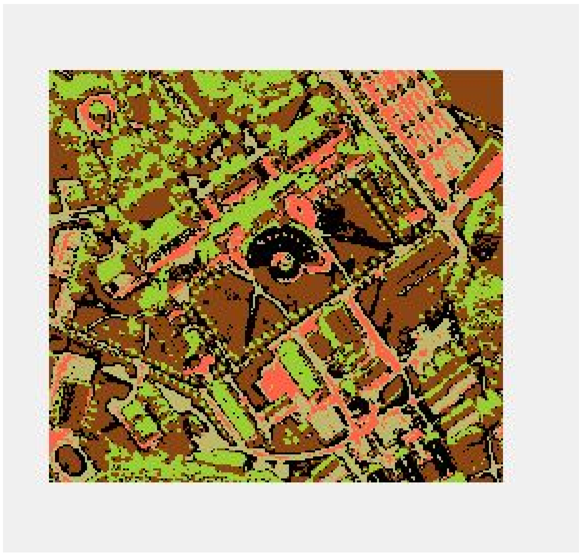
Practical 3 - Part 1

Q1. What interval of intensities did you choose and why?
We decided the intensity intervals based on the histogram of the original image. There were four peaks in the histogram and we chose the three biggest peaks in descending order to decide the intensity interval.

Q2. What colors did you choose to set into the new image? Provide a screenshot of the result.

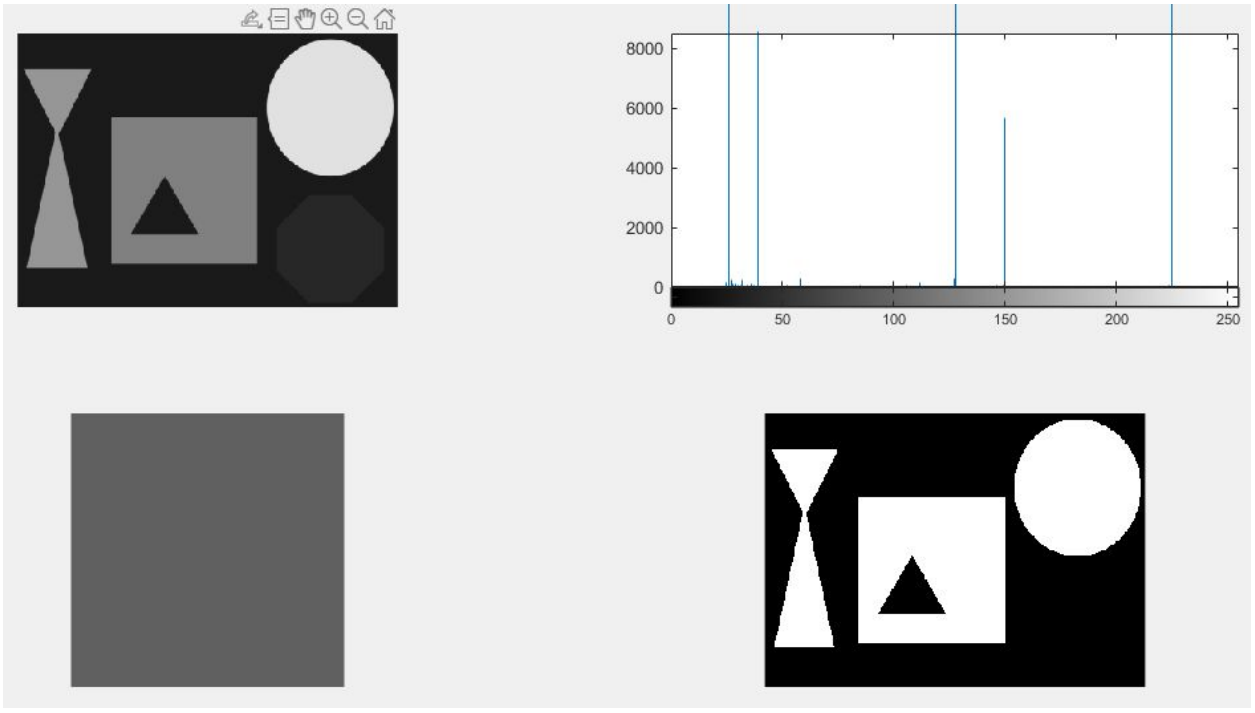
We chose tomato for the residential buildings; yellow green for the trees; dark khaki for the play area around the residential buildings and saddle brown for the big buildings (mostly commercial).
Screenshot shown below.



Q3. Why is it convenient to perform color finding in HSV domain?
It is because the color is separated from its intensity and saturation value. If we pick Hue (which is the first channel of HSV image), we change only the color keeping S and V channels intact.

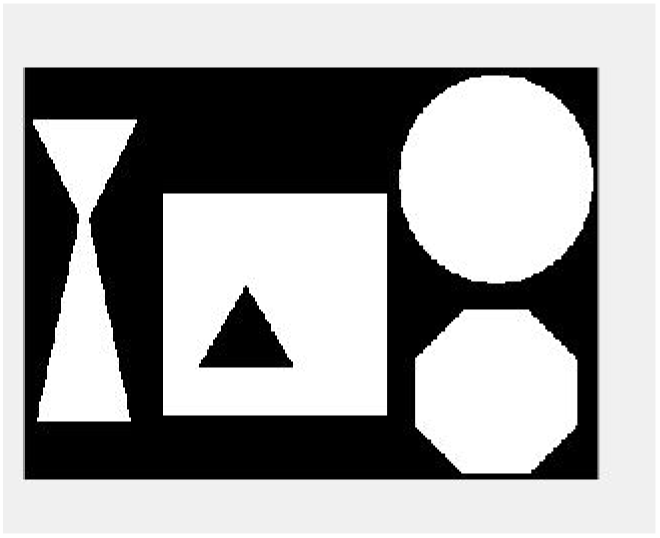
Q4. What value is strong red in H1 and S1?
It is H1 = 0 and S1 = 0.9

Q5. See the results and investigate the obtained foreground imagery, did you get all the objects by this threshold?
Yes, I did. 'Graythresh' chose a threshold that minimized the intraclass variance of the thresholded black and white pixels. The result is a white image with black background, see below.



Q6. What threshold value of gt is used to correctly separate background from foreground?

Looking at the histogram, the darkest image is at X =27, which when normalized gives X = 27/255 = 0.1058.
Substituting gt with this value in bn = imbinarize(I,0.1058) gives below result



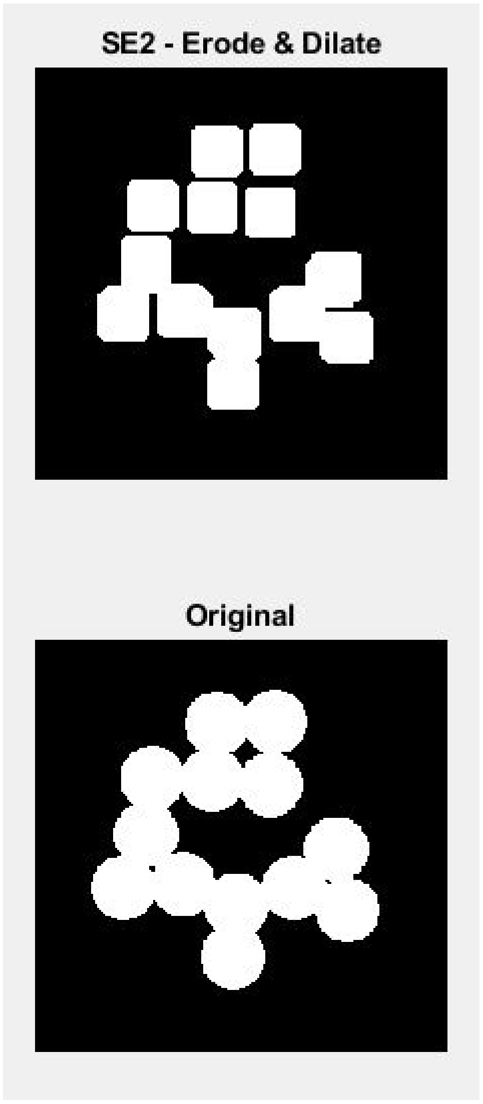
Q7. What is stored in variable num, see help for bwlabel()?
Num stores the number of connected objects found in the black and white image.

Q8. What is the difference between the build in opening filter, which is stored in variable op and the one we created ourselves and stored in variable d? See the resulting image and explain the difference?
Both are the same thing i.e. op is an inbuilt function in Matlab for opening filter. Whereas d is created by first eroding the image and then dilating it which is the definition of opening filter. Thus d is a manual opening filter.

Q9. Describe the structuring element you have stored in variable SE2 and explain the result of erode and dilate with this element.

SE2 is a square structuring element with initial width 3 pixels. The underlying image has circular objects to be detected. Hence the structuring element shape is not ideal.

Also, with a square width of 3 pixel, the erosion function detects the original image as one image. We need to increase the pixel width of the square structuring element to 25 to detect the 13 circles in the original image. With SE2=strel('square', 25), we get below image after erosion and dilation.



Q10. Why is there a difference between the number of components found with SE and SE2.

It is because the SE uses a disc structuring element with radius 14 which is approximately equal to the object in the underlying image. SE2 uses a square structuring element with width much lesser than the radius of the underlying object. (If we increase the width to a much larger number, then the underlying objects start being detected, although not matching the original shape)

Q11. What size of the structuring element makes the edge detection most satisfactory? What happens with the edges if we increase the size of the structuring element?

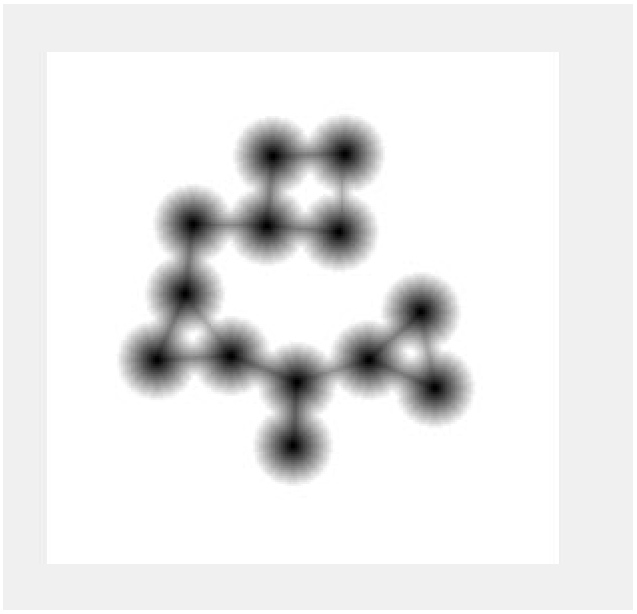
The edges start being clear at the structuring element size 5 for the siluette image. If we increase the size of the structuring element, the edge starts getting thicker and thicker.

For circles, we start getting complete edges from structuring element size 3 already.

Q12. Why should you invert the image?

To perform morphological operations, it is important to have the image foreground as white and background as black. The given image is in opposite colors, hence we invert the image first to make it ready for a morphological filter.

Q13. Study the figures produced by the code. Why do we need the step where we calculate D = -D?
We calculate D = -D to make the background as the underlying object and start working on it. After D = -D, the background will be white as below.



This operation is needed because we want to make the background pixels as infinity i.e. not to include them in our calculation.

Q14. What do the darkest areas inside the blobs represent?
The darkest areas inside the blobs represent the local minima of the catchment basins.

Q15. Was the algorithm successful in segmenting the objects? If not, can you suggest some pre-processing to improve it?

No the algorithm was not successful in segmenting the objects. We add the morphological opening filter before we start calculating the Euclidean distance with command ‘bwdist’.

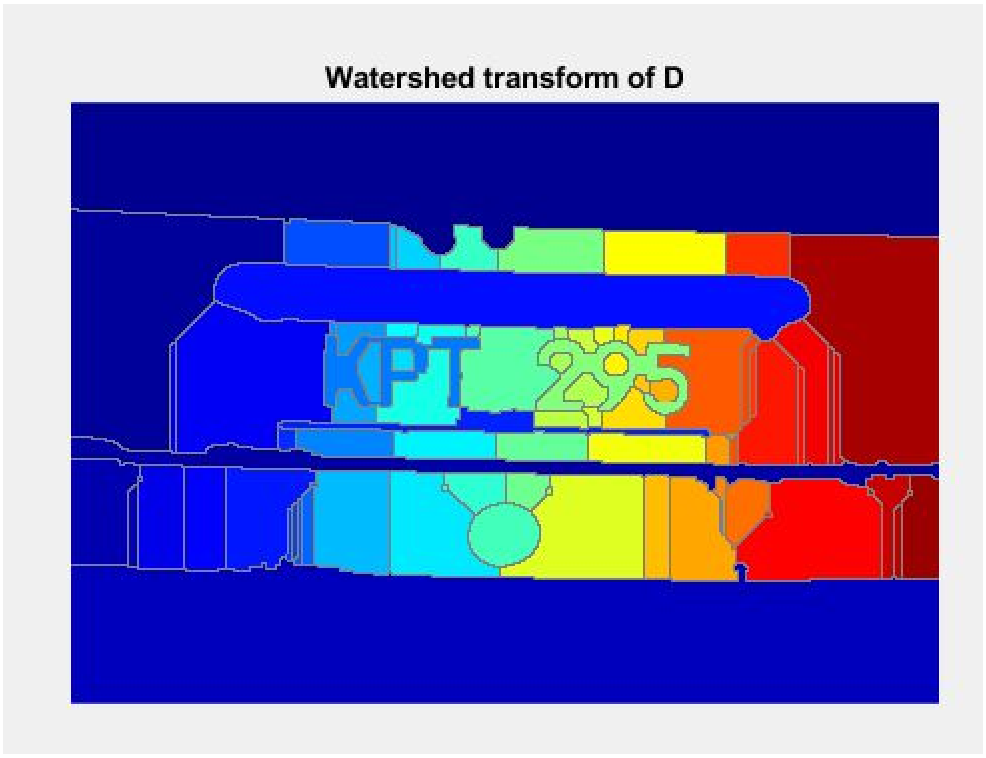
The additional code is as below

```
SE = strel('square', 6);
e=imerode(bw, SE);
d=imdilate(e, SE);
```

[labeled, num]=bwlabel(e, 8)

D = bwdist(~d); and so on…

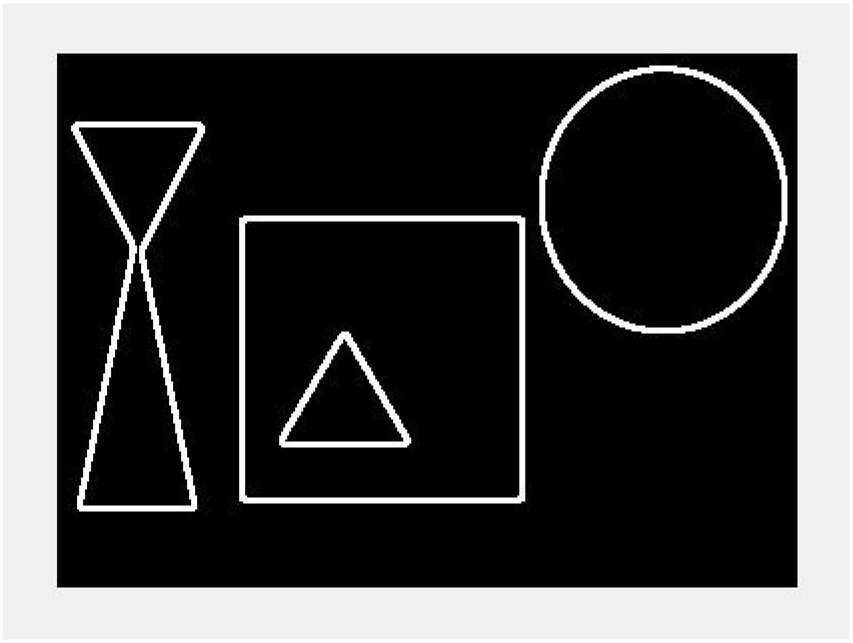
It gives 5 separate elements as below. I personally see the car has 5 separate objects hence I chose the structuring element size to 6 which gives 5 elements.



Q16. Where in the code you decide the amount of objects to be segmented. Try this code with any components image. What is the result? Could you successfully segment the image?

The objects to be segmented are decided in below lines of code.
for d=17:38
    [r,c]=find(L==d); and so the code continues..

Trying this code with 'components1.png' image gave below results. Note we changed the d values to *d=1:4* in the above code because the max(L) values are between 0 and 4.



The 5th object hexagon is not detected in this image because we cannot choose L between 0 and 1 as it is a coordinate/position.