ImageProcessing

Final Project

Blood Types

Abstract:

A blood-group project that helps to detect the type of the project in a short period of type based on image-processing algorithms and methodologies.

System Requirements & Usecases :

This project been made & coded using matlab specially for medical purposes so basically it’s for doctors and nurses.

Coded & Managed by :

Aws Nassar

Problems we faced :

The main problem we had , we couldn’t find any real images that contain blood samples so what we have done , we got some digital images to all the blood types and worked separately on them in order to specify the pattern of blood-types and how to define , divide and get the right methodology to build the algorithm.

Explanation :

How the project works ?

First-step:

By choosing an blood sample as given in the specified files , we inserted and managed to do tests on the following images with the project file , containing all the blood types (ABO+-).

Second-step:

Each blood sample contain x3 slices so we cropped the blood sample into 3 sub-images to process which slice got an anti etc..

Third-step:

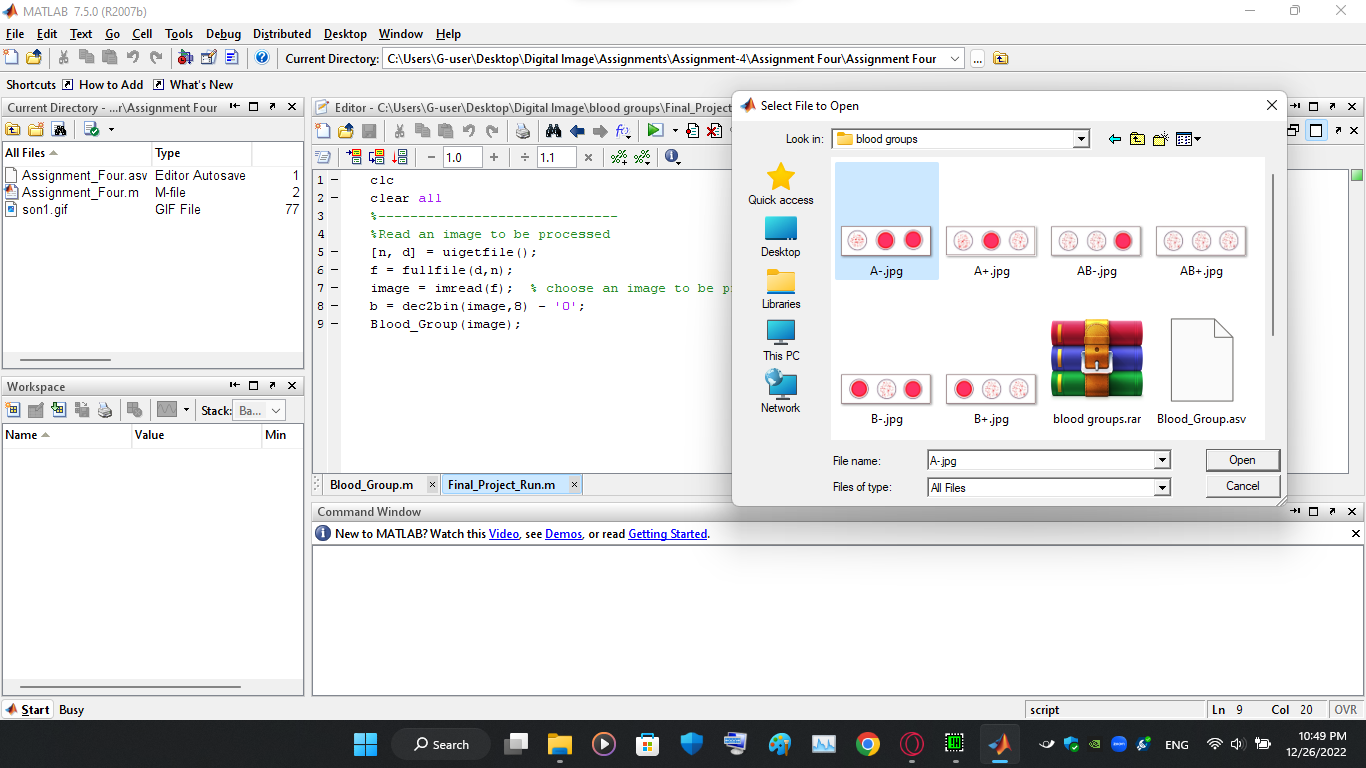
We check the histogram of each slice and compare it to a given calculated pixel-by-pixel process that provides us information whether the given slice is a colored slice or not to proceed into the next step.

Fourth-step:

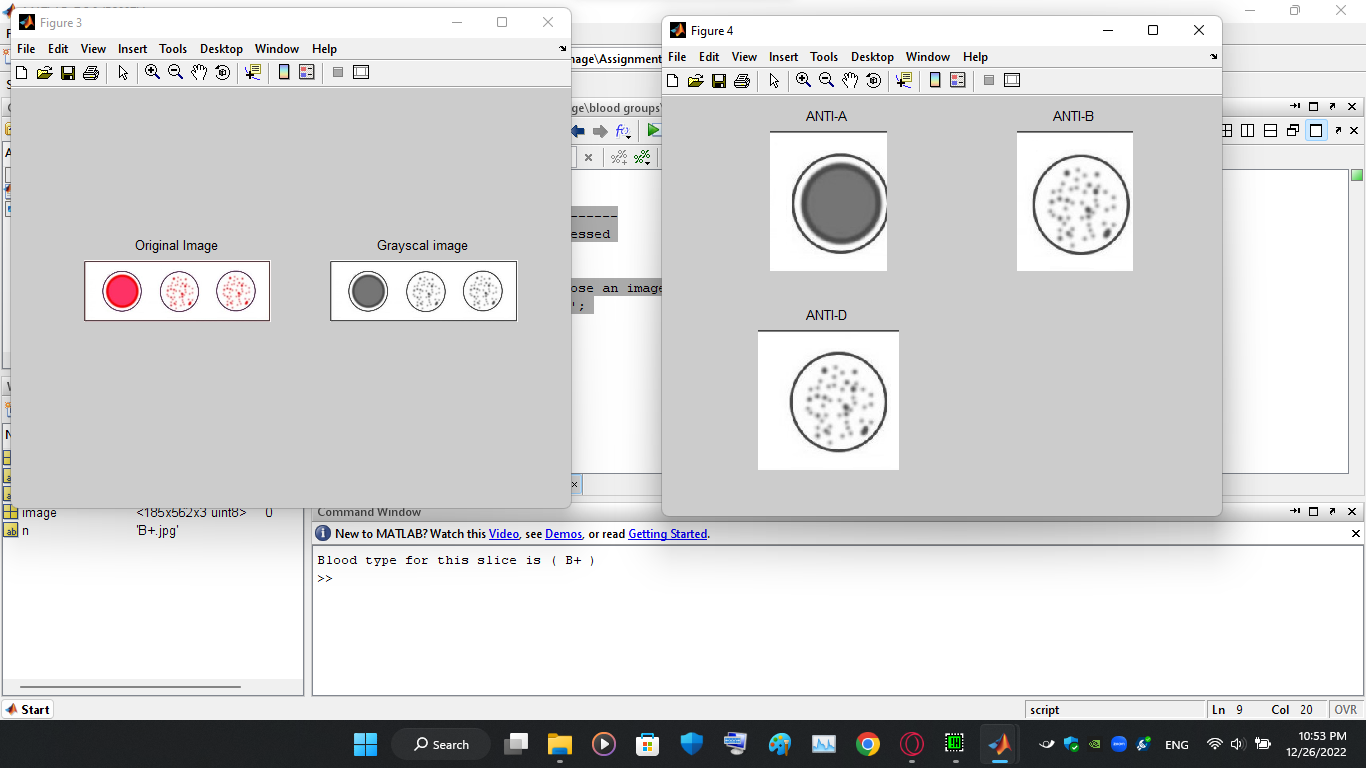
While doing the third step , we have three checking Booleans, 1 and 0. So if the given slice is colored it will be set to 1 else 0 at the end we will have three values for A , B and D(C).

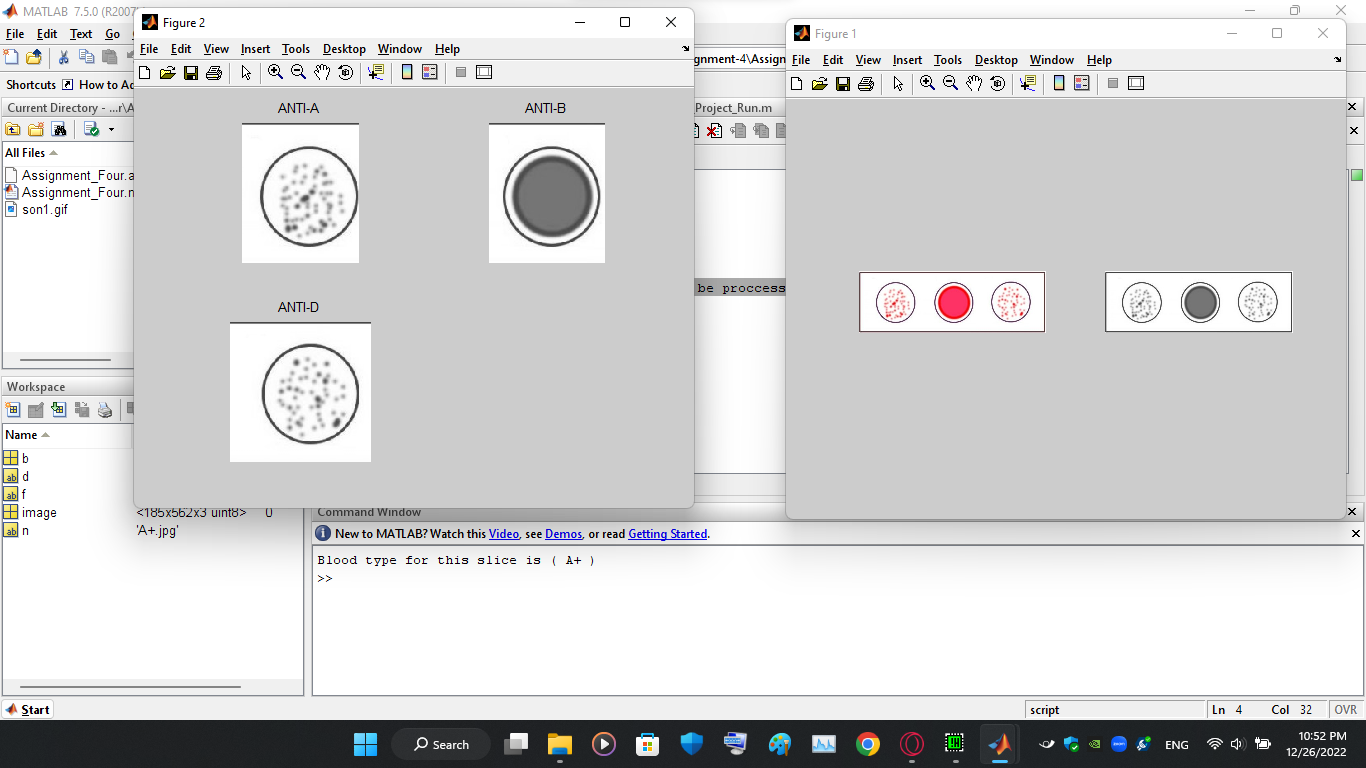
Fifth-step:

Last step in order to give the conclusion about the three Booleans and return the type of the blood within 8 possible types of blood.



Choosing the image:

Slicing & giving the result in console ( B+ ).

Slicing & giving the result in console ( A+ ).

Conclusion:

Making the blood-type matlab program made us aware of how things work using the image processing methods.

We built our own algorithm without the needs of using any actual filters , a suggested filter we wanted to use is the Hit-or-Miss filter but at some point it will not work well with Error percentage might happen.

Our own algorithm might be faster and better than hit or miss by diving the pictures and sampling each one of the division.

Working on digital blood samples gave us a 95% success rate , the 5% left might happen with new pictures but it must work properly if the size is similar to the given one at the attached files.