

Universitatea Tehnică Gheorghe Asachi din Iași

Facultatea de Electronică, Telecomunicații și Tehnologia Informației

Specializarea: Tehnologii și Sisteme de Telecomunicații

LUCRARE DE DIPLOMĂ

Profesor îndrumător:

Conf.Dr.Ing. Cleju Nicolae

Absolvent:

Vieriu-Avădăni Ștefania

Universitatea Tehnică Gheorghe Asachi din Iași

Facultatea de Electronică, Telecomunicații și Tehnologia Informației

Specializarea: Tehnologii și Sisteme de Telecomunicații

Detecția irisului din imagini ale feței

Profesor îndrumător:

Conf.Dr.Ing. Cleju Nicolae

Absolvent:

Vieriu-Avădăni Ștefania

Cuprins

Memoriu justificativ	5
1. CAPITOLUL I: Introducere	7
1.1 Scurt istoric.....	8
1.1.1 Tehnici de procesare digitală.....	9
1.1.2 Senzori de preluare a imaginii.....	10
1.2 Fundamentare teoretică	11
1.2.1 Extensia unei imagini	11
1.2.2 Spații de culoare	12
1.2.3 Rezoluție.....	14
1.2.4 Luminanță	15
1.2.5 Contrast	16
1.2.6 Histograma	17
1.3 Algoritmi de procesare a imaginilor	18
1.3.1 Convoluție	18
1.3.2 Gaussian Blur	19
1.4 Tehnici de procesare a imaginilor	20
1.4.1 Clasificare.....	20
1.4.2 Transformare	22
1.4.3 Filtrul median	23
1.5 Identificarea trăsăturilor	24
1.5.1 Detectarea obiectelor.....	25
1.5.2 Caracteristicile Haar	25
1.6 Operații morfologice pe imagini binare	27
1.6.1 Eroziunea.....	27
1.6.2 Dilatarea	28
1.6.3 Deschiderea (O-opening) și închiderea (C-closing)	29

2 CAPITOLUL II: Mediu de dezvoltare.....	30
2.1 Limbajul de programare Matlab	31
2.2 Limbaje de programare similar cu MATLAB	32
3. CAPITOLUL III: Implementare practică	33
3.1 Descrierea aplicației	33
3.2 Structura aplicației/organigramă	34
3.3 Implementarea programului	35
3.4 Rezultatele simulării.....	43
3.5 Analiza algoritmului.Limitări	45
3.5.1 Blurare	46
3.5.2 Luminozitate crescută.....	47
3.5.3 Contrast.....	48
3.5.4 Influența zgomotului	49
4.CAPITOLUL IV:CONCLUZII	50
5. Anexe	51
6.Referințe.....	57

Memoriu justificativ

Percepția stimulilor vizuali reprezintă cel mai important simț al omului fiind o componentă esențială pentru procesarea informațiilor. O mare parte dintre informațiile preluate și procesate de către creier ne oferă detalii semnificative despre dimensiunea, forma și culoarea unui obiect.

Dezvoltarea continuă este necesară în cazul unor aplicații de preluare și analiză ale unor imagini pentru a permite dispozitivelor să evalueze și să extragă informații pe baza stimulilor vizuali. În zilele noastre, datorită tehnologiei avansate există tipuri de aplicații capabile să ajute sau să înlocuiască integral observatorul uman în multe arii de activitate cum ar fi: condusul pe pilot automat, localizarea și urmărirea obiectelor, identificarea persoanelor sau a locațiilor, etc.

Recunoașterea irisului sau detecția irisului este un proces care constă în utilizarea luminii vizibile sau a luminii infraroșii apropiate pentru a face o fotografie cu contrast ridicat a irisului unei persoane. Aceasta este o metodă automatizată de identificare care utilizează tehnici matematice de recunoașterea modelelor pe imagini sau fișiere video, ale unuia sau ale ambelor irisuri ale ochilor unei persoane, ale căror modele complexe sunt unice, stabile și pot fi văzute de la o anumită distanță.

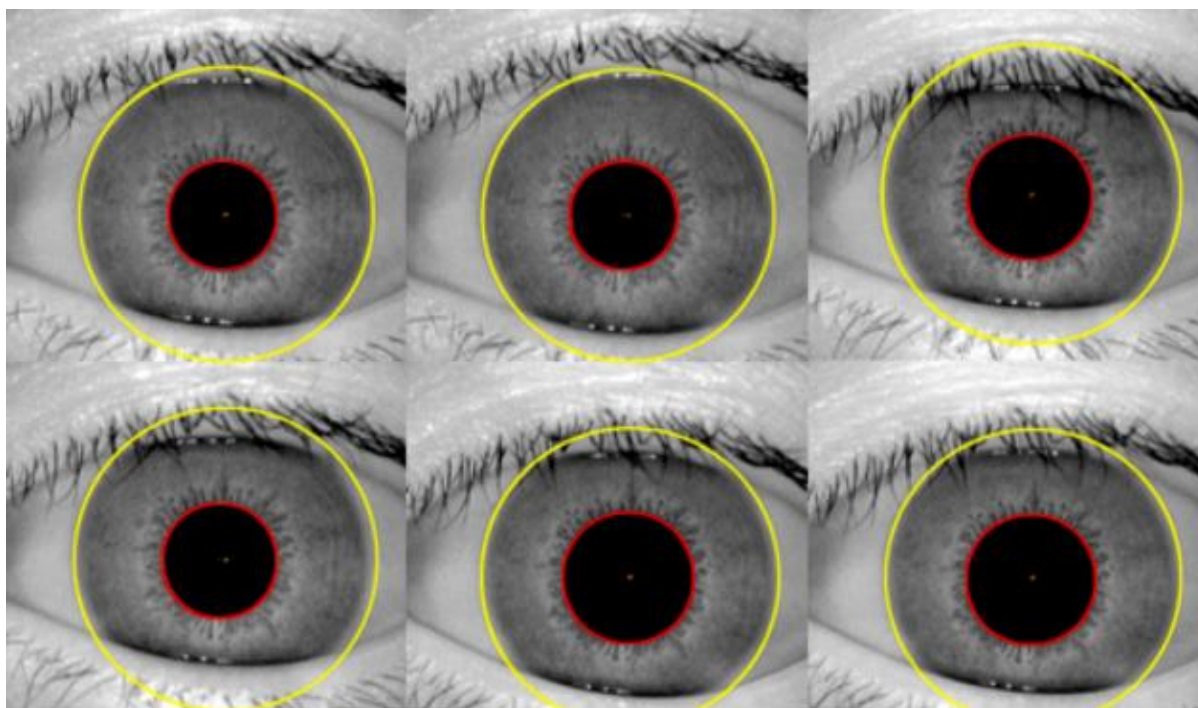


Figura 0.1: Recunoașterea irisului și a pupilei[1]

Camerele de scanare ale irisului pot fi montate pe un perete sau într-o altă locație fixă, sau acestea pot fi și portabile, dar și ținute în mână. Cercetătorii de la Universitatea Carnegie Mellon dezvoltă scanere cu rază lungă de acțiune, ce pot fi folosite pentru a capta imagini pe ascuns de la o distanță de până la 40 de metri.

Acest sistem recunoaște mai întâi fața, apoi recunoaște ochii și apoi determină dacă ochii sunt deschiși sau închiși. Procesul sistemului folosește informațiile obținute din versiunea binară a imaginii pentru a găsi marginile fețelor, îngustând zonele în care pot exista ochii.

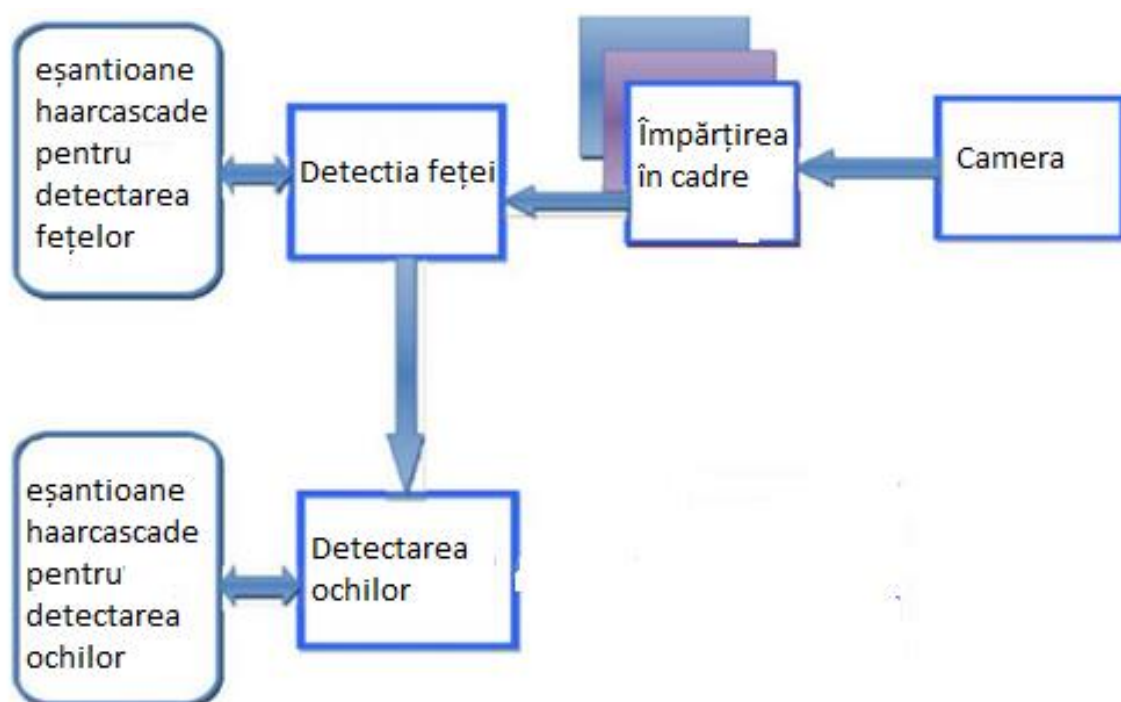


Figura 0.2: Schema aplicației

Pornind cu aceste prime cuvinte, această temă este rezultatul unei curiozități pentru a contribui la îmbunătățirea sistemului de detecție a oboselii prin procesarea imaginilor. Lucrarea „Detectia irisului din imagini ale feței” abordează partea teoretică și partea practică ale unei aplicații de procesare a imaginilor în programul MATLAB, ce oferă funcții de preluare a imaginii, analiză și manipulare prin accesul la imaginile stocate în calculator, sau cele primite de la camera web.

1.CAPITOLUL I: Introducere

Procesarea imaginii este o metodă în care se efectuează unele operații asupra unei imagini, pentru a obține o imagine îmbunătățită sau pentru a extrage unele informații utile din aceasta. Este un tip de procesare a semnalului în care intrarea este o imagine și ieșirea poate fi o imagine sau caracteristici asociate cu imaginea respectivă. Acest tip de detecție se numără printre tehnologiile preferate de majoritatea utilizatorilor, formând un domeniu de bază și în cadrul disciplinelor de inginerie și informatică.

Procesarea imaginii include practic trei pași simpli:

- Importul imaginii prin instrumente de achiziție a imaginii (fotografii, camera web);
- Analiza și manipularea imaginii;
- Ieșirea în care rezultatul poate fi imaginea alterată sau raportul care este bazat pe analiza imaginii;

Există două tipuri de metode utilizate pentru procesarea imaginilor și anume, procesarea imaginii sub format analogic sau sub format digital.

Procesarea analogică a imaginilor poate fi utilizată pentru copiile tipărite, cum ar fi imprimările și fotografiile. Analizii de imagine folosesc diverse elemente fundamentale ale interpretării în timp ce folosesc aceste tehnici vizuale. Tehnicile de procesare a imaginilor digitale ajută la manipularea imaginilor digitale prin utilizarea computerelor. Cele trei faze generale pe care trebuie să le parcurgă toate tipurile de date în timpul utilizării tehnicii digitale sunt preprocesarea, îmbunătățirea și afișarea, extragerea informațiilor.

În funcție de reprezentarea imaginii (sub format analogic sau sub format digital) se pot distinge diferite avantaje și dezavantaje. Procesarea digitală a imaginilor are mai multe avantaje comparativ cu procesarea analogică. Această metodă permite aplicarea unei game largi de algoritmi asupra datelor de intrare prin intermediul unor circuite mult mai ușor de realizat, iar pe de altă parte permit modificarea imaginilor fără riscul de a crea zgomot suplimentar sau de a induce distorsiuni nedorite.

Prin urmare, implementarea digitală poate oferi o performanță mult mai bună în procese simple față de metodele analogice de prelucrare a semnalului, care ar fi imposibile și nejustificate de costisitoare de cele mai multe ori.

1.1 Scurt istoric

Modificarea imaginilor datează încă de pe vremea primelor fotografii realizate în secolul al XIX-lea. Istoria fotografiei a început în antichitate cu descoperirea a două principii critice: proiecția imaginii în camera obscură și observarea că unele substanțe sunt vizibil modificate prin expunerea îndelungată la lumina soarelui. Nu există artefacte sau descrieri care să poată dovedi vreo încercare de a captura imagini cu materiale sensibile la lumină înainte de secolul al XVIII-lea.



Figura 1.1: „Vedere de la fereastra din Le Gras” – prima fotografie din lume care prezintă o ipostază din realitate [2]

Imaginile dezvoltate erau modificate prin intermediul retușurilor cu cerneală sau prin îndepărtarea cernelii prin zgâriere. Fotografiile nedevelopate (negative) erau modificate doar în interiorul camerelor întunecate datorită materialelor folosite care erau fotosensibile.

Aceste tehnici erau în mare parte influențate de limitările tehnologiei. Un exemplu ar fi negativele care puteau fi doar decupate și lipite sau înălbite pentru a lumina cadrul ori cu totul unele obiecte.

Influența asupra evoluției procesării de imagini au avut-o 3 factori: dezvoltarea sistemelor de calcul capabile să proceseze un cumul mare de informații, dezvoltarea metodelor matematice ce se află în spatele algoritmilor de procesare și compresie (crearea și dezvoltarea teoriei matematicii discrete), dar nu în ultimul rând, necesitatea în mai multe industrii a unor aplicații care să permită calculatoarelor să extragă informații importante din imaginile primite de la o camera digitală (în mod special în domeniul militar, medical, astronomie și industrie).

Datorită evoluției extrem de accelerate a tehnologiei, reprezentarea în format digital a devenit din ce în ce mai populară, iar tehnologia a migrat spre acest nou format de reprezentare a datelor fiind ușor stocate, procesate și redistribuite către alte dispozitive pe o varietate extrem de mare de căi de transmisie, cu o rată de detecție a erorilor mult mai bună față de cea a semnalelor analogice.

1.1.1 Tehnici de procesare digitală

Marea majoritate a tehnicilor de procesare digitală a imaginilor au fost dezvoltate după anul 1960, pentru a putea ușura viața oamenilor și au fost utilizate îndeosebi în următoarele aplicații: imagini din satelit, imagistică medicală, teleconferință video și în cele din urmă, în îmbunătățirea fotografiilor și recunoașterea caracterelor. Scopul acestor tehnici de procesare a fost de a îmbunătăți calitatea imaginilor, dar și de a le codifica sau compresa.

După redescoperirea „Transformatei Fourier Rapide (FFT)” s-a făcut posibilă manipularea conținutului de frecvență al semnalelor pe un computer. În cele din urmă, o aplicație remarcabilă realizată în anul 1964 ,avea să revoluționeze procesul de dezvoltare rapid al procesării imaginilor și transmiterea informațiilor. Ea aplică procedee de corecție geometrică și eliminare a zgomotului de pe numeroasele poze realizate de către o probă spațială, generând o primă cartografiere detaliată a lunii.

Dat fiind că puterea de procesare a calculatoarelor creștea considerabil, a apărut posibilitatea ca imaginile să fie procesate în timp real și astfel au apărut primele standarde TV de conversie și compresie a semnalelor audio-video.

1.1.2 Senzori de preluare a imaginii

Sistemul vizual al oamenilor și sistemul digital de preluare a imaginilor au aceleași funcții și scopuri. Ambele analizează datele vizuale, spațiale, având scopul de a extrage informații și de a recunoaște obiectele sau aspectele specifice din imagini. Având în vedere faptul că sistemul vizual uman este prea complex, iar sistemul neuronal nu a fost deslușit complet, chiar și cele mai bune sisteme de calcul și achiziții de imagini nu reușesc să îl egaleze. În loc de a modela sau de a înlocui modul oamenilor de a percepe informații, tehnicile digitale de achiziție și procesare a imaginilor au fost îndreptate către îmbunătățirea acestuia.

Oamenii percep imaginile pe baza stimulilor vizuali, sub formă de fotoni, ce ajung pe cele aproape 130 de milioane de celule fotoreceptoare de pe retină. Celulele fotoreceptoare responsabile de percepția culorii sunt de 3 tipuri, câte un tip pentru câte o porțiune din spectrul luminos: albastru, verde și roșu.

Baza senzorilor moderni de preluare a imaginilor constă în tehnologia metalelor oxidsemiconductoare (MOS), o tehnologie ce a declansat apariția primului tranzistor MOS cu efect de câmp (MOSFET = metal oxid semiconductor field-effect transistor). Proiectat de către Mohamed M. Atalla și Dawon Kahng, în 1959, a dus la dezvoltarea senzorilor de imagine pe bază de semiconductori (CCD și mai târziu, CMOS).

Tehnologia CCD (charge-coupled device) a fost inventată de Willard S. Boyle și George E. Smith, în 1969. În timp ce studiau tehnologia MOS, aceștia au descoperit că o sarcină electrică poate fi comparată cu o bulă magnetică și că poate fi memorată într-o mică capacitate parazită din interiorul unui dispozitiv MOS. După ce au pus mai multe astfel de capacități la un loc și au aplicat o sursă de tensiune asupra acestora, cei doi au construit un circuit semiconductor, ce mai apoi a fost folosit în primele camere video digitale care au transmis imagini către publicul larg .

Senzorul CMOS (CMOS active-pixel sensor) a fost dezvoltat de către o echipă condusă de Eric Fossum, în cadrul NASA, rezultatul fiind depășirea vanzarilor acestor senzori față de vechile tehnologii (CCD).

1.2 Fundamentare teoretică

Procesarea digitală a imaginilor se bazează în practică pe:

- Clasificare
- Extragerea caracteristicilor
- Analiza unui semnal pe mai multe nivele
- Segmentare
- Transformări
- Recunoașterea formelor

Câteva dintre tehnicile folosite în procesarea imaginilor sunt: difuzie anizotropă, editare, restaurare, analiza independentă a componentelor, filtrare mediană, rețele neuronale, ecuații diferențiale parțiale, pixelare, identificarea trăsăturilor, etc.

Pozele digitale sunt caracterizate de o serie de proprietăți influențate de condițiile de mediu în care sunt realizate (contrast, luminozitate, saturație, expunere, etc), aparatul care a realizat fotografia (rezoluție, extensia fișierului - implicit tipul de compresie folosit, rezoluție, spațiu de culoare, etc).

1.2.1 Extensia unei imagini

Un prim pas în procesarea digitală a unei imagini este înfățișată de identificarea formatului și a compresiei sub care aceasta a fost stocată. Aceste extensii sunt împărțite în 2 categorii:

1) Imaginea de tip raster este reprezentată în sisteme de calcul sub forma unei matrici bidimensionale, unde fiecărui pixel îi corespunde intersecția dintre linia și coloana matricei. Ele conțin majoritatea formatelor folosite pentru web:

- GIF (Graphics Interchange Format): utilizat în special în pagini WEB pentru că permite pixeli transparenți, dar conține cel mult 256 culori;
- JPEG(Joint Photographic Experts Groups): un tip de imagine compresată, cu o pierdere minimă a calității fiind utilizată în aplicațiile de consum larg;
- PNG(Portable Network Graphics) a apărut cu menirea de a înlocui formatul GIF și permite pâna la 16 milioane de culori.

2)Imaginea de tip vectorial folosește un sistem de linii și curbe într-un plan cartezian dimensionata după întreaga arie a imaginii. Ea permite o mărire a imaginii fără a distorsiona sau a pierde din calitate. Formatele cele mai întâlnite în rândul imaginilor de tip vectorial sunt: SVG (Scalable Vector Graphics), DXF (AutoCAD drawing exchange format), EPS (Encapsulated PostScript).

1.2.2 Spații de culoare

Pentru a putea fi prelucrate, imaginile grafice sunt digitalizate în primă fază, adică împărțite încât fiecare element să aibă o singură culoare, sau măcar o singură culoare clar dominantă. Acest element numit pixel, deține trei atribute care se pot exprima digital (numeric): culoare, opacitate (transparență) și poziție în matricea în care se divide imaginea.

În cazul în care pixelii sunt foarte mici și numeroși, prezentarea integrală a imaginii pe un ecran sau prin tipărire poate avea o calitate sau o fidelitate optică foarte înaltă ce poate fi asemănată cu calitatea imaginilor tipărite obișnuite (analogice).

O analiză foarte simplă se poate face, spre exemplu, pe o imagine alb-negru (imagine binară). În acest fel, numerele din matricea bidimensională oferă informații legate de conținutul celulei (alb/negru) pentru pixelul cu aceleași coordonate, deci în imagine apar doar pixeli albi și negrii. O imagine binară se obține dintr-o imagine grayscale printr-o operație simplă numită binarizare cu prag (thresholding).



Fig 1.2 Binarizarea unei imagini [3]

Urmatorul tip de imagine este cel în nuanțe de gri. În acest caz, fiecare pixel este de o valoare de la 0 la $2^n - 1$, unde n semnifică numărul de biți ai reprezentării numerice. De cele mai multe ori, imaginile în nuanțe de gri dețin 256 de nuanțe, un număr destul de mare pentru ca diferența dintre zonele cu o nuanță diferită să fie dificil de observat de către ochiul uman.

Crescând în complexitate, ultimul tip de imagine este reprezentat de imaginea color. În acest tip, fiecare pixel este o combinație a celor 3 culori primare: roșu, verde și albastru (RGB), pentru a alcătui toate celelalte culori ale spectrului. Pe lângă sistemul RGB de codare a culorilor, există mai multe spații de culoare.

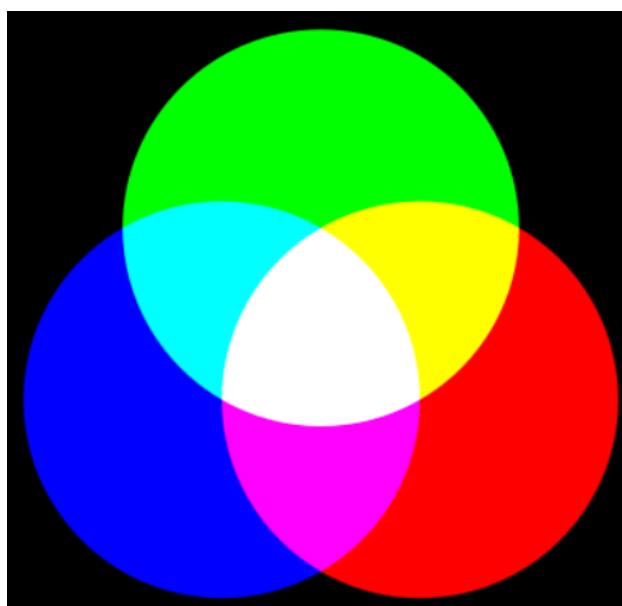


Fig 1.3: Modelul de culoare RGB [4]

Alte spații de culoare bazate pe RGB sunt următoarele: RGBA (bazat pe RGB, la care este adăugat un nou canal, alfa, care indică transparența absolută a pixelului), Adobe RGB (creat pentru compensarea culorilor ce pot fi obținute prin printare), sRGB (utilizat în monitoare, imprimante și web), scRGB și ProPhoto RGB.

CMYK este abrevierea lui „**C**yan, **M**agenta, **Y**ellow, **B**lack”. Aceasta folosește o metodă substractivă de combinare a culorilor (spre deosebire de RGB) și este utilizat în dispozitivele digitale de imprimare color (imprimante, plotere), dar și în mașinile mari (imprimarea offset, rotogravură). Constă în amestecul de substanțe care, reflectând doar o parte din spectrul luminii, apar într-o culoare specifică dorită de utilizator. Acest spațiu de culoare memorează valori ale culorii pentru cyan, magenta, galben, negru, și mai poate fi numită tipărire în patru culori.

HSB cu abrevierea de la „Hue Saturation Brightness”, adică „nuanță, saturație și luminozitate”, este folosit datorită faptului că abordarea sa de a interpreta culorile sub formă de saturație și nuanță este una mult mai naturală decât celelalte abordări.

Modelul HSL (nuanță, saturație, luminanță) este similar cu spațiul HSV, dar stralucirea (eng: brightness) este înlocuită cu luminozitatea (eng: lightness).

1.2.3 Rezoluție

Rezoluția digitală constituie măsura clarității sau a gradului de detaliere a unei imagini digitale. Imaginile sunt eșantionate într-un număr finit de celule dintr-un senzor de imagine, iar prin această metodă, marginea obținută este degradată în raport cu obiectul. Pentru ca imaginea să fie cât mai apropiată de realitate, numărul de celule din senzor trebuie să fie cât mai mare.

Cu cât numărul de pixeli prezenți în imagine este mai mare, nivelul de detalii crește proporțional până la stadiul în care aceasta devine o copie cât mai fidelă a realității. Mai jos avem un exemplu în care se poate observa clar o imagine în mai multe rezoluții:



Figura 1.4: Diferențe de rezoluție [5]

Rezoluția spațială a unei imagini depinde de densitatea spațială și rezoluția optică a dispozitivului folosit pentru capturarea imaginii. Dacă rezoluția optică a dispozitivului de captură este superioară densității spațiale, rezoluția spațială a imaginii obținute este limitată numai de densitatea spațială.

Conform criteriului Nyquist, este nevoie de un interval de eșantionare de cel puțin 2 ori mai mare decât cea mai mare de mare frecvență spațială pe care dorim să o surprindem pentru a obține o fidelitate cât mai mare a imaginii rezultate.

O altă alternativă, pentru criteriul lui Nyquist, este teorema de eșantionare a lui Shannon care susține că dispozitivul cu care se efectuează captura trebuie să folosească un interval de eșantionare mai mic decât jumătatea celei mai mici caracteristici reprezentabile a imaginii optice.

1.2.4 Luminanță

Luminanța este cantitatea de lumină detectată de un ochi uman tipic emis de o suprafață plană. Luminanța este reprezentată de raportul dintre intensitatea luminoasă a unei surse ce generează lumină și proiecția ariei sursei pe un plan. În industrie, acest concept este folosit pentru a caracteriza luminozitatea display-ului unui dispozitiv.

$$L = \frac{d^2\Phi_v}{d\Sigma d\Omega_\Sigma \cos\theta_\Sigma}$$

L = luminanța (strălucirea)

$d^2\Phi_v$ = fluxul luminos ce părăsește unitatea de arie $d\Sigma$, în orice direcție continuă în interiorul unghiului solid $d\Omega_\Sigma$

$d\Sigma$ = unitatea infinitezimală a ariei

$d\Omega_\Sigma$ = unitatea infinitezimală a unghiului solid

θ_Σ = este unghiul dintre normala la suprafața $n\Sigma$ și direcția specificată

1.2.5 Contrast

Termenul de contrast se referă la cantitatea de diferențiere de culoare sau tonuri de gri care există între diferitele caracteristici ale imaginii (atât în imaginile analogice, cât și în cele digitale). Imaginile cu un nivel de contrast mai mare afișează, în general, un grad mai mare de variație a culorii sau a tonurilor de gri decât cele cu contrast mai scăzut.

Contrastul luminanței (Weber Contrast) este definit ca diferența dintre luminozitatea maximă a unei zone dintr-o imagine și luminozitatea dintr-o zonă întunecată, totul fiind împărțit la luminozitatea minimă a pozei.

$$C = (L_{max} - L_{min})/L_{min}$$

Contrastul vârf la vârf (Michelson Contrast) măsoară relația dintre diferența și suma a doua luminozități. Această definiție este des folosită în procesarea semnalelor pentru a determina calitatea unui semnal în raport cu zgomotul.

$$C_{Mich} = (L_{max} - L_{min})/(L_{max} + L_{min})$$

1.2.6 Histograma

Histograma reprezintă un grafic în care sunt reprezentate frecvențele relative ale fiecărei valori ale pixelilor din interiorul unei imagini, pentru fiecare canal, separat. Aceasta ajută la analizarea, și, mai mult chiar, la corectarea contrastului unei imagini. Tehnic vorbind, histograma mapează luminanța, care este definită pornind de la felul în care ochiul uman percepe luminozitatea anumitor culori. Spre exemplu, ochii omului sunt mult mai sensibili la culoarea verde, ceea ce înseamnă că se va vedea culoarea mult mai strălucitor decât albastru.

Fiecare pixel din poza color sau din nuanțele de gri are o valoare a luminanței cuprinsă între 0 și 255. Histograma va mapa în grafic fiecare valoare a luminanței pentru fiecare pixel. Lungimea fiecare linii verticale din histograma arată câți pixeli au o valoare a luminanței de 0, câți de 1, de 2 etc., până la valoarea maximă de 255.

Graficul compus din bare verticale din care este alcătuită histograma arată imaginea distribuției relative de tonuri pe întreaga scală de valori. Imaginea nu va acoperi întreaga scală posibilă de la negru la alb, ceea ce poate sugera că imaginea ar putea fi îmbunătățită din punct de vedere al contrastului.

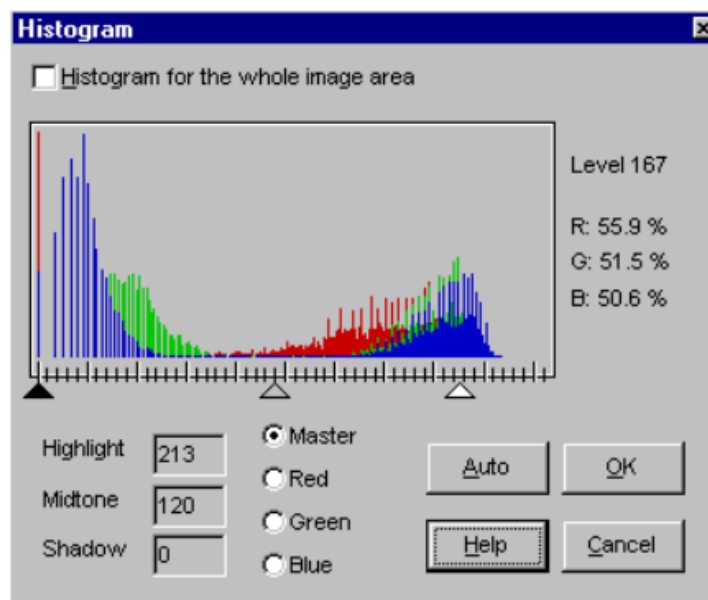


Figura 1.5: Exemplu de histogramă[6]

1.3 Algoritmi de procesare a imaginilor

Pentru a realiza transformări asupra imaginilor, toate tehnicile de procesare se bazează pe metode de calcul matematice. În acest fel, toate tipurile de modificări se rezumă, din punct de vedere matematic, la operații simple ce pot fi realizate cu ușurință de calculator.

1.3.1 Convoluție

Convoluția reprezintă o operație ce descrie evoluția suprapunerii dintre ariile a 2 funcții, atunci când una dintre ele este deplasată pe axa timpului. Aceasta este definită după formula:

$$f(t) * g(t) := \int_{-\infty}^{+\infty} f(\tau)g(t - \tau)d\tau = \int_{-\infty}^{+\infty} f(t - \tau)g(\tau)d\tau;$$

În cazul procesării digitale a semnalelor discrete și implicit a imaginilor, convoluția are expresia:

$$f[n] * g[n] := \sum_{m=-\infty}^{+\infty} f[n]g[n - m] = \sum_{m=-\infty}^{+\infty} f[n - m]g[m];$$

În formulele de mai sus, se poate observa că operația de convoluție este comutativă. Operația de adunare este mult mai ușor de realizat de către un calculator decât calcularea integralei în sine.

1.3.2 Gaussian Blur

Gaussian Blur este adesea numită curbă de clopot, deoarece graficul densității sale de probabilitate arată ca un clopot și se presupune că în timpul oricărei măsurători valorile vor urma o distribuție normală cu un număr egal de măsurători deasupra și sub valoarea medie. Aceasta se obține convoluționând o funcție oarecare, cu o funcție Gaussiană de forma:

$$g(x) = a * \exp\left(-\frac{(x - b)^2}{2 * c^2}\right)$$

- a= înălțimea vârfului graficului
- b=deplasarea centrului față de origine
- c= deviația standard și influențează lățimea graficului

Pentru coeficienții a, b, c sunt valori reale constante și c diferit de 0, această funcție este denumită după renumitul matematician Carl Friedrich Gauss și mai este numită „Clopotul lui Gauss” datorită formei pe care o capătă. Dar convoluția dintre o funcție Gaussiană și imaginea originală, limitează schimbările rapide în intensitățile pixelilor și netezește marginile.

1.4 Tehnici de procesare a imaginilor

1.4.1 Clasificare

Segmentarea

Segmentarea unei imagini în procesarea digitală a imaginii este procesul prin care imaginea este împărțită în unități semnificative. Această tehnică este folosită pentru a obține o reprezentare mai compactă, pentru a extrage obiecte sau ca instrument pentru analiza imaginilor. Ea permite partiționarea imaginilor digitale în seturi de pixeli. Scopul acestui proces este de a simplifica sau de a schimba reprezentarea imaginilor, pentru a putea fi mult mai ușor de analizat. În acest mod, pixelii cu caracteristici similare aparțin aceluiași obiect.

Cea mai simplă metodă de segmentare este binarizarea. Cel mai simplu caz, spre exemplu, este o imagine ce conține un singur obiect sau mai multe individuale care sunt pe un fundal de intensitate diferită comparativ cu cea a obiectelor. Deci, în acest caz, separația dintre obiecte și fundal se realizează prin binarizare.



Fig. 1.6 :Separea obiectelor de fundal prin binarizare [7]

Thresholding (Metoda pragului)

Metoda pragului, numită și „Thresholding”, este o altă metodă de segmentare. În această metodă, porțiunile din imagine sunt clasificate în funcție de o plajă de valori. Deci, se înlocuiește fiecare pixel dintr-o imagine cu un pixel negru dacă intensitatea imaginii este mai mică decât o valoare fixă numită prag, sau un pixel alb dacă intensitatea pixelului este mai mare decât acel prag. Spre exemplu, în imaginea de mai jos se poate observa că după acest proces copacul întunecat devine complet negru, iar zăpada stralucitoare devine complet albă.



Aplicarea pe o imagine color are rezultatul identic. Un prag mare aplicat asupra canalului culorii roșu și un prag mic aplicat pe celelalte canale va determina estomparea totală a obiectelor de culoare roșie din imaginea rezultată.

Segmentare locală

Pe de altă parte, o altă metodă folosită în numeroșii algoritmi de procesare a imaginilor constă în segmentarea locală. Comparativ cu segmentarea generală, această metodă procesează grupuri separate de pixeli, în cele din urmă, fiind asociați pentru a putea recunoaște conturul și caracteristicile unui obiect.

Un pas important, în segmentarea locală, îl constituie reducerea zgomotului. După acest proces, se dorește ca imaginea să păstreze structura obiectelor, dar să se reducă cât mai mult nivelul de perturbații introduse de limitările senzorului sau a rezoluției circuitelor de cuantizare.

Regiunile locale rezultate după procesul de fragmentare conțin un număr mic de pixeli, iar probabilitatea ca un pixel să facă parte din 2 grupuri distincte este foarte mică. Această metodă de a considera grupuri locale de pixeli, apoi de a le reduce zgomotul, constituie un pas important ce susține operațiile de detectare ale marginilor, clasificarea pixelilor și de compresie a imaginilor.

1.4.2 Transformare

Transformarea de imagine implică convertirea imaginii dintr-un domeniu în celălalt. Vizualizarea unei imagini în domenii ca de pildă spațiul Hough sau frecvența, permite identificarea caracteristicilor ce ar putea să nu fie la fel de ușor de detectat în domeniul spațial.

Cele mai întâlnite transformări folosite includ:

Transformarea Hough este o tehnică de extracție a caracteristicilor utilizată în analiza imaginilor, viziunea computerizată și procesarea digitală a imaginilor. Scopul acestei tehnici este de a găsi instanțe imperfecte ale obiectelor dintr-o anumită clasă de forme printr-o procedură de vot.

Deoarece necesită ca caracteristicile dorite să fie specificate într-o formă parametrică, transformarea Hough clasică este utilizată în mod frecvent pentru detectarea curbilor regulate,

cum ar fi linii, cercuri, elipse etc. O transformare Hough generalizată poate fi folosită în aplicații în care o simplă descrierea analitică a unei caracteristici nu este posibilă. Datorită complexității computaționale a algoritmului Hough generalizat, limităm accentul principal al acestei discuții la transformarea Hough clasică.

În ciuda restricțiilor sale de domeniu, transformarea clasică păstrează multe aplicații, deoarece majoritatea pieselor fabricate (și multe părți anatomice examinate în imagini medicale) conțin granițe de caracteristici care pot fi descrise prin curbe regulate. Principalul avantaj al tehnicii de transformare Hough este că are toleranță la golurile în descrierile limitelor caracteristicilor și este relativ neafectată de zgomotul imaginii.



Fig 1.7 Detecția cercurilor folosind metoda transformării Hough [8]

În matematică, **transformata Radon** este transformarea integrală care duce o funcție f definită în plan la o funcție Rf definită pe spațiul (bidimensional) al liniilor din plan, a cărei valoare la o anumită dreaptă este egală cu integrala dreaptă a funcției peste acea linie. Transformarea Radon, al cărui nume se datorează renumitului matematician Johann Radon, a fost introdusă în 1917 și a oferit, pe de altă parte, o formulă pentru transformarea inversă. Analogul complex al transformării Radon mai este cunoscut sub numele de transformarea Penrose.

În domeniul medical, cum ar fi tomografia computerizată, scopul acestei transformate este de a obține imaginea unei secțiuni a obiectului din proiecții obținute cu ajutorul unor radiații

penetrate. Deci, pentru fiecare linie se obține o proiecție unidimensională a unei secțiuni a obiectului.

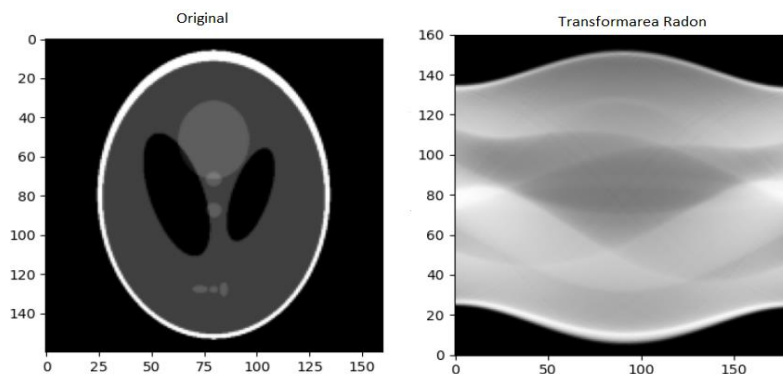


Fig 1.8: Transformata Radon aplicată [9]

1.4.3 Filtrul median

Clasa operațiilor de îmbunătățire cuprinde filtrarea imaginilor, principalul scop fiind eliminarea zgomotului suprapus unei imagini. Prin filtrare se înțelege că la calculul noii valori a unui pixel vor contribui și valorile pixelilor învecinați. Această mulțime de pixeli (vecinătate) poate avea diverse forme și dimensiuni, dar cele mai des întâlnite sunt vecinătățile de formă pătrată cu dimensiuni impare. Filtrul median este o tehnică de filtrare utilizată pentru eliminarea zgomotului din imagini și semnale.

Filtrul median este foarte important în domeniul prelucrării imaginilor, deoarece este cunoscut pentru păstrarea marginilor în timpul eliminării zgomotului.

Filtrul median este potrivit pentru eliminarea zgomotului de tip „sare și piper”. După ordonarea valorilor pixelilor, valorile zgomotului (adică 0 sau 255) se vor situa pe primele sau ultimele poziții, iar la ieșirea filtrului vom avea o valoare diferită de valorile zgomotului.

Există situații în care, după filtrare, după filtrare există pixeli afectați de zgomot. În acest caz, filtrul a fost „străpuns” de zgomot. Este posibil doar dacă atunci când mai mult de jumătate din pixelii selectați de fereastra de filtrare sunt afectați în același mod de zgomot (sare sau piper, 255 sau 0).

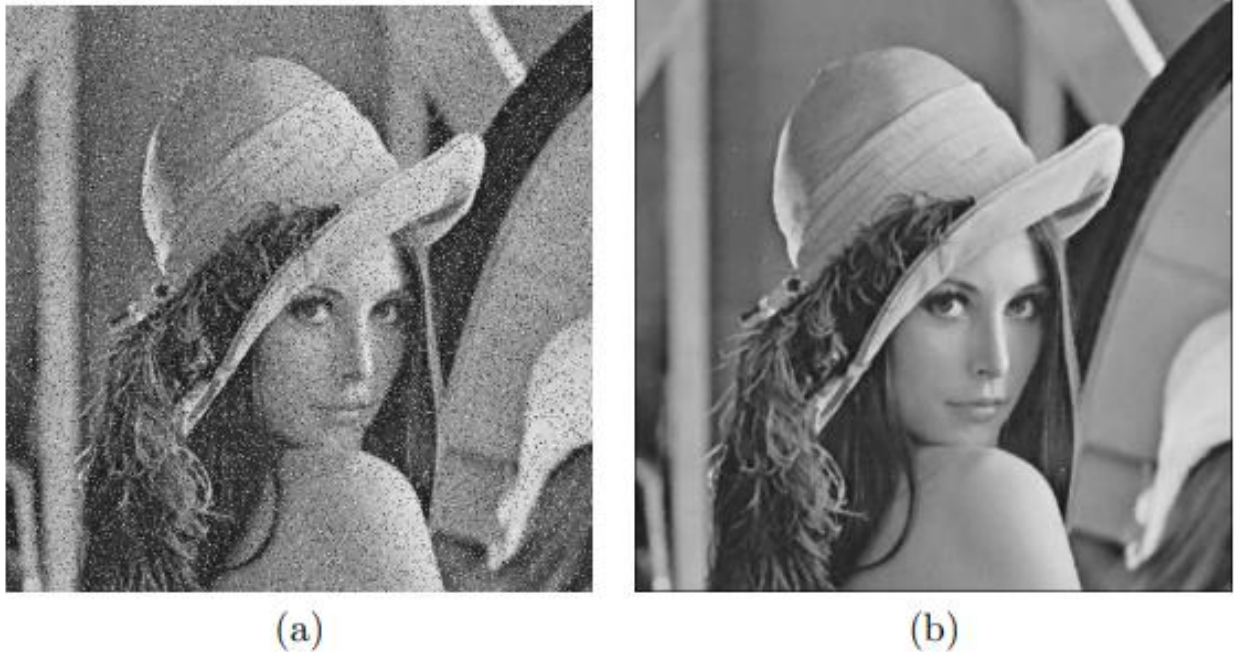


Fig 1.9: Reprezentare a filtrului median: a. Imaginea originală afectată de zgomot „sare și piper”; [10]

b. imaginea filtrată

1.5 Identificarea trăsăturilor

Recunoașterea modelelor în procesarea imaginilor are la bază extragerea caracteristicilor pe baza cărora un set inițial de date măsurate și construirea unor valori rezultate din aceste trasaturi sa fie informative, pentru sprijinirea pașilor ulteriori de identificare.

Extragerea caracteristicilor este bazată pe reducerea substanțială a dimensionalității informațiilor. Dacă datele de intrare dintr-un algoritm de calcul sunt prea mari, iar timpul de procesare este remarcabil și se presupune că sunt informații redundante, acestea pot fi transformate într-un set redus de caracteristici (vector de caracteristici).

Constituieră unei submulțimi ale caracteristicilor incipiente se numește extragere de trăsături. Caracteristicile ce au fost selectate cuprind informațiile principale din datele de intrare, pentru ca sarcina dorită să poată fi executată, folosind această reprezentare redusă în schimbul datelor inițiale.

Analizarea unei cantități considerabile de informație presupune, în regulă, o cantitate mai mare de memorie și putere de calcul. Extragerea caracteristicilor este un termen folosit pentru metodele de elaborare a vectorilor de caracteristici pentru soluționarea acestor probleme, caracterizând în același timp datele cu o acuratețe suficientă.

1.5.1 Detectarea obiectelor

Datorită relației strânse dintre detectarea obiectelor și analiza video și înțelegerea imaginilor, aceasta a atras multă atenție cercetării în ultimii ani. Detectarea obiectelor este o tehnică de viziune computerizată pentru localizarea obiectelor în imagini sau videoclipuri. Algoritmii de detectare a obiectelor utilizează de obicei învățarea automată sau învățarea profundă pentru a produce rezultate semnificative.

Aceasta aplică clasificarea obiectelor distincte și utilizează cutii de delimitare, după cum se arată mai jos.

Detectarea obiectelor este complet interconectată cu alte tehnici similare de viziune pe calculator, cum ar fi segmentarea și recunoașterea imaginilor, care ne ajută să înțelegem și să analizăm scenele din videoclipuri și imagini.

1.5.2 Caracteristicile Haar

Recunoașterea feței înseamnă identificarea persoanei, iar verificarea feței înseamnă verificarea persoanei care se pretinde a fi (prin compararea acesteia într-o anumită bază de date). Aplicațiile sale se regăsesc în diverse domenii, cum ar fi școlile, colegiile, organizațiile, fabricile, locurile publice, supravegherea etc. Aceste tehnici câștigă tot mai mult teren în întreaga lume și, extinse cu recunoașterea emoțiilor umane, aplicațiile sale sunt uriașe.

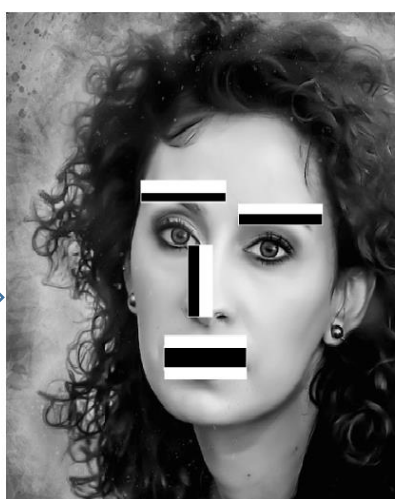
Aspectul cheie în recunoașterea feței este detectarea caracteristicilor relevante ale feței umane, cum ar fi ochii, sprâncenele, nasul, buzele.

În 2001, Paul Viola și Michael Jones au susținut ideea de a utiliza undele Haar și au dezvoltat așa numitele caracteristici de tip Haar. O caracteristică de tip Haar ia în considerare regiunile dreptunghiulare adiacente într-o anumită locație dintr-o fereastră de detecție, însumează intensitățile pixelilor din fiecare regiune și calculează diferența dintre aceste sume. Această diferență este apoi utilizată pentru a clasifica subsecțiunile unei imagini.

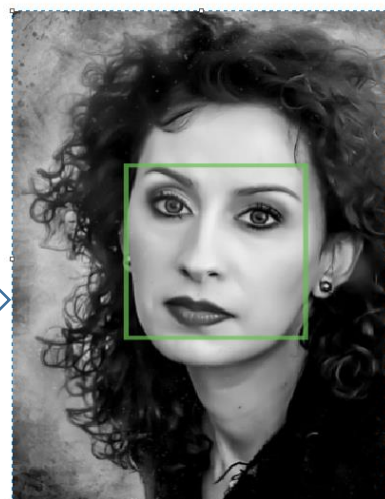
Spre exemplu, în cazul unei fețe umane, se poate observa faptul că, printre toate fețele, regiunea ochilor este mai întunecată decât regiunea obrazilor. Deci, o caracteristică Haar comună pentru detectarea feței este un set de două dreptunghiuri adiacente care se află deasupra regiunii ochilor și a obrazilor. Poziția acestor dreptunghiuri este definită în raport cu o fereastră de detecție care acționează ca un chenar de delimitare a obiectului țintă (în acest caz, fața).



Imaginea 1) : imaginea ce urmeaza a fi prelucrata



Imaginea 2) caracteristicile Haar aplicate pe imaginea 1)



Imaginea 3): fața detectata din imaginea 1)

Pentru a detecta sprâncenele, folosim caracteristica Haar (imaginea (1)), deoarece fruntea și sprâncenele formează pixeli mai ușori - pixeli mai întunecați ca în imagine. În mod similar, pentru a detecta buzele, vom utiliza o caracteristică similară cu Haar (imaginea (3)) cu pixeli mai ușori - mai întunecați - mai luminoși. Pentru a detecta nasul, am putea folosi caracteristica similară Harr mai închisă-ușoară din (imaginea (1)).

1.6 Operații morfologice pe imagini binare

Operațiile morfologice pe imagini afectează forma sau structura obiectelor. Ele se aplică doar pe imagini binare (imaginile de tip alb-negru). Aceste operații morfologice se folosesc ca etape de pre-procesare sau post-procesare a imaginilor (filtrare sau eliminarea protuberanțelor) sau pentru obținerea unei descrieri a formei obiectelor sau a regiunilor (contururi). Operațiile morfologice principale sunt dilatarea și eroziunea. Operațiile de acest tip pot fi adaptate prin selectarea elementului folosit, care determină modul în care vor fi dilatate sau erodate obiectele. Dilatarea mărește obiectele și permite umplerea unor mici goluri, pe când eroziunea micșorează obiectele prin erodarea marginilor obiectelor.

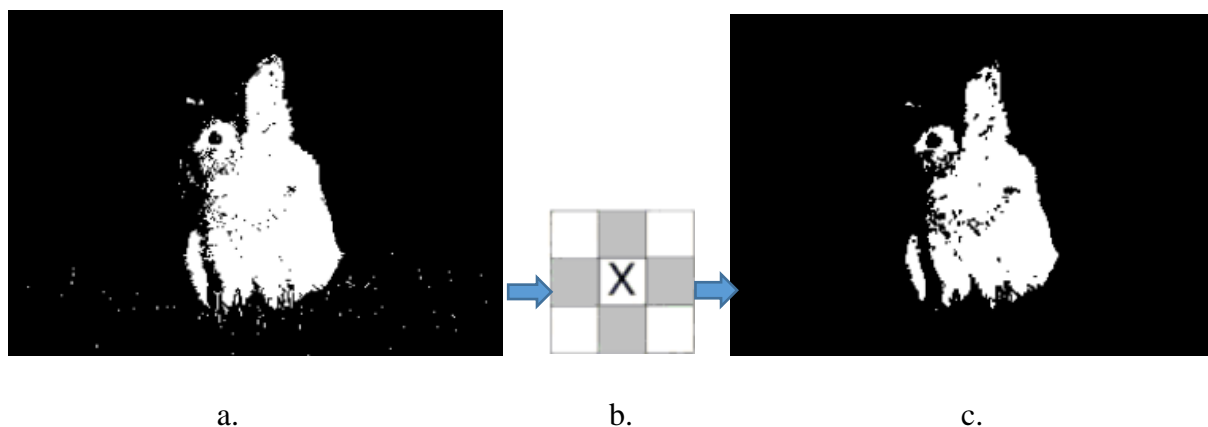
Pixelii „obiect” sunt mulțimea pixelilor asupra cărora se aplică operațiile morfologice, pe când pixelii fundal sunt complementul mulțimii pixelilor „obiect”.

1.6.1 Eroziunea

Operația de eroziune este asemănătoare cu dilatarea, dar anumiți pixeli vor fi transformați în pixeli „fundal”, invers decât la dilatare. Imaginea fiind binară, fiecare pixel este înlocuit ori cu 0 pentru negru, ori cu 1 pentru alb.

Dacă elementul structural pus peste un anumit pixel atinge fundalul, adică un punct din intersecție este negru, pixelul curent este trimis în fundal. Elementul structural este deplasat peste imagine, deci se vor folosi următorii pași:

1. Dacă originea elementului structural se suprapune peste un pixel „fundal” din imagine, atunci nu se realizează nicio modificare și se trece la următorul pixel.
2. Dacă originea elementului structural se suprapune peste un pixel „obiect” din imagine și există cel puțin un pixel „obiect” al elementului structural care se suprapune peste un pixel „fundal” din imagine, atunci pixelul curent din imagine va fi transformat în „fundal”.

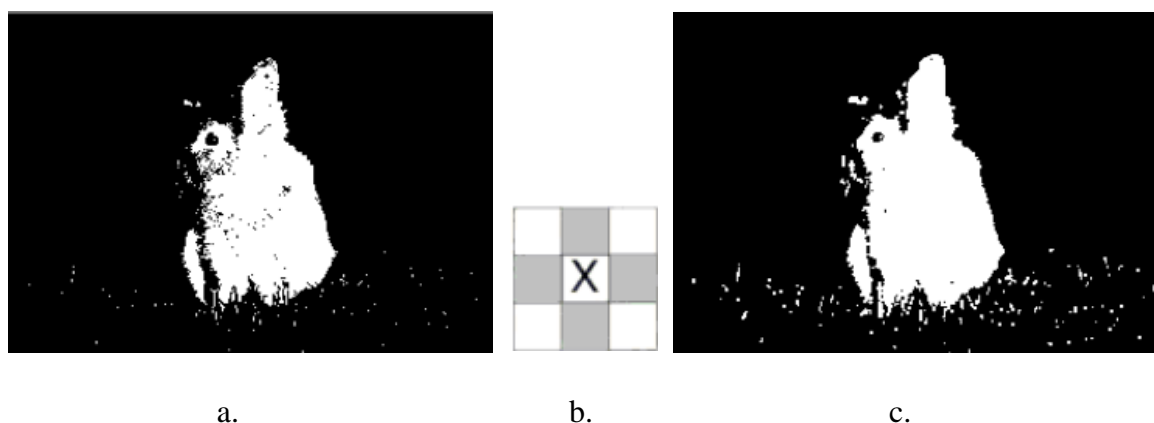


Exemplu de eroziune: a. Imaginea originală; b. Element structural; c. Imaginea rezultată

1.6.2 Dilatarea

Dilatarea este complementarul eroziunii, adică înlocuiește pixelul curent cu valoarea maxima din mulțimea de pixeli intersectată de obiectul structural. Dacă obiectul structural atinge prim planul imaginii, atunci pixelul curent devin alb. În acest fel, prim planul este mai mare decât imaginea inițială. Dilatarea este definită prin următoarea secvență de pași:

1. Dacă originea elementului structural se suprapune cu un pixel „fundal” din imagine, nu se efectuează nicio modificare și se trece mai departe la următorul pixel.
2. Dacă originea elementului structural coincide cu un pixel „obiect” din imagine, atunci toți pixelii acoperiți de elementul structural devin pixeli „obiect”.



Exemplu de dilatare: a. Imaginea originală; b. Element structural; c. Imaginea rezultată

1.6.3 Deschiderea (O-opening) si închiderea (C-closing)

Cu ajutorul dilatării și eroziunii se pot construi operații de ordin mai mare: deschiderea (O-opening) și închiderea (C-closing). Aceste tehnici pot fi utilizate și pentru a găsi forme specifice.

Deschiderea (O-opening) îndepărtează obiectele mici din prim-planul unei imagini (spre exemplu pixelii luminoși), plasându-le în fundal. Deschiderea este un proces în care se efectuează mai întâi o operație de eroziune și apoi o operație de dilatare. În general, acest proces netezește mai bine conturul unui obiect și elimină proeminențele subțiri.

Închiderea (C-closing) elimină mici găuri în prim-plan, cu alte cuvinte elimină pixelii de fundal care se potrivesc elementului de structurare. Aceasta tinde sa netezească secțiuni ale conturilor, dar comparativ cu deschiderea, contopește rupturile înguste și golurile lungi și subțiri, elimină găurile mici și umple golurile din contur. Închiderea este un proces în care se efectuează mai întâi o operație de dilatare, apoi o operație de eroziune.

2. Capitolul II: Mediu de dezvoltare

Pentru a dezvolta partea practică a acestei lucrări de licență, mediul de dezvoltare folosit este MATLAB®. Prin intermediul acestuia, aplicația a fost dezvoltată în limbajul de programare MATLAB.

MATLAB®, cu acronimul de la „Matrix Laboratory”, este o platformă de programare concepută special pentru ingineri și oameni de știință pentru a analiza și proiecta sisteme și produse care ne transformă lumea. Nucleul aplicației MATLAB este limbajul MATLAB, un limbaj bazat pe matrice care permite cea mai naturală exprimare a matematicii computaționale.



Acest program include un editor de cod, cu suport pentru completarea inteligentă a codului.

Ce putem face cu aplicația MATLAB:

- Analiza datelor
- Dezvoltarea de algoritmi
- Crearea de modele și aplicații

MATLAB permite ducerea ideilor de la cercetare la producție prin implementarea în aplicații de întreprindere și dispozitive încorporate, precum și prin integrarea cu Simulink® și Model-Based Design.

MATLAB a evoluat de-a lungul unei perioade de ani, cu aportul a numeroși utilizatori. În mediile universitare, acesta este instrumentul standard de instruire pentru cursurile

introdutive și avansate de matematică, inginerie și științe. În industrie, MATLAB este instrumentul preferat pentru cercetare, dezvoltare și analiză de înaltă productivitate.

2.1 Limbajul de programare MATLAB

MATLAB este un limbaj de înaltă performanță pentru calcul tehnic. Acesta integrează calculul, vizualizarea și programarea într-un mediu ușor de utilizat, în care problemele și soluțiile sunt exprimate în notații matematice familiare. Printre utilizările tipice se numără:

- Matematică și calcul
- Dezvoltarea algoritmilor
- Modelare, simulare și prototipare
- Analiza, explorarea și vizualizarea datelor grafică științifică și inginerescă
- Dezvoltarea de aplicații, inclusiv crearea de interfețe grafice cu utilizatorul

MATLAB este un sistem interactiv al cărui element de date de bază este o matrice care nu necesită dimensionare. Acest lucru vă permite să rezolvați multe probleme tehnice de calcul, în special cele cu formulări matriceale și vectoriale, într-o fracțiune din timpul necesar pentru a scrie un program într-un limbaj neinteractiv scalar, cum ar fi C sau Fortran.

Numele MATLAB înseamnă laborator matricial. MATLAB a fost scris inițial pentru a oferi un acces ușor la software-ul matricial dezvoltat de proiectele LINPACK și EISPACK, care împreună reprezintă stadiul actual al software-ului pentru calculul matricial.

MATLAB dispune de o familie de soluții specifice aplicațiilor, denumite toolbox-uri. Foarte importante pentru majoritatea utilizatorilor de MATLAB, toolbox-urile vă permit să învățați și să aplicați tehnologii specializate. Toolbox-urile sunt colecții complete de funcții MATLAB (fișiere M) care extind mediul MATLAB pentru a rezolva anumite clase de probleme. Domeniile în care sunt disponibile toolbox-uri includ procesarea semnalelor, sisteme de control, rețele neuronale, logică fuzzy, wavelets, simulare și multe altele.

2.2 Limbaje de programare similare cu MATLAB

Matlab este un limbaj de programare care suportă mai multe paradigme de programare. Acesta este un mediu de calcul numeric dezvoltat de MathWorks. Acesta prezintă un limbaj de înaltă performanță și de uz general, dar acest limbaj este strâns integrat cu Java - interpretorul Matlab este scris în Java.

Simulările în C++ arată performanțe superioare, de 100 de ori mai bune în ceea ce privește complexitatea timpului decât o implementare MATLAB echivalentă. De cele mai multe ori, codul în C++ este în mare parte serial și nu se face nicio optimizare de înaltă fidelitate în mod explicit.

C++ are o viteză medie de procesare de peste 500 de ori mai mare decât codul Matlab. Acest lucru nu este valabil numai pentru acest cod, ci poate fi aplicat și pentru orice altă comparație de cod între fișierele MEX Matlab și C++. Prin comparație, avantajele vitezei oferite de C++ depășesc cu mult simplitatea lui MATLAB.

Python este un limbaj de nivel înalt, este mai ușor de utilizat, mai ușor de citit și mai portabil. MATLAB are funcția de matrice, iar Python poate folosi NumPy, iar biblioteca poate obține rezultate similare. MATLAB este mai rapid decât Python, dar Python este mai bun la rularea mai multor sarcini în paralel.

Este posibil ca MATLAB să ruleze lent deoarece are o cantitate limitată de memorie RAM (sub 128 MB). Memoria RAM utilizată de MATLAB în timpul execuției este cuprinsă între 40 MB-60 MB. Browserul HELP poate ocupa încă 12MB. În cazul în care, memorie este limitată (RAM), este posibil ca procesorul să înceapă să utilizeze memoria virtuală (de pe hard disk).

3. Capitolul III: Implementarea practică

Partea practică a acestei lucrări constă într-o aplicație dezvoltată în limbajul de programare MATLAB, în mediul de dezvoltare furnizat de aplicația cu același nume, adică MATLAB. Aplicația oferă mai multe meniuri de selecție prin intermediul unei interfețe text ce facilitează navigarea printre opțiunile furnizate și pentru a eficientiza experiența de utilizare, funcții ce rulează sub fire de execuție diferite pentru diverse funcții consumatoare de resurse.

Aplicația folosește, în mod direct, funcții din librăria MATLAB pentru a facilita accesarea, transferul și editarea pozelor, iar pentru ca utilizatorul să fie mereu în contact cu aplicația, acesta primește un răspuns în timp real cu privire la modificările realizate, fie că este sub formă de mesaj, atunci când a intervenit o eroare, fie că este sub forma unei ferestre în care este afișată imaginea rezultată.

3.1 Descrierea aplicației

Aplicația de procesare a imaginii oferă funcții de detectare a irisului alături de diverse efecte și manipulări ale unei imagini furnizată de către utilizator.

Toate aceste funcții sunt implementate în fire de execuție, pentru ca interacțiunea utilizatorului cu aplicația să fie lipsită de întârzieri, iar meniurile de navigare sunt simple și ușor de utilizat.

Pentru a detecta irisul vom folosi o imagine stocată pe dispozitiv și vom aplica diferite filtre pentru a transforma sau elimina părți din aceasta cum ar fi: încadrarea zonei ochilor și decuparea acesteia, transformarea imaginii originale într-o imagine grayscale, binarizarea imaginii. Toate aceste procese vor fi aplicate pe imaginea originală până se va ajunge la rezultatul final, unde imaginea color va avea irisul încadrat și un mesaj sugestiv.

3.2 Structura aplicației/organigramă

Aplicația are următoarea structură funcțională:

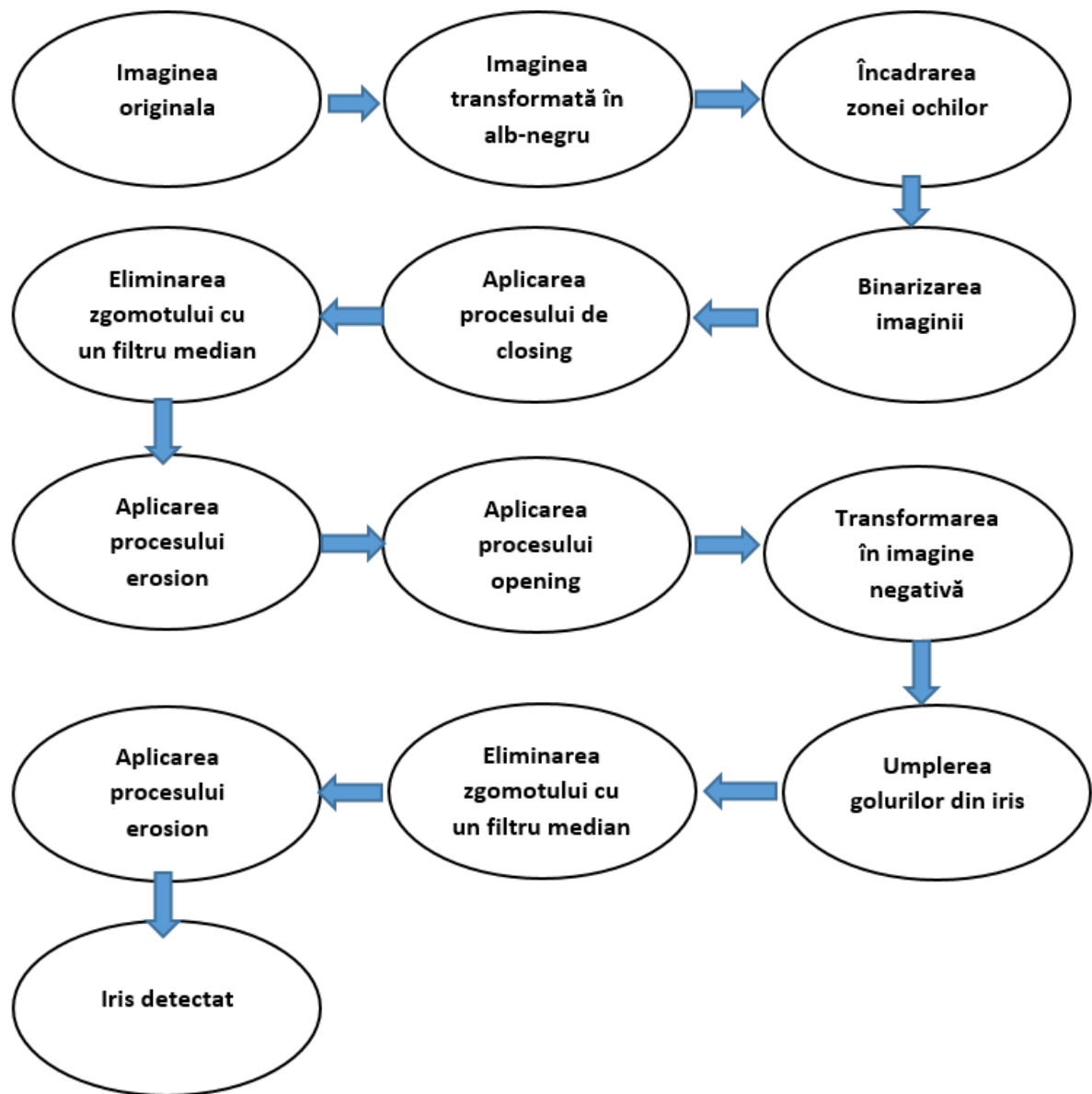


Figura 3.01:Schema bloc a aplicației MATLAB

3.3 Implementarea programului

În cadrul etapei de detecție a irisului vom executa următoarii pași:

Citirea imaginii de pe dispozitiv

Vom citi imaginea stocată pe dispozitiv cu ajutorul funcției `imread()`.

```
clc
clear all
close all

% EyeDetect este o variabila care va avea valoarea dată de return-ul
% funcției apelate cu parametrul 'EyePairBig'
EyeDetect=vision.CascadeObjectDetector('EyePairBig');
%citirea imaginii stocate pe dispozitiv
I=imread('D:\licenta\eye3.jpg');
```

Imaginea originala



Funcția `imread()` citește imaginea din fișierul local specificat prin calea oferită, deducând formatul fișierului din conținutul acestuia. Dacă numele fișierului este un fișier cu mai multe imagini, atunci `imread` citește prima imagine din fișier.

Convertirea imaginii în grayscale (alb-negru)

Imaginea originală de pe dispozitiv este transformată în imagine alb-negru cu ajutorul funcției `rgb2gray()`, iar zona ochilor este încadrată. Funcția `rgb2gray` convertește imaginile RGB în tonuri de gri prin eliminarea informațiilor de nuanță și saturație, păstrând în același timp luminanța.

```
%In variabila igray se stocheaza imaginea data de utilizator in format  
%alb-negru  
igray=rgb2gray(I);  
%In variabila BB se stocheaza zona ochilor  
BB=step(EyeDetect,igray);
```

Imaginea transformata in alb negru



Decuparea imaginii

După ce zona ochilor a fost detectată și încadrată într-o fereastră, ea este decupată cu ajutorul funcției `imcrop()`. Această funcție afișează imaginea indexată în parametrii specificați și creează un instrument de decupare asociat cu imaginea respectivă.

```
%In variabila Eyes se stocheaza imaginea decupata  
Eyes=imcrop(igray,BB);
```

Imaginea decupata



Binarizarea imaginii

Imaginea decupată trece prin procesul de binarizare cu funcția `imbinarize()`. Această funcție creează o imagine binară din imaginea în tonuri de gri 2-D sau 3-D I prin înlocuirea tuturor valorilor care depășesc un prag determinat la nivel global cu valori de o secundă și prin stabilirea tuturor celorlalte valori la 0.

```
% Binarizarea imaginii
imshow(Eyes);
title("Ochii");
BW = imbinarize(Eyes,0.4);
figure
imshow(BW);
title("Binarizarea imaginii (alb-negru)");
imwrite(BW, "BW.png");
```

Binarizarea imaginii decupate



Aplicarea procesului de closing

Pe imaginea binarizată se aplică procesul de closing cu funcția `imclose()`. Aceasta realizează închiderea morfologică a imaginii în scală de gri sau binară BW, utilizând un element de structurare tip disc de dimensiune 2. Operația de închidere morfologică (closing) este o dilatare urmată de o eroziune, folosind același element structurant pentru ambele operații.

```
% Functia strel cu parametru disk va crea o figura in forma de disk cu raza
% egala cu doi
se = strel('disk',2);
closeBW = imclose(BW,se);
figure
imshow(closeBW);
title("Aplicare proces de closing");
imwrite(closeBW, "closeBW.png");
```

Aplicarea proces closing



Eliminarea zgomotului

Se aplică un filtru median peste imagine pentru a reduce zgomotul cu funcția `medfilt2()`. Ea face filtrarea mediană a imaginii `closeBW` în două dimensiuni. Fiecare pixel de ieșire conține valoarea mediană într-o vecinătate de 3×3 în jurul pixelului corespunzător din imaginea de intrare.

```
% Se aplica un filtru median pentru imagine pentru a reduce zgomotul
filter1 = medfilt2(closeBW);
figure
imshow(filter1);
title("Eliminarea zgomotului");
imwrite(filter1,"filter1.png");
```

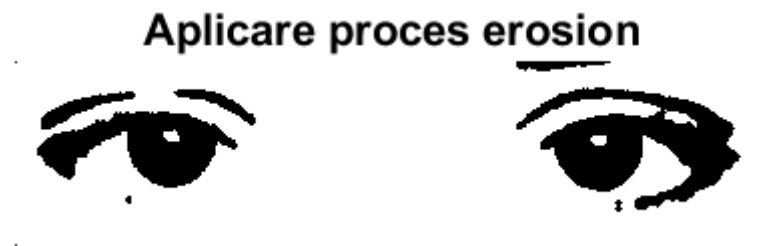
Eliminarea zgomotului



Aplicarea procesului erozion

După ce zgomotul a fost eliminat, peste imaginea rezultată este aplicat procesul de eroziune cu funcția `imerode()`. Funcția erodează imaginea `filter1` în tonuri de gri sau binară, folosind elementul structurant SE.

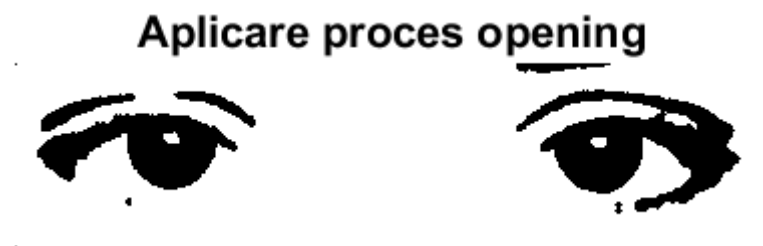
```
% creează un element structurant în formă de disc, unde 2 specifică raza
se3 = strel('disk',2);
%functia imerode erodeaza imaginea stocata dupa filtrarea acesteia
%elimina pixelii de la limitele obiectului
er1=imerode(filter1,se3);
figure
imshow(er1);
title("Aplicare proces erosion");
imwrite(er1,"er1.png");
```



Aplicarea procesului de opening

Funcția `bwareaopen()` elimină toate componentele conectate (obiecte) care au mai puțin de 30 de pixeli din imaginea binară `er1`, producând o altă imagine binară. Această operație este cunoscută sub numele de deschidere de zonă („area opening”).

```
%functia bwareaopen elimina parti din imagine care au mai putini de 30 de
%pixeli
open = bwareaopen(er1,30);
figure
imshow(open);
title("Aplicare proces opening");
imwrite(open,"Aplicare proces de opening.png");
```

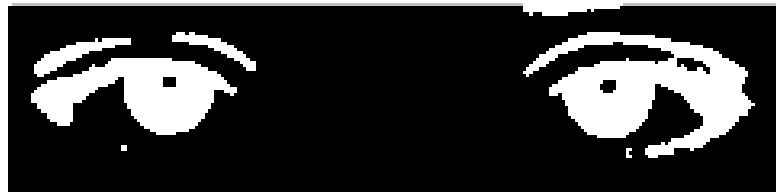


Transformarea în imagine negativă

Imaginea devine o imagine negativă cu ajutorul funcției `imcomplement()`. În cazul unei imagini în tonuri de gri, zonele întunecate devin mai deschise, iar zonele deschise devin mai întunecate.

```
%funcția imcomplement transforma în imagine negativă| variabila open  
neg = imcomplement(open);  
figure  
imshow(neg);  
title("Imagine negativa");  
imwrite(neg, "Imagine_negativa.png");
```

Imagine negativa



Umplerea golurilor

Pentru formarea completă a irisului, golurile proiectate din cauza luminii sunt eliminate cu funcția `imfill()`.

```
%umple golurile proiectate din cauza luminii din iris  
fill = imfill(neg, 'holes');  
figure  
imshow(fill);  
title("Umplerea golurilor");  
imwrite(fill, "Umplerea_golurilor.png");
```

Umplerea golurilor

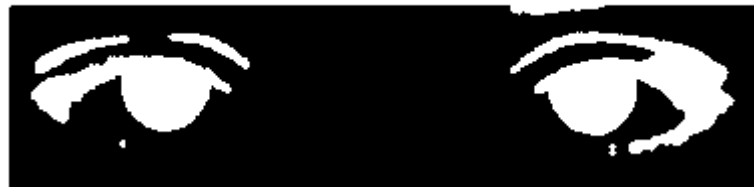


Eliminarea zgomotului

Se elimină din nou zgomotul din imaginea negativă cu funcția `medfilt2()`.

```
%Se aplica un filtru median pentru imaginea stocata in variabila fill ce ia  
%ca valoarea imaginea cu culorile inversate din variabila neg  
filter2 = medfilt2(fill);  
figure  
imshow(filter2);  
title("Eliminarea zgomotului");  
imwrite(filter2,"Eliminarea zgomotului.png");
```

Eliminarea zgomotului



Aplicarea procesului erozion

După ce zgomotul a fost eliminat, peste imaginea rezultată este aplicat procesul de eroziune cu funcția `imerode()`.

```
%Procesul erozion  
se4 = strel('disk',4);  
er2=imerode(filter2,se4);  
figure  
imshow(er2);  
title("Proces de eroziune");  
imwrite(er2,"Erosion.png");
```

Aplicare procesului erozion

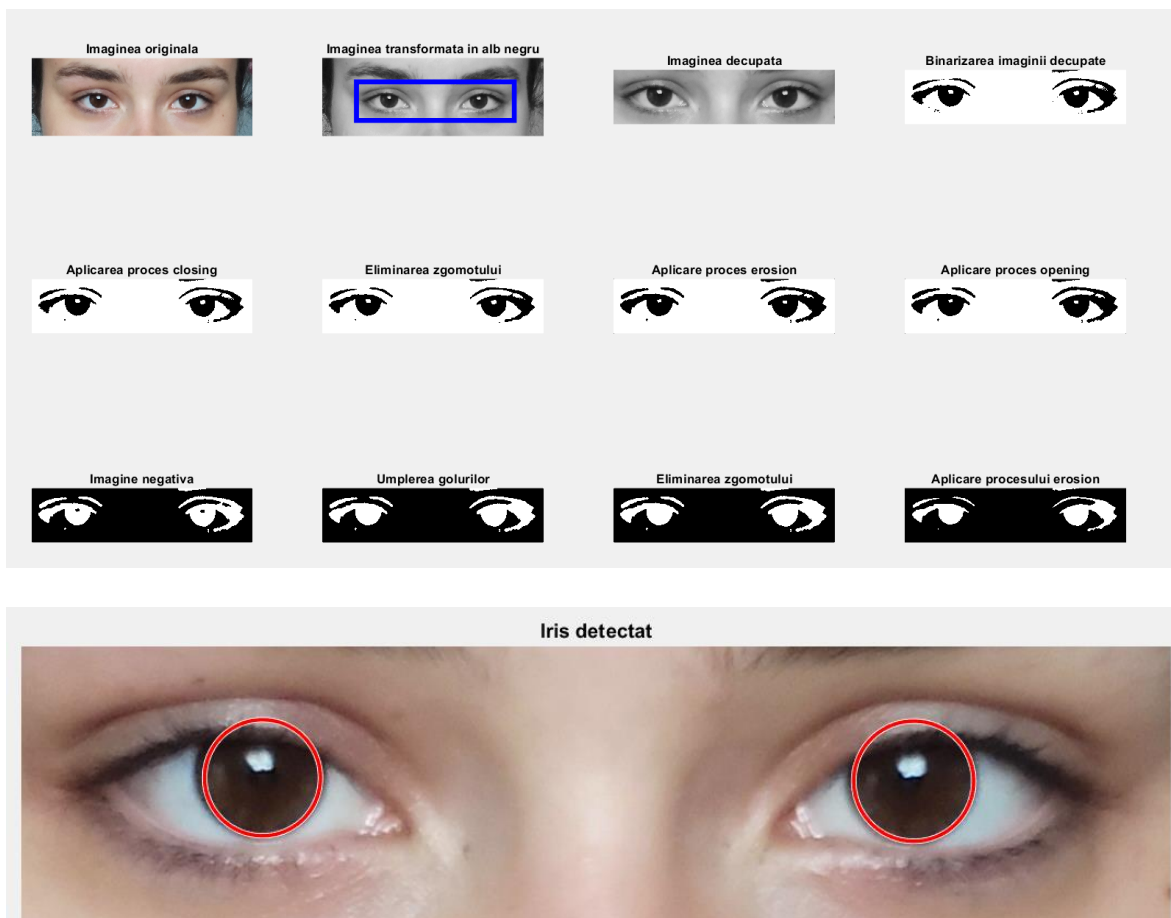


Creearea cercurilor pentru încadrarea irisului

Funcția `imfindcircles` găsește cercuri cu raze în intervalul specificat de `er2`. Argumentul de ieșire suplimentar, `radii`, conține razele estimate corespunzătoare fiecărui centru de cerc din `centers`. În cele din urmă, funcția `viscircles` desenează cercuri cu centrele și razele specificate pe axele curente.

```
%Cu functia imfindcircle se cauta in imagine irisul| care a fost incadrata  
%mai sus  
[centers,radii] = imfindcircles(er2,[30 120],'ObjectPolarity','bright', ...  
    'Sensitivity',0.92)  
  
%Detectarea pupilelor  
figure,imshow(Eyes1)  
title('Iris detectat')  
%functia viscircles creeaza un cerc cu centru si raza stocate mai sus cu  
%functia imfindcircles  
h = viscircles(centers,radii):
```

În concluzie, irisul este detectat.



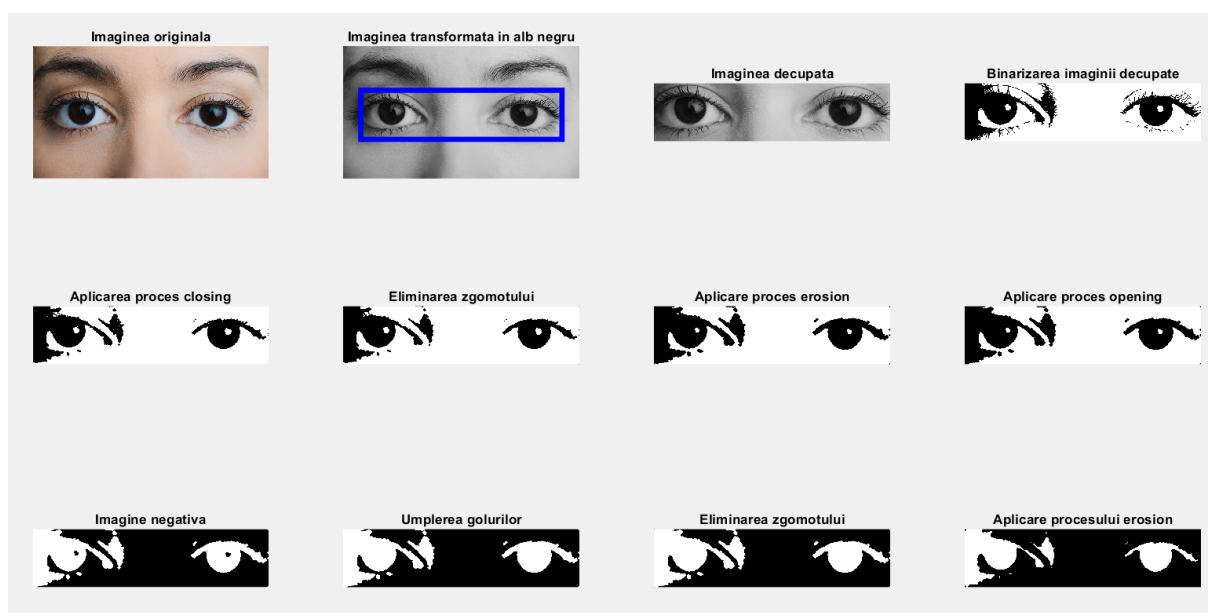
3.4 Rezultatele simulării

Pentru a observa rezultatele obținute în urma simulării am ales trei imagini diferite și am exemplificat fiecare cadru ales cu câte o imagine pentru a vedea care este rezultatul când ochii se află în poziții diferite sau au forme diferite.

În cazul algoritmului de detectare a irisului, rata de detecție este extrem de ridicată, atât timp cât ochii subiecților sunt îndreptați către cameră și nu sunt acoperiți de nimic.

Primul caz:

În cazul ochilor mari și rotunzi este de așteptat ca algoritmul să detecteze cu ușurință irisul. Imaginea este clară, conturul ochilor este vizibil în totalitatea și luminozitatea este potrivită pentru toate etapele programului.



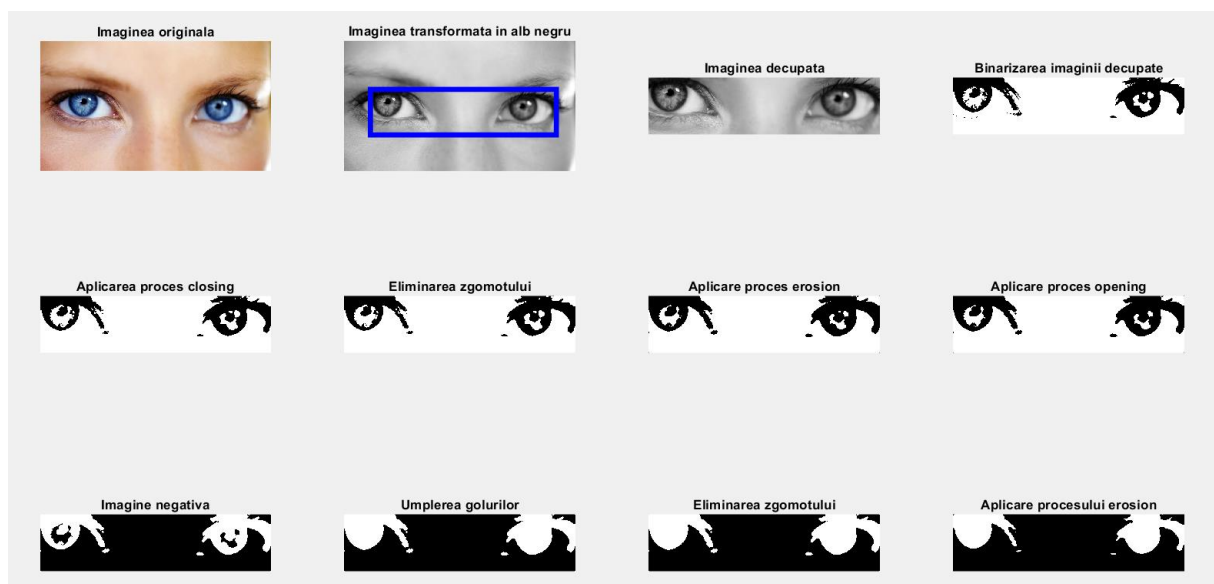
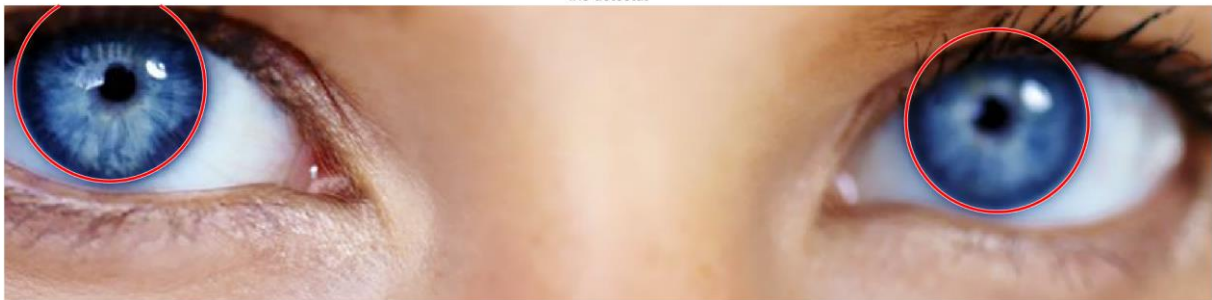
Iris detectat



Al doilea caz:

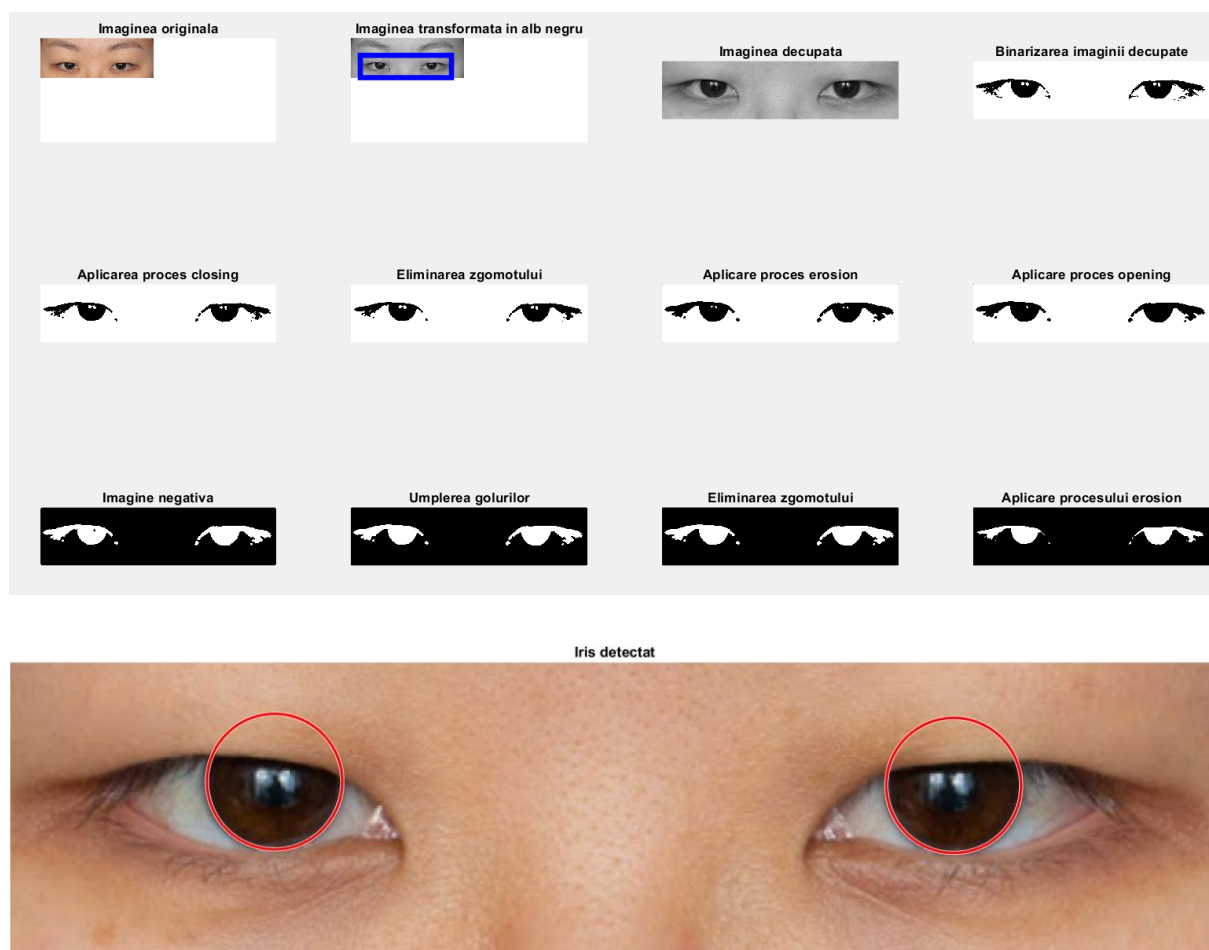
Am ales această imagine deoarece irisul este mult mai deschis comparativ cu primul caz, dar și poziția ochilor este ușor diferită ceea ce ar însemna ca ar putea îngreuna detecția.

Iris detectat



Al treilea caz:

În cazul ochilor alungiți și cu pleoapa căzută este mult mai dificil să fie realizabilă detecția irisului, dar aplicația reușește să ducă la bun sfârșit fiecare etapă.



3.5 Analiza algoritmului.Limitări

Un algoritm de detecție a irisului are unele limitări care ar putea îngreuna detecția corespunzătoare a colțurilor unui obiect. Cele mai importante probleme în irisului sunt: influența rezoluției , influența zgomotului , blurarea , luminozitatea și alte obiecte detectate. În următoarele paragrafe le vom exemplifica pe fiecare în parte pentru a putea observa efectele.

3.5.1 Blurare

Algoritmul de detectare a irisului are rezultate dependente de claritatea pozelor și de nivelul de detalii al acestora. Pentru a putea recunoaște formele cu o precizie cât mai mare, subiectul trebuie să fie în focus, iar conturul său să nu fie acoperit, deoarece acest lucru duce la o clasificare greșită.

Dacă, în unele cazuri, conturul irisului nu este complet vizibil, algoritmul are șanse mai mari să detecteze cercuri în plus din întregul document, după cum se poate observa și în imaginea de mai jos:

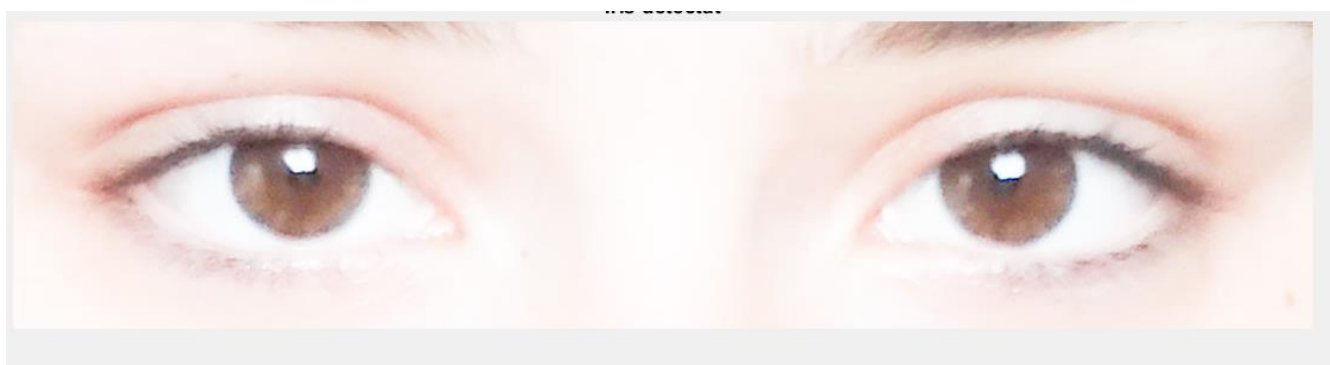
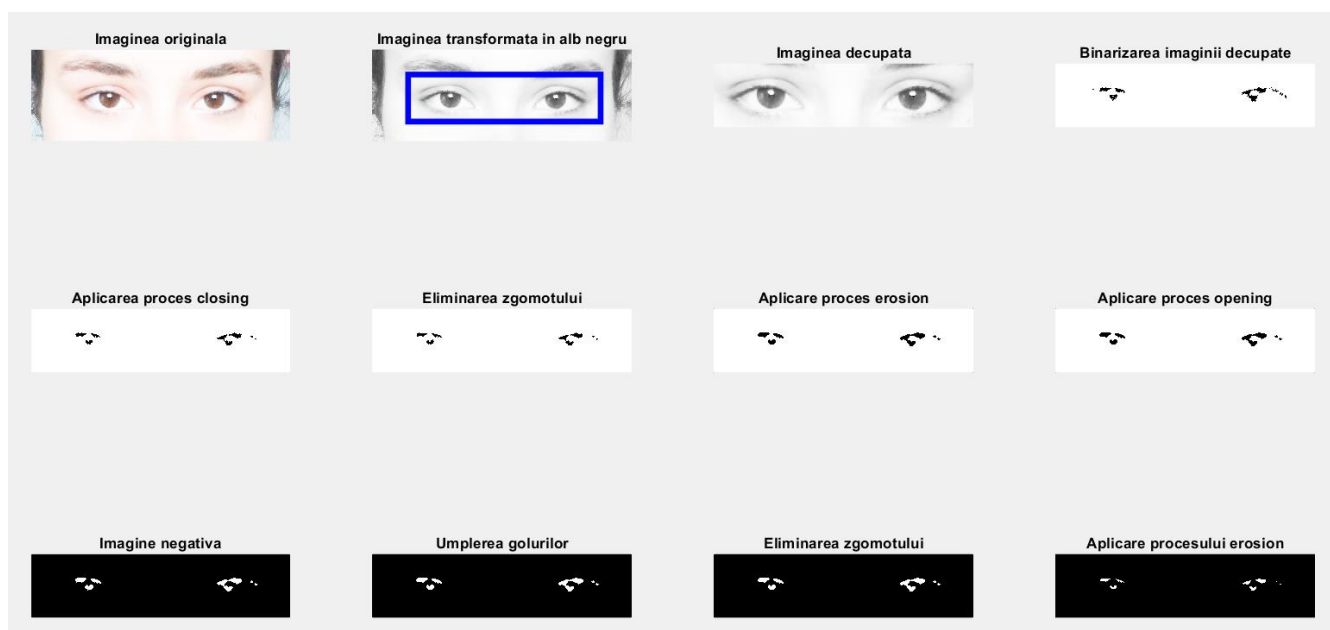


3.5.2 Luminozitate crescută

În aplicația noastră luminozitatea are un rol important și îl vom demonstra în exemplul următor.

Vom adăuga cadrelor noastre un nivel de alb de 0.9 și vom observa cum influențează încadrarea irisului în cerc. După cum se poate observa, programul execută fiecare transformare a imaginii, dar în final încadrarea irisului în cercuri nu este realizată.

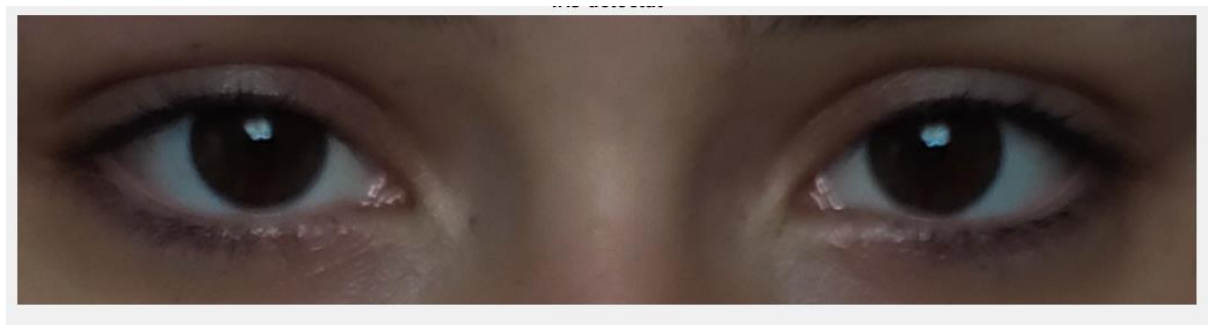
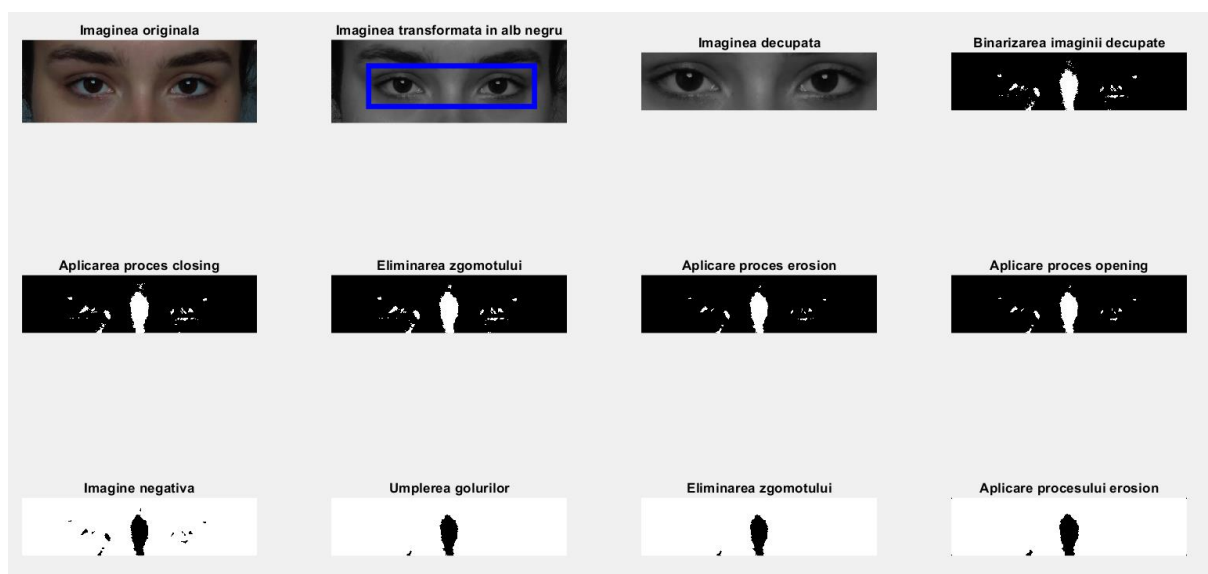
Se încadrează zona ochilor, apoi este decupată, dar în procesul de binarizare imaginea suferă mari schimbări. Scopul ei este de a segmenta imaginea în regiuni de interes și eliminarea regiunilor care nu sunt considerate esențiale. În cazul nostru peretele fiind alb și intensitatea pixelilor fiind crescută, algoritmul nu poate detecta pe deplin conturul ochilor.



Deci, luminozitatea are un rol important influențând în mod decisiv detecția irisului atunci când este aplicată asupra imaginii o valoare aproape de nivelul maxim a nivelului de alb.

3.5.3 Contrast

Creșterea contrastului mărește diferența dintre zonele luminoase și cele întunecate ale imaginii. După procesul de binarizare, separarea regiunilor din imagine nu poate fi realizată clar din cauza pragului de binarizare, de aceea unde lumina a fost mai puternică pixelii au devenit albi, iar regiunea unde lumina era scăzută pixelii au devenit negrii. Când imaginea a devenit negativă, regiunea ochilor nu a mai putut fi distinsă fiind afectat procesul de detecție a irisului.

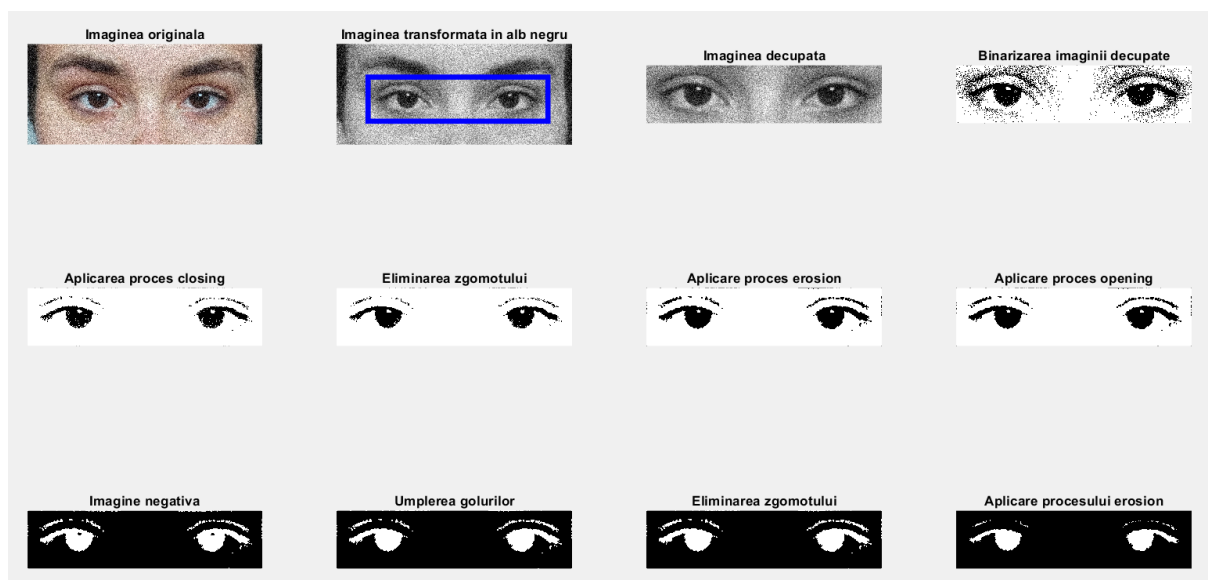


3.5.4 Influența zgomotului

Zgomotul cauzat de biții pierduți sau alterați pot cauza îngreunarea detecției irisului, și implicit încadrarea acestuia într-un cerc.

În modelul de zgomot de tip salt & pepper există doar două valori posibile, a și b. Din aceasta cauză se mai numește și zgomot de tip impuls (speckle). Probabilitatea de apariție a fiecăruia este mai mică de 0.1. La valori mai mari decât acestea, zgomotul va domina imaginea.

Acest tip de zgomot va afecta valorile pixelilor în două moduri: „salt” adică noua valoare a pixelului este 255 (pixelul este alb) sau „pepper” unde noua valoare a pixelului este 0 (pixelul este negru).



4.Capitolul IV: Concluzii

Procesarea imaginilor digitale este un domeniu vast, dar care se dezvoltă într-un ritm alert. Procesarea de imagini este des folosită în multe domenii și ne ușurează viața sau chiar ne ferește de anumite pericole cum ar fi eventualele accidente. Prinzipala utilizare a acestei aplicații o reprezintă optimizarea informației conținute de imagini în vederea analizării de către om.

În această lucrare, obiectivul a fost realizarea cu succes a detecției irisului, prin diferite procese cum ar fi: binarizarea imaginii, eliminarea zgomotului, erodarea imaginii sau procese de opening/closing. Această aplicație are avantajul că indiferent de forma ochilor, detecția nu este influențată, putând detecta cu ușurință irisul.

Ca orice altă aplicație de detecție are și unele dezavantaje, iar în această lucrare am scos în evidență câteva influențe negative asupra detecției corecte și precise a irisului, implicit încadrarea cu ajutorul cercurilor. Principalele influențe negative sunt: luminozitatea crescută, blurarea crescută a imaginii sau zgomotul ridicat.

Aceste obiective au fost, în general, atinse, programul având un flux simplu și rapid. La nivelul aplicației există un timp de așteptare, cauzat de capacitatea de actualizare în timp real a imaginii, suferind trecerea prin procesele de filtrare sau transformare.

Într-o formă mult mai complexă, aplicația are rolul de a ușura munca celor ce muncesc în sistemul medical sau în domeniul bancar, înlocuind sistemele greoaie și care necesită mult timp, bazate pe coduri PIN și parole. Lucrarea se află în formă de prototip, urmând ca, pe viitor, să se aducă îmbunătățiri la funcționarea acesteia. De pildă, captarea imaginilor se va face automat, în timp real, prin utilizarea camerei web. De asemenea, se urmărește o continuitate mai rafinată a segmentării secvenței video și procesarea sau filtrarea acestora. Se va urmări actualizarea cât mai rapidă a proceselor, imaginilor și o interfață mai prietenoasă cu utilizatorul.

Conceput pentru a funcționa drept un prototip, dispozitivul propune câteva funcții pentru ca acestea să fie îndeplinite cu succes, urmând adăugarea unora noi și actualizarea celor existente.

5. Anexă

```
clc

clear all

close all

% EyeDetect este o variabila care va avea valoarea dată de return-ul
% funcției apelate cu parametrul 'EyePairBig'
EyeDetect=vision.CascadeObjectDetector('EyePairBig');

%citirea imaginii stocate pe dispozitiv
I=imread('D:\licenta\eye3.jpg');

%In variabila igray se stocheaza imaginea data de utilizator in format
%alb-negru
igray=rgb2gray(I);

%In variabila BB se stocheaza zona ochilor
BB=step(EyeDetect,igray);

%In variabila Eyes se stocheaza imaginea decupata
Eyes=imcrop(igray,BB);

% Binarizarea imaginii
imshow(Eyes);

title("Ochii");

BW = imbinarize(Eyes,0.4);

figure

imshow(BW);

title("Binarizarea imaginii (alb-negru)");
```

```

imwrite(BW, "BW.png");

% Incadrarea zonei ochilor

% Functia strel cu parametru disk va crea o figura in forma de disk cu raza
% egala cu doi

se = strel('disk',2);

closeBW = imclose(BW,se);

figure

imshow(closeBW);

title("Aplicare proces de closing");

imwrite(closeBW,"closeBW.png");


% Se aplica un filtru median pentru imagine pentru a reduce zgomotul

filter1 = medfilt2(closeBW);

figure

imshow(filter1);

title("Eliminarea zgomotului");

imwrite(filter1,"filter1.png");


% creează un element structurant în formă de disc, unde 2 specifică raza

se3 = strel('disk',2);

%functia imerode erodeaza imaginea stocata dupa filtrarea acesteia

%elimina pixelii de la limitele obiectului

er1=imerode(filter1,se3);

figure

```

```

imshow(er1);

title("Aplicare proces erosion");

imwrite(er1,"er1.png");

%functia bwareaopen elimina parti din imagine care au mai putini de 30 de
%pixeli

open = bwareaopen(er1,30);

figure

imshow(open);

title("Aplicare proces opening");

imwrite(open,"Aplicare proces de opening.png");

%functia imcomplement transforma in imagine negativa in variabila open

neg = imcomplement(open);

figure

imshow(neg);

title("Imagine negativa");

imwrite(neg,"Imagine_negativa.png");

%umple golurile proiectate din cauza luminii din iris

fill = imfill(neg,'holes');

figure

imshow(fill);

title("Umplerea golurilor");

imwrite(fill,"Umplerea golurilor.png");

%Se aplica un filtru median pentru imagina stocata in variabila fill ce ia
%ca valoarea imaginea cu culorile inversate din variabila neg

filter2 = medfilt2(fill);

```

```

figure

imshow(filter2);

title("Eliminarea zgomotului");

imwrite(filter2,"Eliminarea zgomotului.png");

%

se4 = strel('disk',4);

er2=imerode(filter2,se4);

figure

imshow(er2);

title("Eliminarea zgomotului");

imwrite(er2,"Eliminarea zgomotului.png");


subplot(3,4,1),imshow(I)

title('Imaginea originala')

subplot(3,4,2),imshow(igray)

title('Imaginea transformata in alb negru')

rectangle('Position',BB,'LineWidth',4,'LineStyle','-','EdgeColor','b');

subplot(3,4,3),imshow(Eyes)

title('Imaginea decupata')

subplot(3,4,4),imshow(BW)

title('Binarizarea imaginii decupate')

subplot(3,4,5),imshow(closeBW)

title('Aplicarea proces closing')

subplot(3,4,6),imshow(filter1)

title('Eliminarea zgomotului')

subplot(3,4,7),imshow(er1)

```

```

title('Aplicare proces erosion')
subplot(3,4,8),imshow(open)
title('Aplicare proces opening')
subplot(3,4,9),imshow(neg)
title('Imagine negativa')
subplot(3,4,10),imshow(fill)
title('Umplerea golurilor')
subplot(3,4,11),imshow(filter2)
title('Eliminarea zgomotului')
subplot(3,4,12),imshow(er2)
title('Aplicare procesului erosion')

%In BB1 se va stoca din imaginea initiala ochii
BB1=step(EyeDetect,I);
Eyes1=imcrop(I,BB1);
[rows, columns, numberOfColorChannels] = size(Eyes1);
%Încadrarea ochilor
righteye = Eyes1(1:end, 1:round(columns/2), :);
lefteye = Eyes1(1:end, round(columns/2):end, :);

%Cu functia imfindcircle se cauta in imagine pupila care a fost incadrata
%mai sus
[centers,radii] = imfindcircles(er2,[30 120],'ObjectPolarity','bright', ...
    'Sensitivity',0.92)

%Detectarea pupilelor
figure,imshow(Eyes1)

```

```
title('Iris detectat')

%functia viscircles creeaza un cerc cu centru si raza stocate mai sus cu

%functia imfindcircles

h = viscircles(centers,radii);
```


6. Referințe

Figuri:

[1] <https://cnx.org/contents/WeBHosox@1.1:ZSSjvvgF@2/Iris-Recognition-Results-and-Conclusions>

[2] <https://www.nationalgeographic.com/photography/article/milestones-photography>

[3] https://bionescu.aimultimedialab.ro/index_files/tapai/TAPAI_BIonescu_M4.pdf

[4] <https://www.lcipaper.com/kb/what-are-the-differences-between-pantone-cmyk-rgb.html>

[5] https://www.researchgate.net/figure/An-example-comparison-of-the-low-resolution-input-of-the-625-of-k-space-with-the_fig4_349044795

[6] <http://easy-learning.neuro.pub.ro:8888/Laboratoare/L4-Echipamente%20de%20scanare/Echipamente%20de%20scanare/Cap5/Histograma/Histograma.htm>

[7] http://users.utcluj.ro/~rdanescu/pi_c03.pdf

[8] <https://www.mathworks.com/matlabcentral/fileexchange/26978-hough-transform-for-circles>

[9] https://koaha.org/wiki/Trasformata_di_Radon

[10] <https://www.rasfoiesc.com/educatie/informatica/Metode-de-binarizare-a-imagini32.php>

- <https://cnx.org/contents/WeBHosox@1.1:ZSSjvvgF@2/Iris-Recognition-Results-and-Conclusions>
- <https://medium.com/analytics-vidhya/what-is-haar-features-used-in-face-detection-a7e531c8332b>
- <https://www.mathworks.com/help/images/ref/imcomplement.html>
- <https://www.geeksforgeeks.org/matlab-erosion-of-an-image/>
- <https://www.mathworks.com/matlabcentral/answers/334096-why-does-imfindcircles-not-find-circles-in-my-image>

- <https://cimss.ssec.wisc.edu/wxwise/class/aos340/spr00/whatismatlab.htm>
- <https://cmmi.tuiasi.ro/wp-content/uploads/cursuri/Limbaje%20de%20programare%20structurate.%20Aplicatii%20MATLAB.pdf>
- [https://en.wikipedia.org/wiki/Thresholding_\(image_processing\)](https://en.wikipedia.org/wiki/Thresholding_(image_processing))
- https://koaha.org/wiki/Trasformata_di_Radon
- http://www.master-taid.ro/Cursuri/IVOM_curs/ivom_trasataturi%20faciale_2019.pdf
- https://www.mathworks.com/help/images/ref/bwareaopen.html?s_tid=srchtitle_bwareaopen_1
- <https://anale-informatica.tibiscus.ro/download/lucrari/1-2-18-Mark.pdf>
- https://www.miv.ro/books/MIvanovici_PI.pdf
- <https://dokumen.tips/documents/operatii-morfologice.html?page=3>
- <http://alpha.imag.pub.ro/ro/cursuri/archive/color.pdf>
- <https://www.rasfoiesc.com/educatie/informatica/Metode-de-binarizare-a-imagini32.php>
- https://webpace.ulbsibiu.ro/catalina.neghina/Resurse/PI/pdf_PI/L1.pdf
- <https://www.mathworks.com/matlabcentral/answers/334096-why-does-imfindcircles-not-find-circles-in-my-image>