**Nume student: Trandafir Irina, grupa 1130IE**

**Proiect etapa II**

o Enunțul preluat din lista de teme, împreună cu numărul temei.

Pentru etapa a II-a a proiectului am ales tema:

9. Simularea zgomotului unimodal. Filtre de ordine minim și maxim pentru eliminarea zgomotului unimodal.

In cele ce urmeaza voi detalia mai intai functia pentru obtinerea zgomotului unimodal, iar mai apoi functia pentru filtrul de ordine minim/maxim.

o Date de intrare necesare și rezultate obținute.

**nume\_poza** - numele pozei careia urmeaza sa ii fie adaugat zgomotul

**tip\_zgomot** - **0** pentru **piper**, orice **alta valoare** pentru **sare**

**p** - probabilitatea in care trebuie sa se regaseasca un pixel pentru a fi alterat, ia valori intre **0** si **1**

Am apelat functia pentru obtinerea zgomotului unimodal pe imaginea cu Lena, parametrul tip\_zgomot avand valoarea 1 (sare).

A person wearing a hat

Description automatically generated

Figură Lena.png

A person wearing a hat

Description automatically generated

Figură Lena sare zg.png

Apoi am apelat functia pentru obtinerea zgomotului unimodal pe imaginea cu Lena, parametrul tip\_zgomot avand valoarea 0 (piper).

A person wearing a hat

Description automatically generated

Figură Lena piper zg.png

o Descrierea algoritmului utilizat, în pseudocod.

im ← imaginea citita

Afisare imagine

m, n ← dimensiunile imaginii

Pentru fiecare pixel (i,j)

Genereaza r din distributia uniforma intre 0 si 1

Daca r < p

Daca tipul\_zgomotului e de tip piper

im(i,j) ← 0

Altfel

im(i,j) ← 255

Afisare imagine

Daca tip\_zgomot e de tip piper

Scrie imagine cu titlul numele primit ca parametru concatenat cu sirul „piper zg.”

Altfel

Scrie imagine cu titlul numele primit ca parametru concatenat cu sirul „sare zg.”

Citesc imaginea si apoi o afisez. In variabilele m si n stochez numarul de linii si coloane ai matricei de pixeli a imaginii citite anterior. Pentru fiecare pixel in parte generez o probabilitate. Daca probabilitatea respectiva se incadreaza in probabilitatea primita ca parametru, atunci alterez pixelul respectiv. Daca zgomotul e de tip piper atunci pixelii alterati devin negrii, altfel daca zgomotul e sare, pixelii devin albi.

Afisez imaginea cu zgomot unimodal, iar apoi o salvez concatenand numelui pozei cu tipul de zgomot in functie de valoarea parametrului tip\_zgomot.

o Exemple de apel.

zgomot\_unimodal('Lena', 'png', 0, 0.1);

Am incercat si alta modalitate de generare a zgomotului unimodal.

o Date de intrare necesare și rezultate obținute.

**nume\_poza** - numele pozei careia urmeaza sa ii fie adaugat zgomotul

**tip\_zgomot** - **0** pentru **piper**, orice **alta valoare** pentru **sare**

**densitate** - nr de pixeli alterati in medie, ia valori intre **0** si **1**

Am apelat functia pentru obtinerea zgomotului unimodal pe imaginea cu Lena, parametrul tip\_zgomot avand valoarea 0 (piper). Zgomotul nu e asa puternic ca in varianta anterioara.

A person wearing a hat

Description automatically generated

Figură 3 Lena piper zg.png

o Descrierea algoritmului utilizat, în pseudocod.

im ← imaginea citita

Afisare imagine

m, n ← dimensiunile imaginii

ind\_pixeli ← d\*m\*n numere intregi aleatoare in intervalul [0, m\*n] indicii pixelilor ce vor fi alterati

Pentru fiecare index i de la 1 la d\*m\*n

l, c ← ind2sub([m,n], ind\_pixeli(i)) transformarea indicilor in linii si coloane

Daca tipul\_zgomotului e de tip piper

im(i,j) ← 0

Altfel

im(i,j) ← 255

Afisare imagine

Daca tip\_zgomot e de tip piper

Scrie imagine cu titlul numele primit ca parametru concatenat cu sirul „densitate piper zg.”

Altfel

Scrie imagine cu titlul numele primit ca parametru concatenat cu sirul „densitate sare zg.”

Citesc imaginea si apoi o afisez. In variabilele m si n stochez numarul de linii si coloane ai matricei de pixeli a imaginii citite anterior. Generez d\*m\*n numere intregi aleatoare din intervalul [0, m\*n]. Aceste numere reprezinta indicii pixelilor ce vor fi alterati. Daca zgomotul e de tip piper atunci pixelii alterati devin negrii, altfel daca zgomotul e sare, pixelii devin albi.

Afisez imaginea cu zgomot unimodal, iar apoi o salvez concatenand numelui pozei cu sirul „densitate” si tipul de zgomot in functie de valoarea parametrului tip\_zgomot.

o Exemple de apel.

zgomot\_unimodal\_densitate('Lena', 'png', 0, 0.1);

Acum voi detalia functia pentru filtrul de ordin minim sau maxim.

o Date de intrare necesare și rezultate obținute.

**nume\_poza** - numele pozei ce va fi prelucrata

**tip** - tipul pozei ce va fi prelucrata

**minSauMax** - **0** pentru filtru de ordine **minim**, orice **alta valoare** pentru

filtru de ordine **maxim**

**d** - dimensiunea filtrului

Am apelat functia pentru filtrul de ordin maxim (minSauMax = 1) pe imaginea ‚Lena piper zg.png’.

A person wearing a hat

Description automatically generated

Figură Lena piper zg.png

A person wearing a hat

Description automatically generated

Figură Lena piper zg restaurata.png

Am apelat functia pentru filtrul de ordin minim (minSauMax = 0) pe imaginea ‚Lena sare zg.png’.

A person wearing a hat

Description automatically generated

Figură Lena sare zg.png

A person wearing a hat

Description automatically generated

Figură Lena sare zg restaurata.png

o Descrierea algoritmului utilizat, în pseudocod.

im ← imaginea citita

m, n ← dimensiunile imaginii

Afisare imagine

t ← (d+1)/2 numarul de linii si coloane cu care trebuie bordata imaginea

l ← m+d-1 nr linii imagine extinsa

c ← n+d-1 nr coloane imagine extinsa

ime ← matrice plina cu zerouri de dimensiuni l\*c va fi imaginea bordata

f ← matrice plina cu zerouri de dimensiuni l\*c va fi imaginea prelucrata

ime(t:m+t-1,t:n+t-1,:) ← im(:, :) copierea imaginii in centrul matricii cu zerouri

Pentru fiecare pixel (i,j)

subim ← ime(i:i+d-1, j:j+d-1) filtrul in sine

elem ← matricea subim liniarizata

Daca filtrul e de minim

minEl ← elementul minim din vectorul elem

f(i+t+1, j+t+1) ← minEl

Altfel

maxEl ← elementul maxim din vectorul elem

f(i+t+1, j+t+1) ← maxEl

im ← f(t:m+t-1, t:n+t-1,:) convertita la uint8 pentru ca functia zeros intoarce double

Afisare imagine im

Citesc imaginea si apoi o afisez. In variabilele m si n stochez numarul de linii si coloane ai matricei de pixeli a imaginii citite anterior. In variabila t stochez numarul de linii si coloane cu care voi bordui imaginea pentru a o extinde pentru a aplica filtrul de ordin minim sau maxim. In l si c stochez dimensiunile imaginii extinse. Variabila ime este imaginea extinsa (mai intai creez o matrice de dimensiuni m+d-1\*n+d-1 umpluta cu 0, iar apoi incepand cu coltul de coordonate (t,t) si terminand cu coltul (m+t-1, n+t-1) copiez pixelii din imaginea originala). In f voi stoca imaginea restaurata.

Pentru fiecare pixel creez o matrice denumita subim, care este o matrice care are in centru pixelul caruia ii voi asigna valoarea rezultat. Matricea subim incepe de la coordonatele i si j si are d linii si d coloane. Liniarizez matricea si o stochez in vectorul elem. Apoi verific tipul de filtru de ordine. Daca minSauMax este 0, am de-a face cu un filtru de ordine minim si atunci gasesc elementul cu valoare minima din vectorul elem si il asignez pixelului de pe pozitia din mijlocul matricii subim din matricea f (care va contine imaginea rezultat). Daca am de-a face cu un filtru de ordine maxim, tot ce se schimba e faptul ca gasesc elementul cu valoare maxima din vectorul elem. Imaginea prelucrata se regaseste in matricea f incepand cu coltul de coordonate (t,t) si se termina cu coltul de coordonate (m+t-1, n+t-1). Fiindca functia zeros returneaza valori de tip double (f a fost initializata folosind functia zeros), imaginea trebuie convertita la uint8. Rezultatul final il afisez si il salvez.

o Exemple de apel.

filtru\_ordine('Lena piper zg', 'png', 1, 3);