USER MANUAL V4

Image Tampering Detection

TABLE OF CONTENTS

1.0 GENEARAL INFORMATION

1.1 Introduction

Purpose behind this project is to reliably determine whether a digital image has been tampered with using passive forensic techniques. This project involves investigating the performance of a format based technique called PRNU based image forgery detection.

1.2 System Overview

Image Tampering Detection tool performs forgery detection on digital image. This tool is built on Matlab platform based on the PRNU fingerprint technique.

This program uses Bayesian-MRF algorithm which consists of both BM3D and Mihcak denoising method. Currently has the code loaded with digital images from the CanonEOS_10D and Nikon_D200 cameras with different resolution.

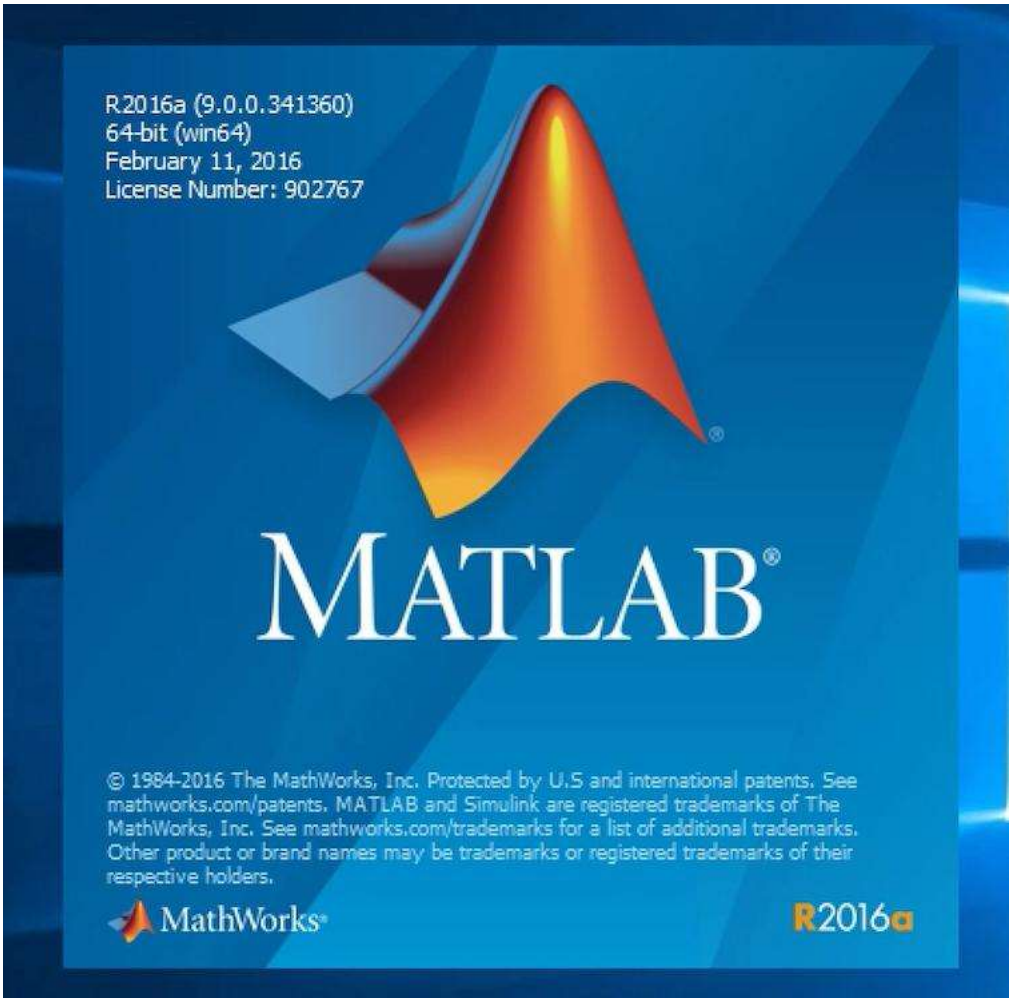
1.3 Acronyms and Abbreviations

PRNU: Photo-response non-uniformity.
MRF: Markov random field.
BM3D: Block-matching and 3D filtering

2.0 SYSTEM SUMMARY

2.1 System Configuration

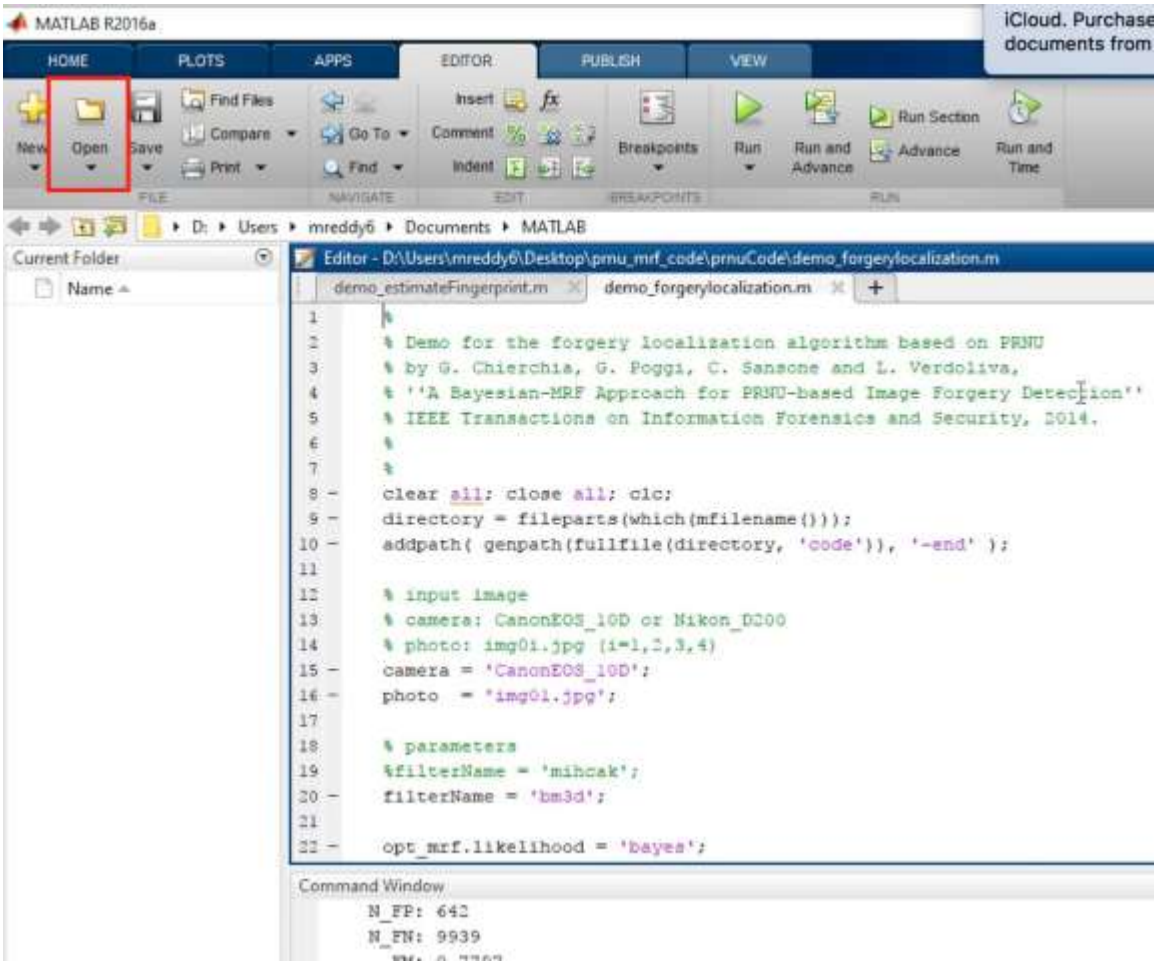
Matlab2014 or MatlabR2016a licensed version is required to run the code and should be on Windows 64-bit operating system.



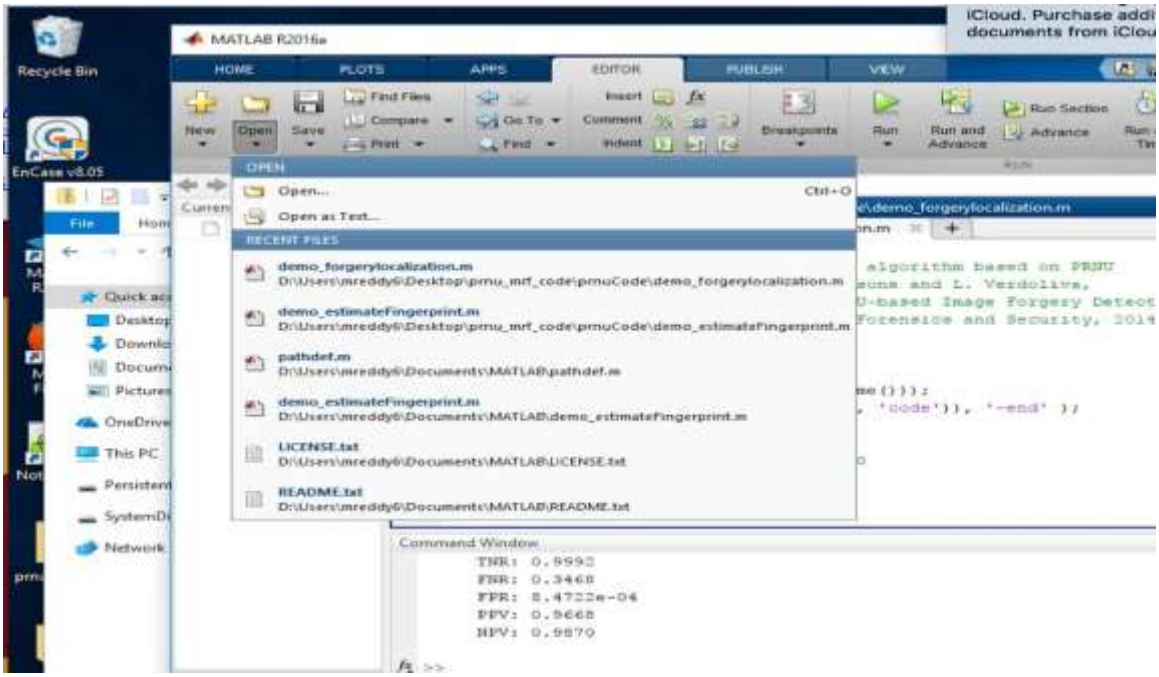
3.0 GETTING STARTED

3.1 Loading the Matlab with code

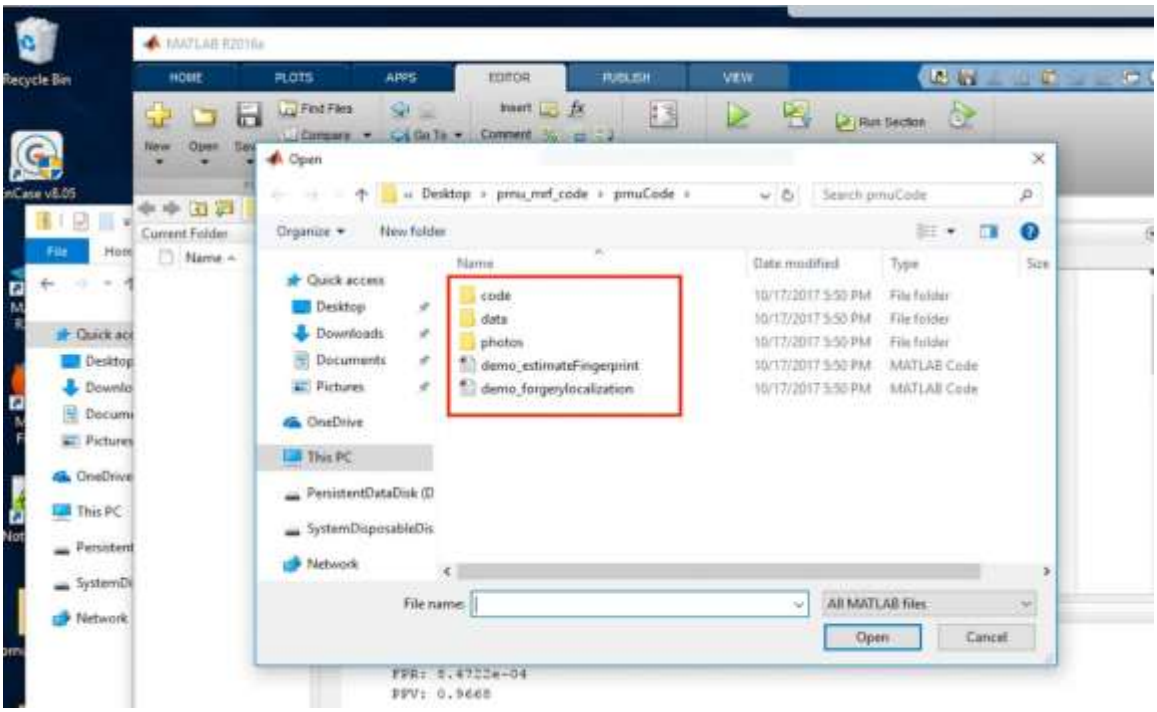
Step 1: Launch the Matlab and click on the open button



Step 2:

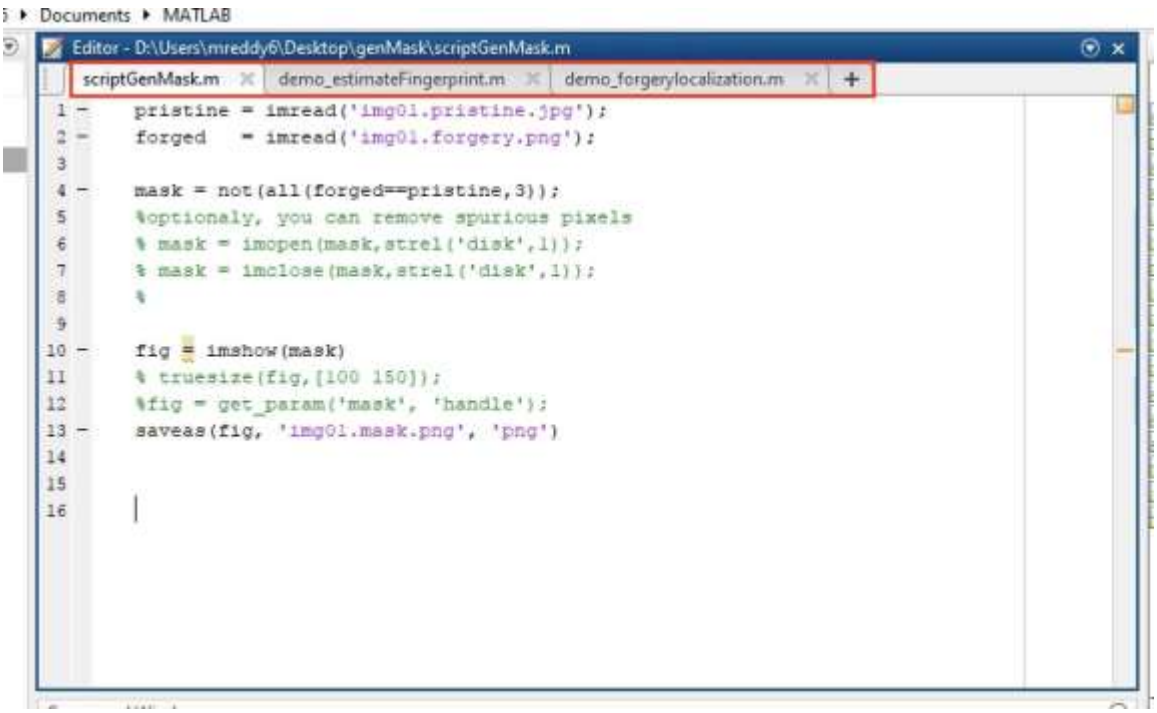


Step 3: Load all the Matlab files that are show in the below screenshots.



4.0 RUNNING THE CODE

After loading the files open the **scriptGenMask.m**, **demo_forgerylocalization.m** and **demo_estimatefingerprint.m** file in the Matlab as shown below,

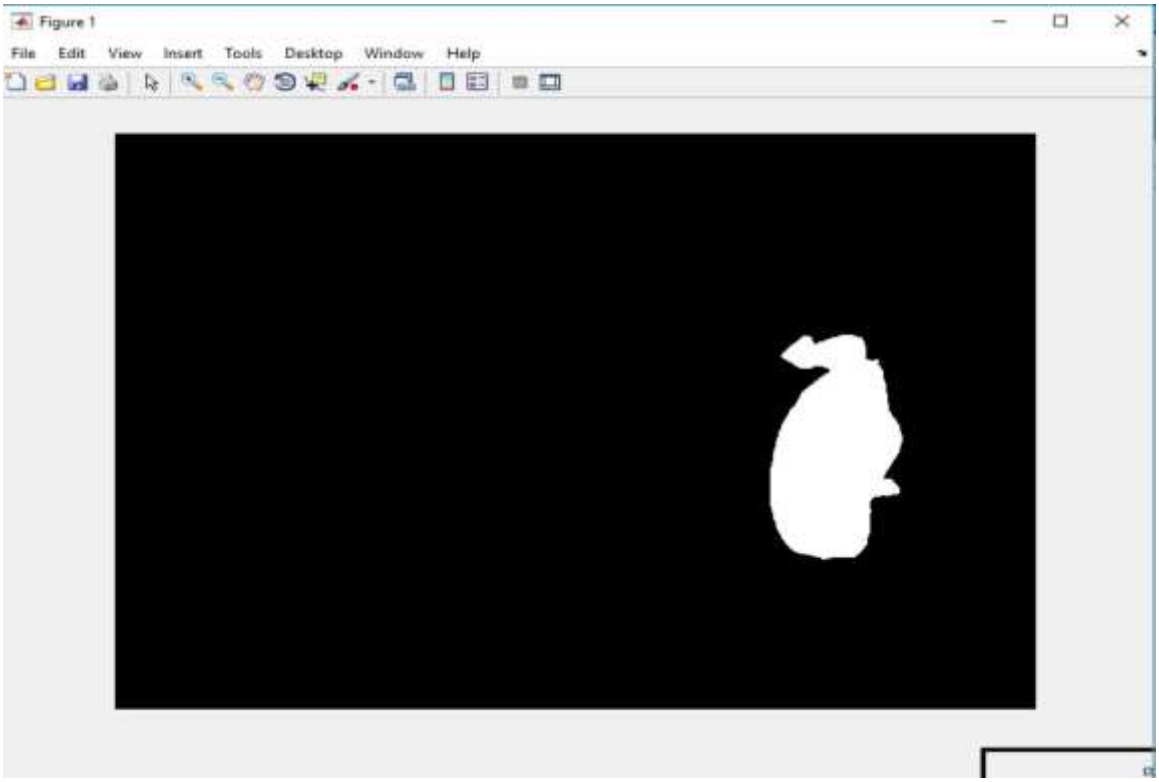


Once the files are loaded run the **scriptGenMask.m** matlab file to generate the **Mask** for which specify the image name for **pristine and forgery images** as shown below**.** Also specify the file name for mask to be saved as. We can change the extension of the file name by changing to jpeg also by giving so.

Note always have the same image name for pristine, forgery and masked one to run it in PRNU tool.

```
Editor - D:\Users\mreddy6\Desktop\genMask\scriptGenMask.m*
scriptGenMask.m*  demo_estimateFingerprint.m  demo_forgerylocalization.m
1 -   pristine = imread('img01.pristine.jpg');
2 -   forged   = imread('img01.forgery.jpg');
3
4 -   mask = not(all(forged==pristine,3));
5     %optionaly, you can remove spurious pixels
6     % mask = imopen(mask,strel('disk',1));
7     % mask = imclose(mask,strel('disk',1));
8     %
9
10 -  fig = imshow(mask)
11    % truesize(fig,[100 150]);
12    %fig = get_param('mask', 'handle');
13 -  saveas(fig, 'img01.mask.png', 'png')
14
15
16
```

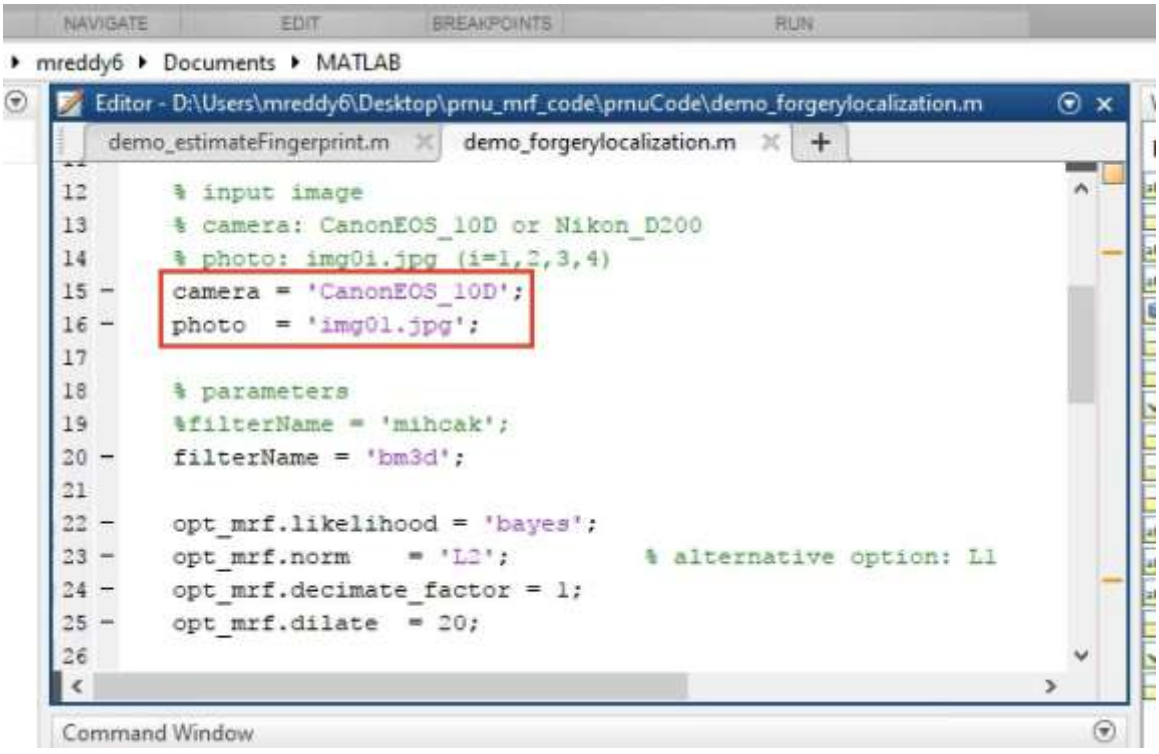
Mask results be like,



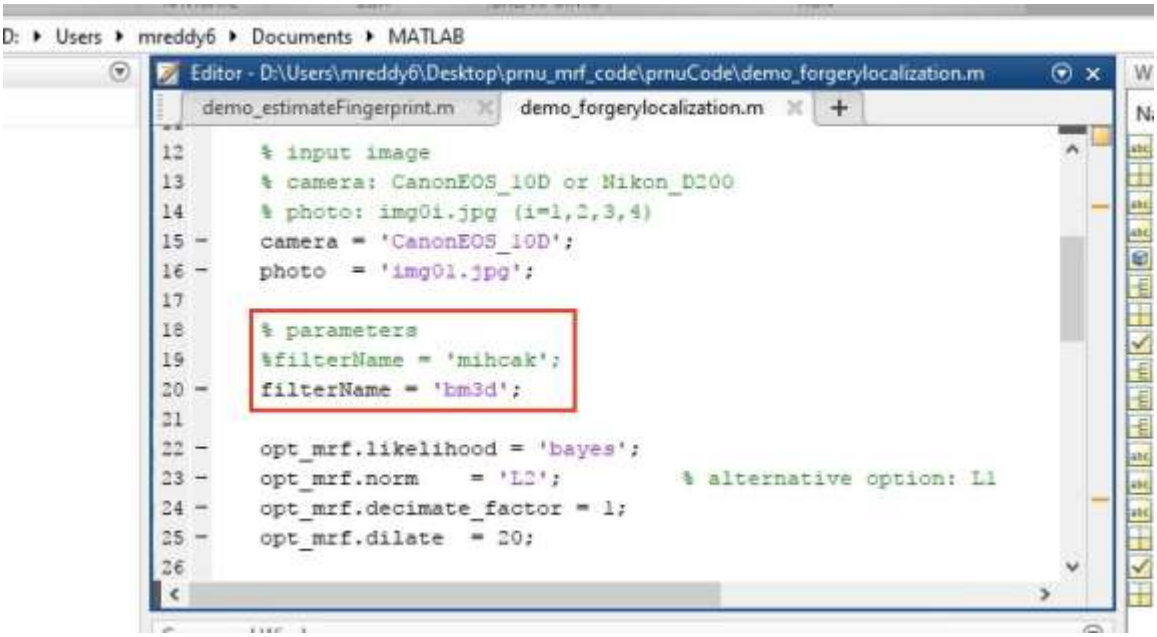
After mask is generated in the **documents\Matlab** folder copy it to the mask folder under **prnucode\photos\CanonEOS_10D\masks** for Canon camera and under **prnucode\photos\Nikon_D200\masks** for Nikon camera.

Once export is completed, specify the **camera folder** name in **demo_forgerylocalization.m** file from which it will select the images and the **image name** as shown in the below screenshot. In the below screenshot

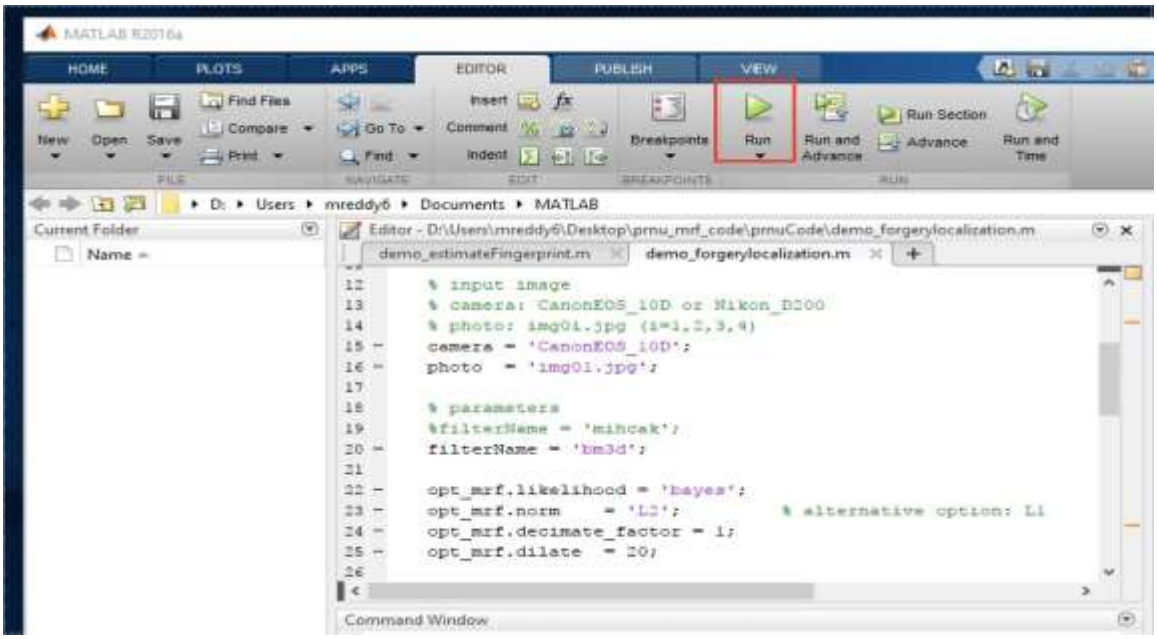
Camera folder: 'CanonEOS_10D'
Photo: 'img01.jpg'



We can change the denoising algorithm either as **'bm3d'** or **'mihcak'** shown below,



Click on the **Run** button as shown below,



 After running the program, it will display the **results** as shown below,

Output consists of Original image, Tampered image and output with accuracy where red area specifies the error portion it has detected, grey area is the missed portion, green is portion it has detected and white portion is actual tampered portion.