

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Karl-Magnus Laikoja

**Inimese näo automaatne kirjeldamine
loomulikus eesti ja inglise keeles**

Bakalaureusetöö (9 EAP)

Juhendaja: Sven Aller, MSc

Tartu 2024

Inimese näo automaatne kirjeldamine loomulikus eesti ja inglise keeles

Lühikokkuvõte:

Käesoleva bakalaureusetöö eesmärgiks on luua ning implementeerida algoritm, millega on võimalik inimese nägu kirjeldada. Lisaeesmärgiks on luua veebirakendus, mis integreeriks algoritmi implementatsiooni ning tagastaks kasutajatele sisendpildist leitud näo kirjelduse loomulikus eesti või inglise keeles. Töö raames analüüsitakse sarnaseid olemasolevaid rakendusi ning põhjendatakse loodud rakenduse eripära. Seejärel kirjeldatakse välja töötatud algoritmi ja valminud rakendusi. Viimaks testitakse rakenduse täpsust ning analüüsitakse selle erinevaid puudujääke ja edasiarendamise võimalusi.

Võtmesõnad:

Inimese nägu, pilditöötlus, pildi analüüs, tehisnägemine, automaatne kirjeldus, loomulik eesti keel, loomulik inglise keel, süsteemidisain

CERCS: P175 Informaatika, süsteemiteooria

Automatic description of the human face in natural Estonian and English languages

Abstract:

The objective of this bachelor's thesis is to create and implement an algorithm which can describe the human face. A secondary goal is to create a web application that integrates said implementation, which returns a description of the face found in a user inputted image in natural Estonian or English language. Several similar applications are analysed within the scope of the thesis and the specialty of the created application is justified. Thereupon the created algorithm and applications are described in detail. Finally, the accuracy of the application is tested, and its shortcomings are analysed with possibilities for further development.

Keywords:

Human face, image processing, image analysis, computer vision, automatic description, natural Estonian language, natural English language, system design

CERCS: P175 Informatics, systems theory

Sisukord

Sissejuhatus	6
1 Olemasolev tarkvara.....	7
1.1 Näotuvastus ja näokirjeldus	7
1.1.1 Amazon Rekognition	7
1.1.2 Deepface.....	8
1.1.3 Cognitec FaceVACS Engine.....	8
1.1.4 Face Recognition.....	8
1.2 Keelemudelid	8
1.3 Analüüs	9
2 Kasutatud tarkvara ja tööriistad	10
2.1 Python	10
2.1.1 Face Recognition.....	10
2.1.2 OpenCV.....	10
2.2 Java.....	11
2.2.1 Spring Boot	11
2.2.2 Apache Maven	11
2.3 Vue	11
2.3.1 Vue CLI.....	12
2.3.2 PrimeVue.....	12
2.4 Node	12
2.5 NPM	12
2.6 VirtualBox.....	13
3 Sisendpildilt leitud näo kirjeldamine	14
3.1 Värvide normaliseerimine	14
3.2 Kirjelduse olemus.....	16

3.3	Üldised atribuudid	18
3.3.1	Naha värv	18
3.3.2	Näo kuju	20
3.4	Juuksed	24
3.4.1	Juuste olemasolu	24
3.4.2	Juuste värv	28
3.5	Näokarvad	29
3.5.1	Näokarvade tihedus	29
3.5.2	Näokarvade värv	31
3.6	Silmad	32
3.6.1	Silmade asetus	33
3.6.2	Silmade kuju	34
3.6.3	Silmade värv	36
3.7	Nina kuju	37
3.8	Valminud andmestik	38
3.9	Täpsust mõjutavad faktorid ning vajakajäämised	39
4	Veebirakendus	41
4.1	Arhitektuur	41
4.1.1	Tagaliides	42
4.1.2	Eesliides	43
4.2	Loomulikus keeles väljundi koostamine	43
4.3	Veebilehe funktsioonid	44
4.4	Turvalisus ja privaatus	44
5	Valminud rakenduse hindamine	46
5.1	Metoodika ja tulemused	46
5.2	Kasutajate tagasiside	47

6	Võimalusi edasiarendamiseks	49
	Kokkuvõte	51
	Viidatud kirjandus	52
	Lisad	59
I.	Kasutajatestimise küsimustik	59
II.	Kasutajatestimise tulemuste tabel	69
III.	Litsents	72

Sissejuhatus

Füüsilised deskriptorid (ingl *descriptor*) [1] on sõnalised kirjeldused inimese erinevatest atribuutidest nagu kehaline vorm, sugu või pikkus. Need on palju kasutust leidnud profileerimise eesmärgil nii-öelda „mustades nimekirjades“. Näiteks säilitab Ameerika Ühendriikide New Jersey osariigi valitsuse Hasartmänguseaduste Jõustamise Osakond (ingl *Division of Gaming Enforcement*) keelunimekirja, kus on avalikustatud hasartmängu asutustest keelatud isikute andmed koos kirjeldustega nende kehalistest omadustest [2]. Seevastu on antud inimestest kirjelduste koostamine manuaalne töö ning on üldiselt piiratud arvu atribuutidega, eelneva näite puhul on välja toodud vaid silma värv ja juuste värv. Kaasnevast pildist on aga võimalik eraldada ka muud kasulikku informatsiooni. Tänapäeval on tehisintellekti ja masinõppe arenguga aina populaarsemaks muutumas erinevad tehisnägemisalgoritmide ning nende implementatsioonid, mille tulemusena on juba automatiseeritud erinevaid pilditöötlusega seotud ülesandeid. Sarnaselt teostatav on füüsiliste deskriptorite laiendamine inimese näo osas ning selle töö automatiseerimine [3].

Käesoleva bakalaureusetöö eesmärk on arendada algoritm, mis suudab sisendpildilt tuvastada inimese näo ning kirjeldada selle erinevaid osi ja atribuute nagu naha värv, silmade kuju ning näokarvade tihedus. Sihiks on see algoritm implementeerida rakenduses, mis suudab tulemusi esitada nii eesti kui inglise keeles. Tulemuseks saadud rakendus peaks olema ka avaliku ressursina kättesaadav. Lisaks on eesmärk luua eraldi veebirakendus, mis võtab kasutajalt sisendpildi ja valitud keele ning koostab kirjeldustulemuste põhjal loomulikus keeles kirjelduse. Selline rakendus võimaldab hõlpsasti arendatud algoritmi kasutada.

Lõputöö raames uuritakse esmalt juba olemasolevaid rakendusi, mis kirjeldavad kindlaid omadusi inimese näo osas või on tänu tehisintellektile võimelised sõltuvalt sisendist seda tegema. Analüüsitakse nende tugevusi ja puudujääke ning leitakse arendamata nišš, mida loodav rakendus täidaks. Seejärel kirjeldatakse algoritmi erinevate sammude ideed ning implementatsiooni. Pärast seda kirjeldatakse arendatud veebirakendust, mille abil saab kasutaja oma enda sisendpildist loomulikus eesti või inglise keeles näokirjelduse genereerida. Viimaks testitakse valminud rakenduse täpsust erinevate sisenditega, andes sellele lõpliku hinnangu, ning analüüsitakse selle puudujääke ja edasiarendamise võimalusi.

1 Olemasolev tarkvara

Järgnev peatükk kirjeldab lühidalt bakalaureusetöö kirjutamise hetkel eksisteerivat tuntumat tarkvara inimese näo tuvastamiseks ning kirjeldamiseks. Analüüsitakse ka erinevate rakenduste tugevusi ja nõrkusi.

1.1 Näotuvastus ja näokirjeldus

Näotuvastus on kiirelt arenev valdkond, mille arengusse on panustanud mitmed ettevõtted ja isegi riigid, kelle huvides võib olla kindla näo sidumine kindla isiksusega. Näiteks on Hiina Zhengzhou linna politseijaoskond võtnud kasutusele näotuvastustarkvaraga varustatud päikeseprillid, mis suudavad kiirelt siduda leitud nägusid andmebaasis oleva näoga [4]. Selle tulemusena on võimalik kuriteos kahtlustatavaid isikuid automaatselt tuvastada. Siinkohal tuleb mainida, et inglise keeles on olemas kaks erinevat mõistet, *face detection* ja *face recognition*. Esimene neist tähendab vaid sisendpildilt või -videolt nägude koordinaatide leidmist ja teine tähendab selle näo sidumist andmebaasis oleva näoga [5]. Eesti keeles kasutatakse aga mõlema kontseptsiooni kirjeldamiseks mõistet **näotuvastus**. Järgnevates peatükkides mõeldakse antud mõistet kasutades nägude koordinaatide leidmist. Kuna näotuvastus on fundamentaalne osa näo kirjeldamisest, siis on vaja välja tuua erinevaid olemasolevaid näotuvastusrakendusi ning kaalutleda nende tugevusi ja nõrkusi. Lisaks tuleb välja tuua nende võimalusi nende kasutamiseks näo kirjeldamise eesmärgil.

Näokirjeldus on erinevalt näotuvastusest üsnagi subjektiivne teema. Erinevad rakendused on võtnud kasutusele erinevad lähenemised ja arusaamised sellest, mida näo kirjeldamine tähendab. Näiteks Deepface [6] suudab sisendi põhjal ennustada vanust, sugu, emotsiooni ja rassi, aga Cognitec FaceVACS Engine [7] leiab sisendi põhjal soo, vanuse, näogeomeetria, prillide olemasolu ja muud. Ühine on kõigi puhul see, et näo kirjeldamine tähendab näost leitatavate või näo põhjal tuletatavate markerite kirjeldamist.

1.1.1 Amazon Rekognition

Amazon Rekognition [8] on pildituvastus- ja videoanalüüsi tarkvara, mis võimaldab kasutada Amazoni poolt arendatud süvaõppe tehnoloogiat kõrge täpsusega näotuvastuseks, kirjeldamiseks ja võrdlemiseks. Kuigi Amazon Rekognition on tuntud nimi näotuvastuse valdkonnas ning pakuks kõrgeima taseme tulemusi seoses tuvastuse täpsusega, on see mõeldud peamiselt ettevõtetele, kes peavad töötleva suurtes kogustes pilte ja videoid. Sõltuvalt kasutusest on see ka tasuline.

1.1.2 Deepface

Deepface [6] on Pythoni moodul näotuvastuseks ja erinevate näo atribuutide analüüsiks. Raamistik kasutab näotuvastuseks mitmeid modernseid mudeleid ning võimaldab sisendpildilt leida näo koordinaadid ja kirjeldada ennustatud vanust, sugu, emotsiooni ja rassi. Kuna moodul on kirjutatud Pythoni programmeerimiskeeles, on see kergesti integreeritav samas keeles kirjutatavasse rakendusse. Deepface'i peamiseks puuduseks on see, et rakendus tuvastab pildilt vaid näo koordinaadid. Näo erinevate osade koordinaate Deepface ei tuvasta.

1.1.3 Cognitec FaceVACS Engine

Cogniteci poolt arendatud FaceVACS Engine [7] on tippteaduse tulemusena loodud kõrge täpsusega rakendus, mis suudab lisaks näotuvastusele leida palju erinevaid portreeomadusi nagu näiteks poos, miimika, juuksestiil, prillide olemasolu. FaceVACS Engine'i suurimaks puuduseks on selle tasulisus.

1.1.4 Face Recognition

Veel üks levinud Pythoni moodul on Face Recognition [9], mis võimaldab mitte ainult näo koordinaadid pildilt leida, vaid ka erinevate näo osade nagu silmad, nina, huuled. Face Recognition võimaldab hõlpsasti kirjeldada pildilt leitud nägu, analüüsides hoopis selle üksikuid osi nagu nina ja silmad.

1.2 Keelemudelid

Viimaste aastate järsk areng keelemudelite rakendatavuses teeb neist ilmselge kandidaadi näokirjeldusrakenduseks. Modernsed mudelid nagu GPT-4 on teatavasti võimelised interpreteerima sisendpilte [10] ning sõltuvalt kasutajasisendist tagastama loomulikus keeles kirjelduse pildilt leitud objektidest. Peamine probleem selliste mudelitega on nende mittedeterministlik väljund. Samasugune sisend ei garanteeri iga kord samasugust väljundit [11], mistõttu on keelemudeli mittetriviaalne kasutus näokirjeldusrakenduses keeruline või isegi võimatu. Lisaks on sellised mudelid ja nende mudelite peale ehitatud rakendused tihti suletud lähtekoodiga. See võib tekitada probleeme seoses isikuandmete kaitse seadustega, kuna antud rakenduste andmete kogumise poliitika ei ole kontrollitav.

1.3 Analüüs

Nii-öelda nägu kirjeldavaid rakendusi on juba olemas, aga neil kõigil leidub puudujääke. Suurettevõtete poolt arendatud tippteaduslikud mudelid annavad väga muljetavaldavaid tulemusi täpsuse osas, kuid ei ole kasutajale kergelt kättesaadavad. Need on paljudel juhtudel tasulised, nõuavad tutvumist programmeerimisliidesega ning vajavad tarkvara enda paigaldamist. Need keskenduvad üldiselt ka väga spetsiifiliste omaduste kirjeldamisele, näiteks FaceVACS Engine annab infot juuksestiili kohta, aga mitte juuksevärvi kui palju üldisema ja võib-olla tähtsama omaduse kohta. Keelemudelid on see-eest suutelised antud puudust lahendama. Nende puhul on probleemiks mittedeterministlik väljund, mille tulemusena võib sama sisendpilt ja sisendtekst anda eri katsetustel erinevas formaadis kirjeldusi. Mainimata ei saa jätta ka asjaolu, et enamike leitud rakenduste väljundid ei genereerita loomulikus keeles. Lisaks piirduvad need üldiselt inglise keelega ning eestikeelseid rakendusi veel autorile teadaolevalt pole.

Väljatoodud Pythoni moodulitest suudab Deepface sooritada ka näokirjeldust, kuid puudujäägiks on selle üldisus. Vanus, sugu, emotsioon ja rass on küll tähtsad omadused, kuid need on niivõrd üldised, et inimene ei suudaks ainuüksi nende põhjal oma mõttepildis kirjeldatud nägu ette kujutada. See-eest pakub Face Recognition unikaalset võimalust: kuna peale näo koordinaatide tuvastatakse ka näo erinevate osade koordinaadid, on võimalik luua neid atribuute kirjeldav algoritm.

Bakalaureusetöö eesmärgiks on Face Recognition moodulit kasutades luua eesti- ja ingliskeelne rakendus, mis annab kasutajale vastavas loomulikus keeles kirjelduse sisendpildilt leitud näo kohta. Näokirjeldus ise koosneb näo erinevate atribuutide nagu silmade värvi, nina kuju, näokarvade tiheduse kategoriseerimisest. Lisaks kirjeldusrakendusele on eesmärgiks luua seda integreeriv veebirakendus, mis koostaks leitud omaduste põhjal loomulikus keeles kirjelduse ning pakuks kasutajale lihtsat võimalust kirjeldusrakenduse kasutuseks.

2 Kasutatud tarkvara ja tööriistad

Järgnevas peatükis kirjeldatakse peamiseid tehnoloogiaid, mis osutusi vajalikuks rakenduste arendamiseks. Lisaks põhjendatakse lühidalt valikuid ja kirjeldatakse olemasolu juhul ka alternatiive.

2.1 Python

Rakenduse tagaliideses kasutatakse sisendpildi töötlemisel ja kirjelduse koostamisel Pythoni. Python sai valitud kirjeldusalgoritmi teostamise keeleks peamiselt selle tõttu, et see pakub palju võimalusi pilditöötluks ning ka väljavalitud näotuvastuse moodulit. Alternatiiviks oli C++, mille jaoks on ka loodud teeke pilditöötluks ja tehisnägemiseks, kuid sobivat näotuvastuse moodulit pole selle jaoks autorile teadaolevalt loodud.

2.1.1 Face Recognition

Face Recognition [9] on Pythoni moodul, mis pakub võimalusi näotuvastuseks. See on võimeline leidma sisendpildilt nägude koordinaadid ning nende nägude seest erinevate näoosade koordinaadid. Moodul suudab tuvastada lõua, kulmude, ninasilla, ninaotsa, silmade ja huulte koordinaadid. Lisaks pakub Face Recognition võimalust võrrelda nägusid erinevates piltides, mille tulemusena on võimalik leitud nägu siduda kindla isikuga. Rakenduses on moodulit kasutatud näo erinevate osade koordinaatide leidmiseks.

2.1.2 OpenCV

OpenCV [12] on Pythoni moodul, mis pakub tehisnägemisfunktsionaalsust. See on avatud lähtekoodiga ja sisaldab rohkem kui 2500 erinevat tehisnägemisalgoritmi [13]. Näiteks on saadaval implementatsioon Canny servatuvastusalgoritmist, mille avaldas John Francis Canny 1986. aastal meetodina sisendpildilt tuvastada servasid arvutusliku lähenemisega [14]. Palju võimalusi on ka pilditöötluks, mis jagunevad üldisemalt värvide manipuleerimiseks, morfoloogilisteks ja geomeetrilisteks muundamisteks. OpenCV pakub ka näotuvastustehnoloogiat, aga bakalaureusetöö kirjutamise hetkel on antud funktsionaalsus veel beeta-testimise faasis. Rakenduses kasutatakse OpenCV moodulit peamiselt pilditöötluks eesmärgil, aga ka selleks, et rakendada tehisnägemise algoritme nagu eelnevalt mainitud Canny algoritm.

2.2 Java

Rakenduse tagaliidese server on kirjutatud programmeerimiskeeles Java. Serveri ülesandeks on vastu võtta sissetulevad näokirjelduse HTTPS päringud, valideerida sisendid, salvestada ning lõpus kustutada pildifail, käivitada Pythoni programm saadud sisenditega ning tagastada päringule vastus. Alternatiiviks oli kasutada serveri jaoks mõnda Pythoni raamistikku nagu Flask [15], kuid rakenduse jõudlus osutus peale algoritmi implementeerimist probleemiks. Probleemi leevendamiseks sai valitud Java, mis peaks olema Pythonist paljudel juhtudel kiirem. Näiteks kulub Pythonil maatriksarvutuste tegemiseks 56 korda rohkem aega kui C puhul, Java on aga ainult 1,69 korda aeglasem [16].

2.2.1 Spring Boot

Spring Boot [17] on Spring raamistiku laiendus, mis võimaldab lihtsat Java veebirakenduste arendamist, käsitledes automaatselt objektide elutsükli. Selle tulemusena saab arendaja keskenduda rohkem rakenduse loogika kui arhitektuuri peale. Lisaks pakub Spring Boot hõlpsat sisseehitatud (ingl *embedded*) Tomcat [18] veebiserveri kasutust ja konfigureerimist. Rakenduses on kasutatud Spring Booti tagaliidese serveri ja loogika arendamiseks.

2.2.2 Apache Maven

Apache Maven [19] on projektihaldus tööriist, mis lihtsustab sõltuvuste (ingl *dependencies*) haldust Java rakendustes. See installeerib need automaatselt vastavalt defineeritud projekti objektimudelile (POM). Rakenduses on kasutatud Apache Maveni selleks, et hallata rakenduse erinevaid sõltuvusi nagu näiteks eelmainitud Spring Boot.

2.3 Vue

Rakenduse eesliidese jaoks on kasutatud Vue.js raamistikku. Vue.js on JavaScripti programmeerimiskeelel põhinev raamistik, mis on loodud kasutajaliideste arendamiseks. Vue.js puhul lisatakse üldise HTML, CSS ja JavaScript kohale komponendipõhise programmeerimismudeli [20]. See annab võimaluse rakenduse eesliideses esitada HTML väljundit olekupõhiselt, mida Vue.js jälgib automaatselt ja uuendab selle muudatusi tuvastades dokumendi objektimudelit (DOM). Vue.js üheks eeliseks on veel see, et väljundi koostamine toimub kliendi poolel. Serveri poolel koosneb rakendus vaid üksikust lehest (niinimetatud *Single-Page Application*), mille tulemusena on kasutaja vaatest veebileht tunduvalt kiirem ja reageerivam. Kuigi Vue.js pakub võimalust ka serveripoolseks esituseks,

on eesliidese loomiseks valitud esimene variant. Rakenduses on kasutatud raamistikku veebilehe jaoks, mida serveerib kasutajale Webpack DevServer [21].

2.3.1 Vue CLI

Vue CLI [22] on tööriist, mis kiirendab ja lihtsustab Vue.js projektide arendamist. Vue CLI haldab arendaja eest erinevaid konfiguratsioone ja integreerib automaatselt mitmed ametlikud pistikprogrammid, mida projekti jaoks võib vaja minna. Rakenduses on kasutatud Vue CLI-d lihtsustamaks eesliidese projekti arendamist.

2.3.2 PrimeVue

PrimeVue [23] on Vue.js jaoks arendatud teek, mis pakub rohkelt erinevaid kohandatavaid ja visuaalselt atraktiivseid komponente. See võimaldab taaskasutada erinevaid kasutajaliidese elemente, muutes vaid nende stiili, sisu või osaliselt funktsionaalsust. Rakenduses on kasutatud PrimeVue komponente tegemaks eesliidest kasutajasõbralikumaks, reageerivamaks ja visuaalselt atraktiivsemaks.

2.4 Node

Node.js [24] on JavaScripti käitussüsteem, mida kasutatakse veebirakenduste arendamiseks. Kuna Node.js-il on asünkroonne sündmusepõhine arhitektuur (ingl *asynchronous event-driven architecture*), on see väga efektiivne süsteem serveri arendamiseks. Sündmused käivituvad iga ühenduse peale, aga kui ühendusi pole, siis Node.js jääb ooteolekusse. Rakenduses on kasutatud Node.js-i selleks, et eesliidese serveri kõrvale luua sekundaarne server. Selle eesmärgiks on suunata sissetulevad päringud pordilt 80 pordile 443, tagades brauserites HTTPS kasutamise. Tulemusena ühendub kasutaja automaatselt eesliidese serveriga, mis kasutab HTTPSi, isegi kui ta defineeris protokolliks HTTP või brauser kasutas seda vaikimisi.

2.5 NPM

Node Package Manager (NPM) [25] on käsurealiides, mis võimaldab kasutajatel jagada koodipakette, laadides need üles ning installides neid npm registrist. Kogu rakenduse eesliidese arendamise protsessi jooksul kasutati NPM-i paketiholduse eesmärgil.

2.6 VirtualBox

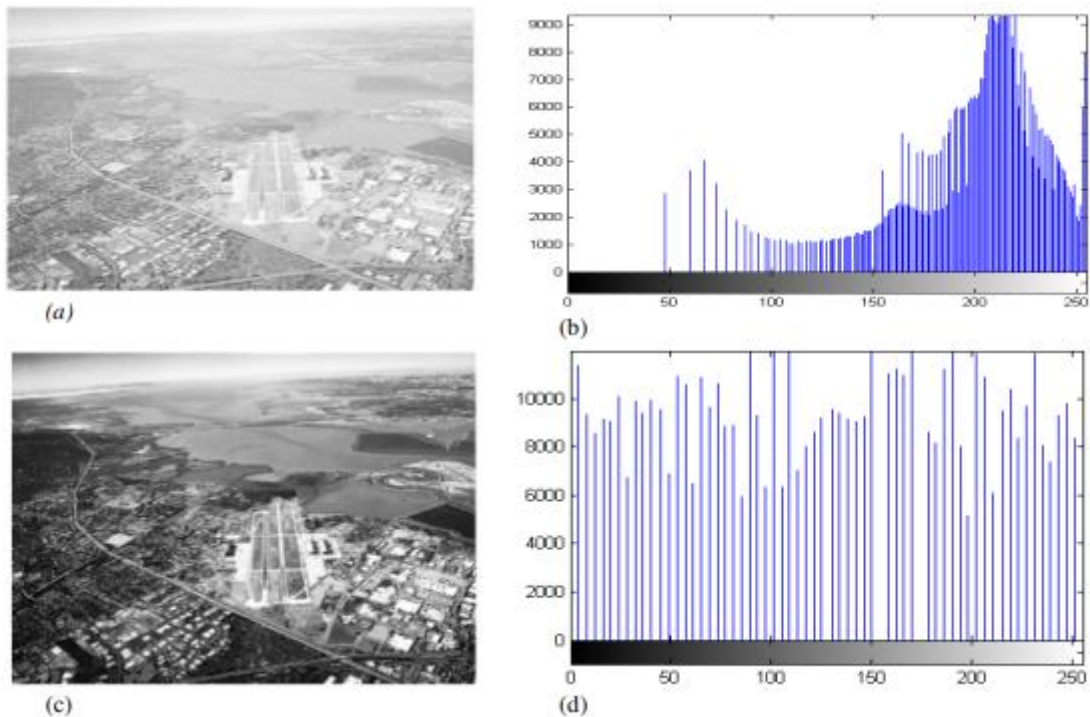
Oracle VM VirtualBox [26] on avatud lähtekoodiga virtualiseerimise tarkvara. Kuna arendamisprotsess hõlmab mitmete sõltuvuste installimist kindlate versioonidega, lihtsustab seda olulisel määral operatsioonisüsteemi tasemel töötav paketi haldustarkvara. Seetõttu sai arendamiskeskonna operatsioonisüsteemiks valitud Ubuntu 22.04.2 LTS. Ubuntu [27] on avatud lähtekoodiga Linuxil põhinev operatsioonisüsteem, mis sisaldab vaikimisi paketi haldustööriista APT (*Advanced Packaging Tool*). Lisaks, kuna rakenduse hostimiseks valitud masin on tänapäevaste standardite järgi vähese jõudlusega, sai sama operatsioonisüsteem valitud ka hostimiseks. Linux vajab võrreldes Windowsiga põhifunktsionaalsuse toimimiseks oluliselt vähem ressursse [28]. Rakenduse hostimine virtuaalmasinas tagab ka parema turvalisuse: võimaliku turvaauku ärakasutamine rakenduses piirdub tekitatud kahju poolest vaid virtuaalmasinaga ning hostmasin ise on kaitstud virtuaalmasina isoleerituse tõttu [29].

3 Sisendpildilt leitud näo kirjeldamine

Järgnevas peatükis on kirjeldatud, milline näeb välja näo kirjeldamise algoritm, mida täpselt kirjeldatakse ning kuidas tulemused genereeritakse. Rakenduse mitmekeelsuse tõttu on mainitud ka lisaks eestikeelsetele määratavatele kategooriatele ingliskeelsed vasted. Erinevad kirjeldatavad atribuudid on jagatud loetavuse eesmärgil üldistatud kategooriatesse, aga meetoodika on kohati erinevate kategooriate alamosade vahel sarnane. Lisaks on välja toodud teadaolevad puudused iga kirjeldusmeetoodika puhul ning seejuures erinevad faktorid, mis mõjutavad tulemuste täpsust negatiivselt.

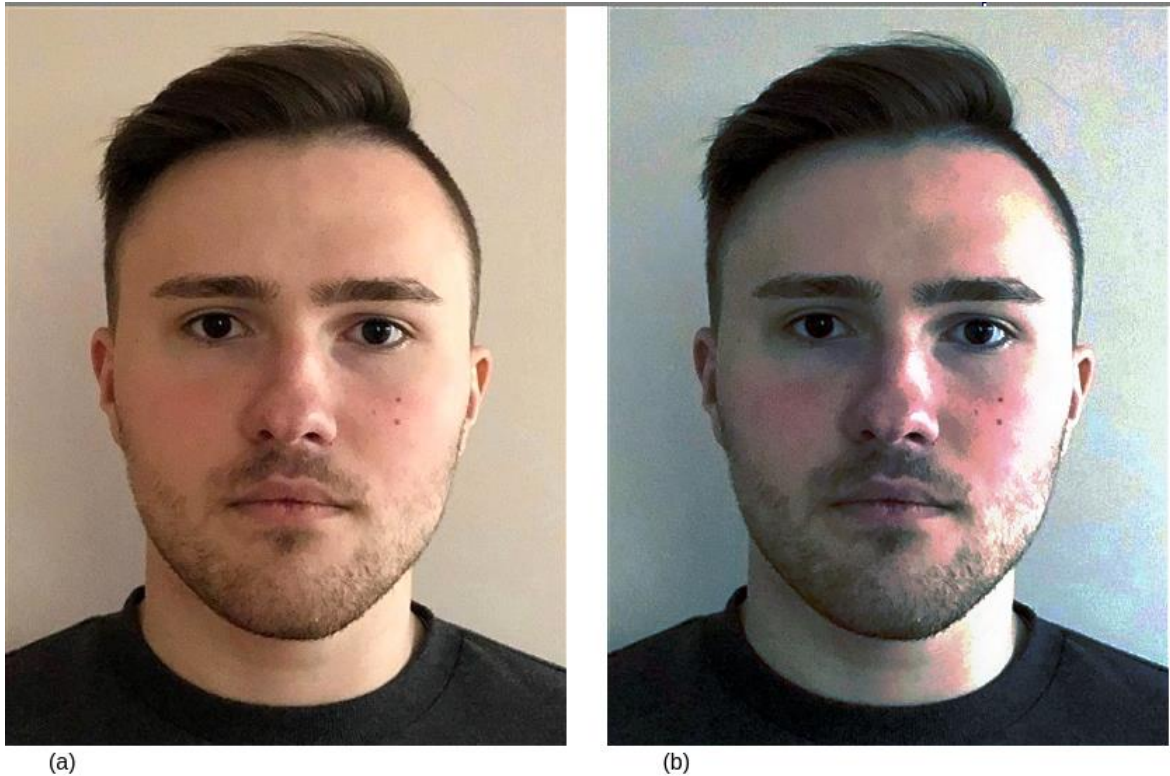
3.1 Värvide normaliseerimine

Kuna suure osa näo kirjeldusest moodustab erinevate värvide klassifitseerimine ning sisendpildi värve mõjutab pildi valgustus, tuleb esimese sammuna sisendpildi värvid normaliseerida. Selle eesmärgiks on suurendada pildi kontrasti ja seekaudu vähendada valgustuse mõju pildi värvidele. Selleks on kasutatud histogrammi võrdustamise (ingl *Histogram Equalization*) meetodit. Histogrammi võrdsustamine on meetod sisendpildi kontrasti suurendamiseks, jaotades pikslite intensiivsuse kogu võimaliku ulatuse peale [30]. Värviga piltide puhul, millel on eraldi kanalid punase, roheline, sinise värvi jaoks (R, G, B), tuleb meetodit rakendada iga värvikanali puhul eraldi. Tulemusena on minimaalne väärtus iga kanali jaoks 0 ja maksimaalne 255 ning värvi intensiivsus on jaotatud nende väärtuste vahele. Näitena on välja toodud joonis 1, mis kujutab halltoonides pilti enne ja pärast meetodi rakendamist koos vastavate histogrammidega.



Joonis 1. (a) näitab sisendpilti enne histogrammi võrdsustamist, (b) näitab sisendpildi halltooni kanali histogrammi, (c) näitab väljundpilti peale histogrammi võrdsustamist ja (d) näitab väljundpildi halltooni kanali histogrammi [30].

Histogrammi võrdsustamine on OpenCV moodulis realiseeritud funktsiooniga *equalizeHist* [31]. Funktsioon võtab sisendiks ühe värvikanaliga pildi, näiteks halltoonides pildi, rakendab histogrammi võrdsustamist ning tagastab tulemuseks saadud pildi. Värviliste piltide puhul, millel on mitu värvikanalit, tuleb kanalid enne lahutada ning funktsiooni neile eraldi rakendada. Hiljem saab kanalid tagasi ühendada ning tulemuseks on normaliseeritud värvidega pilt. Sellise töötluse tulemust kujutab joonis 2.



Joonis 2. (a) näitab sisendpilti enne iga R, G, B värvikanali histogrammi võrdsustamist, kus heledad toonid domineerivad, (b) näitab tulemust peale histogrammide võrdsustamist.

Normaliseeritud värvidega pildid on kontrastsemad ning neis ei domineeri kindlad toonid, mis on sagedane nähtus profiilipiltidel, kuna valgustus võib pilte tehes erineda märgatavalt. Eeltöötamise tulemusena on erinevate värvide analüüs täpsem ning ei sõltu niivõrd suurel määral pildi valgustingimustest või kaamerast endast.

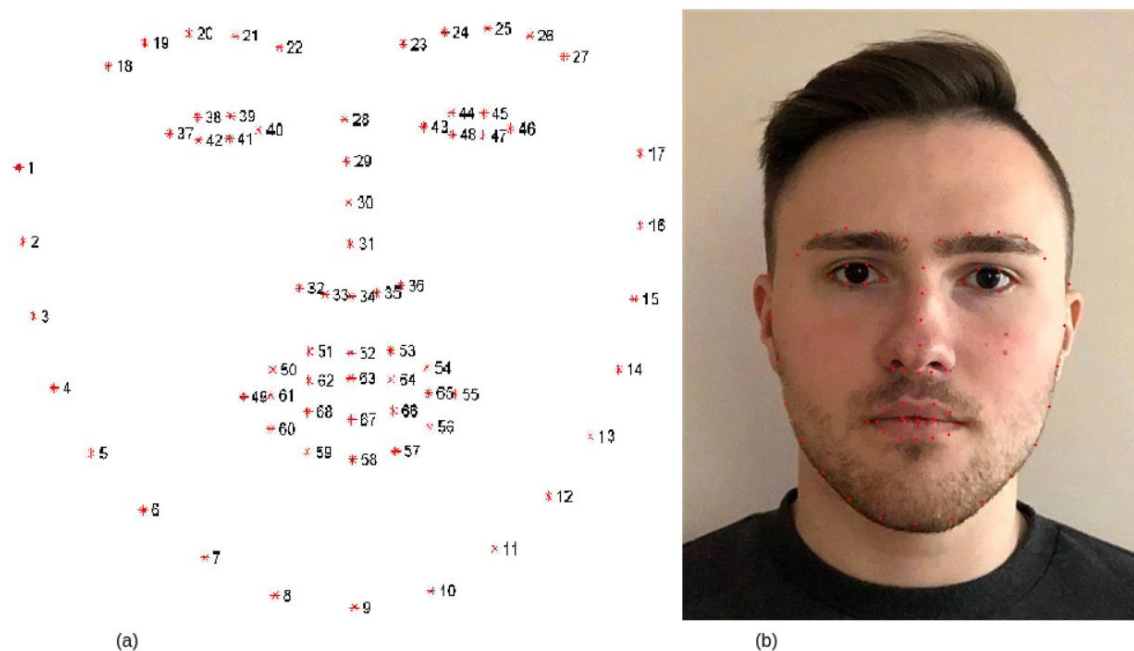
3.2 Kirjelduse olemus

Nägu võib kirjeldada selle üksikuid atribuute nagu silmade värv, nina kuju, juuste olemasolu kategoriseerides. Face Recognition moodul võimaldab funktsiooni *face-landmarks* [32] kasutust, mis leiab sisendiks antud pildist järgnevate näoosade koordinaadid:

- 1) lõug,
- 2) vasak kulm,
- 3) parem kulm,
- 4) ninasild,
- 5) nina ots,
- 6) vasak silm,
- 7) parem silm,

- 8) ülemine huul,
- 9) alumine huul.

Funktsiooni implementeerimiseks kasutab Face Recognition süvaõppe mudeleid moodulist Dlib [33]. Leitavate koordinaatide paiknemist näol ja päris sisendpildil demonstreerib joonis 1.



Joonis 3. (a) Dlib mudelitega leitud koordinaadid [34], (b) leitud koordinaadid punaste täppidena sisendpildil.

Erinevate näoosade või atribuutide kirjeldused annavad eri koguses väärtust lõplikule kirjeldusele. Näiteks annab juuste olemasolu kasutajale vaieldamatult rohkem informatsiooni kui huulte kuju. Kirjeldatavate atribuutide hulka on vaja ka piirata, et vältida info üleküllust. Arvestades Face Recognition ja OpenCV moodulite piiranguid ning autori subjektiivset hinnangut selle kohta, milliseid atribuute tasub kirjeldada, said välja valitud järgnevad:

- 1) üldised atribuudid
 - a. naha värv
 - b. näo kuju
- 2) juuksed
 - a. olemasolu
 - b. värv
- 3) näokarvad

- a. tihedus
- b. värv
- 4) silmad
 - a. värv
 - b. kuju
 - c. asetus
- 5) nina kuju.

Kirjeldusest jäävad välja lõua, huulte ning kulmude analüüs. Kui ei ole tegemist väga eripäraste omadustega, siis ei anna autori hinnangul nende näoosade kategoriseerimine võrreldes teistega oluliselt informatsiooni juurde. Iga atribuudi puhul teeb rakendus spetsialiseeritud analüüsi ja kategoriseerib need eeldefineeritud klasside põhjal.

3.3 Üldised atribuudid

Üldisteks atribuutideks inimese näo puhul on peetud kõige algelisemaid ja esimesena märgatavaid omadusi ehk selle värv ja kuju. Antud omaduste kategoriseerimine annab põhja lõplikule kirjeldusele.

3.3.1 Naha värv

Näokirjelduse protsessi üheks esimeseks sammuks on naha värvi leidmine ja kategoriseerimine, kuna tulemus on vajalik juuste ja näokarvade analüüsimiseks, mida kirjeldatakse hilisemates peatükkides.

Esmalt lõigatakse pildist välja ülemine osa põsest, kasutades Face Recognition mooduli abil saadud x-koordinaate punktidest 10 kuni 13 ja y-koordinaate punktidest 1 kuni 35 (vt joonis 1.a), kuna seal on minimaalne värvide varieeruvus võrreldes naha värviga. See tähendab, et üldiselt pole seal näokarvu või varje teistest näo osadest, mistõttu on sealt leitud värv võimalikult lähedane naha värvile. Lõigatud pilt läbitakse pikslihaaval ning leitakse iga R, G, B värvikanali aritmeetiline keskmine väärtus. Tulemust arvestatakse kui naha värvi RGB väärtust.

Fitzpatricki klassifikatsioonisüsteem on üldlevinud meetod hindamaks naha pigmentatsiooni taset, koosnedes kuuest klassist [35], mida võib kirjeldada järgnevalt:

- 1) kahvatu valge (ingl *pale white*),
- 2) hele (ingl *fair*),
- 3) tumedam valge (ingl *darker white*),

- 4) helepruun (ingl *light brown*),
- 5) pruun (ingl *brown*),
- 6) tumepruun või must (ingl *dark brown or black*).

Näidiseid Fitzpatricki fototüüpidest on näha joonisel 4.



Joonis 4. Näited kuuest Fitzpatricki fototüübist [36].

RGB väärtuste põhjal on võimalik jagada inimese nahavärvi toonid 18 kategooriasse [37], mida on näha joonisel 5.

45 34 30	#2D221E	
60 46 40	#3C2E28	
75 57 50	#4B3932	
90 69 60	#5A453C	
105 80 70	#695046	
120 92 80	#785C50	
135 103 90	#87675A	
150 114 100	#967264	
165 126 110	#A57E6E	
180 138 120	#B48A78	
195 149 130	#C39582	
210 161 140	#D2A18C	
225 172 150	#E1AC96	
240 184 160	#F0B8A0	
255 195 170	#FFC3AA	
255 206 180	#FFCEB4	
255 218 190	#FFDABE	
255 229 200	#FFE5C8	

Joonis 5. Naha värvi 18 kategooriat RGB väärtustega [37].

Joonisel 5 illustreeritud 18 nahavärvi kategooriat on jagatud Fitzpatricki kuue fototüübi vahel nii, et tumedaimast heledaimani järjestuse alusel vastab igale fototübile järgnevast kolmest kategooriast keskmine. Selle tulemuseks on klassifitseerimistabel, mis seab vastavusse igale Fitzpatricki fototübile kindla RGB väärtuse. Tulemusi on näha tabelis 1.

Tabel 1. Naha värvi klassifitseerimistabel RGB väärtustega.

Fitzpatricki fototüüp	Lihtkeelne nimetus	RGB väärtus
Fototüüp I	Kahvatu valge (ingl <i>pale white</i>)	(255, 218, 190)
Fototüüp II	Hele (ingl <i>fair</i>)	(240, 184, 160)
Fototüüp III	Tumedam valge (ingl <i>darker white</i>)	(195, 149, 130)
Fototüüp IV	Helepruun (ingl <i>light brown</i>)	(150, 114, 100)
Fototüüp V	Pruun (ingl <i>brown</i>)	(105, 80, 70)
Fototüüp VI	Tumepruun või must (ingl <i>dark brown or black</i>)	(60, 46, 40)

Naha värvi kategoriseerimiseks võrreldakse naha värvi RGB väärtust erinevate naha värvi kategooriate RGB väärtuste vastu. Iga kategooria puhul leitakse naha värvi kaugus ehk erinevus antud kategooriast ning väikseima erinevuse põhjal tehakse naha värvi kategoriseerimine. Valemina kasutatakse Eukleidilist kaugust kolmes mõõtmes iga värvikanali jaoks (vt valem 1).

$$d(A, B) = \sqrt{(R_A - R_B)^2 + (G_A - G_B)^2 + (B_A - B_B)^2} \quad (1)$$

3.3.2 Näo kuju

Teiseks oluliseks markeriks näo juures, mis mängib tähtsat rolli kogu näo välimuses, on näo kuju. Sõltumata geograafilisest asukohast, vanusest või soost on olemas 5 üldtunnustatud näokuju:

- 1) teemant (ingl *diamond*),
- 2) oblong (ingl *oblong*),
- 3) ovaal (ingl *oval*),
- 4) ümmargune (ingl *round*),
- 5) ruut (ingl *square*) [38].

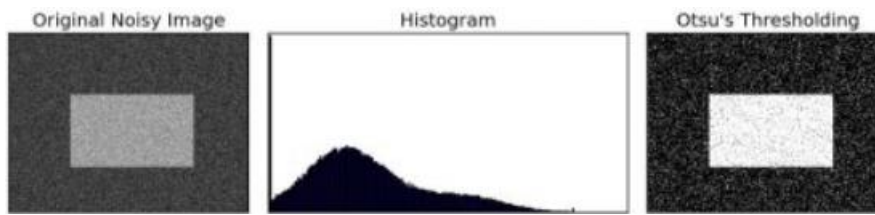
Erinevaid näokujusid illustreerib joonis 6.



Joonis 6. 5 üldtunnustatud näo kuju, vasakult paremale vaadates: teemant, ovaal, oblong, ümmargune, ruut [38].

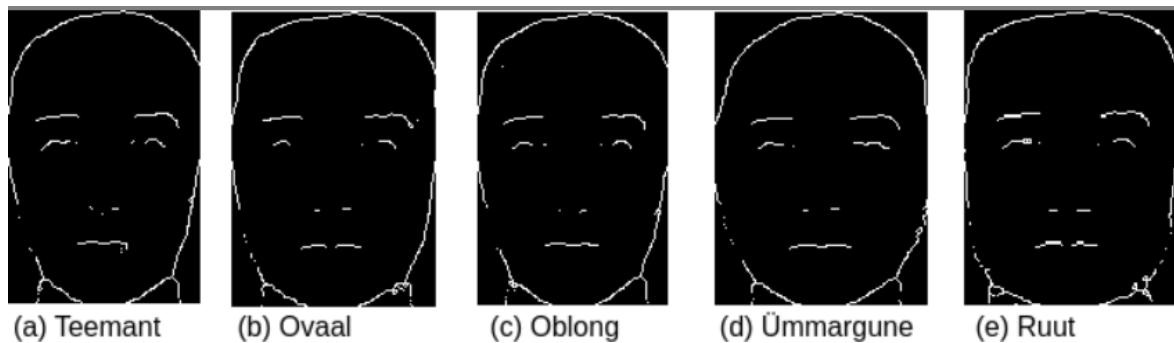
Näo kuju kirjeldamine on loomult mallide sobitamine ja täpseima vaste põhjal kuju klassifitseerimine. Kuna antud sektsioonis on oluline vaid näokuju ehk näo väline kontuur, ei ole hea idee lihtsalt joonisel 6 olevaid pilte sisendpildile sobitada. Sellisel juhul mõjutaksid tulemusi värvid ning erinevad näoomadused.

Värvide probleemi saab lahendada lihtsalt pildi halltoonidesse teisendades. Siiski jäävad peale teisendust mallide sisse erinevused näoomadustes, mis võivad tulemusi mõjutada. Lävendamine Otsu binariseerimisega (ingl *Thresholding with Otsu's Binarization*) on meetod, kus halltoonides pildi pikslid muudetakse mustaks või valgeks (0 või 255) sõltuvalt sellest, kas piksel ületab etteantud lävendit. Selleks lävendiks võetakse pildi histogrammilt leitud optimaalne punkt vähima varieeruvusega kummaski suunas. OpenCV moodulis on algoritm realiseeritud funktsiooniga *threshold*, mille rakendamist on näha joonisel 7 [39].



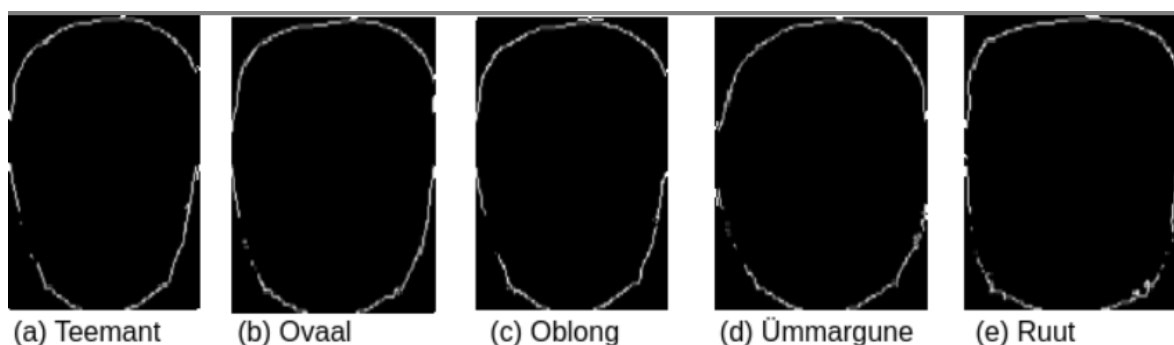
Joonis 7. Lävendamine Otsu binariseerimisega [39]. Tulemuseks on müra vähenemine ja tugev kontuur.

Hõrendamine (ingl *thinning*), teisisõnu skeletoniseerimine (ingl *skeletonization*), on pildi eeltöötlusmeetod, mis kustutab iteratiivselt kontuuride piksleid, kuni alles jääb nii-öelda pildi skelett [40]. OpenCV moodul pakub funktsiooni *ximgproc.thinning* [41], mis võtab eelneva lävendamismeetodi tulemusena saadud binaarse pildi ja rakendab hõrendamist, tagastades pildi skeleti. Algoritmi rakendamist näokuju mallidele illustreerib joonis 8.



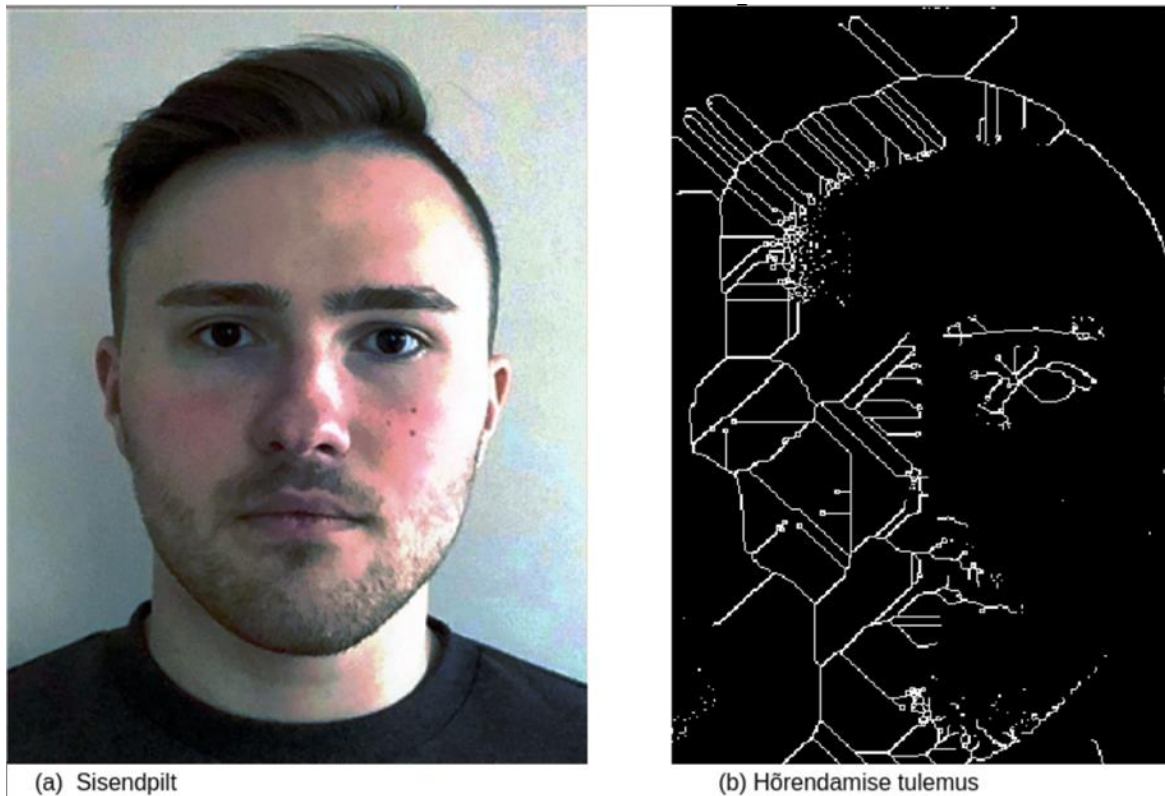
Joonis 8. Näo kuju mallid pärast hõrendamist.

Peale hõrendamist sisaldavad mallid siiski liiga palju detaile. Näokuju määramisel on tähtis vaid pea väliskontuur, aga alles on jäänud ka üleliigseid kontuure nagu kulmud ja silmad. Ebavajalikke detailide eemaldamiseks rakendati käsitööd, mille tulemusi on näha joonisel 9.



Joonis 9. Lõplikud näo kuju mallid.

Sisendpildi näo kuju klassifitseerimiseks on parima täpsuse nimel hea idee sisendpilti sarnaselt mallidele töödelda enne võrdluste tegemist. Kuna aga sisendpildi kohta rakendus eelnevalt midagi ei tea, on välistatud peenhäälestamise võimalus. Kuna kõige tugevam kontuur pildis ei ole ilmtingimata näokontuur, osutus hõrendamine halvaks meetodiks. Mallidega sarnase lävendamise ja hõrendamise tulemust on näha joonisel 10.



Joonis 10. Hõrendamise tulemus sisendpildile rakendades.

Alternatiiviks sai valitud eelmainitud Canny servatuvastusalgoritm, mis on OpenCV moodulis implementeeritud funktsiooniga *Canny* [42]. Tulemuseks saadud pildis on üldjuhul palju rohkem müra, kuid säilib näo väline kontuur, mis on vajalik näo kuju võrdluseks. Canny algoritmi rakendamist illustreerib joonis 11.



Joonis 11. Canny servatuvastusalgoritmi tulemus.

Näo kuju klassifitseerimiseks kasutatakse OpenCV mooduli funktsiooni *matchTemplate*. Funktsioon on implementatsioon mallipõhise sobitamise algoritmist, kus malli võrreldakse nii-öelda „liikuva aknana“ erinevate osadega sisendpildist [43]. Rakendus võrdleb igat näo kuju malli sisendpildiga ning määrab näokuju vastavalt kõige suurema täpsusega mallile.

3.4 Juuksed

Järgmine esiletähtsate atribuut inimese näo juures on tema juuksed. Sisendpildilt leitud näo kirjeldusele annab palju juurde määramine, kas inimesel on juukseid ja mis värvi need on.

3.4.1 Juuste olemasolu

Juuste olemasolu määramisel mängib olulist rolli jälle Canny servatuvastusalgoritm. Sarnaselt näo kuju kirjeldamisele kasutatakse OpenCV mooduli implementatsiooni algoritmist näos leiduvate servade tuvastamiseks. Seejärel kasutatakse Face Recognition moodulist saadud kulmu kõige ülemist y-koordinaati punktis 19 lõikamaks pildist välja lauba koos juustega.

Tulemusele rakendatakse morfoloogilist sulgemist (ingl *morphological closing*), mis tähendab lünkade täitmist kujude vahel [44]. Eesmärk on täita väiksed vahed leitud servade

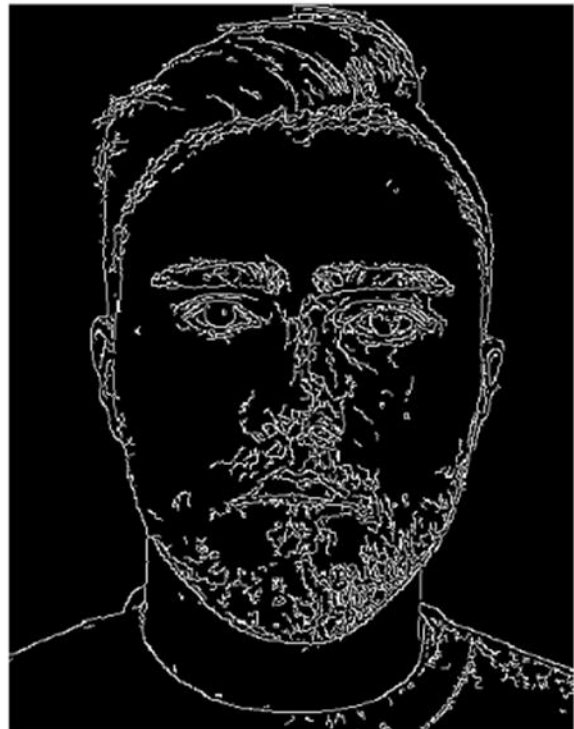
vahel, luues tugeva ühendatud kontuuri juuste ja lauba vahel. Funktsioon *morphologyEx* [45] pakub OpenCV moodulis implementatsiooni antud algoritmist. Sisendiks võetakse pildi ning struktuurielemendi suuruse. Struktuurielemendi kasutatakse iga piksli puhul vaadeldava raadiusena, kus pikslile piisavalt lähedal olevad pikslid (jäävad struktuurielemendi sisse) ühendatakse. Selle tulemusena leitakse ning ühendatakse pikslid, mis on osa mingist lüngast. Erinevate sisendpiltide peal katsetamisega osutus üldiselt parimaks suuruseks olema 7x7 pikslit. Väiksemate struktuurielementidega ei suletud kõiki lünki kontuuride vahel ning suurematega värviti ära pildi osi, mis pole kontuurid.

Järgmise sammuna täidetakse kontuuride sisene ala valge värviga. Eesmärk on saavutada mustvalge pilt, kus valged pikslid vastavad originaalses pildis kas laubal või juustes olevatele pikslitele ning mustad pikslid vastavad taustas olevatele pikslitele. Eesmärgi saavutamise edu sõltub tugevalt sisendpildi servade tugevusest juuste, lauba ja tausta vahel. Esialgu kasutatakse OpenCV mooduli funktsiooni *findContours* [46], mis leiab binaarselt pildilt kontuurid ning tagastab nende vektorid. Välist vektorit ehk kõige välimist leitud kontuuri saab kasutada funktsioonis *fillConvexPoly* [47], et kogu välimisest kontuurist sissejääv ala valgeks värvida. Tulemus on aktsepteeritav ka ebaoptimaalsemate piltide puhul, kus servad pole niivõrd ilmekad.

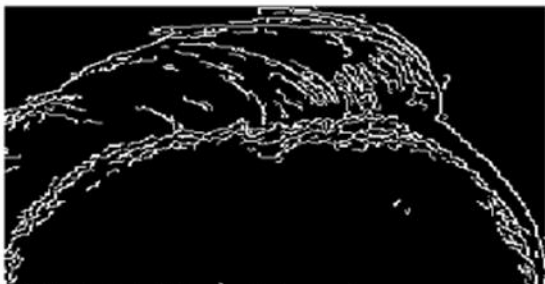
Pilditöötamise sammude rakendamist on näha joonisel 12.



(a) Sisendpilt



(b) Tuvastatud servad



(c) Lõigatud pilt laubast ja juustest



(d) Morfoloogiline sulgemine



(e) Kontuuride tuvastus ja välise kontuuri sisese ala värvimine



(f) Sama seksioon originaalpildist välja lõigatuna

Joonis 12. Pilditötluse sammud leidmaks ning eraldamaks juuksed ja lauba taustast.

Edasi vaadeldakse kõrvuti lauba ja juuste pilti originaalsest pildist (joonis 12.f) ning sama pilti binaarsena ehk nii-öelda maski (joonis 12.e). Ükshaaval piksleid originaalpildis läbides kogutakse eraldi kokku juuste pikslid ja lauba pikslid. Piksleid, millele vastav piksel on maskis must, ignoreeritakse, kuna tegemist on taustaga. Valgete pikslite puhul on tegemist laubaga juhul, kui piksli erinevus naha värvist jääb alla lävendi, vastasel juhul on tegemist

juuste piksliga. Testpiltide manuaalse analüüsi tulemusena (juuste ning lauba eraldamine, pikslite kokkulugemine koos suhtarvu leidmisega, mille puhul rakendus andis võimalikult täpse suhte reaalsusele) osutus keskmiseks lävendiks 0.495. Piksli erinevust naha värvist arvestatakse kui värvikanalite keskmist kauguse ja nahavärvi väärtuse suhet (vt valem 2).

$$\text{erinevus} = \frac{\frac{\sqrt{(R_{\text{piksel}} - R_{\text{nahavärv}})^2}}{R_{\text{nahavärv}}} + \frac{\sqrt{(G_{\text{piksel}} - G_{\text{nahavärv}})^2}}{G_{\text{nahavärv}}} + \frac{\sqrt{(B_{\text{piksel}} - B_{\text{nahavärv}})^2}}{B_{\text{nahavärv}}}}{3} \quad (2)$$

Arvutuste tulemusena on pikslid jagunenud kahte kategooriasse: juuksed ja laup. Hindamaks, kas inimesel on juuksed olemas, peab juuste pikslite arvu ja juuste ning lauba pikslite arvu summa suhe ületama kindlat lävendit. Leidmaks selle lävendi, rakendati eelnevat pilditöötlust ning arvutuskäiku viie erineva eksemplari peal inimestest eri juuste kogusega, mida on näha joonisel 13.



Joonis 13. Eksemplarid inimestest erinevate juuste kogusega. Vasakult paremale vaadates: (a) kiilakas inimene [48], (b) kiilanev inimene [49], (c) hõrenevate juustega inimene [50], (d) väga lühikeste juustega inimene [51], (e) lühikeste juustega inimene [52].

Eksemplaride juuste pikslite arvu suhe lauba ja juuste piksite arvu summaga on kirjeldatud tabelis 2.

Tabel 2. Eksemplaride juuste pikslite arvu suhe lauba ja juuste piksite arvu summaga.

Eksemplar	Suhe
Kiilakas	0.0
Kiilanev	0.07960533786426549
Hõrenev	0.22172339855266684

Väga lühikesed juuksed	0.79091021365938
Lühikesed juuksed	0.7717787682333873

Selleks, et määrata juuste olemasolu, peab leitud suhe ületama kindla lävendi. Autorile teadaolevalt ei leidu allikaid, mis sarnase lävendi defineeriksid. Seetõttu sai lävendiks valitud autori subjektiivsel hinnangul eksemplaride tulemusi analüüsides 0.1. See suhe jääb kiilaneva ja hõrenea eksemplari tulemuste vahele, aga on väga tugevalt kallutatud kiilaneva eksemplari tulemuse poole.

3.4.2 Juuste värv

Eelnevas peatükis on kirjeldatud, kuidas määrata juuste olemasolu. Kui juuksed on tuvastatud, siis järgmine samm on nende värv kategoriseerida. Selleks kasutatakse ära eelnevas sammus leitud juuste pikslid, mille aritmeetilise keskmise RGB väärtuse leitakse need pikslid ükshaaval läbides.

Võimalikud juuksevärvid saab jagada 6 kategooriasse: punane, blond, helepruun, tumepruun, must, muu [53]. Kui välja jätta kunstlikud juuksevärvid ning nimekirja lihtsustada, saab juukseid kategoriseerida järgnevalt:

- 1) punased (ingl *red*),
- 2) blondid (ingl *blonde*),
- 3) pruunid (ingl *brown*),
- 4) mustad (ingl *black*),
- 5) hallid (ingl *grey*).

Eksemplare juuste värvidest on näha joonisel 14.



Joonis 14. Välja lõigatud pildid erinevatest juuste värvidest. Vasakult paremale vaadates: (a) punased, (b) blondid, (c) pruunid, (d) mustad [54], (e) hallid [55].

Iga eksemplari puhul leiti aritmeetiline keskmine RGB väärtus sarnaselt sisendpildile, tulemused on kirjeldatud tabelis 3.

Tabel 3. Juuste värvide eksemplaride keskmised RGB väärtused.

Juuste värv	RGB väärtus
Punane	(161, 104, 80)
Blond	(145, 116, 91)
Pruun	(87, 74, 64)
Must	(46, 52, 51)
Hall	(186, 182, 176)

Kategoriseerimaks juuste värvi, rakendatakse iga kategooria puhul Eukleidilise kauguse valemit (vt valem 1). Lühima kauguse põhjal määratakse sisendpildi ennustatud juuksevärv.

3.5 Näokarvad

Sarnaselt juustele on oluline atribuut inimese näo juures näokarvad. Nende olemasolu, tihedus ja värv mõjutavad tugevalt näo üldist väljanägemist. Koos eelnevalt kirjeldatud omadustega annab näokarvade kirjeldus juba lugejale piisavalt infot, et tekiks mingisugune ettekujutus kirjeldatavast näost.

3.5.1 Näokarvade tihedus

Võrreldes juuste asukoha määramisega on näokarvade määramine tunduvalt hõlpsam, kuna erinevalt juustest jääb lõua, põskede ja ülahuule ala Face Recognition mooduli poolt leitud kordinaatpunktide sisse. Mõne väga ebaoptimaalse pildi puhul on võimalik, et juuste ja lauba asukoht määratakse valesti. Face Recognitioni treenitud mudelid on aga palju usaldusväärsemad. Seetõttu saab olla palju kindlam vaadeldava ala õigsuses ning määrata lisaks näokarvade olemasolule ka nende tihedus. Lisaks pole tarvis teha läbi kulukat pilditöötlust näokarvade määramiseks.

Sisendpildist lõigatakse välja 4 alampilti:

- 1) Lõug (x-koordinaadid punktidest 7 ja 9 ning y-koordinaadid punktidest 5 ja 8),

- 2) vasak põsk (x-koordinaadid punktidest 10 ja 12 ning y-koordinaadid punktidest 57 ja 12),
- 3) parem põsk (x-koordinaadid punktidest 5 ja 7 ning y-koordinaadid punktidest 57 ja 12),
- 4) nina ja ülemise huule vaheline osa (x-koordinaadid punktidest 7 ja 9 ning y-koordinaadid punktidest 33 ja 50).

Neid seksioone peetakse näoosadeks, kus näokarvu peaks mingil määral leiduma, juhul kui neid üldse on. Iga alampildi puhul läbitakse pikslid ükshaaval ja kategoriseeritakse näokarvade piksliteks juhul, kui piksli erinevus naha värvi RGB väärtusest ületab kindlat lävendit. Erinevuse arvutamiseks on kasutatud valemit 2. Erinevuse lävendiks sai valitud 0.25 peale katse-eksituse meetodi rakendamist erinevate näokarvade tihedustega eksemplaride peal, tehes läbi kirjeldatud arvutuskäigu ning analüüsides tulemuseks saadud näokarvapikslite osakaalu kõikide pikslite seas.

Peale piltides olevate pikslite jagamist kahte kategooriasse, näokarvade pikslid ja muud pikslid, on võimalik arvutada näokarvade osakaal. Selle põhjal saab võrdluse teel kategoriseerida näokarvade olemasolu ja tiheduse. Kategooriate piirväärtused said valitud sarnaselt eelnevas paragrahvis kirjeldatud lävendi leidmisele:

- 1) Näokarvu pole, kui leitud näokarvade pikslite arvu osakaal on vähem kui 75%.
- 2) Leidub õrn (ingl *thin*) näokarv, kui osakaal on vahemikus 75-90%.
- 3) Leidub tihe (ingl *thick*) näokarv, kui osakaal ületab 90%.

Näide analüüsi tulemustest on nähtav joonisel 15.



Joonis 15. Näokarvade osakaal näidispildi ning kasutatud alampiltide puhul.

Näokarvade osakaalu leidmiseks kogutud näokarvade pikslid osutuvad vajalikuks ka näokarvade värvi määramisel.

3.5.2 Näokarvade värv

Sarnaselt juuste värvi leidmisele kasutatakse ära eelnevalt leitud näokarvade piksleid (juuste puhul juuste piksleid), mille ükshaaval läbides leitakse pikslite aritmeetiline keskmine RGB väärtus. Keskmised leitakse samamoodi ka iga näokarva värvi kategooria eksemplaride jaoks. Need kategooriad on samad, mis juuste värvi puhulgi.

Näokarvad ning juuksed on mõlemad fundamentaalselt karvkate ning nende erinevad võimalikud värvikategooriad on samad. Küll aga pisierinevuste tõttu, mis tekivad varjudest ja muudest detailidest, tuleb parema täpsuse nimel arvutada kategooriatele vastavad RGB väärtused uute eksemplaride peal päris näokarvadest. Varem arvutatud juuste värvide väärtuste taaskasutamine mõjutaks liialt kategooriat esindava väärtuse täpsust. Kasutatavad eksemplarid on nähtaval joonisel 16.



Joonis 16. Väljalõigatud pildid näokarvade värvide kategooriate eksemplaridest. Vasakult paremale vaadates (a) punane [56], (b) blond [57], (c) pruun [58], (d) must [59], (e) hall [60].

Eksemplaride aritmeetilised keskmised RGB väärtused arvutati väljalõigatud piltide piksleid ükshaaval läbides ning vastavad tulemused on kirjeldatud tabelis 4.

Tabel 4. Näokarvade värvide eksemplaride keskmised RGB väärtused.

Juuste värv	RGB väärtus
Punane	(147, 87, 66)
Blond	(129, 109, 90)
Pruun	(126, 70, 55)
Must	(24, 26, 27)
Hall	(137, 137, 142)

Iga kategooria RGB väärtuse puhul leitakse Eukleidiline kaugus (vt valem 1) sisendpildilt leitud näokarvade keskmisest RGB väärtusest. Lühima kauguse põhjal kategoriseeritakse värv sisendpildis.

3.6 Silmad

Võrreldes naha värvi, näo kuju, juuste või näokarvakattega on silmad pigem väike detail kogu näo kirjelduses, aga need annavad siiski väärtuslikku infot. Sõltuvalt erinevast silmade kujust, värvist või eriti nende asetusest ehk omavahelisest kaugusest võivad muul moel sarnased näod tunduda märgatavalt erinevad. Järgnev peatükk kirjeldab mainitud kolme atribuudi kategoriseerimist.

3.6.1 Silmade asetus

Üks oluline aspekt silmade juures, mis võib inimeselt inimesele tugevalt varieeruda, on silmade horisontaalne asetus näos. Vertikaalselt asuvad silmad umbes näo keskel, võrdsel kaugusel lõuast ja kolju tipust [61]. Lõua ning silmade vahelise distantsi standardhälve on naiste puhul 5,5% keskmisest (11,34 cm) ja meeste puhul 4,9% keskmisest (12,26 cm). Silmade vaheline distants, teisisõnu silmade horisontaalne asetus üksteise suhtes, on tunduvalt suurema variatsiooniga. Selle standardhälve on naiste puhul 9,3% keskmisest (3,02 cm) ja meeste puhul 11,5% keskmisest (3,04 cm) [62]. Arvestades horisontaalse distantsi väiksust võrreldes vertikaalse distantsiga, on variatsioon keskmisest esimeses võrdlemisi märgatav ja teises mitte.

Silmi võib pidada lähedal asetsevateks (ingl *close-set*) või laialt asetsevateks (ingl *wide-set*) sõltuvalt sellest, kas silmade vaheline distants on vähem kui ühe silma laius või rohkem [63]. Eksemplarid kummagi kategooria esindajatest koos silmade vahelise distantsi ja silmade keskmise laiuse suhtega on nähtaval joonisel 17. Suhte leidmiseks on võrreldud horisontaalselt asetsevate pikslite arvu kummaski silmas nende keskmise leidmiseks ning silmade vahel.



(a) Laialt asetsevad silmad: silmade vaheline distants on ~1.24 korda suurem kui keskmine silma laius



(b) Lähedal asetsevad silmad: silmade vaheline distants on ~0.91 korda väiksem kui keskmine silma laius

Joonis 17. Nädispildid laialt asetsevatest silmadest [64] ja lähedal asetsevatest silmadest [65] koos ligikaudsete silmade vahelise distantsi ja keskmise silma laiuse suhetega.

Sama suhte määramiseks sisendpildis eraldatakse kummagi silma koordinaadid Face Recognition moodulist saadud punktidest 37 kuni 42 (parem silm) ja 43 kuni 48 (vasak silm). Kummagi silma suurimast x-koordinaadist lahutatakse madalaim x-koordinaat ehk

leitakse silma laius pikslites. Vasaku silma madalaimast x-koordinaadist parema silma kõrgeima x-koordinaadi lahutamise tulemusena leitakse silmade vaheline distant pikslites. Selle võrdlemisel silmade laiuse aritmeetilise keskmisega leitakse mainitud suhe. Erinevate testpiltidega lähedal ja kaugel asetsevatest silmadest katsetamisega selgus korduv muster, et Face Recognitioni mooduli punktid silmade ääres (parema silma puhul punktid 37 ja 40, vasaku puhul 43 ja 46) ei asetse täielikult silma ääres. Seetõttu jääb märgatav osa silmast horisontaalselt punktidest väljapoole. Puuduse lahendamiseks korrutatakse leitud suhet konstandiga 0.66, mis leiti testpiltide veafaktori aritmeetilise keskmisena. Veafaktoriks peetakse siinkohal arvutatud tulemuse ning manuaalse analüüsi ehk horisontaalsete pikslijadade pikkuse hindamise tulemuse suhet. Sisendpiltide silmad, mille arvutatud suhe jääb alla 1, kategoriseeritakse lähedal asetsevateks. Lävendit ületavad silmad kategoriseeritakse laialt asetsevateks.

3.6.2 Silmade kuju

Mõnel määral väiksem detail silmade juures on nende kuju. Siiski on inimeste vahel varieeruvust selle osas palju. Lisaks üldisele naturaalsele variatsioonile, mis on omane kõikidele kirjeldatavatele atribuutidele, võib silmade kuju rohket erinevust selgitada geograafilisest asukohast tingitud variatsiooniga valguses [66].

Erinevalt eelnevalt mainitud atribuutidele ei ole autorile teadaolevalt olemas üldist teaduslikku konsensust selle osas, missugused silmade kujud on olemas. Erinevad allikad, peamiselt silmakliinikud, väidavad neid kujusid üldiselt ühtemoodi jagunevaks. Küll aga on kõigis väidetes väikesed erinevused, pakutakse välja eri hulki ning mõnel määral erineva kirjeldusega kategooriaid. Lõputöö jaoks sai valitud alamhulk kategooriaid allikast, mis piiraks kategooriaid mõistlikuks arvuks ning maksimeeriks visuaalset erinevust nende vahel. Silmade kuju saab kategoriseerida järgnevalt:

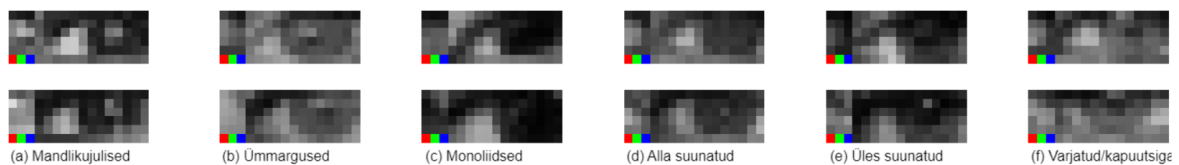
- 1) mandlikujulised (ingl *almond shaped*),
- 2) ümmargused (ingl *round*),
- 3) monoliidsed (ingl *monolid*),
- 4) alla suunatud (ingl *downturned*),
- 5) üles suunatud (ingl *upturned*),
- 6) varjatud/kapuutsiga (ingl *hooded*) [67].

Iga kategooria eksemplarid on nähtaval joonisel 18.



Joonis 18. Eksemplarid erinevatest silmade kujudest. Vasakult paremale vaadates: (a) mandlikujulised, (b) ümmargused, (c) monoliidsed, (d) alla suunatud, (e) üles suunatud, (f) varjatud/kapuutsiga [67].

Sarnaselt näo kuju kategoriseerimisele põhineb silmade kuju määramine erinevate kategooriate mallide sisendpildile sobitamisel, kus kuju kategoriseeritakse kõige täpsema malli põhja. Selleks rakendati Face Recognition moodulit iga kategooria eksemplari peal silmade tuvastamiseks. Seejärel lõigati välja alampilt vasakust silmast punktidega 42 kuni 48 ning paremast silmast punktidega 36 kuni 42. Vältimaks olukorda, et mallide täpsust mõjutavad erinevused värvides, teisendati mallid halltoonidesse. Suur erinevus võrreldes näo kuju määramisega on see, et rohkem morfoloogilise teisendusi pole tarvis. Kõik, mis jääb kummagi silma malli sisse, näiteks silmalaug, on vajalik informatsioon. Lõplikud mallid on nähtaval joonisel 19.



Joonis 19. Silmade kuju mallid. Ülemisel real on vasaku silma, alumisel parema silma mallid.

Sisendpilt läbib samasugust töötlust, nagu sai tehtud mallide loomisel. Pildilt tuvastatakse silmad, kummastki lõigatakse välja alampilt ning tulemused teisendatakse halltoonidesse. Järgnevalt sobitatakse OpenCV mooduli funktsiooniga *matchTemplate* vasaku silma malle sisendpildi vasakule silmale ning parema silma malle paremale silmale. Lihtsustamaks lõplikku kirjeldust ei määrata kummagi silma kuju eraldi. Silmade kuju kategoriseeritakse kõige suurema vasaku ja parema silma malli paari sarnasuste aritmeetilise keskmise põhjal.

3.6.3 Silmade värv

Silmade viimaseks kirjeldatavaks atribuudiks on nende värv. Võimalikud silmade värvid võib jagada kuueks kategooriaks:

- 1) sinine (ingl *blue*),
- 2) pruun (ingl *brown*),
- 3) roheline (ingl *green*),
- 4) hall (ingl *grey*),
- 5) pähkelpruun (ingl *hazel*),
- 6) erandjuhtudel punane (ingl *red*) [68].

Joonisel 20 on näha eksemplare igast kategooriast. Värvide määramiseks on iga eksemplar lõigatud välja minimaalses suuruses riskülikuna, mis sisaldaks kogu silmaläätse. Sisendpilti lõigatakse sarnaselt, kasutades Face Recognition mooduli poolt leitavaid minimaalseid silma ümbritsevaid punkte.



Joonis 20. Silmade värvide eksemplarid. Vasakult paremale vaadates: (a) sinine, (b) pruun, (c) roheline, (d) hall, (e) pähkelpruun, (f) punane [68].

Mainitud eksemplare pikslihaaval läbides leiti vastavad keskmised RGB väärtused, mis on välja toodud tabelis 5.

Tabel 5. Silmade värvide eksemplaride keskmised RGB väärtused.

Silmade värv	RGB väärtus
Sinine	(89, 117, 130)
Pruun	(62, 48, 40)
Roheline	(70, 70, 42)
Hall	(113, 109, 114)

Pähkelpruun	(118, 99, 63)
Punane	(158, 114, 120)

Heterokroomia ehk erinevus silmade värvides on suhteliselt haruldane nähtus, esinemissagedusega umbes 0,256% või 1 inimene 400st [69]. Siiski on erinevus värvis tugevalt märgatav omadus, mistõttu kirjeldatakse kummagi silma värvi eraldi.

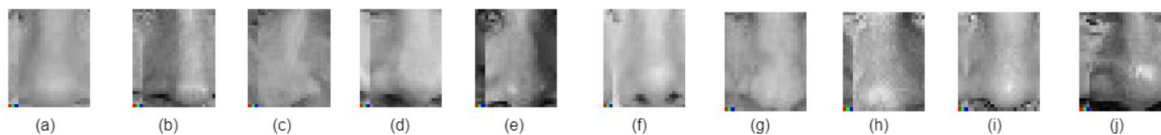
Sisendpildist eraldatakse esialgu vasak silmalääts Face Recognition mooduli punktidega 44 ja 47 ning parem silmalääts punktidega 38 ja 41, sarnaselt eksemplaridega joonisel 20. Kumbki pilt läbitakse pikselhaaval ning leitakse keskmine RGB väärtus, mida võrreldakse iga kategooria RGB väärtuse vastu. Kummagi silma värv määratakse lühima Eukleidilise kauguse (vt valem 1) põhjal.

3.7 Nina kuju

Näo kirjelduse viimaseks atribuudiks on nina kuju. Erinevused selles on, äärmused välja jättes, üldiselt raskemini märgatavad. Sarnaselt silmade kujule ei leidu üldist teaduslikku konsensust erinevate kujude arvus ja jaotuses. Seetõttu valiti kategooriateks alamhulk kategooriaid allikast sedasi, et erinevus kategooriate vahel oleks maksimaalne. Võimalikke nina kujusid on kümme:

- 1) Kreeka ehk sirge nina (ingl *Greek/straight nose*).
- 2) Rooma ehk küürus nina (ingl *Roman/aquiline nose*).
- 3) Kulli/kongus nina (ingl *hawk nose*).
- 4) Nööpnina (ingl *button/snub nose*).
- 5) Nuubia ehk aafrika nina (ingl *Nubian/African nose*).
- 6) Ülespööratud nina (ingl *upturned nose*).
- 7) Kõver nina (ingl *crooked nose*).
- 8) Sibulakujuline nina (ingl *bulbous nose*).
- 9) Lihav nina (ingl *fleshy nose*).
- 10) Ida-aasia nina (ingl *East Asian nose*) [70].

Joonisel 21 on kujutatud eksemplare igast kategooriast. Need on originaalpiltidest välja lõigatud kasutades Face Recognition mooduli punkte 39 ja 42 x-koordinaatide leidmiseks ning punkte 27 ja 34 y-koordinaatide leidmiseks. Lisaks on need halltoonidesse teisendatud.



Joonis 21. Nina kuju mallid. Vasakult paremale vaadates: (a) kreeka [71], (b) rooma [72], (c) kulli/kongus [73], (d) nööp- [74], (e) nuubia [75], (f) ülespööratud [76], (g) kõver [77], (h) sibulakujuline [78], (i) lihav [79], (j) ida-aasia [80] nina.

Joonise 21 pilte kasutatakse kui nina kuju malle. Iga sisendpilt läbib samasuguse töötlust, nagu sai mallide loomiseks tehtud. Seejärel sobitatakse malle ükshaaval sisendpildi nina peale OpenCV mooduli funktsiooniga *matchTemplate*. Kõige täpsema tulemuse saavutanud malli põhjal kategoriseeritakse sisendpildi nina kuju.

Erinevalt näiteks näo kujust pole nina kuju võrdlemisel rohkem pilditöötlust vaja. Nina eraldamisel halltoonides pildina jääb kogu vajalik info pildi sisse ning liigseid detaile, mis mallide täpsust negatiivselt mõjutaks, on minimaalselt. Peamine faktor, mis mallide täpsust ebavõrdselt mõjutaks, on pildi värvid. Need on aga eemaldatud halltoonidesse teisendamise kaudu.

3.8 Valminud andmestik

Pärast iga atribuudi töötlemist on tehtud otsus, millisesse kategooriasse see atribuut kuulub. Valminud andmestik koosneb atribuutide kategooriatest, mis on tähistatud kindlate märksõnadena. Sõltuvalt kasutaja valikust on need märksõnad kas eesti või inglise keeles. Andmestikku kasutatakse hiljem rakenduse eesliideses, et nendest loomulikus keeles kirjelduse koostada. Kirjeldusprogrammi väljundiks ongi see andmestik, jaotatud üldisematesse gruppidesse JSON formaadis. Joonis 22 kujutab näidispilti koos programmi väljundiga mõlemas keeles.



(a) Sisendpilt

```
{
  "üldine": {
    "naha_värv": "hele",
    "näo_kuju": "teemant"
  },
  "näokarvad": {
    "tihedus": "õhuke",
    "värv": "pruun"
  },
  "silmad": {
    "asetus": "lähedal asetsevad",
    "kuju": "alla suunatud",
    "värv_parem": "pruun",
    "värv_vasak": "pruun"
  },
  "nina": {
    "kuju": "rooma"
  },
  "juuksed": {
    "värv": "must",
    "on_olemas": "tõene"
  }
}
```

(b) Näokirjeldus eesti keeles

```
{
  "nose": {
    "shape": "Roman"
  },
  "general": {
    "face_shape": "diamond",
    "skin_colour": "fair"
  },
  "hair": {
    "colour": "black",
    "has_hair": "true"
  },
  "eyes": {
    "colour_right": "brown",
    "colour_left": "brown",
    "shape": "downturned",
    "placement": "close-set"
  },
  "facial_hair": {
    "colour": "brown",
    "thickness": "thin"
  }
}
```

(c) Näokirjeldus inglise keeles

Joonis 22. Näokirjeldusprogrammi väljundandmestik eesti ja inglise keeles.

Tagaliidese töö tulemus on kirjeldatud andmestiku genereerimine ja tagastamine. Märksõnadest loomulikus keeles kirjelduse loomise eest vastutab rakenduse eesliides, mida kirjeldatakse täpsemalt peatükis 4.1.2.

3.9 Täpsust mõjutavad faktorid ning vajakajäämised

Järgnev peatükk kirjeldab peamisi faktoreid, mis mõjutavad sisendpildi näo kirjeldamise täpsust. Need faktorid on tingitud just valitud lahendusest iga atribuudi kirjeldamisel. Teistsuguse lähenemisega oleks võimalik sellised puudujäägid lahendada. Lisaks tuuakse välja kindlad vajakajäämised kirjeldatud metoodikas ning pakutakse võimalusi nende lahendamiseks.

Peamine sarnasus enamike atribuutide kategoriseerimise juures on nii-öelda võrdlus esindajatega. Iga kategooria jaoks on valitud üks esindaja, kelle pildi põhjal luuakse mall või leitakse mingisugune värv. Kuna valim on niivõrd väike, pole aga kategooriat esindav väärtus või mall nii täpne, kui saaks olla. Kui suureks probleemiks see on, sõltub atribuudist ja esindajate valikust. Silma värvi puhul on eri kategooriate esindajad ning arvutatud tulemused tugevalt eristatavad, aga näiteks nina kuju puhul on erinevused peenemad. Sellest võib järeldada, et nina kuju kategoriseerimise täpsus on valimi suuruselt rohkem mõjutatud kui silma värvi oma. Probleemi lahendamiseks aitaks valimi suurendamine realistlikes piirides ning kategooriat esindava väärtus või malli genereerimine nende keskmise põhjal.

Teine silmatorkav puudujääk on rakenduse suur arvutusnõudlikkus, mis on tingitud ebaoptimaalsest lahendusest. Mitmes etapis läbitakse suuri pikslimassiive elemendihaaval

läbi, leidmaks mõnd keskmist väärtust või liigitamaks piksleid. Neid analüüse saaks tegelikult teha ühekordselt, läbides korraga vajalikud osad pildist. Sarnane lähenemine teeks aga rakenduse ülesehituse ebaselgemaks. Lisaks saaks vaadelda korraga mitte üksikut pikslit, vaid kogumikku lähedal asetsevaid piksleid. Võib näiteks teha oletuse, et üksik piksel on väga sarnane seda ümbritsevate pikslitega. See tähendab, et kõiki üheksat polegi vaja ükshaaval vaadelda, vaid saab kasutada keskmist pikslit kui selle grupi esindajat. Sarnane lähenemine vähendaks mitmekordselt töötlushaigega mainitud etappides, kuid mõjutaks kirjelduse täpsust märgatavalt.

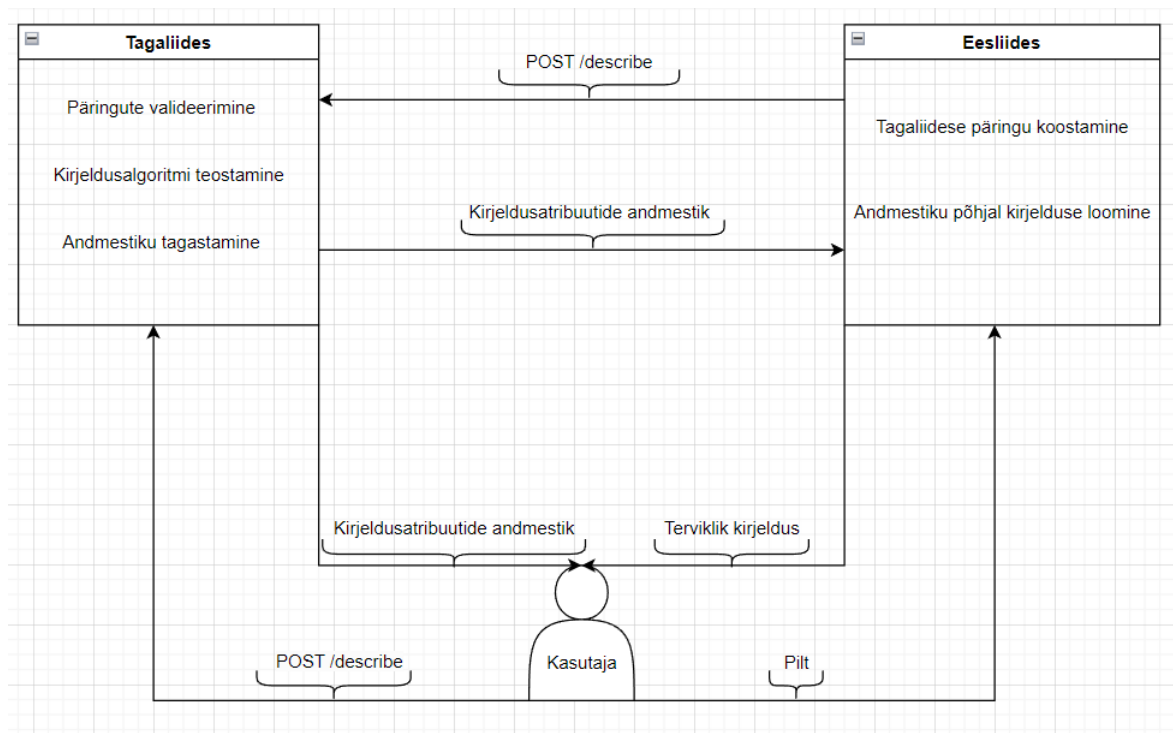
Kolmandaks suureks probleemiks on ebatavapärasused sisendpiltides. Valminud algoritm ei arvesta paljude erinevate võimalike kõrvalekalletega näo atribuutides, mis päris maailmas esinevad suhteliselt tihti. Atribuutide puhul, kus klassifikatsioon tehakse värvide võrdlemise alusel, on selliseks kõrvalekaldeks ebaühtlane või etteaimamatu pigmentatsioon. Näiteks olenevalt silma sisenevast valgusest võib kahel erineval sisendpildil samast inimesest olla erinev silmavärvi klassifikatsioon. Põhjuseks on erinevus pupilli suurusel, mis mõjutab silma keskmist RGB väärtust. Tedretähnid, laikpigmentatsioon ning muud nähtused võivad sarnasel moel mõjutada arvutatud nahavärvi. Lisaks võib esineda probleeme asjaolust, et värvi arvutamiseks ei vaadelda iga kord maksimaalset võimalikku ala. Näiteks vaadeldakse pikemate juustega inimestel nendest ainult lauba osa. Juhul kui inimese juuste juured on ülejäänud juustest erinevat värvi, võib leitud värv olla vale.

4 Veebirakendus

Järgnev peatükk kirjeldab valminud veebirakenduse arhitektuuri ning funktsioone. Lisaks tuuakse välja rakenduse lahendusest tingitud asjakohased detailid seoses rakenduse turvalisuse ja privaatsusega. Rakenduse „Facedescriber“ lähtekood on avalikult kättesaadav GitHubi repositooriumist¹ ning on majutatud samanimelises domeenis².

4.1 Arhitektuur

Rakenduse tagaliides ja eesliides on jaotatud kaheks eraldi teenuseks. Tagaliides vastutab sissetuleva kirjelduspäringu valideerimise ning kirjelduse teostamise eest. Eesliides vastutab veebilehe serveerimise, tagaliidese päringu koostamise ning loomulikus keeles kirjelduse koostamise eest. Joonis 23 annab ülevaate süsteemist ning suhtlusest liidest vahel.



Joonis 23. Ülevaade rakenduse arhitektuurist.

Rakenduse tagaliides on avalik, mis tähendab, et kasutajal on võimalus vältida eesliidese kasutamist ja suhelda otse tagaliideselega. Teenuste eraldatus võimaldab ka ainult tagaliidese kasutamist.

¹ <https://github.com/KarlMagnusLaikoja/face-describer>

² <https://facedescriber.com>

4.1.1 Tagaliides

Rakenduse tagaliidese ülesandeks on oodata sissetulevaid kirjelduspäringuid, valideerida sisendid, rakendada kirjeldusalgoritmi ning tagastada kasutajale vastus. Suhtlusprotokolliks kasutatakse HTTPSi, kasutades sertifitseerimisasutuse Let's Encrypt [81] poolt allkirjastatud ja väljastatud sertifikaate. Let's Encrypti allkirja usaldavad enamused modernseid veebibrausereid.

Tagaliidese server on konfigureeritud käivitama pordil 8080 ning kuulama otspunkti (ingl *endpoint*) „/describe“ sissetulevaid POST päringuid *application/json* sisutüübiga. Sissetulnud päringu validatsioon koosneb kahest etapist. Esmalt valideeritakse päringu sisu JSON skeemi vastu, mis nõuab kahte parameetrit. Esimene on sõne „image“ ehk sisendpilt Base64 kodeeringus. Teine on sõne „language“ võimalike väärtustega „EN“ ja „EE“ ehk soovitud väljundkeel. Skeemi on kujutatud joonisel 24.

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Description Request",
  "description": "A request to apply the face-describer algorithm against a supplied image",
  "type": "object",

  "properties": {

    "image": {
      "description": "The image in base64 encoding",
      "type": "string"
    },
    "language": {
      "description": "The language that the algorithm creates it's output in",
      "type": "string",
      "enum": ["EN", "EE"]
    }
  },

  "required": ["image", "language"]
}
```

Joonis 24. Päringu valideerimiseks kasutatav JSON skeem.

Teiseks etapiks on parameetri „image“ valideerimine. Kasutades regulaaravaldist „ $^([A-Za-z0-9+]{4})*([A-Za-z0-9+]{4}|[A-Za-z0-9+]{3}=[A-Za-z0-9+]{2}==)$$ “ [82] valideeritakse, et parameetri väärtus vastab Base64 kodeeringule.

Pärast sisendite valideerimist dekodeeridakse pilt ning salvestatakse faili, kuna kodeeritud sisend on käsurea jaoks liiga suur. Operatsioonisüsteemi käsuga käivitatakse Pythoni programm, andes sisendiks keele ning pildifaili nime. Pythoni programm rakendab sisendpildile implementeeritud kirjeldusalgoritmi, mille väljundiks on eesti- või ingliskeelne andmestik JSON kujul (joonis 22) või vea korral veakirjeldus. Viimaks kustutakse pildifail ning saadetakse päringu vastus koos andmestikuga või veakoodi ning -kirjeldusega.

4.1.2 Eesliides

Rakenduse eesliidese ülesandeks on kasutajale veebileht serveerida, mis koostab kasutaja sisestatud pildist ning valitud keelest tagaliidese päringu ja loob vastuse põhjal loomulikus keeles kirjelduse. Protokoliks kliendiga ning tagaliidesega suhtlemisel kasutatakse HTTPSi.

Veebilehte serveerib Webpacki server, mis käitub proksina kasutaja ja tagaliidese vahel. Serveeritud veebileht koostab kasutaja sisestatud pildist ja valitud keelest tagaliidese päringu ning edastab selle eesliidese serverile. Server edastab päringu tagaliidesele ning saadud vastuse kliendile tagasi. Lisaks Webpacki serverile on kasutusel ka Node server, mille ainuülesandeks on kõik päringud suunata pordilt 80 pordile 443. Tulemusena on kasutajale peale sunnitud HTTPSi kasutus.

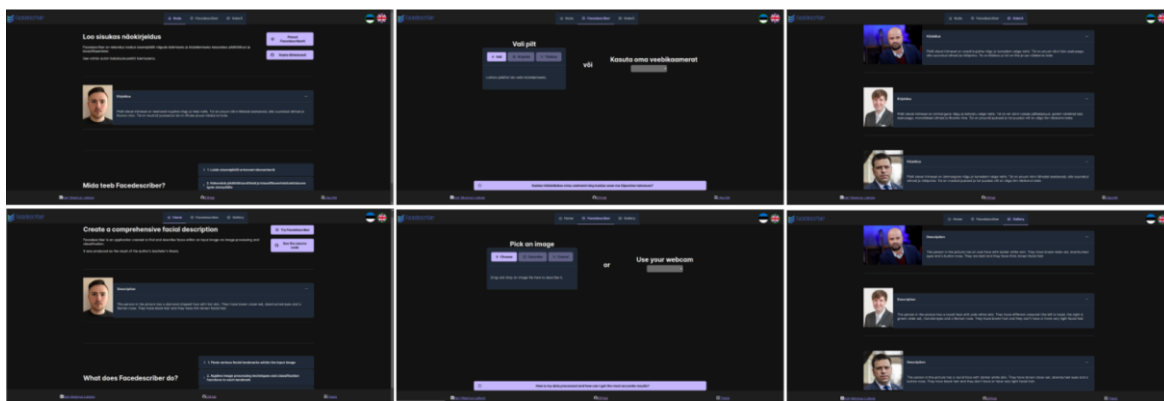
4.2 Loomulikus keeles väljundi koostamine

Päringu vastuse veakoodi põhjal kuvatakse kasutajale kas veakirjeldus või koostatakse saadud märksõnadest loomulikus keeles näokirjeldus. Kõik kirjeldused järgivad ühist malli, kus nii-öelda „lüngad“ asendatakse tagaliidese töö tulemusena saadud märksõnadega. Selle tulemuseks on loomulikus keeles kirjeldus. Sõltuvalt keelest ning erinevatest võimalikest märksõna väärtustest korrigeeritakse kirjelduse ülesehitust, et lõpptulemus oleks keeleliselt korrektne iga võimaliku märksõnade kombinatsiooni puhul. Enamike eestikeelsete atribuutide puhul on vaja teostada vähemalt vormimuudatused, et sõna kasutataks lause sees õiges käandes. Inglisekeelsetele atribuutidele on vaja aga lisada õige artikkel. Kuna atribuutide võimalikud väärtused on teada, pole selleks vaja rakendada morfoloogilist sünteesi, vaid piisab püsikodeeritud (ingl *hardcoded*) vastete kasutamisest. Sarnasel viisil tuleb arvestada võimalike olukordadega, kus andmeid on keskmisest vähem või rohkem. Näiteks kui kirjeldatud näol pole juukseid või näokarvu, siis pole ka nende värve, mistõttu on vajalik vastavat lauset lühendada. Kui aga näiteks silmade värvid on erinevad, siis on

vaja neid mõlemat kirjeldada. Viimaks tuleb arvestada erandlike atribuutidega, mis ei sobi muutmata kujul tekstimalli. Näiteks „nööpnina“ puhul on sõna „nina“ juba osa atribuudist ning tuleb mallist eemaldada, et lõplik lause oleks korrektne.

4.3 Veebilehe funktsioonid

Valminud veebileht on minimaalse funktsionaalsusega, kus peamiseks ülesandeks on pakkuda kasutajale võimalust oma enda üles laetud pilti kirjeldada. Pildi üleslaadimise alternatiiviks on ka veebikaamera kasutus, mille funktsioneerivus sõltub kasutaja operatsioonisüsteemist, näiteks iOS seadmed ei luba kaamera avamist. Lisaks on kuvatud rakenduse demonstreerimiseks mõned näidispildid erinevate näoatribuutidega avaliku elu tegelastest ning töö autorist. Joonis 25 kujutab veebilehte kummaski keeles.



Joonis 25. Veebileht eesti keeles (ülemine rida) ja inglise keeles (alumine rida).

Rakenduse mitmekeelsuse ära kasutamiseks on võimalus veebilehel kuvavat ja tagaliidese päringutes kasutatavat keelt eesti ja inglise keele vahel valida. Paremaks kasutajakogemuseks on ka lühikirjeldus rakendusest ja selle töökäigust. Välja on toodud ka juhend parimate tulemuste saamiseks ning lingid lähtekoodile ja lõputööle.

4.4 Turvalisus ja privaatus

Veebirakenduse disainist tingituna on vajalik mainida võimalike turvariske rakenduses ning kuidas neid on maandatud. Peamine oht, mis tuleneb tagaliidese kasutaja sisendiga operatsioonisüsteemi käsu jooksutamisesest, on käsusüsti (ingl *command injection*) võimalus. Käsusüst on rünne, mis kasutab ära haavatavust kasutajasisendite kasutamisel operatsioonisüsteemi käskude täitmiseks, eesmärgil täita kahjulikke ootamatuid käskude [83]. Ohu maandamiseks on kasutatud tagaliidese päringu „image“ parameetri valideerimiseks kasutatavat regulaaravaldist. Lisaks Base64 kodeeringu formaadi nõudmisele välistab

regulaaravaldis sisendis käsusüsti jaoks kasutatavad sümbolid |, ;, &, \$, >, <, ', ! [84]. JSONi skeema (ingl *schema*) põhine validatsioon keelab käsu süstimise teistesse parameetritesse.

Tagamaks kaitse ettenägemata turvaohutude eest, on rakenduse hostimiseks kasutatud virtuaalmasinat. Virtualisatsioon tagab rakenduse ning kaasnevate turvaohutude isolatsiooni hostmasinast. Lisaks pakub Oracle VM VirtualBox hetktõmmise (ingl *snapshot*) funktsionaalsust, mis võimaldab hõlpsasti taastada virtuaalmasina oleku enne rünnet.

Peale rakenduse turvalisuse on asjakohane kirjeldada rakenduse kasutaja privaatsuse tagamist. Kuna rakenduse loomuses on töödelda kasutaja poolt sisestatud pilte päris inimestest, on Isikuandmete kaitse üldmääruse põhjal tegemist biomeetriliste isikuandmete töötlemisega [85]. Sellega kaasnevad vajalikud arvestused. Kokkuvõtvalt peab mittepüsivate isikuandmete töötlemine olema seaduslik, läbipaistev, vajaduspõhine, asjakohane, minimaalne ning turvaline [86]. Loodud rakendus täidab kõiki mainitud nõudeid. Pildi üleslaadimise lehel on kasutaja jaoks detailselt kirjeldatud, mida sisendpildiga tehakse ning miks need tegevused vajalikud on. Andmete turvalisuse tagab HTTPS, mis kasutab võrguprotokolli TLS (*Transport Layer Security*) ühenduse krüpteerimiseks [87]. TLS krüpteerib ka päringute sees olevad pildid ehk kasutaja isikuandmed.

5 Valminud rakenduse hindamine

Implementeeritud algoritmi rakendatavus päris sisendite peal oleneb selle täpsusest. Järgnev peatükk kirjeldab rakenduse testimise metoodikat, leitud tulemusi ning nende põhjal moodustatud üldhinnangut. Lisaks tuuakse välja kasutajate tagasiside rakenduse kohta.

5.1 Metoodika ja tulemused

Valminud rakenduse testimisvõimalused osutusid ootuspärasest piiratumaks. Autorile teadaolevalt ei leidu avalikult kättesaadavaid andmestikke, mis seoks vastavusse pildi ning kirjeldatud atribuutide klassifikatsiooni. Sellise andmestiku loomine oleks aga osutunud liiga mahukaks. Seetõttu piirduti testimiseks väikese valimiga kasutajatest, kes kasutasid rakendust omaenda portreepildi peal ning võrdlesid saadud tulemusi oodatud tulemustega. Lisaks andsid osalejad oma hinnangud üksteise piltidele.

Testimisest võttis osa 5 osalejat vanusevahemikus 17 – 56 aastat. Igaüks neist tutvus esialgu rakendusega ja selle väljundiga. Seejärel valiti igast osalisest üks portreepilt, mille peal algoritmi katsetada. Pildid jagati laiali igale osalejale ning vastati küsimustiku (vt lisa 1) küsimustele 1 kuni 20. Küsimuste eesmärk oli enne väljundi nägemist saada osaleja oodatud tulemused nii enda kui teiste osalejate piltide kohta. Seejärel laadsid nad pildid ise üles ning võrdlesid saadud tulemusi oma oodatud tulemustega (vt lisa 2).

Tulemustest on näha, et klassifitseerimiste täpsus ei ole ühegi sisendpildi puhul perfektne. Vigu esines rohkem, kui vaadata osalejate oodatud tulemusi üksteise piltide osas. Kuna osalejad valisid enda pilti iseloomustades tihti mitu väärtust, siis oli tulemusel suurem tõenäosus ühega neist ühtida. Teiste osalejate pilte iseloomustati aga konkreetsemalt ühe või kahe väärtusega, mistõttu olid vead tõenäolisemad. Seevastu on enamik erinevusi oodatud ning saadud tulemuste vahel marginaalsed. Oodates näiteks mandlikujulisi silmi, pole ülessuunatud silmad niivõrd vale vastus, kui oleksid ümmargused silmad. Siiski on ka kaheldamatuid eksimusi, näiteks klassifitseeriti enamikel juhtudel heledamad silmad (hallid, sinised, rohelised) pruunideks. Analüüsides sisendpilte lähemalt, selgitab neid eksimusi võrreldes silmavärvi esindajate piltidega suuremad pupillid. Kuna pupillid ise on praktiliselt musta värvi, siis kallutab nende suurenemine silma keskmist värvi tumeda poole. Algoritmi reaalsust peidab ka klassifikatsiooni olemus, kus väljundiks on üksik klass, isegi kui mõne teise kategooria täpsus on väga lähedane. Hea näide sellest on kolmanda osaleja silma värv. Oodati rohelist või halli, aga saadi hoopis vasaku silma värviks roheline ja parema värviks pruun. Klassifikatsioon on selgelt ühe silma puhul vale ja teise puhul õige.

Vale klassifikatsiooni puhul on erinevus kummastki värvist marginaalne, aga kuna pruun on natukene täpsem, siis see ongi lõplik ennustus. Probleemiks osutusid ka pisierinevused võimalike klasside vahel. Tulemustes kordus muster, kus mitme osaleja oodatud tulemus ühtib saadud tulemusega, kuid mitte kõigi.

Üksikuid atribuute vaadates osutus algoritmi täpsuseks 67%, s.t 33,5 atribuuti 50st ühtisid osalejate ootustega enda pildi jaoks. Kui õigeks tulemuseks pidada atribuuti, mida pakkus vähemalt üks osalejatest, siis on täpsus 79%. Grupeerides atribuudid oma loomulikesse klassidesse nagu silmad, näokarvad, juuksed (vt tulemuste tabelit lisas 2) saab hinnata algoritmi täpsust üldisemalt. Arvestades klassi õigesti kirjeldatuks siis, kui vähemalt mingi atribuut sellest ühtib pildi omaniku arvamusega, osutus täpsuseks umbes 73,3%. Kui õigeks vastuseks lugeda ka teiste osalejate ootusi, on see aga umbes 76,7%. Nende tulemuste põhjal võib üldhinnanguna väita, et algoritm on täpne, aga puudujäägid on selged. Selleks, et seda saaks usaldusväärselt kasutada füüsilise deskriptori genereerijana reaalsetes kasutusjuhtudes, peaks algoritm olema rohkem suuteline kõrvalekalletega (näiteks erinevused pupilli suurus) arvestama. Võimalikud klassid peaksid üksteisest ka rohkem erinema, et vähendada erinevusi kasutajate ootustes ning vigu saadud tulemustes. Lisaks oleks tarvilik rakendust testida tunduvalt suurema valimi peal. 5 hästi valitud pilti pole piisav, et saada täielikku ülevaadet algoritmi täpsusest ning kõikvõimalikest puudujääkidest.

5.2 Kasutajate tagasiside

Küsimustiku küsimuste 21 kuni 24 eesmärgiks oli saada üldist tagasisidet väljundi ning rakenduse kohta. Nende eesmärk oli ka panna osalejaid järele mõtlema erinevuste üle oma oodatud ja saadud kirjeldustulemustes. Üldine tagasiside oli positiivne. Toodi välja rakenduse kasutusmugavus, infoküllus ning visuaalne meeldivus. Väljundtekst tundus loomulik, selge ja korrektne, kui välja jätta erinevused oodatud tulemustega. Osalejate sõnul oli kirjeldus piisav, et tekitada mingisuguse vaimse ettekujutuse nende ja teiste osalejate nägudest. Siinkohal aktsepteeriti ka mitmeid erinevusi oodatud ja saadud tulemustes, kuna järele mõeldes leiti sarnasusi ennustatud klasside esindajatega.

Kriitilist tagasisidet anti kirjeldamisprotsessi kohta. Mainiti pikka töötusaega, mille tõttu pidi algoritmi väljundit pikalt ootama. Selle tulemusena võis kohati jääda mulje, et rakendus ei tööta korralikult. Testimise järel mõõdeti töötusaega võrreldes failide suurusega, tulemused on nähtaval tabelis 6.

Tabel 6. Osalejate piltide töötusaeg ning failide suurus.

Osaleja	Töötusaeg sekundites	Pildifaili suurus (KB)
1	30,76	1561
2	14,63	2352
3	37,58	2084
4	15,29	2333
5	15,39	1382

Keskmine töötusaeg oli 22,73 sekundit ning tugevat seost pildifaili suurusega ei leitud. Sellest võib teha järelduse, et töötusaeg sõltub suurel määral pildi morfoloogilisest sisust.

Toodi välja ka mõned selged vead kirjeldustes, mida ei saa seletada pisierinevustega võimalike klasside ning ootuste vahel. Näiteks kui inimesel on rohelised silmad, siis pruuni värvi leidmine on kasutaja vaatest selge viga algoritmis. Vaatamata erinevustele väitsid osalejad siiski, et suurem osa genereeritud kirjeldusest vastab reaalsusele.

Osalejad soovisid muudatusi näha vaid algoritmis endas. Nende sõnul on aktsepteeritav saada oodatust erinevaid tulemusi, kui see erinevus on piisavalt väike, et kasutaja saab tulemusega nõustuda. Kuid mõned tulemused olid selgelt valed, mistõttu oleks vaja analüüsi täpsust parandada. Rakenduse töötusaja lühendamist mainiti ka kui võimalikku parandust, kuid seda peeti pigem pinnapealseks probleemiks.

6 Võimalusi edasiarendamiseks

Peamine puudujääk, mis rakenduse täpsust oluliselt mõjutab, on asjaolu, et algoritm ei arvesta erinevate kõrvalekalletega. Paljud neist esinevad tegelikkuses tihti ning mõjutavad olulisel määral analüüsi käigus leitud väärtusi võrreldes kasutaja poolt oodatud väärtustega. Lihtne näide sellest on tavapärasest eredama valgustuse tõttu suurenenud pupillid. Kuna pupill ise on tüüpiliselt peaaegu must, siis mõjutab selle suurenemine leitud keskmist värvi olulisel määral tumedama poole. Sarnaseid võimalikke kõrvalekaldeid oodatust leidub peaaegu iga kirjeldatava atribuudi jaoks. Laikpigmenditus mõjutab keskmist leitud nahavärvi, ülejäänud juustest erinevat värvi juuksejuured leitud juuste värvi, ebatavapärane juuksestiil leitud näokuju. Võimalik lahendus selliste puudujääkide adresseerimiseks peab suutma neid kõrvalekaldeid ignoreerida, ilma et liialt teavet kaduma läheks. Pupillide probleemi näitel saaks leida silmas olevad kontuurid ning silmavärvi arvutuseks vaadelda vaid pikseleid, mis jäävad väljapoole pupilli kontuuri.

Teine suur puudujääk on asjakohase andmestiku puudus. Rakenduse täpsust mõjutab mingil määral ka asjaolu, et nii-öelda „esindavate väärtuste“ ehk erinevate värviklasside RGB väärtuste või kuju mallide leidmiseks on kasutatud valimit suurusega 1. Niivõrd väikse valimiga ei saa kindel olla, et esindav väärtus vastab võimalikult täpselt reaalsusele. Selle probleemi saaks lahendada koostades piisavas suuruses andmestiku, kus pildid on seoses kirjeldatavate atribuutide väärtustega. Seejärel on võimalik leida esindavad väärtused suurema valimi keskmisena. Sellise andmestiku koostamine võimaldaks ka rakenduse täpsemat testimist.

Rakenduse testimise käigus tuli välja ka asjaolu, et erinevused erinevate võimalike kategooriate vahel ei ole ühtlased. Näiteks on erinevus ovaali- ja oblongikujulise näo vahel tunduvalt väiksem kui ovaali- ja ruudukujulise vahel. Selle tulemusena juhtus olukordi, kus rakendus klassifitseeris mingi atribuudi väga sarnasesse kategooriasse oodatuga ning leitud kaugused kummastki kategooriast olid peaaegu võrdsed. Klassifitseerimise loomu tõttu tagastati vastuseks aga ikkagi kõige lähedasem kategooria. Vältimaks sellistest marginaalsetest erinevustest tulenevaid valesid klassifikatsioone peaks rakendus neil juhtudel klassifitseerima atribuudi mitmesse klassi. Sel juhul peaksid leitud klassid olema ka piisavalt sarnased.

Kasutajakogemuse poolest on veel üheks probleemiks rakenduse töötlusaeg. Algoritmi loomusest tuleneb vajadus läbi töödelda suuri pikslimassiive ning rakendada

erinevaid tehisenägemise algoritme. Selle tulemusena on rakendus arvutusnõudlik ning võtab enamike piltide töötlemiseks paarkümmend sekundit aega. Pikslimassiivide arvutusi on võimalik kiiremini teostada märgatava täpsuse kaoga, käsitledes korraga gruppi lähedal asetsevaid pikseleid, kus keskmine piksel on nende esindaja. Sellise lahenduse kaalumiseks tuleks esialgu eelnevalt mainitud probleemid algoritmi täpsusega lahendada. See lahendus ei parandaks ka töötlusaja probleemi täielikult. Testimise käigus tuli välja, et töötlushaeg pole niivõrd sõltuv faili suuruselt ehk pikslite arvust. Pigem määrab töötlusaja pildi morfoloogia ehk selles leiduv struktuur ja kujud [88]. Sellest võib järeldada, et enamik ajast kulub tehisenägemise algoritmide rakendamisele. Ilma loodud algoritmi fundamentaalselt muutmata pole neid aga võimalik vältida. Suurel määral aitaks töötlushaega parandada ka võimsama protsessoriga hostmasina kasutamine või graafikaprotsessoriga riistvaraline kiirendamine.

Kokkuvõte

Füüsiliste deskriptorite genereerimine on tänapäeval veel aeganõudev käsitöö, mis piirdub üldisemate atribuutide kirjeldamisega. Seevastu on tehisenägemise valdkonna arenguga avanenud võimalus selle protsessi laiendamiseks ning automatiseerimiseks inimese näo detailse kirjelduse loomise eesmärgil.

Käesolevas bakalaureusetöös töötati välja algoritm, mis suudab kirjeldada sisendpildilt leitud inimese näo erinevaid atribuute. Algoritmi implementeerimiseks loodi rakendus „Facedescriber“, mis tagastab tulemusi sõltuvalt kasutaja valikust kas eesti või inglise keeles. Lisaks loodi avalikult kättesaadav veebileht, mille eesmärk on koostada kasutaja üleslaetud pildilt leitud atribuutide põhjal tervikliku loomulikus keeles näokirjelduse. Rakenduse täpsust testisid viis osalejat, kes iseloomustasid üksteise pilte ning võrdlesid oma ootusi saadud tulemustega. Testimise tulemusel leiti, et rakenduse täpsus jääb sõltuvalt hindamiskriteeriumist 67% kuni 79% vahele.

Eesmärk luua rakendus, mis suudaks inimese nägu automaatselt ja detailselt kirjeldada nii loomulikus eesti kui ka inglise keeles, sai täidetud. Kuid selleks, et rakendus täidaks kasutusmalli ka päris maailmas kui füüsilise deskriptori genereerijana, oleks vaja selle täpsust mõjutavad puudujäägid lahendada.

Automaatne näokirjeldus kui tehisenägemise ja pilditöötluse alamvaldkond on olnud väga piiratud senises arengus. Lisaks loodud rakenduse edasistele arendamisvõimalustele on sellele võimalik hulganisti täiendusi juurde mõelda.

Viidatud kirjandus

- [1] Deskriptor, Andmekaitse ja infoturbe portaal AKIT. <https://akit.cyber.ee/term/3721> (10.05.2024).
- [2] Exclusion List - A – Z. Division of Gaming Enforcement. <https://www.njoag.gov/about/divisions-and-offices/division-of-gaming-enforcement-home/exclusion-list/exclusion-list-a-z/> (01.04.2024).
- [3] Yin J., Fang M., Ma W. Automatic labeling of 3D facial acupoint landmarks. Metaverse Research Institute, Guangzhou University, 2024. https://www.researchgate.net/publication/379426269_Automatic_labeling_of_3D_facial_acupoint_landmarks (20.04.2024).
- [4] Luo Y., Guo R. Facial recognition in China: current status, comparative approach and the road ahead. Journal of Law and Social Change, University of Pennsylvania, 2021. <https://scholarship.law.upenn.edu/cgi/viewcontent.cgi?article=1269&context=jlasc> (11.02.2024).
- [5] Ramchandra A., Kumar R. Overview of Face Recognition System Challenges. International Journal of Scientific & Technology Research, 2013. <https://www.ijstr.org/final-print/aug2013/Overview-Of--Face-Recognition-System--Challenges.pdf> (11.02.2024).
- [6] Deepface. Github. <https://github.com/serengil/deepface> (12.02.2024).
- [7] FaceVACS Engine. Cognitec. <https://www.cognitec.com/facevacs-technology.html> (12.02.2024).
- [8] Amazon Rekognition (AMS SSPS). AWS. <https://docs.aws.amazon.com/managedservices/latest/userguide/rekognition.html> (12.02.2024).
- [9] Face Recognition. PyPi. <https://pypi.org/project/face-recognition/> (12.02.2024).
- [10] Image inputs for ChatGPT – FAQ. OpenAI. <https://help.openai.com/en/articles/8400551-image-inputs-for-chatgpt-faq> (12.02.2024).
- [11] Ouyang S. jt. LLM is Like a Box of Chocolates: the Non-determinism of ChatGPT in Code Generation, 2023. <https://arxiv.org/pdf/2308.02828.pdf> (12.02.2024)
- [12] OpenCV-Python. PyPi. <https://pypi.org/project/opencv-python/> (12.02.2024).
- [13] OpenCV is the world's biggest computer vision library. OpenCV. <https://opencv.org/releases/> (12.02.2024).
- [14] Canny J. A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986. https://www.researchgate.net/publication/224377985_A_Computational_Approach_To_Edge_Detection (12.02.2024).

- [15] Flask. <https://flask.palletsprojects.com/en/3.0.x/> (21.04.2024).
- [16] Brihadiswaran G. A Performance Comparison Between C, Java, and Python. Medium, 2020. <https://medium.com/swlh/a-performance-comparison-between-c-java-and-python-df3890545f6d> (21.04.2024).
- [17] Spring Boot. Spring. <https://spring.io/projects/spring-boot> (15.02.2024).
- [18] Apache Tomcat. The Apache Software Foundation. <https://tomcat.apache.org/> (15.02.2024).
- [19] Welcome to Apache Maven. Apache Maven Project. <https://maven.apache.org/> (15.02.2024).
- [20] What is Vue? Vue.js. <https://vuejs.org/guide/introduction.html> (15.02.2024).
- [21] DevServer. Webpack. <https://webpack.js.org/configuration/dev-server/> (15.02.2024).
- [22] Overview. Vue CLI. <https://cli.vuejs.org/guide/> (15.02.2024).
- [23] The Next-Gen UI Suite for Vue.js. PrimeVue. <https://primevue.org/> (15.02.2024).
- [24] About Node.js. Node.js. <https://nodejs.org/en/about> (15.02.2024).
- [25] About npm. Npm Docs. <https://docs.npmjs.com/about-npm> (15.02.2024).
- [26] Welcome to VirtualBox.org! VirtualBox. <https://www.virtualbox.org/> (20.02.2024).
- [27] The story of Ubuntu. Canonical Ubuntu. <https://ubuntu.com/about> (20.02.2024).
- [28] Sulaiman N. S., Raffi A. S. H. A. Comparison of Operating System Performance Between Windows 10 and Linux Mint. International Journal of Synergy in Engineering and Technology, 2021. <https://tatiuc.edu.my/ijset/index.php/ijset/article/download/89/67/> (20.02.2024).
- [29] Hakamian A., Rahmani A. M. Evaluation of isolation in virtual machine environments encounter in effective attacks against memory. Security and Communication Networks, Hindawi, 2015. https://www.researchgate.net/publication/282940674_Evaluation_of_isolation_in_virtual_machine_environments_encounter_in_effective_attacks_against_memory (20.02.2024).
- [30] Patel O., Maravi Y. P. S., Sharma S. A comparative study of histogram equalization based image enhancement techniques for brightness preservation and contrast enhancement. Signal & Image Processing: An International Journal, 2013. <https://arxiv.org/pdf/1311.4033.pdf> (27.02.2024).
- [31] Histograms – 2: Histogram Equalization. OpenCV. https://docs.opencv.org/4.x/d5/daf/tutorial_py_histogram_equalization.html (27.02.2024).

- [32] Face_recognition package. Face Recognition. https://face-recognition.readthedocs.io/en/latest/face_recognition.html (23.02.2024).
- [33] Dlib. Github. <https://github.com/davisking/dlib> (27.02.2024).
- [34] Cholet S., Paugam-Moisy H. Prototype-based Classifier for Automatic Diagnosis of Depressive Mood. International Conference on Biomedical Engineering and Applications, Funchal, 2018. https://www.researchgate.net/publication/327935335_Prototype-based_Classifier_for_Automatic_Diagnosis_of_Depressive_Mood (27.02.2024).
- [35] Charlton M. jt. The effect of constitutive pigmentation on the measured emissivity of human skin. PLOS ONE, 2020. https://www.researchgate.net/publication/347178606_The_effect_of_constitutive_pigmentation_on_the_measured_emissivity_of_human_skin (04.03.2024).
- [36] Oakley A. Fitzpatrick skin phototype. DermNet, 2012. <https://dermnetnz.org/topics/skin-phototype> (04.03.2024).
- [37] Akhand M. A. H., Akash A. S., Mollah A. S. Improvement of Haar Feature Based Face Detection in OpenCV Incorporating Human Skin Color Characteristic. International Journal of Computer Applications & Information Technology, 2016. https://www.researchgate.net/publication/310443424_Improvement_of_Haar_Feature_Based_Face_Detection_in_OpenCV_Incorporating_Human_Skin_Color_Characteristic (04.03.2024).
- [38] Chao Y. Y. Y. jt. Pan-Aasian Consensus – Key recommendations for adapting the World congress of dermatology consensus on combination treatment with injectable fillers, toxins, and ultrasound devices in Asian patients. Journal of Clinical and Aesthetic Dermatology, 2017. <https://pubmed.ncbi.nlm.nih.gov/28979659/> (04.03.2024).
- [39] Image Thresholding. OpenCV. https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html (04.03.2024).
- [40] Saeed K. jt. K3M: A universal algorithm for image skeletonization and a review of thinning techniques. International Journal of Applied Mathematics and Computer Science, 2010. https://www.researchgate.net/publication/220273912_K3M_A_universal_algorithm_for_image_skeletonization_and_a_review_of_thinning_techniques (04.03.2024).
- [41] Extended Image Processing. OpenCV. https://docs.opencv.org/4.x/df/d2d/group_ximgproc.html#ga37002c6ca80c978edb6ead5d6b39740c (04.03.2024).
- [42] Canny Edge Detection. OpenCV. https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html (12.02.2024).

- [43] Template Matching. OpenCV. https://docs.opencv.org/4.x/d4/dc6/tutorial_py_template_matching.html (04.03.2024).
- [44] Rosebrock A. OpenCV Morphological Operations. Pyimagesearch, 2021. <https://pyimagesearch.com/2021/04/28/opencv-morphological-operations/> (04.03.2024)
- [45] Image Filtering. OpenCV. https://docs.opencv.org/4.x/d4/d86/group_imgproc_filter.html#ga67493776e3ad1a3df63883829375201f (04.03.2024).
- [46] Structural Analysis and Shape Descriptors. OpenCV. https://docs.opencv.org/3.4/d3/dc0/group_imgproc_shape.html#ga17ed9f5d79ae97bd4c7cf18403e1689a (04.03.2024).
- [47] Drawing Functions. OpenCV. https://docs.opencv.org/3.4/d6/d6e/group_imgproc_draw.html#ga9bb982be9d641dc51edd5e8ae3624e1f (04.03.2024).
- [48] Alopecia Totalis: Causes, Symptoms & Treatment. Wimpole Clinic. <https://wimpole-clinic.com/blog/alopecia-totalis-causes-symptoms-treatment/> (04.03.2024).
- [49] Jason Alexander. Apple TV. <https://tv.apple.com/us/person/jason-alexander/umc.cpc.jr8zbgh4rpjcfg5rd96rxx8r> (04.03.2024).
- [50] Brendan Fraser 2023 Wallpapers. WallpaperCave. <https://wallpapercave.com/brendan-fraser-2023-wallpapers> (04.03.2024).
- [51] People. Pinterest. <https://www.pinterest.com/pin/384565255675427666/> (04.03.2024).
- [52] The Ultimate Guide to Men's Short Haircuts and Styles. Pinterest. https://ar.pinterest.com/pin/364017582355308277/?amp_client_id=CLIENT_ID%28_%29&mweb_unauth_id= (04.03.2024).
- [53] Morgan M. D. jt. Genome-wide study of hair colour in UK Biobank explains most of the SNP heritability. Nature Communications, 2018. https://www.researchgate.net/publication/329405511_Genome-wide_study_of_hair_colour_in_UK_Biobank_explains_most_of_the_SNP_heritability (05.03.2024).
- [54] Zheng L. jt. Hair transparency decoding in Asia: From stylists' perception to in vitro measurement. Skin Research and Technology, 2021. https://www.researchgate.net/publication/356789415_Hair_transparency_decoding_in_Asia_From_stylists'_perception_to_in_vitro_measurement (05.03.2024).

- [55] Scientists may have discovered why hair turns grey. The Guardian. <https://www.theguardian.com/science/2023/apr/19/scientists-may-have-discovered-why-hair-turns-grey> (05.03.2024).
- [56] Untitled. Pinterest. <https://www.pinterest.com/pin/untitled-a-beard-that-fans-out-in-2022--642255596871342594/> (06.03.2024).
- [57] 19 Best Blonde Beard Styles (2024 Guide). Pinterest. <https://www.pinterest.com/pin/19-best-blonde-beard-styles-2022-guide--353391901998050536/> (06.03.2024).
- [58] The Crucial Step to Cleaning Up Your Beard. Pinterest. <https://www.pinterest.com/pin/the-crucial-step-to-cleaning-up-your-beard--932667404123184967/> (06.03.2024).
- [59] Swag. Pinterest. <https://www.pinterest.com/pin/568016571737478012/> (06.03.2024).
- [60] Portrait of man with grey beard wearing spectacles and hat – stock photo. Gettyimages. <https://www.gettyimages.com/detail/photo/portrait-of-man-with-grey-beard-wearing-spectacles-royalty-free-image/601823085> (06.03.2024).
- [61] Smith K., Kempe V., Wood L. Eye Placement Bias Is Remarkably Robust. i-Perception, 2021. https://www.researchgate.net/publication/351875058_Eye_Placement_Bias_Is_Remarkably_Robust (06.03.2024).
- [62] Odabas B. jt. Comparison of Facial Proportions and Anthropometric Norms Among Turkish Young Adults With Different Face Types. Aesthetic Plastic Surgery, 2008. https://www.academia.edu/20829414/Comparison_of_Facial_Proportions_and_Anthropometric_Norms_Among_Turkish_Young_Adults_With_Different_Face_Types (06.03.2024).
- [63] Conroy P. Eye Shapes. Vision Direct, 2023. <https://www.visiondirect.com.au/optical-centre/eye-shapes/> (06.03.2024).
- [64] Anya Taylor-Joy had a ‘life-changing’ experience on ‘Furiosa’. News Channel 3. <https://kesq.com/entertainment/cnn-entertainment/2022/11/17/anya-taylor-joy-had-a-life-changing-experience-on-furiosa/> (06.03.2024).
- [65] Morfologia Facial. Ana Paula da Cunha, 2017. <https://anapauladacunha.wordpress.com/2017/05/29/morfologia-facial/> (06.03.2024).
- [66] Short A. The relationship between latitudinal light variation and orbit and cranial size in humans. Journal of Historical Archaeology & Anthropological Sciences, 2017.

- <https://medcraveonline.com/JHAAS/the-relationship-between-latitudinal-light-variation-and-orbit-and-cranial-size-in-humans.html> (09.03.2024).
- [67] Callahan C. What is my eyeshape? Today, 2019. <https://www.today.com/style/what-my-eye-shape-best-tips-each-according-makeup-artists-t157085> (09.03.2024).
- [68] What color are your eyes exactly? Eye Doctors of Washington. <https://www.edow.com/general-eye-care/eyecolor/> (11.03.2024).
- [69] Stelzer O. Iris heterochromia: variations in form, age changes, sex dimorphism. Anthropol Anz, 1979. <https://pubmed.ncbi.nlm.nih.gov/485098/> (11.03.2024).
- [70] Different types of nose shapes. Centreforsurgery. <https://centreforsurgery.com/what-are-the-different-types-of-nose-shapes/> (12.03.2024).
- [71] Jennifer Anniston. Biography. <https://www.biography.com/actors/jennifer-aniston> (12.03.2024).
- [72] Ryan Gosling. Wikipedia. https://diq.wikipedia.org/wiki/Ryan_Gosling (12.03.2024).
- [73] Adrien Brody. Rotten Tomatoes. https://www.rottentomatoes.com/celebrity/adrien_brody (12.03.2024).
- [74] Wayne Rooney. <https://www.transfermarkt.com/wayne-rooney/profil/trainer/86680> (12.03.2024).
- [75] Obama, Barack. Eesti Entsüklopeedia. http://entsyklopeedia.ee/artikkel/obama_barak (12.03.2024).
- [76] Emma Stone. MUBI. <https://mubi.com/en/cast/emma-stone> (12.03.2024).
- [77] Disney/Pixar's "Cars 2" Los Angeles Premiere – Arrivals. Gettyimages. <https://www.gettyimages.ie/detail/news-photo/anthony-kiedis-and-everly-bear-kiedis-attend-the-premiere-news-photo/117074734> (12.03.2024).
- [78] Bill Clinton. Raamaturott. https://raamaturott.fandom.com/et/wiki/Bill_Clinton (12.03.2024).
- [79] Mark Ruffalo. Popsugar. <https://www.popsugar.co.uk/Mark-Ruffalo> (12.03.2024).
- [80] Jackie Chan. Wikipedia. https://et.wikipedia.org/wiki/Jackie_Chan (12.03.2024).
- [81] Let's Encrypt. <https://letsencrypt.org/> (30.03.2024).
- [82] How to check whether a string is Base64 encoded or not. Stackoverflow. <https://stackoverflow.com/questions/8571501/how-to-check-whether-a-string-is-base64-encoded-or-not> (25.03.2024).

- [83] Käsusüst, Andmekaitse ja infoturbe portaal AKIT. <https://akit.cyber.ee/term/9356> (30.03.2024).
- [84] Testing for Command Injection. OWASP. https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/07-Input_Validation_Testing/12-Testing_for_Command_Injection (30.03.2024).
- [85] Art 4. GDPR Definitions. Intersoft consulting. <https://gdpr-info.eu/art-4-gdpr/> (30.03.2024).
- [86] Art 5. GDPR Principles relating to processing of personal data. Intersoft consulting. <https://gdpr-info.eu/art-5-gdpr/> (30.03.2024).
- [87] Technical Guidelines. The HTTPS-Only Standard. <https://https.cio.gov/technical-guidelines/> (04.03.2024).
- [88] Ledda A. Mathematical Morphology in Image Processing. Universiteit Gent, 2007. https://www.researchgate.net/publication/236902389_Mathematical_Morphology_in_Image_Processing (12.05.2024).

Lisad

I. Kasutajatestimise küsimustik

1. Missugune järgnevatest piltidest on kõige lähedasem Teie pildile naha värvi poolest? Kahtluse puhul võib märkida mitu vastust.



Phototype I



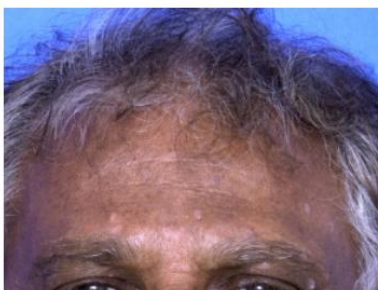
Phototype II



Phototype III



Phototype IV



Phototype V



Phototype VI

Joonis 1. Näited kuuest Fitzpatricki fototüübist [35].

- Kahvatu valge (*Phototype I*)
- Hele (*Phototype II*)
- Tumedam valge (*Phototype III*)
- Helepruun (*Phototype IV*)
- Pruun (*Phototype V*)
- Tumepruun või must (*Phototype VI*)

2. Kuidas kategoriseeriksite naha värvi teiste osalejate piltides? Kahtluse puhul võib märkida mitu vastust.

Osaleja	Naha värv
2	
3	

4	
5	

3. Missugune järgnevatest piltidest on kõige lähedasem Teie pildile näo kuju poolest? Kahtluse puhul võib märkida mitu vastust.



Joonis 2. 5 üldtunnustatud näo kuju, vasakult paremale vaadates: teemant, oblong, ovaal, ümmargune, ruut [37].

- Teemant (*Inverted Triangle/Diamond*)
- Ovaal (*Oval/Heart*)
- Oblong (*Oblong/Rectangle*)
- Ümmargune (*Round*)
- Ruut (*Square/Triangle*)

4. Kuidas kategoriseeriksite näo kuju teiste osalejate piltides? Kahtluse puhul võib märkida mitu vastust.

Osaleja	Näo kuju
2	
3	
4	
5	

5. Kas Teie pildis on Teil juuksed olemas?

- Jah
- Ei

6. Kas Teie hinnangul on teistel osalejatel nende piltides juuksed olemas?

Osaleja	Juuste olemasolu
2	
3	
4	
5	

7. Missugune järgnevatest piltidest on kõige lähedasem Teie pildile juuste värvi poolest? Kahtluse puhul võib märkida mitu vastust.



Joonis 3. Välja lõigatud pildid erinevatest juuste värvidest. Vasakult paremale vaadates: (a) punased, (b) blondid, (c) pruunid, (d) mustad [53], (e) hallid [54].

- Punased (a)
- Blondid (b)
- Pruunid (c)
- Mustad (d)
- Hallid (e)

8. Kuidas kategoriseeriksite juuste värvi teiste osalejate piltides? Kahtluse puhul võib märkida mitu vastust.

Osaleja	Juuste värv
2	
3	
4	
5	

9. Missugused näokarvad on Teil pildis?

- Puuduvad/väga õrnad
- Õrnad
- Tihedad

10. Kuidas hindaksite näokarvu teiste osalejate piltides?

Osaleja	Näokarvad
2	
3	
4	
5	

11. Juhul kui Teil on näokarvad, siis missugune järgnevatest piltidest on kõige lähedasem Teie pildile näokarvade värvi poolest? Kahtluse puhul võib märkida mitu vastust.



Joonis 4. Väljalõigatud pildid näokarvade värvide kategooriate eksemplaridest. Vasakult paremale vaadates (a) punane [55], (b) blond [56], (c) pruun [57], (d) must [58], (e) hall [59].

- Punased (a)
- Blondid (b)
- Pruunid (c)
- Mustad (d)
- Hallid (e)

12. Kuidas kategoriseeriksite näokarvade värvi teiste osalejate piltides? Näokarvade puudumise korral pole vaja lahtrit täita. Kahtluse puhul võib märkida mitu vastust.

Osaleja	Naha värv
2	
3	
4	
5	

13. Missugune järgnevatest piltidest on kõige lähedasem Teie pildile silmade asetuse poolest? Asetuse all mõeldakse silmade horisontaalset kaugust üksteisest.



(a) Laialt asetsevad silmad: silmade vaheline distantis on ~1.24 korda suurem kui keskmine silma laius



(b) Lähedal asetsevad silmad: silmade vaheline distantis on ~0.91 korda väiksem kui keskmine silma laius

Joonis 5. Näidispildid laialt asetsevatest silmadest [63] ja lähedal asetsevatest silmadest [64] koos ligikaudsete silmade vahelise distantisi ja keskmise silma laiuse suhetega.

- Laialt asetsevad (a)
- Lähedal asetsevad (b)

14. Kuidas kategoriseeriksite silmade asetust teiste osalejate piltides?

Osaleja	Silmade asetis
2	
3	
4	
5	

15. Missugune järgnevatest piltidest on kõige lähedasem Teie pildile silmade kuju poolest? Kahtluse puhul võib märkida mitu vastust.



Joonis 6. Eksemplarid erinevatest silmade kujudest. Vasakult paremale vaadates: (a) mandlikujulised, (b) ümmargused, (c) monoliidsed, (d) alla suunatud, (e) üles suunatud, (f) varjatud/kapuutsiga [66].

- Mandlikujulised (a)
- Ümmargused (b)
- Monoliidsed (c)
- Alla suunatud (d)
- Üles suunatud (e)
- Varjatud/kapuutsiga (f)

16. Kuidas kategoriseeriksite silmade kuju teiste osalejate piltides? Kahtluse puhul võib märkida mitu vastust.

Osaleja	Silmade kuju
2	
3	
4	
5	

17. Missugune järgnevatest piltidest on kõige lähedasem Teie pildile silmade värvi poolest? Juhul kui teie silmad on eri värvi, märkige kummagi silma värv eraldi. Kahtluse puhul võib märkida mitu vastust.



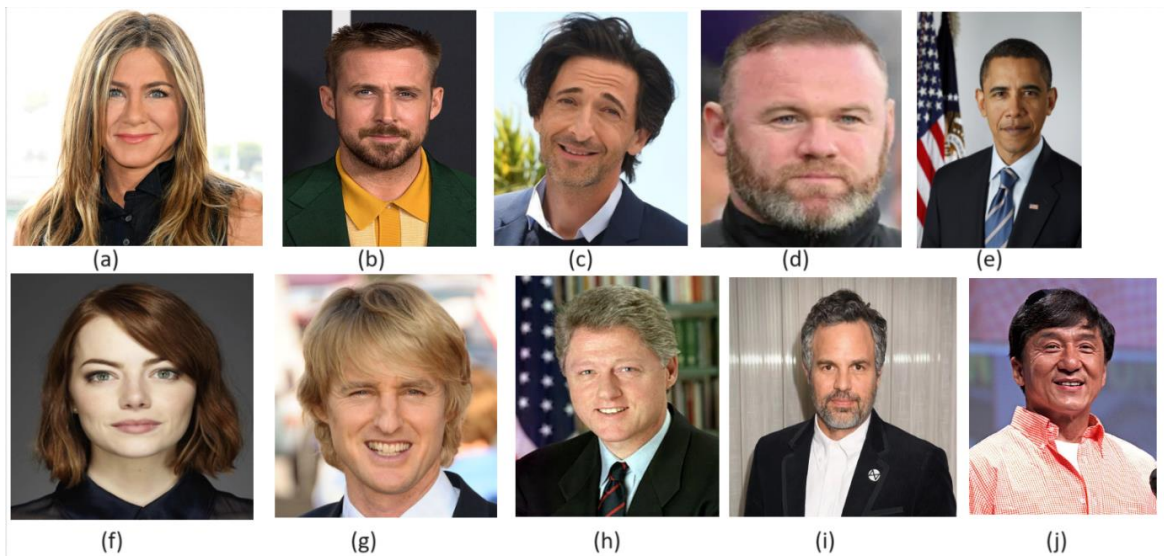
Joonis 7. Silmade värvide eksemplarid. Vasakult paremale vaadates: (a) sinine, (b) pruun, (c) roheline, (d) hall, (e) pähkelpruun, (f) punane [67].

- Sinine (a)
- Pruun (b)
- Roheline (c)
- Hall (d)
- Pähkelpruun (e)
- Punane (f)

18. Kuidas kategoriseeriksite silmade värvi teiste osalejate piltides? Juhul kui nende silmad on eri värvi, märkige kummagi silma värv eraldi. Kahtluse puhul võib märkida mitu vastust.

Osaleja	Naha värv
2	
3	
4	
5	

19. Missugune järgnevatest piltidest on kõige lähedasem Teie pildile nina kuju poolest? Kahtluse puhul võib märkida mitu vastust.



Joonis 8. Nina kujude eksemplarid. Vasakult paremale vaadates: (a) kreeka [70], (b) rooma [71], (c) kulli/kongus [72], (d) nööp- [73], (e) nuubia [74], (f) ülespööratud [75], (g) kõver [76], (h) sibulakujuline [77], (i) lihav [78], (j) ida-aasia [79] nina.

- Kreeka ehk sirge nina (a)
- Rooma ehk küürus nina (b)
- Kulli/kongus nina (c)
- Nööpnina (d)
- Nuubia ehk aafrika nina (e)
- Ülespööratud nina (f)
- Kõver nina (g)
- Sibulakujuline nina (h)
- Lihav nina (i)
- Ida-aasia nina (j)

20. Kuidas kategoriseeriksite nina kuju teiste osalejate piltides? Kahtluse puhul võib märkida mitu vastust.

Osaleja	Nina kuju
2	
3	
4	

21. Kuidas hindaksite väljundi kvaliteeti? Mida kirjeldas rakendus õigesti ja mida valesti?

--

22. Kas väljundtekst tundub loomulik ning tekitab mingisuguse vaimse ettekujutuse kirjeldatud näost? Kas selle sõnastus ja disain on selge või sooviksite muuta midagi?

--

23. Kuidas hindaksite rakendust üldisemalt? Kas sellega on mingeid silmnähtavaid probleeme, vigu või ebaseaduslikusi?

--

24. Kas on mingeid muudatusi, mida rakenduses teeksite?

--

II. Kasutajatestimise tulemuste tabel

		Nahavärv	Näo kuju	Juuksed	Näokarvad	Silmad	Nina
Osaleja 1	Oodatud tulemused (osaleja 1)	Hele või tumedam valge	Ümmargune või ovaal	Pruunid	Puuduvad	Rohelised või hallid, lähedal asetsevad, mandlikujulised või alla suunatud	Kreeka või lihav või nõõpnina
	Oodatud tulemused (osaleja 2)	Tumedam valge	Ümmargune	Pruunid	Puuduvad	Hallid, lähedal asetsevad, mandlikujulised	Nõõpnina
	Oodatud tulemused (osaleja 3)	Tumedam valge	Ovaal	Pruunid	Puuduvad	Hallid, laia asetusega, ümmargused	Kreeka
	Oodatud tulemused (osaleja 4)	Hele	Ovaal	Pruunid	Puuduvad	Rohelised, laia asetusega, ümmargused	Kreeka
	Oodatud tulemused (osaleja 5)	Tumedam valge	Ümmargune	Pruunid	Puuduvad	Sinised, laia asetusega, alla suunatud	Kreeka
	Saadud tulemused	Tumedam valge	Ruudu kujuline	Pruunid	Puuduvad	Pruunid, lähedal asetsevad, alla suunatud	Nõõpnina
Osaleja 2	Oodatud tulemused (osaleja 2)	Hele	Ümmargune	Puuduvad	Õrnad, pruunid või hallid	Pruunid, lähedal asetsevad, mandlikujulised või alla suunatud	Lihav
	Oodatud tulemused (osaleja 1)	Hele	Oblong	Puuduvad	Tihedad, mustad või hallid	Pruunid, lähedal asetsevad, alla suunatud	Nõõpnina
	Oodatud tulemused (osaleja 3)	Tumedam valge	Oblong	Puuduvad	Tihedad, mustad või hallid	Pruunid, lähedal asetsevad, alla suunatud	Sibulakujuline

	Oodatud tulemused (osaleja 4)	Hele	Ümmargune	Puuduvad	Õrnad, pruunid	Pruunid, lähedal asetsevad, mandlikujulised	Ülespööratud
	Oodatud tulemused (osaleja 5)	Tumedam valge	Oblong	Puuduvad	Tihedad, pruunid või hallid	Pruunid, lähedal asetsevad, alla suunatud	Sibulakujuline
	Saadud tulemused	Hele	Ruudu kujuline	Puuduvad	Õrnad, pruunid	Pruunid, laia asetusega, monoliidsed	Nööpnina
Osaleja 3	Oodatud tulemused (osaleja 3)	Hele	Ovaal või teemant	Pruunid	Õrnad või puuduvad, hallid	Rohelised või hallid, lähedal asetsevad, alla suunatud	Sibulakujuline
	Oodatud tulemused (osaleja 1)	Tumedam valge	Oblong	Pruunid	Õrnad, mustad või hallid	Rohelised, laia asetusega, varjatud/kapuutsiga	Kõver
	Oodatud tulemused (osaleja 2)	Tumedam valge	Ovaal	Pruunid	Õrnad, hallid	Rohelised, lähedal asetsevad, alla suunatud	Lihav
	Oodatud tulemused (osaleja 4)	Hele	Ovaal	Pruunid	Õrnad, hallid	Rohelised, lähedal asetsevad, alla suunatud	Sibulakujuline
	Oodatud tulemused (osaleja 5)	Hele	Ovaal	Pruunid	Õrnad, mustad või hallid	Rohelised, laia asetusega, varjatud/kapuutsiga	Kõver
	Saadud tulemused	Hele	Ümmargune	Pruunid	Puuduvad	Vasak roheline ja parem pruun, laia asetusega, alla suunatud	Rooma
Osaleja 4	Oodatud tulemused (osaleja 4)	Hele või tumedam valge	Teemant	Blondid või puuduvad	Puuduvad	Sinised või hallid, lähedal asetsevad, mandlikujulised	Rooma või lihav
	Oodatud tulemused (osaleja 1)	Tumedam valge	Teemant	Blondid	Õrnad, blondid	Hallid, lähedal asetsevad, varjatud/kapuutsiga	Sibulakujuline

	Oodatud tulemused (osaleja 2)	Hele	Ovaal	Blondid	Puuduvad	Hallid, lähedal asetsevad, mandlikujulised	Nööpnina
	Oodatud tulemused (osaleja 3)	Tumedam valge	Teemant	Blondid	Puuduvad	Hallid, laia asetusega, mandlikujulised	Sibulakujuline
	Oodatud tulemused (osaleja 5)	Tumedam valge	Ovaal	Blondid	Puuduvad	Hallid, laia asetusega, mandlikujulised	Sibulakujuline
	Saadud tulemused	Tumedam valge	Ümmargune	Blondid	Puuduvad	Pruunid, lähedal asetsevad, monoliidsed	Köver
Osaleja 5	Oodatud tulemused (osaleja 5)	Hele või tumedam valge	Ovaal	Pruunid	Puuduvad	Pruunid, lähedal asetsevad, mandlikujulised	Rooma
	Oodatud tulemused (osaleja 1)	Tumedam valge	Ovaal	Pruunid	Puuduvad	Hallid, laia asetusega, üles suunatud	Rooma
	Oodatud tulemused (osaleja 2)	Tumedam valge	Oblong	Pruunid	Puuduvad	Pruunid, lähedal asetsevad, mandlikujulised	Kreeka
	Oodatud tulemused (osaleja 3)	Tumedam valge	Ovaal	Pruunid	Puuduvad	Pähkelpruunid, lähedal asetsevad, üles suunatud	Rooma
	Oodatud tulemused (osaleja 4)	Tumedam valge	Oblong	Pruunid	Puuduvad	Hallid, lähedal asetsevad, mandlikujulised	Rooma
	Saadud tulemused	Tumedam valge	Ovaal	Pruunid	Puuduvad	Pruunid, laia asetusega, üles suunatud	Nööpnina

III. Litsents

Lihlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Karl-Magnus Laikoja,

1. annan Tartu Ülikoolile tasuta loa (lihlitsentsi) minu loodud teose **Inimese näo automaatne kirjeldamine loomulikus eesti ja inglise keeles**, mille juhendaja on Sven Aller, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Karl-Magnus Laikoja

14.05.2024