UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI

FACULTATEA DE INFORMATICĂ

LUCRARE DE LICENȚĂ

Peer Group Image Denoising

propusă de

Diana Andreea Ștefanovici

Sesiunea: *iulie, 2019*

Coordonator științific

Lect. Dr. Anca Ignat

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI

FACULTATEA DE INFORMATICĂ

Peer Group Image Denoising

Diana Andreea Ștefanovici

Sesiunea: *iulie, 2019*

Coordonator științifi

Lect. Dr. Anca Ignat

Avizat,

Îndrumător Lucrare de Licenţă

Titlul, Numele şi prenumele _____

Data _____Semnătura _____

DECLARAȚIE privind originalitatea conținutului lucrării de licență

Subsemntatul(a) ……

domiciliul în ……..

născut(ă) la data de ……., identificat prin CNP ……., absolvent(a) al(a) Universităţii „Alexandru Ioan Cuza" din Iaşi, Facultatea de …….. specializarea ……., promoţia ……., declar pe propria răspundere, cunoscând consecinţele falsului în declaraţii în sensul art. 326 din Noul Cod Penal şi dispoziţiile Legii Educaţiei Naţionale nr. 1/2011 art.143 al. 4 si 5 referitoare la plagiat, că lucrarea de licenţă cu titlul: _____

_____

_____elaborată sub îndrumarea dl. / d-na _____, pe care urmează să o susţină în faţa comisiei este originală, îmi aparţine şi îmi asum conţinutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licenţă să fie verificată prin orice modalitate legală pentru confirmarea originalităţii, consimţind inclusiv la introducerea conţinutului său într-o bază de date în acest scop.

Am luat la cunoştinţă despre faptul că este interzisă comercializarea de lucrări ştiinţifice in vederea facilitării falsificării de către cumpărător a calităţii de autor al unei lucrări de licenţă, de diploma sau de disertaţie şi în acest sens, declar pe proprie răspundere că lucrarea de faţă nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Dată azi, …… Semnătură student ……

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „*Peer Group Image Denoising*”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iaşi, 27.06.2019

Absolvent *Diana Andreea Ștefanovici*

_____

(semnătura în original)

ACORD PRIVIND PROPRIETATEA DREPTULUI DE AUTOR

Facultatea de Informatică este de acord ca drepturile de autor asupra programelor-calculator, în format executabil şi sursă, să aparţină autorului prezentei lucrări, *Diana Andreea Ştefanovici*

Încheierea acestui acord este necesară din următoarele motive:

[Se explică de ce este necesar un acord, se descriu originile resurselor utilizate în realizarea

produsului-program (personal, tehnologii, fonduri) şi aportul adus de fiecare resursă.]

Iaşi, *data*

Decan *Adrian Iftene* Absolvent *Diana Andreea Ştefanovici*

_____ _____

(semnătura în original) (semnătura în original)

Cuprins

Introducere

Imaginile obținute în urma unor operații de achiziționare, transfer de date, decompresie a unui fișier imagine compresat în vederea unei transmisii mai rapide sunt afectate de diferite tipuri de degradări care influențează caracteristicile reprezentative ale imaginilor (cele mai importante fiind legate de poziția obiectelor în imagine, forma obiectelor, posibilitatea distingerii obiectelor de fundal). Imaginile obținute în urma acestor perturbații sunt greu de analizat, deorece posibile operații ulterioare, cum ar fi segmentarea, identificarea sau recunoașterea obiectelor din imagine, conduc la rezultate eronate. Mecanismul de degradare este strâns legat de procesele fizice implicate în perturbare: rezoluția finită a senzorului matriceal (la momentul scanării), procesul de refractare (deviere în momentul trecerii de la un mediu la altul), focalizarea greșită, senzații de estompare și mișcare a imaginii la fotografiere, precum și apariția efectelor de tip zgomot (de obicei repartizat normal sau uniform) în procesele de cuantificare, măsurare și transmitere de date.

De cele mai multe ori, degradarea unei imagini constă în apariția zgomotului (Figura 1) și a blurării (Figura 2).

Figura 1. Imaginea originală (A) - Imaginea afectată de zgomot (B)

Figura 2. Imaginea originală (stânga) - Imaginea blurată (dreapta)

Restaurarea imaginilor este procesul de reconstrucție pe baza unor cunoștințe a priori (cunoștințe care nu au fost dobândite în urma experienței, fiind necesare și universale) legate de modelul de degradare. De aceea tehnicile de restaurare sunt orientate în principiu către modelarea matematică a degradării, urmată de aplicarea procesului invers în vederea obținerii imaginii inițiale. O abordare de acest gen implică de obicei formularea unor criterii de tip performanță care conduc la estimări optimale ale rezultatului dorit.

Ca o consecință directă, calitatea tehnicilor de restaurare depinde esențial de acuratețea procesului de modelare a degradării. Au fost dezvoltate o serie de modele matematice, pe baza diferitelor tipuri de cunoștințe a priori considerate.

Unul dintre cele mai utilizate modele de degradare este cel linear, în care se presupune că procesul de deteriorare a imaginii este reprezentat printr-o superpoziție a acesteia ca răspunsul unui impuls *H*, la care eventual este adăugată o componentă zgomot aditiv.

În domeniul spațial, modelul de degradare este exprimat prin convoluția dintre operatorul *h* și imaginea originală *f,* urmată de adăugarea componentei zgomot aditiv *η,* cu rezultat imaginea observată (degradată) *g.* Fie funcția *f* funcție imagine de dimensiune $N \times M$. Pentru $1 \le x \le N$, $1 \le y \le M$, modelul de degradare este exprimat prin relația:

$$g(x, y) = (h \otimes f)(x, y) + \eta(x, y) \qquad (1)$$

În domeniul frecvențelor, relația (1) este exprimată pe baza teoremei de convoluție prin:

$$g(n, m) = h(n, m) * f(n, m) + \eta(n, m) \qquad (2)$$

unde $1 \le n \le N$, $1 \le m \le M$ și *g, h, f, η* desemnează reprezentările Fourier (conversii ale datelor imagine aranjate spațial într-o reprezentare în spațiul de frecvențe).

În acest caz, restaurarea imaginii perturbate *g(x, y)* este procesul de obținere a unei aproximații a originalului *f(x, y)*, presupunând cunoscute informațiile referitoare la procesul de degradare înglobate în forma operatorului *h*. [1]

Filtrarea liniară constă în înlocuirea valorii unui pixel corupt (afectat de zgomot) cu media ponderată a pixelilor necorupți din fereastra (vecinătatea) *W* din

care face parte. O soluție simplă este medierea cu pixelii vecini. Această metodă este puternică în domeniul spațial. Astfel, fiind dată o imagine cu *N*N* pixeli f*(x, y)*, procedura constă în a genera o imagine filtrată *g(x, y)* ale cărei niveluri de gri în fiecare punct *(x, y)* sunt obținute prin medierea valorilor pixelilor lui f conținuți într-o vecinătate a punctului *(x, y)*. Se utilizează relația:

$$g(x, y) = 1/M \Sigma f(n, m), (n, m) \in W$$

De asemenea, există și metode foarte populare de filtrare neliniară. Unul dintre cele mai utilizate filtre este cel median cu ajutorul căruia, într-o fereastră de filtrare (W), valoarea pixelului central se determină ca fiind valoarea mediană a valorilor pixelilor din acea fereastră.

Conținutului tezei presupune în restaurarea imaginilor prin detecția zgomotului folosind metoda Peer Group și apoi eliminarea acestuia aplicând Mean Filter.

Contribuții

Lucrarea curentă prezintă o tehnică de reducere a zgomotului mai nouă, și anume peer group. Ea va fi utilizată pentru detectarea pixelilor corupți dintr-o imagine afectată de zgomot. După încheierea procesului de clasificare al pixelilor în corupți și necorupți, cei afectați vor fi modificați cu ajutorul. Filtrului median.

În ceea ce privește contribuțiile personale, voi prezenta, pe scurt, etapele de dezvoltare a soluției finale.

Alegerea temei a fost curiozitatea cunoașterii unui domeniu nou, și anume procesarea imaginii. M-am documentat și am ajuns la concluzia că detecția și reducerea zgomotului dintr-o imagine stau la baza oricărui proces de prelucare a fotografiilor. Astfel, mi-am îndreptat interesul spre găsire unei metode cât mai optime. Dezvoltarea tehnicii peer group a venit la sugestia doamnei profesor coordonator de a uitiliza o modalitate nouă și nu foarte întâlnită.

În vederea implementării unei modalități cât mai bune, am aplicat algoritmul descris în capitolele următoare. Pentru acest lucru am uitlizat limbajul de programare Python, un limbaj interpretat, de nivel înalt, cu scop general. Python are tipuri de date dinamice și garbage collector. De asemenea suportă mai multe paradigme de programare, inclusiv cea orientată pe obiecte, cât și programarea funcțională.

În dezvoltarea proiectului a fost necesară folosirea bibliotecii OpenCV, disponibilă atât în Python, cât și în Java, C++, C, pentru procesarea imaginilor și video. OpenCV este folosit pentru toate tipurile de analize de imagine și video, cum ar fi recunoașterea și detectarea feței, citirea plăcuței de înmatriculare, editarea forografiilor, viziunea robotică avansată, recunoașterea optică a caracterelor, detectarea și reducerea zgomotului și multe altele.

În continuarea, m-am axat pe documentarea progresivă pentru alcătuirea unei lucrări cât mai explicită și care să includă toate mijloacele necesare unei aplicații de îmbunătățire a imaginilor.

Capitolul 1. Descrierea problemei

Imaginile digitale joacă un rol foarte important în rutina zilnică, cum ar fi cele utilizate în televiziunea prin satelit, monitorizarea inteligentă a traficului, recunoașterea scrisului de mână pe cecuri, validarea semnăturii, imagistica prin rezonanță pe computer și în domeniul cercetării și al tehnologiei, cum ar fi sistemele informatice geografice și astronomia. În imagistica digitală, tehnicile de achiziție și sistemele introduc diferite tipuri de zgomote și artefacte. Reducerea zgomotului este mai importantă decât orice alte sarcini în procesarea, analiza și aplicațiile imaginilor.

Rezervarea detaliilor unei imagini și eliminarea zgomotului aleator, pe cât posibil, este scopul abordărilor de netezire a imaginii. Pe lângă faptul că imaginea zgomotoasă generează o calitate vizuală nedorită, reduce și vizibilitatea obiectelor cu contrast scăzut. Prin urmare, eliminarea zgomotului este esențială în aplicațiile de imagini digitale, pentru a îmbunătăți și a recupera detaliile fine care sunt ascunse în date. În multe cazuri, zgomotul în imaginile digitale se dovedește a fi aditiv în natură, cu o putere uniformă pe toată lărgimea de bandă.

În reducerea zgomotului există întotdeauna un compromis între suprimarea zgomotului și conservarea discontinuităților reale ale imaginii. Pentru a elimina zgomotul fără a netezi excesiv detaliile importante, o tehnică de reducere trebuie să fie adaptabilă spațial. Sunt utilizate diferite tehnici în funcție de modelul de zgomot.

Documentul prezent descrie implementarea unei tehnici noi de reducere a zgomotului din imaginile corupte. Aceasta presupune utilizarea peer grupurilor pentru detectarea pixelilor afectați, apoi aplicarea unui filtru pentru a atenua zgomotului. Se încearcă reconstituirea pe cât posibil a imaginii originale.

Lucrarea este organizată după cum urmează. Prima parte prezintă o introducere în ceea ce înseamnă procesarea imaginilor, apoi vor fi detaliate tehnici utilizate în implementarea soluției, și anume filtrarea imaginii, pixelul, matricea pixelului, trăsături, RGB. Secțiunea următoare conține definirea zgomotului, tipuri ale acestuia și prezentarea câtorva modalități de netezire a imaginilor. Mai apoi este detaliată noțiunea de peer group. În final este descrisă soluția finală și rezultatele obținute în practică.

Capitolul 2. Filtrarea imaginii

Indiferent daca suntem conștienți sau nu de acest lucru, viziunea computerului este peste tot în viața noastră de zi cu zi. Fotografiile filtrate sunt omniprezente în feedurile noastre de social media, articole de știri, reviste, cărți, peste tot. Dacă ne gândim la imagini ca funcții de cartografiere a locațiilor în imagini la valori ale pixelilor, atunci filtrele sunt doar sisteme care formează o imagine nouă și preferabil îmbunătățită dintr-o combinație a valorilor originale ale pixelilor.

Imaginile ca Funcții

Pentru a înțelege mai bine proprietățile inerente ale imaginilor și procedura tehnică folosită pentru a le manipula și procesa, putem gândi o imagine care este compusă din pixeli individuali, ca o funcție *f*. Fiecare pixel are, de asemenea, o valoare proprie. Pentru o imagine în tonuri de gri, fiecare pixel ar avea o intensitate între 0 și 255, 0 fiind negru și 255 alb. f (x, y) ar da apoi intensitatea imaginii în poziția pixelului (x, y), presupunând că este definită pe un dreptunghi, cu un domeniu finit: *f: [a, b] × [c, d] → [0, 255].* (Figura 3)

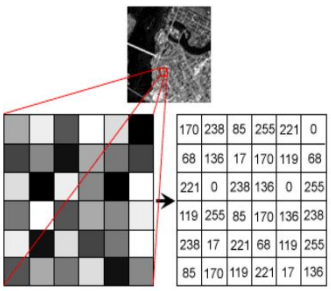


Figura 3. Pixel alb-negru

O imagine color este doar o extensie simplă a acestui lucru. *f (x, y)* este acum un vector de trei valori în loc de unul. Folosind o imagine RGB ca exemplu, culorile sunt construite dintr-o combinație de roșu, verde și albastru (RGB). Prin urmare, fiecare pixel al imaginii are trei canale și este reprezentat ca un vector 1×3. Deoarece cele trei culori au valori întregi de la 0 la 255, există un total de 256 * 256 * 256 = 16.777.216 combinații sau opțiuni de culoare. (Figura 4)

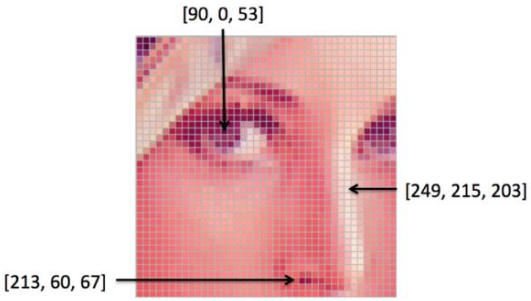


Figura 4. Pixel color

Mai apoi, o imagine poate fi reprezentată ca o matrice cu valorile pixelilor.(Figura 5)

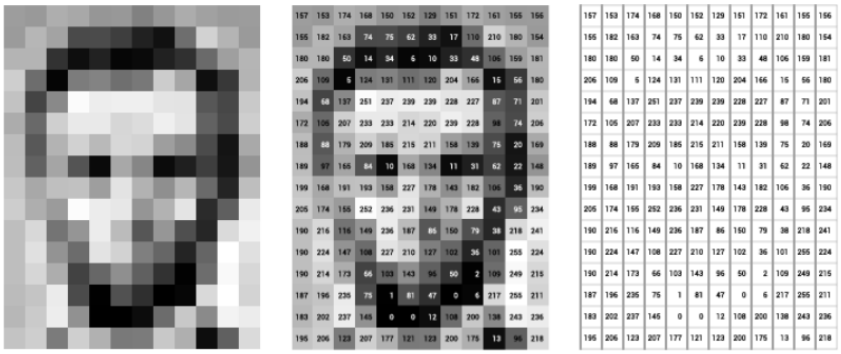


Figura 5. Reprezentarea imaginii sub forma unei matrici

Procesarea imaginii

Există două tipuri principale de procesare a imaginilor: filtrarea imaginilor și deformarea imaginilor. Filtrul de imagine modifică intervalul (adică valorile pixelilor) unei imagini, astfel încât culorile imaginii sunt modificate fără a schimba pozițiile pixelilor, în timp ce deformarea imaginii schimbă domeniul (adică pozițiile pixelilor) unei imagini, unde punctele sunt mapate spre alte puncte fără a schimba culorile.(Figura 6)

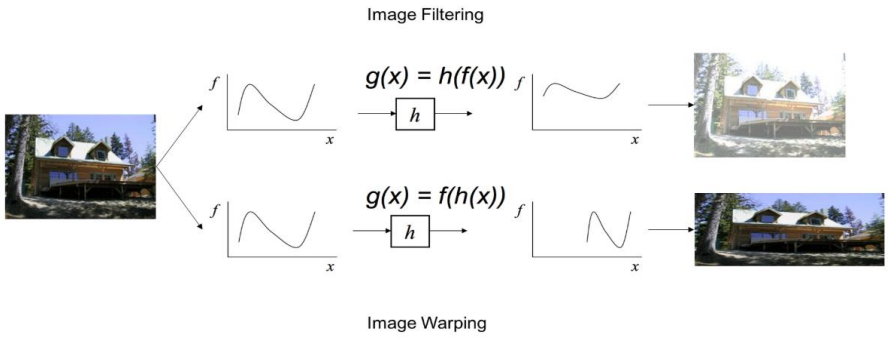


Figura 6. Filtrarea imaginii (sus) - Deformarea imaginii (jos)

Detecția și analiza imaginilor sunt cele mai cunoscute aplicații în domeniul prelucrării imaginilor sau reprezintă o parte importantă a unui sistem de analiză a imaginilor. În prefața uneia dintre cele mai citate cărți din domeniul procesării imaginilor, se apreciază ca un sistem suficient de performant de prelucare a imaginilor ar trebui, ca pentru o imagine care cuprinde un drum pe lângă care trece o apă și peste ea o punte, sa le identifice pe toate dintre cele trei elemente. [2]

Scopul utilizării filtrelor este de a modifica sau de a îmbunătăți proprietățile imaginii și de a extrage informații valoroase din imagini, cum ar fi marginile și colțurile.

Fiecare imagine digitală are un zgomot de fond, însă acesta devine sesizabil de la un anumit grad de intensitate în sus, sub forma unor puncte individuale, cu valoare falsă de luminozitate de culoare. Zgomotul de imagine al luminozității seamănă ca și aspect cu granulația filmului, iar cel al culorii este vizbil sub forma unor pete de culoare.

În lucrarea curentă mă voi axa pe filtrarea imaginilor, mai precis pe reducerea zgomotului și reconstituirea fotografiei originale, atât alb-negru, cât și color.(Figura 7) [3]

Figura 7. Reducerea zgomotului

Ce este un pixel?

Pixelul este cel mai mic element al unei imagini. Fiecare pixel corespunde oricărei valori. Într-o imagine cu scală de 8 biți, valoarea pixelului este cuprinsă între 0 și 255. Valoarea unui pixel în orice punct corespunde intensității fotonilor de lumină care izbucnesc în acel moment. Fiecare pixel stochează o valoare proporțională cu intensitatea luminii din acea locație.[3]

Altfel spus, un pixel este un punct de pe ecran (al calculatorului, al telefonului, al tabletei etc.). Într-o formulare mai tehnică, pixelul desemnează cea mai mică unitate adresabilă a memoriei video a unui monitor. Pixelul este totodată un punct al unei imagini digitale. Făcând o analogie, pixelul este pentru o imagine (sau un monitor) ca atomul pentru materie. (Figura 8)

Figura 8. Pixelii unei porțiuni din imagine (dreapta)

Formatele digitale au nevoie de pixeli. Pentru a putea reprezenta o poză în calculator, ea trebuie împărțită în pătrățele foarte mici, asemenea unei table de șah, fiecare pătrățel având o anumită culoare. Astfel, ea poate fi reprezentată numeric.

Cu cât pixelii sunt mai mici, cu atât calitatea și rezoluția imaginii cresc, devenind mai clară și mai aproape de realitate.[4]

Trăsături

Definirea atributelor se poate face pe 1, 2, 4, 8, 16 sau chiar mai mulți biți pentru fiecare plan de culoare sau transparență. Pixelul definit pe 1 bit formează o imagine alb-negru. Cel pe 8 biți reprezintă o imagine grayscale care suportă 256 de nuanțe de gri, în care fiecare pixel este o cantitate de lumină, adică are doar informații despre intensitate.

Imaginile în nuanțe de gri, un fel de monocrom alb-negru sau gri, sunt compuse exclusiv din nuanțe de gri. Contrastul variază de la cea mai slabă intensitate de negru la cea mai puternică intensitate de alb.[5]

Pixelul poate fi definit și ca 3×8 biți, ce reprezintă o imagine RGB – Red Green Blue. Este un model aditiv de culoare, în care roșu, verde și albastru sunt amestecate în diverse moduri pentru a produce o gamă largă de culori. Fiecare secvență de 8 biți ia valori de la 0 la 255 și poate fi ilustrată ca o funcție. (Figura 9)

$$f(x,y) = \begin{bmatrix} r(x,y) \\ g(x,y) \\ b(x,y) \end{bmatrix}$$

Figura 9. Componentele RGB ale unui pixel

Pixelul definit ca 4×8 biți simbolozează o imagine CMYK – Cyan, Magenta, Yellow, Key. Este un model substractiv utilzat în tipărirea color. Pentru obținerea unei anumite culori sunt combinați pigmenții celor patru culori de bază, cyan (albastru deschis), magenta, galben, negru. Cu ajutorul acestui model se pot reproduce aproape toate culorile din spectrul vizibil, mai puțin cele fluorescente.(Figura 10) [6][7]

Figura 10. Imagine originală (sus) - Imagine separată în culorile CMYK (jos)

Rezoluția

Atunci când vine vorba de rezoluția unei imagini, se are în vedere numărul de linii ce sunt afișate pe orizontală și pe verticală și numărul de pixeli afișați.

Rezoluția digitală reprezintă o măsură a clarității sau a gradului de detaliere a unei imagini digitale. Acestea din urmă sunt forma de memorare și prelucrare a imaginilor obișnuite într-un calculator. Ele pot fi afișate sau tipărite, dar nu pot fi văzute ca atare, direct, ca imaginile reale.[9]

Numărul absolut de pixeli al unei imagini digitale definește rezoluția digitală a acesteia. O imagine dreptunghiulară din memoria calculatorului poate avea, spre exemplu, o rezoluție digitală de 1600 × 1200 pixeli (lungime și lățime). Dacă pixelii sunt pătrați, atunci raportul dintre laturi va fi 1600:1200 sau 4:3. Rezoluția poate fi exprimată și prin numărul total de pixeli. Astfel, pentru exemplul precedent am avea 1920000 px sau 1,92 Mpx (megapixeli).

Pentru imagini reale, tipărite sau afișate pe ecran, are importanță și numărul de pixeli raportat la lungime. Rezoluția optică se exprimă în pixeli pe inch, prescurtat *ppi* (un inch = 25, 4 mm). Imaginile redate grafic pe calculatoare cu sistemul de operare Mac OS au, de obicei, 72 ppi, iar cele cu Windows au 96 ppi.

Deseori cele două rezoluții sunt confundate, cea digitală a unei imagini din memorie cu cea optică fizică a unui aparat (ecran, scanner). Este și termenul de rezoluție la tipar (print resolution). În tehnica tipografică s-a folosit o altă unitate de măsură, *dpi* (dots per inch). Prin natura lor, aceste puncte (dots) sunt tot componente ale imaginii, foarte asemănătoare cu pixelii, doar că sunt mult mai mici.

De cele mai multe ori, numărul de dpi al imprimantelor este mult mai mare decât numărul de ppi al imaginii de tipărit. De exemplu, dacă rezoluția tipografică a imprimantei este de 600 dp, iar imaginea digitală trebuie tipărită cu 96 ppi, atunci pentru fiecare pixel al imaginii vor fi transpuse pe hârtie în medie 6,25 dots pe lungime și 6,25 pe lățime. Acest lucru înseamnă aproximativ 39 de dots pe pixel. [10]

Capitolul 3. Zgomotul

Zgomotul este o specificitate proprie a tuturor formelor de imagistică și pot fi găsite în diverse forme în toate domeniile imaginilor digitale.

Zgomotul imaginii este echivalentul digital al granulațiilor de film (film grain) pentru camerele analogice. În mod alternativ, se poate gândi la acest lucru ca fiind similar cu șuierăturile de fundal subtile pe care le puteți auzi de la sistemul audio când este utilizat la volum maxim. În cazul imaginilor digitale, acest zgomot apare ca speculații aleatorii pe o suprafață netedă și poate degrada semnificativ calitatea imaginii. Deși zgomotul deseori slăbește imaginea, este uneori de dorit, deoarece poate adăuga un aspect de modă veche, care amintește de filmul timpuriu. Uneori chiar poate spori și claritatea aparentă a unei imagini. Zgomotul crește odată cu setarea de sensibilitate a camerei, cu lungimea expunerii, cu temperatura și chiar și cu diferitele modele de camere.[11]

Zgomotul reprezintă informație nedorită ce deteriorează calitatea imaginii. Acesta este definit ca un proces *(n),* care afectează imaginea obținută *(f)* și nu este o parte a semnalului inițial *(s).* Folosind modelul aditiv al zgomotului, acest proces poate fi scris sub forma: [12]

$$f(i, j) = s(i, j) + n(i, j)$$

Cauze ale producerii zgomotului

Zgomotul imaginilor digitale poate proveni din diferite surse. Unele sunt fizice, legate de natura luminii și de artefactele optice, iar altele sunt create în timpul conversia de la semnalul electric la datele digitale. Procesul de achiziție convertește semnalele optice în semnale electrice și apoi în semnale digitale. Este unul din procesele prin care zgomotul este introdus în imaginile digitale. Orice pas al conversiei produce fluctuație cauzată de fenomenele naturale, adăugând valori aleatorii intensității rezultate pentru fiecare pixel. [12]

Deoarece zgomotul degradează calitatea unei imagini, diferite modele au fost investigate pentru a modela zgomotul imaginii în vederea reducerii sau eliminării ulterioare, în diferite etape ale procesului de preluare a imaginii.[13]

Modelarea zgomotului

Zgomotul *(n)* poate fi modelat cu ajutorul unei histograme sau al funcției probabilității densității care este suprapusă pe cea a imaginii originale *(s)*. În continuare vor fi prezentate modelările celor mai comune tipuri de zgomot.

Zgomotul sare și piper (The salt & pepper noise)

În modelul zgomotului sare și piper, numit și zgomot de impuls, sunt posibile doar două valori, *a* si *b,* iar probabilitatea obținerii fiecăruia dintre ele este mai mica decât 0,1 (altfel, zgomotul va domina imaginea). Pentru o imagine de tipul 8 biți/pixel, valoarea tipică a intensității zgomotului de tip *piper (pepper)* este aproape de 0 și pentru cel de tip *sare (salt)* este aproape de 255.

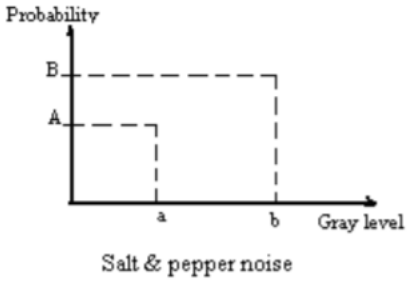


Figura 11. Probability density function for the salt and pepper noise model (PDF)

$$PDF_{salt\&pepper} = \begin{cases} A & for\ g = a\ ("pepper") \\ B & for\ g = b\ ("salt") \end{cases}$$

Figura 11. Ilustrează PDF – ul (probability density function) sau funcția probabilității densității, pentru zgomotul de tip sare și piper, mai exact, dacă media este 0 și varianța este 0.05. Se observă două vârfuri, unul care indică regiunea de luminozitate (unde nivelul de gri sau gray level este mai scăzut), *regiunea a* și unul pentru regiunea întunecată (unde nivelul de gri este mai ridicat), *regiunea b.* Aici, valorile date de PDF atinge punctele minim (regiunea a) și maxim (regiunea b).

Zgomotul de tip sare și piper este cauzat, în general, de multifuncționalitatea celulelor senzorilor de cameră, de eroarea celulelor de memorie sau de sincronizarea erorilor în digitalizarea sau transmisiunea imaginii. [12]

Figura 12 ilustrează o imagine degradată de zgomot de tip sare și piper.

Figura 12. Imagine cu zgomot de tip sare și piper

Zgomotul Gaussian

Zgomotul Gaussian are funcție de probabilitate a densității normală (Gaussiană). Este numit și zgomot electronic deoarece apare în amplificatoare sau detectoare. Este provocat de surse naturale cum ar fi vibrații termale ale atomilor și radiații ale obiectelor calde.

În general, zgomotul Gaussian perturbă valorile de gri (grey values) ale imaginilor digitale. Acesta este motivul pentru care modelul zgomotului Gaussian este conceput și caracterizat esențial de PDF – ul său sau de histograma normalizată pe baza respectării valorilor de gri. Acest lucru este dat de formula:

$$P(g) = \sqrt{\frac{1}{2\pi\sigma^2}} e^{-\frac{(g-\mu)^2}{2\sigma^2}}$$

Unde g = gray value, δ = deviația standard, μ = media

De obicei, modelul matematic al zgomotului Gaussian reprezintă aproximarea corectă a scenariilor reale. În modelul următor, media este egală cu 0, varianța este 0,1 și 256 nivele de gri în terminii PDF – ului său, ilustrat în figura 13.

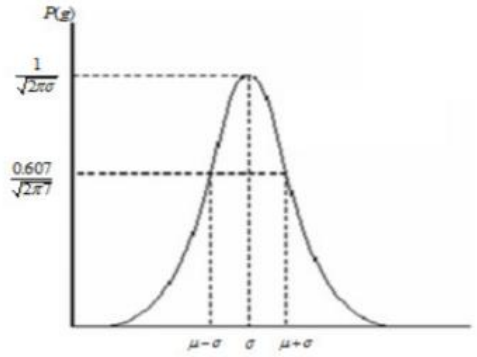


Figura 13. PDF – ul zgomotului Gaussian

Conform acestei egalități, PDF – ul modelului arată că între 70% și 90% dintre valorile pixelilor cu zgomot (noisy pixel) ale imaginii degradate sunt între $\mu - \delta$ și $\mu + \delta$. Forma histogramei normalizate este aproximativ similară în domeniul spectral. [14]

Figura 14 ilustrează o imagine degradată prin intermediul zgomotului de tip Gaussian.

Figura 14. Imagine cu zgomot de tip Gaussian

Zgomotul uniform

Este un model teoretic, simplu de generat și este folosit la degradarea imaginilor pentru evaluarea algoritmilor de restaurare, deoarece oferă un model de zgomot neutru.

Aspectul zgomotului este moștenit din amplitudinea procesului de cuantizare. Este prezentat, în general, potrivit conversiei în date digitale. În acest model, semnalul spre rația zgomotului (SNR) este limitată de valoarea minimă a pixelului, respectiv cea maximă.

Cu distribuția uniformă, valorile nivelelor de gri ale zgomotelor sunt distribuite intr-un domeniu specific, care poate fi întreg domeniul (0 – 255 pentru 8 biți) sau o porțiune mai mică din acest domeniu.[14]

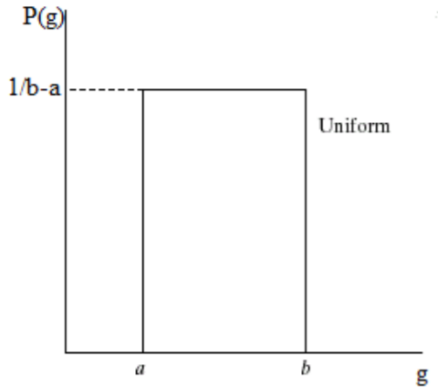


Figura 15. Probability density function for the uniform noise model (PDF)

Pentru modelul zgomotului uniform avem:

$$P(g)=\begin{cases}\dfrac{1}{b-a} & \text{if } a \le g \le b \\ 0 & \text{otherwise}\end{cases} \qquad \mu=\frac{a+b}{2} \quad \sigma^2=\frac{(b-a)^2}{12}$$

Zgomotul Speckle

Acest zgomot este multiplicativ. Aspectul său este văzut în sisteme de imagini coerente cum ar fi laserul, radarul sau acusticele. Zgomotul de tip Speckle poate exista într-o imagine similar cu zgomotul Gaussian. Funcția de probabilitate a densității urmează distribuția gamma și este dată în formula[14]:

$$F(g)=\frac{g^{\alpha-1}e^{\frac{-g}{a}}}{\alpha-1!a^{\alpha}}$$

Figura 16 ilustrează o imagine afectată de zgomotul de tip Speckle cu o varianță de 0,04.

Figura 16. Image with Speckle noise

Zgomotul Gamma

Zgomotul de tip Gamma este, în general, văzut în imaginile bazate pe laser. Acesta urmează distribuția gamma, observată în figura 17[14].

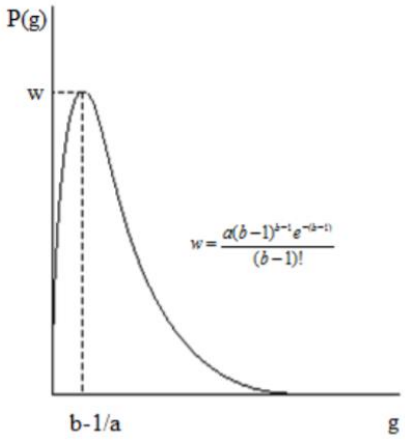


Figura 17.

Pentru modelul zgomotului Gamma avem:

$$P(g) = \begin{cases} \dfrac{a^b g^{b-1} e^{-ag}}{(b-1)!} & \text{for } g \geq 0 \\ 0 & \text{for } g < 0 \end{cases} \qquad \mu = \frac{b}{a} \quad \sigma^2 = \frac{b}{a^2}$$

Modalități de eliminare a zgomotului

Principala provocare în procesarea digitală a imaginilor este eliminarea zgomotului din imaginea originală. Alegerea algoritmului de eliminare a zgomotului depinde de aplicație. Prin urmare, este necesar să avem cunoștințe despre zgomotul prezent în imagine pentru a selecta algoritmul adecvat, lucru prezentat anterior.

Mean Filter (filtru mediu)

Un filtru mediu acționează asupra unei imagini prin uniformizarea acesteia. Reduce variația intensității întreadiacenți. Filtrul mediu nu este altceva decât un filtru spațial simplu al ferestrei glisante care înlocuiește valoarea centrală înfereastra cu media tuturor valorilor pixelilor vecine, inclusiv ea însăși. Imaginea coruptă cu zgomot de sare și piper estesupusă unui filtru mediu, apoi se poate observa că zgomotul dominat este redus. Valorile pixelilor albi și negri din zgomot sunt modificate pentru a fi mai aproape de valorile pixelilor celor din jur. De asemenea, luminozitatea imaginii de intrarerămâne neschimbată datorită utilizării măștii, a cărei coeficienți se însumează până la valoarea unică. Se folosește filtrul mediu înaplicații în care zgomotul din anumite regiuni ale imaginii trebuie eliminat. Cu alte cuvinte, filtrul mediu este utilcând trebuie procesată doar o parte a imaginii.[15]

Filtrarea medie este o metodă simplă, intuitivă și ușor de implementat pentru netezirea imaginilor, adică reducerea variației intensității între un pixel și cel de-al doilea. Este adesea folosit pentru a reduce zgomotul în imagini.

Ideea de filtrare medie este pur și simplu de a înlocui fiecare valoare a pixelului într-o imagine cu valoarea medie a vecinilor săi, inclusiv pe sine. Acest lucru are efectul de a elimina valorile pixelilor care nu sunt reprezentative pentru împrejurimile lor. Filtrarea medie este de obicei considerată ca un filtru de convoluție. Ca și alte convoluții, el se bazează pe un kernel, care reprezintă forma și dimensiunea blocului care urmează să fie eșantionat la calcularea mediei. Adesea se folosește un kernel pătrat de 3 × 3, numit și fereastră, așa cum se arată în figura 18, deși pot fi utilizate și dimensiuni mai mari (de exemplu, 5 × 5 pătrate) pentru o nivelare mai severă. (Rețineți că un kernel mic poate fi aplicat de mai multe ori pentru a produce un efect similar, dar nu identic, ca o singură trecere cu un kernel mare). [16]

Un kernel este o matrice (de regulă) mică a numerelor care este folosită în convoluțiile imaginii. Miezurile cu dimensiuni diferite, care conțin diferite modele de numere, dau naștere unor rezultate diferite în urma convoluției.

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

Figura 18.

Calculând convoluția simplă a unei imagini cu acest kernel, se realizează procesul de filtrare mediu.

Se observă aplicarea filtrului pe o imagine afectată de zgomot de tip sare și piper (Figura 19), folosind ferestre de 3 × 3, respectiv 5 × 5.

Figura 19. Imaginea afectată (stânga) - imagine cu filtru 3 × 3 (mijloc) - imagine cu filtru 5 × 5 (dreapta)

Fotografia din mijloc arată efectul de netezire al imaginii zgomotoase cu un filtru mediu de 3 × 3. Deoarece valorile pixelilor de zgomot sunt adesea foarte diferite de valorile din jur, ele tind să distorsioneze semnificativ media pixelilor calculată de filtrul mediu.

Folosind un filtru mediu de 5 × 5 nu reprezintă o îmbunătățire semnificativă a reducerii zgomotului și, în plus, imaginea este acum foarte neclară.

Aceste exemple ilustrează cele două probleme principale ale filtrului mediu. Un singur pixel cu o valoare foarte nereprezentativă poate afecta în mod semnificativ valoarea medie a tuturor pixelilor din vecinătatea sa. A doua problemă este că, atunci când fereastra de filtrare se întinde pe o margine, filtrul va interpola noi valori pentru pixeli pe margine și astfel va blura acea margine. Aceasta poate fi o problemă dacă sunt necesare muchii ascuțite în ieșire.[16]

Efectul acestui filtru nu este tocmai cel așteptat, motiv pentru care există alte modalități. Una dintre ele ar fi filtrarea mediană.

Filtrul median

Filtrul median urmează, de asemenea, principiul ferestrei în mișcare asemănător cu filtrul mediu. Un kernel 3 × 3, 5 × 5 sau 7 × 7 al pixelilor este scanat pe o matrice de pixeli a întregii imagini. Valoarea mediană a valorilor pixelilor din fereastră este calculată, iar pixelul central al ferestrei este înlocuit cu mediana calculată. Astfel, un singur pixel foarte nereprezentativ într-o vecinătate nu va afecta valoarea mediană. Deoarece cea din urmă trebuie să fie de fapt valoarea unuia dintre pixelii din vecinătate, filtrul median nu creează noi valori nerealiste ale pixelilor atunci când filtrul se întinde pe o margine. Din acest motiv, filtrul median este mult mai bun la păstrarea marginilor ascuțite decât filtrul mediu. Aceste avantaje ajută la reducerea zgomotului uniform dintr-o imagine. [15]

Ca și filtrul mediu, filtrul median ia în considerare fiecare pixel din imagine și privește vecinii din apropiere pentru a decide dacă este sau nu reprezentativ pentru împrejurimile sale. În loc să înlocuiască pur și simplu valoarea pixelilor cu media valorilor pixelilor vecini, ea o înlocuiește cu valoarea mediană a acestor valori. Mediana se calculează mai întâi prin sortarea tuturor valorilor pixelilor din vecinătatea înconjurătoare în ordine crescătoare și apoi înlocuirea pixelului considerat cu valoarea pixelului mediu. (Dacă fereastra în cauză conține un număr par de pixeli, se folosește media celor două valori ale pixelilor medii.) Figura 20 ilustrează un exemplu de calcul.

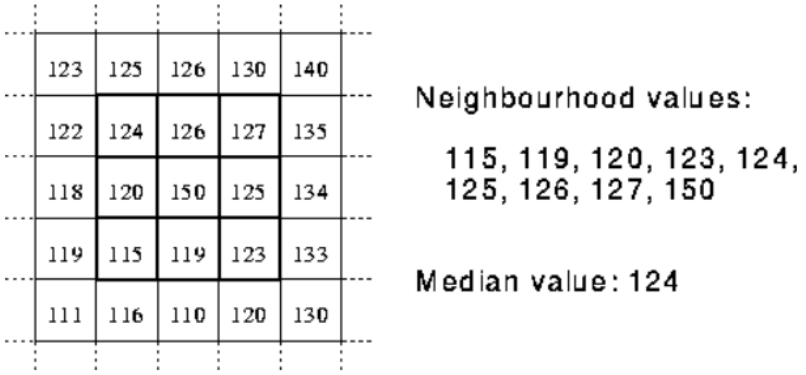


Figura 20.

În figura 20 se efectuează calculul valorii mediane a unui pixel într-o vecinătate. După cum se poate observa, valoarea centrală este 150 în fereastra 3 × 3 și este mai degrabă nereprezentativ pentru pixelii care îl înconjoară. Astfel este înlocuit cu valoarea mediană, adică 124. Ferestrele de dimensiuni mai mari vor produce o nivelare mai severă. [17]

În figura 21 se pot observa efectele filtrului median asupra unei imagini corupte.

Figura 21. Imagine coruptă de zgomot Gaussian (stânga) - Imagine cu filtru median 3 × 3 (dreapta)

Prin calcularea valorii mediane a unei vecinătăți, în comparație cu cea a filtrului mediu, filtrul median are două avantaje principale. Primul ar fi faptul că mediana este o medie mai robustă decât media și, deci, un singur pixel foarte nereprezentativ într-un bloc nu va afecta semnificativ valoarea mediană. Iar cel de-al doilea se referă la valoarea mediană care trebuie să fie de fapt valoarea unuia dintre pixelii din vecinătate, motiv pentru care filtrul median nu creează noi valori ale pixelilor nerealiste atunci când filtrul se extinde pe o margine. Acest lucru face ca filtrul median să fie mult mai bun la păstrarea marginilor ascuțite decât filtrul mediu, ceea ce se poate observa și in figura 21.[17]

Filtrul LMS (Least – Mean – Square Algorithm)

Un filtru adaptiv este un sistem cu un filtru liniar care constă din funcția de transfer limitată de parametrii variabili și un mijloc de reglare a acestor parametri în conformitate cu un algoritm de optimizare. Filtrele lineare adaptive sunt un sistem dinamic liniar cu structură variabilă sau adaptivă și parametri și au proprietatea de a modifica valorile parametrilor lor, adică funcția lor de transfer, în timpul procesării semnalului de intrare, pentru a genera semnal la ieșire care nu conține componente nedorite, zgomot și degradare, precum și semnale de interferență.

Algoritmii adaptivi au fost extensiv studiați în ultimele decade, iar algoritmii adaptivi cei mai populari sunt algoritmul least – mean - square (LMS) și algoritmul recursive – least - square (RLS). Atingerea celor mai bune performanțe ale unui filtru adaptiv necesită utilizarea celui mai bun algoritm adaptiv cu complexitate redusă de calcul și o rată de convergență rapidă.

O abordare foarte simplă în ceea ce privește anularea zgomotului este utilizarea algoritmului LMS dezvoltat de Windrow și Hoff. Acest algoritm utilizează o coborâre de gradient pentru a estima un semnal care variază în timp.

Gradientul imaginii este o schimbare direcțională a intensității sau culorii dintr-o imagine. El reprezintă unul din blocurile fundamentale ale procesării imaginilor. De exemplu, în software-ul grafic pentru editare digitală a imaginilor, gradientul este de asemenea utilizat pentru un amestec treptat de culoare, care poate fi considerat ca o gradare uniformă de la valori joase la valori mari (de la alb la negru). Un alt nume pentru aceasta este progresia culorii.

Metoda de coborâre în gradient găsește un minim, dacă există, luând pași în direcția negativă a gradientului și o face prin ajustarea coeficienților de filtrare pentru a minimiza eroarea. Gradientul este operatorul *“del”* și este aplicat pentru a găsi divergența unei funcții, care este eroarea în raport cu coeficientul *“n”* în acest caz. Algoritmul LMS a fost acceptat de mai mulți cercetători pentru implementarea hardware-ului datorită structurii sale simple. Pentru a le implementa, trebuie să se facă modificări algoritmului LMS inițial deoarece bucla recursivă din formula actualizării filtrului îi împiedică să fie transmisă.[18]

Filtrele adaptive sunt capabile să reducă zgomotul din imaginile non-staționare, adică imagini care au schimbări bruște de intensitate. Astfel de filtre sunt cunoscute pentru capacitatea lor de a urmări în mod automat o circumstanță necunoscută sau când un semnal este variabil, cu puțină cunoaștere a priori despre semnalul care urmează să fie procesat. Un filtru adaptiv face o treabă mai bună în vederea reducerii zgomotului comparativ cu filtrul de medie sau cel mediu. Filtrul adaptiv least – mean - square (LMS) este cunoscut pentru simplitatea sa în calcul și implementare. Algoritmul LMS funcționează bine pentru imagini corupte cu zgomot de tip sare și piper. [19]

Acestea sunt doar câteva dintre modalitățile de reducere a zgomotului din imaginile corupte, Pe parcursul tezei vor fi prezentate și alte metode puse în practică.

Capitolul 4. Peer Group

Introducere

În procesarea imaginilor de peer group identifică un "peer group" pentrufiecare pixel și apoi înlocuiește intensitatea pixelilor cu media setului sau cu ajutorul altei modalități de reducere a coruperii. Doi parametri oferă un control direct asupra căror caracteristici ale imaginiisunt selectiv îmbunătățite: suprafața (numărul de pixeli din caracteristică) șidiametrul ferestrei (mărimea ferestrei necesară pentru detecție).

Reducerea zgomotului și netezirea imaginii sunt folositoare în pașii de pre-procesare al multor aplicații de procesare a imaginilor. Un obiectiv general al acestor aplicații este de a suprima zgomotul în timp ce sunt păstrate informațiile despre margine.

Filtrul median, prezentat anterior, este o alegere populară în vederea reducerii zgomotului de impuls. Această modalitate de filtrare formează o aproximare a unei imagini prin pasarea unei ferestre de filtrare *d* × *d* peste acea imagine. Apoi se ia intensitatea mediană a ferestrei la adresa pixelului respectiv. Motivația utilizării acestui tip de filtru este aceea că mediana păstrează marginile (discontinuități de intensitate).

Pentru cazul zgomotului Gaussian și a impulsului, o metodă neliniară adaptabilă de filtrare multivariată este propusă. Utilizeazăvaloarea medie într-un cartier local de pixeli pentru a estima valoarea originală a pixelului și, prin urmare, poate bloca marginile și detaliile. Alte abordări pentru a netezi în timp ce păstrați granițele includmetodele variationale și filtrarea șocurilor.[20]

Zgomotul, provenit dintr-o varietate de surse, este inerent tuturor senzorilor electronici de imagine și, prin urmare, semnalul zgomot trebuie prelucratprintr-un algoritm de filtrare care anulează componenta de zgomot,păstrând în același timp structurile de imagine originale.Destul de des, imaginile color sunt corupte de zgomotul de impulscauzate de funcționarea defectuoasă a senzorilor în procesul de formare a imaginii,locațiile de memorie defecte în hardware, îmbătrânirea spațiului de stocare al materialelor sau erori de transmisie cauzate procese naturale sau create de om.

Cea mai populară familie de filtre nonlineare folosite pentru reducerea zgomotului de impuls în imaginile color este bazată pe statisticile de ordine. Aceste filtre utilizează ordonarea vectorială a setului de pixeli din fereastra de filtrare pentru a determina monstra de ieșire. Schema de ordonare a vectorului este definită prin sortarea distanțelor agregate:

$$r_k = \sum_{j=1}^{n} \left\| \boldsymbol{x}_k - \boldsymbol{x}_j \right\|_l$$

Multe filtre nonlineare de tip clasa multianuală definesc vectorul median $x_{(1)}$ ca ieșire a operației de filtrare, ca vectori care se deosebesc foarte mult de populația de date ce apare în locații indexate mai mari în secvența ordonată. În metodele clasice, filtrul Median Vector (VMF) este aplicat fiecărui pixel din imagine, indiferent dacă este zgomotos sau nu și, prin urmare, tinde pentru a blura detaliile imaginii, degradând semnificativ calitatea acesteia. Pentru a depăși aceste neajunsuri, multe filtre de comutare, care le înlocuiesc doar pixelii corupți au fost propuși să atenueze problema de netezire a imaginii excesive.

Într-o familie de filtre adaptive a fost introdus ceea ce poate fi vazut ca o modificare a VMF. Design-ulse bazează pe estimarea neparametrică a distribuției de pixeliîn fereastra de filtrare și avantajul său major este că se filtrează componenta de zgomot în timp ce se adaptează la imaginea locală structuri.

De asemenea a fost propusă o schemă de schimbare bazată pe conceptul defiltrare cu vector direcșional. Neajunsul metodei îl reprezintă necesitatea optimizării ponderăriicoeficienții utilizați în algoritmul de detecție a zgomotului. Utilizândschema de prelucrare direcțională, un centru eficient de conservare a detaliilor cu filtre direcționale vectoriale a fost dezvoltat.

O altă schemă împarte pixeli din filtrarea ferestrei în două seturi. Primul constă din pixeli similari cu pixelul central al ferestrei, iar celălalt este compus din acei pixeli, care se deosebesc foarte mult în termeni de distanța euclidiană față de pixelul central. Acest concept de filtrare poate fi adoptat pentru suprimarea zgomotului de impuls, totuși are nevoie de un prag care depinde de intensitatea zgomotului și nu reușește în cazul așa-numitului zgomot "sare și piper".

În lucrarea curentă ne vom axa pe detalierea metodei descrisă în ultima idee, dezvoltând modalitățile de implementare. [21]

Definiție

Un grup de pixeli de tip peer group este definit ca setul de pixeli vecini, care sunt asemănători cu acesta în funcție de o distanță apropiată sau măsură similară. Acest concept a fost utilizat cu succes pentru a elabora algoritmipentru detectarea și suprimarea zgomotului impulsiv în imaginile de culoare gri și color. În această lucrare, voi prezenta o nouă abordare bazată pe peer group, menită să îmbunătățească compromisul între eficiența computațională și calitatea de filtrare a metodelor bazate pe grupul anterior. Eficiența calculului poate fi îmbunătățită folosind o modificare a unei abordări recente care poate fi aplicată numai atunci când se aplică măsura de distanță sau similarăfolosită îndeplinește așa-numita proprietate a inegalității triunghiulare. Îmbunătățirea calității de filtrare esterealizată prin includerea unei etape de rafinare în detectarea zgomotului. Metoda propusă efectuează în conformitate cu următorii pași: Mai întâi, împărțim imaginea în blocuri disjuncte și facem oclasificare rapidă a pixelilor în trei tipuri: necorupți, nediagnosticați și corupți. În al doilea rând,

rafinămconstatările inițiale prin analizarea pixelilor nediagnosticați și, în final, fiecare pixel este clasificat ca fiind coruptsau fără corupție. Apoi, doar pixeli corupți sunt înlocuiți, astfel încât datele de imagine necorupte sunt păstrate.Rezultatele experimentale sugerează că metoda propusă este capabilă să depășească performanțele metodelor de ultimă oră atât în ceea ce privește calitatea filtrării, cât și eficiența computațională.

Detecția și îndepărtarea pixelilor corupți

În această lucrare, reprezentăm imagini color folosind spațiul de culoare pe 8 biți RGB (Red Green Blue), astfel încât fiecare pixel $\boldsymbol{x_i}$ este o componentă vector cu trei elemente al valorilor întregului. Pentru a descrie metoda nouă numită *filtru îmbunătățit rapid grup de grup* (Improved fast peer group filter - IFPGF), vom redefini mai întâi termenul de *peer group* al unui pixel $\boldsymbol{x_i}$ încorporat într-o fereastră *W*, este definit ca un set $\mathscr{P}(\mathbf{x_i}, d) = \{\mathbf{x_j} \in W : \|\mathbf{x_i} - \mathbf{x_j}\|_2 \leqslant d\}$unde $d > 0$ și $\|\bullet\|_2$ reprezintă distanța euclidiană. Este de notat faptul că $\mathscr{P}(\mathbf{x_i}, d)$, pentru fiecare $d > 0$. Distanța Euclidiană poate fi înlocuită cu orice altă distanță sau măsură similară în aceleași condiții.

Metoda propusă este divizată într-o parte de detecție a zgomotului și alta de filtrare a acestuia. Pasul de detecție al zgomotului este împărțită în două faze.

Prima fază

În primă instanță, imaginea este partiționată în blocuri diferite. Pentru fiecare partiție *W* de dimensiune $n \times n$, unde n = 3, 5, 7, ..., centrată în $\boldsymbol{x_i}$, aplicăm următoarea procedură, având o valoare fixă pentru *m* în $\{1, 2, ..., n^2 - 1\}$:

$$\begin{aligned}
&\text{IF } \sharp\mathscr{P}(\mathbf{x_i}, d) \geqslant (m+1) \text{ } (\sharp \text{ denotes set cardinality}) \\
&\text{THEN } \forall \mathbf{x_j} \in \mathscr{P}(\mathbf{x_i}, d), \mathbf{x_j} \text{ is declared as } \textit{non-corrupted} \text{ and} \\
&\forall \mathbf{x_k} \in W - \mathscr{P}(\mathbf{x_i}, d), \mathbf{x_k} \text{ is declared as } \textit{non-diagnosed} \\
&\text{ELSE } \mathbf{x_i} \text{ is declared provisionally as } \textit{corrupted} \text{ and} \\
&\forall \mathbf{x_k} \in W - \{\mathbf{x_i}\}, \mathbf{x_k} \text{ is declared as } \textit{non-diagnosed}
\end{aligned}$$

Idee din spatele acestei proceduri este aceea că, atunci când $\mathscr{P}(\mathbf{x_i}, d)$ are cel puțin $m + 1$ membri, atunci $\boldsymbol{x_i}$ poate fi declarat drept *necorupt.* În plus, nu doar $\boldsymbol{x_i}$ ci și toți ceilalți membri care fac parte din $\mathscr{P}(\mathbf{x_i}, d)$ sunt declarați *necorupți.* De fapt, ținând seama de proprietatea triunghiulară de inegalitate, putem garanta că distanța dintre oricare doi pixeli în $\mathscr{P}(\mathbf{x_i}, d)$ va fi mai mică sau egală cu $2d$. Deci, prin selectarea unei valori potrivite pentru *d,* fiecare pixel din $\mathscr{P}(\mathbf{x_i}, d)$ poate fi declarat drept *necorupt* cu o fiabilitate destul de mare. Această idee implică o reducere importantă a calculelor, deoarece pixelii diagnosticați deja, nu mai trebuie sa fie procesați din nou.

A doua fază

Pentru a doua fază a etapei de detecție, luam în considerare o valoare a lui *m' ∈{1, ..., m - 1}*, și aplicăm următoarea procedură pentru fiecare pixel $\boldsymbol{x_i}$ *nediagnosticat* centrat în fereastra *W.*

IF $\mathscr{P}(\mathbf{x_i}, d)$ contains m' *non-corrupted* pixels
THEN $\mathbf{x_i}$ is declared as *non-corrupted*
ELSE IF $\sharp\mathscr{P}(\mathbf{x_i}, d) \geqslant (m+1)$
THEN $\forall \mathbf{x_j} \in \mathscr{P}(\mathbf{x_i}, d), \mathbf{x_j}$ is declared as *non-corrupted*
ELSE $\mathbf{x_i}$ is declared as *corrupted*

În această fază includem condiția ca $\mathscr{P}(\mathbf{x_i}, d)$ să includă *m'* vecini *necorupți* care sunt luați din pasul de rafinament (anterior). Ideea din spatele acestei condiții este că, dacă un pixel este similar cu alți pixeli *necorupți* atunci, ar trebuie să fie și el *necorupt.* Această condiție ajută la a îmbunătăți acuratețea detectării zgomotului în apropierea marginilor și a regiunilor detaliate unde *m* vecini similari sunt greu de găsit. La finalul acestei etape a procesului de decție, toți pixelii imaginii sunt clasificați în *corupți* și *necorupți*. De punctat este faptul că, anumiți pixeli, care inițial au fost declarați ca și *corupți,* pot fi rediagnosticați drept *necorupți.*

În final, zgomotul detectat este redus prin aplicarea o metodă de filtrare potrivită pentru fiecare pixel corupt. Pot fi uitlizate diferite filtre precum Arithmetic Mean Filter (AMF), numit și filtru de medie și filtru median, o altă metodă. În cazul în care fereastra de filtrare nu conține niciun pixel necorupt, atunci masca (fereastra) trebuie lărgită până când conține cel puțin un pixel necorupt.

Metoda descrisă este rezumată în algoritmul următor (Figura 22).

1 The image is partitioned into disjoint blocks W of size $n \times n, n = 3, 5, \ldots$ centered at $\mathbf{x_i}$, respectively;
2 **foreach** *Block* **do**
3 Compute $\mathscr{P}(\mathbf{x_i}, d)$;
4 **if** $\sharp\mathscr{P}(\mathbf{x_i}, d) \geqslant (m+1)$ **then**
5 $\forall \mathbf{x_j} \in \mathscr{P}(\mathbf{x_i}, d)$, $\mathbf{x_j}$ is declared as *non-corrupted*;
6 $\forall \mathbf{x_k} \in W - \mathscr{P}(\mathbf{x_i}, d), \mathbf{x_k}$ is declared as *non-diagnosed*;
7 **else**
8 $\mathbf{x_i}$ is declared provisionally as *corrupted*;
9 $\forall \mathbf{x_k} \in W - \{\mathbf{x_i}\}, \mathbf{x_k}$ is declared as *non-diagnosed*;
10 **end**
11 **end**
12 **foreach** *non-diagnosed* pixels $\mathbf{x_i}$ **do**
13 Compute $\mathscr{P}(\mathbf{x_i}, d)$;
14 **if** $\mathscr{P}(\mathbf{x_i}, d)$ contains m' *non-corrupted* pixels **then**
15 $\mathbf{x_i}$ is declared as *non-corrupted*;
16 **else**
17 **if** $\sharp\mathscr{P}(\mathbf{x_i}, d) \geqslant (m+1)$ **then**
18 $\forall \mathbf{x_j} \in \mathscr{P}(\mathbf{x_i}, d), \mathbf{x_j}$ is declared as *non-corrupted*;
19 **else**
20 $\mathbf{x_i}$ is declared as *corrupted*;
21 **end**
22 **end**
23 **end**
24 **foreach** *corrupted* pixel $\mathbf{x_i}$ centered in a filtering window W **do**
25 **if** W does not contain any *non-corrupted* pixel **then**
26 **repeat**
27 Enlarge W in size;
28 **until** W contains at least one *non-corrupted* pixel
29 **end**
30 Replace $\mathbf{x_i}$ with the output of the AMF over the *non-corrupted* pixels in W;
31 **end**

Figura 22. Peer Group Filterging Algorithm

În continuare se va încerca o implementare cât mai optimă a algoritmului descris mai sus, în vederea îmbunătățirii vizibile a imaginilor corupte de diferite tipuri de zgomot. Algoritmul de peer group va fi aplicat atât pentru imagini alb-negru, cât și color.

Capitolul 5. Descrierea soluției

În această lucrare este prezentată o nouă abordare a problemei reducerii zgomotului impulsiv în imaginile color și alb-negru. Tehnica propusă se bazează pe evaluarea proprietăților statistice ale unei secvențe sortate de distanțe acumulate utilizate pentru calculul median al vectorului. Detectarea pixelilor corupți se efectuează folosind tehnica peer group care lucrează pe distanțele agregate atribuite fiecăruia din pixeli din fereastra de filtrare. Scopul discernământului este de a împărți pixelii în două clase: un grup de pixeli asemănători cu pixelul central și un set de pixeli care constau în valori extreme în imagine, prin procesul de zgomot. Filtrul descris permite detectarea fiabilă a impulsurilor și datele sale de ieșire comută între mediana vectorului și pixelul original, neperturbat.

Un peer group al unei imagini este definit ca setul de pixeli vecini, care sunt asemănători cu acesta în funcție de un pixel în funcție de distanță. Acest concept a fost utilizat cu succes pentru a elabora algoritmi pentru detectarea și suprimarea zgomotului impulsiv în imaginile de culoare gri și color. În această lucrare, este prezentată o nouă abordare bazată pe grupuri interumane, menită să îmbunătățească compromisului între eficiența computațională și calitatea de filtrare a metodelor bazate pe grupul anterior.

Metoda propusă efectuează în conformitate cu următorii pași. Mai întâi, împărțim imaginea în blocuri disjuncte și facem o clasificare rapidă a pixelilor în trei tipuri: necorupți, nediagnosticați și corupți. În al doilea rând, rafinăm constatările inițiale prin analizarea pixelilor nediagnosticați și, în final, fiecare pixel este clasificat ca fiind corupt sau fără corupție. Apoi, doar pixelii corupți sunt înlocuiți, astfel încât datele de imagine necorupte sunt păstrate.

Structura proiectului:

În primă instanță este efectuată citirea imaginii cu ajutorul funcției din OpenCV, *imread()* ce se dorește a fi prelucrată (Figura 23). Au fost citite două imagini pentru a ajuta la afișare. Funcția *cvtColor()* convertește o imagine de la un spațiu de culoare la altul. În cazul nostru, se face conversia de la BGR (biții sunt în ordine inversă), care este formatul implicit în OpenCV, la BGR. Astfel, culorile imaginilor vor fi cele reale.

```
img = cv2.imread(r'C:\Users\diana\OneDrive\Desktop\Facultate\Python\Licenta\image_denoising-1.png')
img2 = cv2.imread(r'C:\Users\diana\OneDrive\Desktop\Facultate\Python\Licenta\image_denoising-1.png')

img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)
```

Figura 23. Citirea imaginilor

Mai departe sunt preluate lungimea și lățimea imaginii, care vor fi folosite pe tot parcursul proiectului. Acest lucru este posibil uitlizând proprietatea *shape,* care este un vector. Primul element al său reprezintă înălțimea, iar cel de-al doilea lățimea (Figura 24).

```
h = img.shape[0]
w = img.shape[1]
```

Figura 24. Preluarea lungimii și lățimii

Urmează inițializările aferente algoritmului. În urma testelor am constatat că acestea sunt cele mai apropiate valori care ar trebui folosite, pentru d = 35, n = 3, m = 2, m_prim = 1. Adițional, am definit trei liste, care stochează pixelii necorupți (non_corrupted), corupți (corrupted) și nediagnosticați (non_diagnosed). De asemenea, am utilizat și trei dicționare, unul care stochează pentru fiecare pixel, pixelul din centrul ferestrei de filtrare (dictMF), unul care salvează pentru fiecare pixel, valoarea cu care va fi înlocuit, adică cea mediană (dictXi). Pentru fiecare peer group, pixelii sunt adăugați într-un vector, care va fi sortat. Valoarea din mijlocul listei va fi mediana grupului. Cel de-al treilea dicționar reține numărul de pixeli necorupți din peer group (Figura 25).

```
d = 35
n = 3
m = 2
m_prim = 1
non_corrupted = list()
non_diagnosed = list()
corrupted = list()
dictMF = {}
dictXi = {}
dictNrNonCor = {}
```

Figura 25. Inițializări

Mai departe, este definită funcția care creează și returnează un peer group. Aceasta ia ca parametri, pixelul central, distanța *d* și dimensiunea ferestrei *n.* În cadrul acestei proceduri se parcurge fereastra de filtrare pixel cu pixel și, daca distanța euclidiană ale valorilor intensităților culorilor RGB este mai mică sau egală cu distanța prestabilită *d,* atunci pixelul respectiv este asignat grupului. Tot aici se realiză și cate o listă pentru fiecare culoare, Red, Green, Blue. La final sunt sortate aceste liste și se alege din fiecare listă valoarea mediană și se adaugă dicționarului dictXi, pentru a ști cu ce valoarea vom înlocui pixelii corupți (Figura 26).

```
def peerGroup(xi, d, n):
    dictNrNonCor[str(xi[0]) + str(xi[1])] = 0
    P = list()# peer group
    R = list()# Red
    G = list()# Green
    B = list()# Blue
    for i in range(xi[0] - 1, xi[0] + n - 1):
        for j in range(xi[1] - 1, xi[1] + n - 1):
            R.append(img[i, j][0])
            G.append(img[i, j][1])
            B.append(img[i, j][2])
            dictMF[str(i)+str(j)] = str(xi[0]) + str(xi[1])
            if ((xi[0], xi[1])!=(i, j)):
                euclidian_dist = math.sqrt(sum([(int(a) - int(b)) ** 2 for a, b in zip(img[xi[0], xi[1]], img[i, j])]))
                if euclidian_dist <= d:
                    P.append((i, j))
    R.sort()
    G.sort()
    B.sort()
    dictXi[str(xi[0]) + str(xi[1])] = (R[int(n*n/2)], G[int(n*n/2)], B[int(n*n/2)])
    return P
```

Figura 26. Peer Group

Urmează definirea procedurii *diagnose* care oferă o primă diagnosticare a pixelilor. Pentru fiecare peer group, în cazul în care lungimea grupului respectiv este mai mare decât *(m + 1),* atunci pixelul central al ferestrei este declarat nediagnosticat, urmând a fi analizat la pasul următor, iar restul pixelilor din grup sunt clasificați drept necorupți. În cazul în care în peer group sunt mai puțin de (m + 1) pixeli, atunci cel central va fi corupt, iar ceilalți nediagnosticați (Figura 27).

```
def diagnose(P, xi, n, corrupted, non_corrupted, non_diagnosed):

    if len(P) >= (m + 1):
        for i in range(xi[0] - 1, xi[0] + n - 1):
            for j in range(xi[1] - 1, xi[1] + n - 1):
                if ((xi[0], xi[1]) != (i, j)):
                    if (i, j) in P:
                        non_corrupted.append((i, j))
                        dictNrNonCor[str(xi[0]) + str(xi[1])] += 1
                    else:
                        non_diagnosed.append((i, j))
    else:
        corrupted.append(xi)
        img[xi[0], xi[1]][0] = dictXi[dictMF[str(xi[0]) + str(xi[1])]][0]
        img[xi[0], xi[1]][1] = dictXi[dictMF[str(xi[0]) + str(xi[1])]][1]
        img[xi[0], xi[1]][2] = dictXi[dictMF[str(xi[0]) + str(xi[1])]][2]
        for i in range(xi[0] - 1, xi[0] + n - 1):
            for j in range(xi[1] - 1, xi[1] + n - 1):
                if ((xi[0], xi[1]) != (i, j)):
                    if (i, j) not in P:
                        non_diagnosed.append((i, j))
```

Figura 27. Prima diagnoză

După prima fază de diagnosticare, urmează *rediagnose,* care va face clasificarea pixelilor declarați nediagnosticați la pasul anterior. Acest lucru se face astfel. Pentru fiecare pixel nediagnosticat, se creează peer grupul. Dacă numărul de pixeli necorupți din grup este egal cu *m_prim,* atunci pixelul central este necorupt. Dacă sunt mai mult de *m_prim* pixeli necorupți, restul pixelilor din fereastră vor fi și ei necorupți. Iar dacă numărul de pixeli necorupți este mai mic decât *m_prim*, atunci pixelul central va fi corupt (Figura 28).

```
def rediangose(corrupted, non_corrupted, non_diagnosed):
    for x in non_diagnosed:
        xi = x
        P = peerGroup(xi, d, n)
        if dictNrNonCor[str(xi[0]) + str(xi[1])] == m_prim:
            non_corrupted.append(xi)
        elif len(P) >= (m + 1):
            for i in range(xi[0] - 1, xi[0] + n - 1):
                for j in range(xi[1] - 1, xi[1] + n - 1):
                    if ((xi[0], xi[1]) != (i, j)):
                        if (i, j) in P:
                            non_corrupted.append((i, j))
        else:
            corrupted.append(xi)
            img[xi[0], xi[1]][0] = dictXi[dictMF[str(xi[0]) + str(xi[1])]][0]
            img[xi[0], xi[1]][1] = dictXi[dictMF[str(xi[0]) + str(xi[1])]][1]
            img[xi[0], xi[1]][2] = dictXi[dictMF[str(xi[0]) + str(xi[1])]][2]
```

Figura 28. Rediagnosticarea

După ce am definit și construit tot ceea ce aveam nevoie pentru compilarea soluției, se parcurge imaginea pe blocuri de dimensiune n × n. Parcurgerea se face pe baza pixelului aflat în centrul ferestrei de filtrare. Pentru fiecare pixel se creează peer grupul aferent și o primă diagnosticare (Figura 29).

```
for i in range(0, int(h) - n, n):
    for j in range(0, int(w) - n, n):
        xi = (i, j)
        P = peerGroup(xi, d, n)
        diagnose(P, xi, n, corrupted, non_corrupted, non_diagnosed)
```

Figura 29. Parcurgerea pe blocuri

La pasul următor se face o diagnosticare puțin mai amănunțită. Pentru fiecare pixel corupt se verifică dacă grupul din care face parte conține pixeli necorupți sau nu. În cazul în care nu avem pixeli necorupți în fereastra respectivă, atunci dimensiunea blocului se mărește cu 2. Acest lucru se va efectua până când măcar un poxel necorupt aparține grupului respectiv (Figura 30).

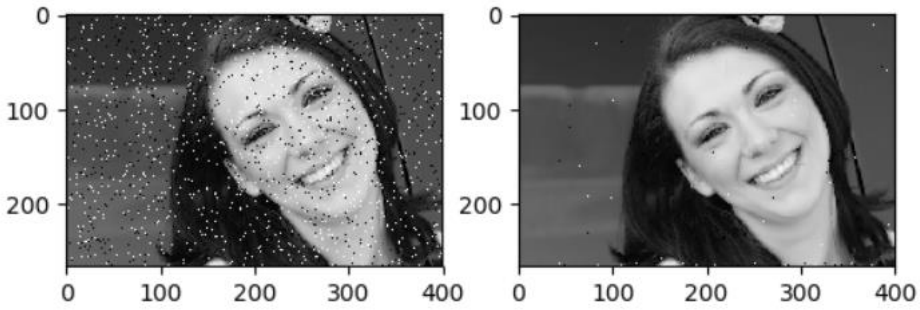
```
for xi in corrupted:
    nn = n
    while (dictNrNonCor[str(xi[0]) + str(xi[1])] == 0) and nn < h:
        print(xi)
        nn += 2
        if (xi[0] + nn < h) and (xi[1] + nn < w):
            P = peerGroup(xi, d, nn)
        print(nn)
        diagnose(P, xi, nn, corrupted, non_corrupted, non_diagnosed)
```

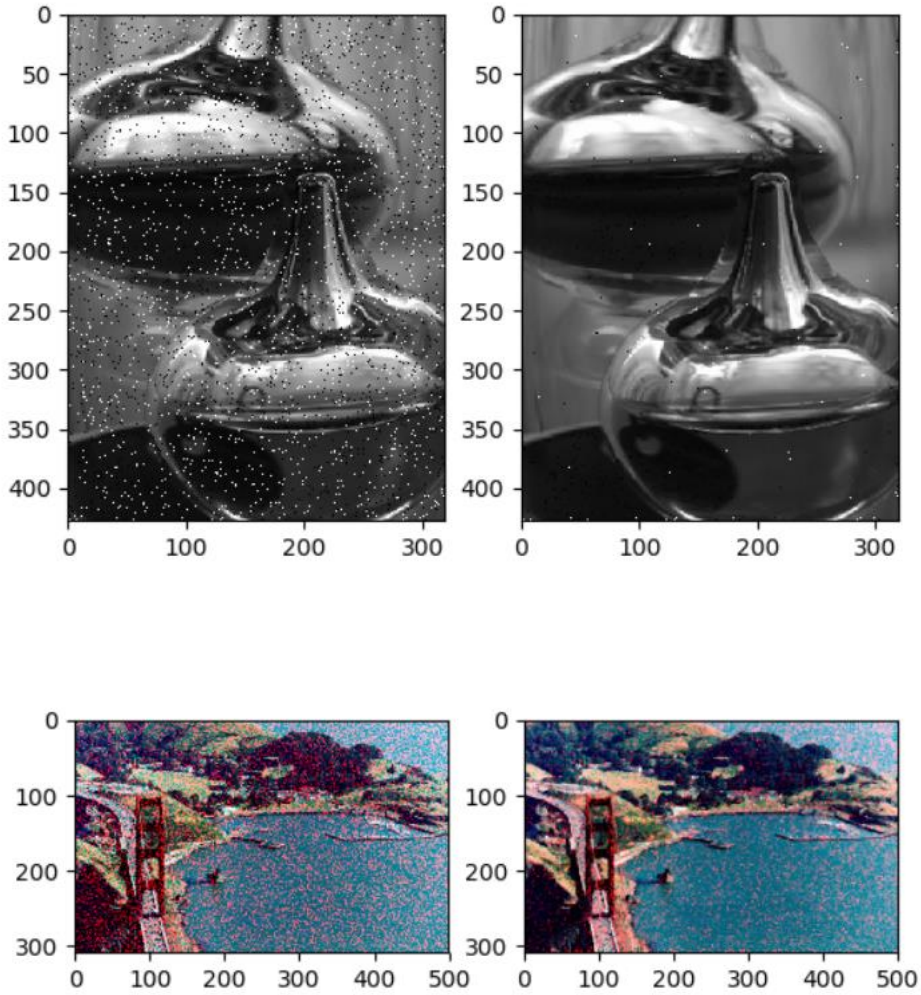
Figura 30. Lărgirea ferestrei

La final, se apelează funcția *rediagnose* pentru a se efectua clasificarea finală a pixelilor. Astfel, la final, toți pixelii vor fi clasificați drept corupți, necorupți. Asignarea noii valori pixelilor corupți se face tot în cadrul acestei proceduri.

Rezulatele implementării

Mai jos se pot observa câteva rezultate ale implementării prezentate. Testele au fost făcute pe imagini alb-negru sau color corupte fie de zgomot de tip sare și piper, fie Gaussian.





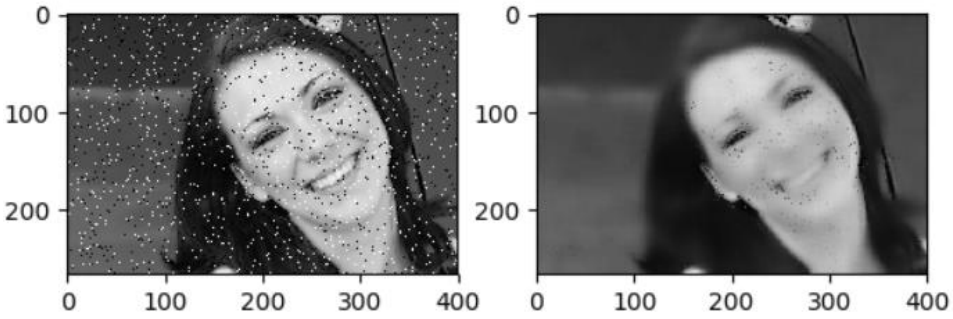
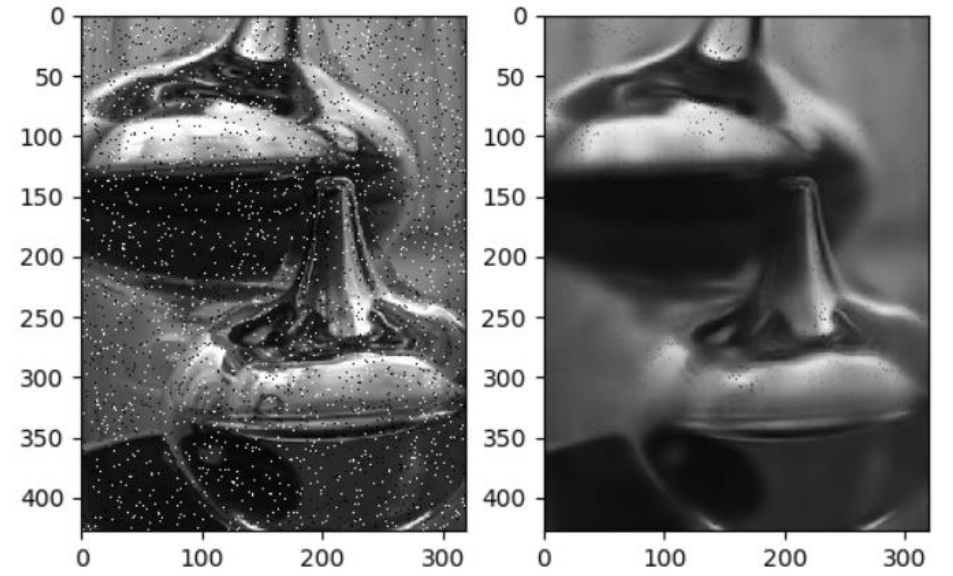
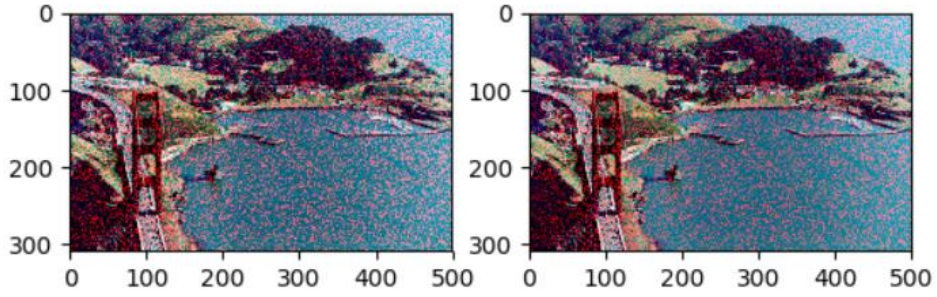
În partea stângă se află imaginea coruptă, iar în dreapta, rezultatul după reducerea zgomotului. Primele două imagini sunt afectate de zgomot de tip sare și piper, iar cea de-a treia de zgomot de tip Gaussian. După cum se observă performanța algoritmului implementat este mai bună în cazul coruperii cu zgomot de tip sare și piper.

Rezultate OpenCV

În cadrul OpenCV există câteva funcții predefinte care realizează reducerea zgomotului în imagini. În continuare vom vedea efectele acestor proceduri, care sunt în număr de două.

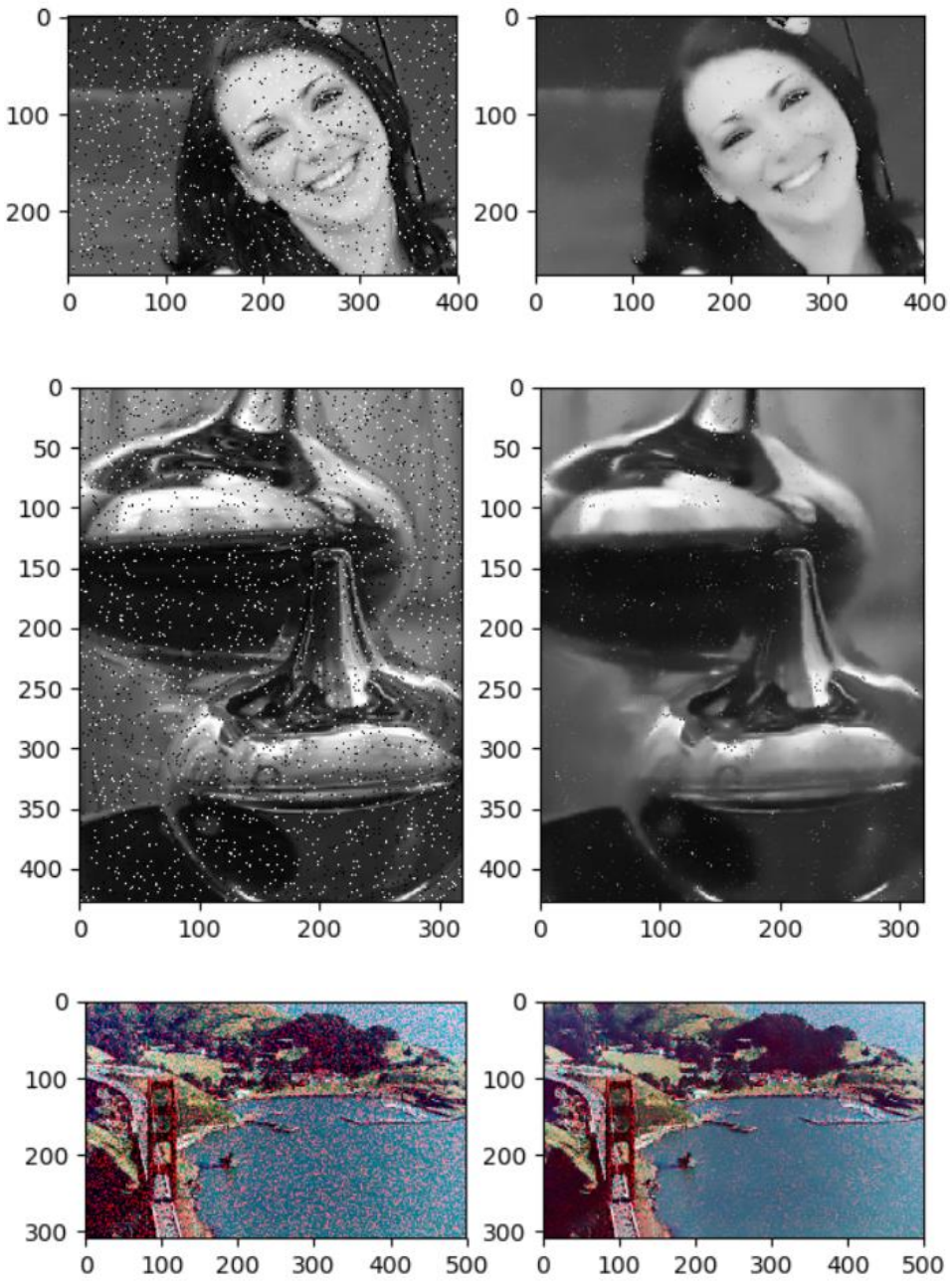
1. cv2.fastNlMeansDenoisingColored()

   Este folosit pentru a înlătura zgomotul (în special Gaussian) din imaginile color.

2. cv2.fastNlMeansDenoising()

Este folosit pentru a înlătura zgomotul din imaginile alb-negru.



Concluzii

Principala provocare în procesarea digitală a imaginilor este eliminarea zgomotului din imaginea originală. În această lucrare au fost analizați algoritmii de reducere a zgomotului dintr-o imagine coruptă existenți și efectuează studiul comparativ. Diferite modele de zgomot, inclusiv aditiviar tipurile de multiplicare sunt discutate în document. Alegerea algoritmului de netezire a imaginii depinde de aplicație. Prin urmare,este necesar să avem cunoștințe despre zgomotul prezent în imagine pentru a selecta algoritmul adecvat.

Imaginile digitale joacă un rol foarte important în rutina zilnică, cum ar fi cele utilizate în televiziunea prin satelit, trafic inteligent monitorizarea, recunoașterea scrisului de mână pe verificări, validarea semnăturii, imagistica prin rezonanță pe calculator și în domeniul cercetării și tehnologie, cum ar fi sistemele de informații geografice și astronomie. În imagistica digitală, tehnicile de achiziție și sistemele introduc diferite tipuri de zgomote și artefacte.

Reducerea zgomotului este mai importantă decât orice alte sarcini din procesarea imaginii, analiză și aplicații. Rezervă detaliile unei imagini și elimină zgomotul aleator pe cât posibil. Pe lângă faptul că imaginea zgomotoasă generează o calitate vizuală nedorită, ea scade și vizibilitatea obiectelor cu contrast scăzut. Prin urmare, eliminarea zgomotului este esențială în aplicațiile de imagini digitale, în scopul de a îmbunătăți și recuperați detaliile fine care sunt ascunse în date

În documentul prezentat s-a dezvoltat în mod principal implementarea tehnicii de peer group în vederea detecției pixelilor corupți. Acest lucru presupune împărțirea imaginii în blocuri de dimensiune n × n și efectuarea unei grupări pentru fiecare fereastră. Pixelii luați în calcul au fost cei care aveau distanța euclidiană a intensității valorilor RGB, mai mică decât un parametru predefinit de noi. Detecția a fost făcută în două etape , astfel încât, la final, toți pixelii să fie clasificați drept corupți sau necorupți. După împărțirea corespunzătoare, pentru pixelii corupți se aplică filtrul median, care presupune ordonarea valorilor intensităților pixelilor din peer group și alegerea celui din mijloc, numit și mediană.

Prin mijloace descrise și dezvoltate s-a ajuns la o soluție finală ce efectuează depistarea și reducerea zgomotului din imaginile corupte.

Bibliografie

[1] http://www.programare.ase.ro/AN/20152016/4.metode%20de%20restaurare%20-%20partea%20I.pdf

[2] http://www.etti.tuiasi.ro/documents/teze_doctorat/2010/rezumat_Ungureanu.pdf

[3] https://ai.stanford.edu/~syyeung/cvweb/tutorial1.html

[4] https://www.tutorialspoint.com/dip/concept_of_pixel.htm

[5] http://geek.m3d1a.ro/2013/07/ce-sunt-pixelii.html

[6] https://en.wikipedia.org/wiki/Grayscale

[7] https://ro.wikipedia.org/wiki/Modelul_de_culoare_CMYK

[8] https://ro.wikipedia.org/wiki/Pixel

[9] https://ro.wikipedia.org/wiki/Rezolu%C8%9Bie_digital%C4%83

[10] https://ro.wikipedia.org/wiki/Pixel

[11] https://www.cambridgeincolour.com/tutorials/image-noise.htm

[12] http://users.utcluj.ro/~igiosan/Resources/PI/L10/PI-L10e.pdf

[13] Julliand, T., Nozick, V., & Talbot, H. (2015, October). Image noise and digital image forensics. In *International Workshop on Digital Watermarking* (pp. 3-17). Springer, Cham.

[14] Boyat, A. K., & Joshi, B. K. (2015). A review paper: Noise models in digital image processing. *arXiv preprint arXiv:1505.03489*.

[15] Image Denoising Techniques: A Review, Sandeep Kaur, Navdeep SinghResearch Scholar, Master of Technology, Department of Computer Engineering, Punjabi University, Patiala ,India Assistant Professor, Department of Computer Engineering, Punjabi University, Patiala, India

[16] https://homepages.inf.ed.ac.uk/rbf/HIPR2/mean.htm

[17] https://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm

[18] Dixit, S., & Nagaria, D. (2017). LMS Adaptive Filters for Noise Cancellation: A Review. *International Journal of Electrical & Computer Engineering (2088-8708)*, *7*(5).

[19] http://www.rroij.com/open-access/image-denoising-techniques-a-review-.php?aid=46252

[20] Kenney, C., Deng, Y., Manjunath, B. S., & Hewer, G. (2001). Peer group image enhancement. *IEEE Transactions on Image Processing*, *10*(2), 326-334.

[21] Camarena, J. G., Gregori, V., Morillas, S., & Sapena, A. (2010). Some improvements for image filtering using peer group techniques. *Image and Vision Computing*, *28*(1), 188-201.

[22] Smolka, B. (2010). Peer group switching filter for impulse noise reduction incolor images. *Pattern Recognition Letters*, *31*(6), 484-495.