EECS: 332 Intro to Computer Vision, MP #1

Connected Component Labeling, Due on 09/28/2017

Karan Shah

Introduction

Connected Component Labeling is a Labeling technique where subsets of connected components are labelled uniquely based on a given heuristic. It is an algorithmic approach of Graph Theory. We develop an algorithm which traverses through the image, labeling pixels based on connectivity and relative values of their neighbors.

Algorithm to perform CCL

ccl(input_image) = [label_img, label_num]

input_image is the input binary image
label_img is the output labelled image
label_num is the number of labels in the labelled image

1. Iterate over every pixel of the image. There are **two passes** to the procedure.
2. Only **foreground pixels** are checked while background pixels are ignored.
3. While checking one pixel, check if the pixels to its **Upper** and **Left** side are both zero, if so, then assign a new value to this pixel in the Output Image, say "**L**" and **increment** the value of "L"
4. While checking a pixel, if either its Upper or Left Pixel is zero, while the other is not zero, set the non-zero value to the value of the pixel in the Output Image.
5. While checking a pixel, if its Upper and Left Pixel are not equal to zero, but equal, then set the value of the Label Image Pixel to be the value of One of them.
6. But, in the previous step, if the Upper and Left pixels are not equal, then we need to create a Mapping Table. Where we map the maximum value between the Upper and Left neighbor to the minimum value between them.
7. The process is repeated to create a mapping table which maps from one pixel to another.
8. Now, in the second pass, we go over each pixel and recursively check if there are any other mappings to that pixel. Eg: If "5" is mapped to "4", but "4" is mapped to "2". We ensure that "5" is mapped to "2"
9. When the pixel values are updated, then the procedure is complete.
10. To display the number of distinct labels, the number of unique values in the lookup table can be returned and displayed. In MATLAB, we can use the function **(size (unique (label_img))**
11. For images which have a lot of background noise, then we can use a size filter. The size filter removes small objects whose size is less than a certain number of pixels
12. This Size Filter is used while running the code on gun.bmp

Result Analysis

As you can see in Fig 1, The function mat2gray has been used, which converts a matrix to a grayscale image. Therefore, we can see the slight differences in Color between the various grayscale values. The different regions have been marked with the red regions. Here, the background (Pixels with value 0) are labelled as well. Therefore, there are 7 different labels in this image.

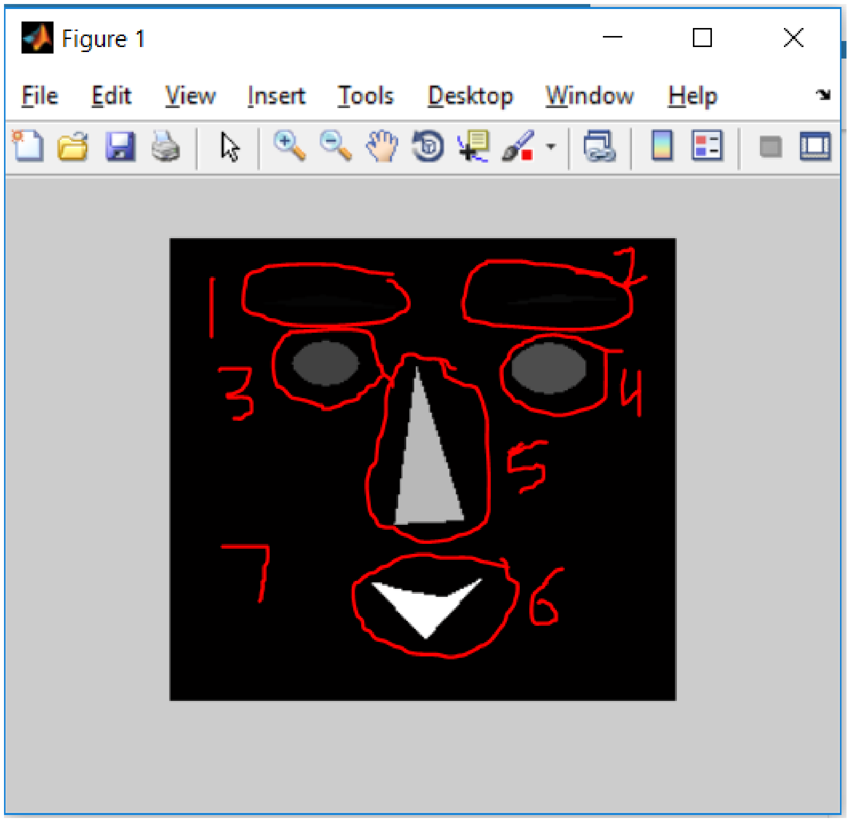


Fig 1. Output of Running CCL on Face.bmp

Also, the second return statement “label_num” is also shown. Which is returning the value of “7”. Thereby proving that there are 7 Labels in the image.

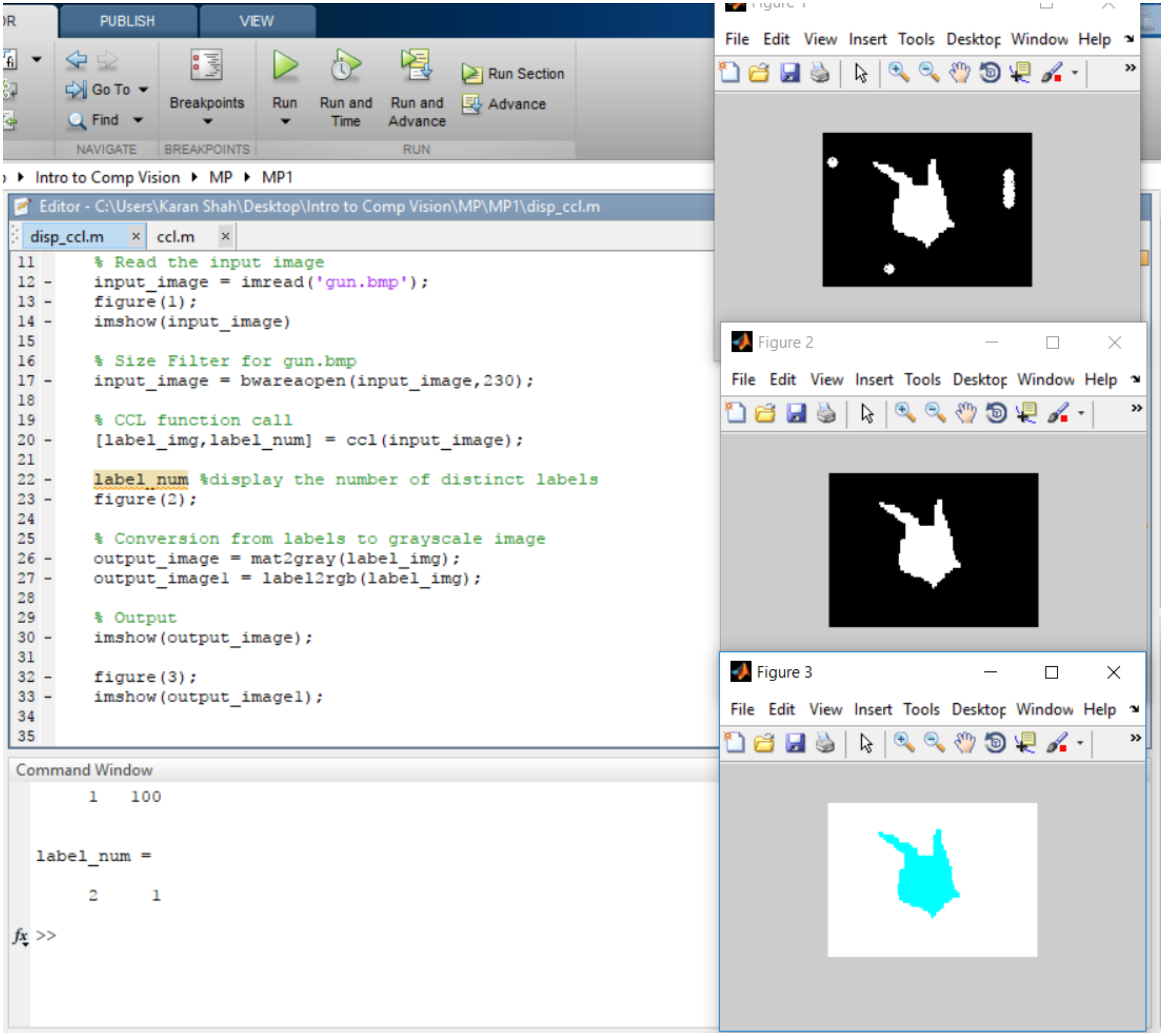
```
label_num =

     7     1
```

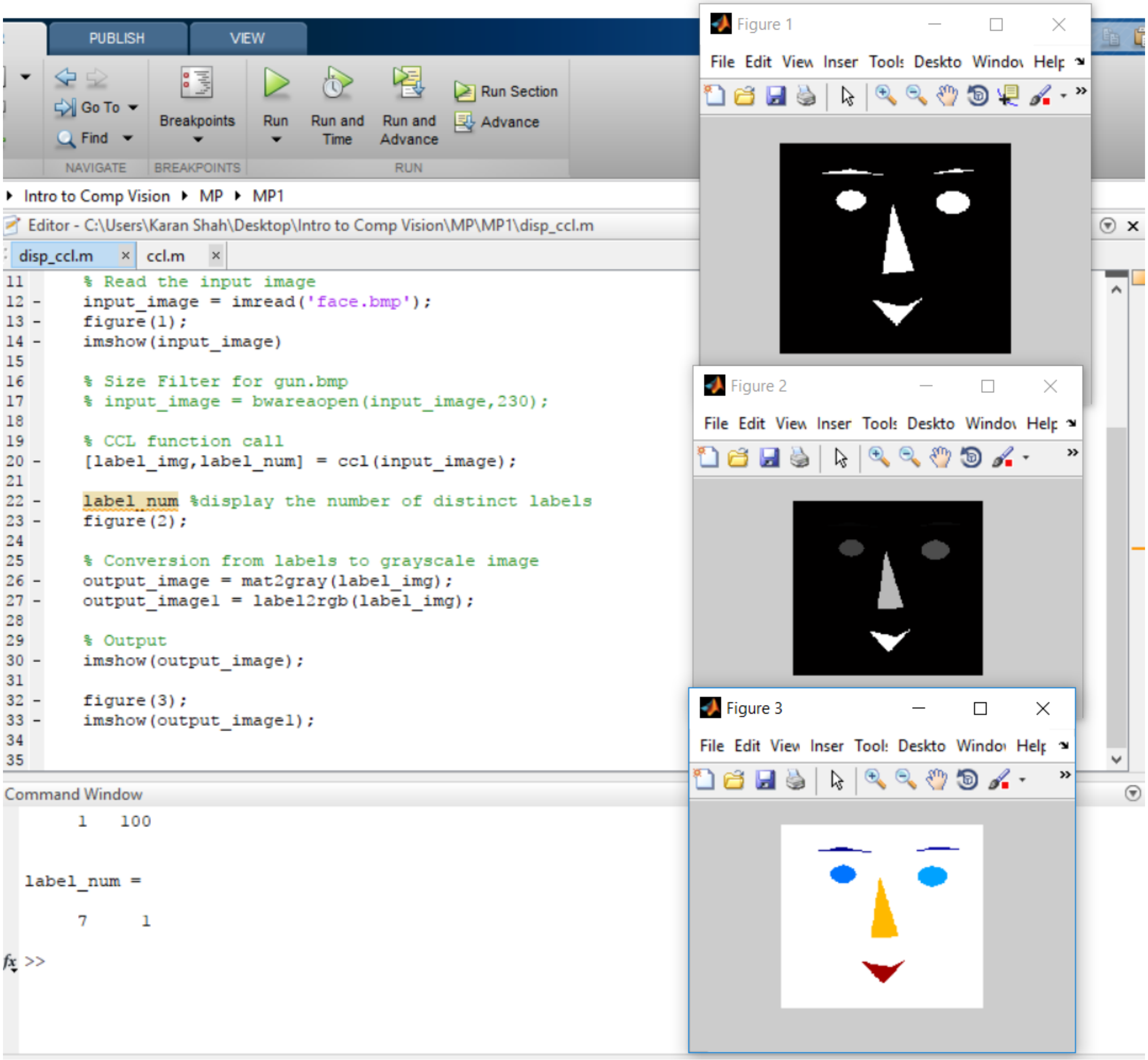
Fig 2. Number of Labels in the Image

MP#1 Results for test images

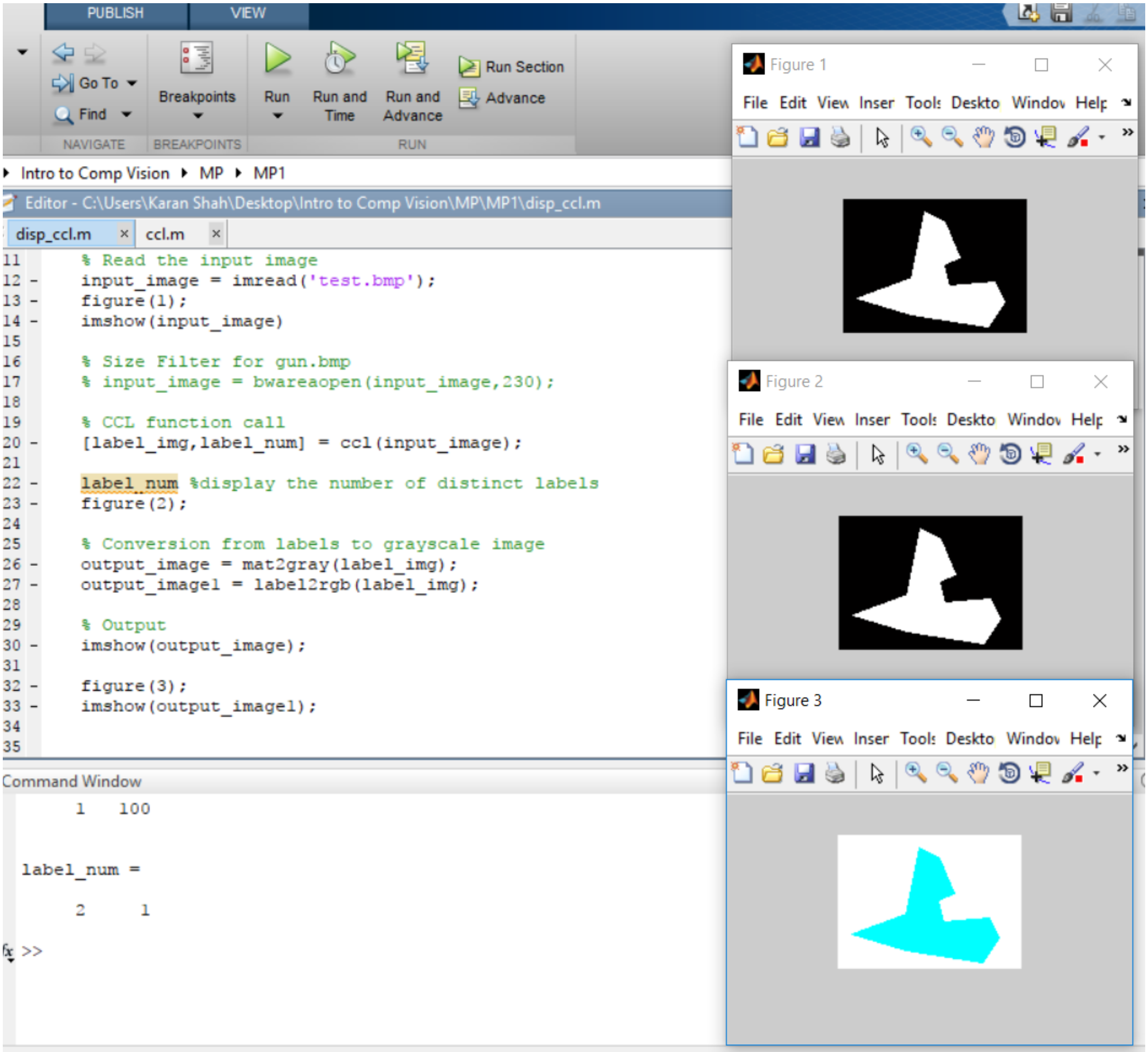
1. Gun



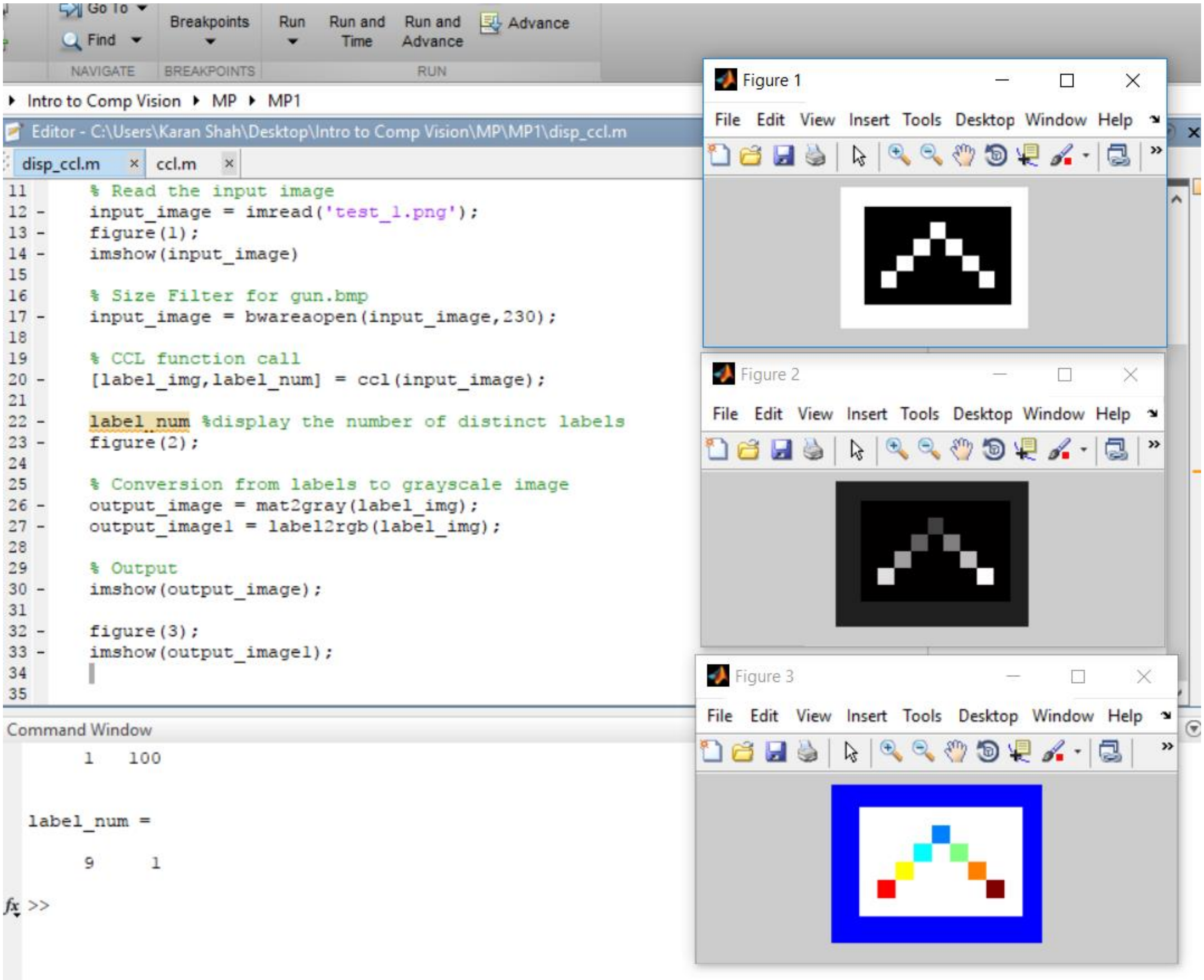
2. Face


PUBLISH
VIEW
Go To
Find
Breakpoints
Run
Run and Time
Run and Advance
Run Section
Advance
NAVIGATE
BREAKPOINTS
RUN
Intro to Comp Vision ▸ MP ▸ MP1
Editor - C:\Users\Karan Shah\Desktop\Intro to Comp Vision\MP\MP1\disp_ccl.m
disp_ccl.m
ccl.m
% Read the input image
input_image = imread('face.bmp');
figure(1);
imshow(input_image)
% Size Filter for gun.bmp
% input_image = bwareaopen(input_image,230);
% CCL function call
[label_img,label_num] = ccl(input_image);
label_num %display the number of distinct labels
figure(2);
% Conversion from labels to grayscale image
output_image = mat2gray(label_img);
output_image1 = label2rgb(label_img);
% Output
imshow(output_image);
figure(3);
imshow(output_image1);
Command Window
1 100
label_num =
7 1
Figure 1
Figure 2
Figure 3

3. Test


PUBLISH
VIEW
Go To
Find
Breakpoints
Run
Run and Time
Run and Advance
Run Section
Advance
NAVIGATE
BREAKPOINTS
RUN
Intro to Comp Vision ▸ MP ▸ MP1
Editor - C:\Users\Karan Shah\Desktop\Intro to Comp Vision\MP\MP1\disp_ccl.m
disp_ccl.m
ccl.m
% Read the input image
input_image = imread('test.bmp');
figure(1);
imshow(input_image)
% Size Filter for gun.bmp
% input_image = bwareaopen(input_image,230);
% CCL function call
[label_img,label_num] = ccl(input_image);
label_num %display the number of distinct labels
figure(2);
% Conversion from labels to grayscale image
output_image = mat2gray(label_img);
output_image1 = label2rgb(label_img);
% Output
imshow(output_image);
figure(3);
imshow(output_image1);
Command Window
1 100
label_num =
2 1
>>
Figure 1
Figure 2
Figure 3
File Edit View Inser Tools Deskto Window Help

4. Test – 1


Go To
Find
Breakpoints
Run
Run and Time
Run and Advance
Advance
NAVIGATE
BREAKPOINTS
RUN
Intro to Comp Vision ▸ MP ▸ MP1
Editor - C:\Users\Karan Shah\Desktop\Intro to Comp Vision\MP\MP1\disp_ccl.m
disp_ccl.m
ccl.m
% Read the input image
input_image = imread('test_1.png');
figure(1);
imshow(input_image)
% Size Filter for gun.bmp
input_image = bwareaopen(input_image,230);
% CCL function call
[label_img,label_num] = ccl(input_image);
label_num %display the number of distinct labels
figure(2);
% Conversion from labels to grayscale image
output_image = mat2gray(label_img);
output_image1 = label2rgb(label_img);
% Output
imshow(output_image);
figure(3);
imshow(output_image1);
Command Window
1 100
label_num =
9 1
>>
Figure 1
Figure 2
Figure 3
File Edit View Insert Tools Desktop Window Help