ADOBI Photo Editor

By Frîncean Antonio-Andrei

Teacher: Mihaela Cîrlugea

Second Year

ETTI_2021

INDEX

Introduction

MATLAB, short for "Matrix Laboratory," started as a simple interactive matrix calculator built in Fortran in the late 1970s. It was later commercialized in 1984 as a programming language, reimplemented in C and enhanced with user functions, toolboxes, and graphics. Over the years, various new data types, structures, and toolboxes were added, and it has been continuously updated with enhancements to its computing environment, including support for parallel computing and GPUs. Today, there are over 60 toolboxes available for specialized technical fields.

About the project

The theme of the project is image processing and editing with the help of MATLAB. The created interface allows the adjustment of certain parameters

that allow the manipulation of a picture in jpg format, at the same time offering advantages such as a music player and exposure histogram.

Theoretical presentation

As I presented before, the theme of the project is photo manipulation. The created interface allows importing an image in .jpg format, and then editing it using the buttons and the slider in the interface.

- The **UPLOAD IMAGE** button allows you to import the image from the file
- The **B&W** button uses predefined function in matlab to turn the image into black and white ("*im2gray*")
- The **HUE+** and **HUE-** buttons adjust the level of green and purple in the image("*hsvImage = rgb2hsv(rgbImage);*

  *hsvImage(:,:,1) = mod(hsvImage(:,:,1)*k,1);*")

- The **RESET** button resets the image to the initial one
- The **EXPOSURE+** buttons and **EXPOSURE-** changes the exposure level of the image made with a simple sum("*I = currentImage ;*

  I1 = I+20 ; ")

- The **HISTOGRAM** button generates a histogram of the image that is first converted to black and white, and then displays the exposure histogram using a predefined function in MATAB
- The **CONTRAST slider** adjusts the contrast level in the image("*contrast_image = imadjust(currentImage, [], [], contrastValue);*")
- The right side of the interface we have **2 secret buttons**
- The two other buttons **PLAY MUSIC** & **STOP MUSIC** that allow starting and stopping a song preset by the developer.

EXPERIMANETAL PART

The UI:

Fig1

The UI created is presented by the picture in Fig1, where all the buttons are presented.

To start using the app we have to click on "Upload IMG" button.

Fig 2

Then we will choose an image in jpg format from the computer.

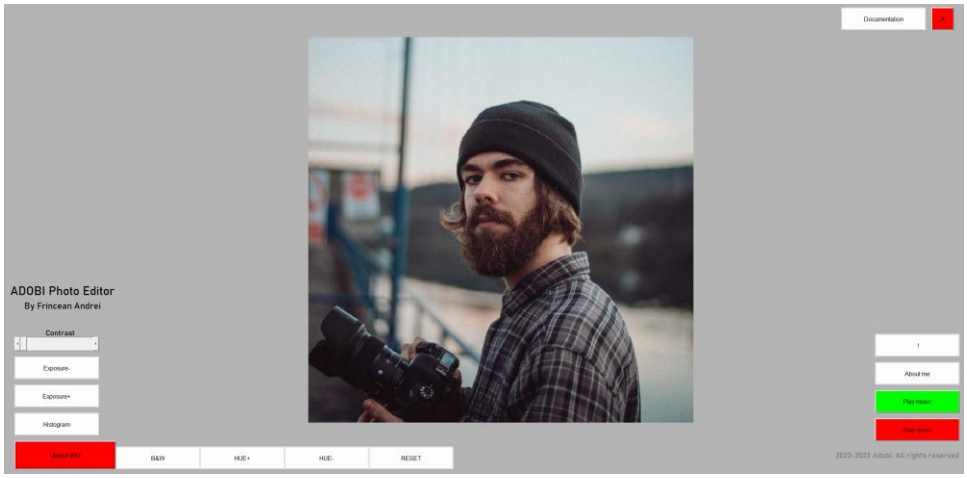


Fig3

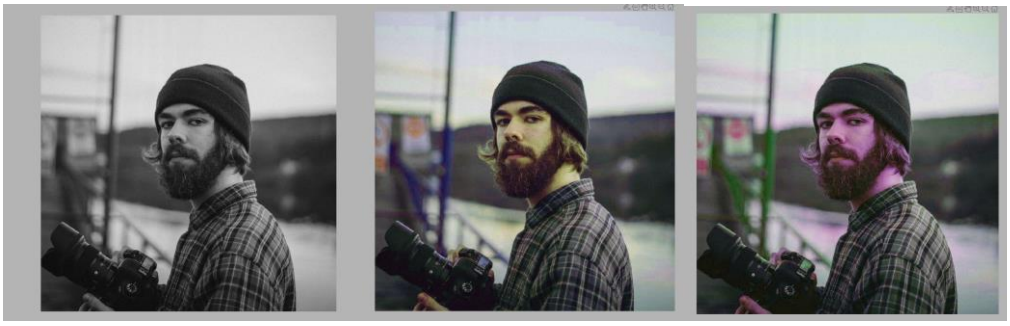
After we import the desired image from the computer, this is how the interface will look, with the imported image centered in the middle of the application screen, as presented in Fig3.

Great!! We can start editing the imported image!

The first thing we can do to change the image is to interact with the row of buttons at the bottom of the interface:

Fig4

The buttons shown above change the color of the image. The figures below show the results of each button (the effects can be combined with each other).



B&W HUE + HUE-

Also, the RESET button does...exactly what you expect.

Resets the image to the initial one, preset with its import.

On the left side of the interface we have the buttons that control the brightness of the image, the slider for contrast and the button for displaying the histogram of the image, as presentet in Fig5.

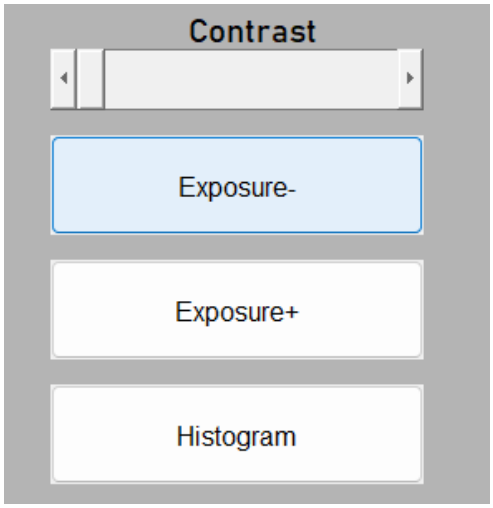


Fig5

The EXPOSURE+ and EXPOSURE- buttons increase the exposure and decrease the exposure of the image, respectively. They can be clicked several times to increase the value of the stake (each click represents 20%).

The slider controls the contrast of the imported image (it doesn't work very well in this version, updates are coming)

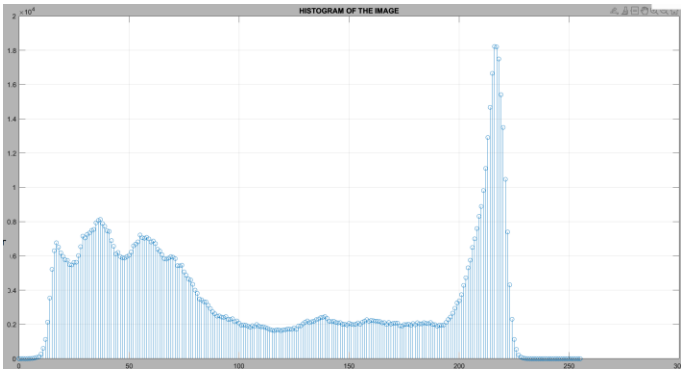


Fig6

The histogram is a type of histogram that acts as a graphical representation of the tonal distribution in a digital image. It plots the number of pixels for each tonal value.

By pressing the histogram button, the application generates the histogram of the imported image based on its tones and brightness, so you can see if the image is correctly exposed and the colors are correctly represented.

Fig7

Figure 7 shows the right side of the UI. This is not a part of photo editing, they are extra elements of the platform, offered for free.

Butoanele PLAY MUSIC si STOP MUSIC pornesc si respectiv opresc muzica prestabilita a programului.

Butoanele PLAY MUSIC si STOP MUSIC pornesc si respectiv opresc muzica prestabilita a programului.

The "About me" button redirects the user to the social media platform of the developer who created this application.

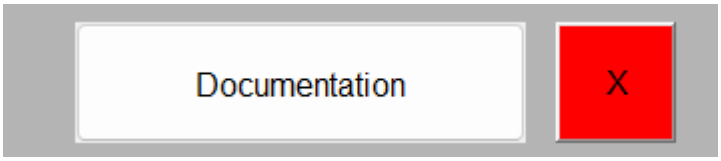


Fig8

In the upper right part, we have the buttons to close the application, as well as the one that redirects the user to the program documentation.

ESSENTIAL CODE BEHIND THE UI

One of the most important aspects behind this application is the possibility to apply several effects over the imported image, without resetting.

To do this, each function loads the image into the counter (currentImage) preset at the beginning of the code, with the exception of the RESET button function which resets the image to the original one.

The global counter declared at the beginning of the code:

```
global currentImage;
global initialImage;
currentImage=[];
```

RESET BUTTON FUNCTION : `function reset_callback(hObject,eventdata) % RESET`

```
imshow(initialImage);
currentImage=initialImage
end
```

At the same time, another challenging part of the code was the contrast slider. To make it work, I created a slider button to which I connected the contrast function:

```
function contrast_callback(hObject, eventdata)
% Get the current value of the slider
contrastValue = get(hObject, 'Value');
if contrastValue < 0
contrastValue = -contrastValue;
end
% Adjust the contrast of the image
contrast_image = imadjust(currentImage, [], [], contrastValue);
currentImage = contrast_image;
% Update the RGB channels
currentImage_r = currentImage(:, :, 1);
currentImage_g = currentImage(:, :, 1);
currentImage_b = currentImage(:, :, 1);
imshow(currentImage);
end
```

This code creates a numeric value based on the position of the slider. Then the value is transposed over the original image using the "imadjust" function, thus gradually increasing the contrast.

Disclaimer! The slider does not work perfectly!

CONCLUSION

This application is created for photo manipulation and it exceeded my expectations. I didn't think it was so easy to create an application that can do almost the same things as one for which you pay monthly subscriptions.

I would have liked to be able to implement more functions, such as lens correction, the possibility to crop and a slider that works more fluidly.

In the future I will try to add more elements that can help me in editing higher resolution pictures.

BIBLIOGRAPHY

Websites used in creating the project:

https://stackoverflow.com/

https://github.com/

www.mathworks.com

www.youtube.com

THE CODE:

```
function buttonPlot

% Create a figure window

f = figure;

f.WindowState = 'maximized';

global currentImage;

global initialImage;

currentImage=[];

%STATIC TEXT

uicontrol('Style', 'text', 'String', 'ADOBI Photo Editor', ...

'Position', [5, 300 200 40], 'FontName', 'Bahnschrift ', 'FontSize', 16, 'ForegroundColor', 'black',
'BackgroundColor','#B4B4B4' );

uicontrol('Style', 'text', 'String', 'Contrast', ...

'Position', [1, 220 200 40], 'FontName', 'Bahnschrift ', 'FontSize', 10, 'ForegroundColor', 'black',
'BackgroundColor','#B4B4B4' );

uicontrol('Style', 'text', 'String', 'By Frincean Andrei', ...

'Position', [5, 270 200 40], 'FontName', 'Bahnschrift', 'FontSize', 12, 'ForegroundColor', 'black',
'BackgroundColor','#B4B4B4' );

set(gcf,'color','#B4B4B4')

uicontrol('Style', 'text', 'String', '2022-2023 Adobi. All rights reserved', ...

'Position', [1440 3 300 40], 'FontName', 'Bahnschrift ', 'FontSize', 10, 'ForegroundColor', '#767676',
'BackgroundColor','#B4B4B4' );

% Creating a push buttons
```

```
my_button = uicontrol( 'Style', 'pushbutton', 'String' , 'Upload IMG',...
'Position', [20 10 180 50],...
'Callback', @upload_callback);
my_button2 = uicontrol('Style', 'pushbutton', 'String', ' B&W ',...
'Position', [200 10 150 40],...
'Callback', @bw_callback);
my_button3 = uicontrol('Style', 'pushbutton', 'String', ' HUE+ ',...
'Position', [350 10 150 40],...
'Callback', @hueup_callback);
my_button4 = uicontrol('Style', 'pushbutton', 'String', ' HUE- ',...
'Position', [500 10 150 40],...
'Callback', @huedown_callback);
my_button5 = uicontrol('Style', 'pushbutton', 'String', ' RESET ',...
'Position', [650 10 150 40],...
'Callback', @reset_callback);
my_button6 = uicontrol('Style', 'pushbutton', 'String', ' Histogram ',...
'Position', [20 70 150 40],...
'Callback', @histogram_callback);
my_button7 = uicontrol('Style', 'pushbutton', 'String', ' Exposure+ ',...
'Position', [20 120 150 40],...
'Callback', @exposureup_callback);
my_button8 = uicontrol('Style', 'pushbutton', 'String', ' Exposure- ',...
'Position', [20 170 150 40],...
'Callback', @exposuredown_callback);
my_button9 = uicontrol('Style', 'pushbutton', 'String', ' Stop music ',...
'Position', [1550 60 150 40],...
'Callback', @audiostop_callback);
my_button10 = uicontrol('Style', 'pushbutton', 'String', ' Play music ',...
'Position', [1550 110 150 40],...
'Callback', @audioplay_callback);
my_button11 = uicontrol('Style', 'pushbutton', 'String', ' About me ',...
```

```
'Position', [1550 160 150 40],...
'Callback', @surprise_callback);
my_button12 = uicontrol('Style', 'pushbutton', 'String', ' ! ',...
'Position', [1550 210 150 40],...
'Callback', @warning_Callback);
my_button13 = uicontrol('Style', 'pushbutton', 'String', ' X ',...
'Position', [1650 790 40 40],...
'Callback', @close_callback);
my_button14 = uicontrol('Style', 'pushbutton', 'String', ' Documentation ',...
'Position', [1490 790 150 40],...
'Callback', @open_pdf);
% Slider
contrast_slider = uicontrol('Style', 'slider', 'Min', 0, 'Max', 10, ...
'Value', 0, 'Position', [20 220 150 25]);
set(my_button, 'BackgroundColor', 'red')
set(my_button9, 'BackgroundColor', 'red')
set(my_button13, 'BackgroundColor', 'red')
set(my_button10, 'BackgroundColor', 'green')
% Afisarea imaginii default
function initializeDefaultMedia
currentImage=imread('nomedia.jpg');
imshow(currentImage);
end
initializeDefaultMedia;
% Functions to be called when button is pressed:
function upload_callback(hObject,eventdata) %Upload picture function
uploadImage=uigetfile('*.*');
currentImage=imread(uploadImage);
initialImage=currentImage;
imshow(currentImage)
end
```

```
function bw_callback(hObject,eventdata)%transforms the image in B&W
% Load image
I = im2gray(currentImage);
currentImage=I
imshow(currentImage);
end
function hueup_callback(hObject,eventdata) %HUE+ function
rgbImage = currentImage
k = 3;
hsvImage = rgb2hsv(rgbImage);
hsvImage(:,:,1) = mod(hsvImage(:,:,1)*k,1);
rgbImage = hsv2rgb(hsvImage);
currentImage=rgbImage;
imshow(currentImage);
end
function huedown_callback(hObject,eventdata) %HUE- function
rgbImage =currentImage
k = -3;
hsvImage = rgb2hsv(rgbImage);
hsvImage(:,:,1) = mod(hsvImage(:,:,1)*k,1);
rgbImage = hsv2rgb(hsvImage);
imshow(rgbImage);
end
function reset_callback(hObject,eventdata) % RESET
imshow(initialImage);
currentImage=initialImage
end
function histogram_callback(hObject,eventdata) %opens the histogram
;
img=rgb2gray(currentImage);
[x, y] = size(img);
```

```
frequency = 1 : 256;
count = 0;
for i = 1 : 256
for j = 1 : x
for k = 1 : y
if img(j, k) == i-1
count = count + 1;
end
end
end
frequency(i) = count;
count = 0;
end
n = 0 : 255;
% Display Histogram
h = stem(n, frequency);
grid on;
title('HISTOGRAM OF THE IMAGE');
set(h, 'ButtonDownFcn', @(src,evnt) delete_hist(src,evnt));
end
function exposureup_callback(hObject,eventdata) %Exposure+ function
I = currentImage ;
I1 = I+20 ;
currentImage=I1
imshow(currentImage) ;
end
function exposuredown_callback(hObject,eventdata) %Exposure- function
I = currentImage ;
I1 = I-20 ;
currentImage=I1
imshow(currentImage) ;
```

```
end
function audiostop_callback(hObject,eventdata) %Pauses the music
global p
[y,Fs]=audioread("Wires.mp3");
player=audioplayer(y,Fs);
p=player;
play(p);
pause(player)
end
function audioplay_callback(hObject,eventdata) %Plays the music
global p
[y,Fs]=audioread("Wires.mp3");
player=audioplayer(y,Fs);
p=player;
play(p);
end
function surprise_callback(hObject,eventdata) %Opens smth
url = 'https://www.instagram.com/franceanandrei/';
web(url)
end
set(contrast_slider, 'Callback', @contrast_callback);% Callback slider.
function contrast_callback(hObject, eventdata)
% Get the current value of the slider
contrastValue = get(hObject, 'Value');
if contrastValue < 0
contrastValue = -contrastValue;
end
% Adjust the contrast of the image
contrast_image = imadjust(currentImage, [], [], contrastValue);
currentImage = contrast_image;
% Update the RGB channels
```

```
currentImage_r = currentImage(:, :, 1);
currentImage_g = currentImage(:, :, 1);
currentImage_b = currentImage(:, :, 1);
imshow(currentImage);
end
function warning_Callback(hObject, eventdata)
opts = struct('WindowStyle','modal',...
'Interpreter','tex');
f = warndlg('\color{black} Press OK if you love Miha<3',...
' Warning!!!', opts);
end
function close_callback(hObject,eventdata)
close(gcf)
end
function open_pdf(~,~)
pdf_path = 'LAB.pdf';
winopen(pdf_path);
end
end
```