**KAUNO TECHNOLOGIJOS UNIVERSITETAS**
**INFORMATIKOS FAKULTETAS**

**Mantas Tekorius**

# Objekto suradimo scenoje identifikavimo tikslumo tyrimai, tikslumo gerinimas prie tų sąlygų, kurioje jis žemas

Baigiamasis magistro projektas

**Vadovas**
Prof. dr. Rytis Maskeliūnas

**KAUNAS, 2017**

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**
**INFORMATIKOS FAKULTETAS**

# Objekto suradimo scenoje identifikavimo tikslumo tyrimai, tikslumo gerinimas prie tų sąlygų, kurioje jis žemas

Baigiamasis magistro projektas
**Programų sistemų inžinerija (621E16001)**

**Vadovas**
(parašas) Prof. dr. Rytis Maskeliūnas
(data)
**Recenzentas**
(parašas) Doc. dr. Tomas Blažauskas
(data)
**Projektą atliko**
(parašas) Mantas Tekorius
(data)

**KAUNAS, 2017**

KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

(Fakultetas)

Mantas Tekorius

(Studento vardas, pavardė)

Programų sistemų inžinerija (621E16001)

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto „Objekto suradimo scenoje identifikavimo tikslumo tyrimai, tikslumo gerinimas prie tų sąlygų, kurioje jis žemas“

**AKADEMINIO SĄŽININGUMO DEKLARACIJA**

20 17 m. gegužės 25 d.

Kaunas

Patvirtinu, kad mano, **Manto Tekoriaus,** baigiamasis projektas tema „Objekto suradimo scenoje identifikavimo tikslumo tyrimai, tikslumo gerinimas prie tų sąlygų, kurioje jis žemas“ yra parašytas visiškai savarankiškai ir 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)*

**TURINYS**

**SANTRAUKA**

Tekorius Mantas. Objekto suradimo scenoje identifikavimo tikslumo tyrimai, tikslumo gerinimas prie tų sąlygų, kurioje jis žemas Magistro baigiamasis projektas / vadovas prof. dr. Rytis Maskeliūnas; Kauno technologijos universitetas, Informatikos fakultetas.

Mokslo kryptis ir sritis: vaizdo atpažinimas

Reikšminiai žodžiai: *vaizdo atpažinimas, objektų atpažinimas, trimatis vaizdas, veidų atpažinimas*

Kaunas, 2017.

Technologijos ir įvairūs sprendimai lengvina mūsų kasdienius gyvenimą ir daro jį paprastesnį. Tokie sprendimai kaip vaizdo atpažinimo sistemos ir jų programinė įranga darosi vis labiau naudojamos ir pritaikomos mūsų kasdienėje veikloje. Šių sistemų panaudojimas yra labai platus. Vaizdo atpažinimo technologijas galime sutikti žaidimuose, papildytoje realybėje, apsaugoje, robotikoje, karinėje pramonėje. Išvardintos sistemos jau gyvuoja kurį laiką, tačiau paprasti, žmonėms pritaikyti sprendimai atsirado visai neseniai. Už jų populiarėjimą galima padėkoti išmaniesiems įrenginiams, kuriuos naudoja kiekvienas iš mūsų. Vaizdo atpažinimas paprastam vartotojui naudos galbūt teikia ir nedaug, tačiau tiems žmonėms, kurie turi regos sutrikimų, tokie sprendimai daro didžiulę įtaką.

Šiame darbe buvo išanalizuoti skirtingi vaizdo atpažinimo būdai, metodai ir algoritmai bei pritaikyti realioje sistemoje. Sistema buvo suprojektuota ir pasirinkus OpenCV biblioteką, realizuota Android įrenginyje. Programėlė sugeba kameros pagalba atpažinti žmonės 79% tikslumu ir apie tai pranešti vartotojui. Sistema kadre atpažįsta žmogų vidutiniškai per 438ms. Atliktame tyrime buvo nustatyta, jog sistema tiksliausiai atpažįsta žmonių veidus naudojant HAAR Cascade klasifikatorių. Geriausi rezultatai užfiksuoti naudojant 1,05 didinimo žingsnio reikšmę ir 4 minimalaus kaimyno skaičiaus reikšmę. Tyrimui buvo panaudojamos skirtingų veidų bruožų žmonių nuotraukos, o taip pat sistema išmėginta ir realioje aplinkoje.

**SUMMARY**

Tekorius Mantas. *Evaluation Of Environment Identification Methods Using A Smartphone Camera* Master‘s thesis supervisor assoc. prof. dr. Rytis Maskeliūnas. The Faculty of Informatics, Kaunas University of Technology.

Research area and field: computer vision

Key words: image recognition, object recognition, 3d, face recognition

Kaunas, 2017

There are a lot of technologies and solutions which makes our lives easier and simpler. Solutions and software like image recognition are becoming more and more popular and used in our daily routine. Image recognition technologies are in video games, augmented reality, security, robotics and military industry. Most of the systems are already in production but solutions for ordinary people just recently appeared. For their popularity we should thanks smart phones which made huge impact in our lives.

This paper concentrates on various image recognition methods, algorithms and approaches and use it in working system. System was designed and using OpenCV library a solution for Android device was created. App is capable to detect humans in camera view and report about it to user with 79% accuracy. While doing a research it was found that HAAR Cascade classifier is most accurate for people face recognition. It was found that scale factor value for best results is 1.05 and minimum neighbors value is 4. In research were used different people face photos and also system were used in real life scenario.

**LENTELIŲ SĄRAŠAS**

**PAVEIKSLŲ SĄRAŠAS**

# 1. ĮVADAS

Technologijos ir įvairūs sprendimai lengvina mūsų kasdienį gyvenimą ir daro jį paprastesnį. Tokie sprendimai kaip vaizdo atpažinimo sistemos ir jų programinės įrangos darosi vis labiau naudojamos ir pritaikomos mūsų kasdienėje veikloje. Šių sistemų panaudojimas yra labai platus. Vaizdo atpažinimo technologijas galime sutikti žaidimuose, papildytoje realybėje, apsaugoje, robotikoje, karinėje pramonėje, neurologijos moksle, signalų apdorojime. Vis didėjant ir augant dirbtinio intelekto tobulinimui, vaizdo atpažinimo sistemos tapo neatsiejama šios srities dalis. Nors šiai dienai ir ne taip dažnai tenka susidurti su šiomis technologijomis, tačiau šios jau kurį laiką keičia mūsų gyvenimą. Šiandienai turbūt didžiausia šios technologijos mada ateina į automobilius. Dauguma automobilių gamintojų aiškiai žengia tuo keliu, jog ateityje žmonėms automobilių vairuoti nebereikės, o tai atliks dirbtinis intelektas ir automobilis važiuos savaime. Tam įgyvendinti yra naudojami įvairūs vaizdo atpažinimo metodai ir būdai, jog automobilis gebėtų atpažinti kelio ženklus, kelio ženklinimą, kitus automobilius, kliūtis ir pėsčiuosius.

Šiame darbe yra analizuojami įvairūs vaizdo atpažinimo būdai ir metodai, bei tiriamas vaizdo atpažinimo tikslumas įvairiose aplinkose ir jo pagerinimo būdai. Analizei ir tyrimui bus pasitelkta sukurta išmaniojo įrenginio programėlė, kuri sugeba realiu laiku atpažinti vaizde esanti žmogų. Programėlė naudoja išmaniojo įrenginio kamerą, o joje fiksuojamuose vaizduose bandoma atrasti žmogaus veidą ir akis. Norint sukurti tiksliai veikiančią programėlę, reikės atlikti įvairius uždavinius.

Pagrindiniai darbo tikslai:

- Pasirinkti tinkamą vaizdo atpažinimo algoritmą
- Nustatyti nuo ko priklauso atpažinimo tikslumas
- Pasirinkti tinkamą vaizdo atpažinimo biblioteką
- Pagerinti atpažinimo spartą
- Pagerinti žmonių veidų atpažinimo tikslumą

# 2. ANALITINĖ DALIS

## 2.1. Vaizdo atpažinimas

Pagrindiniai vaizdo atpažinimo uždavinių tikslai yra sugebėti atpažinti objektus, scenas ir kategorijas. Tai uždaviniai, kurie sutinkami įvairiuose srityse, o dažniausiai dirbtiniame intelekte, paieškose naudojant paveiksliukus, vaizdo medžiagos analizėse, objektų atpažinime ir robotikoje. Vaizdo atpažinimo sąvoka yra labai plati ir abstrakti. Į šią įeiną daugybę skirtingų vaizdo atpažinimo būdų ir metodų. Taip pat vaizdo atpažinimą galima vertinti skirtingai. Iš esmės atpažinti kokia yra daikto spalva jau yra atpažinimas, tačiau taip pat galima atpažinti ir patį objektą. Šiame skyrelyje pabandysime apžvelgti ir išskirti skirtingus vaizdo atpažinimo atvejus.

Atpažinimas dažniausiai skirstomas į dvi dalis. Pirmoji tai kategorijų atpažinimas ir antroji tai specifinis atpažinimas. Specifiniame atpažinime stengiamasi identifikuoti konkretų objektą, vietą, žmogų, kaip pavyzdys būtų Eifelio bokštas, Barakas Obama ar žurnalo viršelis. Kategorijų atpažinime stengiamasi atpažinti skirtingus objektus iš kažkurios bendrinės klasės. Kaip pavyzdys būtų pastatas, automobilis, šuo.

## 2.2. Problematika

Objektų atpažinimas turi daugybę iššūkių. Skirtingi objektai, kurie priklauso tai pačiai kategorijai, vaizde gali atrodyti visiškai skirtingai. Tai priklauso nuo apšvietimo, objekto pozicijos, kameros kampo ir fono, kuriame yra pats objektas. Be šių sunkumų, šiai dienai naudojami algoritmai taip pat susiduria su sunkumais, ypač kuomet reikia dirbti su didelėmis duomenų bazėmis ir apdorojant ypač aukštos raiškos paveikslėlius [4 p. 3].

## 2.3. Būdai

Praktikoje pasitaiko skirtingų vaizdo atpažinimo būdų. Šiame skyrelyje bus pateikti visi populiariausieji ir aprašyti pagrindiniai jų skirtumai

### 2.3.1. Simbolių ar kodų atpažinimas

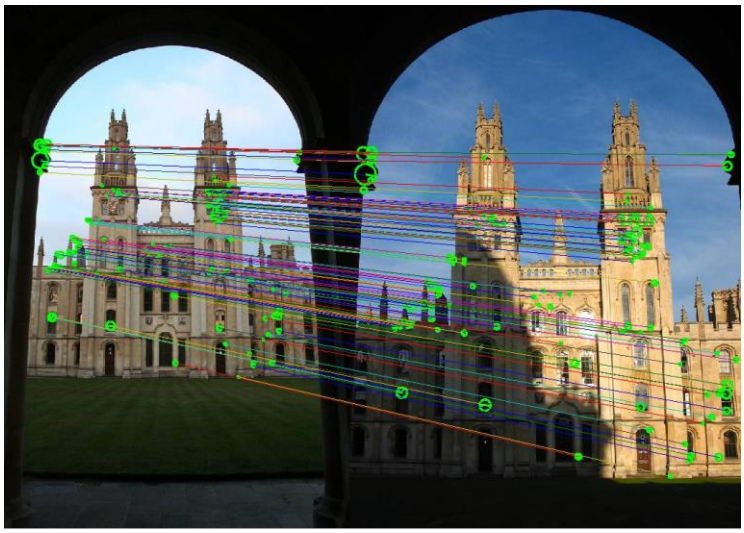
Tai pats paprasčiausias ir lengviausiai suprantamas vaizdo atpažinimas. Šis vaizdo atpažinimas dar dažnai vadinamas optiniu simbolių atpažinimų (*Optical Character Recognition angl.*). Tai kodai, kurie dažniausiai yra tuo pačiu metu suprantami ir žmogui ir elektroniniam skaitytuvui. Tokius kodus dažniausiai randame ant įvairių pakuočių, pašto ir kitų dokumentų. Ši technologija pasirodė dar 1970-ais metais ir yra naudojama iki šiol. Kasdien susiduriame su tokiu būdu parduotuvėse, kuomet yra nuskanuojamos prekės. Tokių kodų nuskaitymą galima atlikti praktiškai su visais išmaniais

įrenginiais, turinčiais vaizdo kamerą. Taip pat verta paminėti, jog toks būdas yra pakankamai patikimas ir tikslus.

Šį sprendimą įgyvendinti yra ganėtinai paprasta. Nesunku rasti visiems viešai prieinamų bibliotekų, kurios skirtos pernaudojimui kuriant savo programėles. Turbūt vienas iš lengviausiai panaudojamų dalykų rinkoje yra brūkšniniai (bar) kodai. Tokį dalyką integruoti į savo aplikaciją nereikalauja daugybės pastangų. Šiek tiek paieškojus internete, galima lengvai rasti bent keletą puikiai veikiančių bibliotekų tokių kaip „ZXing“. [1]

### 2.3.2. Objektų atpažinimas

Šis būdas ir metodas yra naudojamas populiarioje „Google“ programėlėje pavadinimu „Google Goggles“. Programėlės, kurios naudoja tokį būdą, paima nuotrauką, bando ją skaityti ir pritaikydamas tam tikrus algoritmus bando sudėlioti objekto vaizdą. Toliau šis vaizdas yra palyginamas pagal tam tikras panašias detales su kitais paveiksliukais esančiais internete ir tuomet yra išvedamas rezultatas. Tokie algoritmai iš esmės ieško kritiniu detalių paveiksle, kurias turi būtent tik tas objektas. Kaip pavyzdį galima pateikti tam tikrą pastatą kaip Eifelio bokštas ar unikalus piešinys „Mona Lisa“. Tokie objektai yra atpažįstami nepaisant iš kokio kampo ar kokio atstumo yra nufotografuojami. [2]



**pav. 1 Objektų atpažinimas [2]**

### 2.3.3. Objektų klasių atpažinimas

Šis atpažinimas sukelia turbūt daugiausiai problemų vaizdo atpažinimo metoduose ir sprendimuose. Objektų klasių atpažinimas reiškia, jog reikia atpažinti ne tik patį objektą, tačiau sugebėti jį ir klasifikuoti. Kaip pavyzdys būtų, jog programa turi atpažinti kokios veislės yra šuo ar kokios markės yra automobilis. Šio būdo realizavimui daugumai atvejų reikia stipraus dirbtinio intelekto.

### 2.3.4. Senos supratimas ir interpretavimas

Tai yra pats sudėtingiausias atpažinimo atvejis. Senos supratimas ir interpretavimas reiškia, jog programėlei reikia atpažinti keletą objektų vienu metu. Kaip pavyzdys būtų, jog programėlė išanalizavus senoje esančius objektus, turi nuspręsti, jog tai yra tarkim paplūdimys ar sporto salė. Ši problematika yra labai sudėtinga ir nuolat tiriama. Tokie sprendimai privalo naudoti automatinį programėlės mokymąsi iš interneto paveikslėlių ir žymių. [3]

## 2.4. Algoritmai

Vaizdo atpažinimui gali būti naudojami skirtingi jų algoritmai, kurių realizacija yra skirtinga, tačiau rezultatai yra labai panašūs. Skirtingi algoritmai turi savus privalumus ir trūkumus.

Dažniausiai programinėje įrangoje yra sutinkami šie algoritmai:

- Šablonų aptikimas (specifinių objektų atpažinimas)
- Briaunų aptikimas

### 2.4.1. Šablonų aptikimas

Šis algoritmas iki šios dienos yra vis dar tobulinamas. Nors ir bandoma atrasti skirtingų šio algoritmo atmainų, visi jie remiasi tais pačiais principais.

#### 2.4.1.1. Globalus paveikslėlių atpažinimas

Pati paprasčiausia objekto atpažinimo realizacija yra kiekvieno pikselio ryškumą arba spalvos saugojimas ir pagal tam tikrą eilės tvarką to paties šablono paieškojimas paveikslėlyje. Jei pavyksta rasti tą pačią taškų eilę paveikslėlyje, galima daryti prielaidą, jog objektas priklauso tai pačiai klasei. Šį sprendimo būdą dar 1992-aisiais pasiūlė Matthew Turk ir Alex Pentland.

1995-aisiais Hiroshi Murase ir Shree K. Nayar panaudojo šią idėją dirbant su 3D objektais ir sukonstravo sistemą, kuri sugebėjo atpažinti 100 objektų realiu laiku. Esant skirtingoms objektų pozicijoms, suprojektuoti paveikslėliai būdavo paverčiami į eigen-erdvę. Tai erdvė, kuri sudaryta iš eigen-vektorių. Šie vektoriai yra gaunami iš skirtingų to paties objekto paveikslėlių. Hiroshi Murase ir Shree K. Nayar sukurta sistema sugebėjo atpažinti nekintamos pozicijos objektus.

Peter Belhumeur ir Kriegmen praplėtė atpažinimo galimybes pasukdami kitu keliu. Principinių komponenčių analizės (*toliau PCA*) būdu sukurta erdvė buvo suprojektuota taip, jog sumažintų visas dėl rekonstrukcijos atsiradusias klaidas, kuomet paveikslėliai buvo projektuojami keičiant jų dimensijas.

Kiek vėliau buvo pasiūlyta optimizuoti objektų klasės atskyrimą vietoj Fišerio tiesinio diskriminanto analizės sukurtos erdvės. To pasėkoje buvo įgyvendintas „Fišerio veidų“ algoritmas,

kuris skirtingai nuo PCA metodų, galėjo spręsti konkretesnes užduotis kaip pavyzdžiui atskirti žmogų su akiniais ir be akinių.

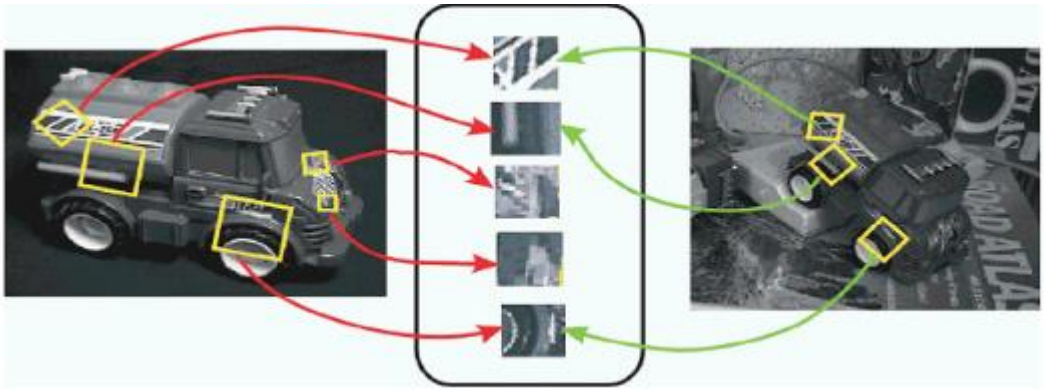
Vis dėlto visi šie būdai turėjo didelį trūkumą, jog visi pavyzdiniai paveikslėliai 2D erdvėje turėjo būti labai panašūs. Tai reiškia, jog bandant apmokyti vieną ar kitą algoritmą prireikdavo daugybės laiko. Pavyzdžiui jei veido atpažinimo klasei buvo panaudoti įvairaus intensyvumo priekinio veido paveikslėliai, tai šiek tiek paslinktas vaizdas ir algoritmas galvos, jog žmogaus nosis bus nebe toje vietoje ir veido atpažinti nepavyks.

1991-aisiais Swain ir Ballard pirmą kartą vaizdo atpažinimui panaudojo spalvotas histogramas. Ši idėja buvo panaudota ir patobulinta 2000-aisiais metais. Tą padarė Schiele ir Crowley. Histogramos buvo perdarytos į kelių dimensijų histogramas, o paveikslėliams atpažinti buvo naudojami artimiausio kaimyno metodai (*neighborhood operators angl.*). Šios histogramos užfiksuodavo tam tikros objekto ypatybės tikimybinį pasiskirstymą. Taip pat Schiele ir Crowley pasiūlė metodą, kuris matuoja skirtingų modelių iš kelių pavyzdinių paveikslėlių didžiausią a-posteriori tikimybę [4 p. 7].

#### 2.4.1.2. Paveikslėlio požymių atpažinimas

Sėkminga šios metodologijos realizacija padarė didžiulę įtaką vaizdo atpažinimo tyrimuose. Tai leido sukurti tiksliai ir efektyviai veikiančias vaizdo atpažinimo sistemas, kurios gebėdavo atpažinti objektus esam skirtingomis sąlygomis

Tai metodika, kuri stengiasi objekte rasti tam tikrus išskirtinius objekto požymius ir juos palyginti su jau paruoštais požymių šablonais. Iš pradžių objektų požymiai yra išrenkami iš abiejų paveikslėlių atskirai. Tuomet jie yra palyginami. Tam naudojami tokie algoritmai kaip SIFT ir SURF. Vis dėlto šių žingsnių neužtenka užtikrinti idealaus objektų atpažinimo, todėl papildomai yra atliekami įvairūs geometriniai pakeitimai.



**pav. 2 Požymių išskyrimas [4. p. 10]**

Pagrindiniai atliekami veiksmai

1. Išskiriami objekto požymiai iš apsimokymui skirto ir pavyzdinio paveikslėlio
2. Sulyginami paveikslėlių požymiai
3. Patikrinama ar sutapę požymiai turi atitinkančias geometrines savybes

Didelę įtaką tobulinant vaizdo atpažinimą padarė nekintamų požymių funkcionalumas. Tai leido algoritmams surasti paveikslėlio požymius ir juos užkoduoti taip, jog net ir atlikus įvairias paveikslėlio transformacijas, objektas būtų ir toliau tiksliai atpažintas [4 p. 9]

2.4.1.2.1. Požymių išskyrimas

Pirmieji žingsniai norint surinkti paveikslėlio požymius, tai atrasti tas paveikslėlio savybes, kurios būtų patikimos net ir esant skirtingoms sąlygomis ir matymo kampams. Jog atrastume tokias vietas, tam yra naudojami detektoriai. Labiausiai vaizdo atpažinime naudojami yra Hessian ir Harris detektoriai.

Hesian detektorius ieško tokių paveikslėlio vietų, jog jas sudarytų dviejų statmenų krypčių išvestinę. Detektorius paremtas antros eilės matricos išvestine. Norint užtikrinti gerą tikslumą, taip pat yra panaudojamas Gauso filtras su parametru σ. Pati Hesiano matrica atrodo taip:

$$H(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma)\ L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma)\ L_{yy}(x,\sigma) \end{bmatrix}; (1)$$

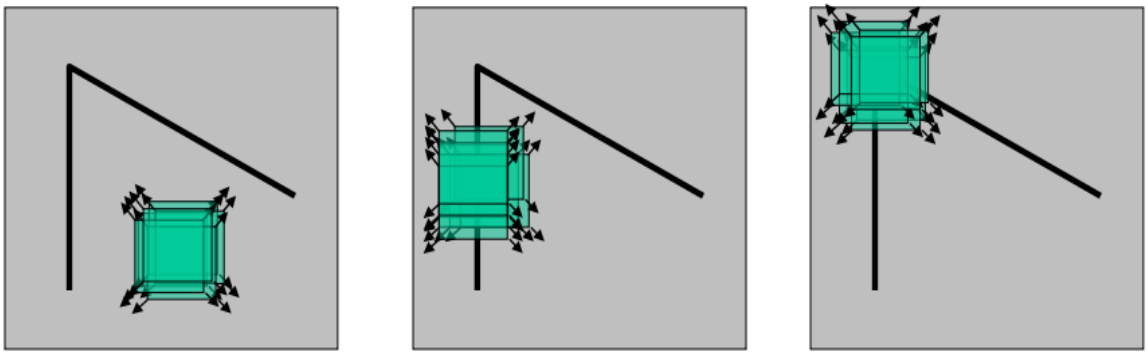
Čia Hesiano matrica apibrėžiama $H(x,\sigma)$ – duotam taškui $x = (x,y), L_{xx}(x,\sigma)$ – Gauso antros eilės išvestinė $\frac{\delta^2}{\delta_{x^2}} g(\sigma)$ su vaizdu $I$ taške $x$; atitinkamai su $L_{xy}(x,\sigma)$ ir $L_{yy}(x,\sigma)$. Kiekvienam vaizdo taškui yra skaičiuojamas Hesiano matricos determinantas $\det(H) = I_{xx}I_{yy} - I_{xy}^2$, kurio reikšmės naudojamos nustatyti požymius ir užtikrinti gerą tikslumą. Dažniausiai reikšmės naudojamos 3x3 langeliui užpildyti. Tuomet langelis yra uždedamas ant paveikslėlio ir paliekami tik tie taškai, kurių reikšmė yra didesnė už 8 tiesioginius greta esančius tašus lango viduje. Detektorius grąžina visas likusias vietas, kurių reikšmės yra didesnės nei prieš tai nustatyta riba $\theta$ [4 p. 13]

**pav. 3 Hesiano detektorius [5]**

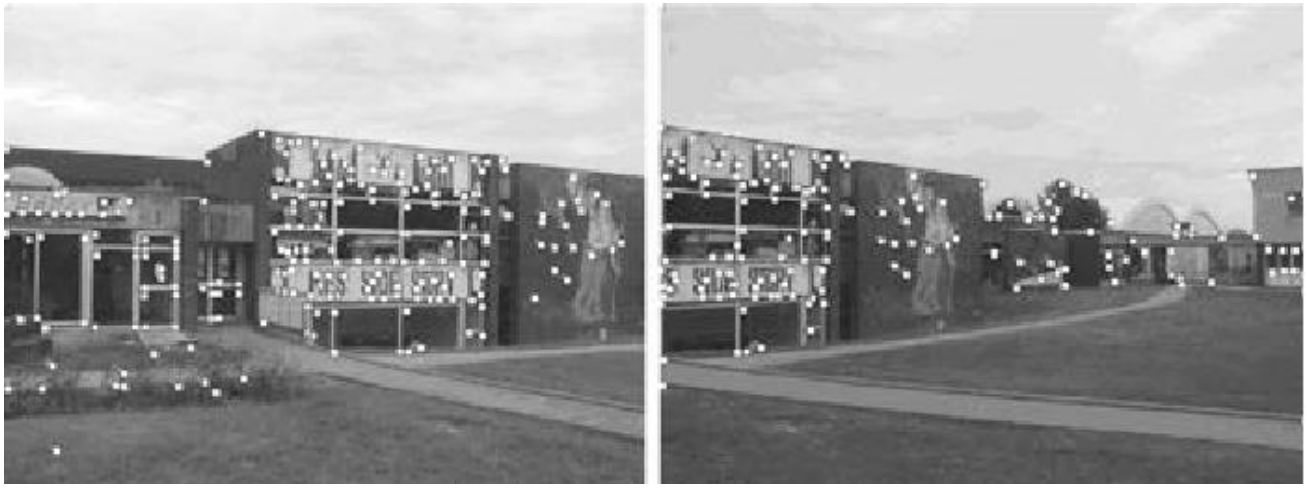
Harris detektorius buvo sukurtas dėl geometrinio stabilumo. Šis detektorius nustato pagrindinius taškus, kurių intensyvumas yra labiausiai pasikeitęs. Praktikoje šie taškai yra dažniausiai objekto kampai. Detektorius ieško taškų $x$, kurių antros eilės išvestinės $C$ prie taško $x$ tri dvi dideles eigen-reikšmes:

$$C(x,\sigma,\tilde{\sigma}) = G(x,\tilde{\sigma}) * \begin{bmatrix} I_x^2(x,\sigma) & I_xI_y(x,\sigma) \\ I_xI_y(x,\sigma) & I_y^2(x,\sigma) \end{bmatrix}; \quad (2)$$

4 pav. pirmajame iliustracijos paveikslėlyje simbolizuojama lygi erdvė, todėl stumdant detektoriaus langelį, intensyvumas iš esmės nesikeičia. Antrajame paveikslėlyje langelis stumdomas ant linijos, todėl intensyvumas taip pat išlieka vienodas. Trečiajame paveikslėlyje pavaizduotas langelio slankiojimas ant kampo ir tai sąlygoja akivaizdų taškelių intensyvumo pokytį.



**pav. 4 Kampų aptikimas [6]**



**pav. 5 Požymių išskyrimas [7]**

2.4.1.2.2. Nekintančio dydžio aptikimas

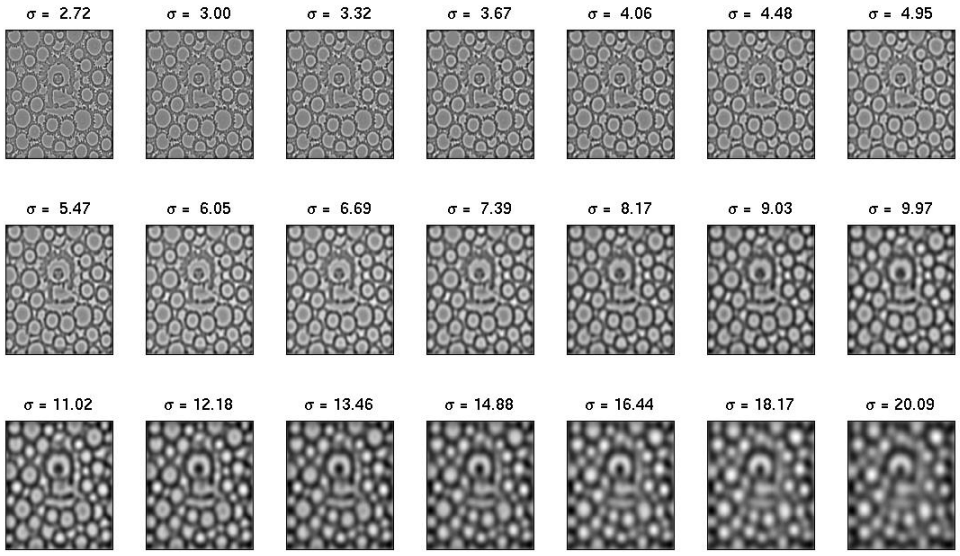
Nors ir Harris ir Hessian detektorių metodai padarė didelę įtaką vaizdo atpažinimo moksle, tačiau jų detektoriai veikė sąlyginai tik su labai nedideliais paveikslėlio dydžio pokyčiais. Tai yra todėl, nes abu detektoriai pasikliauja Gauso išvestinėmis. Jei paveikslėlių dydis per daug skiriasi nuo pavyzdinio paveikslėlio, tuomet rezultatas taip pat bus skirtingas. Nekintamo mastelio aptikimas išsprendžia šią problemą ir leidžia kisti paveikslėlio dydžiui. [8]

**Automatinis mastelio parinkimas**. Pagrindinė šio algoritmo idėja yra dviejų paveikslėlių poroje, kiekvienam požymio taškui pabandyti nustatyti ar kaimyniniai taškai turi tą pačią struktūrą, nepriklausomai nuo paveikslėlio dydžio. Iš esmės tai galima pasiekti tiesiog padarydami daugybę skirtingo dydžio paveikslėlio šablonų ir atlikti $NxN$ palyginimą. Vis dėlto ši operacija reikalauja daug resursų ir laiko, todėl naudojama parašo (*signature angl.*) funkcija.

Ši funkcija matuoja paveikslėlio ypatybes pagal tam tikrą spindulį kaimyniniuose taškuose. To pasėkoje, palyginus dviejų skirtingų dydžių paveikslėlių pagrindinius taškus, gaunamas ganėtinai tikslus rezultatas. Vienintelis skirtumas bus tas, jog vienas paveikslėlis bus arba suspaustas, arba ištemptas dėl pakitusio mastelio.

**Gauso-Laplacian detektorius.** Panaudojus automatinio mastelio parinkimo idėją, Lindeberg pasiūlė detektorių, kuris paveikslėlyje ieško požymių, kurie įgis maksimumą tokiame paveikslėlio dydyje, kuris atspindės tikrąją objekto struktūrą.

$$L(x,\sigma) = \sigma^2 \left( I_{xx}(x,\sigma) + I_{yy}(x,\sigma) \right); (3)$$




**pav. 6 Gauso filtras [9]**

**Harris-Laplacian detektorius**. Tai detektorius, kuris pasižymėjo geresniu objektų atskyrimu lyginant su Gauso-Laplaciano detektoriumi. Šį kartą buvo sukombinuotas jau aprašytas Harris kampų radimo būdas ir kintančio mastelio Lindebergo mechanizmas. Iš pradžių metodas sukuria dvi skirtingų mastelių erdves. Tuomet Harris funkcija lokalizuoja kandidatų pagrindinių požymių taškus, kurie įgyja maksimumą, keičiant jų mastelį. To rezultate, taškai išlieka vienodi keičiant paveikslėlių dydį, pasukimo kampą, apšvietimą ir triukšmą. Vis dėlto iš pradžių sukurtas Harris-Laplaciano detektorius grąžindavo pakankamai mažą kiekį rezultatų. Tai įvyko dėl to, jog buvo įvestas papildomas reikalavimas, jog kiekvienas taškas turėtų dvi skirtingas maksimumo sąlygas vienu metu. Daugumai realių vaizdo atpažinimo programų, mažesnioji reikšmė dažniau būna trūkumas nei privalumas, nes tai sumažina atpažinimą esant skirtingam apšvietimui. Tai ypač yra neveiksminga, kuomet programa bando atpažinti objekto kategoriją. [10]

To pasėkoje buvo atnaujintas Harris-Laplacian detektorius pasiūlant mažesnius tikslumo reikalavimus. Vietoj to, jog būtų ieškomi du maksimumai tuo pačiu metu, metodas pasirenka Laplaciano maksimumą toje pačioje vietoje, kurioje Harris funkcija taip pat pasirenka maksimumą esant bet kokiam paveikslėlio dydžiui. Rezultate šis detektorius randa daugiau taškų su šiek tiek mažesniu tikslumu ir puikiai veikia vaizdo atpažinimo programose, kuriose reikalinga didelė sparta ir dirbama su daug objektų turinčiais vaizdais.

**Hessian-Laplacian detektorius**. Lygiai taip pat kaip ir Harris-Laplacian detektorius, ta pati idėja gali būti pritaikyta ir Hessian metodui. Dirbant su vienodo dyžio paveikslėliais, Hessian-Laplaciano detektorius dažniausiai grąžina daugiau paveikslėlio požymių nei Harris-Laplaciano detektorius su šiek tiek mažesniu pasikartojimu.

2.4.1.2.3. Kintamo regiono aptikimas

Dažniausia praktikoje sutinkami vaizdo atpažinimo uždaviniai, kuomet vaizdas yra kintamas. Todėl stengiamasi surasti tokius paveikslėlio požymius, kurie tiktų ir visiškai pasikeitus matymo kampui. Jei laikytume, jog scenos struktūra, kuri mus domina yra visiškoje plokštumoje, tuomet padarius perspektyvos iškraipymą, paveikslėlio požymiai būtų ir toliau matomi. Vis dėlto, tokie perspektyvos iškraipymai yra neefektyvūs ir turi didelę tikimybę klaidoms, nes dažniausiai požymiai būna sudaryti iš labai mažo kiekio taškų. Kol kintamas mastelis ir pasukimas gali būti apibūdinamas kaip apskritimas, tai kitos geometrinės transformacijos apskritimą pakeičia į elipsę. Tai vyksta dėl to, jog ieškant regionų, elipsė yra patikimesnė ir gali būti pakartotinai ištraukiama iš paveikslėlio savybių [4 p. 20]

**Harris ir Hessian Affine detektorius**.

Tiek Harris-Laplace, tiek Hessian-Laplace detektoriai gali būti praplėsti taikyti gimiringus kintamus regionus. Iš pradžių paimamas apskritimo formos regionas, kuri grąžina originalus nekintamo mastelio detektorius. Tuomet kiekvienoje iteracijoje sukuriama matrica ir paskaičiuojamos eigen reikšmės. Taip gaunama kintamo regiono deformuota elipsės forma. Tuomet paveikslėlis yra transformuojamas taip, jog elipsės forma virstu į apskritimą ir atnaujintų transformuoto paveikslo išdėstymą ir dydį. Procedūra yra kartojama tol kol eigen reikšmės ir antrosios eilės matricos reikšmės yra beveik vienodos.

Šis iteracinis metodas leidžia elipsinių regionų setui prisitaikyti prie lokalių intensyvumo šablonų, tam, jog tokių pačių objektų struktūros sutaptu nepaisant įvykusių deformacijų dėl pasikeitusio matymo kampo. [4 p. 21]

**Maksimaliai stabilūs ekstremalūs regionai**.

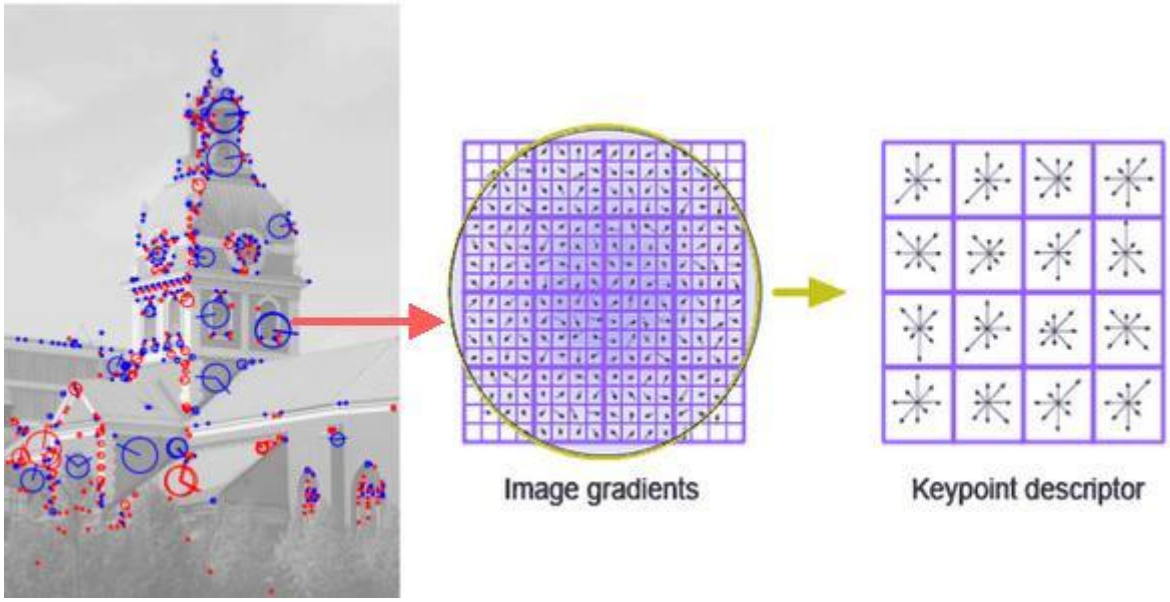
Skirtingą būdą, bandant rasti nekintamos geometrijos regionus, pasiūlė Matas et al. Palyginant su prieš tai aprašytais metodais, kurie pradedami nuo pagrindinių taškų, palaipsniui pridėdavo nekintamus lygius, šis būdas pradeda iš segmentacijos perspektyvos. Metodas paveikslėliui pritaiko takoskyros segmentacijos algoritmą ir išskiria vienodo intensyvumo regionus, kurie išlieka stabilus pritaikant skirtingas slenksčio ribas. Išskirti regionai išlieka stabilus prie skirtingų paveikslėlio sąlygų ir yra patikimi pasikeitus matymo kampui. [4p. 21]

### 2.4.1.3. Vietos ir požymių atpažinimas

Kada regionai yra išrinkti iš paveikslėlio, jų turinys turi būti šifruojamas taip, jog būtų tinkamas deskriptoriui palyginimui.

2.4.1.3.1. SIFT deskriptorius

Vienas populiariausių ir labiausiai naudojamų deskriptorių yra SIFT (Scale Invariant Feature Transform angl.). Šį būdą pasiūlė David G. Lowe dar 1999-aisias metais. Šis metodas puikiai tiko įvairiems uždaviniams spręsti ir buvo suderinamas su visais regionų detektoriai. SIFT deskriptoriaus tikslas pasiekti puikų atpažinimą besikeičiant apšvietimui ir mažiems pozicijos pokyčiams. Tam, jog tai būtų pasiekta, paveikslėlio informacija yra šifruojama į setus, kuriuos sudaro gradiento orientacijos histogramos. Deskriptorius pradeda ieškoti stabilių normalizuotų regionų, panaudodamas vieną iš anksčiau aprašytų detektorių. Pirmajame žingsnyje deskriptoius išskiria taškinius požymius iš paveikslėlio. Juos vėliau aprašo, tai reiškia, jog yra sudaromas taškinių požymių vektorius. Trečiajame žingsnyje vektoriai, kurie paimti iš skirtingų paveikslėlių yra palyginami. [11]

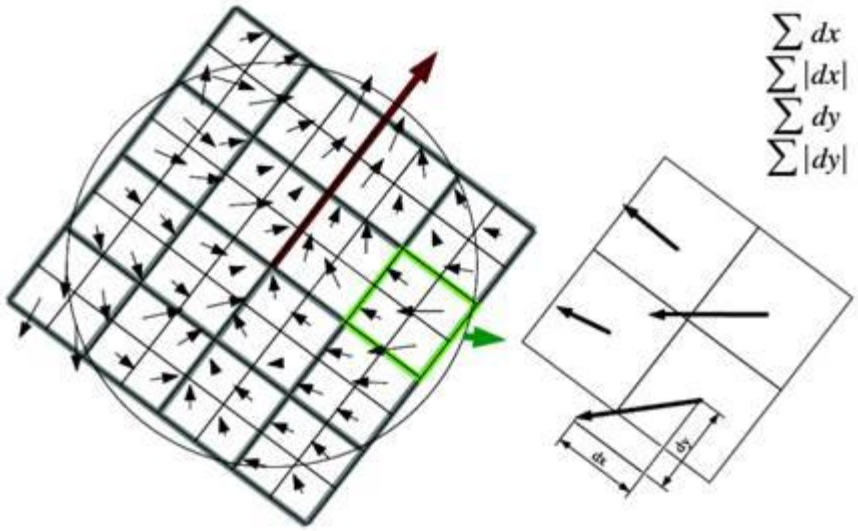



**pav. 7 SIFT deskriptorius [12]**

2.4.1.3.2. SURF deskriptorius

Požymių detektoriams ir deskriptoriams tampant vis populiaresniais, efektyvūs vaizdo atpažinimo būdai darėsi vis svarbesni. Po kiek laiko buvo pasiūlytas efektyvesnis ir spartesnis būdas. 2008-ais metais SURF (Speeded-Up Robust Features angl.) pasiūlė Herbert, et al. Tai buvo efektyvesnė alternatyva nei SIFT.

SURF metodas naudoja Hessian-Laplace regionų detektorius ir savo deskriptorių, kuris apibūdina kaip taškų intensyvumas yra pasiskirstęs nuo šalia esančio kiekvieno požymio [13].




**pav. 8 SURF deskriptorius [14]**

### 2.4.1.4. Haar Cascade

Tai algoritmas, kuris buvo panaudotas pirmame realiai veikiančiame veidų atpažinimo sistemoje. Šis būdas atsirado tuomet, kada dirbant vien su paveikslėlių taškų intensyvumo reikšmėmis trūko spartos ir tikslumo. Dėl šios priežasties Viola ir Jones pasiūlė alternatyva dirbti su Haar bangomis. Harr požymius sudarė kaimyniniai stačiakampiai, kurių taškų intensyvumo suma kiekviename regione buvo lyginami su kitais stačiakampiais. Skirtumas tarp šių sumų buvo skirstomas į kategorijas. Kaip pavyzdys būtų darbas su žmonių veidais. Įprastu atveju vietos aplink žmonių akis yra tamsesnės nei vietos skruosto dalyje. Haar požymių detektorius bandant atpažinti žmonių veidus, naudoja du kaimynystėje esančius stačiakampius su vietomis esančiomis virš akių ir skruosto dalyje [15]

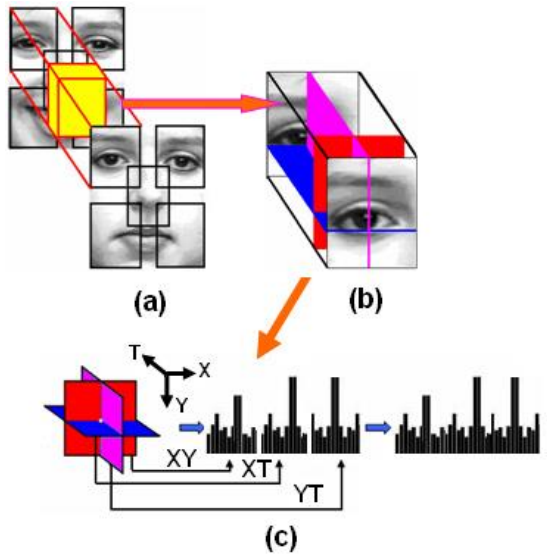
#### 2.4.1.4.1. Cascade klasifikatorius

Cascade klasifikatorių sudaro sąrašas stadijų, kur kiekvieną stadiją sudaro sąrašas apmokytų duomenų. Klasifikatoriai naudojami Haar Cascade algoritme. Sistema ieško objektų stumdydama paieškos langelį per paveikslėlį. Kiekviena klasifikatoriaus stadija žymi specifinį regioną, kuris apibrėžtas pagal dabartinę vietą langelyje kaip teigiama arba neigiama. Teigiama reiškia, jog objektas buvo rastas, neigiama – objektas buvo nerastas. Jei žymėjimas duoda neigiamą rezultatą, tuomet regiono klasifikatorius baigia darbą toje vietoje ir langelis yra perstumiamas į kitą vietą. Jei žymėjimas duoda teigiamą rezultatą, tuomet regionas analizuojamas jau kitoje klasifikatoriaus stadijoje. Klasifikatorius pasako galutinį verdiktą kaip teigiamą rezultatą, kai visos stadijos, įskaitant ir paskutinę sako, jog paveikslėlyje yra rastas objektas.

2.4.1.4.2. LBP klasifikatorius

LBP (Local Binary Pattern angl.) yra labai paprastas ir efektyvūs tekstūrų operatorius, kuris pažymi paveikslėlio taškus, žymint kaimyninius taškus ir apibrėžiant rezultatą kaip binarinį skaičių. Dėl šių paprastų skaičiavimų, LBP tekstūrų operatorius tapo labai populiarus įvairiuose vaizdo atpažinimo sprendimuose [16]

- LBP klasifikatorius gaunamas padalinant langelį į celes (pavyzdžiui 16 x 16)
- Kiekvienas celės taškas palyginamas su kitais kaimyniniais taškais.
- Kur taško reikšmė yra didesnė nei kaimyno, ten įrašomas 0, kitu atveju įrašomas 1. Taip gaunamas 8 skaitmenų binarinis skaičius
- Sukuriama histograma su pasikartojančiais binariniais skaičiais.
- Sujungiamos celių histogramos ir gaunamas paveikslėlio požymių vektorius.




**pav. 9 LBP klasifikatorius [17]**

## 2.5. Pasaulyje egzistuojantys sprendimai

Pasaulyje egzistuoja daugybę skirtingų vaizdo atpažinimo sprendimų. Kai kurie iš jų nėra pasiekiami įprastiniam žmogaus naudojimui, tačiau yra ir programų, kurios gali palengvinti įprastas žmonių operacijas kaip paieška internete.

### 2.5.1. „InfoEye“

Tai „Sony“ produktas „InfoEye“. Įmonė sugalvojo naują ir inovatyvų būdą ieškoti informacijos. Programėlės principas nėra sudėtingas – nufotografuojamas objektas apie kurį norime sužinoti daugiau informacijos ir leidžiama programėlei surasti informaciją. Ši būna pateikiama už kelių

sekundžių išmaniojo įrenginio ekrane. Tai veikia su visais mums žinomais objektais kaip Eifelio bokštas ar Didžiojo Beno laikrodis. Dar vienas šios programėlės panaudojimas yra bar kodų skanavimas. Paimama prekė, nuskanuojamas jos bar kodas ir yra pateikiama detalesnė informacija apie prekę ar gaminį.

### 2.5.2. „CamFind“

Ši programėlė skirta išmaniesiems įrenginiams yra skirta paprastesnei paieškai internete. Naudojant išmaniojo įrenginio vaizdo kamerą užfiksuojami vaizdai ir yra atliekama paieška internete. Atlikus paiešką ir suradus rezultatų, objekto pavadinimas yra ištariamas garsu ir yra pateikiamos nuorodos apie susijusį objektą. Tarkime nuskanavus kompiuterio klaviatūrą, yra duodamos nuorodos į įvairias parduotuves, kur galima rasti būtent tokią kompiuterio klaviatūrą [18].

### 2.5.3. „Google Goggles“

Dar viena alternatyva „CamFind“ programėlei. Šią programėlę pristatė „Google“ kompanija. Tai vaizdo atpažinimo programėlė, skirta ieškoti informacijos pateikus paveikslėlius. Programėlė puikiai veikia nufotografavus tam tikrus objektus ar vietoves. Nufotografavus tokią vietą, išmaniajame įrenginyje pasirodo daug informacijos susijusios su objektu. Taip pat programėlė puikiai veikia skanuojant bar kodus. Tokia funkcija pasitarnauja norint sužinoti daugiau apie konkrečią prekę [19].

## 2.6. Objektų atpažinimo bibliotekos ir įrankiai

Rinkoje egzistuoja jau nemažai paruoštų sprendimų, kuriuos galime pritaikyti vaizdo atpažinimo uždaviniams spręsti.

### 2.6.1. OpenCV

Tai atviro turinio biblioteka, kuri išleista su BSD licencija, todėl yra visiškai nemokama tiek moksliniams, tiek komerciniams tikslams. Biblioteka yra parašyta C++, C, Python kalbomis ir turi Java palaikymą. Šiuo metu biblioteka veikia Windows, Linux, Mac OS, iOS ir Android operacinėse sistemose. OpenCV buvo sukurtas įvairiems vaizdo atpažinimo užduotims spręsti ir ypač sukoncentruotas realaus laiko programinei įrangai. Dauguma metodų yra labai optimizuoti ir geba išnaudoti kelių branduolių procesoriaus spartesniam veikimui. Biblioteka turi didžiulę bendruomenę, o jos parsisiuntimai jau viršijo 14-ą milijonų. Verta paminėti, jog ji yra naudojama mene, minų kasyklose, žemėlapiuose ir robotų industrijoje [20].

### 2.6.2. Integrating Vision Toolkit (IVT)

Atviro turinio biblioteka, kuri parašyta naudojant C++ programavimo kalbą. Biblioteka siūlo įvairias paveiksliukų apdorojimo galimybes. Verta paminėti, jog bibliotekos kodas yra aiškiai suprantamas, todėl esant poreikiui galima modifikuoti pagal savo poreikius. Veikia ant Windows, Linux ir Mac OS X operacinių sistemų [21]. Pagrindiniai bibliotekos akcentai:

- Turi SIFT detektoriaus realizaciją
- Turi Harris kampų atpažinimo realizaciją
- Turi pilnai integruotą kameros modelį
- Turi daugybę filtrų ir dydžio keitimo galimybių
- Turi realizuotą K - D medžio algoritmą.

### 2.6.3. Visualization Toolkit (VTK)

Tai atviro turinio biblioteka, kuri skirta vaizdo apdorojimui. Pati biblioteka parašyta naudojant C++, Java r Python programavimo kalbas. Taip pat turi realizuotų daugybę algoritmų kaip daugiakampių sumažinimu, tinklelio glotninimu, karpymu ir kt. Verta paminėti, jog įrankis geba išnaudoti paralelinį apdorojimą ir integruoja įvairias duomenų bazes [22].

### 2.6.4. TensorFlow

Tai atviro turinio įrankis kūrėjams, kuris leidžia naudoti mašininio mokymosi algoritmus. Produktas yra sukurtas Google kompanijos ir pristatytas 2015-ais metais. Šis įrankis gali būti pritaikytas ant įvairių ir skirtingų įrenginių kaip išmanieji telefonai, planšetės ir gali būti naudojami netgi ypač sudėtingose sistemose, turinčias ypač galingas aparatūrines įrangas. TensorFlow leidžia vartotojams apmokyti skirtingus algoritmus, panaudoti neuroninius tinklus ir pritaikyti realizacija įvairiose srityse kaip vaizdo atpažinimas, balso atpažinimas ar robotika [23].

### 2.6.5. Bibliotekų palyginimas

Analizės metu atrastos bibliotekos buvo palygintos pagal penkis kriterijus:

- Veikimas Windows platformoje
- Veikimas Linux platformoje
- Veikimas Mac OS platformoje
- Veikimas Android platformoje
- Galimybė dirbti su kadrais iš vaizdo kameros

Visos keturios analizuojamos bibliotekos veikia populiariausiose Windows, Linux ir Mac OS platformose. Vis dėlto, tik OpenCV ir TensorFlow turi galimybę veikti Android operacinėje

sistemoje. Taip pat mūsų darbe labai svarbu, jog būtų galima dirbti su kadrais, tiesiai iš vaizdo kameros. Šią galimybę turi OpenCV, IVT ir TensorFlow bibliotekos (Žr. lentelė 1).

**lentelė 1 bibliotekų palyginimas**

| | OpenCV | IVT | VTK | TensorFlow |
|---|---|---|---|---|
| Veikia Windows platformoje | Taip | Taip | Taip | Taip |
| Veikia Linux platformoje | Taip | Taip | Taip | Taip |
| Veikia Mac OS platformoje | Taip | Taip | Taip | Taip |
| Veikia Android platformoje | Taip | Ne | Ne | Taip |
| Leidžia dirbti su kadrais iš vaizdo kameros | Taip | Taip | Ne | Taip |

## 2.7. Išvados

Atlikus literatūros analizę, buvo nuspręsta rinktis OpenCV vaizdo apdorojimo biblioteką dėl jos palaikymo Android operacinėje sistemoje. Sistemos tikslas buvo veikti kaip Android programėle ir vaizdams naudoti įrenginio vaizdo kamerą. Dėl šių priežasčių tai buvo vienintelė opcija, kuri tenkino sistemos reikalavimus. Atlikus metodų ir algoritmų analizę, buvo nuspręsta rinktis Haar cascade metodiką. Ši metodika puikiai tinka, nes klasifikatorių apmokymas yra paprastas procesas, o rasti žmonių veidų paveikslėlius yra labai paprasta. Klasifikatorių apmokymas nebuvo sudėtingas, o algoritmo veikimo rezultatai tenkino sistemos reikalavimus.

# 3. PROJEKTINĖ DALIS

## 3.1. Pagrindinis funkcionalumas ir veikimo principas

Pagrindinės sukurtos sistemos funkcijos yra analizuoti kameroje esanti vaizdą, atpažinti esančius žmonių veidus ir pranešti vartotojui, jog vaizde yra užfiksuoti ieškomi vaizdai.

Sukurta „Android“ įrenginio programėlė geba sparčiai atpažinti vaizde esančius žmonių veidus ir vibracijos pagalba pranešti vartotojui. Žmogui užfiksuoti reikalingas žmogaus veido kontūras ir akys.

## 3.2. Reikalavimų analizė

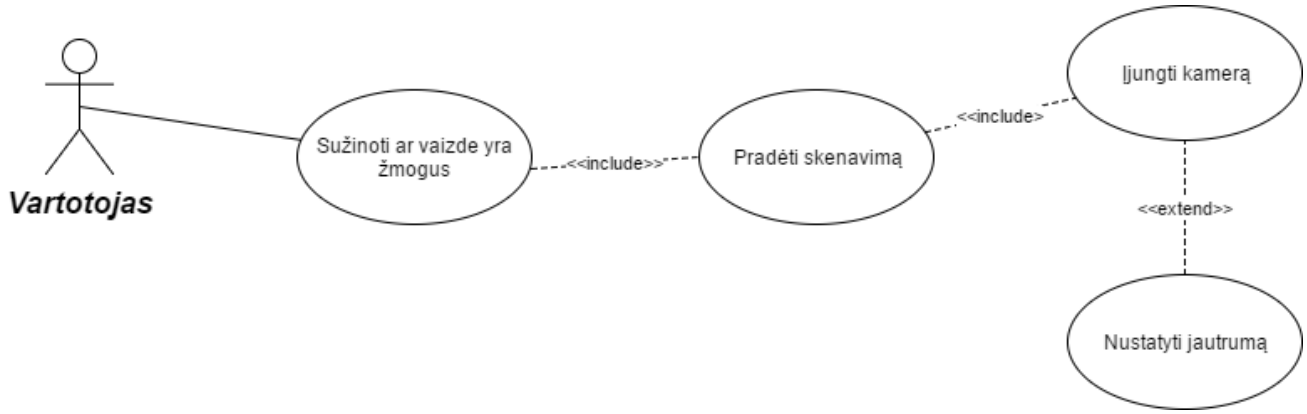
Esminiai sistemos reikalavimai:

- Vartotojui turi galėti valdyti programėlę balsu
- Programėlė atpažįsta vaizde esančius žmonių veidus
- Vartotojai turi galėti gauti informacija balsu ar kitomis ne vaizdinėmis priemonėmis

### 3.2.1. Nefunkciniai reikalavimai

- Skenavimas turi trukti iki 3s.
- Rezultatai turi būti pranešami tik aptikus objektą
- Programėlė turi gebėti atpažinti daugiau nei 3 objektus vienu metu

## 3.3. Panaudos atvejų diagrama



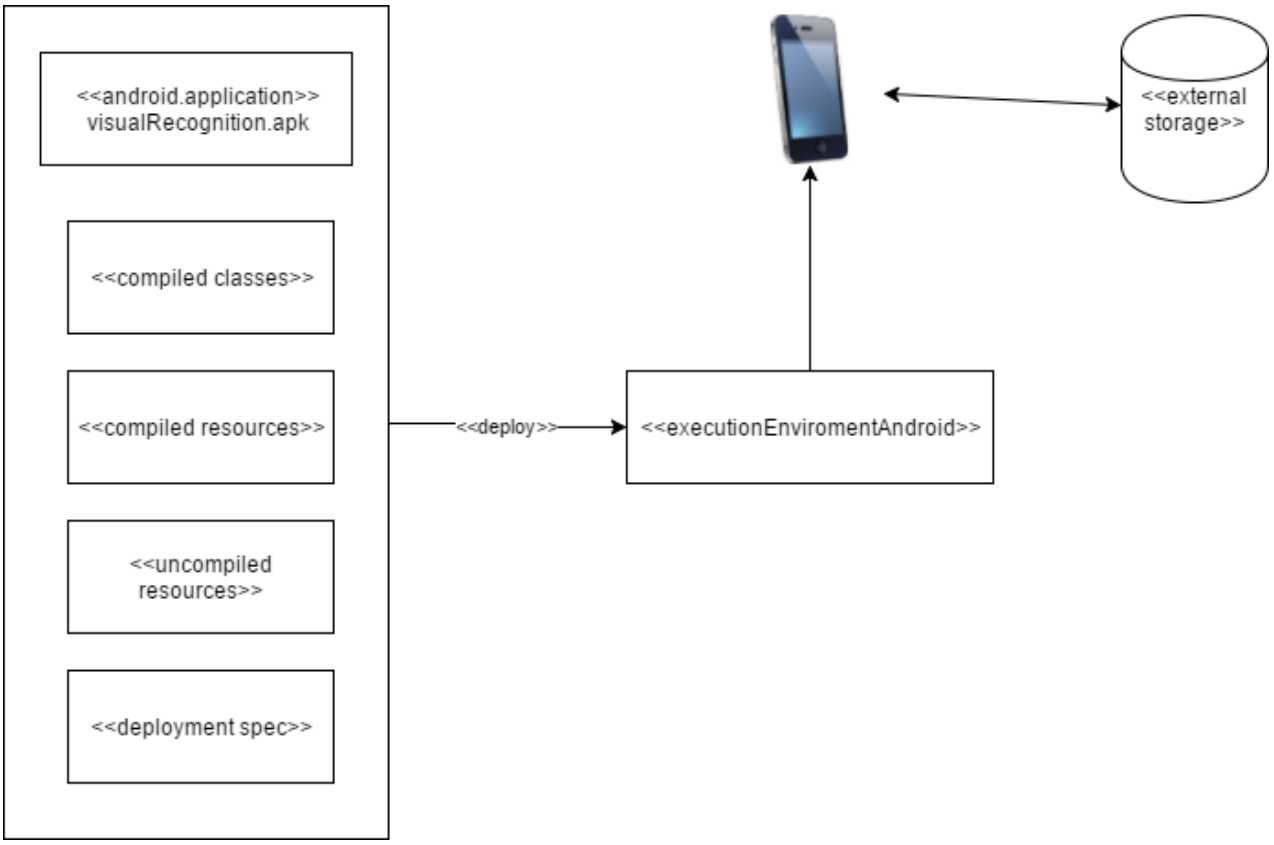

**pav. 10 PA diagrama**

Programėlės panaudos atvejis yra labai specifinis, tai yra sužinojimas ar kameros pagalba fiksuojame vaizde yra žmogus. Į tai įeina skenavimo pradėjimas. Vartotojas gali pats balso komanda pasirinkti kada pradėti skenavimą, tam jog būtų saugomi resursai. Į skenavimą įeina kameros įjungimas. Vartotojas turi galimybę pasirinkti kada įjungti kamerą. Šis panaudos atvejis praplečiamas jautrumo nustatymu. Vartotojas gali pasirinkti kokiu jautrumu skanuoti vaizdą.

## 3.4. Sistemos projektavimas

### 3.4.1. Išdėstymo vaizdas

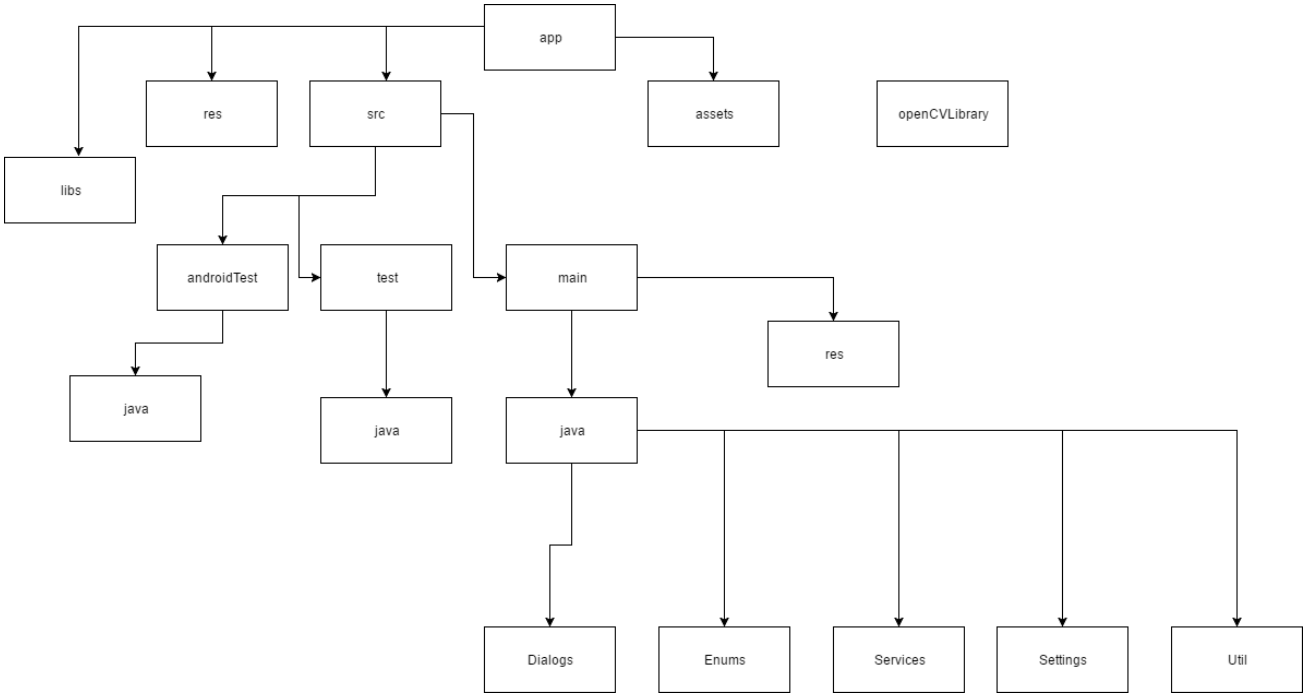
Žemiau esančiojo diagramoje pateikiama kaip atrodo galutinės sistemos išdėstymas.




**pav. 11 Išdėstymo vaizdas**

Pati schema yra labai paprasta. Iš pradžių yra surenkamas aplikacijos failas .apk, kuriame yra kompiliuotos klasės, kompiliuoti resursai, nekompiliuoti resursai ir išskleidimo ypatybės. Šis failas yra patalpinamas į „Android“ įrenginį ir jame instaliuojamas į vidinę atmintį. To pilnai pakanka, jog būtų galima naudotis programėle.

### 3.4.2. Sistemos statinis vaizdas




**pav. 12 Statinis sistemos vaizdas**

Sistema suskaidyta į šiuos paketus:

- app – pagrindinis programėlės katalogas
    - libs – įvairioms bibliotekoms laikyti skirtas katalogas
    - res – resursams skirtas katalogas
    - src – kodui skirtas katalogas
        - androidTest – testams, kurie testuoja android įrenginio funkcijas, skirtas katalogas
            - java – java kodo testams skirtams katalogas
        - test – vienetiniams testams skirtas katalogas
            - java – java kodo testams skirtas katalogas
        - main – pagrindiniam programėlės kodui skirtas katalogas
            - java – java kodas
                - Dialogs – skirtas dialogo klasėms kaip nustatymų langas
                - Enums – skirtas programos klasifikatoriams saugoti kaip balso komandos
                - Services – skirtas servisams tokiems kaip balso atpažinimo servisas, kuris turi balso atpažinimo logiką
                - Settings – paketas skirtas programos nustatymams

        - Util – pagalbinės funkcijos
    - Res – programos resursai, xml failai, kurie apibrėžia programėlės langų dizainą
- openCvLibrary – pagrindinė vaizdo atpažinimo bibliotekos katalogas, kuriame yra visi reikalingi metodai vaizdo atpažinimui

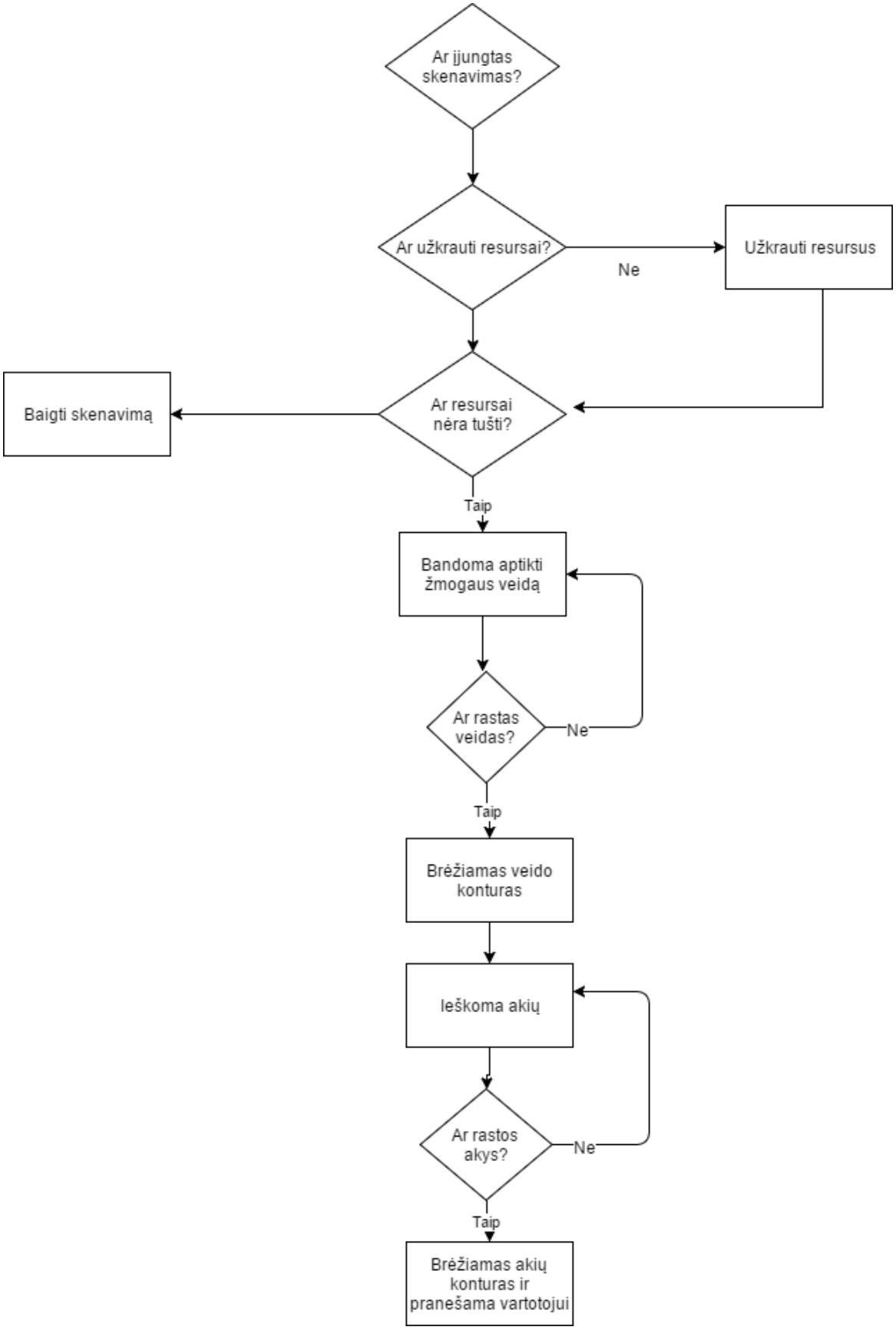
### 3.5. Vaizdo atpažinimas

Tam, jog būtų galima kameros vaizde atpažinti žmonių veidus, reikėjo sukurti pagrindinį metodą, kuris panaudotų tinkamus OpenCV bibliotekos algoritmus. Pats algoritmas atrodo taip:

1. Metodas iškviečiamas pasikeitus kameros vaizdo kadrui. Tai vyksta maždaug 30 kartų per sekundę
2. Patikrinama ar skenavimas yra įjungtas
3. Bandoma užkrauti resursus. Tai yra veido ir akių Haar Cascade šablonai
4. Patikrinama ar pavyko užkrauti resursus
5. Sukuriami tušti stačiakampių vektorių objektai, skirti saugoti veidams ir akims
6. Pasinaudoję OpenCV metodais, bandoma aptikti vaizdo kadre žmonių veidus
7. Jei pavyko rasti, brėžiamas kontūras aplink veidą
8. Radus veidą, bandoma ieškoti žmogaus akių esančiame veido kontūre
9. Jei pavyko rasti akis, tuomet brėžiamas kontūras aplink akis
10. Vartotojui yra pranešama, jog pavyko rasti vaizde žmogaus veidą

Detali algoritmo schema pateikta pav. 13.

### 3.5.1. Algoritmo schema




**pav. 13 Algoritmo schema**

### 3.5.2. Algoritmo realizacija

Apačioje pateikta pagrindinio algoritmo realizacija. Šis metodas yra iškviečiamas pasikeitus kameros vaizde esančiam kadrui. Kamera sugeba paimti 30 kadrų per sekundę, todėl šis metodas yra kviečiamas nuolatos, o skaičiavimai vyksta kiekvienam kadrui atskirai.

```
    public Mat onCameraFrame(CameraBridgeViewBase.CvCameraViewFrame inputFrame) {

        mRgba = inputFrame.gray();
        mRgbaF = inputFrame.rgba();

        if (MainSettings.scanning) {

            if (!MainSettings.resourcesLoaded) {

                File mCascadeFile = null;
                File eyesCascadeFile = null;

                InputStream is = null;
                InputStream is2 = null;

                if (MainSettings.method == ScanningMethod.HAAR) {
                    is = getResources().openRawResource(R.raw.haarcascade_frontalface_default);
                    is2 = getResources().openRawResource(R.raw.frontal_eyes);

                    File cascadeDir = getDir("cascade", Context.MODE_PRIVATE);

                    mCascadeFile = new File(cascadeDir, "haarcascade_frontalface_default.xml");
                    eyesCascadeFile = new File(cascadeDir, "frontal_eyes.xml");

                } else if (MainSettings.method == ScanningMethod.LBP) {
                    is = getResources().openRawResource(R.raw.lbpcascade_frontalface);
                    is2 = getResources().openRawResource(R.raw.frontal_eyes);

                    File cascadeDir = getDir("cascade", Context.MODE_PRIVATE);

                    mCascadeFile = new File(cascadeDir, "lbpcascade_frontalface.xml");
                    eyesCascadeFile = new File(cascadeDir, "frontal_eyes.xml");
                }


                loadResources(is, mCascadeFile);
                loadResources(is2, eyesCascadeFile);

                cascadeClassifier = new CascadeClassifier(mCascadeFile.getAbsolutePath());

                cascadeClassifier.load(mCascadeFile.getAbsolutePath());

                eyesClassifier = new CascadeClassifier(eyesCascadeFile.getAbsolutePath());

                eyesClassifier.load(eyesCascadeFile.getAbsolutePath());

                MainSettings.resourcesLoaded = true;
            }

            if (!cascadeClassifier.empty() && !eyesClassifier.empty()) {

                MatOfRect objects = new MatOfRect();

                MatOfRect eyesObjects = new MatOfRect();

                cascadeClassifier.detectMultiScale(mRgba, objects, SCALE_FACTOR, MIN_NEIGHBOURS,
0, new Size(400, 400), new Size(1000, 1000));

                Rect[] dataArray = objects.toArray();
```

```
                for (Rect rect : dataArray) {
                    Imgproc.rectangle(mRgbaF, rect.tl(), rect.br(), new Scalar(0, 255, 0, 255),
3);

                    eyesClassifier.detectMultiScale(mRgba, eyesObjects, SCALE_FACTOR,
MIN_NEIGHBOURS, 0, new Size(100, 100), new Size(1000, 1000));
                    Rect[] eyesDataArray = eyesObjects.toArray();

                    for (Rect eye : eyesDataArray) {
                        Imgproc.rectangle(mRgbaF, eye.tl(), eye.br(), new Scalar(255, 0, 0, 255),
3);
                    }

                    if (eyesDataArray.length != 0) {
                        final long duration = System.currentTimeMillis() - startTime;

                        Log.i("Duration time" + MainSettings.method.name(),
String.valueOf(duration));

                    }
                }

                startTime = System.currentTimeMillis();
            }
        }

        return mRgbaF;
}
```

### 3.5.3. OpenCV pagalbinis metodas

Tam, kad atpažintume objektus esančius vaizde, buvo panaudotas OpenCV bibliotekos detectMultiScale metodas. Šis metodas dirba su Haar cascade klasifikatoriais, kurie dar prieš naudojimą turi būti paruošti. Tai reiškia, jog klasifikatoriai yra apmokami paduodant daugybę pavyzdinių žmonių veidų ir žmonių akių paveikslėlių. Programoje yra užkraunamas klasifikatorius ir saugomas CascadeClassifier objekte. Jis turi load metodą, kuriam paduodamas .xml failas kaip klasifikatoriaus resursai. Pagrindinis detectMultiScale metodas priima tokius parametrus:

- Paveikslėlis – matrica, kurioje yra paveikslėlio taškai
- Objektai – stačiakampių vektoriai, kuriame saugomi atpažinti objektai
- Mažinimo koeficientas – reikšmė, kuri nusako kiek turi būti mažinamas paveikslėlis kiekviename mažinimo žingsnyje
- Minimalus kaimynų skaičius – reikšmė, kuri nusako kiek kiekvienas kandidatas turi turėti kaimynų, jog ją išlaikytų
- Minimalus dydis – minimalus objekto dydis
- Maksimalus dydis – maksimalus objekto dydis

Padavus atitinkamas reikšmes, metodas užpildo paduotą stačiakampių vektorių parametrą rastais objektais, kuriuos po to galime naudoti kontūro brėžimui aplink rastą objektą. Kaip pavyzdys

```
cascadeClassifier.detectMultiScale(mRgba, objects, 1.04, 4, 0, new Size(400, 400), new
Size(1000, 1000));
```

### 3.5.4. Haar cascade klasifikatorius

Šiam uždaviniui spręsti buvo panaudoti Haar cascade klasifikatoriai, kurio struktūrą aprašo žmonių veidų bruožus. Šis klasifikatorius buvo apmokytas su teigiamais ir neigiamais rezultatais. Tai reiškia, jog buvo paimta didelis kiekis žmonių veidų paveikslėlių ir didelis kiekis ne žmonių veidų paveikslėlių. Taip yra apmokamas klasifikatorius, kuris po to naudojamas realiuose uždaviniuose. Klasifikatoriaus reikšmės pateikiamos žemiau esančioje lentelėje (Žr. lentele 2).

**lentelė 2 HAAR klasifikatoriaus reikšmės**

| | |
|---|---|
| Aukštis | 24 |
| Plotis | 24 |
| Stadijos | 25 |

### 3.5.5. LBP klasifikatorius

Taip pat programėlė galima pasirinkti ir LBP klasifikatorių. Šio struktūra taip pat skirta žmonių veidams atpažinti. Šio struktūra labai panašį į Haar cascade, tačiau turi šiek tiek mažiau stadijų, o reikšmės saugos kaip binariniai skaičiai. Klasifikatoriaus reikšmės pateikiamos žemiau esančioje lentelėje (Žr. lentelė 3).

**lentelė 3 LBP klasifikatoriaus reikšmės**

| | |
|---|---|
| Aukštis | 24 |
| Plotis | 24 |
| Stadijos | 20 |

### 3.5.6. Klasifikatorių apmokinimas

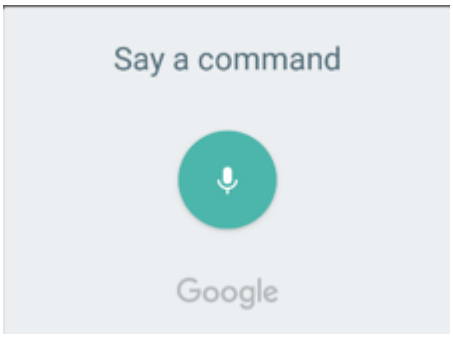
Norint apsimokyti savo klasifikatorių, kuris vėliau bus naudojamas įvairiems objektams atpažinti, iš pradžių reikės surinkti daugybę norimo objekto paveikslėlių. Rekomenduojama surinkti bent 1000 teigiamų ir 1000 neigiamų paveikslėlių. Norint supaprastinti rinkimo procesą, galima nufilmuoti tam tikrą objektą ir iškarpyti kadrus iš vaizdo medžiagos. Paruošiant paveikslėlius reikia apkarpyti taip, jog būtų matomas tik objektas, be jokių kitų pašalinių objektų. Taip pat apmokinant klasifikatorių reikalingi ir paveikslėliai, kurie neturi norimo atpažinti objekto. Dar geriau jei paveikslėliai būtų labai panašus į tuos, kurie turi objektą tik be pačio objekto.

OpenCV biblioteka siūlo dvi skirtingas programas („haar training“ ir „traincascade“), kurios skirtos apmokyti savo klasifikatorius. „Traincascade“ programėlė leidžia naudoti keletą gijų vienu metu ir šitaip sumažina apmokymui skirtą laiką. Šis procesas yra reiklus tiek kompiuterio resursams,

tiek laikui. Nustatyta, jog klasifikatorių LBP klasifikatorių apmokinimas trunka net iki 10 kartų trumpiau nei HAAR cascade [24].

### 3.6. Balso atpažinimas

Programėlė buvo skirta žmonėms, kurie turi regos sutrikimų, todėl vartotojo sąsaja neturi įprastinio valdymo lietimu. Visas valdymas yra balso komandomis. Iš esmės yra keletas balso komandų, kurias pasakius, yra vykdomi vienokie ar kitokie veiksmai. Šiam funkcionalumui buvo panaudotas standartinis Android SDK balso atpažinimo funkcionalumas. Tą patį funkcionalumą naudoja ir Google paieška, kuri leidžia balsu atlikti įvairias paieškas internete (Žr. pav. 14). Vaizdo atpažinimo sistemoje panaudota balso komandų surinkimas ir rezultatų pateikimas. Tai reiškia, jog vartotojui pasakius keletą žodžių, jie grąžinami sąrašo pavidalu, o juose jau galima ieškoti konkrečių žodžių ir taip priskirti norimus veiksmus.




**pav. 14 Balso komandos įvedimas**

# 4. TYRIMO IR EKSPERIMENTINĖ DALIS

## 4.1. Tyrimų aprašas

Vaizdo atpažinimo sistemose svarbiausia yra kaip tiksliai yra gebama atpažinti tam tikrus objektus. Tokiose sistemose tikslumas yra pagrindinis ir kritiškiausias kriterijus. Dėl šios priežasties bus ištirta bent 10 skirtingų nuotraukų, kuriose yra skirtingų bruožų žmonių veidai. Taip pat bus atliktas tyrimas ir realioje aplinkoje esant šviesiam ir blankiam apšvietimui su realiu autoriaus veidu. Tiriant ir testuojant veidų atpažinimą bus keičiami šie parametrai:

- Keičiama paveikslėlio didinimo reikšmė programėlėje
- Keičiamas minimalus kaimynų skaičius
- Keičiamas klasifikatorius

### 4.1.1. Tyrimų paskirtis

**lentelė 4 tyrimų sąrašas**

| Tyrimo numeris | Paskirtis |
|---|---|
| 1 | Nustatyti vaizdo atpažinimo tikslumą |
| 2 | Nustatyti vaizdo atpažinimo spartą |

### 4.1.2. Tyrimui naudojama įranga

Tyrimui yra naudojama tokia aparatūrinė bei aparatūrinė įranga

- LG G Flex 2 išmanusis telefonas
    - 6.0.1 Android operacinė sistema
    - Qualcomm MSM8994 Snapdragon 810 procesorius
    - 2gb RAM atmintis
    - 13 MP galinė vaizdo kamera

### 4.1.3. Tyrime naudotų nuotraukų specifikacija

Žemiau pateiktoje lentelėje (lentelė 5) pateiktos tyrime naudotų nuotraukų charakteristikos.

**lentelė 5 nuotraukų specifikacijos**

| Nuotraukos Nr. | Raiška | Bendras taškų skaičius | DPI |
|---|---|---|---|
| 1 | 400 x 533 | 213,200 | 150 |
| 2 | 650 x 366 | 237,900 | 300 |
| 3 | 920 x 518 | 476,560 | 96 |
| 4 | 207 x 243 | 50,301 | 96 |
| 5 | 287 x 380 | 109,060 | 96 |
| 6 | 668 x 800 | 534,400 | 96 |
| 7 | 183 x 275 | 50,325 | 96 |
| 8 | 222 x 227 | 50,394 | 96 |
| 9 | 225 x 225 | 50,625 | 96 |
| 10 | 615 x 626 | 384,990 | 96 |

## 4.2. Eksperimentas Nr. 1

Siekiant nustatyti vaizdo atpažinimo tikslumą, buvo pasirinkta 10 skirtingų žmonių veidų nuotraukų. Eksperimento metu buvo naudojamas tiek Haar, tiek LBP klasifikatorius. Norint išsiaiškinti kokios turi būti paveikslėlio didinimo ir minimalaus kaimynų skaičiaus reikšmės, reikšmės buvo keičiamos nuo mažiausios įmanomos iki didžiausios veikiančios, tol kol bus randamas optimaliausios.

### 4.2.1. Tikslumo tyrimo rezultatai

Šiame skyrelyje apžvelgti tikslumo tyrimo rezultatai.

#### 4.2.1.1. Tyrimas su nuotraukomis

Šiame skyrelyje  apžvelgiami tyrimo rezultatai, kurie buvo atlikti su dešimt skirtingų nuotraukų (*prieduose 33 pav., 34 pav., 35 pav. 36 pav., 37 pav., 38 pav., 39 pav., 40 pav., 41 pav., 42 pav.)*. Nuotraukos buvo rodomos ekrane, o naudojant Android programėlę ir įrenginio kamerą, skenuojamos ir apdorojamos.

4.2.1.1.1. Tyrimas keičiant didinimo žingsnio reikšmę

Lentelėse (*lentelė 13 , lentelė 14., lentelė 15 ., lentelė 16., lentelė 17., lentelė 18., lentelė 19., lentelė 20., lentelė 21., lentelė 22.)*. pateikti tyrimo rezultatai, kuriuose detaliai pateikti skirtingų nuotraukų vaizdo atpažinimo su skirtingomis didinimo žingsnio reikšmėmis elgsena. Lentelėse yra

sužymėtos didinimo žingsnio reikšmės ir išskirti HAAR ir LBP klasifikatoriai. Kiekvieno tyrimo metu buvo žymimi simboliai, jei veidas buvo atpažintas, lentelėje pažymima (+) simboliu, jei veidas nebuvo atpažintas, lentelėje buvo pažymima (-) simboliu. Toks pat žymėjimas galioja ir stulpelyje „Klaidos atpažinime“. Jei bandant atpažinti žmogaus veidą pasitaikydavo netikslumų, kaip akys ne toje vietoje ar užfiksuoti du veidai, tačiau pats tiriamas veidas būdavo atpažintas, tuomet laikome, jog atpažinimas įvyko, bet yra klaidų. 16 pav. pateiktas pavyzdys kuomet veidas buvo atpažintas, tačiau jame pasitaikė klaidų. Du stačiakampiai parodo, jog programėlė mato žmogaus dvi poras akių.

Atliekant tyrimus su pirmąją nuotrauka (pav. 33), buvo užfiksuota, jog prie visų nuo 1,01 iki 1,50 didinimo žingsnio reikšmių, naudojant HAAR cascade klasifikatorių, žmonių veidai buvo atpažinti. Šią reikšmę pakeitus į 2, klasifikatorius nesugebėjo atpažinti žmogaus veido. Taip pat atpažinimo metu pasitaikė klaidų prie 1,01 ir 1,5 didinimo žingsnio reikšmės. Tai parodo, jog esant labai mažai ir labai didelei didinimo žingsnio reikšmei, atpažinimo tikimybė sumažėja bent keliais kartais. LBP klasifikatorius pirmą nuotrauką atpažino su visomis didinimo žingsnio reikšmėmis. Verta paminėti, jog visi atpažinimai buvo su klaidomis ir netikslumais (Žr. lentelė 13).

Tyrimas su antrąja nuotrauka (pav. 34) parodė identiškus rezultatus kaip ir pirmąją. Keičiant didinimo žingsnį nuo 1,01 iki 1,5, veidas nuotraukoje buvo atpažintas. Klaidos pasitaikė didinimo žingsnio reikšmėms esant 1,01 ir 1,5. Esant didinimo žingsniui 2, veidas nuotraukoje nebuvo atpažintas. LBP klasifikatorius elgėsi kaip ir pirmos nuotraukos tyrimo metu, 1,01 – 2 didinimo žingsnių intervale jis atpažino veidą nuotraukoje. Vis dėlto visi atpažinimai rodė netikslumus (Žr. lentelė 14).

Naudojant tyrime trečią nuotrauką (pav. 35), tyrimo rezultatai buvo lygiai tokie patys kaip ir pirmais dviem atvejais. Didinimo žingsnio reikšmei esant 1,01 – 1,5 HAAR cascade klasifikatorius atpažino veidą, o esant reikšmei 2, veidas nebuvo atpažintas. Netikslumai pasikartojo reikšmėms esant 1,01 ir 1,5. LBP klasifikatorius atpažino veidą prie visų tyrime naudotų reikšmių, tačiau visais atvejais kartojosi netikslumo klaidos (Žr. lentelė 15).

Ketvirtos nuotraukos (pav. 36) tyrimas parodė tą pačią tendenciją. HAAR cascade klasifikatorius atpažino nuotrauką keičiant didinimo žingsnio reikšmę nuo 1,01 iki 1,5. Kaip ir su ankstesnėmis nuotraukomis, klaidos pasireiškė esant mažiausiai įmanomai reikšmei 1,01 ir 1,5. Reikšmei esant 2, veidas nebuvo atpažintas. LBP klasifikatorius atpažino visus veidus reikšmėms esant 1,01 – 2. Klaidos aptiktos esant visoms tyrime naudotoms didinimo žingsnio reikšmėms (Žr. lentelė 16).

Atliekant tyrimą su penkta nuotrauka (pav. 37), rezultatai užfiksuoti lygiai tokie patys kaip ir su pirmomis keturiomis nuotraukomis. HAAR cascade klasifikatorius nesugebėjo atpažinti veido tik esant didinimo žingsnio reikšmei 2. Klaidos pasikartojo reikšmėms esant 1,01 ir 1,5. LBP

klasifikatorius veidus atpažino esant reikšmėms 1,01-2. Klaidos pasikartojo esant visoms tyrime galimoms reikšmėms (Žr. lentelė 17).

Šeštosios nuotraukos (pav. 38) tyrimas parodė kiek kitokius rezultatus. Veidą atpažinti pavyko reikšmėms esant 1,01 – 1,5. Vis dėlto lyginant su kitomis nuotraukomis, buvo užfiksuotos klaidos didinimo reikšmei esant 1,01, 1,02, 1,03, ir 1,5. Šioje tyrimo stadijoje galime pastebėti, jog šeštoji nuotrauka išsiskiria savo taškų skaičiumi. Ją sudaro 534,400 taškų (Žr. lentelė 5). Dėl mažo didinimo žingsnio reikšmės ir dėl didelio kiekio taškų, bandant atpažinti nuotrauką, atsirado netikslumų ir klaidų. LBP klasifikatorius pasikeitimų lyginant su prieš tai aprašytomis nuotraukomis neparodė. Esant visoms, skirtingoms, tyrime naudotomis reikšmėmis, veidas buvo atpažintas ir visi atpažinimai turėjo netikslumų (Žr. lentelė 18).

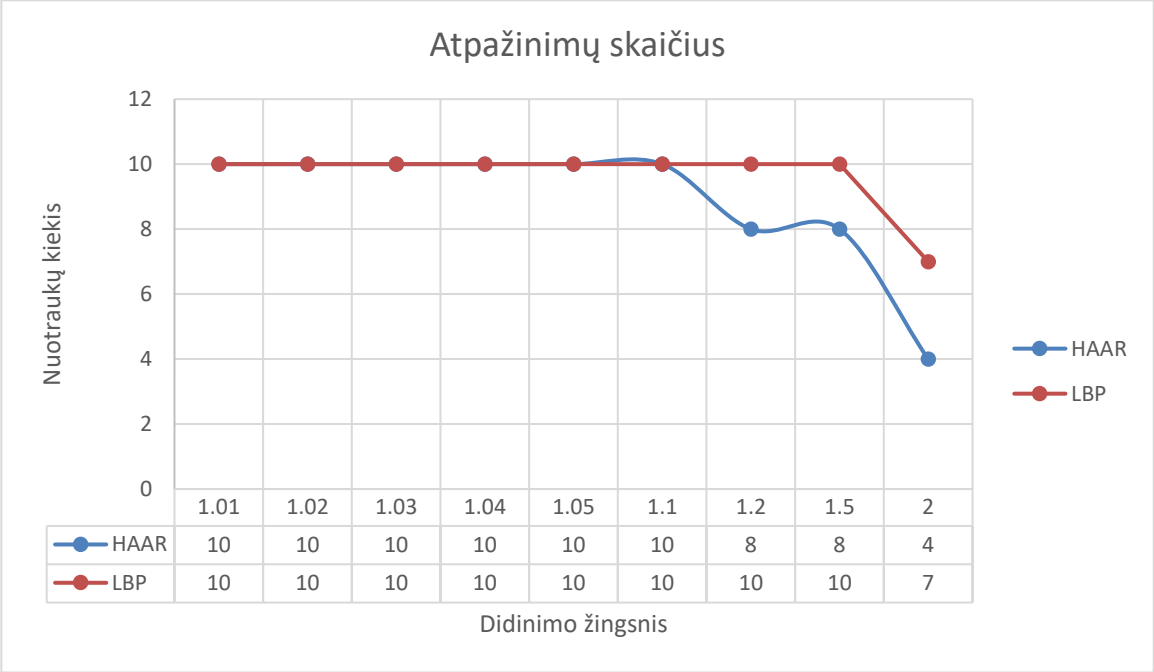
Atliekant tyrimą su septinta nuotrauka, (pav. 39) HAAR cascade klasifikatorius atpažino veidą nuotraukoje, didinimo žingsnio reikšmei esant 1,01 – 1,5. Jei prieš tai atlikti tyrimai su 1-5 nuotraukomis parodė netikslumus prie didinimo žingsnio reikšmės 1,01 ir 1,5, tai šios nuotraukos tyrimas parodė netikslumus esant 1,02 ir 1,03 reikšmėms. Septinta nuotrauka pasižymi labai mažu taškų skaičiumi - 50,325 (Žr. lentelė 5). Galime daryti prielaidą, jog dėl šios priežasties, esant mažoms didinimo reikšmėms, buvo užfiksuotos atpažinimo klaidos. Taip pat tiriant LBP klasifikatoriaus tikslumą, pastebėta, jog skirtingai nei prieš tai atliktuose tyrimuose, esant didinimo žingsnio reikšmei 2, veidas nebuvo atpažintas. Esant reikšmėms 1,01 - 1,5, veidas nuotraukoje buvo atpažintas (Žr. lentelė 19)

Aštuntos nuotraukos (pav. 40) tyrimas užfiksavo HAAR cascade vaizdo atpažinimą esant didinimo žingsniui 1,01 – 1,1. Visi šie atpažinimai įvyko be jokių klaidų. Verta paminėti, jog prie reikšmių 1,2, 1,5 ir 2, veidas nuotraukoje atpažintas nebuvo. LBP klasifikatorius veidą atpažino keičiant didinimo žingsnio reikšmę nuo 1,01 iki 1,5. Visi šie atpažinimai turėjo klaidų. Reikšmę nustačius 2, LBP klasifikatorius veido neatpažino (Žr. lentelė 20).

Naudojant tyrime devintą nuotrauką (pav. 41), pasikartojo analogiška situacija kaip ir aštuntos nuotraukos tyrime. HAAR Cascade klasifikatorius prie didinimo žingsnio reikšmių 1,01 – 1,1 sugebėjo tiksliai, be klaidų, atpažinti veidą nuotraukoje. Esant reikšmėms 1,2 – 2, klasifikatorius aptikti veido paveikslėlyje nesugebėjo. LBP klasifikatorius atpažino visus veidus su klaidomis, išskyrus prie reikšmės 2, kuomet veidas nebuvo atpažintas (Žr. lentelė 21).

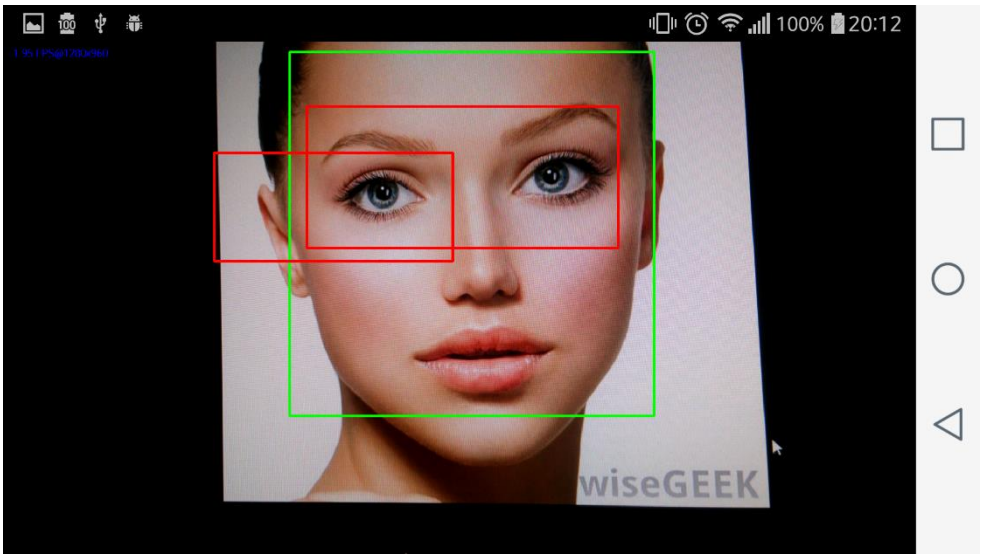
Dešimtos nuotraukos (pav. 41) tyrimas parodė kiek kitokius rezultatus. Paskutinis nuotraukos pavyzdys pasižymėjo dideliu taškų skaičiumi - 384,990 (Žr. lentelė 5). HAAR cascade klasifikatorius sugebėjo atpažinti veidą esant visoms skirtingoms tyrime naudojamoms reikšmėms. Klaidų pasitaikė tik vieninteliu atveju, kuomet didinimo reikšmė buvo 1,5. LBP klasifikatorius taip pat sugebėjo atpažinti visus veidus. Verta paminėti, jog visi atpažinimai įvyko su klaidomis ir netikslumais (Žr. lentelė 22).

15. pav pavaizduoti bendri atpažinimų rezultatai. Tai reiškia, jog veido nuotraukoje veidas buvo atpažintas. Tai gali reikšti, jog buvo netikslumų ar nukrypimų. Kaip pavyzdys būtų daugiau nei kelių akių atpažinimas ar veido atpažinimas kitoje vietoje nei iš tiesų (Žr. pav. 16). Šioje tyrimo fazėje svarbiausia buvo, jog būtent rodomas veidas būtų atpažintas. Taip buvo atliktas tyrimas su dešimt nuotraukų, bei su skirtingomis devyniomis didinimo žingsnio reikšmėmis. Buvo pradėta nuo mažiausios įmanomos 1,01 reikšmės ir didinamos mažiausia galima reikšme 0,01 iki 1,05. Po to ši reikšmė buvo didinama didesniais žingsniais ir žiūrima kaip elgiasi algoritmas bei pati programa. Tyrimas parodė, jog nuo 1,01 iki 1,1 žingsnio, naudojant tiek HAAR, tiek LBP klasifikatorių, atpažinti buvo visi veidai. Pakeitus didinimo žingsnio reikšmę į 1,2, pastebėta, jog naudojant HAAR klasifikatorių, šis nebesugebėjo atpažinti kelių nuotraukų. Tas pats kartojosi ir su 1,5 didinimo žingsnio reikšme. Padidinus žingsnį iki 2, LBP klasifikatorius atpažino 7 iš 10 nuotraukų, o HAAR tik 4 iš 10.



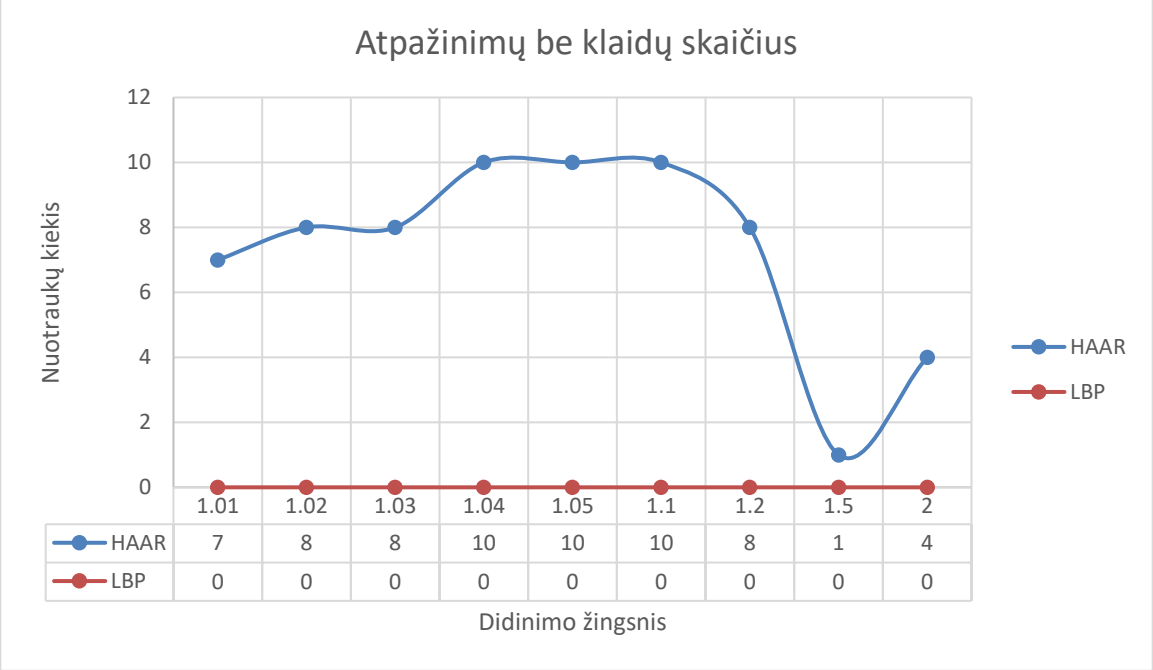

| | 1.01 | 1.02 | 1.03 | 1.04 | 1.05 | 1.1 | 1.2 | 1.5 | 2 |
|---|---|---|---|---|---|---|---|---|---|
| HAAR | 10 | 10 | 10 | 10 | 10 | 10 | 8 | 8 | 4 |
| LBP | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 7 |

**pav. 15 Atpažinimų skaičius didinant žingsnį**




**pav. 16 Atpažinimas su klaidomis**

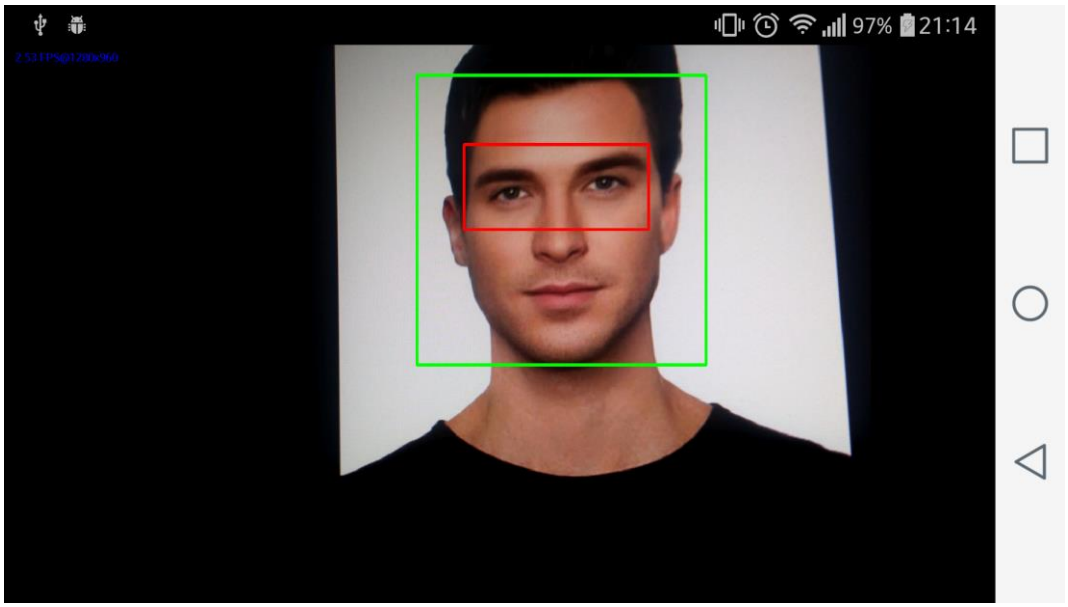
17 pav. parodytas tyrimas, kuriame vykdytas tas pats vaizdo atpažinimas su tomis pačiomis nuotraukomis, tačiau pavaizduoti rezultatai, kurie buvo įvykdyti be klaidų. Tai reiškia, jog jei buvo nuotrauka atpažinta, bet su klaidomis, tai ji nebuvo užfiksuota šiame paveiksle. Šis grafikas ryškiai parodo LBP klasifikatoriaus netikslumą. Visais atpažinimo atvejais, klasifikatorius darė klaidas ir rodė netikslumus. Tuo tarpu HAAR klasifikatorius geriausiai pasirodė esant 1,04, 1,05 ir 1,1 didinimo žingsnio reikšmėms. Esant šioms reikšmėms, visos nuotraukos buvo atpažintos be jokių klaidų ir netikslumų.




| | 1.01 | 1.02 | 1.03 | 1.04 | 1.05 | 1.1 | 1.2 | 1.5 | 2 |
|---|---|---|---|---|---|---|---|---|---|
| HAAR | 7 | 8 | 8 | 10 | 10 | 10 | 8 | 1 | 4 |
| LBP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**pav. 17 Atpažinimų be klaidų skaičius keičiant didinimo žingsnį**

18 pav. užfiksuotas normalus programėlės veikimas, kuomet rodomoje nuotraukoje yra atpažintas žmogaus veidas. Žalias kontūras žymi žmogaus veidą, o raudonas kontūras žymi žmogaus akis.




**pav. 18 Nuotrauka, kurioje užfiksuotas veidas be klaidų**

4.2.1.1.2. Tyrimas keičiant minimalų kaimynų skaičiaus reikšmę

Šiame tyrime buvo atlikti analogiški veiksmai, tačiau šį kartą buvo keičiama minimalus kaimynų skaičius. Didinimo žingsnio reikšmė pasirinkta 1,01.

Atliekant tyrimus su pirmąją nuotrauka (pav. 33), užfiksuota, jog keičiant minimalaus kaimynų skaičiaus reikšmę nuo 2 iki 50, veido atpažinimas, naudojant HAAR cascade klasifikatorių, nuotraukoje buvo sėkmingas. Esant reikšmei 100, atpažinti veido nepavyko. LBP klasifikatorius esant visoms, tyrime naudojamoms, reikšmėms, atpažino veidą nuotraukoje. Verta paminėti, jog LBP klasifikatorius darė klaidas prie visų reikšmių, išskyrus 50 ir 100. (Žr. lentelė 23).

Tyrimas su antrąja nuotrauka (pav. 34) parodė kiek geresnius rezultatus nei su pirmąja. HAAR cascade klasifikatorius atpažino veidą kintant reikšmėms nuo 2 iki 50, o visi atpažinimai įvyko be jokių klaidų. Vieninteliu atveju veido atpažinti nepavyko, kuomet minimalaus kaimynų skaičiaus reikšmė buvo 100. LBP klasifikatorius atpažino veidą esant reikšmėms 2 – 100. Taikant reikšmę 50 ir 100, buvo užfiksuotos klaidos. (Žr. lentelė 24).

Naudojant tyrime trečią nuotrauką (pav. 35), užfiksuota, jog esant minimalaus kaimynų skaičiaus reikšmei 2-50, HAAR cascade klasifikatorius atpažino nuotraukoje esantį žmogaus veidą. Vieninteliu atveju esant reikšmei 2, buvo aptiktos atpažinimo klaidos. Prie reikšmės 100, veidas atpažintas nebuvo. LBP klasifikatoriaus elgsena lygiai tokia pati kaip ir antros nuotraukos tyrime. (Žr. lentelė 25).

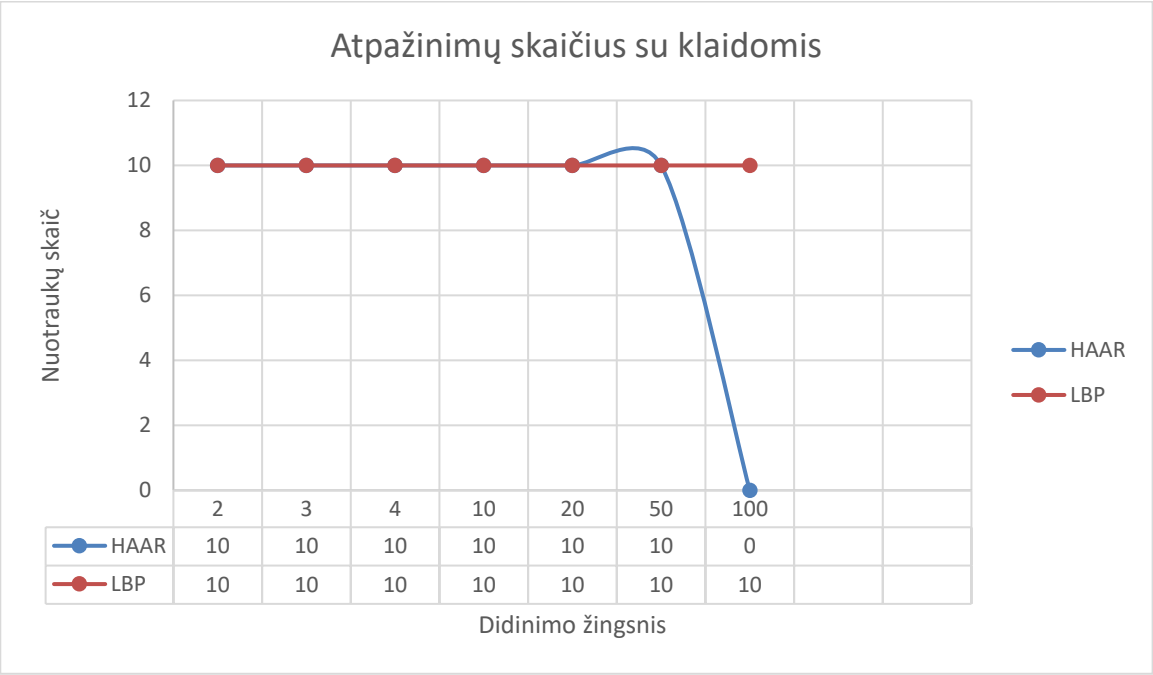
Ketvirtos nuotraukos (pav. 36) tyrimas parodė labai panašius rezultatus. HAAR Cascade klasifikatorius puikiai veikė prie minimalaus kaimynų skaičiaus reikšmės 2-50. Visi veido atpažinimai įvyko be klaidų. Reikšmei esant 100, veidas nebuvo atpažintas. LBP klasifikatorius atpažino nuotraukas prie visų reikšmių, tiesa, reikšmėms esant 2-20, buvo užfiksuotos klaidos (Žr. lentelė 26).

Atliekant tyrimą su penkta nuotrauka (pav. 37), gauti lygiai tokie patys rezultatai kaip ir ketvirtos nuotraukos tyrime. (Žr. lentelė 27).

Tyrimas su šeštąja nuotrauka (pav. 38) parodė tą pačią tendenciją kaip ir ketvirtos, bei penktos nuotraukos tyrimas. HAAR cascade klasifikatorius vėl atpažino veidą be jokių klaidų prie minimalaus kaimynų skaičiaus reikšmės 2-50. Nustačius reikšmę 100, veidas nuotraukoje atpažintas nebuvo. LBP klasifikatorius atpažino veidus prie visų tyrime naudotų reikšmių, tačiau be klaidų tik esant 50 ir 100. (Žr. lentelė 28).

Atliekant tyrimą su septinta nuotrauka (pav. 39), aštunta (pav. 40), devinta (pav. 41) ir dešimta (pav. 42) užfiksuoti tie patys rezultatai kaip ir tyrime su šešta nuotrauka. (Žr. lentelė 29, lentelė 30, lentelė 31, lentelė 32).

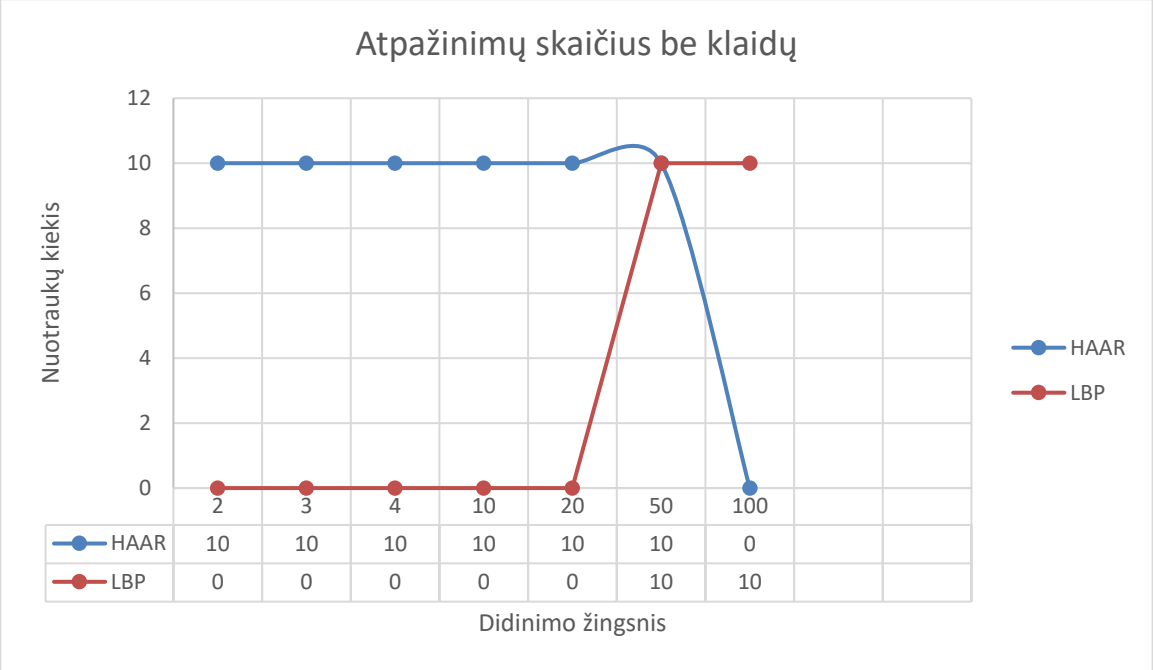
19 pav. matosi bendri tyrimo rezultatai. Rezultatai buvo labai panašūs viso tyrimo metu. Keičiant reikšmę nuo 2 iki 50, visais atvejais abu klasifikatoriai atpažino veidus. Vis dėlto pakeitus reikšmę į 100, HAAR klasifikatorius nebesugebėjo atpažinti veido, o LBP parodė teigiamą rezultatą.




| | 2 | 3 | 4 | 10 | 20 | 50 | 100 |
|---|---|---|---|---|---|---|---|
| HAAR | 10 | 10 | 10 | 10 | 10 | 10 | 0 |
| LBP | 10 | 10 | 10 | 10 | 10 | 10 | 10 |

**pav. 19 Atpažinimų skaičius su klaidomis keičiant didinimo žingsnį**

Lyginant rezultatus, kuomet buvo užfiksuoti veidai su klaidomis ir be jų, pasirodo akivaizdūs tikslumo skirtumai tarp dviejų klasifikatorių. Jei keičiant reikšmes nuo 2 iki 20 abu klasifikatoriai atpažįsta veidus, tai LBP tai daro su klaidomis. Tik padidinus reikšmę iki 50, LBP klasifikatorius atpažino tiksliai, o HAAR klasifikatorius esant reikšmei 100 nesugebėjo to padaryti.




**pav. 20 Atpažinimas be klaidų**

### 4.2.1.2. Tyrimas su realiais pavyzdžiais

Šiame skyrelyje atpažinimas buvo vykdomas realiomis sąlygomis. Tai reiškia, jog pasinaudojus Android įrenginiu, programėle ir įrenginio vaizdo kamera, buvo skenuojamas realus vaizdas, šiuo atveju kuriame yra autoriaus veidas. Vienu atveju tyrimas atliktas esant menkam apšvietimui. Tyrimai buvo atliekami tamsiu paros metu, kambario aplinkoje.

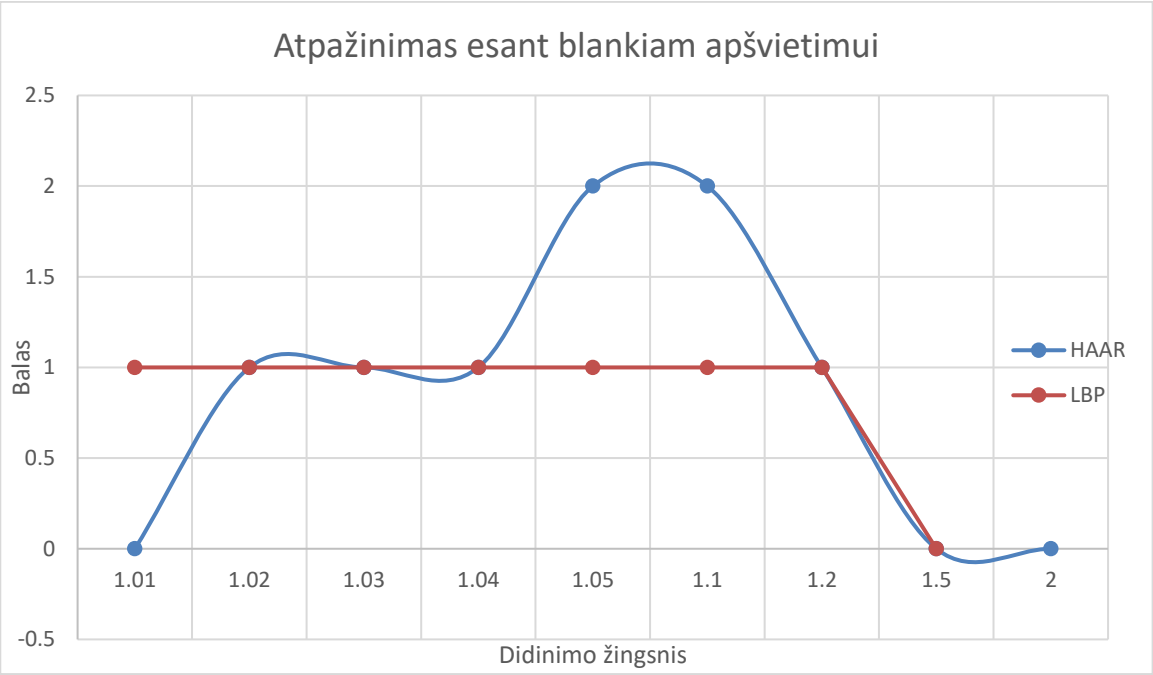
4.2.1.2.1. Tyrimas keičiant didinimo žingsnio reikšmę

21 pav. pateikti tyrimo rezultatai. Tyrimo procedūra prilygsta prieš tai vykusiems tyrimams su nuotraukomis. Atlikus tyrimą, pastebėta, jog programėlė veikia kiek kitaip nei dirbant su nuotraukomis. Grafiko Y ašyje sužymėti tikslumui skirti įverčiai. Įverčių reikšmės:

- 0 – neatpažinta
- 1 – atpažinta su klaidomis
- 2 – atpažinta be klaidų

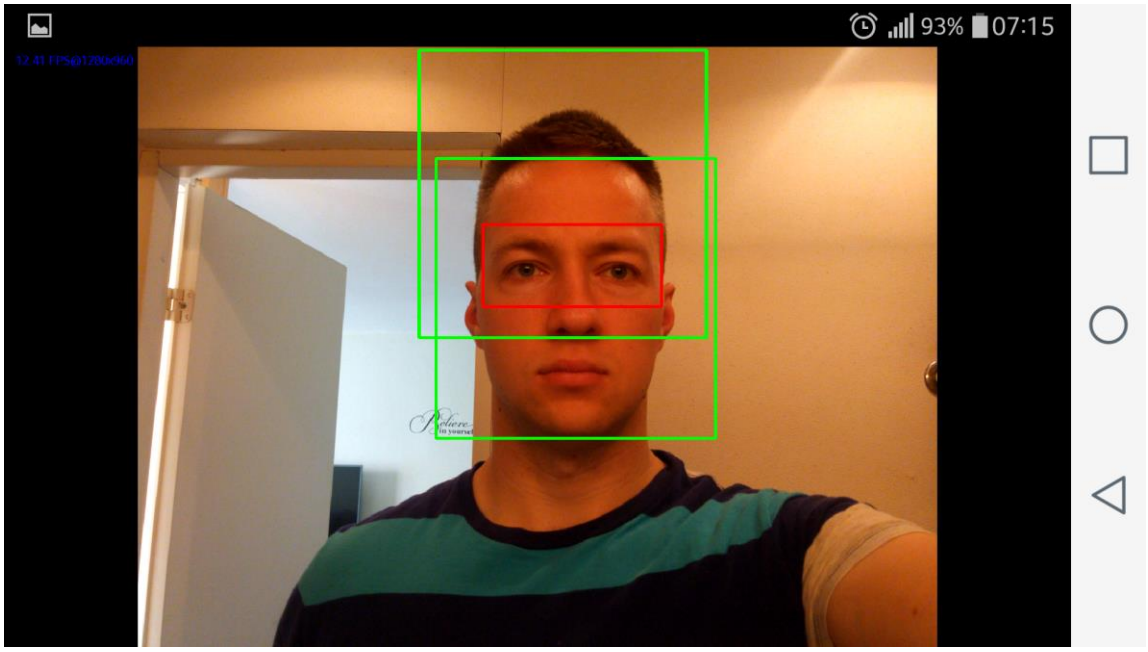
Naudojant HAAR klasifikatorių, tiksliausiai atpažinimas vyko didinimo žingsnio reikšmei esant 1,05 ir 1,1. Kitoms reikšmėms esant veidas buvo atpažintas su klaidomis, o prie 1,01, 1,5 ir 2

reikšmės veidas buvo visiškai neatpažintas. LBP klasifikatorius atpažino dažniau, tačiau su netikslumais ir klaidomis. Keičiant didinimo žingsnio reikšmę nuo 1,01 iki 1,2, veidas buvo atpažintas su klaidomis, o pakeitus į 1,5 ir 2, programėlė atpažinti veido nesugebėjo.



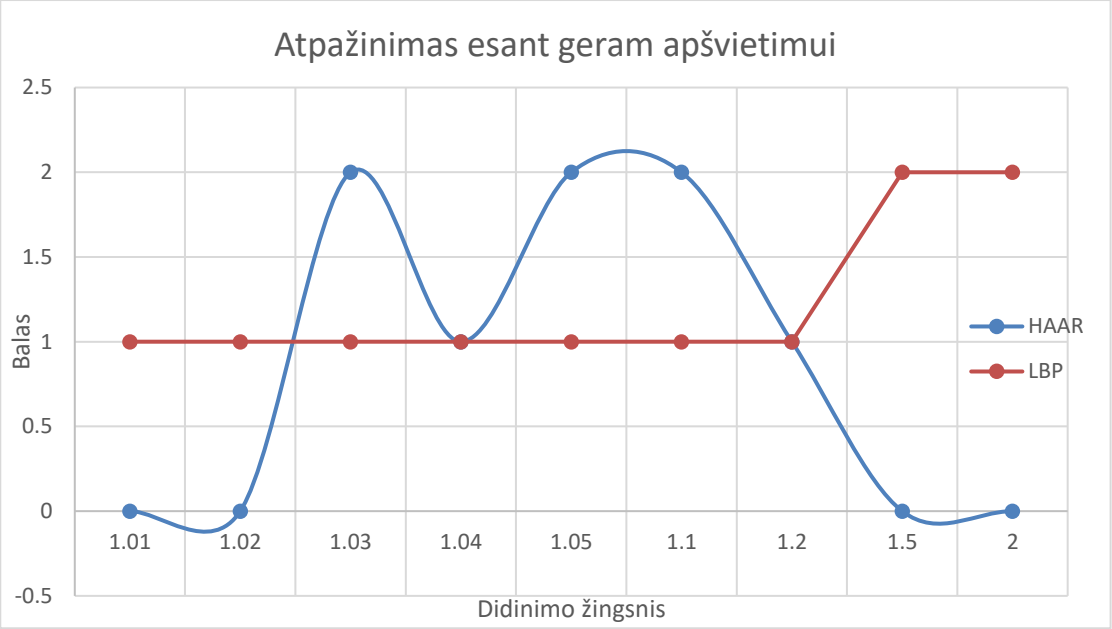

**pav. 21 Atpažinimas esant blankiam apšvietimui**

22 pav. užfiksuotas veido atpažinimas, kuomet didinimo reikšmė 1,02. Matome, jog veidą su akimis pavyko atpažinti, tačiau taip pat rodomas dar vienas žalias stačiakampis. Tai reiškia, jog įvyko atpažinimo klaida ir programėlė mato du veidus



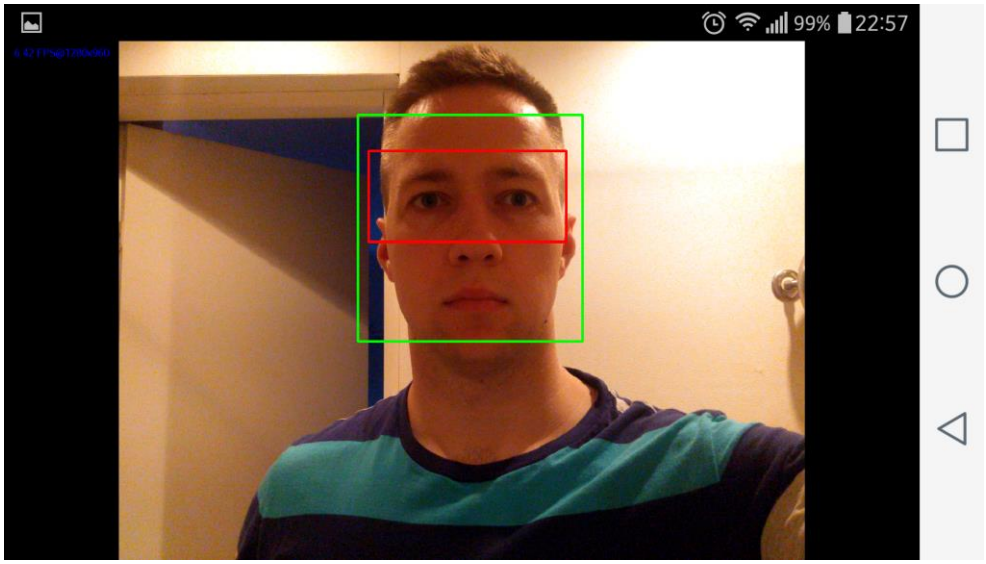

**pav. 22 Atpažinimas gerame apšvietime su klaidomis esant didinimo žingsniui 1,02**

Kiek kitokie užfiksuoti rezultatai esant geram apšvietimui. 23 pav. matome, jog naudojant HAAR klasifikatorių, programėlė elgėsi panašiai kaip ir esant blankiam apšvietimui. Geriausi rezultatai užfiksuoti esant 1,03, 1,05 ir 1,1 didinimo žingsnio reikšmėms.



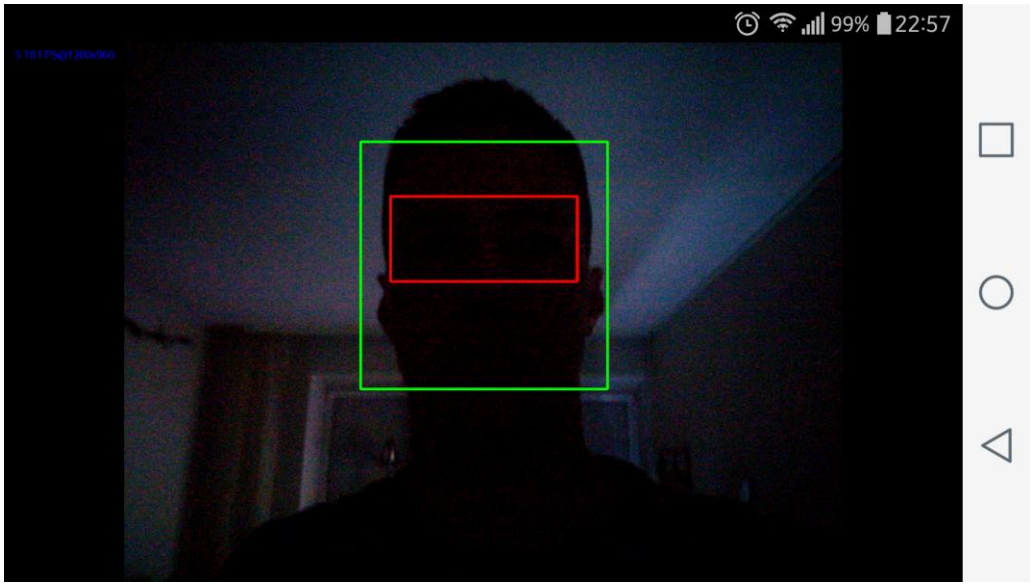

**pav. 23 Atpažinimas esant geram apšvietimui**

24 pav. užfiksuotas tikslus vaizdo atpažinimas. Šis pavyzdys užfiksuotas, kuomet didinimo žingsnio reikšmė buvo 1,05.




**pav. 24 Atpažinimas be klaidų gerame apšvietime esant didinimo žingsniui 1,05**

25 pav. taip pat užfiksuotas tikslus veido atpažinimas netgi esant labai blankiam apšvietimui. Čia didinimo žingsnio reikšmė 1,05.



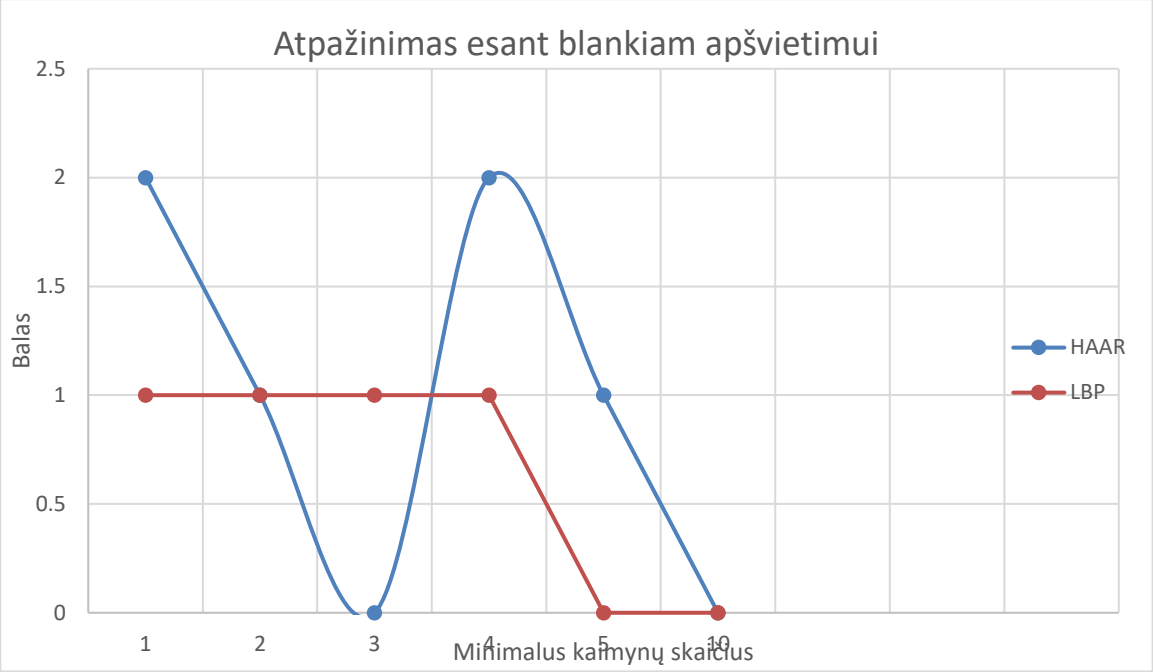

**pav. 25 Atpažinimas be klaidų blankiame apšvietime esant didinimo žingsniui 1,05**

Tyrimas keičiant minimalų kaimynų skaičiaus reikšmę

Taip pat tyrimas realioje aplinkoje buvo atliktas ir keičiant minimalų kaimynų skaičiaus reikšmę. Čia Y ašyje sužymėti tikslumui nustatyti įverčiai:

- 0 – neatpažinta
- 1 – atpažinta su klaidomis
- 2 – atpažinta be klaidų

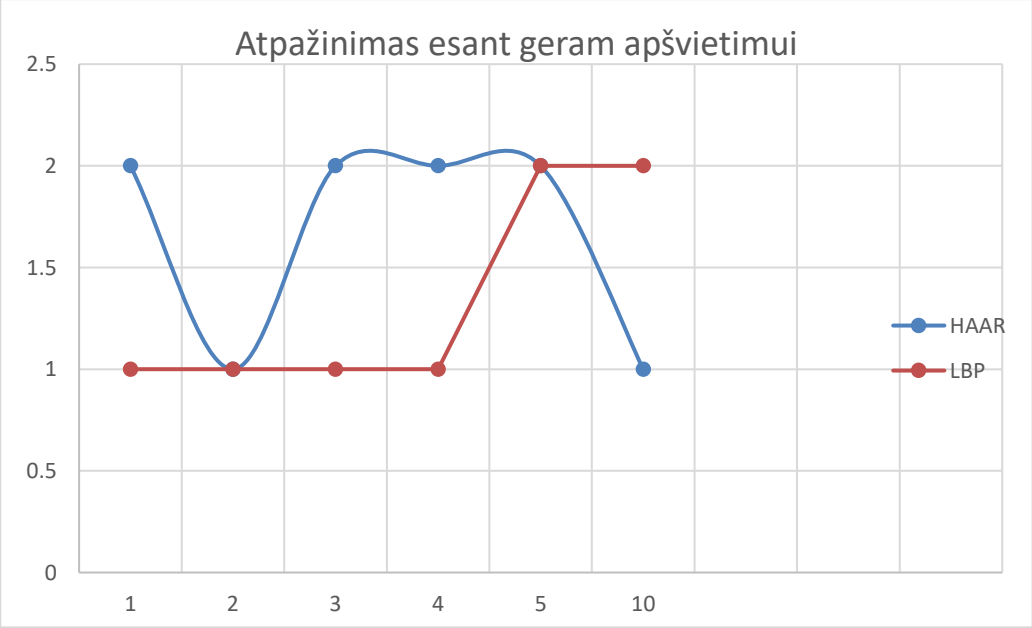
Tyrime didinimo žingsnio reikšmė pasirinkta 1,05. Šis atliktas tyrimas parodė įdomius rezultatus, HAAR klasifikatoriaus pagalba tiksliausiai pavyko atpažinti minimalaus kaimynų skaičiaus reikšmei esant 1 ir 4. Šiais atvejais nepasirodė jokios klaidos ir veidas buvo atpažintas tiksliai. Reikšmėms esant 2 ir 5 veidai buvo atpažinti su klaidomis. Pakeitus reikšmę į 10, veidas nebuvo atpažintas. Kiek kitokia situacija su LBP klasifikatoriumi. Šis sugebėdavo atpažinti veidą esant reikšmei nuo 1 iki 4. Vis dėlto kartu su atpažinimu pasirodydavo klaidos ir netikslumai. Esant reikšmėms 5 ir 10 LBP nesugebėjo atpažinti veido realioje aplinkoje. 26 pav. užfiksuoti rezultatai esant blankiam aplinkos apšvietimui.




**pav. 26 Atpažinimas esant blankiam apšvietimui**

Analogiški tyrimai buvo atliekami ir esant geram apšvietimui, kuomet natūralioje aplinkoje buvo panaudotas dirbtinis šviesos šaltinis. 27 pav. matome, jog HAAR klasifikatorius elgėsi kiek geriau nei esant blankiam apšvietimui. Esant minimaliam kaimynų skaičiui 1, 3, 4, 5 vaizdo atpažinimas įvyko be jokių klaidų. Kiek blogesni rezultatai buvo prie reikšmių 2 ir 10.

LBP klasifikatorius skirtingai nei HAAR, parodė prastesnį atpažinimo tikslumą. Esant reikšmėms nuo 1 iki 4, veidas buvo atpažintas su netikslumais ir klaidomis. Pakeitus reikšmę į 5 ir 10, atpažinimai, turintys klaidų, dingo.




**pav. 27 Atpažinimas esant geram apšvietimui**

#### 4.2.1.3. Sunaudojami resursai

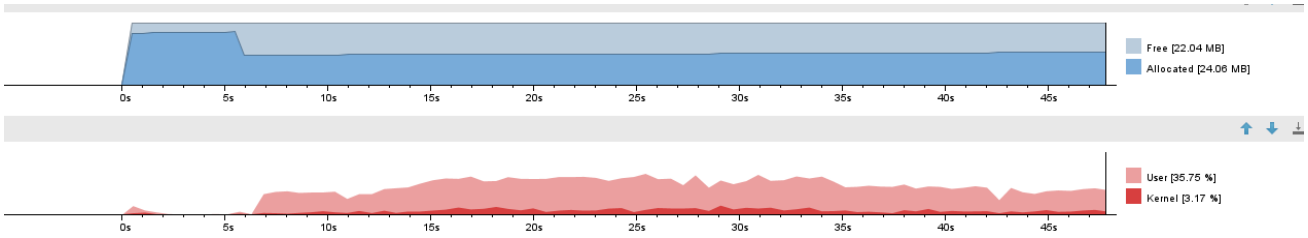
Taip pat tyrimo metu buvo ir pabandyta pasižiūrėti į sunaudojamus programėlės resursus. Pastebėta, jog procesoriaus sunaudojimas ypač aptinkant objektus yra didžiausias. Aptikus objektą, tyrimui naudoto įrenginio procesorius buvo naudojamas 54% (Žr. lentelė 6). Procesas naudoja daug procesoriaus resursų, nes jam per vieną sekundę reikia apdoroti apie 30 kadrų per sekundę, jį apdoroti ir išvesti rezultatus. Vieno kadro dydis 1280 x 720 taškų.

- Didinimo skaičius – 1,5
- Minimalus kaimynų skaičius – 4

**lentelė 6 resursų sunaudojimas**

| **Būsena** | **RAM atmintis** | **CPU sunaudojimas** |
|---|---|---|
| Įjungta kamera | 24mb | 20% |
| Įjungtas skenavimas | 25mb | 35% |
| Aptinkamas objektas | 26mb | 54% |

28 pav. užfiksuoti resursų pokyčiai. Aukštesnėje grafoje matoma operatyvios atminties sunaudojimas. Žemesnėje grafoje matoma centrinio procesoriaus sunaudojimas.




**pav. 28 Sunaudojami resursai**

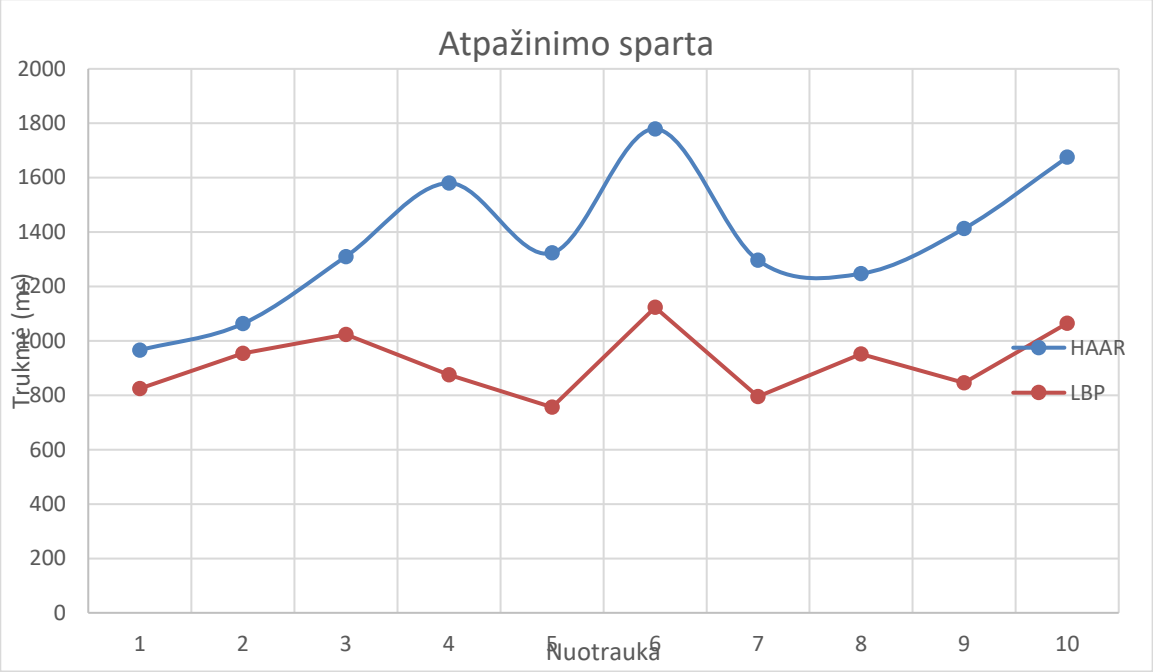
### 4.3. Eksperimentas Nr. 2

Tyrimo metu išsiaiškinus tiksliausią metodą ir tiksliausias didinimo, bei minimalaus kaimynų skaičiaus reikšmes, šios buvo naudojamos spartos tyrime. Tam, jog pasiektume galimai tiksliausiai ir sparčiausiai veikiančią vaizdo atpažinimo sistemą, šiame tyrime buvo šiek tiek koreguojama didinimo žingsnio reikšmė. Tyrimas buvo atliekamas su nuotraukomis ir realiu vaizdu.

- Didinimo žingsnis 1,01
- Minimalus kaimynų skaičius 4

29 pav. pavaizduoti spartos tyrimo rezultatai. Matome, jog HAAR klasifikatorius užtrunka šiek tiek daugiau laiko ir nusileidžia sparta LBP klasifikatoriui. Tiesa, šiame tyrime buvo nekreipiama

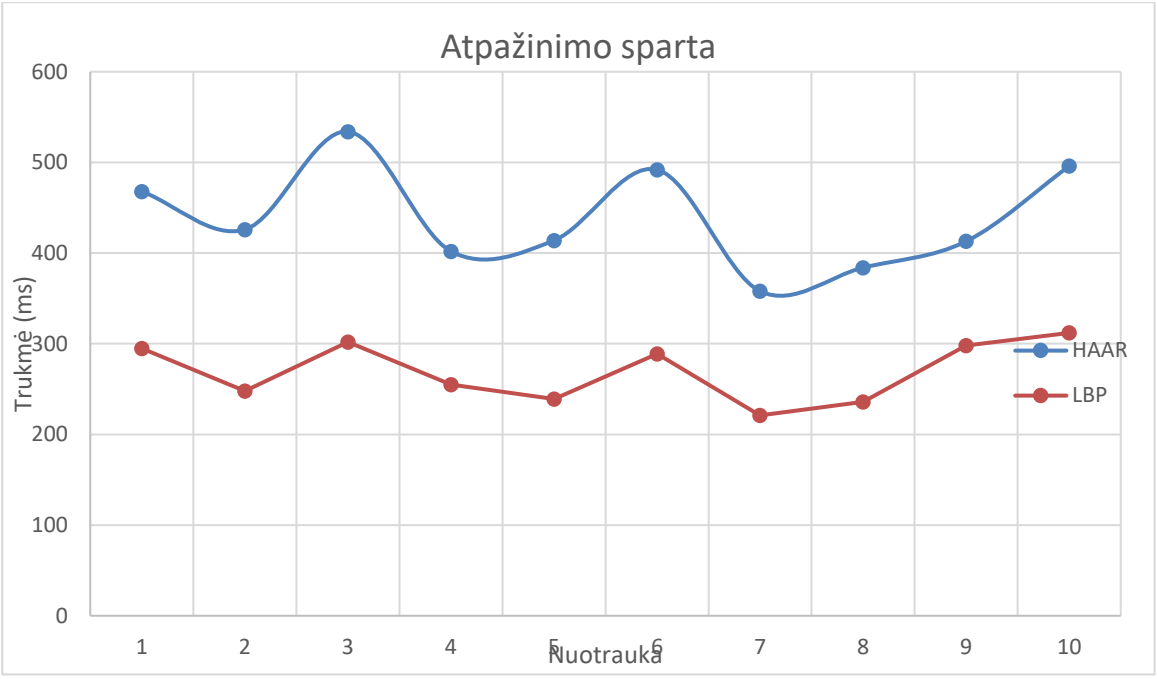
dėmesio į klaidas ir netikslumus, o laikas paimtas kuomet buvo atpažintas veidas. HAAR klasifikatorius trumpiausiai užtruko su pirma nuotrauka, veidui atpažinti prireikė 966 ms. Ilgiausiai užtruko su šešta nuotrauka – 1779 ms. Atitinkamai LBP klasifikatorius geriausius ir blogiausius rezultatus parodė su tomis pačiomis nuotraukomis (824ms ir 1123ms). Visi rezultatai pateikti 35 lentelėje.




**pav. 29 Atpažinimo sparta didinimo žingsnio reikšmei esant 1,01**

- Didinimo žingsnis 1,05
- Minimalus kaimynų skaičius 4

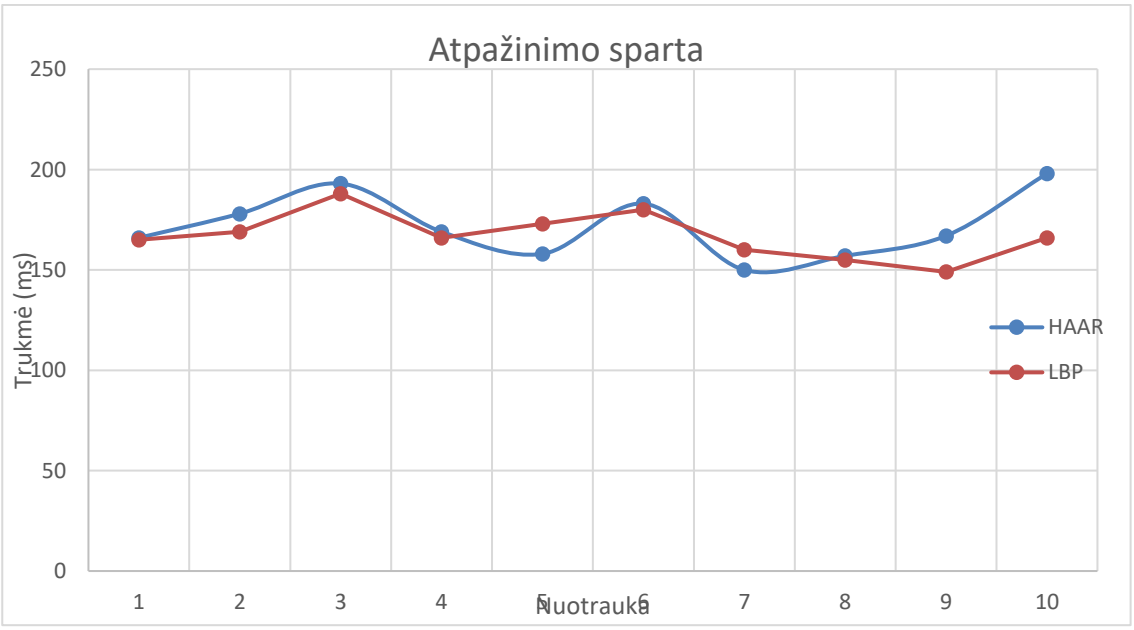
Spartesnis vaizdo atpažinimas užfiksuotas pakeitus didinimo žingsnio reikšmę iki 1,05. Rezultatai pavaizduoti pav. 30. Vis dar HAAR klasifikatorius spartos atžvilgiu buvo lėtesnis nei LBP. HAAR klasifikatorius ilgiausiai užtruko su trečia nuotrauka, veidui atpažinti prireikė 534ms. Sparčiausias rezultatas užfiksuotas su 7-a nuotrauka – 358 ms. LBP klasifikatorius taip pat sparčiausiai atpažino 7-ą nuotrauką, tą padarė per 221 ms. Vidutiniškai vaizdo atpažinimas naudojant 1,05 didinimo žingsnio reikšmę paspartėjo 3,11 karto naudojant HAAR cascade klasifikatorių. Naudojant LBP klasifikatorių, vaizdo atpažinimas paspartėjo 3,42 karto.




**pav. 30. Atpažinimo sparta didinimo žingsnio reikšmei esant 1,05**

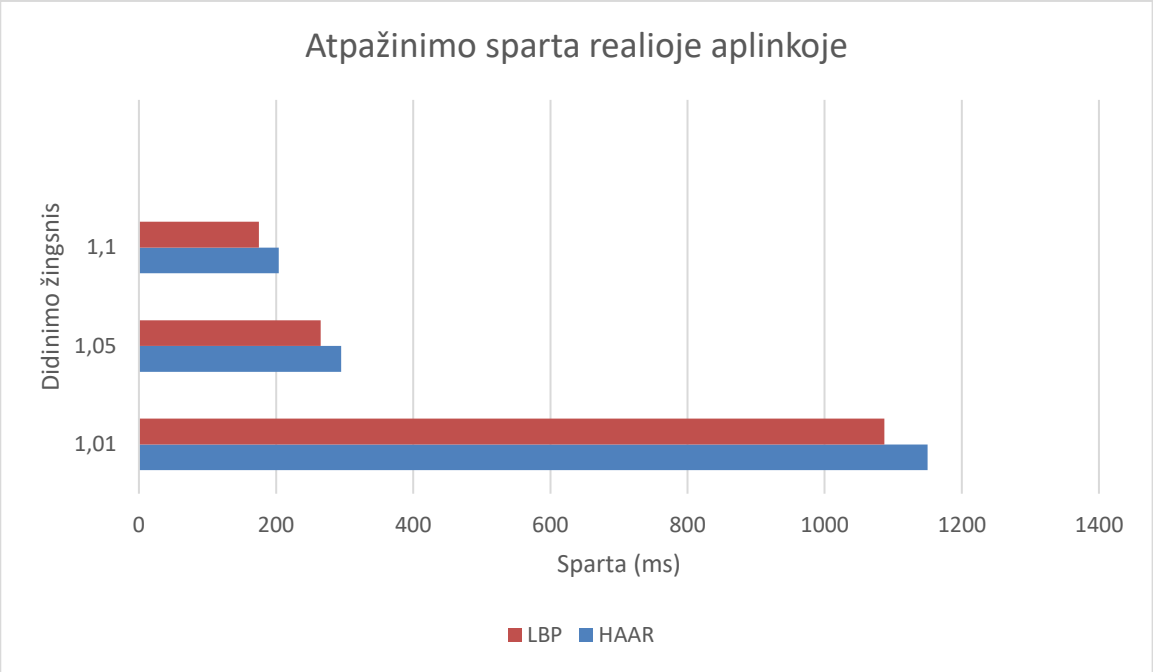
- Didinimo žingsnis 1,10
- Minimalus kaimynų skaičius 4

Dar padidinus didinimo žingsnio reikšmę iki 1,10, rezultatai beveik supanašėjo, o kartais HAAR klasifikatorius suveikdavo sparčiau nei LBP (Žr. pav.31). Taip pat verta paminėti, jog abiejų klasifikatorių trukmė ženkliai paspartėjo. Lyginant su ankstesne reikšme. HAAR klasifikatorius geriausią rezultatą parodė su septinta nuotrauka – 150ms. LBP klasifikatorius su devinta nuotrauka – 149. Nuo pradinės 1,01 reikšmės, padidinus iki 1,1, HAAR cascade klasifikatorius paspartėjo 7,98 karto, o LBP 5,44 karto.




**pav. 31 Atpažinimo sparta didinimo žingsnio reikšmei esant 1,1**

Tyrimas buvo atliktas ir realioje aplinkoje. Buvo panaudotos tos pačios didinimo reikšmės 1,01, 1,05 ir 1,1. Rezultatai buvo labai panašūs kaip ir apdorojant nuotraukas. Kuo didesnė reikšmė, tuo spartesnis atpažinimas. Palyginimui HAAR klasifikatoriui prie 1,01 reikšmės prireikė 1150ms, o esant 1,1 reikšmei – 204ms. Atitinkamai elgiasi ir LBP klasifikatorius, prie 1,01 reikšmės 1087ms, o prie 1,1 – 175ms. Didinant reikšmę yra sumažinamas skaičiavimų kiekis, tačiau padarius ją per didelę, kaip jau parodė ankstesni tyrimai, prarandamas tikslumas. Situacija ženkliai keičiasi pakeitus didinimo žingsnio reikšmę į 1,5 ir didesnę, kuomet užfiksuotas net 90 % mažesnis tikslumas (Žr. lentelė 7).




**pav. 32 Atpažinimo sparta realioje aplinkoje**

### 4.4. Rezultatų palyginimai

Žemiau pateiktose lentelėse surašyti atpažintų veidų kiekiai prie skirtingo didinimo žingsnio reikšmių (Žr. lentelė 7, lentelė 8) ir prie skirtingos minimalaus kaimynų skaičiaus reikšmės (Žr. lentelė. 9, lentelė 10). Šioje vietoje yra lyginami tik atpažinimai, kurie įvyko be jokių klaidų ir netikslumų. Tai reiškia, jog buvo tiksliai atpažintas žmogaus veidas. Lentelėse pateikti tyrimo su nuotraukomis rezultatai. Tolimesni skaičiavimai apskaičiuoja procentinę atpažinimo išraišką. Pirmoji apskaičiuoja HAAR Cascade klasifikatoriaus atpažinimą prie skirtingų didinimo žingsnio reikšmių.

$$\frac{0.7 + 0.8 + 0.8 + 1 + 1 + 1 + 0.8 + 0.1 + 0.4}{9} = 0.73$$

Žemiau esantys skaičiavimai paskaičiuoja HAAR Cascade klasifikatoriaus atpažinimo vidutinį tikslumą keičiant minimalaus kaimynų skaičiaus reikšmę.

$$\frac{1 + 1 + 1 + 1 + 1 + 1 + 0}{7} = 0.85$$

Žemiau esantys skaičiavimai paskaičiuoja LBP klasifikatoriaus atpažinimo tikslumą keičiant minimalaus kaimynų skaičiaus reikšmę.

$$\frac{0 + 0 + 0 + 0 + 0 + 1 + 1}{7} = 0.28$$

Žemiau esantys skaičiavimai paskaičiuoja kiek kartų tiksliau atpažįsta vienas ar kitas klasifikatorius.

$$\frac{0.73 + 0.85}{2} \div \frac{0 + 0.28}{2} = 5.64$$

Šiuo atveju gauta, jog HAAR cascade klasifikatorius, be klaidų atpažįsta net 5,64 karto tiksliau nei LBP klasifikatorius.

**lentelė 7 atpažinimų rezultatų palyginimai HAAR Cascade klasifikatoriumi keičiant didinimo žingsnio reikšmę**

| Didinimo žingsnio reikšmė | Atpažintų veidų kiekis be klaidų | Atpažintų veidų kiekio procentinė išraiška |
|---|---|---|
| 1,01 | 7 | 70 % |
| 1,02 | 8 | 80 % |
| 1,03 | 8 | 80 % |
| 1,04 | 10 | 100 % |
| 1,05 | 10 | 100 % |
| 1,1 | 10 | 100% |
| 1,2 | 8 | 80 % |
| 1,5 | 1 | 10 % |
| 2 | 4 | 40 % |

**lentelė 8 atpažinimų rezultatų palyginimai LBP klasifikatoriumi keičiant didinimo žingsnio reikšmę**

| Didinimo žingsnio reikšmė | Atpažintų veidų kiekis be klaidų | Atpažintų veidų kiekio procentinė išraiška |
|---|---|---|
| 1,01 | 0 | 0 % |
| 1,02 | 0 | 0 % |
| 1,03 | 0 | 0 % |
| 1,04 | 0 | 0 % |
| 1,05 | 0 | 0 % |
| 1,1 | 0 | 0% |
| 1,2 | 0 | 0 % |
| 1,5 | 0 | 0 % |
| 2 | 0 | 0 % |

**lentelė 9 atpažinimų rezultatų palyginimai HAAR klasifikatoriumi keičiant minimalaus kaimynų skaičiaus reikšmę**

| Minimalaus kaimynų skaičiaus reikšmė | Atpažintų veidų kiekis be klaidų | Atpažintų veidų kiekio procentinė išraiška |
|---|---|---|
| 2 | 10 | 100 % |
| 3 | 10 | 100 % |
| 4 | 10 | 100 % |
| 10 | 10 | 100 % |
| 20 | 10 | 100 % |
| 50 | 10 | 100% |
| 100 | 0 | 0 % |

**lentelė 10 atpažinimų rezultatų palyginimai HAAR klasifikatoriumi keičiant minimalaus kaimynų skaičiaus reikšmę**

| Minimalaus kaimynų skaičiaus reikšmė | Atpažintų veidų kiekis be klaidų | Atpažintų veidų kiekio procentinė išraiška |
|---|---|---|
| 2 | 0 | 0 % |
| 3 | 0 | 0 % |
| 4 | 0 | 0 % |
| 10 | 0 | 0 % |
| 20 | 0 | 0 % |
| 50 | 10 | 100% |
| 100 | 10 | 100 % |

Vidutinė atpažinimo sparta buvo gauta sudėjus visas vienos didinimo žingsnio reikšmės atpažinimo spartos rezultatų reikšmes (Žr. lentelė 35, lentelė 36, lentelė 37, lentele 38, 39) ir padalinus iš tyrimo objektų (10) skaičiaus. Vidutinė atpažinimo sparta taikant HAAR Cascade klasifikatorių pateikta lentelėje (Žr. lentelė 11). LBP klasifikatoriaus rezultatai pateikti kitoje lentelėje (Žr. lentelė 12). Pakeitus didinimo žingsnio reikšmę iš 1,01 į 1,05, gauname: $\frac{1365}{438} = 3.11$. Tai reiškia, jog vaizdo atpažinimas vidutiniškai paspartėja 3,11 karto. Pakeitus reikšmę iš 1,01 į 1,1, gauname: $\frac{1365}{171} = 7.98$. Toks rezultatas parodo, jog vaizdo atpažinimas gali būti paspartintas net 7,98 karto.

**lentelė 11 vidutinė atpažinimo sparta taikant HAAR Cascade klasifikatorių**

| Didinimo žingsnio reikšmė | Vidutinė atpažinimo sparta (ms) |
|---|---|
| 1,01 | 1365 |
| 1,05 | 438 |
| 1,10 | 171 |

Apdorojant vaizdą ir naudojant LBP klasifikatorių, pakeitus didinimo žingsnio reikšmę iš 1,01 į 1,05, gauname: $\frac{921}{269} = 3.42$. Toks pokytis leidžia paspartinti vaizdo atpažinimą 3,42 karto. Pakeitus reikšmę iš 1,01 į 1,1, gauname: $\frac{921}{169} = 5.44$. Toks rezultatas parodo, jog vaizdo atpažinimas gali būti paspartintas net 5,44 karto.

**lentelė 12 vidutinė atpažinimo sparta taikant LBP klasifikatorių**

| Didinimo žingsnio reikšmė | Vidutinė atpažinimo sparta (ms) |
|---|---|
| 1,01 | 921 |
| 1,05 | 269 |
| 1,10 | 169 |

Palyginant HAAR Cascade ir LBP klasifikatorius tarpusavyje, gauname tokius rezultatus:

$$\frac{1365}{921} = 1.48$$

$$\frac{438}{171} = 2.56$$

$$\frac{171}{169} = 1.011$$

Paskaičiavus vidurkį gauname:

$$\frac{1.48 + 2.56 + 1.011}{3} = 1.68$$

Pažiūrėjus į visus rezultatus, matome, jog LBP klasifikatorius veikia sparčiau. Vidutiniškai LBP klasifikatorius yra 1,68 karto spartesnis.

# 5. IŠVADOS

- Atlikus literatūros analizę ir pasiaiškinus skirtingų metodų ir algoritmų veikimo principus, jų privalumus bei trūkumus, buvo pasirinktas HAAR cascade klasifikatorių algoritmas, nes jis leidžia turėti iš anksto paruoštus apmokytus klasifikatorius.
- Atlikus eksperimentą su programine įranga, buvo nustatyta, jog vaizdo atpažinimas priklauso nuo didinimo žingsnio reikšmės, minimalaus kaimynų skaičiaus reikšmės ir pasirinkto klasifikatoriaus. Keičiant didinimo žingsnio reikšmes nuo 1,01 iki 2 buvo gauti 70 % ir 40% tikslumas.
- Atlikus analizę ir palyginus tris skirtingas vaizdo atpažinimo bibliotekas, buvo pasirinkta OpenCV, nes ši vienintelė yra tinkama Android operacinei sistemai ir turi realizuotą analizės metu išsirinktą HAAR cascade klasifikatorių algoritmą.
- Atlikus tyrimą su skirtingais vaizdo atpažinimo klasifikatoriais ir skirtingomis didinimo žingsnio bei minimalaus kaimynų skaičiaus reikšmėmis, buvo nustatyta, jog didinimo žingsnio reikšmė daro didžiulę įtaką atpažinimo spartai. Tyrimo metu išsiaiškinta, jog didinant reikšmę, algoritmo veikimo sparta didėja net iki 7,98 karto. Optimaliausia pasirinkta reikšmė – 1,05. Nuo pradinės programos realizacijos, pakeitus reikšmę pavyko paspartinti vaizdo atpažinimą apie 927 ms, o tai yra net 3,11 karto. Pasirenkant LBP klasifikatorių, atpažinimas įvyksta 68% sparčiau, nes vaizdo apdorojimui ir skaičiavimams naudojami tik sveiki skaičiai.
- Atlikus tyrimą buvo išsiaiškinta, jog tinkamiausios klasifikatorius yra HAAR, nes įvertinus jo tikslumą keičiant didinimo žingsnio ir minimalaus kaimynų skaičiaus reikšmę, jis yra tikslesnis už LBP 5,64 karto.

## 6. LITERATŪRA


[1] Optical Character Recognition (OCR) [interaktyvus] [žiūrėta 2016 01 26]. Prieiga per: http://www.microscan.com/en-us/technology/OpticalCharacterRecognition.aspx

[2] Recognition of object instances practical [interaktyvus] [žiūrėta 2016 01 26]. Prieiga per: http://www.robots.ox.ac.uk/~vgg/practicals/instance-recognition/index.html

[3] Towards Total Scene Understanding [interakyvus] [žiūrėta 2016 01 27]. Prieiga per http://vision.stanford.edu/projects/totalscene/

[4] Kristen Grauman, Bastian Leibe 2011 Visual Object Recognition ISBN: 9781598299694 [interaktyvus]. [žiūrėta 2017-04-30]. Prieiga per: https://books.google.lt/books?id=lAQGBvdm3UsC&lpg=PP1&dq=visual%20recognition&hl=lt&pg=PP1#v=onepage&q=visual%20recognition&f=false

[5] Affine Covariant Region Detectors [interaktyvus]. 2007 [žiūrėta 2017-05-23] Prieiga per: http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html

[6] Harris Corner Detector & Scale Invariant Feature Transform (SIFT) [interaktyvus]. [žiūrėta 2017-05-23] Prieiga per: http://slideplayer.com/slide/8695490/

[7] Ezzeddine Zagrouba An efficient image-mosaicing method based on multifeature matching [interaktyvus] 2009 [žiūrėta 2017-05-23] Prieiga per: https://www.researchgate.net/figure/225236416_fig4_Fig-4-Interest-points-extraction-with-ImpHarris-detector

[8] Krystian Mikolajczyk, Cordelia Schmid An Affine Invariant Interest Point Detector [interaktyvus]. Prancūzija 2002, [žiūrėta 2017-05-01] Prieiga per: SpringerLink

[9] James Fishbaugh Blob Detection by Scale-Space Filtering [interaktyvus]. [žiūrėta 2017-05-21] Prieiga per: http://www.cs.utah.edu/~jfishbau/advimproc/project1/

[10] Robert Fisher Simon Perkins Ashley Walker Erik Wolfart Laplacian/Laplacian of Gaussian [interaktyvus]. 2003 [žiūrėta 2017-04-28] Prieiga per: http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm

[11] David G. Lowe, International Journal of Computer Vision University of British Columbia IJCV 2004

[12] Ravimal Badara Bag-of-Features Descriptor on SIFT Features with OpenCV [interaktyvus]. 2014 [žiūrėta 2017-05-23] Prieiga per: https://www.codeproject.com/Articles/619039/Bag-of-Features-Descriptor-on-SIFT-Features-with-O

[13] Jacob Toft Pedersen Study group SURF: Feeature detection & description [interaktyvus] 2011 [žiūrėta 2017-05-03] Prieiga per: http://cs.au.dk/~jtp/SURF/report.pdf

[14] Caleb Woodruff Feature Detection and Matching [interakyvus]. 2013 [žiūrėta 2017-05-23] Prieiga per: https://courses.cs.washington.edu/courses/cse576/13sp/projects/project1/artifacts/woodrc/index.htm
[15] Sander Soo Object detection using Haar-cascade Classifier Institute of Computer Science, University of Tartu [interaktyvus]. [žiūrėta 2017-05-04] Prieiga per: http://ds.cs.ut.ee/Members/artjom85/2014dss-course-media/Object%20detection%20using%20Haar-final.pdf
[16] Jo Chang-yeon Face Detection using LBP features [interaktyvus] 2008 Prieiga per: http://cs229.stanford.edu/proj2008/Jo-FaceDetectionUsingLBPfeatures.pdf
[17] Mati Pietikäinen Local Binary Patterns [interaktyvus] 2010. Prieiga per: http://www.scholarpedia.org/article/Local_Binary_Patterns
[18] CamFind Search The Physical World [interaktyvus] [žiūrėta 2017-05-05] Prieiga per: http://camfindapp.com/
[19] Google, „Google Googles“ [žiūrėta 2017-05-14]. Prieiga per internetą: http://www.google.com/mobile/goggles/
[20] OpenCV 3 Computer Vision Application Programming Cookbook – Third Edition [interaktyvus] Prieiga per: https://www.packtpub.com/application-development/opencv-3-computer-vision-application-programming-cookbook-third-edition
[21] Integrating Vision Toolkit [žiūrėta 2017-04-28]. Prieiga per internetą: http://ivt.sourceforge.net/
[22] Visualization Toolkit [žiūrėta 2017-05-12]. Prieiga per internetą: http://www.vtk.org/
[23] Martin Abadi, Ashish Agarwal, Paul Berham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowich, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, ´ Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, ´ Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems [interaktyvus] 2015 [žiūrėta 2017-05-22]. Prieiga per: https://arxiv.org/pdf/1603.04467.pdf
[24] Thorsten Ball Train Your Own OpenCV HAAR Classifier [interaktyvus]. 2013 [žiūrėta 2017-05-22]. Prieiga per: http://coding-robin.de/2013/07/22/train-your-own-opencv-haar-classifier.html

## 7. TERMINŲ IR SANTRUMPŲ ŽODYNAS

- **JAVA –** objektinio programavimo kalba
- **Android –** išmaniųjų įrenginių operacinė sistema
- **APP –** programėlė skirta išmaniesiems įrenginiams
- **HAAR Cascade –** paveikslėlio požymių klasifikatorius
- **LBP** *(Local Binary Pattern)* – paveikslėlio požymių klasifikatorius, kurio struktūrą sudaro binariniai skaičiai
- **CPU** *(Central Processing Unit)* – pagrindinis įrenginio procesorius, kuris atlieka pagrindinius veiksmus ir operacijas
- **RAM** *(Random Access Memory)* – operatyvinė įrenginio atmintis, kurioje saugomas laikini programai veikti reikalingi duomenys
- **OCR** *(Optical Character Recognition)* – optinis simbolių atpažinimas
- **OpenCV –** biblioteka skirta vaizdo medžiagos analizei
- **SIFT** *(Scale Invariant Feature Transform)* – vaizdo atpažinimo algoritmas, kuris išskiria paveiksliuko savybes
- **SURF** *(Speed Up Robost Features)* – vaizdo atpažinimo algoritmas, kuris pasižymi tuo, jog nepriklauso nuo paveikslėlio dydžio
- **Regionas –** paveikslėlio vieta, kurioje yra išskirtiniai paveikslėlio požymiai
- **2D –** dvimatis (sudaro ilgis ir plotis)
- **3D –** trimatis (sudaro ilgis, plotis ir aukštis)
- **Eigen erdvė –** Eigen vektorių rinkinys
- **A-posteriori –** informacija iš anksčiau
- **Objektas –** kūnas, kurį sudaro linijos ir spalvos
- **Raiška –** paveikslėlį sudarančių taškų suma
- **Taškas** *(pixel)* – pats mažiausias paveikslėlio vienetas (spalva kvadratėlis).
- **DPI** *(dots per inch)* – taškų kiekis viename colyje

# 8. PRIEDAI

## 8.1. Tyrimo rezultatai

Čia pateikti tyrimo rezultatų priedai kaip rezultatų lentelės.

### 8.1.1. Atpažinimo tyrimas

**lentelė 13 pirmos nuotraukos atpažinimo tyrimas**

| | Haar cascade | | LBP klasifikatorius | |
|---|---|---|---|---|
| Didinimo žingsnis | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 1,01 | + | + | + | + |
| 1,02 | + | - | + | + |
| 1,03 | + | - | + | + |
| 1,04 | + | - | + | + |
| 1,05 | + | - | + | + |
| 1,10 | + | - | + | + |
| 1,20 | + | - | + | + |
| 1,50 | + | + | + | + |
| 2,00 | - | - | + | + |

**lentelė 14 antros nuotraukos atpažinimo tyrimas**

| | Haar cascade | | LBP klasifikatorius | |
|---|---|---|---|---|
| Didinimo žingsnis | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 1,01 | + | + | + | + |
| 1,02 | + | - | + | + |
| 1,03 | + | - | + | + |
| 1,04 | + | - | + | + |
| 1,05 | + | - | + | + |
| 1,10 | + | - | + | + |
| 1,20 | + | - | + | + |
| 1,50 | + | + | + | + |
| 2,00 | - | - | + | + |

**lentelė 15 trečios nuotraukos atpažinimo tyrimai**

| | Haar cascade | | LBP klasifikatorius | |
|---|---|---|---|---|
| Didinimo žingsnis | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 1,01 | + | + | + | + |
| 1,02 | + | - | + | + |
| 1,03 | + | - | + | + |
| 1,04 | + | - | + | + |
| 1,05 | + | - | + | + |
| 1,10 | + | - | + | + |
| 1,20 | + | - | + | + |
| 1,50 | + | + | + | + |
| 2,00 | - | - | + | + |

**lentelė 16 ketvirtos nuotraukos atpažinimo tyrimai**

| | Haar cascade | | LBP klasifikatorius | |
|---|---|---|---|---|
| Didinimo žingsnis | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 1,01 | + | + | + | + |
| 1,02 | + | - | + | + |
| 1,03 | + | - | + | + |
| 1,04 | + | - | + | + |
| 1,05 | + | - | + | + |
| 1,10 | + | - | + | + |
| 1,20 | + | - | + | + |
| 1,50 | + | + | + | + |
| 2,00 | - | - | + | + |

**lentelė 17 penktos nuotraukos atpažinimo tyrimai**

| | Haar cascade | | LBP klasifikatorius | |
|---|---|---|---|---|
| Didinimo žingsnis | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 1,01 | + | + | + | + |
| 1,02 | + | - | + | + |
| 1,03 | + | - | + | + |
| 1,04 | + | - | + | + |
| 1,05 | + | - | + | + |
| 1,10 | + | - | + | + |
| 1,20 | + | - | + | + |
| 1,50 | + | + | + | + |
| 2,00 | - | - | + | + |

**lentelė 18 šeštos nuotraukos atpažinimo tyrimai**

| | Haar cascade | | LBP klasifikatorius | |
|---|---|---|---|---|
| Didinimo žingsnis | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 1,01 | + | + | + | + |
| 1,02 | + | + | + | + |
| 1,03 | + | + | + | + |
| 1,04 | + | - | + | + |
| 1,05 | + | - | + | + |
| 1,10 | + | - | + | + |
| 1,20 | + | - | + | + |
| 1,50 | + | + | + | + |
| 2,00 | - | - | + | + |

**lentelė 19 septintos nuotraukos atpažinimo tyrimai**

| | Haar cascade | | LBP klasifikatorius | |
|---|---|---|---|---|
| Didinimo žingsnis | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 1,01 | + | - | + | + |
| 1,02 | + | + | + | + |
| 1,03 | + | + | + | + |
| 1,04 | + | - | + | + |
| 1,05 | + | - | + | + |
| 1,10 | + | - | + | + |
| 1,20 | + | - | + | + |
| 1,50 | + | - | + | + |
| 2,00 | - | - | - | - |

**lentelė 20 aštuntos nuotraukos atpažinimo tyrimai**

| | Haar cascade | | LBP klasifikatorius | |
|---|---|---|---|---|
| Didinimo žingsnis | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 1,01 | + | - | + | + |
| 1,02 | + | - | + | + |
| 1,03 | + | - | + | + |
| 1,04 | + | - | + | + |
| 1,05 | + | - | + | + |
| 1,10 | + | - | + | + |
| 1,20 | - | - | + | + |
| 1,50 | - | - | + | + |
| 2,00 | - | - | - | - |

**lentelė 21 devintos nuotraukos atpažinimo tyrimai**

| | Haar cascade | | LBP klasifikatorius | |
|---|---|---|---|---|
| Didinimo žingsnis | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 1,01 | + | - | + | + |
| 1,02 | + | - | + | + |
| 1,03 | + | - | + | + |
| 1,04 | + | - | + | + |
| 1,05 | + | - | + | + |
| 1,10 | + | - | + | + |
| 1,20 | - | - | + | + |
| 1,50 | - | - | + | + |
| 2,00 | - | - | - | - |

**lentelė 22 dešimtos nuotraukos atpažinimo tyrimai**

| | Haar cascade | | LBP klasifikatorius | |
|---|---|---|---|---|
| Didinimo žingsnis | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 1,01 | + | - | + | + |
| 1,02 | + | - | + | + |
| 1,03 | + | - | + | + |
| 1,04 | + | - | + | + |
| 1,05 | + | - | + | + |
| 1,10 | + | - | + | + |
| 1,20 | + | - | + | + |
| 1,50 | + | + | + | + |
| 2,00 | + | - | + | + |

8.1.1.1.1. Tyrimas keičiant minimalų kaimynų skaičių

**lentelė 23 pirmos nuotraukos atpažinimo tyrimai**

| | Haar cascade | | LBP klasifikatorius | |
|---|---|---|---|---|
| Minimalus kaimynų skaičius | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 2 | + | - | + | + |
| 3 | + | + | + | + |
| 4 | + | - | + | + |
| 10 | + | - | + | + |
| 20 | + | - | + | + |
| 50 | + | - | + | - |
| 100 | - | - | + | - |

**lentelė 24 antros nuotraukos atpažinimo tyrimai**

| | Haar cascade | | LBP klasifikatorius | |
|---|---|---|---|---|
| Minimalus kaimynų skaičius | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 2 | + | - | + | + |
| 3 | + | - | + | + |
| 4 | + | - | + | + |
| 10 | + | - | + | + |
| 20 | + | - | + | + |
| 50 | + | - | + | - |
| 100 | - | - | + | - |

**lentelė 25 trečios nuotraukos atpažinimo tyrimai**

| | Haar cascade | | LBP klasifikatorius | |
|---|---|---|---|---|
| Minimalus kaimynų skaičius | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 2 | + | + | + | + |
| 3 | + | - | + | + |
| 4 | + | - | + | + |
| 10 | + | - | + | + |
| 20 | + | - | + | + |
| 50 | + | - | + | - |
| 100 | - | - | + | - |

**lentelė 26 ketvirtos nuotraukos atpažinimo tyrimai**

| | Haar cascade | | LBP klasifikatorius | |
|---|---|---|---|---|
| Minimalus kaimynų skaičius | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 2 | + | - | + | + |
| 3 | + | - | + | + |
| 4 | + | - | + | + |
| 10 | + | - | + | + |
| 20 | + | - | + | + |
| 50 | + | - | + | - |
| 100 | - | - | + | - |

**lentelė 27 penktos nuotraukos atpažinimo tyrimai**

| | Haar cascade | | LBP klasifikatorius | |
|---|---|---|---|---|
| Minimalus kaimynų skaičius | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 2 | + | - | + | + |
| 3 | + | - | + | + |
| 4 | + | - | + | + |
| 10 | + | - | + | + |
| 20 | + | - | + | + |
| 50 | + | - | + | - |
| 100 | - | - | + | - |

**lentelė 28 šeštos nuotraukos atpažinimo tyrimai**

| | Haar cascade | | LBP klasifikatorius | |
|---|---|---|---|---|
| Minimalus kaimynų skaičius | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 2 | + | - | + | + |
| 3 | + | - | + | + |
| 4 | + | - | + | + |
| 10 | + | - | + | + |
| 20 | + | - | + | + |
| 50 | + | - | + | - |
| 100 | - | - | + | - |

**lentelė 29 septintos nuotraukos atpažinimo tyrimai**

| | Haar cascade | | LBP klasifikatorius | |
|---|---|---|---|---|
| Minimalus kaimynų skaičius | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 2 | + | - | + | + |
| 3 | + | - | + | + |
| 4 | + | - | + | + |
| 10 | + | - | + | + |
| 20 | + | - | + | + |
| 50 | + | - | + | - |
| 100 | - | - | + | - |

**lentelė 30 aštuntos nuotraukos atpažinimo tyrimai**

| | Haar cascade | | LBP klasifikatorius | |
|---|---|---|---|---|
| Minimalus kaimynų skaičius | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 2 | + | - | + | + |
| 3 | + | - | + | + |
| 4 | + | - | + | + |
| 10 | + | - | + | + |
| 20 | + | - | + | + |
| 50 | - | - | + | - |
| 100 | - | - | + | - |

**lentelė 31 devintos nuotraukos atpažinimo tyrimai**

| | Haar cascade | | LBP klasifikatorius | |
|---|---|---|---|---|
| Minimalus kaimynų skaičius | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 2 | + | - | + | + |
| 3 | + | - | + | + |
| 4 | + | - | + | + |
| 10 | + | - | + | + |
| 20 | + | - | + | + |
| 50 | + | - | + | - |
| 100 | - | - | + | - |

**lentelė 32 dešimtos nuotraukos atpažinimo tyrimai**

| | HAAR klasifikatorius | | LBP klasifikatorius | |
|---|---|---|---|---|
| Minimalus kaimynų skaičius | Atpažinimas | Klaidos atpažinime | Atpažinimas | Klaidos atpažinime |
| 2 | + | - | + | + |
| 3 | + | - | + | + |
| 4 | + | - | + | + |
| 10 | + | - | + | + |
| 20 | + | - | + | + |
| 50 | + | - | + | - |
| 100 | - | - | + | - |

### 8.1.2. Testavimas su realiais objektais

**lentelė 33 tyrimo rezultatai realioje aplinkoje keičiant didinimo žingsnio reikšmę**

| | | | HAAR | LBP |
|---|---|---|---|---|
| **Didinimo skaičius** | **Minimalus kaimynų skaičius** | **Apšvietimas** | **Rezultatas** | **Rezultatas** |
| 1,01 | 1 | Blankus | Neatpažinta | Atpažinta su klaidomis |
| | | Geras | Neatpažinta | Atpažinta su klaidomis |
| 1,02 | | Blankus | Taip, su daugybę neatitikimų | Atpažinta su klaidomis |
| | | Geras | Neatpažinta | Atpažinta su klaidomis |
| 1,03 | | Blankus | Atpažinta, su daugybę klaidų | Atpažinta su klaidomis |
| | | Geras | Atpažinta | Atpažinta su klaidomis |
| 1,04 | | Blankus | Atpažinta, bet su klaidomis | Atpažinta su klaidomis |
| | | Geras | Atpažinta | Atpažinta su klaidomis |
| 1,05 | | Blankus | Atpažinta | Atpažinta su klaidomis |
| | | Geras | Atpažinta | Atpažinta su klaidomis |
| 1,10 | | Blankus | Atpažinta | Atpažinta su klaidomis |
| | | Geras | Atpažinta | Atpažinta su klaidomis |
| 1,20 | | Blankus | Atpažinta, bet akys rodomos ne toje vietoje | Atpažinta su klaidomis |
| | | Geras | Atpažinta, bet akys rodomos ne toje vietoje | Atpažinta su klaidomis |
| 1,50 | | Blankus | Neatpažinta | Neatpažinta |
| | | Geras | Neatpažinta | Atpažinta |
| 2 | | Blankus | Neatpažinta | Neatpažinta |
| | | Geras | Neatpažinta | Atpažinta |

**lentelė 34 tyrimo rezultatai realioje aplinkoje keičiant minimalų kaimynų skaičiaus reikšmę**

| | | | **HAAR** | **LBP** |
|---|---|---|---|---|
| **Didinimo skaičius** | **Minimalus kaimynų skaičiųs** | **Apšvietimas** | **Rezultatas** | |
| 1,05 | 1 | Blankus | Atpažinta | Atpažinta su klaidomis |
| | | Geras | Atpažinta | Atpažinta su klaidomis |
| | 2 | Blankus | Atpažinta su daug neatitikimų | Atpažinta su klaidomis |
| | | Geras | Atpažinta su daug neatitikimų | Atpažinta su klaidomis |
| | 3 | Blankus | Neatpažinta | Atpažinta su klaidomis |
| | | Geras | Atpažinta | Atpažinta su klaidomis |
| | 4 | Blankus | Atpažinta | Atpažinta su klaidomis |
| | | Geras | Atpažinta | Atpažinta su klaidomis |
| | 5 | Blankus | Atpažinta su netikslumais | Neatpažinta |
| | | Geras | Atpažinta | Atpažinta |
| | 10 | Blankus | Neatpažinta | Neatpažinta |
| | | Geras | Atpažinta su netikslumais | Atpažinta |

### 8.1.3. Spartos tyrimas

- Didinimo žingsnis 1,01
- Minimalus kaimynų skaičius 4

**lentelė 35 spartos tyrimo rezultatai**

| Nuotrauka | HAAR | LBP |
|---|---|---|
| 1 | 966 ms | 824 ms |
| 2 | 1063 ms | 954 ms |
| 3 | 1310 ms | 1023 ms |
| 4 | 1580 ms | 875 ms |
| 5 | 1323 ms | 756 ms |
| 6 | 1779 ms | 1123 ms |
| 7 | 1296 ms | 795 ms |
| 8 | 1247 ms | 952 ms |
| 9 | 1413 ms | 846 ms |
| 10 | 1675 ms | 1065 ms |

**lentelė 36 spartos tyrimo rezultatai realioje aplinkoje**

| HAAR | LBP |
|---|---|
| 1150 ms | 1087 ms |

- Didinimo žingsnis 1,05
- Minimalus kaimynų skaičius 4

**lentelė 37 spartos tyrimo rezultatai**

| Nuotrauka | HAAR | LBP |
|---|---|---|
| 1 | 468 ms | 295 ms |
| 2 | 426 ms | 248 ms |
| 3 | 534 ms | 302 ms |
| 4 | 402 ms | 255 ms |
| 5 | 414 ms | 239 ms |
| 6 | 492 ms | 289 ms |
| 7 | 358 ms | 221 ms |
| 8 | 384 ms | 236 ms |
| 9 | 413 ms | 298 ms |
| 10 | 496 ms | 312 ms |

**lentelė 38 spartos tyrimo rezultatai realioje aplinkoje**

| HAAR | LBP |
|---|---|
| 295 ms | 265 ms |

- Didinimo žingsnis 1,10
- Minimalus kaimynų skaičius 4

**lentelė 39 spartos tyrimo rezultatai**

| Nuotrauka | HAAR | LBP |
|---|---|---|
| 1 | 166 ms | 165 ms |
| 2 | 178 ms | 169 ms |
| 3 | 193 ms | 188 ms |
| 4 | 169 ms | 166 ms |
| 5 | 158 ms | 173 ms |
| 6 | 183 ms | 180 ms |
| 7 | 150 ms | 160 ms |
| 8 | 157 ms | 155 ms |
| 9 | 167 ms | 149 ms |
| 10 | 198 ms | 166 ms |

**lentelė 40 spartos tyrimo rezultatai realioje aplinkoje**

| HAAR | LBP |
|---|---|
| 204 ms | 175 ms |

## 8.2. Tyrimo metu naudotos nuotraukos

**pav. 33 Pirma nuotrauka**

**pav. 34 Antra nuotrauka**

**pav. 35 Trečia nuotrauka**

**pav. 36 Ketvirta nuotrauka**

**pav. 37 Penkta nuotrauka**


**pav. 38 Šešta nuotrauka**

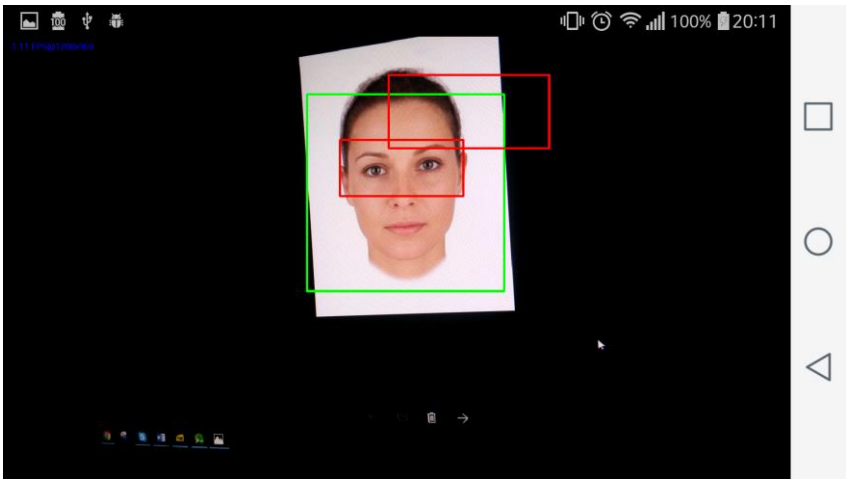
**pav. 39 Septinta nuotrauka**

**pav. 40 Aštunta nuotrauka**

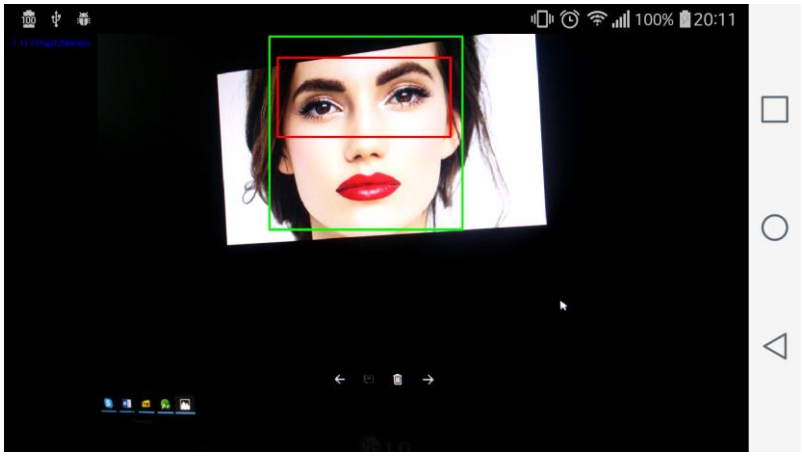
**pav. 41 Devinta nuotrauka**

**pav. 42 Dešimta nuotrauka**

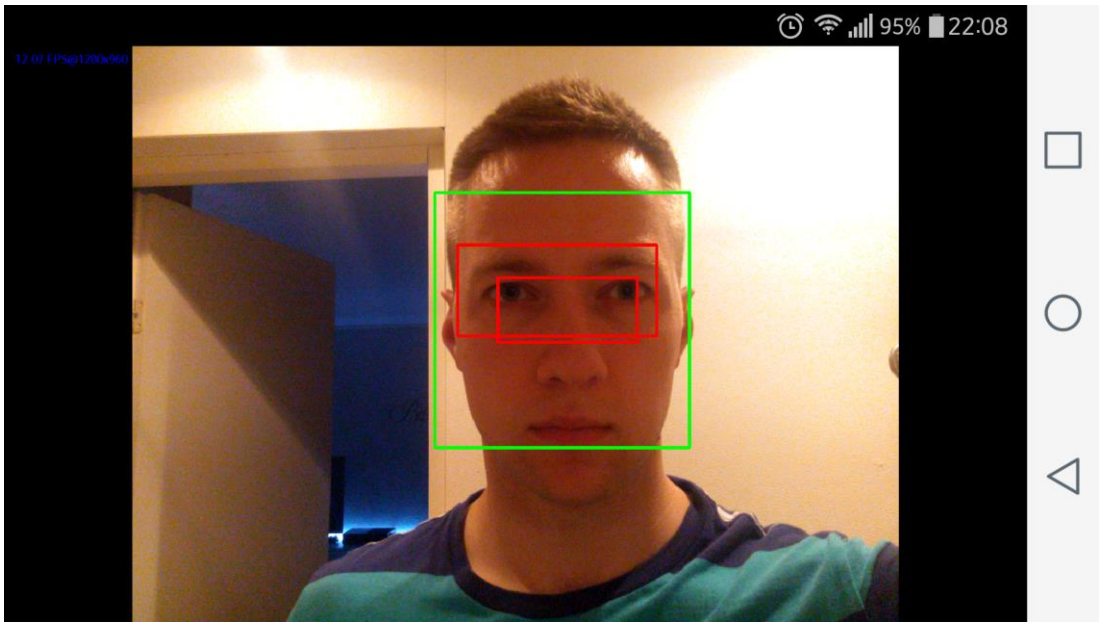
## 8.3. Tyrimo ekrano vaizdai




**pav. 43. Atpažinimas su klaidomis**




pav. 44. Tikslus atpažinimas




**pav. 45. Atpažinimas su klaidomis esant 1,3 didinimo žingsniui**