yra priskiriama simbolio reikšmė (klasė). Šio tyrimo mokymų bazė buvo suformuota iš 1100 pavyzdžių arba 50 pavyzdžių kiekvienai klasei.

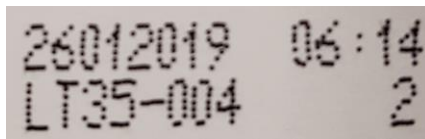
2.1 Pirmoji tyrimo dalis

Pirmoje tyrimo dalyje atliktas informacijos identifikavimas, panaudojus 17 skirtingų klasifikatorių ir deskriptorių kombinacijas, keičiant klasifikatorių parametrus, kai simbolio dydis 30×20 pikselių. Tyrimo metu naudoti k -artimiausių kaimynų KNN, atraminių vektorių mašinų SVM, dirbtinių neuroninių tinklų ANN, Naive Bajeso, sprendimų medžio klasifikatoriai ir HOG, pikselių vektorius, LBP, PCA deskriptoriai. Keičiami klasifikatorių parametrai ir jų vertės pateiktos 2.4 lentelėje.

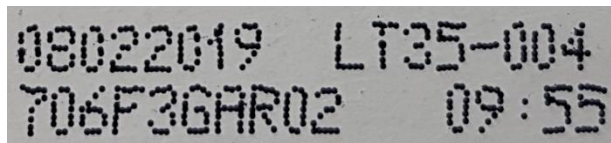
2.4 lentelė. Keičiami klasifikatorių parametrai

Klasifikatorius	Keičiamas parametras	Keitimo ribos
<i>K</i> -artimiausi kaimynai	kaimynų skaičius	1 – 50
Atraminė vektorių mašina	polinomo laipsnis	1 – 4
Dirbtinis neuronų tinklas	paslėptų sluoksnių skaičius	1 – 50
Naive Bajesas	–	–
Sprendimų medis	pastabų skaičius per šaką	1 – 50

Pirmoje tyrimo dalyje identifikuota dviejų skirtingų formatų nuotraukų informacija (žr. 2.5 pav. ir 2.6 pav.).



2.5 pav. Tyrimo nuotrauka Nr. 1

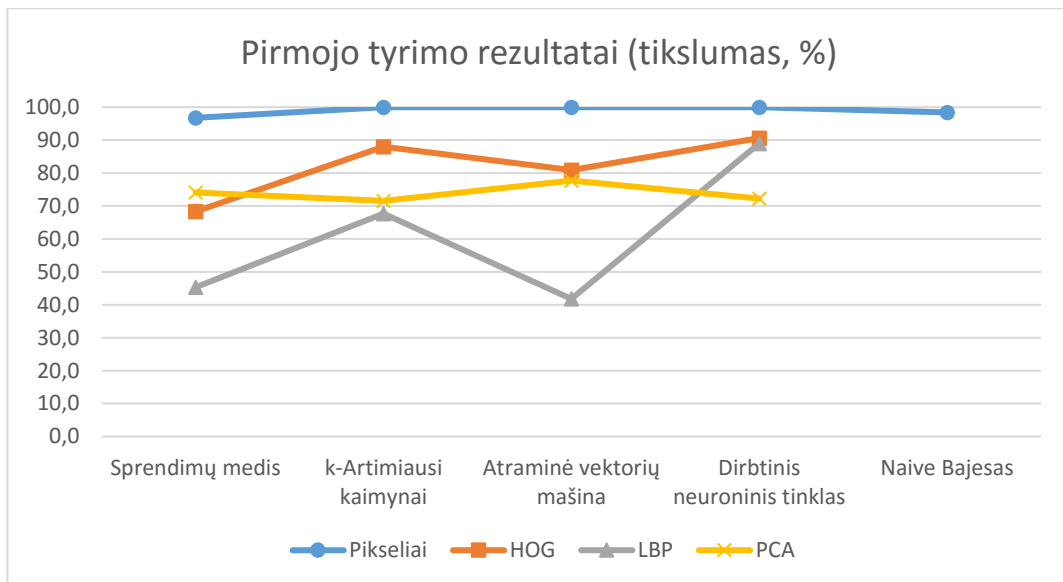


2.6 pav. Tyrimo nuotrauka Nr. 2

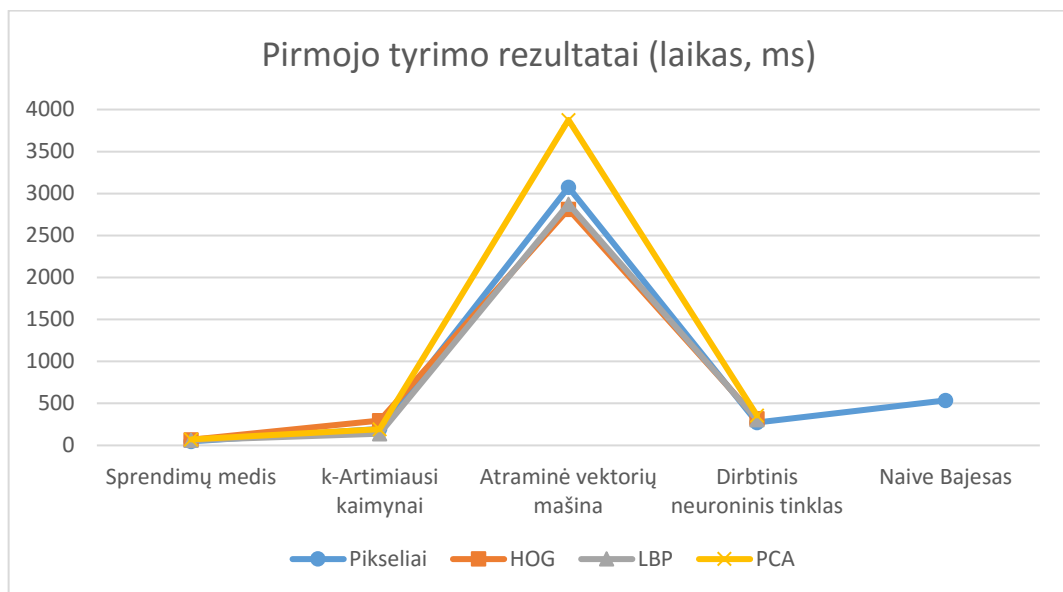
Visi pirmojo tyrimo rezultatai pateikti prieduose (žr. 1 – 10 lentelėse). Iš tyrimo duomenų išrinkti geriausi rezultatai, pagal skirtingus klasifikatorius bei deskriptorius, ir pateikti 2.5 lentelėje bei 2.7 ir 2.8 paveiksluose. O mažėjimo tvarka surūšiuoti rezultatai pateikti 2.6 lentelėje.

2.5 lentelė. Pirmojo tyrimo rezultatai

Klasifikatorius / Deskriptorius	Pikseliai		HOG		LBP		PCA	
	Tikslumas, %	Laikas, ms	Tikslumas, %	Laikas, ms	Tikslumas, %	Laikas, ms	Tikslumas, %	Laikas, ms
Sprendimų medis	96,8	48	68,3	68	45,3	63	74,1	72
<i>k</i> -Artimiausi kaimynai	100,0	195	88,1	297	67,7	138	71,6	195
Atraminė vektorių mašina	100,0	3075	80,9	2813	41,8	2875	77,7	3877
Dirbtinis neuroninis tinklas	100,0	273	90,6	320	89,0	309	72,2	355
Naive Bajesas	98,4	537	-	-	-	-	-	-



2.7 pav. Pirmojo tyrimo rezultatai (tikslumas, %)



2.8 pav. Pirmojo tyrimo rezultatai (laikas, ms)

2.6 lentelė. Pirmojo tyrimo rezultatai pagal tikslumą mažėjimo tvarka

Nr.	Klasifikatorius	Deskriptorius	Parametras	Tikslumas, %	Laikas, ms
1	<i>k</i> -Artimiausi kaimynai	Pikseliai	41	100,0	195
2	Dirbtiniai neuroniniai tinklai	Pikseliai	12	100,0	273
3	Atraminė vektorių mašina	Pikseliai	4	100,0	3075
4	Naive Bajesas	Pikseliai	-	98,4	537
5	Sprendimų medis	Pikseliai	43	96,8	48
6	Dirbtiniai neuroniniai tinklai	HOG	30	90,6	320
7	Dirbtiniai neuroniniai tinklai	LBP	10	89,0	309
8	<i>k</i> -Artimiausi kaimynai	HOG	47	88,1	297
9	Atraminė vektorių mašina	HOG	4	80,9	2813
10	Atraminė vektorių mašina	PCA	3	77,7	3877

Iš tyrimo rezultatų (žr. 2.6 lentelę) matyti, kad tiksliausi rezultatai gauti naudojant pikselių vektorių deskriptorių (vidutinis tikslumas – 99 %). Geriausias visų deskriptorių tikslumas (žr. 2.7 pav.) gautas panaudojus dirbtinių neuroninių tinklų klasifikatorių (deskriptorių vidutinis tikslumas – 88 %). Sparčiausias klasifikatorius – sprendimų medis (deskriptorių vidutinis laikas – 63 ms), kuris vaizdus apdorojo vidutiniškai 4 – 8 kartus greičiau nei kiti klasifikatoriai. O lėčiausias – atraminės vektorių mašinos klasifikatorius (vidutinis laikas – 3255 ms).

2.2 Antroji tyrimo dalis

Antrojoje tyrimo dalyje atliktas informacijos identifikavimas, panaudojus geriausias pirmojo tyrimo metu gautas klasifikatorių ir deskriptorių kombinacijas, kai keičiamas simbolio metmenų dydis (60×40 ; 30×20 ; 24×16 ; 15×10). Tyrimo metu naudojamos nuotraukos, kaip ir pirmajame tyrime (žr. 2.5 pav. ir 2.6 pav.).

Antrojo tyrimo rezultatai pateikti 2.7 lentelėje, o rezultatai, surūšiuoti pagal tikslumą mažėjimo tvarka, pateikti 2.8 lentelėje.

2.7 lentelė. Antrojo tyrimo rezultatai (tirtos 2 nuotraukos)

Simbolio dydis				60 x 40		30 x 20		24 x 16		15 x 10	
Nr.	Klasifikatorius	Deskriptorius	Parametras	Tikslumas, %	Laikas, ms	Tikslumas, %	Laikas, ms	Tikslumas, %	Laikas, ms	Tikslumas, %	Laikas, ms
1	<i>k</i> -Artimiausi kaimynai	Pikseliai	41	95,5	656	100,0	195	97,7	147	98,4	186
2	Dirbtiniai neuroniniai tinklai	Pikseliai	12	98,4	329	100,0	273	96,1	286	100	281
3	Atraminė vektorių mašina	Pikseliai	4	100	3648	100,0	3075	3,9	3075	90,6	2796
4	Naive Bajesas	Pikseliai	-	96,1	1398	98,4	537	97,7	414	100	291
5	Sprendimų medis	Pikseliai	43	91,3	50	96,8	48	93,8	47	95,4	49
6	Dirbtiniai neuroniniai tinklai	HOG	30	100	334	90,6	320	86,1	312	-	-
7	Dirbtiniai neuroniniai tinklai	LBP	10	59,2	311	89,0	309	81,9	302	82,6	302
8	<i>k</i> -Artimiausi kaimynai	HOG	47	97,7	457	88,1	297	85,1	292	-	-
9	Atraminė vektorių mašina	HOG	4	100	3458	80,9	2813	88,3	2806	-	-
10	Atraminė vektorių mašina	PCA	3	68	6167	77,7	3877	57,9	3595	62,8	3402

2.8 lentelė. Antrojo tyrimo rezultatai pagal tikslumą mažėjimo tvarka (tirtos 2 nuotraukos)

Nr.	Klasifikatorius	Deskriptorius	Parametras	Simbolio dydis	Tikslumas, %	Laikas, ms
1	<i>k</i> -Artimiausi kaimynai	Pikseliai	41	30 x 20	100,0	195
2	Dirbtiniai neuroniniai tinklai	Pikseliai	12	30 x 20	100,0	273
3	Dirbtiniai neuroniniai tinklai	Pikseliai	12	15 x 10	100,0	281
4	Naive Bajesas	Pikseliai	-	15 x 10	100,0	291
5	Dirbtiniai neuroniniai tinklai	HOG	30	60 x 40	100,0	334
6	Atraminė vektorių mašina	Pikseliai	4	30 x 20	100,0	3075
7	Atraminė vektorių mašina	HOG	4	60 x 40	100,0	3458
8	Atraminė vektorių mašina	Pikseliai	4	60 x 40	100,0	3648
9	<i>k</i> -Artimiausi kaimynai	Pikseliai	41	15 x 10	98,4	186
10	Dirbtiniai neuroniniai tinklai	Pikseliai	12	60 x 40	98,4	329

Iš rezultatų (žr. 2.7 lentelę) matyti, jog didžiausias vidutinis tikslumas (92,1 %) pasiektas naudojant 30×20 dydžio simbolius. Rezultatai parodo, jog keičiantis simbolio dydžiui, keičiasi ir apdorojimo laikas. Didžiausias apdorojimo laiko pakitimas (lyginant 60×40 ir 15×10

simbolių apdorojimo laikus) užfiksuotas naudojant Naive Bajeso klasifikatorių (4,8 karto). O mažiausiai apdorojimo laikas kito naudojant sprendimo medžio (1,03 karto) ir dirbtinių neuroninių tinklų (1,17 karto) klasifikavimo metodus.

Kadangi naudojant dirbtinių neuroninių tinklų klasifikatorių algoritmo apdorojimo laikas nežymiai pakito, tai buvo pasirinkta atlikti kitą tyrimą naudojant šį klasifikatorių bei keičiant deskriptoriaus metodą ir simbolio dydį. Tyrimo rezultatai pateikti 2.9 lentelėje.

2.9 lentelė. Antrojo tyrimo rezultatai, naudojant dirbtinių neuroninių tinklų deskriptorių

Simbolio dydis				60 x 40		30 x 20		24 x 16		15 x 10	
Nr.	Klasifikatorius	Deskriptorius	Parametras	Tikslumas, %	Laikas, ms	Tikslumas, %	Laikas, ms	Tikslumas, %	Laikas, ms	Tikslumas, %	Laikas, ms
1	Dirbtiniai neuroniniai tinklai	Pikseliai	12	98,4	329	100,0	273	96,1	286	100	281
2	Dirbtiniai neuroniniai tinklai	HOG	12	100	398	76,9	328	87,6	337	-	-
3	Dirbtiniai neuroniniai tinklai	LBP	12	66	326	81,9	338	84,3	338	77,1	332
4	Dirbtiniai neuroniniai tinklai	PCA	12	52,1	385	37,4	351	56,9	336	52,6	327

Iš rezultatų (žr. 2.9 lentelę) matyti, jog identifikavimo tikslumas tiesiogiai nepriklauso nuo simbolio dydžio. Mažinant simbolio matricos dydį greitėja algoritmo apdorojimo laikas, kai naudojami pikselių vektorius, PCA ir HOG deskriptoriai. Tačiau naudojant LBP deskriptorių ir keičiant simbolio dydį, apdorojimo laikas nekinta.

2.3 Trečioji tyrimo dalis

Trečiojoje tyrimo dalyje atliktas informacijos identifikavimas, panaudojus geriausias antrojo tyrimo metu gautas klasifikatoriaus, deskriptoriaus ir simbolio dydžio kombinacijas bei keičiant klasifikatoriaus parametrus. Tyrimo metu buvo identifikuojama penkiasdešimt nuotraukų su skirtinga, ant pakuotės užpurkšta, informacija.

Trečiojo tyrimo rezultatai pateikti prieduose 11 – 20 lentelėse, o rezultatai, surūšiuoti pagal tikslumą mažėjimo tvarka, pateikti 2.10 lentelėje.

2.10 lentelė. Trečiojo tyrimo rezultatai (tirtos 50 nuotraukų)

Nr.	Klasifikatorius	Deskriptorius	Parametras	Simbolio dydis	Tikslumas, %	Laikas, ms
1	Dirbtiniai neuroniniai tinklai	Pikseliai	32	15 x 10	99,6	266
2	Dirbtiniai neuroniniai tinklai	Pikseliai	25	60 x 40	99,1	308
3	Atraminė vektorių mašina	Pikseliai	2	60 x 40	99,0	4288
4	Dirbtiniai neuroniniai tinklai	Pikseliai	46	30 x 20	98,9	313
5	k -Artimiausi kaimynai	Pikseliai	2	15 x 10	98,6	97
6	Atraminė vektorių mašina	HOG	2	60 x 40	98,5	3169
7	Dirbtiniai neuroniniai tinklai	HOG	42	60 x 40	98,4	361
8	Naive Bajesas	Pikseliai	–	15 x 10	97,7	209
9	k -Artimiausi kaimynai	Pikseliai	19	30 x 20	97,0	193
10	Atraminė vektorių mašina	Pikseliai	1	30 x 20	96,8	3278

Iš trečiojo tyrimo rezultatų (žr. 2.10 lentelė) matyti, kad geriausias vidutinis tikslumas (99,6 %) pasiektas naudojant dirbtinių neuroninių tinklų klasifikatorių (paslėptų sluoksnių skaičius – 32), pikselių vektorių deskriptorių bei pasirinkus simbolio dydį – 15×10 . Informacijos identifikavimo metu, klasifikavimo metodai dažniausiai sumaišydavo „0“ su „D“, „T“ su „7“ ir „5“ su „6“.

2.4 Ketvirtoji tyrimo dalis

Ketvirtojoje tyrimo dalyje atliktas informacijos identifikavimas, panaudojus trečiojo tyrimo klasifikatoriaus, deskriptoriaus ir simbolio dydžio kombinacijas bei padidinus mokymo duomenų bazės dydį. Mokymo duomenų bazės dydis buvo padidintas nuo 1100 simbolių iki 2500 simbolių (žr. 2.11 lentelę). Tyrimo metu identifikuojamos tos pačios penkiasdešimt nuotraukų, kaip ir trečiojo tyrimo metu.

2.11 lentelė. Mokymo duomenų bazės simboliai ir jų kiekis

Simbolis	ASCII	Kiekis
–	45	74
0	48	388
1	49	177
2	50	160
3	51	152
4	52	138
5	53	125
6	54	66
7	55	97
8	56	105
9	57	127
:	58	64
A	65	83
B	66	82
C	67	103
D	68	77
E	69	81
F	70	83
G	71	80
L	76	88
R	82	83
T	84	67

Ketvirtojo tyrimo rezultatai pateikti prieduose 21 – 30 lentelėse, o rezultatai, surūšiuoti pagal tikslumą bei spartą mažėjimo tvarka, pateikti 2.12 lentelėje.

2.12 lentelė. Ketvirtojo tyrimo rezultatai (tirtos 50 nuotraukų)

Nr.	Klasifikatorius	Deskriptorius	Parametras	Simbolio dydis	Tikslumas, %	Laikas, ms
1	Dirbtiniai neuroniniai tinklai	Pikseliai	28	15 x 10	99,8	257
2	Dirbtiniai neuroniniai tinklai	Pikseliai	46	30 x 20	99,8	336
3	Dirbtiniai neuroniniai tinklai	HOG	36	60 x 40	99,8	355
4	Dirbtiniai neuroniniai tinklai	Pikseliai	42	60 x 40	99,8	362
5	Atraminė vektorių mašina	Pikseliai	1	30 x 20	99,7	3323
6	Atraminė vektorių mašina	Pikseliai	1	60 x 40	99,7	4398
7	k–Artimiausi kaimynai	Pikseliai	3	15 x 10	99,6	142
8	k–Artimiausi kaimynai	Pikseliai	8	30 x 20	99,6	343
9	Atraminė vektorių mašina	HOG	3	60 x 40	99,6	3395
10	Naive Baijesas	Pikseliai	–	15 x 10	99,3	235

Iš ketvirtojo tyrimo rezultatų (žr. 2.12 lentelę) matyti, jog padidinus mokymo duomenų bazės dydį nuo 1100 simbolių iki 2500 simbolių, tikslumas padidėjo nuo 0,2 % iki 2,9 %, lyginant su trečiojo tyrimo rezultatais. O algoritmo apdorojimo laikas vidutiniškai padidėjo 13 %. Didžiausią tikslumą (99,8 %) pavyko pasiekti naudojant dirbtinių neuroninių tinklų klasifikatorių.

IŠVADOS IR REZULTATAI

1. Baigiamojo magistro darbo projekte atlikta vaizdų apdorojimo literatūros analizė informacijos identifikavimo tema. Apžvelgta k -artimiausių kaimynų KNN, atraminių vektorių mašinų SVM, dirbtinių neuroninių tinklų ANN, Naive Bajeso ir sprendimų medžio klasifikatorių metodika bei HOG, LBP, PCA, SIFT, SURF, Haar-like požymių deskriptorių metodika.
2. Sudarytas informacijos identifikavimo algoritmas naudojant *MATLAB* programine įranga. Algoritmas geba identifikuoti nuotraukos informaciją, naudojant k -artimiausių kaimynų KNN, atraminių vektorių mašinų SVM, dirbtinių neuroninių tinklų ANN, Naive Bajeso ir sprendimo medžio klasifikatorius bei HOG, LBP, PCA ir pikselių vektorių deskriptorius.
3. Iš pirmojo tyrimo rezultatų (žr. 2.6 lentelę) matyti, kad tiksliausi rezultatai buvo gauti naudojant pikselių vektorių deskriptorių (vidutinis tikslumas – 99 %). Trumpiausias algoritmo apdorojimo laikas gautas naudojant sprendimų medžio klasifikatorių (vidutinis laikas – 63 ms), kuris vaizdus apdorojo vidutiniškai 4 – 8 kartus greičiau nei kiti klasifikatoriai. O ilgiausias algoritmo apdorojimo laikas gautas naudojant atraminės vektorių mašinos klasifikatorių (vidutinis laikas – 3255 ms).
4. Iš antrojo tyrimo rezultatų (žr. 2.7 lentelę) matyti, jog keičiant simbolio dydį, keičiasi ir apdorojimo laikas. Didžiausias apdorojimo laiko pakitimas (lyginant 60×40 ir 15×10 simbolių apdorojimo laikus) užfiksuotas naudojant Naive Bajeso klasifikatorių (4,8 karto). Tyrimas parodė, jog identifikavimo tikslumas tiesiogiai nepriklauso nuo simbolio dydžio (žr. 2.9 lentelė).
5. Iš trečiojo tyrimo rezultatų (žr. 2.9 lentelę) matyti, jog geriausius rezultatus (50 nuotraukų informacijos identifikavimo vidutinis tikslumas – 99,6 %) pavyko pasiekti naudojant dirbtinių neuroninių tinklų klasifikatorių (paslėptų sluoksnių skaičius – 32), pikselių vektorių deskriptorių bei pasirinkus simbolio dydį 15×10 . Informacijos identifikavimo metu, klasifikavimo metodas dažniausiai sumaišydavo „0“ su „D“, „T“ su „7“ ir „5“ su „6“.
6. Iš ketvirtojo tyrimo rezultatų (žr. 2.12 lentelę) matyti, jog padidinus mokymo duomenų bazės dydį nuo 1100 simbolių iki 2500 simbolių, tikslumas padidėjo nuo 0,2 % iki 2,9 %, lyginant su trečiojo tyrimo rezultatais. Didžiausią tikslumą (99,8 %) pavyko pasiekti naudojant dirbtinių neuroninių tinklų klasifikatorių.

LITERATŪROS SĄRAŠAS

1. Holz D., Behnke S. Fast Edge-Based Detection and Localization of Transport Boxes and Pallets in RGB-D Images for Mobile Robot Bin Picking. *Proceedings of ISR 2016: 47st International Symposium on Robotics* [interaktyvus]. Munich: VDE, 2016 [žiūrėta 2017-04-30]. ISBN 978-3-8007-4231-8. Prieiga per: IEEE Xplore.
2. Hocenski Ž., Matić T., Vidović, I. Technology transfer of computer vision defect detection to ceramic tiles industry. *International Conference on Smart Systems and Technologies (SST)* [interaktyvus]. Osijek: IEEE, 2016 [žiūrėta 2017-04-30]. ISBN 978-1-5090-3720-9. Prieiga per: IEEE Xplore.
3. Nava-Dueñas C. F., Gonzalez-Navarro F. F. OCR for Unreadable Damaged Characters on PCBs Using Principal Component Analysis and Bayesian Discriminant Functions. *2015 International Conference on Computational Science and Computational Intelligence (CSCI)* [interaktyvus]. Las Vegas: IEEE, 2015 [žiūrėta 2017-04-30]. ISBN 978-1-4673-9795-7. Prieiga per: IEEE Xplore.
4. Singh G., Sachan M., Multi-Layer Perceptron (MLP) Neural Network Technique for Offline Handwritten Gurmukhi Character Recognition. *2014 IEEE International Conference on Computational Intelligence and Computing Research (ICIC)* [interaktyvus]. Coimbatore: IEEE, 2014 [žiūrėta 2016-06-04]. ISBN 978-1-4799-3975-6. Prieiga per: IEEE Xplore.
5. Deshpande S., Shriram R. Real time text detection and recognition on hand held objects to assist blind people. *International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)* [interaktyvus]. Pune: IEEE, 2016 [žiūrėta 2017-05-01]. ISBN 978-1-5090-2080-5. Prieiga per: IEEE Xplore.
6. Yuanxing Z., Jing G., Chui L., Shumin H., Yong G., Qingmao H. Licence Plate Location Based on Haar-like Cascade Classifiers and Edges. *2010 Second WRI Global Congress on Intelligent Systems (GCIS)* [interaktyvus]. Wuhan: IEEE, 2010 [žiūrėta 2016-06-03]. ISSN 2155-6091. Prieiga per: IEEE Xplore.
7. Huang. T. S. Computer Vision: Evolution and Promise [interaktyvus]. Urbana: CERN 1996 [žiūrėta 2017-04-29]. Prieiga per: CERN Document Server.
8. Wu Y. An Introduction to Computer Vision. *EECS 432-Advanced Computer Vision Notes Series I* [interaktyvus]. Evanston [žiūrėta 2017-04-29]. Prieiga per: <http://users.eecs.northwestern.edu/~yingwu/teaching/EECS432/Notes/intro.pdf>

9. Dervinis, Donatas. *Vaizdų apdorojimas: mokamoji knyga* [interaktyvus]. Vilnius: TEV, 2012 [žiūrėta 2016-06-10]. ISBN 978-609-433-107-7. Prieiga per: <https://www.ebooks.ktu.lt/einfo/451/vaizdu-apdorojimas/>
10. Mehta H., Singla S., Mahajan A. Optical character recognition (OCR) system for Roman script & English language using Artificial Neural Network (ANN) classifier. *International Conference on Research Advances in Integrated Navigation Systems (RAINS)* [interaktyvus]. Bangalore: IEEE, 2016 [žiūrėta 2017-04-30]. ISBN 978-1-5090-1111-7. Prieiga per: IEEE Xplore.
11. Mazlan M. J., Ibrahim Z., Kasiran Z. Text–Frame Classification. *2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)* [interaktyvus]. Langkawi: IEEE, 2016 [žiūrėta 2016-06-18]. ISBN 978-1-4673-7124-7. Prieiga per: IEEE Xplore.
12. Dalal N., Triggs B. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* [interaktyvus]. San Diego: IEEE, 2005 [žiūrėta 2017-04-16]. ISSN 1063-6919. Prieiga per: IEEE Xplore.
13. Goyal A., Khandelwal K., Keshri P. Optical Character Recognition for Handwritten Hindi. *CS229 Machine Learning* [interaktyvus]. Stanford: 2010 [žiūrėta 2017-04-16]. [interaktyvus] [žiūrėta 2017 m. balandžio 16 d.] Prieiga per: <https://pdfs.semanticscholar.org/3bcf/694b910758cbcd61bfb84d5e3776fb005a9b.pdf>
14. Feng K., Yuan F. Static hand gesture recognition based on HOG characters and support vector machines. *2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA)* [interaktyvus]. Toronto: IEEE, 2013 [žiūrėta 2017-04-15]. ISBN 978-1-4799-2716-6. Prieiga per: IEEE Xplore.
15. Tikader A., Puhan N. B. Histogram of oriented gradients for English–Bengali script recognition. *2014 International Conference for Convergence of Technology (I2CT)* [interaktyvus]. Pune: IEEE, 2014 [žiūrėta 2017-04-16]. ISBN 978-1-4799-3759-2. Prieiga per: IEEE Xplore.
16. Ojala T., Pietikainen M., Maenpää T. Multiresolution gray–scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 24, Issue: 7, Jul 2002)* [interaktyvus]. Univ: IEEE, 2002, 971 – 987 [žiūrėta 2017-04-16]. ISSN 0162-8828. Prieiga per: IEEE Xplore.

17. Hassan T., Khan H. A. Handwritten Bangla Numeral Recognition using Local Binary Pattern. *2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)* [interaktyvus]. Dhaka: IEEE, 2015 [žiūrėta 2016-06-19]. ISBN 978-1-4673-6676-2. Prieiga per: IEEE Xplore.
18. Maenpaa T., Pietikainen M. Classification with color and texture: jointly or separately? *Pattern Recognition (Volume: 37, Issue: 8, Aug 2004)* [interaktyvus]. Oulu: Pergamon, 2004, 1629 – 1640 [žiūrėta 2017-04-16]. Prieiga per: Science Direct.
19. Ahonen T., Hadid A., Pietikainen M. Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 28, Issue: 12, Dec. 2006)* [interaktyvus]. Oulu: IEEE, 2006, 2037 – 2041 [žiūrėta 2017-04-16]. ISSN 0162-8828. Prieiga per: IEEE Xplore.
20. Chen X., Qi C. A super-resolution method for recognition of license plate character using LBP and RBF. *2011 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)* [interaktyvus]. Santander: IEEE, 2011 [žiūrėta 2017-04-16]. ISBN 978-1-4577-1623-2. Prieiga per: IEEE Xplore.
21. Dzemyda Gintautas, Kurasova Olga, Žilinskas Julius. *Daugiamatčių duomenų vizualizavimo metodai: Vadovėlis informatikos krypties doktorantams ir magistrantams* [interaktyvus]. Vilnius: Mokslo aidai, 2008 [žiūrėta 2017-04-15]. ISBN 978-9986-680-42-0. Prieiga per:
<http://web.vu.lt/mii/j.zilinskas/DzemydaKurasovaZilinskasDDVM.pdf>
22. Sharma P. K., Deori M., Kaur B. Radon Transform and PCA based feature extraction to design an Assamese Character Recognition system. *2012 3rd National Conference on Emerging Trends and Applications in Computer Science (NCETACS)* [interaktyvus]. Shillong: IEEE, 2012 [žiūrėta 2017-04-15]. ISBN 978-1-4577-0748-3. Prieiga per: IEEE Xplore.
23. Hao M., Qiao Z. Identification of the Pesticide Fluorescence Spectroscopy based on the PCA and KNN. *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)* [interaktyvus]. Chengdu: IEEE, 2010 [žiūrėta 2017-03-26]. ISBN 978-1-4244-6542-2. Prieiga per: IEEE Xplore.
24. Dallas G. *Principal Component Analysis 4 Dummies: Eigenvectors, Eigenvalues and Dimension Reduction* [interaktyvus]. 2013 [žiūrėta 2017-04-15]. Prieiga per:
<https://georgemdallas.wordpress.com/2013/10/30/principal-component-analysis-4-dummies-eigenvectors-eigenvalues-and-dimension-reduction/>

25. Du M., Ding S., Jia H. Study on density peaks clustering based on k-nearest neighbors and principal component analysis. *Knowledge-Based Systems (Volume 99)* [interaktyvus]. Amsterdam: Elsevier, 2016, 135–145 [žiūrėta 2017-03-26]. ISSN 0950-7051. Prieiga per: Science Direct.
26. Prasad S., Ramkumar B. Passive copy-move forgery detection using SIFT, HOG and SURF features. *IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)* [interaktyvus]. Bangalore: IEEE, 2016 [žiūrėta 2017-04-30]. ISBN 978-1-5090-0774-5. Prieiga per: IEEE Xplore.
27. Jaiswal G., Yadav A. S., Kumar S. Advanced approach for face recognition using Gabor Wavelet and SURF detector (GWSS). *International Conference System Modeling & Advancement in Research Trends (SMART)* [interaktyvus]. Moradabad: IEEE, 2016 [žiūrėta 2017-04-30]. ISBN 978-1-5090-3543-4. Prieiga per: IEEE Xplore.
28. Bay H., Tuytelaars T., Gool L. V. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding (Volume 110, Issue 3, Jun 2007)* [interaktyvus]. Amsterdam: Elsevier, 2008 [žiūrėta 2017-04-30]. ISSN 1090-235X. Prieiga per: Science Direct.
29. Sledevič T. Modifikuoto požymių vaizde išskyrimo SURF algoritmo objektui sekti realiuoju laiku įgyvendintas laiku programuojamoje loginėje matricoje. *Mokslas – Lietuvos ateitis, Vaizdų technologijos T III* [interaktyvus]. Vilnius: 2013; 5(2): 74-78 [žiūrėta 2017-04-29]. ISSN 2029-2341. Prieiga per: <http://www.mla.vgtu.lt/index.php/mla/article/viewFile/mla.2013.12/pdf>
30. Ithaya R. P., Meneeswaran K. Robust Real Time Face detection automatically from video sequence based on Haar features. *2014 International Conference on Communication and Network Technologies (ICCNT)* [interaktyvus]. Sivakasi: IEEE, 2014 [žiūrėta 2016-06-04]. ISBN 978-1-4799-6266-2. Prieiga per: IEEE Xplore.
31. Zhao M, Qian H. M., Jiao R. Y. Robust object tracking via sparse representation based on compressive collaborative Haar-like feature space. *2016 International Conference on Audio, Language and Image Processing (ICALIP)* [interaktyvus]. Shanghai: IEEE, 2016. ISBN 978-1-5090-0654-0. Prieiga per: IEEE Xplore.
32. Tan S. An effective refinement strategy for KNN text classifier. *Expert Systems with Applications (Volume 30, Issue 2, Feb 2006)* [interaktyvus]. Amsterdam: Elsevier, 2006, 290–298 [žiūrėta 2017-04-13]. ISSN 0957-4174. Prieiga per: Science Direct.

33. Wang L., Zhao X. Improved kNN Classification Algorithms Research in Text Categorization. *2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)* [interaktyvus]. Yichang: IEEE, 2012 [žiūrėta 2016-06-4]. ISBN 978-1-4577-1415-3. Prieiga per: IEEE Xplore.
34. Yong Z., Youwen L., Shixiong X. An Improved KNN Text Classification Algorithm Based on Clustering. *Journal of Computers (Volume 4, No. 3, Mar 2009)* [interaktyvus]. Jiangsu: 2009 [žiūrėta 2017-04-22]. Prieiga per: <http://www.jcomputers.us/vol4/jcp0403-08.pdf>
35. Kansheng S., Lemin L., Haitao L.. An improved KNN text classification algorithm based on density. *2011 IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)* [interaktyvus]. Beijing: IEEE, 2011 [žiūrėta 2017-04-23]. ISBN 978-1-61284-204-2. Prieiga per: IEEE Xplore.
36. Hwang W. J., Wen K. W. Fast kNN classification algorithm based on partial distance search. *Electronics Letters (Volume: 34, Issue: 21, Oct 1998)* [interaktyvus]. Taiwan: IET, 1998, 2062-2063 [žiūrėta 2017-04-22]. ISSN 0013-5194. Prieiga per: IEEE Xplore.
37. Lin Y., Wang J. Research on Text Classification Based on SVM–KNN. *2014 5th IEEE International Conference on Software Engineering and Service Science (ICSESS)* [interaktyvus]. Beijing: IEEE, 2014 [žiūrėta 2016-06-04]. ISBN 978-1-4799-3279-5. Prieiga per: IEEE Xplore.
38. Lishuang L., Tingting M., Degen H.. Extracting location names from Chinese texts based on SVM and KNN. *Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE '05)* [interaktyvus]. Wuhan: IEEE, 2005 [žiūrėta 2017-03-26]. ISBN 0-7803-9361-9. Prieiga per: IEEE Xplore.
39. Park S. S., Shin Y. G., Jung W. K., Ahn D. K., Jang D. S. Document form and character recognition using SVM. *AIP Conference Proceedings (Volume 1148, Issue 2)* [interaktyvus]. Hersonissos: AIP, 2009, 180–183 [žiūrėta 2017-03-26]. Prieiga per: <https://koreauniv.pure.elsevier.com/en/publications/document-form-and-character-recognition-using-svm>
40. Zhijie L., Xueqiang L., Kun L. Study on SVM Compared with the other Text Classification Methods. *2010 Second International Workshop on Education Technology and Computer Science (ETCS)* [interaktyvus]. Wuhan: IEEE, 2010 [žiūrėta 2017-04-23]. ISBN 978-1-4244-6389-3. Prieiga per: IEEE Xplore.

41. Zhai X., Bensaali F., Sotudeh R. OCR-based neural network for ANPR. *2012 IEEE International Conference on Imaging Systems and Techniques (IST)* [interaktyvus]. Manchester: IEEE, 2012 [žiūrėta 2017-04-13]. ISSN 1558-2809. Prieiga per: IEEE Xplore.
42. Tuli R. Character Recognition In Neural Networks Using BackPropagation Method. *2013 IEEE 3rd International Advance Computing Conference (IACC)* [interaktyvus]. Ghaziabad: IEEE, 2013 [žiūrėta 2017-03-27]. ISBN 978-1-4673-4529-3. Prieiga per: IEEE Xplore.
43. Sahoo B. K., Meikap B. C. Artificial neural network approach for rheological characteristics of coal-water slurry using microwave pre-treatment. *International Journal of Mining Science and Technology (Volume 27, Issue 2, Mar 2017)* [interaktyvus]. Amsterdam: Elsevier, 2017, 379–386 [žiūrėta 2017-03-27]. ISSN 2095-2686. Prieiga per: Science Direct.
44. Hau-Wen C., Hung-Sik K., Shuyang L. Comparative Study on Subject Classification of Academic Videos Using Noisy Transcripts. *2010 IEEE Fourth International Conference on Semantic Computing (ICSC)* [interaktyvus]. Pittsburgh: IEEE, 2010 [žiūrėta 2017-04-13]. ISBN 978-1-4244-7912-2. Prieiga per: IEEE Xplore.
45. StatSoft: *Naive Bayes Classifier Introductory Overview* [interaktyvus] [žiūrėta 2017-04-13]. Prieiga per: <http://www.statsoft.com/Textbook/Naive-Bayes-Classifier>
46. Bidgoli A. M., Boraghi M. A language independent text segmentation technique based on naive bayes classifier. *2010 International Conference on Signal and Image Processing (ICSIP)* [interaktyvus]. Chennai: IEEE, 2010 [žiūrėta 2017-04-13]. ISBN 978-1-4244-8594-9. Prieiga per: IEEE Xplore.
47. Wei Z., Mei X., Jian-Feng X. A decision tree algorithm for license plate recognition based on bagging. *2012 International Conference on Wavelet Active Media Technology and Information Processing (ICWAMTIP)* [interaktyvus]. Chengdu: IEEE, 2012 [žiūrėta 2017-04-14]. ISBN 978-1-4673-1684-2. Prieiga per: IEEE Xplore.
48. OpenCV 2.4.13.0 documentation: *The Machine Learning Library, Decision Trees* [interaktyvus]. 2014 [žiūrėta 2016-04-20] Prieiga per: http://docs.opencv.org/2.4/modules/ml/doc/decision_trees.html
49. Ruggieri S. Efficient C4.5 [classification algorithm]. *IEEE Transactions on Knowledge and Data Engineering (Volume: 14, Issue: 2, Mar/Apr 2002)* [interaktyvus]. Italy: IEEE, 2002 [žiūrėta 2017-04-14]. ISSN 1041-4347. Prieiga per: IEEE Xplore.

50. Quinlan R. *C4.5: Programs for Machine Learning* [interaktyvus]. London: Morgan Kaufmann, 1993 [žiūrėta 2017-04-19]. ISBN 1-55860-238-0. Prieiga per: Google Books.
51. Tan P. N., Steinbach M., Kumar V. *Introduction to Data Mining*. [interaktyvus]. Boston: Addison-Wesley, 2013 [žiūrėta 2017-04-15]. ISBN 978-0133128901. Prieiga per: <http://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf>
52. Freund Y., Schapire R. E. Experiments with a New Boosting Algorithm. *Machine Learning: Proceedings of the Thirteenth International Conference* [interaktyvus]. Boston: Morgan Kaufmann, 1996, 148-156 [žiūrėta 2017-04-24]. Prieiga per: <http://www.cis.upenn.edu/~mkearns/teaching/COLT/boostingexperiments.pdf>

PRIEDAI

1 lentelė. Pirmojo tyrimo sprendimo medžio klasifikatoriaus rezultatų duomenys (tikslumas, %)

Pastabų skaičius per šaką	Deskriptorius	Pikseliai		HOG		LBP		PCA	
	Vaizdo Nr.	1	2	1	2	1	2	1	2
1		100,0	93,5	81,8	54,8	54,5	25,8	68,2	64,5
2		100,0	93,5	81,8	54,8	54,5	25,8	68,2	64,5
3		100,0	93,5	81,8	54,8	54,5	25,8	68,2	64,5
4		100,0	93,5	81,8	54,8	54,5	25,8	72,7	64,5
5		100,0	93,5	81,8	54,8	50,0	25,8	72,7	64,5
6		100,0	93,5	81,8	54,8	50,0	29,0	72,7	64,5
7		100,0	93,5	81,8	54,8	50,0	25,8	68,2	64,5
8		100,0	93,5	81,8	54,8	50,0	25,8	68,2	64,5
9		100,0	93,5	81,8	54,8	50,0	22,6	68,2	64,5
10		100,0	93,5	81,8	45,2	50,0	25,8	68,2	64,5
11		100,0	93,5	81,8	45,2	50,0	25,8	68,2	64,5
12		100,0	93,5	81,8	45,2	50,0	35,5	68,2	64,5
13		100,0	93,5	81,8	45,2	45,5	35,5	68,2	64,5
14		100,0	93,5	81,8	45,2	45,5	35,5	68,2	64,5
15		100,0	93,5	77,3	48,4	45,5	35,5	72,7	64,5
16		100,0	93,5	77,3	48,4	45,5	35,5	72,7	64,5
17		100,0	93,5	77,3	48,4	45,5	41,9	72,7	64,5
18		100,0	93,5	77,3	48,4	45,5	41,9	72,7	64,5
19		100,0	93,5	77,3	48,4	45,5	41,9	72,7	64,5
20		100,0	93,5	77,3	48,4	45,5	41,9	72,7	64,5
21		100,0	93,5	77,3	48,4	45,5	41,9	72,7	64,5
22		100,0	93,5	77,3	48,4	45,5	41,9	72,7	64,5
23		100,0	93,5	77,3	51,6	45,5	41,9	72,7	64,5
24		100,0	93,5	77,3	51,6	45,5	41,9	77,3	64,5
25		100,0	93,5	77,3	51,6	45,5	41,9	77,3	64,5
26		100,0	93,5	77,3	51,6	45,5	41,9	77,3	67,7
27		100,0	93,5	77,3	51,6	45,5	41,9	77,3	67,7
28		100,0	93,5	77,3	51,6	45,5	41,9	77,3	67,7
29		100,0	93,5	77,3	51,6	45,5	41,9	77,3	67,7
30		100,0	93,5	72,7	51,6	45,5	41,9	77,3	67,7
31		100,0	93,5	72,7	51,6	45,5	41,9	77,3	71,0
32		100,0	93,5	72,7	51,6	45,5	41,9	77,3	71,0
33		100,0	93,5	72,7	51,6	45,5	45,2	77,3	71,0
34		100,0	93,5	72,7	51,6	45,5	45,2	77,3	71,0
35		100,0	93,5	72,7	51,6	45,5	45,2	77,3	71,0
36		100,0	93,5	72,7	51,6	40,9	35,5	77,3	71,0
37		100,0	93,5	72,7	51,6	40,9	35,5	77,3	71,0
38		100,0	93,5	72,7	51,6	40,9	35,5	72,7	71,0
39		100,0	93,5	72,7	51,6	40,9	35,5	72,7	71,0
40		100,0	93,5	72,7	51,6	40,9	29,0	72,7	71,0
41		100,0	93,5	72,7	51,6	40,9	29,0	72,7	71,0
42		100,0	93,5	72,7	51,6	40,9	29,0	72,7	71,0
43		100,0	93,5	77,3	51,6	40,9	29,0	72,7	71,0
44		100,0	93,5	77,3	51,6	40,9	29,0	72,7	71,0
45		100,0	93,5	77,3	48,4	40,9	29,0	72,7	71,0
46		100,0	93,5	77,3	48,4	40,9	29,0	72,7	71,0
47		100,0	93,5	77,3	48,4	40,9	29,0	68,2	71,0
48		100,0	93,5	77,3	51,6	40,9	32,3	68,2	71,0
49		100,0	93,5	77,3	48,4	40,9	32,3	63,6	71,0
50		100,0	93,5	77,3	48,4	40,9	32,3	63,6	71,0

2 lentelė. Pirmojo tyrimo sprendimo medžio klasifikatoriaus rezultatų duomenys (laikas, ms)

Pastabų skaičius per šaka	Deskriptorius	Pikseliai		HOG		LBP		PCA	
	Vaizdo Nr.	1	2	1	2	1	2	1	2
1		122	78	85	112	87	100	91	113
2		57	63	70	85	69	83	76	91
3		54	74	68	82	66	82	73	108
4		50	59	65	76	62	73	70	87
5		51	56	65	74	63	72	81	81
6		49	55	64	73	62	71	69	80
7		49	55	62	74	61	71	68	80
8		49	55	64	74	63	72	69	80
9		50	66	64	74	61	70	69	80
10		49	55	63	73	62	69	68	79
11		48	53	62	73	61	69	68	79
12		48	52	62	73	60	69	68	94
13		49	53	63	73	61	69	68	79
14		49	52	63	71	60	69	76	79
15		48	52	63	71	59	68	67	79
16		48	54	62	72	59	69	67	83
17		48	52	62	71	60	68	67	78
18		48	51	62	70	60	68	76	77
19		48	53	62	70	59	67	67	77
20		48	51	61	69	60	67	67	77
21		48	51	62	69	59	67	67	83
22		48	51	62	69	60	67	71	77
23		48	50	62	70	59	67	77	76
24		49	50	62	69	59	66	66	77
25		57	50	62	70	59	66	66	84
26		48	50	62	70	59	66	66	76
27		48	50	62	70	59	66	67	76
28		47	50	62	69	59	66	66	85
29		47	50	61	69	59	65	66	76
30		47	50	62	69	59	66	68	83
31		48	50	62	69	59	66	66	77
32		47	50	62	70	59	67	67	85
33		47	51	61	84	59	70	66	101
34		47	59	60	75	58	74	66	82
35		48	53	60	76	58	68	67	87
36		47	51	61	70	58	67	74	91
37		47	50	61	70	58	66	66	82
38		48	50	62	70	59	66	68	78
39		48	49	62	68	58	65	70	76
40		47	50	61	68	59	65	66	80
41		47	49	61	67	58	65	66	75
42		48	49	64	68	59	65	70	79
43		47	49	61	68	58	65	66	76
44		48	49	63	68	59	65	79	82
45		48	49	62	68	59	65	68	75
46		51	49	64	69	61	64	71	75
47		50	49	64	68	61	65	69	75
48		50	49	61	69	58	65	66	75
49		50	49	62	69	58	65	66	75
50		56	49	61	68	58	65	66	75

3 lentelė. Pirmojo tyrimo k -artimiausių kaimynų klasifikatoriaus rezultatų duomenys (tikslumas, %)

Kaimynų skaičius k	Deskriptorius	Pikseliai		HOG		LBP		PCA	
	Vaizdo Nr.	1	2	1	2	1	2	1	2
1		100,0	96,8	90,9	71,0	77,3	45,2	45,5	38,7
2		100,0	96,8	95,5	67,7	72,7	45,2	68,2	41,9
3		100,0	96,8	100,0	67,7	77,3	51,6	68,2	48,4
4		100,0	96,8	100,0	64,5	72,7	48,4	63,6	41,9
5		100,0	96,8	100,0	64,5	77,3	58,1	59,1	41,9
6		100,0	96,8	100,0	67,7	81,8	48,4	54,5	38,7
7		100,0	96,8	95,5	67,7	72,7	54,8	59,1	38,7
8		100,0	96,8	95,5	67,7	77,3	54,8	59,1	45,2
9		100,0	96,8	95,5	67,7	72,7	54,8	54,5	38,7
10		100,0	96,8	95,5	71,0	77,3	51,6	63,6	38,7
11		100,0	96,8	90,9	74,2	77,3	58,1	63,6	45,2
12		100,0	96,8	90,9	74,2	77,3	54,8	68,2	48,4
13		100,0	96,8	90,9	71,0	72,7	41,9	68,2	48,4
14		100,0	96,8	95,5	67,7	72,7	54,8	72,7	51,6
15		100,0	96,8	95,5	71,0	72,7	54,8	72,7	45,2
16		100,0	96,8	95,5	71,0	68,2	51,6	72,7	54,8
17		100,0	96,8	95,5	71,0	68,2	54,8	72,7	51,6
18		100,0	96,8	95,5	71,0	68,2	54,8	77,3	54,8
19		100,0	96,8	95,5	71,0	68,2	54,8	77,3	58,1
20		100,0	96,8	95,5	71,0	68,2	54,8	77,3	58,1
21		100,0	96,8	95,5	67,7	68,2	54,8	72,7	61,3
22		100,0	96,8	95,5	67,7	68,2	51,6	68,2	61,3
23		100,0	96,8	95,5	64,5	72,7	45,2	72,7	61,3
24		100,0	96,8	95,5	64,5	63,6	41,9	72,7	61,3
25		100,0	96,8	95,5	64,5	63,6	41,9	81,8	61,3
26		100,0	96,8	95,5	64,5	63,6	41,9	72,7	64,5
27		100,0	96,8	95,5	61,3	68,2	41,9	63,6	64,5
28		100,0	96,8	95,5	67,7	68,2	38,7	68,2	64,5
29		100,0	96,8	95,5	67,7	68,2	38,7	63,6	71,0
30		100,0	96,8	95,5	71,0	63,6	38,7	68,2	67,7
31		100,0	96,8	95,5	67,7	63,6	38,7	63,6	71,0
32		100,0	96,8	95,5	71,0	63,6	35,5	59,1	67,7
33		100,0	96,8	95,5	67,7	63,6	38,7	63,6	67,7
34		100,0	96,8	95,5	67,7	63,6	35,5	59,1	64,5
35		100,0	96,8	95,5	67,7	63,6	32,3	63,6	67,7
36		100,0	96,8	95,5	67,7	63,6	35,5	63,6	67,7
37		100,0	96,8	95,5	67,7	63,6	32,3	63,6	64,5
38		100,0	96,8	95,5	67,7	63,6	29,0	63,6	67,7
39		100,0	96,8	95,5	71,0	63,6	35,5	63,6	64,5
40		100,0	100,0	95,5	74,2	59,1	29,0	63,6	64,5
41		100,0	100,0	95,5	74,2	59,1	25,8	63,6	61,3
42		95,5	100,0	95,5	74,2	59,1	32,3	63,6	61,3
43		95,5	100,0	95,5	77,4	59,1	25,8	63,6	61,3
44		95,5	100,0	95,5	77,4	59,1	29,0	59,1	58,1
45		95,5	100,0	95,5	77,4	59,1	32,3	59,1	61,3
46		95,5	100,0	95,5	80,6	59,1	32,3	54,5	61,3
47		95,5	100,0	95,5	80,6	59,1	35,5	68,2	61,3
48		95,5	100,0	95,5	80,6	59,1	35,5	54,5	64,5
49		95,5	100,0	90,9	80,6	59,1	35,5	59,1	61,3
50		95,5	96,8	90,9	80,6	59,1	35,5	59,1	58,1

4 lentelė. Pirmojo tyrimo k -artimiausių kaimynų klasifikatoriaus rezultatų duomenys (laikas, ms)

Kaimynų skaičius k	Deskriptorius	Pikseliai		HOG		LBP		PCA	
	Vaizdo Nr.	1	2	1	2	1	2	1	2
1		190	262	147	228	143	192	200	248
2		178	246	131	186	125	167	170	233
3		170	237	127	174	122	158	176	224
4		168	228	121	176	116	146	163	218
5		170	226	122	174	130	146	167	219
6		175	226	135	180	116	157	163	237
7		169	225	120	182	115	150	171	217
8		167	226	119	188	114	164	163	230
9		168	225	120	198	116	166	163	215
10		167	224	119	208	145	178	178	215
11		168	225	124	216	134	191	165	226
12		167	225	129	255	135	197	166	231
13		168	225	138	253	138	224	171	237
14		167	224	137	243	163	220	162	267
15		167	224	142	245	151	228	168	271
16		168	224	163	264	155	236	165	255
17		169	223	152	251	158	246	159	213
18		168	224	156	263	162	253	161	217
19		169	223	161	271	171	275	164	210
20		168	223	168	278	171	274	161	212
21		168	223	183	277	176	281	168	213
22		170	223	174	285	198	291	164	225
23		168	223	181	297	179	298	161	210
24		169	222	183	302	182	306	159	211
25		169	222	188	311	185	316	168	221
26		169	221	210	318	191	340	167	208
27		168	223	198	329	195	336	162	219
28		170	221	203	351	201	344	166	217
29		169	224	207	338	203	351	163	208
30		168	222	212	343	220	353	162	219
31		169	222	217	330	215	344	166	220
32		168	223	220	341	219	355	161	215
33		168	233	226	347	224	364	161	253
34		171	226	228	348	226	351	160	233
35		168	224	242	363	233	359	161	227
36		168	223	240	358	238	345	161	210
37		169	222	244	342	241	351	160	210
38		168	223	250	345	244	353	172	223
39		168	222	254	338	266	361	164	211
40		169	222	260	348	253	355	161	217
41		168	222	261	340	259	350	161	211
42		168	222	264	343	260	347	166	207
43		169	222	252	351	252	357	165	220
44		168	220	278	353	260	361	172	210
45		168	221	264	343	261	347	160	209
46		169	220	269	351	264	372	177	206
47		167	221	255	338	268	342	169	217
48		167	221	266	348	256	353	177	207
49		170	221	267	350	262	351	161	215
50		169	221	255	358	255	344	164	210

5 lentelė. Pirmojo tyrimo dirbtinių neuroninių tinklų klasifikatoriaus rezultatų duomenys (tikslumas, %)

Paslėptų sluoksnių skaičius	Deskriptorius	Pikseliai		HOG		LBP		PCA	
	Vaizdo Nr.	1	2	1	2	1	2	1	2
1		18,2	9,7	22,7	6,5	13,6	29,0	9,1	3,2
2		18,2	35,5	68,2	54,8	50,0	38,7	27,3	22,6
3		27,3	45,2	81,8	77,4	59,1	67,7	22,7	29,0
4		81,8	61,3	90,9	74,2	86,4	80,6	36,4	32,3
5		95,5	93,5	81,8	64,5	86,4	77,4	31,8	41,9
6		95,5	96,8	90,9	71,0	90,9	80,6	50,0	29,0
7		100,0	41,9	72,7	77,4	86,4	64,5	31,8	25,8
8		86,4	100,0	81,8	71,0	81,8	74,2	68,2	16,1
9		100,0	96,8	77,3	80,6	77,3	64,5	50,0	48,4
10		95,5	100,0	81,8	67,7	90,9	87,1	50,0	51,6
11		90,9	100,0	77,3	64,5	77,3	64,5	36,4	38,7
12		100,0	100,0	77,3	58,1	81,8	74,2	54,5	29,0
13		100,0	96,8	90,9	67,7	86,4	64,5	40,9	48,4
14		90,9	100,0	95,5	80,6	77,3	77,4	59,1	54,8
15		100,0	100,0	86,4	77,4	72,7	61,3	45,5	48,4
16		95,5	100,0	90,9	74,2	81,8	77,4	63,6	38,7
17		100,0	100,0	81,8	74,2	86,4	64,5	45,5	64,5
18		90,9	96,8	100,0	67,7	81,8	83,9	50,0	54,8
19		100,0	100,0	77,3	80,6	86,4	67,7	50,0	64,5
20		95,5	100,0	86,4	80,6	77,3	61,3	68,2	74,2
21		95,5	100,0	90,9	67,7	72,7	71,0	72,7	51,6
22		100,0	100,0	95,5	83,9	90,9	64,5	72,7	51,6
23		100,0	100,0	86,4	77,4	86,4	67,7	68,2	58,1
24		100,0	100,0	95,5	74,2	86,4	67,7	59,1	48,4
25		100,0	100,0	100,0	74,2	86,4	71,0	63,6	54,8
26		100,0	100,0	90,9	71,0	86,4	58,1	72,7	54,8
27		100,0	100,0	95,5	77,4	77,3	64,5	86,4	58,1
28		100,0	100,0	90,9	71,0	81,8	61,3	81,8	45,2
29		100,0	100,0	95,5	80,6	86,4	74,2	54,5	67,7
30		100,0	100,0	90,9	90,3	86,4	67,7	68,2	64,5
31		100,0	100,0	95,5	80,6	77,3	67,7	59,1	64,5
32		100,0	100,0	90,9	74,2	77,3	67,7	63,6	54,8
33		100,0	100,0	86,4	71,0	81,8	77,4	72,7	61,3
34		100,0	100,0	100,0	71,0	77,3	83,9	77,3	51,6
35		100,0	100,0	95,5	80,6	77,3	61,3	77,3	54,8
36		100,0	100,0	90,9	77,4	81,8	61,3	63,6	54,8
37		100,0	100,0	95,5	80,6	72,7	83,9	68,2	61,3
38		100,0	100,0	90,9	71,0	81,8	67,7	63,6	58,1
39		100,0	100,0	95,5	77,4	86,4	67,7	72,7	61,3
40		100,0	100,0	95,5	67,7	77,3	71,0	77,3	54,8
41		100,0	100,0	90,9	83,9	81,8	71,0	72,7	54,8
42		100,0	100,0	100,0	71,0	81,8	67,7	81,8	54,8
43		100,0	100,0	90,9	77,4	86,4	83,9	77,3	54,8
44		100,0	100,0	90,9	67,7	90,9	74,2	59,1	61,3
45		100,0	100,0	95,5	83,9	77,3	67,7	63,6	48,4
46		100,0	100,0	90,9	71,0	90,9	61,3	77,3	61,3
47		100,0	100,0	95,5	77,4	72,7	77,4	81,8	58,1
48		100,0	100,0	90,9	77,4	81,8	67,7	68,2	58,1
49		100,0	100,0	90,9	74,2	81,8	61,3	68,2	64,5
50		100,0	100,0	95,5	80,6	90,9	71,0	68,2	61,3

6 lentelė. Pirmojo tyrimo dirbtinių neuroninių tinklų klasifikatoriaus rezultatų duomenys (laikas, ms)

Paslėptų sluoksnių skaičius	Deskriptorius	Pikseliai		HOG		LBP		PCA	
	Vaizdo Nr.	1	2	1	2	1	2	1	2
1		417	333	293	1475	287	455	681	427
2		258	322	277	436	269	387	368	405
3		249	319	273	408	267	381	369	403
4		247	312	268	402	261	367	332	367
5		257	313	270	419	263	361	320	379
6		243	307	267	377	262	358	317	374
7		238	310	266	373	261	355	298	398
8		239	308	268	387	263	353	318	379
9		240	314	268	367	262	355	306	395
10		236	310	268	368	262	356	305	406
11		239	308	266	379	260	363	361	424
12		234	311	266	363	259	356	305	415
13		240	313	267	363	259	356	301	422
14		244	318	268	362	261	355	299	405
15		244	319	266	371	259	356	326	372
16		250	325	264	391	260	357	318	386
17		243	319	267	388	262	374	305	370
18		242	322	265	365	259	354	311	397
19		245	320	266	360	262	355	318	401
20		247	319	266	359	259	353	319	393
21		247	323	266	359	260	353	310	391
22		252	321	269	359	259	351	339	392
23		263	322	265	358	261	354	336	380
24		246	323	265	362	261	354	293	391
25		244	323	268	360	261	354	305	437
26		245	320	267	360	262	358	315	382
27		245	329	270	363	261	350	288	422
28		245	324	268	359	262	350	288	380
29		247	327	271	367	261	350	319	402
30		246	327	267	372	263	353	298	406
31		247	326	270	394	261	352	302	419
32		258	358	265	400	260	389	315	391
33		292	393	269	397	261	385	296	442
34		266	365	269	387	260	356	326	388
35		279	404	267	387	262	351	321	397
36		269	374	268	383	260	348	330	405
37		256	372	265	412	260	347	290	447
38		270	347	271	385	261	346	297	372
39		257	355	272	353	263	347	313	401
40		295	361	267	348	261	345	306	390
41		283	392	268	347	261	345	324	424
42		298	384	270	349	263	346	337	410
43		264	351	267	347	260	348	355	394
44		326	352	272	348	265	348	362	418
45		278	349	273	344	262	346	357	420
46		282	360	283	347	277	348	366	392
47		283	340	270	349	263	349	396	401
48		274	378	274	346	260	355	402	406
49		269	362	268	346	265	349	367	418
50		288	356	300	346	260	349	365	426

7 lentelė. Pirmojo tyrimo atraminių vektorių mašinos klasifikatoriaus rezultatų duomenys (tikslumas, %)

Polinomo laipsnis	Deskriptorius	Pikseliai		HOG		LBP		PCA	
	Vaizdo Nr.	1	2	1	2	1	2	1	2
1		100	100	90,91	67,74	36,36	16,13	54,55	48,39
2		95,46	100	90,91	70,97	36,36	16,13	77,27	61,29
3		95,46	93,55	90,91	70,97	50	22,58	90,91	64,52
4		100	100	90,91	70,97	54,55	29,03	86,36	67,74

8 lentelė. Pirmojo tyrimo atraminių vektorių mašinos klasifikatoriaus rezultatų duomenys (laikas, ms)

Polinomo laipsnis	Deskriptorius	Pikseliai		HOG		LBP		PCA	
	Vaizdo Nr.	1	2	1	2	1	2	1	2
1		3558	4494	2584	3340	2711	3582	3483	4604
2		2792	4085	2330	3295	2501	3389	3116	4395
3		2386	3322	2340	3295	2457	3387	3138	4615
4		2530	3620	2343	3282	2402	3347	3189	4534

9 lentelė. Pirmojo tyrimo Naive Bajeso klasifikatoriaus rezultatų duomenys (tikslumas, %)

Nr.	Deskriptorius	Pikseliai		HOG		LBP		PCA	
	Vaizdo Nr.	1	2	1	2	1	2	1	2
1		100	96,77	-	-	-	-	-	-

10 lentelė. Pirmojo tyrimo Naive Bajeso klasifikatoriaus rezultatų duomenys (laikas, ms)

Nr.	Deskriptorius	Pikseliai		HOG		LBP		PCA	
	Vaizdo Nr.	1	2	1	2	1	2	1	2
1		541	533	-	-	-	-	-	-

11 lentelė. Trečiojo tyrimo k-artimiausių kaimynų klasifikatoriaus rezultatų duomenys (deskriptorius – pikselių vektorius; simbolio dydis – 30 x 20)

Kaimynų skaičius k	Tikslumas, %	Laikas, ms	Kaimynų skaičius k	Tikslumas, %	Laikas, ms
1	95,7	199	26	96,4	192
2	97,0	198	27	96,3	192
3	96,0	197	28	96,5	192
4	96,5	193	29	96,6	193
5	96,7	193	30	96,8	192
6	96,7	193	31	96,6	193
7	96,5	192	32	96,7	192
8	96,4	194	33	96,7	193
9	96,7	194	34	96,7	193
10	96,7	193	35	96,8	193
11	96,9	194	36	96,8	192
12	96,8	194	37	96,9	193
13	96,8	192	38	96,8	191
14	96,7	193	39	96,8	193
15	96,9	192	40	96,8	191
16	96,7	194	41	96,8	193
17	96,5	193	42	97,0	191
18	96,8	193	43	97,0	193
19	97,0	193	44	96,8	190
20	96,8	192	45	96,8	193
21	96,9	193	46	96,7	190
22	96,8	193	47	96,6	193
23	96,5	193	48	96,7	191
24	96,6	193	49	96,5	193
25	96,6	192	50	96,4	191

12 lentelė. Trečiojo tyrimo dirbtinių neuroninių tinklų klasifikatoriaus rezultatų duomenys (deskriptorius – pikselių vektorius; simbolio dydis – 30 x 20)

Paslėptų sluoksnių sk.	Tikslumas, %	Laikas, ms	Paslėptų sluoksnių sk.	Tikslumas, %	Laikas, ms
1	11,4	265	26	97,4	278
2	27,6	265	27	98,3	278
3	42,3	266	28	96,6	279
4	52,5	262	29	98,1	280
5	92,2	262	30	98,9	280
6	84,9	263	31	97,2	281
7	90,2	265	32	98,2	308
8	92,1	263	33	98,7	309
9	89,3	268	34	98,7	309
10	94,4	264	35	98,0	310
11	95,7	266	36	98,5	309
12	94,6	265	37	97,2	311
13	94,5	267	38	98,2	308
14	98,1	273	39	98,0	310
15	98,8	274	40	98,4	311
16	97,8	274	41	98,8	312
17	97,1	275	42	97,9	309
18	97,1	275	43	98,5	307
19	98,4	275	44	97,4	311
20	98,1	276	45	98,3	329
21	96,7	278	46	98,9	313
22	98,5	276	47	98,5	319
23	98,3	277	48	97,9	329
24	95,6	278	49	96,6	323
25	98,7	278	50	98,1	329

13 lentelė. Trečiojo tyrimo dirbtinių neuroninių tinklų klasifikatoriaus rezultatų duomenys (deskriptorius – pikselių vektorius; simbolio dydis – 15 x 10)

Paslėptų sluoksnių sk.	Tikslumas, %	Laikas, ms	Paslėptų sluoksnių sk.	Tikslumas, %	Laikas, ms
1	23,6	265	26	99,3	264
2	69,7	265	27	99,1	264
3	19,6	265	28	99,2	264
4	70,3	264	29	98,6	265
5	88,5	262	30	98,9	265
6	83,0	261	31	98,8	265
7	82,3	261	32	99,6	266
8	93,7	262	33	98,9	265
9	95,3	262	34	98,9	266
10	94,7	262	35	99,4	265
11	97,3	263	36	99,2	266
12	95,2	262	37	99,1	268
13	98,3	263	38	99,0	266
14	97,6	262	39	99,2	266
15	98,1	262	40	99,1	266
16	98,7	266	41	99,0	268
17	99,0	262	42	99,1	268
18	98,4	263	43	99,0	266
19	97,8	263	44	99,1	267
20	99,2	263	45	99,3	268
21	98,5	264	46	99,2	267
22	98,3	264	47	99,0	270
23	98,3	264	48	99,2	268
24	99,0	265	49	99,2	268
25	99,1	264	50	99,4	269

14 lentelė. Trečiojo tyrimo dirbtinių neuroninių tinklų klasifikatoriaus rezultatų duomenys (deskriptorius – HOG; simbolio dydis – 60 x 40)

Paslėptų sluoksnių sk.	Tikslumas, %	Laikas, ms	Paslėptų sluoksnių sk.	Tikslumas, %	Laikas, ms
1	8,5	304	26	96,8	321
2	18,2	306	27	97,0	320
3	59,0	305	28	96,7	320
4	43,9	305	29	97,5	320
5	86,2	304	30	97,8	321
6	94,2	305	31	97,4	321
7	90,0	306	32	97,6	348
8	92,8	306	33	97,9	357
9	92,2	306	34	97,7	353
10	93,7	309	35	97,3	355
11	95,6	307	36	97,4	354
12	94,8	307	37	97,7	354
13	96,0	314	38	98,1	356
14	95,8	315	39	97,6	351
15	96,2	314	40	97,8	350
16	96,6	317	41	97,5	356
17	97,1	316	42	98,4	361
18	97,6	316	43	96,6	354
19	97,4	318	44	96,7	360
20	97,0	316	45	97,5	356
21	97,3	317	46	98,0	370
22	97,8	318	47	97,4	375
23	96,9	318	48	97,6	368
24	98,3	320	49	97,4	367
25	98,2	319	50	97,4	364

15 lentelė. Trečiojo tyrimo dirbtinių neuroninių tinklų klasifikatoriaus rezultatų duomenys (deskriptorius – Pikselių vektorius; simbolio dydis – 60 x 40)

Paslėptų sluoksnių sk.	Tikslumas, %	Laikas, ms	Paslėptų sluoksnių sk.	Tikslumas, %	Laikas, ms
1	11,7	266	26	99,0	310
2	24,1	270	27	97,2	312
3	42,7	273	28	98,8	314
4	75,7	280	29	98,2	313
5	84,4	281	30	96,8	315
6	59,4	282	31	98,4	315
7	92,3	283	32	98,7	348
8	93,6	285	33	98,4	345
9	95,9	287	34	97,9	351
10	97,4	288	35	99,0	349
11	96,6	289	36	98,2	354
12	97,1	290	37	98,7	356
13	96,8	292	38	98,6	360
14	98,2	294	39	98,9	360
15	95,9	294	40	98,3	367
16	97,9	296	41	98,6	361
17	98,9	298	42	98,2	370
18	98,4	300	43	98,0	381
19	98,2	300	44	98,8	371
20	98,0	302	45	98,3	378
21	98,1	304	46	98,2	380
22	97,9	304	47	98,6	383
23	98,7	306	48	98,8	392
24	97,5	308	49	98,8	381
25	99,1	308	50	97,6	392

16 lentelė. Trečiojo tyrimo k-artimiausių kaimynų klasifikatoriaus rezultatų duomenys (deskriptorius – Pikselių vektorius; simbolio dydis – 15 x 10)

Kaimynų skaičius <i>k</i>	Tikslumas, %	Laikas, ms	Kaimynų skaičius <i>k</i>	Tikslumas, %	Laikas, ms
1	97,8	100	26	97,4	93
2	98,6	97	27	97,4	93
3	97,9	98	28	97,3	93
4	97,9	93	29	97,2	93
5	97,7	94	30	97,1	93
6	97,8	94	31	97,0	93
7	97,7	93	32	97,3	93
8	97,5	94	33	97,2	93
9	97,9	93	34	97,0	94
10	97,8	94	35	97,0	94
11	97,8	94	36	97,0	93
12	97,8	93	37	97,1	94
13	97,6	93	38	97,2	93
14	97,7	93	39	97,0	93
15	97,5	93	40	96,9	94
16	97,7	93	41	97,0	93
17	97,6	93	42	97,0	93
18	97,7	94	43	97,1	94
19	97,9	93	44	96,8	94
20	97,8	93	45	96,9	93
21	97,9	94	46	96,8	93
22	97,6	94	47	97,0	93
23	97,3	93	48	96,8	93
24	97,3	93	49	96,9	94
25	97,5	93	50	96,7	94

17 lentelė. Trečiojo tyrimo atraminių vektorių mašinos klasifikatoriaus rezultatų duomenys (deskriptorius – Pikselių vektorius; simbolio dydis – 30 x 20)

Polinomo laipsnis	Tikslumas, %	Laikas, ms	Polinomo laipsnis	Tikslumas, %	Laikas, ms
1	96,8	3278	3	89,9	2807
2	96,5	3520	4	91,9	2998

18 lentelė. Trečiojo tyrimo atraminių vektorių mašinos klasifikatoriaus rezultatų duomenys (deskriptorius – HOG; simbolio dydis – 60 x 40)

Polinomo laipsnis	Tikslumas, %	Laikas, ms	Polinomo laipsnis	Tikslumas, %	Laikas, ms
1	98,1	3114	3	98,2	3239
2	98,5	3169	4	98,0	3296

19 lentelė. Trečiojo tyrimo atraminių vektorių mašinos klasifikatoriaus rezultatų duomenys (deskriptorius – Pikselių vektorius; simbolio dydis – 60 x 40)

Polinomo laipsnis	Tikslumas, %	Laikas, ms	Polinomo laipsnis	Tikslumas, %	Laikas, ms
1	98,7	4140	3	98,5	3381
2	99,0	4288	4	98,6	3503

20 lentelė. Trečiojo tyrimo Naive Bajeso klasifikatoriaus rezultatų duomenys (deskriptorius – Pikselių vektorius; simbolio dydis – 15 x 10)

Nr.	Tikslumas, %	Laikas, ms
1	97,7	209

21 lentelė. Ketvirtojo tyrimo k-artimiausių kaimynų klasifikatoriaus rezultatų duomenys (deskriptorius – pikselių vektorius; simbolio dydis – 30 x 20)

Kaimynų skaičius <i>k</i>	Tikslumas, %	Laikas, ms	Kaimynų skaičius <i>k</i>	Tikslumas, %	Laikas, ms
1	99,6	354	26	99,3	337
2	99,6	355	27	99,3	335
3	99,6	371	28	99,3	341
4	99,6	366	29	99,3	339
5	99,6	359	30	99,3	341
6	99,5	363	31	99,3	339
7	99,6	343	32	99,3	341
8	99,6	343	33	99,3	340
9	99,6	344	34	99,3	341
10	99,5	344	35	99,4	340
11	99,5	348	36	99,3	341
12	99,5	343	37	99,3	339
13	99,5	343	38	99,3	341
14	99,5	351	39	99,4	342
15	99,5	344	40	99,3	343
16	99,4	346	41	99,3	335
17	99,4	343	42	99,3	337
18	99,4	345	43	99,3	340
19	99,4	344	44	99,3	343
20	99,3	348	45	99,3	342
21	99,3	348	46	99,3	345
22	99,3	345	47	99,3	343
23	99,3	344	48	99,3	343
24	99,3	345	49	99,3	343
25	99,3	351	50	99,3	343

22 lentelė. Ketvirtojo tyrimo dirbtinių neuroninių tinklų klasifikatoriaus rezultatų duomenys (deskriptorius – pikselių vektorius; simbolio dydis – 30 x 20)

Paslėptų sluoksnių sk.	Tikslumas, %	Laikas, ms	Paslėptų sluoksnių sk.	Tikslumas, %	Laikas, ms
1	33,5	261	26	99,7	278
2	57,4	268	27	99,6	277
3	74,7	265	28	99,5	280
4	92,3	265	29	99,7	277
5	92,1	265	30	99,6	277
6	98,4	264	31	99,7	276
7	98,7	264	32	99,7	312
8	99,1	267	33	99,8	311
9	99,1	265	34	99,6	311
10	99,2	266	35	99,7	311
11	99,3	270	36	99,7	311
12	99,5	267	37	99,6	310
13	99,8	270	38	99,7	314
14	99,5	272	39	99,8	317
15	99,5	273	40	99,7	323
16	99,5	275	41	99,7	328
17	99,8	273	42	99,7	333
18	99,7	276	43	99,7	335
19	99,7	274	44	99,7	331
20	99,5	273	45	99,7	331
21	99,4	273	46	99,8	336
22	99,5	275	47	99,7	329
23	99,6	274	48	99,7	344
24	99,7	274	49	99,7	335
25	99,7	275	50	99,8	338

23 lentelė. Ketvirtojo tyrimo dirbtinių neuroninių tinklų klasifikatoriaus rezultatų duomenys
(deskriptorius – pikselių vektorius; simbolio dydis – 15 x 10)

Paslėptų sluoksnių sk.	Tikslumas, %	Laikas, ms	Paslėptų sluoksnių sk.	Tikslumas, %	Laikas, ms
1	50,6	254	26	99,7	268
2	77,6	258	27	99,5	257
3	90,2	259	28	99,8	257
4	90,2	258	29	99,6	257
5	94,8	259	30	99,7	257
6	94,7	258	31	99,6	258
7	94,0	249	32	99,7	258
8	98,4	254	33	99,8	258
9	98,9	255	34	99,6	257
10	98,6	254	35	99,7	258
11	99,3	254	36	99,6	259
12	98,9	254	37	99,7	259
13	99,1	255	38	99,6	258
14	99,5	255	39	99,8	259
15	99,5	255	40	99,7	258
16	99,5	255	41	99,7	259
17	99,5	255	42	99,6	259
18	99,6	255	43	99,8	259
19	99,5	257	44	99,7	259
20	99,7	256	45	99,5	262
21	99,6	256	46	99,6	260
22	99,7	256	47	99,7	261
23	99,7	256	48	99,7	260
24	99,6	269	49	99,7	267
25	99,6	283	50	99,7	263

24 lentelė. Ketvirtojo tyrimo dirbtinių neuroninių tinklų klasifikatoriaus rezultatų duomenys
(deskriptorius – HOG; simbolio dydis – 60 x 40)

Paslėptų sluoksnių sk.	Tikslumas, %	Laikas, ms	Paslėptų sluoksnių sk.	Tikslumas, %	Laikas, ms
1	50,6	322	26	99,3	314
2	65,0	300	27	99,7	315
3	80,4	301	28	99,4	314
4	88,6	299	29	99,4	314
5	94,5	300	30	99,2	316
6	97,8	299	31	99,6	316
7	92,8	299	32	99,5	364
8	91,5	303	33	99,6	390
9	98,4	301	34	99,8	384
10	99,2	301	35	99,7	352
11	99,0	302	36	99,8	355
12	99,0	302	37	99,5	359
13	99,3	307	38	99,4	355
14	99,4	309	39	99,5	351
15	99,0	309	40	99,5	353
16	99,5	308	41	99,6	361
17	99,5	310	42	99,7	350
18	99,5	312	43	99,7	353
19	98,5	311	44	99,5	359
20	99,4	310	45	99,5	368
21	99,6	313	46	99,7	370
22	99,7	312	47	99,8	369
23	99,4	312	48	99,5	360
24	99,5	313	49	99,7	378
25	99,5	314	50	99,8	371

25 lentelė. Ketvirtojo tyrimo dirbtinių neuroninių tinklų klasifikatoriaus rezultatų duomenys
(deskriptorius – Pikselių vektorius; simbolio dydis – 60 x 40)

Paslėptų sluoksnių sk.	Tikslumas, %	Laikas, ms	Paslėptų sluoksnių sk.	Tikslumas, %	Laikas, ms
1	37,5	268	26	99,5	309
2	55,4	268	27	99,7	309
3	69,4	271	28	99,5	310
4	85,7	278	29	99,5	312
5	97,3	278	30	99,8	313
6	98,4	280	31	99,7	316
7	98,5	281	32	99,7	349
8	98,7	282	33	99,7	355
9	98,8	284	34	99,7	343
10	99,0	285	35	99,8	343
11	99,5	287	36	99,7	355
12	99,6	289	37	99,7	345
13	99,4	289	38	99,7	355
14	99,4	293	39	99,6	355
15	99,5	294	40	99,8	354
16	99,5	294	41	99,7	365
17	99,7	297	42	99,8	362
18	99,3	297	43	99,7	372
19	99,8	299	44	99,8	371
20	99,7	301	45	99,8	371
21	99,6	302	46	99,6	374
22	99,6	302	47	99,7	385
23	99,6	304	48	99,7	386
24	99,5	305	49	99,6	375
25	99,5	307	50	99,6	388

26 lentelė. Ketvirtojo tyrimo k-artimiausių kaimynų klasifikatoriaus rezultatų duomenys (deskriptorius – Pikselių vektorius; simbolio dydis – 15 x 10)

Kaimynų skaičius <i>k</i>	Tikslumas, %	Laikas, ms	Kaimynų skaičius <i>k</i>	Tikslumas, %	Laikas, ms
1	99,6	154	26	99,1	139
2	99,6	143	27	99,2	139
3	99,6	142	28	99,1	139
4	99,4	140	29	99,2	139
5	99,3	140	30	99,2	140
6	99,3	141	31	99,2	139
7	99,4	139	32	99,2	139
8	99,4	139	33	99,2	139
9	99,5	140	34	99,2	139
10	99,2	139	35	99,2	140
11	99,4	139	36	99,2	140
12	99,2	139	37	99,2	140
13	99,2	138	38	99,1	139
14	99,1	139	39	99,2	139
15	99,1	139	40	99,2	139
16	99,1	139	41	99,2	139
17	99,1	139	42	99,2	138
18	99,1	139	43	99,2	139
19	99,1	139	44	99,2	139
20	99,1	139	45	99,2	139
21	99,1	139	46	99,1	139
22	99,1	139	47	99,2	140
23	99,0	140	48	99,2	139
24	99,0	139	49	99,2	139
25	99,0	139	50	99,2	139

27 lentelė. Ketvirtojo tyrimo atraminių vektorių mašinos klasifikatoriaus rezultatų duomenys
(deskriptorius – Pikselių vektorius; simbolio dydis – 30 x 20)

Polinomo laipsnis	Tikslumas, %	Laikas, ms	Polinomo laipsnis	Tikslumas, %	Laikas, ms
1	99,7	3323	3	99,5	2860
2	99,7	3679	4	4,1	3187

28 lentelė. Ketvirtojo tyrimo atraminių vektorių mašinos klasifikatoriaus rezultatų duomenys
(deskriptorius – HOG; simbolio dydis – 60 x 40)

Polinomo laipsnis	Tikslumas, %	Laikas, ms	Polinomo laipsnis	Tikslumas, %	Laikas, ms
1	99,2	3114	3	99,6	3395
2	99,4	3190	4	99,5	3586

29 lentelė. Ketvirtojo tyrimo atraminių vektorių mašinos klasifikatoriaus rezultatų duomenys
(deskriptorius – Pikselių vektorius; simbolio dydis – 60 x 40)

Polinomo laipsnis	Tikslumas, %	Laikas, ms	Polinomo laipsnis	Tikslumas, %	Laikas, ms
1	99,7	4398	3	4,1	3855
2	99,7	4555	4	99,6	4044

30 lentelė. Ketvirtojo tyrimo Naive Bajeso klasifikatoriaus rezultatų duomenys (deskriptorius – Pikselių vektorius; simbolio dydis – 15 x 10)

Nr.	Tikslumas, %	Laikas, ms
1	99,3	235

MATLAB programos kodas

Simbolių išskyrimas

```

clc;
close all;
% Nuotraukos pavadinimas
pavadinimas = '1';
% Slenkstin? riba nuotraukos binariavimui
riba = 130;
% Paskutinio simbolio numeris
simbolioNr = 0;
PilkaNuotrauka = imread(strcat('Cuttet/',pavadinimas, '.jpg'));
% Nuotraukos dydžiu radimas
% spalvos turi būti = 1.
[eilutes, stulpeliai, spalvos] = size(PilkaNuotrauka);
if spalvos > 1
    % Jei spalvos > 1, tai nuotrauka nera pilkos spalvos
    % Konvertuojama nuotrauka i pilka spalva.
    PilkaNuotrauka = PilkaNuotrauka(:, :, 2);
end

% Pilka nuotrauka paverčiama i baltai juoda nuotrauka
BaltaiJuodaNuotrauka = PilkaNuotrauka > riba;
BaltaiJuodaNuotrauka = medfilt2(BaltaiJuodaNuotrauka, [3 3]);
% Teksto eilu?i? radimas vaizde
% Nuotrauka sumuojama horizontaliai
VertikalusVektorius = sum(BaltaiJuodaNuotrauka, 2);
% Iskiriamos linijos su tekstu
dydis = size(BaltaiJuodaNuotrauka);
Riba = 0.99*dydis(2);
EilutesSuTekstu = VertikalusVektorius < Riba;
% Ieskoma virsutiniu ir apatiniu teksto liniju
diff1 = diff(EilutesSuTekstu);
VirsutinesLinijos = find(diff1 == 1);
ApatinesLinijos = find(diff1 == -1);
VirsutineEiluteMem = 0;
for k = 1 : length(VirsutinesLinijos) % Ieskoma teksto eiluciu
    ApatineEilute = ApatinesLinijos(k);

```



```

if VirsutineEiluteMem == 0
    VirsutineEilute = VirsutinesLinijos(k);
else
    VirsutineEilute = VirsutineEiluteMem;
end

EilutesPlotis = ApatineEilute - VirsutineEilute;
if EilutesPlotis < eilutes/3.5
    if VirsutineEiluteMem == 0
        VirsutineEiluteMem = VirsutinesLinijos(k);
    end

    continue
else
    VirsutineEiluteMem = 0;
end

TekstoEilute = BaltaiJuodaNuotrauka(VirsutineEilute:ApatineEilute, :);
% Simboli? radimas eilut?je
% Sumuojama nuotrauka vertikaliam
HorizontalusVektorius = sum(TekstoEilute, 1);
Riba2 = 0.95*abs(VirsutineEilute-ApatineEilute);
SimbolioStulpelis = HorizontalusVektorius < Riba2;
KaireSimbolio = find(diff(SimboliosStulpelis) == 1);
DesineSimbolio = find(diff(SimboliosStulpelis) == -1);
if isempty(KaireSimbolio) || isempty(DesineSimbolio)
    continue;
end

for j = 1 : length(KaireSimbolio) % Ieskoma simboliu
    KairysisStulpelis = KaireSimbolio(j);
    DesinysisStulpelis = DesineSimbolio(j);
    Simbolis = TekstoEilute(:, KairysisStulpelis:DesinysisStulpelis);
    VertikalusVektorius2 = sum(Simbolis, 2);
    dydis2 = size(Simbolis);
    EilutesSuTekstu2 = VertikalusVektorius2 < dydis2(2);
    SimbolioIlgis = abs(DesinysisStulpelis - KairysisStulpelis);
    K = dydis2(1)/12;
    if SimbolioIlgis < K
        continue
    else
        % Ieskoma simboliu apatines ir virsutines linijos
        diff2 = diff(EilutesSuTekstu2);
        VirsutinesLinijos2 = find(diff2 == 1);
        ApatinesLinijos2 = find(diff2 == -1);
        if isempty(VirsutinesLinijos2) & isempty(ApatinesLinijos2)
            Simbolis2 = Simbolis;
        else
            if VertikalusVektorius2(1) == dydis2(2)
                VirsutinesLinijos3 = VirsutinesLinijos2(1);
            else
                VirsutinesLinijos3 = 1;
            end

            if VertikalusVektorius2(end) == dydis2(2)
                ApatinesLinijos3 = ApatinesLinijos2(end);
            else
                ApatinesLinijos3 = dydis2(1);
            end

            Simbolis2 = Simbolis(VirsutinesLinijos3:ApatinesLinijos3, :);
        end

        simbolioNr = simbolioNr + 1;
        % Simbolio nuotraukos išsaugojimas
        kelias = strcat('Photos/', num2str(simbolioNr), '.jpg');
        imwrite(Simbolis2, kelias);
    end
end
end
end

```

Mokymo duomenų bazės kūrimas

```

clc;
close all;
% Matricos duomenis saugoti
Klases = [];
SavybesPikseliai = [];

```

```

SavybesHOG = [];
SavybesLBP = [];
SavybesPCA = [];
KlasesNN = [];
SavybesPikseliaiNN = [];
SavybesHOGNN = [];
SavybesLBPNN = [];
SavybesPCANN = [];
SimboliuSarasas = {'45', '48', '49', '50', '51', '52', '53', '54', '55', '56', '57',
'58', '65', '66', '67', '68', '69', '70', '71', '76', '82', '84'}
% Simbolio matemenys
aukstis = 15;
plotis = 10;
dydis = [num2str(aukstis), 'x', num2str(plotis)];
% Nuotrauk? nuskaitymas
for i = 1 : length(SimboliuSarasas)
    Skyrius = SimboliuSarasas{i}
    Aplankalas = ['DataBase\'', Skyrius];
    D = dir([Aplankalas, '\*.jpg']);
    SimbSarasas = {D.name};
    SimbKiekis = length(SimbSarasas);
    if SimbKiekis > 0
        for j = 1 : SimbKiekis
            Kelias = ['DataBase\'', Skyrius, '\', SimbSarasas{j}];
            Vaizdas = imread(Kelias);
            % Vaizdo binarizavimas
            BaltaiJuodaNuotrauka = Vaizdas > 160;
            % Simbolio matmen? keitimas
            SumazintasSimbolis = imresize(BaltaiJuodaNuotrauka, [aukstis plotis]);
            Klases = [Klases; str2num(Skyrius)];
            SkyriusNN = zeros(22,1);
            SkyriusNN(i,1) = 1;
            KlasesNN = [KlasesNN SkyriusNN];
            SimbDydis = size(SumazintasSimbolis);
            % Simboliai aprašomi deskriptoriais
            KoeficientaiPikseliai = reshape(SumazintasSimbolis,
1, SimbDydis(1)*SimbDydis(2));
            KoeficientaiHOG = extractHOGFeatures(SumazintasSimbolis);
            KoeficientaiLBP = extractLBPFeatures(SumazintasSimbolis);
            MatKoeficientai = pca(double(SumazintasSimbolis));
            KoefDydis = size(MatKoeficientai);
            KoeficientaiPCA = reshape(MatKoeficientai, 1, KoefDydis(1)*KoefDydis(2));
            SavybesPikseliai = [SavybesPikseliai; KoeficientaiPikseliai];
            SavybesHOG = [SavybesHOG; KoeficientaiHOG];
            SavybesLBP = [SavybesLBP; KoeficientaiLBP];
            SavybesPCA = [SavybesPCA; KoeficientaiPCA];
            KoeficientaiPikeseliaiNN = rot90(KoeficientaiPikseliai);
            KoeficientaiHOGNN = rot90(KoeficientaiHOG);
            KoeficientaiLBPNN = rot90(KoeficientaiLBP);
            KoeficientaiPCANN = rot90(KoeficientaiPCA);
            SavybesPikseliaiNN = [SavybesPikseliaiNN KoeficientaiPikeseliaiNN];
            SavybesHOGNN = [SavybesHOGNN KoeficientaiHOGNN];
            SavybesLBPNN = [SavybesLBPNN KoeficientaiLBPNN];
            SavybesPCANN = [SavybesPCANN KoeficientaiPCANN];
        end
    end
end

% Mokymo duomen? baz?s fail? suk?rimas
dlmwrite(strcat('Duomenys/', dydis, '/Klases.txt'), Klases);
dlmwrite(strcat('Duomenys/', dydis, '/Pikseliai.txt'), SavybesPikseliai);
dlmwrite(strcat('Duomenys/', dydis, '/HOG.txt'), SavybesHOG);
dlmwrite(strcat('Duomenys/', dydis, '/LBP.txt'), SavybesLBP);
dlmwrite(strcat('Duomenys/', dydis, '/PCA.txt'), SavybesPCA);
dlmwrite(strcat('Duomenys/', dydis, '/KlasesNN.txt'), KlasesNN);
dlmwrite(strcat('Duomenys/', dydis, '/PikseliaiNN.txt'), SavybesPikseliaiNN);
dlmwrite(strcat('Duomenys/', dydis, '/HOGNN.txt'), SavybesHOGNN);
dlmwrite(strcat('Duomenys/', dydis, '/LBPNN.txt'), SavybesLBPNN);
dlmwrite(strcat('Duomenys/', dydis, '/PCANN.txt'), SavybesPCANN);

```

Teksto identifikavimas

```
clc;
closeall;
% Matricos rezultatams saugoti
NuskaitytiSimboliai = [];
NuskaitytiSimboliaiM = [];
LaikasM = [];
TeisingiSimbM = [];
TikslumasM = [];
KintamasisM = [];
BendrasTikslumasM = [];
BendrasLaikasM = [];
aukstis = 15;
plotis = 10;
SavybiuMetodai = {'Pikseliai', 'HOG', 'LBP', 'PCA'};
SavybiuMetodas = string(SavybiuMetodai(1)); % Savybi? metodo pasirinkimas
Klasifikatoriai = {'KNN', 'SVM', 'DTree', 'NN', 'NBayes'};
Klasifikatorius = string(Klasifikatoriai(4)); % Klasifikatoriaus pasirinkimas
% Parametr? kitimo ribos h
for h = 1:50
    LaikasM = [];
    TikslumasM = [];
    Kintamasis = h;
    A = dlmread('info2.txt');
    [m, n] = size(A);
    aukstisstr = num2str(aukstis);
    plotisstr = num2str(plotis);
    % Nuskaitomi klasifikatori? apmokymo duomenys
    if Klasifikatorius == 'NN'
        Klases = dlmread(char(strcat('Duomenys/', aukstisstr, 'x', plotisstr, '/KlasesNN.txt')));
        Savybes =
    dlmread(char(strcat('Duomenys/', aukstisstr, 'x', plotisstr, '/', SavybiuMetodas, 'NN.txt')));
    else
        Klases = dlmread(char(strcat('Duomenys/', aukstisstr, 'x', plotisstr, '/Klases.txt')));
        Savybes =
    dlmread(char(strcat('Duomenys/', aukstisstr, 'x', plotisstr, '/', SavybiuMetodas, '.txt')));
    end

    % Apmokomi klasifikatoriai
    if Klasifikatorius == 'KNN'
        Modelis = fitcknn(Savybes, Klases, 'NumNeighbors', Kintamasis);
    end

    if Klasifikatorius == 'SVM'
        t = templateSVM('KernelFunction', 'polynomial', 'PolynomialOrder', Kintamasis)
        Modelis = fitcecoc(Savybes, Klases, 'Learners', t);
    end

    if Klasifikatorius == 'DTree'
        Modelis = fitctree(Savybes, Klases, 'MinParentSize', Kintamasis);
    end

    if Klasifikatorius == 'NN'
        Modelis = patternnet(Kintamasis);
        Modelis.divideFcn = 'dividettrain';
        [Modelis, tr] = train(Modelis, Savybes, Klases);
    end

    if Klasifikatorius == 'NBayes'
        Modelis = fitcnb(Savybes, Klases, 'Distribution', 'mn');
    end

    % Nuotrauk? nuskaitymas
    for l = 1:m
        Atsakymas = [];
        NuskaitytiSimboliai = [];
        kelias = A(1, l);
        FotoStr = char(strcat('DataBase/', num2str(kelias), '.jpg'));
        riba = A(1, 2);
        if A(1, 30) == 0
            ilgis = 21;
        else
            ilgis = 30;
        end

        for i=3:ilgis+2
            simbolis = A(1, i);
            Atsakymas = [Atsakymas simbolis];
        end
    end
end
```

```

tic;
PilkaNuotrauka = imread(FotoStr);
% Nuotraukos dydži? radimas
% spalvos turi b?ti = 1.
[eilutes, stulpeliai, spalvos] = size(PilkaNuotrauka);
if spalvos > 1
    % Jei spalvos > 1, tai nuotrauka n?ra pilkos spalvos
    % Konvertuojama nuotrauka ? pilk? spalv?.
    PilkaNuotrauka = PilkaNuotrauka(:, :, 2);
end

% Pilka nuotrauka paver?iama ? baltai juod? nuotrauk?
BaltaiJuodaNuotrauka = PilkaNuotrauka > riba;
BaltaiJuodaNuotrauka = medfilt2(BaltaiJuodaNuotrauka, [3 3]);
% Nuotrauka sumuojama horizontaliai
VertikalusVektorius = sum(BaltaiJuodaNuotrauka, 2);
% I?skiriamos linijos su tekstu
dydis = size(BaltaiJuodaNuotrauka);
Riba = 0.99*dydis(2);
EilutesSuTekstu = VertikalusVektorius < Riba;
% Ieškoma viršutini? ir apatini? teksto linij?
diff1 = diff(EilutesSuTekstu);
VirsutinesLinijos = find(diff1 == 1);
ApatinesLinijos = find(diff1 == -1);
% I?skiriami simboliai iš teksto
for k = 1 : length(VirsutinesLinijos) % Ieškoma teksto eilu?i?
    VirsutineEilute = VirsutinesLinijos(k);
    ApatineEilute = ApatinesLinijos(k);
    TekstoEilute = BaltaiJuodaNuotrauka(VirsutineEilute:ApatineEilute, :);
    % Sumuojama nuotrauka vertikaliai
    HorizontalusVektorius = sum(TekstoEilute, 1);
    Riba2 = 0.95*abs(VirsutineEilute-ApatineEilute);
    SimbolioStulpelis = HorizontalusVektorius < Riba2;
    KaireSimbolio = find(diff(SimbolioStulpelis) == 1);
    DesineSimbolio = find(diff(SimbolioStulpelis) == -1);
    if isempty(KaireSimbolio) || isempty(DesineSimbolio)
        continue;
    end

    for j = 1 : length(KaireSimbolio) % Ieškoma simboli?
        KairysisStulpelis = KaireSimbolio(j);
        DesinysisStulpelis = DesineSimbolio(j);
        Simbolis = TekstoEilute(:, KairysisStulpelis:DesinysisStulpelis);
        VertikalusVektorius2 = sum(Simbolis, 2);
        dydis2 = size(Simbolis);
        EilutesSuTekstu2 = VertikalusVektorius2 < dydis2(2);
        SimbolioIlgis = abs(DesinysisStulpelis - KairysisStulpelis);
        K = dydis2(1)/7;
        if SimbolioIlgis < K
            continue
        else
            % Ieškoma simboli? apatin?s ir virsutin?s linijos
            diff2 = diff(EilutesSuTekstu2);
            VirsutinesLinijos2 = find(diff2 == 1);
            ApatinesLinijos2 = find(diff2 == -1);
            if isempty(VirsutinesLinijos2) & isempty(ApatinesLinijos2)
                Simbolis2 = Simbolis;
            else
                if VertikalusVektorius2(1) == dydis2(2)
                    VirsutinesLinijos3 = VirsutinesLinijos2(1);
                else
                    VirsutinesLinijos3 = 1;
                end

                if VertikalusVektorius2(end) == dydis2(2)
                    ApatinesLinijos3 = ApatinesLinijos2(end);
                else
                    ApatinesLinijos3 = dydis2(1);
                end

                % Rastas simbolis
                Simbolis2 = Simbolis(VirsutinesLinijos3:ApatinesLinijos3, :);
            end

            % Pakei?iamas simbolio dydis
            SumazintasSimbolis = imresize(Simbolis2, [aukstis plotis]);
            % Simbolio požymiai aprašomi deskriptoriais
            if SavybiuMetodas == 'Pikseliai'
                SimbDydis = size(SumazintasSimbolis);
            end
        end
    end
end

```

```

        Koeficientai = double(reshape(SumazintasSimbolis,
1,SimbDydis(1)*SimbDydis(2)));
    end

    if SavybiuMetodas == 'HOG'
        Koeficientai = extractHOGFeatures(SumazintasSimbolis);
    end

    if SavybiuMetodas == 'LBP'
        Koeficientai = extractLBPFeatures(SumazintasSimbolis);
    end

    if SavybiuMetodas == 'PCA'
        MatKoeficientai = pca(double(SumazintasSimbolis));
        KoefDydis = size(MatKoeficientai);
        Koeficientai = reshape(MatKoeficientai, 1,KoefDydis(1)*KoefDydis(2));
    end

    % Atliekamas klasifikavimas
    if Klasifikatorius ~= 'NN'
        Nusksimbolis = predict(Modelis, Koeficientai);
    end

    if Klasifikatorius == 'NN'
        Koeficientai2 = rot90(Koeficientai);
        Isejimai = Modelis(Koeficientai2);
        Nusksimbolis = VisiSimboliai(vec2ind(Isejimai));
    end

    NuskaitytiSimboliai = [NuskaitytiSimboliai char(Nusksimbolis)];
end
end
end

Laikas = toc;
TeisinguSimbSk = sum( Atsakymas == NuskaitytiSimboliai );
Tikslumas = TeisinguSimbSk / length(Atsakymas) * 100;
if ilgis == 21
    for z = 1:9
        NuskaitytiSimboliai = [NuskaitytiSimboliai'0'];
    end
end

LaikasM = [LaikasM; Laikas];
TeisingiSimbM = [TeisingiSimbM; TeisinguSimbSk];
TikslumasM = [TikslumasM; Tikslumas];
KintamasisM = [KintamasisM; Kintamasis];
NuskaitytiSimboliaiM = [NuskaitytiSimboliaiM; NuskaitytiSimboliai];
end

% Randamas vidutinis tikslumas ir laikas bei kaupiami rezultatai duomenys
BendrasTikslumas = mean(TikslumasM);
BendrasLaikas = mean(LaikasM);
BendrasTikslumasM = [BendrasTikslumasM; BendrasTikslumas];
BendrasLaikasM = [BendrasLaikasM; BendrasLaikas];
end

```