

HOMEWORK – 7A

NOTE:- Put folder in the same working directory

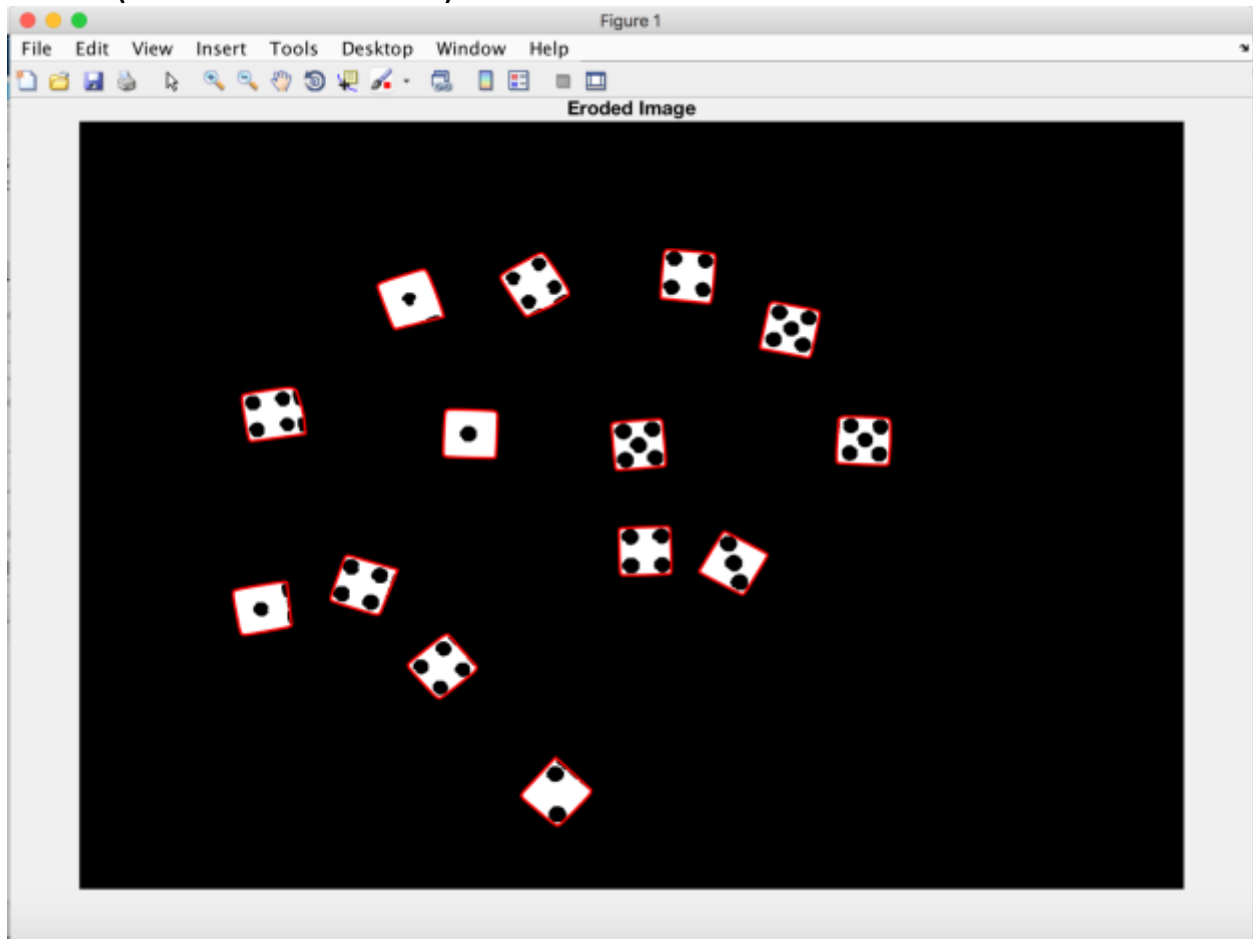
FOLDER OF IMAGES SHOULD NOT BE IN TEST_IMAGES RATHER .m AND FOLDER SHOULD BE AT SAME LEVEL

Image chaining \ Steps followed to count dice and dots on dice:

1. First parse if given parameter is file or folder. If its file start from step 2 else for every file in folder start from step 2.
2. Read the image.
3. If the image is tall, rotate it to get the landscape view.
4. Company's bicycle logo is red so extracted red channel.
5. Applied Otsu's method(graythresh) to find threshold.
6. Used this threshold to obtain a binary image. This will remove some noise from the image.
7. To remove small white spots(noise) applied erosion on the binary image using Morphological structure of type disk and size 11. This will also separate if 2 dices have touching edges.
8. Used bwlabeled to extract all the connected components in the image. These components are dice and remaining may be noise that we will handle later by putting threshold on area.
9. Using regionprops measured properties like Area, ConvexHull and BoundingBox.
 1. ConvexHull – To get coordinates of polygon to draw red boundary.
 2. BoundingBox – To get boundaries of the dice which I am using to crop the image and select only region which has dice.
 3. Area - To threshold the min and max area to minimize false dice selection.
10. For every component obtained by bwlabeled, we first try to determine if it is a dice or not. I have used min and max area threshold to determine if it is a dice or not.
11. If it is a dice then we increase the dice count and start calculating dots on the face of dice.
12. To calculate dots, I cropped the binary image to get the portion containing the dice.
13. On the cropped image, I have applied dilation with morphological structure size as 22. We need this larger value of the morphological structure because we are dealing only with dice.
14. By doing this it will apart the dots on face of dice if they are connected.
15. I took imcomplement on the image obtained in step 13 to convert white dice into black and black dots into white. I did this so that I can apply bwlabeled directly on the image to extract white dots.
16. If number of dots is zero or greater than six then we add the count of unknown. This scenario may come up when two dice are connected.
17. Else increase the count respectively.
18. Repeat step 11 to 16 for all dice.
19. I have used cell array to store the statistical data(output).
20. Displayed the output.

So, overall I have learnt so many new things from this homework. It gave me good idea about how morphology is used. Learnt to use so many new methods/functions like bwlabeled, regionprops, graythresh, imcrop. In a nutshell, after removing some noise I tried to identify the number of dice and then cropped every dice to form a separate image of dice and worked on this separate image to count dots on the face of dice one by one. Have done this for all the dice. At the end processed the output to print it in the manner required. Used cell array to store the final output and displayed it.

See output on next page...

OUTPUT (IF FOLDER NAME IS GIVEN)

I Its a directory so program will run for all the files in dir

INPUT FILE is set_0/img_6858__unk.jpg

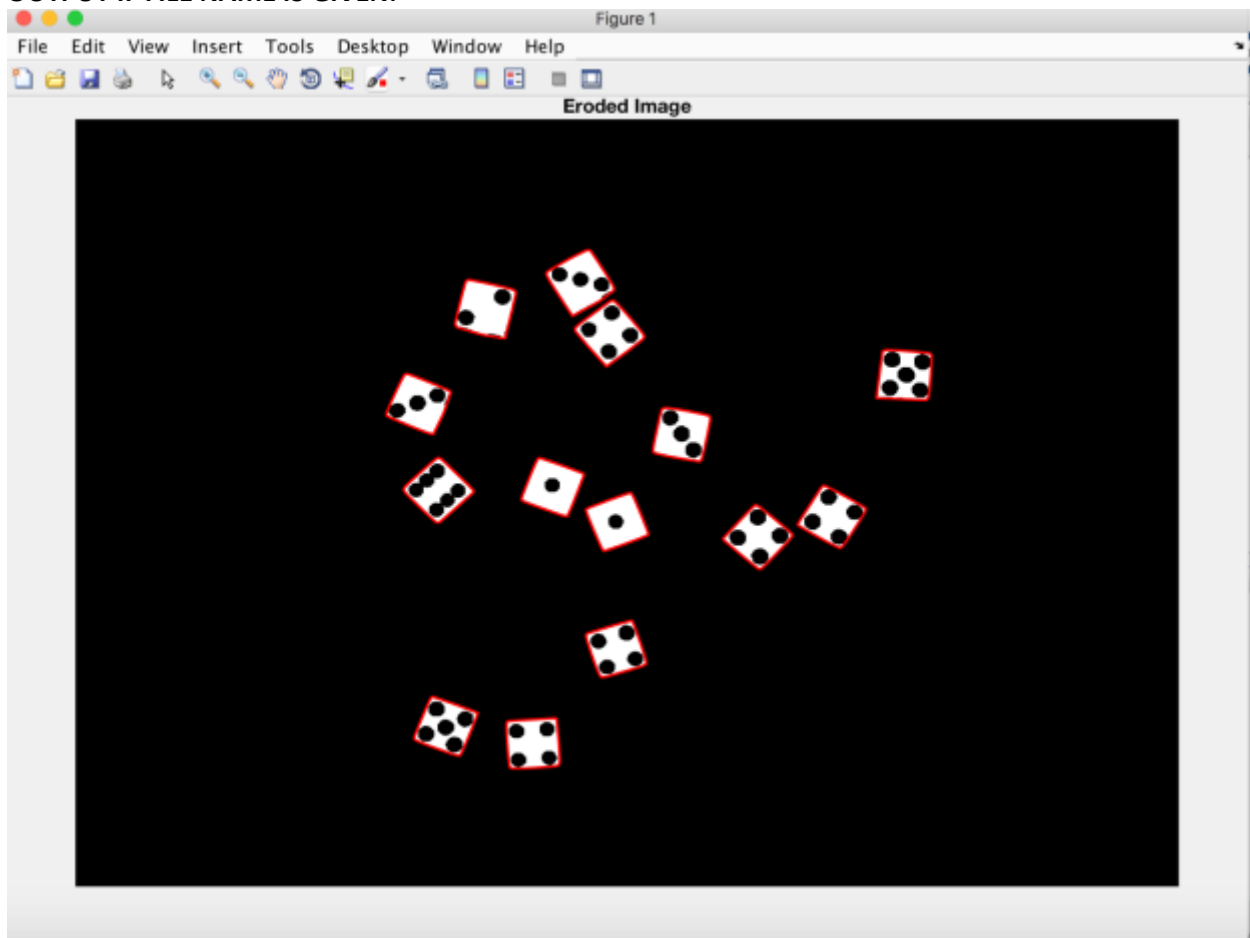
Number of dice/dices :14

cell =

9×2 cell array

'NUMBERS'	'COUNT'
'Number of ones'	[3]
'Number of twos'	[1]
'Number of threes'	[1]
'Number of fours'	[6]
'Number of fives'	[3]
'Number of sixes'	[0]
'Number of unknowns'	[0]
'TOTAL DOTS'	[47]

OUTPUT IF FILE NAME IS GIVEN:



Its a file

INPUT FILE is img_7130__unk.jpg

Number of dice/dices :14

cell =

9×2 cell array

'NUMBERS'	'COUNT'
'Number of ones'	[2]
'Number of twos'	[1]
'Number of threes'	[3]
'Number of fours'	[5]
'Number of fives'	[2]
'Number of sixes'	[1]
'Number of unknowns'	[0]
'TOTAL DOTS'	[49]