

ЎЗБЕКИСТОН РЕСПУБЛИКАСИ  
ОЛИЙ ВА ЁРТА МАХСУС ТАЪЛИМ ВАЗИРЛИГИ  
НАМАНГАН МУҲАНДИСЛИК - ҚУРИЛИШ  
ИНСТИТУТИ

Енергетика ва саноатни ахборотлаштириш факультети

Информатика ва ахборот технологиялари кафедраси

**БИТИРУВ  
МАЛАКАВИЙ ИШИ**

*Мавзу: “Локал бинар образлар” оператори ёрдамида  
таъсир белгиларини ажратиш алгоритмлари ва  
дастурини ишлаб чиқиш*

Бажарди:

БМИ раъбари:

Зокиров Фаррухмирзо

т.ф.н. М. Тўхтасинов

## Кириш

Бизга маълумки, ҳозирда дунёда интеллектуал компьютер тизимларидан кенг фойдаланилмоқда ва улар янада ривожлантирилмоқда. Таъкидлаш керакки, бу каби интеллектуал компьютер тизимларининг аксарияти рақамли тасвирларни қайта ишлаш ва таниб олиш илмий масалалари билан боғлиқдир. Шу сабабли ҳам, тасвирларни қайта ишлаш усулларини ўрганиш, уларни такомиллаштириш ва амалиётга тадбиқ этиш масалаларини ҳал қилиш муҳим вазифалардан бири ҳисобланади.

Ҳукуматимиз томонидан АКТ тегишли бўлган бир нечта фармон ва қарорлар қабул қилинди. Бунга мисол қилиб “Ўзбекистонлик дастурий таъминот ишлаб чиқарувчиларни рағбатлантиришни янада кучайтириш чора-тадбирлари тўғрисида”ги ва ”Замонавий ахборот-коммуникация технологияларини янада жорий этиш ва ривожлантириш чора тадбирлари тўғрисида”ги Президент қарори ҳамда “Компютерлаштиришни янада ривожлантириш ва ахборот-коммуникация технологияларини жорий этиш тўғрисида”ги Президент фармонини келтиришимиз мумкин. Бундан ташқари Ўзбекистон Республикаси Президенти Шавкат Миромонович Мирзияевнинг ”2017 -2021 йилларда электротехника саноатини бошқаришни янада такомиллаштириш, уни жадал ривожлантириш ва диверсификациялаш чора -тадбирлари тўғрисидаги” 2017 йил 13 февралдаги ПҚ - 2772 - сонли қарори қабул қилинди.

**Битирув ишининг долзарблиги.** Бугунги кунда рақамли тасвирларнинг белгиларини аниқлаш ва таниб олиш билан боғлиқ илмий-амалий масалаларни ҳал қилиш долзарб масалалардан ҳисобланади. Бу каби масалаларда асосий босқичдаги ишлардан бири бу – рақамли тасвирларнинг идентификацион белгиларини аниқлашдир. Ҳозирда дунё тажрибасида “Локал бинар образлар” оператори ёрдамида тасвир белгиларини ажратиш яхши натижалар бермоқда. Шу сабабли ҳам биз бу йўналишда илмий-амалий ишни бажариш долзарб масала деб ҳисоблаган ҳолда, БМИ мавзусини «“Локал бинар образлар”

оператори ёрдамида тасвир белгиларини ажратиш алгоритмлари ва дастурини ишлаб чиқиш» деб номладик.

**Битирув малакавий ишининг мақсади.** Ушбу битирув ишининг мақсади рақамли тасвирлардаги объектларни таниб олиш имкониятни берувчи идентификацион белгиларни “Локал бинар образлар” оператори асосида аниқлаш учун усуллар, алгоритмлар ҳамда амалий дастурни ишлаб чиқишдан иборат.

**Битирув малакавий ишининг вазифаси.** БМИнинг вазифаси мавзу доирасида амалий дастур ишлаб чиқиш ва амалиётга тадбиқ этишдир. Шунингдек, олинган илмий-амалий ва назарий натижаларни соҳа фан ривожини учун қўллашдан иборат.

**Битирув ишининг объекти.** БМИнинг объекти рақамли тасвирлар.

**Мавзунинг предмети.** Рақамли тасвирларнинг идентификацион белгиларини “Локал бинар образлар” оператори асосида аниқлаш усуллари, алгоритмлари ҳамда амалий дастурлаш технологиялари.

**Битирув ишининг назарий ва амалий аҳамияти.** БМИ иши рақамли тасвирларга оид ахборотларга ишлов бериш, тасвирда идентификацион белгиларни аниқлаш ва таниб олиш билан боғлиқ ахборотларни автоматик қайта ишлаш ва баҳолаш ҳақида бўлиб, унда тасвирларга сонли ишлов беришнинг назарий ва амалий масалалари қаралади. Олинган назарий натижалар фан соҳаси учун учун қўшимча илмий манба сифатида самарадор бўлишлиги билан аҳамиятлидир. Ишнинг амалий аҳамияти ишлаб чиқилган амалий дастурларнинг фойдаланувчиларга тақдим этилиши, қолаверса ундан ишлаб чиқаришда фойдаланиш мумкинлиги билан изоҳланади.

## **1.1. Илмий йўналишнинг ҳамда танланган мавзунинг долзарблиги**

Ҳозирги кунда, яъни ахборот технологиялари асрида инсониятнинг энг долзарб муаммоларидан бири – бу ахборотни қайта ишлаш масалаларидир. Чунки ахборот хажми кундун-кунга ошиб бормоқда. Инсон ҳаётида ахборотни сақлаш ва уларни қайта ишлаш масаласини ҳал этилиши унинг тараққиётига жуда катта таъсир кўрсатади. Ахборот табиатда турли-туман шаклларда учраши ва уни инсон сезги органлари (қулоқ, бурун, кўз, таъм билиш ва тери) орқали қабул қилиши, уларни қайта ишлаб, турларга ажрата олишида ўзига яраша қийинчиликларни келтириб чиқаради. Айниқса, компьютер техникаси ва технологиялари билан боғлиқ бўлган ахборот, яъни электрон ахборот оқими, айниқса, катта миқдорни ташкил этмоқда.

Ахборотни бир қанча турлари мавжудки, улар информатика (ёки ахборот технологиялари) асосини ташкил этади. Ахборотнинг турларини таҳлил қилиш асосида биз ўзимиз қабул қилаётган ахборотнинг 80% дан ортиғини тасвирлар ташкил қилишини кўрамыз. Қолган ахборот инсоннинг бошқа сезги органлари ёрдамида қабул қилинади. Масалан қулоқ ёрдамида эшитиш, бурун ёрдамида ҳид билиш, кўлда пайпаслаб сезиш ва бошқ. Агар инсоннинг умри давомида кўриш аъзоси орқали оладиган ахбороти жамланса, у жуда катта хотирани эгаллайди ва уларни сақлаш учун кўплаб хотира қурилмалари керак бўлади. Тасвирларни қайта ишлаш, таҳлил қилиш ва таниб олиш зарурати фазодан туриб ернинг табиий ресурсларини ўрганишда, ҳаракатланаётган жисмнинг траекториясини аниқлашда, тасвирларни визуаллаштириш, тасвирдаги объектларни таниб олиш каби масалаларда пайдо бўлади.

Ҳозирги кунда рақамли тасвирлар, айниқса видеотасвирлар билан боғлиқ илмий-амалий масалаларни ҳал этиш дунёда долзарб бўлиб қолди. Бунинг асосий сабабларидан бири – бу компьютер хотираларида сақланаётган ахборотларнинг аксарияти рақамли (видео) тасвирлар ҳиссасига тўғри келиши ҳамда шу йўналишдаги масалаларнинг тобора долзарблик касб этаётганлигидир.

Дунёда жиноятчиликни олдини олиш, хавфсизликни таъминлаш, турли илмий-ишлаб чиқариш йўналишларида рақамли тасвирларни илмий таҳлил орқали хулосалар қилиш ҳозирги замоннинг долзарб масалалардан ҳисобланади.

Ҳозирги кунда илмий таҳлил қилиниши лозим бўлган тасвирларни асосан куйидагиларга ажратиш мумкин.

- 1) Харакатдаги тасвирлар. Бундай тасвирлар турига юриб кетаётган одам, ҳаракатдаги автомобил ва ҳ.к.ларни мисол қилиб кўрсатиш мумкин.
- 2) Биометрик тасвирлар (одам юзи, бармоқ изи, кўз қорачиғи ва ҳ.к.).
- 3) Биологик ёки тиббиётга оид тасвирлар (электрон микроскоп ёрдамида олинган хужайралар тасвирлари ва ҳ.к.).
- 4) Космик тасвирлар (спутник ёрдамида олинган ер сатҳи ёки турли бошқа сайёралар сатҳи тасвирлари ва ҳ.к.).
- 5) Хариталар ва турли геологик тасвирлар.

Юқоридаги масалаларда асосан тасвирларга ишлов бериш, уларнинг ўзига хос белгиларини ажратиш олиш ва таниб олиш масалалари ҳал этилиши лозим бўлади. Бизнинг мавзуимиз айнан белгиларни аниқлаш, яъни тасвирларнинг идентификацион белгилар фазосини шакллантириш масаласига бағишланган. Бизга маълумки, бу йўналишда бир қатор усул ва алгоритмлар ишлаб чиқилган. Сўнгги йилларда “Локал бинар образлар” деб номланувчи операторлар оиласига мансуб усул ва алгоритмлар дунё тажрибасида ютуқларга эришмоқда. Ушбу БМИ да биз ҳам шу йўналишда кичик илмий – тадқиқот ишларини ўтказиш ҳамда “Локал бинар образлар” операторларини ўзлаштириб, улар асосида рақамли тасвирларнинг идентификацион белгилар фазосини шакллаштиришни мақсад қилганмиз.

Юқоридагилардан хулоса қилиш мумкинки, танланган БМИ мавзуси долзарб ҳамда унинг натижалари илмий-амалий жиҳатдан аҳамиятлидир.

## **1.2. “Локал бинар образлар” операторининг рақамли тасвирлар билан боғлиқ илмий-амалий тадқиқотлардаги аҳамияти**

Олимлар дастлаб “Локал бинар образлар” (ЛБО) операторидан текстураларни таҳлил қилишда фойдаланишган. Кейинроқ эса, уни юз тасвири учун қўллаб кўришган. Ҳозирги кунга қадар ЛБОнинг бир нечта вариантлари ишлаб чиқилган. Бу йўналишда чоп этилган илмий мақолаларни таҳлил қилиб айтиш мумкинки, ЛБО оператори рақамли тасвирдаги объектларнинг белгилар фазосини ишончли аниқлай олади.

Шунингдек, ЛБО операторини қўллашгача бўлган бир қатор ишлар мавжудки, ушбу босқичдаги ишлар учун муҳим алгоритмларни ишлаб чиқиш зарур. Хусусан, тасвирда бизни қизиқтирган соҳани соҳани аниқлаш, яъни зарурий ишчи соҳаси (ишчи матрицаси)ни ажратиб олиш ва зарур ҳолларда ажратиб олинган соҳани геометрик ёки ёруғлик нуқтаи назаридан нормаллаштириш каби масалалар олдимизда туради.

Таъкидлаш керакки, ҳозирги кунда рақамли тасвирларни қайта ишлаш орқали яратилаётган интеллектуал компьютер тизимларига бўлган талаб кун сайин ортиб бормоқда. Бу каби масалаларда энг муҳим илмий-амалий масалалардан бири бу – рақамли тасвирдаги объектнинг ўзига хос идентификацион белгиларини ажратиб олишдир. Бу масала бевосита тизимнинг якуний натижасига катта таъсир кўрсатади. Сабаби, ишончли белгилар фазоси шакллангандагина, кейинги босқичдаги ишлар самарадор бўлади. Шу маънода, ЛБО оператори ишончли белгиларни аниқлашда бизга яхши кўмакчи бўлиши мумкин. Ҳозирги кунда дунё тажрибасида бу операторлардан самарали фойдаланилмоқда. Уларни янада такомиллаштириш устида олимлар яна тадқиқотлар олиб боришмоқда.

Шу сабабли ҳам бу йўналишда янги тадқиқотлар олиб бориш, ЛБО операторларидан самарали фойдаланиш кўникмасига эга бўлиш ҳамда ундан амалиётда кенг фойдаланиш масаласи долзарбдир.

### **1.3. Масаланинг қўйилиши ва унинг илмий-амалий аҳамияти**

Тадқиқот ишининг мақсади рақамли тасвирларнинг ўзига хос белгиларини “Локал бинар образлар” операторлари асосида аниқлаш алгоритмлари ва дастурий таъминот яратишдан иборатдир.

Мақсадга эришиш учун қуйидаги вазифаларни бажариш талаб этилади:

- рақамли тасвирларни қайта ишлаш усулларини ўрганиш, таҳлил қилиш;
- рақамли тасвирларда бизга керакли соҳани ажратиб олиш алгоритмларини ишлаб чиқиш;
- ЛБО оператори хусусиятларини ўрганиш ва у асосида тасвир белгилар фазосини шакллантириш;
- алгоритмлар асосида дастур таъминот яратиш.

Ишлаб чиқиладиган алгоритмлар ва дастурлар юқоридаги бўлимларда таъкидлаганимиздек, илмий-амалий аҳамият касб этади. Назарий ва амалий натижалардан ишлаб чиқаришда, таълимда, қолаверса, тасвирларни қайта ишлаш билан боғлиқ бошқа турли тармоқларда фойдаланиш мумкин бўлади.

## 2.1. Масалани ечиш учун зарур бўлган дастлабки маълумотлар

Бу бўлимда биз масалани ечиш учун бизга зарур бўлган дастлабки маълумотлар билан танишамиз. Бу маълумотлар бизга дастлабки билимларимизни мустаҳкамлашга ҳамда улар асосида қўйилган масалани ҳал этишга имконият беради.

### 2.1.1. Рақамли тасвир хусусиятлари

Тасвирлар бизнинг кўзимизга турли ранглар аралашмаларидан иборат ҳолда кўринади. Яъни, кўзимизга кўринаётган ҳар қандай объект ўзининг сиртида физик жиҳатдан турли рангларни акс эттириши натижаси ўлароқ, уларни кўз орқали англаймиз. Яъни, табиатнинг ўзи бизга кўриш неъматини берганки, бу неъмат биз яшаётган дунё неъматларининг ичида энг улуғларидан бири ҳисобланади.

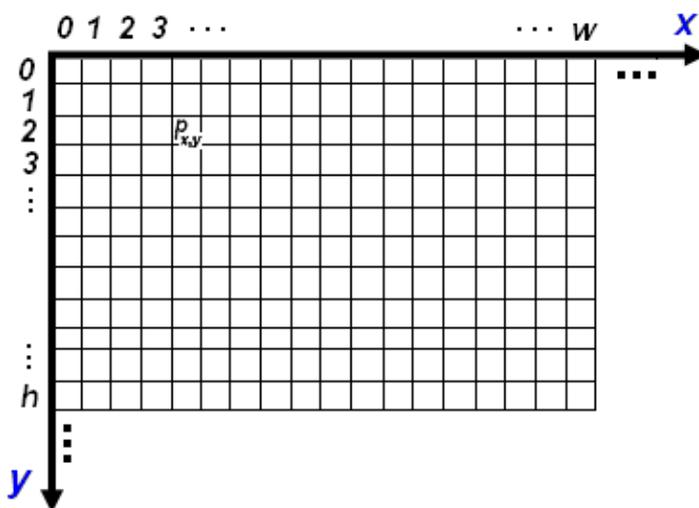
Энди, бевосита рақамли тасвирларни ҳосил қилиш технологиялари ва рақамли тасвир хусусиятлари билан қисқача танишамиз.

Рақамли тасвирнинг бевосита компьютер хотирасида қандай кўринишда сақланишини кўриб чиқамиз. Бизга маълумки, компьютерларнинг ясси экранларида тасвирлар асосан икки ўлчовли (2D) массив (ёки матрица) шаклида ифодаланади. Икки ўлчовли тасвирни матрица шаклидаги  $x$ ,  $y$  координаталар ўқиға жойлаш мумкин. Компьютерда  $y$  координата ўқининг ўсиши юқоридан пастга қараб ўзгаради. Демак, бу ҳолда координаталар боши матрицанинг чап юқори бурчагида жойлашади. Тасвирнинг эни –  $w$  ва бўйи –  $h$  каби ифодаланиб, улар матрицанинг устунлари ва қаторлари сонини ифодалайди (2.1-расм).

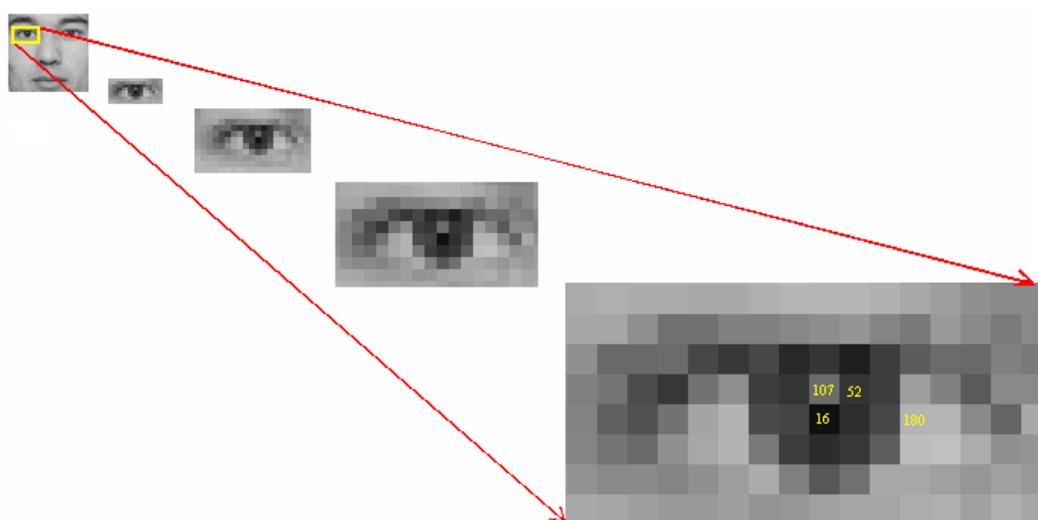
Агар тасвир *рангли* бўлса, битта пикселда учта ранг (**Red**-қизил, **Green**-яшил, **Blue**-кўк) аралашмалари мавжуд бўлиб, уларнинг ҳар бири  $0\div 255$  оралиғида тусланади (ўзгаради). Агар тасвир *кулранг* (ёки нимранг; ингл.“Grayscale”; рус.“Полутоновый”) бўлса, у ҳолда пикселдаги ранг қийматлари фақат битта ранг компонентидан иборат бўлиб, у  $0\div 255$  оралиғида,

яъни қора (0)дан оқ (255)гача бўлган ранглардан иборат бўлади. Кулранг тасвирлар ҳозирги кунда маиший ҳаётимизда деярли учрамайди. Лекин, уларни олдинроқ, яъни бир неча ўн йиллар олдин кўп учратганмиз. Масалан, олдинги (эски) телевизорларимиз ҳозиргидек рангли эмас, кулранг (одамлар тилида оқ-қора дейилган) бўлган, шунингдек, фотосуратлар ҳам худди шундай бўлган ва ҳ.к.

Кулранг тасвир (юз тасвири)нинг бир бўлагини катталаштирилган ҳолати ва ундаги пикселларда ранг қийматларининг намуна кўринишини 2.2-расмда кўришингиз мумкин.



2.1-расм. Икки ўлчовли координаталар ўқи бўйича тасвир матрицасининг кўриниши.



2.2-расм. Кулранг (юз) тасвири бўлаги ва ундаги пиксел ранг қийматлари.

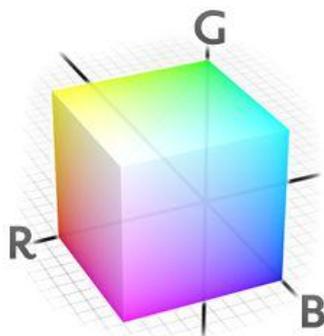
Пикселларда ранглар ифодаланар экан, ранг системалари хакида ҳам қисқача тўхталиб ўтишимиз жоиз.

Ранг системаларини (фазоларини) ишлаб чиқиш ва стандартлаштириш бўйича ҳозирги кунда махсус комиссия иш юритади. Бу комиссия Австрия давлатида жойлашган бўлиб, CIE (Commission International de l'Eclairage) халқаро комиссияси деб номланади [77]. Ҳозирги қадар бир қатор дунё олимлари томонидан ранг тадқиқотлари натижалари бўйича илмий ишланмалар шу комиссия тасдиғидан ўтган ва стандартлаштирилган. Хусусан, ранг системалари бўйича маълумотлар, уларнинг хоссалари, фойдаланиш тармоқлари, улар устида математик алмаштириш формулалари кабилар тўлиқ берилган. Бу илмий марказнинг, яъни комиссиянинг фаолиятига 100 йилдан ошди. Шу давр мобайнида халқаро комиссия томонидан кўплаб ранг системалари стандартлаштирилди. Бу ранг системалари илмий тадқиқотлар ва ишлаб чиқаришда жуда аҳамиятлидир.

Ранглар билан боғлиқ турли илмий-амалий масалалар учун турлича ранг системаларидан фойдаланилади. Биз кўпроқ компьютерлар экранида акс этувчи RGB ранг системаси асосидаги рақамли тасвирлар билан ишлаймиз. Лекин, бошқа турли амалий масалаларда бошқа ранг системалари, масалан, XYZ, Lab, HSI, HLS, CMYK кабилардан фойдаланилади. Бу ранг системалари ўзининг хоссалари ва функциясига кўра турли илмий-амалий масалаларни ҳал этишда кўл келади.

Тадқиқотларимизда асосан RGB ранг системаси билан ишлаймиз. Шу сабабли қуйида ушбу ранг системасига кенгроқ тўхталамиз.

RGB ранг системаси компьютер экрани учун мослаштирилган (техник) ранг ҳисобланади (2.3-расм). Унда уч хил ранг аралашмаси мавжуд бўлиб, улар Red (қизил), Green (яшил) ва Blue (кўк) ранглардир. Ҳар бир ранг миқдори  $0 \div 255$  оралиғида бўлади. Демак, учта рангнинг бирлик миқдорлари асосида  $256^3=16777216$  та рангни ифодалаш мумкин бўлади.



2.3-расм. Уч координатали RGB ранг фазоси.

### 2.1.2. Рақамли тасвир белгилари ҳақида

Рақамли тасвирнинг ўзига хос белгиларидан турли мақсадларда фойдаланилади. Масалан, тасвирни ёки тасвирдаги объектни таниб олиш (идентификациялаш), турли хулосалар ёки прогнозлар қилишда фойдаланиш мумкин.

Тасвирнинг таниб олинувчи (идентификацион) белгиларини тўғри ажратиб олиш муҳим аҳамият касб этади. Сабаби, ана шу белгилардан келгусида тасвирдаги объектни таниб олиш ёки бошқа илмий хулосалар қилишда фойдаланилади.

Ҳар бир тасвир ўзига хос хусусиятларга, хусусан, ўз белгиларига эга. “Белги” сўзи илмий адабиётларда инглиз тилида “Feature” ва рус тилида “Признак” каби учрайди.

Тасвир белгиларининг турлари кўп. Масалан, пиксел ёруғлиги билан боғлиқ белгилар, тасвирдаги объектнинг шакллари бўйича белгилар, маълум бир нуқталар орасидаги масофалар бўйича белгилар ва шу каби бошқа белгилардан фойдаланиш мумкин.

Белгиларни аниқлаш учун турли усуллардан фойдаланилади. Хусусан, бевосита тасвирларни қайта ишлаш усуллари, статистик формулалар, Хаф алмаштиришлари ва шу каби бошқа усуллар ёрдамида тасвирдаги белгиларни аниқлаш (топиш) мумкин.

Тасвир хусусияти ҳамда қўйилган масаланинг моҳиятидан келиб чиқиб, тасвир белгиларини турли кўринишда ва турли усуллар билан аниқланади. Қуйида бир қатор маълум ва машҳур тасвир белгилари билан танишамиз.

## Манба тасвир белгилари

Манба (дастлабки) тасвир деб компьютер хотирасига киритилган ва асосан бирор кўринишдаги ишлов берилмаган кулранг тасвирга тушунилади.

Кўпинча тасвирларни таниш таҳлил қилинганда манба тасвирдан турли тартибли асос ва ҳосила белгилар ажратиш имконини берадиган бошланғич белгилар билан чекланади. Улардан кўпроқ ишлатиладигани қуйидагилар:

1. Алоҳида нуқта ёруғлик даражаси -  $f_{i,j}$ . Бу белги асосий ҳисобланади ва деярли барча қолган белгилар унинг асосида аниқланади;

2. Тасвирдаги максимал ёруғлик даражаси -  $f_{max}$ .

3. Минимал ёруғлик даражаси -  $f_{min}$ . Бунда  $f_{min}$  ва  $f_{max}$  ларнинг қийматлари тасвир қийматлари ўзгариш оралиғини аниқлаш имконини беради;

4. Тасвирнинг ўртача ёруғлик даражаси:  $f_{\bar{y}p} = \frac{1}{MN} \sum_{i=1}^N \sum_{j=1}^M g_{ij}$ , бу ерда  $N \times M$  тасвир ўлчамларидир.

5. Маълум бир қийматли нуқталар сони -  $K_f$ . Одатда қуйидаги кўринишда жадвал тузилади:

<b>Нуқтанинг ёруғлик қиймати</b>	$X_0$	$X_1$	$X_2$	.....	$X_z$
<b>Шу қийматли нуқталар сони</b>	$K_0$	$K_1$	$K_2$	.....	$K_z$

Бу жадвал тасвир элементлари ёруғлик қийматлари гистограммасини беради.  $K_i$  қийматларни  $N \times M$  кўпайтмага бўлиб, ёруғлик тақсимоли эҳтимоллиги  $P_i$  ни олиш мумкин. Ушбу гистограмма асосида кўпсонли гистограмма белгилари (статистик белгилар) олинади, улардан энг кенг тарқалганлари қуйида келтирилган (6+11).

6. Тасвир энергияси:  $E = \sum_{i=1}^L p_i^2$ .

7. Тасвир ёруғлиги ўрта қиймати –  $f_{\bar{y}p}$ ;  $f_{\bar{y}p} = 1/MN \sum_{i=1}^N \sum_{j=1}^M f_{ij}$ , бу ерда  $N, M$  – тасвир ўлчами.

8. Тасвир ёруғлик дисперсияси -  $\sigma^2 = \sum_{i=1}^L (x_i - \sum_{i=1}^L x_i P_i)^2 P_i$ , бу ерда  $L$  - максимал ёруғлик даражаси,  $P_i$  - ўртача  $|F - \mu|$  тасвирнинг ёруғлик тақсимоти гистограммаси киймати,  $\mu$  - математик кутилма.

9. Ёруғлик қийматлари гистограммаси асимметрияси:

$$A = \frac{\sum_{i=1}^k (x_i - \sum_{i=1}^n x_i P_i)^3 P_i}{\left( \sqrt{\sum_{i=1}^n (x_i - \sum_{i=1}^m x_i P_i)^2 P_i} \right)^3} .$$

10. Ёруғлик қийматлари гистограммаси эксцесси:

$$E_x = \frac{\sum_{i=1}^n (x_i - \sum_{i=1}^m x_i P_i)^4 P_i}{\left( \sqrt{\sum_{i=1}^L (x_i - \sum_{i=1}^m x_i P_i)^2 P_i} \right)^4} .$$

11. Тасвир энтропияси:  $H = -\sum_{i=1}^L p_i \log_2 p_i$ .

12. Маълум қийматли нуқталар жуфтликлари сони. Уларни ҳисоблаш бир вақтда пайдо бўлиш матричасини (БВПБМ) ҳосил қилади, бу матрица асосида жуфтликлар тақсимотининг икки ўлчамли гистограммаси тузилади. Бу гистограмма қатор белгиларини ҳисоблаш имконини беради. Қуйида уларни базилари (13+21) келтирилган.

13. БВПБМ энергияси:  $\mathcal{E} = \sum_{i=1}^L \sum_{j=1}^L [M_{ij}[r, \theta]]^2$ .

14. БВПБМ автокорреляция коэффиценти:  $KA = \sum_{i=1}^L \sum_{j=1}^L ij M_{ij}[r, \theta]$ , бу ерда  $M_{ij}[r, \theta]$  - ВПБМ элементлари.

15. БВПБМ ковариация коэффиценти:  $KK = \sum_{i=1}^L \sum_{j=1}^L (i - \mu)(j - \mu) M_{ij}[r, \theta]$ , бу ерда  $\mu$  - математик кутилма (БВПБМ ўртача қиймати).

16. Гистограмма инерция моменти:  $\mu = \sum_{i=1}^L \sum_{j=1}^L (i - j)^2 M_{ij}[r, \theta]$ .

17. Ўртача абсолют айирма:  $AP = \sum_{i=1}^L \sum_{j=1}^L |i - j| M_{ij}[r, \theta]$ .

18. Тескари айирма:  $OP = \sum_{i=1}^L \sum_{j=1}^L M_{ij}[r, e] / (1 + (i - j)^2)$ .

19. БВПБ энтропияси:  $H = -\sum_{i=1}^L \sum_{j=1}^L M_{ij}[r, \theta] \log_2(M_{ij}[r, \theta])$ .

20. Тасвирнинг Фурье спектри:

$$F(u, v) = \frac{1}{NM} \sum_{m=1}^N \sum_{n=1}^N F(m, n) \exp\left(\frac{12\pi(um + vn)}{MN}\right),$$

бу ерда  $F(m, n)$  - тасвир элементлари.

21. Тасвирнинг Уолш спектри:  $V(u, v) = \sum_{m=1}^M \sum_{n=1}^N F(m, n) W_{u,v}(m, n);$

Уолш алмаштириш матрицаси  $W_{u,v}(m, n) = (-MxN)^{-1} \sum_{j=1}^{n-1} (m, n)_{n-1-j}^T (m, n)_j$  кўринишга

эга. Бу ерда  $(m, n)_{n-1-j}^T$  - функция номерининг аксилтартибли Грей код рақамлари,  $(m, n)_j$  – нуқта номери иккилик код рақамлари.

22. Тасвирнинг Адамар спектри:

$$A(u, v) = \sum_{m=1}^M \sum_{n=1}^N F(m, n) HAD_{u,v}(m, n);$$

бунда  $HAD_{u,v}(m, n) = 1/MN (-1)^{\sum_{j=1}^{n-1} (m, n)_j^i (m, n)_j}$ ,

$(m, n)_j^i$  - Грей кодининг (тўғри тартибли) рақамлари.

23. Уолш - Адамар спектри (УА) -  $VA(u, v)$ . Турли типдаги спектрларни ҳисоблашнинг бошқа алгоритмлари ҳам мавжуд.

24. Қўшни элементлар қийматлари айирмаси  $-\Delta f = f_1 - f_2 - f = f_1 - f_2$ . Бу айирмани ҳисоблаш манба тасвир белгилар: максимал, минимал, ўртача ёруғликлар, гистограмма белгилари, спектрлар ва ҳоказоларни аниқлаш имконини берувчи тасвир ҳосил қилади.

25. Маълум узунликли ва йўналишли бир ҳил қийматлардан ташкил топган чизиқлар сони, яъни такрорланишлар кетма-кетлиги. Бу белгидан чизиқлар узунликлари гистограммаси тузиш ва чизиқларнинг статистик хусусиятларини аниқлаш мумкин.

Турли масалаларни ечиш учун зарур бўлган белгиларни аниқловчи яна кўплаб усуллар мавжуд. Шунини айтиб ўтиш керакки, белгиларни кўпчилиги бутун тасвирни эмас, балки сатр, устун ёки бирор бўлак таҳлили асосида аниқланади.

### Нуқтавий объектлар белгилари

Нуқтавий объект деб ўзаро боғлиқ бўлган нуқталарни сони 9 тадан ошмайдиган ва мустақил аҳамиятга эга қисмларга бўлинмайдиган объектларга айтилади. Нуқтавий объект (НО)ларнинг белгиларидан кенгрок кўлланиладиганлари куйидагилар:

1. Алоҳида нуқтанинг ёруғлик қиймати:  $f_{i,j}$ .

2. НО ўртача ёруғлик қиймати:  $f_{\text{урт}} = \frac{1}{kl} \sum_{i=1}^k \sum_{j=1}^l f_{i,j}$ .

3. Тасвирдаги НО лар сони:  $n$ .

4. НО юзаси –  $s$ . У НОдаги нуқталар сонига тенг.

5. Тасвирдаги НО ларнинг умумий юзаси:  $S = \sum_{j=1}^n s_j$

6. НО оғирлик маркази координаталари:

$$x_m = 1/s \sum_i \sum_j i \cdot f_{ij}, \quad y_m = 1/s \sum_i \sum_j j \cdot f_{ij},$$

бу ерда  $f_{i,j}$  ва  $s$  лар ҳар бир объектга алоҳида аниқланади.

7. Оғирлик марказидан координата марказигача бўлган масофа:  $r_1 = \sqrt{x_m^2 + y_m^2}$ .

8. НОлар орасидаги масофа:  $r_2 = \sqrt{|x_{m1} - x_{m2}|^2 + |y_{m1} - y_{m2}|^2}$ .

9. НОлар юзалари гистограммаси. Бу гистограмма асосида тасвирда НОларнинг бир қатор статистик хусусиятлари аниқланади.

10. НОлар орасидаги масофалар гистограммаси. Уларнинг “тўп-тўп”лиги ва НОларнинг тасвирда жойлашишининг бир қатор статистик хоссаларини аниқлаш имконини беради.

11. Тасвирнинг НОлар билан тўйинганлиги:  $A = \frac{S}{MN}$ .

Бундан ташқари НО ларнинг яна бир қанча инерция моменти, ташқи чизилган тўртбурчак ўлчамлари ва бошқа белгиларини аниқлаш мумкин.

Ўз навбатида, НОли тасвирларни манба тасвир деб қараб, унда тасвирга хос белгиларни ҳам ажратиш мумкин.

### Узуқ чизиқли объектларнинг белгилари

Бу ерда узлуксиз чизиқларнинг узлуксиз дискрет кўриниши бўлган битта нуқта қалинликдаги чизиқлар кўрилади. Одатда, бундай объектли тасвирлар бинар бўлади. Шунинг учун куйида тасвирдаги бир қалинликдаги қора (объект) нуқталар муҳокама этилади ва уларнинг белгилари асосан геометрик

хусусиятларини ифодалайди. Бу белгилардан кўпроқ фойдаланиладиганлари қуйидагилардир:

1. Биринчи тартибли тугун нуқталари сони –  $k_1$ . Бу тугунлар чизиқнинг четки (бошланғич ва охири) нуқталаридир.
2. Учинчи тартибли тугун нуқталари сони –  $k_3$ . Бу  $T$  ёки  $Y$  кўринишидаги учтали шоҳлаш нуқталаридир.
3. Тўртинчи тартибли шоҳлаш нуқталари –  $k_4$ . Бу икки чизиқнинг ўзаро кесишиш нуқталаридир.
4. Чизиқ узунлиги –  $L$ . У чизиқ нуқталари сонига пропорционал;
5. Чизиқ сони –  $n$ .
6. Барча чизиқларнинг умумий узунлиги:  $L = \sum_{i=1}^n l_i$ .
7. Чизиқнинг  $(i,j)$  нуқтадаги эгрилиги  $\tau_{ij} = \frac{\Delta\varphi}{\Delta d}$ .

Бу ерда  $\Delta\varphi$ -чизиқнинг шу нуқтада ўзгариш бурчаги,  $\Delta d$  - эгри чизиқ узунлиги.

8. Чизиқларнинг бурилишлар сони –  $\alpha$ .
9. Чизиқнинг маҳаллий максимумлари сони -  $\beta$ .
10. Чизиқнинг маҳаллий минимумлари сони -  $\tau$ .
11.  $x$  ўқидаги чизиқнинг проекция узунлиги  $l^x = |x_{\max} - x_{\min}|$

Бу ерда  $x_{\max}, x_{\min}$  -  $x$  чизиқ элементлари координатасининг экстремал қийматлари.

12.  $y$  ўқидаги чизиқнинг проекция узунлиги  $l^y = |y_{\max} - y_{\min}|$ .

Бу ерда  $y_{\max}, y_{\min}$  - чизиқнинг энг катта ва энг кичик ординалари.

13. Тасвирнинг чизиқларга тўйинганлиги  $B = \frac{L}{MN}$ .
14. Чизиқларнинг четки нуқталари орасидаги энг қисқа масофа –  $r$ . Икки нуқта орасидаги масофа ифодасидан аниқланади.
15. Чизиқ учлари орасидаги масофанинг чизиқ узунлигига нисбати:  $R = r/l$ .
16. Чизиқнинг абсциссага проекцияси узунлигининг чизиқ узунлигига нисбати:  $D^x = l^x / l$ .
17. Чизиқнинг ордината проекцияси узунлигининг чизиқ узунлигига нисбати  $D^y = l^y / l$ .

Белгиларни санашни узоқ давом эттириш мумкин. Улар узунлиқлар ва эгриликлар гистограммалари асосида аниқланадиган белгилар, 1,3,4- тартибли

тугунларнинг чизиклар узунликларига ёки сонига нисбатлари, локал экстремумлар нисбатлари ва шунга ўхшаш бошқа белгилардир.

### Ёпиқ чизикли объектларнинг белгилари

Бу ерда ҳам узук чизикли объектларга айtilган гаплар ўринли. Фарқи - чизикларнинг учлари туташганлигидан ва қўшимча белгиларнинг мавжудлигидандир. Шунинг учун фақат қўшимча белгиларни муҳокама қиламиз.

1. Объектларнинг ичма-ич тартиби –  $V$ .

2. Объектларнинг марказ координаталари:

$$X_m = 1/S \sum_i \sum_j i \cdot f_{ij}, \quad Y_m = 1/S \sum_i \sum_j j \cdot f_{ij} .$$

3. Объект марказлари орасидаги масофа:  $R_{kl} = \sqrt{(x_{mk} - x_{ml})^2 + (y_{mk} - y_{ml})^2}$

4. Ташқи чизилган тўртбурчак ўлчамлари:

$$a = |x_{\max} - x_{\min}| \cos \alpha, \quad b = |y_{\max} - y_{\min}| \cos \alpha .$$

5. Объект маркази ва координата боши орасидаги масофа:  $r = \sqrt{x_m^2 + y_m^2} .$

6. Объектларни  $x$  ўқига нисбатан инерция моменти:  $\mu^x = \sum_i \sum_j f_{ij} (i - x_m)^2$

7. Объектларни  $y$  ўқига нисбатан инерция моменти:  $\mu^y = \sum_i \sum_j f_{ij} (j - y_m)^2$

8. Объектларнинг аралаш инерция моменти:  $\mu^{xy} = \sum_i \sum_j f_{ij} (i - x_m)(j - y_m)$

9. Объектларнинг асосий инерция моменти:

$$M_{1,2} = \frac{\mu^x + \mu^y}{2} \pm \sqrt{\frac{1}{4}(\mu^x - \mu^y)^2 + (\mu^{xy})^2} .$$

10. Объектларни катта ўқи узунлиги (Мартин диаметри):  $O_1 = \max_j (dF_j)$  , бу

ерда  $dF_j$  – Фере диаметри.

11. Объектнинг кичик ўқи узунлиги (Мартини эни):  $O_2 = \min_i (dF_i)$ .

12. Объект эксцентриситети:  $e = \sqrt{\frac{(I \bullet \varepsilon_1)}{(I \bullet \varepsilon_2)}} .$

Бу ерда  $\bullet$  белгиси -  $I(\mu_x, \mu_y, \mu_{xy})$ ,  $\varepsilon_1(\cos^2 \alpha, \sin^2 \alpha, -\sin^2 \alpha)$ ,  $\varepsilon_2(\sin^2 \alpha, \cos^2 \alpha, \sin^2 \alpha)$  координатали векторларнинг скаляр кўпайтмаси белгиси,  $\alpha$  – объектнинг оғиш бурчаги.

13. Объект юмалоқлиги:  $Q = \frac{P^2}{4\pi S}$ , бу ерда  $P$  - периметр,  $S$  - объект юзаси.
14. Фере диаметри:  $F: F = p_0 / \pi$ , бу ерда  $P_0$  - объектга ташқи чизилган энг кичик қавариқ чизиқнинг периметри;
15. Объект чизиғининг бўлакланганлик даражаси:  $I = \frac{n}{S}$ , бу ерда  $n$  - тасвирдаги чизиқлар сони.
16. Объектнинг квадратга мослик коэффиценти:  $J = \frac{P}{4H}$ , бу ерда  $P$ -периметр,  $H$ -объектнинг Мартин эни.
17. Объект чизиқларининг спираллик даражаси:  $C = \frac{P}{2L}$ , бу ерда  $L$  - объектнинг Мартин диаметри.

### Юзали объектларнинг белгилари

Элементлари сони тўққиздан ошиқ бўлган ва мустақил қисмларга ажратиш мумкин бўлган объектларга юзали объектлар дейилади. Уларнинг белгиларини аниқлаш учун манба тасвир белгиларини аниқлаш алгоритмларидан фойдаланиш мумкин, чунки ҳар бир объектга тасвир ёки унинг фрагменти деб қараш мумкин. Демак, куйида манба тасвирниқидан фарқли белгиларни муҳокама қиламиз.

1. Объект юзаси –  $s$ , унинг элементлар сонига тенг.
2. Объектдагри тешиқлар сони –  $d$ ;
3. Объектдаги тешиқ юзаси  $s^d$ . Тешиқ юзасидаги нуқталар сонига тенг.
4. Объектнинг умумий майдони:  $q = s + \sum_{i=1}^d s_i^d$ .
5. Тасвирдаги объектлар сони –  $n$ .
6. Барча объектларнинг йиғинди юзаси:  $S = \sum_{j=1}^n q_j$
7. Объект илма тешиқлиги:  $u = d / q$ .
8. Тасвирнинг объектларга тўйинганлиги:  $v = \frac{s}{MN}$ .
9. Объект оғирлик маркази координаталари:  
 $x_m = 1/s \sum_i \sum_j i \cdot f_{ij}$ ,  $y_m = 1/s \sum_i \sum_j j \cdot f_{ij}$ .
10. Абцисса ўқиға нисбатан инерция моменти:  $\mu^x = \sum_i \sum_j f_{ij} (i - x_m)^2$ .

$$11. \text{ Ордината ўқига нисбатан инерция моменти: } \mu^y = \sum_i \sum_j f_{ij} (j - y_m)^2 .$$

$$12. \text{ Аралаш инерция моменти } \mu^{xy} = \sum_i \sum_j f_{ij} (i - x_m)(j - y_m) .$$

$$13. \text{ Инерция бош моментлари: } M_{1,2} = \frac{\mu^x + \mu^y}{2} \pm \sqrt{\frac{1}{4}(\mu^x - \mu^y)^2 + (\mu^{xy})^2} .$$

Юзали объектлар учун айтиб ўтилганлардан ташқари Эйлер сони, объектларнинг ичма-ичлиги, объектлар орасидаги масофа, объект қобиғи параметри ва юзаси каби кўплаб белгиларни аниқлаш мумкин. Агар юзали объектларнинг контури олинса, ёпиқ чизиқли объектларга эга бўламиз, натижада юқоридаги муҳокамалар ҳам юзали объектлар учун ўринли бўлади.

### Моментлар

Тасвир белгиларини аниқлаш ва уларни таққослашда моментлардан кенг фойдаланилади. **Момент** қуйидагича ҳисобланади.

$$\mu_{pq} = \sum_{x=1}^n \sum_{y=1}^n x^p y^q f(x, y) .$$

Бу ерда  $f(x, y)$  - тасвирнинг  $x, y$  - координатасидаги пиксел қиймати.  $n$  – тасвир ўлчами (эни ва бўйи).

Кўпинча  $p$  ва  $q$  лар 0 дан 3 даражагача бўлади. Баъзи моментларнинг ўз маъноси бор. Масалан:

1. Агар  $p, q=0$  бўлса - энергия ёки нолинчи даражали момент деб аталади ва бу ҳолда  $f(x, y)$  бўйича сумма аниқланади.
2. Агар  $p=1, q=0$  ёки  $p=0, q=1$  бўлса, биринчи даражали моментлар деб аталади. Биринчи даражали моментлар орқали оғирлик марказини аниқлаш мумкин, яъни:

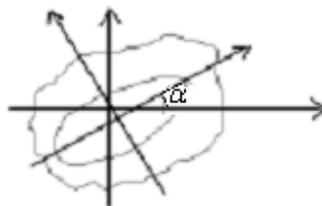
$$\bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}},$$

*ёки бу қуйидаги билан тенг кучли*

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i, \quad \bar{y} = \frac{1}{N} \sum_{i=1}^N y_i .$$

3. Агар  $p=2, q=0$  ёки  $p=0, q=2$  бўлса – инерция моменти ёки иккинчи даражали момент деб аталади. У асосий инерция ўқини аниқлайди. Ўз навбатида ушбу моментлардан фойдаланиб, асосий ўқнинг оғиш бурчагини қуйидагича аниқлаш мумкин:

$$\alpha = \frac{1}{2} \operatorname{arctg} \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} .$$



Тасвирдаги объектнинг силжишига нисбатан инвариантликни сақлаш мақсадида **марказий моментдан** фойдаланилади ва у қуйидагича ҳисобланади:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p \cdot (y - \bar{y})^q f(x, y) .$$

У ҳолда тасвирнинг бурилишига, силжишига ва ойна (зеркальной) кўринишига инвариант моментларни қуйидагича ҳисоблаш мумкин:

$$\begin{aligned} \Phi_1 &= \mu_{20} + \mu_{02} , \\ \Phi_2 &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 , \\ \Phi_3 &= (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 , \\ \Phi_4 &= (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2 , \\ \Phi_5 &= (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12}) [(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + \\ &+ (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03}) [3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] , \\ \Phi_6 &= (\mu_{20} - \mu_{02}) [(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}) , \\ \Phi_7 &= (3\mu_{21} - \mu_{03})(\mu_{30} - \mu_{12}) [(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] - \\ &- (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03}) [3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] . \end{aligned}$$

Аффин алмасиришларга, хусусан масштаблашга нисбатан инвариант моментларни эса, қуйидагича аниқлаш мумкин:

$$\eta_{pq} = \frac{\mu_{pq}}{(\mu_{00})^\gamma} , \text{ бу ерда } \gamma = \frac{p+q}{2} + 1 .$$

Бу формула даражалар  $p+q \geq 2$  бўлган ҳолат учун ўринлидир.

У ҳолда, тасвирнинг жойлашуви, масштаби ва ориентациясига нисбатан инвариант моментлар қуйидагича ҳисобланади:

$$\begin{aligned}
M_1 &= \eta_{20} + \eta_{02}, \\
M_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2, \\
M_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2, \\
M_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2, \\
M_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\
&+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2], \\
M_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}), \\
M_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} - \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - \\
&- (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].
\end{aligned}$$

### 2.1.3. “Локал бинар образлар” оператори ҳақида

“Локал бинар образлар” (ЛБО), инглиз тилида “Local Binary Pattern” (LBP) операторларидан ҳозирги кунда рақамли тасвирлар билан боғлиқ илмий-амалий масалаларда идентификацион белгиларни ажратиб олиш масаласини ҳал этишда самарали фойдаланилмоқда. Бу оператордан кулранг тасвирда кўшни пикселларнинг ранг қийматлари бўйича ўзаро ранг фарқлари бўйича муносабатларини аниқловчи ва шу асосда тасвир матрицаси учун белгилар матрицасини ҳосил қилиш учун фойдаланилади. Демак, тасвирда қанча пиксел мавжуд бўлса, ЛБО белгилари ҳам шунча миқдорда бўлади.

ЛБО операторидан тасвирларни бирга-бир таққослашда самарали фойдаланилади. ЛБО асосида тасвирларни таққослашда ЛБО гистограммаларидан фойдаланиш мумкин. Шунингдек, тадқиқотларда ЛБО матрицасини бўлақларга ажратиб олиш ва ундан кейин шу майда бўлақчалар учун қурилган гистограммаларни таққослаш тавсия этилади. Шунини таъкидлаш керакки, ЛБО матрицасини таққослашда тасвир ҳажми ёки ЛБО гистограммалари бир хил масштабга келтирилиши лозим.

Ушбу БМИ да тасвирларни таққослаш масаласига урғу берилмайди. Бизнинг мавзу таққослашгача бўлган муҳим жараён, яъни тасвирнинг идентификацион белгиларини ЛБО оператори ёрдамида ишончли аниқлаш усуллари, алгоритмлари ҳамда дастурларини ишлаб чиқишдан иборат.

Яна шуни таъкидлаш керакки, ушбу БМИда тадқиқот объекти сифатида рақамли тасвир олинган. Айнан қайсидир объект тасвири эмас, балки умумий тасвир олинган. Бу тасвир одам юзи тасвири бўлиши мумкин, бармоқ изи бўлиши мумкин, автомобил ёки шу каби бошқа объектлар бўлиши мумкин.

Мисол тариқасида, куйида (2.4-расм) одам юзи (олдиндан аниқланган ва ажратиб олинган) тасвири ва ЛБО натижаси кўрсатилган.



а)



б)

2.4-расм. Кулранг юз тасвири ва унинг ЛБО матрицаси.

## 2.2. “Локал бинар образлар” оператори ёрдамида тасвир белгиларини ажратиб олиш усули ва алгоритмлари

Олимлар дастлаб “Локал бинар образлар” (ЛБО) операторидан текстураларни таҳлил қилишда фойдаланишган. Кейинроқ эса, уни юз тасвири учун қўллаб кўришган. Ҳозирги кунга қадар ЛБОнинг бир нечта вариантлари ишлаб чиқилган. Бу йўналишда чоп этилган илмий мақолаларни таҳлил қилиб айтиш мумкинки, ЛБО оператори тасвирларнинг, хусусан юз тасвирларининг белгилар фазосини ишончли аниқлай олади.

Фараз қилайлик, бизда ажратиб олинган  $w \times h$  ўлчамли кулранг юз тасвири (ранг қийматлари 0 дан 255 оралиғида) мавжуд (2.5-расм).



2.5-расм. Кулранг юз тасвири.

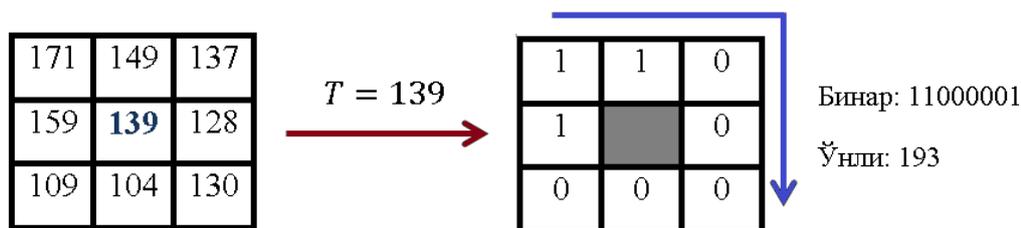
ЛБО операторида ҳисоблаш ишлари берилган кулранг тасвирнинг ҳар бир координатаси учун унинг атрофидаги 8 та қўшни пиксел ранг қийматларини ҳисобга олган ҳолда амалга оширилади.

Ҳақиқат шундай:  $3 \times 3$  ўлчамли ойнада марказий қийматни бўсаға қиймати ( $T$ ) сифатида олинади. Кейин, 8 та атрофдаги қўшни ранг қийматлари  $T$  билан

таққосланади. Агар қаралаётган қўшни ранг қиймати  $T$  дан катта бўлса, унга “1” қиймати, акс ҳолда эса, “0” қиймати берилади. Худди шу тарзда барча 8 та қўшнилари ҳам таққосланади. Натижада “0” ва “1” лардан иборат 8 та иккилик рақамлар пайдо бўлади. Шундан сўнг, бирлаштирилган 8 та рақамдан иборат иккилик сонни ўнликка ўтказилади (2.6-расм). Кулранг тасвир матрица қийматларини ЛБО оператори ёрдамида алмаштириш қуйидаги формула ёрдамида амалга оширилади.

$$LBP(x_c, y_c) = \sum_{n=0}^7 S(i_n - i_c) \cdot 2^n ; \quad S(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} .$$

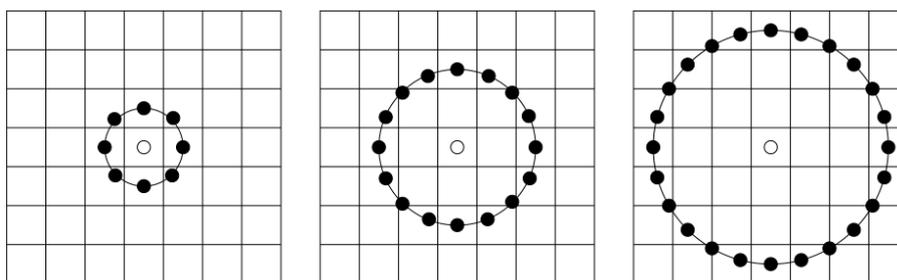
Бу ерда  $x_c, y_c$  – ишчи ойна ( $3 \times 3$ ) маркази координатаси;  $i_c$  – ойна марказидаги пиксел ранг қиймати;  $i_n$  – ойна атрофидаги 8 та қўшни пикселлар ранг қиймати.



2.6-расм. ЛБО операторининг умумий схемаси.

Таъкидлаш керакки, биз юқорида кўрган ЛБО оператори  $3 \times 3$  ўлчамли ойна учун, яъни 1 радиусли ва 8 та қўшнили ойна учун мўлжалланган. Радиус ва қўшнилари қиймати ўзгарган ҳолатлар учун ҳам ЛБОни ҳисоблаш мумкин. 2.7-расмда 1, 2 ва 3 радиусли ишчи ойналарни кўришимиз мумкин. Бунда кўриш мумкинки, радиуслар ортиши билан қўшни нуқталар сони ҳам ортиб боради.

ЛБО оператори ёрдамида юз тасвирини кодлаштириш натижаларидан намуналарни 2.1-жадвалда кўришимиз мумкин.

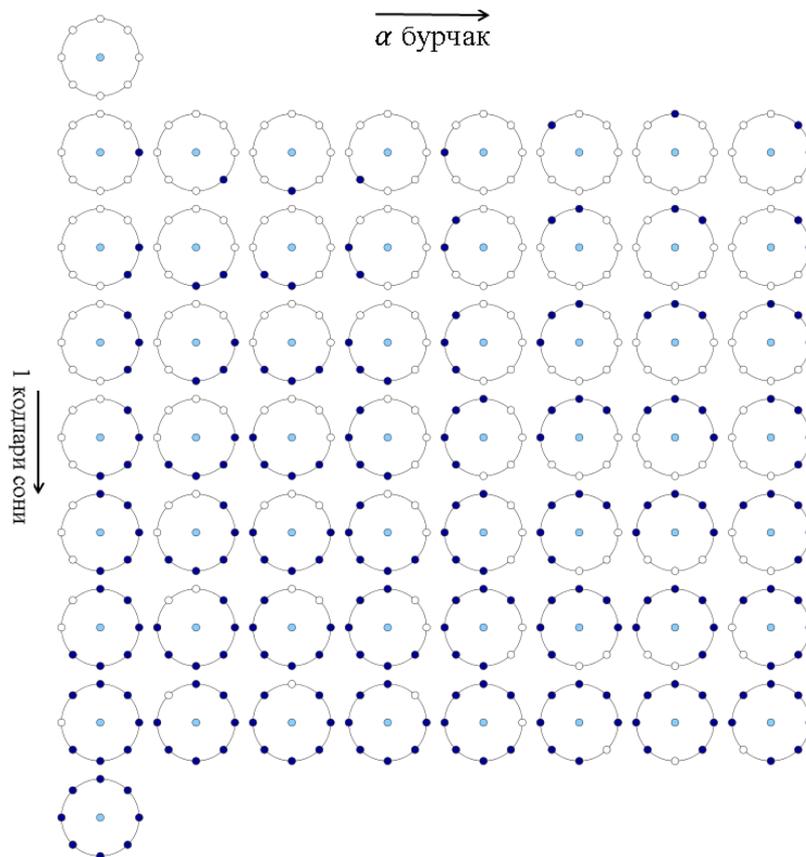


2.7-расм. 1, 2 ва 3 радиусли ишчи ойналар.

Кулранг тасвир	1 радиусли ЛБО	2 радиусли ЛБО
		
		
		
		
		

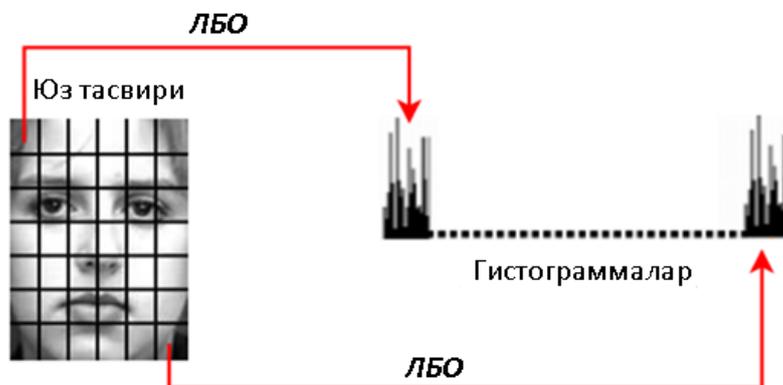
Шунингдек, тадқиқотчилар ишчи ойнадаги бинар қийматли образларни турларга ажратишган. Тасвирдаги муҳим белгиларни ажратиш учун “Текис йўналган” (Uniform) образлардан фойдаланилади. “Uniform” образ деб, 8 та рақамдан иборат иккилик кодларда “0” дан “1” га ёки аксинча ўтишлар сони 2 тадан ошмаган ҳолатларга айтилади. Масалан, 11000001 коди - Uniform, чунки ундаги ўтишлар сони 2 та, яъни унда “1” дан “0” га, кейин яна “0” дан “1” га ўтиш мавжуд. 10011000 коди эса Uniform эмас, ундаги ўтишлар сони 3 га тенг.

2.8-расмда “Текис йўналган” 58 та образлар келтирилган. Унда иккилик коднинг “1” рақами миқдори ҳамда уларнинг қандай бурчак остида жойлашганлигини кўришимиз мумкин.



2.8-расм. “Текис йўналган” (Uniform) 58 та образлар.

ЛБО белгилар фазосини шакллантириш ЛБОга алмаштирилган тасвир матрицаси учун гистограммалар қуриш орқали амалга оширилади. Кўпинча ЛБО тасвир матрицаси бир нечта бўлақларга ажратилади ва ҳар бир бўлақ учун гистограммалар қурилади. Масалан, юз тасвирини  $7 \times 7$  ўлчамда бўлақлаб олиш ва 49 та тасвир бўлағи учун алоҳида ЛБОларни ҳисоблаш ҳамда улар асосида белгилар векторини қуриш мумкин (2.9-расм).



2.9-расм.  $7 \times 7$  бўлақли юз тасвири учун ЛБО гистогрмаалаларни қуриш.

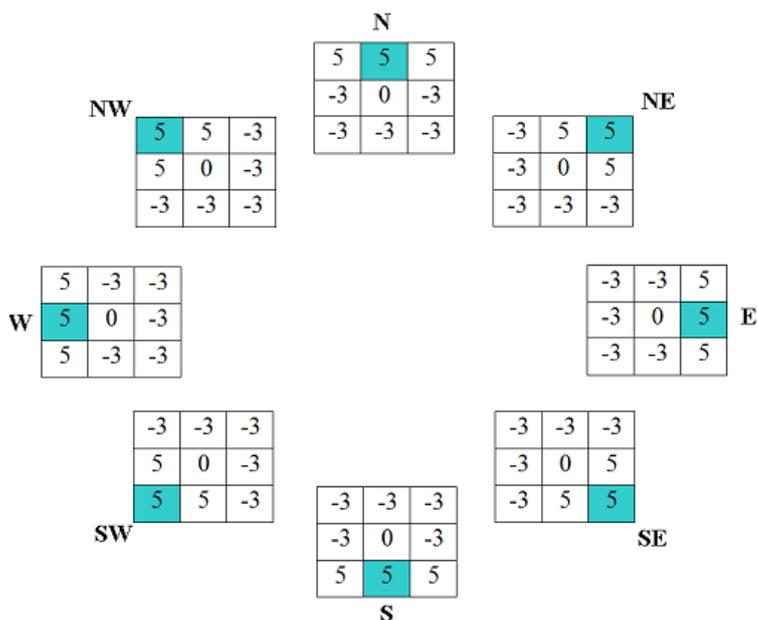
### 2.3. “Локал йўналган образлар” оператори ёрдамида тасвир белгиларини ажратиб олиш усули ва алгоритмлари

Ушбу бўлимда ЛБО оиласига мансуб бўлган, ЛБО дан бироз фарқ қилувчи “Локал йўналган образлар” (ЛЙО) оператори билан танишамиз. Бу оператор ҳақида тўлиқроқ маълумотларни инглиз тилидаги манбалардан топишимиз мумкин.

Ҳозирги кунда “Локал йўналган образлар” - ЛЙО (инглизчада “Local Directional Pattern” - LDP) операторидан рақамли тасвирларни тавсифлашда, хусусан, таниб олишда кенг фойдаланилмоқда. Айниқса, бу оператордан юз тасвирларининг муҳим белгиларини ажратиб олишда самарали фойдаланилмоқда.

ЛЙО тасвир ҳалақитларига, айниқса, тасвирнинг нотекис ёруғлик тақсимланишига турғун бўлган усул ҳисобланган.

ЛЙО операторида чегара ажратувчи Кирш (Kirsch) ниқобларидан фойдаланилади. Бизга маълумки, Кирш ниқобларидан 3×3 ишчи ойнанинг марказига нисбатан 8 та қўшни нуқталар қийматларини ўзгартириш (ниқоб ёрдамида қайта ҳисоблаш) орқали чегаравий нуқталарни аниқлашда фойдаланилади. Кирш ниқобларини қуйида кўришимиз мумкин (2.10-расм).



2.10-расм. Чегара аниқловчи 8 йўналишли Кирш ниқоблари.

Бу ерда *N* – North (Шимол), *S* - South (Жануб), *W* – West (Ғарб), *E* – East (Шарқ), *NW* - North West (Шимолий ғарб), *SE* - South East (Жанубий шарқ), *SW* - South West (Жанубий ғарб), *NE* - North East (Шимолий шарқ) йўналишларини англатади.

Тасвирнинг ҳар бир пиксели ва унинг саккизта атроф қийматлари Кирш ниқоблари бўйича қайтадан ҳисобланади. Масалан, кулранг тасвирда қаралаётган локал соҳада 3×3 ишчи ойна қийматлари қуйидагича бўлсин (2.11-расм).

25	52	31
210	40	60
180	125	77

2.11-расм. Дастлабки (намуна) локал соҳа қийматлари.

Дастлаб, *N* – North (Шимол) ниқобини қўллаймиз. Ҳисоблаш қуйидагича бўлади:

$$N = (5 * 25) + (5 * 52) + (5 * 31) + (-3 * 60) + (-3 * 77) + (-3 * 125) + (-3 * 180) + (-3 * 210) + (0 * 40) = -1416.$$

Демак, локал соҳанинг шимолида жойлашган 52 сони ўрнига -1456 сонини қўямиз. Худди шу тартибда дастлабки локал соҳанинг бошқа йўналишлари учун ҳам мос бўлган Кирш ниқоблари ҳисоблаб чиқилади. У ҳолда 4-расмдаги локал соҳанинг 8 та йўналишлари учун қийматлар қуйидагича ўзгаради (2.2-жадвал):

2.2-жадвал. ЛЙО натижалари.

Йўналиш	NW- Шимолий ғарб	N– Шимол	NE- Шимолий шарқ	E– Шарқ	SE- Жанубий шарқ	S– Жануб	SW- Жанубий ғарб	W– Ғарб
Пиксел қиймати	25	52	31	60	77	125	180	210
Ниқоб қиймати	16	-1456	-1136	-936	-184	776	1840	1040

Шундай қилиб, локал соҳа қийматлари бутунлай ўзгарди (2.12-расм). Таъкидлаш керакки, Кирш ниқобларида марказ доим 0 га тенг бўлади.

16	-1456	-1136
1040	0	-936
1840	776	-184

2.12-расм. Натижа – ниқоб қийматлари.

Энди бу қийматларни бинарлаштириш амалга оширилади. ЛЙОга бағишланган мақолаларда бинарлаштиришнинг турли вариантлари таклиф этилган. Тадқиқотлар давомида улар ичидан энг самарадорини танлаб олдик. У билан қуйида танишамиз.

Дастлаб, саккизта натижавий ниқоб қийматлари ўсиш тартибида тартибланади ва битта тўпلام (массив)га жойлаштирилади, яъни:

$$G^{(8)} = \{g_1, \dots, g_4, g_5, \dots, g_8\}.$$

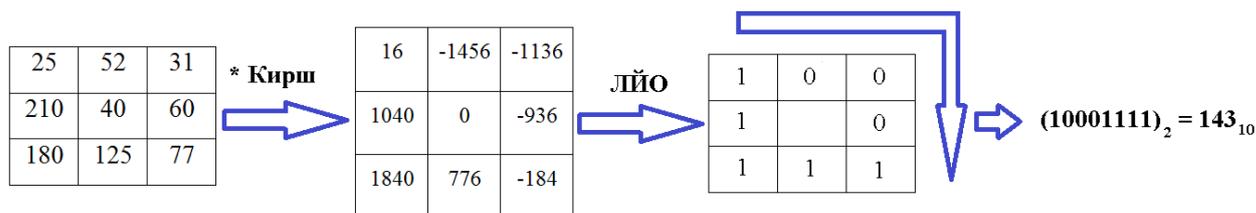
Натижада юқорида ҳисоблаб чиққан қийматларимиз қуйидагича кўриниш олади:

$$G^{(8)} = \{-1456, -1136, -936, -184, 16, 776, 1040, 1840\}.$$

Тартибланган саккизта элементнинг биринчи ярмини, яъни дастлабки 4 та кичик қийматларни “0” га, қолганлари (катталари)ни “1” га алмаштираемиз. Натижада дарчада саккизта “0” ва “1” рақамлар пайдо бўлади. Уларни соат стрелкаси бўйлаб кетма-кет жойлаймиз ва 8 та рақамдан иборат иккилик сонга эга бўламиз. Сўнгра, ушбу иккилик сонни ўнли сонга ўтказамиз, яъни:

$$LDP(x_c, y_c) = \sum_{i=1}^8 S(g_i) \cdot 2^i, \quad S(x) = \begin{cases} 1 & \text{агар } x \geq g_4, \\ 0 & \text{акс холда.} \end{cases}$$

ЛЙО операторининг умумий иш схемаси (2.13-расм) қуйидагича бўлади (намуна қийматлар мисолида):



2.13-расм. ЛЙО операторининг умумий иш схемаси.

Демак, юқоридаги ҳисоблашлар орқали битта ўнли сонга (0 дан 255 гача ораликда) эга бўламиз. Кейин биз бу сонни ойна марказига жойлаймиз. У ҳолда, юқорида намуна сифатида олган дастлабки локал соҳанинг марзий қиймати – 40 сони ўрнига 143 сони алмашади.

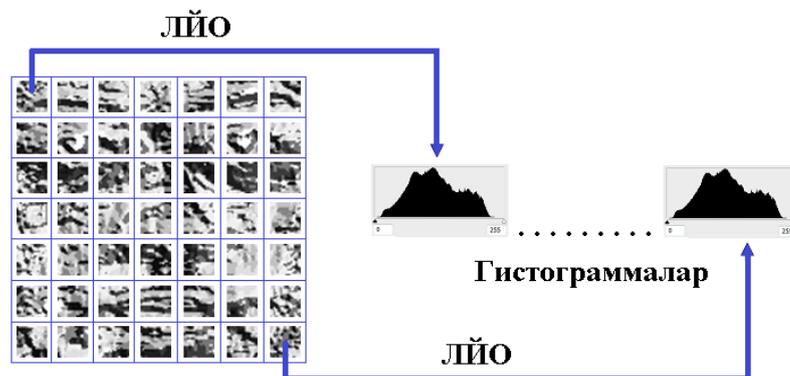
Демак, ЛЙО оператори асосидаги ҳисоблаш ишлари тасвирнинг бошқа пикселлари учун ҳам амалга оширилади ва натижада ЛЙО тасвири (ёки матрицаси) ҳосил бўлади. Қуйида ажратиб олинган юз тасвири (2.14а-расм) ҳамда унинг ЛЙО натижаси (2.14b-расм) кўрсатилган.



а) б)  
2.14-расм. Юз тасвири (а) ва унинг ЛЙО натижаси (б).

#### 2.4. ЛБО ва ЛЙО белгиларни таққослаш усули

Идентификацион белги сифатида ЛБО ёки ЛЙО матрицаси (тасвири)нинг гистограммаси олинади. Лекин, тасвирнинг гистограммасини тўлалигича олиш келгусида таниб олиш натижасига салбий таъсир кўрсатиши мумкин. Шу сабабли ҳам тадқиқотчилар ЛБО (ёки ЛЙО) тасвирни дастлаб  $N$  та бўлакларга ажратиб олишади ва шундан кейин ҳар бир бўлак учун гистограммалар ҳисобланади. Кўпинча  $7 \times 7$  ўлчамда бўлаклар амалга оширилади (масалан, юз тасвири учун). У ҳолда  $7 * 7 = 49$  та гистограммалар ҳосил бўлади ва улар кетма-кет бирлаштирилади. Бу жараённинг схемаси қуйида (2.15-расм) кўрсатилган.



2.15-расм. 7×7 ўлчамли ЛЙО асосида белгилар фазосини куриш.

Белгиларини, яъни гистограммаларни таққослаш учун бир қатор усуллардан фойдаланиш мумкин. Масалан, *Корреляция (Correlation)*, *Хи-квадрат (Chi-Square)*, *Кесимиша (Intersection)*, *Бхаттачария масофаси (Bhattacharyya distance)* ва ҳ.к. Тадқиқотларда *Хи-квадрат* усули самаралироқ эканлиги аниқланган. Қуйида *Хи-квадрат* формуласини кўрамиз.

$$\chi^2(H_1, H_2) = \sum_{i=0}^{255} \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)} .$$

Агар тасвирлар ўхшаш бўлса, юқоридаги формула асосида олинадиган натижа 0 га яқин бўлади. Таъкидлаш керакки, тасвирларни таққослашдан олдин уларни бир хил ўлчамга келтириш (масштаблаш орқали) зарур.

### 3.1. Дастурий таъминот ишлаб чиқишда фойдаланиладиган дастурий ва техник воситалар таҳлили

#### 3.1.1. OpenCV кутубхонасининг имкониятлари

Ҳозирда рақамли тасвирларни очиш, намоиш қилиш, таҳрирлаш ва қайта ишлашга мўлжалланган турли дастурлаш воситалари ёки кутубхоналар мавжуд. Улар орасида OpenCV кутубхонаси кенг тарқалган. Чунки, бу кутубхона очик кодли, текин ва кенг имкониятли ҳисобланади. Шу сабабли, биз ҳам ушбу кутубхона функцияларидан фойдаландик. Қуйида ушбу кутубхона ҳақида қисқача тўхталамиз.

OpenCV (*Open Source Computer Vision Library*) бу – компьютерли кўриш ва тасвирларга ишлов бериш алгоритмлари ҳамда сонли алгоритмларнинг очик кодли библиотекаси. С/С++ да яратилган. Илмий ва тижорий мақсадларда эркин фойдаланиш мумкин.

OpenCV библиотекасининг асосий модуллари қуйидагилардан иборат: **cxcore** - ядро, асосий маълумотлар тузилмалари ва алгоритмларини ўз ичига олади, яъни:

- кўп ўлчовли сонли массивлар устида асосий амаллар;
- матрицали алгебра, математик функциялар, тасодифий сонлар генератори;
- хотирага ёзиш / XML га/дан маълумотлар тузилмасини тиклаш;
- 2D графиканинг асосий функциялари.

**CV** – тасвирларга ишлов бериш ва компьютерли кўриш модули. Қуйидаги функциялари мавжуд:

- тасвирлар устида асосий амаллар (филтрлашлар, геометрик алмаштиришлар, ранг фазоларини алмаштириш ва ҳ.к.);
- тасвирларни таҳлил қилиш (фарқли белгиларни танлаш, морфология, контур ажратиш, гистограммалар);
- ҳаракат таҳлили, объектни кузатиш;
- объектларни топиш, хусусан юз тасвирини;
- камераларни калибрлаш, фазовий тузилмаларни қайта тиклаш элементлари.

**Highgui** - тасвирлар ва видеони киритиш/чиқариш, фойдаланувчи интерфейсини яратиш модули. Қуйидаги функцияларни ўз ичига олади.

- камерадан ва видео файлдан тасвирни олиш, статик тасвирларни ўқиш/ёзиш;
- оддий UI ни ташкиллаш учун функциялар.

OpenCV нинг кейинги версияларида модуллар функционал фойдаланиш бўйича кичикроқ модулларга ажратилди. Хусусан:

**opencv\_core** – ядро: асосий тузилмалар, ҳисоблашлар (математик функциялар, тасодифий сонлар генерацияси, дискрет Фурье ва Косинус алмаштиришлар, XML да киритиш/чиқариш в ш.к.);

**opencv\_imgproc** – тасвирларга ишлов бериш (филтрлар, алмаштиришлар ва ҳ.к.);

**opencv\_highgui** – оддий UI, тасвирлар ва видеони юклаш/сақлаш;

**opencv\_ml** – машинали ўқитиш усуллари ва моделлари (SVM, қарор қабул қилиш ва ҳ.к.);

**opencv\_features2d** – турли идентификаторлар (SURF).

**opencv\_video** – объект харакатини таҳлил қилиш ва кузатиш (оптик оқим, харакат андозаси, фонни йўқотиш);

**opencv\_objdetect** – тасвирда объектларни топиш (Хаар вейвлети, HOG ва ҳ.к.);

**opencv\_calib3d** – камера калибровкаси, стерео-мосликни излаш ва уч ўлчовли маълумотларни қайта ишлаш элементлари;

**opencv\_flann** – яқин қўшниларни тезкор излаш кутубхонаси (FLANN);

**opencv\_gpu** – OpenCV нинг баъзи функцияларини CUDA (NVidia) ҳисобига тезлатиш.

### **OpenCVда кўп фойдаланиладиган функциялар**

Тасвирларни қайта ишлаш ва намойиш қилиш учун OpenCV да махсус типлар мавжуд бўлиб, асосан *IplImage* ва *Mat* типларидан кўп фойдаланилади. Қуйида кўриб чиқиладиган функциялар асосан ана шу типли тасвирларни қайта

ишлашга мўлжалланган. Функцияларни кичик-кичик дастур намуналари негизида кўриб чиқамиз.

Масалан, 620 қатор ва 440 та устундан иборат 8 битли, 3 каналли *IplImage* ҳамда *Mat* типли тасвирларни яратиш учун қуйидаги функциялардан фойдаланамиз:

```
int height =620; // тасвир бўйи
int width =440; // тасвир эни
IplImage* tasvir = cvCreateImage(cvSize(height, width),8,3); // IplImage типли tasvir номли тасвирни
яратиш
Mat matritsa(Size(height, width), CV_8UC3); // Mat типли matritsa номли тасвирни яратиш
```

Тасвирни қора ранг билан бўяш қуйидагича:

```
cvSet(tasvir,cvScalar(0,0,0));
```

Қуйида эса, дискдан тасвир файлини хотирага юклаш ва экранда намойиш қилиш учун кичик дастурни кўрамиз:

```
IplImage* img = cvLoadImage("c:\\tasvir.jpg"); // Дискдан тасвирни img га ўқиш
cvNamedWindow("Example1", CV_WINDOW_AUTOSIZE ); // Example1 намойиш ойнасини яратиш
cvShowImage("Example1", img ); // намойиш ойнасига img тасвирни юклаш ва кўрсатиш
cvWaitKey(0); // кутиб туриш
cvReleaseImage( &img ); // img тасвирни хотирадан ўчириш
cvDestroyWindow("Example1"); // Example1 намойиш ойнасини хотирадан ўчириш
```

Тасвирни бир неча маълумотлари билан намойиш қилиш дастури қуйида берилган:

```
#include <cv.h>
#include <highgui.h>
#include <stdlib.h>
#include <stdio.h>

IplImage* image =0;
IplImage* src =0;

int main(int argc, char* argv[])
{
    // тасвир номи биринчи параметр билан берилмоқда
    char* filename = argc ==2? argv[1]:"Image0.jpg";
    // тасвирани оламиз
    image = cvLoadImage(filename,1);
    // тасвирни клонлаймиз (нўсхалаймиз)
    src = cvCloneImage(image);

    printf("[i] image: %s\n", filename);
    assert( src !=0);

    // тасвирни намойиш қилиш учун ойна
    cvNamedWindow("original", CV_WINDOW_AUTOSIZE);

    // тасвирни кўрсатамиз
    cvShowImage("original", image);
```

```

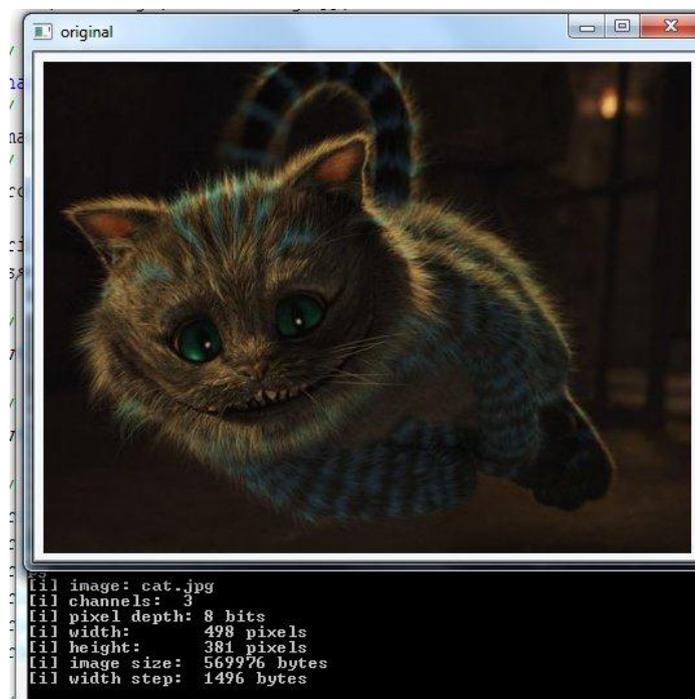
// тасвир тўғрисидаги ахборотларни (консол ойнасида) чиқарамиз
printf("[i] channels:  %d\n",      image->nChannels );
printf("[i] pixel depth: %d bits\n",  image->depth );
printf("[i] width:      %d pixels\n", image->width );
printf("[i] height:     %d pixels\n", image->height );
printf("[i] image size: %d bytes\n",  image->imageSize );
printf("[i] width step: %d bytes\n",  image->widthStep );

// тугма босилишини кутамиз
cvWaitKey(0);

// ресурсларни бўшатамиз
cvReleaseImage(& image);
cvReleaseImage(&src);
// ойнани щчираамиз
cvDestroyWindow("original");
return 0;
}

```

Натижа куйидагича (3.1-расм.) бўлади.



3.1-расм. Тасвирни маълумотлари билан намойиш қилиш.

Кейинги дастур видео-файлни намойиш қилиш (овозсиз, ҳар бир кадрни кетма-кет кўрсатиш) учун хизмат қилади:

```

cvNamedWindow( "Example2", CV_WINDOW_AUTOSIZE ); // "Example2"намойиш ойнасини яратиш
CvCapture* capture = cvCreateFileCapture("d:\video.avi"); // видеофайлни хотирага юклаш
IplImage* frame; // тасвир учун хотирадан жой ажратиш
while(1) { // тугалланмас такрорлаш жараёнини бошлаш (цикл боши)
frame = cvQueryFrame( capture ); // видеодан битта кадрни (тасвирни) олиш
if( !frame ) break; // агар тасвирни ололмаса, циклдан чиқиб кетиш
cvShowImage( "Example2", frame ); // битта кадрни (тасвирани) намойиш қилиш
char c = cvWaitKey(33); // 33 секунд кутиб туриш (кадрлар орасидаги вақт)
}

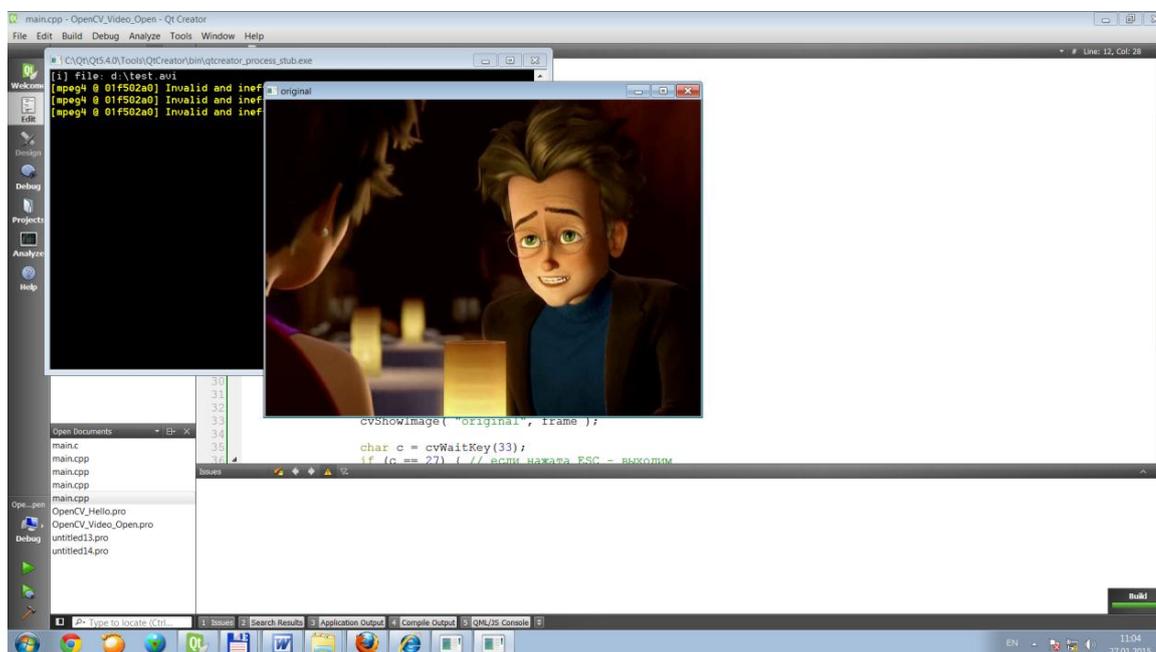
```

```

if( c == 27 ) break;           // агар Esc босилса, чиқиб кетиш
}                               // цикл охири
cvReleaseCapture( &capture ); // видеони хотирадан тозалаш
cvDestroyWindow( "Example2" );//Example2 намойиш ойнасини хотирадан ўчириш

```

Натижа қуйидагича (3.2-расм.) бўлади.



3.2-расм. OpenCVда видеофайлни намойиш қилиш.

Қуйидаги функция эса, тасвирни Гауссли текислаш (фильтрлаш) учун хизмат қилади:

```

cvSmooth( image, out, CV_GAUSSIAN, 3, 3 ); // image кирувчи тасвир 3×3 ойна ўлчамида филтрланади
// ва натижа тасвир out га берилади.

```

Қуйидаги дастур ёрдамида дастлабки тасвирни (3.3-расм) ҳамда уни 7×7 ойна ўлчамида филтрлаб (3.4-расм) намойиш қилинади:

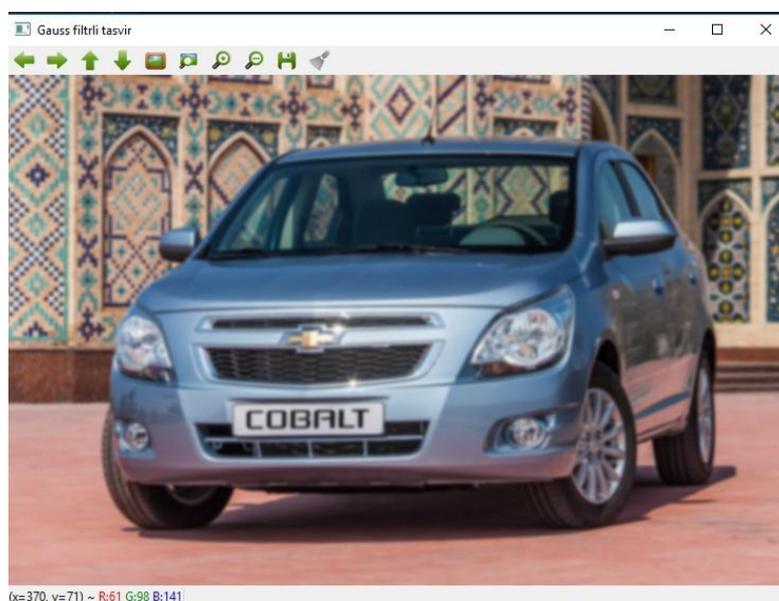
```

IplImage* img = cvLoadImage("c:\\car\\2.jpeg");
cvShowImage("Kiruvchi tasvir", img);
IplImage* out = cvCreateImage(cvGetSize(img), IPL_DEPTH_8U, 3);
cvSmooth( img, out, CV_GAUSSIAN, 7, 7 );
cvShowImage("Gauss filtrli tasvir", out);
cvWaitKey();

```



3.3-расм. Дастлабки тасвир.



3.4-расм. 7×7 Гаусс фильтри қўллангандан сўнг.

Шу ўринда OpenCV нинг оддий маълумотлар типлари тўғрисида қисқача тўхталамиз.

Структура	Ўз ичига олади	Ифодалайди
CvPoint	int x, y	Тасвирдаги нуқта
CvPoint2D32f	float x, y	$R^2$ даги нуқта
CvPoint3D32f	float x, y, z	$R^3$ даги нуқта
CvSize	int width, height	Тасвир ҳажми
CvRect	int x, y, width, height	Тасвир бўлаги (қисми)
CvScalar	double val[4]	RGBA қийматлар

Масалан, *myImg* тасвирида чап юқори бурчаги (5, 10), яъни  $x=5$ ,  $y=10$  ва ўнг куйи бурчаги (20, 30) координатали тўртбурчакни оқ рангда чизиш учун куйидагича дастур кодини ёзиш керак бўлади:

```
cvRectangle( myImg, cvPoint(5,10), cvPoint(20,30), cvScalar(255,255,255) );
```

OpenCV да куйидагича тасвир турлари мавжуд:

Тасвир типи (Macro)	Тасвир пиксел типи
IPL_DEPTH_8U	Unsigned 8-bit integer (8u)
IPL_DEPTH_8S	Signed 8-bit integer (8s)
IPL_DEPTH_16S	Signed 16-bit integer (16s)
IPL_DEPTH_32S	Signed 32-bit integer (32s)
IPL_DEPTH_32F	32-bit floating-point single-precision (32f)
IPL_DEPTH_64F	64-bit floating-point double-precision (64f)

Куйидаги жадвалда матрица ва тасвирнинг асосий операторлари келтирилган.

Функция	Таърифи
cvAbs	Массив (матрица) даги барча элементларнинг абсолют қиймати
cvAbsDiff	Иккита массив ўртасидаги фарқларнинг абсолют қиймати
cvAbsDiffS	Массив ва скаляр ўртасидаги фарқларнинг абсолют қиймати
cvAdd	Иккита массивни элементлари бўйича қўшиш
cvAddS	Массив ва скалярни элементлари бўйича қўшиш
cvAddWeighted	Иккита массивни элементлари бўйича вазн (вес)ли қўшиш (алфа қориштириш)
cvAvg	Массивдаги барча элементларнинг ўртача қиймати
cvAvgSdv	Массивдаги барча элементларнинг абсолют қиймати ва стандарт четлашиши
cvCalcCovarMatrix	n-ўлчовли вектор тўпламининг ковариансини ҳисоблаш
cvCmp	Иккита массивнинг барча элементларига танланган таққослаш операторини қўллаш.
cvCmpS	Скалярга нисбатан массивга танланган таққослаш операторини қўллаш.
cvConvertScale	Қийматнинг танловли қайта ўлчови орқали массив типини ўзгартириш.
cvConvertScaleAbs	Танловли қайта ўлчов орқали абсолют қийматдан кейин массив типини ўзгартириш.
cvCopy	Бир массивдаги элементларни бошқасига нусхалаш.
cvCountNonZero	Массивдаги нолга тенг бўлмаган элементларни ҳисоблаш.
cvCrossProduct	Уч ўлчовли иккита векторни кўпайтириш.
cvCvtColor	Массив ранг каналларини бир ранг фазосидан бошқасига ўзгартириш (алмаштириш).
cvDet	Квадрат матрицанинг детерминантини ҳисоблаш
cvDiv	Элементлар бўйича бир массивни бошқасига бўлиш.
cvDotProduct	Иккита векторни скаляр (ички) кўпайтириш.

cvEigenVV	Квадрат матрицанинг хос сон ва хос векторларини Ҳисоблаш
cvFlip	Танланган ўқ бўйича массивни тўнтариш (ойнали намоиш қилиш).
cvGEMM	Умумлашган матрица кўпайтириш.
cvGetCol	Массивнинг устун кесимидан элементларни нусхалаш
cvGetCols	Массивнинг турли қўшни устунларидан элементларни нусхалаш.
cvGetDiag	Диагонал бўйича массив элементларини нусхалаш.
cvGetDims	Массив ўлчамини қайтаради.
cvGetDimSize	Массивнинг барча ўлчамлари ҳажмини қайтаради.
cvGetRow	Массивнинг қатор кесимидан элементларни нусхалаш
cvGetRows	Массивнинг турли қўшни қаторларидан элементларни нусхалаш.
cvGetSize	Икки ўлчовли массив ҳажмини олиш ва уни CvSize шаклида қайтариш.
cvGetSubRect	Массив бир бўлаги элементларини нусхалаш.
cvInRange	Массив элементини бошқа иккита массив қийматлари ичида мавжудлигини текшириш.
cvInRangeS	Массив элементини бошқа иккита скаляр ўртасида мавжудлигини текшириш.
cvInvert	Квадрат матрицани инвертлаш (барча элемент қийматлари 255 дан айрилади).
cvMahalonobis	Иккита вектор орасидаги Махалонобис масофасини ҳисоблаш.
cvMax	Иккита массивнинг элементлари бўйича махсимумлар операцияси.
cvMaxS	Массив ва скаляр ўртасида элементлари бўйича махсимумлар операцияси.
cvMerge	Бир нечта бир каналли тасвирларни битта кўп каналли тасвирга бирлаштириш.
cvMin	Иккита матрицада элементлар бўйича минимумлар операцияси
cvMinS	Массив ва скаляр ўртасида элементлар бўйича минимумлар операцияси
cvMinMaxLoc	Массивда минимум ва максимум қийматларни топиш.
cvMul	Иккита массивни элементлар бўйича кўпайтириш.
cvNot	Массивни ҳар бир элементини битли инвертлаш.
cvNorm	Иккита массив ўртасидаги нормаланган корреляцияни ҳисоблаш.
cvNormalize	Массивдаги элементларни бирор сонгача номаллаш.
cvOr	Иккита массивнинг элементлари бўйича “ЁКИ” (OR) бит-даражаси.
cvOrS	Массив ва скалярнинг элементлари бўйича “ЁКИ” (OR)бит-даражаси.
cvReduce	Берилган амал бўйича икки ўлчовли массивни векторга айлантирмоқ.
cvRepeat	Бир массив мажмуини бошқасига қуриш.
cvSet	Массивнинг барча элементларига берилган сон қийматини ўрнатиш.
cvSetZero	Массивнинг барча элементларига 0 қиймати ўрнатилади (берилади).
cvSetIdentity	Массивнинг барча элементларига диагонал бўйича 1,

	қолганига эса 0 қиймати берилади.
cvSolve	Чизиқли тенгламалар системасини ечиш.
cvSplit	Кўп каналли массивни турли бир каналли массивларга бўлиш.
cvSub	Бир массивдан иккинчисини элементлар бўйича айириш.
cvSubS	Скалярни массивдан элементлар бўйича айириш.
cvSubRS	Массивни скалярдан элементлар бўйича айириш.
cvSum	Массивни барча элементлари суммаси (йиғиндиси).
cvSVD	Икки ўлчовли массивнинг ягона бўлиниш қийматини ҳисоблаш.
cvSVBkSb	Тескари алмаштиришнинг ягона қийматини ҳисоблаш.
cvTrace	Массив белгиларини ҳисоблаш.
cvTranspose	Массивнинг барча элементларини диагонал бўйича ўзгартириш.
cvXor	Иккита массив элементлари бўйича XOR (истисно ЁКИ) бит – даражаси.
cvXorS	Массив ва скаляр ўртасидаги массив элементлари бўйича XOR бит – даражаси.
cvZero	Массивнинг барча элементларига 0 қиймати берилади.

### OpenCV да чизиш

Баъзан тасвир устида бирор фигурани чизиш керак бўлиб қолади. OpenCV да тўғри чизиқ, тўртбурчак, айлана ва шу каби геометрик фигураларни чизиш функциялари мавжуд. Улар билан қуйида танишамиз.

*Тўғри чизиқ:*

```
void cvLine( CvArr* array, CvPoint pt1, CvPoint pt2, CvScalar color,
            int thickness = 1, int connectivity = 8 );
```

Бу ерда CvArr – тасвир, pt1 – чизиқнинг бошланиш координатаси, pt2 – чизиқнинг якунланиш координатаси, color – чизиқ ранги, thickness – чизиқ калинлиги, connectivity – чизиқ тури.

*Тўғри тўртбурчак:*

```
void cvRectangle( CvArr* array, CvPoint pt1, CvPoint pt2, CvScalar color, int thickness = 1 );
```

*Айлана:*

```
void cvCircle( CvArr* array, CvPoint center, int radius, CvScalar color, int thickness = 1, int connectivity = 8 );
```

Бу ерда center – айлана маркази, radius – айлана радиуси.

*Эллипс:*

```
void cvEllipse( CvArr* img, CvPoint center, CvSize axes, double angle,
               double start_angle, double end_angle, CvScalar color,
               int thickness = 1, int line_type = 8 );
```

Бу ерда axes – ўқ ҳажми, start\_angle – бошланғич бурчак, end\_angle – охири бурчак.

*Полигонлар:*

Полигонларни қуйидаги уч хил функция ёрдамида чизиш мумкин.

```
void cvFillPoly( CvArr* img, CvPoint** pts, int* npts, int contours, CvScalar color, int line_type = 8 );
```

```
void cvFillConvexPoly( CvArr* img, CvPoint* pts, int npts, CvScalar color, int line_type = 8 );
```

```
void cvPolyLine( CvArr* img, CvPoint** pts, int* npts, int contours, int is_closed,  
                CvScalar color, int thickness = 1, int line_type = 8 );
```

*Матнлар:*

```
void cvPutText( CvArr* img, const char* text, CvPoint origin, const CvFont* font, CvScalar color );
```

## HighGUI модули функциялари

Ушбу бўлимда HighGUI модули функциялари билан танишамиз.

*Намойиш ойнасини яратиш:*

```
int cvNamedWindow(const char* name,int flags = CV_WINDOW_AUTOSIZE);
```

*Тасвирни юклаш:*

```
IpImage* cvLoadImage(const char* filename,int iscolor = CV_LOAD_IMAGE_COLOR);
```

*Тасвирни сақлаш:*

```
int cvSaveImage(const char* filename,const CvArr* image);
```

*Тасвирни намойиш қилиш:*

```
void cvShowImage( const char* name, const CvArr* image );
```

Сичқонча ходисалари (босилиши):

```
void CvMouseCallback(int event,int x,int y,int flags,void* param);
```

Бу ерда event га қуйидаги ходисалар боғланган:

Event (ходиса)	Рақам қиймати
CV_EVENT_MOUSEMOVE	0
CV_EVENT_LBUTTONDOWN	1
CV_EVENT_RBUTTONDOWN	2
CV_EVENT_MBUTTONDOWN	3
CV_EVENT_LBUTTONUP	4
CV_EVENT_RBUTTONUP	5
CV_EVENT_MBUTTONUP	6
CV_EVENT_LBUTTONDOWNCLK	7
CV_EVENT_RBUTTONDOWNCLK	8
CV_EVENT_MBUTTONDOWNCLK	9

## OpenCVда тасвирларни қайта ишлаш

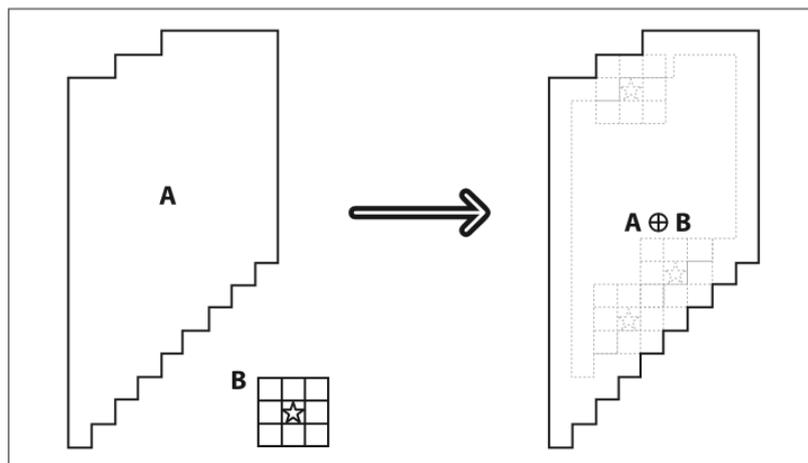
*Текислаш:*

```
void cvSmooth( const CvArr* src, CvArr* dst, int smoothtype = CV_GAUSSIAN, int param1 = 3, int param2 = 0, double param3 = 0, double param4 = 0 );
```

Бу ерда Smoothtype ўзгарувчиси қуйига текислаш турларини ўз ичига олиши мумкин: CV\_BLUR, CV\_BLUR\_NO\_SCALE, CV\_MEDIAN, CV\_GAUSSIAN, CV\_BILATERAL.

*Морфологик Кенгайтириш (Dilation):*

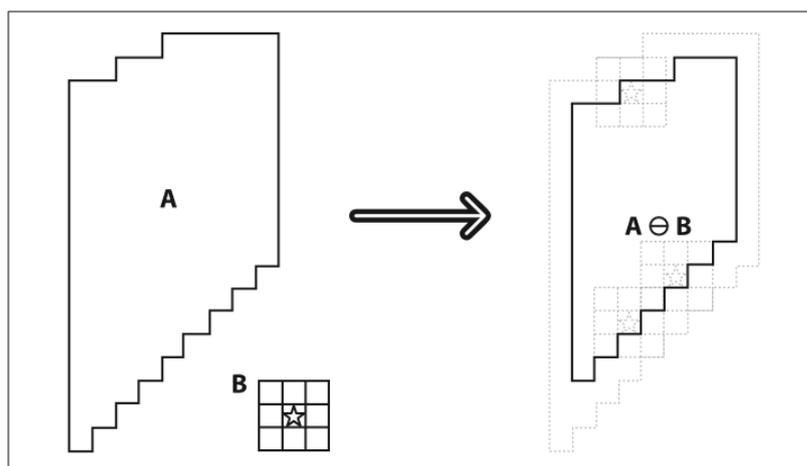
```
void cvErode( IplImage* src, IplImage* dst, IplConvKernel* B = NULL, int iterations = 1 );
```



3.5-расм. Морфологик кенгайтириш.

*Морфологик Емириш (Erosion):*

```
void cvDilate( IplImage* src, IplImage* dst, IplConvKernel* B = NULL, int iterations = 1 );
```



3.6-расм. Морфологик емириш.

*Оқимли тўлдириш (Flood Fill):*

```
void cvFloodFill(IplImage* img, CvPoint seedPoint, CvScalar newVal, CvScalar loDiff =  
cvScalarAll(0), CvScalar upDiff = cvScalarAll(0), CvConnectedComp* comp = NULL, int  
flags = 4, CvArr* mask = NULL );
```

*Ҳажми ўзгартириш (Resize):*

```
void cvResize( const CvArr* src, CvArr* dst, int interpolation = CV_INTER_LINEAR );
```

Бу ерда *Interpolation* ўзгарувчиси қуйидагиларни ўз ичига олиши мумкин: CV\_INTER\_NN, CV\_INTER\_LINEAR, CV\_INTER\_AREA, CV\_INTER\_CUBIC.

*Бўсағалаш (Threshold):*

```
double cvThreshold( CvArr* src, CvArr* dst, double threshold, double max_value,  
int threshold_type );
```

Бу функция глобал бўсаға қиймати орқали тасвирни бўлаклаш (масалан бинарлаштириш орқали) учун хизмат қилади. Ундаги *threshold\_type* ўзгарувчиси қуйидаги бўсағалаш турларинини ўз ичига олади:

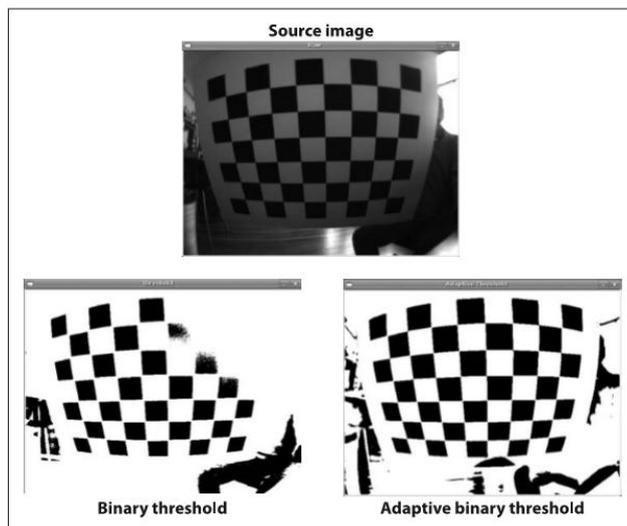
```
CV_THRESH_BINARY, CV_THRESH_BINARY_INV, CV_THRESH_TRUNC,  
CV_THRESH_TOZERO_INV, CV_THRESH_TOZERO.
```

*Адаптив бўсағалаш (Adaptive Threshold):*

```
void cvAdaptiveThreshold( CvArr* src, CvArr* dst, double max_val, int adaptive_method =  
CV_ADAPTIVE_THRESH_MEAN_C, int threshold_type = CV_THRESH_BINARY, int  
block_size = 3, double param1 = 5 );
```

Бу функцияда бўсағалаш блоklarда, яъни кичик соҳаларда амалга оширилади.

Қуйидаги расмда (3.7-расм) берилган тасвирни глобал ва адаптив бўсағалаш ёрдамида бинарлаштириш намуналари кўрсатилган.



3.7-расм. Берилган тасвир ҳамда унинг глобал ва адаптив бўсағалаш натижаси.

## OpenCV да тасвир ўзгартиришлар

*Ўрама (Свёртка):*

“Ўрама” усулида маълум бир ҳажмдаги (масалан,  $3 \times 3$  ўлчамли ядро) ойна ичида қийматлар тасвирга мосланади ва тасвирдаги мос координаталардаги қийматлар ойнадаги мос қийматлар билан кўпайтирилади ва уларнинг йиғиндиси йиғилади. Йиғинди натижа тасвирнинг ишчи ойна марказига қўйилади. Буни формула кўринишида қуйидагича ифодалаш мумкин.

$$H(x, y) = \sum_{i=0}^{M_i-1} \sum_{j=0}^{M_j-1} I(x+i-a_i, y+j-a_j) G(i, j)$$

OpenCVда бундай филтрлашни қуйидаги процедура ёрдамида амалга оширилади:

```
void cvFilter2D( const CvArr* src, CvArr* dst, const CvMat* kernel,
CvPoint anchor = cvPoint(-1,-1) );
```

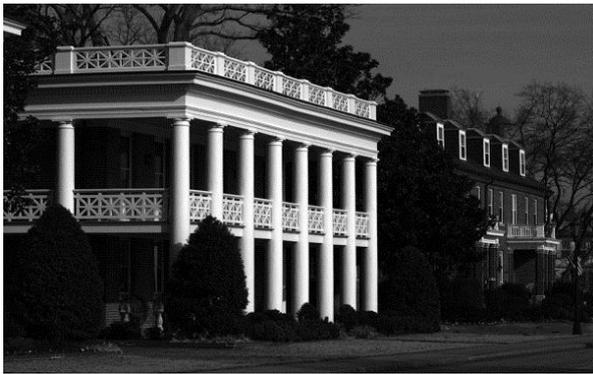
Бу ерда kernel (ядро) га олдиндан мавжуд ишчи ойналарни (масалан, Собель ишчи ядроси) қўйишимиз мумкин, ёхуд ўзимиз янги ядро қуриб олишимиз ҳам мумкин.

*Градиентлар ва Собель ҳосиласи:*

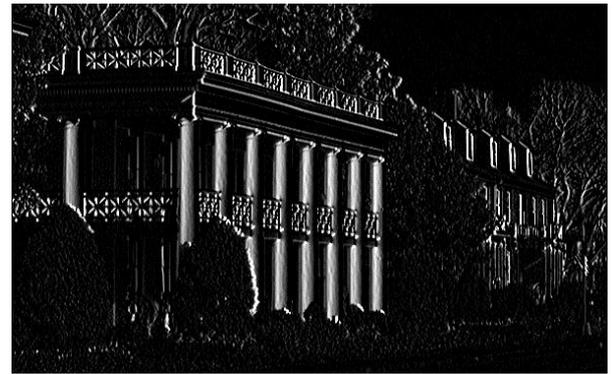
Тасвир градиентларини (чегараларини) аниқлашда энг кўп фойдаланиладиган усуллардан бири Собель ҳисобланади. Собель функцияси қуйида берилган:

```
cvSobel( const CvArr* src, CvArr* dst, int xorder, int yorder, int aperture_size = 3 );
```

Қуйидаги расмда (3.8-расм) берилган тасвир ва унинг Собел градиенти натижаси кўрсатилган.



a)



b)

3.8-расм. Берилган тасвир (a) ва унинг Собел градиенти натижаси (b)

*Лаплас (Laplace) градиенти:*

```
void cvLaplace(const CvArr* src,CvArr* dst,int apertureSize= 3);
```

*Канни (Canny) градиенти:*

```
void cvCanny( const CvArr* img, CvArr* edges, double lowThresh, double highThresh,  
int apertureSize= 3);
```

### **OpenCV да Хаф (Hough) алмаштиришлари**

*Хаф алмаштириши ёрдамида чизиқларни топиш:*

```
CvSeq* cvHoughLines2(CvArr* image,void* line_storage,int method,double rho,double  
theta,int threshold,double param1 = 0,double param2 = 0);
```

*Хаф алмаштириши ёрдамида айланаларни топиш:*

```
CvSeq* cvHoughCircles(CvArr* image, void* circle_storage,int method,double dp,double  
min_dist,double param1 = 100,double param2 = 300,int min_radius = 0,int max_radius =  
0);
```

### **OpenCV да Аффин алмаштиришлар**

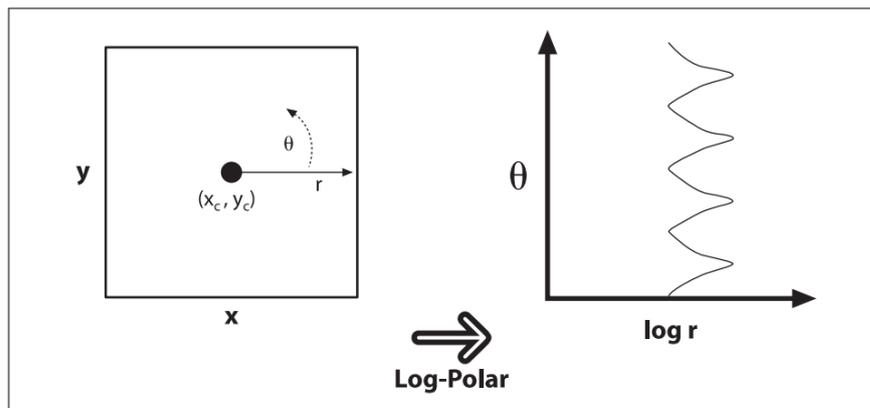
*Тасвирни буриш:*

```
CvMat* cv2DRotationMatrix( CvPoint2D32f center, double angle, double scale,  
CvMat* map_matrix );
```

*Поляр алмаштириш:*

```
void cvLogPolar(const CvArr* src, CvArr* dst, CvPoint2D32f center, double m,  
int flags = CV_INTER_LINEAR | CV_WARP_FILL_OUTLIERS );
```

Поляр алмаштириш деб, марказдаги нуқтага нисбатан қаралаётган нуқта(пиксел) нинг жойлашув координатаси бўйича бурчаги -  $\theta$  ва масофасини -  $r$  кўрсатувчи янги фазога айтилади (3.9-расм).



3.9-расм. Поляр алмаштириш схемаси.

*Дискрет-косинус алмаштириши:*

```
void cvDCT( const CvArr* src, CvArr* dst, int flags );
```

*Масофавий алмаштириши:*

```
Void cvDistTransform(const CvArr* src,CvArr* dst,int distance_type = CV_DIST_L2,  
int mask_size = 3,const float* kernel = NULL,CvArr* labels = NULL);
```

*Гистограммаларни тўғрилаш:*

```
void cvEqualizeHist( const CvArr* src, CvArr* dst );
```

Таъкидлаш керакки, OpenCV да яна кўплаб функциялар мавжуд, хусусан тасвирларни қайта ишлаш, тасвирнинг идентификацион белгиларини ажратиш олиш ва таниб олиш билан боғлиқ функциялар фойдаланувчиларга катта қулайлик туғдиради. Улар тўғрисида кенгроқ маълумотларни олиш учун OpenCV функциялари ҳақида ёритилган адабиётлардан фойдаланиш мумкин.

### 3.1.2. Qt Creator C++ дастурий воситаси имкониятлари

Qt Creator C++ дастурлаш воситаси ҳозирги кунда дастурчиларнинг энг самарадор воситасига айланиб улгурган. Ахборот-коммуникацион технологиялар билан шуғулланувчи дунёнинг бир қатор йирик компаниялари

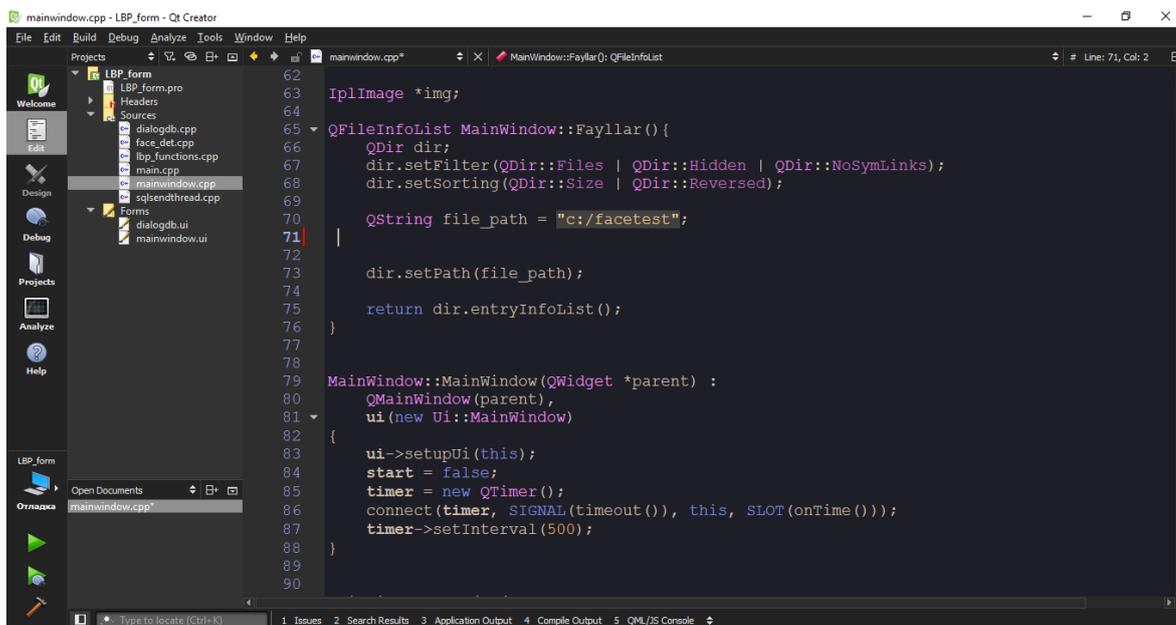
ўз маҳсулотларини Qt C++ дастурлаш воситасида яратмоқдалар. Бундай компанияларга мисол сифатида Adobe, Amazon, AMD, Boshc, Blackberry, Cannon, Cisco Systems, Disney, Intel, IBM, Panasonic, Pioneer, Philips, Oracle, HP, Goober, Google, NASA, NEC, Neonway, Nokia, Samsung, Siemens, Sony, Xerox, Xilinx, Yamahalarни келтириш мумкин.

Шу билан бирга ҳозирги кунда маълум ва машҳур бўлган бир қатор компьютер дастурлари ҳам айнан Qt C++ библиотекалари ёрдамида яратилган. Хусусан: “KDE Software Compilation 4” иш столи, “Autodesk Maya” уч ўлчовли графика таҳрирлагичи, “Skype” интернет – пейджер, “Adobe Photoshop Album” рақамли тасвирларга ишлов бериш дастури, “Google Earth” тармоқли дунё харитаси, “VLC media player” ва ҳ.к.

Айниқса, Qt C++ нинг мультимедиавий дастурлаш имкониятлари юқори. Хусусан, рақамли тасвирларни қайта ишлаш учун кенг имкониятлар яратилган.

Шунингдек, Qt C++ дастурлаш воситасининг яна бир ижобий томони шундаки, у кроссплатформали ҳисобланади, яъни у ёрдамида яратилган дастуримиз деярли барча тизимларда (Windows, Mac OS X, Linux, UNIX, iOS, Android ва ҳ.к.) ишлай олади.

Qt Creator дастурлаш воситасининг умумий кўриниши қуйида келтирилган (3.10-расм).



3.10-расм. Qt Creator дастурлаш воситасининг умумий кўриниши.

Qt Creator дастурлаш воситасининг бир қатор стандарт визуал объектлари (*Виджетлари*) орқали тасвирларни қайта ишлаш мумкин. Шундай бўлсада, рақамли тасвирларни қайта ишлашга мўлжалланган OpenCV очик библиотеказини Qt Creator га боғлаш орқали янада юқорироқ самарадорликка эришиш мумкин. Qt Creator га OpenCV ни боғлашни қуйида кўрамиз.

Бизда OpenCV нинг библиотекази мавжуд ва уни бирор дискка, масалан “d” дискка Opencv папкаси ичига сақлаган бўлсак, у ҳолда Qt Creatorда яратган проектимиз (дастуримиз)га тегишли *.pro* кенгайтмали файл ичига қуйидагича буйруқлар кетма-кетлигини ёзамиз:

```
INCLUDEPATH += d:\opencv\include
LIBS += -L"d:/OpenCV/bin"
LIBS += -lopencv_core2410 -lopencv_highgui2410 -lopencv_imgproc2410 -
lopencv_features2d2410 -lopencv_objdetect2410
```

Бунда сўнги қаторда биз фойдаланадиган библиотеказалар рўйхати келтирилган. Сиз бу жойга ўзингизга керак бўладиган библиотеказани қўйишингиз мумкин.

Дастур кодида эса, OpenCV библиотеказарини қуйидагича боғлаймиз:

```
#include <OpenCV/cv.h>
#include "opencv2/core/core.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
using namespace cv;
```

Шу билан OpenCV библиотеказари имкониятларидан Qt Creator да бемалол фойдаланишимиз мумкин бўлади.

### 3.1.3. Ишда фойдаланилган техник воситалар таҳлили

Масалани ҳал қилишда фойдаланилган техник таъминотларга қуйидагиларни киритиш мумкин.

- Компьютер.
- Маълумот ташувчи дисклар.
- Фотоаппарат.

Юқорида келтирилган техник қурилмаларнинг қисқача вазифаларини кўриб чиқайлик.

#### 1) Компьютер.

Асосий восита ҳисобланиб, унда процессор, доимий ва тезкор хотира, монитор, клавиатура, сичқонча каби асосий қурилмалар мавжуд бўлади. Бунда дастурни яратиш ва намойиш этиш ишлари бажарилади. Компьютернинг техник параметрлари: процессор - Dual Core 3.0 ГГц, тезкор хотира - 2 ГБ.

#### 2) Маълумот ташувчи дисклар.

Ахборотларни бир жойдан иккинчи жойга кўчириб ўтиш учун CD ёки Flash дисклар зарур бўлади.

#### 3) Фотоаппарат.

Рақамли фотоаппаратлар ёрдамида олинган тасвирларни компьютер хотирасига олиш мумкин (3.11-расм). Тасвирларнинг сифати мегапикселларнинг катта ёки кичиклигига боғлиқ. Шу ўринда, “Мегапиксел” (Мп) тушунчасини изоҳлаб ўтсак. Мега – бу миллион деган маънони беради. Масалан, 1 Мегапиксел ўлчовимиз 1 млн. та пиксел дегани. Мисол учун тасвиримизнинг ўлчами 1920×1200 ҳажмда бўлса, у ҳолда бу тасвирнинг пикселлардаги ҳажми 2,3 Мп бўлади. Яъни, бунда тасвирнинг эни 1920 та пиксел (ёки нуқта), бўйи эса 1200 та пиксел, жами пикселлар сони  $1920 \times 1200 = 2304000$  та пикселга тенг бўлади ва ҳ.к. Турли рақамли тасвирлар билан боғлиқ қурилмалар, хусусан, фотоаппаратларнинг асосий техник хусусиятлари уларнинг битта кадрга (ёки матрицага) қанча пикселларни сиғдира олиши билан баҳоланади. Яъни, битта кадрда қанчалик кўп

пикселларни ўзида қамраб олса, ана ўша фотоаппаратнинг тасвирлар сифати юқори бўлади.



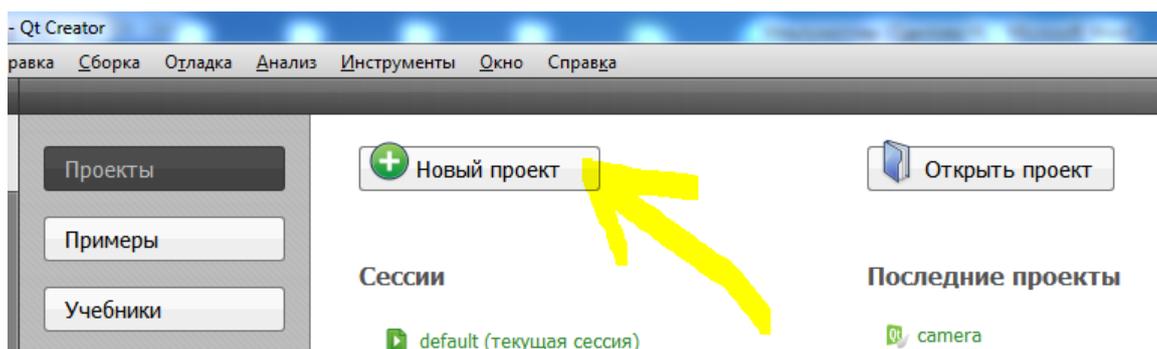
3.11-расм. Фотоаппарат ёрдамида рақамли тасвир ҳосил қилиш.

### **3.2. Алгоритмлар асосида амалий дастурлаш яратиш технологияси**

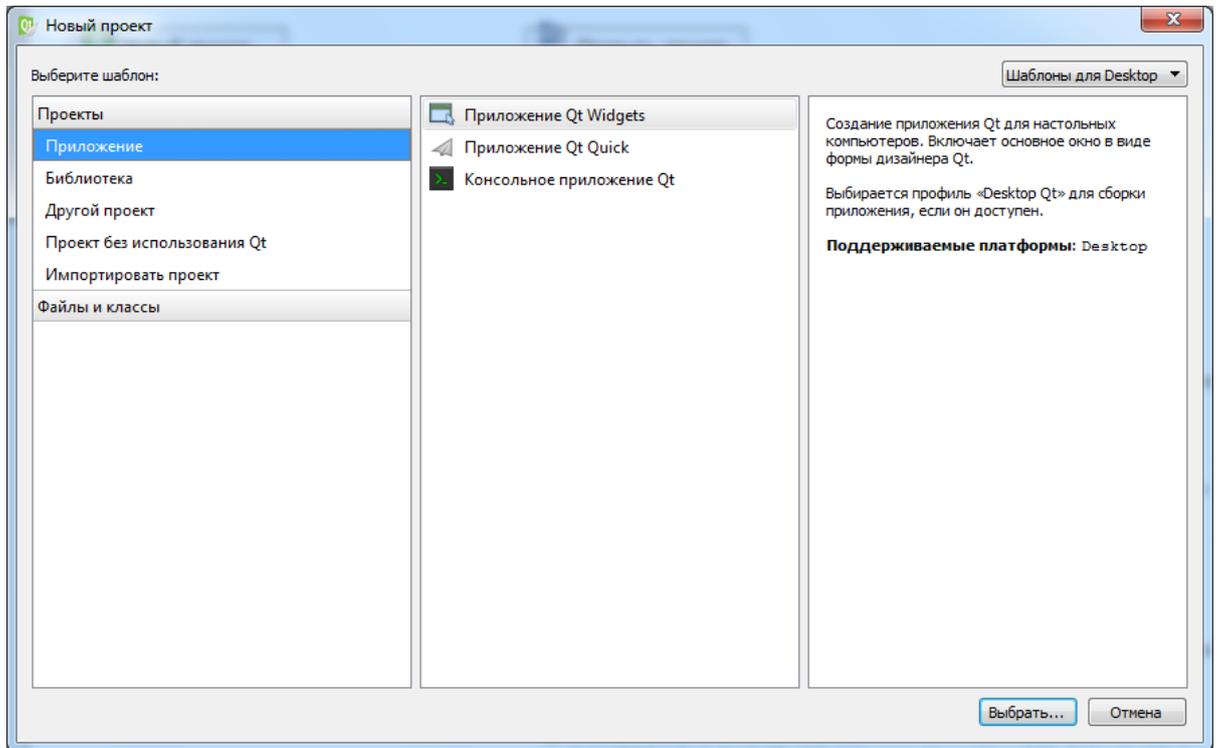
Дастур OpenCV библиотекаси ва Qt Creator C++ дастурий воситалари ёрдамида ишлаб чиқилди. Биз дастлаб Qt Creator да янги лойиҳа яратиб оламиз. Бунинг учун Qt Creator асосий лойиҳа ойнасида “Янги лойиҳа” (Новый проект) тугмасини босамиз (3.12-расм). Натижада лойиҳа турини танлаш учун диалогли ойна ҳосил бўлади (3.13-расм). У ердан “Приложение” ва “Приложение Qt Widgets” лойиҳа турини танлаймиз ҳамда “Выбрать...” тугмасини босамиз. Шундай сўнг лойиҳа номи ва унинг жойлашувини танлаш диалог ойнаси ҳосил бўлади (3.14-расм). Бу ойнада керакли ўзгартиришларни амалга оширганимиздан сўнг “Далее” тугмасини босамиз. Натижада “Комплектни танлаш” диалог ойнаси ҳосил бўлади (3.15-расм). Бу ойнадаги параметрларни ўзгаришсиз қолдириб, яна “Далее” тугмасини босамиз. Шундан сўнг синфлар ҳақида маълумот берувчи диалог ойна ҳосил бўлади (3.16-расм). Бу ойнада синфларга стандарт ном берилган бўлиб, агар хоҳласак, уларнинг номларини ўзгартиришимиз мумкин. Кейин бу ойнада ҳам “Далее” тугмасини босамиз. Натижада “Лойиҳани бошқариш” деб номланган сўнгги диалог ойнаси ҳосил бўлади (3.17-расм). Бу ойнада “Настроить...” номли тугмасини босиш орқали бир қатор параметрларни созлаш мумкин. Лекин, бу бизга шарт эмас, ҳеч нарсани ўзгартирмасдан “Завершить” тугмасини босамиз. Натижада янги

лойиханинг дастур тахрирлаш ойнаси пайдо бўлади (3.18-расм). Ойнанинг чап қисмида “Формы” бўлимидан `mainwindow.ui` ни танлаб, сичқончанинг чап тугмасини икки марта босиш орқали лойиханинг формали кўринишини чиқарамиз (3.19-расм). Бу ойна асосий визуал ишчи ойнаси ҳисобланади. У ерда ойнанинг чап томонида турли вазифаларга мосланган “Объектлар” (компонентлар) жойлашган. Улардан бирини сичқонча ёрдамида танлаб, марказдаги формага жойлаш мумкин. Жойланган фаол объектнинг хусусиятлари ўнг томондаги “Хусусиятлар” ойнаси параметрларини созлаш орқали шакллантирилади.

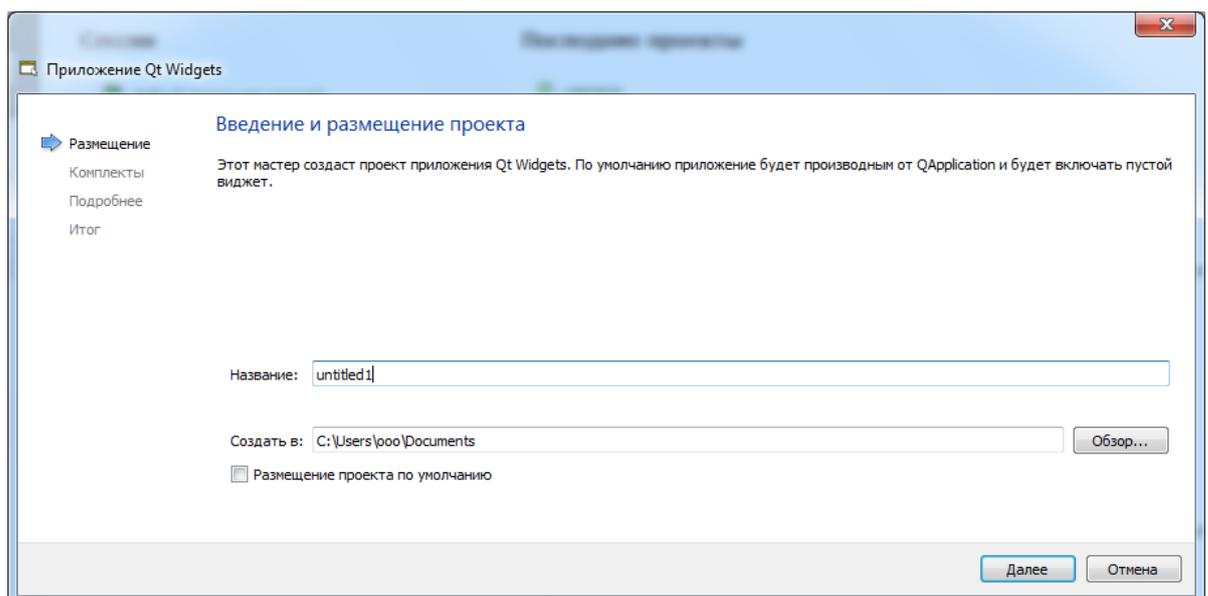
Объектлар билан ишлаш “Сигнал-слот” технологиясида амалга оширилади. Масалан, босилувчи тугма “`QPushButton`” объектини формага жойлаб, унинг устида сичқончанинг ўнг тугмаси босиб, “Перейти к слоту” меню буйруғини танлаб босамиз (3.20-расм). Натижада турли сигналлар вариантлари учун диалог ойнаси ҳосил бўлади (3.21-расм). У ерда керакли слотни танлаймиз. Масалан, “`clicked()`”, яъни “босилди” слотини танлаб “Ок” тугмасини босамиз. Натижада “`QPushButton`” босилганда рўй бериши лозим бўлган жараён учун дастур ёзишга тахрир ойнаси очилади (3.22-расм). Шу тариқа объектлар учун зарурий буйруқлар дастур коди ойнасида ёзилади.



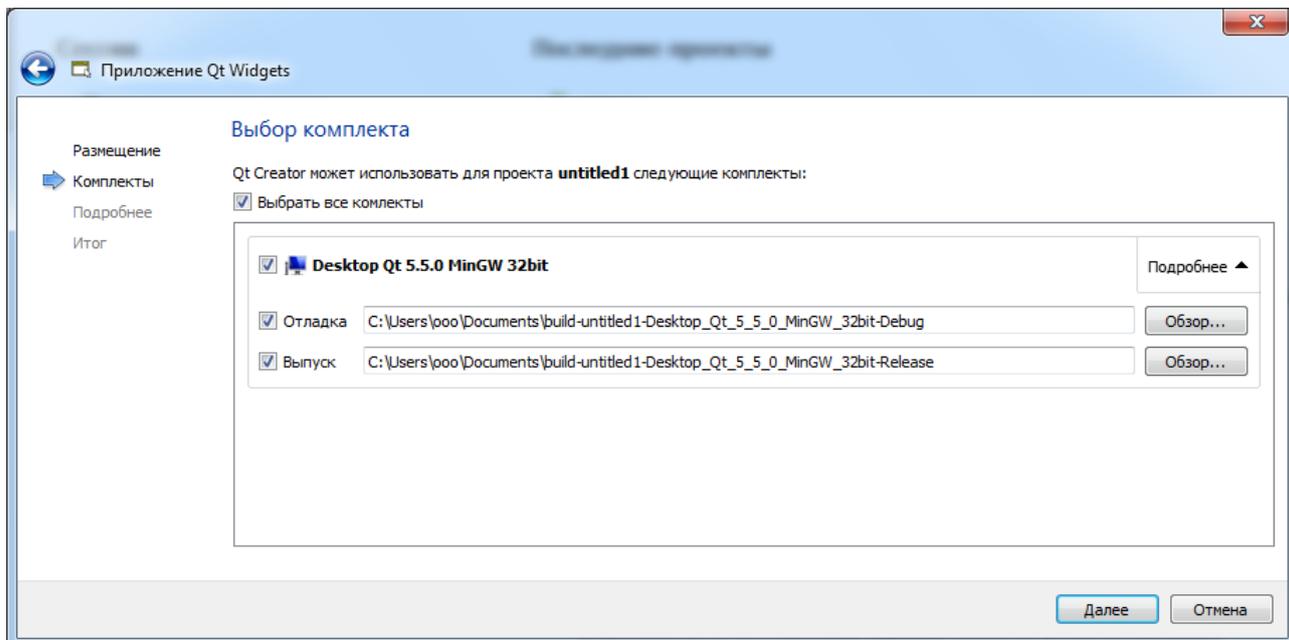
3.12-расм.



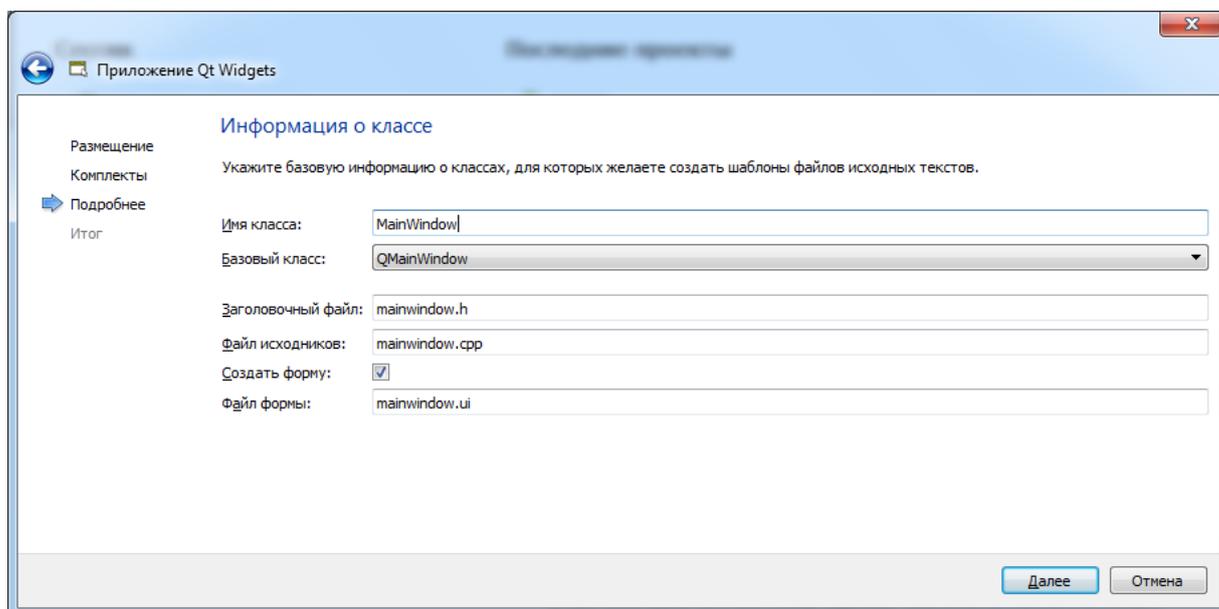
3.13-рasm.



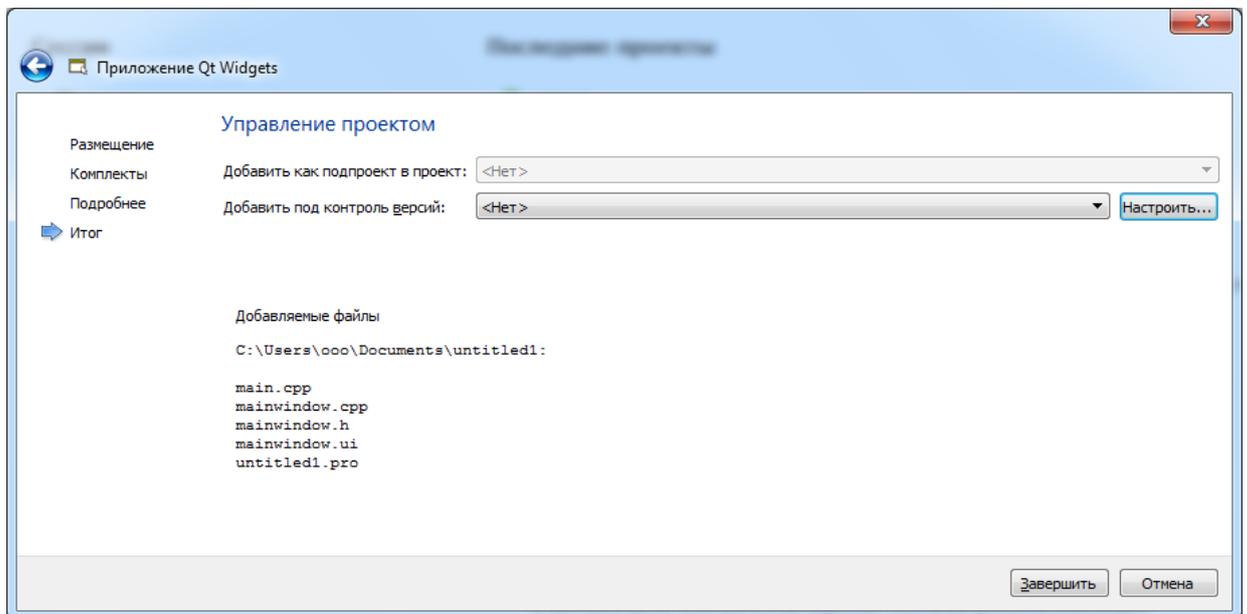
3.14-рasm.



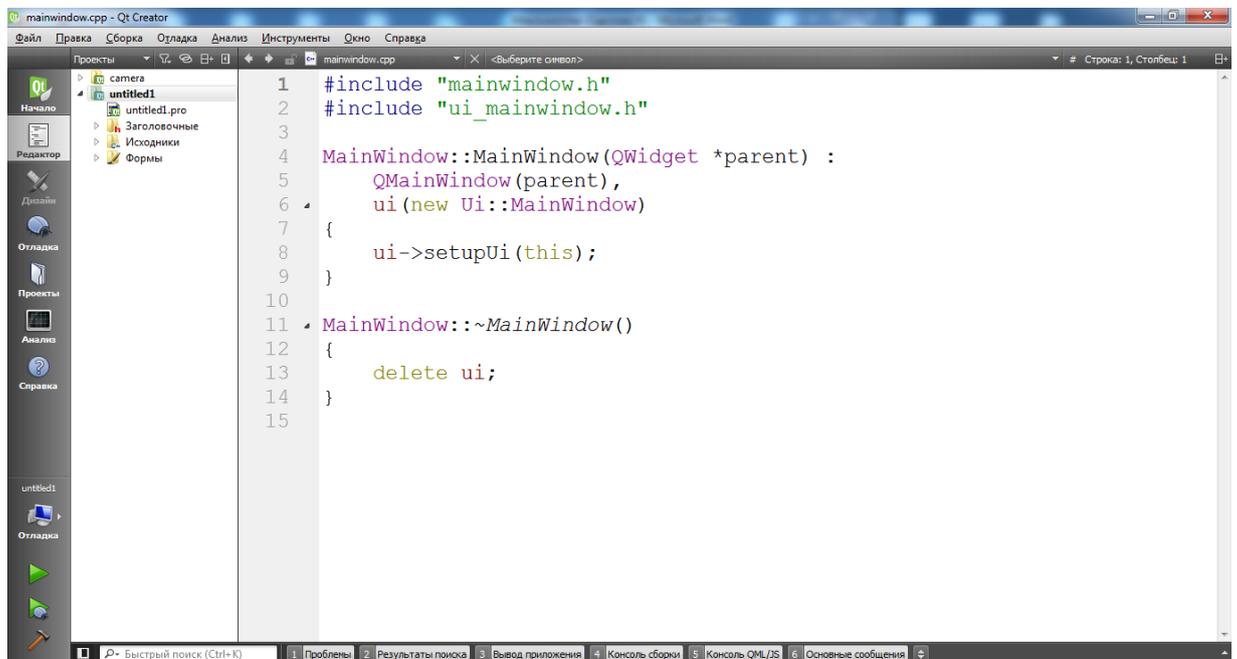
3.15-рasm.



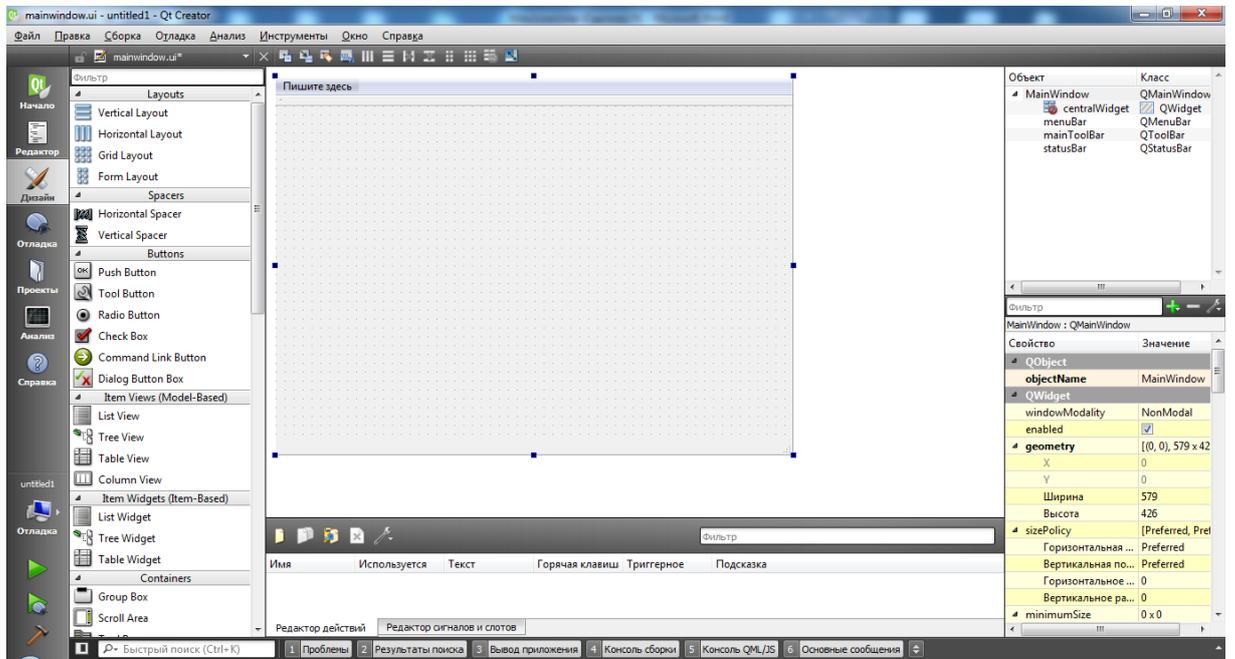
3.16-рasm.



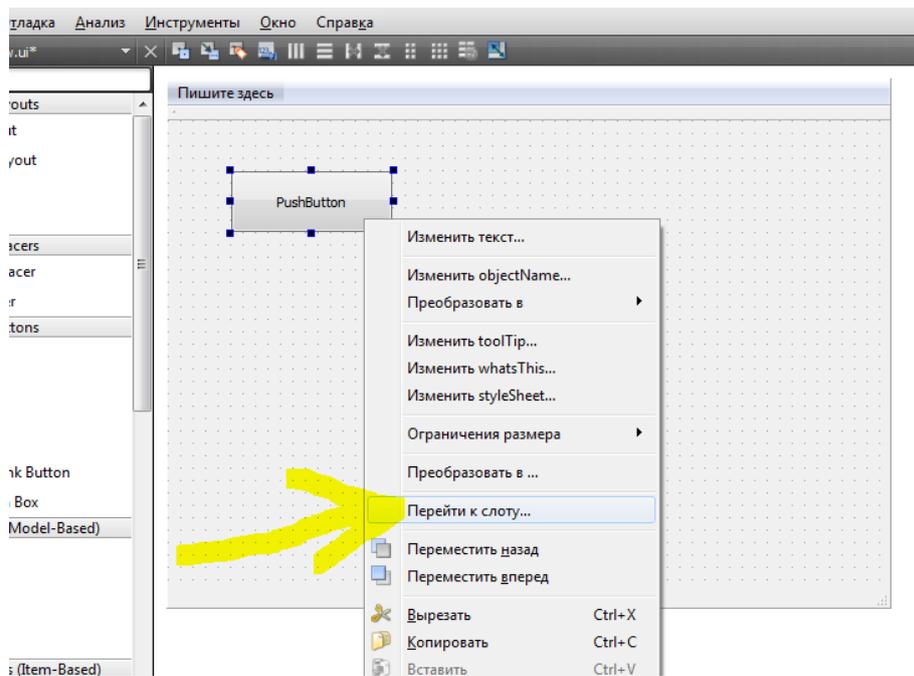
3.17-рasm.



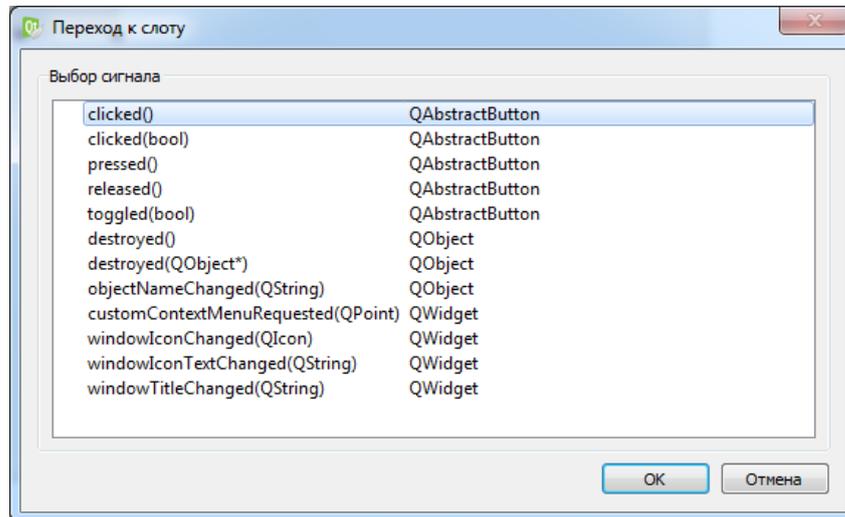
3.18-рasm.



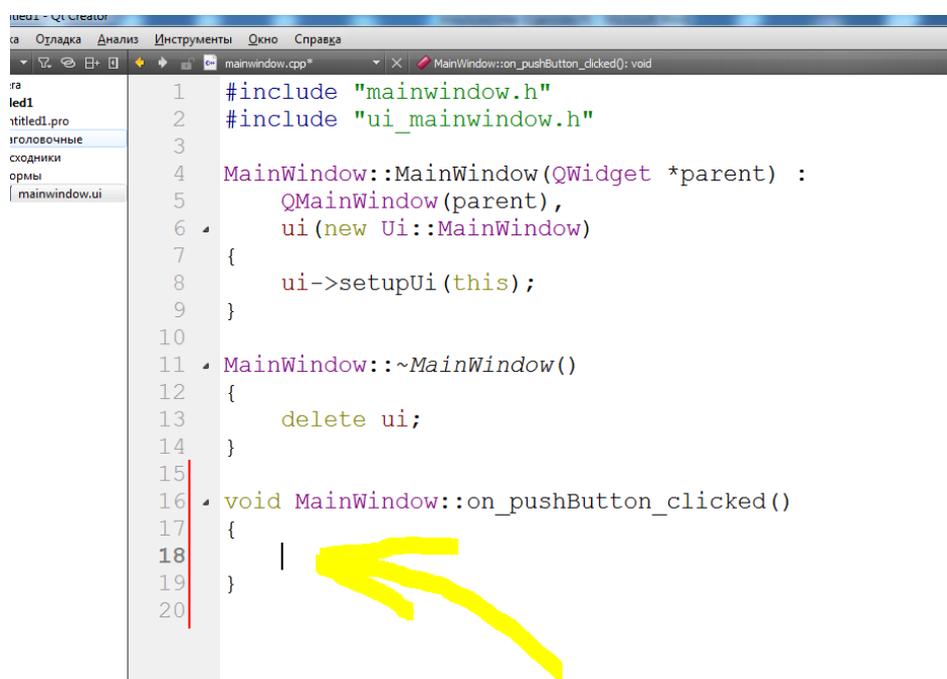
3.19-рasm.



3.20-рasm.



3.21-расм.



3.22-расм.

Энди, бизда OpenCV библиотекасини Qt Creatorга боғлаш масаласи бор. Бунинг учун биз лойиханинг “pro” файли (бизнинг мисолда “untitled1.pro” файли) ичига куйидаги дастур кодини кўшамиз.

```

INCLUDEPATH += d:\OpenCV_24\include
LIBS += -L"d:/OpenCV_24/bin"
LIBS += -lopencv_core2410 -lopencv_highgui2410 -lopencv_imgproc2410 -
lopencv_features2d2410 -lopencv_objdetect2410

```

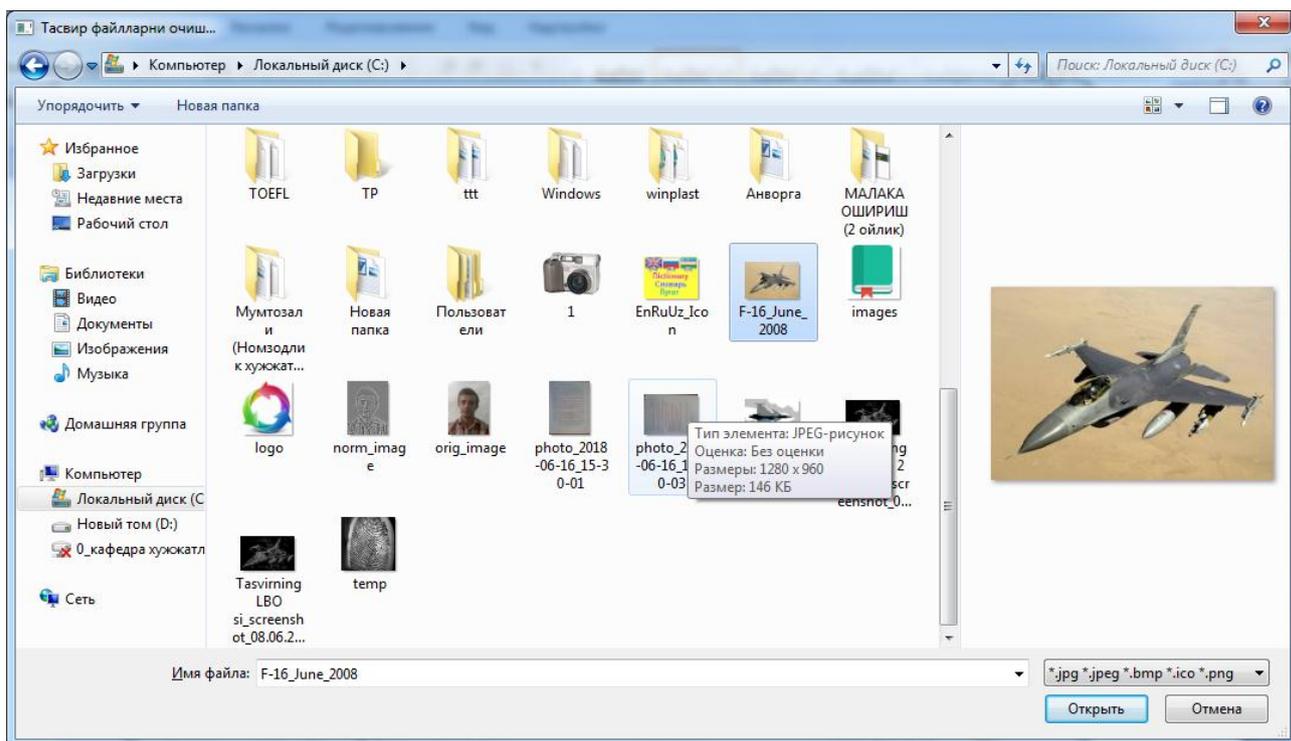
Бунда кўнишимиз мумкинки, OpenCV библиотекасининг тегишли файллари d:/OpenCV\_24/ папкасида жойлашган. Биз бу папкани ихтиёрий жойда сақлашимиз мумкин. Фақат, унга йўлни кўрсатишда адашмасак бўлгани.

Амалий дастур Qt Creator C++ дастурлаш воситаси ёрдамида яратилди.

Унда асосан куйидаги объект (виджет)лардан фойдаланилди:

- 1) **Action** – юқори менюни ташкил қилиш учун фойдаланилади.
- 2) **GroupBox** – чегарали сохани ажратиш учун хизмат қиладиган ишчи майдон.
- 3) **gridLayout** – маълум бир объектларни гурухлаб олиш имкониятини беради.
- 4) **PushButton** – босилувчи тугма, бирор амални боғлаш учун керак.
- 5) **Line** – чегарани ажратувчи чизиқ.
- 6) **ListWidget** – рўйхатни визуал намойиш қилиш ва бошқариш объекти.

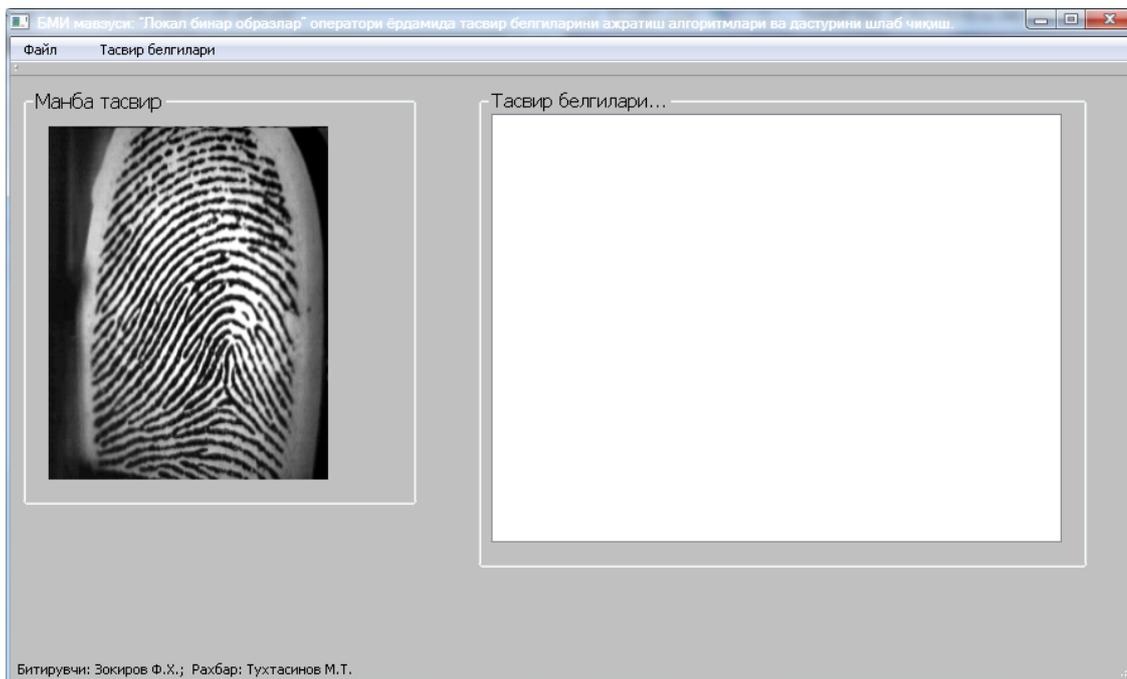
Тасвир файлларни очиш учун QFileDialog модулидан фойдаланилди (3.23-расм).



3.23-расм.

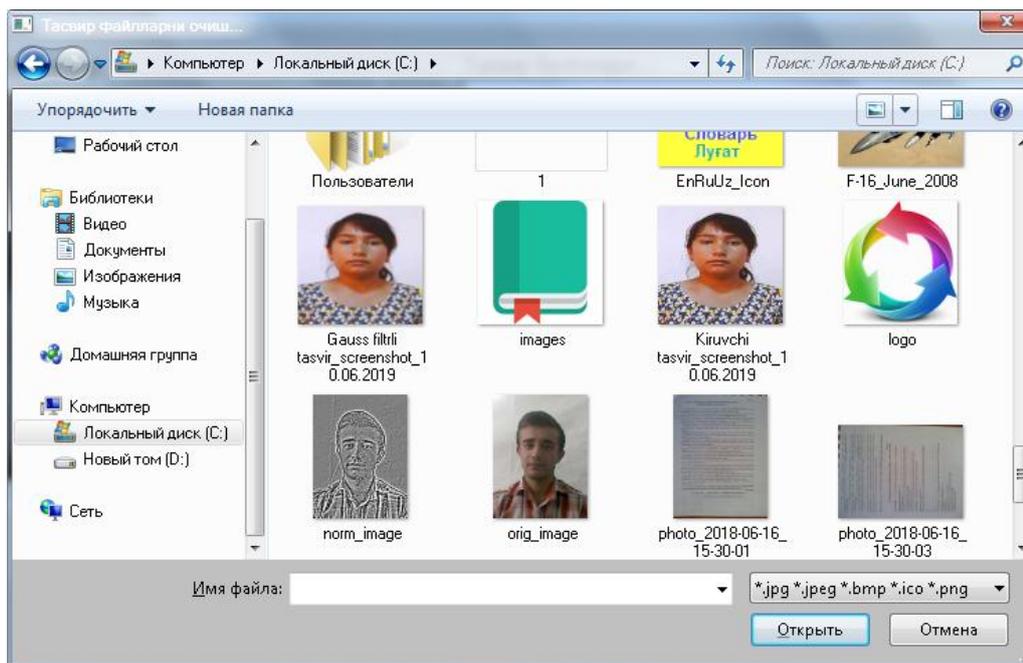
### 3.3. “Local Binary Pattern” дастуридан фойдаланиш бўйича йўриқнома

Дастур “Local Binary Pattern.exe” деб номланади. Уни ишга туширганда куйидагича ойна ҳосил бўлади (3.24-расм.)



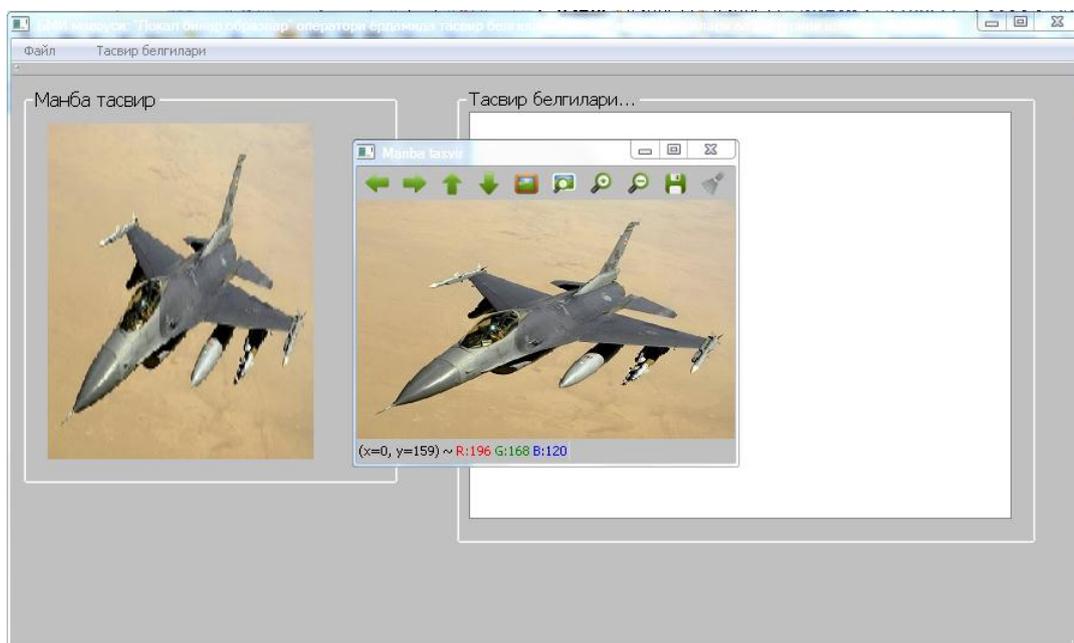
3.24-расм. Бош ойна.

Дастур ойнасини асосан 3 та қисмга ажратиш мумкин: 1-юқори меню буйруқлари, 2-манба тасвир соҳаси, 3-тасвир белгилари соҳаси. Файл менюсида “Очиш” ва “Чиқиш” меню буйруқлари мавжуд. “Очиш” меню буйруғи ёрдамида хотирадан тасвирни очиш, яъни юклаш мумкин. Тасвирни очиш учун “Очиш” меню буйруғи босилганда махсус диалогли ойна ҳосил бўлади (3.25-расм).



3.25-расм. “Тасвир файлларни очиш” диалогли ойнаси.

Керакли тасвир танлангандан сўнг у иккита ойнада пайдо бўлади. Биринчи тасвир бош ойнанинг “Манба тасвир” соҳасида, иккинчи тасвир эса OpenCV нинг силжувчан ойнасида кўринади (3.26-тасвир).



3.26-расм.

Танланган тасвирнинг идентификацион белгиларини “Локал бинар образлар” (ЛБО) операторлари ёрдамида аниқлаш учун юқори менюнинг “Тасвир белгилари” меню буйруғига муурожаат қиламиз. У ерда ЛБОнинг турли вариантлари мавжуд (3.27-расм).

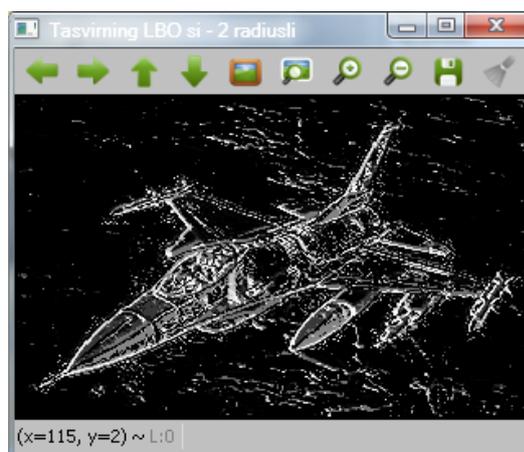


3.27-расм.

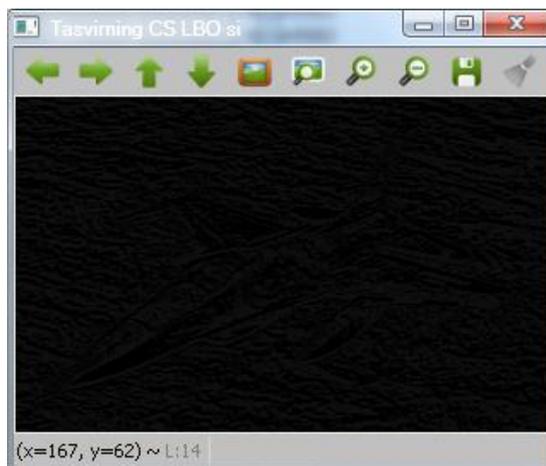
3.27-расмда кўриниб турибдики, у ерда ЛБО оиласига мансуб беш хил усул мавжуд. Танланган тасвир (3.26-расмдаги самолёт тасвири) учун шу усуллардаги натижаларни куйида кўрамиз (3.28-3.32-расмлар).



3.28-расм. Тасвирнинг ЛБО асосидаги белгилари.



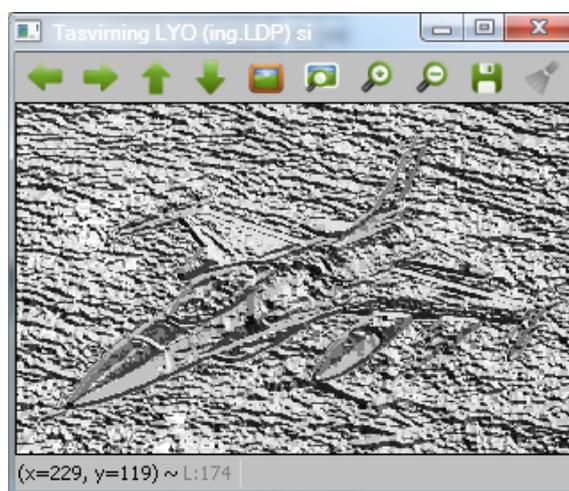
3.29-расм. Тасвирнинг ЛБО (2-радиусли) асосидаги белгилари.



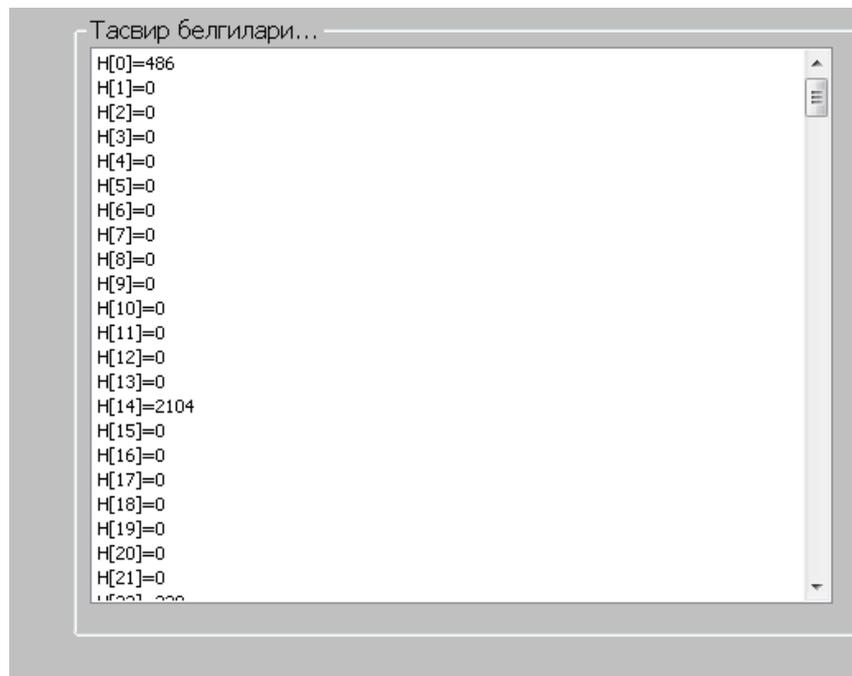
3.30-расм. Тасвирнинг ЛБО (марказий симметрик) асосидаги белгилари.



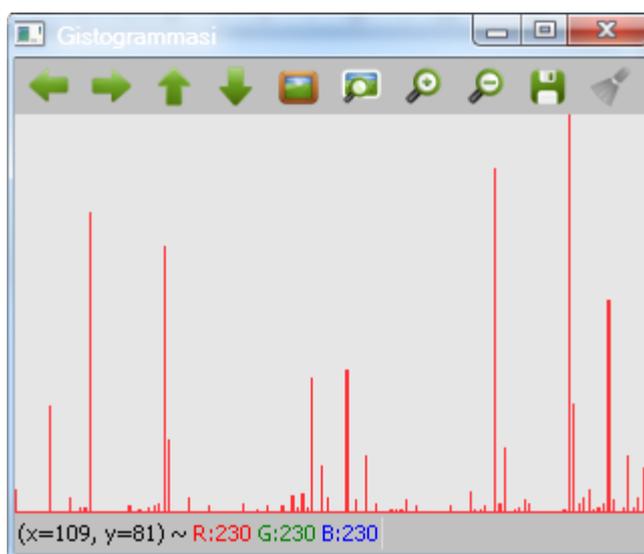
3.31-расм. Тасвирнинг ЛБО (марказий симметрик, 2 радиусли) асосидаги белгилари.



3.32-расм. Тасвирнинг ЛЙО (локал йўналган образлар) асосидаги белгилари. Ҳар бир аниқланган ЛБО белгилар матрицаси, яъни тасвирини келгусида уларни таққослаш йўли билан таниб олиш масаласи ҳал этилиши мумкин. Таққослашда белгилар матрицаларининг гистограммаларидан фойдаланилади. Яъни, бевосита матрица қийматлари эмас, балки гистограмма қийматлари таққосланади. Дастурда натижа (белгили) тасвирнинг гистограммаси қийматли (3.33-расм) ҳамда график (3.34-расм) кўринишда курилади.

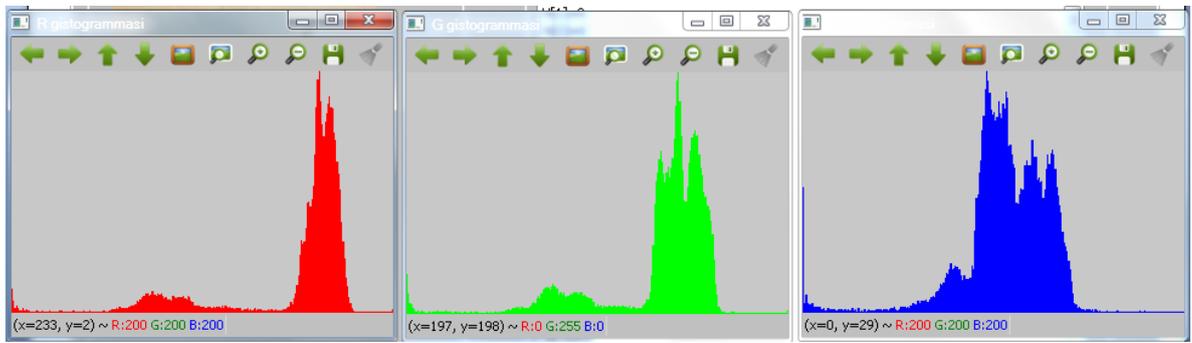


3.33-расм. Гистограмма қийматлари.



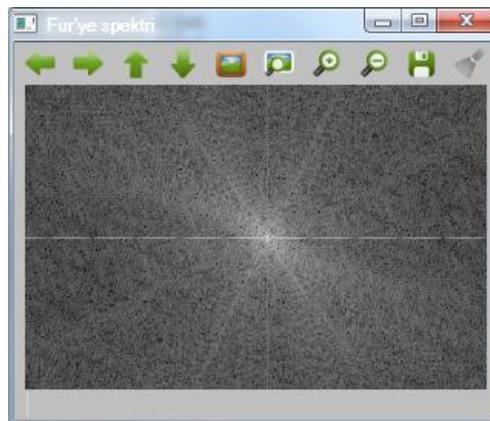
3.34-расм. Гистограмма графиги.

Шунингдек, белгилар сифатида ЛБОдан ташқари бир неча белгиларни аниқлаш дастуримизда мавжуд. Масалан, очилган рангли тасвирнинг RGB ранг шкалалари бўйича гистограммалар графиги (3.35-расм).



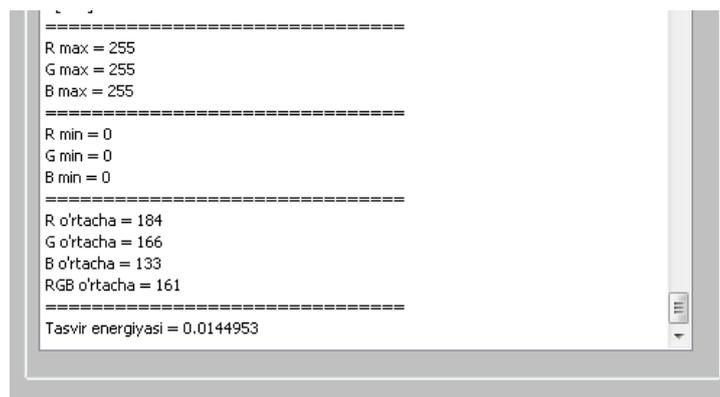
3.35-расм. Очилган рангли тасвир (3.26-расмдаги самолёт тасвири)нинг RGB гистограммалари.

Шунингдек, тасвирнинг Фурье спектрини ҳам аниқлаш мумкин (3.36-расм.)



3.36-расм. Очилган рангли тасвир (3.26-расмдаги самолёт тасвири)нинг Фурье спектри.

Ва ниҳоят, очилган тасвир ранглари қийматлари бўйича максимал, минимал, ўртача ва энергия каби белгиларни аниқлаш мумкин (3.37-расм).



3.37-расм.

## **IV-боб. Ҳаёт фаолияти хавфсизлиги**

### **4.1. Мухандис-педагоглар фаолиятини самарали ташкил этишда меҳнат муҳофазаси масалалари**

Мухандис-педагоглар фаолиятини самарали ташкил этишда меҳнат муҳофазаси масалалари ва уларнинг ижобий ечими муҳим аҳамият касб этади. Меҳнат шароитларини яхшилаш ва хавфсизликни таъминлаш мақсадида хавфсизлик техникаси хизмати кўзда тутилган. Хавфсизлик техникаси хизматининг асосий вазифалари ташкилот ва корхоналарда ишлаб чиқариш травматизми сабабларини бартараф этиш ишларини бажариш, меҳнат шароитларини яхшилаш, хавфсизлик техникаси ва химоя воситаларини такомиллаштириш, ишлаб чиқариш маданиятини кўтариш, ташкилий-техник ва санитария-гигиена тадбирларини ишлаб чиқиш ва амалга ошириш бўйича ишлаб чиқариш ва техник хизматларнинг ишини назорат қилишдан иборат. Ташкилотларда ва қишлоқ хужалик корхоналарида меҳнат муҳофазаси, хавфсизлик техникаси ва ёнғиндан муҳофаза қилишни ташкил қилиш бўйича инженерлик лавозими мавжуд бўлиб, бу инженернинг вазифасига корхона ишчиларига хавфсизлик техникаси ва ишлаб чиқариш санитариясидан таълим беришни ташкил қилишда иштирок этиш ва уларнинг билимларини текшириш киради. У жамоанинг касаба уюшмаси билан биргаликда меҳнат шароитини яхшилашга оид тадбирлар ишлаб чиқади ва уларнинг бажарилиш чорасини кўради. Меҳнат муҳофазаси бўйича инженер ишлаб чиқариш билан боғлиқ бўлган бахциз ходисалар юз берган вазиятни ва унинг сабабларини текширишда иштирок этади, травматизм тўғрисида ҳисоботлар тузади, унинг сабабларини ўрганади ва уларни бартараф қилиш чораларини ишлаб чиқади.

Таълим тизимида меҳнат муҳофазасининг умумий ҳолати, уқув муассасаларининг раҳбарларига юклатилади. Улар куйидагиларни ташкил этиши кўзда тутилади:

- бутун ўқув процессида ўқувчилар, талабалар ва ишчи ходимларнинг хавфсизлигини таъминлаш;

- мактаб, лицейда, коллеж ва ОТМларда хавфсизлик техникаси бўйича плакатлар, инструкциялар ва бошқа норматив ҳужжатлар билан таъминлаш;

- ишлаб чиқаришда ва ўқув процессида содир бўлган бахциз ходисаларни текширишни ташкил қилиш ва унга сабаб бўлган камчиликларни тузатиш.

- ҳар қандай оғир ёки грuxли бахциз ходисаларни тезда вилоят халқ таълими бошқармасига, прокуратурага ва вилоят касаба уюшмасига хабар етказишни ташкил қилиш керак;

- касаба уюшма билан биргаликда ҳар чоракда меҳнат муҳофазаси бўйича тадбирларнинг бажарилишини текшириб боради;

- меҳнат муҳофазаси ҳолати бўйича жавоб берувчи шахснинг ҳисоботи тингланади ва содир бўлган камчиликлар тузатилиб борилади;

- доимо ўқув хоналарнинг ҳаво иклими, чангланганлик ва захарли газларнинг бор-юклиги текширилишини шунингдек дезинфекция қилишни ташкил қилади;

- ҳар йили электроустановакаларнинг ерга уланишини, изоляциясини текширтириб боради;

- ўз вақтида ўқув хоналарнинг ремонтини ташкил қилади (потолок, поллар ва хоказолар).

Ҳаёт фаолияти хавфсизлиги ва меҳнат муҳофазаси масалалари ҳамма ишчи-ҳодимларнинг, талабаларнинг ва ўқувчиларнинг амал қилишлари шарт бўлган умумий қоидалар мажмуидир. Шунинг учун ишга янги қабул қилинган барча ишчилар иш ўринларида кириш инструктажи ва бирламчи инструктаждан ўтганларидан кейингина ишга қўйилишлари мумкин.

Бундан ташқари даврий инструктаж, пландан ташқари инструктаж, кушимча инструктаж ва ойлик ўқиш курслардан иборат.

**Кириш инструктажини** тегишли соҳанинг бош мутахассиси утказди. У меҳнат муҳофазасига оид қонунчилик тўғрисидаги асосий маълумотларни, ички тартиб-қоидалар ва корхонадаги хавфсизлик қоидалари тўғрисида гапириб беради. Кириш инструктажи суҳбат ёки лекция тарзида ўтилади. Кириш инструктажини ўтишда кўпроқ техника асбоблари, кўрғазмали қуроллар ёрдамида ўтиш керак. Инструктаж ўтгандан кейин албатта журнал тутиш керак

ва бу журналга ўқувчилар ва талабалар инструктаж ўтганлиги хақида кўл кўйиши керак.

**Бирламчи инструктаж**ни ўқитувчи ёки участка бошлиги хар бир талаба билан ёки ишчи билан олиб боради. Бирламчи инструктажга ишчининг ўз ихтисослиги бўйича вазифалари, иш ўрнини хавфсиз ташкил қилиш тартиби билан таништириш, асбоб-ускуна ва химоя воситаларини ишлатиш ҳамда улардан фойдаланиш қоидалари, хавфсиз ишлаш усуллари, ёнғинни ўчириш воситалари ва сигнализациянинг турар жойи ҳамда ишлатиш усуллари, шунингдек, хавф туғилган тақдирда ўзини қандай тутиш кераклиги билан таништирилади. Инструктаж ўтувчи ишчининг ёки талабанинг хавфсизлик қоидалари бўйича билимини текшириб кўради, агар у меҳнат муҳофазасининг асосий талабларини, қоидаларини ўрганган бўлса уни ишлашга рухсат беради ва журналга қайд қилиб кўяди.

**Даврий инструктаж** - ишчи-ходимларга ва талабаларга камида ярим йилда бир марта иш жойида ўтказилади. Унинг мақсади - ишчиларнинг хотирасида хавфсизлик техникаси ва ёнғинга қарши курашиш қоидаларидан билимларини мустахкамлаш учун, ҳамда электроустановкаларга хизмат кўрсатувчи ходимлар учун эса камида 3 ойда бир марта бевосита иш ўринларида ўтказилади.

**Режадан ташқари инструктаж** - хавфсизлик қоидалари ёки асбоб-ускуна ва хавфсиз ишлашга таъсир қилувчи бошқа хар қандай факторлар ўзгарганида, шунингдек, бирор ходим томонидан травма, авария ёки ёнғин чиқишига олиб келадиган даражада хавфсизлик қоидалари бузилганда ўтказилади.

**Кўшимча инструктаж** эса бахциз ходисадан кейин, сўнгги инструктаж берилган санадан катъи назар ҳамма ишловчилар билан ўтказилади.

Ойлик ўқиш курсларида кучланиши 1000 В ва ундан юқори бўлган электроустановкаларга хизмат кўрсатувчи ходимлар, баландликда ишловчи монтажчилар, изоляцияловчилар, электр-газ пайвандчилар ва юқори хавф манбаида ишлайдиган ишчилар ўқийди. Улар ташкилот бош инженери

тасдиқлаган программа асосида ишларнинг хавфсиз усуллари ва қабулларига уқитади. Уқитиш тугагандан сўнг ишчининг кўрсатилган ишларнинг усуллари ва ишлаб чиқариш санитарияси талабларини ўзлаштирганлиги юзасидан олган билими текширилиб, унга гувоҳнома берилади.

#### **4.2. Компютерлардан фойдаланишда уқув аудиторияларига қўйиладиган талаблар**

Таълим жараёнида, айниқса амалий, тажриба ва мустақил таълим соатларида электрон таълим ресурсларидан фойдаланиш - бевосита компютер ва замонавий АКТ асосида амалга оширилади. Компютер технологиялари билан жихозланган ўқув аудиторияларида кислороднинг концентрацияланиши пасайиши, озоннинг кўпайиши бу эса ўз навбатида хаво атмосфераси таркибидаги озоннинг чегаравий қийматларидан ҳам ошиши каби ҳолатлар кузатилиши мумкин. Хавонинг ионли таркиби бузилади. Ўқув хонасида агар компютерлар ёқилган ва ҳеч ким бўлмаса, манфий ионлар миқдори чегаравий қийматдан ошиб кетиши мумкин, бу эса одамлар кайфиятини ёмонлашувига, бош оғриғининг пайдо бўлишига, ишлаш қобилиятининг пасайишига олиб келиши мумкин.

Хаво таркибидаги ионларнинг мумкин бўлган чегаралари қуйидагича:

1. Хонада фойдаланувчилар бўлса, хаво органик табиатли антропоген моддалар ва углерод диоксиди билан ифодаланади. Шу сабабли компютерли хоналарда ҳама иқлимий соҳалар учун оптимал ҳарорат намлик режимини таъминловчи оқимли сурувчи вентеляцияга эга бўлиш тавсия этилади.

2. Ҳарорат ва нисбий намликнинг оптимал ва чегаравий папраметрлари.

Оқимли сурувчи вентеляция бўлмаган вақтда хавонинг маиший кондиционерларнинг унумдорлигига машиналардан, одамлардан, қуёш радиациясидан, сунъий ёритиш манбаларидан бўладиган ортиқча иссиқлик миқдorigа боғлиқ равишда ўрнатиладиган кондиционерларнинг сонини ҳисоблаш мумкин. Ҳисоблашлар вентеляциялар бўйича муҳандис томонидан

хисобланиши керак. Кондиционерлар меъёр кийматидан ошмайдиган шовқинни хосил қилмаслиги керак. Кондиционерлар ишлаётган вақтда хона эшиклари ёпиқ бўлиши керак. Оқимли сўрувчи вентеляция ва кондиционерлар бўлмаган вақтда хонани хар бир танаффусда ва хар қандай хавода шамоллатишни ташкил этиб туриш керак. Агар об-хаво шароитлари йўл қўйса, дарсларни очик деразалари хонада хам ўтказиш мумкин.

Компютерлар билан жихозланган уқув аудиториялари юқори шовқинли ва вибрацияли хоналардан (машина заллари, гимнастика заллар, мусика махсулотлари учун кабинетлар, устохоналар ва х.к.) шунингдек кўча магистралларидан, катта транспорт қатнови кўчаларидан узоқроқ бўлиши керак. Бундай уқув аудиторияларда шовқин даражаси 40, хаво иситиш тизими, вентеляция ва кондиционер ишлаганда эса 35, уқув машгулотлари вақтида эса шовқин 50 дан ошмаслиги керак.

Шовқинни пасайтириш мақсадида компютер хоналарида товушни ютувчи юқори самарали материаллар ишлатилиши мақсадга мувофик. Хонада товушни ютувчи қушимча восита сифатида - деразалардаги огир, зич пардалар хизмат қилиши мумкин.

Улар девордан 15 - 20 см масофада осилган булиши керак. Бундай парда эни дераза энидан 2 марта катта, бир хил рангда ва уқув хонаси ранги билан мос тушган булиши керак.

Хар хил полимер материаллар ишлатилган компютер хоналарида олиб борилган гигиеник тадқиқотлар шуни кўрсатадики, хавода хар хил фенол, формалдегид, хлорли винил аммиак, толуол, хоказоларга эга булган оқим мавжуд экан. Улар мумкин булган чегаравий таъсир кўрсатиши ва инсоннинг мехнат қобилиятини кескин пасайтириши мумкин. Хаво хароратининг ошиши хона хаво мухитидан химик моддаларнинг мавжудлиги бир хил даражада сезилмайдиган ёқимсиз хид туғдиради. Бу моддалар концентрацияси об - хаво шароитлари, ташқаридаги хаво харорати, атмосфера босими, шамолнинг тезлиги ва хона деразаларига йўналиши ва хоказолар билан аниқланади.

Ёмгирли об - хавода ташқи хаво хароратининг кўтарилишида сўрувчи вентеляция иши ёмонлашади.

Шу сабаларга кўра, компьютер хоналари деворларини безашда хавога зарарли химик моддалар ва бирлашмалар ажратувчи синтез синтетик матириаллардан фойдаланишга йўл қўйилмайди. Хонанинг шипи ок-кўк фон балан оқланиши, деворлари эса яшил рангга оқланиши керак. Бу ранглар ва офтоб нурланиши бизга керакли ранг иқлимини яратиб беради.

Електр ҳавфсизлиги: Компьютер хонасида ҳамма жихозлар электр токида ишлайди, шунинг учун электр шикастланишига учраш мумкин. Компьютерларни ерга улаш ва ерга улаш химоясига талаблар. Ҳамма компьютерларда электр тармоғига улаш учун махсус система ишлатилади ва унда "0" улаш химояси қўлланилган. "0" га улаш химояси бу "0" симини корпусларга боғлаш ва хар хил иссиқликда ишлайдиган автоматларни ишга туширувчи системадир.

Шахсий компьютердан фойдаланиш қоидалари: Шахсий компьютерни ишга тайёрлаш ва ўчириш тартиблари мавжуд. Авваламбор компьютернинг тармоқ кучланишига мослигини текшириш керак. Шахсий компьютерлар 220 ёки 110 кучланишли электр тармоқларида ишлаши мумкин. Шахсий компьютерлар учун кучланишнинг ўзгариши, айниқса кескин ўзгаришлар ҳавfli бўлиши мумкин. Шунинг учун махсус стабилизаторлардан ёки электр қувватининг узлуксизлигини таъминловчи - УРС қурилмасидан фойдаланиш тавсия этилади. Махсус УПС қурилмаси электр қувватини ўзгармас холда ушлаб туради, ҳамда электр манбаи ўчирилгандан кейин муайян вақт давомида компьютер ишлашини таъминлаб туради.

## Хулоса

Ишда таъкидланганидек, ушбу БМИда бажарилган ишлар ҳозирги кунда дунёда ривожланиб бораётган илм-фан йўналишларидан бири бўлган “Рақамли тасвирларни қайта ишлаш ва таниб олиш” (инг. “Digital Image Processing and Recognition”) илмий йўналишининг муҳим босқичдаги ишлари, яъни тасвирдаги объектнинг идентификацион белгиларини аниқлаш масалаларини ҳал қилиш билан боғлиқдир. Бу босқичдаги ишларнинг сифати бевосита яқуний компьютер тизимларининг аниқлигига таъсир кўрсатади. Шу маънода, бажарилган ишлар долзарб аҳамият касб этади.

Ушбу БМИда амалга оширилган ишларни асосан қуйидагиларга ажратиш мумкин:

- рақамли тасвир хусусиятлари ва уни қайта ишлаш усул ва алгоритмлари тадқиқ қилинди;
- рақамли тасвирнинг идентификацион белгилари тадқиқ қилинди;
- “Локал бинар образлар” операторининг хусусиятлари тадқиқ қилинди;
- “Локал бинар образлар” оператори ёрдамида тасвир белгиларини ажратиб олиш усули ва алгоритмлари асосида амалий дастур ишлаб чиқилди;
- амалий дастурни яратишда ҳозирда дунё тажрибасида кенг фойдаланилаётган OpenCV кутубхонасининг имкониятлари қаралди;
- амалий дастурни яратишда энг замонавий дастурлаш воситаси ҳисобланган Qt Creator C++ дастурий воситасидан фойдаланилди;
- алгоритмлар асосида амалий дастурни яратиш технологияси ишлаб чиқилди;
- ишлаб чиқарилган “Local Binary Pattern” дастури тестдан ўтказилди ундан фойдаланиш бўйича йўриқнома ишлаб чиқилди ва ҳ.к.

Ишлаб чиқилган амалий дастур натижалари асосида хулоса қилиш мумкинки, “Локал бинар образлар” оператори асосида ишончли белгиларни ажратиб олиш мумкин. Яъни, бу белгилар келгусида тасвирдаги объектларни таққослаш, яъни таниб олиш масаласини ишончли ечиш имкониятини беради.

Таъкидлаш керакки, “Локал бинар образлар” операторлари оиласига мансуб яна бир қатор бошқа операторлар ҳам мавжуд. Ушбу БМИда уларнинг назарий асослари ҳам қаралиб, улар асосида дастурий натижалар ҳам олинди. Бу операторларнинг хусусиятлари ва якуний натижалари асосида келажакда қўйиладиган масала учун улардан зарурийсини танлаш мумкин.

Демак, хулоса қилиш мумкинки, “Локал бинар образлар” оператори асосида рақамли тасвирнинг муҳим идентификацион белгиларини ишончли аниқлаш мумкин ва ундан келажакда тасвирларни таниб олиш билан боғлиқ турли илмий-амалий масалаларни ҳал қилишда самарали фойдаланиш мумкин.

Ўйлаймизки, бу йўналишда Республикамизда ҳам илмий-тадқиқот ишларини олиб бориш ва ўз навбатида улар асосида импорт ўрнини босувчи миллий дастурларни ишлаб чиқиш муҳим масаладир. БМИда бажарилган ишлар ушбу соҳада қўйилган яна бир ижобий қадам бўлади, деб умид қиламиз.

## Фойдаланилган адабиётлар

1. Ўзбекистон Республикаси Президентининг «Ахборот – коммуникация технологияларини янада ривожлантиришга оид кўшимча чора тadbирлар тўғрисида»ги Қарори. 21.03.2012 й. №ПҚ-1730.
2. Баркамол авлод - Ўзбекистон тараққиётининг пойдевори. (Ўзбекистон Республикасининг «Таълим Тўғрисида» ва «Кадрлар тайёрлаш миллий дастури тўғрисида»ги қонунлар). -Т.: «Шарк», 1998. -64 б.
3. Ўзбекистон Республикаси Олий ва ўрта махсус таълим вазирлиги тизимида ахборот- коммуникация технологияларини жорий этиш ва ривожлантириш концепцияси, Тошкент, 2013 й.
4. Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Prentice Hall. -2002. -793 p.
5. Pratt, William K. Digital image processing : PIKS Scientific inside / William K. Pratt.—, 4th ed. – 2007. – 782 p.
6. Форсайт Дэвид А., Понс Джин. Компьютерное зрение. Современный подход = Computer Vision: A Modern Approach. — М.: Вильямс, 2004. — 928 с.
7. Ahonen, T., Hadid, A., and Pietikainen, M. Face Recognition with Local Binary Patterns. Computer Vision - ECCV, pp. 469–481. 2004.
8. Gary Bradski, Adrian Kaehler. Learning OpenCV, Computer Vision with the OpenCV Library, -USA, 2008, -580 p.
9. Prateek Joshi, David Millán Escrivá, Vinícius Godoy. OpenCV By Example, - UK, 2016, -270 p.
10. Фарберман Б. Илғор педагогик технологиялар. –Тошкент. ЎзР ФА ”Фан” нашриёти, 2000. – 146 б.
11. Ишмухаммедов Р., Абдукодиров А., Пардаев А. Таълимда инновацион технологиялар. –Т.: Истеъдод, 2008. – 180 б.
12. Ишматов Қ. Педагогик технология. Ўқув қўлланма, Наманган, 2004.
13. Михридинова М.Т. “Таълимда ахборот технологиялари” фанидан маъруза матни, ТАТУ, Тошкент, 2011 й., 144 б.

14. Эргашев Х., Умумий экология, Тошкент, Ўзбекистон, 2003 й.
15. Sultonov P. Ekologiya va atrof-muhitni muhofaza qilish asoslari. Toshkent, 2007. -238 b.
16. Рахимова Х. ва бошқ. Мехнатни муҳофаза қилиш: Олий ўқув юртлари талабалари учун ўқув қўлланма, -Т.: Ўзбекистон, 2003. -216 б.
17. Макс Шлее. Qt 5.3 Профессиональное программирование на C++. -СПб.: «БХВ-Петербург», 2015. - С. 928.
18. [www.ziyonet.uz](http://www.ziyonet.uz)
19. [www.edu.uz](http://www.edu.uz)
20. [www.lex.uz](http://www.lex.uz)
21. [www.gov.uz](http://www.gov.uz)
22. [www.google.com](http://www.google.com)
23. [www.en.wikipedia.org](http://www.en.wikipedia.org)
24. [www.twirpx.com](http://www.twirpx.com)
25. [www.computergraphics.ru](http://www.computergraphics.ru)
26. <http://opencv.org>
27. <http://biometrics.cse.msu.edu>
28. <http://www.biometrics.org>

## Илова

### Дастур матни

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include "opencv2/opencv.hpp"
#include "QDebug"
#include "math.h"
#include "QFileDialog"
#include "qmessagebox.h"

using namespace cv;

Mat LBP(Mat inp_img);
Mat LBP2r(Mat inp_img);

Mat TP_LBP(Mat cs_lbp_img);

Mat CS_LBP(Mat inp_img);
Mat CS_LBP2r(Mat inp_img);

Mat LDP(const Mat &inp_img);

void histogramBuild(Mat img);

int his[256];

Mat globImg;
Mat img;
void DFT();

void glcm(const Mat img, vector<double> &vec_energy);

/// Global variables
Mat src, src_gray;

int max_thresh = 255;

char* source_window = "Source image";
char* corners_window = "Corners detected";

/// Function header
void cornerHarris_demo( int, void* );

//////////

bool applyGray = false;
bool applyBlur = false;
bool applySobel = false;

String windowNameFace = "Face";
void applyFilters();
void grayCallback(int state, void* userData);
void rgbCallback(int state, void* userData);
void blurCallback(int state, void* userData);
void sobelCallback(int state, void* userData);

using namespace std;
```

```
Mat x2y2, xy, mtrace, x_derivative,
y_derivative, x2_derivative, y2_derivative,
xy_derivative, x2g_derivative,
y2g_derivative, xyg_derivative, dst, dst_norm,
dst_norm_scaled;
int thresh = 128;

void onTrackbar( int, void* );

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);

    this->setWindowTitle("БМИ мавзуси: "Локал бинар
образлар" оператори ёрдамида тасвир белгиларини
ажратиш алгоритмлари ва дастурини шлаб чиқиш.");
    this->statusBar()->showMessage("Битирувчи: Зокиров
Ф.Х.; Рахбар: Тухтасинов М.Т.");

    QString imgPath = QApplication-
>applicationDirPath()+"/temp.jpg";
    string imgPath2 = imgPath.toStdString();
    globImg = imread(imgPath2);
}

MainWindow::~MainWindow()
{
    delete ui;
}

void DFT(){

    cvtColor(globImg, img, COLOR_BGR2GRAY);

    int M = getOptimalDFTSize( img.rows );
    int N = getOptimalDFTSize( img.cols );
    Mat padded;
    copyMakeBorder(img, padded, 0, M - img.rows, 0, N -
img.cols,
        BORDER_CONSTANT, Scalar::all(0));
    Mat planes[] = {Mat_<float>(padded),
Mat::zeros(padded.size(), CV_32F)};
    Mat complexImg;
    merge(planes, 2, complexImg);
    dft(complexImg, complexImg);
    // compute log(1 + sqrt(Re(DFT(img))**2 +
Im(DFT(img))**2))
    split(complexImg, planes);
    magnitude(planes[0], planes[1], planes[0]);
    Mat mag = planes[0];
    mag += Scalar::all(1);
    log(mag, mag);
    // crop the spectrum, if it has an odd number of rows or
columns
    mag = mag(Rect(0, 0, mag.cols & -2, mag.rows & -2));
    int cx = mag.cols/2;
```

```

int cy = mag.rows/2;

// so that the origin is at the image center
Mat tmp;
Mat q0(mag, Rect(0, 0, cx, cy));
Mat q1(mag, Rect(cx, 0, cx, cy));
Mat q2(mag, Rect(0, cy, cx, cy));
Mat q3(mag, Rect(cx, cy, cx, cy));
q0.copyTo(tmp);
q3.copyTo(q0);
tmp.copyTo(q3);
q1.copyTo(tmp);
q2.copyTo(q1);
tmp.copyTo(q2);
normalize(mag, mag, 0, 1, CV_MINMAX);
imshow("Fur'ye spektri", mag);
normalize(mag, mag, 0, 255, CV_MINMAX);
mag.copyTo(img);
}

//////////

void MainWindow::on_action_triggered()
{
    QString str = QFileDialog::getOpenFileName(0, "Тасвир
файлларни очиш...", QDir::currentPath(), "*.jpg *.jpeg
*.bmp *.ico *.png");

    String file;

    file = str.toStdString();
    QString qFile;
    qFile= QString::fromStdString(file);

    if (file.length()<=1 || QFile::exists(qFile)==false) return;

    globlmg = imread(file);

    imshow("Manba tasvir", globlmg);

    cvtColor(globlmg, globlmg, CV_BGR2RGB);
    QImage imgIn = QImage((uchar*) globlmg.data,
globlmg.cols, globlmg.rows, globlmg.step,
QImage::Format_RGB888);
    QPixmap PixImage = QPixmap::fromImage(imgIn);
    QPixmap scaledImage = PixImage.scaled(ui->label-
>size(),
Qt::KeepAspectRatio,
Qt::FastTransformation);
    ui->label->setPixmap(scaledImage);
}

void MainWindow::on_action_destroyed()
{
}

//////////

// rearrange the quadrants of Fourier image
// LBP - Local Binary Pattern
Mat LBP(Mat inp_img){

    int t = 5;

    int binar[8];

    Mat Image;
    if (inp_img.channels()==3)
        cvtColor(inp_img, Image, CV_BGR2GRAY);
    else Image = inp_img;

    Mat lbp(inp_img.rows, inp_img.cols, Image.type());

    int center = 0;
    int center_lbp = 0;

    for (int row = 1; row < Image.rows-1; row++)
    {
        for (int col = 1; col < Image.cols-1; col++)
        {
            center = Image.at<uchar>(row, col) + t;
            center_lbp = 0;

            if ( center <= Image.at<uchar>(row-1, col-1) )
                { center_lbp += 1; binar[0]=1; }

            if ( center <= Image.at<uchar>(row-1, col) )
                { center_lbp += 2; binar[1]=1; }

            if ( center <= Image.at<uchar>(row-1, col+1) )
                { center_lbp += 4; binar[2]=1; }

            if ( center <= Image.at<uchar>(row, col+1) )
                { center_lbp += 8; binar[3]=1; }

            if ( center <= Image.at<uchar>(row+1, col+1) )
                { center_lbp += 16; binar[4]=1; }

            if ( center <= Image.at<uchar>(row+1, col) )
                { center_lbp += 32; binar[5]=1; }

            if ( center <= Image.at<uchar>(row+1, col-1) )
                { center_lbp += 64; binar[6]=1; }

            if ( center <= Image.at<uchar>(row, col-1) )
                { center_lbp += 128; binar[7]=1; }

            // uniform ni xisoblash
            char uniform=0;
            for (int i=0; i<=6; i++)
                if (binar[i] != binar[i+1])
                    uniform++;

            if (uniform <= 2)
                lbp.at<uchar>(row, col) = center_lbp;
        }
    }

    return lbp;
}

```

```

void MainWindow::on_action_2_triggered()
{
    if (globlmg.empty()==true) {
        QMessageBox::critical(this, "Хабар!", "Хотирага
тасвир юкланмаган!");
        return;
    }

    Mat LBP_Img = LBP(globlmg);
    imshow("Tasvirning LBO si", LBP_Img);

    histogramBuild(LBP_Img);

    // gistogramma qiymatlarini chiqarish
    ui->listWidget->clear();
    for (int i=0; i<256; ++i) {
        QString str = "H[" +
QString::number(i)+"="+QString::number(his[i]);
        ui->listWidget->addItem(str);
    }
}

////////////////////
void histogramBuild(Mat lmg){
    if (lmg.empty()) return;

    int w = lmg.cols;
    int h = lmg.rows;

    // gistogrammani to'ldirish
    for (int i=0; i<256; ++i) his[i] = 0;

    for (int j = 0; j < h-1; ++j)
        for (int i = 0; i < w-1; ++i)
            {
                uchar R = lmg.at<uchar>(j,i);
                his[R]++;
            }

    Vec3b colorWhite;
    colorWhite.val[0] = 230;
    colorWhite.val[1] = 230;
    colorWhite.val[2] = 230;

    Point pt1, pt2;

    // gistogramma uchun rang...
    CvScalar colorH;
    colorH.val[0]=55;
    colorH.val[1]=50;
    colorH.val[2]=255;

    Mat imgH;
    imgH.create(200, 256, CV_8UC3);
    // tozalash
    for (int j = 0; j < 200; ++j)
        for (int i = 0; i < 256; ++i)
            imgH.at<Vec3b>(j,i) = colorWhite;

    // max R ni topish
    int max = 0;
    for (int i=0; i<256; ++i)
        if (his[i]>max) max = his[i];

    // R uchun gistogrammani chizish
    for (int i=0; i<256; ++i){
        pt1.x = i;
        pt1.y = 199;

        pt2.x = i;
        int k = his[i] * 200 / max;
        pt2.y = (200-k);

        line(imgH, pt1, pt2, colorH);
    }

    imshow("Gistogrammasi", imgH);
}

////////////////////
// CS-LBP - Center-Symmetric Local Binary Pattern
Mat CS_LBP(Mat inp_img){

    int binar[8];

    Mat Image;
    if (inp_img.channels()==3)
        cvtColor(inp_img, Image, CV_BGR2GRAY);
    else Image = inp_img;

    Mat lbp(inp_img.rows, inp_img.cols, Image.type());

    int center = 0;
    int center_lbp = 0;

    for (int row = 1; row < Image.rows-1; row++)
    {
        for (int col = 1; col < Image.cols-1; col++)
        {

            center_lbp = 0;

            if ( Image.at<uchar>(row, col+1) -
Image.at<uchar>(row, col-1) >= 0)
                center_lbp += 1;

            if ( Image.at<uchar>(row+1, col+1) -
Image.at<uchar>(row-1, col-1) >= 0)
                center_lbp += 2;

            if ( Image.at<uchar>(row+1, col) -
Image.at<uchar>(row-1, col) >= 0)
                center_lbp += 4;

            if ( Image.at<uchar>(row+1, col-1) -
Image.at<uchar>(row-1, col+1) >= 0)
                center_lbp += 8;
            lbp.at<uchar>(row, col) = center_lbp;
        }
    }

    return lbp;
}

```

```

// CS_LBP asosida TP_LBP ni xisoblash

Mat TP_LBP(Mat cs_lbp_img){

Mat Image;
if (cs_lbp_img.channels()==3)
    cvtColor(cs_lbp_img, Image, CV_BGR2GRAY);
else Image = cs_lbp_img;

    Mat lbp(cs_lbp_img.rows, cs_lbp_img.cols,
cs_lbp_img.type());

    int center = 0;
    int center_lbp = 0;

    for (int row = 1; row < Image.rows-1; row++)
    {
        for (int col = 1; col < Image.cols-1; col++)
        {
            center_lbp = 0;

            int Cp=Image.at<uchar>(row, col);
            int C0=Image.at<uchar>(row, col-1);
            int C1=Image.at<uchar>(row-1, col-1);
            int C2=Image.at<uchar>(row-1, col);
            int C3=Image.at<uchar>(row-1, col+1);
            int C4=Image.at<uchar>(row, col+1);
            int C5=Image.at<uchar>(row+1, col+1);
            int C6=Image.at<uchar>(row+1, col);
            int C7=Image.at<uchar>(row+1, col-1);

            double T=0.01;

            if ( abs( abs(C0-Cp) - abs(C2-Cp) ) >= T )
                center_lbp += 1;

            if ( abs( abs(C1-Cp) - abs(C3-Cp) ) >= T )
                center_lbp += 2;

            if ( abs( abs(C2-Cp) - abs(C4-Cp) ) >= T )
                center_lbp += 4;

            if ( abs( abs(C3-Cp) - abs(C5-Cp) ) >= T )
                center_lbp += 8;

            if ( abs( abs(C4-Cp) - abs(C6-Cp) ) >= T )
                center_lbp += 16;

            if ( abs( abs(C5-Cp) - abs(C7-Cp) ) >= T )
                center_lbp += 32;

            if ( abs( abs(C6-Cp) - abs(C0-Cp) ) >= T )
                center_lbp += 64;

            if ( abs( abs(C7-Cp) - abs(C1-Cp) ) >= T )
                center_lbp += 128;

            lbp.at<uchar>(row, col) = center_lbp;

        }
    }
}

```

```

return lbp;
}

// CS-LBP - 2 radiusli
Mat CS_LBP2r(Mat inp_img){

Mat Image;
if (inp_img.channels()==3)
    cvtColor(inp_img, Image, CV_BGR2GRAY);
else Image = inp_img;

    Mat lbp(inp_img.rows, inp_img.cols, Image.type());

    // soxani qoraytirish
    for (int row = 0; row < Image.rows; row++)
        for (int col = 0; col < Image.cols; col++)
            lbp.at<uchar>(row, col) = 0;

    int center = 0;
    int center_lbp = 0;

    for (int row = 6; row < Image.rows-6; row++)
    {
        for (int col = 6; col < Image.cols-6; col++)
        {
            center_lbp = 0;

            int n0, n1,n2,n3,n4,n5,n6,n7;

            int sum=0;
            for (int j=row-1; j<=row+1; j++)
                for (int i=col+2; i<=col+4; i++)
                    sum += Image.at<uchar>(j, i);
            n0= sum / 9;

            sum=0;
            for (int j=row+2; j<=row+4; j++)
                for (int i=col+2; i<=col+4; i++)
                    sum += Image.at<uchar>(j, i);
            n1= sum / 9;

            sum=0;
            for (int j=row+2; j<=row+4; j++)
                for (int i=col-1; i<=col+1; i++)
                    sum += Image.at<uchar>(j, i);
            n2= sum / 9;

            sum=0;
            for (int j=row+2; j<=row+4; j++)
                for (int i=col-4; i<=col-2; i++)
                    sum += Image.at<uchar>(j, i);
            n3= sum / 9;

            sum=0;
            for (int j=row-1; j<=row+1; j++)
                for (int i=col-4; i<=col-2; i++)
                    sum += Image.at<uchar>(j, i);
            n4= sum / 9;

            sum=0;
            for (int j=row-4; j<=row-2; j++)
                for (int i=col-4; i<=col-2; i++)
                    sum += Image.at<uchar>(j, i);

```

```

n5= sum / 9;

sum=0;
for (int j=row-4; j<=row-2; j++)
    for (int i=col-1; i<=col-1; i++)
        sum += Image.at<uchar>(j, i);
n6= sum / 9;

sum=0;
for (int j=row-4; j<=row-2; j++)
    for (int i=col+2; i<=col+4; i++)
        sum += Image.at<uchar>(j, i);
n7= sum / 9;

if (n0-n4>=0) center_lbp += 1;
if (n1-n5>=0) center_lbp += 2;
if (n2-n6>=0) center_lbp += 4;
if (n3-n7>=0) center_lbp += 8;

//if (center_lbp >= 1)
lbp.at<uchar>(row, col) = center_lbp;
}
}

return lbp;
}

// LBP 2 radiusli
Mat LBP2r(Mat inp_img){

int t = 5;

int binar[8];
Mat Image;
if (inp_img.channels()==3)
    cvtColor(inp_img, Image, CV_BGR2GRAY);
else Image = inp_img;

Mat lbp(inp_img.rows, inp_img.cols, Image.type());

int center = 0;
int center_lbp = 0;

// chekadagi 2 ta piksel soxani qoraytirish
for (int row = 0; row <= 1; row++) // tepasi
    for (int col = 0; col < Image.cols; col++)
        lbp.at<uchar>(row, col) = 0;

for (int row = Image.rows-2; row < Image.rows; row++)
// pasti
    for (int col = 0; col < Image.cols; col++)
        lbp.at<uchar>(row, col) = 0;

for (int col = 0; col <= 1; col++) // chapi
    for (int row = 0; row < Image.rows; row++)
        lbp.at<uchar>(row, col) = 0;

for (int col = Image.cols-2; col < Image.cols; col++) //
o'ngi
    for (int row = 0; row < Image.rows; row++)
        lbp.at<uchar>(row, col) = 0;

```

```

//////////
// LBP ni xisoblash
for (int row = 2; row < Image.rows-2; row++)
{
    for (int col = 2; col < Image.cols-2; col++)
    {
        for (int i=0; i<=7; i++) binar[i]=0;

        center = Image.at<uchar>(row, col) + t;
        center_lbp = 0;

        if ( center <= (Image.at<uchar>(row-1, col-1) +
            Image.at<uchar>(row-2, col-1) +
            Image.at<uchar>(row-2, col-2) +
            Image.at<uchar>(row-1, col-2) ) / 4 )
            center_lbp += 1;

        if ( center <= (Image.at<uchar>(row-1, col)+
            Image.at<uchar>(row-2, col) ) / 2 )
            center_lbp += 2;

        if ( center <= (Image.at<uchar>(row-1, col+1)+
            Image.at<uchar>(row-2, col+1)+
            Image.at<uchar>(row-2, col+2)+
            Image.at<uchar>(row-1, col+2)) / 4 )
            center_lbp += 4;

        if ( center <= (Image.at<uchar>(row, col+1)+
            Image.at<uchar>(row, col+2) ) / 2 )
            center_lbp += 8;

        if ( center <= ( Image.at<uchar>(row+1, col+1)+
            Image.at<uchar>(row+1, col+2)+
            Image.at<uchar>(row+2, col+1)+
            Image.at<uchar>(row+2, col+2)) / 4 )
            center_lbp += 16;

        if ( center <= ( Image.at<uchar>(row+1, col) +
            Image.at<uchar>(row+2, col)) / 2 )
            center_lbp += 32;

        if ( center <= ( Image.at<uchar>(row+1, col-1)+
            Image.at<uchar>(row+1, col-2)+
            Image.at<uchar>(row+2, col-1)+
            Image.at<uchar>(row+2, col-2) ) / 4 )
            center_lbp += 64;

        if ( center <= ( Image.at<uchar>(row, col-1)+
            Image.at<uchar>(row, col-2)) / 2 )
            center_lbp += 128;

        if (center_lbp >= 1)// && uniform <= 2)
            lbp.at<uchar>(row, col) = center_lbp;
        else
            lbp.at<uchar>(row, col) = 0;
    }
}

return lbp;
}

```

```

////////// Local Directional Pattern
Mat LDP(const Mat &inp_img)
{
    int East[9]={5, 5, 5, -3, 0, -3, -3, -3, -3};
    int Northeast[9]={-3, 5, 5, -3, 0, 5, -3, -3, -3};
    int North[9]={-3, -3, 5, -3, 0, 5, -3, -3, 5};
    int Northwest[9]={-3, -3, -3, -3, 0, 5, -3, 5, 5};
    int West[9]={-3, -3, -3, -3, 0, -3, 5, 5, 5};
    int Southwest[9]={-3, -3, -3, 5, 0, -3, 5, 5, -3};
    int South[9]={5, -3, -3, 5, 0, -3, 5, -3, -3};
    int Southeast[9]={5, 5, -3, 5, 0, -3, -3, -3, -3};

    int mask_result[8];

    Mat Image;
    if (inp_img.channels()==3)
        cvtColor(inp_img, Image, CV_BGR2GRAY);
    else
        Image = inp_img;

    Mat ldp(inp_img.rows, inp_img.cols, Image.type());

    // umumiy qoraytirish
    for (int row = 0; row < ldp.rows; row++)
        for (int col = 0; col < ldp.cols; col++)
            ldp.at<uchar>(row, col) = 0;

    int center = 0;

    for (int row = 1; row < Image.rows-1; row++)
    {
        for (int col = 1; col < Image.cols-1; col++)
        {
            int vr=0;

            int kk=0;
            int sum=0;
            for (int i=col-1; i<=col+1; ++i)
                for (int j=row-1; j<=row+1; ++j)
                {
                    int f = Image.at<uchar>(j, i);
                    sum += f * East[kk];
                    kk++;
                }

            mask_result[vr] = sum;
            ////////////
            vr++;
            kk=0;
            sum=0;
            for (int i=col-1; i<=col+1; ++i)
                for (int j=row-1; j<=row+1; ++j)
                {
                    int f = Image.at<uchar>(j, i);
                    sum += f * Northeast[kk];
                    kk++;
                }

            mask_result[vr] = sum;
            ////////////
            vr++;
            kk=0;
            sum=0;
        }
    }
}

```

```

for (int i=col-1; i<=col+1; ++i)
    for (int j=row-1; j<=row+1; ++j)
    {
        int f = Image.at<uchar>(j, i);
        sum += f * North[kk];
        kk++;
    }

mask_result[vr] = sum;

//////////
vr++;
kk=0;
sum=0;
for (int i=col-1; i<=col+1; ++i)
    for (int j=row-1; j<=row+1; ++j)
    {
        int f = Image.at<uchar>(j, i);
        sum += f * Northwest[kk];
        kk++;
    }

mask_result[vr] = sum;

//////////
vr++;
kk=0;
sum=0;
for (int i=col-1; i<=col+1; ++i)
    for (int j=row-1; j<=row+1; ++j)
    {
        int f = Image.at<uchar>(j, i);
        sum += f * West[kk];
        kk++;
    }

mask_result[vr] = sum;

//////////
vr++;
kk=0;
sum=0;
for (int i=col-1; i<=col+1; ++i)
    for (int j=row-1; j<=row+1; ++j)
    {
        int f = Image.at<uchar>(j, i);
        sum += f * Southwest[kk];
        kk++;
    }

mask_result[vr] = sum;

//////////
vr++;
kk=0;
sum=0;
for (int i=col-1; i<=col+1; ++i)
    for (int j=row-1; j<=row+1; ++j)
    {
        int f = Image.at<uchar>(j, i);
        sum += f * South[kk];
        kk++;
    }
}

```

```

mask_result[vr] = sum;

////////////////////
vr++;
kk=0;
sum=0;
for (int i=col-1; i<=col+1; ++i)
    for (int j=row-1; j<=row+1; ++j)
    {
        int f = Image.at<uchar>(j, i);
        sum += f * Southeast[kk];
        kk++;
    }

mask_result[vr] = sum;

int tmp_mas[8];
for (int d=0; d<=7; ++d)
    tmp_mas[d]=mask_result[d];

int elements = sizeof(mask_result) /
sizeof(mask_result[0]);
std::sort(mask_result, mask_result + elements);

/*
int sred=0;
sum=0;
for (int d=0; d<=7; ++d)
    sum += mask_result[d];

sred = sum / 8;
*/

int center_ldp = 0;

for (int d=0; d<=7; ++d)
    if (tmp_mas[d]>mask_result[4]-1)
        center_ldp += pow(2,d+1);

if (center_ldp>0)
    ldp.at<uchar>(row, col) = center_ldp;
else
    ldp.at<uchar>(row, col) = 0;
}
}
return ldp;
}

```

```

void MainWindow::on_action_3_triggered()
{
    if (globlmg.empty()==true) {
        QMessageBox::critical(this, "Хабар !", "Хотирага
тасвир юкланмаган !");
        return;
    }
}

```

```

Mat LBP_Img = LBP2r(globlmg);
imshow("Tasvirning LBO si - 2 radiusli", LBP_Img);

histogramBuild(LBP_Img);

```

```

// gistogramma qiymatlarini chiqarish
ui->listWidget->clear();
for (int i=0; i<256; ++i) {
    QString str = "H[" +
QString::number(i)+"="+QString::number(his[i]);
    ui->listWidget->addItem(str);
}

void MainWindow::on_action_5_triggered()
{
    if (globlmg.empty()==true) {
        QMessageBox::critical(this, "Хабар !", "Хотирага
тасвир юкланмаган !");
        return;
    }
}

```

```

Mat LBP_Img = CS_LBP(globlmg);
imshow("Tasvirning CS LBO si", LBP_Img);

```

```

histogramBuild(LBP_Img);

```

```

// gistogramma qiymatlarini chiqarish
ui->listWidget->clear();
for (int i=0; i<256; ++i) {
    QString str = "H[" +
QString::number(i)+"="+QString::number(his[i]);
    ui->listWidget->addItem(str);
}
}

```

```

void MainWindow::on_action_6_triggered()
{
    if (globlmg.empty()==true) {
        QMessageBox::critical(this, "Хабар !", "Хотирага
тасвир юкланмаган !");
        return;
    }
}

```

```

Mat LBP_Img = CS_LBP2r(globlmg);
imshow("Tasvirning CS LBO si - 2 radiusli", LBP_Img);

```

```

histogramBuild(LBP_Img);

```

```

// gistogramma qiymatlarini chiqarish
ui->listWidget->clear();
for (int i=0; i<256; ++i) {
    QString str = "H[" +
QString::number(i)+"="+QString::number(his[i]);
    ui->listWidget->addItem(str);
}
}

```

```

void MainWindow::on_actionRGB_triggered()
{
    if (globlmg.empty()) return;
}

```

```

int w = globlmg.cols;
int h = globlmg.rows;

```

```

int hisR[256];
int hisG[256];

```

```

int hisB[256];

for (int i=0; i<256; ++i){
    hisR[i] = 0; hisG[i] = 0; hisB[i] = 0;
}

for (int j = 0; j < h-1; ++j)
    for (int i = 0; i < w-1; ++i)
    {

        uchar R = globlmg.at<Vec3b>(j,i)[0];
        uchar G = globlmg.at<Vec3b>(j,i)[1];
        uchar B = globlmg.at<Vec3b>(j,i)[2];

        hisR[R]++;
        hisG[G]++;
        hisB[B]++;

    }

Vec3b colorWhite;
colorWhite.val[0] = 200;
colorWhite.val[1] = 200;
colorWhite.val[2] = 200;

Point pt1, pt2;

// R uchun gistogramma...
CvScalar colorR;
colorR.val[0]=0;
colorR.val[1]=0;
colorR.val[2]=255;

Mat imgR;
imgR.create(200, 256, CV_8UC3);
// tozalash
for (int j = 0; j < 200; ++j)
    for (int i = 0; i < 256; ++i)
        imgR.at<Vec3b>(j,i) = colorWhite;

// max R ni topish
int maxR = 0;
for (int i=0; i<256; ++i)
    if (hisR[i]>maxR) maxR = hisR[i];

// R uchun gistogrammani chizish
for (int i=0; i<256; ++i){
    pt1.x = i;
    pt1.y = 199;

    pt2.x = i;
    int k = hisR[i] * 200 / maxR;
    pt2.y = (200-k);

    line(imgR, pt1, pt2, colorR);
}

imshow("R gistogrammasi", imgR);

// G uchun gistogramma...
CvScalar colorG;
colorG.val[0]=0;
colorG.val[1]=255;
colorG.val[2]=0;

Mat imgG;
imgG.create(200, 256, CV_8UC3);
// tozalash
for (int j = 0; j < 200; ++j)
    for (int i = 0; i < 256; ++i)
        imgG.at<Vec3b>(j,i) = colorWhite;

// max G ni topish
int maxG = 0;
for (int i=0; i<256; ++i)
    if (hisG[i]>maxG) maxG = hisG[i];

// G uchun gistogrammani chizish
for (int i=0; i<256; ++i){
    pt1.x = i;
    pt1.y = 199;

    pt2.x = i;
    int k = hisG[i] * 200 / maxG;
    pt2.y = (200-k);

    line(imgG, pt1, pt2, colorG);
}

imshow("G gistogrammasi", imgG);

// B uchun gistogramma...
CvScalar colorB;
colorB.val[0]=255;
colorB.val[1]=0;
colorB.val[2]=0;

Mat imgB;
imgB.create(200, 256, CV_8UC3);
// tozalash
for (int j = 0; j < 200; ++j)
    for (int i = 0; i < 256; ++i)
        imgB.at<Vec3b>(j,i) = colorWhite;

// max B ni topish
int maxB = 0;
for (int i=0; i<256; ++i)
    if (hisB[i]>maxB) maxB = hisB[i];

// B uchun gistogrammani chizish
for (int i=0; i<256; ++i){
    pt1.x = i;
    pt1.y = 199;

    pt2.x = i;
    int k = hisB[i] * 200 / maxB;
    pt2.y = (200-k);

    line(imgB, pt1, pt2, colorB);
}

imshow("B gistogrammasi", imgB);

}

void MainWindow::on_action_8_triggered()

```

```

{
    int w = globlmg.cols;
    int h = globlmg.rows;

    uchar maxR = 0;
    uchar maxG = 0;
    uchar maxB = 0;

    for (int j = 0; j < h-1; ++j)
        for (int i = 0; i < w-1; ++i)
            {
                uchar R = globlmg.at<Vec3b>(j,i)[0];
                uchar G = globlmg.at<Vec3b>(j,i)[1];
                uchar B = globlmg.at<Vec3b>(j,i)[2];

                if (R>maxR)
                    maxR = R;
                if (G>maxG)
                    maxG = G;
                if (B>maxB)
                    maxB = B;
            }

    QString strR = "R max = " + QString::number(maxR);
    QString strG = "G max = " + QString::number(maxG);
    QString strB = "B max = " + QString::number(maxB);

    //ui->listWidget->clear();
    ui->listWidget-
>addItem("=====");
    ui->listWidget->addItem(strR);
    ui->listWidget->addItem(strG);
    ui->listWidget->addItem(strB);
}

void MainWindow::on_action_9_triggered()
{
    int w = globlmg.cols;
    int h = globlmg.rows;

    uchar minR = 255;
    uchar minG = 255;
    uchar minB = 255;

    for (int j = 0; j < h-1; ++j)
        for (int i = 0; i < w-1; ++i)
            {
                uchar R = globlmg.at<Vec3b>(j,i)[0];
                uchar G = globlmg.at<Vec3b>(j,i)[1];
                uchar B = globlmg.at<Vec3b>(j,i)[2];

                if (R<minR)
                    minR = R;
                if (G<minG)
                    minG = G;
                if (B<minB)
                    minB = B;
            }

    QString strR = "R min = " + QString::number(minR);
    QString strG = "G min = " + QString::number(minG);
    QString strB = "B min = " + QString::number(minB);

    //ui->listWidget->clear();
    ui->listWidget-
>addItem("=====");
    ui->listWidget->addItem(strR);
    ui->listWidget->addItem(strG);
    ui->listWidget->addItem(strB);
}

void MainWindow::on_action_10_triggered()
{
    int w = globlmg.cols;
    int h = globlmg.rows;

    int sumR = 0;
    int sumG = 0;
    int sumB = 0;

    for (int j = 0; j < h-1; ++j)
        for (int i = 0; i < w-1; ++i)
            {
                uchar R = globlmg.at<Vec3b>(j,i)[0];
                uchar G = globlmg.at<Vec3b>(j,i)[1];
                uchar B = globlmg.at<Vec3b>(j,i)[2];

                sumR += R;
                sumG += G;
                sumB += B;
            }

    int urtaR = sumR / (h*w);
    int urtaG = sumG / (h*w);
    int urtaB = sumB / (h*w);
    int urtaRGB = (urtaR + urtaG + urtaB) / 3;

    QString strR = "R o'rtacha = " + QString::number(urtaR);
    QString strG = "G o'rtacha = " + QString::number(urtaG);
    QString strB = "B o'rtacha = " + QString::number(urtaB);
    QString strRGB = "RGB o'rtacha = " +
    QString::number(urtaRGB);

    ui->listWidget-
>addItem("=====");
    ui->listWidget->addItem(strR);
    ui->listWidget->addItem(strG);
    ui->listWidget->addItem(strB);
    ui->listWidget->addItem(strRGB);
}

void MainWindow::on_action_11_triggered()
{
    if (globlmg.empty()) return;
    // kulrangga o'tkazish
    Mat img_gray;
    cvtColor(globlmg, img_gray, COLOR_BGR2GRAY);
}

```

```

int w = img_gray.cols;
int h = img_gray.rows;

int his[256];

for (int i=0; i<256; ++i) his[i] = 0;

for (int j = 0; j < h-1; ++j)
    for (int i = 0; i < w-1; ++i)
        {
            uchar k = img_gray.at<uchar>(j,i);
            his[k]++;
        }

// yoruglik taqsimoti ehtimolligi
double hP[256];
for (int i=0; i<256; ++i) hP[i]= (double) his[i] / (w*h);

// energiyani xisoblash
double sum = 0;
for (int i=0; i<256; ++i)
    sum += pow(hP[i], 2);

QString strP = "Tasvir energiyasi = " +
QString::number(sum);

    ui->listWidget-
>addItem("=====");
    ui->listWidget->addItem(strP);
}

void MainWindow::on_action_13_triggered()
{
    DFT();
}

void MainWindow::on_action_4_triggered()
{
    exit(0);
}

void MainWindow::on_action_7_triggered()
{
    if (globImg.empty()==true) {
        QMessageBox::critical(this, "Хабар !", "Хотирага
тасвир юкланмаган !");
        return;
    }

    Mat LBP_Img = LDP(globImg);
    imshow("Tasvirning LYO (ing.LDP) si ", LBP_Img);

    histogramBuild(LBP_Img);

    // gistogramma qiymatlarini chiqarish
    ui->listWidget->clear();
    for (int i=0; i<256; ++i) {
        QString str = "H[" +
QString::number(i)+"]="+QString::number(his[i]);
        ui->listWidget->addItem(str);
    }
}

```