

**ŠIAULIŲ UNIVERSITETAS**  
**TECHNOLOGIJOS FAKULTETAS**  
**ELEKTRONIKOS INŽINERIJOS KATEDRA**

Sergėj Liaškėvič

**ŽMONIŲ FIGŪRŲ PAIEŠKA VAIZDUOSE**

Magistro darbas

**Vadovas**

prof. dr. G. Daunys

ŠIAULIAI, 2012

**ŠIAULIŲ UNIVERSITETAS**  
**TECHNOLOGIJOS FAKULTETAS**  
**ELEKTRONIKOS INŽINERIJOS KATEDRA**

**TVIRTINU**  
Katedros vedėjas  
prof. dr. G.Daunys

2012 06 16

**ŽMONIŲ FIGŪRŲ PAIEŠKA VAIZDUOSE**

Elektronikos inžinerijos magistro darbas

**Recenzentas**

doc. dr. N. Ramanauskas

2012 06 11

**Vadovas**

prof. G. Daunys

2012 06 16

**Atliko**

RM0 gr. stud.

S. Liaškėvič

2012 06 16

ŠIAULIAI, 2012

## SANTRAUKA

Liaškėvič S. Žmonių figūrų paieška vaizduose: Elektronikos inžinerijos magistro darbas/mokslinis vadovas prof. G. Daunys; Šiaulių universitetas, Technologijos fakultetas, Elektronikos inžinerijos katedra. – Šiauliai, 2012. – 39 p.

Elektronikos inžinerijos magistro darbo tema yra aktuali įvairiose šiuolaikinėse modernių technologijų vystymosi ir pritaikymo srityse. Kompiuterinė rega siekia kurti algoritmus, kurie vis tobuliau leistų mašinoms atkartoti žmogaus vieną svarbiausių jutimų – regą, gebėjimą apdoroti vaizdinę informaciją, suprasti jos turinį.

Šiuolaikinių atpažinimo modelių metodai yra požymiais grįsti algoritmai. Šiame darbe žmonių atpažinimui vaizde pasirinktas orientuotų gradientų histogramų (HOG) metodas. Išanalizuotas HOG metodo veikimo principas, apžvelgti kiti šio metodo pagrindu veikiančys metodai.

Žmonių atpažinimo eksperimentiniam tyrimui pasirinktas vaizdo integravimo dalimis atpažinimo metodas. Tyrimui atlikti panaudotas Matlab kompiuterinei programai sukurtas algoritmas. Eksperimentų metu nustatyti ir išskirti veikimo algoritmo trūkumai ir netikslumai.

## SUMMARY

Liaškėvič S. Localization of Humans in the Images: Master thesis of electronics engineer / research advisor Prof.Dr. G. Daunys, Šiauliai University, Technological Faculty, Electronics Engineering Department. – Šiauliai, 2012. – 39 p.

The theme of Master project of Electronics engineer is actual in various modern high – tech development and application domains. Computer vision develops algorithms to make machines more advanced to replicate one of the most important human senses – sight, the ability to process visual information, understand its contents.

Modern methods of detection models are based on the features of algorithms. The histogram of oriented gradients (HOG) method performs the identification of people at a picture. We analyzed the principle of operating of HOG method and also took overview of few other operating methods based on HOG.

We made experimental investigation of people detection using discriminatively trained part based model. An algorithm is developed for using in Matlab program to study the method. The performance deficiencies and inaccuracies of the experiment were identified and isolated.

## TURINYS

SANTRAUKA.....	3
TURINYS .....	5
LENTELIŲ SĄRAŠAS .....	6
PAVEIKSLŲ SĄRAŠAS .....	7
ĮVADAS .....	8
1. LITERATŪROS APŽVALGA .....	9
1.1. Orientuotų gradientų histogramų (HOG) metodas pagal DALAL-TRIGGS .....	9
1.2. Orientuotų gradientų histogramų metodas panaudojant kaskadinio atmetimo mechanizmą .....	11
1.2.1. Kintamo dydžio blokai .....	13
1.2.2. Kaskado apmokymas .....	13
1.3. HOG BOOSTING metodas .....	13
1.4. Vaizdo integravimo dalimis metodas .....	16
1.5. SVM įgyvendinimas .....	18
1.6. ADABOOST klasifikatorius.....	23
1.6.1. Tikslų funkcija .....	24
1.6.2. Atnaujinimo schema.....	24
1.7. HAAR ir HOG palyginimas .....	26
2. VAIZDO INTEGRAVIMO DALIMIS METODO TYRIMAS.....	30
3. SLENKSTINĖS VERTĖS, PANAUDOTOS VAIZDO INTEGRAVIMO DALIMIS METODO TYRIME, NUSTATYMAS.....	33
IŠVADOS .....	37
LITERATŪRA .....	39

**LENTELIŲ SĄRAŠAS**

Lentelė 1 Vieno lygio bandymo duomenys .....	27
Lentelė 2 Daugiapakopio bandymo duomenys.....	28
Lentelė 3.1 Objekto Nr.1 aptikimo duomenys iš 3.1 (a) paveikslo .....	34
Lentelė 3.2 Objekto Nr.2 aptikimo duomenys iš 3.1 (a) paveikslo .....	34
Lentelė 3.3 Objekto Nr.1 aptikimo duomenys iš 3.1 (b) paveikslo .....	35
Lentelė 3.4 Objekto Nr.2 aptikimo duomenys iš 3.1 (b) paveikslo .....	35
Lentelė 3.5 Objekto Nr.1 aptikimo duomenys iš 3.1 (c) paveikslo .....	36
Lentelė 3.6 Objekto Nr.2 aptikimo duomenys iš 3.1 (c) paveikslo .....	36

## PAVEIKSLŲ SĄRAŠAS

1.1 pav. Požymių išgavimo grandinė.....	9
1.2 pav. Vaizdo gradientų histogramų skaičiavimas.....	10
1.3 pav. Gradiento histogramos priklausomai nuo krypčių skaičiaus.....	10
1.4 pav. Kaskadinis atmetimo metodas.....	12
1.5 pav. Gradientų vaizdo kitimas priklausomai nuo bloko dydžio.....	14
1.6 pav. Požymių išrinkimas.....	14
1.7 pav. Modelio tikslumo kreivė.....	16
1.8 pav. Žvaigždinė struktūra.....	17
1.9 pav. Tiesinė SVM.....	19
1.10 pav. Dvi kategorijos, išskirtos hypererdve.....	20
1.11 pav. Daug alternatyvių hypererdvių.....	20
1.13 pav. Hypererdvės plotis ir atraminiai vektoriai.....	21
1.14 pav. Klasių išskyrimo palengvinimas.....	22
1.15 pav. Nepašalinamas triukšmas.....	22
1.16 pav. AdaBoost.....	23
1.17 pav. Vieno lygio bandymo rezultatas.....	28
1.18 pav. Atpažinimo kaskados būdu rezultatai.....	29
2.1 pav. Tanki minia.....	31
2.2 pav. Išplaukęs vaizdas.....	31
2.3 pav. Gulintys žmonės.....	31
2.4 pav. .Neišskirti žmonės.....	31
2.5 pav. Ne visas žmogus kadre.....	31
2.6 pav. Krentantis žmogus.....	32
2.7 pav. Žmogus šalia stulpo.....	32
2.8 pav. Atpažinimas su slenkstine verte.....	32
3.1 pav. Tyrimui naudotos nuotraukos.....	33
3.2 pav. Paveiksluota iš vidutinio nuotolio.....	33
3.3 pav. Fotografuota iš didžiausio nuotolio.....	34
3.4 pav. Paveiksluota iš arti.....	35

## IVADAS

Objektų atpažinimas ir sekimas – tai šiandien labai aktuali tema daugelyje žmonių veiklos sričių:

- Automobilių pramonėje, ypač dabar, populiarėjant inovatoriškoms idėjoms apie transporto priemonių vairavimą be vairuotojo. Keliant saugumo laipsnį, panaudojant kompiuterinę regą kaip aktyvų vairuotojo asistentą.
- Įvairių objektų atpažinime ir, jei reikia, sekime. Tai žmonių srautų stebėjimas, apsaugos sistemos, teisėsaugos institucijų poreikiai.
- Žmogaus būsenos stebėjimui – įvairios gestais valdomos komandos, veido mimikos pokyčiai, kūno laikysena ir būseną.

Kompiuterinė rega siekia kurti algoritmus, kurie vis tobuliau leistų mašinoms atkartoti žmogaus vieną svarbiausių jutimų – regą, gebėjimą apdoroti vaizdinę informaciją, suprasti jos turinį, išskirti, atpažinti bei sekti dominantį objektą ar reiškinių vaizde.

Dabartinių atpažinimo modelių metodai yra požymiais grįsti algoritmai, kurių pagalba sistemos atlieka įvairias operacijas tikslui pasiekti:

- Detektavimas – vaizde esančių elementų priskyrimas kokiam tai kategorijai.
- Atpažinimas – gebėjimas atpažinti objektą iš turimų duomenų apie jį.
- Kategorijų atskyrimas.
- Žiūrėjimo taškas erdvėje.
- Judesio atpažinimas (pokyčio nustatymas).

Darbo tikslas – ištirti kompiuterinės regos galimybes atpažinti žmogų vaizduose, panaudojant orientuotų gradientų histogramų (HOG) metodą.

Darbui keliami uždaviniai:

- Atlikti HOG pagrindu sukurtų ir naudojamų vaizdų atpažinimo metodų analizę.
- Pasirinkti tinkamas priemones ir pagal vieną iš metodų praktiškai atlikti žmonių atpažinimą vaizduose.
- Susisteminti gautus rezultatus, panagrinėti galimų klaidų, kaip neatpažinimo ar klaidingo atpažinimo, atsiradimo priežastis. Padaryti išvadas.



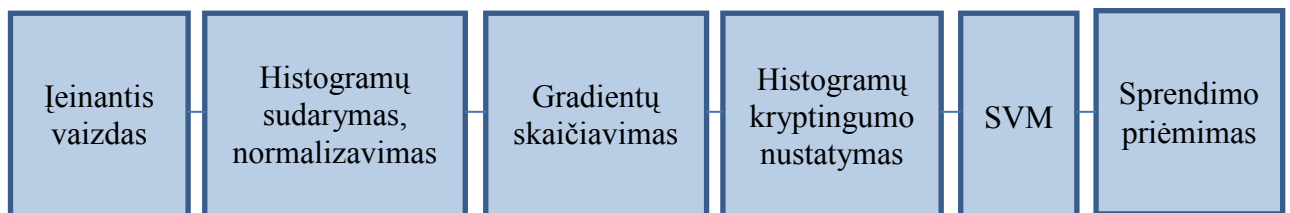
## 1. Literatūros apžvalga

### 1.1. Orientuotų gradientų histogramų (HOG) metodas pagal Dalal-Triggs

Metodą sugalvojo Novneet Dalal ir Bill Triggs [1]. Savo darbe autoriai nagrinėjo, ieškojo gero (tikslaus) objekto atpažinimo metodo testavimui pasitelkiant tiesinę atraminių vektorių mašiną (linear SVM). Autoriai peržiūrėjo ir išnagrinėjo jau esančius kraštų (edges) ir gradientų pagrindu sukurtus deskriptorius ir priėjo išvados, jog jų pasiūlytas orientuotų gradientų histogramų deskriptorių tinklas žymiai lenkia esančias požymių aibes žmogaus atpažinimui. Autoriai tyrė kiekvieno žingsnio įtaką galutiniam rezultatui pasiekti – geras gradientų sugrupavimas (vaizdo sudalinimas į celes, blokus), orientacijos krypčių skaičius (orientation binning), kontrasto normalizavimas persidengiančiuose deskriptoriaus blokuose – viskas svarbu geram rezultatui pasiekti.

Žmonių aptikimas vaizde yra sudėtinga užduotis dėl skirtingų formų ir daugybės pozų, kurias gali priimti. Pirmas klausimas – tai labai tikslus ypatybių rinkinys, galintis išskirti žmogų visu ūgiu, esant įvairiam apšvietimo lygiui ir būdai. Lokaliai normalizuoti HOG deskriptoriai puikiai atlieka darbą palyginus su kitais ypatybių rinkiniais. Siūlomas deskriptorius primena kraštais orientuotas histogramas, SIFT metodą, bet skiriasi tuo, kad skaičiuoja tankiame tinklelyje išsidėsčiusių celių požymius ir naudoja vietinį kontrasto normalizavimą, siekiant pagerinti rezultatą. Tiesinis SVM naudojamas dėl paprastumo ir galimybės pagreitinti skaičiavimus. Autoriai sukūrė savo pėsčiųjų duomenų bazę iš 1800 vaizdų, kur žmonės vaizduojami įvairiose pozose ir esant įvairiam vaizdui fone.

The Dalal-Triggs modelis naudoja pastovaus dydžio slenkantį langą, sudarytą iš 16x16 pikselių normalizuotų orientuotų gradientų histogramų (HOG) blokų. Filtras naudojamas visose pozicijose ir įvairiais dydžiais apibrėžtame vaizde. HOG algoritmas duoda išskirtinį atpažinimo tikslumą. Tai lemia maži deskriptoriaus histogramų blokai bei vietinis bloko normalizavimas, kas išryškina jų santykinį veikimą.



1.1 pav. Požymių išgavimo grandinė

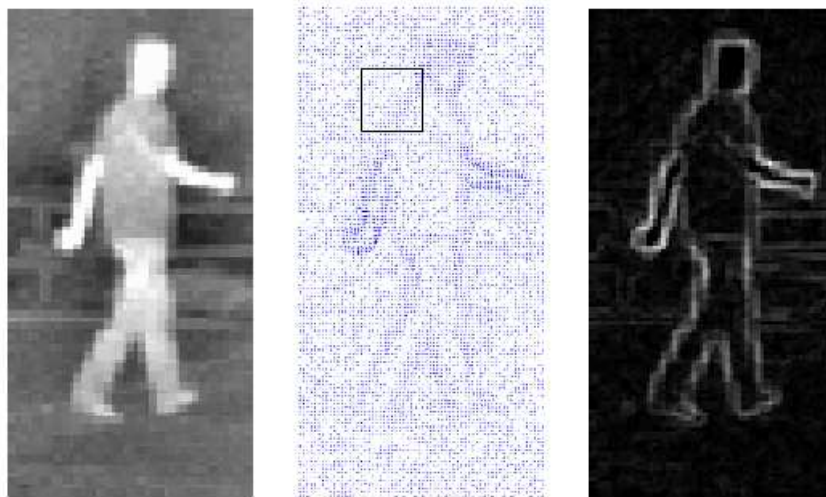
Šio metodo esmė tokia: objekto vieta ir forma gali būti apibūdinama gradientais arba kraštų kryptimis net neturint pradinės informacijos apie gradiento ar krašto pozicijos koordinatės.

Praktikoje tai atliekama sudalinant vaizdo langą į mažus pasiskirsčiusius regionus (celes) ir paskaičiuojant kiekvienai celei vienos dimensijos 1-D HOG kryptį ar kraštų orientaciją pikseliais celėje. Tam, kad sumažinti apšvietimo įtaką, rekomenduojama normalizuoti kontrastą kiekvienoje celėje. Tai atliekama akumuliuojant energijos tankio pasiskirstymą per didesnę blokų skaičių (blokas sudaro kelios celės) ir tuo įverčiu normalizuoja energiją kiekvienoje celėje tame bloke. Taigi tokie normalizuoti deskriptoriai ir laikomi HOG deskriptoriais.

Metodas pradedamas įėjimo signalo gama korekcija. Kai įėjimas (spalvotas vaizdas) turi RGB spalvų paletę, gradientas skaičiuojamas kiekvienai spalvai atskirai ir išskiriamas didžiausias. Detektorius yra jautrus būdui, kuriuo paskaičiuojami gradientai. Skaičiavimams panaudotas horizontalus filtras  $[-1 \ 0 \ 1]$  ir transponuotas  $[-1 \ 0 \ 1]^T$ .

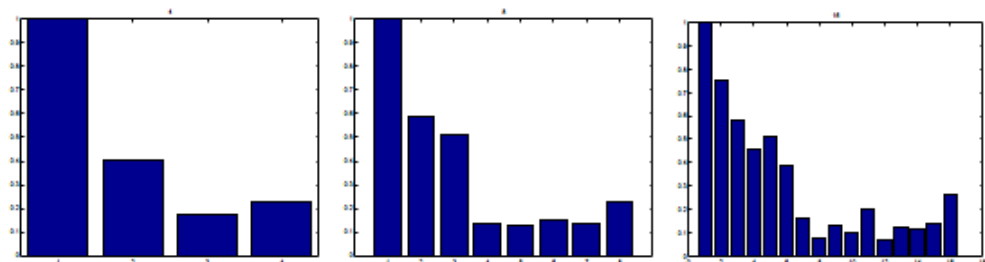
Gautus įverčius galima priskirti erdvinėms kryptims (orientation bins), dažniausiai imant tarp  $0^\circ$  ir  $180^\circ$  (be ženklų) ir tarp  $0^\circ$  ir  $360^\circ$  (su ženklų).

Turint kryptių skaičių skaičiuojamos gradientų histogramos kiekvienai celei priskiriant svorių įverčius atitinkamai kryptiai.



1.2 pav. Vaizdo gradientų histogramų skaičiavimas.

Kairėje – originalus vaizdas, centre – gradientų kryptys, dešinėje – normalizuotas gradientų vaizdas



1.3 pav. Gradientų histogramos priklausomai nuo kryptių skaičiaus.

Kairėje – 4, centre – 8, dešinėje – 16

Paskaičiavus histogramas visoms celėms bloke galima sudaryti deskriptorių, apibūdinamą deskriptoriaus vektoriumi. Imant klasifikatorių kaip filtrą galima paskaičiuoti įvertį (gradiento svorį) kiekvienai celei:

$$f(x) = \beta \times \Phi(x); \quad (1)$$

Čia:  $\beta$  – filtras;

$X$  – vaizdas tam tikroje pozicijoje ir skalėje;

$\Phi(x)$  – požymių vektorius;

Tam, kad sumažinti apšvietimo įtaką, kas gali iškraipyti histogramų turinį, reikia atlikti normalizavimą. Normalizavimas atliekamas grupėje celių, sudarančių bloką, po to grupėje bloką, sudarančių detektoriaus langą. Yra daug normalizavimo schemų, dažniausiai naudojama formulė L2-norm:

$$f = \frac{v}{\sqrt{\|v\|_2^2 + \varepsilon^2}}; \quad (2)$$

Čia:  $v$  – deskriptoriaus vektorius;

$\varepsilon$  – maža reguliavimo konstanta;

Pagrindiniai reikalavimai efektyviam modelio veikimui – tikslumas ir greitaveikiškumas. Viena iš priežasčių, kodėl paprasti modeliai gali veikti praktiškai geriau nei sudėtingi, yra ta, kad pastarųjų modelių apmokymas yra sudėtingas. Dėl šios priežasties HOG metode apmokymui naudojama tiesinė atraminių vektorių mašina (SVM). Tai binarinis klasifikavimo algoritmas, ieškantis sprendimo pateikiamoje duomenų erdvėje.

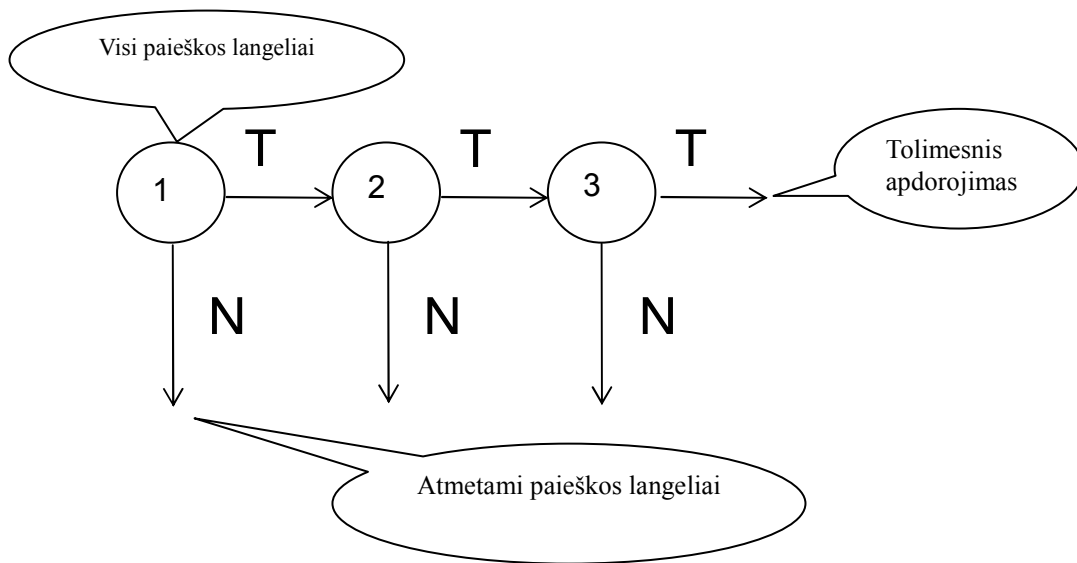
HOG požymių privalumai:

- Gerai išryškina siluetus ir ribas;
- Atsparūs apšvietimo pokyčiams, nežymiems objekto pasislinkimams;
- Skaičiuojami per nustatytą laiką, nepriklausomai nuo dydžio;

## 1.2. Orientuotų gradientų histogramų metodas panaudojant kaskadinio atmetimo mechanizmą

Deja, Dalal ir Triggs pasiūlytas metodas gali apdoroti tik 320x240 matmenų vaizdą viename kadre (frame) per sekundę ir apdoroti 800 aptikimo langų vienoje nuotraukoje. Buvo pasiūlytas

HOG metodo patobulinimas, kuris leido padidinti klasifikatorių (langu) skaičių nuo 800 iki 12800 [2]. Tai pasiekta sujungus kaskadinio atmetimo modelį su HOG požymiais. Autoriai ištyrė ir nustatė, jog fiksuoto dydžio blokai (pagal Dalal-Triggs) nėra pakankamai informatyvūs. Todėl sukūrė kintamo dydžio, kintančios vietos ir skalės didesnę klasifikatorių langu aibę. Su AdaBoost atrenkami labiausiai tinkantys blokai. Tokiu būdu pasiektas labai neblogas rezultatas, kuris beveik prilygo atpažinimui realiu laiku. Taigi vietoj standartinio Dalal-Triggs HOG metodo ir 105 bloku, šis metodas atrankai turi 5031 bloką. Dar daugiau, pirmi keli kaskado atmetimo lygiai, kurie atmeta daugiausiai detektavimo langu, naudoja didelius blokus, o maži blokai naudojami vėlesniuose lygiuose.



1.4 pav. Kaskadinis atmetimo metodas

Šis būdas panašus į Dalal-Triggs HOG metodą, tačiau turi du pagrindinius skirtumus. Pirma, naudoja Gauso funkciją ir trilijinę (tri-linear) interpoliaciją konstruojant HOG kiekvienam blokui. Anksčiau minėtam būdai šie žingsniai netinka. Antra, Dalal ir Triggs naudoja L2 normalizavimą kiekvienam blokui. Vietoj to, čia naudojamas L1- norm būdas, pagal kurį yra greičiau atliekami skaičiavimai:

$$f = \frac{v}{(\|v\|_1 + \varepsilon)}; \quad (3)$$

Čia:  $v$  – deskriptoriaus vektorius;  
 $\varepsilon$  – maža reguliavimo konstanta;

### 1.2.1. Kintamo dydžio blokai

Dalal Triggs naudojo pastovaus dydžio langą, 105 bloką, 16x16 pikselių. Kaskadiniame atmetimo būde naudojamas 64x128 dydžio detektoriaus langas, blokai gali kisti nuo 12x12 iki 64x128 pikselių dydžio. Santykis tarp bloko pločio ir ilgio gali būti 1:1, 1:2, 2:1. Parinktas mažas žingsnis (4,6,8 pikseliai) priklausomai nuo bloko dydžio. Galiausiai 5031 blokas yra suvedami į 64x128 dydžio detektoriaus langą, kur kiekvieną sudaro trisdešimt šešių dimensijų (36 - D) histogramų sąryšio vektorius, surišantis 9 kryptis 2x2 sub-regionuose. Šių blokų naudojimas turi pranašumą. Originalus fiksuoto dydžio 105 blokų detektorius atkoduoja labai mažai informacijos, taigi patobulinto metodo privalumas, jog daugiausiai informacijos turintys blokai atrenkami AdaBoost algoritmu. Naudojami HOG požymiai yra nekintantys ir atsparūs nedideliems pokyčiams, ir prie viso to kintamo dydžio blokai gali aprėpti globalias sritis.

### 1.2.2. Kaskado apmokymas

Pristatomame modelyje kiekvienas požymis yra trisdešimt šešių dimensijų (36 – D) vektorius, kuris naudojamas aprašyti bloką. Silpni klasifikatoriai atskirai skaičiuojami naudojant tiesinę SVM. Galiausiai generuojant (vykdant) kiekvieną iš galimų 5031 blokų kiekviename lygyje reikia didelių laiko sąnaudų. Todėl naudojamas atrankos metodas. Pagal jį imama tik 250 blokų per ciklą. Kiekvienam kaskado lygiui konstruojamas stiprus klasifikatorius, susidedantis iš kelių silpnesnių (tiesinių SVM). Kiekviename lygmenyje vis pridedama po silpnesnį klasifikatorių, kol pasiekama reikiama kokybė.

### 1.3. HOG Boosting metodas

Metodas pagrįstas sustiprintais (patobulintais) HOG požymiais. Šie požymių deskriptoriai paskaičiuoti remiantis apskaičiuotomis kvadratų blokų histogramomis. Histogramų skaičiavimas atliekamas keturis kartus greičiau nei HOG integravimo metodas [2]. Naudojant AdaBoost HOG požymius atrinkimui ir SVM kaip silpną klasifikatorių, sukurtas greitas žmogaus klasifikatorius su puikiu aptikimo rodikliu.

Pasiūlytoje struktūroje klasifikavimo apmokymas atliekamas naudojant Adaboost. Pritaikymas (boosting) pagrįstas stiprių klasifikatorių, sukonstruotu iš kelių silpnesnių klasifikatorių tiesinės kombinacijos. Iš vaizdų išskiriami ir detektoriumi atrenkami geriausi žmogaus HOG požymiai.

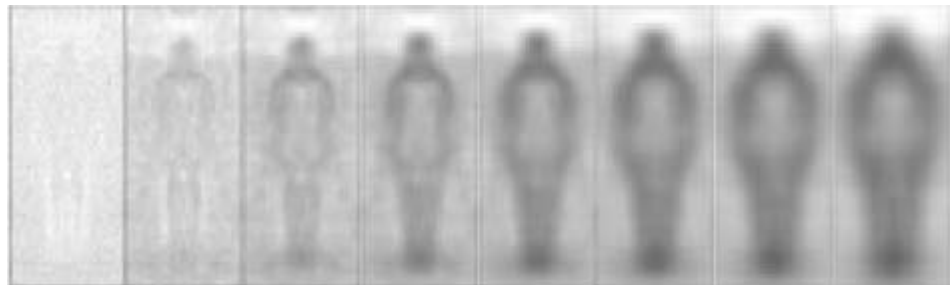
Pasiūlytas modifikuotas HOG būdas, paremtas kvadratinų blokų histogramų rekursyviu skaičiavimu [3]. Tai gali būti iki keturių kartų greičiau nei gradiento integravimas.

Pirmiausia vaizdas verčiamas į juoda-baltą (greyscale) vaizdą. Tada gradiento modelis ir fazė yra paskaičiuojami panaudojant horizontalų ir vertikalų (transponuotą) filtrą  $[-101]$ ,  $[-101]^T$ .

Kitas žingsnis yra klasterizavimas, gradiento orientavimas. Gradiento vaizdas padalintas į devynias kryptis (binus), kiekvienas atitinka orientaciją nuo  $0^0$  iki  $180^0$ .

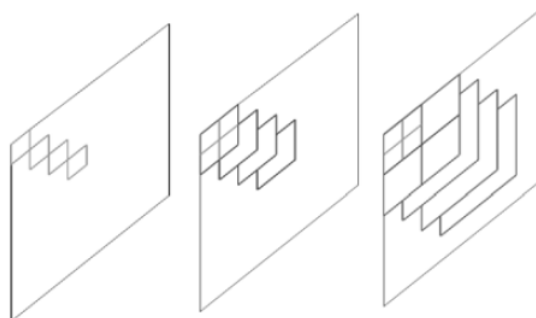
Kiekvienas pikselis patalpinamas į atitinkamą krypties vienetą. Dabar paskaičiuojamos kvadratinų blokų histogramos. Šie blokai yra pagrindiniai komponentai tolimesniam HOG konstravimui.

Žiūrint į 1.5 paveikslą blokų svarba išryškėja nuo keturių pikselių dydžio. Tada  $2^2, 2^3, 2^4, 2^5$  dydžio kvadratiniai blokai skaičiuojami su keturių pikselių žingsniu.



1.5 pav. Gradientų vaizdo kitimas priklausomai nuo bloko dydžio  
 $\{2 \times 2, 4 \times 4, 8 \times 8, 12 \times 12, 16 \times 16, 20 \times 20, 24 \times 24, 32 \times 32\}$

Kaip parodyta 1.6 paveiksle, kiekvienas blokas gali būti apskaičiuotas kaip keturių blokų suma iš ankstesnio lygio arba šešiolikos blokų suma iš dviejų ankstesnių lygių.



1.6 pav. Požymių išrinkimas

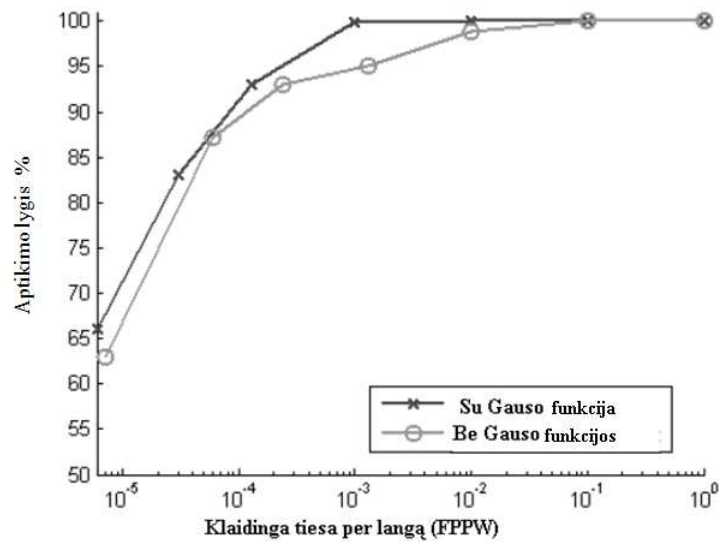
Po tokio išankstinio vaizdo apdorojimo požymio skaičiavimas gali būti atliktas labai greitai nepriklausomai nuo požymio dydžio. Pagrindinis skirtumas tarp šio metodo ir histogramų integravimo tas, kad kiekviena histograma gali būti pasiekta tik su viena prieiga vietoj keturių (nes skaitmuo tiesiai nuskaitomas iš atminties), padarant požymių išgavimą keturis kartus greitesniu.

Klasifikavimo algoritmas paremtas Adaboost metodu. Duota imčių aibė  $X = \{x_1, x_2, \dots, x_n\}$  ir aibė silpnų klasifikatorių  $h_t$ , kurie priskiria kiekvienam elementui  $-1$  arba  $+1$  reikšmę nepriklausomai nuo to ar elementas priklauso klasei, ar ne. AdaBoost sukuria stiprų klasifikatorių  $f(x) = \sum_{t=1}^T \alpha_t h_t(x)$ , kuris yra tiesinė  $h_t$  kombinacija. Priauginimas (Boosting) pasiekiamas keletą kartų keičiant svorius. Pradžioje visi svoriai turi tą patį reikšmingumą (svorį), bet po to, kai silpnas klasifikatorius yra pridodamas, svorių reikšmė yra atnaujinama. Imčių svoriai esant teisingai klasifikacijai išlaiko tą pačią reikšmę, kuomet esant neteisingai klasifikacijai, svorio reikšmė didėja, suteikiant tai imčiai didesnę svarbą sekančioje iteracijoje. Tokiu pačiu būdu, kiekvienos iteracijos metu klasifikacijos koeficiento reikšmė  $\alpha_t$  yra skaičiuojama, priklausomai nuo silpno klasifikatoriaus tinkamumo. Šiuo atveju silpnas klasifikatorius yra SVM, susijęs su atskiru HOG požymiu, rūšiuojamu priklausomai nuo jo klasifikavimo galimybių (pajėgumų). SVM apmokomas visų požymių. Atrenkami geriausias klasifikavimo savybes turintys SVM. Šiuo atveju, AdaBoost atnaujinama reikšmingumo svorius kiekvienai imčiai ir suskaičiuoja  $\alpha_t$  koeficientą. Tada vėl iš naujo apmokomas SVM su jau skirtingomis svorių reikšmėmis. Ciklas tęsiasi tol, kol tam tikras atpažinimo lygis yra pasiekiamas arba kol stiprus klasifikatorius nepasiekia tam tikro sudėtingumo.

Buvo ištestuotas detektorius su MIT pėsčiųjų duomenų baze. Ši duomenų bazė sudaryta iš 924 vaizdų, kurie yra  $128 \times 64$  pikselių dydžio ir žmogus matomas visu ūgiu. Panaudota 800 teigiamų iš MIT ir 800 neigiamų iš bet kur kitur pavyzdžių. Kiekvienai vaizdų imčiai išskyrė 619 požymių, kurie yra sumos:

- 29 x 13 požymių  $16 \times 16$  pikselių dydžio,
- 25 x 9 požymių  $32 \times 32$  pikselių dydžio,
- 17 x 1 požymių  $64 \times 64$  pikselių dydžio.

Šis pasiūlytas būdas yra tarsi kompromisas tarp Dalal-Triggs HOG sprendimo su 105 blokais, fiksuoto dydžio, lėto skaičiavimo ir Zhu, kur naudojami 5031 blokai, kintamo dydžio ir santykio greitesnis HOG sprendimas. Buvo atlikti keli bandymų modeliai. Vieno iš jų idėja buvo panaudoti Gauso svorių pridėjimo funkciją, kad padidinti skaičiavimo greitį, siekiant gauti geresnį sprendinį. Panaudojant Gauso funkciją standartinis nuokrypis parenkamas lygus pusei požymio dydžio. Tai sumažina mažų pokyčių įtaką ir padidina klasifikatoriaus stabilumą. Paveiksle (žr. 1.7 pav.) pavaizduotos dvi kreivės, kur viena atspindi modelį su Gauso funkcija, kita be. Abiejų kreivių aptikimo lygis labai panašus, nors su Gauso funkcija vietomis geresnis kokiais 5%. Būtu gal ir ne labai vertas dėmesio pranašumas, jei ne tai, jog aptikimo greitis su Gauso funkcija yra 4 kartus didesnis.



1.7 pav. Objekto atpažinimo tikslumas

Pasiūlytas metodas, apibrėžiantis naują būdą HOG požymiams skaičiuoti, paremtas kvadratinėmis bloku histogramomis ir yra keturis kartus greitesnės nei Zhu sprendimo, paremto histogramų integravimu.

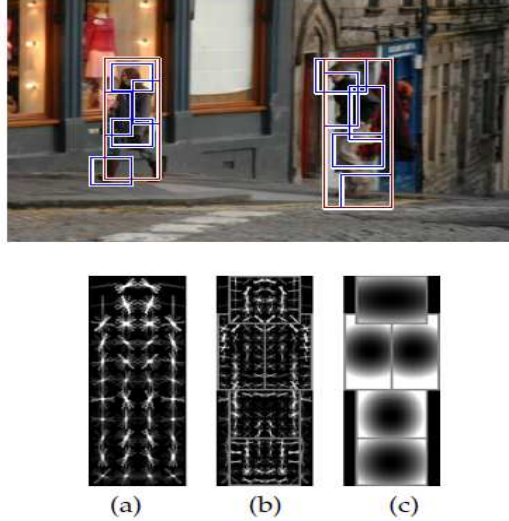
#### 1.4. Vaizdo integravimo dalimis metodas

Šis būdas paremtas vaizdinės struktūros rėmeliu. Vaizdinė struktūra nusako objektą kaip kolekciją dalių, organizuotų kaip deformuojama konfigūracija. Kiekviena dalis saugo savyje duomenis apie užfiksuotą objektą jame, o ryšys tarp tų dalių yra kaip spinduliai, stygos. Turint sudėtingus duomenų masyvus (on difficult datasets) deformuojamų dalių modeliai dažnai atliekami pasinaudojant paprastesniais modeliais, tokiais kaip HOG.

Tuo tarpu, kai deformuojami modeliai gali sukaupti daugybę variacijų informacijos, pavienis deformuojamas modelis yra ne pakankamai išraiškingas, kad apibūdintų turtingą objekto kategoriją. Aprašomas maišytas modelis gali spręsti sudėtingų vaizdų kategorijų išskyrimo problemas.

Buvo pasiūlyta idėja, kaip praturtinti HOG metodą, panaudojant žvaigždinės struktūros, dalimis pagrįstą (part-based) modelį, apibrėžiant šakninį filtrą (root) (tai būtų Dalal-Triggs HOG filtras) plus aibę dalinių (pagalbinių) filtrų, asocijuotų deformacijos modelių (deformation models) [4]. Žvaigždinio modelio įvertis tam tikroje pozicijoje ir tam tikroje skalėje būtų dydis, susidedantis iš šakninio filtro duotame taške plus suma iš aplink esančių maksimumų bei minus deformacijos įvertis, apibūdinamas nukrypimu nuo idalios krypties link šaknies. Paveiksle (žr. 1.8 pav.) pavaizduota žvaigždinė struktūra, kur a – šakninis filtras; b – daliniai filtrai; c – dalinių sąryšio su šakniniu atvaizdavimas.





1.8 pav. Žvaigždinė struktūra

Nagrinėjamame modelyje daliniai filtrai renka požymius būdami dvigubai tankesnėje erdvėje (rezoliucijoje) negu šakninės, todėl modelis gaunasi dvigubai tankesniame tinklelyje. Turime modelį, kur šakninis filtras aptinka ir išskiria objektą, o pagalbiniai daliniai filtrai išskiria objekto dalis. Modelių treniravimui naudojamas latent SVM (LSVM) – paslėptų atraminių vektorių metodas. Jo išraiška tokia

$$f_{\beta}(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z) \quad (4)$$

Čia:  $\beta$  – modelio parametrų vektorius (filtras);

$Z$  – paslėpti duomenys (neatvaizduojami);

$\Phi(x, z)$  – požymių vektorius;

$\beta$ -tai sąryšys tarp šakninio, dalinių filtrų ir deformacijos,  $z$  – objekto konfigūracijos specifikacija,  $\Phi(x, z)$  – sąryšis tarp sublangų iš požymių piramidės ir dalinės deformacijos požymių.

**HOG požymiai.** Tegul  $\theta(x, y)$  ir  $r(x, y)$  yra atitinkamai  $(x, y)$  vaizdo pikselio gradiento kryptis ir dydis. Gradientas apskaičiuotas pasinaudojant baigtiniu filtru  $[-1, 0, +1]$  ir jo transponuota forma. Spalvotam vaizdui naudojamas spalvinis kanalas su didžiausia gradiento reikšme apibrėžti  $\theta$  ir  $r$  kiekvienam pikseliui.

Gradiento orientacija kiekviename pikselyje padalinama į vieną iš  $p$  reikšmių, naudojant kontrasto jautrumo (B1) arba nejautrumo (B2) apibrėžimą.

$$B_1(x, y) = \text{round} \left( \frac{p\theta(x, y)}{2\pi} \right) \text{ mod } p \quad (5)$$

$$B_2(x, y) = \text{round} \left( \frac{p\theta(x, y)}{\pi} \right) \text{ mod } p \quad (6)$$

Apibrėžiamas pikselio lygio požymio žemėlapis, kuris apibūdina erdvinę gradiento histogramą kiekviename pikselyje.

Tegul  $b \in \{0, \dots, p-1\}$  ribos virš krypčių vienetų (orientation bins). Požymių vektorius taške  $(x, y)$  yra

$$F(x, y)_b = \begin{cases} r(x, y) & \text{if } b = B(x, y) \\ 0 & \text{kitu atveju} \end{cases} \quad (7)$$

F galima laikyti kaip orientuotų kraštinių žemėlapi su p kryptiniu kanalu. Kiekvienam pikseliui parenkamas kanalas diskretizuojant gradiento kryptingumą (orientaciją). Gradiento dydis yra kraštinės jėgos (edge strength) dydis.

**Erdvinis sukaupimas.** Tegul F būna pikselių eilės  $w \times h$  dydžio vaizdo požymių žemėlapis. Tegul  $k > 0$  būna parametras, nusakantis kvadrato formos srities perimetro ilgį. Apibrėžiamas stačiakampio formos tankus celių tinklelis ir bendri pikselių eilės požymiai, kad gauti celėmis paremtą (cell-based) požymių žemėlapi C su požymio vektoriais  $C(i, j)$ , kur

$$0 \leq i \leq \left\lfloor \frac{w-1}{k} \right\rfloor \quad (8)$$

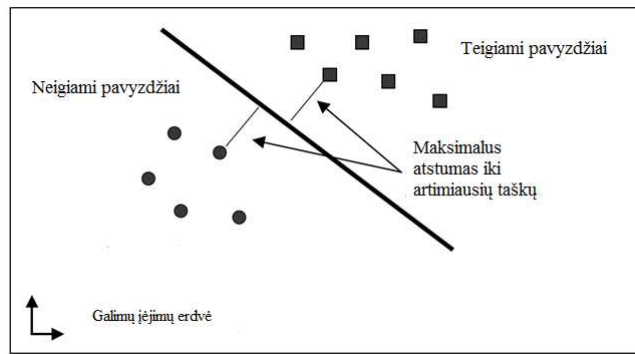
$$0 \leq j \leq \left\lfloor \frac{h-1}{k} \right\rfloor \quad (9)$$

Šis supaprastinimas duoda atsparumą smulkioms deformacijoms ir sumažina požymių žemėlapio dydį. Paprasčiausias būdas supaprastinti požymius tai įpiešti kiekvieną pikselį  $(x, y)$  į celę  $(\lfloor \frac{x}{k} \rfloor, \lfloor \frac{y}{k} \rfloor)$  ir apibrėžti celės požymių vektorius kaip pikselių eilės požymių sumą (arba vidurkis) toje celėje.

Kaip ir daugelyje kitų vaizdų atpažinimo metoduose, taip ir HOG metode viena svarbiausių sudėtinių dalių yra klasifikatoriaus apmokymas. Kad aiškiau suprasti atpažinimo modelio veikimą, smulkiau panagrinėsime kai kuriuos klasifikatorius, jų įgyvendinimo būdus.

## 1.5. SVM įgyvendinimas

SVM – support vector machine – atraminių vektorių metodas. Kaip jau buvo minėta, SVM – tai binarinis klasifikavimo algoritmas, ieškantis sprendimo pateikiamoje duomenų erdvėje.

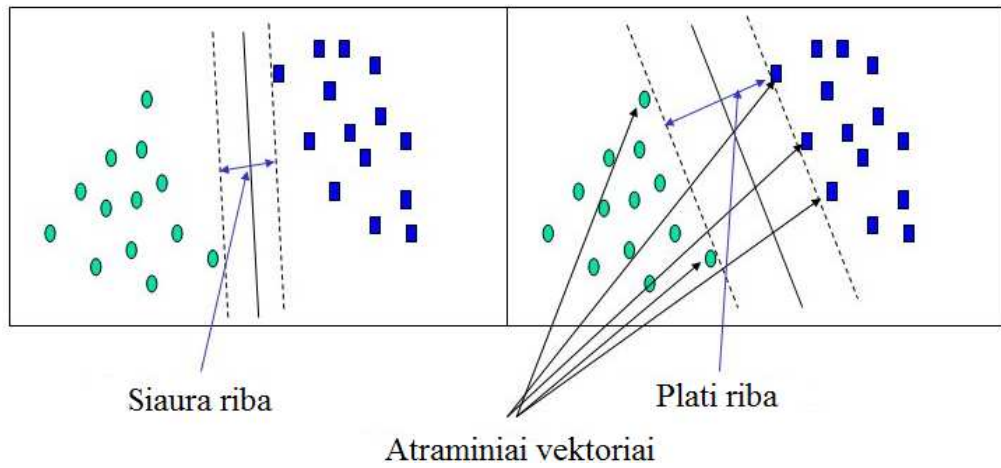


1.9 pav. Tiesinė SVM

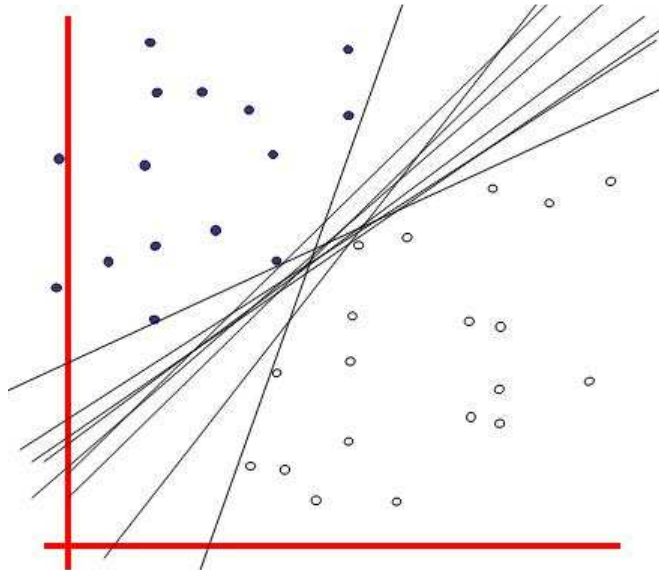
SVM išrado ir pirmą kartą aprašė Vladimiras Vapnikas 1979 metais. Tiesinėje formoje SVM yra hypererdvė, atskirianti teigiamų duomenų aibę nuo neigiamų su maksimaliu ribos plotiu (margin) (žr. 1.10 pav.). Hypererdvės ribos plotis apibrėžiamas atstumu iki artimiausio teigiamo ir artimiausio neigiamo taško.

Prognozuojamas kintamasis vadinamas atributu (attribute), o transformuotas atributas, naudojamas apibrėžti hypererdvei vadinamas požymiu. Labiausiai tinkančių objekto apibūdinančių požymių išrinkimas vadinamas požymių atranka. Požymių aibė, aprašanti atskirą funkcinį vienetą (pavyzdžiui spėjimų vienetų matricos stulpelis), vadinama vektoriumi. Taigi SVM modeliavimo tikslas yra rasti optimalią hypererdvę, kuri atskirtų vektorių grupes taip, kad vienos kategorijos objektai būtų vienoje hypererdvės pusėje, o kitos kitoje. Artimiausi hypererdvės vektoriai vadinami atraminiais vektoriais [7].

Kad būtų aiškiau, imame paprastą pavyzdį: turime didelę aibę elementų, priklausančių dviem skirtingoms klasėms dviejų dimensijų erdvėje. Vienos kategorijos elementai yra kairiajame apatiniame kampe, kitos kategorijos elementai – dešiniajame viršutiniame kampe (žr. 1.9 pav.). Yra be galo daug galimų plokštumų, skiriančių dvi kategorijas (žr. 1.11 pav.). Iškyla pagrindinis klausimas: kuri geriausia, kaip apibrėžti optimaliausia?



1.10 pav. Dvi kategorijos, išskirtos hypererdve



1.11 pav. Daug alternatyvių hypererdvių

Plokštuma, kurios ribų plotis didesnis (žr. 1.10 pav. dešinėje) yra pranašesnė prieš kitą.

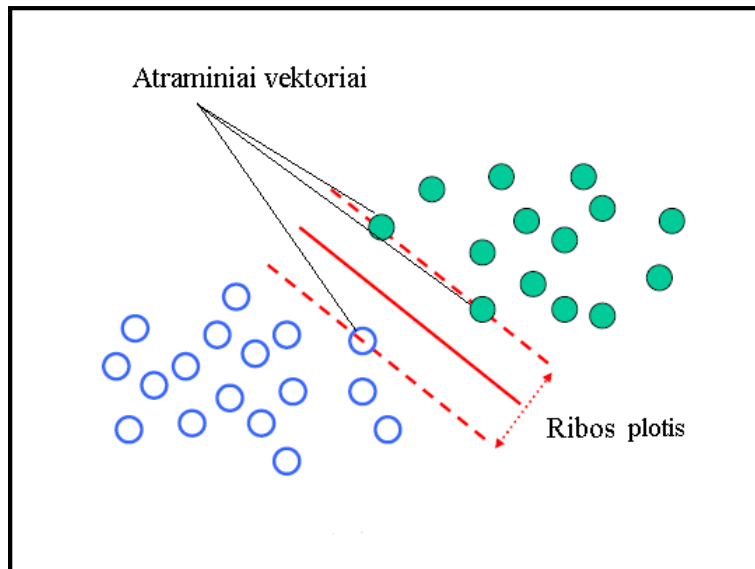
Esant duomenims  $(x_1, y_1), \dots, (x_n, y_n)$  skiriamosios hypererdvės radimui sudaromos sekančios sąlygos

$\forall_i \in (1, \dots, N)$  rasti  $w$  tokią, kad

$$\begin{cases} w^T X_i + b \geq +1, & \text{jei } y_i = +1 \\ w^T X_i + b \leq -1, & \text{jei } y_i = -1 \end{cases} \quad (10)$$

$$\text{Arba} \quad y_i(w^T x_i + b) - 1 \geq 0, \forall_i \quad (11)$$

Sudarome ribą tarp skirtingų klasių elementų (žr. 1.13 pav.)



1.13 pav. Hypererdvės plotis ir atraminiai vektoriai

Tikslas – optimizuoti ribos plotį. Kuo mažesnis svorio vektorius  $w$ , tuo didesnis ribos plotis  $\frac{2}{\|w\|}$ .

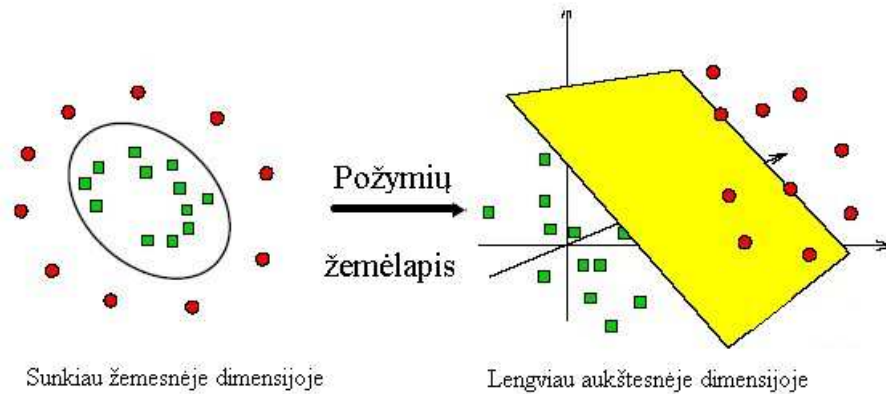
Kuo mums geras didesnis hypererdvės, skiriančios klases, plotis. Jei ir padarytume smulkią (nedidelę) klaidą brėžiant ribą, plotis suteikia galimybę išvengti klaidos. Kitas aspektas būtų išvengti lokalaus minimumo bei pasiekti geresnio klasifikavimo [5].

[6] Jei analizuojamą užduotį sudarytų dviejų kategorijų kintamieji su dviem prognozavimo kintamaisiais, o kategorijų taškai galėtų būti padalinti (išskirti) tiesia linija, tai viskas būtų paprasta. Deja, praktikoje SVM turi susidoroti su:

- Daugiau negu du prognozavimo kintamieji;
- Taškų išskyrimas ne tiesia linija;
- Atvejais, kuomet pavienius klasių taškus neįmanoma arba netikslinga išskirti;
- Klasifikacija, kuomet turima daugiau nei dvi kategorijas;

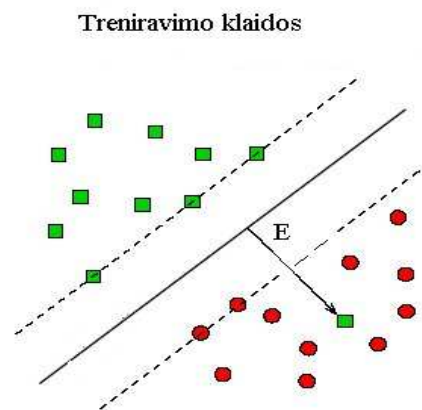
Kuomet tiesios linijos linksta, reikia netiesinių dalijančių linijų. Tokiu atveju naudojama branduolio (kernel) funkcija išdėlioti duomenis į skirtingas erdves, kur hypererdvė gali duomenis išskirti. Kitaip tariant, branduolio funkcija gali transformuoti duomenis į aukštesnės dimensijos erdvę, kad galima būtų juos išskirti (žr. 1.14 pav.).

## Išskyrimas yra lengvesnis aukštesnėje dimensijoje



1.14 pav. Klasių išskyrimo palengvinimas

Tačiau gali būti, kad net ir Kernel SVM negali atlikti klasifikavimo arba gali sudaryti modelį su tokiu dideliu kiekiu požymių vektorius dimensijų, kad nebeatskirs duomenų tipo, pertekliniai duomenys tiesiog tampa triukšmu (over fitting). Kad palengvinti atskyrimą, suteikti klasifikatoriui lankstumo, SVM turi parametą  $E$  (žr. 1.15 pav.), kuris kontroliuoja kompromisą tarp kelių praleistų treniravimo klaidų, kurios dažniausiai atsiranda dėl triukšmo ir tikslios ribos išlaikymo. Tai sukuria minkštą atskyrimo erdvę (soft margin), kuri praleidžia nedidelį nesutapimą.



1.15 pav. Nepašalinamas triukšmas

Minimizavimo lygtis:

$$\frac{\min}{w} \frac{1}{2} \|w\|^2 + C \sum_i \max(0, 1 - y_i(w^T x_i + b)) \quad (12)$$

Kiekvienas  $x_i$  taškas gali papulti į tris kategorijas:

- Yra už ribos  $y_i(w^T x_i + b) > 1$ , ir neturi įtakos

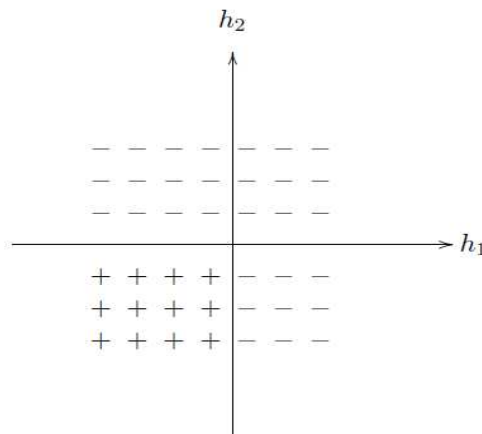
- Gali būti ant ribos  $y_i(w^T x_i + b) = 1$ , nedaro nuostolio, bet dalyvauja kaip atraminis vektorius.
- Gali būti tarp ribų, tuomet jo įvertis pakeičiamas į tinkamą, taškas ištraukiamas į užribį

### 1.6. AdaBoost klasifikatorius

AdaBoost – tai dviejų klasių (reikšmių) klasifikatorius: +1 (taip), -1 (ne). Modelio principas – stiprus klasifikatorius, sudarytas iš kelių silpnų klasifikatorių.

Kaip ir SVM AdaBoost veikia sujungdamas kelių elementų balsus. Tik vietoj atraminių vektorių AdaBoost naudoja silpnąsias imtis.

Paveiksle pavaizduota (žr. 1.16 pav.), kad nei  $h_1$  nei  $h_2$  neduoda tobulo elemento, bet AdaBoost kombinuoja su jais, kad pasiekti „gerą“ kombinaciją.



1.16 pav. AdaBoost

Kitas žingsnis, duomenys yra persveriami, kad padidinti klasifikuotų imčių svarbą. Šis procesas tęsiasi, kiekviename žingsnyje (proceso periodas) silpniausio svorio elementas yra išmetamas.

Algoritmas atrodo sekančiai:

1. Nustatomi lygūs visų imčių svoriai, randamas maksimalus  $h_1$

$$\sum_i y_i h(x_i) \quad (13)$$

2. Atliekamas klasifikuotų imčių persverimas
3. Randamas kitas  $h$  maksimumas  $\sum_i y_i h(x_i)$ , randamas šio klasifikatoriaus svoris  $\alpha$

4. Grįžtama prie antro punkto

Galutinis klasifikatorius atrodo sekančiai:

$$\text{sgn}(\sum_{t=1}^T \alpha_t h_t(x)) \quad (14)$$

Čia:  $\alpha$  – klasifikatoriaus svoris;

$h$  – imties elementas;

### 1.6.1. Tikslo funkcija

Norint sugebėti rasti svorį, atsakantį į kiekvieną klasifikatorių, reikia suformuluoti tikslo funkciją ir rasti  $\alpha$ , kad ją minimizuoti.

Siekama minimizavimo funkcija

$$\sum_i 1_{y_i \neq \text{sgn}(\sum_{t=1}^T \alpha_t h_t(x_i))} \quad (15)$$

Ši funkcija netiesinė ir sunkiai minimizuojama, todėl naudojama sekanti tiesinė funkcija, kuri apibrėžia viršutinę hipererdvės ribą:

$$\text{Obj} = \sum_{i=1}^n e^{-y_i (\sum_{t=1}^T \alpha_t h_t(x_i))} \quad (16)$$

Kita šios funkcijos išraiška:

$$y_i = \text{sgn}(\sum_{t=1}^T \alpha_t h_t(x_i)) \quad (17)$$

- Jei  $\text{sgn}(y_i) = \text{sgn}(\sum_{t=1}^T \alpha_t h_t(x_i))$ , tai  $y_i (\sum_{t=1}^T \alpha_t h_t(x_i))$  yra didelis teigiamas, todėl Obj yra mažas.

- Jei  $\text{sgn}(y_i) \neq \text{sgn}(\sum_{t=1}^T \alpha_t h_t(x_i))$ , tai  $y_i (\sum_{t=1}^T \alpha_t h_t(x_i))$  yra didelis neigiamas, todėl Obj yra didelis.

Šiai funkcijai reikia platesnės atskyrimo hipererdvės pločio, todėl geriau treniruojasi. Ji vienodai ir visur lygi ir diferencijuojasi.

### 1.6.2. Atnaujinimo schema

Aprašomas metodas, kaip rasti  $\alpha_t$  kiekvienoje algoritmo pakopoje:



- (Esami balsai):  $\text{Senas}(x) = \sum_t \alpha_t h_t(x)$
- (Esamas objektas):  $\sum_i e^{-y_i \sum_t \alpha_t h_t(x)}$

Pridedame naują klasifikatorių  $h_n(x)$  su nauju svoriu  $\alpha_n$ :

- (Nauji balsai):  $\text{Senas}(x) + \alpha_n \times h_n(x)$
- (Naujas objektas):  $\sum_i e^{-y_i(\text{old}(x) + \alpha_n h_n(x))}$

Santykis  $\frac{\text{senas}}{\text{naujas}}$  objekto

$$\frac{\text{senas objektas}}{\text{naujas objektas}} = \sum_{i=1}^p w_i e^{-y_i \alpha_n h_n(x_i)} \quad (18)$$

$$\text{Čia: } w_i = \frac{e^{-y_i \text{old}(x_i)}}{\sum_i e^{-y_i \text{old}(x_i)}} = \frac{e^{-y_i \text{old}(x_i)}}{\text{sum}}$$

Taip pat galima pastebėti:

$$\text{sum}_i w_i = \frac{\text{sum}}{\text{sum}} = 1$$

$W_i$  yra svoris, kuris gali būti priskirtas kiekvienam naujam elementui kiekvienoje imtyje. Tai tenkina imčių svorių reikalavimus:

- Jei  $y_i = \text{sgn}(\text{old}(x_i))$ , tai svoris yra mažas
- Jei  $y_i \neq \text{sgn}(\text{old}(x_i))$ , tai svoris yra didelis

Kad gauti naujo elemento svorį  $\alpha_n$ , priimama, kad esamas objektas yra fiksuotas (nekintantis). Domina rasti tokį  $\alpha_n$ , kad su fiksuotu  $h_n$  objekto funkcija yra minimizuota. Santykis  $\frac{\text{senas}}{\text{naujas}}$  objekto gali būti užrašytas sekančiai:

$$\frac{\text{senas objektas}}{\text{naujas objektas}} = \sum_{i: y_i = h_n(x)} w_i e^{-\alpha_n} + \sum_{i: y_i \neq h_n(x)} w_i e^{\alpha_n}, \quad (19)$$

$$\text{Čia: } 1 - \epsilon = \sum_{i: y_i = h_n(x)} w_i;$$

$$\epsilon = \sum_{i: y_i \neq h_n(x)} w_i \text{ yra klaidos lygis (error rate) ant persvertų imčių.}$$

Todėl:

$$\frac{\text{senas objektas}}{\text{naujas objektas}} = (1 - \epsilon) e^{-\alpha_n} + \epsilon e^{\alpha_n} \quad (20)$$

$$(1 - \epsilon) e^{-\alpha_n} + \epsilon e^{\alpha_n} \geq 2\sqrt{(1 - \epsilon)\epsilon} \quad (21)$$

Sumos minimumas bus, kai:

$$(1-\epsilon)e^{-\alpha_n} = +\epsilon e^{\alpha_n} \quad (22)$$

Todėl optimalus  $\alpha_n$  yra:

$$\alpha_n = \frac{1}{2} \log \frac{1-\epsilon}{\epsilon} \quad (23)$$

### 1.7. Haar ir HOG palyginimas

Yra du šiuo metu plačiausiai objektų atpažinime naudojami deskriptoriai (požymių rinkikliai), tai yra Haar filtras ir orientuotų gradientų histogramų (HOG) metodas [7].

Automobilių, pėsčiųjų atpažinimas gatvėje yra labai aktuali ir perspektyvi tema automobilizmo pramonėje, kuomet visi gamintojai stengiasi įdiegti vis daugiau intelektualių valdymo, stebėjimo, atpažinimo sistemų naujai kuriamuose automobiliuose, kas, be jokios abejonės, prideda žymios vertės ir konkurencingumo milžiniškoje pasaulinėje automobilių pramonėje.

Požymių atsiradimo metodai mokinami objektų klasių treniruojant klasifikatorius, kuomet pateikiami vaizdai ir pasakoma kuriame yra ieškomas objektas, o kuriame ne.

Taigi, naudojant kaskadinio atmetimo metodą, naudojamas AdaBoost klasifikavimo metodas. Šio metodo esmė yra stiprus klasifikatorius, susidedantis iš kelių silpnesnių klasifikatorių. Haar ir HOG naudojami sudaryti požymių vektorius objektui atpažinti. Šių vektorių dydis yra ženkliai didesnis nei pikselių skaičius vaizde. Visų išskirtų požymių panaudojimas yra neadekvatus laiko sąnaudų atžvilgiu norint pasiekti reikiamą tikslumą, o, be to, daugelis šių požymių neturi savyje jokios svarbios informacijos (triukšmas). Naudojamas tam tikras požymių kiekis imtyje.

**Haar silpnas klasifikatorius** apibrėžiamas kaip silpna binarinė funkcija  $g$  (24):

$$g \begin{cases} 1, & \text{kai } p_j f_j < p_j \theta_j \\ 0, & \text{kitu atveju} \end{cases} \quad (24)$$

Čia  $f_j$  - požymio vertė;

$\theta_j$  - požymio slenkstis;

$p_j$  - slenksčio lyginumas (parity);

**HOG silpnas klasifikatorius.** Šiuo atveju skaičiuojamas ne požymio įvertis, o imamas atstumas tarp vaizdo histogramos  $h_j$  ir modelio histogramos  $m_j$ . Modelis paskaičiuojamas kaip vidurkio histograma tarp visų teigiamų vaizdų pavyzdžių. Kiekvienai  $h_j$  vaizdo histogramai turime atitinkamą modelio histogramą  $m_j$ . Objekto modelis sudarytas ir AdaBoost atrinktų  $m_j$ , kurie geriausiai atskirs ieškomą klasę nuo kitos klasės. Silpną klasifikatorių apibrėžiame kaip funkciją  $g$  (25):

$$g \begin{cases} 1, & \text{kai } d(h_j m_j) < \theta_j \\ 0, & \text{kitu atveju} \end{cases} \quad (25)$$

Čia  $d(h_j m_j)$  – Bhattacharyya atstumas tarp histogramų  $h_j$  ir  $m_j$ ;

$\theta_j$  – atstumo požymio slenkstis;

Bandymams buvo panaudota : 1000 teigiamų vaizdų apmokymui, 490 patikrinimui, 81 testavimui ir 250 neigiamų vaizdų.

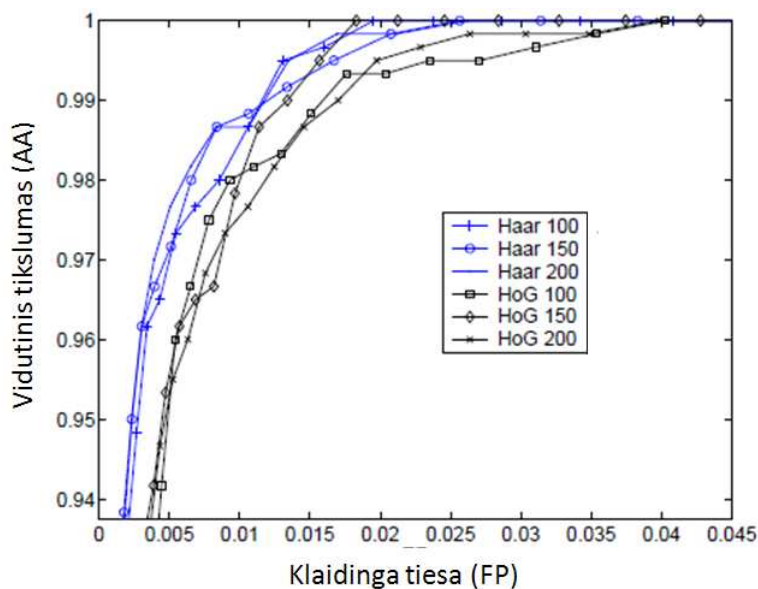
Atliktas vieno lygio bandymas su abejais modelių tipais sudarant stiprų klasifikatorių iš 100, 150 ir 200 Haar ir HOG požymių, panaudojant diskretinį AdaBoost algoritmą. Rezultatų vertinimui panaudotas aptikimo lygis ir klaidinga tiesa (false positive).

Gauti rezultatai pateikiami lentelėje 1 ir paveiksle 1.17 pav.

**Lentelė 1**

**Vieno lygio bandymo duomenys**

<i>Klasifikatorius</i>	<i>Aptikimo lygis (%)</i>	<i>Klaidinga tiesa</i>	<i>Laikas</i>
HOG – 100	69.0	1289	3.52
HOG – 150	72.5	1218	4.20
HOG – 200	83.1	1228	5.02
Haar – 100	96.5	1443	2.61
Haar – 150	95.7	1278	3.93
Haar – 200	95.8	1062	5.25



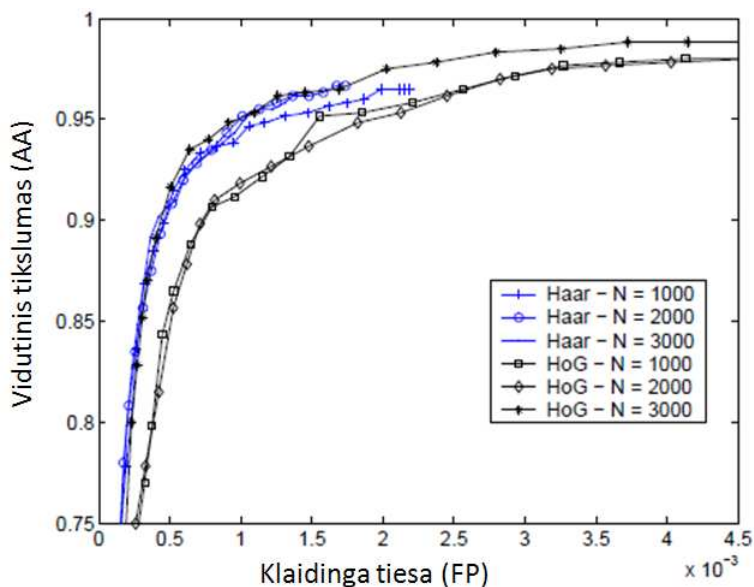
1.17 pav. Vieno lygio bandymo rezultatas

Vėliau bandymą pakartojo panaudojant kaskadinio atmetimo metodą. Pirmo bandymo variantas papildomas keičiamu požymių skaičiumi ir neigiamų pavyzdžių skaičiumi. Gauti duomenys pateikti lentelėje 2 ir paveiksle 1.18 pav.

Lentelė 2

#### Daugiapakopio bandymo duomenys

<i>Klasifikatorius</i>	<i>Kaskados lygiai</i>	<i>Požymių skaičius</i>	<i>Neigiamų pavyzdžių skaičius</i>	<i>Aptikimo lygis (%)</i>	<i>Klaidinga tiesa</i>	<i>Laikas</i>
Haar	12	306	1000	94.3	598	0.75
Haar	12	332	2000	94.0	490	0.71
Haar	12	386	3000	93.5	445	0.59
HOG	12	147	1000	96.5	935	0.51
HOG	12	176	2000	96.1	963	0.59
HOG	11	192	3000	96.6	954	0.55



1.18 pav. Atpažinimo kaskados būdu rezultatai

### Apibendrinant rezultatus.

Apie HOG metodą: didinant požymių skaičių, aptikimo lygis didėja, klaidingos tiesos dydis praktiškai nesikeičia.

Apie Haar metodą: didinant požymių skaičių, klaidingos tiesos dydis mažėja, aptikimo lygis beveik nekinta.

Antru atveju, abiejų modelių elgsena beveik identiška pirmam vieno lygio klasifikavimui, Haar ir HOG metodai pasiekė panašų atpažinimo lygį, tik HOG tam reikėjo perpus mažesnio požymių skaičiaus.

Didinant požymių skaičių, HOG metodo atpažinimo lygis didėja, klaidinga tiesa išlieka tokia pati, kuomet Haar metodui klaidinga tiesa mažėja.

## 2. Vaizdo integravimo dalimis metodo tyrimas

HOG metodo veikimo principo nagrinėjimui buvo pasirinktas vaizdo integravimo dalimis metodas, aprašytas 1.4. skyrelyje. Žmogaus atpažinimui panaudotas metodas aprašomas [4][8]. Atpažinimui panaudotos atsitiktinės nuotraukos, rastos internete.

Panaudotas metodas – tai mišinys iš tokių komponentų:

- HOG požymiai;
- Efektyvūs algoritmai, skirti dalimis integruojamiems objektams atpažinti;
- Diskriminatyvus apmokymas su paslėptais kintamaisiais ( latent SVM);

Užduotis – pateiktose nuotraukose aptikti ir pažymėti žmones. Apdorotas nuotraukas galime suskirstyti į keturias kategorijas:

- Visi objektai atpažinti;
- Tik dalis objektų atpažinta;
- Ieškomi objektai neatpažinti;
- Kiti objektai atpažinti kaip ieškomi (klaidingai atpažinta).

Atpažinimui didelę įtaką turi pačios nuotraukos dydis (skyra pikseliais). Nuotraukose, kurių dydis yra labai didelis, atpažįstami visi žmonės, Jei nuotraukos mažos ir objektai maži, tai dažniausiai objektai neaptinkami. Kitos pastebėtos neatpažinimo priežastys:

- pavaizduota tanki minia, sunku išskirti kontūrus (žr. 2.1 paveiksle);
- išplaukęs vaizdas (žr. 2.2 paveiksle);
- du greta arba vienas už kito stovintys žmonės dažnai atpažįstami kaip vienas (žr. 2.4 paveiksle);
- žmogaus laikysena – jei guli, pasviręs arba nesimato kokios tai kūno dalies – tai dažniausiai neatpažįsta (žr. 2.3; 2.5; 2.6 paveiksluose);
- Pakankamo dydžio, tačiau stovintis prie kokio tai objekto, pvz pakelės stulpo (žr. 2.7 paveiksle);



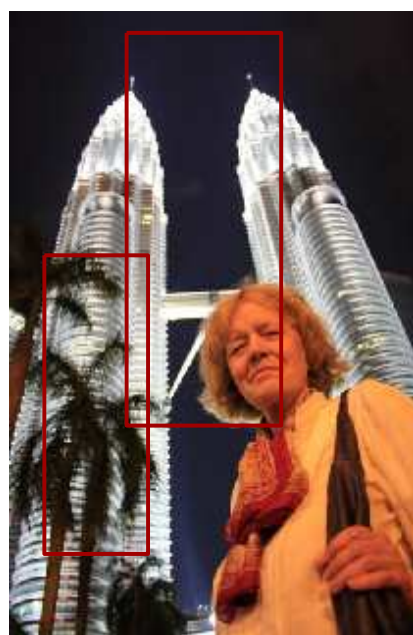
2.1 pav. Tanki minia



2.4 pav. .Neišskirti žmonės



2.2 pav. Išplaukęs vaizdas



2.5 pav. Ne visas žmogus kadre



2.3 pav. Gulintys žmonės



2.6 pav. Krentantis žmogus



2.7 pav. Žmogus šalia stulpo

Kitos neatpažinimo priežastys – klasifikatorius apmokytas tik tam tikro tipo žmonių atvaizdų (tam tikra laikysena), per mažai pozų variacijos pavyzdžių, HOG fiksuotas skaičiavimo laikas bei uždėtas atpažinimo slenkstis ( atpažįstamas žmogus turi būti tam tikro minimalaus dydžio (žr. 2.8 pav.)).



2.8 pav. Atpažinimas su slenkstine verte



### 3. Slenkstinės vertės, panaudotos vaizdo integravimo dalimis metodo tyrime, nustatymas

Kad nustatyti naudojamą slenkstį, nuo kurio vertės objektas atpažįstamas kaip ieškomas, padarėme keletą skaitmeninių nuotraukų (žr. 3.1 pav.), kuriose vaizduojamos dvi moterys, einančios gatve, viena priešais kitą. Ta pati situacija atvaizduota trijose nuotraukose, bet skirtingais stebėjimo kampais. Originalus nuotraukos dydis (skyra) 4500x3000 pikselių. Nuotraukoje artimesnį žmogų pažymėsime Nr. 1, tolimesnį Nr. 2.



a

b

c

3.1 pav. Tyrimui naudotos nuotraukos

Vaizdų atpažinimui naudojome Matlab programinę įrangą, o algoritmo kodą panaudojome iš šaltinio (), kuris taip pat buvo panaudotas ir antrame skyriuje. Prieš apdorojant nuotraukas Matlab programa, jos buvo sumažintos IrfanView programa.

Moterų dydį nuotraukose nustatėme Matlab programos vaizdų redaktoriaus *imtool* pagalba. Duomenis surašėme lentelėse 3.2; 3.3; 3.4; 3.5; 3.6;



3.2 pav. Paveiksluota iš vidutinio nuotolio

## Objekto Nr.1 aptikimo duomenys iš 3.1 (a) paveikslo

Nuotraukos dydis pikseliais	Objekto Nr. 1 dydis pikseliais	Rezultatas
4500x3000	457	Atpažino
2200x1500	232	Atpažino
2000x1350	206	Atpažino
1800x1200	186	Atpažino
1600x1050	161	Atpažino
1350x900	139	Atpažino
1125x750	117	Atpažino
900x600	93	Atpažino
675x450	70	Neatpažino

## Objekto Nr.2 aptikimo duomenys iš 3.1 (a) paveikslo

Nuotraukos dydis pikseliais	Objekto Nr. 2 dydis pikseliais	Rezultatas
4500x3000	237	Atpažino
2200x1500	119	Atpažino
2000x1350	105	Atpažino
1800x1200	94	Atpažino
1600x1050	84	Atpažino
1350x900	73	Neatpažino
1125x750	60	Neatpažino
900x600	47	Neatpažino
675x450	35	Neatpažino



a)

b)

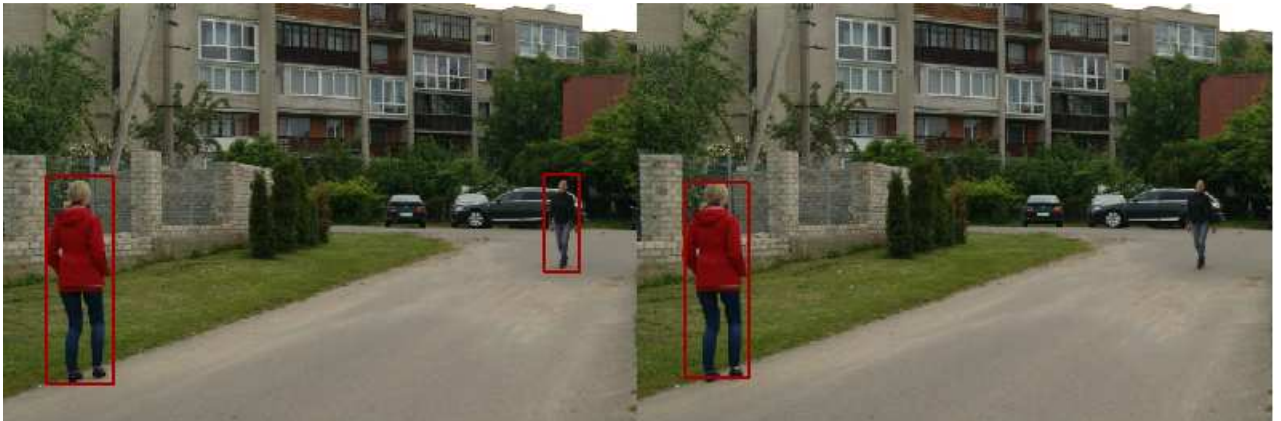
3.3 pav. Fotografuota iš didžiausio nuotolio: a) originalus vaizdas; b) išdidintas fragmentas su žmonių figūromis

**Objekto Nr.1 aptikimo duomenys iš 3.1 (b) paveikslo**

Nuotraukos dydis pikseliais	Objekto Nr. 1 dydis pikseliais	Rezultatas
4500x3000	239	Atpažino
2200x1500	124	Atpažino
2000x1350	111	Atpažino
1800x1200	97	Atpažino
1600x1050	85	Neatpažino
1350x900	74	Neatpažino
1125x750	61	Neatpažino
900x600	49	Neatpažino
675x450	36	Neatpažino

**Objekto Nr.2 aptikimo duomenys iš 3.1 (b) paveikslo**

Nuotraukos dydis pikseliais	Objekto Nr. 2 dydis pikseliais	Rezultatas
4500x3000	175	Atpažino
2200x1500	89	Atpažino
2000x1350	81	Atpažino
1800x1200	72	Atpažino
1600x1050	61	Neatpažino
1350x900	53	Neatpažino
1125x750	43	Neatpažino
900x600	35	Neatpažino
675x450	27	Neatpažino



3.4 pav. Paveiksluota iš arti

Lentelė 3.5

**Objekto Nr.1 aptikimo duomenys iš 3.1 (c) paveikslu**

Nuotraukos dydis pikseliais	Objekto Nr. 1 dydis pikseliais	Rezultatas
4500x3000	707	Atpažino
2200x1500	360	Atpažino
2000x1350	323	Atpažino
1800x1200	289	Atpažino
1600x1050	218	Atpažino
1350x900	218	Atpažino
1125x750	180	Atpažino
900x600	145	Atpažino
675x450	107	Atpažino

Lentelė 3.6

**Objekto Nr.2 aptikimo duomenys iš 3.1 (c) paveikslu**

Nuotraukos dydis pikseliais	Objekto Nr. 2 dydis pikseliais	Rezultatas
4500x3000	296	Atpažino
2200x1500	154	Atpažino
2000x1350	144	Atpažino
1800x1200	120	Atpažino
1600x1050	94	Atpažino
1350x900	91	Atpažino
1125x750	78	Neatpažino
900x600	61	Neatpažino
675x450	47	Neatpažino

Atpažinimo slenkstis iš vidutinio nuotolio paveiksluotoje nuotraukoje (žr. 3.2 pav.) objekto Nr.1 yra 70 pikselių, objekto Nr. 2 yra 73 pikseliai.

Atpažinimo slenkstis iš tolimiausio nuotolio paveiksluotoje nuotraukoje (žr. 3.3 pav.) objekto Nr.1 yra 85 pikseliai, objekto Nr. 2 yra 61 pikselis.

Atpažinimo slenkstis iš mažiausio nuotolio paveiksluotoje nuotraukoje (žr. 3.4 pav.) objektas Nr.1 yra atpažįstamas visų testuojamų dydžių nuotraukose, objekto Nr. 2 yra 78 pikseliai.

Apibendrinant rezultatus, galima pastebėti, kad to paties vaizdo, bet skirtingų žiūrėjimo kampų atpažinimo slenksčiai skiriasi, kaip ir toje pačioje nuotraukoje esančių objektų.

Pažymėtina, kad, apdorojant didelės skyros nuotraukas, sugaištama daugiau laiko nei mažesnių. Todėl, naudojant šį atpažinimo metodą praktikoje, svarbu parinkti optimalų nuotraukos dydį, kad pasiekti kompromisą tarp norimo apdorojimo greičio ir tikslumo.

## IŠVADOS

Darbo tikslas – išanalizuoti HOG metodo panaudojimą žmonių atpažinimui vaizduose. Išanalizavus didelį kiekį literatūros, aprašyti HOG metodo ir jo papildytų būdų veikimo principai.

Dalal – Triggs HOG metodo privalumai: dėl fiksuotų blokų dydžio požymiai skaičiuojami per nustatytą laiką, nepriklausomai nuo dydžio. Dėl histogramų lokalaus normalizavimo metodas atsparus apšvietumo pokyčiams, nežymiems objekto pasislinkimams. Modelio apmokymui naudojama tiesinė SVM, tai supaprastina ir pagreitina skaičiavimus.

Taip pat panagrinėti keli patobulinti HOG metodai – kaskadinio atmetimo metodas ir Boosting HOG metodas.

Kaskadinis atmetimo mechanizmas naudoja AdaBoost klasifikatorių, sudarytą iš kelių silpnų tiesinių SVM klasifikatorių. Histogramoms sudaryti naudojami kintamo dydžio blokai, todėl jų kiekis nuo 105, naudojamų standartiniame metode, padidėjo iki 5031. Tai įgalina sudaryti didesnę požymių variaciją. Histogramų normalizavimui naudojama L1 norm formulė vietoj L2 norm, naudojamos standartiniame HOG metode.

Boosting HOG metodas – tai kompromisas tarp Dalal – Triggs metodo ir kaskadinio atmetimo metodo. Tai greitas metodas. Greitaveikiškumą lemia metodo veikimas, kuomet atliekamas išankstinis vaizdo apdorojimas, klasifikatoriui pateikiami tik naudingos informacijos turintys kvadratiniai blokai. Anot literatūros šaltinių, metodas yra keturis kartus greitesnis nei kaskados, už standartinį HOG metodą dar daugiau.

Tiriamajam darbui surinkom žmonių atvaizdų nuotraukų, rastų internete, duomenų bazę. Matlab programoje pasinaudojus šaltinyje [9] pateiktu programos algoritmu, aprašančiu vaizdo integravimo dalimis metodą, atlikome žmonių atpažinimą. Gautus rezultatus galime suskirstyti į keturias kategorijas:

- Visi objektai atpažinti;
- Tik dalis objektų atpažinta;
- Ieškomi objektai neatpažinti;
- Kiti objektai atpažinti kaip ieškomi (klaidingai atpažinta).

Taip pat buvo atlikta atpažinimo slenkstinės vertės paieška eksperimento būdu.

Nustatyta, jog toje pačioje nuotraukoje esančių skirtingų objektų slenkstinės atpažinimo vertės skiriasi, kaip ir skiriasi tų pačių objektų slenkstinės atpažinimo vertės skirtingose nuotraukose. Apibendrinant rezultatus, galima teigti, jog gautų rezultatų kokybę labiausiai lemia požymių rinkiniai ir efektyvus klasifikatoriaus apmokymas.

## LITERATŪRA

1. Dalal N., Triggs B.. Histograms of oriented gradients for human detection / Computer vision and pattern recognition: tarptautinės konferencijos pranešimų medžiaga. San Diegas, Kalifornija, JAV, 2005. p. 886 – 893.
2. Zhu Q., Yeh M., Cheng K., Avidan S.. Fast human detection using cascade of histograms of oriented gradients / Conference on computer vision and pattern recognition – volume 2 (CVPR'06): tarptautinės konferencijos pranešimų medžiaga. Niujorkas, JAV, 2006. p. 1491 – 1498.
3. Pedersoli M., Gonzales J., Chakraborty B., Villanueva J.. Boosting histograms of oriented gradients for human detection / 2 nd Computer vision: Advances in research and development: tarptautinės konferencijos pranešimų medžiaga. Bellatera, Ispanija, 2007. p. 1 – 6.
4. Felzenszwalb P., Girshick R., McAllester D., Ramanan D.. Object detection with discriminatively trained part based models / IEEE transactions on pattern analysis and machine intelligence, vol.32, No 9 [interaktyvus] 2010, [žiūrėta 2012-05-10]
5. Jakkula V.. Tutorial on support vector machine (SVM) / School of EECS, Washington State University. Prieiga per internetą:  
< <http://www.ccs.neu.edu/course/cs5100f11/resources/jakkaluta.pdf> >
6. SVM – Support vector machines. < <http://www.dtrek.com/svm.htm> >
7. Negri P., Clady X., Prevost L.. Benchmarking Haar and histograms of oriented gradients features applied to vehicle detection. Prieiga per internetą:  
< [http://pablonegri.free.fr/Publications/files/ICINCO2007\\_negri.pdf](http://pablonegri.free.fr/Publications/files/ICINCO2007_negri.pdf) >
8. Discriminatively trained deformable part models. Version 4 (Balandžio 21, 2010). Prieiga per internetą:  
< <http://www.cs.brown.edu/~pff/latent/> >