

# BACKGROUND CHANGING USING THRESHOLDING ON MATLAB



**Ahsan Ali Kamlo**  
**2K18-IT-15**

## Project Report

# Background changing using thresholding on MATLAB

<https://github.com/Developer-AhsanAli/Computer-Vision-Project.git>

## Introduction

This division process is called image segmentation and thresholding is one of the popular techniques for image segmentation. It has low computational cost when compared to other algorithms. Image thresholding works on the principle of pixel classification. It divides an image into segments depending upon the pixel attributes. This technique applies on each pixel and by comparing it to a specific threshold value decides whether the picture belongs to an object or background.

## Background

In this project I have used thresholding technique to change the background of my image on the basis on color extraction model on MATLAB.

By using thresholding technique on MATLAB, I have created filtered mask of image and generated its function and applied that function on my image to create mask and super imposed the compliment mask filter with other image to change its background.

## Problem Analysis

Thresholding is a type of image segmentation, where we change the pixels of an image to make the image easier to analyze. In thresholding, we convert an image from color or grayscale into a binary image, i.e., one that is simply black and white.

Image thresholding is a simple, yet effective, way of partitioning an image into a foreground and background. It is the most effective in images with high levels of contrast.

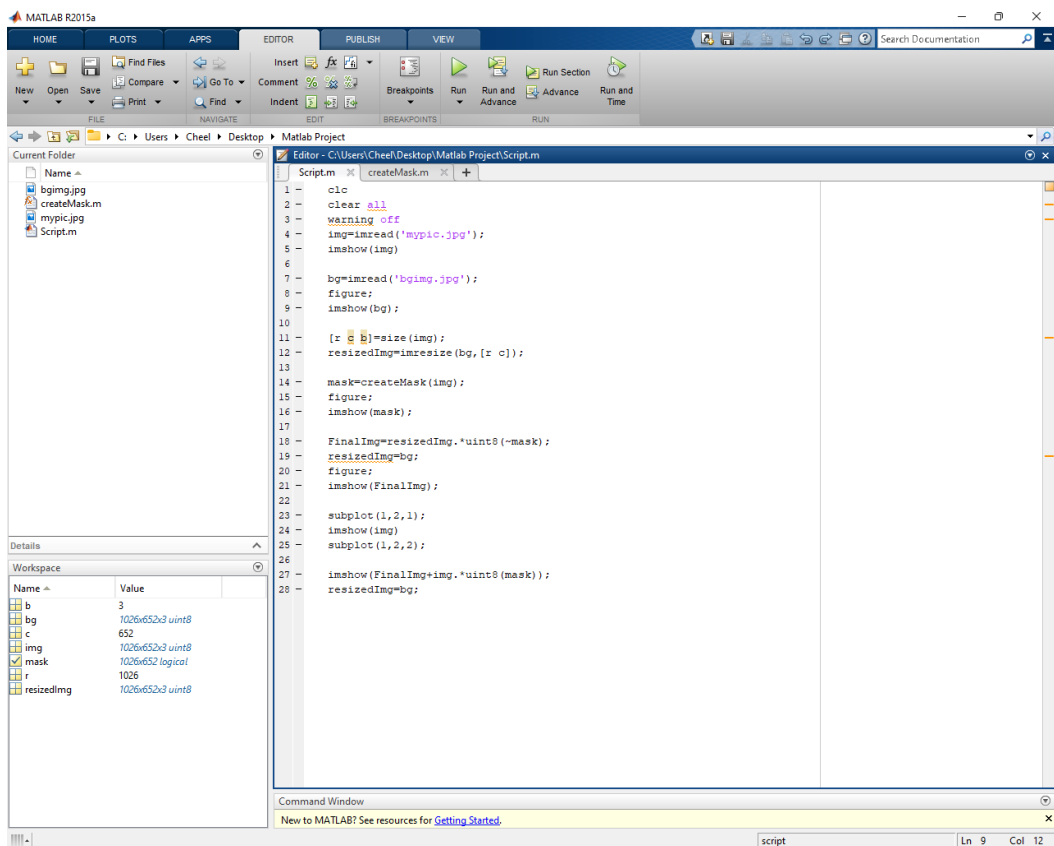
## Implementation

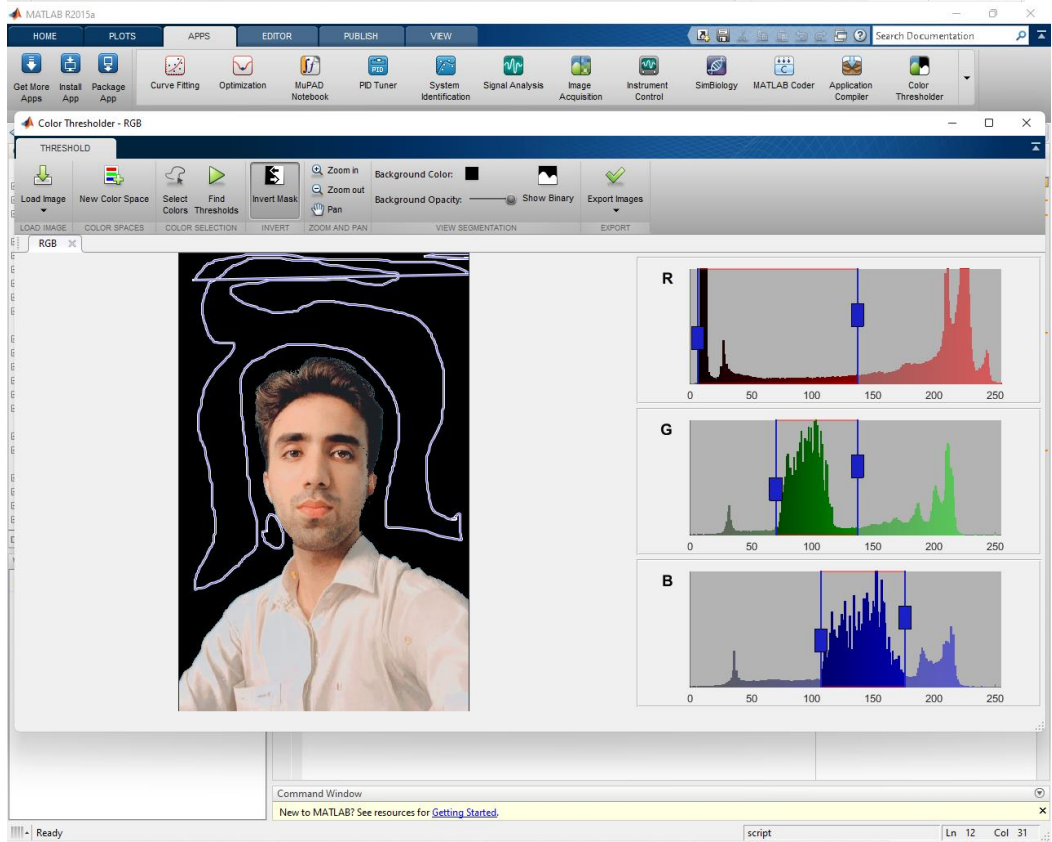
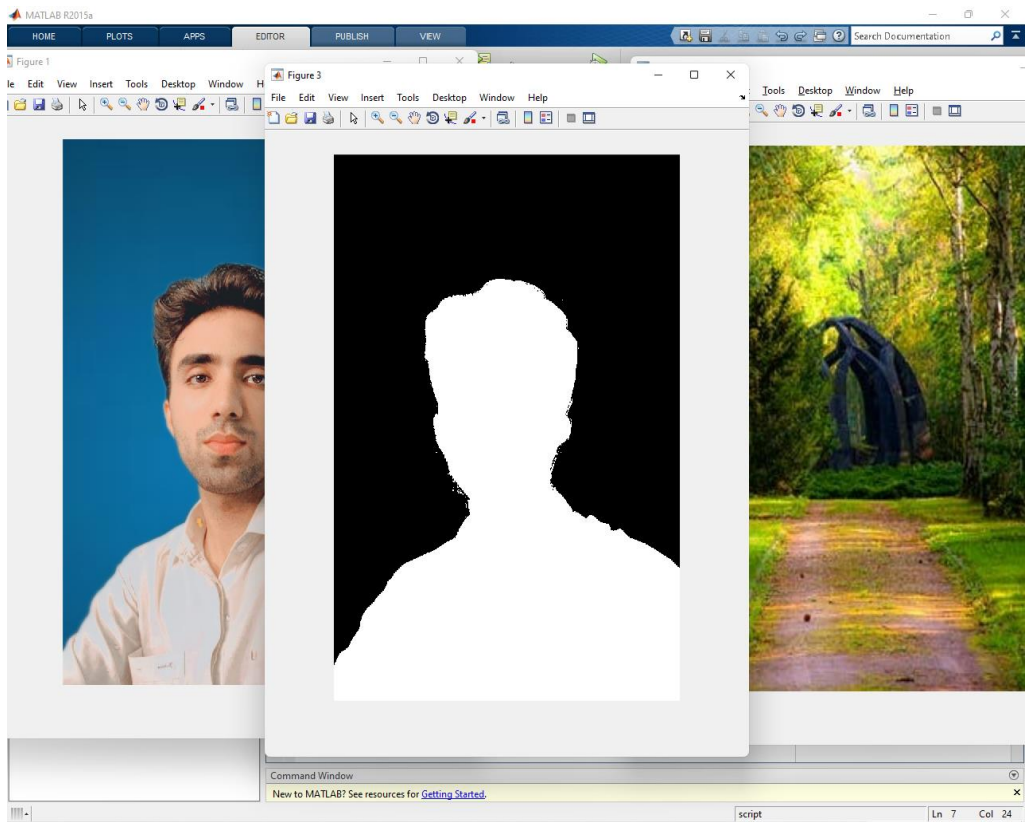
- Selected image for making its mask
- Selected image for background
- Resized both images to be of same size
- Applied Thresholding on image and created Mask function
- Applied that mask function on the image
- Multiplied compliment of filtered image with background image to get the final image

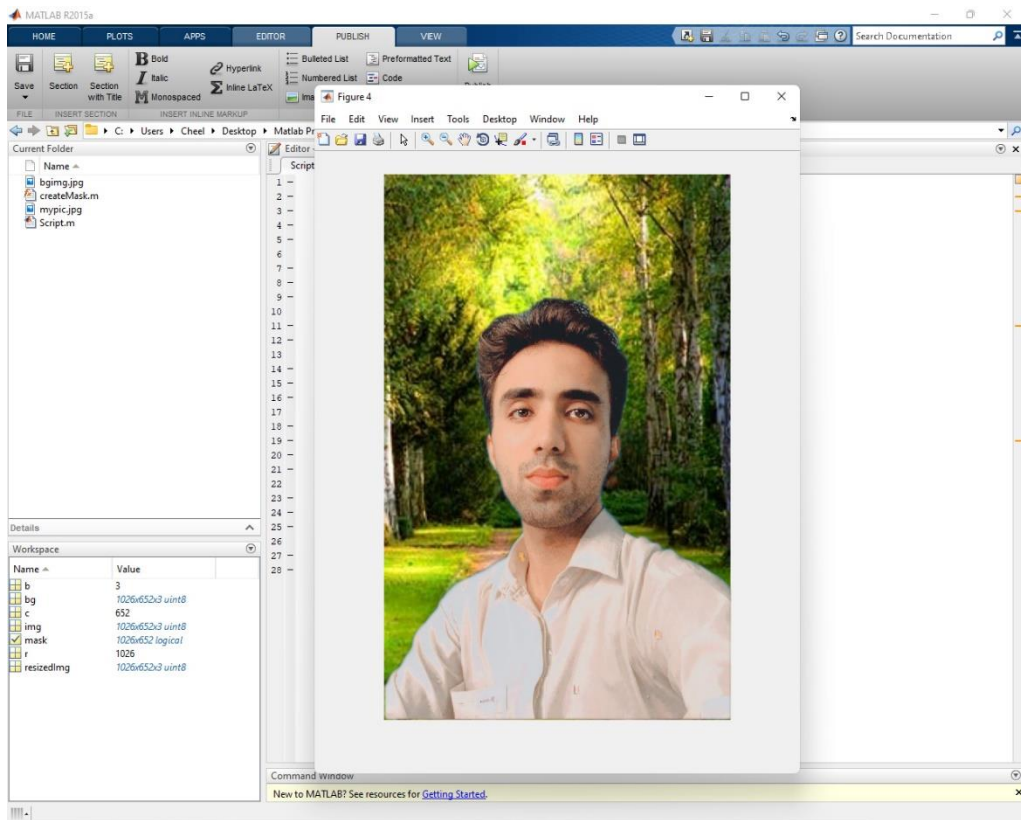
## Conclusion

By using Thresholding technique in MATLAB I have easily changed the background of my image without using any third party application like Photoshop.

## OUTPUT







## Script Code

```
clc
clear all
warning off
img=imread('mypic.jpg');
imshow(img)

bg=imread('bgimg.jpg');
figure;
imshow(bg);

[r c b]=size(img);
resizedImg=imresize(bg,[r c]);

mask=createMask(img);
figure;
imshow(mask);

FinalImg=resizedImg.*uint8(~mask);
resizedImg=bg;
figure;
imshow(FinalImg);

subplot(1,2,1);
imshow(img)
subplot(1,2,2);

imshow(FinalImg+img.*uint8(mask));
resizedImg=bg;
```

## CreateMask Function Code

```
function [BW,maskedRGBImage] = createMask(RGB)
%createMask Threshold RGB image using auto-generated code from
colorThresholder app.
% [BW,MASKEDRGBIMAGE] = createMask(RGB) thresholds image RGB using
% auto-generated code from the colorThresholder App. The colorspace and
% minimum/maximum values for each channel of the colorspace were set in the
% App and result in a binary mask BW and a composite image maskedRGBImage,
% which shows the original RGB image values under the mask BW.

% Auto-generated by colorThresholder app on 26-Jul-2021
%-----

% Convert RGB image to chosen color space
I = RGB;
```

```

% Define thresholds for channel 1 based on histogram settings
channel1Min = 6.000;
channel1Max = 90.000;

% Define thresholds for channel 2 based on histogram settings
channel2Min = 64.000;
channel2Max = 118.000;

% Define thresholds for channel 3 based on histogram settings
channel3Min = 99.000;
channel3Max = 174.000;

% Create mask based on chosen histogram thresholds
BW = (I(:,:,1) >= channel1Min ) & (I(:,:,1) <= channel1Max) & ...
      (I(:,:,2) >= channel2Min ) & (I(:,:,2) <= channel2Max) & ...
      (I(:,:,3) >= channel3Min ) & (I(:,:,3) <= channel3Max);

% Invert mask
BW = ~BW;

% Initialize output masked image based on input image.
maskedRGBImage = RGB;

% Set background pixels where BW is false to zero.
maskedRGBImage(repmat(~BW,[1 1 3])) = 0;

```