

ŠIAULIŲ UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Oleg Mazec

Informatikos magistro specialybės II kurso dieninio skyriaus studentas

Fotografijų analizės metodai ir jų taikymai
Image Processing Methods and their Practical
Applications

Magistro darbas

Darbo vadovas:

Lekt. G. Felinskas

Recenzentas:

Doc. R. Laurutis

Šiauliai, 2012

Turinys

Įvadas	2
1. Analitinė dalis	3
1.1. Vaizdų paieškos sistemos.....	3
1.1.1. Bing Image Search	4
1.1.2. Gazopa image search.....	5
1.1.3. Imense image search portal	6
1.1.4. Img(Rummager)	6
1.2. Vaizdų analizės metodai	7
1.2.1. Histogramos ir jų panaudojimas vaizdų analizėje.....	10
1.2.2. Galinis histogramos projektavimas (<i>angl. Backprojecting</i>).....	11
1.2.3. Panašių vaizdų paieška naudojant histogramų palyginimą	12
1.3. Objektų paieška ir atpažinimas	12
1.3.1. Veidų atpažinimo metodai	14
1.3.2. Tikriniai veidai (<i>angl. Eigenfaces</i>).....	14
1.3.3. Fišerio veidai (<i>angl. Fisherface</i>).....	15
1.3.4. Lokalių binarinių šablonų histograma.....	17
2. Projektinė dalis.....	18
2.1. OpenCV 2.4 biblioteka.....	18
2.2. Integrating Vision Toolkit.....	20
2.3. AForge.Net Framework	20
2.4. Įrankių pasirinkimo analizė.....	21
2.5. Vaizdų analizės įrankio pradinis projektas.....	22
3. Galutinis projekto aprašymas.....	23
3.1. Veidų atpažinimo modulis	24
3.2. Problemos ir jų sprendimo būdai	26
3.3. Darbo rezultatų analizė	27
3.3.1. Įrankio ir metodų testavimas.....	27
3.3.2. Realizuotų metodų analizė	28
4. Išvados	34
Naudota literatūra:.....	35
Anotacija	37
Priedai	38
1. CD turinys	38
2. Programos naudojimas	38

Ivadas

Sparčiai besivystančių technologijų pagrindinis tikslas yra palengvinti žmonių darbą ir pagerinti jų gyvenimo kokybę. Jei iš pradžių jos buvo naudojamos tam, kad nevarginti žmonių sunkiais pasikartojančiais darbais arba sudėtingais skaičiavimais, tai su sparčių kompiuterinių technologijų vystymusi bandoma perimti ir kitus žmogaus veiksmus – analizę ir sprendimų priėmimą. Tai lėmė spartų dirbtinio intelekto vystymą. Dabar jis skirstomas į daug sričių – žinių atvaizdavimą, mokymąsi, natūralaus teksto apdorojimą ir kitas. Viena iš jo pritaikymo sričių yra skaitmeninių vaizdų apdorojimas, pavyzdžiui: medicinoje, transporto priemonėse (pvz. kelio ženklų atpažinimas), filmuose, fotografijoje, teksto atpažinimo sistemose, apsaugos sistemose ir kitur.

Viena iš taikymo sričių – fotografijų paieška pagal sistemos turinį. Šios sistemos atlieka fotografijų analizę – fotografijose ieško objektų atitinkančių paieškos kriterijus (raktinius žodžius arba kitas fotografijas). Jos turėtų pakeisti šiuo metu naudojamą sistemas, kurios yra nepatikimos, nes atlieka paiešką tik pagal tekstą arba skaitmeninių failų meta duomenis. Keletas pavyzdžių būtų: „Bing image search“, „Imense image search portal“ ir kiti. Šios sistemos yra internetinės paieškos sistemos, bet dėl didelio kiekio fotografijų, kurias gali sukaupti asmuo arba įmonė, paieškos sistemos skirtos asmeniniams kompiuteriams arba veikiančios tinkluose taip pat yra aktualios ir vystomos.

Šio darbo tikslas yra išanalizuoti skaitmeninių vaizdų apdorojimo metodus, ir sukurti įrankį, skirtą skaitmeninių fotografijų paieškai, analizei ir palyginimui pagal jų turinį.

Darbo uždaviniai:

- Apžvelgti egzistuojančias paieškos pagal turinį sistemas;
- Apžvelgti skaitmeninių vaizdų analizės metodus ir atrinkti tuos, kurie bus naudojami tolimesnei analizei;
- Susipažinti su egzistuojančiomis vaizdų analizės bibliotekomis ir jų galimybėmis;
- Parinkti projekto realizavimo įrankius;
- Realizuoti ir palyginti pasirinktus apdorojimo metodus;
- Sukurti vaizdų analizės įrankį.

1. Analitinė dalis

1.1. Vaizdų paieškos sistemos

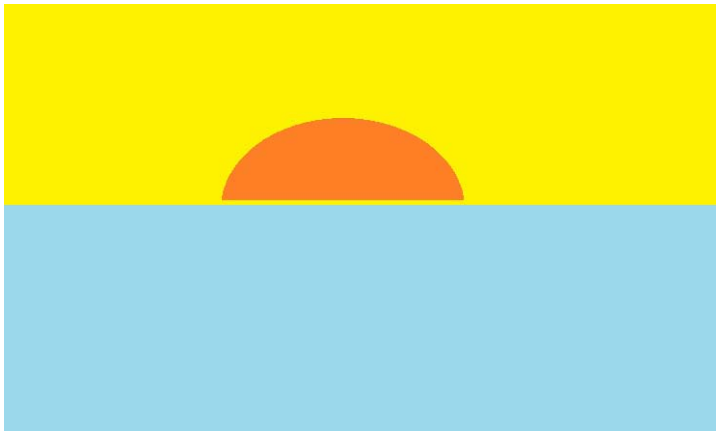
Kai kalbama apie vaizdų paiešką pagal turinį reikia išskirti keletą svarbių dalykų. Pirmas – paieška gali būti atliekama internete, lokaliai arba tinklu pasiekiamuose duomenų kaupimo įrenginiuose. Antras – paieška gali būti atliekama lyginant pavyzdinį vaizdą su turimais ir atrenkant tinkamiausius, arba išanalizuojant turimus vaizdus ir automatiškai priskiriant žymes(angl. automatic image tagging) ir po to atliekant paiešką pagal raktinius žodžius, nurodytus vartotojo arba gautus po pavyzdinio vaizdo automatinės analizės.

Jeigu paieška internete nuo paieškos tinkle arba lokaliuose įrenginiuose skiriasi tik duomenų kiekiu, tai pačios paieškos metodai tarp vaizdų lyginimo ir žymių lyginimo skiriasi labai stipriai. Tiek vieną, tiek kitą būdą naudoja gana daug paieškos įrankių. Toliau bus aprašyti keli iš jų, ir bus palyginti ir įvertinti jų paieškos rezultatai. Priklausomai nuo paieškos tipo bus naudojami raktinių žodžių rinkiniai ir/arba pavyzdiniai vaizdai:



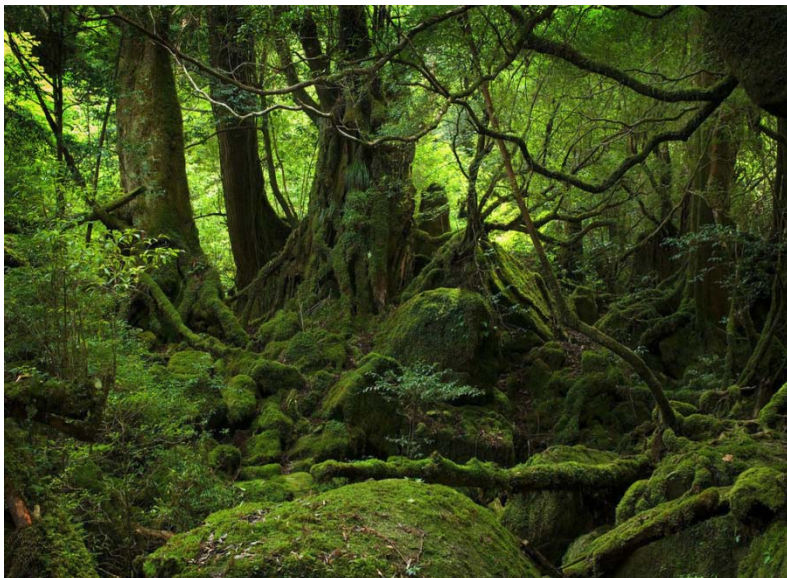
Pav. 1 Paieškos vaizdas nr.1.

Pirmas vaizdas – aukštos kokybės vaizdas, turintis nedaug spalvų ir formų, parodantis kraštovaizdį. Panašių vaizdų yra daug, todėl atliekant paiešką nereikia galvoti apie tai, kad vaizdas yra labai unikalus ir panašių vaizdų rasti neįmanoma. Su šiuo vaizdu bus susieta keletas paprastų žodžių: mountains, sky, desert, orange.



Pav. 2 Paieškos vaizdas nr.2.

Antras vaizdas – grubiai nupieštas vaizdas, sudarytas tik iš kelių geometrinių figūrų ir trijų spalvų. Atitinkantys raktiniai žodžiai: sunset, blue, water, yellow, orange.



Pav. 3 Paieškos vaizdas nr.3.

Trečias vaizdas – aukštos kokybės fotografija, su daug detalių ir objektų, ir viena dominuojančia spalva. Raktiniai žodžiai: forest, green, trees.

1.1.1. Bing Image Search

Ši paieškos sistema išleista Microsoft kompanijos 2009 metais. Ji leidžia atlikti vaizdų internete paiešką pagal raktinius žodžius ir pagerinti rezultatai nurodant dominuojančias spalvas, dydį ir kitus parametrus. Neleidžia atlikti paieškos pagal pavyzdį. Paieškos rezultatai:

Pagal pirmą paiešką nebuvo rasta tokio pat vaizdo, bet buvo rastas didelis kiekis tikrai atitinkančių raktinius žodžius ir panašių į pirmąjį paieškos vaizdą. Pav.4 buvo vienas pirmųjų rastų vaizdų. Tačiau tarp rezultatų buvo ir nemažai vaizdų atitinkančių tik vieną raktinį žodį ir pasirinktą pagrindinę spalvą.



Pav. 4 Paieškos rezultatas Bing image search.

Antros paieškos rezultatai buvo panašūs į pirmosios, tokio vaizdo nebuvo, bet panaudojant kelis raktinius žodžius, buvo rastas didelis kiekis atitinkančių vaizdų.

Trečios paieškos metu pagal raktinius žodžius buvo rastas lygiai toks pat vaizdas kaip Pav.3. tarp pirmųjų rezultatų nurodžius dominuojančią spalvą ir tai, kad ieškoma fotografija.

1.1.2. Gazopa image search

Ši paieškos sistema išleista 2008 metais Hitachi kompanijos. Ji leidžia atlikti vaizdų paiešką internete pagal raktinius žodžius, pagal pavyzdinį vaizdą arba nupieštą tinklapyje vaizdą.

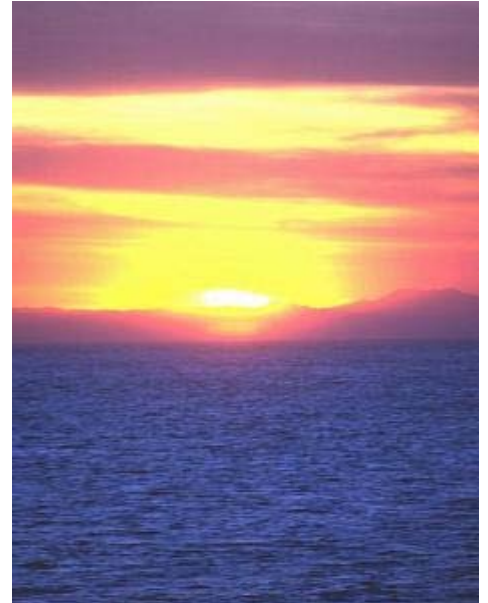
Atlikus pirmo vaizdo paiešką pagal pavyzdį rezultatas – rastas didelis lygiai kiekis tokių pat vaizdų kaip Pav.1 ir labai didelis kiekis panašių vaizdų. Atlikus paiešką pagal raktinius žodžius rezultatas buvo blogesnis, nes tokio pat vaizdo nebuvo ir buvo daug neatitinkančių raktinius žodžius vaizdų.

Antro vaizdo paieška buvo įdomesnė, buvo rasta daug panašių vaizdų tiek geometrijos, tiek spalvų atžvilgiu, taip pat buvo rasta ir fotografija atitinkanti nupieštą vaizdą, tačiau bendrai vertinant paieškos rezultatą ji buvo beprasmis. Atlikus paiešką pagal raktinius žodžius taip pat buvo rasta nemažai atitinkančių jų prasmę vaizdų, bet kaip ir pirmuoju atveju, buvo taip pat daug visiškai juos neatitinkančių.

Trečio vaizdo paieška kaip ir pirmo buvo sėkminga. Buvo rastas didelis kiekis lygiai tokių pat arba panašių vaizdų. Tačiau atlikus paiešką pagal raktinius žodžius buvo rasta labai nedaug atitinkančių paiešką rezultatų.

1.1.3. Imense image search portal

Šis portalas leidžia atlikti vaizdų paiešką pagal raktinius žodžius ir taip pat leidžia atsisiųsti įrankį leidžiantį atlikti paiešką pagal pavyzdinį vaizdą (Similar search plugin for adobe cs4 bridge). Vaizdai, kuriuos galima rasti šiame puslapyje yra sužymėti panaudojant „Imense autotagger“, kuris analizuoja vaizdus ir priskiria jiems žymes. Sukurtas Imense kompanijos ir išleistas 2008 metais. Atliktų testų rezultatai buvo:



Pav. 5 Imense paieškos rezultato vaizdas.

Pirmos paieškos pagal raktinius žodžius rezultatai buvo geri. Didžioji dalis vaizdų pilnai atitiko raktinius žodžius ir buvo rasti vaizdai panašūs į Pav.1.

Antros paieškos rezultatai gerai atitiko raktinių žodžių prasmę

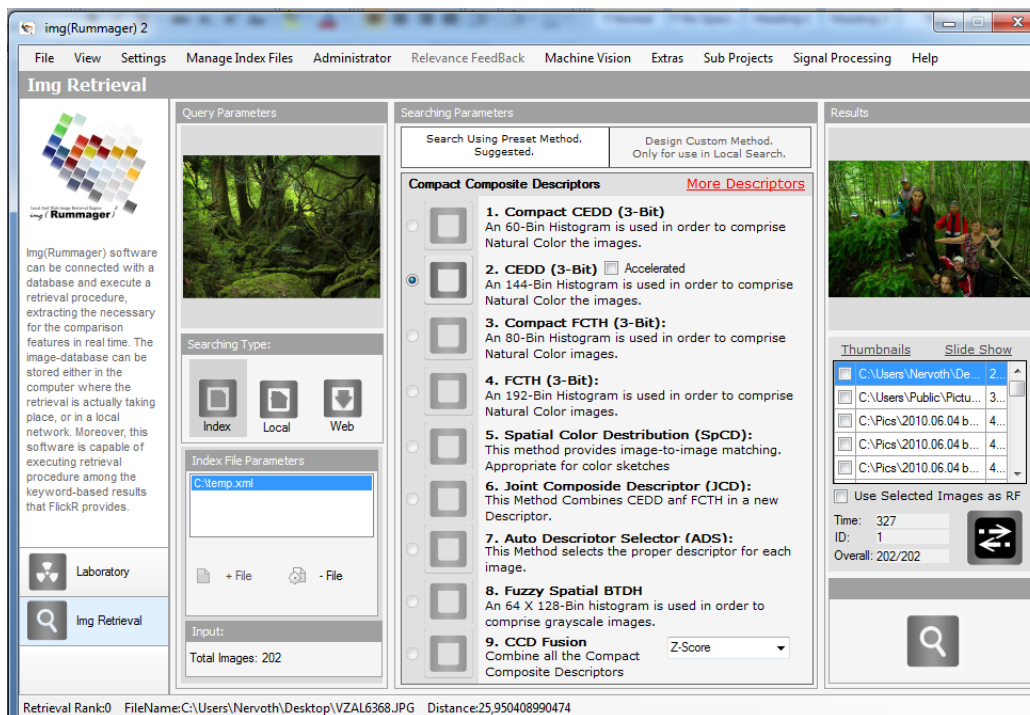
Visi trečios paieškos rezultatai pilnai atitiko raktinius žodžius.

Dėl to, kad atlikti Similar search įskiepio testo nepavyko, negalima komentuoti paieškos pagal pavyzdinį vaizdą.

1.1.4. Img(Rummager)

Tai programa skirta namų kompiuteriams, kuri gali atlikti vaizdų analizę ir paiešką pagal turinį. Ją sukūrė Savvas A. Chatzichristofis. Ši programa naudoja daug atviro kodo bibliotekų (OpenCV, Aforge.net ir kt.). Dėl to, kad ji pritaikyta darbui su lokaliais arba tinklo resursais, tikros paieškos internete nėra atliekamos (yra galimybė atlikti paiešką pagal raktinius žodžius ir vaizdą Flickr svetainėje, bet tokių paieškų rezultatai yra netikslūs). Tam, kad atlikti testą pirma buvo sukurtas 200 skirtingų vaizdų indeksų failas (indeksavimo metu visi failai buvo išanalizuoti programa) ir tuomet buvo atliekama testinių vaizdų paieška.

Pirmas ir antras vaizdai buvo tarp indeksuojamų vaizdų rinkinio. Jie buvo rasti ir nurodyti kaip pats panašiausias vaizdas į ieškomą. Trečio vaizdo nebuvo tarp indeksuojamų, bet buvo panašų į jį. Šios paieškos geriausią rezultatą galima pamatyti Pav. 6.



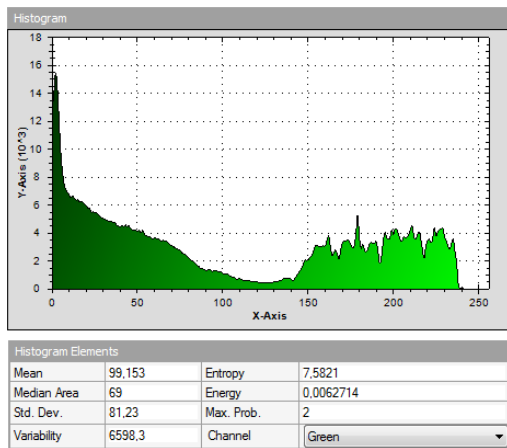
Pav. 6 img(Rummager) programos paieškos rezultatai su trečiu testiniu vaizdu.

1.2. Vaizdų analizės metodai

Kai kalba eina apie vaizdų paieškos pagal turinį sistemas, reikia išskirti du skirtingus tipus pirmas – paieška pagal pavyzdinį vaizdą(angl. content based image retrieval), antras – paieška pagal raktinius žodžius(angl. automatic image annotation and retrieval). Pirmuoju atveju vaizdai ieškomi pagal tai kiek spalvos, formos arba tekstūros yra panašios, paieškos imties vaizduose, į pavyzdinį vaizdą. Antruoju atveju visi paieškos imties vaizdai yra automatiškai sužymimi pagal tai, kokie objektai ten rasti, kokios dominuojančios spalvos ir kita. Atlikus vaizdų žymėjimą, paieška vyksta taip pat kaip ir įprastuose paieškos sistemose, t.y pagal raktinius žodžius arba taip pat sužymėjus pavyzdinį vaizdą.

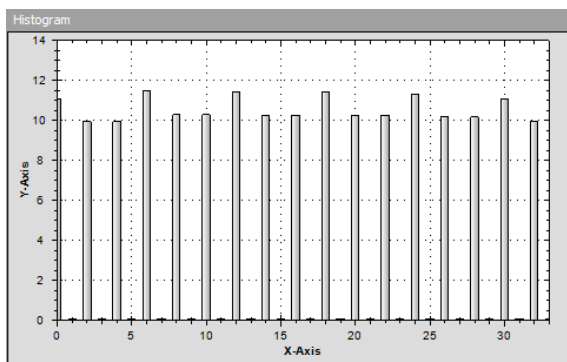
Atliekant paiešką pagal pavyzdį dažniausiai naudojamas vaizdų lyginimas pagal vaizdų nuotolį vienas nuo kito. Šis metodas lygina vaizdų panašumus įvairiuose dimensijose pavyzdžiui spalva, tekstūra ir kitos. Kuo nuotolis yra mažesnis, tuo vaizdai yra panašesnis.

Kai tikrinamas nuotolis pagal spalvas, yra sukuriama vaizdų spalvų histograma(Pav.7), kuri nurodo kiek kokių spalvų yra vaizduose. Sudėtingesni metodai lygina spalvų histogramas pagal pozicijas vaizde arba tarpusavyje. Tačiau yra ir kitų būdų kaip lyginti vaizdus pagal spalvas neskaitant histogramų.



Pav. 7 Vaizdo RGB spalvų histograma, žalios spalvos atžvilgiu.

Kai vaizdai lyginami pagal tekstūrą, ieškomi tekstūros elementai (angl. texels) ir kaip jie išdėstyti. Surinkus tokių elementų rinkinius ir nustačius kokioms klasifikacijom jie priskiriami, galima nustatyti panašumą tarp vaizdų. Tokie rinkiniai gali būti atvaizduojami tiek matricomis, tiek histogramomis (pav. 8).



Pav. 8 Gabor tekstūrų histograma.

Kitas būdas yra palyginimas pagal formą. Jos nustatomos panaudojus segmentaciją, kontūrų paiešką arba formų filtrus. Yra ir nemažai kitų būdų palyginti atvaizdus tarpusavyje, pavyzdžiui, pasinaudojus kitais filtrais, transformacijomis ir kitais būdais.



Pav. 9 Kontūrų išskyrimas.

Atliekant paiešką pagal raktinius žodžius svarbi yra automatinė vaizdų anotacija. Ji yra pagrindinis dalykas išskiriantis tokias paieškos pagal turinį sistemas nuo paprastų paieškos pagal raktinius žodžius sistemas. Šiuo atveju vaizdams yra automatiškai generuojami metaduomenys (raktiniai žodžiai). Tokias sistemas galima įsivaizduoti kaip klasifikatorius, kurie išskiria tam tikras ypatybes vaizduose ir priskiria juos prie tam tikrų klasių (raktinių žodžių). Būtent čia naudojami skirtingi dirbtinio intelekto metodai, nes sistemą reikia apmokyti atpažinti ypatybes ir priskirti tam tikrus raktinius žodžius. Čia gali būti naudojami labai skirtingi dirbtinio intelekto metodai, pavyzdžiui: genetiniai algoritmai, dirbtiniai neuronų tinklai, klasterizavimas ir kita. Pagrindinis tokių paieškos sistemų trūkumas yra ilgas ir sudėtingas sistemos apmokymas, tačiau jos rezultatai gali būti pranašesni už ankščiau minėtą paiešką pagal pavyzdį.

Top 15 Computer-Predicted Tags
ALIPR is like a child trying to learn about the world. Please help us to teach ALIPR. Check those correctly annotated words.

landscape mountain valley rock_form animal
 plant people wild canyon molecule
 snow ski winter texture natural

Thought of other terms missed by ALIPR? Please add here, separated by commas ''
 and make the picture searchable

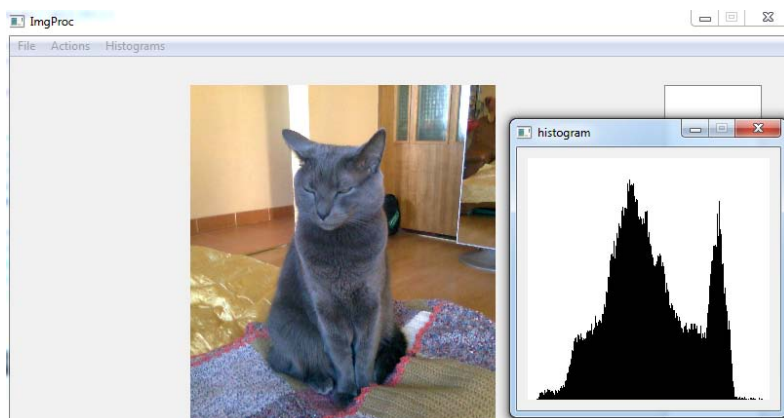
Optional information:
Picture title
URL to see related pictures
Copyright (hypertext ok)

Pav. 10 Alipr automatinės vaizdų anotacijos pavyzdys.

1.2.1. Histogramos ir jų panaudojimas vaizdų analizėje

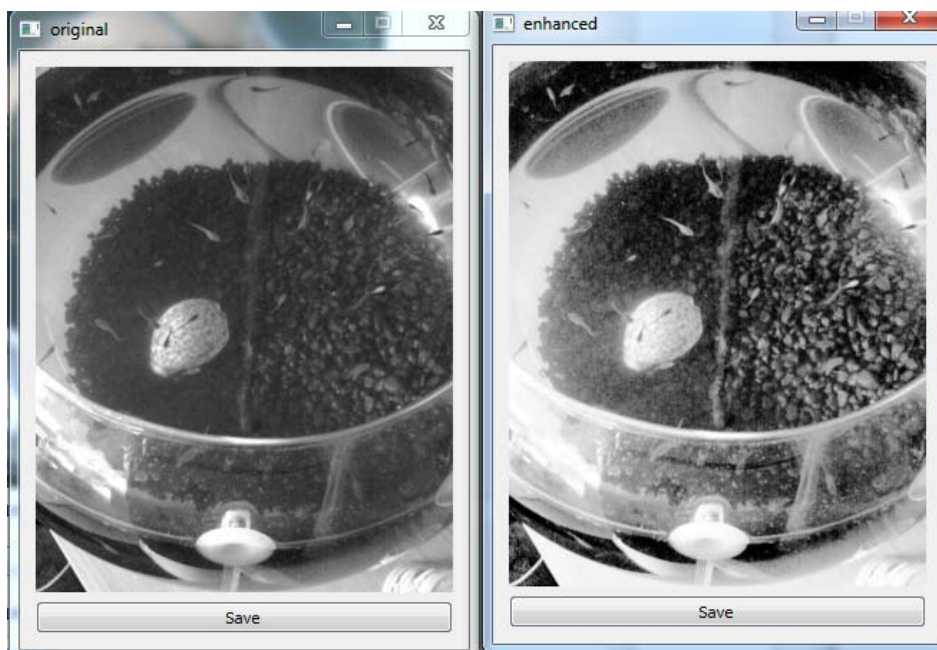
Skaitmeninė fotografija yra sudaryta iš skirtingų spalvų pikselių. Jų pasiskirstymas fotografijoje yra labai svarbi charakteristika. Ji gali būti naudojama tam, kad surasti specifinį turinį fotografijoje, ja manipuluoti ir kt.

Spalvų histograma – lentelė, kurioje aprašytas pikselių, turinčių konkrečią spalvą, skaičius. Pustonių histograma turės 256 įrašus. 0 įrašas parodo kiek pikselių turi reikšmę 0, 1 įrašas – pikselių skaičius su reikšme 1 ir t.t.



Pav. 11 Spalvotas vaizdas, ir jo pustonių histograma

Paprasčiausi histogramų panaudojimo būdai yra vaizdų transformacija. Pavyzdžiui, vaizdo kontrasto pagerinimas. Pav. 12 mes matome, kad ne visas spalvų ryškumo spektras yra naudojamas. Mes galime ištempti histogramą, taip pagerinant vaizdo kontrastą.



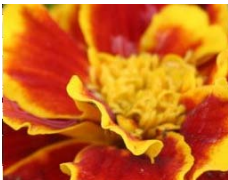
Pav. 12 Pradinis vaizdas kairėje, transformuotas dešinėje.

Mums reikia rasti minimalią ir maksimalią ryškumo reikšmes. Po to mes galime pertvarkyti ryškumo reikšmes taip, kad minimali reikšmė būtų 0, o maksimali 255. Visos kitos reikšmės bus pakeičiamos tiesiškai.

1.2.2. Galinis histogramos projektavimas (*angl. Backprojecting*).

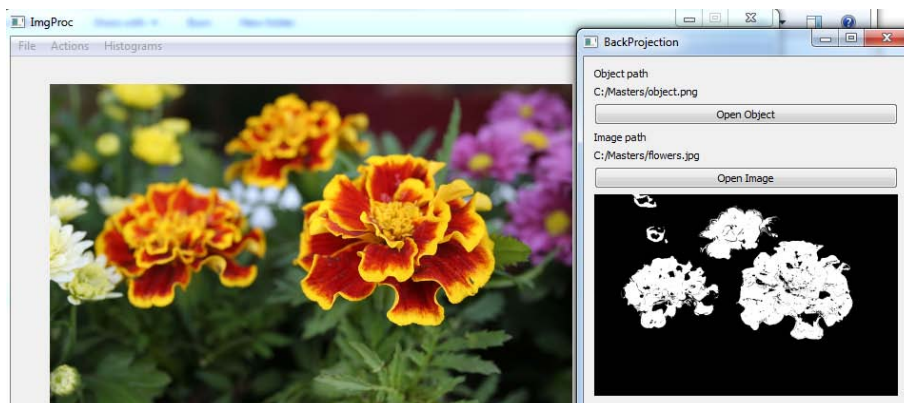
Histograma taip pat gali būti vaizdo turinio charakteristika. Jeigu mes paimsime vaizdo dalį, kur rodoma tekstūra arba objektas, tai šios vaizdo dalies histograma gali būti naudojama kaip funkcija nurodanti tikimybę su kuria pasirinktas pikselis jai priklauso. Tai galima atlikti naudojant galinį histogramos projektavimą. Bendras algoritmo aprašymas būtų toks:

1. Nustatome slenkstį, kuris bus naudojamas tam, kad parodyti binarinį žemėlapią rodantį paieškos rezultatus.



Pav. 13 Ieškomo objekto pavyzdys

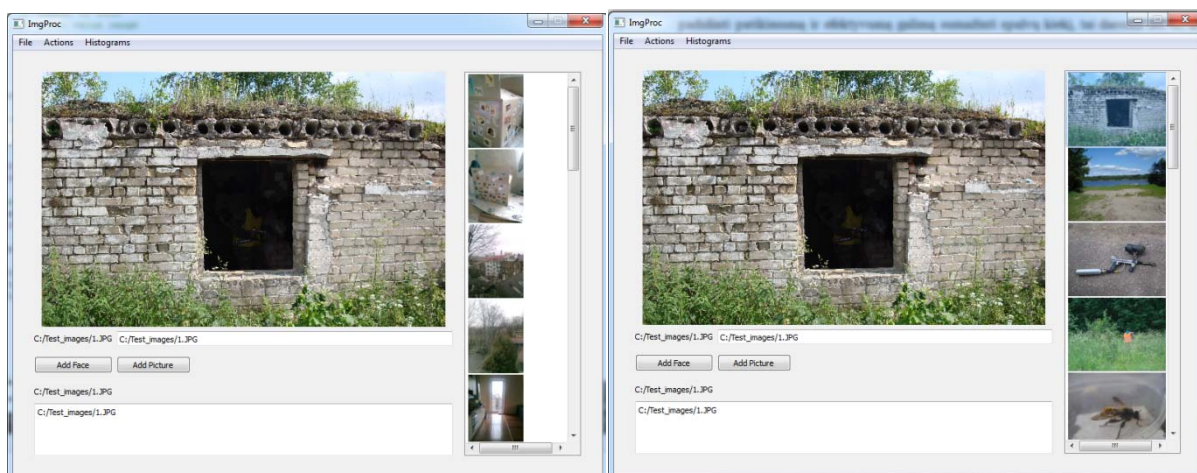
2. Randame ir normalizuojame ieškomo objekto (Pav.13) arba tekstūros histogramą.
3. Atliekame galinę histogramos projekciją į vaizdą, kuriame atliekame paiešką (Pav. 14). Kiekvienas pikselis vaizde keičiamas atitinkama tikimybės reikšme iš normalizuotos ieškomo objekto histogramos.
4. Panaudojame slenksčio parametrus, kad sukurti binarinį vaizdą (Pav. 14). Balti pikseliai parodo vietas kur buvo rastas nurodytas turinys.



Pav. 14 Kairėje vaizdas, kuriame atliekame paiešką, dešinėje – galinės projekcijos rezultatas.

1.2.3. Panašių vaizdų paieška naudojant histogramų palyginimą.

Vaizdų paieška pagrįsta turiniu yra svarbi problema kompiuterinio vaizdo apdorojime. Ji susideda iš vaizdų rinkinio, kuris turi panašų turinį į užklauso vaizdą, paieškos. Kaip mes matėme iš prieš tai buvusių pavyzdžių, histogramos gana efektyviai aprašo vaizdo turinį, dėl to mes galime jas naudoti panašių vaizdų paieškai. Svarbiausia yra tai, kad suskaičiuoti vaizdų panašumui mums reikia palyginti tik vaizdų histogramas. Tam, kad padidinti patikimumą ir efektyvumą galimą sumažinti spalvų kiekį, tai daroma dėl to, kad didžioji dauguma histogramų palyginimo metodų lygina tik konkrečios spalvos reikšmes ir nežiūri į kitas šalia esančias panašias spalvas. Yra kelios funkcijos, kurios naudojamos nustatyti panašumui: Koreliacijos, Chi-kvadrato, sankirtos, Bhattacharyya nuotolis ir kt.



Pav. 15 Vaizdo paieška naudojant histogramų palyginimą. Kairėje pradinis nesurūšiuotas sąrašas ir vaizdas, dešinėje surūšiuotas sąrašas po palyginimo.

1.3. Objektų paieška ir atpažinimas.

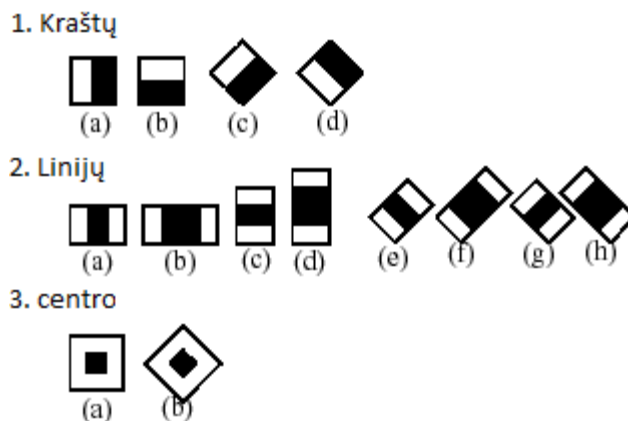
Prieš tai minėti objektų paieškos būdais galima atlikti konkrečių objektų paiešką, bet jie netinka objektų klasės paieškai. Gauti rezultatai tinka tik labai bendrai vaizdų paieškai. Jeigu mes norime rasti ir atpažinti konkrečius objektus mums reikės pasinaudoti kitais metodais. Visų pirma mums reikės išskirti bruožus, kurie apibūdina objektų klasę ir kuriuos mes naudosime tam, kad atpažinti kuriai klasei priklauso rastas objektas. Šiam tikslui pasiekti mes turime nemažai metodų. Kai kurie iš jų bus aprašyti toliau, taip pat bus pateiktos šių metodų realizacijos ir analizė. Bet prieš pradėdant objekto atpažinimą, mums vaizde reikia surasti objektą, su kuriuo vyks tolimesni veiksmai.

Objekto paieškai mes naudosime kaskadomis padidintą klasifikatorių(angl. cascade boosted classifier). Šis metodas naudos dviejų tipų bruožų identifikavimą – Haar bruožus ir lokalius binarinius šablonus.

Kaskadomis padidintas klasifikatorius, tai sprendimų medis, su bent dviem lapais. Kiekvienas išplaukiantis klasifikatorius susideda iš paprastesnių klasifikatorių, kurie yra iš eilės pritaikomi mus dominančiam vaizdo regionui, dėl to jis vadinamas kaskadiniu klasifikatoriumi. Padidintas (angl. boosted) – reiškia tai, kad kiekvieno žingsnio klasifikatoriai yra sudėtingi. Jie yra sukuriami iš bazinių klasifikatorių panaudojant vieną iš padidinimo(angl. boosting) metodų. Mūsų atveju Haar tipo bruožai ir lokalūs binariniai šablonai yra įvesties duomenys baziniams klasifikatoriams. Tam, kad apmokyti klasifikatorių, reikia kelių didelių duomenų rinkinių:

- Pozityvių vaizdų rinkinys – vaizdai, kuriuose yra tik tas objektas, kuris priklauso klasei, kurią mes norime išmokyti atpažinti;
- Negatyvių vaizdų rinkinys – vaizdai, kuriuose nėra ieškomų objektų klasės;

Natūralių vaizdų rinkinys – vaizdai, kuriuose ieškomos objektų klasės egzempliorius yra fone.



Pav. 16 Haar tipo bruožai naudojami OpenCV klasifikatoriuje.

Haar tipo bruožai sudaryti iš dviejų arba trijų sujungtų juodų ir baltų keturkampių(Pav. 16). Haar tipo bruožo reikšmė yra skirtumas tarp pikselių pustomių sumos juodame ir baltame keturkampių regionuose. Palyginus su neapdirbtomis pikselių reikšmėmis, jie gali sumažinti arba padidinti klasių arba jų vidinių objektų skirtumų kintamumą, tokiu būdu palengvinant klasifikaciją.

Šie bruožai gali būti labai greitai paskaičiuoti naudojant vientisus vaizdus. Toks vaizdas taške $P(x,y)$ yra visų pikselių virš ir į kairę nuo savęs (įskaitant pikseli(x,y)) suma.

Tam, kad rasti objektą vaizde, jis yra tikrinamas langu, kuriame yra Haar tipo bruožas. Šis bruožas yra silpnas klasifikatorius, paremtas visais naudojamais klasifikatoriais. Kai jis randa tinkamą vaizdo vietą, pradedama tikrinti naudojant kitus klasifikatorius.

Pagrindinė darbo dalis bus skirta veidų atpažinimui (kaip vieno iš objektų klasių atpažinimo atveju). Toliau darbe bus aprašyti veidų atpažinimo metodai, jie bus realizuoti ir išanalizuoti.

1.3.1. Veidų atpažinimo metodai

Svarbi vaizdų analizės dalis yra veidų atpažinimas, tai yra sekantis žingsnis po objekto aptikimo ir turi savo atskirus metodus.

Statistiniuose metoduose, kiekvienas vaizdas yra atvaizduojamas d ypatybių požiūriu. Todėl į jį yra žiūrima kaip į vektorių d -matėje erdvėje. Dėl to, kad šių duomenų dimensija yra labai didelė, reikia parinkti teisingus statistinius įrankius tam, kad išgauti ir išanalizuoti pagrindinę daugdarą. Šie įrankiai turi apibrėžti veido erdvę vaizde ir išgauti reikiamus duomenis. Kas po to leistų atpažinti linijas, kreives, plokštumas ir hiperplokštumas, kurios atskiria veidus priklausančius skirtingoms klasėms.

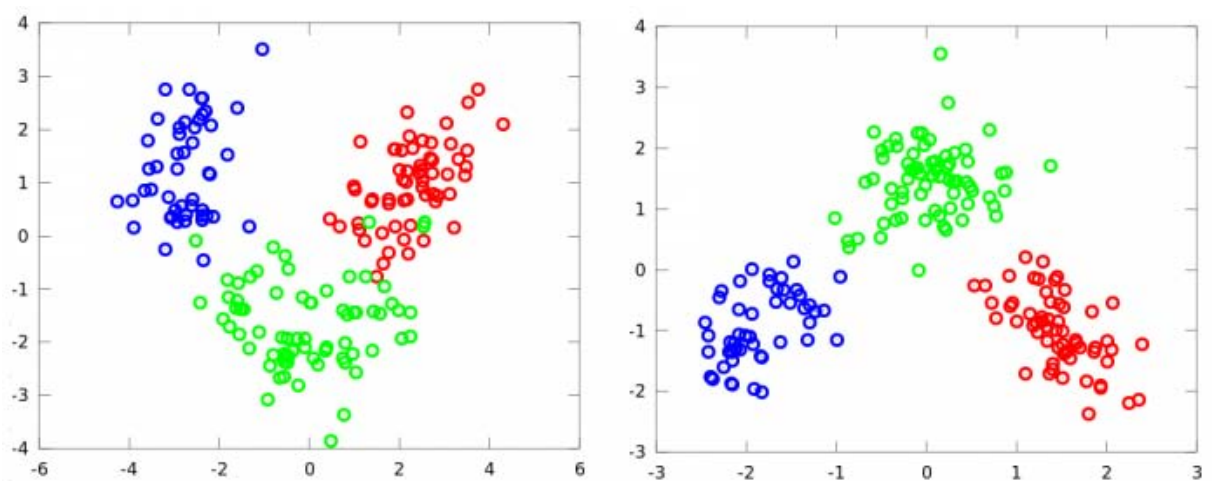
1.3.2. Tikriniai veidai (*angl. Eigenfaces*)

Vienas dažniausiai naudojamų statistinių metodų yra pagrindinių komponentų analizė (toliau PKA). Pagrindinių komponentų analizė yra būdas surasti šablonus duomenyse ir atvaizduoti duomenis pabrėžiant jų skirtumus ir panašumus. Dėl to, kad rasti šablonus dideliame duomenų kiekyje, kur grafiškai atvaizduoti juos yra sudėtinga, yra sunku, PKA yra labai galingas įrankis duomenų analizėje. Viena iš to priežasčių yra ta, kad suradus šablonus jų dydį galima sumažinti be didelių informacijos praradimų.

Vienas žinomiausių metodų naudojantis PKA yra tikrinių veidų metodas (L. Sirovich ir M. Kirby 1987m). Šis metodas naudoja nedidelį rinkinį bruožų, kuriais galima apibūdinti skirtumą tarp veidų. Tikslas yra rasti kovariacinės pasiskirstymo matricos tikrinius vektorius (*angl. eigenvectors*), kitaip vadinamus tikriniais veidais (*angl. eigenfaces*), naudojant apmokomąjį veidų vaizdų rinkinį. Kiekvienas veidas yra atvaizduojamas gautų tikrinių vektorių tiesine kombinacija. Atpažinimas yra atliekamas projektuojant naują vaizdą į tikrinių veidų aibę ir klasifikuojant veidą lyginant jo poziciją veidų erdvėje su žinomų veidų pozicijomis.

Tikrinių veidų atpažinimo metodo algoritmas:

- Sukuriame žinomų asmenų veidų fotografijų M dydžio rinkinį. Jis turėtų būti sudarytas iš kelių kiekvieno asmens fotografijų, besiskiriančių veido išraiškomis ir apšvietimu;
- Suskaičiuojame $M \times M$ dydžio matrica L , randame jos tikrinius vektorius, reikšmes ir pasirenkame tik tuos tikrinius veidus, kurių tikrinės reikšmės yra didžiausios.
- Sujungiamo normalizuotus apmokymo rinkinius, kad sukurti tikrinių veidų aibę.
- Kiekvienam žinomam asmeniui apskaičiuojame klasės vektorius, išlygindami tikrinių veidų šabloną, gautą iš pradinių asmens veido vaizdų. Parenkame slenkstį, kuris aprašo maksimalų leidžiamą nuotolį tarp veidų klasių ir slenkstį, kuris nurodo maksimalų nuotolį nuo veidų erdvės.
- Kiekvienam naujam veidui, kurį mes norime identifikuoti, suskaičiuojame šablono vektorius, nuotolį iki kiekvienos žinomos klasės ir nuotolį iki veidų erdvės. Jeigu atstumas iki kokios nors iš žinomų asmenų klasės yra mažesnis už nurodytą slenkstį, identifikuojame naują veidą kaip tą asmenį. Jeigu tokių nėra, bet atstumas iki veidų erdvės yra mažesnis už nurodytą slenkstį, mes turime nežinomą veidą. Priešingu atveju, objektas yra ne veidas.



Pav. 17 Normalizuoti klasių pasiskirstymo grafai. Kairėje tikrinių veidų, dešinėje Fišerio veidų.

1.3.3. Fišerio veidai (*angl. Fisherface*)

Dar vienas statistinis metodas naudojamas veidų atpažinime yra tiesinių diskriminantų analizė (*angl. Linear Discriminant Analysis*). Šis metodas naudojamas tam, kad rasti tiesines ypatybių kombinacijas, išlaikant klasių nepriklausomumą.

Fišerio veidų metodas yra vienas iš veidų atpažinimo metodų, naudojančių tiesinę diskriminantų analizę. Pagrindinis skirtumas nuo tikrinių veidų metodo yra tas, kad Fišerio veidų metodas bando maksimaliai padidinti skirtumas tarp skirtingų klasių ir maksimaliai sumažinti skirtumą tarp klasės vidinių objektų (Pav.17).

Bendras Fišerio veidų algoritmas:

1. Sukuriame veidų vaizdų matricą X , kurios kiekvienas stulpelis atitinka vienam vaizdui. Kiekvienas vaizdas yra priskiriamas atitinkamai vektoriaus C klasei.
2. Projektuojame matricą X į $(N-c)$ -matę erdvę kaip P , su rotacine matrica W_{pca} , kuri buvo rasta panaudojant PKA. Čia N – pavyzdžių skaičius X matricoje, o c – unikalių klasių skaičius.
3. Suskaičiuojame projekcijos P išsibarstymą tarp klasių – S_b :

$$S_b = \sum_{i=1}^c \left[N_i \cdot (\text{mean}_i - \text{mean}_p) \cdot (\text{mean}_i - \text{mean}_p)^T \right]$$

Čia:

Mean_p – bendras P vidurkis.

Mean – klasės i vidurkis P erdvėje.

N – pavyzdžių skaičius klasėje i .

4. Suskaičiuojame vidinių klasės objektų išsibarstymą P , kaip S_w :

$$S_w = \sum_{i=1}^c \sum_{X_k \in X} \left[(X_k - \text{mean}_p) \cdot (X_k - \text{mean}_p)^T \right]$$

Čia:

X – klasės i pavyzdžiai.

x_k – X pavyzdys.

5. Pritaikome standartinę teisinio diskriminanto analizę ir maksimizuojame santykį tarp klasių išsibarstymo ir klasių vidinių objektų išsibarstymo. Atsakymas bus rinkinys apibendrintų tikrinių vektorių W_{fld} , sudarytų iš S_b ir S_w atitinkančių savo tikrinėms reikšmėms.
6. Randame Fischerio veidus $W = W_{pca} * W_{fld}$.

1.3.4. Lokalių binarinių šablonų histograma

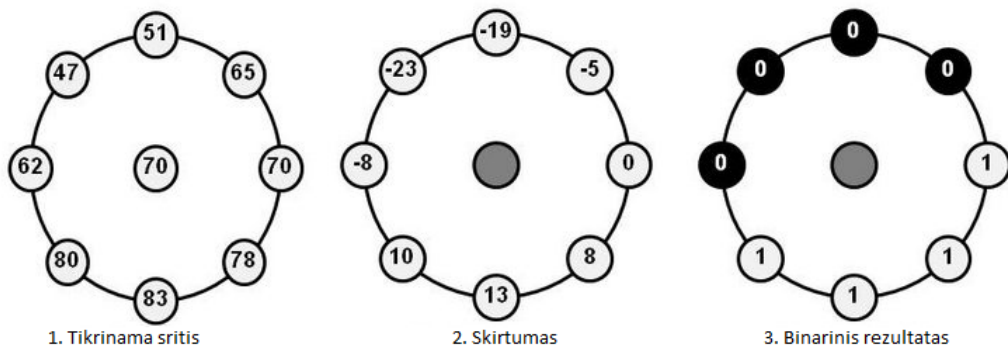
Lokalus binarinis šablonas (*angl. Local binary pattern*) – paprastas, bet labai efektyvus tekstūrų operatorius, kuris žymi vaizdo pikselius tikrinant slenkstines gretimų pikselių reikšmes ir skaityti rezultatą kaip binarinį skaičių. Dėl šio metodo skiriamųjų sugebėjimų ir skaičiavimo lengvumo, šis operatorius tapo labai populiariu įrankiu skirtingose srityse. Viena svarbiausių jo ypatybių yra atsparumas pustonių pasikeitimui, pavyzdžiui, dėl apšvietimo svyravimo. Dėl skaičiavimo patogumo lokalaus binarinio šablono operatorius leidžia gana lengvai atlikti vaizdų analizę realiu laiku.

LBP pikselio reikšmė yra skaičiuojama taip:

Mes turime pikselį (x_c, y_c) . Tuomet to taško LBP reikšmė bus:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad s(x) = \begin{cases} 1, & \text{if } x \geq 0; \\ 0, & \text{otherwise.} \end{cases}$$

Čia P yra atrenkami pikseliai apskritime spinduliu R.



Pav. 18 LBP skaičiavimas

Pav. 18 pavaizduoto pikselio galutinis rezultatas būtų:

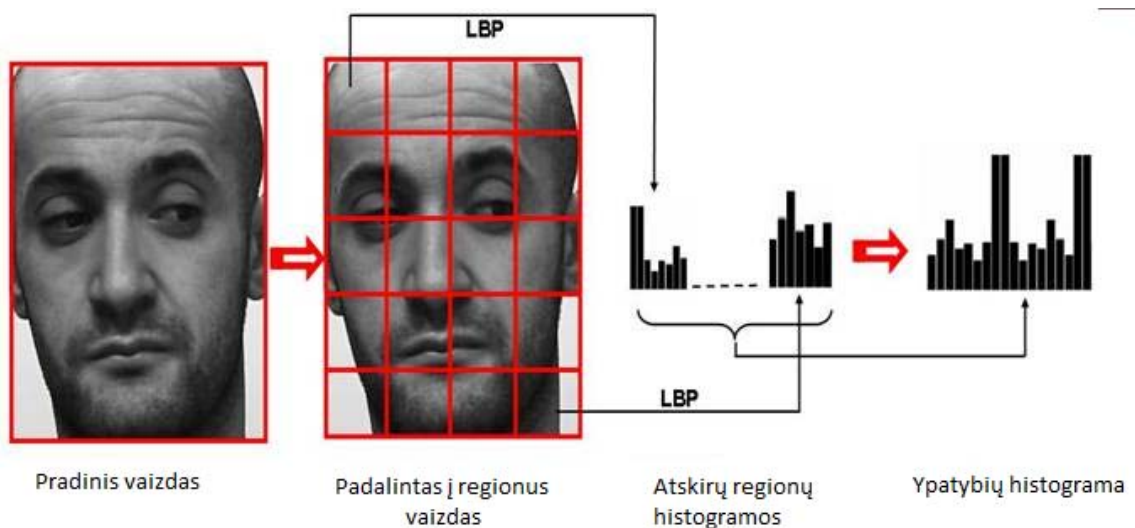
$$1*1+1*2+1*4+1*8+0*16+0*32+0*64+0*128 = 15$$

LBP yra naudojami uniforminiai šablonai. Jie gali būti naudojami tam, kad sumažinti ypatybių vektorius dydį. Lokalus binarinis šablonas yra vadinamas uniforminiu, jeigu jame yra ne daugiau 2 bitinių perėjimų iš 0 į 1 arba atvirkščiai, kai bitų seka yra peržiūrima ratu, pvz. : 00000000 (0 perėjimų), 011111000 (2 perėjimai) – šie šablonai yra uniforminiai, 110011001 (4 perėjimai), 01010010 (6 perėjimai) – ne uniforminiai šablonai. Tokiu atveju žymint šablonus, kiekvienam unikaliai uniforminiam šablonui priskiriama atskira žymė, o visi likę šablonai žymimi viena bendra žyme. Pavyzdžiui, jeigu mes nauduosime $R = 8$, yra 256

galimi šablonai, 58 iš jų – uniforminiai. Tokiu būdu mes turėsime 59 skirtingas žymes. Kai mes turime LBP sužymėtą vaizdą $f(x,y)$, mes galime apskaičiuoti LBP histogramą:

$$H_i = \sum_{x,y} I\{f_L(x,y) = i\}, i = 0, \dots, n - 1,$$

Čia n – skirtingų žymių skaičius, f_L – pikselio reikšmė suskaičiuota LBP.



Pav. 19 LBP ypatybių histogramos sukūrimas.

Panaudojant šias histogramas, vėliau galima atlikti objektų paieškos ir atpažinimo veiksmus.

2. Projektinė dalis

Kiekvienam uždaviniui spręsti reikia sukurti arba rasti atitinkamus įrankius. Kuriant fotografijų analizės programų sistemą gali tekti susidurti su daug užduočių, pavyzdžiui, sistemos apmokymų atpažinti objektus arba veidus ir atlikti kitus vaizdų analizės veiksmus. Tam, kad nereikėtų realizuoti visus metodus ir funkcijas nuo nulio verta pasinaudoti jau turimomis bibliotekomis. Keletas tokių bibliotekų būtų OpenCV, IVT, AForge.Net. Toliau bus pateikti trumpi bibliotekų ir jų galimybių aprašymai.

2.1.OpenCV 2.4 biblioteka

OpenCV tai atviro kodo biblioteka. Ji platinama pagal BSD licenzija, kas leidžia ją naudoti tiek moksliniuose darbuose tiek komerciniuose. OpenCV biblioteka buvo pradėta kurti Intel korporacijoje 1999 metais Gario Bradski. Pirmoji alfa versija buvo pristatyta IEEE konferencijoje (IEEE Conference on Computer Vision and Pattern Recognition) 2000 metais. OpenCV 1.0 buvo išleista 2006 metais. 2008 metais kitos korporacijos (pagrindinė iš jų –

Willow Garage) pradėjo remti OpenCV projektą ir buvo išleista 1.1 versija. 2009 metais buvo išleista antra šios bibliotekos versija, kurioje buvo pakeistas interfeisas ir pagerintas bibliotekos darbas. Šiuo metu yra platinama OpenCV 2.4 versija.

Ši biblioteka gali būti taikoma: objektų atpažinimui, veidų ir gestų atpažinimui, judesių sekimui, segmentacijai ir kitur. Joje yra per 500 skirtingų funkcijų, kuriuos leidžia atlikti didžiąją dalį veiksmų reikalingų realaus laiko kompiuterinio matymo (*angl. computer vision*) realizavimui. Ši biblioteka turi funkcijas:

- Atminties tvarkymui, algebrinės operacijas(su matricomis ir kt.), klaidų tvarkymas ir siūlų (*angl. threads*) palaikymas prareikus paralelinių skaičiavimų.
- Statinių ir dinaminių struktūrų ir veiksmų su jomis atlikimas, specialios piešimo funkcijos.
- Apmokymo funkcijos (*angl. machine learning*): statistiniai modelių realizacijos, sprendimų medžiai, neuronų tinklai ir kitų metodų realizacijos
- Taip pat yra funkcijos interfeiso kūrimui, multimedijos įrenginių duomenų įvedimui ir išvedimui ir kt.

Ši biblioteka gali būti naudojama Windows, Linux ir MacOS operacinėse sistemose ir gali būti įdiegta į kelias skirtingas programavimo aplinkas: Visual C++, Eclipse IDE, C++ builder, DevCpp IDE. Tai leidžia palengvinti darbą nes nereikia priprasti prie naujų aplinkų. Taip pat galima sukompiliuoti ir pritaikyti šią biblioteka ir kitoms programavimo aplinkoms, kurios palaiko Cmake.

Kitas šios bibliotekos bruožas yra tas, kad ją gali naudoti programuotojai dirbantys skirtingomis kalbomis. Ši biblioteka turi C++,C ir python versijas, taip pat šiuo metu daromas java interfeisas, kurį galima bus naudoti su Android SDK.

Šia biblioteka naudojasi ir padeda ja vystyti nemažas kiekis kompanijų: Itseez West, Vizmatic, Willow garage , Zeitera ir kitos(yra minima, kad Google yra viena iš šios bibliotekos vartotojų) ir kitos. Neskaitant įmonių ją naudoja didelis kiekis mokslininkų, studentų ir privačių asmenų.

Dėl to, kad ši biblioteka yra gana plačiai naudojama ji turi gerą dokumentaciją, daug pavyzdžių padedančių susipažinti su šios bibliotekos funkcijos, taip pat yra išleista keletas knygų apie darbą su OpenCV.

2.2. Integrating Vision Toolkit

IVT tai kompiuterinio vaizdo (*angl. computer vision*) biblioteka su objektiškai orientuota architektūra. Ji buvo sukurta Karlsruhe institute. Pirmoji jos versija buvo išleista 2005 metais. Nuo 2009 metų prie bibliotekos palaikymo prisidėjo Keyetech kompanija, kuri siūlo apmokymus ir komercinius sprendimus pagrįstus IVT. Naujausia IVT versija – IVT 1.3.6, buvo išleista 2011 metais.

Kaip ir OpenCV ši biblioteka gali būti taikoma daugelyje skirtingų sričių ir jos teikiamos galimybės yra labai panašios. Teigiama, kad ji veikia taip pat greitai kaip ir OpenCV(arba greičiau), bet jos funkcionalumas yra mažesnis(dalis funkcijų yra pateikiamos kaip OpenCV funkcijų apvalkalai). Jos teikiamos funkcijos:

- Vaizdų apdorojimas
- Integruotas GUI kūrimas(Windows, MacOS, Linux ir QT įrankiais)
- Objektų sekimo algoritmai
- Įvairūs matematiniai algoritmai ir struktūros
- Ypatybių paieškos, išskyrimo ir atvaizdavimo funkcijos ir kt.

Ši biblioteka yra multi platforminė, ir gali būti naudojama su bet kuria operacine sistema palaikančia C++ kompiliatorius. Ji naudoja C++ kalbą ir nesunkiai suderinama su skirtingomis programavimo aplinkomis.

Ji turi pilną dokumentaciją sugeneruotą su DoxiGen kuri pateikia aiškius funkcijų aprašymus, o pavyzdžių rinkinys parodo, kaip naudotis biblioteka. Keyetech taip pat turi komercinę šios bibliotekos versiją, kuri turi optimizuotus algoritmus skirtingoms platformoms ir kuri yra automatiškai naudojama IVT jeigu yra įdiegta kompiuteryje.

2.3. AForge.Net Framework

Priešingai nei IVT ir OpenCV, AForge.Net yra bibliotekų rinkinys. Jis yra sukurtas su C# ir pritaikytas naudojimui su .Net. Paskutinė AForge.Net versija buvo išleista 2012 metais – ver.2.2.4.

AForge.Net kaip ir dvi ankstesnės bibliotekos suteikia daug galimybių ir kalbant apie dirbtinio intelekto dalį suteikia net daugiau negu OpenCV ar IVT. AForge.Net bibliotekų funkcijos:

- AForge.Imaging biblioteka – didžiausia AForge.Net biblioteka. Suteikia galimybes apdoroti, pagerinti ir transformuoti vaizdus pavyzdžiui: skirtingi filtrai, spalvų koregavimo filtrai, tekstūrų generatoriai ir filtrai ir kiti
- AForge.Vision biblioteka – suteikia skirtingas funkcijas tam, kad atlikti judesių atpažinimą ir analizę.
- AForge.Video biblioteka – suteikia galimybę dirbti su video duomenimis.
- AForge.Neuro biblioteka – pateikia paprastesnių neuronų tinklų architektūrų ir jų apmokymo algoritmų realizacijas.
- AForge.Genetic biblioteka – pateikia funkcijas, kurios padeda spręsti įvairius genetinių algoritmų, programavimo uždavotius (chromosomų realizacijos, atrinkimo metodai ir kita).
- Taip pat yra ir kitų su dirbtinių intelektų susijusių bibliotekų arba bibliotekų teikiančių pagalbines funkcijas.

AForge.Net yra pastoviai tobulinamas ir turi gerą dokumentaciją. Taip pat yra labai daug gerai aprašytų pavyzdžių kaip kurti skirtingas programas pasinaudojant šio įrankio galimybėmis.

Kaip galime matyti, AForge.Net funkcinės galimybės yra platesnės nei kitų dviejų bibliotekų. Tačiau dėl to, kad ji gali būti naudojama tik su .Net, jos pritaikymo galimybės yra šiek tiek mažesnės nei kitų.

2.4.Įrankių pasirinkimo analizė

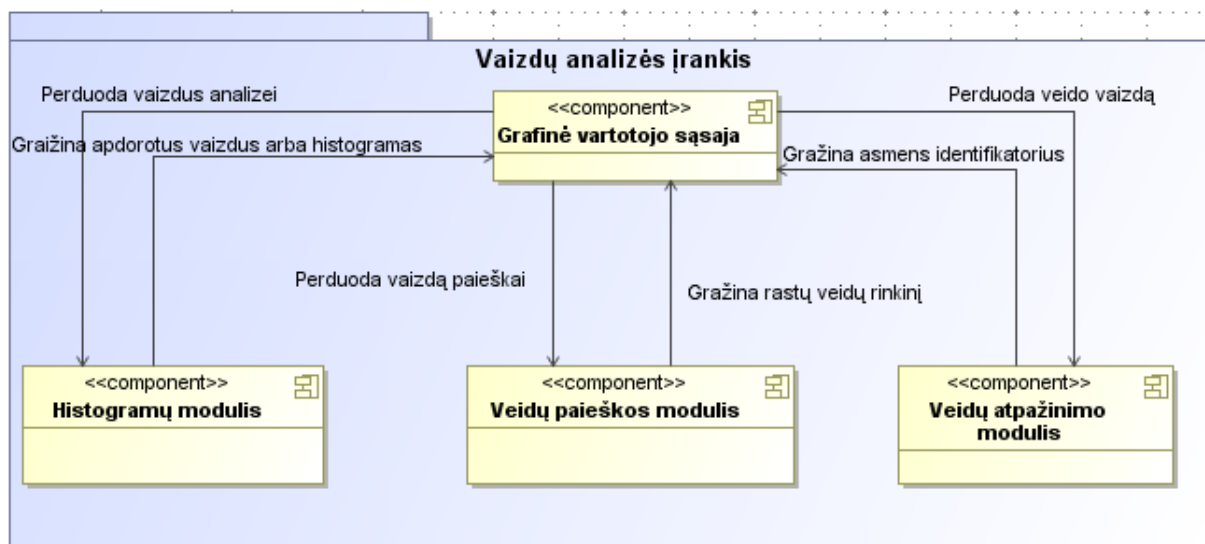
Kaip bazė vaizdų analizės metodams buvo pasirinkta „OpenCV“ biblioteka. Didžiausi jos privalumai yra nepriklausomumas nuo programavimo aplinkos, galimybė lengvai perkelti projektą į kitas platformas (Linux, Mac OS, Android) ir didžiausias kiekis pagalbinių funkcijų darbui su vaizdais. Taip pat pastovus ir dažni bibliotekos atnaujinimai ir galimybė tvarkyti kodą pačiam, jeigu išskyla būtinybė. Kitas priežastis yra ta, kad pasinaudojus visomis prieš tai minėtomis bibliotekomis darbi su „OpenCV“ buvo lengviau.

Pasirinkta programavimo aplinka buvo QtCreator su QT 4.6 biblioteka. Ši programavimo aplinka ir biblioteka lengvai sujungiami su „OpenCV“. Ši aplinka taip pat leidžia labai greitai ir paprastai kurti grafines vartotojo sąsajas ir atvaizduoti „OpenCV“ apdorotus vaizdų duomenis. Ji yra nemokama, laisvai pasiekama ir atviro kodo.

2.5. Vaizdų analizės įrankio pradinis projektas

Projektas bus padalintas į keletą atskirų, tarpusavyje nesusietų modulių. Kiekvienas iš šių modulių galės dirbti apskritai, be kitų modulių pagalbos ir neįtakos jų veikimo. Tai reikalinga todėl, kad duomenys su kuriais bus dirbama netgi testinėje aplinkoje yra gana dideli, dėl ko dirbti su visais moduliais ir tuo pačiu duomenimis yra neįmanoma. Pagrindiniai įrankio moduliai (Pav. 20) bus:

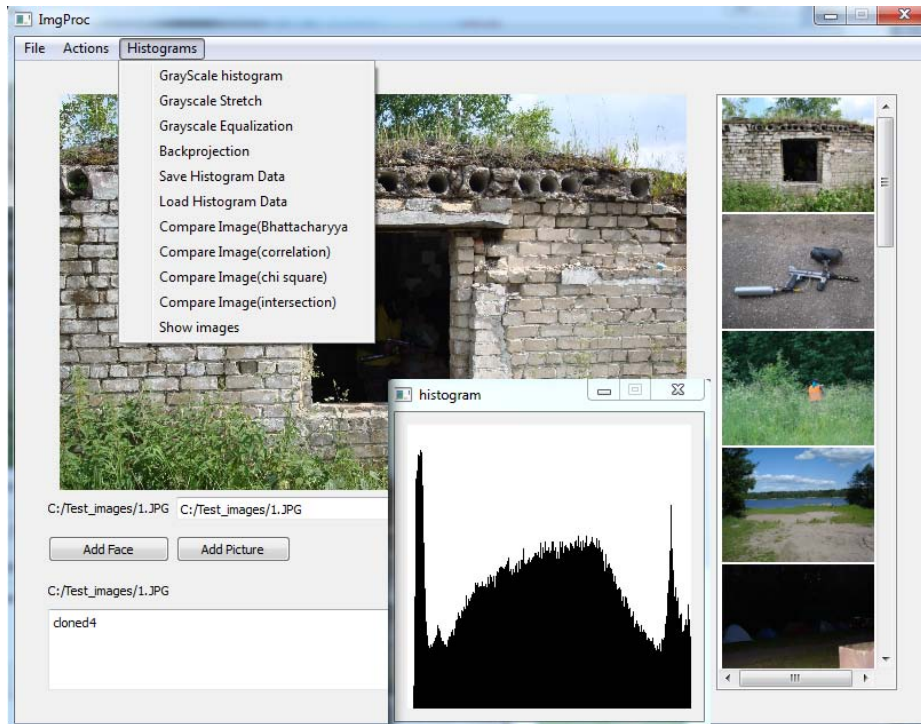
- Grafinė vartotojo sąsaja
- Veidų aptikimo modulis – atsakingas už veidų aptikimą ir paruošimą tolimesnei analizei.
- Veidų atpažinimo modulis – atsakingas už atpažinimą ir sistemos apmokymą.
- Histogramų modulis – atsakingas už visus metodus susijusius su histogramomis. Jų atvaizdavimą, saugojimą ir bazinę analizę.



Pav. 20 Įrankio komponentų diagrama.

3. Galutinis projekto aprašymas

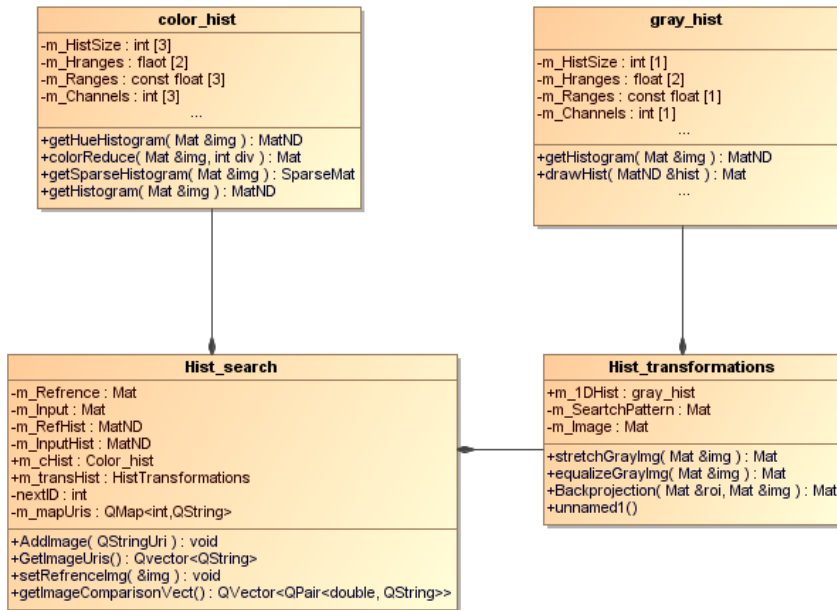
Pirmas sukurtas modulis buvo, histogramų analizės modulis(Pav. 21).



Pav. 21 Histogramų modulis ir veikimas sukurtame vaizdų analizės įrankyje.

Jį sudaro spalvų histogramų klasė, pustoniu histogramų klasė, vaizdų palyginimo klasė, vaizdų transformacijos klasės(Pav. 22).

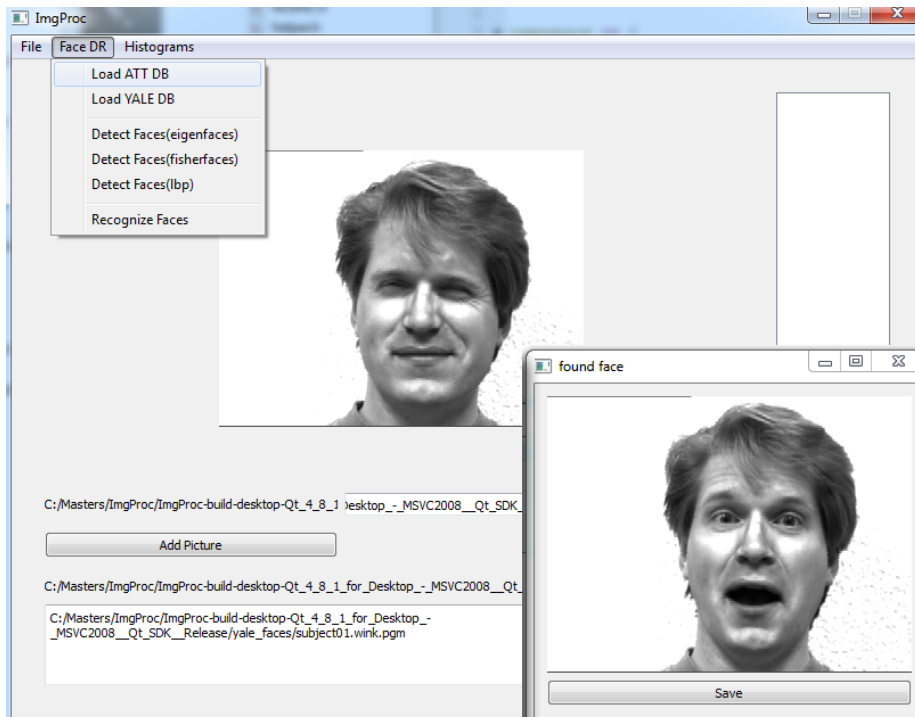
- Color_hist - Spalvų histogramų klasė. Gauną vaizdą, sukuria jo spalvų histogramą, sumažina spalvų kiekį.
- Gray_hist - Pustoniu histogramų klasė. Gauną vaizdą, sukuria jo pustoniu histogramą ir prireikus ją grafiškai atvaizduoja.
- Hist_Transformations – vaizdų transformacijų, panaudojant histogramas, klasė. Atlieką galinį vaizdų projektavimą, vaizdo kontrasto ištempimą ir išlyginimą.
- Hist_search – vaizdų palyginimo ir atrankos klasė naudojanti histogramas. Atlieka palyginimą naudojant kelias skirtingas funkcijas. Taip pat saugo informaciją apie vaizdus, kurie yra įtraukti į palyginimo sąrašą. Pagrindinis interfeisas susisiekiant su pirmomis trimis klasėmis.



Pav. 22 Histogramų modulio diagrama

3.1. Veidų atpažinimo modulis

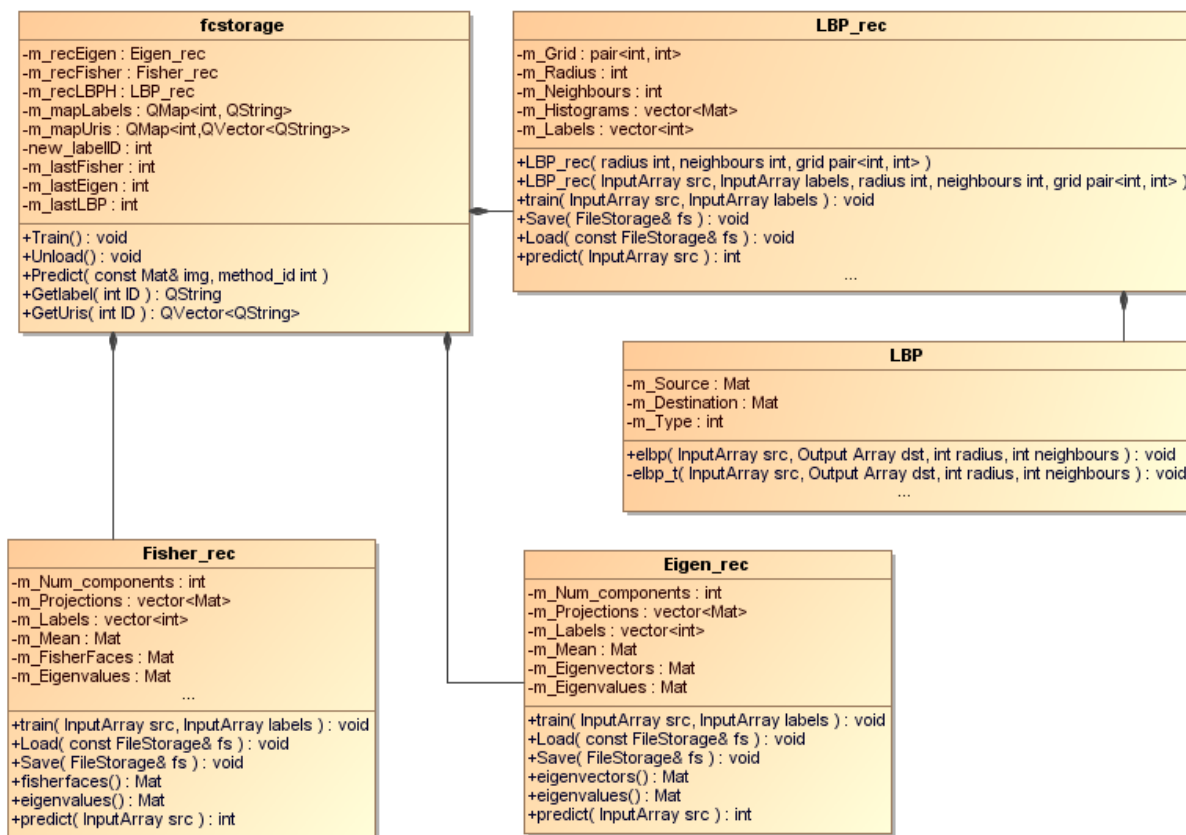
Antras sukurtas modulis buvo veidų atpažinimo modulis(Pav. 23). Sukurtas modulis dirba su dviem testinėm vaizdų bazėm(AT&T ir Yale) ir realizuoja tris minėtus veidų atpažinimo metodus – tikrinių veidų, Fischerio veidų ir lokalių binarinių šablonų.



Pav. 23 Veidų atpažinimo modulio funkcijos ir veikimas sukurtame įrankyje.

Visas modulis yra sudarytas iš 5 pagrindinių klasių(Pav. 24):

- Fcstorage – apjungianti klasė, kuri leidžia dirbti su atskirų metodų klasėmis, saugo ir nuskaito bendrą metodų ir modulio informaciją.
- LBP – klasė naudojama lokalių binarinių šablonų skaičiavimui.
- Fisher_rec – klasė naudojamą veidų atpažinimui Fischerio veidų pagalba. Atsakinga už sistemos apmokymą naudojant testines duomenų bazes it nustato asmenį, kuriam priklauso veidas. Saugo visus susijusius su šiuo metodu duomenis ir gali juos išsaugoti į tekstinį failą ir nuskaityti juos iš failo.
- LBP_rec – klasė naudojama veidų atpažinimui LBP pagalba. Atsakinga už sistemos apmokymą ir asmens nustatymą pagal veidą. Saugo visus susijusius su šiuo metodu duomenis ir gali juos išsaugoti į tekstinį failą ir nuskaityti juos iš failo.
- Eigen_rec – klasė naudojama veidų atpažinimui Tikrinių veidų pagalba. Atsakinga už sistemos apmokymą naudojant testines duomenų bazes it nustato asmenį kuriam priklauso veidas. Saugo visus susijusius su šiuo metodu duomenis ir gali juos išsaugoti į tekstinį failą ir nuskaityti juos iš failo.

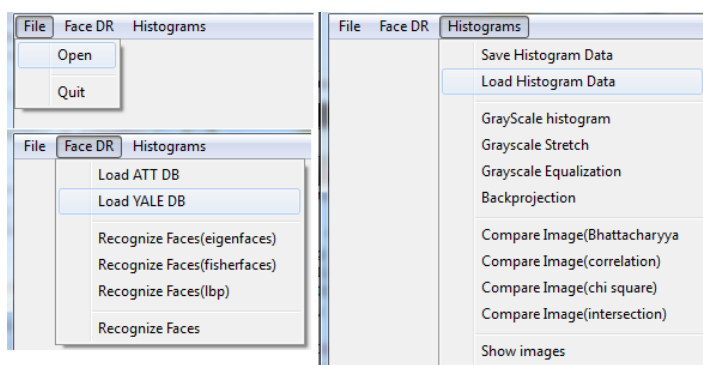


Pav. 24 Veidų atpažinimo modulio diagrama

Trečias sukurtas modulis buvo grafinė vartotojo sąsaja. Visi kitų modulių metodų iškvietimai ir rezultatų atvaizdavimas vyksta šios modulio pagalba. Paleidus įrankį pirmi veiksmai, kuriuos galima atlikti yra vaizdo atidarymas arba duomenų bazių užkrovimas (Pav. 25). Priklausomai nuo veiksmų, kuriuos mes norėsime atlikti, mums reikės:

Norint panaudoti histogramų modulį – atidaryti vaizdą („file->open“). Jeigu mes norėsime atlikti palyginimą su kitais vaizdais, mums taip pat reikės įkrauti histogramų duomenis („Load Histogram Data“), arba patiems sukurti naują rinkinį(atidarant vaizdą ir spaudžiant mygtuką „Add Picture“).

Norint panaudoti veidų atpažinimo modulį - atidaryti vaizdą ir užkrauti vieną iš veidų duomenų bazių („Load ATT DB“ arba „Load Yale DB“). Dabar galima bandyti atlikti veidų paiešką naudojant vieną iš trijų realizuotų metodų.



Pav. 25 Pradiniai veiksmai dirbant su įrankiu.

3.2.Problemos ir jų sprendimo būdai

1. Pirmoji problema buvo vaizdų analizės rezultatų atvaizdavimas. Dėl to, kad grafinė vartotojo sąsaja buvo sukurta su QT biblioteka, o ne naudojant „OpenCV“ iškilo nesuderinamumas tarp naudojamų vaizdų saugojimo klasių. Tam, kad išspręsti šią problemą teko sukurti funkciją, kuri pirma konvertuoja turimą `cv::Mat` objektą į RGB tipą(pradinis `cv::Mat` objektas gali būti pustoniu formatu arba turėti BGR spalvų tvarką), o po to turimą `cv::Mat` objektą konvertuojame į `QImage` klasės objektą.
2. Antra problema buvo vaizdų analizės modulio inicializavimas. Dėl nemažo duomenų kiekio pirmas šio modulio inicializavimas užtrunka gana ilgai. Todėl buvo bandoma išsaugoti visus duomenis į serializuotą failą, kuriame būtų saugoma visa modulio būseną po pirmos inicializacijos. Bandant išspręsti šia problemą tapo aišku, kad naudojant QT bibliotekos ir `c++` suteikiamus serializavimo metodus išsaugoti modulio būseną yra neįmanoma, dėl sudėtingos `cv::Mat` struktūros. Todėl buvo nuspręsta naudoti atskirus išsaugojimo ir užkrovimo metodus bendrai modulio (atitinkamų objektų būsenos

saugomos naudojant QT serializavimo metodus) ir „openCV“(cv::Mat duomenis išsaugomi į atitinkamus .yml formato failus) dalims.

3.3. Darbo rezultatų analizė

Sukurtas vaizdų analizės įrankis. Šis įrankis gali būti apmokytas atpažinti veidus trimis skirtingais būdais: tikrinių veidų metodas, Fišerio veidų metodas, lokalių binarinių šablonų metodas. Taip pat įrankis gali atlikti bazinę vaizdų analizę ir lyginimą naudojant histogramas ir jų transformacijos metodus. Įrankyje realizuoti metodai turi skirtingus plusus ir minusus, kurių testavimas ir analizė bus aprašyti sekančiuose skyriuose. Buvo realizuotas konvertavimas tarp QT ir OpenCV duomenų tipų ir modulių būsenų išsaugojimas, tam, kad pagreitinti ir palengvinti darbą su įrankių, panaudojant skirtingus duomenų išsaugojimo metodus.

3.3.1. Įrankio ir metodų testavimas

Įrankio testavimui buvo panaudotos jau minėtos AT&T ir Yale veidų duomenų bazės, taip pat buvo sukurtas nedidelis rinkinys paprastų fotografijų. Naudojant šiuos testinius duomenis buvo atliktas įrankio testavimas ir naudojamų metodų analizė.

Paprastų fotografijų rinkinys yra sudarytas iš 17 skirtingo dydžio ir aukštos kokybės spalvotų JPG formato fotografijų. Dalis fotografijų atvaizduoja panašius objektus skirtingame apšvietime(Pav. 26).



Pav. 26 fotografijų rinkinio pavyzdys naudojamas histogramų modulio testavimui

Dėl to, kad veidų aptikimo ir atpažinimo metodams reikės daug apmokomųjų vaizdų bus naudojamos jau sukurti ir paruošti analizei vaizdų rinkiniai. Pirmas – Yale veidų rinkinys (Pav. 27) . Jį sudaro 165 pustoniai vaizdai. 15 asmenų, po 11 skirtingų fotografijų kiekvienam asmeniui: skirtingas apšvietimas, fotografijos su ir be akinių, skirtingos veido išraiškos. Antras rinkinys – AT&T(Pav. 28) veidų rinkinys. Sudarytas iš 40 asmenų, po 10 fotografijų kiekvienam asmeniui. Čia taip pat skiriasi veidų išraiškos, apšvietimas ir kt. Bet šioje bazėje visi vaizdai buvo padaryti ant tamsaus fono.



Pav. 27 Yale vieno asmens nuotraukų rinkinys



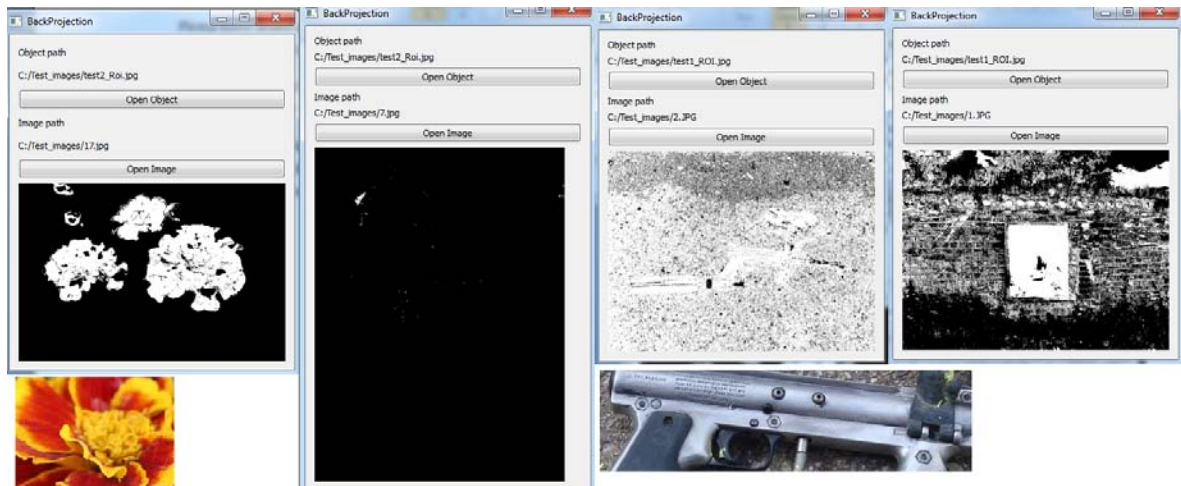
Pav. 28 AT&T vieno asmens nuotraukų rinkinys.

3.3.2. Realizuotų metodų analizė

Pirma buvo analizuoti metodai naudojantys histogramas. Šiuo atveju tai galinio histogramos projektavimo metodas ir histogramų palyginimo metodai.

Tam, kad atlikti galinio histogramos projektavimo metodo testavimą taip pat buvo sukurti dominančio objekto vaizdai. Šių vaizdų histogramos buvo projektuojamos į mus dominantį vaizdą.

Kuo šviesesnis taškas rezultato vaizde tuo geriau sutampa regionai tarp dominančio regiono vaizdo ir analizuojamo vaizdo. Pav. 29 mes matome galinio projektavimo rezultatus naudojant skirtingus dominančio objekto vaizdus ir skirtingus analizuojamus vaizdus.



Pav. 29 Galinio projektavimo pavyzdžiai.

Kaip matome net iš šių paprastų pavyzdžių, galinis histogramos projektavimas gali pateikti labai skirtingus rezultatus. Pagrindinė problema yra, kad jis naudoja tik spalvų histogramas ir neatsižvelgia į objektų ypatybės, kontūrus ir yra labai priklausomas nuo apšvietimo. Dėl šių priežasčių tiksliai objektų paieškai toks metodas netinka. Bet ji galima pritaikyti objektų paieškoje, kaip vieną iš netinkamų vaizdų atmetimo metodų. Taip sumažinant apdorojamų sudėtingesniais metodais vaizdų kiekį.

Sekantis metodas yra vaizdų paieška lyginant vaizdų histogramas. Šiuo atveju 17 minėtų fotografijų buvo įvestos į histogramų palyginimo bazę. Toliau, kiekviena iš šių fotografijų buvo lyginama su visomis esančiomis bazėje fotografijomis, ir buvo atvaizduojamas fotografijų sąrašas jų panašumo mažėjimo tvarka. Tai buvo atliekama su kiekvienu iš keturių realizuotų lyginimo funkcijų: Koreliacijos, Chi-kvadrato, sankirtos, Bhattacharyya nuotolio.

Tikrinamo vaizdo numeris	Vaizdų sąrašas pradedant nuo labiausiai tinkančio vaizdo														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	11	16	13	12	8	15	14	7	4	2	10	6	5	3	9
2	6	16	15	8	11	12	1	13	14	10	6	5	9	3	7
3	1	10	4	11	12	13	16	15	14	5	8	2	9	6	7
4	1	11	16	12	15	8	2	13	14	10	6	5	9	3	7
5	11	12	1	4	8	14	13	3	10	7	9	16	2	15	6
6	2	15	16	8	11	12	1	13	7	4	14	10	9	3	5
7	8	1	11	16	15	12	13	14	2	6	9	10	4	5	3
8	11	12	16	15	7	1	13	14	2	6	4	9	10	5	3
9	10	16	15	8	11	14	12	2	7	13	1	4	6	5	3
10	14	9	12	15	11	16	13	1	8	2	4	7	3	6	5
11	12	13	8	14	1	15	16	7	2	10	4	5	6	9	3
12	11	13	14	8	1	16	15	10	2	7	4	6	9	5	3
13	14	12	11	16	1	15	8	10	2	7	4	6	9	5	3
14	13	12	11	15	1	10	8	16	2	7	9	4	6	5	3
15	16	2	11	8	13	14	1	6	12	10	7	4	9	3	5
16	15	2	11	8	1	13	12	6	14	7	10	4	9	3	5

Lentelė nr. 1 Bhattacharyya funkcijos vaizdų atrinkimo rezultatai

Tikrinamo vaizdo numeris	Vaizdų sąrašas pradedant nuo labiausiai tinkančio vaizdo														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	11	12	8	16	13	15	4	14	7	6	2	5	10	9	3
2	6	16	15	8	11	12	13	4	1	14	9	10	7	5	3
3	4	1	5	10	11	13	12	14	6	15	16	8	9	2	7
4	1	11	6	8	16	12	15	5	13	14	2	10	9	7	3
5	4	11	12	1	8	14	6	13	7	10	3	9	16	15	2
6	2	15	16	8	4	11	12	1	13	7	14	9	10	5	3
7	8	1	11	16	15	12	6	13	14	4	9	2	10	5	3
8	11	12	15	16	1	7	13	6	4	14	2	9	10	5	3
9	10	4	11	15	16	8	12	14	6	7	2	13	1	5	3
10	14	9	12	11	13	15	16	4	1	8	6	2	7	5	3
11	12	13	8	14	1	15	16	4	7	2	5	6	10	9	3
12	11	13	14	8	1	16	15	4	10	2	6	7	5	9	3
13	14	12	11	1	16	15	8	4	2	10	6	7	9	5	3
14	13	12	11	1	15	10	8	16	4	6	2	7	9	5	3
15	16	11	2	6	8	13	1	12	14	4	7	10	9	5	3
16	15	2	8	11	6	1	13	12	4	14	7	10	9	5	3

Lentelė nr. 2 chi-kvadrato funkcijos vaizdų atrinkimo rezultatai

Tikrinamo vaizdo numeris	Vaizdų sąrašas pradedant nuo labiausiai tinkančio vaizdo														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	11	12	8	15	4	7	13	16	5	14	2	6	3	10	9
2	6	16	15	8	1	4	11	12	13	9	7	10	14	3	5
3	5	1	11	10	4	12	13	8	15	14	16	9	2	6	7
4	8	1	11	15	6	2	12	16	5	13	3	10	9	14	7
5	11	1	4	12	3	8	13	14	9	10	7	2	16	15	6
6	2	15	8	16	4	1	11	12	7	13	9	14	10	3	5
7	1	8	11	15	12	16	13	6	2	14	4	9	10	5	3
8	15	11	16	6	2	12	1	4	13	7	14	9	10	5	3
9	10	16	15	8	11	12	13	2	1	14	4	7	6	5	3
10	12	14	9	13	11	1	15	8	16	3	2	4	6	7	5
11	12	1	8	13	15	14	4	5	16	7	2	6	10	9	3
12	11	13	1	8	14	15	10	16	4	2	7	6	5	9	3
13	14	12	11	1	15	16	8	2	7	10	9	4	6	3	5
14	13	12	11	1	15	10	8	7	16	9	2	4	6	3	5
15	16	8	6	2	1	11	4	13	12	14	7	9	10	3	5
16	2	15	8	6	1	11	13	12	4	7	9	14	10	3	5

Lentelė nr. 3 koreliacijos funkcijos vaizdų atrinkimo rezultatai

Tikrinamo vaizdo numeris	Vaizdų sąrašas pradedant nuo labiausiai tinkančio vaizdo														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	11	8	12	16	15	7	13	14	4	2	10	9	3	6	5
2	16	15	6	8	11	13	12	1	14	9	10	7	4	5	3
3	1	10	11	13	12	4	14	16	15	8	5	9	2	7	6
4	1	11	8	16	12	15	14	2	13	9	10	7	6	3	5
5	11	1	12	8	14	13	4	7	3	10	9	16	2	15	6
6	2	15	16	8	1	11	12	13	7	9	14	4	10	5	3
7	8	1	11	16	12	15	13	14	9	2	10	6	4	5	3
8	11	15	1	16	12	7	13	14	2	6	4	9	10	5	3
9	10	11	8	16	15	12	7	14	2	1	13	4	6	3	5
10	14	9	12	11	13	16	15	1	8	2	3	7	4	6	5
11	12	1	13	8	15	14	16	7	2	10	9	4	5	6	3
12	11	14	13	1	8	16	15	10	2	7	9	4	6	3	5
13	14	12	11	1	15	8	16	2	10	7	9	4	6	3	5
14	13	12	11	1	10	8	15	16	2	7	9	4	6	3	5
15	16	2	11	8	1	13	12	6	14	7	10	9	4	3	5
16	15	2	8	11	1	12	13	14	7	6	10	9	4	3	5

Lentelė nr. 4 sankirtos funkcijos vaizdų atrinkimo rezultatai

Kaip matome iš lentelių visos skaičiavimo nuncijuos duoda skirtingus rezultatus. Vieninteliai šiek tiek panašius ryšius rezultatus duodančios funkcijos yra Sankirtos ir Bhattacharyya. Jeigu mes pažvelgsim į gautas sekas ir palyginsim rezultatus vizualiai tai galime pastebėti, kad panašius vaizdus blogiausiai aptiko koreliacijos metodas, likę trys metodai panašius vaizdus aptiko teisingai. Šiek tiek daugiau besiskiriančius vaizdus (7 ir 8) geriausiai aptiko Bhattacharyya funkcija, sankirtos ir chi-kvadrato funkcijos pasirodė vienodai, o blogiausiai suveikė koreliacijos funkcija. Skirtingas fotografijas, kurios neturi labai panašių porų, geriausiai atrinko chi-kvadrato funkcija.

Taigi, po atliktų bandymų galima teigti, kad histogramų palyginimo metodas yra vienas iš galimų vaizdų paieškos ir rūšiavimo sprendimų, kuris veikia gana gerai bendrais atvejais ir gali būti pritaikytas fotografijų kopijų(arba labai panašių fotografijų) paieškai. Tačiau reikia pabrėžti, kad neatlikus histogramų dimensijų sumažinimo ir kitų transformacijų, šie metodai yra lėti, ir gaunami analizės rezultatai – histogramos, yra labai didelės.

Sekanti analizuojamų metodų grupė buvo veidų analizės metodai. Tam, kad atlikti jų analizę buvo naudojamos minėtos AT&T ir Yale veidų bazės. Metodų analizė buvo vykdoma sekančių būdų:

Veidų analizės modulis buvo inicijuojamas AT&T veidų baze. Tai buvo daroma septynis kartus, kiekvieną kartą mažinant vaizdų kiekį, kuris naudojamas asmens klasifikavimui.

Toliau iš eilės buvo bandoma atpažinti kiekvieną asmenį esantį duomenų bazėje panaudojant vieną iš jo vaizdų, kuris neįėjo į inicializavimo imtį. Tai buvo atliekama su kiekvienu iš trijų realizuotų metodų.

Gauti rezultatai buvo lyginami, ir buvo išvesti kiekvieno metodo atpažinimo tikslumo koeficientai priklausomai nuo apmokymo imties dydžio.

Tie patys veiksmai buvo atliekami ir su Yale duomenų baze.

Apmokymo imties dydis	Tikrinių veidų metodas	Fišerio veidų metodas	Lokalių binarinių šablonų metodas
9	0,90	0,95	0,93
8	0,90	0,90	0,93
7	0,90	0,93	0,93
6	0,90	0,93	0,93
5	0,83	0,83	0,90
4	0,78	0,83	0,85
3	0,76	0,80	0,78
2	0,71	0,66	0,71
vidurkis	0,84	0,85	0,87

Lentelė nr. 5 Veidų atpažinimo koeficientai naudojant AT&T bazę

Apmokymo imties dydis	Tikrinių veidų metodas	Fišerio veidų metodas	Lokalių binarinių šablonų metodas
10	0,25	0,19	0,50
9	0,25	0,06	0,50
8	0,31	0,19	0,50
7	0,25	0,13	0,50
6	0,19	0,19	0,50
5	0,25	0,06	0,50
4	0,19	0,19	0,50
3	0,19	0,19	0,50
vidurkis	0,23	0,15	0,50

Lentelė nr. 6 Veidų atpažinimo koeficientai naudojant Yale bazę

Analizuojant rezultatus mes galime pastebėti, kad dirbant su AT&T baze, atpažinimo rezultatai tarp metodų skiriasi gana nežymiai. Taip pat matome, kad tikslumas tiesiogiai priklauso nuo apmokymo imties dydžio. Bet iš visų metodų LBP metodo rezultatų vidurkis yra didžiausias, nors ir nežymiai. Pagrindinė tokio nežymaus skirtumo tarp rezultatų priežastis yra ta, kad AT&T bazė yra lengva, visuose vaizduose vienodas fonas, išraiškos ir apšvietimas skiriasi nežymiai. Dėl to visi metodai lengvai sugeba klasifikuoti asmenis.

Dirbant su Yale duomenų baze, gauti rezultatai yra įdomesni. Kaip matome, nepriklausomai nuo apmokymo imties LBP metodas rodo vienodą rezultatą, kuris yra žymiai geresnis negu kitų dviejų metodų. Taip pat mes matome, kad gauti atpažinimo koeficientai yra daug mažesni už gautus naudojant AT&T bazę. Tokio skirtumo tarp rezultatų priežastis yra ta, kad vaizdai Yale bazėje labiau skiriasi vienas nuo kito, ypač apšvietimu. Būtent atsparumas į tokius skirtumus ir buvo testuojamas su šia baze. Iš apmokymo imčių buvo išimtas vaizdas, kurio apšvietimo lygis smarkiai skyrėsi nuo kitų, ir būtent jis buvo naudojamas kaip vaizdas kurį reikėjo identifikuoti. Taigi mes galime teigti, kad LBP metodas yra atspariausias šviesos pasikeitimams ir teoriškai geriausiai tinka veidų atpažinimui bendrais atvejais.

Atliekant bandymus buvo išsaugomi klasifikatoriai, sugeneruoti kiekvienu atskiru metodu. Tai yra dar vienas svarbus dalykas, į kurį reikia atsižvelgti pasirenkant atpažinimo metodą konkrečiam atvejui. Kai moduliai apmokomi naudojant visą Yale vaizdų bazę sugeneruotų failų dydžiai yra:

- Tikrinių veidų failas („eigenfacesYale.yml“) – 45Mb
- Fišerio veidų failas („fisherfacesYale.yml“) – 4.5 Mb
- LBP histogramų failas („lbphfacesYale.yml“) – 15.5 Mb

Kaip matome, jų dydžiai smarkiai skiriasi. Tikrinių veidų failas užima daugiausiai vietas, ir idealiu atveju šis metodas duoda blogiausių (nors ir nežymiai) rezultatus, bet turint tikrinius veidus, mes galime išgauti originalų pustonių vaizdą (Fišerio ir LBP metodai to padaryti negali), kas pašalina būtinybę pridėti visus veidų failus atskirai ir yra pagrindinis šio metodo pliusas atsižvelgiant į turimus rezultatus. Fišerio veidų failas yra pats mažiausias, o rezultatai geroje sąlygose yra labai aukšti, nors metodas ir nėra labai atsparus šviesos pasikeitimams. Šis metodas gali būti gerai pritaikytas sistemai, kurioje yra riboti resursai (pvz. išmanieji telefonai). Trečias metodas geriausiai tinka kai visi vaizdai yra saugomi kitoje sistemos dalyje arba jų originalai yra smarkiai skiriasi nuo naudojamų atpažinimo procese, reikalingi geri rezultatai ir analizuojami vaizdų rinkiniai yra sudėtingi.

4. Išvados

1. Atsižvelgiant į teorinę analizę, pagrindinis pasirinktas vaizdų analizės būdas – histogramų analizė. Šie analizės metodai yra paprasti, greiti, leidžia lengvai išsaugoti ir analizuoti gautą informaciją.
2. Atlikus darbo rezultatų analizę buvo nustatyta, kad programoje realizuotas histogramų palyginimo metodas gali būti naudojamas panašių vaizdų paieškai. Vaizdų kopijas ir labai panašius vaizdus šis metodas (priklausomai nuo pasirinktos funkcijos) aptinka iki 100 % tikslumo.
3. Pritaikius skirtingus vaizdų analizės metodus buvo sukurtas įrankis, kuris leidžia atlikti vaizdų transformacijas, paiešką pagal turinį, naudojant kelis skirtingus metodus, ir objektų (veidų) atpažinimą. Atlikus testavimą buvo nustatyta, kad, priklausomai nuo veidų rinkinio sudėtingumo ir naudojamų metodų, tikslumas yra tarp 12–95 % (žr. 3.3 Darbo rezultatų analizė).
4. Buvo nustatyta, kad bazinių vaizdų analizės metodų neužtenka, kad atlikti tikslią objektų ir vaizdų paiešką. Tam reikalingi sudėtingesni metodai, kurie atlieka klasifikaciją ir sumažina analizuojamų duomenų kiekį.
5. Atlikus eksperimentus, buvo nustatyta, kad iš trijų realizuotų veidų atpažinimo metodų geriausius rezultatus dirbant su sudėtingais veidų rinkiniais duoda lokalių binarinių šablonų histogramų palyginimo metodas. Bet priklausomai nuo reikalavimų (veidų rinkinių sudėtingumo, turimų resursų kiekio, poreikio atstatyti vaizdą iš turimų analizės duomenų), Fišerio ir tikrinių veidų algoritmai irgi gali būti sėkmingai naudojami veidų atpažinimui.

Naudota literatūra:

1. M. Kirby, L. Sirovich. *Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 1, 1990.
2. T. J. Stonham. *Practical face recognition and verification with wisard*. In H. D. Ellis, editor, *Aspects of face processing*. Kluwer Academic Publishers, 1986.
3. L. Sirovich and M. Kirby. *Low-dimensional procedure for the characterization of human faces*. *Journal of the Optical Society of America A - Optics, Image Science and Vision*, 4(3):p.519–524, 1987.
4. M. Turk and A. Pentland. *Eigenfaces for recognition*. *Journal of Cognitive Neuroscience*, 3(1):p.71–86, 1991.
5. T. Kohonen. *Self-organization and associative memory*. Springer-Verlag, Berlin, 1989.
6. P. Viola, M. Jones. *Rapid Object Detection using a boosted cascade of simple features*. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2001*, Volume: 1, Issue: C, Publisher: IEEE Comput. Soc, p511-518, 2001.
7. Peter N. Belhumeur, J. Heapanha, D. Kriegman. *Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection* *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, p. 711-720, 2001.
8. Ahonen, T., Hadid, A. and Pietikäine. *Face Description with Local Binary Patterns: Application to Face Recognition*. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(12):2037-2041, 2006.
9. IVT library [interaktyvus] [žiūrėta 2012m. Kovo 22d.]. Prieiga per internetą: <<http://ivt.sourceforge.net/index.html>>
10. Img(Rummager)[interaktyvus] [žiūrėta 2012m. gegužės 1d.]. Prieiga per internetą: <http://chatzichristofis.info/?page_id=213>
11. Bing Image search[interaktyvus] [žiūrėta 2012m. gegužės 1d.]. Prieiga per internetą: <<http://www.bing.com/?scope=images&nr=1&FORM=NOFORM>>
12. Imense image search portal [interaktyvus] [žiūrėta 2012m. gegužės 1d.]. Prieiga per internetą: <<http://www.imense.com/>>
13. Gazopa image search[interaktyvus] [žiūrėta 2012m. gegužės 1d.]. Prieiga per internetą: <<http://www.gazopa.com/>>
14. TinEye image search[interaktyvus] [žiūrėta 2012m. gegužės 1d.]. Prieiga per internetą: <<http://www.tineye.com/>>

15. AForge.Net framework[interaktyvus] [žiūrėta 2012m. kovo 18d.]. Prieiga per internetą: <<http://www.aforogenet.com/framework/>>
16. OpenCV library[interaktyvus] [žiūrėta 2011m. gegužės 4d.]. Prieiga per internetą: <<http://opencv.willowgarage.com/wiki/>>

Anotacija

Lietuviškai

Autorius: Oleg Mazec

Tema: Fotografijų analizės metodai ir jų taikymai

Šiaulių universitetas 2012

Šio darbo tikslas yra sukurti įrankį, skirtą skaitmeninių fotografijų paieškai, analizei ir palyginimui pagal turinį. Šiam tikslui pasiekti buvo apžvelgtos egzistuojančios paieškos pagal turinį sistemos ir vaizdų analizės metodai. Remiantis analizės rezultatais buvo parinkti metodai, kurie vėliau buvo realizuoti programoje. Sukurta programa atlieka histogramų analizę ir palyginimą. Taip pat atliekamas veidų atpažinimas trimis skirtingais metodais.

English

Author: Oleg Mazec

Subject: Image Processing Methods and their Practical Applications

Šiauliai University 2012

The goal of this work is to create an application used for searching, comparing and analyzing digital images based on their content. For this purpose a number of existing image search and analysis tools and methods were reviewed. Based on the review results methods, which were later used in the application, were selected and analyzed further. The created application performs image histogram analysis and comparison using various methods. It also performs face recognition using three different methods.

Priedai

1. CD turinys

- Šakniniame kataloge yra elektroninės magistro darbo aprašymo versijos („aprasymas.doc“ ir „aprasymas.pdf“).
- Šakniniame kataloge yra archyvas ImgProc.rar. Kuriame yra sukurtas įrankis, visos jo veikimui reikalingos bibliotekos, ir testiniai duomenų rinkiniai, kurie buvo naudojami šiame darbe.

2. Programos naudojimas

- Reikia išarchyvuoti „ImgProc.zip“ į norimą katalogą ir paleisti ImgProc.exe. Jeigu išmetama klaida, kad trūksta *.dll, įdiekite „Microsoft Visual C++ 2008 Redistributable Package (x86)“, tai galima padaryti paleidus „vcredist_x86.exe“, esantį šakniniame kataloge.
- Prieš atliekant kokius nors veiksmus, priklausomai nuo funkcijų, kurias norima atlikti, reikia užkrauti vieną iš veidų duomenų bazių („Face RD“ -> „Load ATT DB“ arba „Load Yale DB“) arba histogramų bazę („histograms“ -> „Load histograma data“).
- Sekantis veiksmas, kurį reikia atlikti tai atidaryti vaizdą („File“ -> „Open“). Atidarius vaizdą jį galima atlikti jo analizę arba pridėti prie Histogramų duomenų – „Add Picture“ mygtukas (jeigu norima papildyti esamą duomenų rinkinį arba sukurti naują).
- Projektas naudoja duomenis, kurie yra „att_faces“ ir „yale_faces“ kataloguose. Todėl jų negalima ištrinti.
- Yra katalogas su fotografijomis „Test_images“. Čia yra fotografijos, kurios buvo naudojamos testavimo metu.
- Taip pat yra .dat ir .yaml failai, kuriuose saugomi modulių duomenys naudojami pagreitintam sistemos inicializavimui.