**Chapter-4**

**IMPLEMENTATION & USER INTERFACE**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**4.1 Installing MATLAB**

Download the installer from the MathWorks website or locate your MATLAB DVDs.

When you select an installer on the website, you receive the files that the installer needs in order to run on your selected platform, not the product files. Run the installer to select products that you want to install on the computer where you are logged in. The installer downloads the products that it needs from MathWorks, if necessary.

Downloaded installer file — If you click Run when you select the installer on the MathWorks website, the installer starts automatically. If you saved the installer instead, double-click the self-extracting installer file in the folder where you saved it. The name of the self-extracting installer file is matlab\_release\_$ARCH, where release is the release you are installing (for example R2016b) and $ARCH is the selected architecture.

The installer should start automatically after file extraction is complete.

**4.2 Installing Webcam Support Package**

Downloaded installer file — If you click Run when you select the installer on the MathWorks website, the installer starts automatically. If you saved the installer instead, double-click the self-extracting installer file in the folder where you saved it. The name of the self-extracting installer file is matlab\_release\_$ARCH, where release is the release you are installing (for example R2016b) and $ARCH is the selected architecture.

Webcam support is available through MATLAB Add-Ons. Using this installation process, you download and install the following files on your host computer

1. MATLAB files for Webcam support
2. An example that shows how to acquire images using a Webcam
3. The USB Webcams support package documentation

To install the support package:

1. On the MATLAB Home tab, in the Environment section, click Add-Ons > Get Hardware Support Packages.
2. In the Add-On Explorer, scroll to the Hardware Support Packages section, and click show all to find your support package.
3. Refine the list by selecting Imaging/Cameras in the Refine by Hardware Type section on the left side of the Explorer.
4. In the Imaging/Cameras list, select the MATLAB Support Package for USB Webcams.

**4.3 Installing MySQL**

Default installation on any version of Windows is now much easier than it used to be, as MySQL now comes neatly packaged with an installer. Simply download the installer package, unzip it anywhere, and run setup.exe.

Default installer setup.exe will walk you through the trivial process and by default will install everything under C:\mysql.

Test the server by firing it up from the command prompt the first time. Go to the location of the mysqld server which is probably C:\mysql\bin, and type:

mysqld.exe –console

If all went well, you will see some messages about startup and InnoDB. If not, you may have a permissions issue. Make sure that the directory that holds your data is accessible to whatever user (probably mysql) the database processes run under.

MySQL will not add itself to the start menu, and there is no particularly nice GUI way to stop the server either. Therefore, if you tend to start the server by double clicking the mysqld executable, you should remember to halt the process by hand by using mysqladmin, Task List, Task Manager, or other Windows-specific means.

**4.4 Connect MySQL to MATLAB**

If the JDBC driver for MySQL is not installed on your computer, find the link on the Driver Installation page to install the driver. Follow the instructions to download and install this driver on your computer.

1. Run the prefdir command in the Command Window. The output of this command is a file path to a folder on your computer.
2. Close MATLAB® if it is running.
3. Navigate to the folder and create a file called javaclasspath.txt in the folder.
4. Open javaclasspath.txt. Add the full path to the database driver JAR file in javaclasspath.txt. The full path includes the path to the folder where you downloaded the JAR file from the database provider and the JAR file name. For example, C:\DB\_Drivers\mysql-connector-java-5.1.17-bin.jar. Save and close javaclasspath.txt.
5. Restart MATLAB.
6. Alternatively, you can use javaaddpath to add your JDBC driver to the dynamic Java® class path. For details about static and dynamic class paths, see Java Class Path (MATLAB).

This step is required only for connecting to Database Explorer. If you want to use the command line to connect to your database, see Connect to MySQL Using JDBC Driver and Command Line

1. Open Database Explorer by clicking the Apps tab on the MATLAB Toolstrip. Then, select Database Explorer from the Database Connectivity and Reporting section in the apps gallery. Alternatively, enter dexplore at the command line. If no data sources are set up, a message box opens. Click OK to close it. Otherwise, the Connect to a Data Source dialog box opens. Click Cancel to close this dialog box.
2. Click the Database Explorer tab, and then select New > JDBC.
3. Select MYSQL from the Vendor list. After selecting the vendor, if you did not add the JDBC driver file path to the Java class path, this dialog box displays this message at the bottom. Address this message by following the steps described in Step 2.
4. Enter the database server name in the Server Name field, port number in the Port Number field, user name in the Username field, password in the Password field, and database name in the Database field.
5. Click Test to test the connection. If your connection succeeded, Database Explorer displays Connection Successful!
6. Enter a data source name in the Data Source Name field in the Create a New JDBC data source dialog box. Use a new data source name that does not appear in the existing list of data source names. Click Save. The new JDBC data source appears in the list of data sources in the Connect to a Data Source dialog box.
7. If this time is the first time that you are creating a data source using Database Explorer, the New file to store JDBC connection parameters dialog box opens. Use this dialog box to create a MAT-file that saves your specified data source information for future Database Explorer sessions. This MAT-file name is stored in setdbprefs('JDBCDataSourceFile') and is valid for all MATLAB sessions.
8. Navigate to the folder where you want to put the MAT-file, specify a name for it that includes a .mat extension, and click Save.
9. After you complete the data source setup, connect to the MySQL database using Database Explorer or the command line with the JDBC connection.

**4.5 User Interface**

The project is completely deployed on the MATLAB and MySQL database.  
The main feature of this project is to take image as an input and find the equivalent record from the database.

**4.6 Screenshots**

This subsection contains the snapshots of user interface and implemented project.

Figure 4.1:The very first page of our project is login page.

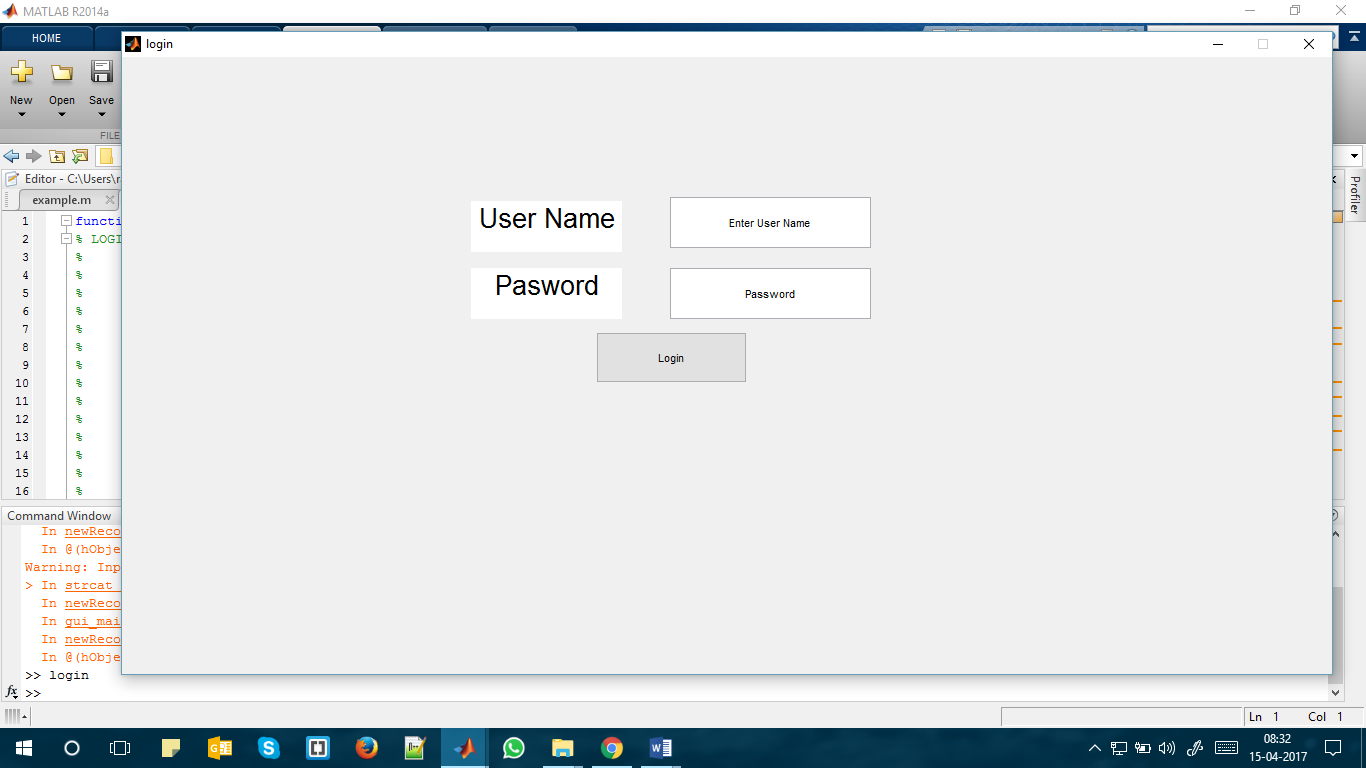


Figure 4.1: LoginPage

Figure 4.2: This is the front GUI of our project.

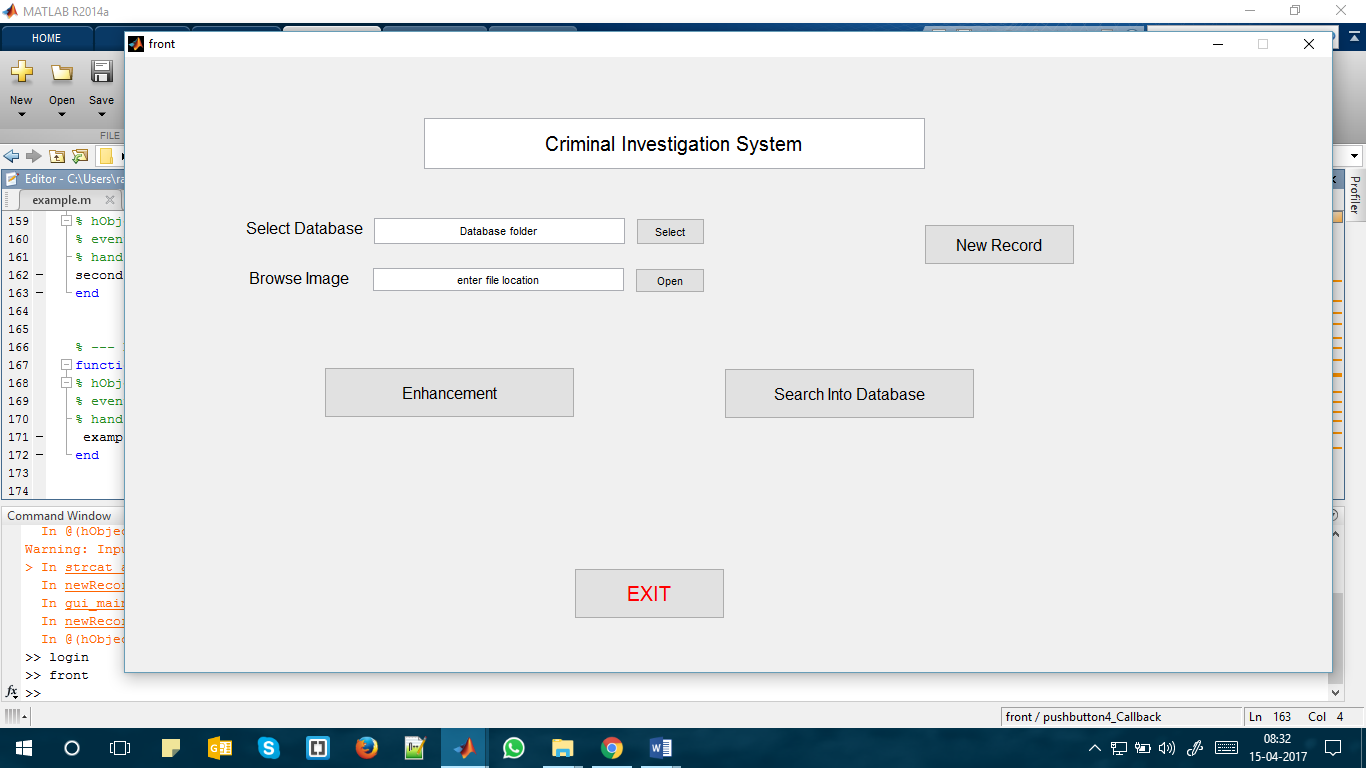


Figure 4.2: Main Page

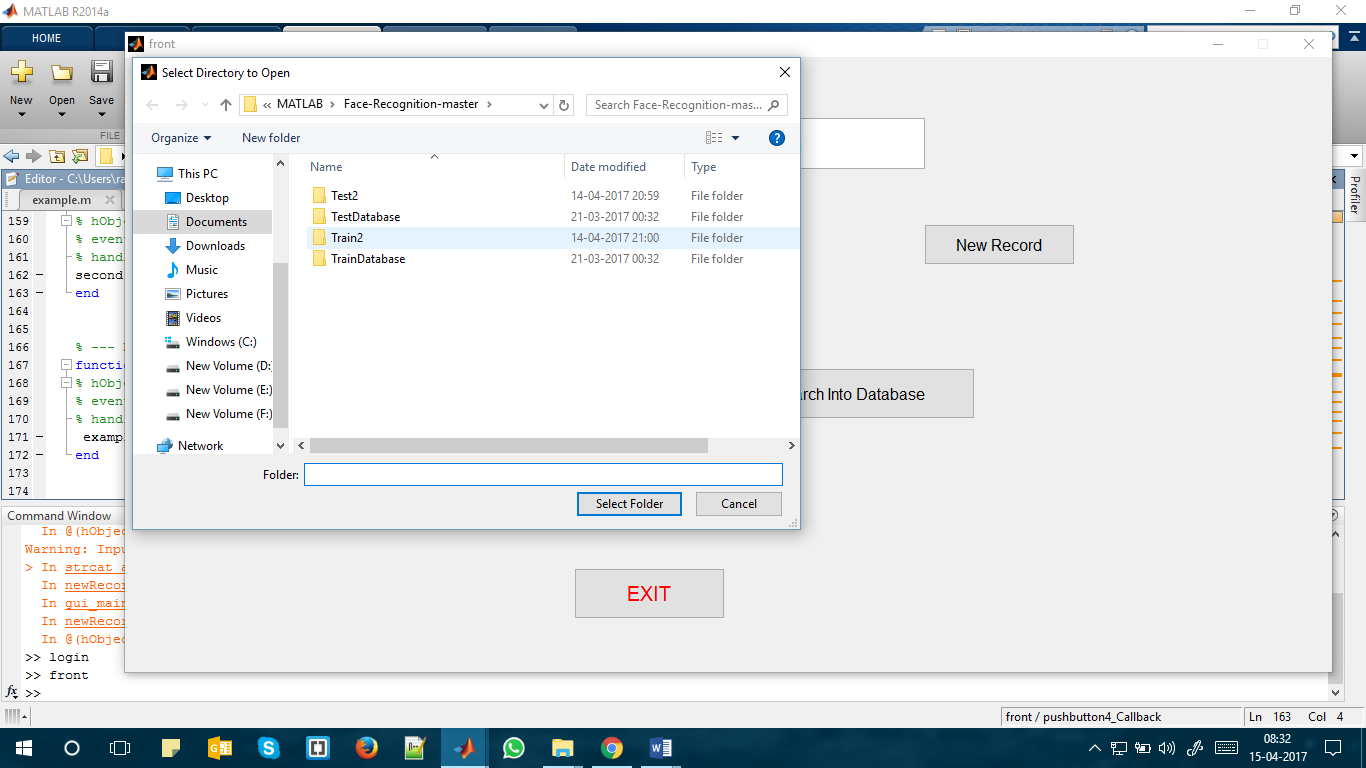
Figure 4.3: This is the process of selecting dataase folder for training .

Figure 4.3: select database

Figure 4.4: This is select image page, where user selects the image from the device.

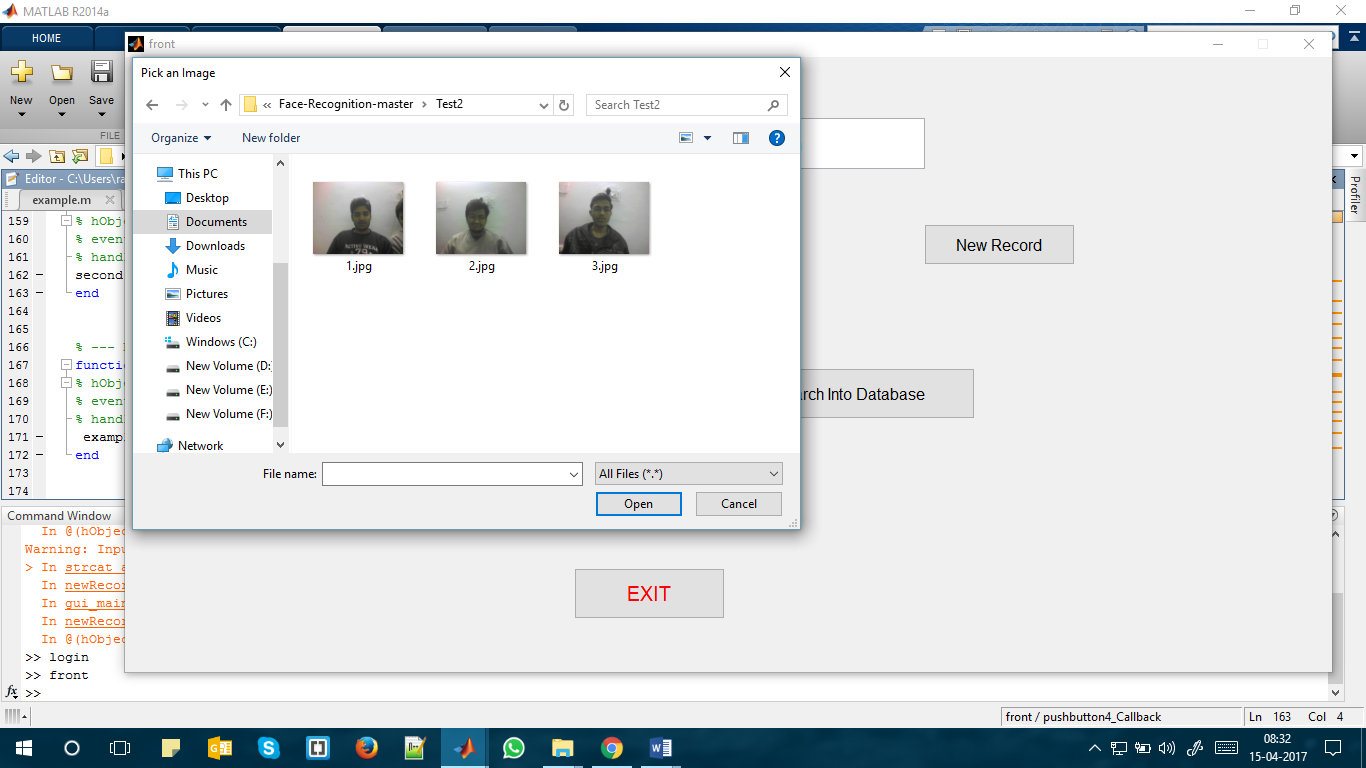


Figure 4.4: select image

Figure 4.5:Image enhancement page.

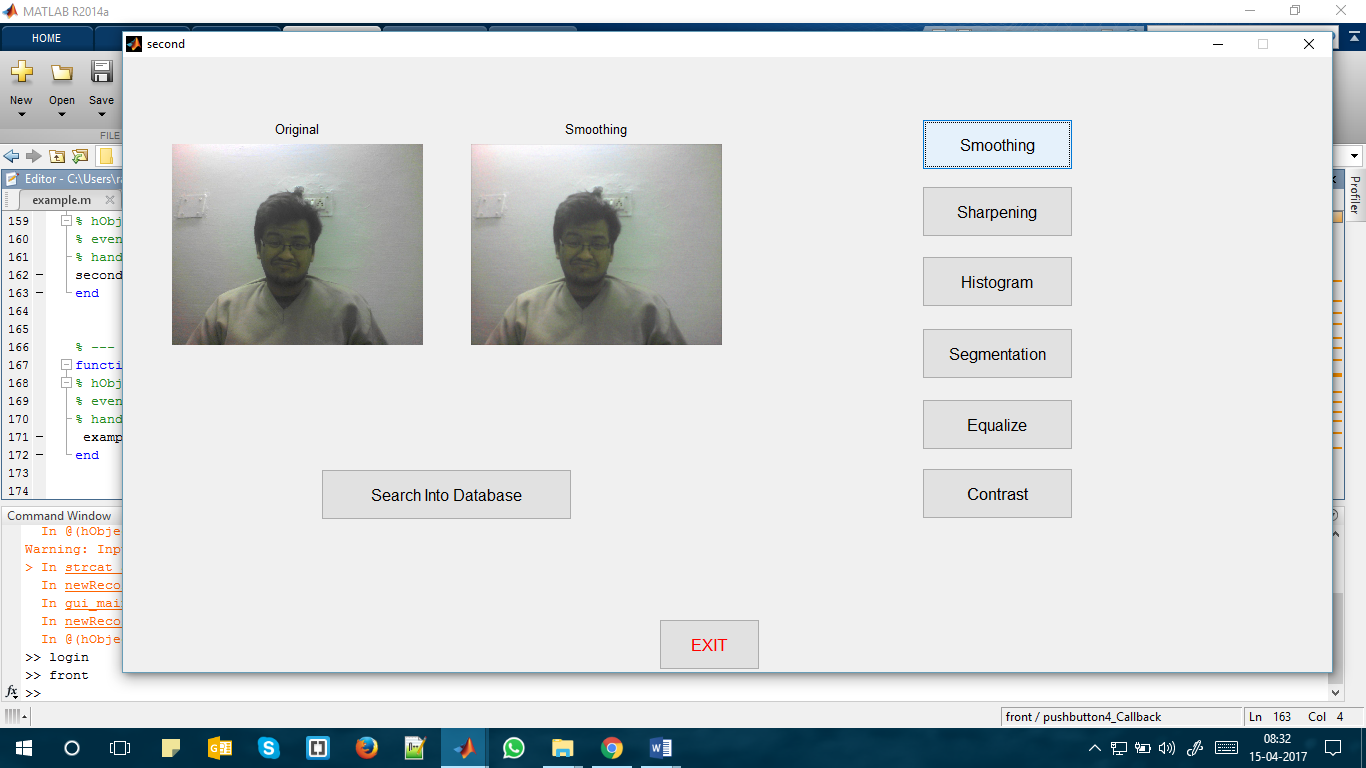


Figure 4.5: Enhancement page

Figure 4.6:Image enhancement : Smoothening.

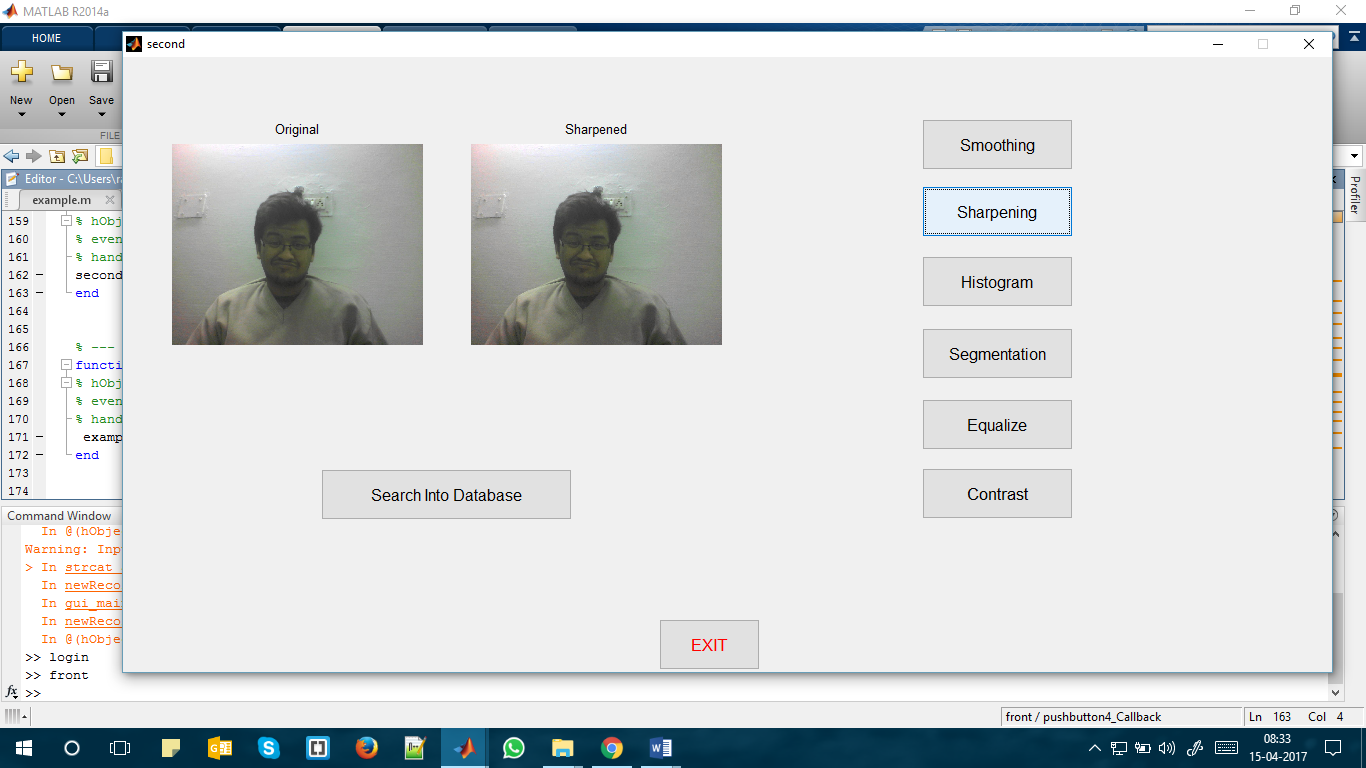


Figure 4.6: Smoothening

Figure 4.7:Image enhancement :Sharpening.

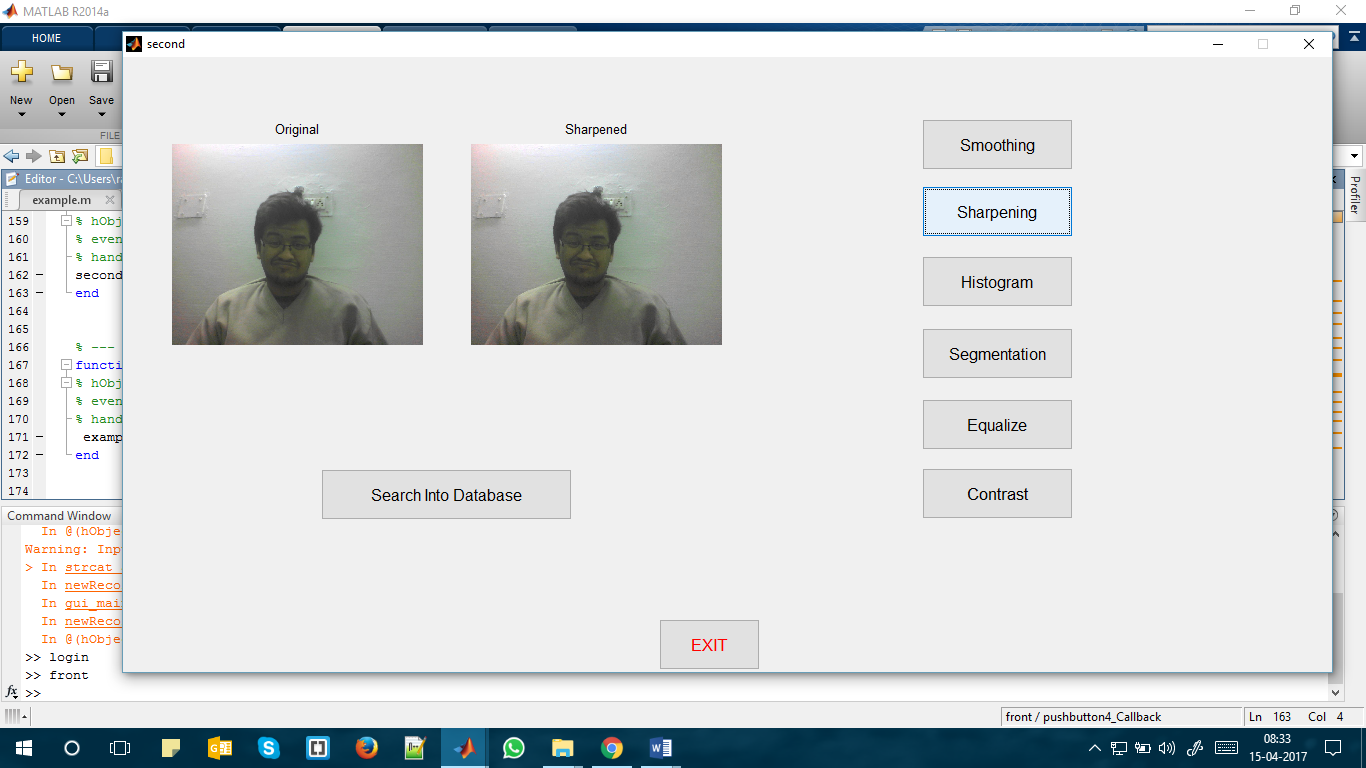


Figure 4.7: Sharpenig

Figure 4.8:Image enhancement :Histogram.

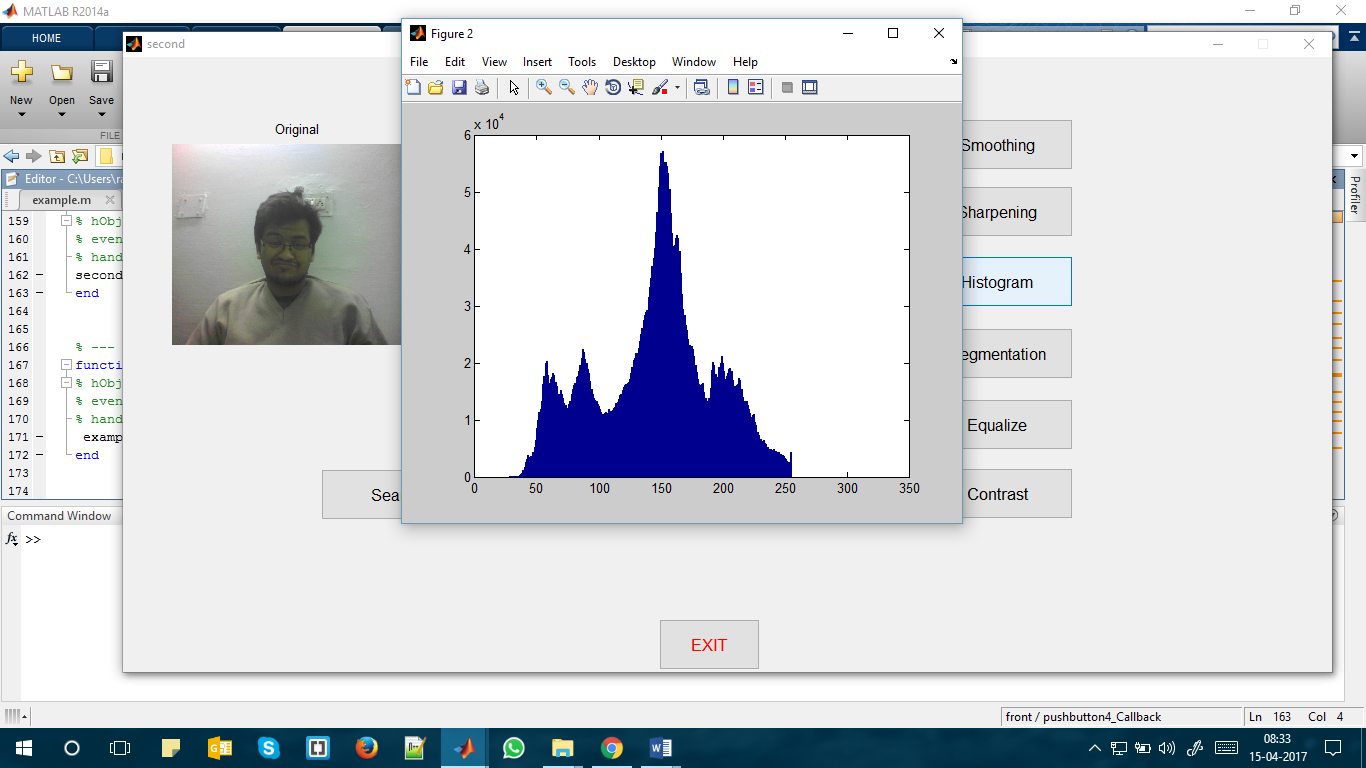


Figure 4.8: Histogram

Figure 4.9 :Image enhancement : Segmentation.

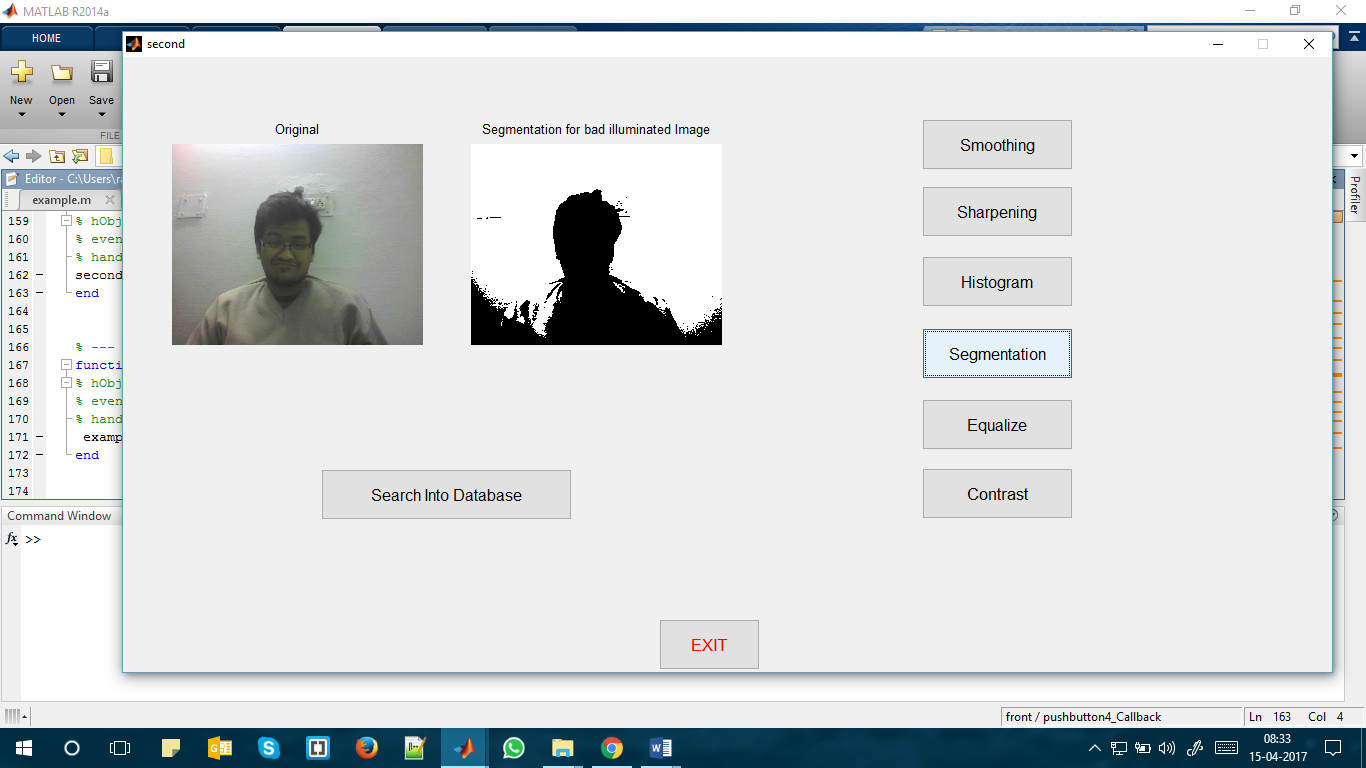


Figure 4.9: Segmentation

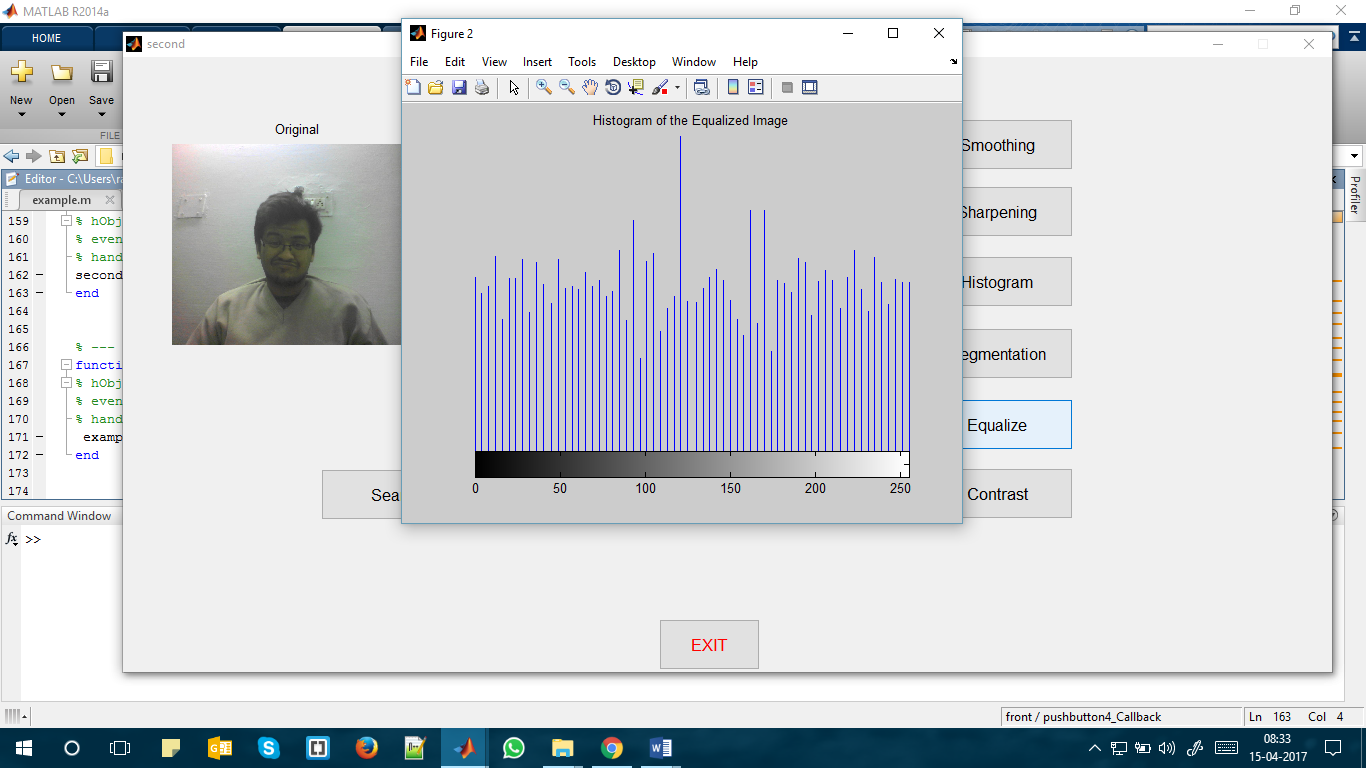


Figure 4.10 : Image enhancement : Equalize.

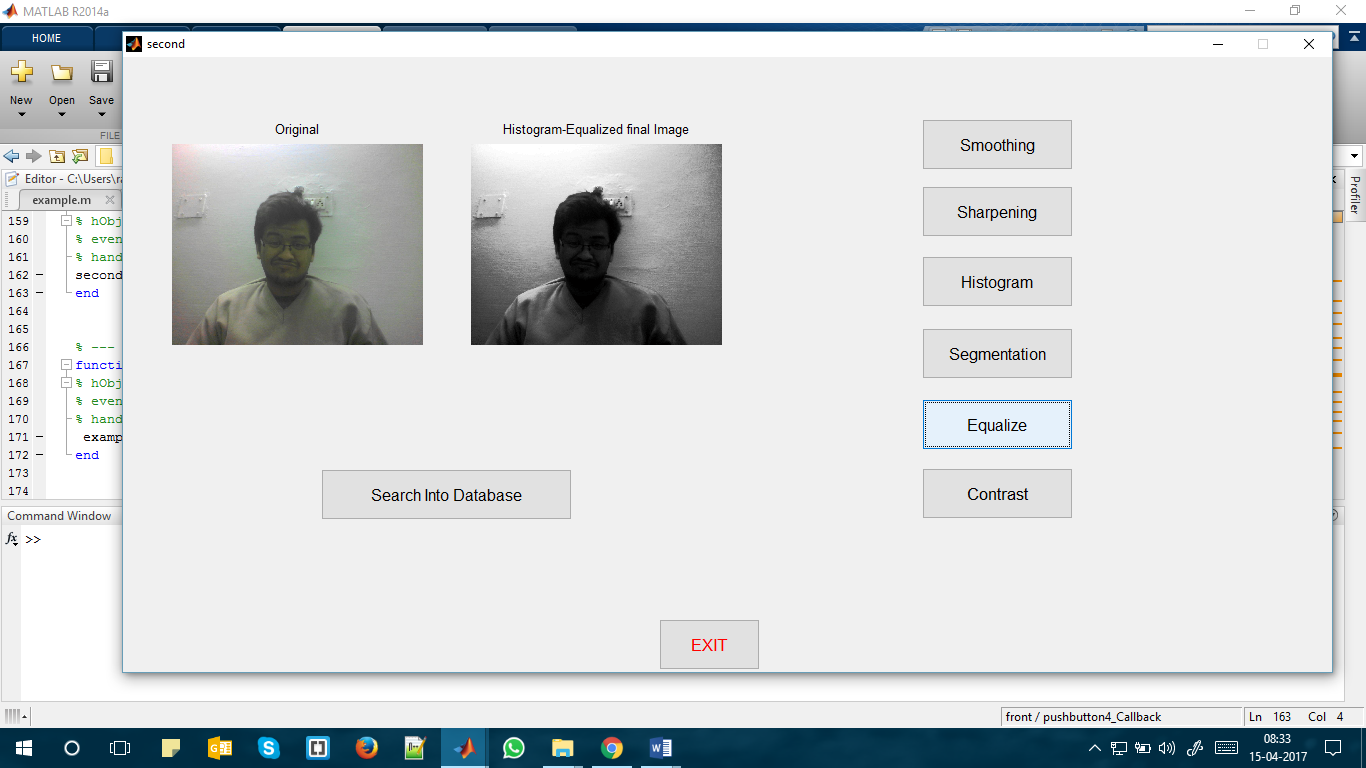


Figure 4.11: Equalize

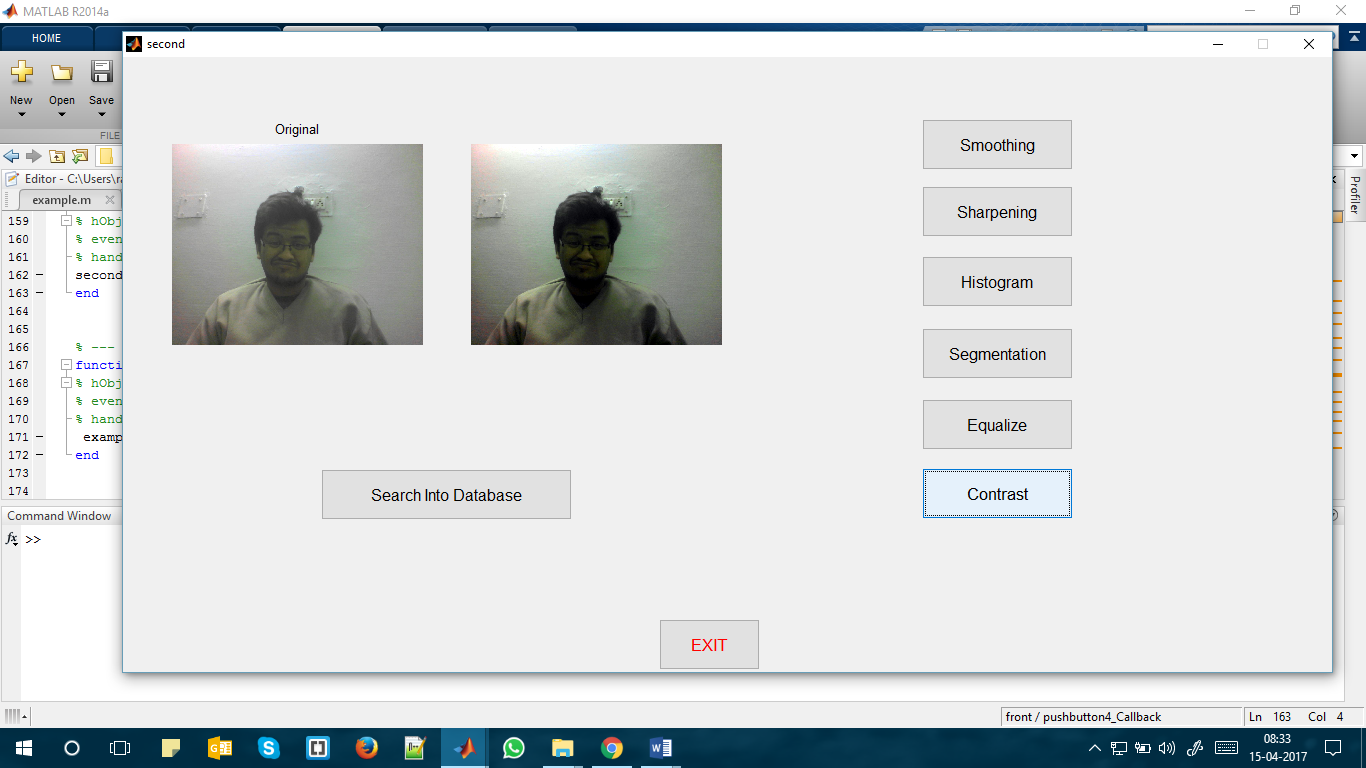


Figure 4.12: Contrast



Figure 4.13: Search Database

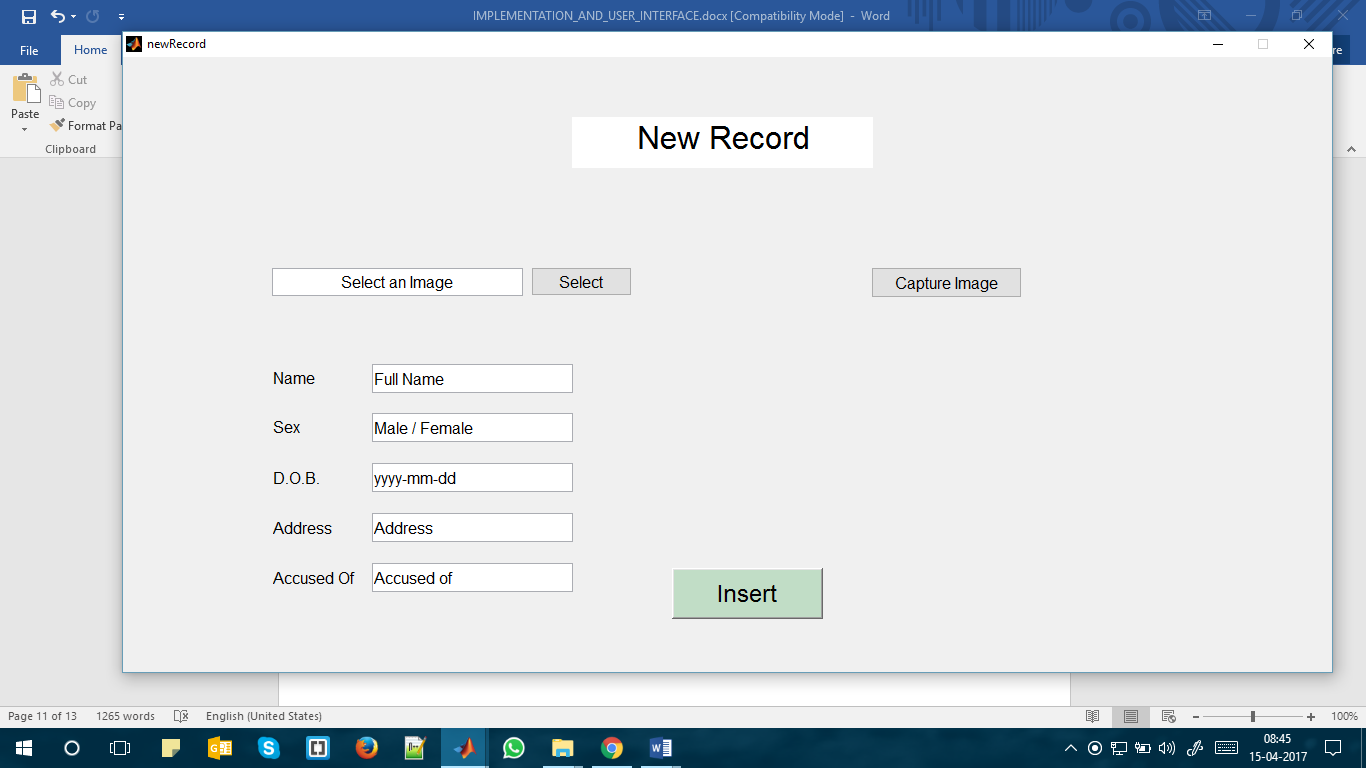


Figure 4.14: New Record

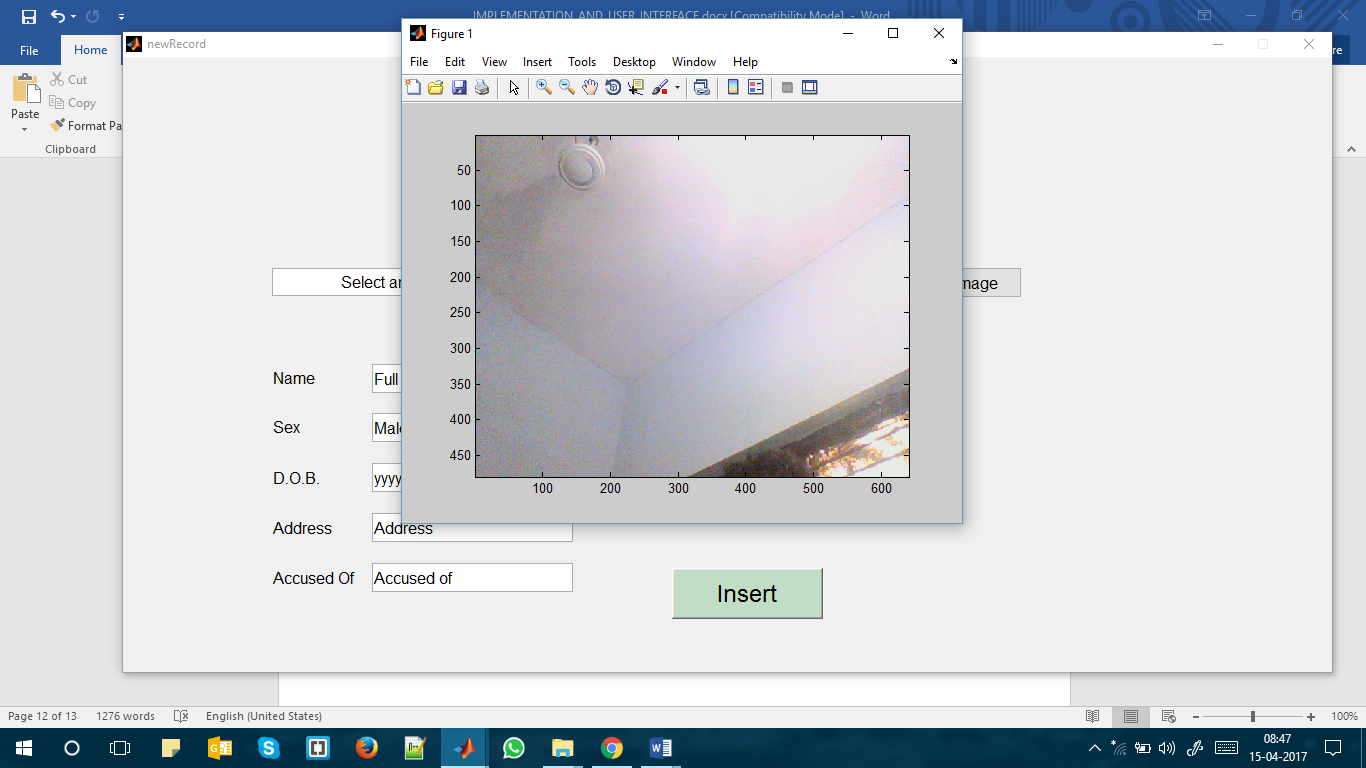


Figure 4.15: Capture image