Faculty of Engineering, University of Jaffna
Department of Computer Engineering
EC1011 – Computing
Lab 08 - Domain Specific Programming Languages

Date: 17 August 2022 **Duration: 3 Hours**

Instructions:

- Any plagiarized work will be given 0 marks.
- Submit your lab work as a pdf file named ***Lab0Z_2021EXXX.pdf*** *(XXX – Your Registration Number & Z is your lab number)* on/before given deadline via team **Teams EC1011** → **Assignments** → **Lab_08**
- Final submission should contain **lab report as pdf** format and all **.m files**.
- Prepare your lab report with the screen shots of your answers.
- Note all images necessary are also provided separately.
- Include a title with your index number for all images.

Image Processing Toolbox

A color image is just a 3 dimensional matrix, the RGB format which is the common format for images has three layers of 2D matrices. Each layer contains the intensity level variation of the pixels (a pixel is the smallest area of the screen which can emit controlled light independent of similar adjacent portions) across the screen for a particular color. In the RGB format the 3 layers correspond to the primary colors red blue and green. The position of each element on the matrix directly corresponds the physical position of the pixel. The superposition of the colors produced by each layer forms the image. Figure 1 below shows the above description visually.

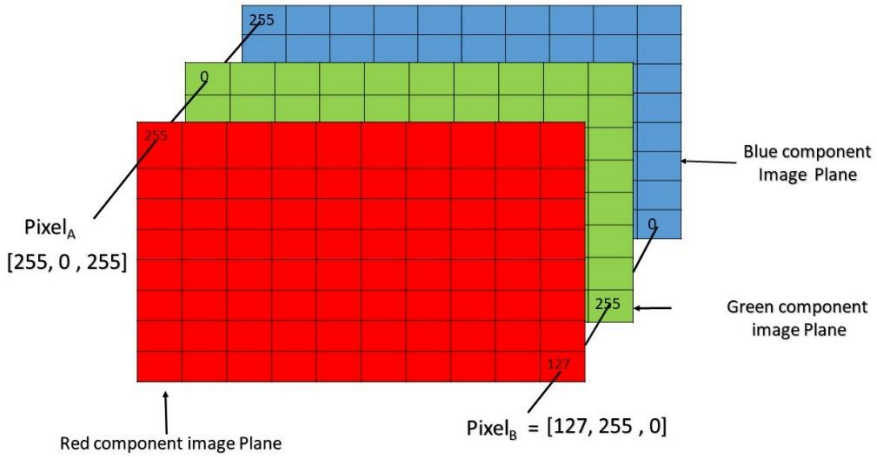

Figure 1: An RGB image -source: geeksforgeeks.org

As illustrated above each pixel contains its intensity level for each pixel for each color plane, commonly 8-bits (bit depth) is used to represent this intensity level. When 8 bits is used the maximum intensity is represented by 255 and the minimum intensity is represented by 0. In MATLAB these values are stored in the data type of unsigned integer of 8 bits (uint8 in short).

Another common format like RGB is the grayscale format. This format ignores the color information it only contains the intensity, this can be obtained in MATLAB by the command **'rgb2gray'**. A comparison of the two types is shown in the figure below.

Figure 2:RGB and grayscale comparison

1. Image Basics

a. Use the command **'imread'** to import Figure 3 (ensure the image is in the current folder of MATLAB and you are using the relevant group image) into the variable A.

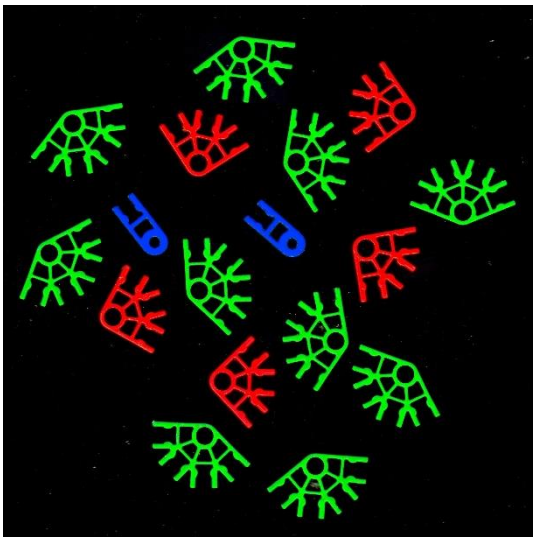

Figure 3: Toy Parts

Hint: Use full name. Example to import an image named test.png into the variable F the command would be,

>> F= imread('test.png'); % remember the semicolon

b. Use **'imshow'** to display the image you stored in A.

c. Show A is a 3 dimensional matrix by using the **'size'** command.

d. The first second and third layers of the image correspond to red, green and blue respectively. The red layer can be extracted and store it in the variable R by the command, **>> R=A(: , : , 1); % the first and second colons (:) mean to keep all elements in the x and y directions respectively and '1' is used to extract the first layer.** Likewise extract the red, green and blue layers in to the variables R, G and B respectively. Then use **imshow** to view each layer separately in a subplot including the three layers and the original image. Observe the objects of the corresponding color are brighter in their respective layer.

e. Obtain the grayscale image, by using the command **'rgb2gray'** and store the output in the variable BW. Then output it in a figure with 'imshow'.

2. Image Rotation

a. Import figures 4 and 5 into the variables P and L respectively.

Figure 4: Portrait Mode

Figure 5: Landscape Mode

b. Rotate the figure in P by 45 degrees in the clockwise and anti-clockwise using the command **'imrotate'** sense and store the results in the variables PCW and PACW respectively. Then using subplot show those images in a single figure with appropriate titles. (**Hint:** B=imrotate(P,70); will rotate P by 70 degrees counter clockwise and store it in B, to rotate clockwise simply use -70.)

c. Flip the figure in P horizontally and vertically using the command **'flip'** and store the results in the variables PFH and PFV respectively. Then using subplot show those images in a single figure with appropriate titles. (**Hint:** B=flip(P,1); will flip P vertically and store it in B, to flip horizontally it will be flip(P,2) instead.)

d. The following table and the figures 6 and 7 illustrate how the image should be depending on the final angle. You have to write a function named **'phoneRotate'** which will input the number of degrees the phone has to be turned (from the portrait position in the clockwise direction, note **input can be negative** for counter clockwise rotation also **input can exceed 360°**) and display the final image of the phone after turning the said degrees. (**Hint: use a combination of rotation and flip or just rotation and some conditional statements.**)

Table 1: Relationship between angle and Orientation

Range of final angle 'x'	Orientation
$0° \leq x \leq 45°$	Portrait
$45° < x \leq 135°$	Landscape
$135° < x \leq 225°$	Portrait
$225° < x \leq 315°$	Landscape
$315° < x \leq 360°$	Portrait

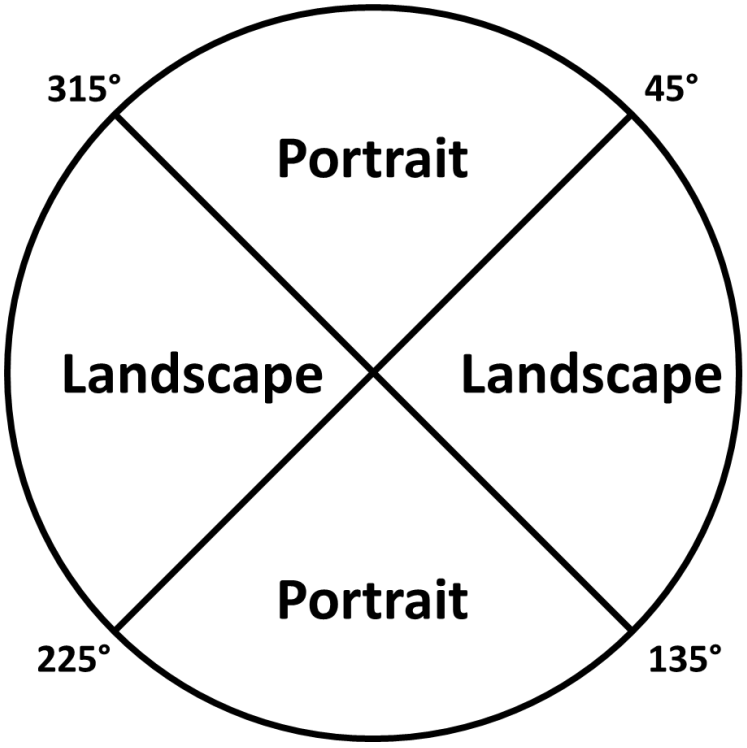

Figure 6: Relationship between angle and Orientation

e. Using the above function create a subplot of output images for 4 rotation angles the **angles you should use are given at the end of the report.**

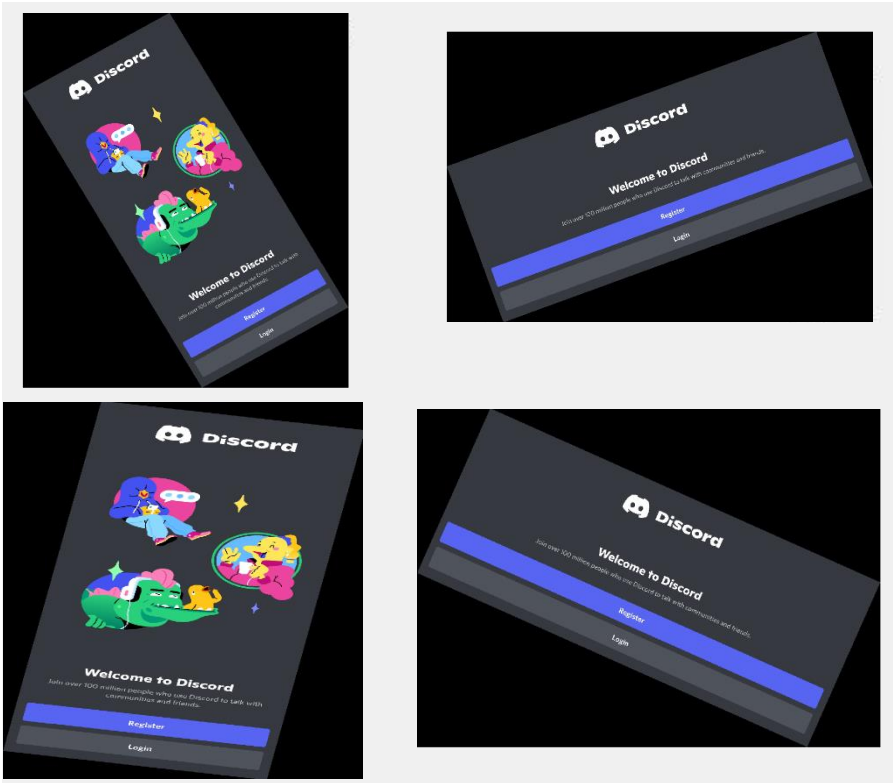

Figure 7: Example Outputs for Phone Rotation

3. Image filters

Image filters are used for various purposes for like blur, sharpen, detect edges and remove noise. One such image filter is the average filter it can be used to blur images.

Average Filer

1. This filter is applied to a gray scale image or applied layer by layer to a color image then combined into one.

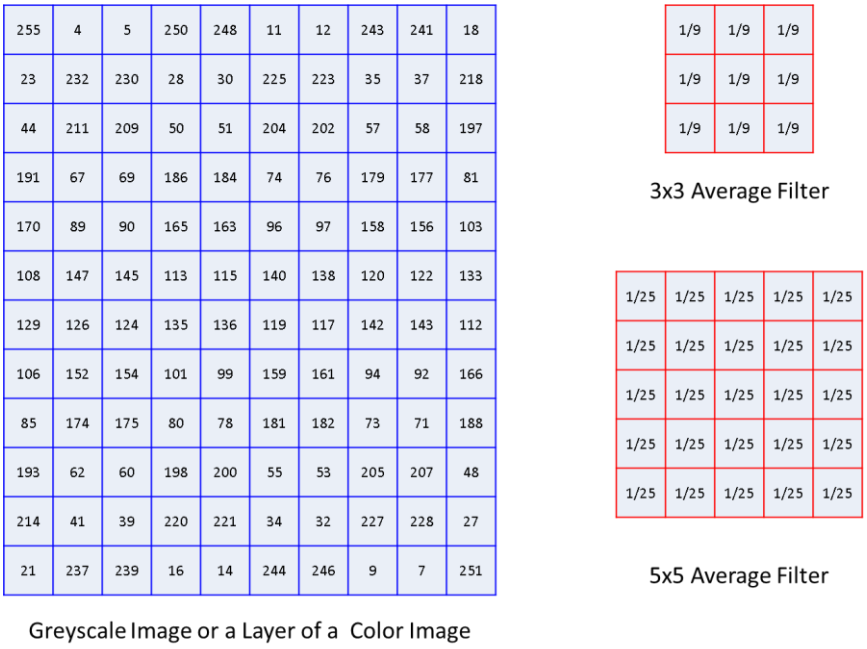

Figure 8: Example image and average filters

2. For this filer the center of a square matrix of odd number of rows is moved on to each pixel. This square matrix is called the filter and is also known as the kernel. Its special properties to be an average filter is that all its elements are the same and the sum of all elements is one.

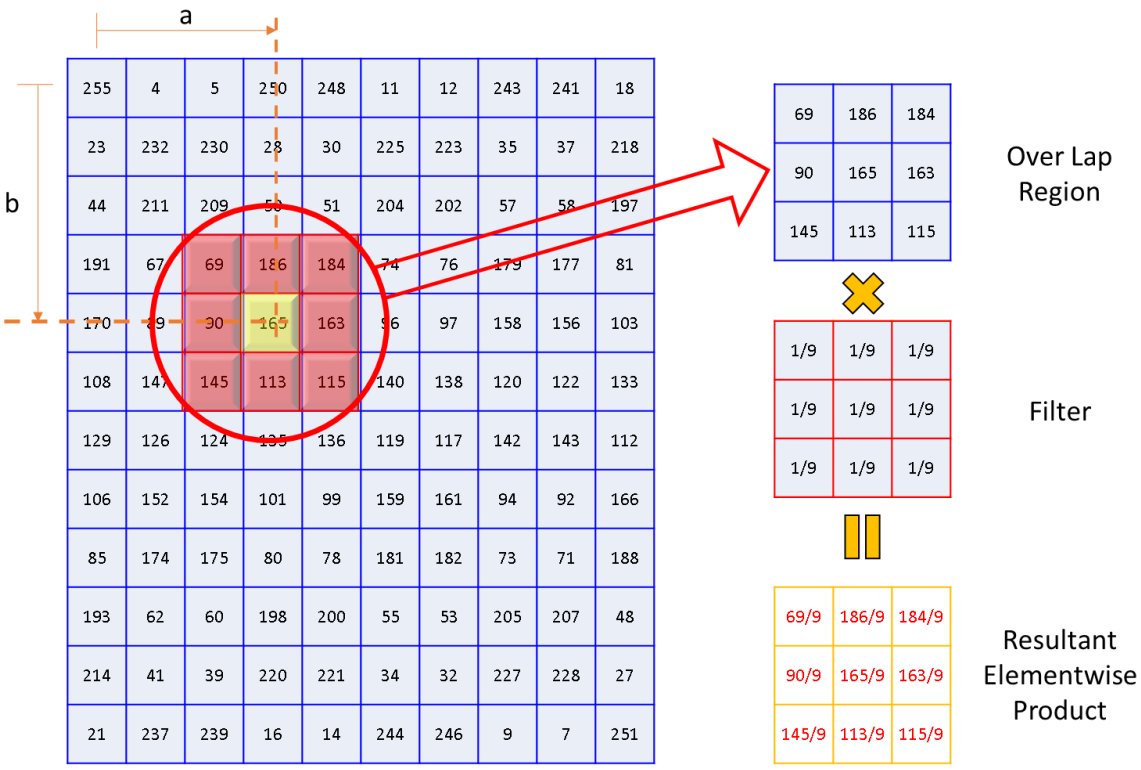

Figure 9: Applying 3x3 Filter to Image at Pixel (a, b)

3. Next elements overlapping in the filter and the image pixel are multiplied elementwise to form a result matrix. The sum of all the elements in the resulting matrix gives the pixel value of the output image's corresponding pixel (Usually the output image is of the same size of the input image.)

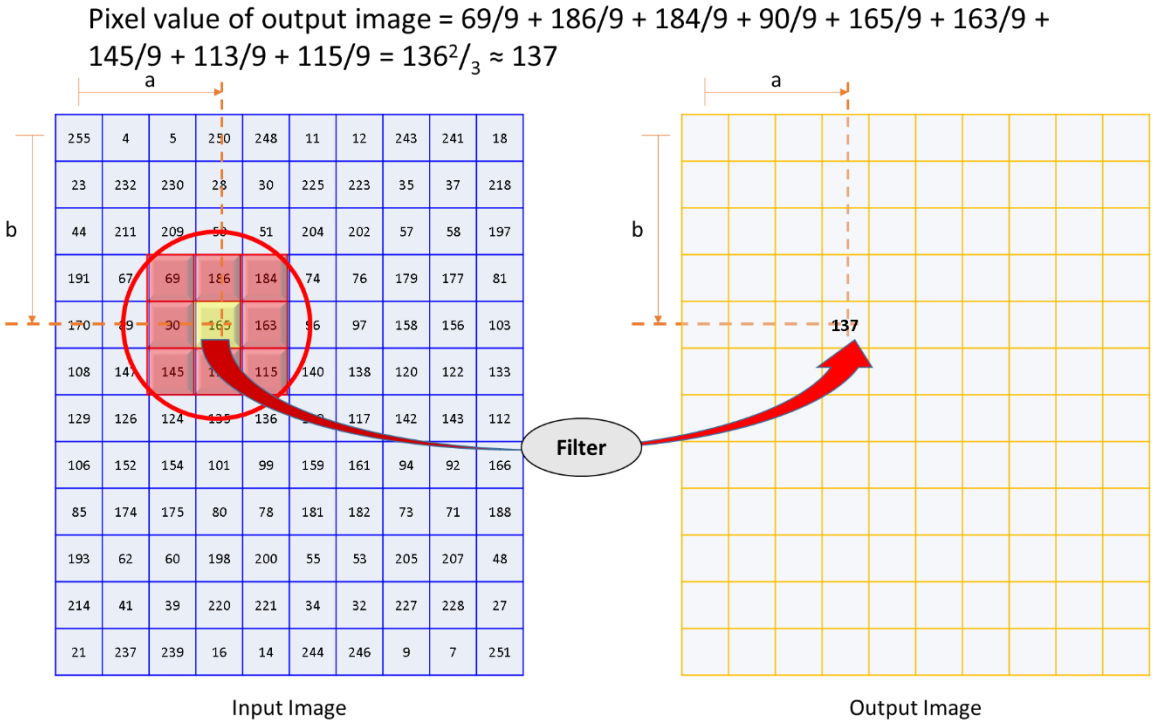

Figure 10: Filter result being transferred to output image

4. The process is repeated for each pixel until the entire image is formed. At edges and corners when the filter matrix exceeds the image boundaries non overlapping elements are considered to have a value of zero in the method of zero padding (there are also other methods too to deal with this scenario.)

0	0	0	0	0	0	0	0	0	0	0
0	255	4	5	250	248	11	12	243	241	18
0	23	232	230	28	30	225	223	35	37	218
0	44	211	209	50	51	204	202	57	58	197
0	191	67	69	186	184	74	76	179	177	81
0	170	89	90	165	163	96	97	158	156	103
0	108	147	145	113	115	140	138	120	122	133
0	129	126	124	135	136	119	117	142	143	112
0	106	152	154	101	99	159	161	94	92	166
0	85	174	175	80	78	181	182	73	71	188
0	193	62	60	198	200	55	53	205	207	48
0	214	41	39	220	221	34	32	227	228	27
0	21	237	239	16	14	244	246	9	7	251

Figure 11: When filter exceeds boundaries virtual outer pixels are considered to be zero

a. The first step in preforming the average filter is to make the kernel. As all elements should be the same use the **'ones'** command to create a 21x21 matrix of ones and store it in k. **(Hint: ones(2,3) makes a matrix of ones with 2 rows and 3 columns)**
b. Now make the sum of all elements in k to be equal to unity (the value 1) keeping all its elements same as each other. **(Hint: k=k/sum of all elements in k)**
c. Now apply the filter to the image in the figure 12 given below, by using the command **'imfilter'** and store the result in the variable **'blur'**. **(Hint: B=imfilter(A, k) applies the filter with kernel 'k' on to the image stored in 'A' and stores the result in 'B'.**

Figure 12: Figure to be blurred

d. Output the result on ‘blur’ on to a figure.

4. Image Histogram

Image histogram for a grayscale image or a layer of color image is a bar graph of number of pixels containing a particular intensity vs the intensity level. Since we are using 8 bit depth it will be a bar graph showing the number of pixels containing each intensity level from 0 to 255. These plots can be used to segment images in to objects or segment objects from the background. Below is a grayscale image and its histogram. The light background has the more pixels close to a higher intensity level of 230 and the dark object on it has many pixels with lower intensity close to 100, thus giving the shape of the histogram. Thus a mid-value around 160 can be used to septate the object from the background, in this case any pixel less than 160 can be considered the object.

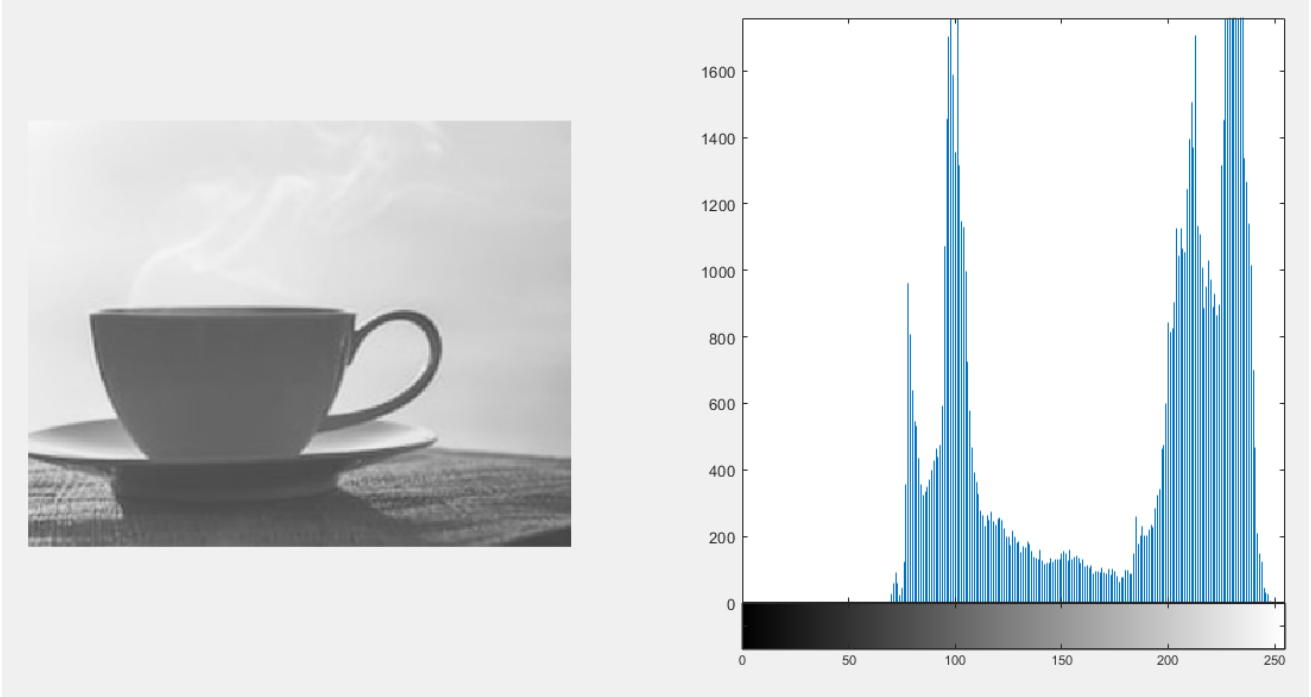

Figure 13: Grayscale image and its histogram

a. Use **'imhist'** to obtain the histogram of the grayscale image of figure 3 obtained in question 1 part e. **(Hint: imhist(A) gives the histogram of the image stored in A)**
b. Observe the histogram and suggest a suitable intensity level (threshold) to separate the dark background from the toys. (**Hint:** The background is dark so the first heap in the histogram should denote the background)
c. The above intensity level can also be calculated statistically using the command **'graythresh'**. Use this command to obtain the threshold. **(T= graythresh (BW) stores a suitable threshold**

to separate the image into 2 segments as a fraction of the maximum intensity (255) in T, thus actually the required value is T=255* graythresh (BW))

d. Now obtain a binary image containing the toys by using the above threshold and display it on a figure. **(Hint: B=BW>T, where BW, T and B are the grayscale image, threshold and the Binary image respectively)**

e. You will see lots of white spots (noise) in addition to the toys, to prevent this blur the grayscale image with a filter of size 25x25 and then perform steps c. and d. on it.

f. Now use the command **'[~,n]=bwlabel(B)'** to obtain the number of toy pieces in the variable n, where B is the binary image obtained from the blurred image in part e. (The command counts the number of non-connected objects.

g. Now write a script to count the number of red pieces. **(Hint: use the specific color layer instead of the grayscale image)**

Table 2: List of angles for question 2 part e

Index No.	Angles in degrees			
2021/E/001	**-1455**	**1184**	**1237**	**-828**
2021/E/002	**-1079**	**-1027**	**920**	**674**
2021/E/003	**-377**	**830**	**1645**	**-1153**
2021/E/004	**-1465**	**1869**	**-165**	**628**
2021/E/005	**684**	**-271**	**-567**	**1025**
2021/E/006	**1801**	**-1674**	**1271**	**-1565**
2021/E/007	**-392**	**1520**	**-1222**	**2107**
2021/E/008	**1431**	**-608**	**-1230**	**1334**
2021/E/009	**719**	**1547**	**-498**	**-465**
2021/E/010	**-1421**	**1901**	**908**	**-1156**
2021/E/011	**-1832**	**840**	**-932**	**2078**
2021/E/012	**-1841**	**-228**	**512**	**2085**
2021/E/013	**733**	**426**	**-1629**	**-124**
2021/E/014	**-1762**	**441**	**-580**	**616**
2021/E/015	**693**	**-970**	**2000**	**-1136**
2021/E/016	**-690**	**-276**	**1257**	**995**
2021/E/017	**-1774**	**-605**	**1995**	**2037**
2021/E/018	**-1431**	**1204**	**939**	**-1526**
2021/E/019	**1469**	**-239**	**-911**	**1718**
2021/E/020	**-681**	**1198**	**899**	**-1147**
2021/E/021	**-1766**	**1567**	**-1615**	**999**
2021/E/022	**-1785**	**-615**	**1226**	**1702**
2021/E/023	**-330**	**1540**	**1267**	**-825**
2021/E/024	**-754**	**838**	**-916**	**1687**
2021/E/025	**760**	**-610**	**-189**	**1380**
2021/E/026	**1044**	**779**	**-888**	**-778**
2021/E/027	**1762**	**-1675**	**501**	**-456**
2021/E/028	**-1045**	**1898**	**1620**	**-1160**
2021/E/029	**-1780**	**1516**	**902**	**-805**
2021/E/030	**-370**	**835**	**-879**	**1344**
2021/E/031	**680**	**1146**	**-150**	**-1213**

2021/E/032	**-1071**	**1850**	**501**	**-1561**
2021/E/033	**-1835**	**1172**	**-572**	**635**
2021/E/034	**-1079**	**1178**	**-1597**	**1673**
2021/E/035	**-1804**	**1200**	**543**	**-805**
2021/E/036	**-378**	**1161**	**930**	**-459**
2021/E/037	**-367**	**-664**	**540**	**648**
2021/E/038	**1048**	**-651**	**1231**	**-482**
2021/E/039	**322**	**1524**	**-1289**	**-1212**
2021/E/040	**-715**	**1505**	**-1596**	**2046**
2021/E/041	**368**	**-645**	**571**	**-455**
2021/E/042	**1791**	**1894**	**-1660**	**-1165**
2021/E/043	**-1109**	**798**	**-928**	**1666**
2021/E/044	**383**	**-263**	**-1238**	**1363**
2021/E/045	**-343**	**1508**	**-1664**	**2073**
2021/E/046	**1041**	**446**	**-173**	**-1513**
2021/E/047	**-739**	**458**	**941**	**-478**
2021/E/048	**-1113**	**1164**	**504**	**-800**
2021/E/049	**-707**	**1865**	**1650**	**-1128**
2021/E/050	**1051**	**1854**	**-922**	**-1498**
2021/E/051	**-1836**	**1514**	**-539**	**1311**
2021/E/052	**742**	**-1745**	**-1655**	**1690**
2021/E/053	**1778**	**413**	**-870**	**-442**
2021/E/054	**-331**	**1186**	**-1646**	**1334**
2021/E/055	**-1482**	**-1750**	**927**	**1706**
2021/E/056	**402**	**-1388**	**-172**	**2062**
2021/E/057	**-1773**	**443**	**1280**	**-1170**
2021/E/058	**-1841**	**1930**	**-1237**	**1355**
2021/E/059	**1839**	**-992**	**-1644**	**621**
2021/E/060	**-1480**	**433**	**1987**	**-87**
2021/E/061	**-722**	**1868**	**-912**	**1364**
2021/E/062	**1051**	**-1668**	**1972**	**-779**
2021/E/063	**-1095**	**819**	**576**	**-72**
2021/E/064	**1044**	**-273**	**496**	**-1133**

2021/E/065	**-733**	**1915**	**-1626**	**2064**
2021/E/066	**1799**	**-295**	**1596**	**-807**
2021/E/067	**-1806**	**1924**	**-190**	**961**
2021/E/068	**1038**	**-1700**	**505**	**-818**
2021/E/069	**1472**	**-1751**	**-163**	**1753**
2021/E/070	**1786**	**-997**	**1589**	**-1142**
2021/E/071	**1109**	**1859**	**-184**	**-1519**
2021/E/072	**-1088**	**-1031**	**922**	**1679**
2021/E/073	**1770**	**1862**	**-576**	**-466**
2021/E/074	**1082**	**794**	**-870**	**-1502**
2021/E/075	**-1777**	**-1667**	**1236**	**633**
2021/E/076	**-716**	**1864**	**1235**	**-1546**
2021/E/077	**1835**	**-283**	**-896**	**1696**
2021/E/078	**745**	**-1722**	**-158**	**1385**
2021/E/079	**-750**	**840**	**870**	**-1530**
2021/E/080	**-1784**	**1571**	**1974**	**-771**
2021/E/081	**-1774**	**1515**	**-1645**	**973**
2021/E/082	**1829**	**-1694**	**878**	**-812**
2021/E/083	**-1794**	**828**	**1642**	**-787**
2021/E/084	**-1399**	**413**	**-1260**	**992**
2021/E/085	**355**	**-599**	**-1287**	**973**
2021/E/086	**-1758**	**1572**	**1273**	**-1138**
2021/E/087	**-1406**	**423**	**-189**	**1754**
2021/E/088	**366**	**-1309**	**1282**	**-76**
2021/E/089	**755**	**795**	**-1644**	**-844**
2021/E/090	**-697**	**1539**	**925**	**-844**
2021/E/091	**-1071**	**1179**	**1975**	**-1211**
2021/E/092	**393**	**-1390**	**1235**	**-94**
2021/E/093	**1074**	**1501**	**-1238**	**-102**
2021/E/094	**1474**	**-1753**	**522**	**-1558**
2021/E/095	**363**	**-1333**	**-1286**	**1714**
2021/E/096	**-1454**	**850**	**-178**	**2110**
2021/E/097	**-695**	**1510**	**1596**	**-826**

2021/E/098	**1435**	**-273**	**-200**	**646**
2021/E/099	**-1832**	**1178**	**-578**	**2099**
2021/E/100	**679**	**1875**	**-858**	**-1182**
2021/E/101	**-713**	**-972**	**1938**	**632**
2021/E/102	**320**	**-313**	**-1244**	**639**
2021/E/103	**697**	**-1373**	**-215**	**662**
2021/E/104	**-1123**	**-949**	**942**	**1677**
2021/E/105	**-685**	**1910**	**532**	**-411**
2021/E/106	**1448**	**-296**	**1272**	**-1503**
2021/E/107	**397**	**810**	**-560**	**-796**
2021/E/108	**-737**	**-253**	**504**	**981**
2021/E/109	**-355**	**-957**	**1219**	**666**
2021/E/110	**367**	**819**	**-1645**	**-448**
2021/E/111	**323**	**-1709**	**1608**	**-69**
2021/E/112	**1475**	**-629**	**-1616**	**999**
2021/E/113	**-691**	**1861**	**867**	**-776**
2021/E/114	**-391**	**1875**	**-197**	**1385**
2021/E/115	**1758**	**1193**	**-522**	**-835**
2021/E/116	**1778**	**-952**	**-543**	**968**
2021/E/117	**-1825**	**452**	**-1266**	**1371**
2021/E/118	**1410**	**1126**	**-516**	**-1506**
2021/E/119	**1103**	**1564**	**-1658**	**-1198**
2021/E/120	**-1798**	**-1728**	**558**	**619**
2021/E/121	**683**	**-1731**	**858**	**-97**
2021/E/122	**361**	**-598**	**-910**	**1367**
2021/E/123	**-1046**	**-633**	**534**	**652**
2021/E/124	**-1765**	**1506**	**-1588**	**1369**
2021/E/125	**677**	**-979**	**516**	**-1168**
2021/E/126	**1841**	**414**	**-581**	**-776**
2021/E/127	**-1792**	**-255**	**1993**	**624**
2021/E/128	**1470**	**1560**	**-220**	**-446**
2021/E/129	**-1066**	**1889**	**-138**	**2092**
2021/E/130	**-402**	**1566**	**-1584**	**992**

2021/E/131	**372**	**436**	**-925**	**-425**
2021/E/132	**1066**	**431**	**-209**	**-1179**
2021/E/133	**1779**	**-265**	**-500**	**2089**
2021/E/134	**1104**	**808**	**-1604**	**-817**
2021/E/135	**1111**	**1882**	**-1623**	**-1501**
2021/E/136	**-347**	**1519**	**-1622**	**667**
2021/E/137	**1794**	**-1730**	**-497**	**1360**
2021/E/138	**1113**	**-1024**	**-868**	**666**
2021/E/139	**687**	**-957**	**1269**	**-1551**
2021/E/140	**-1055**	**1852**	**1960**	**-1541**
2021/E/141	**739**	**-608**	**1296**	**-489**
2021/E/142	**-398**	**1133**	**1942**	**-458**
2021/E/143	**-1481**	**1858**	**1582**	**-1509**
2021/E/144	**-1443**	**1138**	**-938**	**1369**
2021/E/145	**1479**	**-1000**	**1226**	**-833**
2021/E/146	**-360**	**-257**	**2002**	**2052**
2021/E/147	**1844**	**1922**	**-941**	**-806**
2021/E/148	**-734**	**1910**	**1937**	**-821**
2021/E/149	**-1756**	**1166**	**-907**	**964**
2021/E/150	**1088**	**-302**	**-220**	**646**
2021/E/151	**-367**	**1169**	**-1650**	**1005**
2021/E/152	**1470**	**-602**	**-889**	**2026**
2021/E/153	**391**	**-1009**	**1996**	**-1183**
2021/E/154	**328**	**-996**	**1967**	**-1172**
2021/E/155	**-1423**	**1887**	**519**	**-804**
2021/E/156	**1811**	**-233**	**-885**	**620**
2021/E/157	**-1451**	**478**	**1597**	**-1136**
2021/E/158	**1102**	**-228**	**1250**	**-471**
2021/E/159	**-1456**	**-1690**	**541**	**2096**
2021/E/160	**1801**	**-980**	**1964**	**-488**
2021/E/161	**1791**	**-1333**	**-939**	**652**
2021/E/162	**-339**	**1902**	**1222**	**-844**
2021/E/163	**-1440**	**1852**	**-1225**	**971**

2021/E/164	**1774**	**1165**	**-942**	**-777**
2021/E/165	**-1424**	**-1721**	**1222**	**986**
2021/E/166	**1419**	**833**	**-136**	**-823**
2021/E/167	**1452**	**-663**	**-1264**	**1004**
2021/E/168	**-1820**	**766**	**-911**	**1344**
2021/E/169	**1439**	**-290**	**2024**	**-838**
2021/E/170	**1418**	**-263**	**937**	**-415**
2021/E/171	**754**	**-1753**	**1291**	**-469**
2021/E/172	**1122**	**-1315**	**-928**	**586**
2021/E/173	**737**	**-1310**	**-1227**	**1676**
2021/E/174	**1787**	**-1750**	**559**	**-406**
2021/E/175	**-400**	**1540**	**-875**	**1711**
2021/E/176	**40**	**120**	**160**	**280**
2021/E/177	**-925**	**646**	**436**	**1435**
2021/E/178	**-209**	**2099**	**431**	**-1832**
2021/E/179	**-500**	**-1182**	**-265**	**679**
2021/E/180	**-1604**	**632**	**808**	**-713**
2021/E/181	**-1623**	**639**	**1882**	**320**
2021/E/182	**-1622**	**662**	**1519**	**697**
2021/E/183	**-497**	**1677**	**-1730**	**-1123**
2021/E/184	**-868**	**-411**	**-1024**	**-685**
2021/E/185	**1269**	**-1503**	**-957**	**1448**
2021/E/186	**1960**	**-796**	**1852**	**397**
2021/E/187	**1296**	**981**	**-608**	**-737**
2021/E/188	**1942**	**666**	**1133**	**-355**
2021/E/189	**1582**	**-448**	**1858**	**367**
2021/E/190	**-938**	**-69**	**1138**	**323**
2021/E/191	**1226**	**999**	**-1000**	**1475**
2021/E/192	**2002**	**-776**	**-257**	**-691**
2021/E/193	**-941**	**1385**	**1922**	**-391**