VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
GRAFINIŲ SISTEMŲ KATEDRA

Rokas Semėnas

APŠVIETIMO NORMALIZAVIMO ALGORITMŲ TYRIMAS VEIDAMS ATPAŽINTI
ILLUMINATION NORMALIZATION ALGORITHM ANALYSIS IN FACE RECOGNITION

Baigiamasis magistro darbas

Informacinių technologijų studijų programa, valstybinis kodas 621E14004
Multimedijos informacinių sistemų specializacija
Informatikos inžinerija

Vilnius, 2017

Vilniaus Gedimino technikos universitetas
Fundamentinių mokslų fakultetas
Grafinių sistemų katedra

ISBN ISSN
Egz. sk. .....
Data .....-.....-.....

Antrosios pakopos studijų **Informacinių technologijų** programos magistro baigiamasis darbas 4

Pavadinimas	**Apšvietimo normalizavimo algoritmų tyrimas veidams atpažinti**
Autorius	**Rokas Semėnas**
Vadovas	**Romualdas Baušys**

Kalba: lietuvių

Anotacija

Baigiamajame magistro darbe nagrinėjamos veido atpažinimo ir apšvietimo normalizavimo technologijos. Pasiūlytas naujas efektyvus apšvietimo normalizavimo algoritmas, kuris realizuotas naudojant "OpenCV" ir "MS Visual Studio C++" programinę įrangą. Ištirtas siūlomo metodo efektyvumas bei nustatyti optimalūs PCA, LDA ir LBP veido atpažinimo metodų parametrai. Eksperimentiniai tyrimai atlikti naudojant Multimedijos ir kompiuterinio dizaino laboratorijos veidų duomenų bazę.

Darbą sudaro 5 dalys: įvadas, veido atpažinimo ir apšvietimo normalizavimo algoritmų apžvalga, siūlomo metodo integracijos kartu su "OpenCV" metodologijos aprašymas bei eksperimentinis sukurtos sistemos įvertinimas. Pateikiamos darbo išvados.

Darbo apimtis - 77 p. teksto be priedų, 23 iliustr., 10 lent., 56 bibliografiniai šaltiniai.

Prasminiai žodžiai: veido atpažinimas, apšvietimo normalizacija, OpenCV, PCA, LDA, LBP

Vilnius Gediminas Technical University
Faculty of Fundamental Sciences
Department of Graphical Systems

ISBN ISSN
Copies No. .....
Date .....-.....-.....

Master Degree Studies **Information Technologies** study programme Master Graduation Thesis 4

Title	**Illumination Normalization Algorithm Analysis in Face Recognition**
Author	**Rokas Semėnas**
Academic supervisor	**Romualdas Baušys**

Thesis language: Lithuanian

Annotation

This thesis examines facial recognition and illumination normalization technologies. The new and more efficient lighting normalization algorithm, which is implemented by using "OpenCV" and "MS Visual Studio C ++" software is proposed. The effectiveness of the proposed method is investigated and the optimal PCA, LDA and LBP method parameters are determined. Experimental studies are carried out by using the Multimedia and computer design laboratory face database.

The work consists of 5 parts: introduction, face recognition and lighting normalization algorithm analysis, the methodology of proposed method integration with "OpenCV" and description of the experimental results. Thesis summary is presented.

Work scope - 77 p. of text, 23 pictures, 10 tables, 56 references.

Keywords: face recognition, illumination normalization, OpenCV, PCA, LDA, LBP

TURINYS

PAVEIKSLŲ IR GRAFIKŲ SĄRAŠAS....4

LENTELIŲ SĄRAŠAS....5

SĄVOKŲ IR TERMINŲ ŽODYNAS....6

ĮVADAS....7

1.LITERATŪROS APŽVALGA....9

1.1.Kompiuterinė rega....9

1.2.Veido atpažinimo metodai....15

1.3.Veido atpažinimo programinės įrangos apžvalga....24

1.4.Apšvietimo normalizavimo modelio apžvalga....28

1.5.Skyriaus apibendrinimas....30

2.APŠVIETIMO NORMALIZAVIMO IR VEIDO ATPAŽINIMO ALGORITMAI....31

2.1.Veido aptikimas....31

2.2.Standartinis veido nuotraukos normalizavimas....32

2.3.Siūlomas apšvietimo normalizavimo metodas....32

2.4.Apšvietimo normalizacijos algoritmai....33

2.4.1.*Gama* korekcija....34

2.4.2.*CLAHE* metodas....34

2.4.3.Histogramos išlyginimas....35

2.4.4.*Single–scale Retinex* modelis....35

2.4.5.*Multi–scale Retinex* modelis....35

2.4.6.*Single–scale Self Quotient* modelis....36

2.4.7.*Multi–scale Self Quotient* modelis....36

2.4.8.Homomorfinis filtravimas....36

2.4.9.Pleksiformo imitacijos normalizacija....37

2.4.10.Tano ir Trigso metodas....37

2.5.Veido atpažinimas....38

2.5.1.Tikrinių veidų metodas....38

2.5.2.Fišerio veidų metodas....41

2.5.3.Vietinio dvejetainio modelio metodas....44

2.6.Skyriaus apibendrinimas....46

3.KURIAMOS PROGRAMOS INTEGRACIJA *OPENCV*....47

3.1.Programos architektūra....47

3.2.Aplikacijos veikimo ypatumai....49

3.3.Skyriaus apibendrinimas....53

4.SIŪLOMO APŠVIETIMO NORMALIZAVIMO ALGORITMO EKSPERIMENTINIS ĮVERTINIMAS....54

4.1.Biometrinių sistemų vertinimo metodikos....54

4.2.Duomenų bazės paruošimas....57

4.3.Eksperimentinio tyrimo rezultatai....59

IŠVADOS....69

LITERATŪROS SĄRAŠAS....71

PAVEIKSLŲ IR GRAFIKŲ SĄRAŠAS

1 pav. Veido atpažinimo metodika....17

2 pav. X. Xie normalizavimo modelis....29

3 pav. Kraštinių požymiai, kvadratiniai požymiai, linijų požymiai....31

4 pav. Siūlomas apšvietimo normalizacijos modelis....33

5 pav. *CLAHE* metodas....34

6 pav. Histogramos išlyginimas....35

7 pav. Tikrinių veidų apmokymo ir testavimo algoritmas....39

8 pav. Fišerio veidų apmokymo ir testavimo algoritmas....42

9 pav. *LBP* dvejetainio kodo apskaičiavimas....44

10 pav. *LBP* aptinkamų požymių pavyzdys....45

11 pav. Sukurtos programos *UML* diagrama....48

12 pav. *CSV* failo pavyzdys....49

13 pav. Standartinio veido stačiakampio apskaičiavimas....50

14 pav. Tikėtinumų pasiskirstymo kreivės....54

15 pav. Plotas po *ROC* kreive....56

16 pav. Nuotraukų scenos išdėstymas....57

17 pav. Pirmojo atrinkto algoritmų rinkinio *ROC* kreivės naudojant *PCA*....61

18 pav. Pirmojo atrinkto algoritmų rinkinio *ROC* kreivės naudojant *LDA*....63

19 pav. Pirmojo atrinkto algoritmų rinkinio *ROC* kreivės naudojant *LBP*....64

20 pav. Antrojo atrinkto algoritmų rinkinio *ROC* kreivės naudojant *PCA*....65

21 pav. Antrojo atrinkto algoritmų rinkinio *ROC* kreivės naudojant *LDA*....66

22 pav. Antrojo atrinkto algoritmų rinkinio *ROC* kreivės naudojant *LBP*....67

23 pav. Veido atpažinimo koeficiento priklausomybė nuo regiono dydžio....68

LENTELIŲ SĄRAŠAS

1 lentelė. *Matlab* ir *OpenCV* palyginimas.....27

2 lentelė. Sprendimų matrica.....55

3 lentelė. Veido nuotraukų apšvietimo pavyzdžiai.....58

4 lentelė. Apšvietimo normalizavimo algoritmų junginių rezultatai (%).....60

5 lentelė. Pirmojo atrinkto algoritmų rinkinio *TP* ir *FP* santykiai naudojant *PCA*.....62

6 lentelė. Pirmojo atrinkto algoritmų rinkinio *TP* ir *FP* santykiai naudojant *LDA*.....63

7 lentelė. Pirmojo atrinkto algoritmų rinkinio *TP* ir *FP* santykiai naudojant *LBP*.....64

8 lentelė. Antrojo atrinkto algoritmų rinkinio *TP* ir *FP* santykiai naudojant *PCA*.....65

9 lentelė. Pirmojo atrinkto algoritmų rinkinio *TP* ir *FP* santykiai naudojant *LDA*.....66

10 lentelė. Antrojo atrinkto algoritmų rinkinio *TP* ir *FP* santykiai naudojant *LBP*.....67

SĄVOKŲ IR TERMINŲ ŽODYNAS

1. **3D** – trijų dimensijų vaizdas (angl. *Three dimension – 3D*);
2. **API** – aplikacijos programavimo aplinka (angl. *Aplication programing interface*);
3. **CLAHE** – Adaptyvus nuo kontrasto priklausomas histogramos normalizavimo metodas (angl. *Contrast limited adaptive histogram equalization – CLAHE*);
4. **CSV** – duomenų aprašymo metodas (angl. *Comma separated values*);
5. **FP** – Klaidingas priėmimas (angl. *False positive – FP*);
6. **FPR –** Klaidingo priėmimo lygis (angl. *False positive rate – FPR*);
7. **LBP –** Vietinis dvejetainis modelis (angl. *Local binary pattern – LBP*);
8. **LDA –** Linijinio diskriminanto analizė (angl. *Linear discriminant analysis – LDA*);
9. **MSR –** apšvietimo normalizavimo algoritmas (angl. *Multi–scale retinex*);
10. **MSQ –** apšvietimo normalizavimo algoritmas (angl. *Multi–scale self quotient*);
11. **PCA –** Pagrindinių komponenčių analizė (angl. *Principal component analysis – PCA*);
12. **PIN** – skaitmeninis kodas, skirtas identifikuoti banko kortelę naudojantį asmenį;
13. **SSR –** apšvietimo normalizavimo algoritmas (angl. *Single–scale retinex*);
14. **SSQ –** apšvietimo normalizavimo algoritmas (angl. *Single–scale self quotient*);
15. **TP –** Teisingas priėmimas (angl. *True positive – TP*);
16. **TPR –** Teisingo priėmimo lygis (angl. *True positive rate – TPR*).

ĮVADAS

Problema ir aktualumas

Asmens identifikavimas pagal jo veido struktūrą yra vienas iš pagrindinių polinominių kompiuterinės regos uždavinių. Dėl plataus pritaikymo spektro, ši problema išlieka tyrimų objektu tiek pramogų industrijoje, tiek saugumo srityse. Šiame darbe nagrinėjama biometrinių sistemų problema yra itin aktuali, kadangi individualūs žmogaus išvaizdos ir elgsenos požymiai yra teoriškai nesuklastojami ir ypač patogūs naudotojo atžvilgiu. Daugybę skirtingų slaptažodžių ir *PIN* kodų galėtų būti pakeisti viena paprasta veido požymių analize. Nuo pat 1970 metų iki šių dienų, veido atpažinimo ir asmens identifikavimo problemos tyrimams buvo skirta daugybė išteklių, o pirmosios, visuomenei prieinamos, sistemos paplito kartu su išmaniaisiais mobiliojo ryšio įrenginiais. Verta paminėti, kad ši technologija vis dar yra pramoginio lygio ir dažnas naudotojas savo informacijos saugumo jai nepatiki.

Viena pagrindinių priežasčių, nulėmusių gana ribotą veido atpažinimo technologijų pritaikymą saugumo srityje yra nepastovus atpažinimo tikslumo koeficientas. Dauguma geriausių identifikavimo rezultatų yra pasiekiami visiškai ar iš dalies valdomoje aplinkoje, naudojant specializuotą įrangą ir daug sistemos resursų reikalaujančius algoritmus. Realiomis sąlygomis, veido atpažinimo metodai gali susidurti su tokiomis problemomis kaip: asmens galvos posūkio kampas; veido išraiškų įtaka požymių išskyrimui; dalinis arba visiškas asmens uždengimas kitais objektais; vaizdo ar fotokameros rezoliucijos ir triukšmo problema; prasto apšvietimo įtaka paveikslo kokybei.

Šiame darbe testuojami standartiniai nuotraukų analizės metodai: *Pagrindinių komponenčių analizė*, *Linijinė diskriminanto analizė* bei *Vietiniai dvejetainiai modeliai*. Visi šie algoritmai yra pakankamai jautrūs informacijos nuokrypiams, susidariusiems dėl nagrinėjamose nuotraukose esančių neapšviestų regionų. Norint padidinti veido atpažinimo tikslumo koeficientą, privaloma išankstinė paveikslo normalizacija, atsižvelgiant į galimo nuokrypio priežastį. Šio tyrimo tikslas buvo patikrinti standartinių apšvietimo normalizavimo algoritmų apjungimo naudą asmens atpažinimo tikslumui, naudojant nespecializuotas foto kameras ir *OpenCV* biblioteką.

Darbo tikslas

Sukurti apšvietimo normalizavimo metodą ir realizuoti jį *OpenCV* programos aplinkoje. Ištestuoti praktikoje naudojamus algoritmus ir jų junginius bei nustatyti jų tikslumui įtaką turinčius parametrus. Darbo metu **spręsti uždaviniai**:

1. Esamų technologijų ir veido atpažinimo metodikų analizė;
2. Efektyvaus apšvietimo normalizacijos modelio sukūrimas;
3. Siūlomo apšvietimo normalizavimo modelio integravimas *OpenCV*;
4. Sukurto apšvietimo normalizacijos metodo eksperimentinis įvertinimas.

Darbo naujumas ir praktinė nauda

Darbo metu tobulinta biometrinių kompiuterinės regos sistemų sritis – veido atpažinimo technologijos. Ištirti įvairūs apšvietimo normalizavimo algoritmai ir sukonstruoti jų deriniai, nagrinėjama jų įtaka veido atpažinimo tikslumui. Standartiniams veido atpažinimo metodams nustatyti optimalūs algoritmų junginiai. Sukurtas apšvietimo normalizavimo modelis, kuris leidžia praktiškai pagerinti veido atpažinimo koeficientą ir gali būti pritaikytas kuriamose biometrinėse asmenų identifikavimo sistemose.

1. LITERATŪROS APŽVALGA

Šiame skyriuje aptariama apibendrinta kompiuterinės regos (angl. *Computer vision – CV*) atsiradimo istorija, sprendžiami uždaviniai, kylančios problemos ir jų sprendimo būdai bei praktinis egzistuojančių sistemų pritaikymas. Didelis dėmesys skiriamas veidų atpažinimo metodams ir normalizacijos problemoms, trumpai aprašyta ir palyginta pagrindinė praktikoje naudojama programinė įranga. Aptariamos skyriaus išvados.

1.1. Kompiuterinė rega

Kompiuterinės regos uždaviniai ir aplikacijos pradėjo formuotis 1960 metais kartu su dirbtinio intelekto sistemomis. Pagrindinis šių tyrimų tikslas buvo sukurti matančias mašinas, kurios galėtų kopijuoti žmogaus elgseną ir vaizdo suvokimą – trumpiau tariant, išanalizuoti nuotrauką ir suprasti jos turinį. Tuo metu užduotis atrodė lengvai įgyvendinama dėl spartaus dirbtinio intelekto algoritmų kūrimo ir išorinio finansavimo, tačiau jau kitame dešimtmetyje ši problema buvo išskirstyta į daugybę skirtingų kategorijų [36].

1970 metais Masačiusetso technologijų institutas (angl. *Massachusetts Institute of Technology*) sukūrė kompiuterinės regos kursą. Tais pačiais metais buvo pradėtos organizuoti kompiuterinės regos konferencijos. Šiame dešimtmetyje didžiausią pasiekimą padarė D. Maras, 1978 metais pasiūlęs *iš apačios į viršų* (angl. *Bottom–up*) analizės idėją. Taikant žemo lygio nuotraukos apdorojimo algoritmus, paveiksliuke išskiriamos linijos ir paverčiamos 2.5 dimensijų scenos vizualizacija. Naudojant struktūrinės analizės metodus iš jos galima ištraukti užfiksuotų objektų 3 dimensijų modelį – tai ko siekė to laikmečio mokslininkai [36].

1980 metais daug dėmesio pradėta skirti kiekybinei nuotraukų analizei. Nustatyta, kad priklausomai nuo atliekamos užduoties, D. Maro pasiūlytas metodas gali būti sunkiai įgyvendinamas, o daugumai analizių atlikti 3 dimensijų scenos modelio kūrimas nėra būtinas. Tuo metu buvo atlikta daug tyrimų siekiant optimizuoti linijų ir objektų kontūrų aptikimo metodikas, taip pat daug išteklių skirta objektų formos išskyrimo (angl. *Shape–from–X*) problemoms spręsti. Šiame dešimtmetyje suformuluoti variacijų optimizavimo ir nuotraukų normalizacijos uždaviniai [36].

1990 metais suaktyvėjo tyrinėjimai biometrinių sistemų srityje, ištobulintos objektų sekimo ir judesio fiksavimo technologijos. Pasiūlytas veido atpažinimo algoritmas ir jam skirti statistinės analizės metodai. Šiame laikotarpyje didžiausi pasiekimai pastebimi kompiuterinės grafikos srityje. Sukurti nuotraukų apjungimo metodai, sprendžiama morfingo

(angl. *Morphing*) problema. Pasiūlyta paveikslėliais pagrįsta 3 dimensijų modeliavimo technologija, kuri sukuria foto realistiškus objektus panaudodama nuotraukų rinkinius.

2000 metais sparčiai pradėjo plisti požymiais pagrįstų atpažinimo sistemų kūrimas, gilinamas ryšys tarp kompiuterinės regos ir kompiuterinės grafikos sričių. Toliau sprendžiama nuotraukų sujungimo problema, tobulinamos vizualizavimo technologijos. Šiuo laikotarpiu itin daug dėmesio skirta siekiant išspręsti charakteringų objekto požymių išskyrimo problemas ir kompiuterio apmokymo uždavinius. Sukurti scenos ir vietovės atpažinimo algoritmai, realizuota panašių paveikslėlių paieškos sistema [46].

2010 metais plačiai paplito neuroniniai tinklai ir save apmokančios kompiuterinės regos sistemos. Realizuotas objektų atpažinimo algoritmų pritaikymas išmaniuosiuose įrenginiuose, pastebimas biometrinių sistemų paplitimas. Daug dėmesio skiriama žmogaus, o ypač veido aptikimui ir atpažinimui. Optimizuoti anksčiau pasiūlyti algoritmai ir sparčiai tobulinamos jų versijos. Atsiranda papildytos realybės technologijos.

Kompiuterinė rega vis dar yra aktyviai plėtojama ir aktuali mokslo sritis, praktiškai pritaikoma:

1. **Rašytam ar spausdintam tekstui atpažinti**. Ši technologija turi daug realizavimo būdų. Ji pritaikoma daugumos išsivysčiusių šalių automobilių stovėjimo aikštelėse, kai norima identifikuoti į teritoriją įvažiuojančias ir išvažiuojančias transporto priemones [1, 47]. Bibliotekose bei informacijos centruose ši technologija padeda suskaitmeninti, ir taip išsaugoti senus dokumentus, laikraščius ar kitą istoriškai svarbią informaciją ir padaryti ją prieinamą visuomenei. Mokiniams prieinama formulių atpažinimo ir matematikos užduočių sprendimo programėlė, taip pat programa, padedanti rašytinį tekstą paversti skaitmenine jo versija.
2. **Medicininių nuotraukų analizėje**. Automatizuoti egzistuojantys kompiuterinės regos metodai gali padėti aptikti ir nustatyti ligos sukėlėjus ir diagnozę. Magnetinio rezonanso, rentgeno spindulių ar panašių medicininių nuotraukų kompiuterinė analizė gali padėti laiku aptikti sunkiai pastebimus darinius ar atkreipti gydytojo dėmesį į potencialią ligą. Šiuo metu praktikoje naudojamos sistemos pagrįstos ne visiškai automatizuotu sprendimų priėmimu, o įtarimą keliančių regionų identifikavimu ir pateikimu. Ši technologija dažnai naudojama vėžio dariniams aptikti [32, 49].

3. **Nuotraukų redagavime.** Šiuolaikinė pramogų ir reklamos industrija neapsieitų be kompiuterinės regos pasiekimų. Požymių palyginimo technologijos yra plačiai naudojamos kuriant panoramines nuotraukas, aptinkant ir iškerpant nepageidaujamus objektus ar šešėlius. *Google* kompanija šią technologiją naudoja tiek privatumo užtikrinimui, tiek automatizuotam informacijos surinkimui. *Google Street view* projekto metu surinktos nuotraukos turėjo būti sujungtos tarpusavyje (angl. *Stitching*). Šiuose paveiksliukuose taip pat galėjo būti daug naudingos informacijos, kurią reikėjo surinkti. Tai gali būti: pastatų ir transporto priemonių numeriai, parduotuvių pavadinimai, logotipai [1]. Atrinkus kompanijai svarbią informaciją, technologija yra pakankamai pajėgi atpažinti ir užtušuoti žmonių veidus ir kitą privačią informaciją.

4. **Objekto atkūrimas 3 dimensijų erdvėje – fotogrametrija** (angl. *Photogrammetry*). Naudojama automatizuotam miestų ir aplinkinių teritorijų modeliavimui bei objekto nuotraukų pavertimui erdviniu kūnu. Šią technologiją naudoja *3D* modeliuotojai ir filmų kūrėjai, norėdami sukurti foto realistines scenas. *NASA* šį kompiuterinės regos pritaikymą naudoja už žemės ribų – Marso planetoje [50]. Kosmoso agentūros sukurtas tyrinėjimų visureigiai fotografuoja planetos reljefą ir iš gautų failų sudarinėja *3D* gylio modelius ir panoramines nuotraukas. Kompiuterinė rega šiems aparatams padeda orientuotis nežinomame planetos paviršiuje.

5. **Kompiuterinės grafikos technologijos kino industrijoje** (angl. *Computer–generated imagery – CGI*). Objektų sekimas video medžiagoje ir sinchronizacija su *3D* modelių aplinka. Dominuojančių taškų identifikacija ir sekimas kadruose, padeda sukurti realistiškai atrodančius efektus. Kino kūrėjai sutaupo daugybę lėšų, kurios kitu atveju turėtų būti skirtos nepraktiškų specialiųjų efektų kūrimui. Kompiuterinė rega taip pat palengvina video failo post–produkciją. Automatizuotos video failų apdorojimo sistemos yra pajėgios atlikti spalvų korekcijas, aptikti pašalinius objektus, stabilizuoti kameros vaizdą, atstatyti sugadintą kadrų kokybę, sujungti atskirus scenų kadrus ar išskirti dominantį video failo segmentą [36].

6. **Judesio fiksavimo technologijose** (angl. *Motion capture*). Plačiai taikomos žaidimų ir kino industrijoje. 2010 metais pasirodžiusi *xBox kinect* sistema buvo

apmokyta atpažinti žaidėją, identifikuoti jo kūno dalis ir atskirti gestus. Žmogaus atpažinimu pagrįstos sistemos buvo sėkmingai integruotos į pramoginius kūrinius ir susilaukė pakankamai didelio visuomenės dėmesio. Be to, judesio fiksavimas kuriant kompiuterinius žaidimus padeda sukurti itin realistiškas *3D* animacijas ir fiziškai teisingus modelius. Kino industrijoje technologija naudojama tais pačiais tikslais, tačiau išnaudojamas ir papildomas *3D* erdvės aspektas – sukurtas kompiuterinis personažas gali būti įkeltas į video medžiagą daug paprasčiau.

7. **Saugomų objektų priežiūroje**. Kompiuterinė rega gali padėti aptikti patalpoje ar teritorijoje judančius objektus ir įspėti apsaugos darbuotojus. Ši sistema taip pat efektyvi, kai norima nustatyti ar neautorizuoti asmenys bandė pakenkti stebėjimo įrangai – pasukti ar uždengti kamerą. Technologija praktikoje pritaikoma ir norint identifikuoti filmuojamoje teritorijoje esančių objektų kiekį bei rinkti statistinius duomenis [36].

8. **Autonominėje navigacijoje**. Objektų atpažinimo funkcijos plačiai naudojamos autonominėse sistemose. 2010 metais pasirodę *Google* mašinos prototipai be vairuotojo įsikišimo jau nuvažiavo daugiau kaip 2 milijonus mylių. 2016 metais *Tesla* kompanijos autopilotu valdomas automobilis [51] sėkmingai numatė greitkelyje įvyksiančią avariją ir sugebėjo jos išvengti. Automatizuoti gelbėjimo robotai taip pat išnaudoja šią technologiją. Protingos mašinos ekstremaliomis sąlygomis gali padėti aptikti pavojingoje situacijoje atsidūrusius žmones ar suteikti gelbėjimo tarnyboms informaciją apie bendrą žalą ir realią situaciją. Robotai naudojantys kompiuterinės regos technologijas gali be žmogaus įsikišimo nubraižyti paieškos teritorijos planą, identifikuoti kelyje esančias kliūtis, o tuo pačiu ir aptikti ieškomus objektus.

9. **Biometrinėse sistemose**. Žmogaus pirštų atspaudų, rainelės ar veido atpažinimo sistemos naudojamos kaip papildomos apsaugos priemonės pastaraisiais metais tapo populiarios tarp išmaniųjų įrenginių gamintojų ir naudotojų. Kai telefonai tapo asmeninių kompiuterių atmaina, tapo svarbu apsaugoti vartotojo informaciją nuo pašalinių asmenų. Be to, daugybė *išmanių* funkcijų padeda vartotojui paprasčiau naudotis įrenginiu. Tai gali būti puslapių slinkimas, nuotraukų fiksavimas, skambučio atmetimas ar ekrano suaktyvinimas, pagrįstas gesto ar veido atpažinimu [19].

Praktikoje šiuo metu plačiai naudojamos požymiais pagrįstos objektų aptikimo ir atpažinimo sistemos. Pagrindinė šių metodų idėja yra aptikti dominuojančius apmokymo nuotraukų regionus, ir iš jų sudaryti nekintančių požymių vektorius – deskriptorius (angl. *Descriptors*). Požymiai gali būti dviejų kategorijų: lokalūs arba globalūs.

Globalūs požymiai skirti nuotraukos kaip visumos apibūdinimui – dominuojančios spalvos, tekstūros ar objektų kiekiui identifikuoti ir aprašyti nuotraukos turinį. Šie požymiai dažniausiai naudojami pirminiuose paieškos duomenų bazėse etapuose. Nuotrauka tokiu atveju yra aprašoma vienu daugiadimensinių požymių vektoriumi. O atliekant dviejų nuotraukų palyginimą, iš tiesų yra lyginami jų vektoriai. Pavyzdžiui – apmokymui pateikus paveiksliuką, kuriame daug raudonos spalvos, o testavimui visiškai baltą nuotrauką, pamatysime, kad atstumas tarp šių nuotraukų požymių vektorių yra pakankamas atmesti testavimo nuotrauką kaip nežinomą. Šis metodas iš dalies yra naudojamas *Google* ir *Microsoft* kompanijų paieškos sistemose [36].

Lokalūs požymiai skirti objektų identifikacijai, kadangi jie aptinka dominuojančias ypatybes visame paveikslėlyje. Jie taip pat plačiai naudojami mobiliuosiuose įrenginiuose ir aplikacijose, dėl greito duomenų apdorojimo ir santykinai gerų rezultatų. Lokalūs požymiai apibūdina nuotrauką kaip dominuojančių regionų, kurie nepriklauso nuo posūkio kampo, mastelio ar apšvietimo, rinkinį. Šie požymiai turi būti pasikartojantys ir būdingi visoms apmokymo nuotraukoms [36, 40]. Tai reiškia, kad 2 identiškų objektų paveiksliukai turi generuoti tuos pačius dominuojančius regionus. Be to, šie požymiai turi būti pakankamai unikalūs, kad juos būtų galima išskirti iš bendro nuotraukos fono ar tuo atveju kai objektas yra iš dalies uždengtas. Praktikoje tai yra regionai, kurie stipriai skiriasi nuo kaimyninių pikselių, o tai susiję su nuotraukoje esančios struktūros pokyčiais. Pastarąjį dešimtmetį požymių išskyrimas buvo vykdomas naudojant šiuos algoritmus ir patobulintus jų junginius ar atmainas:

1. ***SIFT*** (angl. *Scale invariant feature transform – SIFT*) yra požymių detektorius ir deskriptorius, pasiūlytas D. Lovie 1999 metais ir ištobulintas 2004 metais. Šis algoritmas yra nejautrus mastelio, apšvietimo, poslinkio ir orientacijos pokyčiams. Metodas veikia keturių žingsnių sistema: ekstremumų paieška mastelio erdvėje, dominuojančių regionų paieška, šių požymių orientacijos nustatymu bei deskriptoriaus sugeneravimu [40].

Pirmame žingsnyje, siekiant išskirti stabilių požymių regionus, yra naudojama *Gauso piramidė*. Norint užtikrinti, kad šie regionai bus identiški mastelio erdvėje ir *2D* nuotraukoje, praktikoje yra naudojami dviejų gretimų šios piramidės sluoksnių atimtis. Kiekvienas nuotraukoje esantis pikselis lyginamas su aštuoniais gretimais taškais (kernelio dydis 3x3), o tada atitinkantys taškai ieškomi skirtingo mastelio nuotraukose – aptinkami dominuojantys taškai [46].

Šioje taškų erdvėje toliau ieškoma požymių ir jų orientacijos. Kiekvienam požymiui yra priskiriama santykinė orientaciją, atsižvelgiant į kitus išskirtus taškus. Tai padeda užtikrinti, kad požymiai bus nejautrūs nuotraukos posūkių variacijoms. Paskutiniame algoritmo žingsnyje yra sukuriamas 128 dimensijas turintis *SIFT* deskriptorius [46].

2. ***SURF*** (angl. *Speed up robust features – SURF*) yra požymių išskyrimo ir deskriptorių sudarymo metodas. Kaip ir kiti lokalių požymių detektoriai, *SURF* yra atsparus posūkio ir mastelio pokyčiams. Be to, šis metodas yra daug greitesnis už *SIFT* ir savo kokybe neatsilieka nuo jo. Dominuojantiems taškams aptikti algoritmas naudoja integralų paveikslėlį ir *Hessian* matricą. Jos determinantas atitinka pikselių ryškumą nagrinėjamame taške. Norint užtikrinti *SURF* atsparumą nuotraukos posūkio kampui, pirmiausiai nustatoma požymio orientacija erdvėje.

   Aplink dominuojantį tašką nubrėžiamas $6s$ spindulio apskritimas, kai s yra nuotraukos mastelis. Jo viduje skaičiuojamas *Haar bangelių* (angl. *Haar–wavelet*) atsakas x ir y kryptimis. Gauti rezultatai pasveriami naudojant *Gauso* funkciją ir atvaizduojami kaip vektoriai, atitinkamai ordinatės ir abscisės ašyse. Jų sudėtis atitinka taško krypties vektorių. *SURF* deskriptorius aprašo intensyvumo pasiskirstymą, dominuojančio pikselio aplinkoje, nustatyta kryptimi [17].

3. ***FAST*** (angl. *Features from Accelerated Segment Test – FAST*) metodą 2005 metais pasiūlė E. Rostenas ir T. Drumondas. Nuotraukos segmento testo kriterijus yra pagrįstas 16–kos apie kampo kandidatą apskritimu išsidėsčiusių kaimyninių pikselių analize. Lokalus požymis užfiksuojamas tada, kai bent 12 tarpusavyje besiribojančių taškų intensyvumas yra aukščiau arba žemiau nustatytos ribos ir centrinio pikselio. Analizė optimizuojama įvertinant 4 pikselių, išsidėsčiusių statmenai ir horizontaliai centrui, reikšmes. Požymis aptinkamas tik jei 3 iš šių

pikselių tenkina slenksčio sąlygą. Aptikus dominuojantį tašką, jo požymių vektorius sudaromas iš aplinkinių pikselių [14].

4. ***BRIEF*** (angl. *Binary Robust Independent Elementary Features – BRIEF*) pasiūlytas 2011 metais [28]. Algoritmo idėja yra paprasta ir labai tiesioginė: duotai nuotraukai sukuriamas atsitiktinai sugeneruotų taškų porų masyvas, kurio pagalba apskaičiuojamos bitų reikšmės. Jei nagrinėjamo pikselio reikšmė yra mažesnė už jo porinio pikselio reikšmę – rašomas vienetas, bet kokiu kitu atveju – nulis. *BRIEF* deskriptorius yra n – dimencinė bitų eilė. Šis metodas generuoja netikėtinai gerus rezultatus, turint omenyje atsitiktinumais pagrįstą algoritmo veikimą.
5. ***ORB*** (angl. *Oriented FAST and rotated BRIEF*) algoritmas buvo pasiūlytas 2011 metais, kaip alternatyva *SIFT* ir *SURF* metodams [15]. Dominuojantiems taškams aptikti *ORB* naudoja *FAST* algoritmo metodiką – pikselis klasifikuojamas kaip kampas, jei iš aplink jį išsidėsčiusių 16 kaimyninių pikselių, 12–kos šalia esančių intensyvumas yra aukščiau arba žemiau centrinio pikselio reikšmės ir slenksčio sudėties ribos. *ORB* naudoja modifikuotą *BRIEF* deskriptoriaus versiją, norint padaryti jį nepriklausomu nuo nuotraukos posūkio kampo.

Kaip buvo minėta, lokalius ir globalius požymius taip pat galima naudoti biometrinėse sistemose. Parinkus tinkamą klasifikavimo algoritmą, galima pakankamai tiksliai identifikuoti asmenį, tačiau esantys metodai nėra tobuli. Šiuo metu egzistuojančios sistemos daugiausiai naudoja piršto atspaudu ar veido atpažinimu pagrįstą technologiją, o tai ypač populiariu tarp išmaniųjų telefonų naudotojų. Visgi veido atpažinimas vartotojo atžvilgiu yra laikomas draugiškesniu metodu. Piršto atspaudo paėmimas reikalauja specializuotos įrangos ar jos komponentės, o šiuo metu beveik visi išmanieji įrenginiai ir nešiojami kompiuteriai turi priekinę vaizdo kamerą, kurią galima panaudoti veido identifikacijai [19, 26].

1.2. Veido atpažinimo metodai

Veidų atpažinimo problema yra tokia pat sena, kaip kompiuterinės regos sistemos. Nuo pat dirbtinio intelekto atsiradimo programuotojai ir kompiuterių inžinieriai stengėsi apmokyti mašinas atskirti objektus, o ypač žmones. Skaitmeninis asmens atpažinimas yra itin

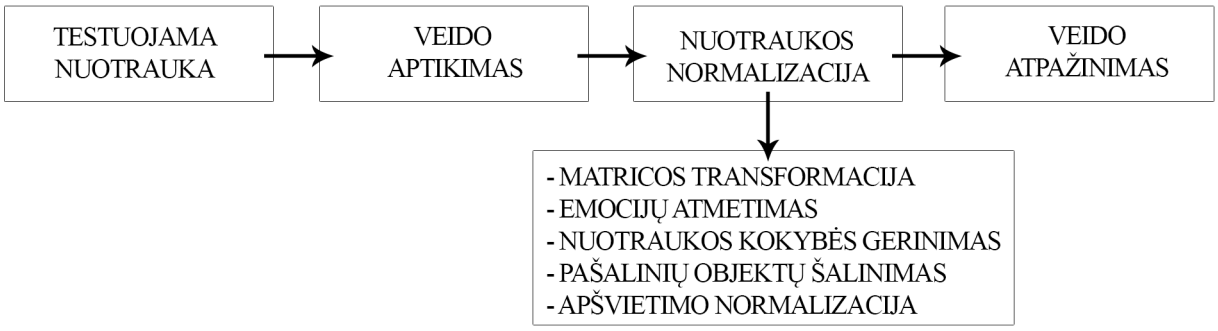
svarbus dėl savo plataus praktinių galimybių pritaikymo spektro. Pavyzdžiui, atsiradus išmaniesiems įrenginiams, telefonuose buvo įdiegta automatizuota veidų atpažinimo programa, skirta naudoti kaip papildoma įrenginio duomenų apsauga. *Facebook* savo veidų identifikavimo algoritmus naudoja nuotraukų žymėjimams, *Instagram* veido aptikimo algoritmus pritaiko pramoginėms reikmėms – nuotraukų modifikacijoms, o *Adobe Photoshop* ir *Picasa* programiniai įrankiai padeda suskirstyti šeimos narių nuotraukas į atskirus katalogus [56]. Šios programos taip pat gali būti pritaikytos bankuose ar visuomenės saugumo užtikrinimo įrenginiuose, siekiant sustiprinti sienų kontrolę ar nustatyti ieškomų asmenų buvimo vietą.

Gebėjimas atpažinti veidus yra viena pagrindinių žmogaus savybių, kurią bandoma imituoti kompiuterinėse sistemose. Siekiant tobulinti kompiuterinės regos algoritmus, 2006 metais P. Sinha atliko tyrimą ir susistemino žinomus biologinius žmonių naudojamus atpažinimo kriterijus. Šį sąrašą sudaro 19 pastebėjimų, kuriuos galima pritaikyti kuriamiems biometriniams algoritmams [34]:

1. Žmonės gali atpažinti žinomus veidus, net jei nuotraukų rezoliucija yra ypatingai maža.
2. Gebėjimas atskirti prastos kokybės veido nuotraukas priklauso nuo to, kaip gerai pažįstamas žmogus.
3. Vien aukšto dažnio informacijos nepakanka užtikrinti gerai atpažinimo kokybei.
4. Veido požymiai yra vertinami ne atskirai, o kaip visuma.
5. Iš visų veido dalių, antakiai yra svarbiausi asmens atpažinimui.
6. Požymių identifikavimas nepriklauso nuo veido nuotraukos pločio ir aukščio dimensijų.
7. Veido forma žmogaus atmintyje išlieka šiek tiek karikatūrizuota.
8. Ilgalaikis veido matymas gali padėti suformuoti jo prototipą.
9. Odos pigmentacija tokia pat svarbi kaip veido ir jo dalių forma.
10. Pigmentacija svarbi tada, kai sunku išskirti veido formą dėl prastos nuotraukų kokybės.
11. Kontrastų išvertimas į kitą pusę turi didelę neigiamą įtaką veido atpažinimui.

12. Apšvietimo pokyčiai turi įtakos atpažinimo kokybei.
13. Vaizdo suvokimas yra pagrįstas laikinomis asociacijomis.
14. Veido judėjimas palengvina jo atpažinimą.
15. Biologinė regos sistema pradeda atpažinimą nuo veido aptikimo.
16. Pirmaisiais gyvenimo metais, regos sistema ir objektų suvokimas pasikeičia iš dalinės į holistinę.
17. Žmogaus regos sistema skiria papildomus neuronų resursus veido atpažinimui.
18. Žmogaus atsako į veido nuotrauką greitis yra apie 120 ms.
19. Žmogaus identifikavimą ir emocijų atpažinimą vykdo skirtingos smegenų dalys.

Šiais pastebėjimais paremtų veido atpažinimo ir asmens identifikavimo procedūrų seka bendruoju atveju yra vienoda: pirmiausiai algoritmui pateikiama analizuojama nuotrauka. Rankiniu būdu arba naudojant automatizuotus žmogaus veido aptikimo metodus, išskiriamas analizei tinkamas regionas. Šis plotas yra normalizuojamas ir išskiriamas iš bendro ploto – sukuriamas standartinis veido stačiakampis. Toliau paveiksliukas yra normalizuojamas, pagal tai su kokiomis problemomis susiduria, ir palyginamas su duomenų bazėje esančiu nuotraukų rinkiniu. Remiantis panašumais yra atliekama asmens identifikacija. Šio algoritmo vizualizaciją matoma 1 paveiksle.



1 pav. Veido atpažinimo metodika

Bandant perkelti žmogaus biologinius veidų atpažinimo gebėjimus į kompiuterines sistemas, šis procesas pasirodė sudėtingai realizuojamas, tačiau įmanomas. 1987 metais pasirodė geriausiai žinoma tų laikų veidų atpažinimo programa, sukurta T. Kohoneno. Jis įrodė, kad paprastas dirbtinis intelektas, naudojantis neuroninius tinklus ir apmokymo

nuotraukas, sugeba atpažinti žmonių veidus. Deja, dėl itin nepalankių algoritmo reikalavimų (veido nuotrauka turėjo būti griežtai normalizuota ir sulygiuota) šis problemos sprendimo metodas nebuvo tinkamas realaus pasaulio uždaviniams spręsti [36].

Informatikai keletą metų bandė modifikuoti T. Kohoneno pasiūlytą metodą, testuodami atpažinimo algoritmus, paremtus globalių bei lokalių požymių išskyrimu ar jų junginiu. 1991 metais M. Turkas ir A. Pentlandas sukūrė *Tikrinių veidų* metodą, o asmens atpažinimui pasiūlė naudoti pagrindinių komponenčių analizę – *PCA* [31]. 1996 metais T. Ojala sukūrė *Vietinio dvejetainio modelio* požymių išskyrimo metodą, tekstūrų analizei atlikti [48]. 1997 metais P. N. Belhumeras pristatė Linijinę diskriminanto analizę – *LDA* ir *Fišerio veidų* metodą [33]. Šie algoritmai buvo modifikuojami nuo pat jų sukūrimo ir yra iki šiol yra naudojami kaip pagrindas dabartinėms veidų atpažinimo sistemoms.

Kiek naujesniu veido atpažinimo algoritmu yra laikomas *Gaboro filtrų* metodas, pasiūlytas 1946 metais ir sparčiai išpopuliarėjęs 2000–2016 metais. Šis algoritmas paremtas *Gauso* ir sinuso bei kosinuso eksponentės $s(x, y)$ sandauga. Naudojama eksponentė turi 2 parametrus: kryptį ir dažnį. Pagal nutylėjimą *Gaboro* filtrus naudojančios veido atpažinimo sistemos pritaiko 8 krypčių ir 5 mastelių filtrus. Šitaip sudaromi regionų požymių rinkiniai, išsidėstę apie centrinį *Gauso* filtro tašką. Šie požymiai vadinami *Gaboro* bangelėmis (angl. *Gabor wavelets*), o jų visuma apibūdina nuotrauką. Veido atpažinimo metu lyginamos apmokymo nuotraukų ir testavimo nuotraukų bangelės. Pagrindinis *Gaboro* filtrų metodo trūkumas – lėtas apskaičiavimo laikas.

Nepaisant nuolatinio esamų sistemų tobulinimo ir naujų metodikų kūrimo, veidų atpažinimo algoritmai nėra tobuli ir praktikoje susiduria su daugybe problemų. Pavyzdžiui, atliekant asmens paiešką duomenų bazėse yra itin svarbus informacijos apdorojimo greitis, o šiuolaikinių socialinių tinklų saugomas nuotraukų kiekis yra milžiniškas. Sprendžiant šią problemą būtina atpažinimo ir klasifikavimo sistemų optimizacija. Net naudojant nedidelį apmokymo ir testavimo nuotraukų masyvą, veido atpažinimo algoritmai gali susidurti su šiomis problemomis:

1. **Veido bruožų analizė, nepriklausoma nuo vartotojo pozos** (angl**.** *Poze Invariant Face Recognition – PIFR*) problema. Realiame pasaulyje užfiksuojamos nuotraukos dažniausiai yra gana mažos rezoliucijos, o asmenys nežiūri tiesiai į kamerą. Bet kuriuo laiko momentu subjekto galva gali būti pasukta visomis trimis ašimis neapibrėžtu kampu. Taip pat verta paminėti, kad žmogus gali būti nusisukęs

nuo kameros daugiau kaip 90 laipsnių kampu – šiuo atveju susiduriama su itin didele duomenų praradimo problema. Negana to, jei nagrinėjamas paveikslėlis ar video failas yra mažos rezoliucijos, dėl didelės pikselizacijos yra itin sunku aptikti dominuojantį tašką ar atlikti paveiksliuko deformacijas, kaip pavyzdžiui, atsukti veidą į priekį. Veido posūkio kampo normalizacija yra pagrindinis metodas siekiant išspręsti *PIFR* uždavinius.

Didžiausias šių algoritmų plėtojimas prasidėjo 2009 metais, kai dėmesys buvo sutelktas į orientyrais pagrįstos sistemos tipo algoritmus (angl. *Landmark Detection Free*). Jie išnaudoja *SIFT* tipo detektorius, kurie padeda išskirti pasikartojančius taškus nuotraukose, nepaisant didelių nagrinėjamo atvaizdo deformacijų. Pagrindinė šių algoritmų problema yra laiko sąnaudos. Jei siekiama itin didelio tikslumo palyginimo metu – kas yra privaloma saugumo srityje – procesas gali trukti nuo keleto minučių iki keleto valandų, priklausomai nuo pradinės nuotraukos informacijos ir duomenų bazės dydžio. Savaime suprantama, kad toks resursų naudojimas yra neoptimalus.

Iki 2015 metų nuo pozos priklausančių veidų atpažinimo metodų tikslumas buvo gana mažas dėl aukščiau įvardintų problemų. 2016 metais *Facebook* kompanija, naudodama savo nuotraukų duomenų bazes, sukūrė pakankamai tikslius rezultatus pateikiantį algoritmą. Testų metu šis metodas 83 % tikslumu sugebėdavo atpažinti žmogaus veidą, net jei nuotraukos kokybė buvo itin prasta ar dalis veido buvo uždengta. Tyrimui buvo panaudota 40 000 nuotraukų, o algoritmas apmokytas panaudojus beveik 250 milijardų nuotraukų, esančių *Facebook* serveriuose [56]. Verta paminėti kad po šios įmonės sėkmės naujai sukuriamų *PIFR* metodų tikslumas dažnai viršija 80 % tikslumo ribas, tačiau atlikti eksperimentai dažnai nebuvo taikyti ekstremaliomis sąlygomis, kai susiduriama su dviem ar daugiau šiame sąraše įvardytų problemų [11].

2. **Veido išraiškų atmetimas.** Veido išraiška yra matoma psichologinės žmogaus būsenos, intencijų, asmenybės ir atliekamos veiklos reprezentacija. Analizuojamas subjektas bet kuriuo laiko momentu gali šypsotis, būti susiraukęs ar rodyti kitokias veido išraiškas nei ramybės būsenoje. Be to, analizei gali būti naudojama video medžiaga ar nuotrauka, kurioje žmogus kalba. Tokiu atveju nagrinėjamą pradinę informaciją iškraipo ne tik emocijos, kurios lydi tariamus žodžius, bet ir paties

garso tarimo vizuali reprezentacija – tariant raides, keičiasi žandikaulio forma, raumenų pozicijos bei gali susidaryti vietiniai šešėliai, kurių įtaka aprašyta paskutiniame šio sąrašo punkte [10].

Jei žmogus yra užsimerkęs ar itin stipriai iškreipęs savo veidą – pavyzdžiui išraiškingai pakėlęs vieną antakį ar perkreipęs lūpas – dauguma atpažinimo algoritmų pateiks klaidingus rezultatus. Pirmiausia nukenčia veido regionų aptikimas. Pavyzdžiui, jei algoritmas akies zoną išskiria remdamasis vyzdžio aptikimu, o viena ar abi akys nagrinėjamame paveiksliuke yra užmerktos, šie dominuojantys taškai nebus aptikti. Be to, programa gali būti ne pakankamai apmokyta atpažinti atitinkamą vartotoją, jei palyginimui naudojamos panašios jo rimties būsenos nuotraukos.

Šiuo metu nėra sukurto optimalaus algoritmo, skirto veido išraiškų atmetimui. Pagrindinė emocijos aptikimo ir šalinimo seka yra lygi veido atpažinimo sekai. Pirmiausiai metodui pateikiamas nagrinėjamas video failas arba nuotrauka, kurie dažniausiai yra pilkų atspalvių tipo. Po to seka pirminio apdorojimo etapas, kurio metu atliekamas triukšmo šalinimas, šviesos intensyvumo reguliavimas bei geometrinis standartizavimas kuris padeda išvengti posūkio, objekto dydžio ir poslinkio sukurtų nuokrypių. Geometrinė normalizacija dažniausiai atliekama remiantis nosies šnervių ar akies vyzdžių buvimo vieta. Šiame etape taip pat atliekama veido segmentacija į regionus, remiantis jų pozicija, spalva, forma ar kitais aspektais.

Trečias etapas – veido dalių savybių ištraukimas (angl. *Feature extraction*). Šiuo metodu taškinė pikselių informacija yra paverčiama aukštesnio lygio reprezentacija. Tai gali būti forma, judesys ar gylio informacija Šie duomenys apdorojami klasifikatoriais ir galiausiai programa priima sprendimą, ar žmogus yra rimties būsenos ar rodo kitokią emociją [10]. Jei emocija yra aptinkama, taikant matematinius modeliavimo procesus galima nustatyti apytikslią dominuojančių taškų buvimo vietą, kai subjektas neutralus. Tačiau šiuo atveju susiduriama su programos realizacijai skirto laiko problema. Dažniausiai atpažinimo procesą norima atlikti realiu laiku – iki 1 sekundės – tad sprendžiant veido išraiškų šalinimo problemą yra itin aktualus pirminio apdorojimo ir klasifikatorių greitis.

3. **Veidai gali būti uždengti kitais objektais**, kaip, pavyzdžiui, akiniais nuo saulės ar paprastais akiniais, kurie atspindi objektus ar apšvietimo srautą. Atpažinimui kenkia ir natūralūs dariniai, kaip plaukai ar barzda. Pavyzdžiui, programos apmokymui buvo naudotos švariai nusiskutusio vyro nuotraukos, o pateikiamose testavimo nuotraukose jis užsiauginęs barzdą. Taip pat dominuojančių taškų išskyrimui gali trukdyti kasdieniai objektai ar kiti asmenys ir jų veidai, kurie užstoja nagrinėjamą subjektą. Pagrindinis šios problemos aspektas yra tas, kad skirtingai nei veido posūkio kampo ar apšvietimo problemoje negalima padaryti pradinės nuotraukos normalizacijos. Analizei reikalingi duomenys yra visiškai prarandami dėl pašalinių objektų kaltės.

   Su šia problema susidoroti yra sunkiausia. Iki šiol buvo atlikti keli bandymai kaip išvengti didelio programinės sistemos sulėtėjimo, kai prarandama dalis veido atpažinimui reikalingos informacijos. Pirmasis algoritmuose naudojamas metodas yra veido suskaidymas į segmentus, kuris padeda nustatyti, kurie subjekto regionai yra dalinai uždengti. Toliau naudojami veido savybių grafikai, kurie aprašo priklausomybes tarp objektų, kaip, pavyzdžiui, atstumas tarp akių ar nosies pozicija lupų atžvilgiu. Remiantis šiais ir programos apmokymo duomenimis, uždengtiems veido plotams yra pritaikomi matematiniai modeliai, kurie padeda apytiksliai nustatyti dingusių savybių buvimo vietą ir charakteristiką. Iš esmės duomenų praradimo problema yra keičiama į informacijos atkūrimo ir objekto atpažinimo problemą [18].

   Atlikti tyrimai taip pat parodė, kad dėl duomenų praradimo stipriai nukenčia tradiciniai veidų atpažinimo metodai. Pavyzdžiui, tuo atveju kai veidą dengia akiniai nuo saulės nei *PCA* nei *LDA* algoritmai neviršija 50 % patikimumo ribos. O kai yra uždengiama apatinė veido dalis *PCA* patikimumas gali nukristi iki 27 % o *LDA* iki 44 % [18].

4. **Vaizdo ar foto kamerų skiriamosios gebos ir kokybės problema**. Gaunamos nuotraukos yra rastrinio pavidalo, tad žemos rezoliucijos nuotraukos ar video failai stipriai apsunkina analizės procesą. Jei nagrinėjamas subjektas nebuvo nufotografuotas ar nufilmuotas iš arti, yra didelė tikimybė, kad lyginimui reikalingi regionai bus nepakankamai detalūs dominuojančių taškų aptikimui. Dėl

pikselizacijos ypač nukenčia smulkios dalys kaip, pavyzdžiui, akys ar lūpų forma, nosies galiukas.

Ši problema itin aktuali kai veidų atpažinimą norima pritaikyti stebėjimo kamerose. Šiuo atveju pastebimi itin dideli duomenų praradimai ne tik dėl nuolat kintančios subjekto pozos, bet ir dėl triukšmo bei pikselizacijos. Šios problemos atsiranda dėl neapibrėžtų kintamųjų kiekio ir didelių atstumų tarp įrangos ir objekto [20, 45]. Šiai problemai spręsti taip pat nėra sukurto optimalaus algoritmo. Dažniausiai naudojamas metodas yra super rezoliucijos nuotraukų naudojimas kartu su identiškais itin mažos rezoliucijos duomenimis programos apmokymo procese. Taikant šį algoritmą yra naudojamas iteracinis procesas, kuris palaipsniui bando pritaikyti didelės pikselizacijos paveikslėlius prie aukštos kokybės nuotraukų. Tokiu būdu yra sudaromos matematinės transformacijų tarpusavio priklausomybės, kurios naudojamos asmens atpažinimui. Pirminio apdorojimo metu yra stengiamasi kaip įmanoma labiau modifikuoti pradinį paveikslėlį, kad atliekant lyginimą su aukštos rezoliucijos duomenimis būtų galima išskirti kuo daugiau veido regionų ir detalių [45].

Deja, šis normalizacijos metodas patikimai veikia tik kontroliuojamoje aplinkoje. Vaizdo ir foto kamerų skiriamosios gebos problema yra itin aktuali – 2014 metais atlikto tyrimo metu paaiškėjo trys esminiai faktoriai, turintys didžiausią įtaką. Tai yra atstumas tarp foto kameros ir subjekto, kameros tipas ir jos jutiklių jautrumas bei skiriamosios gebos parametrai. Apšvietimo, judesio greičio ir pozos variacijos šiai problemai turi daug mažiau įtakos [45].

5. **Apšvietimo problema**. Veidai gali būti pilnai arba dalinai apšviesti, o tai įtakoja naudojamų algoritmų kokybę. Siekiant pagerinti atpažinimo programų rezultatus yra būtina atsižvelgti į du esminius problemų šaltinius: vietinius ir globalius šešėlius.

Vietiniai šešėliai gali vizualiai pakeisti individualių veido bruožų, pavyzdžiui, akių, lūpų ar nosies formas. Taip pat, jie prisideda prie veido zonos ribų iškraipymo [42]. Globalūs šešėliai gali ne tik užmaskuoti atskiras veido dalis, bet ir jį patį sulyginti su paveikslėlio fonu. Keičiant šviesos kitimo kryptį, keičiasi ir šešėlių dydis bei pozicija. Taip pat pakeičiamas odos atspindžių ryškumas ir

bendras nuotraukos kontrastas, kas daro įtaką daugumai veido atpažinimo algoritmų.

Nors kompiuterinėse regos sistemose ši problema yra itin aktuali ir nuo jos kenčia standartiniai veidų atpažinimo algoritmai, optimalus apšvietimą normalizuojantis metodas dar nebuvo sukurtas. Veidų atpažinimui siūloma naudoti su apšvietimu nesusijusias charakteristikas – objekto kraštines, tūrį, dominuojančių taškų pozicijas. Nagrinėjant šią kompiuterinės regos problemą buvo sukurta daugybė histograminių ir fotometrinių metodų, kurie naudojami išankstiniam nuotraukos normalizavimui.

Siekiant sulyginti paveiksliuke esančius kontrastus, bei pagerinti nuotraukų kokybę, 1977 metais R. A. Humelis pasiūlė naudoti histogramų transformacijas (angl. *Histogram equalization – HE*). Tai yra vienas labiausiai paplitusių apšvietimo normalizavimo metodų, kurio metu paveiksliuko pikselių reikšmes bandoma paskirstyti taip, kad jos būtų vienodos visoms ryškumo reikšmėms [39]. 1987 metais paplito adaptyvūs histogramos išlyginimo metodai (angl. *Adaptive histogram equalization – AHE*), kurie išsprendė savo pirmtako greičio problemą, bei pagerino smulkių kontrastų išskyrimą. 1987 metais pasirodė *CLAHE* metodas, kuris plačiai naudojamas ir šiuolaikinėse biometrinėse sistemose [16]. 2010 metais M. Leščinskis susitemino pagrindinius histogramų normalizavimo metodus ir pateikė jų tikslumo statistikas [29].

2003 metais S. Shan pasiūlė keturių regionų histogramos išlyginimo metodą, pagrįstą *Gama* korekcija. 2010 metais X. Tanas ir B. Trigsas *Gama* korekcijas pritaikė išankstinio nuotraukos apdorojimo metu. Šis normalizavimo metodas dėl savo paprastumo tapo populiariu 2013–2016 metų mokslinių darbų objektu [7, 38]. Fotometrinės apšvietimo normalizacijos yra taip pat plačiai nagrinėjama sritis. 1997 metais D. J. Džobsonas pasiūlė *vieno mastelio tinklainės modeliavimo – SSR* apšvietimo normalizavimo algoritmą, bei jį papildantį *MSR* metodą. *SSR* ir *MSR* naudojamas *Gauso* kernelis yra skirtas įvertinti regione esančius dominuojančius pikselius, o logaritminė transformacija padeda suspausti pikselių reikšmių dinaminį diapazoną ir šitaip panaikinti apšvietimo triukšmą [9,13]. *Tinklainės* tipo algoritmai yra plačiai pritaikomi šiuolaikinėse kompiuterinės regos sistemose [2].

2004 metais H. Vangas remdamasis t*inklainės* modeliais, suformavo *SSQ* algoritmą. Esminė metodo idėja yra padalinti paveiksliuką iš jo paties išblukintos versijos. Apšvietimas normalizuojamas – nuotraukoje lieka tik išsiskiriantys pikseliai ir nedideli nuokrypiai. Kaip ir *Tinklainės* atveju, siūlomas metodas turi jį papildantį *MSQ* modelį, kuris normalizuoja nuotrauką ir padeda išskirti dominuojančius taškus naudodamas skirtingo dydžio *Gauso* kernelius [30].

Kiti efektyvūs ir dažnai naudojami sprendimo metodai: šviesos intensyvumo reguliavimas, tikrinės erdvės (angl. *Eigenbase*) skaidymas, duomenų bazės išplėtimas bei homomorfinis filtravimas [37] taikant diskretinę kosinuso transformaciją. Apmokymo duomenų bazės išplėtimo atveju veido atpažinimui naudojami dideli duomenų rinkiniai, kuriuos sudaro skirtingai apšviesti galvų modeliai ir ieškomo asmens nuotraukos. Taikant kosinusų transformacijas iš nagrinėjamo paveiksliuko yra pašalinamas žemo dažnio triukšmas, o tai paryškina esmines nuotraukoje esančio veido dalis. Tikrinės erdvės skaidymas dirba su matricos transformacijomis ir išnaudoja tikrinių veidų metodą, kuris aprašytas 2.5.1 darbo skyriuje. Itin populiarus ir efektyvus yra pirmasis paminėtas variantas – šviesos intensyvumo reguliavimas. Šis metodas pritaiko *Gama* korekcijas, kontrasto intensyvumo reguliavimą bei *LBP* atpažinimo algoritmą.

1.3. Veido atpažinimo programinės įrangos apžvalga

Praktikoje veidams aptikti ir atpažinti dažniausiai naudojama profesionali programinė įranga: *Matlab* programa ir *OpenCV* biblioteka. ***Matlab*** – aukšto lygio inžinerinio ir mokslinio skaičiavimo kalba, skirta algoritmų kūrimui ir testavimui, duomenų modeliavimui, simuliavimui ir sistemų prototipų kūrimui. Ši programa taip pat naudojama duomenų analizei, vizualizavimui, kompiuterinės regos sistemų realizavimui ir mokslinių programų grafinių vartotojo sąsajų kūrimui. *Matlab* programinės įrangos sistemą sudaro penkios pagrindinės dalys [52]:

1. ***Matlab* programavimo kalba** – tai aukšto lygio objektiškai orientuota masyvų ir matricų apdorojimo kalba, kontroliuojanti duomenų įvesties, išvesties, ir jų struktūros modeliavimą. Ši kalba tinkama tik *Matlab* aplinkai ir leidžia sukurti tiek mažos apimties prototipines programėles, tiek išbaigtas specializuotas programas. Šia kalba parašytas programas galima sukompiliuoti kaip portabilų *C* arba *C++* programinį kodą, tačiau yra keletas apribojimų. Ne visi *Matlab* naudojami

algoritmai ir įrankiai yra palaikomi perkėlimo metu. Visiškai nepalaikoma grafikos sistema, taip pat apriboti įvesties ir išvesties failai. Dar viena problema – *Matlab* kalba parašytos programos greitis. Sukompiliuota *C* programa šiuo atžvilgiu gali būti daug pranašesnė nei jos atitikmuo *Matlab* aplinkoje [52].

2. **Darbinė aplinka** – grafinė vartotojo sąsaja, kuri palengvina darbą su *.m* failais. Kaip ir visose standartinėse aplinkose, *Matlab* integruotas duomenų įkėlimas, išsaugojimas, įrankių juostos bei vartotojui draugiška klaidų aptikimo ir taisymo (angl. D*ebug*) sistema.
3. **Grafikos sistema** – *Matlab* turi vidinę aukšto lygio *2D* ir *3D* duomenų atvaizdavimo sistemą. Ji palaiko duomenų vizualizavimą, animaciją, paveiksliukų apdorojimą ir grafikų atvaizdavimą. Taip pat, šią sistemą galima pritaikyti kuriant visiškai funkcionalią grafinę vartotojo sąsają, skirtą duomenų vizualizavimui savarankiškoje *Matlab* programoje.
4. **Matematinių funkcijų biblioteka** – ją sudaro didelis skaičiavimo algoritmų rinkinys. Į jį įtrauktos praktikoje dažnai naudojamos funkcijos, tokios kaip sinusų ar kosinusų apskaičiavimas, bei palyginti sudėtingos matematinės išraiškos, tokios kaip *Furjė* transformacijos ar matricų tikrinių vektorių apskaičiavimas. Taip pat, šioje sistemoje yra realizuoti pagrindiniai kompiuterinės regos algoritmai.
5. ***Matlab API*** – leidžia eksportuoti *Matlab* programinį kodą *C* arba *FORTRAN* kalbomis. Taip pat leidžiamas *Matlab* funkcijų iškvietimas iš išorinių programavimo kalbų: *C*, *C++, .NET*, *FORTRAN*, *JAVA* ir *Python*. O nuo *2015b* programos versijos ir atvirkštinis variantas – išorine kalba parašytų bibliotekų pritaikymas *Matlab* programoje.

Industrijoje *Matlab* naudojamas kaip standartinis programinis paketas, skirtas atlikti didelius skaičiavimų kiekius ar itin didelės apimties mokslines analizes. Tai gremėzdiškas komercinis paketas, kurio standartinės licenzijos kaina siekia 2000 eurų. Į šią kainą neįskaičiuotos papildomos bibliotekos. Pagrindinis šio programinio paketo privalumas – itin gera dokumentacija, programos palaikymas ir greitas reagavimas į klaidų pranešimus.

Atsižvelgiant į šio rašto darbo reikalavimus, galima teigti, kad *Matlab* yra daugiau nei pajėgus atlikti visas suformuluotas užduotis. Veido aptikimui yra siūlomas *Viola–Johnes*

algoritmas, o analizei atlikti galime pritaikyti vieną iš standartinių metodų: *PCA, LDA* arba *LBP*. Pagrindinė problema yra šio programinio paketo kaina bei ribotas portabilumas tarp skirtingų sistemų.

OpenCV – (angl. *Open source Computer Vision*) nemokama atviro kodo kompiuterinės regos ir vaizdų apdorojimo biblioteka, paremta *BSD* licenzija. Ji buvo sukurta siekiant palengvinti komercinių produktų kūrimą bei paspartinti kompiuterinės regos srities programų tobulinimą. *OpenCV* turi per 2500 klasikinių bei optimizuotų, naujos kartos algoritmų, skirtų vaizdo apdorojimui. Dauguma algoritmų yra skirti objektų sekimui ir atpažinimui video failuose, objektų ir žmonių klasifikavimui, *3D* objektų modeliavimui naudojant *2D* grafikos objektus, super rezoliucijos nuotraukų kūrimui bei veidų aptikimui ir atpažinimui.

Praktinis sistemos pritaikymas įgyvendinamas tiek mėgėjiškuose startuolių projektuose, tiek profesionalų sukurtose sistemose. Šią biblioteką naudoja tokios kompanijos kaip *Google*, *Intel*, *IBM*, *Sony* ir *Microsoft*. Pagrindiniai rinkoje naudojami *OpenCV* algoritmai yra pritaikomi nuotraukų sujungimams *Google Street view* programoje bei video klastojimo (angl. *Video tampering*) aptikimo programų realizacijoje. Kinijoje ši biblioteka panaudota norint efektyviai prižiūrėti kasybos įrangos kokybę, robotikoje *OpenCV* padeda užtikrinti sėkmingą automatizuotų mašinų navigaciją bei objektų atpažinimą. Fabrikuose šios bibliotekos pagalba yra skenuojamos prekių etiketės ir valdomas gamybos ciklas, o pastaraisiais metais, daug dėmesio skiriama biometrinių sistemų kūrimui [54].

OpenCV biblioteką palaiko pagrindinės operacinės sistemos: *Linux*, *Windows* bei *MAC OS*, taip pat *Android OS* ir *iOS* įrenginiai. *OpenCV* programinį kodą galima rašyti su *C*, *C++*, *Java*, *Python* programavimo kalbomis, palaikoma ir darbinė *Matlab* aplinka. Kadangi biblioteka parašyta *C++* programavimo kalba, jos greitis yra pakankamas realaus laiko aplikacijų funkcionalumo užtikrinimui. Be to, programinis kodas parašytas taip, kad esant galimybei naudotų *SSE* (angl. *Streaming SIMD Extensions*) ir *MMX* (angl. *MultiMedia eXtention*) instrukcijas. *MMX* procesoriaus panaudojimas gali paspartinti sistemos darbą su multimedijos produkcija net iki 60 %. *SSE* technologija leidžia siųsti didesnį duomenų kiekį panaudojant tik vieną instrukciją, tai paspartina apskaičiavimus susijusius su kompiuterine *3D* grafika. Šias sistemas sukūrė Intel korporacija. Verta paminėti ir tai, kad naujose *OpenCV* versijose pasirodžiusiose nuo 2010 metų pridėtas *CUDA* palaikymas, kuris užtikrina spartesnį

vaizdų apdorojimą. Apskaičiavimai atliekami naudojant grafinę vaizdo plokštę vietoj *CPU* gali pagreitėti nuo šešių iki trisdešimties kartų.

Atsižvelgiant į šio rašto darbo reikalavimus, galima pabrėžti, kad ši biblioteka pajėgi atlikti visas suformuluotas užduotis. *OpenCV* turi integruotus tris klasikinius algoritmus, skirtus veidų atpažinimui – *PCA*, *LDA* ir *LBPH* bei *Haar* požymių filtrų rinkinį, skirta veido bei jo dalių aptikimui ir išskyrimui [53, 54].

1 lentelė. *Matlab* ir *OpenCV* palyginimas

	Matlab	*OpenCV*
Standartinės licenzijos kaina	~2000 eurų	Nemokama
Platformos	• *Linux* • *Windows* • *MacOS*	• *Linux* • *Windows* • *Mac OS* • *Android* • *iOS*
Grafinė vartotojo aplinka	Yra	Nėra
Tiesioginis išmaniųjų įrenginių palaikymas	Nėra (būtina *C* kodo kompiliacija ir pernešimas, ribotas įrankių pasirinkimas)	Yra
Programavimo kalba	• *Matlab* • galimas *C*, *C++*, *FORTRAN*, *.NET*, *Python*, *Java* susiejimas su *Matlab* aplinka.	• *C* • *C++* • *Java* • *Python*
Minimalūs sistemos reikalavimai	• *Intel* arba *AMD* x86–64 procesorius • Disko talpa: 4–6 GB • Atmintis: 2 GB, arba 4 GB jei naudojamas *Simulink*	• *Intel* arba *AMD* x86–64 procesorius x86_64 • Disko talpa: 2–3 GB
	Veido atpažinimo technologija	
Siūlomi veido aptikimo algoritmai	• *Viola–Jhones*	• *Viola–Jhones*
PCA, *LDA*, *LBP* palaikymas	Yra (reikalingas programavimas)	Yra (kaip standartinis metodas)
Skaičiavimų greičio akceleracija	Yra	Yra

Kaip matoma 1 lentelėje, abi dažniausiai naudojamos nuotraukų analizės programos integruoja visus bazinius veidų aptikimo ir atpažinimo algoritmus. Abi sistemos turi skaičiavimų akceleracija, o veido aptikimui nuotraukoje siūlo vienodus metodus. Kadangi abi programos tinkamos iškeltai užduočiai realizuoti, priimant sprendimą atsižvelgta į programas papildančius parametrus.

Matlab programinė įranga yra komercinė. Kitaip nei *OpenCV* atveju, tai užtikrina greitą programos klaidų taisymą ir visos sistemos palaikymą. Be to, *Matlab* turi grafinę vartotojo sąsają kuri palengvina darbą su programa. Kita vertus, *OpenCV* yra greitesnis ir labiau optimizuotas, taip pat reikalauja mažesnių sistemos resursų. Ši biblioteka labiau taikoma realaus laiko aplikacijų kūrimui ir puikiai veikia išmaniuosiuose įrenginiuose. Dar vienas privalumas – tai yra nemokama programa, palaikoma ir modifikuojama bendruomenės. Dėl šių priežasčių baigiamojo darbo užduotims realizuoti buvo pasirinkta *OpenCV* biblioteka.

1.4. Apšvietimo normalizavimo modelio apžvalga

Apšvietimo normalizavimo uždavinys yra polinominis. Daugybė egzistuojančių veido atpažinimo algoritmų praktikoje dažniausiai naudoja *PCA*, *LDA*, *LBP*. Šie metodai yra pakankamai jautrūs apšvietimo variacijoms, o standartinis apšvietimo normalizavimas dažniausiai vykdomas iš karto visam paveiksliukui, arba išskiriamos tik didelio dažnio komponentės.

Sprendžiant šią problemą 2008–2011 metais X. Xie savo moksliniuose darbuose iškėlė pagrindinę šių procedūrų problemą [44]. Atlikti eksperimentiniai tyrimai parodė, kad asmens identifikavimui gali pakakti vien žemo dažnio komponenčių analizės. Esant nedideliam šviesos triukšmui, atpažinimo tikslumo koeficientai tokiu atveju gali siekti 85–96 %, o ekstremaliomis sąlygomis ~10 %.

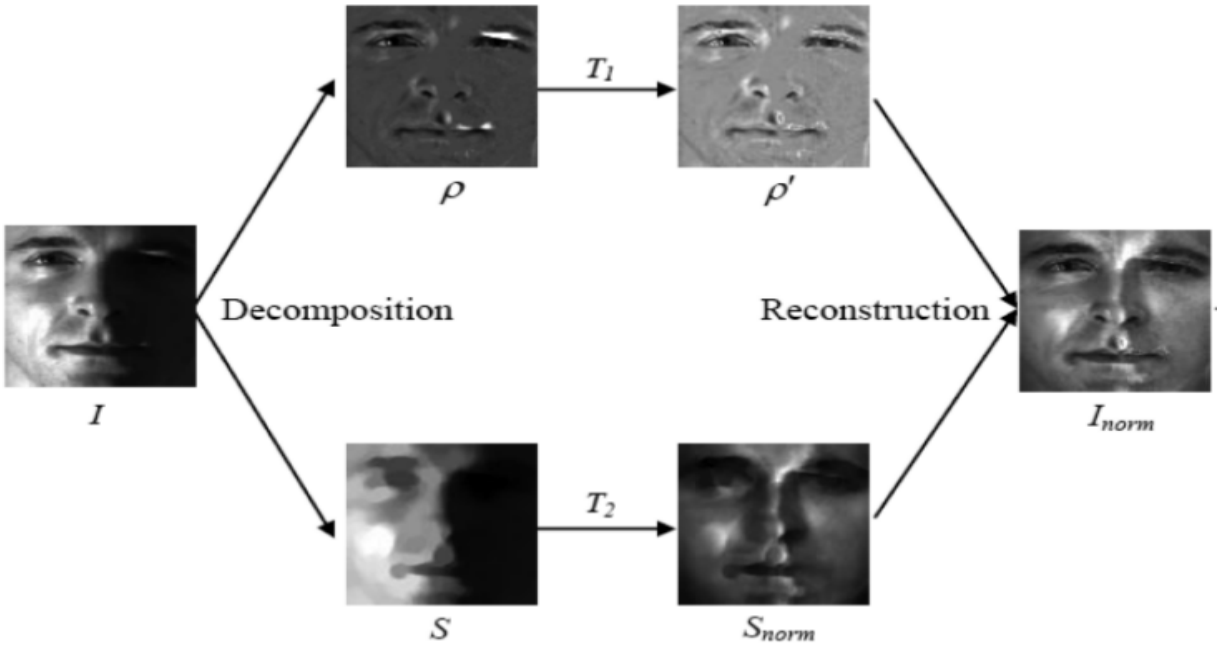
Remdamasis šiais tyrimais X. Xie apšvietimo normalizavimo problemą siūlo spręsti pritaikant paveiksliuko dekompozicijos metodą. Naudojant šį algoritmą, nagrinėjamai nuotraukai pirmiausiai yra pritaikoma logaritminė transformacija – sandaugos modelis konvertuojamas į sudėties modelį (1). Tai ne tik suteikia galimybę panaudoti standartinius dekompozicijos metodus, bet ir iš dalies pašalina apšvietimo triukšmą [44].

$$\log(I(\mathrm{x,y}))=\log(p)+\log(S)=\mathrm{v}+\mathrm{u} \qquad (1)$$

Po šios transformacijos yra pritaikomas LTV (angl. *Logarithmic total variation*) modelis (2).

$$\hat{u} = \operatorname{argmin}\left[\int |\nabla \hat{u}| + \lambda \|f - u\|\right]_{L^1}$$
$$\hat{v} = \mathrm{f} - \hat{u} \tag{2}$$

Veido nuotrauka yra išskaidoma į nedidelius požymius (angl. *Small–scale features*) bei didelius požymius (angl. *Large–scale features*) turinčias dalis $p \approx \exp(\hat{v})$, $S \approx \exp(\hat{u})$ ir kiekviena jų normalizuojama atskirai. Siūlomo metodo vizualizaciją pateikta 2 pav.



2 pav. X. Xie normalizavimo modelis

LTV modelis yra tinkamas kai reikalingas nuotraukose esančių linijų išsaugojimas, tačiau ekstremaliomis sąlygomis įmanomas pikselių peršvietimo susidarymas. Siekiant išspręsti šią problemą, aukšto dažnio komponentėm yra pritaikomas vidurkinimo filtravimas. Naudojant NxN dydžio kernelį, nagrinėjamame plote esantys pikseliai, kurių reikšmės yra didesnės už nustatytą slenkstį Q, yra suvidurkinami priklausomai nuo lokalių reikšmių. Žemo dažnio komponentėms normalizuoti siūloma naudoti diskrečiąją kosinuso transformaciją (angl. *Discrete cosine transformation*) ir aukštų dažnių filtrą [44].

1.5. Skyriaus apibendrinimas

Kompiuterinės regos uždaviniai ir juos sprendžiančios sistemos susiformavo 1960–1970 metais. Šių programų pritaikymo spektras apima pramogų, kino industrijos, asmens ir visuomenės apsaugos, medicinos, mokslo ir verslo sritis. Dauguma sukurtų programų yra pagrįstos lokalių ir globalių požymių išskyrimu ir aptikimu. Optimizuojant algoritmus šiuos požymius dažnai siekiama padaryti padaryti nepriklausomais nuo posūkio ar mastelio variacijų, pirminio apdorojimo metu taikomos nuotraukų normalizacijos.

Veidams atpažinti praktikoje dažnai naudojami standartiniai *Tikrinių veidų*, *Fišerio veidų* ir *Vietinio dvejetainio modelio* metodai. Šie algoritmai taip pat pagrįsti lokalių bei globalių požymių palyginimu. Praktikoje veido atpažinimo programos susiduria su 5 pagrindinėmis problemomis: galvos posūkio kampo, emocijų atmetimo, vaizdo ir foto kamerų skiriamosios gebos, informacijos praradimo dėl kitų scenos objektų ir apšvietimo problemomis. Norint padidinti veido atpažinimo koeficientą, yra būtina normalizuoti nagrinėjamą nuotrauką, o apšvietimo problema yra mažiausiai priklausoma nuo turimos įrangos ir visada sutinkama ne valdomoje aplinkoje.

Apšvietimo normalizavimas yra polinominis uždavinys, kuriam spręsti sukurta daug histograminių ir fotometrinių metodų. Dauguma jų laikomi standartiniais metodais, kurie nuolat optimizuojami ir tobulinami, siekiant gauti kuo geresnius algoritmo tikslumo koeficientus. 2008–2011 metais X. Xie iškėlė šių algoritmų problemą ir pastebėjo, kad didelio mastelio požymių zonos turi svarbios informacijos, kuri padeda pagerinti veido atpažinimo rezultatus. P. Sinha suformuluotos 19 biologinio atpažinimo tezių patvirtina šią teoriją ir gali būti pritaikytos naujam apšvietimo normalizavimo metodui.

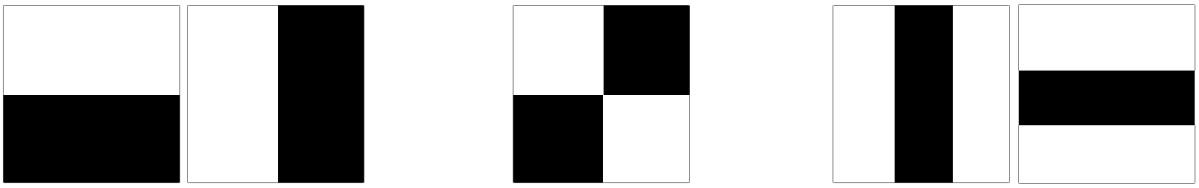
Kuriamo metodo realizavimui ir testavimui galima pritaikyti vieną iš 2 dažniausiai naudojamų kompiuterinės regos ir analizės programų – *Matlab* arba *OpenCV*. Programiniam kodui realizuoti siūloma naudoti atviro kodo *OpenCV* biblioteką, dėl mažų sistemos reikalavimų bei kainos. Be to, ši programinė įranga turi integruotus *PCA*, *LDA*, *LBP* analizės modulius bei gali būti paprastai integruota skirtinguose įrenginiuose. Pagrindinis *OpenCV* privalumas – sistema sukurta realaus laiko kompiuterinės regos programoms realizuoti.

2. APŠVIETIMO NORMALIZAVIMO IR VEIDO ATPAŽINIMO ALGORITMAI

Šiame skyriuje aprašomi veido aptikimo ir atpažinimo metodikoje naudojami algoritmai bei pateikiamas siūlomas apšvietimo normalizavimo modelis. Aprašomi šiame algoritme naudojamos apšvietimo normalizacijos ir jų matematinės išraiškos. Pateikiamos *Vietinių dvejetainių modelių*, *Tikrinių* bei *Fišerio veidų* naudojamos metodikos.

2.1. Veido aptikimas

OpenCV biblioteka turi integruotą objektų atpažinimo klasę *features2d*. Šiame darbe nagrinėjami veidai aptinkami panaudojus *Haar* požymių filtro technologiją ir *Viola–Jones* algoritmą. *Haar* klasifikatorius apmokomas naudojant masyvą teigiamo (nuotraukos kuriose yra veidas) ir neigiamo rezultatų nuotraukų. Tada apskaičiuojamas požymių kernelis, gaunamas iš balto stačiakampio pikselių sumos atėmus juodo stačiakampio pikselius [55]. Galimi rezultatai matomi 3 paveiksle. Kairėje pusėje atvaizduoti kraštinių požymiai, centre – kvadrato, o dešinėje linijų požymiai:



3 pav. Kraštinių požymiai, kvadratiniai požymiai, linijų požymiai

Veido aptikimo metu susidariusių požymių kiekis yra per didelis, kad būtų racionalus kompiuterio sąnaudų atžvilgiu, tad optimizavimui yra naudojamas *AdaBoost* metodas. Klasifikatoriaus apmokymo metu, nuotraukų rinkiniams yra pritaikomi visi požymiai ir ieškomas mažiausiai klaidų paliekantis slenkstis. Praktiškai sukuriamas neuroninis tinklas – algoritmo pradžioje visoms nuotraukoms suteikiamas vienodas svoris. Po apmokymo ciklo, teisingai priskirtoms nuotraukoms svoriai yra padidinami ir apmokymo ciklas kartojamas. Šis procesas kartojamas ir paklaidos bei svoriai yra perskaičiuojami tol, kol pasiekiamas tenkinantis aptikimo tikslumo koeficientas [35].

Haar klasifikatorius yra sudarytas iš pasverto mažesnių klasifikatorių rinkinio, galutinis metodas veido aptikimui naudoja apie 6000 požymių. Verta paminėti tai, kad 200 požymių pakanka norint aptikti frontalų veidą 95 % tikslumu. Siekiant optimizuoti

kompiuterio resursus klasifikatorius netaikomas visiems paveiksliuko pikseliams. Tyrimo metu nuotrauka suskaidoma į paieškos zonas – bandoma atskirti fono ir kandidato į veidą plotus. Kiekvienam pikselių blokui pritaikoma tik dalis požymių paieškos, jei testas praeinamas, pridedama ir likusių požymių paieška. Jei tenkinami visi algoritmo etapai, programa nuotraukos dalį identifikuoja kaip veidą. $384 x 288$ pikselių dydžio nuotraukos analizavimas trunka 0,7 sekundės [55].

2.2. Standartinis veido nuotraukos normalizavimas

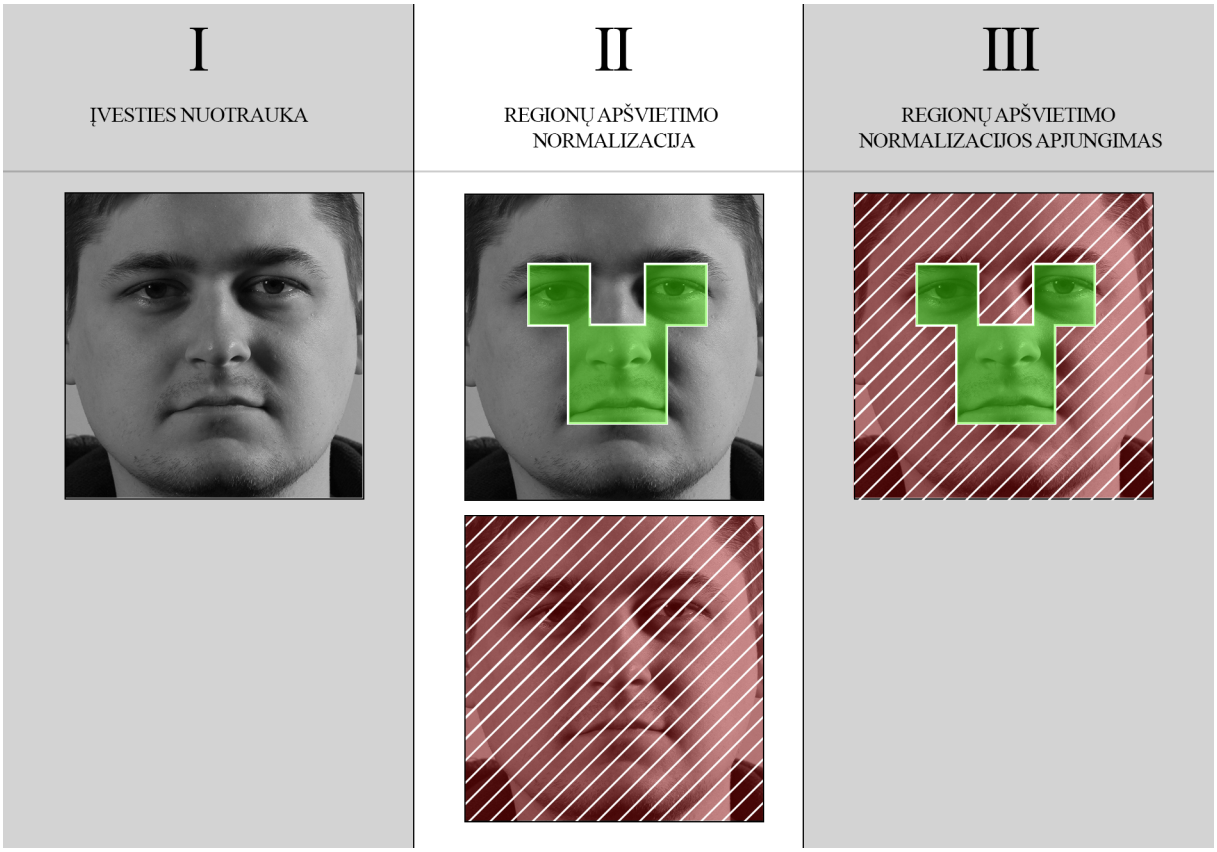
Prieš apmokant ar testuojant veidų atpažinimo programas, kiekvienam apmokymo ir testavimo nuotraukų masyvui yra būtina suvienodinti visus pradinius duomenis. Pirmiausiai yra nustatomas standartinio stačiakampio $n{\cdot}m$ dydis. Šis parametras privalo būti vienodas tiek apmokant algoritmus, tiek atpažįstant veidus.

Normalizavimo algoritmą galima skaidyti į keletą etapų: pirmiausia nuotraukoje yra aptinkamas asmens veidas. Tam naudojami *Haar* požymių filtrai. Išskirto dominančio kvadrato viduje tokiu pat metodu yra atliekama dviejų akių paieška. Apskaičiuojamos jų centrų x_1, x_2 ir y_1, y_2 koordinatės. Jei poslinkio kampas tarp y_1 ir y_2 yra santykinai mažas, nuotrauka nagrinėjama toliau. Kitu atveju vykdoma matricos transformacija arba nuotraukos pašalinimas iš apmokymo–atpažinimo duomenų masyvo.

Toliau aptinkama nosies galiuko pozicija, apskaičiuojamas tarpo tarp dviejų akių bei nosies centras. Atrandamas santykinis veido centras $P(x, y)$. Remiantis šia koordinate, brėžiamas standartinis veido stačiakampis. Jo dydis privalo būti vienodas visoms nuotraukoms, kaip ir atstumas nuo stačiakampio viršaus iki akių linijos. Veido centro x ašies koordinatė, visada bus standartinio stačiakampio pločio centre.

2.3. Siūlomas apšvietimo normalizavimo metodas

X. Xie siūlomo veido zonų dekompozicijos ir rekonstrukcijos modelio rezultatai [44] patvirtina teoriją, kad požymių atskyrimas gali pagerinti atpažinimo kokybę. Remiantis šiais pastebėjimais ir siekiant supaprastinti apšvietimo normalizavimo metodą, taikome prielaidą, kad kiekvieno asmens mažų požymių zonos yra išsidėsčiusios aplink pagrindines veido dalis – akių centrus, nosies bei lūpų regionus. Dideles veido požymių zonas sudaro kaktos sritis, skruostai ir smakras. Loginiame lygmenyje atskyrę šiuos blokus galime tiesiogiai atlikti šviesos triukšmo šalinimus skirtingoms nuotraukos dalims, o gautus rezultatus sujungti (4 pav).



4 pav. Siūlomas apšvietimo normalizacijos modelis

Nustatytoms užduotims realizuoti, darbo metu naudojama *OpenCV* biblioteka. Veido aptikimas ir iškirpimas nuotraukoje vykdomas pasitelkus standartinę *Haar* požymių filtro technologiją. Siekiant įgyvendinti darbo metu iškeltą užduotį – sugeneruoti standartinį veido stačiakampį bei normalizuoti apšvietimo parametrus – buvo panaudota 10 normalizacijos algoritmų ir 3 standartiniai veido atpažinimo algoritmai. Visi naudojami metodai yra aprašyti 2.4 šio darbo skyriuje.

2.4. Apšvietimo normalizacijos algoritmai

Veidams atpažinti naudojamas apšvietimo normalizacijas galima suskirstyti į fotometrines ir histogramines. Remiantis *Lamberto* modeliu, bet kurį natūralų vaizdą $I(x,y)$ matematiškai galime aprašyti kaip apšvietimo ir atspindžio sandaugą: $I(x,y)=L(x,y)*R(x,y)$ [29]. Šiame skyriuje yra aprašyti 10 algoritmų, išnaudojančių šią savybę ir padedančių normalizuoti apšvietimo triukšmą: vietinius bei globalius šešėlius. Šie algoritmai dažniausiai yra laikomi standartiniais apšvietimo normalizavimo metodais, kurie naudojami norint ištestuoti sukurtą normalizavimo metodą.

2.4.1. *Gama* korekcija

Gama nusako sąryšį tarp skaitmeninės pikselio spalvos išraiškos ir jo tikrojo skaisčio (angl. *Luminance*). Skirtingai nei žmogaus akis, fotoaparato jutikliai priima tiesioginį fotonų srautą, krentantį į matricą, o tai iškraipo gaunamą informaciją. *Gama* korekcija dar yra vadinama *Power Law* transformacija. Normalizuojant paveiksliuką, pirmiausia pikselių spalvos intervalai pakeičiami iš [0–255] į [0–1]. Tada kiekvienam nuotraukos pikseliui yra perskaičiuojama nauja reikšmė pagal (3) formulę [7, 38].

$$I(x,y) = 255.0 \cdot I(x,y)^{q} \tag{3}$$

2.4.2. *CLAHE* metodas

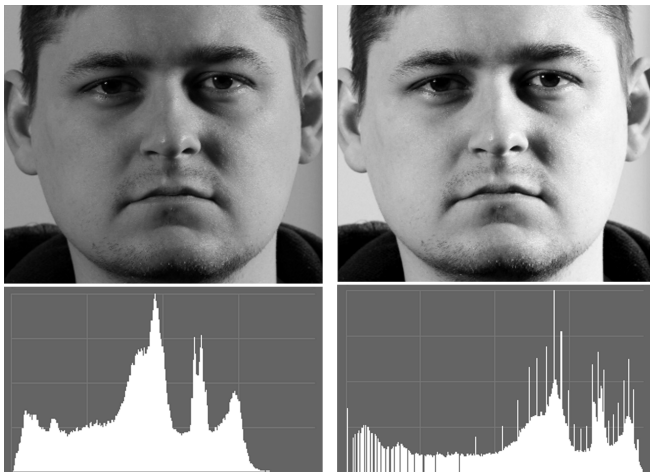
CLAHE, tai kompiuterinės regos algoritmas, skirtas nuotraukų kontrasto išlyginimui. Skirtingai nei įprastas Histogramos lyginimas – *CLAHE* metodas sukuria keletą histogramų atskiroms paveiksliuko vietoms. Tada perskirstomos skaisčio reikšmės. Šis metodas dažnai naudojamas kai norima išskirti nagrinėjamo objekto linijas ir vietinius kontrastus [16].

OpenCV programoje yra numatytas paveikslėlio suskaidymas į 8x8 dydžio stačiakampius. Kiekviename jų atliekama įprastinė histogramų normalizacija, su iš anksto nustatytais kontrasto limitais – 40. Jei dėl triukšmo ar kitų priežasčių apskaičiuotos pikselių reikšmės viršija limitą, jos yra paskirstomos per visą nuotrauką ir sulyginamos su nustatyta riba (5 pav.).

5 pav. *CLAHE* metodas

2.4.3. Histogramos išlyginimas

Histograma tai grafinis nuotraukos intensyvumo lygių pasiskirstymas, nurodantis kiek tam tikros rūšies pikselių yra paveiksle. Histogramos išlyginimas naudojamas siekiant pilkumo lygius padaryti pastovius ir išryškinti vietinius objekto kontrastus. Normalizacija realizuojama ištempiant histogramą – mažos pikselių reikšmės padidinamos, o dideli lokalūs klasteriai yra sumažinami iki kol gaunamas apylygis intensyvumo pasiskirstymas [29, 38, 39]. Kaip matoma 6 pav. – centre susidaręs pikselių reikšmių masyvas buvo išskleistas horizontaliai. *OpenCV* biblioteka turi integruotą metodą, skirtą histogramos normalizavimui.



6 pav. Histogramos išlyginimas

2.4.4. *Single–scale Retinex* modelis

SSR normalizacija vaizdą $I(x,y)$ laiko apšvietimo $L(x,y)$ ir atspindžio $R(x,y)$ sandauga. Siekiant sumažinti nuotraukos pikselių dinaminį diapazoną yra naudojamas *Gauso* funkcijos branduolys $F(x,y)$ ir logaritminės atimties metodas (4) [2, 13]:

$$I_{\mathrm{SSR}}(\mathrm{x,y}) = \log\left(I(\mathrm{x,y})\right) - \log\left[F(\mathrm{x,y}) * I(\mathrm{x,y})\right] \tag{4}$$

2.4.5. *Multi–scale Retinex* modelis

MSR yra papildyta *SSR* algoritmo versija. Šiuo atveju sujungiami keli skirtingų dydžių *NxN* Gauso filtrai su skirtingais svoriais w_n, kurių suma lygi vienetui. Bendruoju atveju *MSR* formulė užrašoma taip (5) [2, 13, 41]:

$$I_{\mathrm{MSR}}(\mathrm{x,y})=\sum_{\mathrm{n}=1}^{N} w_n\left[\log\left(I(\mathrm{x,y})\right)-\log\left(F(\mathrm{x,y})*I(\mathrm{x,y})\right)\right] \tag{5}$$

čia: $I(x,y)$ – pradinės nuotraukos matrica, $F(x,y)$ – *Gauso* filtru paveiktos pradinės matricos rezultatas, w_n – svoris.

2.4.6. *Single–scale Self Quotient* modelis

SSQ normalizacijos metodas apšvietimą analizuoja kaip žemo dažnio komponentę. Testuojama nuotrauka yra padalinama iš *Gauso* blukinimo metodu išfiltruotos savo kopijos [29, 30], o bendroji *SSQ* formulė (6) aprašoma:

$$I_{\mathrm{SSQ}}(\mathrm{x,y})=\frac{I(\mathrm{x,y})}{F(\mathrm{x,y})*I(\mathrm{x,y})} \tag{6}$$

čia: $I(x,y)$ yra nagrinėjamos nuotraukos matrica, o $F(x,y)$ – *Gauso* filtru išblukinta nuotraukos kopija.

2.4.7. *Multi–scale Self Quotient* modelis

SSQ metodo rezultatai tiesiogiai priklauso nuo teisingai parinkto *Gauso* filtro kernelio dydžio. Jei jis bus per mažas, nukentės normalizacijos procesas, o jei per didelis – ties požymių kraštinėmis gali susidaryti papildomo triukšmo [29,30]. Siekiant išspręsti šią problemą, galima panaudoti nelinijinę funkciją T su svorį w_n, turinčiais *Gauso* funkcijos kerneliais pagal (7) formule.

$$I_{\mathrm{MSSQ}}(\mathrm{x,y})=\sum_{\mathrm{n}=1}^{N} T\, w_n\left[\frac{I(\mathrm{x,y})}{F(\mathrm{x,y})*I(\mathrm{x,y})}\right] \tag{7}$$

2.4.8. Homomorfinis filtravimas

Šis metodas normalizacijos metu taip pat naudoja apšvietimo ir atspindžio modelį. Kadangi apšvietimas yra žemo dažnio komponentė, atliekamas filtravimas dažnių srityje. Pirmiausiai panaudojant natūrinio logaritmo funkciją yra atsiejami $L(\mathrm{x,y})$ ir $R(\mathrm{x,y})$. Tada, naudojant *Furjė* transformaciją FT nuotrauka yra konvertuojama į dažnių sritį. Panaudojamas

aukštų dažnių filtras ir įvykdoma atvirkštinė *Furjė* transformacija IFT pagal (8). Atlikus šią normalizaciją yra sukuriama analizės atlikimui paruošta veido nuotrauka [12, 25, 29].

$$\mathrm{I}'(\mathrm{x,y}) = \mathrm{e}^{\mathrm{IFT}(\mathrm{FT}(\log(I(\mathrm{x,y}))H))} \tag{8}$$

2.4.9. Pleksiformo imitacijos normalizacija

Pleksiformo imitacijos normalizacija yra atliekama siekiant suvidurkinti visų nuotraukos pikselių spalvas. Pirmiausiai nuotrauka yra išblukinama panaudojus platų *Gauso* žemų dažnių filtrą, o tada siaurą tokį patį filtrą [22]. Procesas yra lėtas, tačiau atsparus nuotraukos objekto posūkio variacijai. Normalizuotam vaizdui apskaičiuoti yra naudojamos (9) ir (10) formulės.

$$I(\mathrm{x,y}) = 0.5 + \mathrm{const}_1 \cdot \partial \tag{9}$$

$$I(\mathrm{x,y}) = 0.5 + \mathrm{const}_2 \cdot \left(\partial + \frac{\partial}{\sqrt{I(\mathrm{x,y})}} \right) \tag{10}$$

čia: 0.5 žymi vidutinę pikselių reikšmę, const_1 ir const_2 parenkama taip, kad veido pilkumo lygmenys kistų nuo 0 iki 1. Jei vaizdas aplink nagrinėjamą tašką $P(\mathrm{x,y})$ keičiasi nedaug, tai ∂ reikšmė bus arti vidurkio 0.5. Kvadratinė šaknis naudojama, jei norima išryškinti smulkesnes nuotraukos dalis.

2.4.10. Tano ir Trigso metodas

Tano ir Trigso *(angl. Tan and Triggs)* metodas apima keturis apšvietimo normalizacijos etapus. Pirmiausiai atliekama paprasta *Gama* korekcija – tamsios nuotraukos dalys pašviesinamos, šviesios nugesinamos. Toliau vykdoma pilkų lygių normalizacija. Atliekamas aukštų dažnių filtravimas taikant *Gauso* formulę. Trečiame žingsnyje galima, tačiau neprivaloma, atlikti maskavimo procesą. Jo metu yra pašalinami pertekliniai veido požymiai, kurie neaktualūs atpažinimo testavimo metu. Galiausiai vykdomas globalus kontrastų išlyginimas – šiame etape nuotraukoje vis dar gali būti dideli pikselių reikšmių

nuokrypiai, palyginus su jų aplinkos vidurkiu [43]. Šių regionų išlyginimas vyksta panaudojus dviejų veiksmų aproksimaciją, pagal (11) ir (12) formules:

$$I(\mathrm{x,y}) = \frac{I(\mathrm{x,y})}{\left(\mathrm{mean}\left(\left|I(\mathrm{x',y'})\right|^{a}\right)\right)^{1/a}} \tag{11}$$

$$I(x,y) = \frac{I(x,y)}{\left(mean\left(min\left(T, \left|I(x',y')\right|\right)^{a}\right)\right)^{1/a}} \tag{12}$$

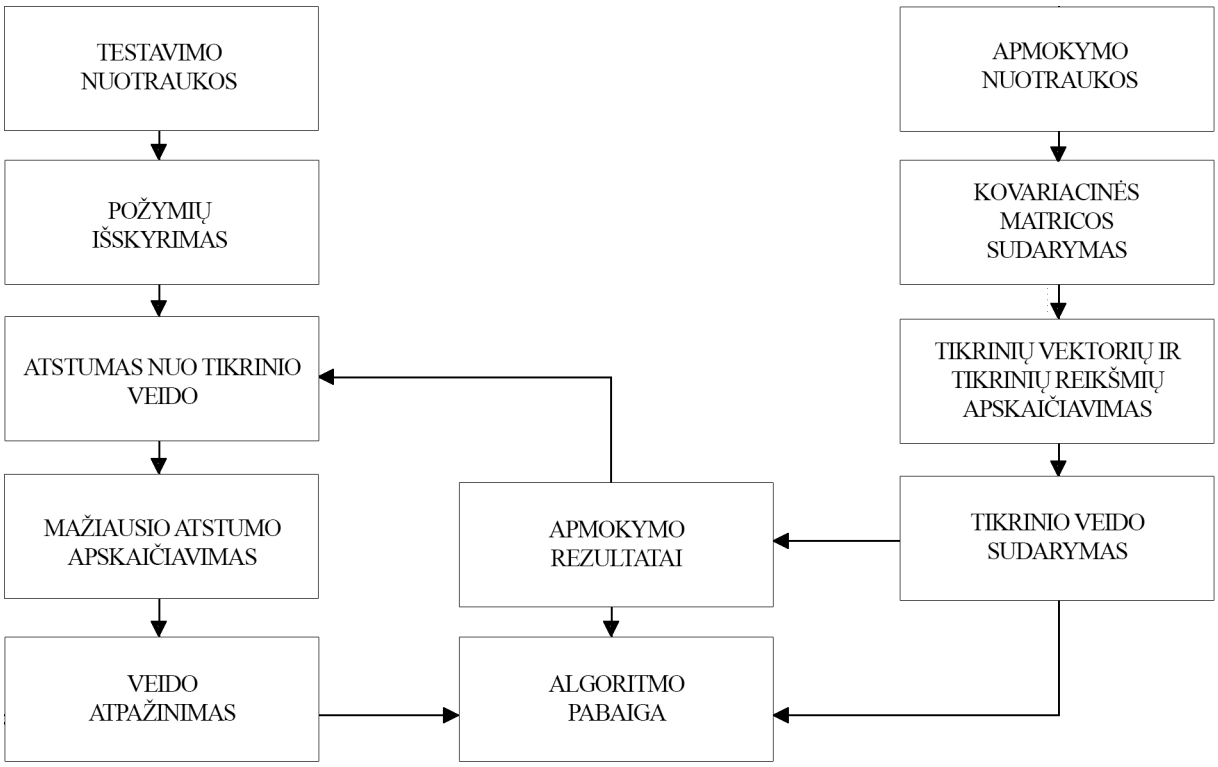
čia: a – aukšto suspaudimo lygio eksponentė, kuri sumažina aukštas reikšmes turinčius pikselius. T – slenkstis, kuris naudojamas po pirmos normalizacijos likusių išsiskiriančių pikselių reikšmių nurėžimui.

2.5. Veido atpažinimas

Veidų atpažinimui *OpenCV* bibliotekoje yra integruoti trys standartiniai algoritmai: *PCA*, *LDA* bei *LBP*. Kiekvienam jų apmokymo metu reikia pateikti minimaliai bent dvi vienodo dydžio $m \cdot n$ nuotraukas. Siekiant geriausių atpažinimo rezultatų, rekomenduojama naudoti standartinius nuotraukos ir apšvietimo normalizavimo metodus, aprašytus 2.4 skyriuje.

2.5.1. Tikrinių veidų metodas

Pagrindinių komponenčių analizė dar vadinama *PCA* yra viena dažniausiai taikomų požymių išskyrimo būdų. Veidų atpažinimo problemose jos dalinis variantas dar yra vadinamas *Tikrinių veidų* (angl. *Eighenfaces*) metodu. Tai statistinis metodas, naudojantis daugiadimensinius požymių vektorius. *Tikrinių veidų* algoritmas buvo pasiūlytas 1987 metais ir yra laikomas atskaitos tašku naujai sukurtų išankstinio apdorojimo algoritmų testavimui [5, 21, 31, 33]. Esminė algoritmo idėja yra iš nuotraukos ištraukti ir užkoduoti požymius, o tada palyginti juos su duomenų bazėje esančiu rinkiniu. Šio metodo loginė schema matoma 7 paveiksle.



7 pav. Tikrinių veidų apmokymo ir testavimo algoritmas

Algoritmo realizavimas yra gana paprastas. Analizuojamą nuotrauką $I(x, y)$ galima apibrėžti kaip $n \cdot n$ dydžio matricą su 8bitų intensyvumo reikšmėmis kiekvienoje celėje n. Tą pačią nuotrauką dar galime laikyti ir N^2 –dimencinės erdvės vektoriumi – paveiksliukas, kurio dydis 512x512 px gali būti išreikštas kaip taškas 262`144 dimensijos erdvėje, o panašių nuotraukų rinkinys susitelks į vieną klasterį. Gerai normalizuotų duomenų rinkiniai turėtų nepersidengti tarpusavyje.

Apmokymo metu tikrinio veido algoritmui pirmiausiai yra pateikiami nagrinėjamų subjektų nuotraukų rinkiniai. Matricas galime aprašyti $\Gamma = [\Gamma_1, \Gamma_2, \Gamma_{3,\ldots}, \Gamma_i]$ kur Γ_i atitinka apmokyme naudojamo paveiksliuko reikšmių vektorių. Norint išskirti tik tam subjektui būdingas savybes yra sukuriamas vidutinis veidas – visų Γ_n vektorių vidurkis (13).

$$\Psi = \frac{1}{M} \sum_{I=1}^{M} \Gamma_i \tag{13}$$

Kiekvienas duomenų rinkinyje esantis veidas skiriasi nuo vidurkio vektoriumi $\Phi_i = \Gamma_n - \Psi$. Toliau algoritme yra apskaičiuojamas ortonormalių vektorių rinkinys, geriausiai apibūdinantis informacijos pasiskirstymą. Panaudojant skirtumų matricą yra apskaičiuojama kovariacinė matrica C , pagal formulę (14).

$$C = A \cdot A^T = \frac{1}{M} \sum_{i=1}^{M} \Phi_i \cdot \Phi_1{}^T \tag{14}$$

Kovariacinė matrica C padeda surasti tikrinius duomenų vektorius u_k ir tikrines reikšmes λ_k . Pagrindinė problema ta, kad ši matrica yra labai aukštame dimensijos lygmenyje – $(N_x \cdot N_y) \cdot (N_x \cdot N_y)$. Jei analizuojama nuotrauka yra 512x512 tai jos kovariacinė matrica užims 262`144x262`144. Todėl būtinas skaičiavimų supaprastinimas. Pirmiausiai apskaičiuojami tikriniai vektoriai v_i iš $L = A \cdot A^T$ matricos, kurios dydis lygus MxM . Jeigu $Lv_i = \lambda_i v_i$, tai egzistuoja toks santykis (15):

$$\begin{aligned} A \cdot Lv_i &= \lambda_i A \cdot v_i \Rightarrow \\ AA^T Av_i &= \lambda_i A \cdot v_i \Rightarrow \\ CAA^T v_i &= \lambda_i A \cdot v_i \end{aligned} \tag{15}$$

Iš šios priklausomybės matome, kad v_i yra kovariacinės matricos tikriniai vektoriai. Ši savybė praktiškai sumažina apskaičiavimų kiekį iki subjektų nuotraukų masyvo dydžio ir nenaudojant daug resursų suformuoja tikrinių veidų rinkinius (16).

$$u_i = \sum_{k=1}^{M} v_{lk} \Phi_i l = 1,2,\ldots,M \tag{16}$$

Tikriniai veidai yra požymių rinkiniai, kurie apibūdina skirtumus tarp analizėje naudojamų nuotraukų [3,33]. Jų rinkinys yra išsaugojamas kaip masyvas $U = [u_{1,} u_{2,} u_{3,} \ldots u_n]_{NxR}$ kuris yra naudojamas veido atpažinimo procese. Šie apmokymo rezultatai dažniausiai yra išsaugomi išoriniame *.xml* tipo faile ir gali būti praplėsti esant poreikiui. Turint apmokymo duomenis, nauja veido nuotrauka gali būti transformuota į tikrinių vektorių komponentes pagal (17) formulę:

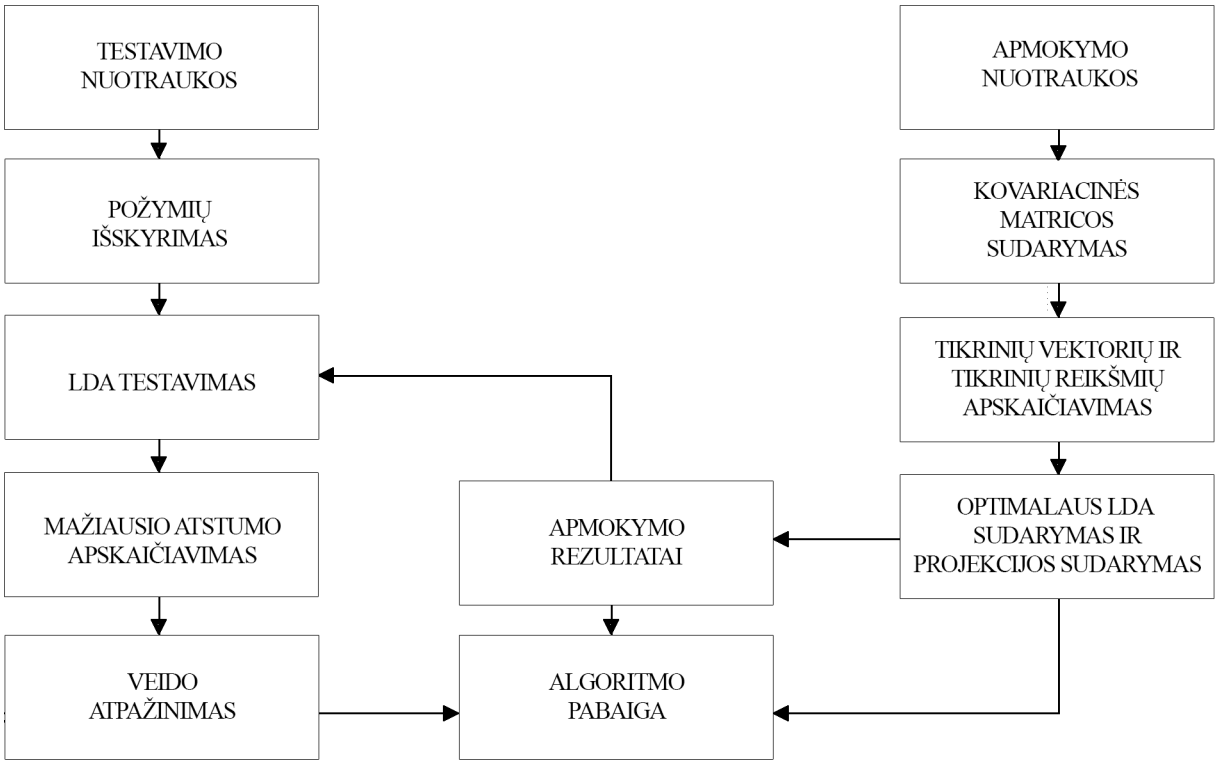
$$\Omega = U^T(\Gamma - \Psi) = [\omega_1, \omega_2, \omega_{3,\ldots}, \omega_n]_{Rx1} \quad (17)$$

Čia ω_i yra svoris, nurodantis kiekvieno tikrinio veido įtaką naujai pateikto veido nuotraukos transformacijai. Atpažinimo metu šis vektorius gali būti panaudotas norint rasti *Euklidinį* atstumą tarp analizuojamų ir duomenų bazėje esančių nuotraukų svorių vektorių $\mathcal{E} = min|\Omega - \Omega_i|$. Jei gautas atstumas tenkina sąlygą $\mathcal{E} < \theta$, kai θ yra nustatytas slenkstis, analizei pateiktas veidas yra laikomas atpažintu ir bus priskirtas atitinkamam subjektui.

Praktikoje *PCA* apmokymo procesas yra paprastai realizuojamas ir automatizuojamas. Nuotraukos verčiamos į statistinius duomenų rinkinius, šis metodas lengvai pritaikomas dideliems duomenų rinkiniams ir galimas pritaikyti realaus laiko aplikacijoms. Tačiau *PCA* labai jautrus apšvietimui ir dideliam duomenų išsiblaškymui dėl nuotraukos posūkio ar mastelio keitimo. Optimaliems rezultatams gauti yra būtinas fotografavimas kontroliuojamoje aplinkoje bei nuotraukoje esančio veido frontali pozicija [5].

2.5.2. Fišerio veidų metodas

Kitas praktikoje dažnai naudojamas veidų atpažinimo metodas yra *Fišerio veidų* (angl. *Fisher faces*) algoritmas. Šis metodas yra dalinis 1936 metais sukurto linijinio diskriminanto analizės – *LDA* atvejis, skirtas padidinti tarp klasinius duomenų pasiskirstymus ir sumažinti klasių viduje esančių duomenų nuokrypius. *LDA* laikomas klasikiniu tekstūros atpažinimo klasifikatoriumi, kuris plačiai naudojamas veido identifikavimo uždaviniams spręsti [21]. Algoritmo loginė schema matoma 8 paveiksle.



8 pav. Fišerio veidų apmokymo ir testavimo algoritmas

Algoritmas išnaudoja klasifikavimo savybę, kad normalizuotų nuotraukų klasių variacijos priklauso nuo apmokymo nuotraukų masyvo $X=[x_{1,}x_{2,}x_{3,}\ldots,x_n]$ variacijų, o gautas klases galima atskirti linija. Kadangi kiekvieno veidų masyvo nuotraukos yra pažymimos etikete, *Fišerio veidų* metodas kiekvienai sudaromai požymių klasei taiko skirtingus dimensijų kiekį mažinančius linijinius metodus. Bendrąja forma *LDA* sudaromas naudojant tarp klasinio pasiskirstymo matricos (18) ir vidinės klasės duomenų pasiskirstymo (19) formules [33].

$$S_B=\sum_{i=1}^{c} N_i(\mu_i-\mu)(\mu_i-\mu)^T \tag{18}$$

$$S_W=\sum_{i=1}^{c}\sum_{x_i\in X_i}(x_k-\mu_i)(x_k-\mu_i)^T \tag{19}$$

čia: N_i yra apmokymui pateiktų pavyzdžių kiekis, o – C generuojamų klasių skaičius. μ – vidutinis nuotraukų rinkinio požymių vektorius. x_k yra *k*–toji apmokymo masyvo nuotrauka.

Kaip ir *Tikrinio veido* metodas, *Fišerio veidų* algoritmas naudoja suvidurkintą nuotraukų požymių vektorių (1). Klasės vidinis duomenų pasiskirstymas aprašomas kaip pavyzdžių reikšmių sklaida apie rinkinio vidurkį [3, 5, 33].

Atskiriant duomenis toliau yra ieškomas W_{opt}, kuris maksimizuotų tarp klasinį ir klasės viduje esančių duomenų pasiskirstymą [24]. Jei S_W yra ne vienetinis (angl. *Non–singular*) W_{opt} apskaičiuojamas pagal (20) formulę.

$$W_{opt} = argmax \frac{|W^T S_B W|}{|W^T S_W W|} = [W_1, W_2, W_{3,\ldots}, W_n] \tag{20}$$

čia: $W_n \; |n = 1,2,3\ldots,n$ – tikriniai S_B ir S_W vektoriai. Jei S_W yra vienetinis (angl. S*ingular*), jo dimensijų mažinimui pritaikomas *PCA* metodas pagal (21) formulę, bei *FLD* (22). Tokiu atveju $W_{opt} = W_{FLD} W_{PCA}$.

$$W_{PCA} = argmax |W^T S_T W| \tag{21}$$

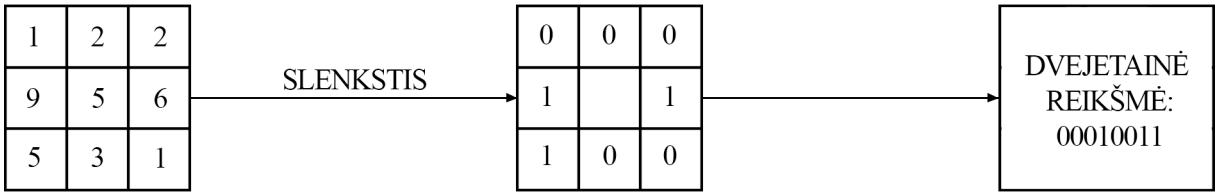
$$W_{FLD} = argmax \frac{|W^T W_{PCA}{}^T S_B W_{PCA} W|}{|W^T W_{PCA}{}^T S_W W_{PCA} W|} = [W_1, W_2, W_{3,\ldots}, W_n] \tag{22}$$

Abiem atvejais veido atpažinimo metu sprendimas priimamas naudojant artimiausio kaimyno metodiką, aprašytą 2.5.1 skyriuje.

Lyginant *Tikrinio veido* ir *Fišerio veido* metodus yra pastebimas didelis sunaudojamų kompiuterio resursų ir atpažinimo tikslumo skirtumas. *Tikrinio veido* metodas yra itin priklausomas nuo apšvietimo svyravimų – gaunama paklaida gali siekti net 48 % nuokrypį. Kita vertus, *Fišerio veidų* metodas pateikia tikslesnius rezultatus ir sunaudoja daugiau kompiuterinės įrangos resursų. Verta paminėti ir tai, kad abu metodai tinkami nagrinėti frontalioms veido nuotraukoms. *LDA* yra daug mažiau priklausomas nuo apšvietimo sąlygų. Jei apmokymui skirtų nuotraukų kiekis yra didelis, *Fišerio veidų* metodas tikslumu pralenkia *Tikrinių veidų* metodą.

2.5.3. Vietinio dvejetainio modelio metodas

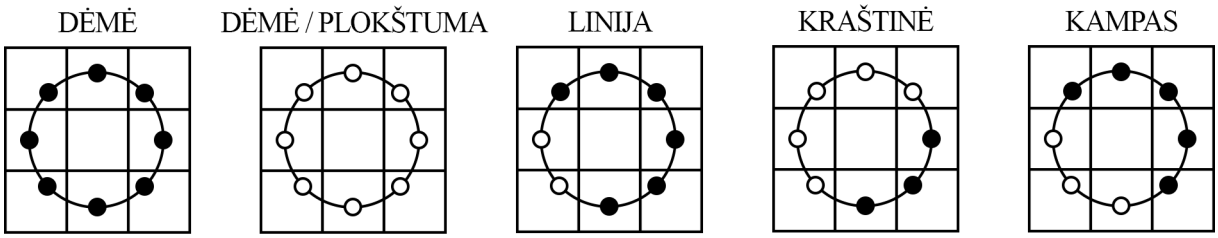
Vietinis dvejetainis modelis dar vadinamas *LBP* yra 1996 metais T. Ojalos sukurtas požymių išskyrimo metodas. Pirmiausiai jis buvo pritaikytas kaip tekstūros klasifikavimo algoritmas, tačiau dėl paprastumo ir pakankamai gerų rezultatų tapo populiarus veido atpažinimo problemai spręsti. Šiuo metu išplėsta *LBP* versija *LBPH* yra laikoma viena iš standartinių analizės metodų *OpenCV* bibliotekoje. Šio algoritmo idėja yra į analizuojamą nuotrauką žiūrėti ne į kaip aukšto lygio vektorinius duomenis, o tiesiog aprašyti vietinius duomenų pokyčius – skirtingai nei holistiniai *PCA* ir *LDA* metodai *LBP* yra lokalus. Originaliame *LBP* metode T. Ojala siūlo naudoti 3x3 dydžio kernelį [48]. Šiame bloke esantys pikseliai yra perskaičiuojami naudojant centrinę reikšmę kaip slenkstį. Jei slenksčio reikšmė yra didesnė arba lygi kaimyniniam pikseliui, jo vietoje įrašomas vienetas, jei mažesnė – nulis. Kaip matoma 9 paveiksle – kiekvienam taškui yra gaunamas binarinis kodas. Priklausomai nuo pilkos spalvos intensyvumo lygių aštuoniuose kaimyniniuose pikseliuose, analizuojamas regionas gali būti išreikštas $2^8 = 256$ skirtingais variantais – dvejetainiu modeliu [4, 5, 6, 8].



9 pav. *LBP* dvejetainio kodo apskaičiavimas

Pagrindinis standartinio *LBP* metodo trūkumas yra tas, kad 3x3 dydžio kernelio nepakanka jei norima išskirti dominuojančius požymius didelį plotą užimančiuose regionuose. Šiai problemai spręsti buvo pasiūlytas išplėstinis *Vietinių dvejetainių modelių* metodas dar žinomas kaip tolygaus rašto algoritmas (angl. *Uniform patterns*). Algoritmo metu galima įvertinti tokio dydžio regionus, kokio reikalauja sprendžiamas uždavinys. Bendruoju atveju regionas aprašomas kaip vienodu atstumu nuo centrinio pikselio nutolusių taškų rinkinys. Šie mėginių ėmimo taškai (angl. *Sampling points*) išdėstyti ratu, vienodais atstumais vieni nuo kitų. Dvejetainis kodas yra generuojamas skaičiuojant slenksčio skirtumus tarp centro ir į tašką patenkančio pikselio. Jei mėginio ėmimo taškas nepatenką į konkretų nuotraukos pikselį, yra pritaikomas bitiesinės (angl. *Bilinear*) interpoliacijos metodas. Tai leidžia panaudoti neriboto dydžio mėginių ėmimo taškų masyvą regione ir apskaičiuoti jo dvejetainę reikšmę. Be to išplėstas *LBP* metodas tampa ne toks priklausomas nuo nuotraukos

transformacijų, kadangi 8 bitų paveikslėlyje yra išlaikomas pikselių intensyvumas visam regionui [4, 8]. *Tolygaus rašto* algoritmas taip pat padeda išskirti konkrečias nuotraukoje esančios struktūros dalis (10 pav.).



10 pav. *LBP* aptinkamų požymių pavyzdys

Veidų atpažinimui dažniausiai naudojamas *LBP* histogramų metodas – *LBPH*. Subjekto nuotrauką galima apibrėžti kaip lokalių tekstūrų rinkinį. Remiantis šia savybe, paveiksliuką galima sudalyti į nustatyto dydžio $m_i = n \cdot n$ regionus $M = [m_1, \ldots, m_i]$, kai n yra nelyginis pikselių skaičius. Tada, remiantis aprašytomis metodikomis, kiekvienam jų yra apskaičiuojamos binarinio kodo reikšmės. Nagrinėjamam paveiksliuko blokui R_i iš gautų dvejetainių reikšmių yra sugeneruojama histograma, kurią galima panaudoti kaip regiono deskriptorių. Šių histogramų suma sudaro visos nuotraukos histogramą (23).

$$H_{i,j} = \sum_{x,y} I\{f_1(x,y) = i\} I\{(x,y) \in R_j\}, i = 0, \ldots, n-1, j = 0, \ldots, m-1 \tag{23}$$

Ši histograma laiko lokalių požymių, tokių kaip linijos, taškai ar plokštumos, pasiskirstymo nuotraukoje informaciją. Be to, algoritmo metu yra sugeneruojamas trijų dimensijų požymių vektorius: pikselių lygmenyje nustatomi dvejetainiai požymiai, regiono lygmenyje iš jų sukuriamos regiono histogramos, globaliame lygmenyje yra sudedamos lokalių histogramų reikšmės ir sukuriamas veido deskriptorius.

Gautą veido nuotraukos histogramą galima panaudoti kaip požymių vektorių, reikalingą statistinei matematinei analizei atlikti [4]. Tokiu atveju yra lyginimas panašumas tarp testavimui duoto paveiksliuko I_T ir apmokymo rinkinyje esančių nuotraukų I_{DB} pagal vieną iš praktikoje siūlomų metodų: histogramų susikirtimo (angl. *Histogram intersection*)

(24), Log–tikimybės statistikos (angl. *Log–likelihood statistic*) (25), Chi kvadratinės statistikos (angl. *Chi square statistic* $\left(x^2\right)$) (26):

$$D\left(I_T, I_{DB}\right)=\sum_{j=1}^{k^2}\left(\sum_{i=1}^{p(p-1)+3} min\left(I_{Tij}, I_{DBij}\right)\right) \tag{24}$$

$$L\left(I_T, I_{DB}\right)=\sum_{j=1}^{k^2}\left(-\sum_{i=1}^{p(p-1)+3} I_{Tij}\log I_{DBij}\right) \tag{25}$$

$$x^2\left(I_T, I_{DB}\right)=\sum_{j=1}^{k^2}\left(\sum_{i=1}^{p(p-1)+3} \frac{\left(I_{Tij}-I_{DBij}\right)^2}{I_{Tij}+I_{DBij}}\right) \tag{26}$$

Vietinių dvejetainių modelių metodas yra vienas stipriausių požymių išskyrimo algoritmų. Paprasta matematinė forma ir tolerancija monotoniškam pilkų lygių pikselių pokyčiui lėmė spartų jo išplitimą ir tolimesnį tobulinimą. Tipinis *LBP* analizės objektas – žmogaus veidas, o remiantis atliktais tyrimais šis algoritmas pranoksta standartinį *Tikrinių veidų* metodą.

Deja, kaip ir kiti tekstūros analizės metodai, *Vietinių dvejetainių modelių* algoritmas yra itin jautrus triukšmui, išblukinimui, objekto deformacijoms ir ekstremaliems apšvietimo nuokrypiams. Žmogaus veidas, kaip trijų dimensijų struktūra pasižymi visomis gylio apšvietimo savybėmis. Priklausomai nuo šviesos šaltinio pozicijos erdvėje, susidaro lokalūs ir globalūs šešėliai, taip pat galimi atspindžiai. Siekdami pagerinti asmens identifikavimo koeficientą, prieš apmokydami ir testuodami *OpenCV LBPH* metodą privalome normalizuoti šiuos triukšmus visose nuotraukose.

2.6. Skyriaus apibendrinimas

Veido atpažinimo seka bendruoju atveju yra vienoda: testuojamoje nuotraukoje pirmiausiai yra surandamas veidas, jis iškerpamas, normalizuojamas ir atpažįstamas. Veido aptikimui *OpenCV* biblioteka naudoja standartinius *Viola–Johnes* ir *Haar* požymių aptikimo algoritmus. Naudojant regiono iškirpimo metodus yra sukuriama standartinio veido stačiakampio nuotrauka, kurią galima normalizuoti pritaikant siūlomą algoritmą.

Veidų apšvietimo problemai spręsti yra siūlomas naujas tiesioginis apšvietimo normalizavimo modelis, pagrįstas smulkių ir didelių požymių turinčių regionų atskiromis

modifikacijomis. Apšvietimo korekcijos yra atliekamos naudojant dviejų algoritmų junginį iš 10–ties standartinių normalizavimo metodų. Pirmasis algoritmas naudojamas akių, nosies ir lūpų regiono normalizavimui, antrasis – kaktos, žandų bei likusių regionų korekcijoms. Siūlomo apšvietimo normalizavimo būdo rezultatų testavimas atliekamas naudojant standartinius veido atpažinimo algoritmus: *Tikrinių veidų*, *Fišerio veidų* bei *Vietinio dvejetainio modelio* metodus.

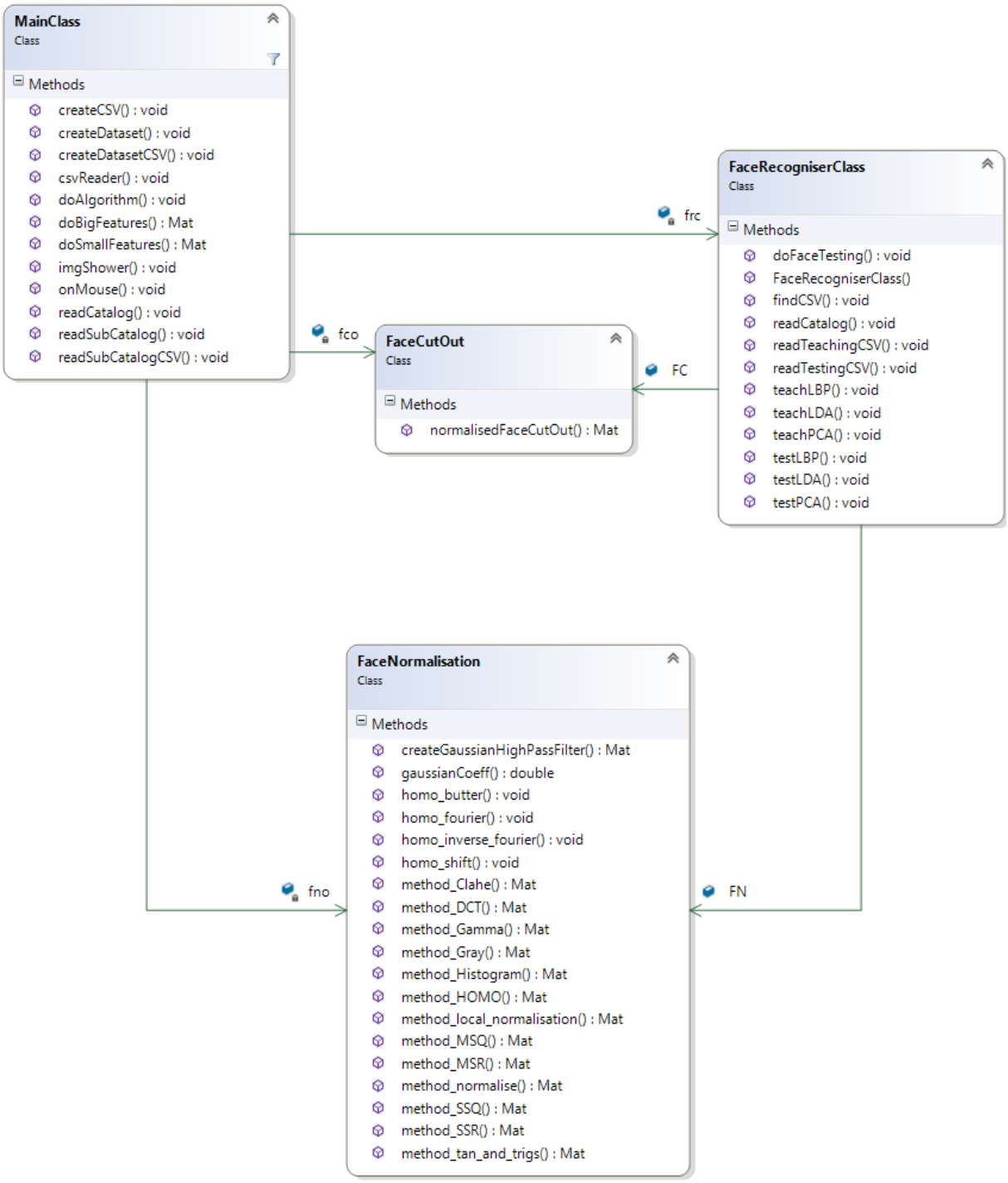
3. KURIAMOS PROGRAMOS INTEGRACIJA *OPENCV*

Siūloma apšvietimo normalizavimo metodika buvo įgyvendinta ir ištestuota naudojant *Microsoft Visual Studio 2015* programinę įrangą ir atviro kodo *OpenCV* biblioteką. Programa parašyta objektine *C++* kalba, o atskiras jos klases galima lengvai modifikuoti ar perkelti į naujai kuriamus projekto failus. Šiame skyriuje aprašyta kuriamos programos architektūra, kuriamo šviesos normalizavimo modelio realizavimo metodika bei programos veikimo principai.

3.1. Programos architektūra

Darbo metu sukurtą apšvietimo normalizavimo ir veido atpažinimo programos kodą sudaro 4 tarpusavyje susijusios klasės: *Main.cpp*, *FaceCutOut.cpp*, *FaceNormalisation.cpp*, *FaceRecogniser.cpp*. Pagrindinė klasė *Main ( )* sukuria primityvią *MS–DOS* valdymo sąsają skirtą vartotojo patogumui. *FaceCutOut ( )* esantys metodai padeda iš apmokymo nuotraukos išskirti ir suspausti veido regioną. *FaceNormalisation ( )* integruoja 10–ties apšvietimo normalizavimo algoritmų ir jų papildinių metodus, o *FaceRecogniser ( )* dalyvauja *PCA*, *LDA* ir *LBP* algoritmų apmokymo bei testavimo procese.

Darbo metu realizuotas programinis kodas veikia iteraciniu principu. Vartotojui tereikia pagrindiniame meniu nurodyti norimus atlikti veiksmus, o duomenų struktūrų paruošimai, normalizacijos procesas, veido atpažinimo algoritmų apmokymas ir rezultatų apskaičiavimai yra automatizuoti. 11 paveiksle pateikta sukurtos programos *UML* diagrama.



11 pav. Sukurtos programos *UML* diagrama

3.2. Aplikacijos veikimo ypatumai

Duomenų struktūros paruošimas – atidarius programą vartotojas pirmiausia privalo nurodyti kelią iki šakninio veidų duomenų bazės katalogo, esančio jo kompiuterio sistemoje. Programa automatiškai atrenka visus šiame kataloge esančius duomenų rinkinius ir jų viduje laikomas nuotraukas. Šiuo atveju laikoma, kad viename kataloge yra vieno asmens veidų rinkinys, o jo identifikavimas vykdomas pagal katalogo pavadinimą – numerį.

Pirmiausiai programa nustato kokio dydžio yra duomenų bazė, ir sukuria atitinkamą kiekį katalogų bei *CSV* failų, reikalingų rezultatų išsaugojimui. Sekantis etapas priklauso nuo to, ar buvo pasirinktas proceso automatizavimas ar atliktas rankinis veido regionų pažymėjimas. Pirmuoju atveju programa vykdo automatizuota veido iškirpimą ir sulygiavimą naudodama *Haar* požymių filtrus ir *Viola–Johnes* algoritmą, aprašytus 2.1 šio darbo skyriuje. *Tačiau* jei šis metodas nesuveikia, ar prastai aptinka veido regionus, vartotojas gali rankiniu būdu pažymėti akis ir nosį. Tokiu atveju koordinatės kiekvienam asmeniui yra išsaugomos regionų duomenų *CSV* faile, kuris naudojamas veido srities iškirpimui. Darbo metu buvo pastebėta, kad automatinis procesas veikia itin prastai ir yra labai priklausomas nuo nuotraukų apšvietimo. Net keičiant pradinius algoritmo parametrus, esant dideliems apšvietimo svyravimams nuotraukose arba nebuvo aptinkamos akių ir nosies zonos, arba jų buvo randama daugiau nei iš tiesų yra.

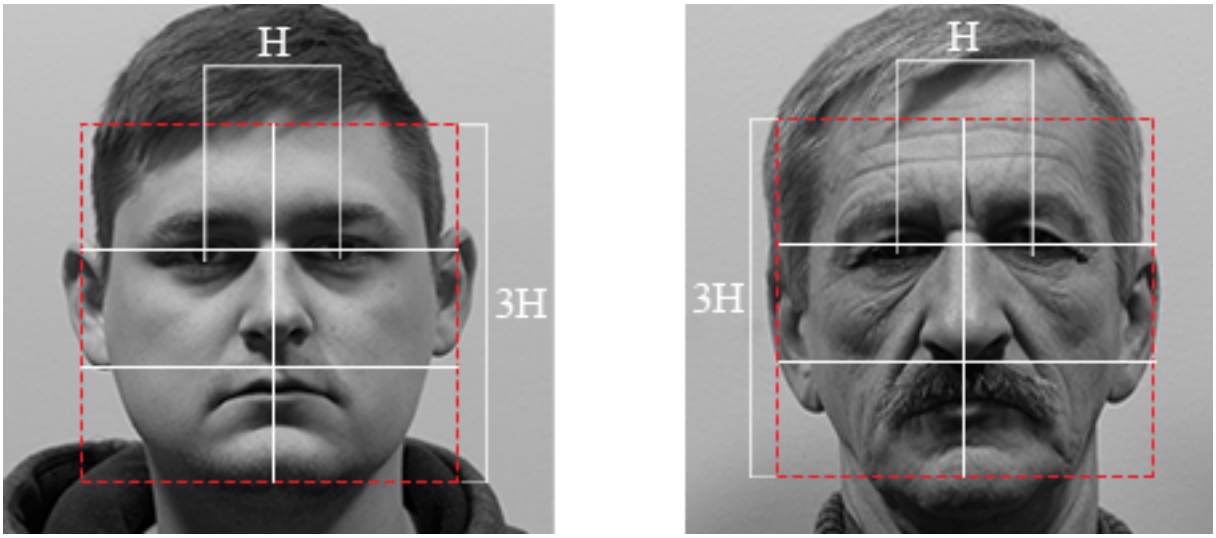
Verta paminėti, kad visos programos dalys gali veikti atskirai, priklausomai nuo naudotojo poreikių. Turint iš anksto paruoštą *CSV* failą ir veidų nuotraukas, galima duomenų bazės analizė, praleidžiant paruošimo etapą. Taip pat, galimas ir nuotraukų normalizavimas, neatliekant analizės, jei siekiama duomenų rinkinį perkelti į kitą mašiną. Sugeneruoto *CSV* failo pavyzdys matomas 12 paveiksle.

```
74  D:\The Library\Work\C++
    projects\OpenCv\Magistrinis\Magistrinis\Dataset\subject_21\subject_5_9\11.JPG,
75  D:\The Library\Work\C++
    projects\OpenCv\Magistrinis\Magistrinis\Dataset\subject_21\subject_5_9\12.JPG,
76  D:\The Library\Work\C++
    projects\OpenCv\Magistrinis\Magistrinis\Dataset\subject_21\subject_5_9\13.JPG,
77  D:\The Library\Work\C++
    projects\OpenCv\Magistrinis\Magistrinis\Dataset\subject_21\subject_5_9\14.JPG,
78  D:\The Library\Work\C++
    projects\OpenCv\Magistrinis\Magistrinis\Dataset\subject_21\subject_5_9\15.JPG
```

12 pav. *CSV* failo pavyzdys

Veido sulygiavimas ir iškirpimas – Atlikus duomenų paruošimą, pirmasis normalizacijos žingsnis yra sulyginti visos duomenų bazės veidų plotus, surasti centrus ir suvienodinti akių aukštį. Šiuos veiksmus atlieka klasė *FaceCutOut.cpp ( )*. Standartinio stačiakampio sudarymas pasiekiamas naudojant 3 koordinates: akių centrų ir nosies pozicijų. Naudojant atstumus tarp šių taškų X ir Y koordinačių, surandamas veido centras ir nustatoma, ar subjekto galva nėra pasukta.

Verta paminėti, kad biologiniu požiūriu kiekvieno žmogaus veido forma nėra taisyklinga. Akių pozicijos natūraliai gali būti aukščiau arba žemiau viena kitos, todėl net jei subjekto galva nėra pasukta, Y koordinatė tokiose nuotraukose skirsis. Ta pati taisyklė galioja ir nosies pozicijai – jos X, Y koordinatė gali būti pasislinkusi į šoną nuo realaus veido centro. Parašytas algoritmas šiuo atveju įvertina visus 3 parametrus ir bando apskaičiuoti standartinio stačiakampio plotą (13 pav.). Tačiau tik tuo atveju, jei skirtumas tarp abiejų akių Y koordinačių įeina į leidžiamo slenksčio ribas – nenustatomas veido posūkis, pagal Z ašį.



13 pav. Standartinio veido stačiakampio apskaičiavimas

Atlikus pirminę analizę, algoritmo veikimo metu sekančiame etape nustatomas veido horizontalus centras – vidurys tarp abiejų akių, atsižvelgiant į nosies poziciją. Ši koordinatė naudojama kaip atskaitos taškas visiems nuotraukų iškirpimams. Atstumas nuo nuotraukos viršaus iki akių linijos yra konstanta H, o stačiakampio plotis ir aukštis apskaičiuojami atsižvelgiant į tarpą tarp abiejų akių ir yra lygūs $3H$. Gaunamas rezultatas – suvienodinti veido nuotraukų rinkiniai, kuriuose akių aukštis yra toje pačioje pozicijoje. Net ir naudojant

standartinę normalizaciją tarp skirtingų asmenų nuotraukų yra matoma gana aiškūs žmogaus veido neproporcingumo pavyzdžiai.

Programos metu sukuriami ir į katalogus pagal subjekto numerį yra suskirstomi pilkų lygių 512x512 px dydžio nuotraukų rinkiniai (nuotraukų dydi galima keisti pagal poreikį), kurie yra paruošti apšvietimo normalizacijos algoritmams.

Apšvietimo normalizavimas – sugeneruoti iškirpti veido plotai yra normalizuojami pritaikant siūloma smulkių požymių turinčių regionų atskyrimo metodiką nuo pagrindinio nuotraukos ploto. Šį procesą atlieka *FaceNormalisation.cpp* klasės metodai, kviečiami iš pagrindinės programos funkcijos. Sukurtoje programoje yra realizuoti 10 skirtingų algoritmų, aprašytų 2 skyriuje. Nagrinėjant duomenų rinkinį principu *visi su visais* vienai nuotraukai iš viso yra atliekama 100 normalizacijos ciklų.

Kiekvienas iš naudojamų apšvietimo normalizavimo metodų, programos realizacijoje reikalauja naudotojo siunčiamų duomenų: iš duomenų bazės nuskaitytos nuotraukos bei algoritmui reikalingų parametrų įvedimo. Šiuo atveju, modifikuojama nuotrauka turi būti NxN dydžio veido kvadratas. Pageidautina, kad N parametras neviršytų 500 px, kadangi nuotraukų rinkinio testavimo etape galimas kompiuterio atminties perkrovimas ir *OpenCV* bibliotekos lūžimas.

Siūloma apšvietimo normalizacija vienu metu yra atliekama dviems identiškoms nuotraukoms. Programos veikimo metu naudojamas dvigubas *C++* programavimo kalbos *for* ciklas ir kviečiami iteracijos numeracijas i bei j atitinkantys algoritmai, kai $0<i,j<10$. Bazinis algoritmas i skirtas didelių veido plotų – žandų, kaktos ir smakro srities apšvietimo normalizacijai atlikti. Jį papildantis j numerį atitinkantis metodas bando pašalinti šviesos triukšmus ties smulkesniais objektais – akimis, nosimi ir lūpomis. Kai algoritmai persidengia vienas su kitu – ciklo metu kviečiamas $i=j$ algoritmas, gaunama paprastoji apšvietimo normalizacija, kurią galima naudoti kaip atskaitos tašką rezultatų palyginimo procese. Galutiniame siūlomo apšvietimo normalizavimo metodo etape vykdomas šių nuotraukų plotų sujungimas.

Atlikus vieną iteraciją, kviečiantysis programos metodas išsaugo normalizuotą nuotrauką atitinkamame kataloge *subject_i_j*, kurio pavadinimas sugeneruojamas automatiškai. Taip pat į apmokymo bei testavimo *CSV* tipo failus yra įrašomas šio katalogo kelias ir subjekto identifikacijos numeris. Pagal numatymą, 22 nuotraukos yra naudojamos

atpažinimo programos apmokymui, o 11 testavimui, tačiau sukurto programinio kodo naudotojas gali bet kada pakeisti šiuos parametrus ir pritaikyti programą ar jos dalis savo produkto realizacijai.

Algoritmų apjungimas. Programai atlikus apšvietimo normalizacijas, atliekamas paskutinis siūlomo algoritmo etapas – normalizuotų nuotraukos regionų apjungimas. Už šią dalį yra atsakingas *Main.cpp* klasės metodas *doAlgorithm ( )*. Kaip buvo minėta, siūlomas normalizavimo metodas atskiria didelius veido plotų regionus, kuriuose gali būti nedaug požymių bei mažus veido regionus, kuriuose gali būti didelis atpažinimui tinkamų požymių kiekis. Sukurtoje programoje realizuotas tiesioginis veido zonų iškirpimas ir apjungimas į bendrą nuotrauką, nenaudojant jokių kitų normalizavimo metodų.

Pirmiausiai atitinkamiems nuotraukos regionams yra atliekamos 2 skirtingos apšvietimo normalizacijos. Toliau algoritmas naudoja konkrečios nuotraukos regionų X ir Y koordinačių rinkinius, tam kad būtų galima apie juos apibrėžti nustatyto dydžio kvadratus ir juos iškirpti. Šio proceso metu gauti nuotraukos regionai yra perkeliami ir sujungiami su kita normalizuota nuotrauka. Kadangi koordinačių sistemos tarp šių paveiksliukų sutampa, gaunamas neiškraipytas vaizdas. Pagrindinė šio metodo problema yra ta, kad jei naudojami algoritmų junginiai vizualiai skiriasi savo rezultatais, aplink sujungtas zonas matomi ryškūs kontrastų skirtumai. Tai gali turėti pakankamai didelę įtaką veido atpažinimo testų rezultatams.

Siūlomo normalizavimo metodo testavimas. Atlikus duomenų bazės nuotraukų normalizaciją bei turint paruoštus apmokymo – testavimo *CSV* failus yra galima veido atpažinimo analizė. Šiuos veiksmus atlieka klasė *FaceRecogniser.cpp*. Pirmame testų etape yra sukuriami tušti matricos ir sveikojo skaičiaus tipo vektoriai. Juose yra laikomos veidų rinkinio nuotraukos ir atitinkami subjektų identifikatoriai, dar vadinami etiketėmis. Programa nurodytame šakniniame kataloge suranda *csv_teach.txt* ir *csv_test.txt CSV* tipo failus, kuriuose yra nuorodos į apmokymui ir testavimui skirtų nuotraukų katalogus. Taip pat, jei neegzistuoja, yra sukuriami r*esults_pca_x_y.txt*, *results_lda_x_y.txt* bei *results_lbp_x_y.txt* failai, kuriuose įrašomos skaitinės rezultatų reikšmės. Šiuose failuose yra aprašoma kiek kartų subjektas buvo teisingai atpažintas, koks buvo apšvietimas ir koks algoritmų junginys buvo panaudotas apšvietimui normalizuoti. Šiuos duomenis galima perkelti į *Microsoft Excel* tipo programinę aplinką ir grafiškai atvaizduoti – supaprastinti analizės procesą.

Testavimo metu yra naudojami 2–ame šio rašto darbo skyriuje aprašyti *OpenCV* bibliotekos *PCA*, *LDA* ir *LBP* metodai. Iteraciniu būdu programa nuskaito visus *CSV* apmokymo failus ir nurodyta eilės tvarka sudaro veidų požymių rinkinius. Tada vykdomas testavimo procesas. Algoritmams pateikiamas nuotraukų, kurios nebuvo naudojamos apmokymui, masyvas ir tikrinamas atpažinimo tikslumo koeficientas. Atlikus vieno algoritmo junginio testavimo ciklą, visa informacija yra išsaugoma išoriniuose *CSV* rezultatų failuose, o matricų ir sveikųjų skaičių tipo vektorių atmintis yra išlaisvinama. Apmokymo–testavimo procesas pradedamas kitam normalizuotam veidų rinkiniui.

Verta paminėti, kad visos pateikiamos nuotraukos privalo būti vieno kanalo pilkų lygių, o apmokymui reikalingi bent du vieno asmens paveikslai. Siūlomo apšvietimo normalizacijos algoritmo testavimo metodika ir rezultatai aprašomi ketvirtame šio darbo skyriuje.

3.3. Skyriaus apibendrinimas

Siūlomas apšvietimo normalizavimo modelis buvo realizuotas naudojant *OpenCV* biblioteką bei pritaikytas *Windows* operacinei sistemai. Programą sudaro 4 pagrindinės dalys atsakingos už algoritmų iškvietimą, testavimo nuotraukų analizę ir veido aptikimą, siūlomo apšvietimo modelio implementaciją bei testų atlikimą. Sukurta sistema veiksmus atlieka iteraciniu principu ir gali būti naudojama kaip visuma arba atskirų klasių rinkinys.

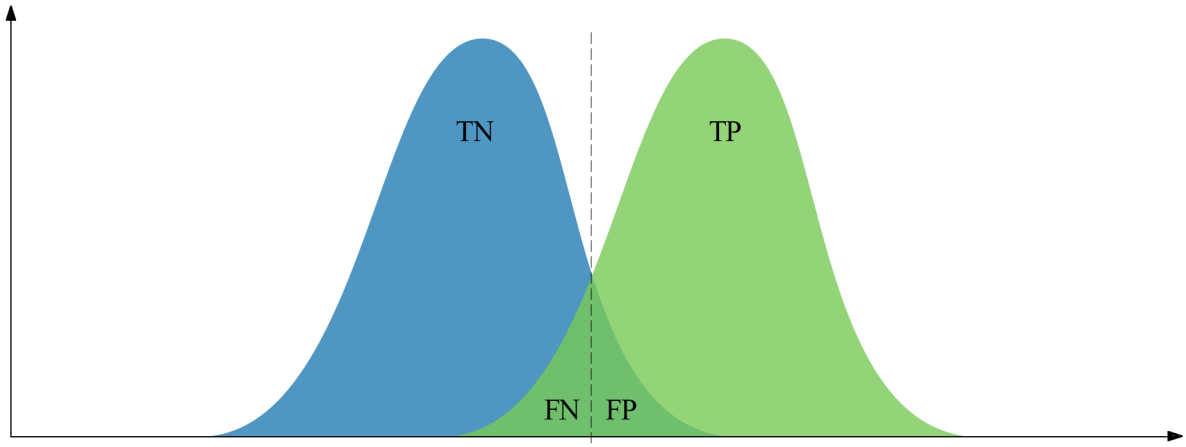
Pagrindiniai programos metu naudojami veidų duomenų bazės katalogai saugomi *DATASET* aplanke. Jo struktūrą sudaro numeruoti subjektų nuotraukų katalogų rinkiniai ir nuotraukos regionų koordinates aprašantys *CSV* dokumentai. Siūlomo metodo testavimas atliekmas naudojant standartinius veido atpažinimo algoritmus, integruotus *OpenCV* bibliotekoje. Programos metu sugeneruoti skaičiavimų rezultatai yra išsaugomi *RESULTS* kataloge sukuriamuose *CSV* failuose.

4. SIŪLOMO APŠVIETIMO NORMALIZAVIMO ALGORITMO EKSPERIMENTINIS ĮVERTINIMAS

Šiame skyriuje aprašytos pagrindinės biometrinių sistemų vertinimo metodikos. Detaliai aprašyta naudojamos veidų duomenų bazės struktūra, sudarymo metodika ir naudojama įranga. Pateikiami siūlomos apšvietimo normalizavimo metodikos testai ir aptariami jų rezultatai.

4.1. Biometrinių sistemų vertinimo metodikos

Biometrinės sistemos yra vertinamos pagal jų gebėjimą teisingai atpažinti objektą, šiuo atveju – veidą. Naudojant subjekto apmokymo nuotraukų masyvą, kiekvienam asmeniui sukuriamas jo požymių modelis. Lyginimo metu tarp pateiktų paveiksliukų ir išsaugotų subjektų modelių sukuriama aibė panašumo įverčių, kuriuos galima grafiškai atvaizduoti tikėtinumų pasiskirstymo kreivėse (14 pav.).



14 pav. Tikėtinumų pasiskirstymo kreivės

Idealiu atveju pasiskirstymo kreivės turėtų nepersikirsti, tai užtikrintų visišką atpažinimo klaidų nebuvimą. Praktikoje, biometrinių duomenų euklidiniai atstumai tarp žinomų ir nežinomų veidų požymių klasių yra persidengiantys. Todėl atsiranda tokių sprendimo priėmimo algoritmo klaidų, kai svetimas veidas priimamas kaip žinomas, o žinomas veidas atmetamas kaip svetimas. Vertinant biometrines sistemas privaloma nustatyti kokio pobūdžio klaidos daro įtaką esminiam sistemos funkcionalumui. Pavyzdžiui, apsaugos sistemoms svetimo asmens identifikavimas kaip žinomo gali būti kritinis.

Sprendimų priėmimo programose gražinamas tik vienas rezultatas – priimta arba atmesta. Šį atsakymą dar gali papildyti sprendimo užtikrintumo koeficientas, arba euklidinio atstumo skaitinė išraiška. Priimto sprendimo būsena gali būti apibūdinta reikšme iš 2 lentelės.

2 lentelė. Sprendimų matrica

	Žinoma nuotrauka	Nežinoma nuotrauka
Sistemos priimta	*TP*	*FP*
Sistemos atmesta	*FN*	*TN*

Tuo atveju, kai sistema priima žinomą nuotrauką, gaunama teisingo priėmimo būsena (angl. *True positive – TP*). Atmetant nežinomą nuotrauką – teisingo atmetimo būsena (angl. *True negative – TN*). Sistemos priimtos nežinomos nuotraukos ir sistemos atmestos žinomos nuotraukos atitinkamai laikomos klaidingo priėmimo (angl. *False positive – FP*) ir klaidingo atmetimo (angl. *False negative – FN*) būsenomis [23].

Priklausomai nuo sistemos paskirties, didinti duomenų atskyrimą ir mažinti klaidų tikimybes galima naudojant optimalų slenksčio parametrą. Tokiu atveju yra mažinamas sistemos jautrumas ir didinamas tikslumas – arba atvirkščiai. Klasifikatoriaus jautrumas, dar vadinamas teisingo priėmimo lygiu (angl. *True positive rate – TPR*), yra apibrėžiamas kaip sėkmingai identifikuotų žinomų subjektų TP ir jų kiekio N santykis (27) [23].

$$TPR = \frac{TP}{N} = \frac{TP}{TP + FN} \tag{27}$$

Klasifikatoriaus tikslumas, dar vadinamas teisingo atmetimo lygiu (angl. *True negative rate – TNR*), nurodo sistemos teisingai atmestų nežinomų subjektų TN ir jų kiekio N santykį (28) [23].

$$TNR = \frac{TN}{N} = \frac{TN}{TN + FP} \tag{28}$$

Klaidingo atmetimo lygis (angl. *False negative rate – FNR*) nurodo sistemos neteisingai atmestų žinomų subjektų FN ir testavimui pateiktų nuotraukų kiekio N santykį

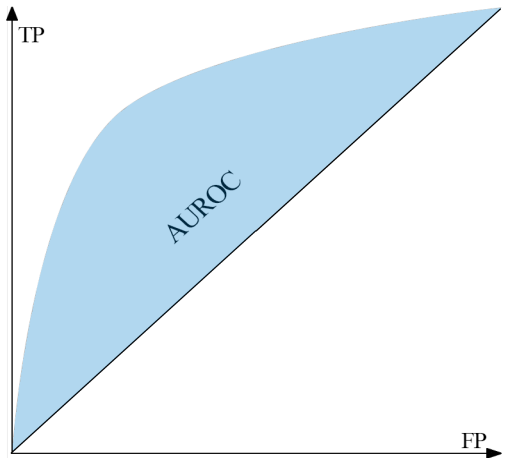
(29). *FNR* reikšmės didėjimas pastebimas, kai slenksčio parametras naudojamas dideliam klasifikatoriaus tikslumui užtikrinti [23].

$$FNR = \frac{FN}{N} = \frac{FN}{TP + FN} \tag{29}$$

Klaidingo priėmimo lygis (angl. *False positive rate – FPR*) nurodo sistemos neteisingai priimtų subjektų FP ir testavimui pateiktų nuotraukų kiekio N santykį (30). Didelė *FPR* reikšmė nurodo, kad slenksčio parametras naudojamas klasifikatoriaus jautrumui padidinti [23].

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN} \tag{30}$$

ROC kreivė (angl. *Receiver operator characteristics*) yra praktikoje dažnai naudojamas įrankis, skirtas vizualiems siūlomų algoritmų įverčiams pateikti. Grafas sudaromas brėžiant *TPR* ir *FPR* reikšmių perėjimus atitinkamai ordinatės ir abscisės ašyse. *ROC* reikšmės svyruoja nuo 0 iki 1, o grafiko įstrižainė naudojama kaip kokybės vertinimo atskaitos linija. Plotas po ROC kreive (angl. *Area under ROC – AUROC*) nusako siūlomų algoritmų tikslumą (15 pav.) [23, 27].



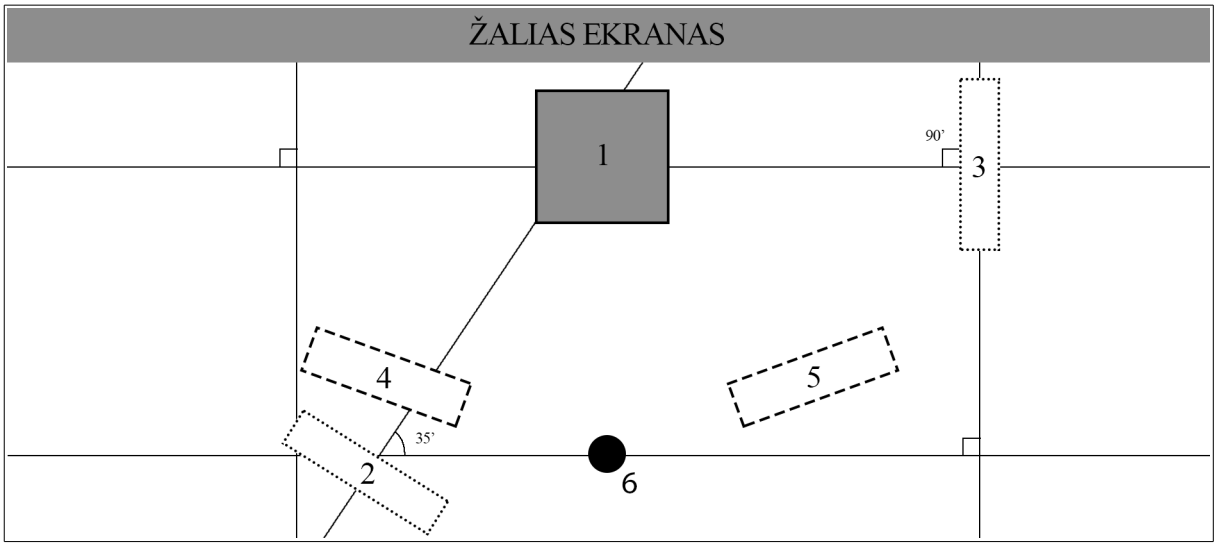
15 pav. Plotas po *ROC* kreive

Idealiu atveju *AUROC* turėtų užimti visą grafo plotą, tačiau praktikoje ši reikšmė svyruoja $0{,}5 \leqslant AUROC \leqslant 1$ intervale. Nubrėžtame grafike, algoritmams įvertinti dažnai naudojami tokie klasifikavimo slenksčiai: 0,9–1 tobulas; 0,8 – 0,9 geras; 0,7 – 0,8 patenkinamas; 0,6 – 0,7 prastas; 0,5 – 0,6 bevertis.

4.2. Duomenų bazės paruošimas

Dauguma literatūroje apžvelgtų apšvietimo normalizavimo algoritmų testų buvo atlikti panaudojant *Jeilio A* arba *Jeilio B* veidų duomenų bazes [20]. Jas sudaro nuo 15 iki 38 individų su 11–64 nuotraukų kiekvienam. Kadangi šie duomenų bazių rinkiniai yra nebeprieinami vartotojui, o panašių, internete randamų nuotraukų rinkinių, imtys yra per mažos, siekiant ištestuoti naują veido apšvietimo normalizavimo metodiką, buvo sukurta nauja veidų duomenų bazė.

Metodikos analizei naudojamas duomenų rinkinys buvo sukurtas valdomoje aplinkoje. Apšvietimui naudota žaliojo ekrano sistema su 5 skirtingo galingumo lempomis: 2 pakabinamomis, 2 pastatomomis ir 1 prožektoriaus tipo. Fotografavimas atliktas naudojant „*Canon EOS 650D*“ fotoaparatą ir „*Canon EF 85mm f/1.8 USM*“ foto objektyvą. 16 paveiksle atvaizduotas scenos išdėstymas.



16 pav. Nuotraukų scenos išdėstymas

Čia 1 numeriu pažymėta fotografuojamo subjekto sėdėjimo vieta. 2 ir 3 numeriai – pastatomi šešių lempų šviestuvai. Numeris 2 pasukta 30 laipsnių kampu, o numeris 3–90 laipsnių kampu. 4–5 pakabinami keturių lempų šviestuvai, kurios sceną apšviečia iš viršaus

~30–45 laipsnių kampu. 6 numeriu pažymėtas „*Canon EOS 650D*“ fotoaparatas ir jo trikojis stovas. Sudarant duomenų bazę, buvo nufotografuota 30 asmenų su 11 skirtingo apšvietimo variantų, matomų 3 lentelėje.

3 lentelė. Veido nuotraukų apšvietimo pavyzdžiai

Visiška prieblanda	Kairysis 30 laipsnių apšvietimas 100 %	Kairysis 30 laipsnių apšvietimas 30 %
Dešinysis 90 laipsnių apšvietimas 100 %;	Dešinysis 90 laipsnių apšvietimas 30 %	Kairysis 30 laipsnių ir dešinysis 90 laipsnių apšvietimai 100 %
Viršutinis kairysis ir dešinysis apšvietimai 100 %	Viršutinis kairysis 100 %	Viršutinis dešinysis 100 %

Apatinis prožektoriaus apšvietimas ~ 70 laipsnių kampu (pridengtas)	Apatinis prožektoriaus apšvietimas ~ 90 laipsnių kampu (pridengtas)	

Siekiant praplėsti apmokymo ir testavimo imtį, kiekvienam apšvietimo atvejui buvo padaryta ~55–70 nuotraukų. Iš šio rinkinio buvo atmesti prastos kokybės elementai – nesufokusuotos, daug triukšmo turinčios nuotraukos. Taip pat, pašalinti paveikslai, kuriuose subjektas yra užsimerkęs, nukreipęs akis, šypsosi arba atlieka kitus veiksmus, kurie didina atpažinimo paklaidą. Atlikus pirminį filtravimą iš viso vienam asmeniui priskirtos 33 nuotraukos, po 3 kiekvienam apšvietimo variantui. Sukurtame nuotraukų rinkinyje yra 12 moterų ir 18 vyrų, amžiaus grupėje nuo 18 iki 54 metų. Visos nuotraukos frontalios, nedengiamos pašalinių ar fono objektų, o fotografuojami subjektai nerodo emocijų. Bendras nuotraukų kiekis – 990.

4.3. Eksperimentinio tyrimo rezultatai

Darbo metu sukurta duomenų bazė yra suskaidoma į du masyvus. 15–kos žmonių nuotraukos sudaro mokymo imtį, o likusių – testavimo. Pirmo eksperimento metu yra siekiama sumažinti nagrinėjamų algoritmų aibę. *Tikrinių veidų*, *Fišerio veidų* ir *Vietinių dvejetainių modelių* metodai yra apmokomi naudojant 19–kos asmenų 2 geros kokybės nuotraukas. Testavimui pateikiamos 3 itin prastos kokybės veido nuotraukos, su daug lokalių ir globalių šešėlių bei papildomo triukšmo. Apšvietimo normalizavimas atliekamas taikant siūlomą metodiką, algoritmai testuojami principu *visi su visais.* 4 lentelėje pateikiami suvidurkinti šių junginių tikslumo koeficientai. Testai atliekami su visais trim analizės metodais. Šiuo atveju algoritmų numeracija atitinka 2 skyriuje aprašytą algoritmų išdėstymo tvarką:

1. *Gama* korekcija;
2. *CLAHE* metodas;
3. Histogramos išlyginimas;
4. *Single–scale self quotient* metodas;
5. *Multi–scale self quotient* metodas;
6. *Single–scale retinex* metodas;
7. *Multi–scale retinex* metodas;
8. Homomorfinis filtravimas;
9. Pleksiformo imitacijos normalizacija;
10. Tano ir Trigso metodas.

4 lentelė. Apšvietimo normalizavimo algoritmų junginių rezultatai (%)

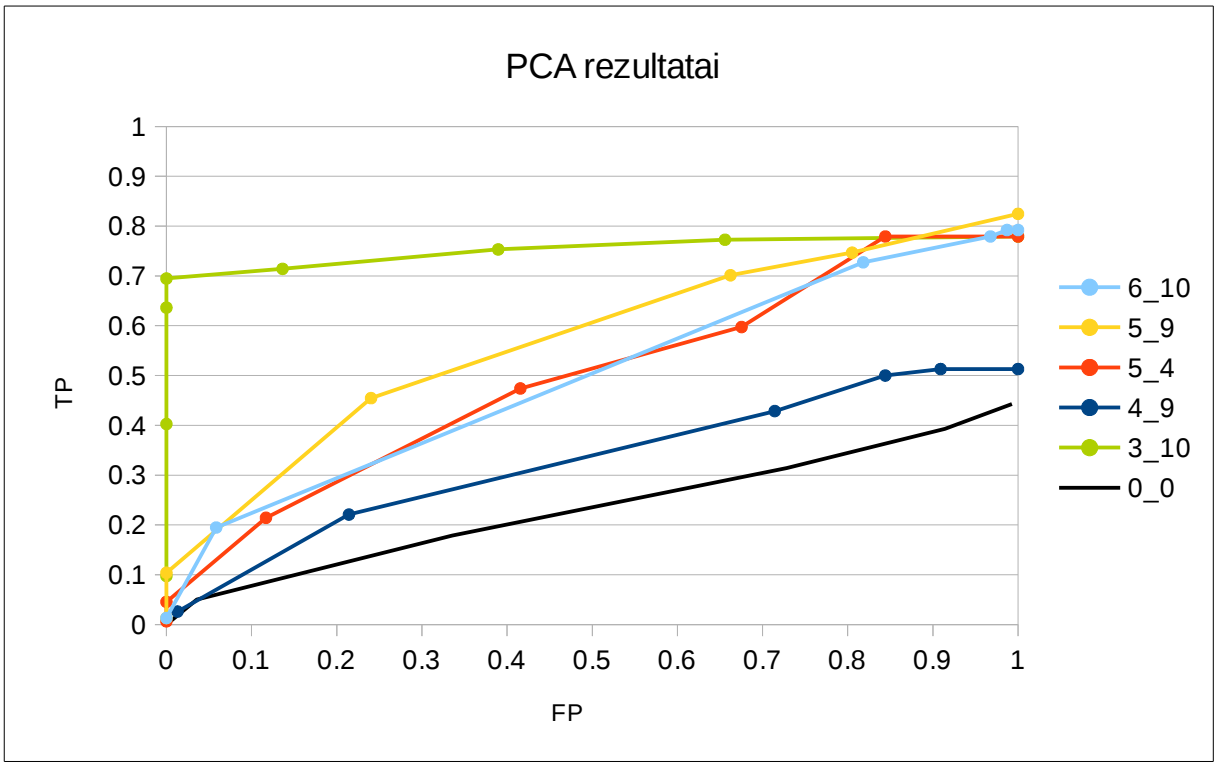
		Mažo mastelio regionų požymių normalizavimo algoritmas									
		1	2	3	4	5	6	7	8	9	10
Didelio mastelio požymių regionų normalizavimo algoritmas	1	44	46	47	44	50	47	43	43	44	43
	2	36	32	36	36	36	36	31	31	29	29
	3	48	42	27	39	35	26	49	51	50	57
	4	53	41	26	35	33	29	37	49	*57*	51
	5	50	43	55	*64*	44	48	54	54	*64*	54
	6	51	43	38	47	39	31	53	53	47	56
	7	25	26	39	44	42	41	19	35	30	16
	8	27	26	32	32	23	23	23	23	26	26
	9	26	23	35	40	33	36	36	29	16	24
	10	33	30	27	36	47	31	43	44	44	30

Kaip matoma lentelėje – smulkius požymius turinčių veido regionų normalizavimo algoritmai yra išdėstyti stulpeliais. Juos papildantys, mažai požymių turinčių didelių regionų apšvietimo normalizavimo algoritmai – eilutėmis. Jų susikirtimo laukelyje nurodytas *PCA*, *LDA*, *LBP* tikslumo vidurkis, išreikštas procentais. Pilkai pažymėtoje įstrižainėje pateikiamas

vienodais metodais normalizuotos nuotraukos atpažinimo koeficientas, o gauta reikšmių matrica yra ne veidrodinė, dėl skirtingų normalizacijos algoritmų užimamo regionų ploto.

Dėl apmokymo aibės dydžio ir testavimo metu naudojamų prastos kokybės nuotraukų, įstrižainėje gaunami žemi tikslumo koeficientai. Prasčiausius rezultatus – 16 % sugeneravo *Pleksiformo imitacijos* metodas, 19 % pateikė *Homomorfinis filtravimas*. Aukščiausią atpažinimo koeficientą – 44 % sugeneravo paprasta *Gama korekcija* bei *Multi–scale self quotient* metodas.

Pirmasis eksperimentinio tyrimo etapo tikslas – išrinkti 10 aukštus rezultatų vidurkius pateikiančius apšvietimo normalizavimo algoritmų junginius. Žalia spalva pažymėti 5 geriausius atpažinimo vidurkius sugeneravę metodai, oranžine – 5 junginiai, kurie turi aukštą pagerinimo santykį, palyginus su pagrindinės įstrižainės reikšme. Aukščiausias tikslumo koeficientas šiame rinkinyje yra 64 %. Žemiausias – 36 %, tačiau ši reikšmė yra 2,25 karto didesnė už pagrindinės įstrižainės reikšmę. Pirmojo rinkinio metodų *PCA* rezultatų *TP* bei *FP* reikšmės pateikiamos 5 lentelėje. *PCA ROC* kreivė atvaizduota 17 paveiksle.



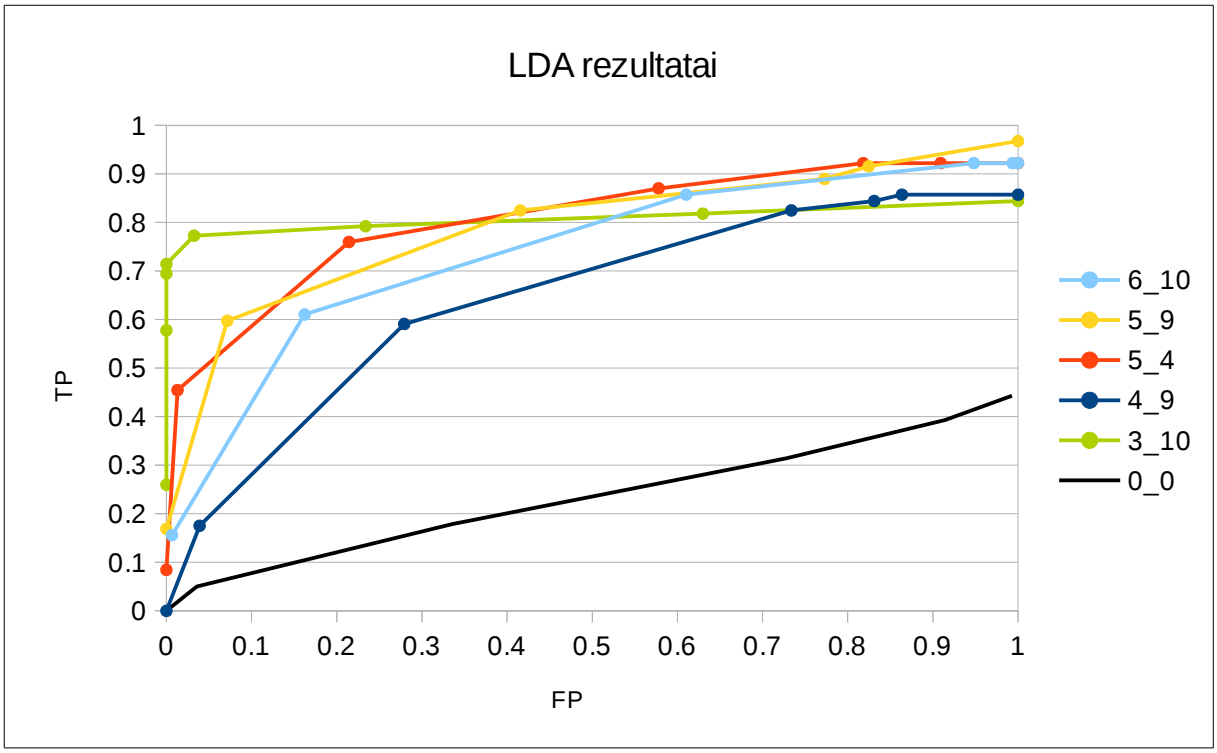
17 pav. Pirmojo atrinkto algoritmų rinkinio *ROC* kreivės naudojant *PCA*

5 lentelė. Pirmojo atrinkto algoritmų rinkinio *TP* ir *FP* santykiai naudojant *PCA*

3 – 10		4 – 9		5 – 4		5 – 9		6 – 10	
TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
0.1	0	0	0	0.01	0	0.01	0	0.01	0
0.4	0	0.03	0.01	0.05	0	0.1	0	0.19	0.06
0.64	0	0.22	0.21	0.21	0.12	0.45	0.24	0.73	0.82
0.69	0	0.43	0.71	0.47	0.42	0.7	0.66	0.78	0.97
0.71	0.14	0.5	0.84	0.6	0.68	0.75	0.81	0.79	0.99
0.77	0.66	0.51	0.91	0.78	0.84	0.82	1	0.79	1
0.78	1	0.51	1	0.78	1	0.82	1	0.79	1

17 paveiksle juoda linija *0_0* žymi nenormalizuotų nuotraukų ROC kreivę, kuri yra daug žemiau nei įstrižainė. Tai parodo, kad testavimo duomenų atskyrimas yra itin prastas – pastebimas žemas jautrumas ir atpažinimo koeficientas. 6–10, 4–9 ir 5–4 apšvietimo normalizavimo algoritmų junginiai pagerino pradinį rezultatą, tačiau visi jie yra arti spėjimo ribos. 5–9 algoritmų junginys pasižymi 82 % tikslumu, esant didžiausiam jautrumo parametrui. Šiuo atveju geriausiai duomenų atskyrimą *PCA* veido atpažinimo metodui atliko *Histogramos išlyginimo* algoritmas, skirtas didelių veido plotų normalizacijai, bei *Tano ir Trigso* metodas, kuriuo normalizuojamas apšvietimas akių, nosies ir lūpų regione. Šio algoritmų junginio jautrumas siekia 78 %, o *FPR* esant patenkinamam požymių atskyrimui tarp klasių - 12 %.

Pirmojo algoritmų apjungimo rinkinio *LDA* rezultatų *TP* ir *FP* reikšmės pateikiamos 6 lentelėje. Kaip matoma 18 paveiksle visos *LDA* metodu testuojamų algoritmų junginių *ROC* kreivės, išskyrus nenormalizuoto rinkinio, yra aukščiau įstrižainės. Duomenų atskyrimą šiuo atveju geriausiai atliko *Histogramos išlyginimo* bei *Tano ir Trigso* metodų junginys. Veido atpažinimo tikslumo koeficientas, esant optimaliam slenksčio parametrui, siekia 77 %, o *FPR* – 3 %. Aukščiausias šiuo metodu pasiekas tikslumas lygus 84 %. Nustačius aukščiausią jautrumo parametrą, 5–9 algoritmų junginys pasiekia 97 % veido atpažinimo tikslumo koeficientą. 4–9 bei 6–10 junginiai siekia 92 % procentų ribą, tačiau *SSR* bei *Tano ir Trigso* metodai geriau atskiria žinomų veidų aibę.

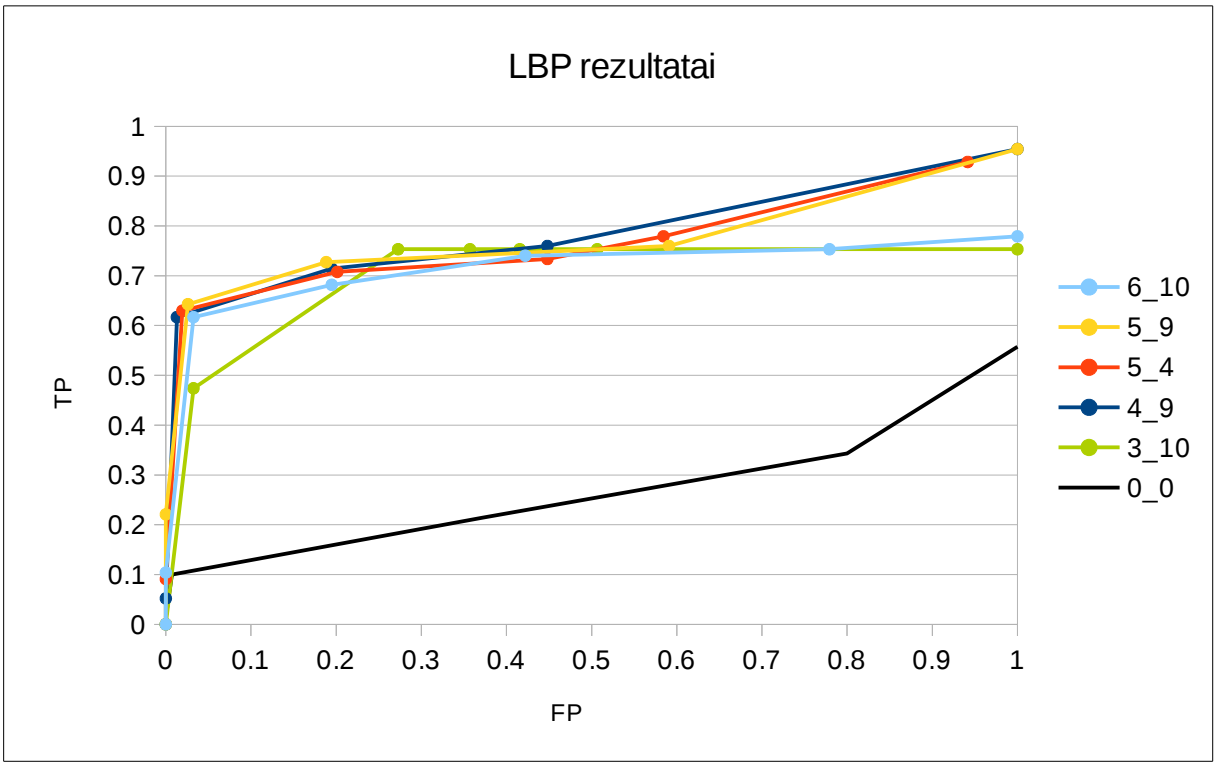


18 pav. Pirmojo atrinkto algoritmų rinkinio *ROC* kreivės naudojant *LDA*

6 lentelė. Pirmojo atrinkto algoritmų rinkinio *TP* ir *FP* santykiai naudojant *LDA*

3 – 10		4 – 9		5 – 4		5 – 9		6 – 10	
TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
0.26	0	0	0	0.08	0	0	0	0.16	0.01
0.58	0	0.18	0.04	0.45	0.01	0.17	0	0.61	0.16
0.71	0	0.59	0.28	0.76	0.21	0.6	0.07	0.86	0.61
0.77	0.03	0.82	0.73	0.87	0.58	0.82	0.42	0.92	0.95
0.79	0.23	0.84	0.83	0.92	0.82	0.89	0.77	0.92	0.99
0.82	0.63	0.86	0.83	0.92	0.91	0.92	0.82	0.92	0.99
0.84	1	0.86	1	0.92	1	0.97	1	0.92	1

Kaip matoma 19 paveiksle – 5 geriausius rezultatus pateikiantys algoritmų junginiai geriausią klasifikavimą atlieka naudojant *LBP* veidų atpažinimo metodą. Šiuo atveju 4–9, 5–4 ir 5–9 algoritmų junginiai pateikė panašius tikslumo koeficientus, siekiančius 94–95 %. Esant geram duomenų atskyrimui, *FPR* gali siekti 1 % ir 63 % tikslumą. Pastebima ir tai, kad 3–10 metodų junginys, priešingai nei *LDA* ir *PCA* analizės atveju, sugeneravo blogiausius atpažinimo koeficientus ir atskyrė veidų požymių rinkinius. Šio grafiko *TP* ir *FP* reikšmės pateiktos 7 lentelėje.

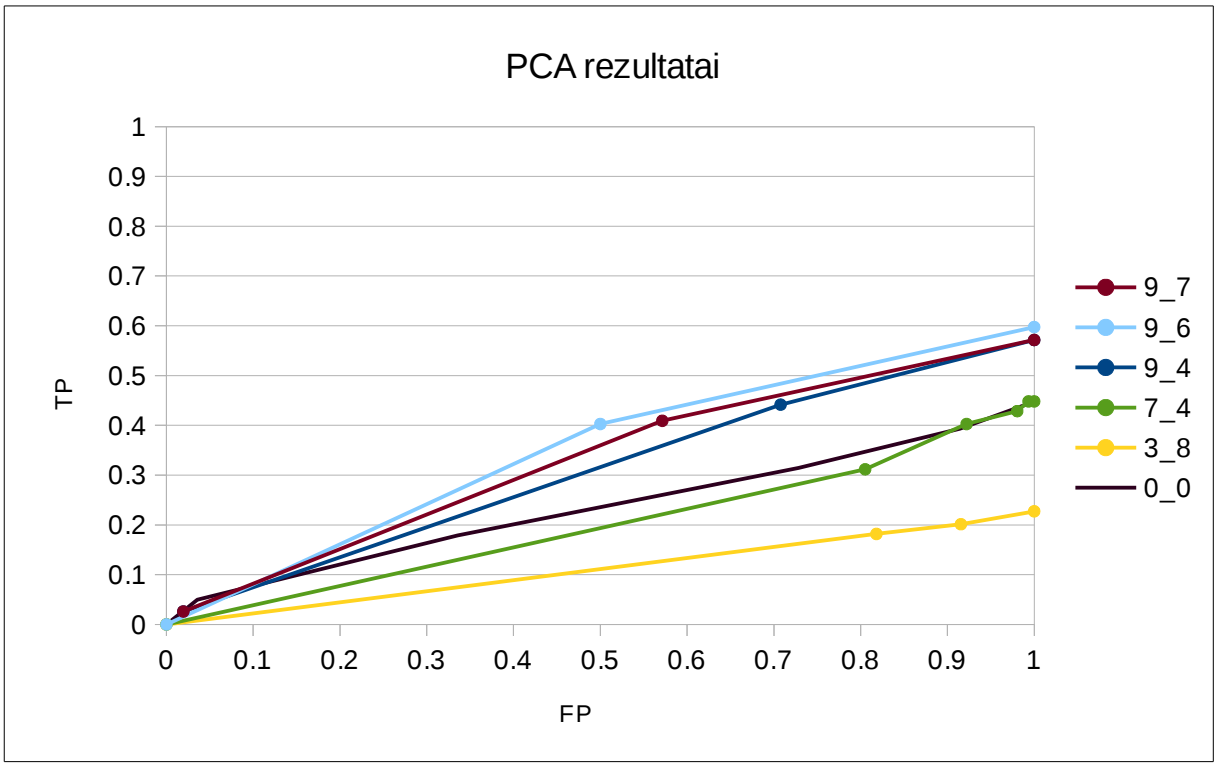


19 pav. Pirmojo atrinkto algoritmų rinkinio *ROC* kreivės naudojant *LBP*

7 lentelė. Pirmojo atrinkto algoritmų rinkinio *TP* ir *FP* santykiai naudojant *LBP*

3 – 10		4 – 9		5 – 4		5 – 9		6 – 10	
TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
0	0	0	0	0	0	0	0	0	0
0.47	0.03	0	0	0.09	0	0.22	0	0.1	0
0.75	0.27	0.05	0	0.63	0.02	0.64	0.03	0.62	0.03
0.75	0.36	0.62	0.01	0.71	0.2	0.73	0.19	0.68	0.19
0.75	0.42	0.71	0.19	0.73	0.45	0.76	0.59	0.74	0.42
0.75	0.51	0.76	0.45	0.78	0.58	0.95	1	0.75	0.78
0.75	1	0.95	1	0.93	0.94	0.95	1	0.78	1

Eksperimentinio algoritmo įvertinimo metu ištestuoti ir 5 algoritmų junginiai, kurie santykinai pagerina įstrižainės algoritmų reikšmes. 8 lentelėje pateikiamos šių metodų *TP* ir *FP* reikšmės, o 20 paveiksle matomos antrojo rinkinio *ROC* kreivės, kai taikomas *PCA* veidų atpažinimo metodas.



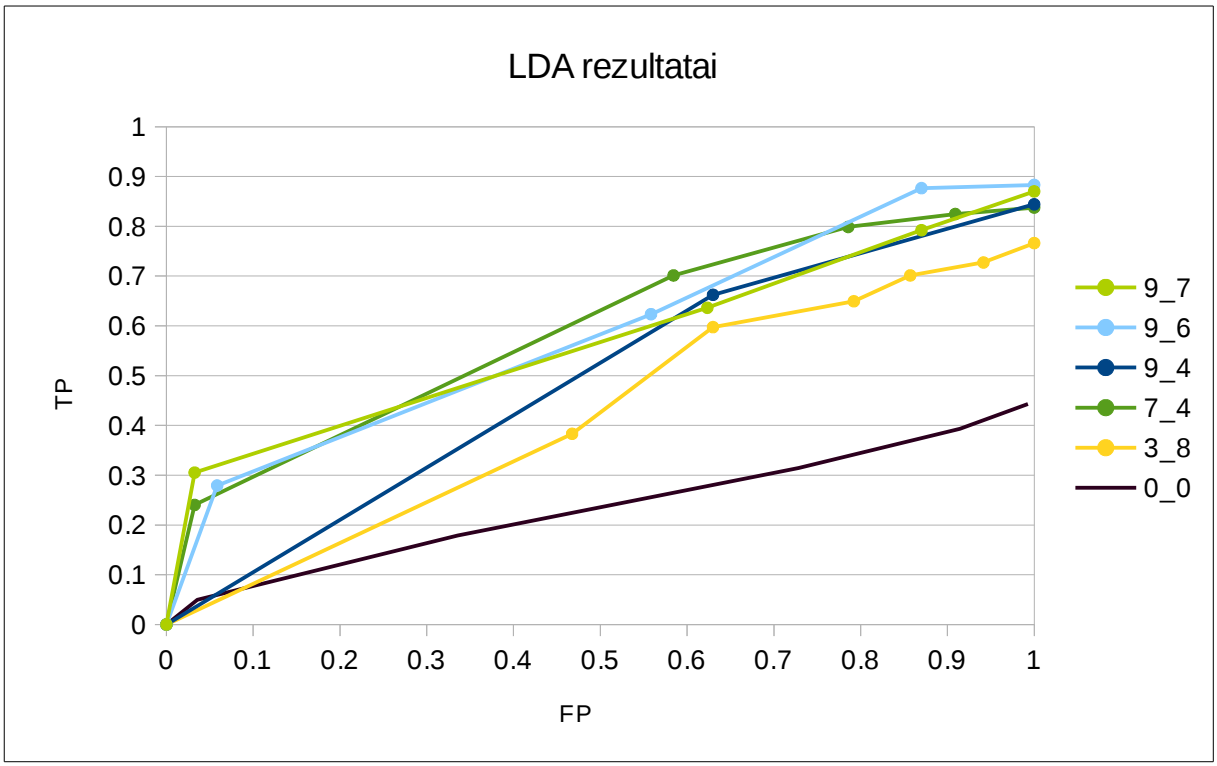
20 pav. Antrojo atrinkto algoritmų rinkinio *ROC* kreivės naudojant *PCA*

8 lentelė. Antrojo atrinkto algoritmų rinkinio *TP* ir *FP* santykiai naudojant *PCA*

3 – 8		7 – 4		9 – 4		9 – 6		9 – 7	
TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
0	0	0	0	0.03	0.02	0	0	0	0
0	0	0.31	0.81	0.44	0.71	0	0	0	0
0	0	0.40	0.92	0.57	1	0	0	0	0
0	0	0.43	0.95	0.57	1	0	0	0	0
0.18	0.82	0.45	0.98	0.57	1	0	0	0.03	0.02
0.2	0.92	0.45	0.99	0.57	1	0.4	0.5	0.41	0.57
0.23	1	0.45	1	0.57	1	0.6	1	0.57	1

Kaip matoma 20 paveiksle – siūlomo apšvietimo normalizavimo metodo *MSR* ir *SSQ* algoritmų junginys bei *Histogramos išlyginimo* ir *Homomorfinio filtravimo* metodų apjungimas generuoja prastesnius *PCA* rezultatus nei nenormalizuotų nuotraukų rinkinys. 9–7, 9–6 bei 9–4 algoritmų junginiai pagerina veido atpažinimo koeficientą, tačiau jis neviršija 60 % ribos.

Taikant siūlomą apšvietimo normalizavimo metodiką ir antro rinkinio algoritmus *LDA* yra pastebimi su *PCA* koreliuojantys rezultatai. 21 paveiksle matomos *ROC* kreivės yra arti įstrižainės, tad duomenų atskyrimas yra prastesnis nei pirmo algoritmų rinkinio. Aukščiausias veido atpažinimo koeficientas siekia 88 %. Prasčiausią rezultatą sugeneruoja 3–8 algoritmų junginys. 9 lentelėje pateikiami *TP* ir *FP* rezultatai *LDA* atpažinimo metodui.



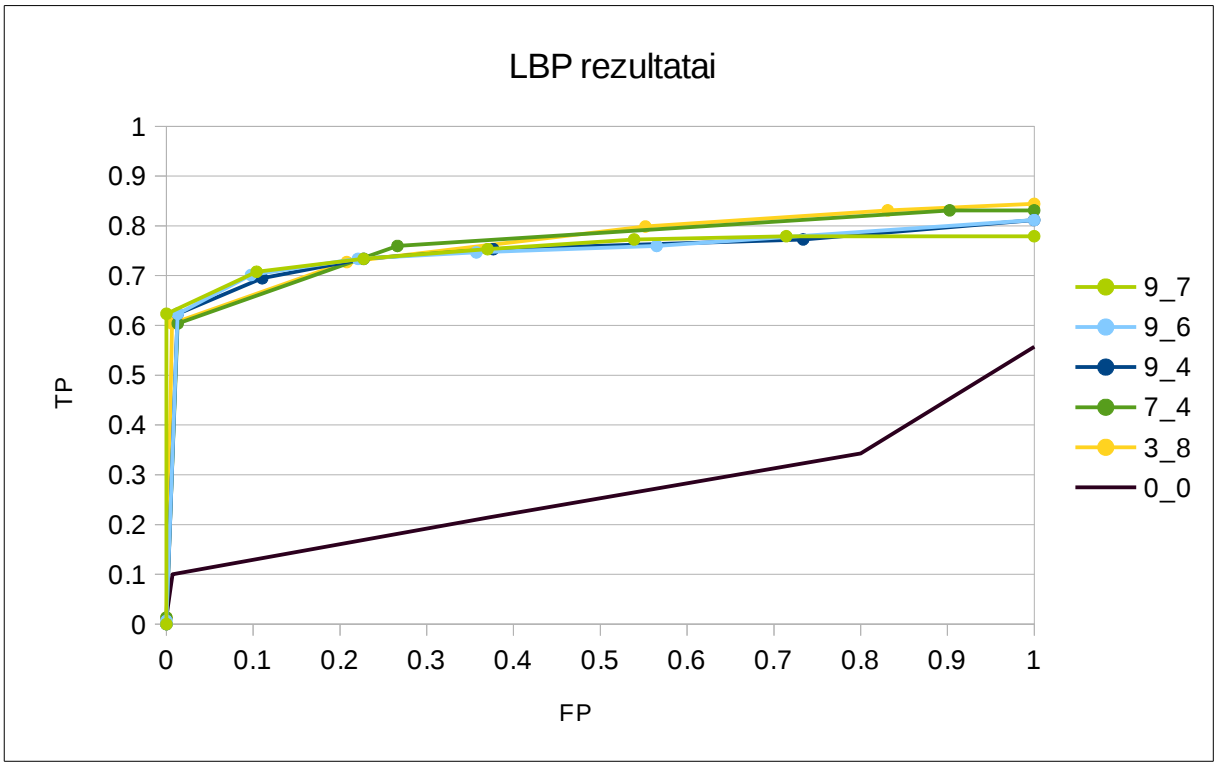
21 pav. Antrojo atrinkto algoritmų rinkinio *ROC* kreivės naudojant *LDA*

9 lentelė. Pirmojo atrinkto algoritmų rinkinio *TP* ir *FP* santykiai naudojant *LDA*

3 – 8		7 – 4		9 – 4		9 – 6		9 – 7	
TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
0	0	0	0	0	0	0	0	0	0
0.38	0.47	0.24	0.03	0.66	0.63	0.28	0.06	0.31	0.03
0.6	0.63	0.7	0.58	0.84	1	0.62	0.56	0.64	0.62
0.65	0.79	0.8	0.79	0.84	1	0.88	0.87	0.79	0.87
0.70	0.86	0.82	0.91	0.84	1	0.88	1	0.87	1
0.73	0.94	0.84	1	0.84	1	0.88	1	0.87	1
0.77	1	0.84	1	0.84	1	0.88	1	0.87	1

LBP analize ištestuotas apšvietimo normalizavimo metodas pateikė pirmam algoritmų junginių rinkiniui panašius rezultatus. Kaip matoma 22 paveiksle, požymių

atskyrimas tarp žinomų ir nežinomų veidų duomenų aibės yra itin tikslus. Tikslumo parametras yra artimas pirmo rinkinio tikslumui, tačiau pastebimas prastesnis jautrumas (10 lentelė). Aukščiausias veido atpažinimo koeficientas siekia 84 %, žemiausias *FPR* – 2 %.



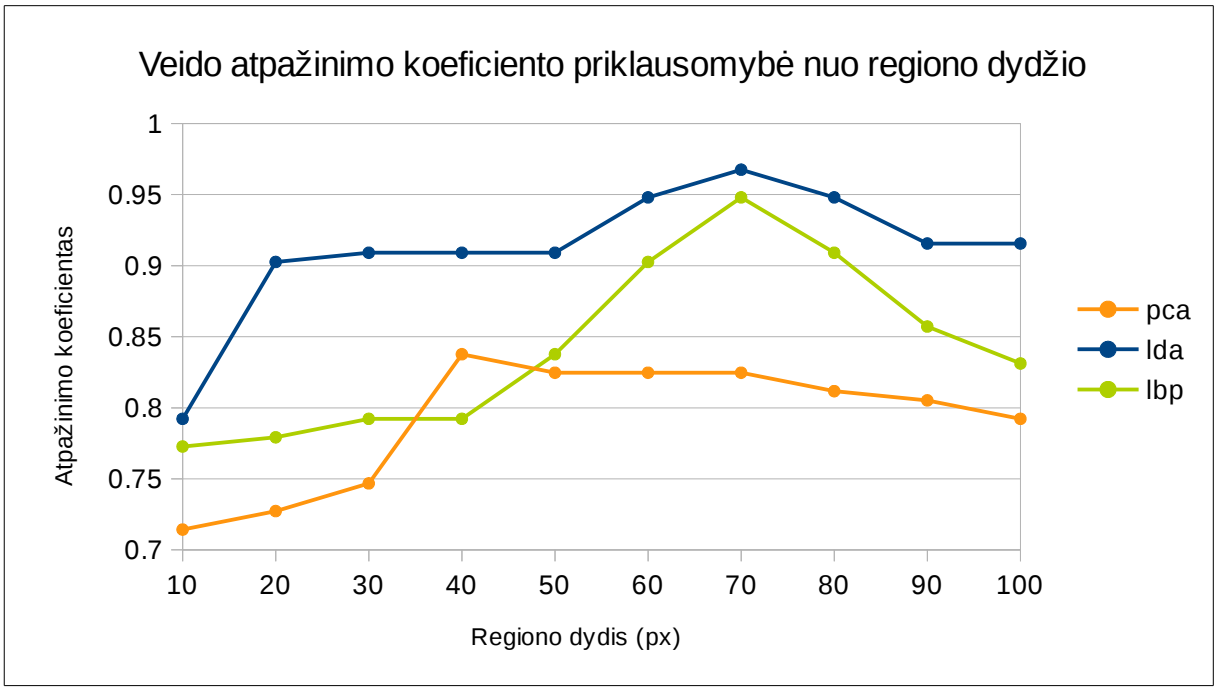
22 pav. Antrojo atrinkto algoritmų rinkinio *ROC* kreivės naudojant *LBP*

10 lentelė. Antrojo atrinkto algoritmų rinkinio *TP* ir *FP* santykiai naudojant *LBP*

3 – 8		7 – 4		9 – 4		9 – 6		9 – 7	
TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
0	0	0	0	0	0	0	0	0	0
0.01	0	0.01	0	0.62	0.01	0.01	0	0.62	0
0.6	0.01	0.6	0.01	0.69	0.11	0.62	0.01	0.71	0.1
0.73	0.21	0.76	0.27	0.73	0.23	0.7	0.1	0.73	0.23
0.8	0.55	0.83	0.9	0.75	0.38	0.75	0.36	0.75	0.37
0.83	0.83	0.83	1	0.77	0.73	0.76	0.56	0.77	0.71
0.84	1	0.83	1	0.81	1	0.81	1	0.78	1

Atlikus 5 algoritmo junginių, kurie pateikia geriausius atpažinimo koeficientus, ir 5 junginių, kurie santykinai pagerina rezultatus, pastebėta, kad geriausias apšvietimo normalizavimo metodas *PCA* metodui yra junginys 3–10. *LDA* metodui atrinkti du algoritmų

junginiai, 3–10 bei 5–9. Pastarasis, esant didžiausiam jautrumo parametrui, pateikia 97 % tikslumą. *Vietinio dvejetainio modelio* algoritmas, visiems atrinktiems algoritmų junginiams sugeneravo panašius aukštos kokybės rezultatus. 4–9 apšvietimo normalizavimo algoritmų junginys siekia 95 % tikslumo koeficientą ir šiek tiek geriau atskiria apmokymo duomenų klases nei panašų rezultatą generuojantis 5–9 algoritmų junginys. Visi metodai pagerina nenormalizuotų nuotraukų atpažinimo koeficientą ~ 2–7 kartus. Kaip matoma 23 paveiksle, veido atpažinimo koeficientas svyruoja priklausomai nuo daug požymių turinčių regionų normalizavimo dydžio.



23 pav. Veido atpažinimo koeficiento priklausomybė nuo regiono dydžio

PCA metodo žemiausia reikšmė pastebima, kai mažų veido regionų kvadratų kraštinės lygios 10 px. Atpažinimo kokybė pastebimai auga iki 84 %, esant 40 px kvadrato kraštinėms ir lėtai mažėja didinant šį parametrą. *LDA* ir *LBP* metodai veido atpažinimo piką – 98 % bei 95 % pasiekia esant ~70 px išskiriamo regiono kraštinės ilgiui. Iš visų 3 algoritmų, *LBP* atpažinimo kokybė labiausiai priklauso nuo optimalaus regiono dydžio parametro parinkimo. Analizuojant šią kreivę pastebimas staigus atpažinimo koeficiento kilimas ir kritimas. Tikslumo koeficientas parinkus optimalų smulkių požymių regiono dydį svyruoja tarp 10–20 %.

IŠVADOS

Darbo metu sukurta funkcionuojanti veido atpažinimo sistema, integruota kartu su *OpenCV*. Pasiūlytas naujas apšvietimo normalizavimo modelis ir atliktas jo eksperimentinis įvertinimas.

1. **Literatūros analizės** metu nustatyta, kad apšvietimo normalizavimo problema yra viena pagrindinių problemų, su kuriomis susiduria šiuolaikinės veido atpažinimo sistemos, o optimalus šio uždavinio sprendimui skirtas algoritmas dar nėra sukurtas. Remiantis šiuo pastebėjimu suformuluotas naujas efektyvus apšvietimo normalizacijos modelis, o jam realizuoti atrinkta 10 moksliniuose darbuose dažnai aptinkamų nuotraukos normalizavimo algoritmų.

   Pasiūlymas: Siekiant užtikrinti aukštesnį veido atpažinimo koeficientą, būtų efektyvu vienai nuotraukai taikyti keletą skirtingų normalizacijų. Šis procesas sulėtintų programos skaičiavimo laiką, tačiau pagerintų sistemos kokybę.

2. Sukurtai veido atpažinimo programai pasiūlytas **efektyvus apšvietimo normalizacijos modelis**. Siūlomo metodo ir nenormalizuotų nuotraukų atpažinimo koeficientai, esant skirtingoms slenksčio reikšmėms, skiriasi ~ 2–7 kartus. Tiesioginis regionų atskyrimas ir normalizavimas palengvina matematinę algoritmo išraišką, yra lengvai suprantamas ir realizuojamas programinėje aplinkoje.

   Pasiūlymas: Siekiant pagerinti apšvietimo normalizavimo algoritmo kokybę, siūloma įvertinti ne tik standartinių, bet ir naujai sukurtų algoritmų junginių įtaką veido atpažinimo koeficientui.

3. Siūlomo **apšvietimo normalizavimo modelis** buvo **integruotas kartu su *OpenCV***. Naudojama programinė įranga palengvina veido atpažinimo programos kūrimą. *OpenCV* biblioteka turi automatizuotą veido aptikimo algoritmą, taip pat *PCA*, *LDA* ir *LBP* algoritmų apmokymo ir testavimo metodus. Sistema dirba sklandžiai, o skaičiavimai atliekami greitai. *OpenCV* objektinė programos struktūros prigimtis leidžia sukurtą produktą naudoti kaip visumą arba atskiromis dalimis. Naudojama biblioteka yra optimali sukurtos biometrinės sistemos realizacijai.

Pasiūlymas: Algoritmo testavimo metu pastebėta, kad standartinis *Haar* požymių klasifikatorius prastai aptinka veidus dėl esamų šviesos nuokrypių. Siekiant geresnės programos automatizavimo kokybės, pirmiausia siūloma normalizuoti nuotrauką, o tada pritaikius *FAST*, *ORB* arba kitą algoritmą, aptikti akies, nosies bei lūpų regionus.

4. Sukurto apšvietimo normalizacijos metodo **eksperimentinis įvertinimas** patvirtino hipotezę, kad atskiras veido regionų apšvietimo normalizavimas gali pagerinti asmens identifikavimo kokybę. *PCA* metodui nustatytas optimalus algoritmų junginys – *Histogramos išlyginimas* bei *Tano ir Trigso* metodas – pagerino maksimalų veido atpažinimo tikslumo koeficientą nuo 45 % iki 84 %, kai smulkių regionų kraštinės ilgis lygus 40 px. *LDA* metodo atpažinimo koeficientas buvo pagerintas nuo 45 % maksimalios reikšmės iki 98 %, kai regiono kraštinių dydžiai lygūs 70 px. Optimalus algoritmų junginys šiuo atveju yra *Pleksiformo imitacijos normalizacijos* bei *Multi–scale self quotient* metodai. *LBP* algoritmo atpažinimo lygis buvo pagerintas nuo 57 % iki 95 %, kai regiono dydis lygus 70 px. Šiam metodui naudojami *Single–scale self quotient* ir *Pleksiformo imitacijos normalizacijos* algoritmai.

   Pasiūlymas: Taikant siūlomą apšvietimo normalizavimo metodiką, geriausius rezultatus sugeneravo *LBP* ir *LDA* metodai. *PCA* veido atpažinimo algoritmas buvo pakankamai nepastovus. Būsimus algoritmo gerinimo eksperimentus rekomenduojama atlikti atsisakant šio metodo.

LITERATŪROS SĄRAŠAS

Knygos ir straipsniai:

1. Agarwal A.; Goswami S. 2016; *A*n Efficient Algorithm for Automatic Car Plate Detection and Recognition, *Second International Conference on Computational Intelligence and Communication Technology;* [Interaktyvus] [žiūrėta 2017–02–10], Prieiga per internetą : <http://ieeexplore.ieee.org/document/7546687/>
2. Petro A. B.; Sbert C.; Morel J. M. 2014; *Multiscale Retinex,* University of Balearic Islands, ENS Cachan; [Interaktyvus] [žiūrėta 2016–05–12], Prieiga per internetą :

   < http://www.ipol.im/pub/art/2014/107/article.pdf >
3. Martinez A. M.; Kak A. C. 2001; PCA versus LDA*, IEEE Transactions on pattern analysis and machine inteligence* 23; [Interaktyvus] [žiūrėta 2016–05–12], Prieiga per internetą : < http://www2.ece.ohio–state.edu/~aleix/pami01.pdf>
4. Rahim A.; Hossain N.; Wahid T.; Azam S. 2013; Face Recognition using Local Binary Patterns (LBP), *Global Journal of Computer Science And Technology* 13 (4); ISSN: 0975–4350;
5. Singh A.; Singh S. K.; Tiwari S. 2012; Comparison of Face Recognition Algorithms on Dummy Faces, *The International Journal of Multimedia & Its Applications (IJMA) 4* (4); [Interaktyvus] [žiūrėta 2016–05–12], Prieiga per internetą :

   < http://airccse.org/journal/jma/4412ijma11.pdf >
6. Reddy B. A.; Raju T. A. 2015; Normalization Techniques Fusion for Face Recognition under Difficult Illumination changes, I*nternational Research Journal of Engeneering and Technology* 2(6), p. 997–1003; [Interaktyvus] [žiūrėta 2016–05–15], Prieiga per internetą: <https://www.irjet.net/archives/V2/i6/IRJET–V2I6151.pdf>
7. Olisah C. C. C.; Sulong G.; Okonkwo Uche A. K. C.; Hashim S. Z. M. 2013; Illumination Normalization for Edge–Based Face Recognition Using the Fusion of RGB Normalization and Gamma Correction, *IEEE International Conference on Signal and Image Processing,* p. 412–416; ISBN: 978–1–4799–0269–9;

8. Huang D.; Shan C.; Ardebilian M.; Wang Y.; Chen L. 2011; Local Binary Patterns and Its Application to Facial Image Analysis: A Survey, *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 41 (6), p.765–781; ISSN: 1094–6977;

9. Jobson D. J.; Rahman Z.; Woodell G. A. 1997; Properties of a Center/Surround Retinex; *IEEE Transactions on Image Processing* 6 (3), p. 451–462; ISSN: 1057–7149;

10. Cibelushi C. C.; Bourel F. 2002; Facial Expression Recognition: A Brief Tutorial Overview; [interaktyvus] [žiūrėta 2016–01–12], Prieiga per internetą: <http://homepages.inf.ed.ac.uk/rbf/Vonline/LOCAL_COPIES/CHIBELUSHI1/CCC_FB_FacExprRecCVonline.pdf>

11. Ding C.; Tao D. 2015; A Comprehensive survey on Pose–Invariant face recognition, *ACM Transactions on Intelligent Systems and Technology (TIST)* 7(3), [Interaktyvus] [žiūrėta 2015–10–12], Prieiga per internetą: <http://arxiv.org/pdf/1502.04383.pdf>

12. Prabhakar C. J.; Kumar P. U. Praveen 2011; *An Image Based Technique for Enchancement of Underwater Images,* International Journal of Machine Intelligence 3 (4), p. 217–224; [Interaktyvus] [žiūrėta 2016–05–15], Prieiga per internetą :

    <https://arxiv.org/ftp/arxiv/papers/1212/1212.0291.pdf >

13. Jobson D. J.; Rahman Z.; Woodell G. A. 1997; A Multiscale Retinex for Bridging the Gap Between Color Images and the Human Observation of Scenes, *IEEE Transactions on image processing*, 6 (7), p. 965–976; ISSN: 1941–0042;

14. Rosten E.; Drummond T. 2006; Machine learning for high–speed corner detection, *European Conference on Computer Vision,* Austrija, Gegužės 7–13 d.; p. 430–443; ISBN: 978–3–540–33832–1;

15. Rublee E.; Rabaud V.; Konolige K.; Bradski G. 2011; ORB: an efficient alternative to SIFT or SURF, *IEEE International Conference on Computer Vision*; ISSN: 1550–5499;

16. Garcia G. B.; Mercado J. O.; Torres G. A.; Perez G. S.; Meana H. P. 2011; Face Identification Based on Contrast Limited Adaptive Histogram Equalization (CLAHE), *The 2011 International Conference on Image Processing, Computer Vision, and Pattern Recognition*;

17. Bay H.; Tuytelaars T.; Gool L. V. 2008; SURF: Speeded Up Robust Features, *Computer Vision and Image Understanding* 110 (3), p. 346–359;

18. Ekenel H. K.; Stiefelhagen R. 2009; Why is Facial Occlusion a Challenging Problem?, *ICB '09 Proceedings of the Third International Conference on Advances in Biometrics,* p. 299–308; [Interaktyvus] [žiūrėta 2016–01–15], Prieiga per internetą: <http://face.cs.kit.edu/download/publications/ICB_2009_239.pdf>

19. Liu H. 2014; Face Detection and Recognition on Mobile Devices; Morgan Kaufmann 2015; ISBN: 9780124171282;

20. Delac K.; Grgic M.; Bartlet M.S 2008; Recent advances in face recognition; Viena, Austrija, InTech, [Interaktyvus] [žiūrėta 2015–10–12]; Prieiga per internetą: <http://www.face–rec.org/journals–books/Delac_Grgic_Bartlett_Recent_Advances_in_Face_Recognition.pdf>

21. Chan L. H.; Salleh S. H.; Ting C. M.; Ariff A. K. 2008; Face Identification and Verification Using PCA and LDA, *International Symposium on Information Technology*; ISSN: 2155–8973;

22. Cheong Y. K.; Yap V. V.; Nisar H. 2013; A Robust Face Recognition Algorithm under Varying Illumination using Adaptive Retina Modeling, *2013 International Symposium on Computational Models for Life Sciences,* p. 155–164; [Interaktyvus] [žiūrėta 2016–05–12], Prieiga per internetą :

<https://www.researchgate.net/publication/260829200_A_Robust_Face_Recognition_Algorithm_under_Varying_Illumination_using_Adaptive_Retina_Modeling>

23. Davis J.; Goadrich M. 2006; The Relationship Between Precision–Recall and ROC Curves, *Proceedings of the 23rd International Conference on Machine Learning*;

24. Yang J.; Yu H.; Kunz W. 2016; An Efficient LDA Algorithm for Face Recognition, *International Journal of Computer Applications*; [Interaktyvus] [žiūrėta 2016–05–12], Prieiga per internetą: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.73.1741&rep=rep1&type=pdf>

25. Zhu J.; Zheng W.; Lai J. 2013; Logarithm Gradient Histogram: A General Illumination Invariant Descriptor for Face Recognition, *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*; [Ineraktyvus]

[žiūrėta 2016–06–08], Prieiga per internetą : <http://ieeexplore.ieee.org/document/6553738/>

26. Bertok K.; Fazekas A. 2016; Face Recognition on Mobile Platforms, *7th IEEE International Conference on Cognitive Infocommunications*; p. 37–42; ISBN: 978–1–5090–2645–6;

27. Toh K.; Kim J.; Lee S. 2008; Maximizing area under ROC curve for biometric score fusion, *The Journal of the Pattern recognition Society* 41; ISSN: 0031–3203;

28. Calonder M.; Lepetit V.; Ozuysal M.; Trzcinski T.; Strecha C.; Fua P. 2012; BRIEF: Computing a Local Binary Descriptor Very Fast, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (7), p. 1281–1298; ISSN: 0162–8828;

29. Leszcynski M. 2010; Image Preprocessing for Illumination Invariant Face Verification, *Journal of Telecommunications and Information Technology* 4, p. 19–25; [Interaktyvus] [žiūrėta 2016–06–10], Prieiga per internetą: <http://www.nit.eu/czasopisma/JTIT/2010/4/19.pdf >

30. Nishiyama M.; Kozakaya T.; Yamaguchi O. 2008; Illumination Normalization using Quatient Image–based Techniques, InTech; [Interaktyvus] [žiūrėta 2016–06–10]; Prieiga per internetą: <http://cdn.intechopen.com/pdfs–wm/5895.pdf >

31. Turk M.; Pentland A. 1991; Eigenfaces for Recognition, *Journal of Cognitive Neuroscience* 3 (1), p. 71–86; ISSN: 0898–929X;

32. Ayache N. 1995; Medical Computer Vision, Virtual Reality and Robotics, *Elsevier Science B.V. Image and Vision Computing* 13 (4); ISBN 978–3–540–49197–2;

33. Belhumeur P. L.; Hespanha J. P.; Kriegman D. J. 1997; Eigenfaces vs Fisherfaces: recognition using class specific linear projection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (7), p. 711–720; ISSN: 0162–8828;

34. Sinha P.; Balas B.; Ostrovsky Y.; Russel R. 2006; Face Recognition by Humans: Nineteen Results All Computer Vision Researchers Should Know About, *Proceedings of the IEEE* 94 (11); ISSN: 0018–9219;

35. Viola P.; Jhones M. 2001; Rapid Object Detection using Boosted Cascade of Simple Features, *Accepted Conference in Computer Vision and Pattern Recognition*; [Interaktyvus] [žiūrėta 2016–05–12], Prieiga per internetą :

<https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola–cvpr–01.pdf>

36. Szeliski R. 2010; Computer Vision: Algorithms and Applications, *Springer*; p. 1–29; 205–267; ISBN: 978–1848829343;

37. Saleh S. A. M.; Ibrahim H. 2012; Mathematical Equations for Homomorphic Filtering in Frequency Domain: A Literature Survey, *International Conference on Information and Knowledge Managament*; [Interaktyvus] [žiūrėta 2016–06–12], Prieiga per internetą : <http://www.ipcsit.com/vol45/015–ICIKM2012–M0029.pdf >

38. Shan S.; Gao W.; Cao B.; Zhao D. 2003; Illumination Normalization for Robust Face Recognition Against Varying Lighting Conditions, *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*; ISBN: 0–7695–2010–3;

39. Dhillon S. S.; Sharma S. 2016; A review on Histogram Equalization Techniques for Contrast Enchancement in Digital Image, *International Journal of Innovation and Advancement in Computer Science* 5 (1), p. 9–13; ISSN: 2347–8616;

40. Tuytelaars T.; Mikolajczyk K. 2007; Local Invariant Future Detectors: A Survey, *Computer Graphics and Vision* 3 (3), p. 177–280;

41. Yu T.; Meng X.; Zhu M.; Han M. 2016; An Improved Multi–Scale Retinex Fog and Haze Image Enhancement Method, *International Conference on Information System and Artificial Intelligence*; ISBN: 978–1–5090–1586–3;

42. Braje W. L.; Kersten D.; Tarr M. J.; Troje N. F. 1998; Illumination effects in face recognition, *Psychobiology* 26 (4), p. 371–380; [Interaktyvus] [žiūrėta 2016–01–12], Prieiga per internetą: <http://www.biomotionlab.ca/Text/braje_etal_98.pdf>

43. Tang X.; Triggs B. 2007; Enhanced Local Texture Feature Sets for FaceRecognition Under Difficult Lighting Conditions, *IEEE Transactions on Image Processin* 19 (6), p. 1635–1650; ISSN: 1057–7149;

44. Xie X.; Zheng W.; Lai J.; Yuen P. C.; Suen C. Y. 2011; Normalization of Face Illumination Based on Large–and–Small–Scale Features; *IEEE Transactions on image processing* 20 (7); [Interaktyvus] [žiūrėta 2016–06–12]; Prieiga per internetą :

<http://isee.sysu.edu.cn/~zhwshi/Research/PreprintVersion/Normalization%20of%20Face%20Illumination%20Based%20on%20Large–%20and%20Small–%20Scale%20Features.pdf >

45. Xu X.; Liu W.; Li L. 2014; Low Resolution Face Recognition in Surveillance Systems, *Journal of Computer and Communications* 2, p. 70–77 ;

46. Zhou X.; Wang K.; Fu J. 2016; A Method of SIFT Simplifying and Maching Algorithm Improvement, *International Conference on Industrial Informatics – Computing Technology, Intelligent Technology, Industrial Information Integration*; [Interaktyvus] [Žiūrėta 2016–02–14], Prieiga per internetą: <http://ieeexplore.ieee.org/document/7823496/>

47. Shi Y.; Li H. Y.; Soong F. K. 2007; A United Framework for Symbol Segmentation and Recognition of Handwritten Mathematical Expressions, *IEEE – Ninth International Conference on Document Analysis and Recognition,* p. 854–858; ISBN: 978–0–7695–2822–9

48. Ojala T.; Pietikainen M.; Harwood D. 1996; A comparative study of texture measures with classification based on feature distributions, *Pattern Recognition* 29.

49. Tahmoush D.; Samet H. 2016; Computer Vision and Learning Approaches to Medical Imaging, *Second International ECCV Workshop*; [Interaktyvus] [žiūrėta 2017–01–15], Prieiga per internetą: <https://ai2–s2–pdfs.s3.amazonaws.com/e5bc/0644d3be87acdd403b182f95c212b8630a3f.pdf>

50. Mathhies L.; Maimone M.; Johnson A.; Cheng Y.; Willson R.; Villalpando C.; Goldberg S.; Huertas A. 2007; Computer vision on Mars, *International Journal of Computer Vision* 75 (67); [Interaktyvus] [žiūrėta 2016–02–12], Prieiga per internetą:

<http://www.ri.cmu.edu/pub_files/pub4/matthies_larry_2007_1/matthies_larry_2007_1.pdf>

Interneto šaltiniai:

51. Conner E. 2016; Autonomous Vehicles: the Tesla Model S; [Interaktyvus] [žiūrėta 2017–02–10] Prieiga per internetą: <http://www.pitt.edu/~elc99/WA3Final.pdf>

52. Mathworks 2017; Matlab; [Interaktyvus] [žiūrėta 2017–02–15],Prieiga per internetą: <https://se.mathworks.com/products/matlab/features.html>

53. OpenCV 2015; Face recognition with open cv; [Interaktyvus] [žiūrėta 2015–10–12], Prieiga per internetą: <http://docs.opencv.org/modules/contrib/doc/facerec/facerec_tutorial.html>

54. OpenCV 2016; OpenCV Features; [Interaktyvus] [žiūrėta 2016–05–10], Prieiga per internetą: <http://opencv.org/>

55. OpenCV 2016; Face detection using Haar Cascades; [Interaktyvus] [žiūrėta 2016–05–10], Prieiga per internetą: <http://docs.opencv.org/master/d7/d8b/tutorial_py_face_detection.html#gsc.tab=0>

56. Price R. 2015; Facebook‘s scarily accurate facial–recognition tech can now recognise you even if yoit can‘t see your face; [Interaktyvus] [žiūrėta 2015–10–12], Prieiga per internetą: <http://uk.businessinsider.com/facebook–facial–recognition–technology–ai–cant–see–face–2015–6>