Șerban Mădălina-Camelia, 1115

**Proiect pentru disciplina Tehnici de Prelucrare a Imaginilor**

**Partea 1**

**6. Implementarea procedurii “unsharp masking”. 1p**

Procedura „unsharp mask” este utilizată pentru a îmbunătăți o imagine, prin evidențierea texturii și a detaliilor. Imaginea este îmbunătățită prin extragerea unei variante nivelate din imaginea originală și adăugarea rezultatului obținut la imaginea originală.

function [] = unsharp\_mask (nume,alpha,tip)

%filtru unsharp masking pentru prelucrarea unei imagini grayscale/RGB

%I: nume - numele fișierului cu imaginea de prelucrat

% alpha - valoare care definește transparența

% are valori [0,1]

% 0,valori apropiate de 0-evidențierea mai multor detalii

% 1,valori apropiate de 1-detaliile sunt evidențiate

% tip - tip fișier pentru imaginea rezultată

%E: -

%încărcare imagine

poza=imread(nume);

[m,n,p]=size(poza); %[nr linii, nr coloane, nr plane]

%aplicare filtru pe fiecare plan al imaginii

R=zeros(m,n,p); %R-imaginea filtrată

for k=1:p

h=fspecial('unsharp',alpha);

R(:,:,k)=imfilter(poza(:,:,k),h);

endfor

R=uint8(R);

%afișare imagine originală și filtrată

figure

subplot(1,2,1), imshow(poza), title('Imaginea originală');

subplot(1,2,2), imshow(R), title('Imaginea sharpened');

fi=[nume num2str(alpha) '-UMf.' tip];

imwrite(R,fi,tip);

end

%Exemple de apel:

%unsharp\_mask('dog.jpg',1,'jpg');

%unsharp\_mask('dog.jpg',0,'jpg');

%unsharp\_mask('leaves.jpg',0,'jpg');

%unsharp\_mask('leaves.jpg',1,'jpg');

Descrierea algoritmului utilizat:

* Încărcarea imaginii originale și alocarea spațiului necesar
* Aplicarea filtrului pe fiecare plan al imaginii:

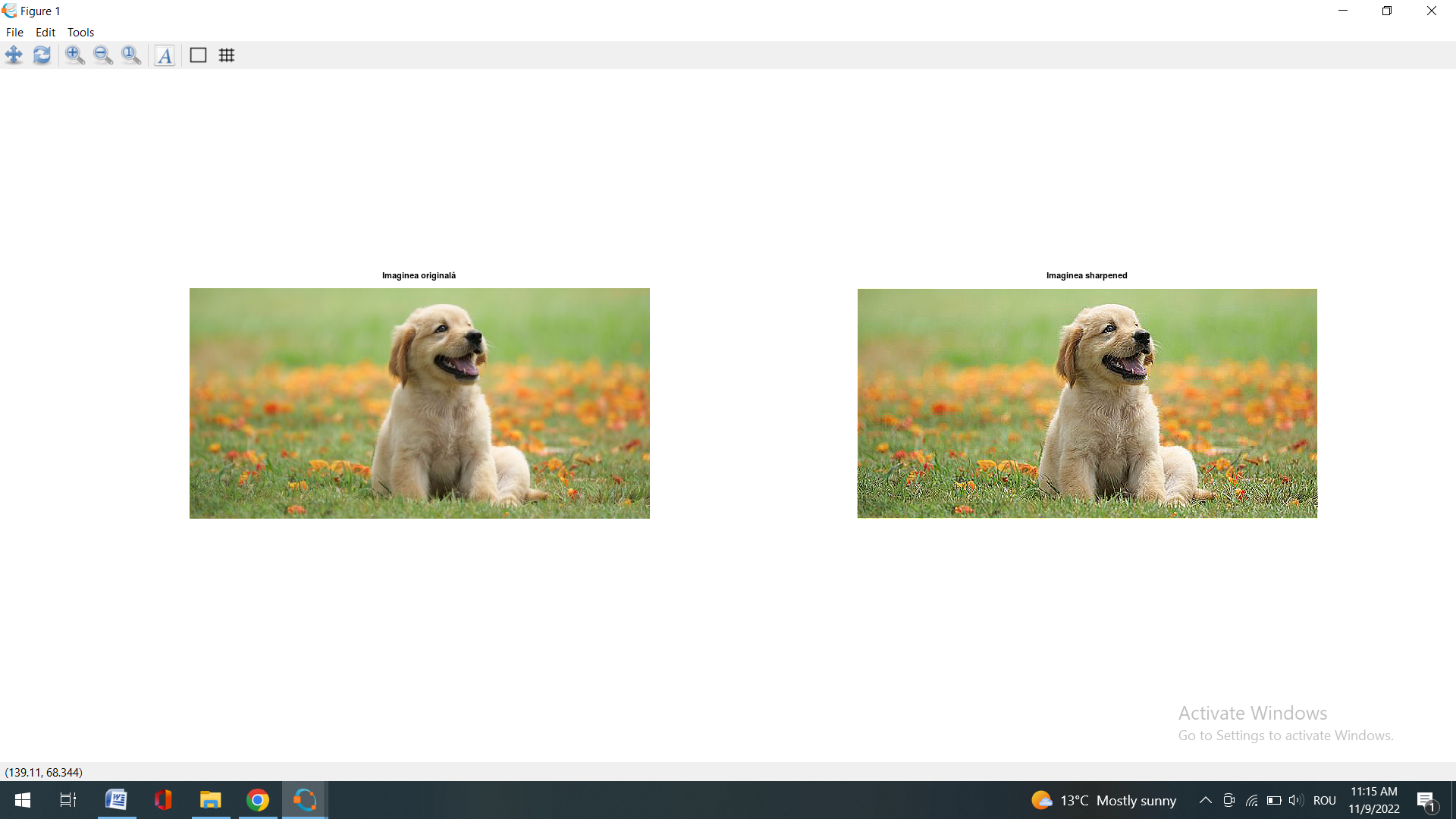
1. definirea unei imagini care va fi filtrată; are aceleași dimensiuni ca imaginea originală
2. aplicarea filtrului pe fiecare plan al imaginii
3. transformarea imaginii filtrate în int8, deoarece s-a lucrat în double

* afișarea celor două imagini (originală și filtrată):

1. definirea figurilor unde se vor afișa imaginile
2. crearea numelui fișierului pentru imaginea filtrată
3. scrierea imaginii filtrate

Rezultate obținute:

Imagine RGB utilizând alpha=0



Imagine grayscale utilizând alpha=0

