EE569 Digital Image Processing

HOMEWORK #2

Issued: 01/29/2020 Due: 02/16/2020

Yao Fu
6786354176
yaof@usc.edu

EE 569 Homework #2
February 16, 2020

Problem 1: Edge Detection (50%)

(a) Sobel Edge Detector (10%)

(b) Canny Edge Detector (10%)

(c) Structured Edge (15%)

(d) Performance Evaluation (15%)

1.1 Abstract and Motivation

Edge detection including a great many mathematical approaches aims at discovering and identifying points in digital images where brightness of images changes sharply, which, technically speaking, means images' discontinuities. Those points representing discontinuities of images are typically organized into several curved line segments named edges. Individuals are supposed to attach great significance to edge detection because it plays a crucial role in image processing and computer vision. According to the paper[1], discontinuities in image brightness are generally summed up as: discontinuities in depth, discontinuities in orientation of the image surface, changes in material properties and variations in images' visual illumination.

In order to understand the basic mechanism of edge detection, which is a stepping stone to middle or high levels ranging from image analysis to machine vision, in this report, I am about to implement three edge detectors: the first one is the Sobel Edge Detector that I will use C++ to realize it, the second one is the Canny Edge Detector that I will use OpenCV in Visual Studio 2019 to operate it, and the last one is the Structured Edge method that I will learn from someone's source code. What's more, I will use MATLAB 2019b to do the performance evaluation of my resulting edge maps by three ways. Necessary resulting images, tables and figures will be presented later in my report.

1.2 Approach and Procedures

(a) Sobel Edge Detector

In this part, I will use C++ to implement the Sobel edge detector.

Step1: Convert RGB images to grayscale images by using the following formula

$$Y = 0.2989 \times R + 0.5870 \times G + 0.1140 \times B$$

Step2: Use my own convolution function with the Sobel filter written by C++ to acquire x-gradient and y-gradient separately and shown the results.

The Sobel filters are

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -2 & 0 & +1 \end{bmatrix},\ G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

Step3: Utilize the Euclidean distance formula to calculate the magnitude and then normalize the gradient magnitude map.

$$Pixel\_normalized = 255 \times \frac{Pixel - Minimum(Pixel)}{Maximum(Pixel) - Minimum(Pixel)}$$

Step4: Attain an edge map with the best visual performance after finishing the performance evaluation in part (d).

(b) Canny Edge Detector

The Canny edge detector, developed by John F. Canny in 1986, is widely used in the field of image processing or computer vision, which is capable of utilizing a multi-stage algorithm to detect a broad range of edges in images. Generally speaking, the process of Canny edge detection algorithm can be summed up to the following 5 steps:

Step 1: Use a Gaussian filter to denoise the image that we want to detect its edge.

Step 2: Find out the intensity gradients' magnitude and orientation of the image.

$$\text{Gradient: } \nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} {\partial f}/{\partial x} \\ {\partial f}/{\partial y} \end{bmatrix}$$

$$Magnitude: |\nabla f| = \sqrt{g_x^2 + g_y^2}$$

$$Orientation: \theta = \tan^{-1}\left[\frac{g_y}{g_x}\right]$$

Step 3: Apply non-maximum suppression to get rid of spurious response to edge detection.

Non-maximum suppression means suppressing all the non-maximum pixels in a certain patch of images in order to realize edge thinning, indicating that locations with the sharpest change of intensity values will be kept. After the step 2, we are able to have the gradient image for reference, by which we can compare the current pixel's edge strength with the edge strength of pixels in the positive or negative gradient directions. At that moment, if the edge strength belonging to the current pixel is the largest one in terms of other pixels in the certain patch with the same orientation, that current pixel will be preserved. Otherwise, it will be suppressed immediately.

Step 4: Apply double threshold to determine potential edges.

After carrying out the non-maximum suppression in step 3, the remaining edge pixels are likely to represent more real edges of an image. However, there are still some fake edge pixels, probably caused by noise, color variation and so on. In order to handle that situation, it is essential for us to remove edge pixels with weak gradient values and keep edge pixels with high ones.

Therefore, we have to apply double threshold values to distinguish potential edges. If the gradient value of an edge pixel is higher than the high threshold value, it will be marked as a strong edge pixel so that it will be preserved. If that gradient value is smaller than the high threshold value and larger than the low one, it will be marked as a weak edge pixel for further selection. If that pixel's value is smaller than the low threshold value, it will be suppressed immediately. In this way, we are in the position to thin images' edge further.

Step 5: Track edge by hysteresis: Finalize the detection of edges by suppressing all the edges that are weak and not connected to strong edges from the step 4.

I will use the OpenCV from [2] in Visual Studio 2019 to implement the Canny edge detector.

(c) Structured Edge

i. SE detection algorithm

With the help of the paper [3], I summarize the Structured Edge detection algorithm as follows:

Step 1: Training: In this step, our goal is to build a decision forest that is an ensemble of a number of independent trees f_t. First of all, define a mapping of the form:

$$M: Y \to Z$$

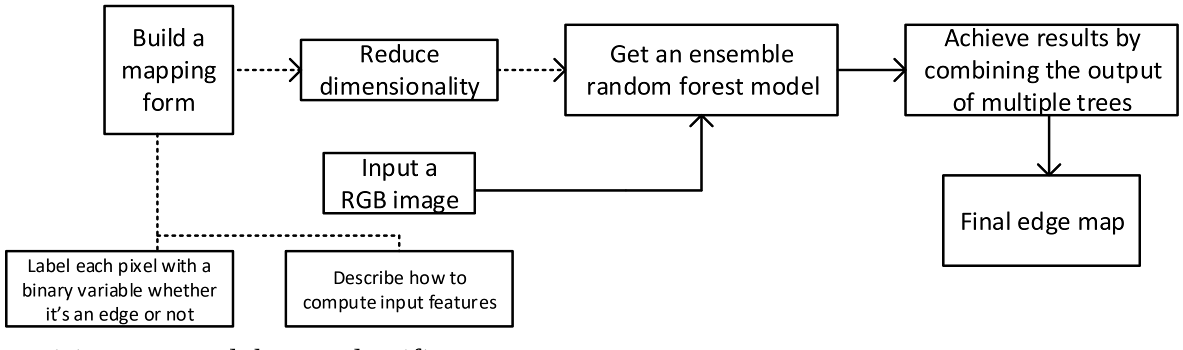
such that we can approximate the dissimilarity of $y \in Y$ by computing Euclidean distance in Z. Then, try to reduce dimensionality of Z by sampling m dimensions of it and further utilizing the Principal Component Analysis (PCA). Finally, use standard information gain criteria based on Shannon entropy or Gini impurity to build an ensemble model by combining a set of n labels $y_1 \dots y_n$ into a single prediction for both training, which means setting leaf labels, and testing, which means merging predictions.

Step 2: Apply our structured forests to edge detection. We regard an RGB or RGBD image as input. Then try to label each pixel with a binary variable indicating whether the pixel contains an edge or not. Next, we describe how can we compute input features x, the mapping functions $M_\varphi$used to determine splits, and the ensemble model used to combine multiple predictions.

Step 3: Make random forests achieve results by combining the output of multiple trees, meaning that we try to make multiple overlapping edge maps $y' \in Y'$ averaged to yield a soft edge response.

Step 4: Get the final edge map.

The flowchart is shown as follows:


Build a mapping form
Reduce dimensionality
Get an ensemble random forest model
Achieve results by combining the output of multiple trees
Input a RGB image
Final edge map
Label each pixel with a binary variable whether it's an edge or not
Describe how to compute input features

ii. Decision trees and the RF classifier

A decision tree is a structure similar to flowcharts, where each internal node represents a "test" on an attribute, each branch represents the outcome of the test, and each leaf node represents a decision represented by a class label. We have to make classification rules to denote paths from roots to leaves in order to build a decision tree.

Random forests are an ensemble learning model or a method for the tasks ranging from classification to regression operated by constructing a certain number of decision trees and outputting the class labels that is either the mode (classification) or mean prediction (regression) of several individual trees.

iii. Practice

I will use the MATLAB source code from [4] to implement the Structured Edge detector.

(d) Performance Evaluation

In this part, I will perform quantitative comparison between edge maps obtained by three edge detectors. It’s a fundamental reality that the essential goal of edge detection is to enable computers to generate contours accepted by humans. Hence, we need the edge map provided by individuals (called the ground truth) to evaluate the visual quality of the edge map from our algorithms.

However, different people may have distinct opinions about what exactly the edges are in a given image. In order take the opinion diversity into account, we usually take the mean of a certain performance measure for each ground truth ranging from the mean precision to the mean recall. To evaluate the visual quality of generated edge maps, we should identify the error. Every pixel in an edge map will be either of the following four classes:

i. ***True positive***: Edge pixels of edge maps match edge pixels of the relevant ground truth.
ii. ***True negative***: Non-edge pixels of edge maps match non-edge pixels in the ground
truth.
iii. ***False positive***: Edge pixels of edge maps correspond to the non-edge pixels in the ground truth, meaning that those are fake edge pixels our algorithms have wrongly identified.
iv. ***False negative***: Non-edge pixels in the edge map correspond to the true edge pixels in the ground truth, meaning that those are edge pixels our algorithms have missed.

The performance of an edge detection algorithm can be measured using the F measure, which is a function of the precision and the recall.

$$Precision: P = \frac{\#\ of\ True\ Positive}{\#\ of\ True\ Positive + \#\ of\ False\ Positive}$$

$$Recall: R = \frac{\#\ of\ True\ Positive}{\#\ of\ True\ Positive + \#\ of\ False\ Negative}$$

$$F = 2 \cdot \frac{P \cdot R}{P + R}$$

1.3 Experimental Results

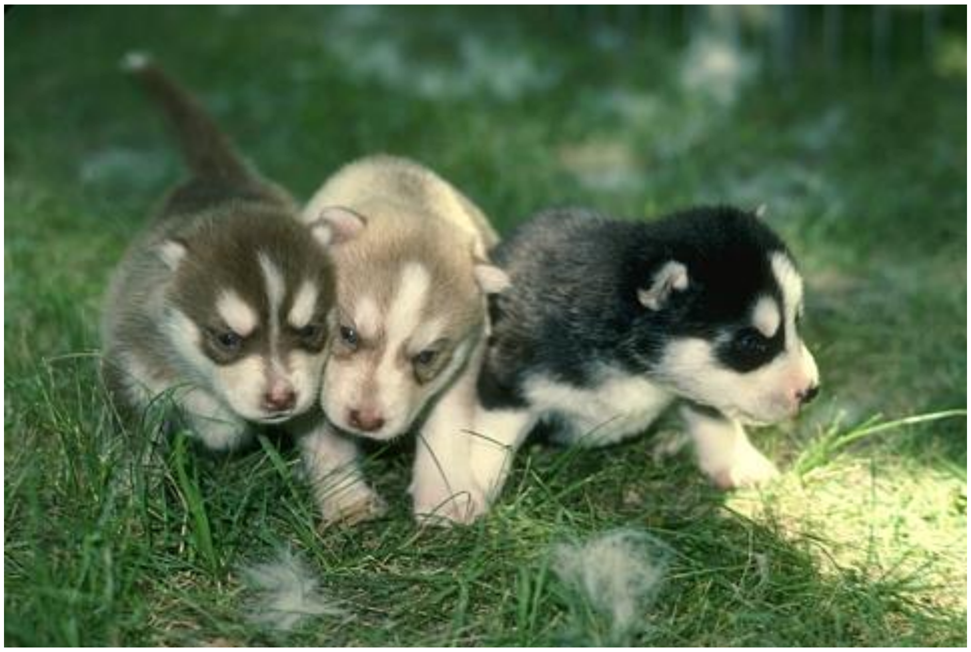


Figure. 1.1 The original “Dog” image

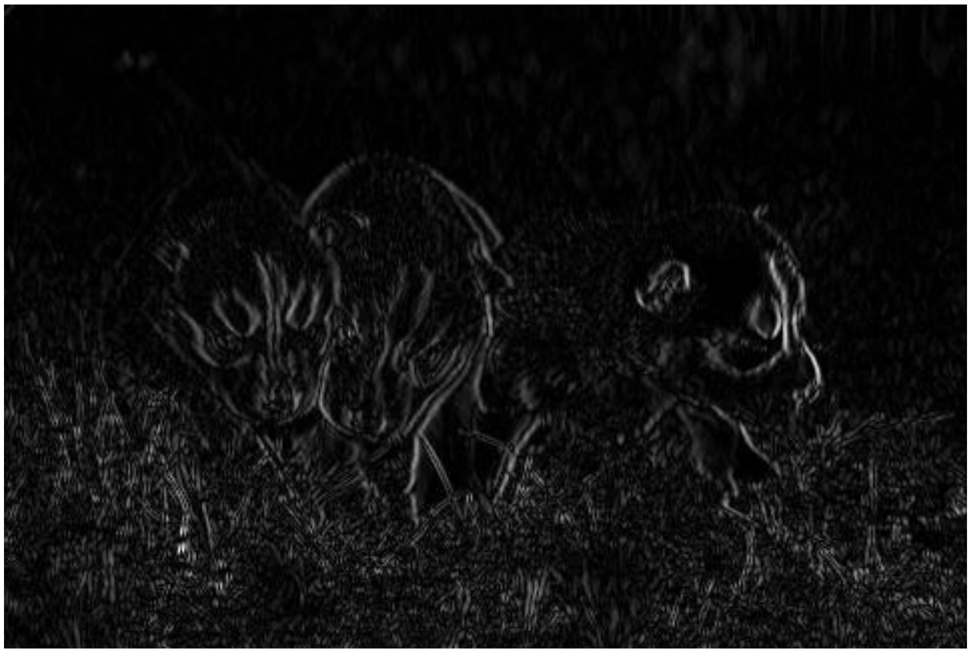


Figure. 1.2 The resulting “Dogs” x-gradient by using the Sobel filter

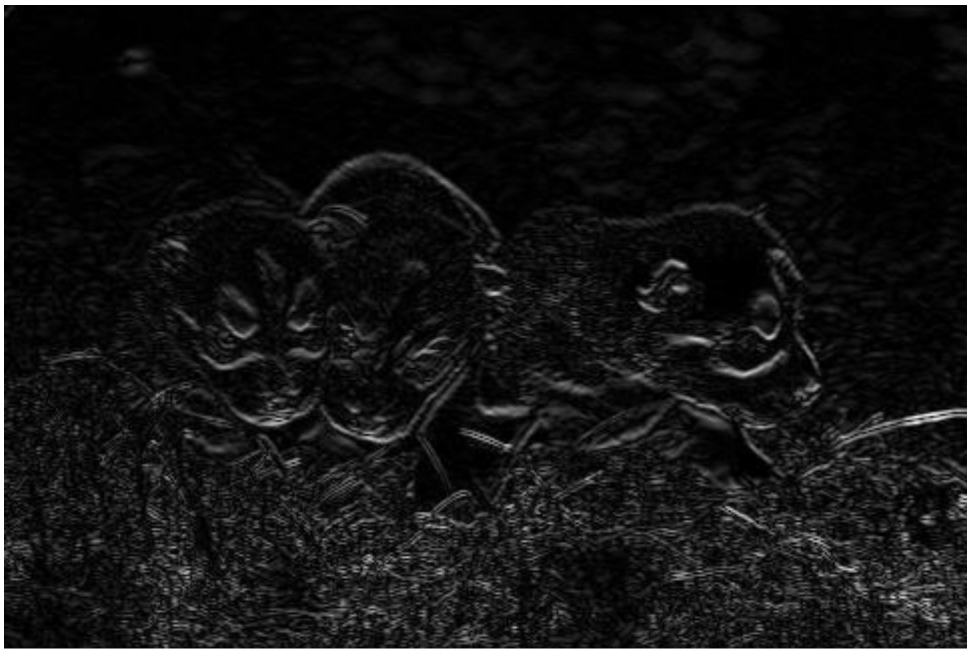


Figure. 1.3 The resulting “Dogs” y-gradient by using the Sobel filter

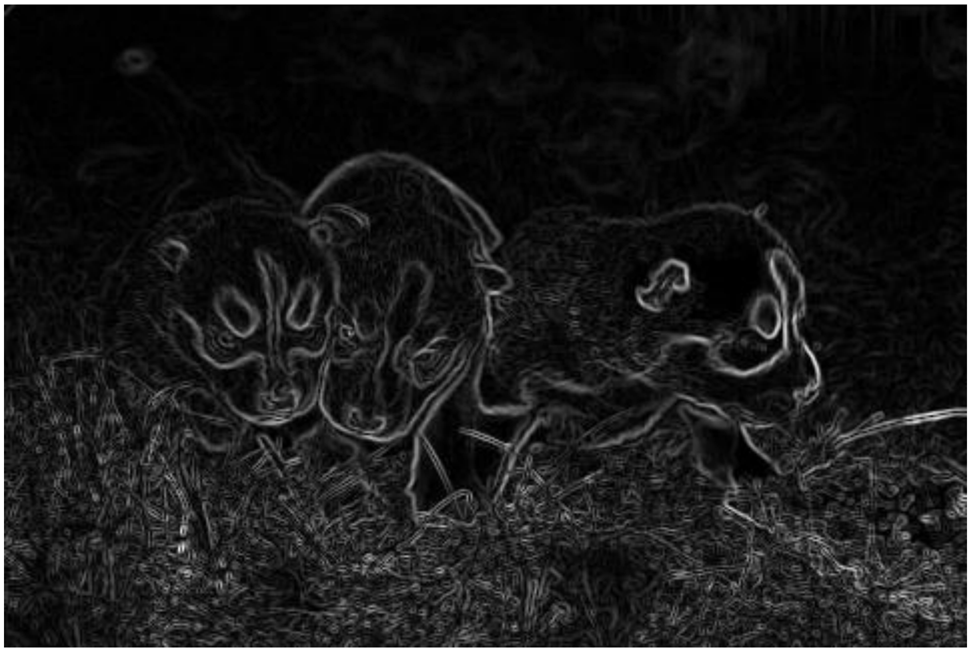


Figure. 1.4 The normalized gradient magnitude map of “Dogs” by using the Sobel filter

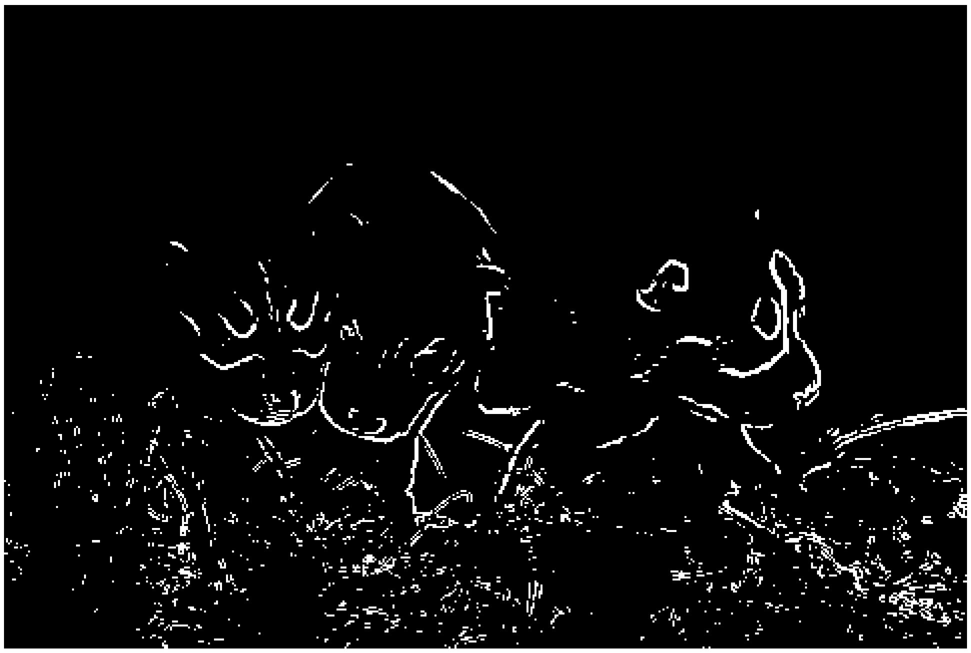


Figure. 1.5 The final edge map of “Dogs” with the threshold 0.34

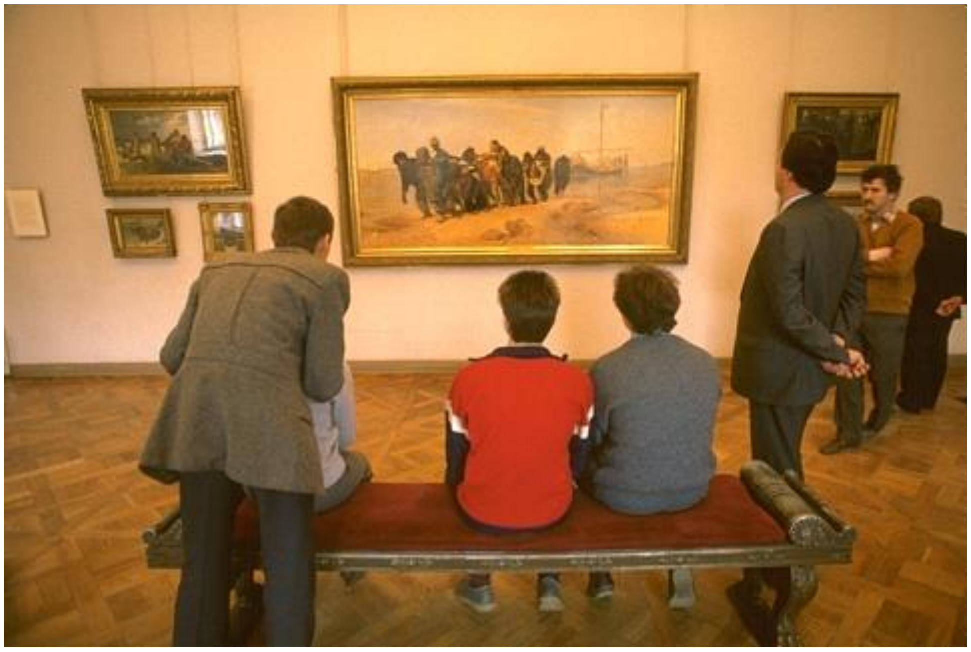


Figure. 1.6 The original “Gallery” image

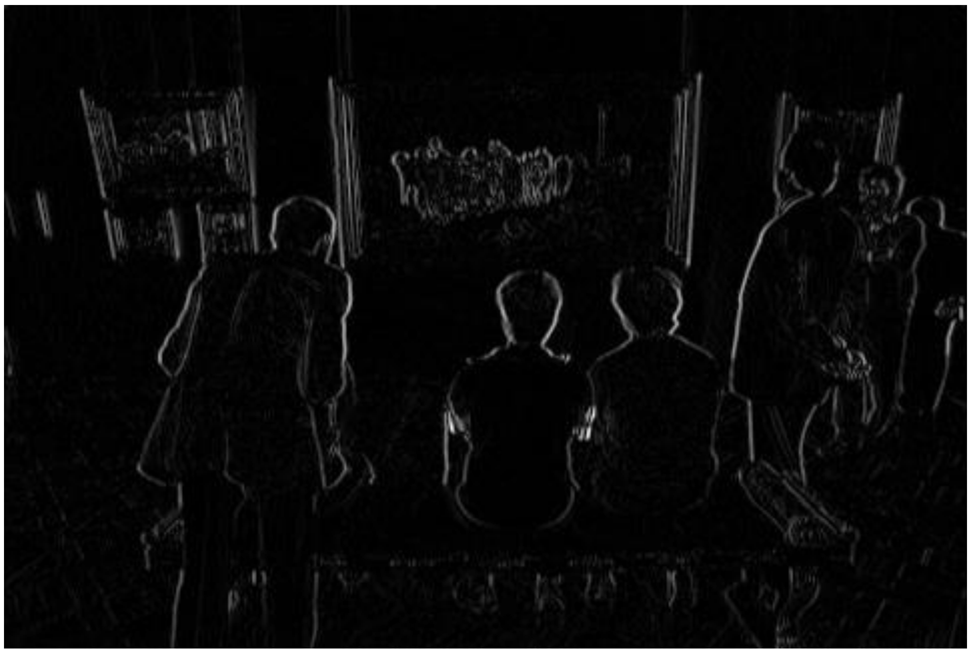


Figure. 1.7 The resulting “Gallery” x-gradient by using the Sobel filter

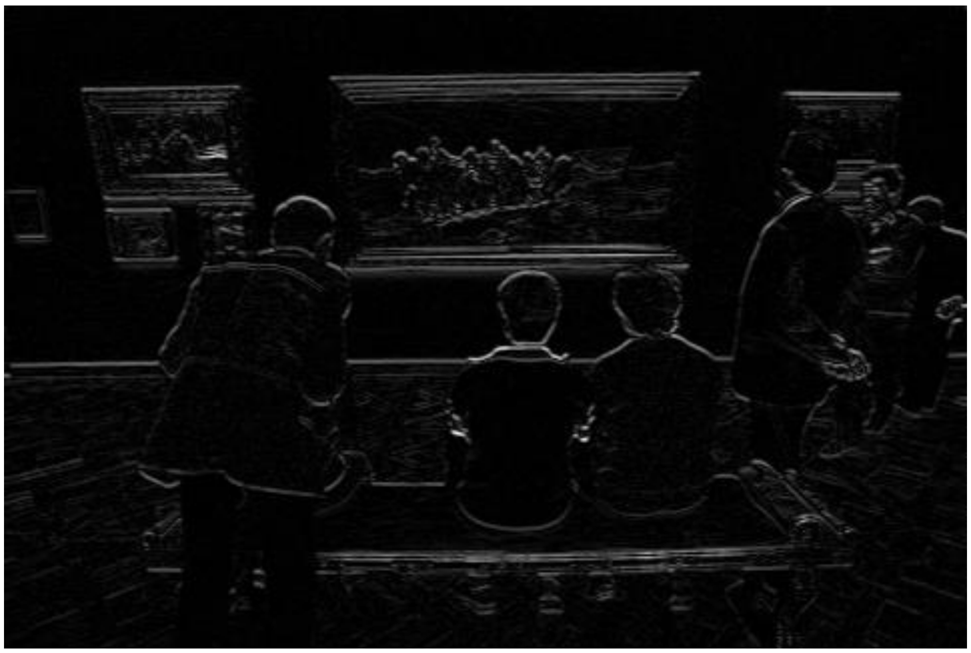


Figure. 1.8 The resulting “Gallery” y-gradient by using the Sobel filter

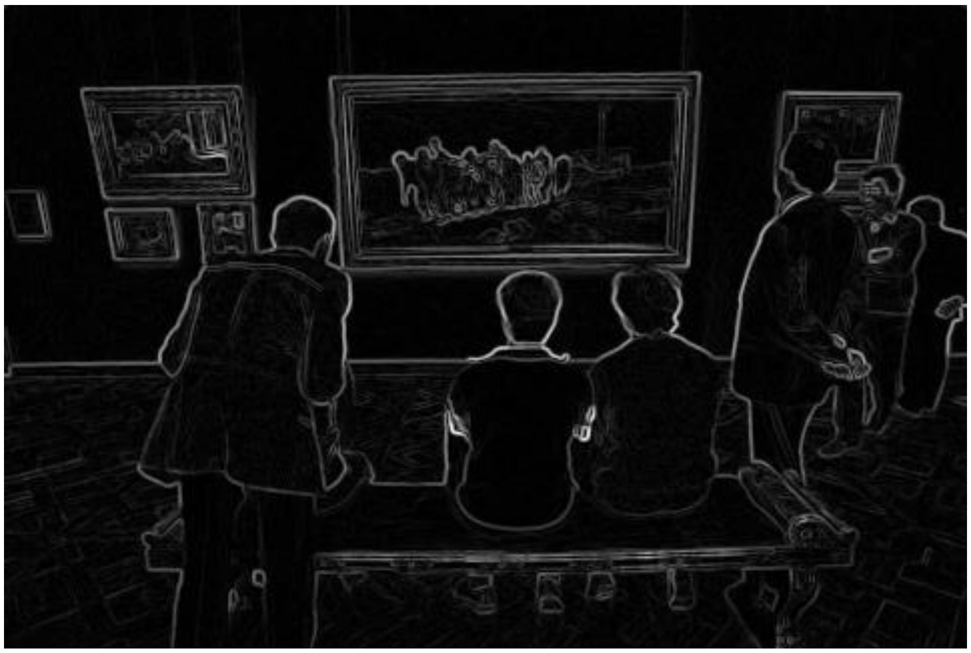


Figure. 1.9 The normalized gradient magnitude map of “Gallery” by using the Sobel filter

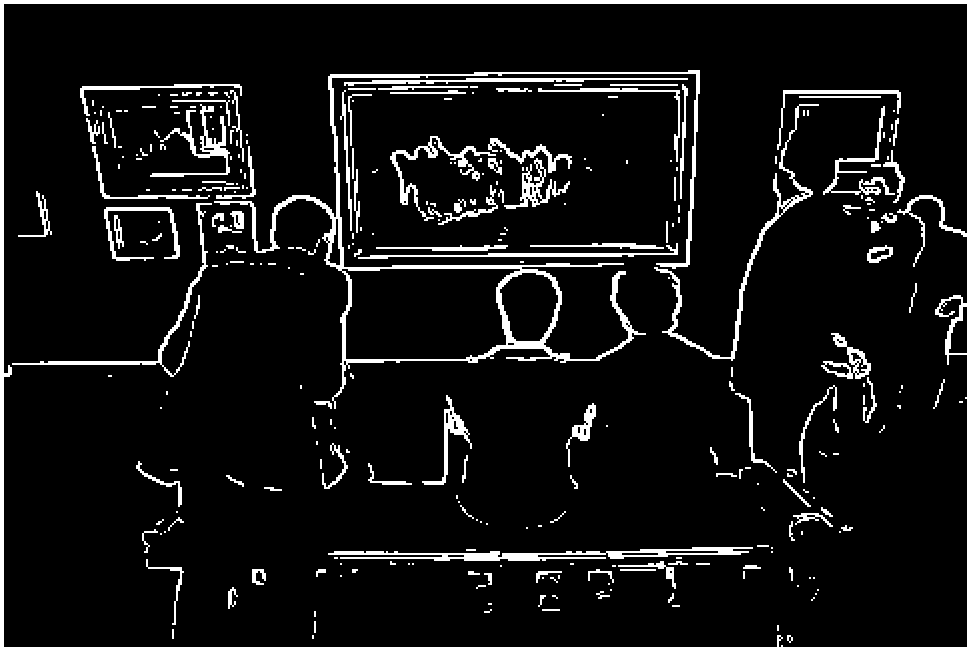


Figure. 1.10 The final edge map of “Gallery” with the threshold 0.22

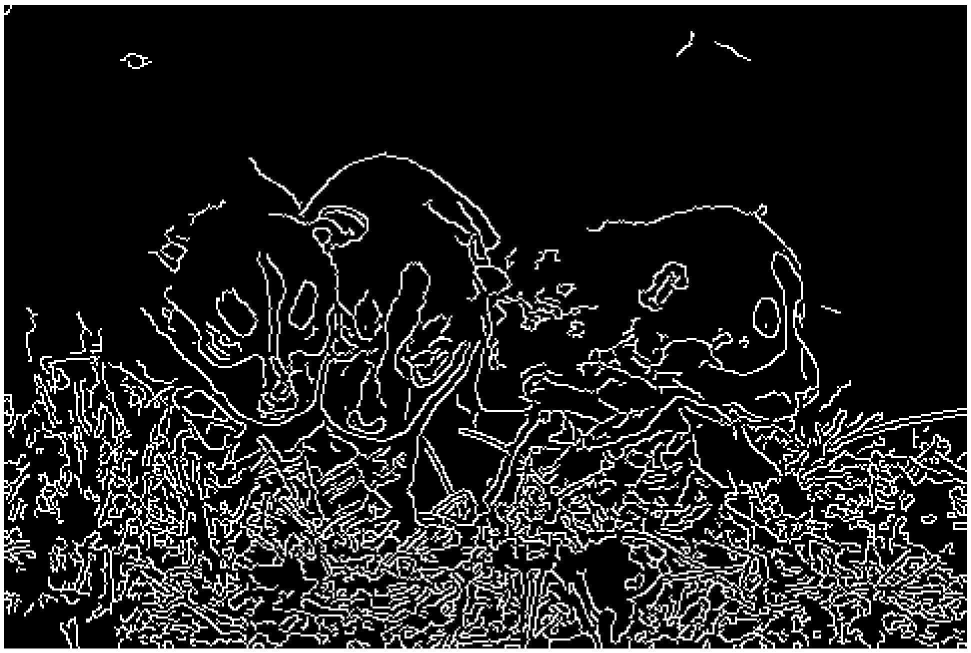


Figure. 1.11 The Canny edge map of “Dogs”
with the low threshold 50 (0.196) and the high threshold 100 (0.392)

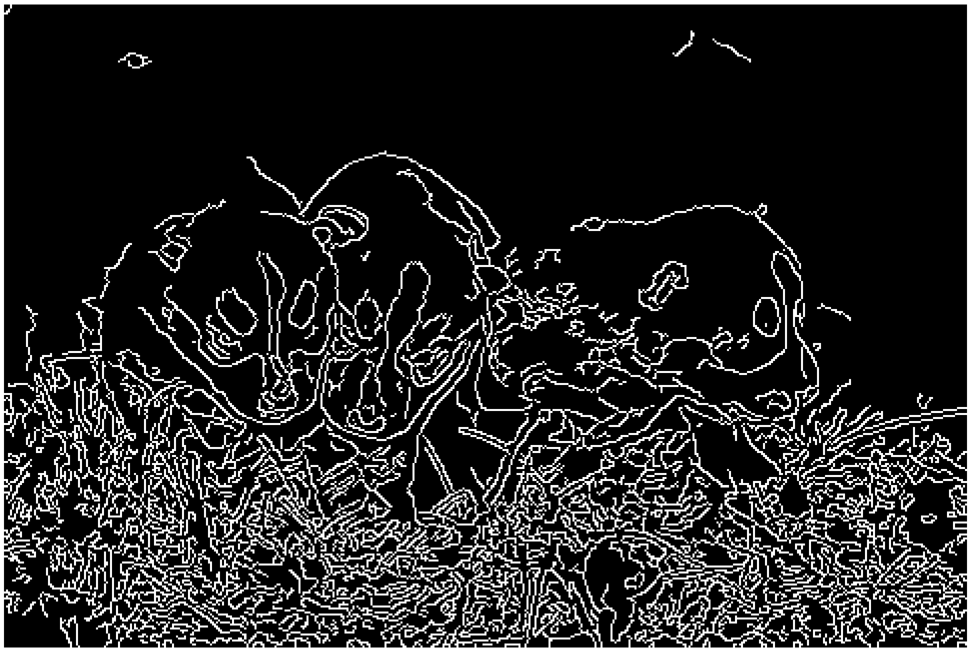


Figure. 1.12 The Canny edge map of “Dogs”
with the low threshold 55 (0.216) and the high threshold 110 (0.431)

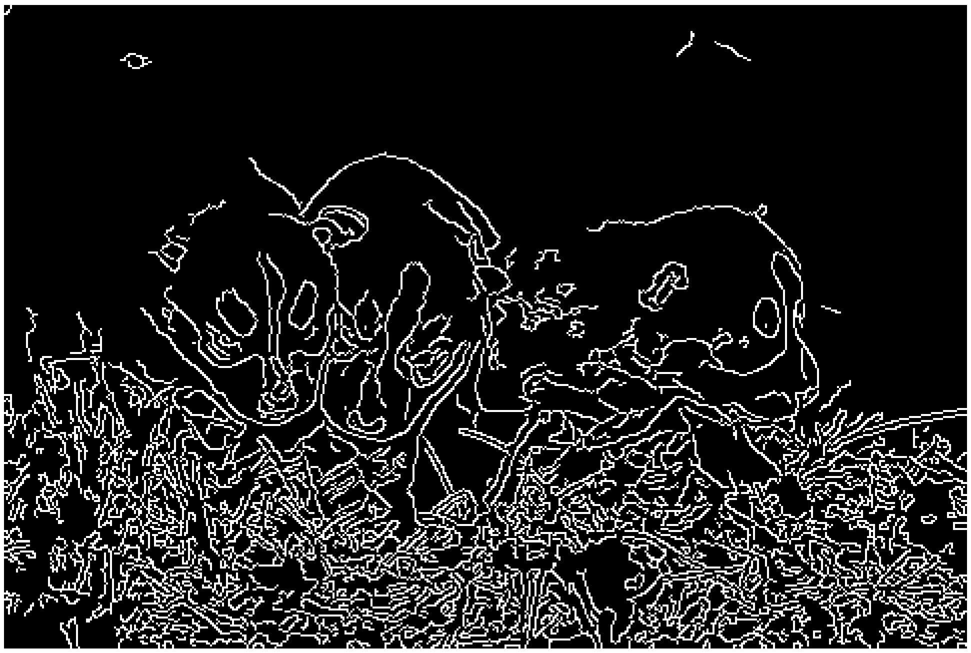


Figure. 1.13 The Canny edge map of “Dogs”
with the low threshold 60 (0.235) and the high threshold 120 (0.471)

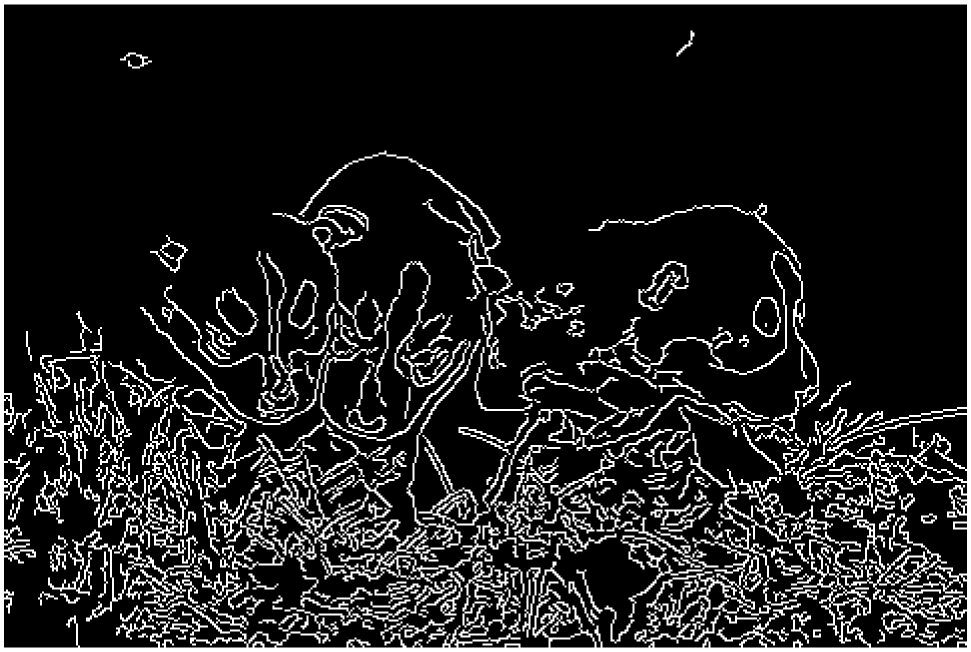


Figure. 1.14 The Canny edge map of “Dogs”
with the low threshold 65 (0.254) and the high threshold 130 (0.510)

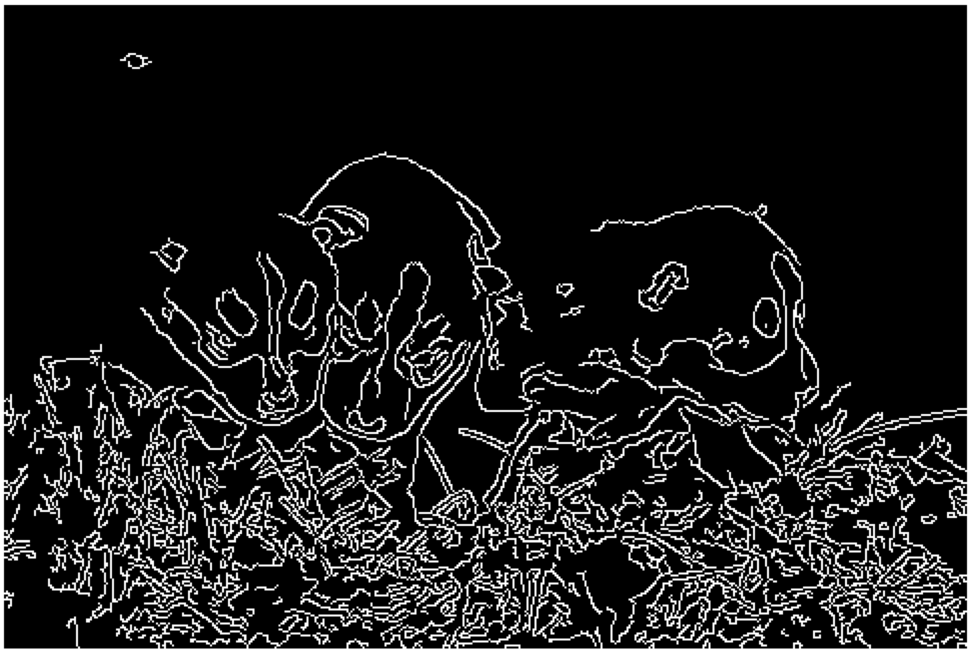


Figure. 1.15 The Canny edge map of “Dogs”
with the low threshold 70 (0.275) and the high threshold 140 (0.549)

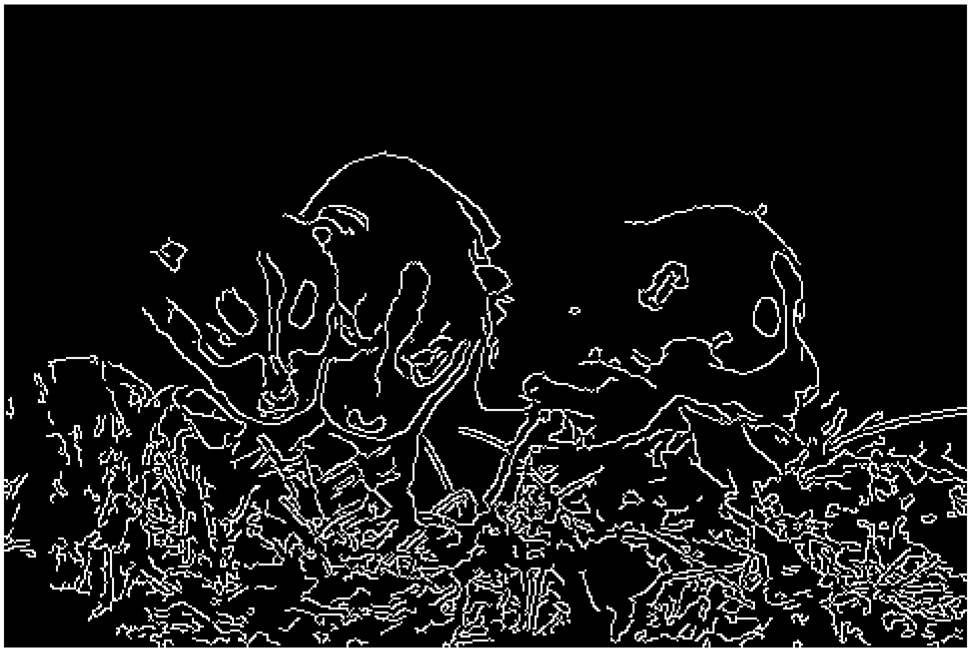


Figure. 1.16 The Canny edge map of “Dogs”
with the low threshold 80 (0.313) and the high threshold 160 (0.627)

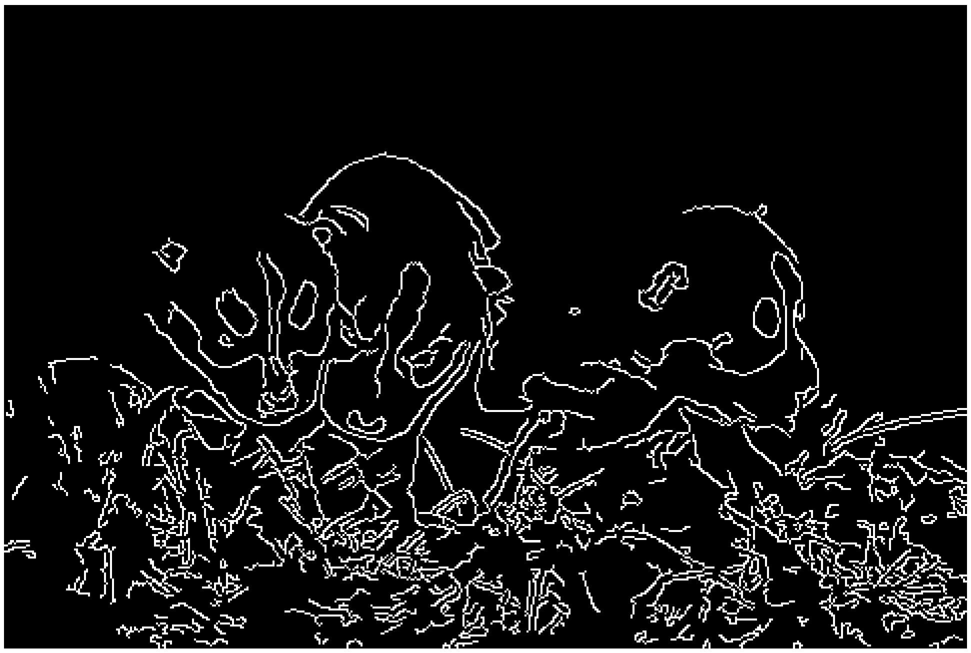


Figure. 1.17 The Canny edge map of “Dogs”
with the low threshold 90 (0.353) and the high threshold 180 (0.701)

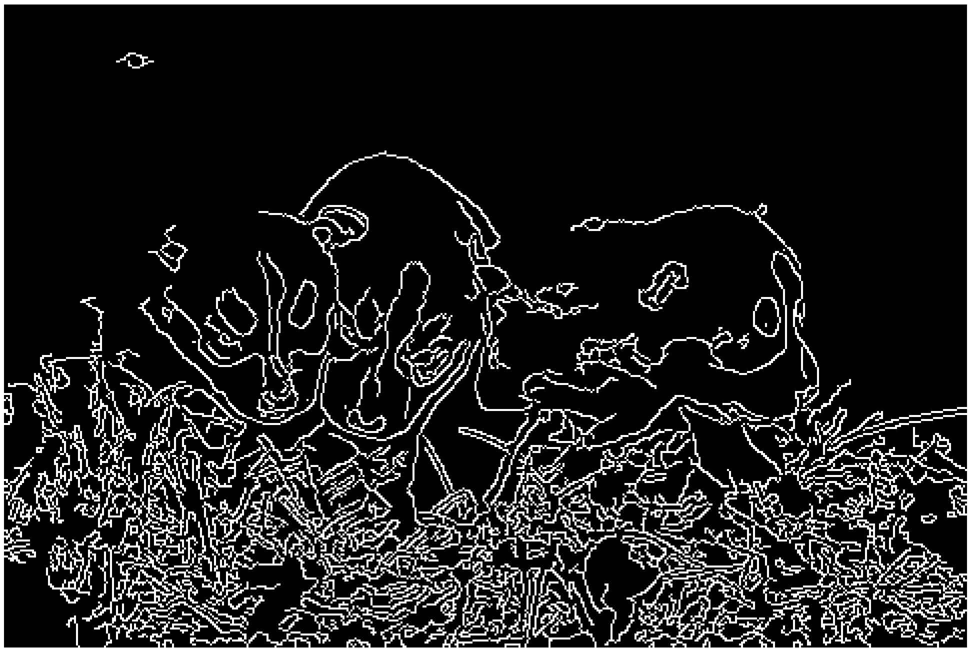


Figure. 1.18 The Canny edge map of “Dogs”
with the low threshold 50 (0.196) and the high threshold 150 (0.588)

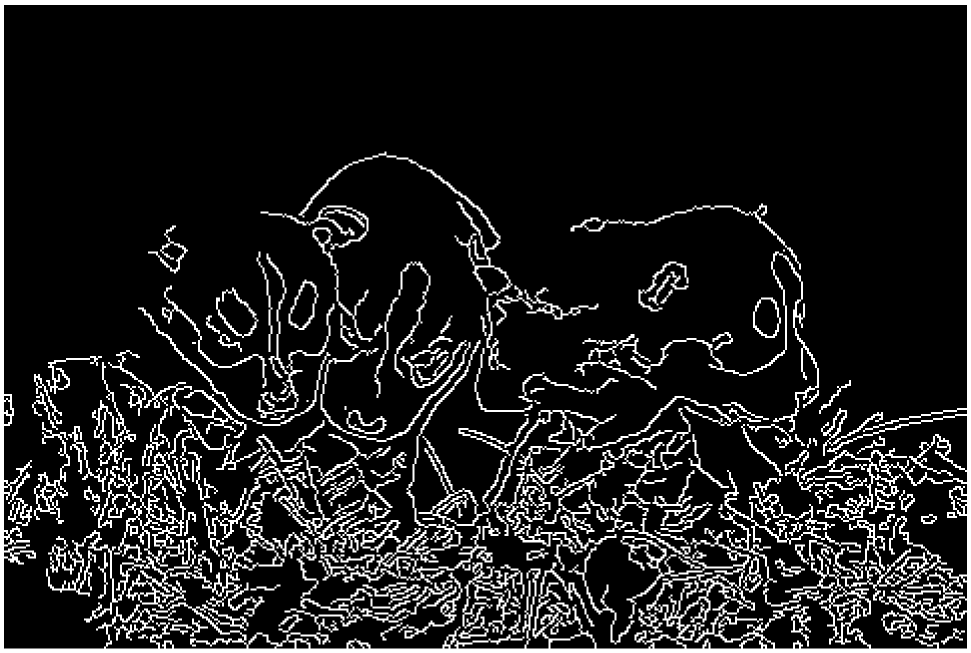


Figure. 1.19 The Canny edge map of “Dogs”
with the low threshold 55 (0.216) and the high threshold 165 (0.647)

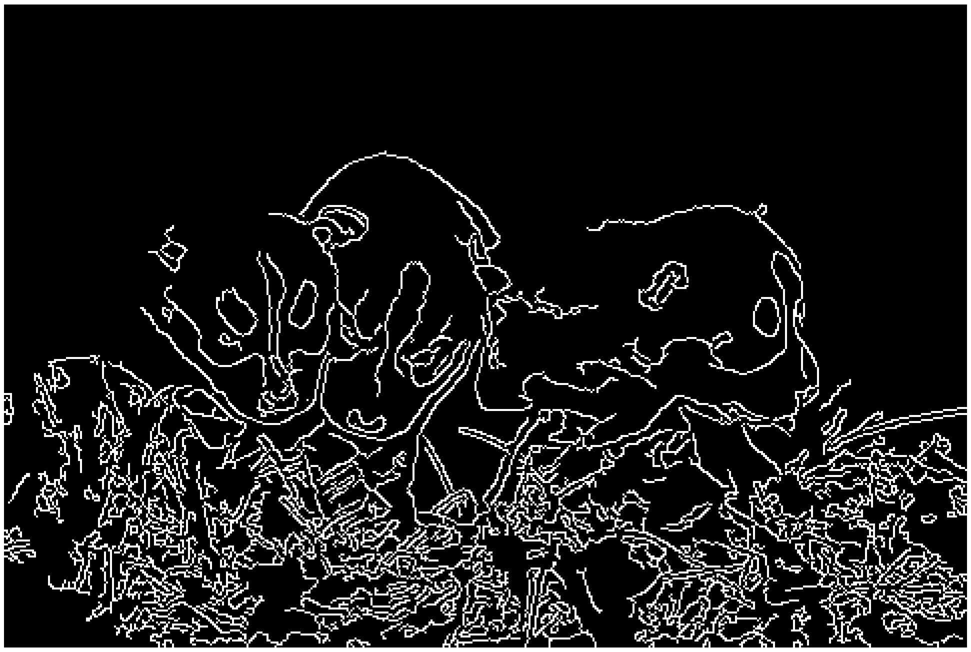


Figure. 1.20 The Canny edge map of “Dogs”
with the low threshold 60 (0.235) and the high threshold 180 (0.706)

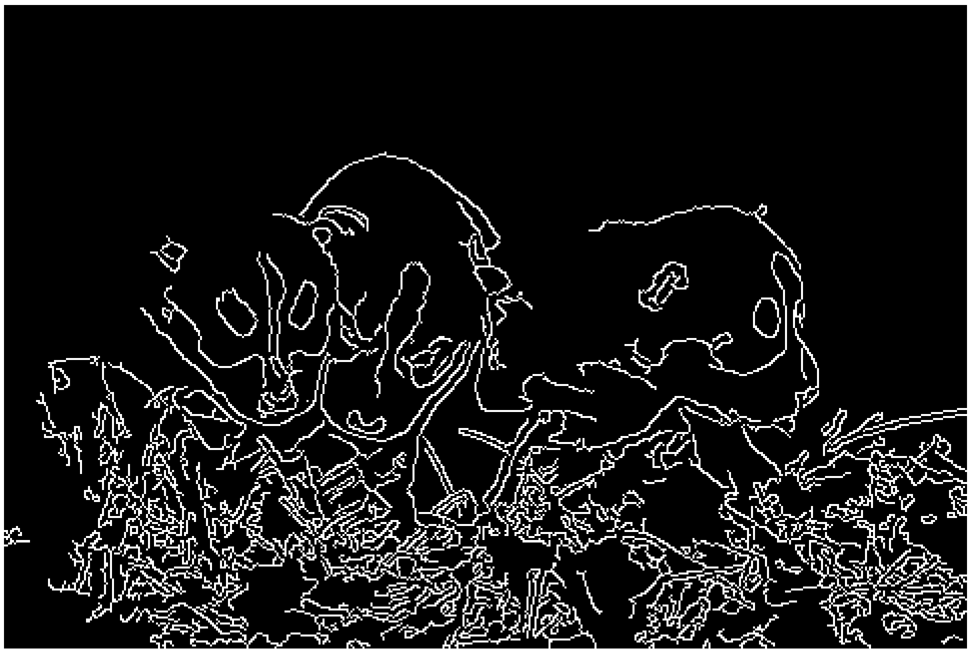


Figure. 1.21 The Canny edge map of “Dogs”
with the low threshold 65 (255) and the high threshold 195 (0.765)

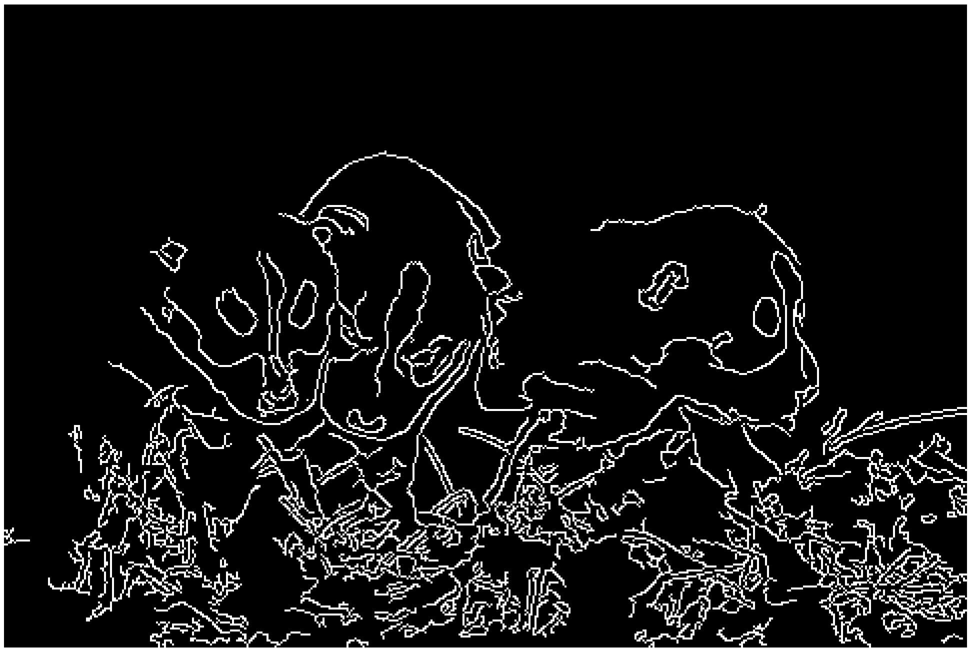


Figure. 1.22 The Canny edge map of “Dogs”
with the low threshold 70 (275) and the high threshold 210 (0.824)

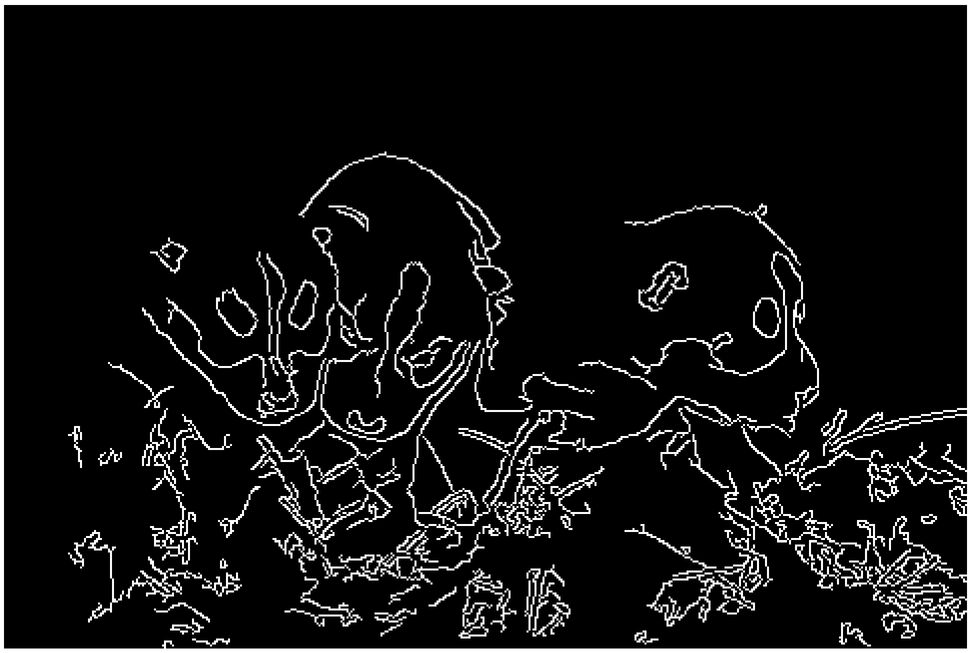


Figure. 1.23 The Canny edge map of “Dogs”
with the low threshold 80 (0.314) and the high threshold 240 (0.941)

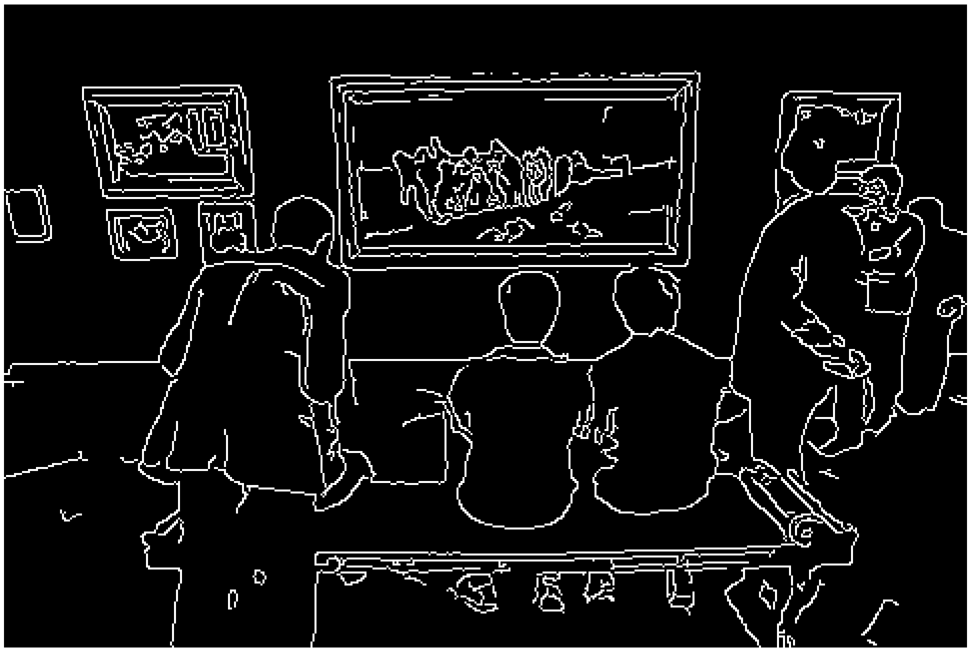


Figure. 1.24 The Canny edge map of “Gallery”
with the low threshold 50 (0.196) and the high threshold 100 (0.392)

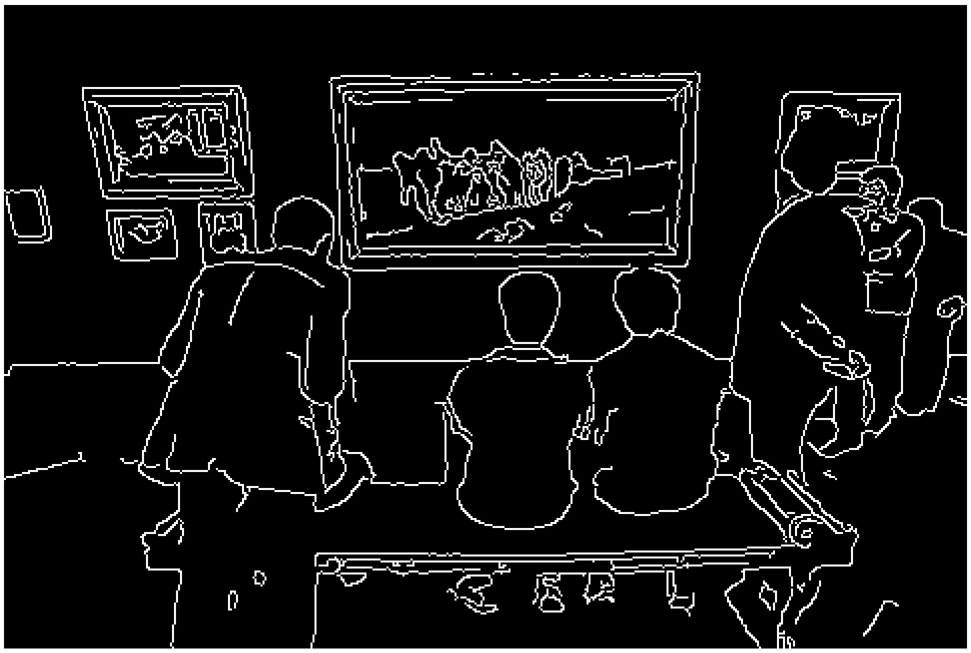


Figure. 1.25 The Canny edge map of “Gallery”
with the low threshold 55 (0.216) and the high threshold 110 (0.431)

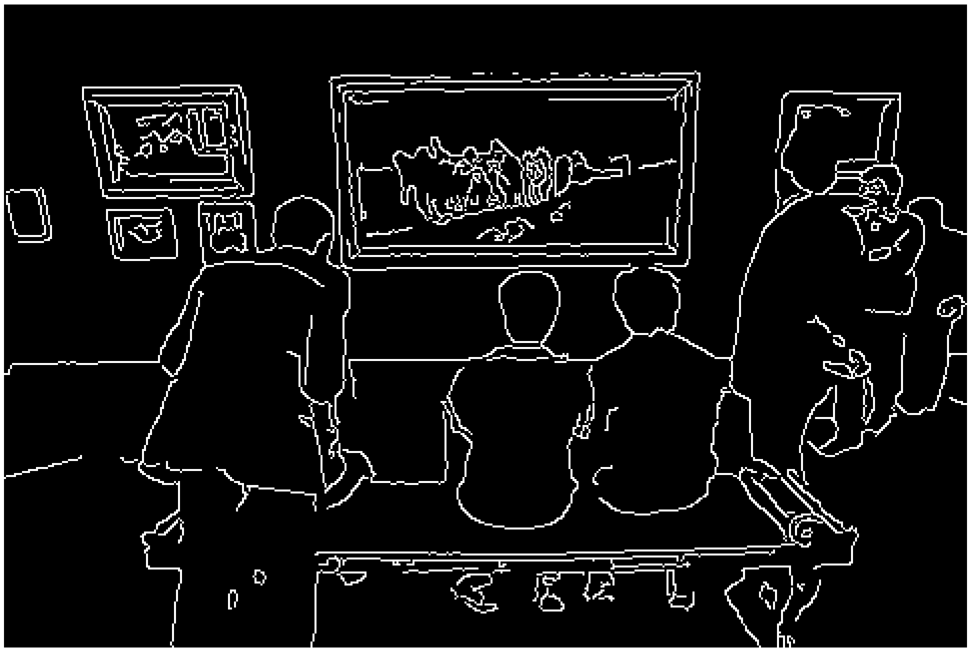


Figure. 1.26 The Canny edge map of “Gallery”
with the low threshold 60 (0.235) and the high threshold 120 (0.471)

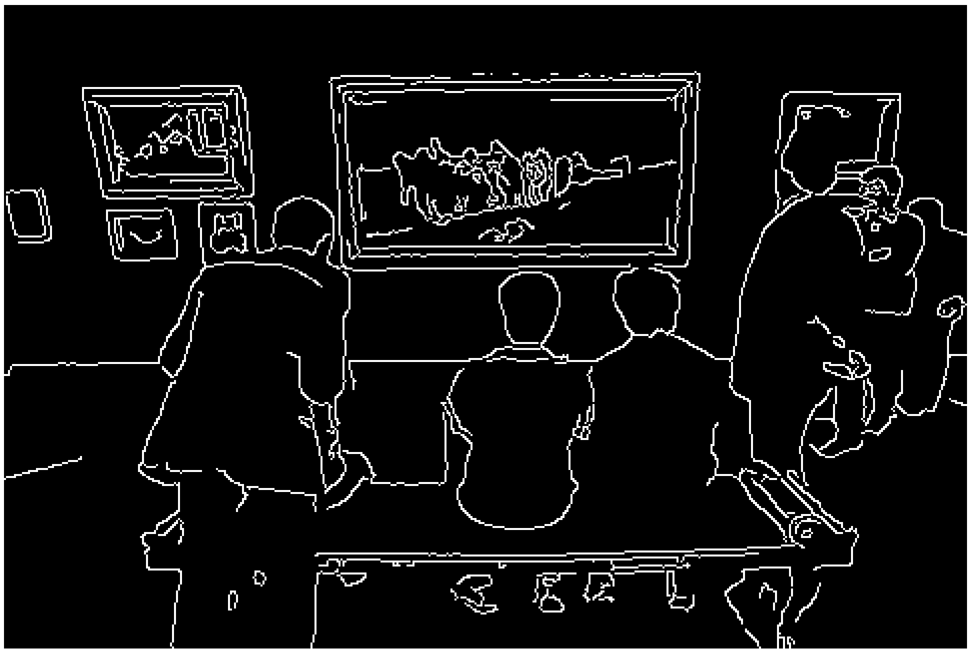


Figure. 1.27 The Canny edge map of “Gallery”
with the low threshold 65 (0.254) and the high threshold 130 (0.510)

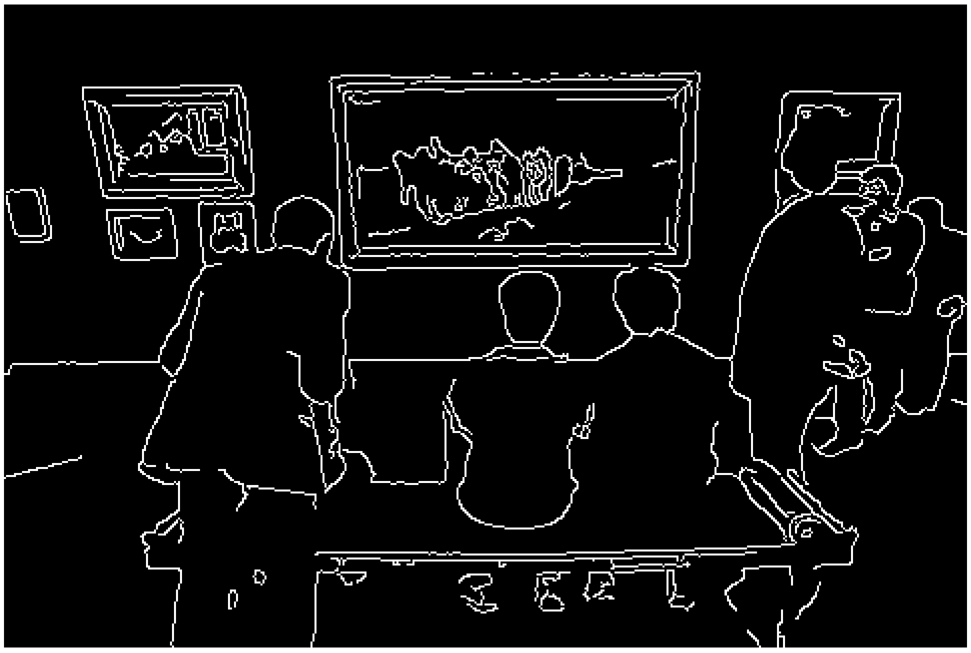


Figure. 1.28 The Canny edge map of “Gallery”
with the low threshold 70 (0.275) and the high threshold 140 (0.549)

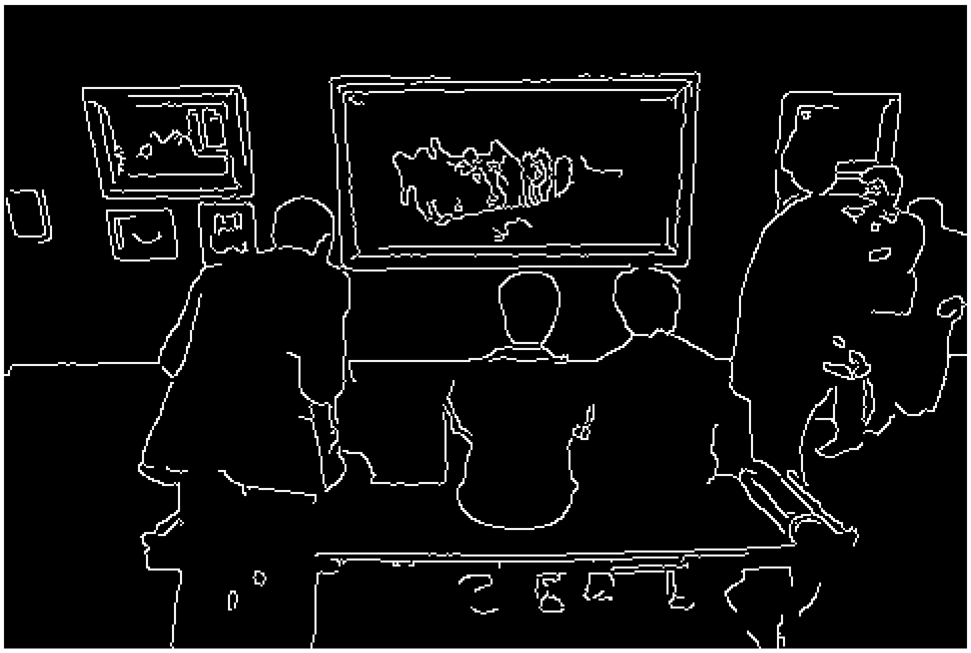


Figure. 1.29 The Canny edge map of “Gallery”
with the low threshold 80 (0.313) and the high threshold 160 (0.627)

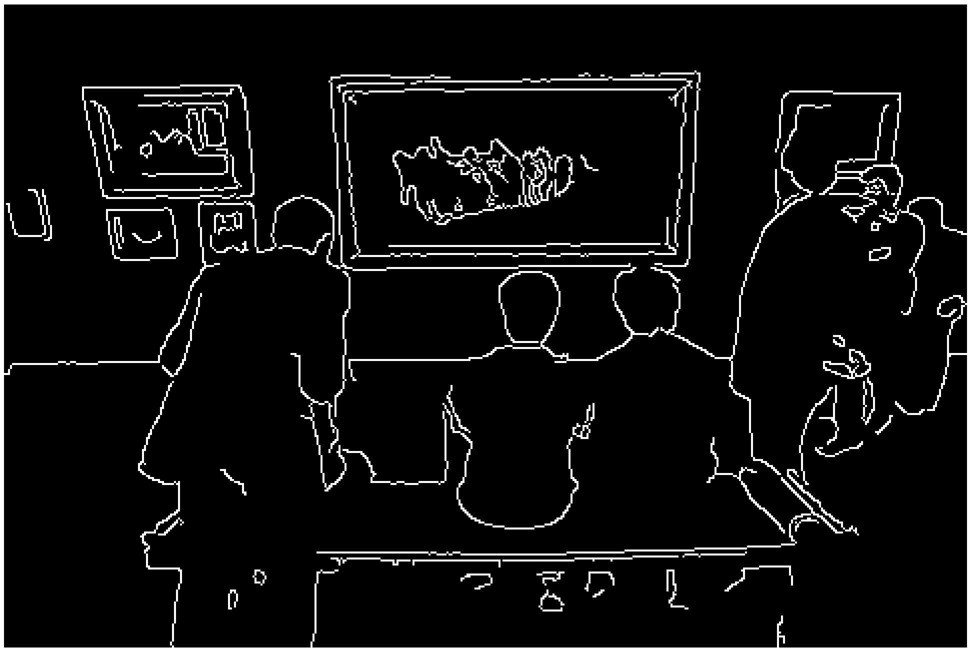


Figure. 1.30 The Canny edge map of “Gallery”
with the low threshold 90 (0.353) and the high threshold 180 (0.701)

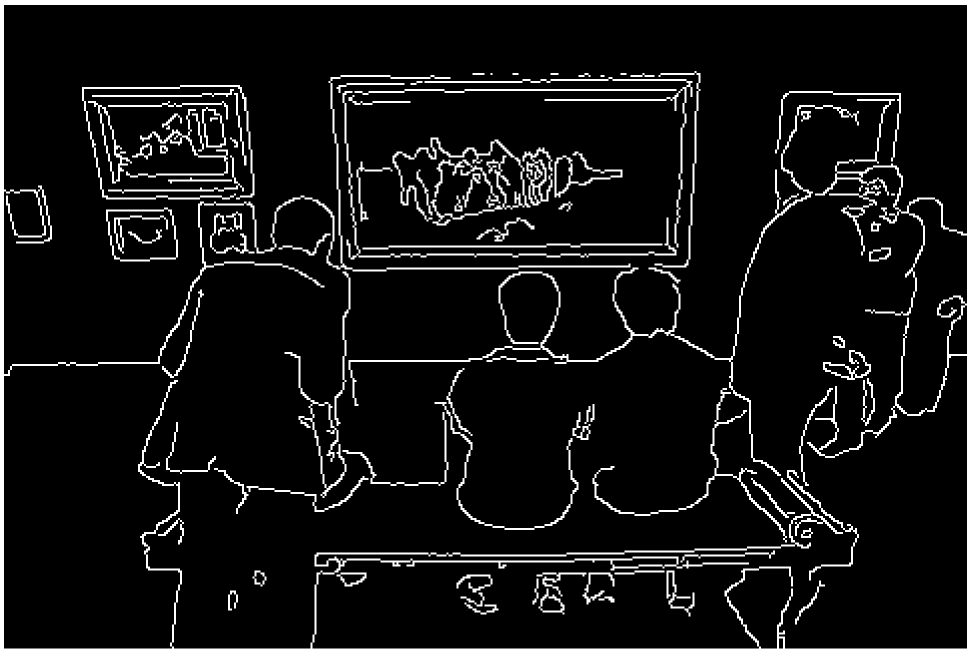


Figure. 1.31 The Canny edge map of “Gallery”
with the low threshold 50 (0.196) and the high threshold 150 (0.588)

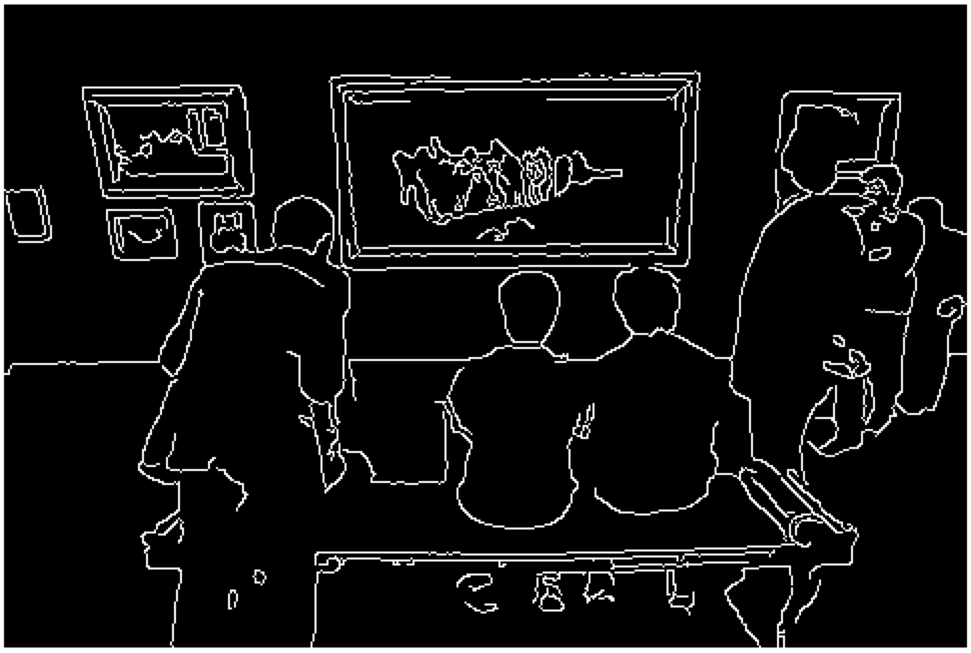


Figure. 1.32 The Canny edge map of “Gallery”
with the low threshold 55 (0.216) and the high threshold 165 (0.647)

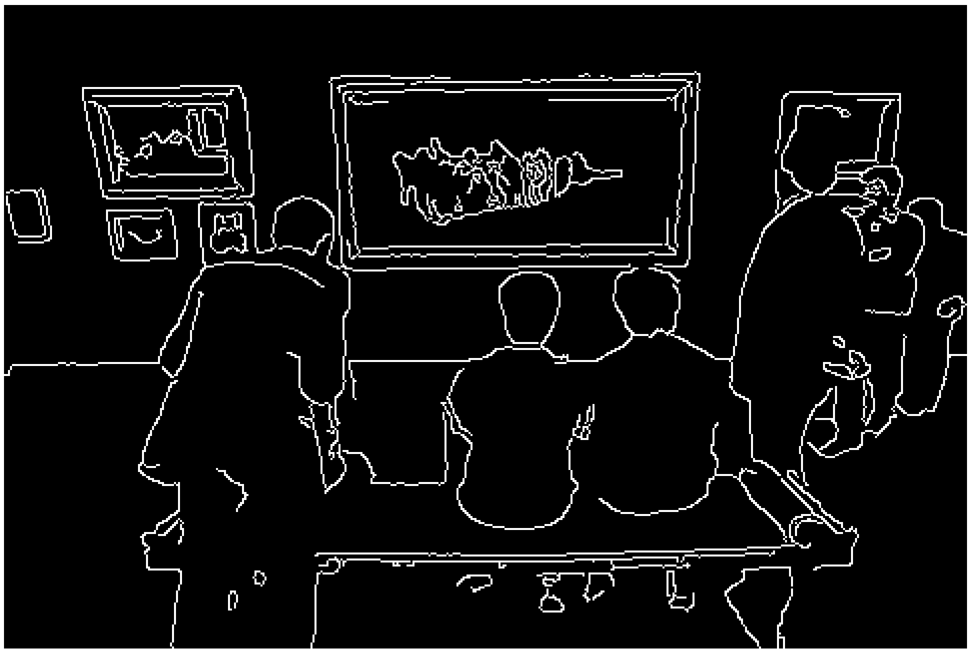


Figure. 1.33 The Canny edge map of “Gallery”
with the low threshold 60 (0.235) and the high threshold 180 (0.706)

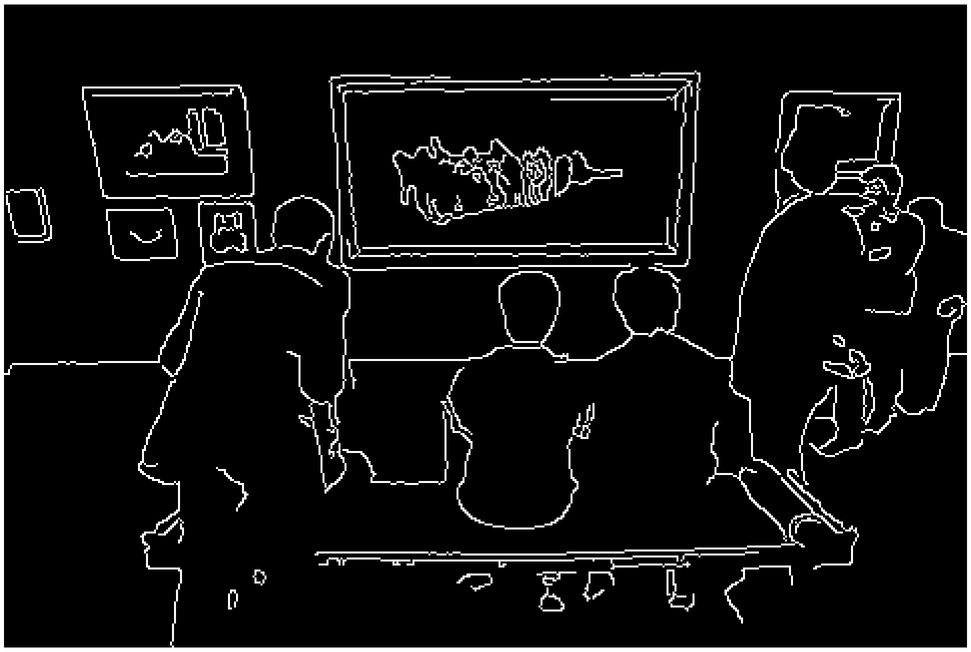


Figure. 1.34 The Canny edge map of “Gallery”
with the low threshold 65 (255) and the high threshold 195 (0.765)

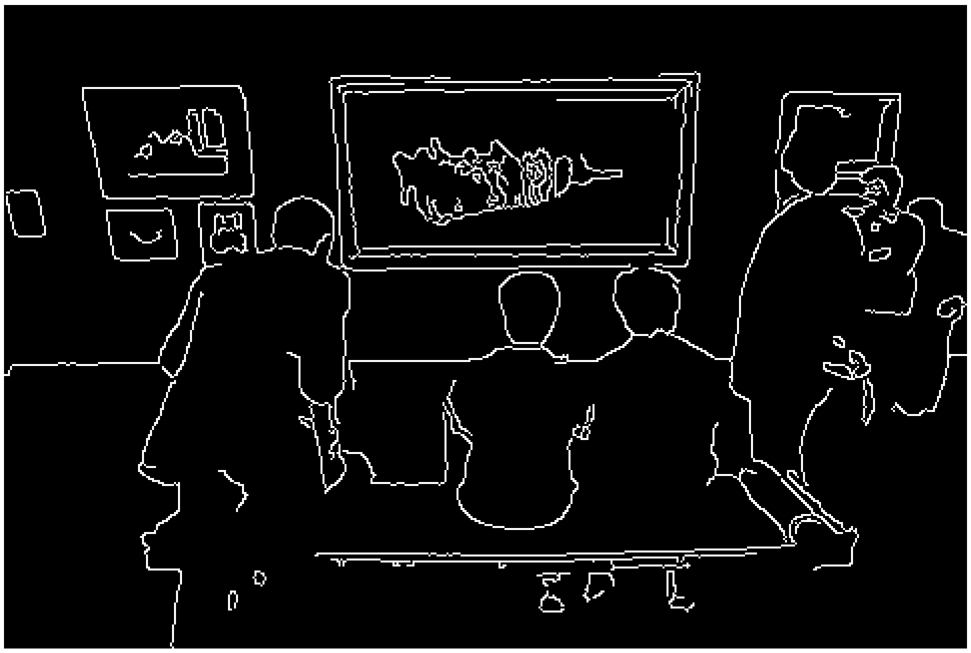


Figure. 1.35 The Canny edge map of “Gallery”
with the low threshold 70 (275) and the high threshold 210 (0.824)

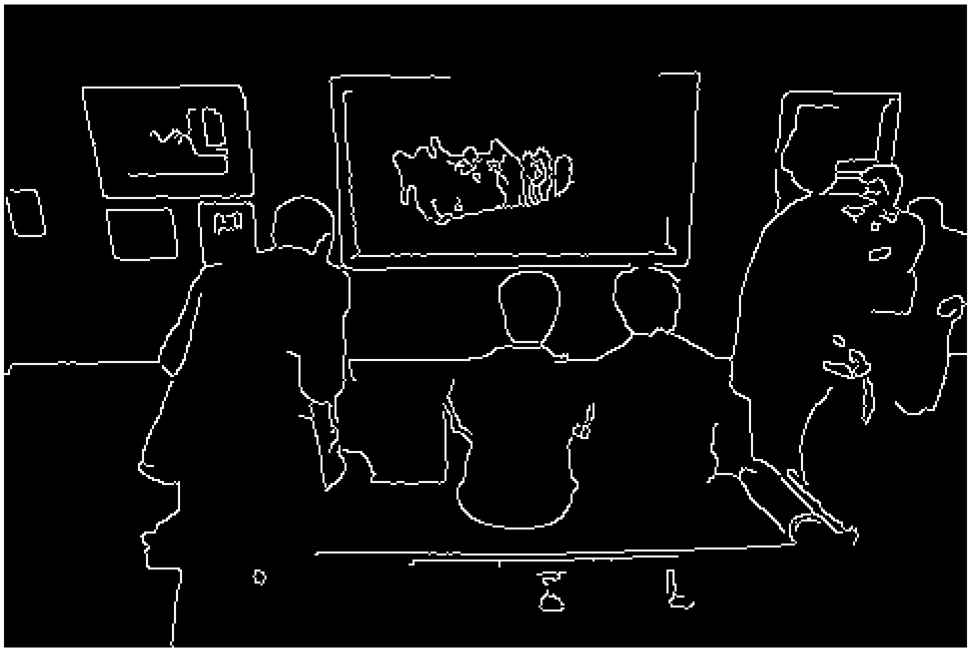


Figure. 1.36 The Canny edge map of “Gallery”
with the low threshold 80 (0.314) and the high threshold 240 (0.941)

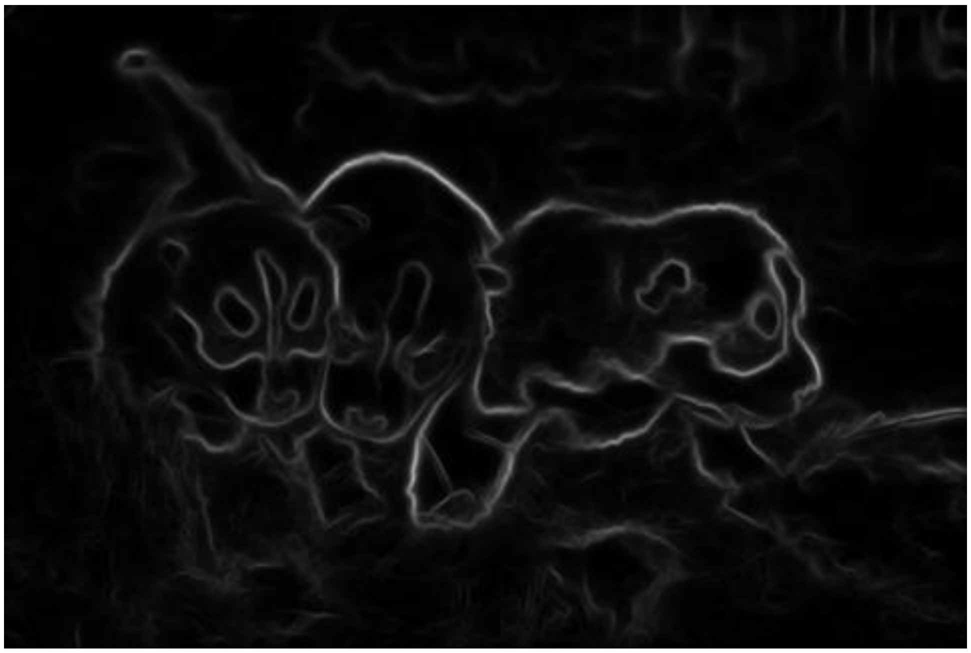


Figure. 1.37 The Structure-Edge map of “Dogs”

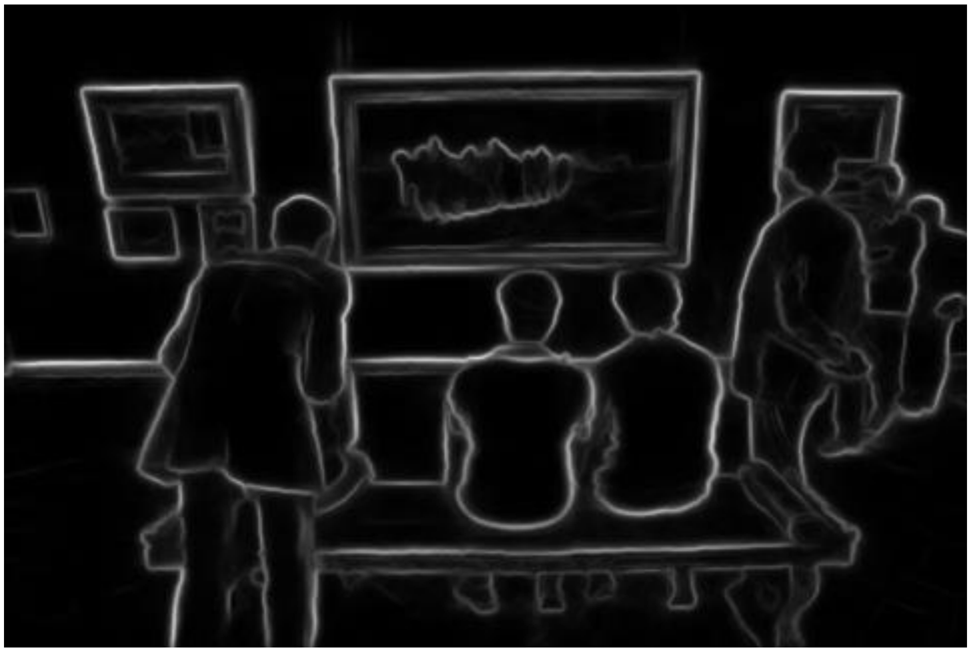


Figure. 1.38 The Structure-Edge map of “Gallery”

Table. 1.1 The performance evaluation for the normalized gradient magnitude of " Dogs"

	The 1st GT_R	The 1st GT_P	The 2nd GT_R	The 2nd GT_P	The 3rd GT_R	The 3rd GT_P	The4th GT_R	The 4th GT_P	The 5th GT_R	The 5th GT_P	Means_R	Means_P	F scores
Threshold 1	0.854005168	0.05191847	0.864470003	0.109217296	0.749667111	0.06633154	0.862095532	0.043946118	0.84770017	0.097710403	0.835588	0.073825	0.135664
Threshold 2	0.965762274	0.0512267	0.98197078	0.108244243	0.941855304	0.072711075	0.959167951	0.042660362	0.975809199	0.098135965	0.964913	0.074596	0.138485
Threshold 3	0.968346253	0.049948352	0.976064657	0.104628303	0.951620062	0.071440472	0.966101695	0.041784679	0.981942078	0.096031455	0.968815	0.072767	0.135366
Threshold 4	0.96124031	0.048974756	0.93472179	0.098969819	0.960497115	0.07122404	0.946070878	0.040417339	0.978194208	0.094493631	0.956145	0.070816	0.131865
Threshold 5	0.949612403	0.048723898	0.877525645	0.093569771	0.958277852	0.071561153	0.937596302	0.040338084	0.974446337	0.094796155	0.939492	0.069798	0.129942
Threshold 6	0.927002584	0.049064861	0.781162574	0.085923343	0.942299157	0.072588642	0.923728814	0.040995658	0.963884157	0.09672787	0.907615	0.06906	0.128354
Threshold 7	0.923126615	0.050279723	0.722101337	0.081735337	0.93608522	0.074205693	0.916024653	0.041835263	0.961158433	0.099257591	0.891699	0.069463	0.128885
Threshold 8	0.910206718	0.052408406	0.655268884	0.078408034	0.924101198	0.077440952	0.888289676	0.042886368	0.954684838	0.104221685	0.86651	0.071073	0.131371
Threshold 9	0.898578811	0.05414769	0.632266086	0.079177858	0.913448735	0.08011211	0.873651772	0.044143408	0.950255537	0.10856787	0.85364	0.07323	0.134888
Threshold 10	0.872739018	0.056508282	0.602735468	0.08110256	0.897913893	0.084616028	0.848998459	0.046093358	0.934923339	0.114773298	0.831462	0.076619	0.140308
Threshold 11	0.844315245	0.059428	0.584084551	0.085436275	0.881047492	0.090255991	0.815100154	0.048106216	0.917887564	0.122493521	0.808487	0.081144	0.147486
Threshold 12	0.833333333	0.061897222	0.576313335	0.088959263	0.870838881	0.094141356	0.80046225	0.049853654	0.910732538	0.128256801	0.798336	0.084622	0.153023
Threshold 13	0.813953488	0.065281592	0.567609574	0.094606497	0.859742565	0.100357494	0.782742681	0.05263976	0.898807496	0.136676856	0.784571	0.089912	0.161336
Threshold 14	0.805555556	0.068030551	0.564811937	0.099127114	0.855747892	0.105182761	0.775038521	0.054882706	0.894378194	0.143207856	0.779106	0.094086	0.167897
Threshold 15	0.789405685	0.072522255	0.556419024	0.106231454	0.847758544	0.113353116	0.761941448	0.058694362	0.883134583	0.153827893	0.767732	0.100926	0.178399
Threshold 16	0.776485788	0.075783368	0.54833696	0.111216191	0.842432312	0.119664586	0.751155624	0.061471534	0.870868825	0.161149991	0.757856	0.105857	0.185766
Threshold 17	0.766149871	0.080570652	0.542119988	0.118478261	0.833111407	0.127513587	0.737288136	0.065013587	0.857240204	0.170923913	0.747182	0.1125	0.195556
Threshold 18	0.758397933	0.084163739	0.535903015	0.123593089	0.822902796	0.132912754	0.729583975	0.067890171	0.847018739	0.178220661	0.738761	0.117356	0.202538
Threshold 19	0.743540052	0.0897956	0.522225676	0.131065689	0.803817133	0.141285692	0.719568567	0.072866282	0.818057922	0.187314714	0.721442	0.124466	0.212304
Threshold 20	0.733204134	0.09384043	0.510724277	0.135841257	0.782956059	0.145845391	0.709553159	0.076147168	0.79693356	0.193385697	0.706674	0.129012	0.218191
Threshold 21	0.708010336	0.098587748	0.488032328	0.141225151	0.730581447	0.148061527	0.692604006	0.08086714	0.76592845	0.202212827	0.677031	0.134191	0.223987
Threshold 22	0.678940568	0.10406971	0.461299347	0.146945242	0.68575233	0.152985444	0.664869029	0.085454005	0.728109029	0.211605109	0.643794	0.140212	0.230273
Threshold 23	0.659560724	0.107677705	0.444513522	0.150812065	0.657789614	0.15629614	0.643297381	0.08806159	0.704599659	0.218097448	0.621952	0.144189	0.234105
Threshold 24	0.629844961	0.112638632	0.419645633	0.155961183	0.622281403	0.161968577	0.61633282	0.092421442	0.680408859	0.230707024	0.593703	0.150739	0.240433
Threshold 25	0.61627907	0.117155839	0.404724899	0.159891932	0.603639592	0.167014614	0.600924499	0.095787793	0.665076661	0.239715093	0.578129	0.155913	0.245593
Threshold 26	0.589147287	0.123560493	0.380167858	0.165695705	0.566799822	0.173011787	0.578582435	0.101747731	0.635775128	0.252811272	0.550095	0.163365	0.251917
Threshold 27	0.570413437	0.127767327	0.366179671	0.170452901	0.545938748	0.177977138	0.56394453	0.105918102	0.619080068	0.262914195	0.533111	0.169006	0.256649
Threshold 28	0.547157623	0.134915578	0.346285359	0.177445046	0.503772747	0.180790061	0.538520801	0.111341191	0.592844974	0.277158331	0.505716	0.17633	0.261487
Threshold 29	0.532945736	0.140186916	0.33043208	0.180628717	0.483355526	0.185046729	0.51540832	0.113678845	0.57274276	0.285641461	0.486977	0.181037	0.263949
Threshold 30	0.514211886	0.149147461	0.315511346	0.19018175	0.462494452	0.195240772	0.498459168	0.121229155	0.542078365	0.298107551	0.466551	0.190781	0.27082
Threshold 31	0.481912145	0.155676127	0.295617035	0.19845576	0.43186862	0.203046745	0.475346687	0.12875626	0.510051107	0.312395659	0.438959	0.199666	0.274481
Threshold 32	0.456072351	0.158687345	0.276966118	0.200269724	0.405681314	0.205439425	0.454545455	0.132614071	0.484156729	0.319397617	0.415484	0.203282	0.272996
Threshold 33	0.430232558	0.166084788	0.259558595	0.208229426	0.375943187	0.211221945	0.427580894	0.13840399	0.45451448	0.332668329	0.389566	0.211322	0.274007
Threshold 34	0.414082687	0.172080537	0.248989742	0.215033557	0.359076787	0.217181208	0.412942989	0.143892617	0.432367973	0.340671141	0.373492	0.217772	0.275126
Threshold 35	0.386950904	0.17880597	0.231271371	0.222089552	0.328894807	0.22119403	0.38366718	0.148656716	0.400681431	0.351044776	0.346293	0.224358	0.272298
Threshold 36	0.365633075	0.182286634	0.216972334	0.224798712	0.309809143	0.224798712	0.369029276	0.154267311	0.380579216	0.359742351	0.328405	0.229179	0.269963
Threshold 37	0.343669251	0.190271817	0.200186509	0.230329041	0.288060364	0.23211731	0.340523883	0.158082976	0.350255537	0.367668097	0.304539	0.235694	0.26573
Threshold 38	0.333979328	0.198388335	0.192726142	0.23791251	0.278739458	0.240982348	0.325885978	0.162317728	0.337308348	0.379892556	0.293728	0.243899	0.266504
Threshold 39	0.313953488	0.2060195	0.179048803	0.244171259	0.260985353	0.24925816	0.30816641	0.169563374	0.311073254	0.387028402	0.274645	0.251208	0.262404
Threshold 40	0.295865633	0.207146088	0.16847995	0.245137947	0.24545051	0.250113071	0.293528505	0.172320217	0.293696763	0.389868838	0.259404	0.252917	0.25612
Threshold 41	0.276485788	0.212301587	0.154491763	0.246527778	0.2268087	0.253472222	0.277349769	0.178571429	0.273253833	0.39781746	0.241678	0.257738	0.24945
Threshold 42	0.246770026	0.211869107	0.138638483	0.247365502	0.202840657	0.253466445	0.249614792	0.179700499	0.248381601	0.404326123	0.217249	0.259346	0.236438
Threshold 43	0.226744186	0.210179641	0.128691327	0.247904192	0.18819352	0.253892216	0.234976888	0.182634731	0.230664395	0.405389222	0.201854	0.26	0.227267
Threshold 44	0.201550388	0.209818426	0.110972956	0.240080699	0.165113182	0.250168124	0.213405239	0.186281103	0.20988075	0.414256893	0.180185	0.260121	0.212897
Threshold 45	0.19121447	0.214648296	0.104134287	0.242929659	0.157123835	0.256707759	0.198767334	0.187092096	0.197614991	0.420594634	0.169771	0.264394	0.206771
Threshold 46	0.17377261	0.21763754	0.092943736	0.241909385	0.140701287	0.256472492	0.181047766	0.19012945	0.180919932	0.42961165	0.153877	0.267152	0.195277
Threshold 47	0.162144703	0.218641115	0.086415915	0.242160279	0.131380382	0.257839721	0.167950693	0.18989547	0.170017036	0.43466899	0.143582	0.268641	0.187141
Threshold 48	0.149870801	0.224588577	0.078333851	0.243949661	0.11939636	0.260406583	0.153312789	0.192642788	0.157069847	0.446272991	0.131597	0.273572	0.17771
Threshold 49	0.140180879	0.222108495	0.073981971	0.243602866	0.112294718	0.258955988	0.146379045	0.194472876	0.148892675	0.447287615	0.124346	0.273286	0.170922
Threshold 50	0.123385013	0.217787913	0.064656512	0.237172178	0.098979139	0.254275941	0.130970724	0.193842645	0.134923339	0.451539339	0.110583	0.270924	0.157059
Threshold 51	0.111757106	0.218710493	0.058128691	0.236409608	0.089214381	0.254108723	0.1201849	0.19721871	0.123679727	0.458912769	0.100593	0.273072	0.147025
Threshold 52	0.102713178	0.218707015	0.053465962	0.236588721	0.082556591	0.255845942	0.111710324	0.199449794	0.114480409	0.462173315	0.092985	0.274553	0.138921
Threshold 53	0.088501292	0.212403101	0.045694747	0.227906977	0.071904128	0.251162791	0.096302003	0.19379845	0.101873935	0.463565891	0.080855	0.269767	0.124419
Threshold 54	0.080749354	0.206270627	0.041653715	0.221122112	0.066134043	0.245874587	0.087827427	0.188118812	0.095059625	0.46039604	0.074285	0.264356	0.115979
Threshold 55	0.074935401	0.211678832	0.037923531	0.222627737	0.059920107	0.246350365	0.083204931	0.197080292	0.088586031	0.474452555	0.068914	0.270438	0.109839
Threshold 56	0.071059432	0.212765957	0.03605844	0.224371373	0.056813138	0.247582205	0.078582435	0.19729207	0.085178876	0.483558994	0.065538	0.273114	0.10571
Threshold 57	0.058139535	0.195227766	0.030774013	0.214750542	0.047492233	0.232104121	0.06779661	0.190889371	0.07427598	0.472885033	0.055696	0.261171	0.091812
Threshold 58	0.052971576	0.189814815	0.027665527	0.206018519	0.041278296	0.215277778	0.060862866	0.18287037	0.068824532	0.467592593	0.050321	0.252315	0.083907
Threshold 59	0.043927649	0.176623377	0.023313646	0.194805195	0.033732801	0.197402597	0.050847458	0.171428571	0.060306644	0.45974026	0.042426	0.24	0.072105
Threshold 60	0.040697674	0.178977273	0.020826857	0.190340909	0.029738127	0.190340909	0.046995378	0.173295455	0.056558773	0.471590909	0.038963	0.240909	0.067078
Threshold 61	0.035529716	0.176282051	0.01802922	0.185897436	0.026187306	0.189102564	0.040832049	0.169871795	0.049744463	0.467948718	0.034065	0.237821	0.059593
Threshold 62	0.031653747	0.173144876	0.016164128	0.183745583	0.023080337	0.183745583	0.036209553	0.166077739	0.046678024	0.48409894	0.030757	0.238163	0.054479
Threshold 63	0.027131783	0.1640625	0.013677339	0.171875	0.019973369	0.17578125	0.030816641	0.15625	0.042589438	0.48828125	0.026838	0.23125	0.048094
Threshold 64	0.025193798	0.174887892	0.012744793	0.183856502	0.018641811	0.188340807	0.029275809	0.170403587	0.038500852	0.506726457	0.024871	0.244843	0.045156
Threshold 65	0.023901809	0.177884615	0.012123096	0.1875	0.017754106	0.192307692	0.027734977	0.173076923	0.034752981	0.490384615	0.023253	0.244231	0.042464
Threshold 66	0.021317829	0.183333333	0.010568853	0.188888889	0.01509099	0.188888889	0.024653313	0.177777778	0.03032368	0.494444444	0.020391	0.246667	0.037668
Threshold 67	0.019379845	0.173410405	0.009636307	0.179190751	0.013759432	0.179190751	0.022342065	0.167630058	0.028279387	0.479768786	0.018679	0.235838	0.034617
Threshold 68	0.01744186	0.178807947	0.008392913	0.178807947	0.011984021	0.178807947	0.020030817	0.17218543	0.024190801	0.470198675	0.016408	0.235762	0.030681
Threshold 69	0.016149871	0.176056338	0.007771215	0.176056338	0.011096316	0.176056338	0.018489985	0.169014085	0.022487223	0.464788732	0.015199	0.232394	0.028532
Threshold 70	0.015503876	0.188976378	0.007460367	0.188976378	0.010652463	0.188976378	0.017719569	0.181102362	0.021124361	0.488188976	0.014492	0.247244	0.027379
Threshold 71	0.012273902	0.166666667	0.005906124	0.166666667	0.0084332	0.166666667	0.014637904	0.166666667	0.018398637	0.473684211	0.01193	0.22807	0.022674
Threshold 72	0.010981912	0.161904762	0.005284426	0.161904762	0.007545495	0.161904762	0.013097072	0.161904762	0.016013629	0.447619048	0.010585	0.219048	0.020193
Threshold 73	0.010981912	0.180851064	0.005284426	0.180851064	0.007545495	0.180851064	0.013097072	0.180851064	0.014310051	0.446808511	0.010244	0.234043	0.019628
Threshold 74	0.010981912	0.197674419	0.005284426	0.197674419	0.007545495	0.197674419	0.013097072	0.197674419	0.012947189	0.441860465	0.009971	0.246512	0.019167
Threshold 75	0.010335917	0.205128205	0.004973578	0.205128205	0.007101642	0.205128205	0.012326656	0.205128205	0.012265758	0.461538462	0.009401	0.25641	0.018136
Threshold 76	0.010335917	0.222222222	0.004973578	0.222222222	0.007101642	0.222222222	0.012326656	0.222222222	0.011243612	0.458333333	0.009196	0.269444	0.017786
Threshold 77	0.009043928	0.225806452	0.004351881	0.225806452	0.006213937	0.225806452	0.010785824	0.225806452	0.00988075	0.467741935	0.008055	0.274194	0.015651
Threshold 78	0.009043928	0.25	0.004351881	0.25	0.006213937	0.25	0.010785824	0.25	0.009199319	0.482142857	0.007919	0.296429	0.015426
Threshold 79	0.008397933	0.254901961	0.004041032	0.254901961	0.005770084	0.254901961	0.010015408	0.254901961	0.008177172	0.470588235	0.00728	0.298039	0.014213
Threshold 80	0.008397933	0.26	0.004041032	0.26	0.005770084	0.26	0.010015408	0.26	0.008177172	0.48	0.00728	0.304	0.01422
Threshold 81	0.007751938	0.279069767	0.003730183	0.279069767	0.005326232	0.279069767	0.009244992	0.279069767	0.00681431	0.465116279	0.006574	0.316279	0.012879
Threshold 82	0.006459948	0.27027027	0.003108486	0.27027027	0.004438526	0.27027027	0.00770416	0.27027027	0.005451448	0.432432432	0.005433	0.302703	0.010673
Threshold 83	0.006459948	0.294117647	0.003108486	0.294117647	0.004438526	0.294117647	0.00770416	0.294117647	0.005110733	0.441176471	0.005364	0.323529	0.010554
Threshold 84	0.005167959	0.285714286	0.002486789	0.285714286	0.003550821	0.285714286	0.006163328	0.285714286	0.004088586	0.428571429	0.004291	0.314286	0.008467
Threshold 85	0.004521964	0.291666667	0.00217594	0.291666667	0.003106968	0.291666667	0.005392912	0.291666667	0.003407155	0.416666667	0.003721	0.316667	0.007356
Threshold 86	0.003875969	0.285714286	0.001865092	0.285714286	0.002663116	0.285714286	0.004622496	0.285714286	0.002385009	0.333333333	0.003082	0.295238	0.006101
Threshold 87	0.002583979	0.235294118	0.001243394	0.235294118	0.001775411	0.235294118	0.003081664	0.235294118	0.001362862	0.235294118	0.002009	0.235294	0.003985
Threshold 88	0.002583979	0.266666667	0.001243394	0.266666667	0.001775411	0.266666667	0.003081664	0.266666667	0.001362862	0.266666667	0.002009	0.266667	0.003989
Threshold 89	0.00129199	0.153846154	0.000621697	0.153846154	0.000887705	0.153846154	0.001540832	0.153846154	0.000681431	0.153846154	0.001005	0.153846	0.001996
Threshold 90	0.00129199	0.333333333	0.000621697	0.333333333	0.000887705	0.333333333	0.001540832	0.333333333	0.000681431	0.333333333	0.001005	0.333333	0.002003
Threshold 91	0.000645995	0.25	0.000310849	0.25	0.000443853	0.25	0.000770416	0.25	0.000340716	0.25	0.000502	0.25	0.001003
Threshold 92	0.000645995	0.25	0.000310849	0.25	0.000443853	0.25	0.000770416	0.25	0.000340716	0.25	0.000502	0.25	0.001003
Threshold 93	0.000645995	0.25	0.000310849	0.25	0.000443853	0.25	0.000770416	0.25	0.000340716	0.25	0.000502	0.25	0.001003
Threshold 94	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 95	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 96	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 97	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 98	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 99	0	0	0	0	0	0	0	0	0	0	0	0	0
Means	0.302619215	0.15973863	0.2157415	0.18066047	0.294063583	0.183537827	0.299061493	0.147262546	0.323146284	0.315633724			
Mean F scores	0.209102016		0.196648673		0.226011858		0.197347904		0.319345827				0.2297
Best F scores													0.275126

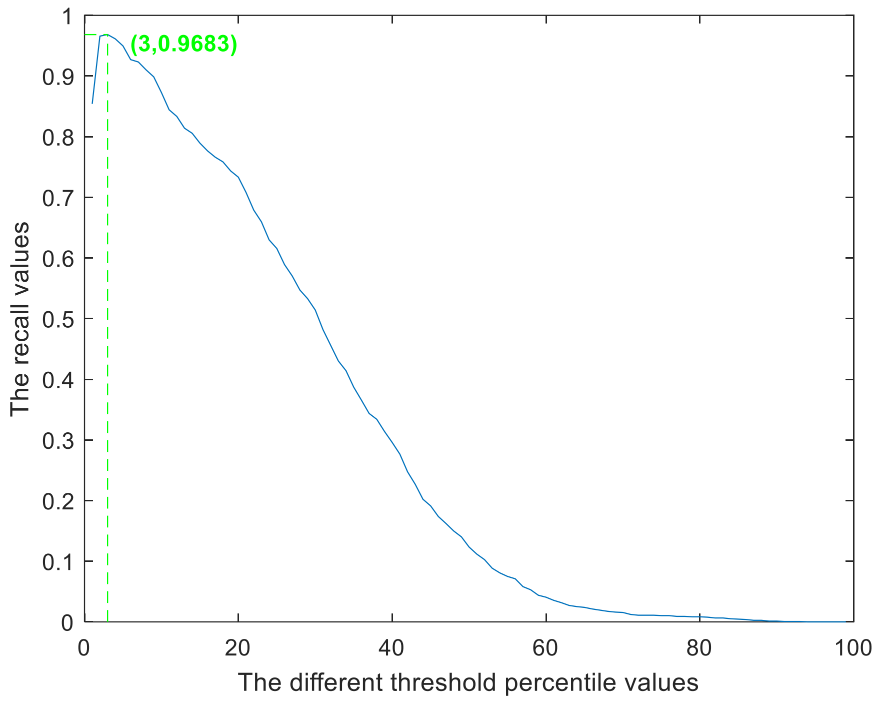


Figure. 1.39 The recall values under different threshold values for the “Dogs” ’s Sobel result with the 1st ground truth image

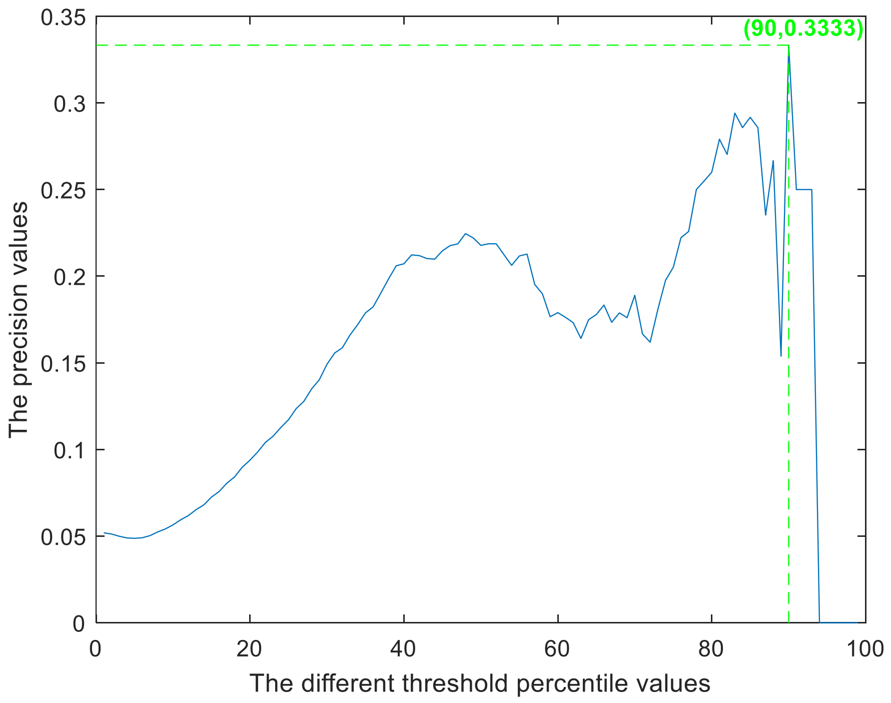


Figure. 1.40 The precision values under different threshold values for the “Dogs” ’s Sobel result with the 1st ground truth image

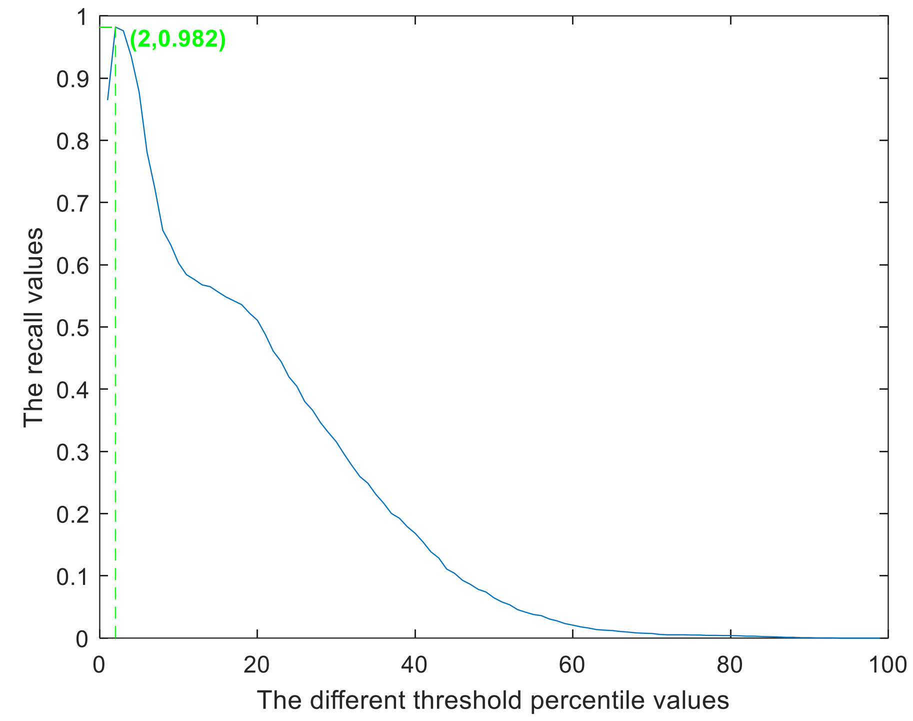


Figure. 1.41 The recall values under different threshold values for the “Dogs” ’s Sobel result with the 2nd ground truth image

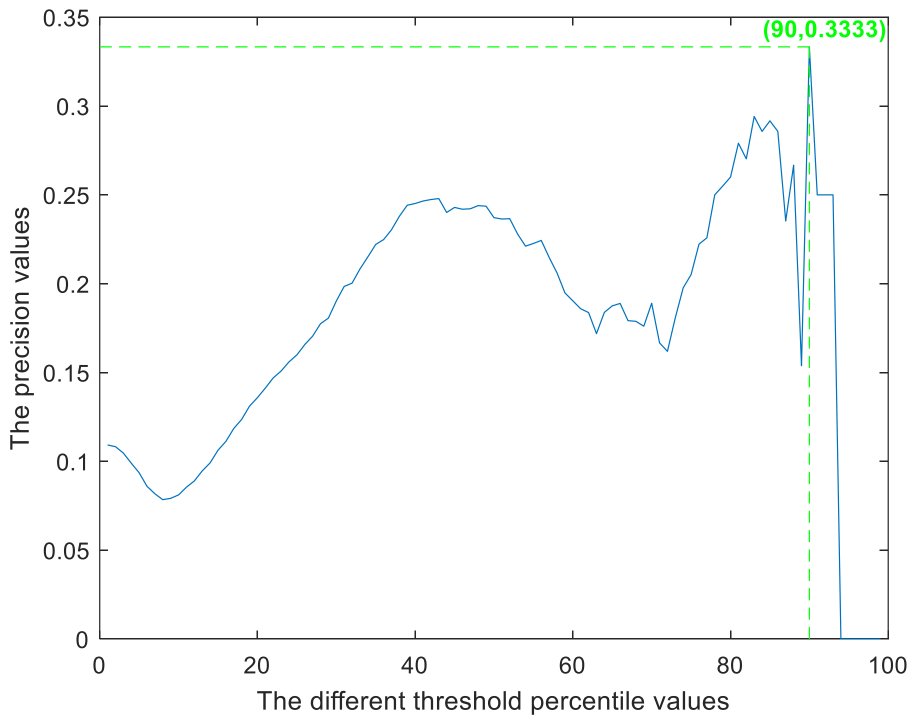


Figure. 1.42 The precision values under different threshold values for the “Dogs” ’s Sobel result with the 2nd ground truth image

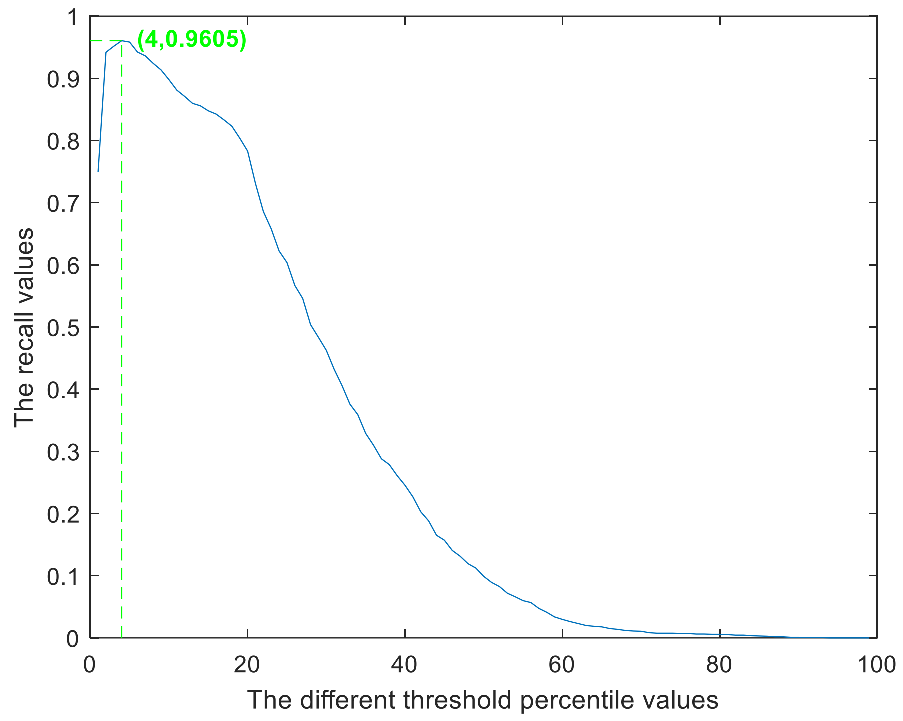


Figure. 1.43 The recall values under different threshold values for the “Dogs” ’s Sobel result with the 3rd ground truth image

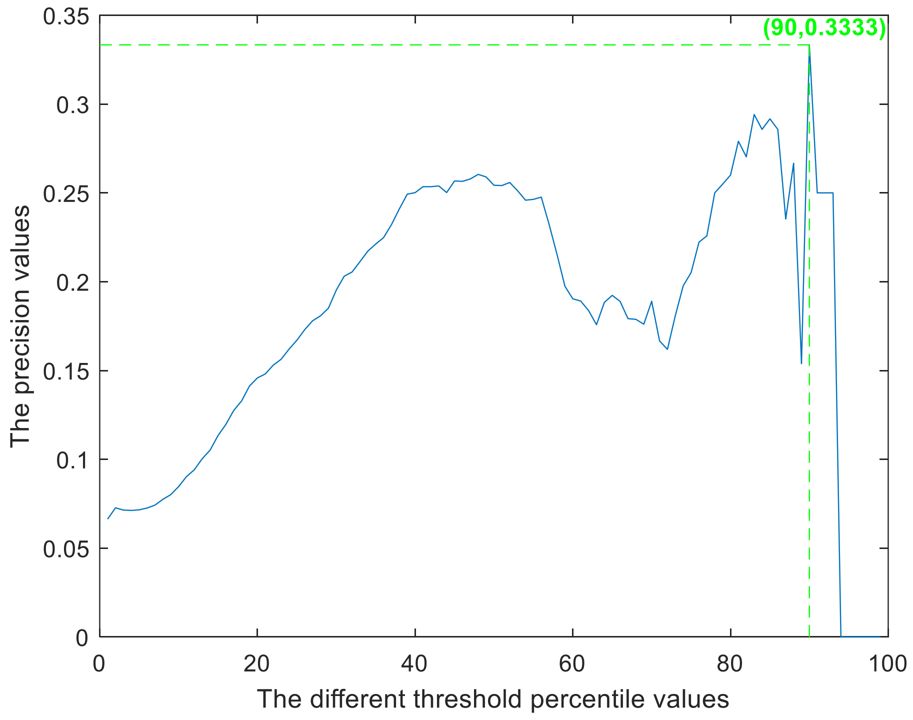


Figure. 1.44 The precision values under different threshold values for the “Dogs” ’s Sobel result with the 3rd ground truth image

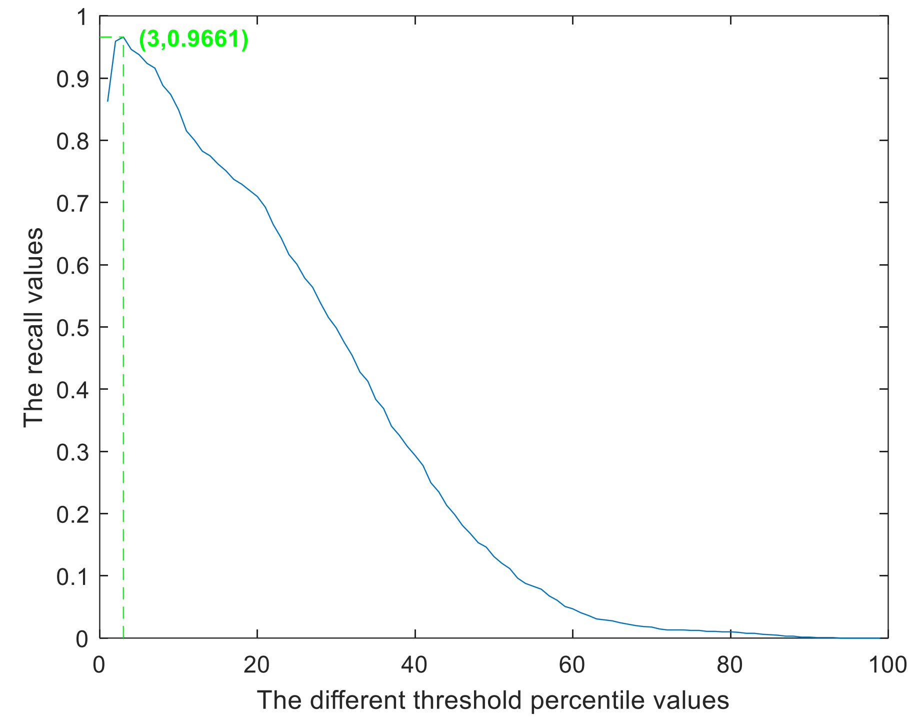


Figure. 1.45 The recall values under different threshold values for the “Dogs” ’s Sobel result with the 4th ground truth image

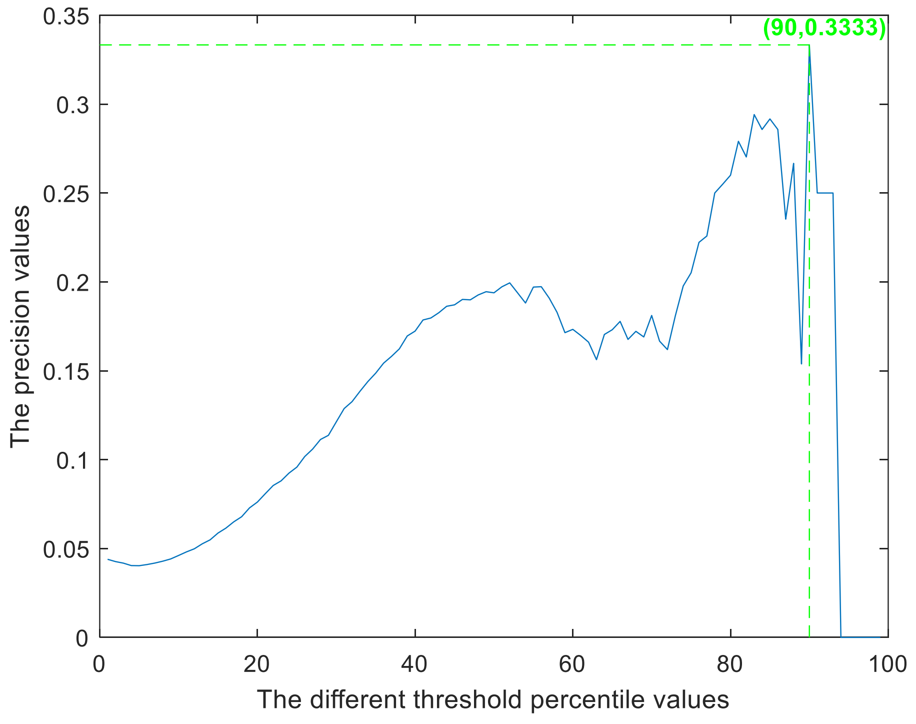


Figure. 1.46 The precision values under different threshold values for the “Dogs” ’s Sobel result with the 4th ground truth image

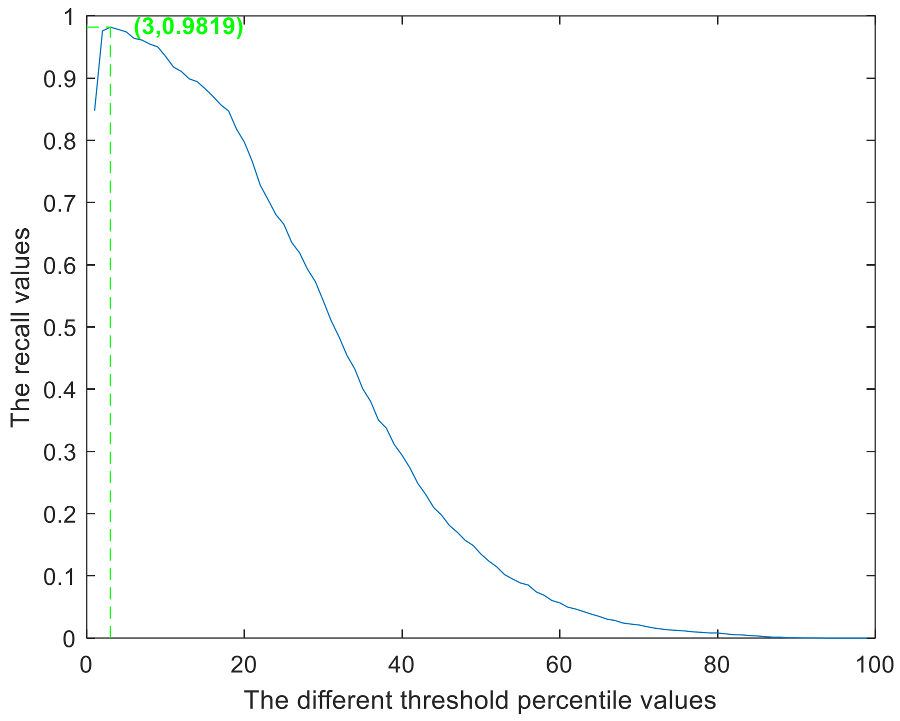


Figure. 1.47 The recall values under different threshold values for the “Dogs” ’s Sobel result with the 5th ground truth image

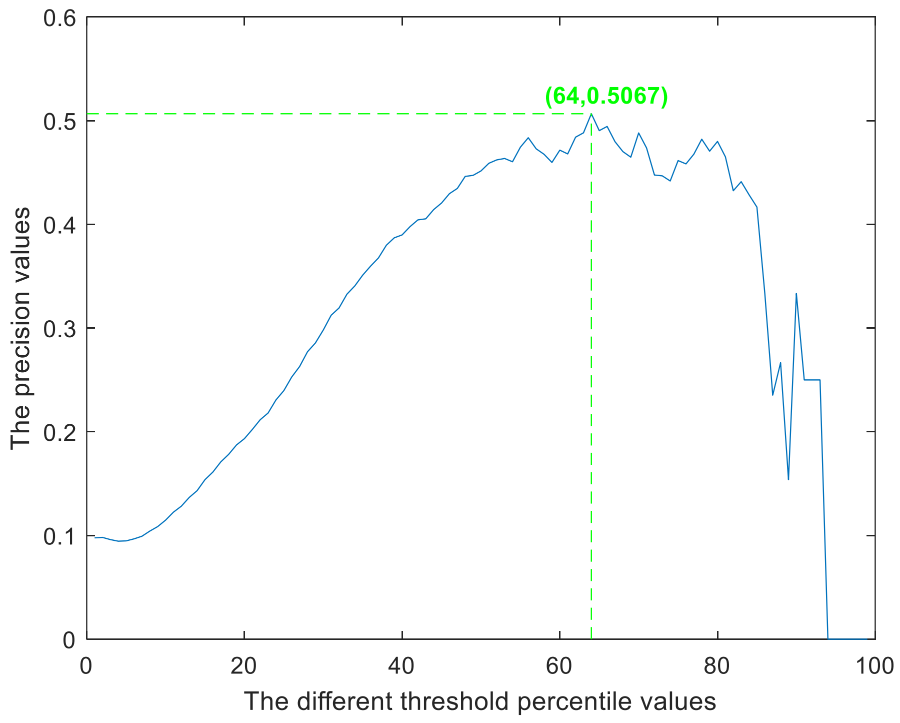


Figure. 1.48 The precision values under different threshold values for the “Dogs” ’s Sobel result with the 5th ground truth image

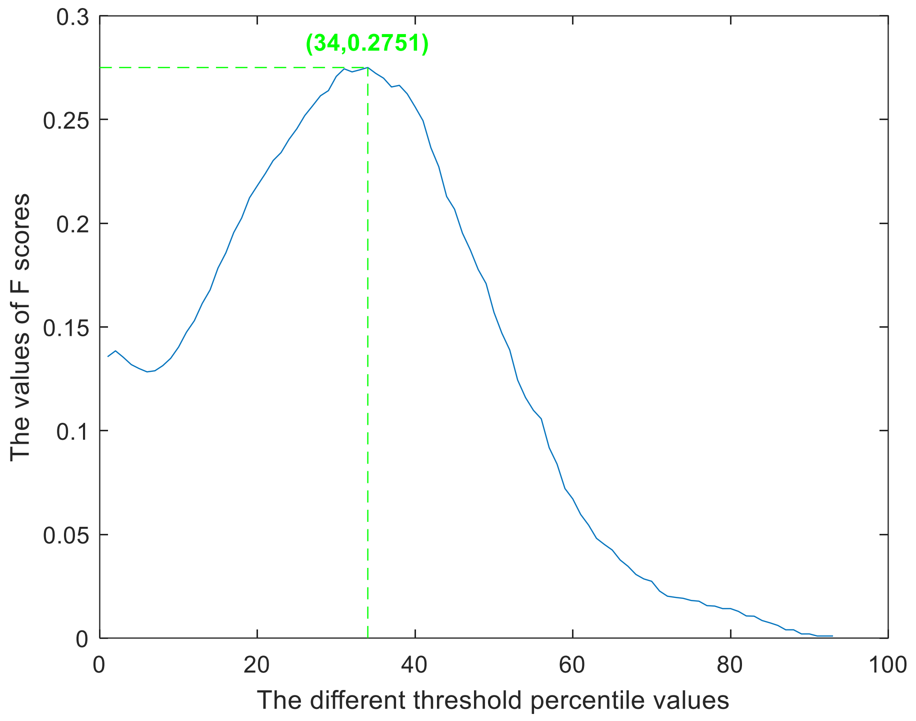


Figure. 1.49 The F values under different threshold values for the “Dogs” ’s Sobel result among 5 ground truth images

Table. 1.2 The performance evaluation for the normalized gradient magnitude of “ Gallery”

	The 1st GT_R	The 1st GT_P	The 2nd GT_R	The 2nd GT_P	The 3rd GT_R	The 3rd GT_P	The4th GT_R	The 4th GT_P	The 5th GT_R	The 5th GT_P	Means_R	Means_P	F scores
Threshold 1	0.99056816	0.100606696	0.988676397	0.089613174	0.991491373	0.095680139	0.9875191	0.088427151	0.985628743	0.131397683	0.98878	0.10114	0.183517
Threshold 2	0.98854705	0.129818042	0.985153498	0.115456074	0.985582605	0.122976201	0.98573612	0.114128992	0.987681779	0.170249786	0.98654	0.13053	0.230549
Threshold 3	0.98585223	0.159607344	0.977604429	0.141247046	0.98085559	0.150881658	0.98573612	0.140701691	0.981864842	0.208652972	0.98238	0.16022	0.275504
Threshold 4	0.99011902	0.206578269	0.984901862	0.183385653	0.98085559	0.194443143	0.99210392	0.182495432	0.977416595	0.267675585	0.98508	0.20692	0.341995
Threshold 5	0.98854705	0.23647596	0.987418218	0.210797744	0.980146537	0.22277733	0.9875191	0.208272898	0.973139435	0.305560032	0.98335	0.23678	0.381656
Threshold 6	0.98225915	0.275857719	0.984650226	0.246783552	0.972583314	0.259523209	0.97962303	0.242558022	0.967493584	0.356647326	0.97732	0.27627	0.430775
Threshold 7	0.97597126	0.300075951	0.981378963	0.269281226	0.964783739	0.281847683	0.97376465	0.263964648	0.963045338	0.38866257	0.97179	0.30077	0.459361
Threshold 8	0.96631484	0.330035281	0.973578259	0.296747967	0.955329709	0.310016874	0.96688742	0.291148949	0.956201882	0.428670041	0.96366	0.33132	0.493108
Threshold 9	0.95800584	0.348216472	0.966532461	0.313525426	0.947530135	0.327238593	0.96001019	0.307648355	0.950213858	0.453350747	0.95646	0.35	0.512466
Threshold 10	0.94700202	0.373516386	0.959486663	0.337732507	0.936658	0.351018601	0.94803872	0.329672276	0.940461933	0.486891054	0.94633	0.37577	0.537932
Threshold 11	0.92948574	0.393328899	0.940613991	0.355221895	0.920349799	0.370046565	0.930973	0.34733441	0.926946108	0.514872185	0.92967	0.39616	0.555575
Threshold 12	0.91107119	0.404728651	0.924509311	0.366520351	0.904277948	0.381683958	0.91619969	0.358838787	0.913430282	0.532621708	0.9139	0.40888	0.564983
Threshold 13	0.88210195	0.420917274	0.903371917	0.384697814	0.878988419	0.398521217	0.89658686	0.377196742	0.892557742	0.559044149	0.89072	0.42808	0.578248
Threshold 14	0.86346283	0.429945209	0.889531958	0.395281226	0.860789411	0.407245891	0.88537952	0.388683887	0.878699743	0.574303925	0.87557	0.43909	0.584875
Threshold 15	0.83179879	0.438810567	0.862103674	0.405876081	0.83171827	0.416893733	0.8660214	0.402795877	0.857142857	0.593531572	0.84976	0.45158	0.589754
Threshold 16	0.81203683	0.444499078	0.847005536	0.413767671	0.813519263	0.423110018	0.85506877	0.41266134	0.844824636	0.607006761	0.83449	0.46021	0.59325
Threshold 17	0.78621154	0.453908985	0.821841973	0.423440944	0.785866225	0.431090367	0.83851248	0.426811876	0.824123182	0.624530014	0.81131	0.47196	0.596763
Threshold 18	0.76555131	0.4610495	0.802466029	0.431295645	0.763649255	0.436975926	0.82322975	0.437111171	0.81043627	0.640654585	0.79307	0.48142	0.599138
Threshold 19	0.74376825	0.473210459	0.779818822	0.44277754	0.740959584	0.447921132	0.8074376	0.452921846	0.790419162	0.660094299	0.77248	0.49539	0.603653
Threshold 20	0.72692567	0.482774049	0.764720684	0.453243848	0.723233278	0.456375839	0.79215487	0.46383296	0.771770744	0.672781506	0.75576	0.5058	0.606019
Threshold 21	0.7044689	0.496675111	0.740563664	0.465959468	0.699598204	0.468651045	0.77254203	0.480208993	0.746963216	0.691260291	0.73283	0.52055	0.608713
Threshold 22	0.68268583	0.512907036	0.718671364	0.481862662	0.67619948	0.482706259	0.74987264	0.496709971	0.720273738	0.710308757	0.70954	0.5369	0.611264
Threshold 23	0.66561868	0.52018252	0.702063412	0.48964549	0.65705507	0.487890488	0.7343352	0.505967006	0.702994012	0.721130221	0.69241	0.54496	0.609903
Threshold 24	0.63844599	0.532596478	0.676396578	0.503559386	0.627038525	0.497002623	0.70427916	0.517984264	0.668776732	0.73229674	0.66299	0.55669	0.605205
Threshold 25	0.62070514	0.538791423	0.657523905	0.509356725	0.610493973	0.503508772	0.68644931	0.525341131	0.648417451	0.738791423	0.64472	0.56316	0.601184
Threshold 26	0.59443072	0.553071458	0.632360342	0.52507313	0.582604585	0.515043878	0.65792155	0.539699122	0.613344739	0.749059758	0.61613	0.57639	0.595599
Threshold 27	0.57893555	0.559461806	0.617765476	0.532769097	0.56511463	0.518880208	0.6400917	0.545355903	0.592643285	0.751736111	0.59891	0.58164	0.590149
Threshold 28	0.54884348	0.562874251	0.591092099	0.540994933	0.534152683	0.520497467	0.61156393	0.552970981	0.560650128	0.754721327	0.56926	0.58641	0.577709
Threshold 29	0.53042892	0.56412706	0.572974333	0.543826128	0.516190026	0.521614521	0.5962812	0.559111536	0.543199316	0.758299498	0.55181	0.5894	0.569986
Threshold 30	0.49764204	0.564873821	0.542778057	0.54983431	0.48262822	0.52052001	0.57182883	0.572266123	0.514285714	0.766250319	0.52183	0.59475	0.55591
Threshold 31	0.46238491	0.569413717	0.511323603	0.561946903	0.447411959	0.523506637	0.54457463	0.591261062	0.479041916	0.774336283	0.48895	0.60409	0.540455
Threshold 32	0.43700876	0.567512394	0.488173125	0.565762613	0.423540534	0.522601341	0.52241467	0.598133567	0.453892216	0.773694955	0.46501	0.60554	0.526049
Threshold 33	0.39726027	0.568262127	0.450931052	0.575650498	0.38406996	0.522004497	0.47962303	0.60488275	0.414200171	0.777706393	0.42522	0.6097	0.501016
Threshold 34	0.3741298	0.570743405	0.428032209	0.582733813	0.363507445	0.526892771	0.45415181	0.610825625	0.389392643	0.779719082	0.40184	0.61418	0.485824
Threshold 35	0.33774983	0.571428571	0.38827378	0.586246201	0.330891042	0.531914894	0.41594498	0.620440729	0.353293413	0.784574468	0.36523	0.61892	0.459378
Threshold 36	0.31866158	0.578711256	0.367136387	0.59502447	0.315291893	0.544045677	0.39684157	0.635399674	0.331223268	0.789559543	0.34583	0.62855	0.446174
Threshold 37	0.29103975	0.588823262	0.335933568	0.606542481	0.28882061	0.555202181	0.36423841	0.64970468	0.299572284	0.795547478	0.31592	0.63916	0.422843
Threshold 38	0.27689198	0.599708171	0.320583795	0.619649805	0.274166864	0.564202335	0.34360672	0.656128405	0.280410607	0.797178988	0.29913	0.64737	0.40919
Threshold 39	0.24949472	0.603476372	0.29089079	0.627919609	0.247931931	0.569799022	0.31125828	0.66376969	0.252010265	0.800108637	0.27032	0.65301	0.382357
Threshold 40	0.2301819	0.599415205	0.269501761	0.626315789	0.229496573	0.567836257	0.29011717	0.666081871	0.233875107	0.799415205	0.25063	0.65181	0.362053
Threshold 41	0.20682686	0.6	0.243834927	0.631270358	0.207043252	0.570684039	0.26260825	0.671661238	0.211120616	0.803908795	0.22629	0.6555	0.336433
Threshold 42	0.18459466	0.596949891	0.219426271	0.633260712	0.185062633	0.568627451	0.23790117	0.678286129	0.188708298	0.801016703	0.20314	0.65563	0.310174
Threshold 43	0.17538738	0.610633307	0.208354303	0.647380766	0.175372252	0.580140735	0.22312787	0.684910086	0.174508127	0.797498045	0.19135	0.66411	0.297098
Threshold 44	0.16034134	0.622493461	0.190991444	0.661726242	0.160245805	0.591107236	0.20275089	0.693984307	0.156030796	0.795117698	0.17407	0.67289	0.276591
Threshold 45	0.15270604	0.637898687	0.182687469	0.681050657	0.152682581	0.606003752	0.19230769	0.708255159	0.145765612	0.799249531	0.16523	0.68649	0.266352
Threshold 46	0.14237593	0.655635988	0.168847509	0.693898656	0.14275585	0.624612203	0.1777891	0.721820062	0.132934132	0.803516029	0.15294	0.6999	0.251027
Threshold 47	0.1369863	0.677777778	0.159788626	0.705555556	0.137319783	0.645555556	0.16861946	0.735555556	0.124379812	0.807777778	0.14542	0.71444	0.241652
Threshold 48	0.12710532	0.710163112	0.146955209	0.732747804	0.127156701	0.675031368	0.15588385	0.767879548	0.110008554	0.806775408	0.13342	0.73852	0.226012
Threshold 49	0.12126656	0.725806452	0.138651233	0.740591398	0.121484283	0.690860215	0.14773306	0.779569892	0.10402053	0.817204301	0.12663	0.75081	0.216712
Threshold 50	0.11385583	0.747787611	0.128334172	0.752212389	0.113448357	0.707964602	0.13550688	0.784660767	0.09546621	0.82300885	0.11732	0.76313	0.203378
Threshold 51	0.10375028	0.754901961	0.116758933	0.758169935	0.103757977	0.717320261	0.1235354	0.79248366	0.086056459	0.821895425	0.10677	0.76895	0.187508
Threshold 52	0.09880979	0.759930915	0.111222949	0.763385147	0.098794611	0.72193437	0.11742231	0.796200345	0.081437126	0.822107081	0.10154	0.77271	0.179489
Threshold 53	0.08803054	0.758220503	0.099396074	0.764023211	0.08886788	0.727272727	0.10570555	0.80270793	0.071856287	0.81237911	0.09077	0.77292	0.162463
Threshold 54	0.08219178	0.760914761	0.092853548	0.767151767	0.082722761	0.727650728	0.09780948	0.798336798	0.067065868	0.814968815	0.08453	0.7738	0.152409
Threshold 55	0.07612845	0.786542923	0.086562657	0.798143852	0.077286693	0.758700696	0.0899134	0.819025522	0.061248931	0.83062645	0.07823	0.79861	0.142498
Threshold 56	0.0716371	0.793532338	0.08178158	0.808457711	0.072559679	0.763681592	0.08456444	0.825870647	0.057142857	0.830845771	0.07354	0.80448	0.134756
Threshold 57	0.06332809	0.801136364	0.072471062	0.818181818	0.064051052	0.769886364	0.07590423	0.846590909	0.049615056	0.823863636	0.06507	0.81193	0.120491
Threshold 58	0.05928587	0.8	0.068193256	0.821212121	0.060033089	0.76969697	0.07208355	0.857575758	0.0465355	0.824242424	0.06123	0.81455	0.113892
Threshold 59	0.05456995	0.80730897	0.062908908	0.830564784	0.055306074	0.777408638	0.06622517	0.863787375	0.042429427	0.823920266	0.05629	0.8206	0.105349
Threshold 60	0.04918033	0.808118081	0.057121288	0.837638376	0.050342709	0.78597786	0.05985736	0.867158672	0.038665526	0.833948339	0.05103	0.82657	0.096132
Threshold 61	0.04109589	0.809734513	0.04781077	0.840707965	0.041834082	0.783185841	0.05068772	0.880530973	0.031993157	0.827433628	0.04268	0.82832	0.081185
Threshold 62	0.03660454	0.802955665	0.043029693	0.842364532	0.037343418	0.778325123	0.04559348	0.881773399	0.028913601	0.832512315	0.0383	0.82759	0.073206
Threshold 63	0.03413429	0.821621622	0.040261701	0.864864865	0.035216261	0.805405405	0.04151808	0.881081081	0.027031651	0.854054054	0.03563	0.84541	0.068383
Threshold 64	0.02964294	0.814814815	0.035228988	0.864197531	0.030725597	0.802469136	0.03616913	0.87654321	0.023609923	0.851851852	0.03108	0.84198	0.059938
Threshold 65	0.02739726	0.829931973	0.032209361	0.870748299	0.028362089	0.816326531	0.03311258	0.884353741	0.0213858	0.850340136	0.02849	0.85034	0.055139
Threshold 66	0.02425331	0.857142857	0.028686462	0.904761905	0.025053179	0.841269841	0.02852776	0.888888889	0.018306245	0.849206349	0.02497	0.86825	0.048535
Threshold 67	0.0222322	0.876106195	0.026170106	0.920353982	0.022926022	0.85840708	0.02572593	0.89380531	0.016766467	0.867256637	0.02276	0.88319	0.044384
Threshold 68	0.01976196	0.88	0.023402114	0.93	0.020562515	0.87	0.02266938	0.89	0.01471343	0.86	0.02022	0.886	0.039541
Threshold 69	0.01841455	0.901098901	0.021640664	0.945054945	0.01914441	0.89010989	0.0208864	0.901098901	0.013515825	0.868131868	0.01872	0.9011	0.036679
Threshold 70	0.01616887	0.911392405	0.019124308	0.962025316	0.016780903	0.898734177	0.01808456	0.898734177	0.011462789	0.848101266	0.01632	0.9038	0.032069
Threshold 71	0.01414777	0.926470588	0.016607952	0.970588235	0.014653746	0.911764706	0.01579215	0.911764706	0.009751925	0.838235294	0.01419	0.91176	0.027946
Threshold 72	0.01302493	0.920634921	0.015349774	0.968253968	0.013471992	0.904761905	0.01451859	0.904761905	0.008896493	0.825396825	0.01305	0.90476	0.025733
Threshold 73	0.01167752	0.912280702	0.01383996	0.964912281	0.012053888	0.894736842	0.01299032	0.894736842	0.007869974	0.807017544	0.01169	0.89474	0.023071
Threshold 74	0.01055468	0.921568627	0.012581782	0.980392157	0.010872134	0.901960784	0.01171676	0.901960784	0.007014542	0.803921569	0.01055	0.90196	0.020852
Threshold 75	0.00943184	0.913043478	0.011323603	0.97826087	0.009690381	0.891304348	0.0104432	0.891304348	0.00615911	0.782608696	0.00941	0.8913	0.018623
Threshold 76	0.00830901	0.925	0.010065425	1	0.008508627	0.9	0.00916964	0.9	0.005474765	0.8	0.00831	0.905	0.01646
Threshold 77	0.00718617	0.914285714	0.008807247	1	0.007326873	0.885714286	0.00789608	0.885714286	0.004619333	0.771428571	0.00717	0.89143	0.01422
Threshold 78	0.00651246	0.90625	0.00805234	1	0.006617821	0.875	0.00713194	0.875	0.00427716	0.78125	0.00652	0.8875	0.012942
Threshold 79	0.00561419	0.892857143	0.007045798	1	0.005672418	0.857142857	0.00611309	0.857142857	0.003592814	0.75	0.00561	0.87143	0.011144
Threshold 80	0.00538962	0.888888889	0.006794162	1	0.005436067	0.851851852	0.00585838	0.851851852	0.003421728	0.740740741	0.00538	0.86667	0.010694
Threshold 81	0.00449135	0.869565217	0.00578762	1	0.004490664	0.826086957	0.00483953	0.826086957	0.002737382	0.695652174	0.00447	0.84348	0.008892
Threshold 82	0.00359308	0.842105263	0.004781077	1	0.003545261	0.789473684	0.00382068	0.789473684	0.002053037	0.631578947	0.00356	0.81053	0.007086
Threshold 83	0.00336852	0.833333333	0.004529441	1	0.00330891	0.777777778	0.00356597	0.777777778	0.00188195	0.611111111	0.00333	0.8	0.006634
Threshold 84	0.00291938	0.8125	0.00402617	1	0.002836209	0.75	0.00305655	0.75	0.001539778	0.5625	0.00288	0.775	0.00573
Threshold 85	0.00291938	0.8125	0.00402617	1	0.002836209	0.75	0.00305655	0.75	0.001539778	0.5625	0.00288	0.775	0.00573
Threshold 86	0.00291938	0.8125	0.00402617	1	0.002836209	0.75	0.00305655	0.75	0.001539778	0.5625	0.00288	0.775	0.00573
Threshold 87	0.00247024	0.785714286	0.003522899	1	0.002363507	0.714285714	0.00254712	0.714285714	0.001197605	0.5	0.00242	0.74286	0.004825
Threshold 88	0.00224568	0.833333333	0.003019628	1	0.002127157	0.75	0.00229241	0.75	0.001026518	0.5	0.00214	0.76667	0.004273
Threshold 89	0.00202111	0.818181818	0.002767992	1	0.001890806	0.727272727	0.0020377	0.727272727	0.000855432	0.454545455	0.00191	0.74545	0.003819
Threshold 90	0.00179654	0.8	0.002516356	1	0.001654455	0.7	0.00178299	0.7	0.000684346	0.4	0.00169	0.72	0.003366
Threshold 91	0.00179654	0.8	0.002516356	1	0.001654455	0.7	0.00178299	0.7	0.000684346	0.4	0.00169	0.72	0.003366
Threshold 92	0.00134741	0.75	0.002013085	1	0.001181754	0.625	0.00127356	0.625	0.000342173	0.25	0.00123	0.65	0.002459
Threshold 93	0.00089827	0.666666667	0.001509814	1	0.000709052	0.5	0.00076414	0.5	0	0	0.00078	0.53333	0.00155
Threshold 94	0.00089827	0.666666667	0.001509814	1	0.000709052	0.5	0.00076414	0.5	0	0	0.00078	0.53333	0.00155
Threshold 95	0.00044914	0.666666667	0.000754907	1	0.000472701	0.666666667	0.00050942	0.666666667	0	0	0.00044	0.6	0.000874
Threshold 96	0.00022457	0.5	0.000503271	1	0.000236351	0.5	0.00025471	0.5	0	0	0.00024	0.5	0.000487
Threshold 97	0.00022457	1	0.000251636	1	0.000236351	1	0.00025471	1	0	0	0.00019	0.8	0.000387
Threshold 98	0.00022457	1	0.000251636	1	0.000236351	1	0.00025471	1	0	0	0.00019	0.8	0.000387
Threshold 99	0.00022457	1	0.000251636	1	0.000236351	1	0.00025471	1	0	0	0.00019	0.8	0.000387
Means	0.30431873	0.660591733	0.320627005	0.707287647	0.301590712	0.62601758	0.32812588	0.657609915	0.306280945	0.638955954			
Mean F scores	0.41668206		0.441234142		0.40707072		0.43780257		0.414076161				0.4234
Best F scores													0.611264

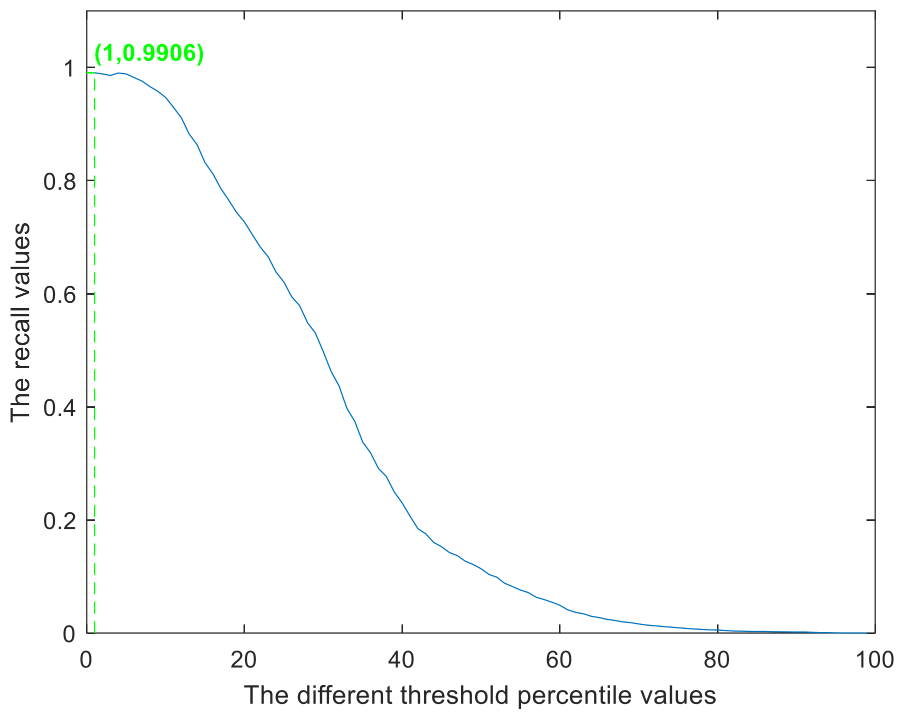


Figure. 1.50 The recall values under different threshold values
for the “Gallery” ’s Sobel result with the 1st ground truth image

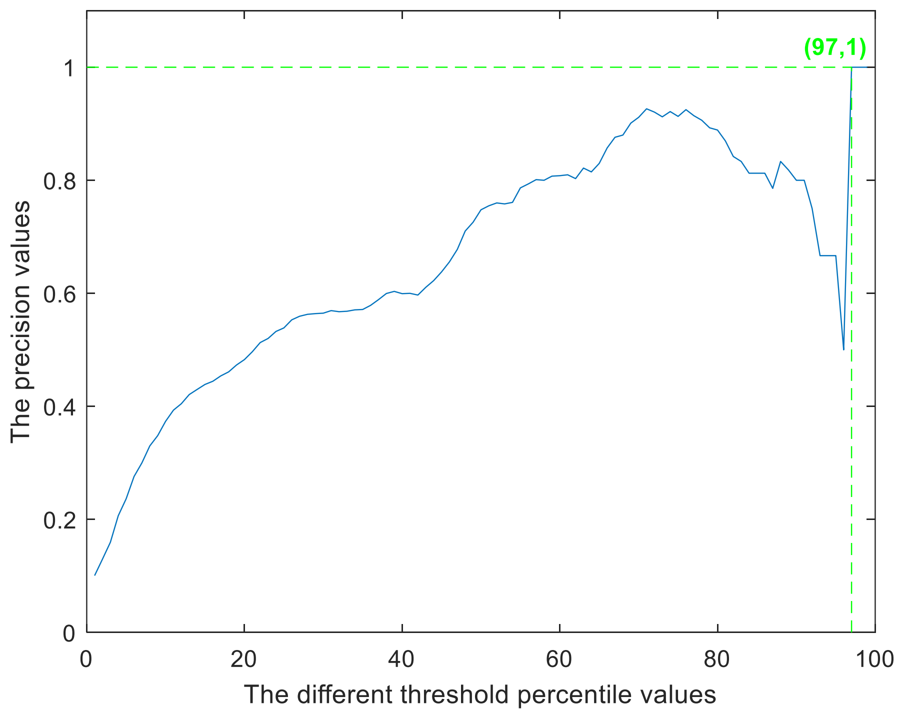


Figure. 1.51 The precision values under different threshold values
for the “Gallery” ’s Sobel result with the 1st ground truth image

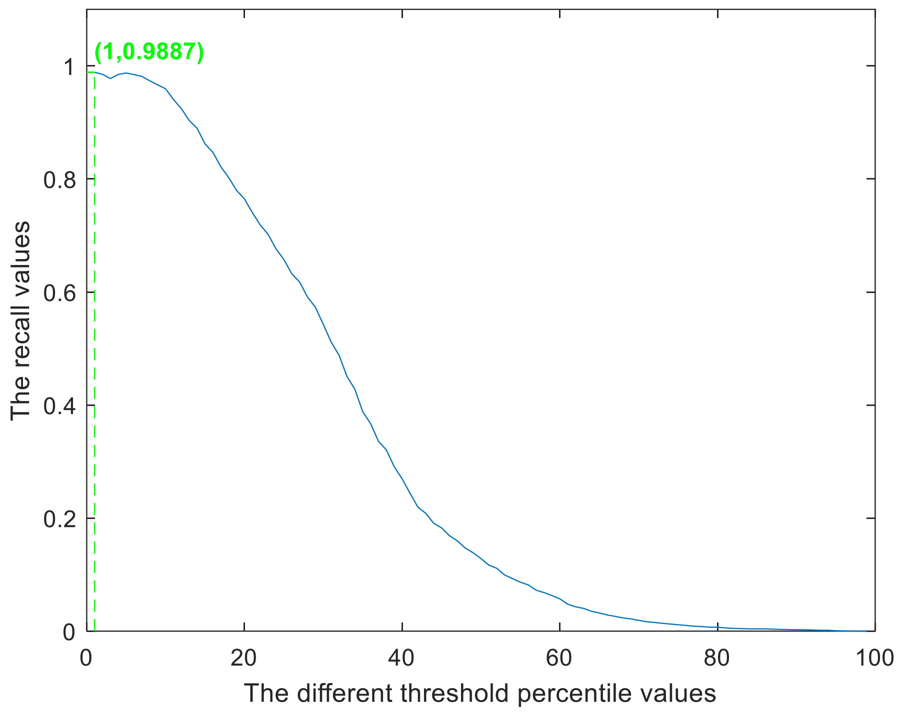


Figure. 1.52 The recall values under different threshold values for the “Gallery” ’s Sobel result with the 2nd ground truth image

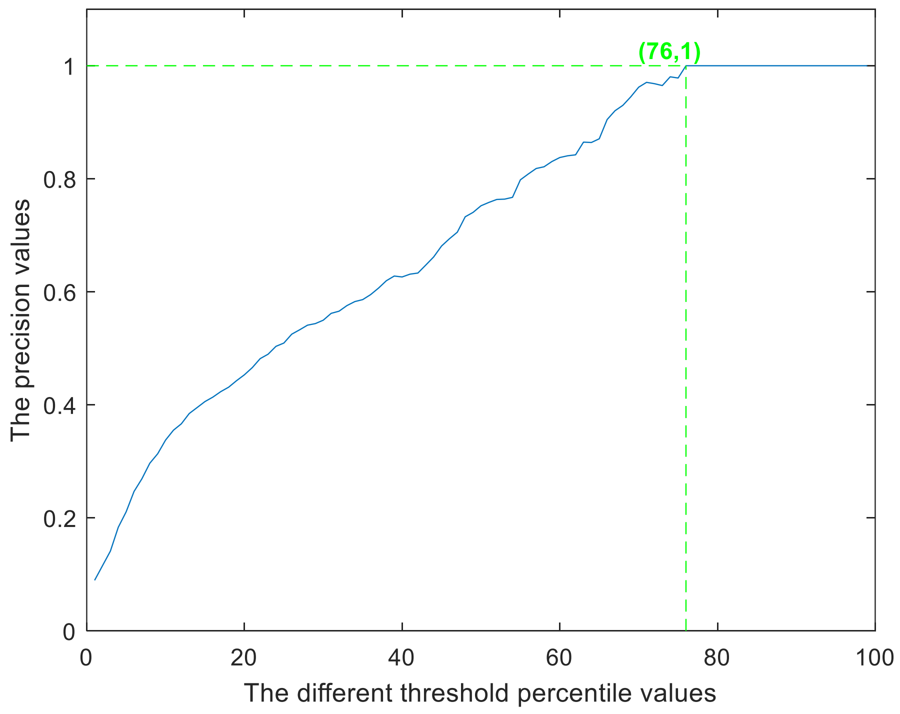


Figure. 1.53 The precision values under different threshold values for the “Gallery” ’s Sobel result with the 2nd ground truth image

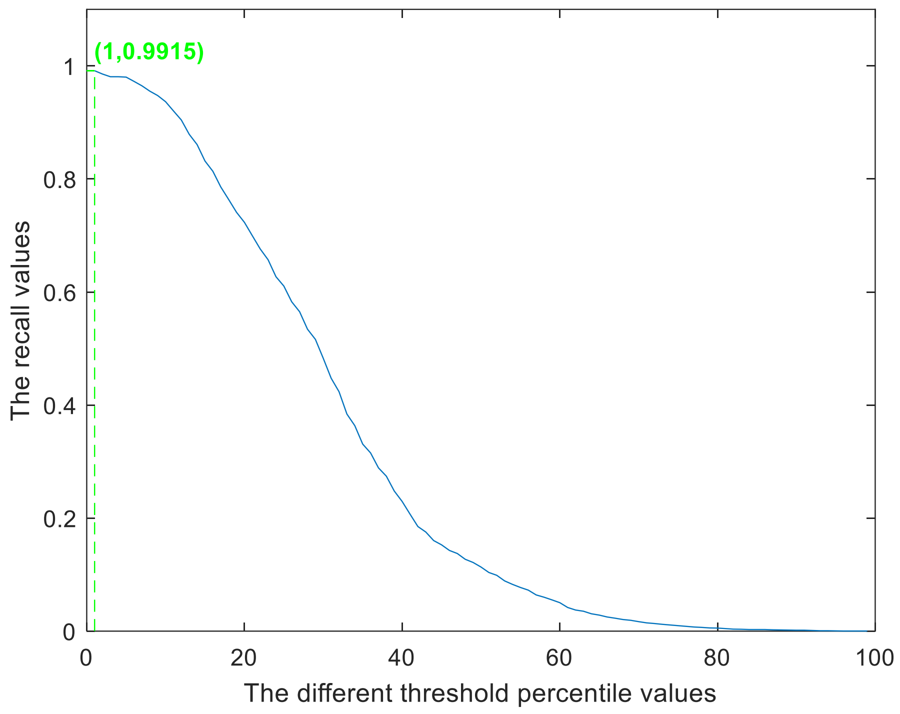


Figure. 1.54 The recall values under different threshold values for the “Gallery” ’s Sobel result with the 3rd ground truth image

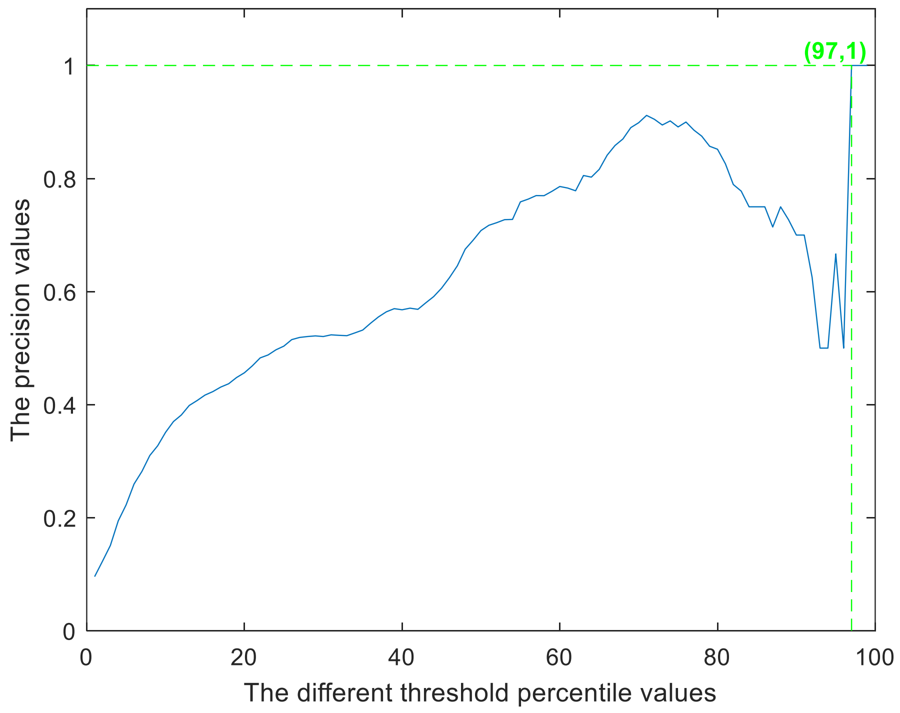


Figure. 1.55 The precision values under different threshold values for the “Gallery” ’s Sobel result with the 3rd ground truth image

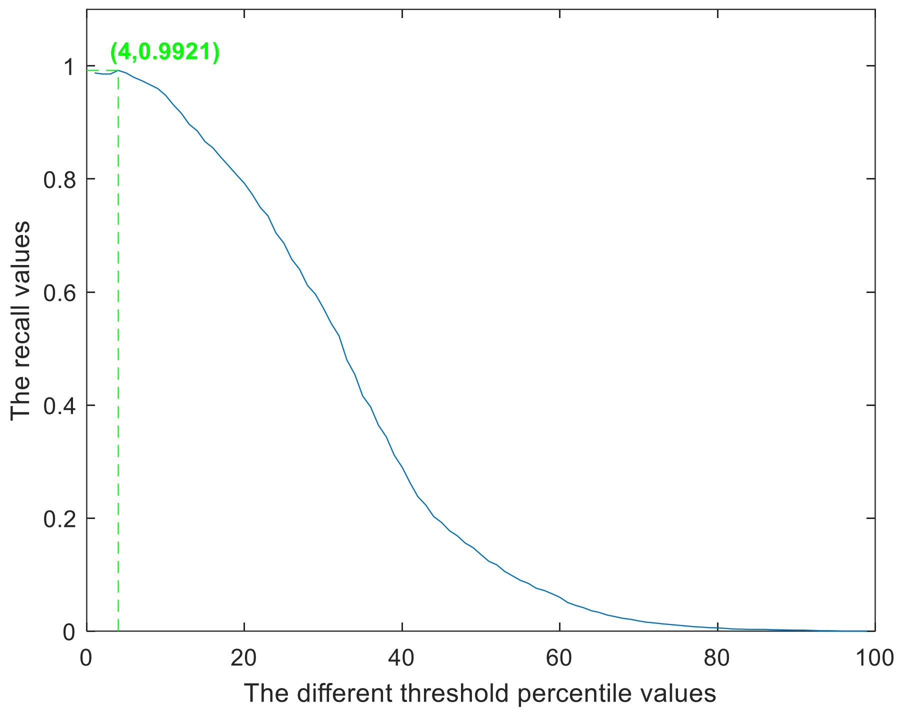


Figure. 1.56 The recall values under different threshold values for the “Gallery” ’s Sobel result with the 4th ground truth image

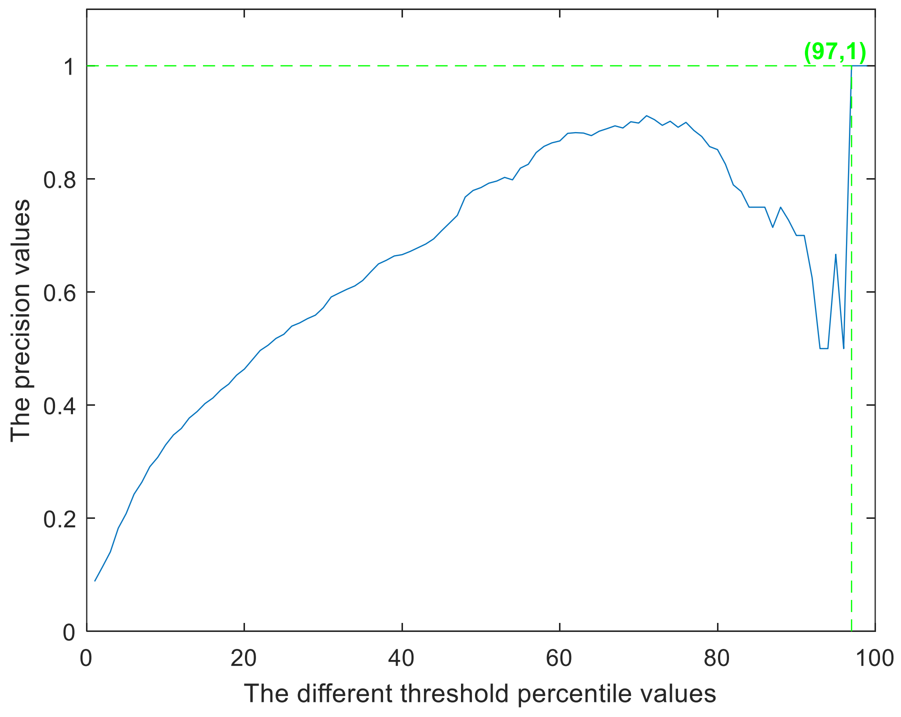


Figure. 1.57 The precision values under different threshold values for the “Gallery” ’s Sobel result with the 4th ground truth image

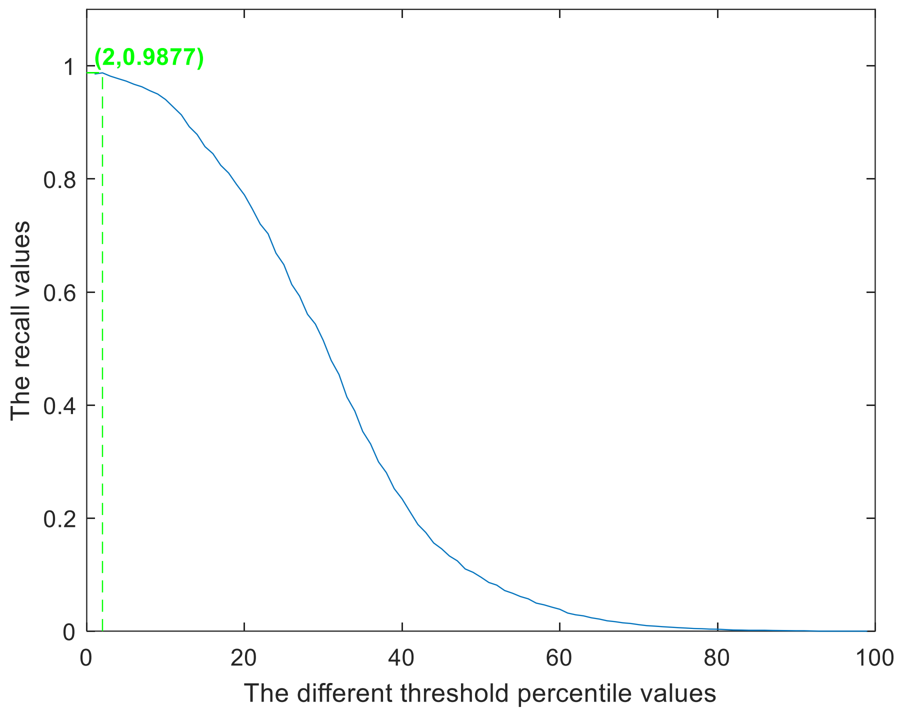


Figure. 1.58 The recall values under different threshold values for the “Gallery” ’s Sobel result with the 5th ground truth image

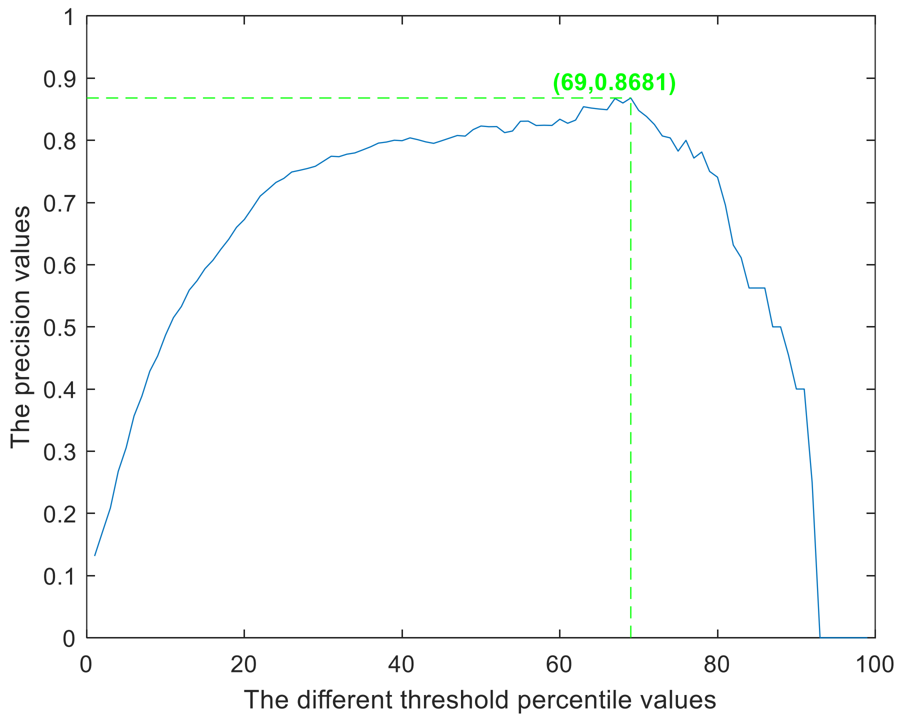


Figure. 1.59 The precision values under different threshold values for the “Gallery” ’s Sobel result with the 5th ground truth image

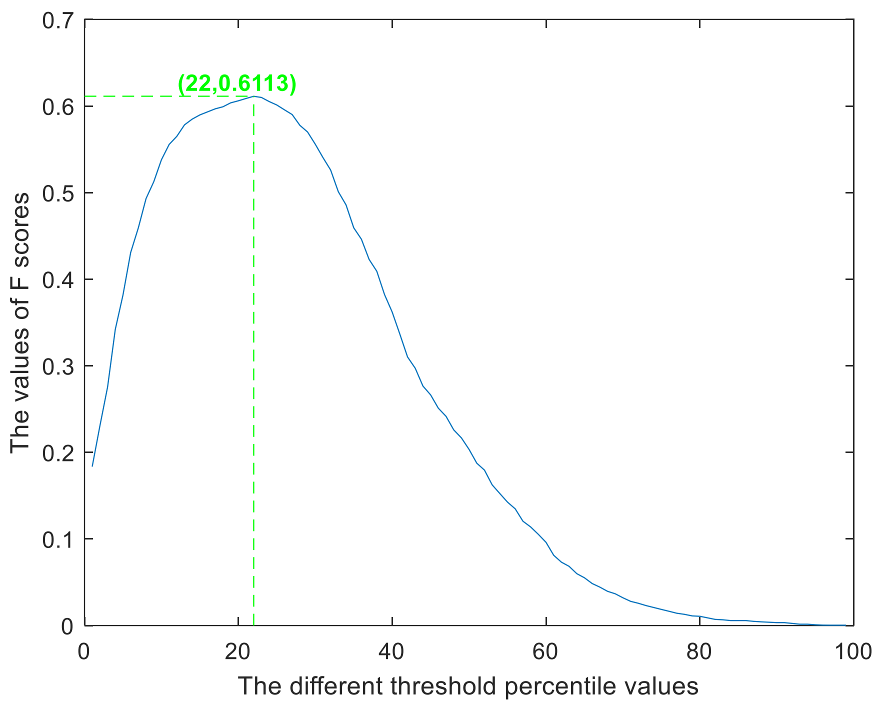


Figure. 1.60 The F values under different threshold values
for the “Gallery” ’s Sobel result among 5 ground truth images

Table. 1.3 The performance evaluation for the Canny edge map of “ Dogs”

			The 1st ground truth			The 2nd ground truth			The 3rd ground truth			The 4th ground truth			The 5th ground truth			Mean F
Sample	**Low threshold**	**High threshold**	**Recall**	**Precision**	**F score**	**Recall**	**Precision**	**F score**	**Recall**	**Precision**	**F score**	**Recall**	**Precision**	**F score**	**Recall**	**Precision**	**F score**	
1	50	100	0.8339	0.088	0.1592	0.5859	0.1285	0.2108	0.8666	0.1331	0.2308	0.8181	0.0724	0.133	0.907	0.1815	0.3025	0.20726
2	55	110	0.8739	0.0847	0.1544	0.6046	0.1218	0.2027	0.8951	0.1263	0.2213	0.852	0.0692	0.1281	0.9334	0.1715	0.2898	0.19926
3	60	120	0.8339	0.088	0.1592	0.5859	0.1285	0.2108	0.8666	0.1331	0.2308	0.8181	0.0724	0.133	0.907	0.1815	0.3025	0.20726
4	65	130	0.8119	0.0908	0.1633	0.5716	0.1328	0.2155	0.844	0.1373	0.2362	0.7943	0.0745	0.1362	0.8931	0.1893	0.3124	0.21272
5	70	140	0.8048	0.1	0.1779	0.5604	0.1447	0.2301	0.8254	0.1493	0.2529	0.785	0.0818	0.1482	0.8856	0.2087	0.3378	0.22938
6	80	160	0.7757	0.1148	0.2	0.5405	0.1663	0.2543	0.7699	0.1659	0.273	0.7565	0.0939	0.1671	0.851	0.2389	0.373	0.25348
7	90	180	0.7389	0.1316	0.2233	0.4933	0.1825	0.2664	0.6887	0.1785	0.2835	0.6933	0.1035	0.1801	0.8217	0.2774	0.4147	0.2736
8	50	150	0.8248	0.0897	0.1618	0.5722	0.1293	0.2109	0.8542	0.1352	0.2334	0.8081	0.0737	0.135	0.8954	0.1846	0.3061	0.20944
9	55	165	0.7925	0.0931	0.1667	0.5561	0.1358	0.2183	0.8227	0.1407	0.2403	0.7665	0.0755	0.1375	0.8645	0.1926	0.315	0.21556
10	60	180	0.7873	0.1018	0.1803	0.5436	0.1461	0.2303	0.8112	0.1526	0.2569	0.7642	0.0828	0.1495	0.8588	0.2105	0.3382	0.23104
11	65	195	0.7783	0.1109	0.1941	0.5321	0.1575	0.2431	0.7739	0.1605	0.2658	0.7642	0.0913	0.1631	0.8425	0.2276	0.3583	0.24488
12	70	210	0.7363	0.1217	0.2089	0.4911	0.1687	0.2511	0.686	0.165	0.2661	0.7203	0.0998	0.1754	0.8207	0.2572	0.3917	0.25864
13	80	240	0.6931	0.1472	0.2428	0.4308	0.1902	0.2639	0.6422	0.1985	0.3033	0.691	0.1231	0.2089	0.8061	0.3246	0.4628	**0.29634**

Table. 1.4 The performance evaluation for the Canny edge map of “ Gallery”

			The 1st ground truth			The 2nd ground truth			The 3rd ground truth			The 4th ground truth			The 5th ground truth			Mean F
Sample	**Low thres**	**High thres**	**Recall**	**Precision**	**F score**	**Recall**	**Precision**	**F score**	**Recall**	**Precision**	**F score**	**Recall**	**Precision**	**F score**	**Recall**	**Precision**	**F score**	
1	50	100	0.9123	0.4404	0.594	0.925	0.3985	0.557	0.9025	0.4139	0.5676	0.9104	0.3874	0.5436	0.9056	0.5738	0.7025	0.5929
2	55	110	0.9073	0.4639	0.6111	0.9197	0.4197	0.5764	0.8978	0.4362	0.5871	0.9055	0.4082	0.5627	0.8989	0.6033	0.722	0.61186
3	60	120	0.8861	0.4805	0.6231	0.9009	0.436	0.5876	0.8797	0.4532	0.5982	0.889	0.425	0.5751	0.8847	0.6297	0.7357	0.62394
4	65	130	0.8494	0.4839	0.6165	0.8709	0.4428	0.5871	0.8435	0.4565	0.5924	0.8632	0.4335	0.5772	0.8627	0.645	0.7381	0.62226
5	70	140	0.8269	0.4932	0.6179	0.8498	0.4524	0.5904	0.8269	0.4932	0.6178	0.8518	0.4479	0.5871	0.849	0.6646	0.7456	0.63176
6	80	160	0.7941	0.5164	0.6258	0.8262	0.4794	0.6068	0.7919	0.4893	0.6049	0.8333	0.4777	0.6073	0.8168	0.6972	0.7522	0.6394
7	90	180	0.755	0.5351	0.6263	0.7933	0.5017	0.6147	0.7537	0.5075	0.6066	0.8228	0.5141	0.6328	0.7757	0.7216	0.7477	**0.64562**
8	50	150	0.88	0.4899	0.6294	0.8953	0.4449	0.5944	0.8694	0.4599	0.6016	0.8912	0.4374	0.5868	0.8863	0.6477	0.7484	0.63212
9	55	165	0.856	0.5043	0.6347	0.8729	0.4589	0.6016	0.8454	0.4732	0.6067	0.8753	0.4546	0.5984	0.8611	0.6658	0.751	0.63848
10	60	180	0.8291	0.5122	0.6332	0.8493	0.4682	0.6037	0.8224	0.4828	0.6084	0.8685	0.473	0.6125	0.8394	0.6806	0.7517	0.6419
11	65	195	0.794	0.5318	0.637	0.8171	0.4884	0.6114	0.789	0.5022	0.6137	0.8203	0.4844	0.6091	0.7926	0.6968	0.7416	0.64256
12	70	210	0.7486	0.5397	0.6272	0.7726	0.4971	0.605	0.749	0.513	0.609	0.792	0.5034	0.6155	0.7619	0.721	0.7409	0.63952
13	80	240	0.6833	0.6081	0.6435	0.7171	0.5695	0.6348	0.6779	0.5732	0.6212	0.7088	0.5561	0.6232	0.6833	0.6081	0.6435	0.63324

Table. 1.5 The performance evaluation for the structured edge map of “ Dogs”

	The 1st GT_R	The 1st GT_P	The 2nd GT_R	The 2nd GT_P	The 3rd GT_R	The 3rd GT_P	The4th GT_R	The 4th GT_P	The 5th GT_R	The 5th GT_P	Means_R	Means_P	F scores
Threshold 1	0.65956072	0.08654743	0.663972645	0.181062982	0.517532179	0.098838688	0.65177196	0.071713147	0.581942078	0.144782572	0.61496	0.11659	0.196013
Threshold 2	0.79844961	0.11684628	0.709978241	0.215919833	0.659565024	0.140480242	0.81201849	0.099640764	0.704599659	0.195500094	0.73692	0.15368	0.254316
Threshold 3	0.85206718	0.14332283	0.703139571	0.245789416	0.699511762	0.171248506	0.86132512	0.121482125	0.737308348	0.235140715	0.77067	0.1834	0.296283
Threshold 4	0.91085271	0.18090839	0.69878769	0.288426995	0.766977364	0.221709007	0.93913713	0.156402361	0.833390119	0.313831152	0.82983	0.23226	0.362929
Threshold 5	0.91343669	0.19923911	0.684488654	0.310271945	0.778961385	0.247287586	0.92604006	0.169367338	0.868824532	0.359306749	0.83435	0.25709	0.393066
Threshold 6	0.92054264	0.22814601	0.664594343	0.342299071	0.765201953	0.27601665	0.91910632	0.191002241	0.896763203	0.421389689	0.83324	0.29177	0.432197
Threshold 7	0.91731266	0.25199645	0.648119366	0.370008872	0.733688415	0.293345164	0.90909091	0.209405501	0.875979557	0.456255545	0.81684	0.3162	0.455913
Threshold 8	0.88501292	0.27822908	0.596518495	0.389723801	0.682645362	0.312347684	0.88520801	0.233346872	0.860306644	0.512794475	0.78194	0.34529	0.479037
Threshold 9	0.87661499	0.29732691	0.571961455	0.403155126	0.666222814	0.328878176	0.87288136	0.248247151	0.842930153	0.54206836	0.76612	0.36394	0.493455
Threshold 10	0.85271318	0.32163743	0.540565744	0.423732942	0.641367066	0.352095516	0.85208012	0.269493177	0.820783646	0.586988303	0.7415	0.39079	0.511827
Threshold 11	0.84302326	0.35548897	0.505750699	0.443203486	0.620949845	0.381095068	0.84283513	0.29801144	0.798637138	0.638518113	0.72224	0.42326	0.53373
Threshold 12	0.82105943	0.37703945	0.483369599	0.46128745	0.605858855	0.404924354	0.8220339	0.316523286	0.77955707	0.678730345	0.70238	0.4477	0.546836
Threshold 13	0.80103359	0.40142441	0.451041343	0.469731303	0.593874834	0.433149885	0.80585516	0.338620912	0.760817717	0.722887664	0.68252	0.47316	0.558875
Threshold 14	0.7874677	0.42223762	0.434566366	0.484239694	0.578339991	0.451333563	0.78659476	0.353654311	0.735945486	0.748181501	0.66458	0.49193	0.565364
Threshold 15	0.75064599	0.43948563	0.406279142	0.494326776	0.548158012	0.467095308	0.75885978	0.372541602	0.698126065	0.774962176	0.63241	0.50968	0.564449
Threshold 16	0.72222222	0.45080645	0.390425863	0.506451611	0.524189969	0.476209675	0.72265023	0.378225805	0.663373083	0.785080642	0.60457	0.51935	0.558728
Threshold 17	0.68475452	0.46107003	0.372085794	0.520661155	0.494451842	0.484558502	0.68181818	0.384949977	0.624190801	0.7968682	0.57146	0.52962	0.549741
Threshold 18	0.64728682	0.46453407	0.353124029	0.526657392	0.468264536	0.489105236	0.65177196	0.392211403	0.594889267	0.809457576	0.54307	0.53639	0.539705
Threshold 19	0.62273902	0.482	0.336959901	0.541999997	0.443408788	0.499499998	0.62788906	0.407499998	0.563543441	0.826999996	0.51891	0.5516	0.53475
Threshold 20	0.60400517	0.49340369	0.322350016	0.547229549	0.428761651	0.50976253	0.61633282	0.422163586	0.541396934	0.838522423	0.50257	0.56222	0.530717
Threshold 21	0.55943152	0.50762016	0.294684489	0.555685812	0.395916556	0.522860489	0.57858243	0.440211017	0.493696763	0.849355212	0.46446	0.57515	0.513907
Threshold 22	0.53423773	0.52675159	0.27634442	0.566242035	0.377718597	0.542038213	0.55007704	0.454777067	0.463713799	0.866878975	0.44042	0.59134	0.504835
Threshold 23	0.5122739	0.53544902	0.262356233	0.569885209	0.359076787	0.546252528	0.52850539	0.463200537	0.442589438	0.877110055	0.42096	0.59838	0.494225
Threshold 24	0.48126615	0.55721765	0.243394467	0.585639487	0.335552597	0.565445022	0.50077042	0.486163048	0.407836457	0.895287951	0.39376	0.61795	0.481014
Threshold 25	0.46576227	0.57313195	0.235001554	0.60095389	0.324012428	0.580286164	0.48459168	0.499999996	0.388415673	0.906200311	0.37956	0.63211	0.474306
Threshold 26	0.4373385	0.58869565	0.219769972	0.614782603	0.302707501	0.593043473	0.45377504	0.512173909	0.359795571	0.918260862	0.35468	0.64539	0.457775
Threshold 27	0.42183463	0.59041591	0.212620454	0.618444841	0.293386596	0.597649181	0.43836672	0.514466541	0.347870528	0.923146465	0.34282	0.64882	0.4486
Threshold 28	0.39470284	0.60078662	0.199253963	0.630285146	0.273413227	0.605703042	0.41217257	0.526057025	0.320613288	0.925270394	0.32003	0.65762	0.430536
Threshold 29	0.37855297	0.60040983	0.189928505	0.626024584	0.260985353	0.60245901	0.39522342	0.525614749	0.306984668	0.923155728	0.30633	0.65553	0.417543
Threshold 30	0.3624031	0.61311475	0.180913895	0.636065567	0.249445184	0.614207644	0.37827427	0.536612016	0.29165247	0.935519115	0.29254	0.6671	0.406717
Threshold 31	0.34043928	0.62514827	0.169101647	0.645314346	0.234354194	0.626334512	0.35362096	0.544483979	0.268824532	0.935943049	0.27327	0.67544	0.389107
Threshold 32	0.32170543	0.6295828	0.159154492	0.647281913	0.222370173	0.633375466	0.33744222	0.553729449	0.252810903	0.938053085	0.2587	0.6804	0.374862
Threshold 33	0.29651163	0.64466291	0.146720547	0.662921339	0.207279183	0.655898867	0.31432974	0.5730337	0.227597956	0.938202234	0.23849	0.69494	0.355107
Threshold 34	0.28359173	0.64749262	0.139881878	0.663716804	0.197514425	0.656342173	0.30046225	0.57522123	0.217376491	0.941002936	0.22777	0.69676	0.343302
Threshold 35	0.25839793	0.65789473	0.127447933	0.674342094	0.18153573	0.672697357	0.27503852	0.587171043	0.195911414	0.945723669	0.20767	0.70757	0.32109
Threshold 36	0.25129199	0.68849556	0.123406901	0.702654855	0.176209498	0.702654855	0.26733436	0.614159281	0.183986371	0.955752195	0.20045	0.73274	0.314778
Threshold 37	0.23578811	0.7005758	0.116257383	0.717850274	0.165113182	0.714011503	0.25192604	0.627639143	0.170017036	0.957773494	0.18782	0.74357	0.299887
Threshold 38	0.22739018	0.71983639	0.111283805	0.732106324	0.158899245	0.732106324	0.24268105	0.644171766	0.161499148	0.969325134	0.18035	0.75951	0.291483
Threshold 39	0.2125323	0.74099097	0.104134287	0.754504488	0.148246782	0.752252235	0.23035439	0.673423408	0.148551959	0.98198196	0.16876	0.78063	0.277526
Threshold 40	0.20219638	0.74346792	0.09884986	0.7553444	0.140701287	0.752969103	0.2211094	0.681710198	0.141737649	0.988123492	0.16092	0.78432	0.267045
Threshold 41	0.19379845	0.7614213	0.094187131	0.769035513	0.13448735	0.769035513	0.21494607	0.708121809	0.133219761	0.992385762	0.15413	0.8	0.258458
Threshold 42	0.18410853	0.77027025	0.088591856	0.770270249	0.127385708	0.775675655	0.20338983	0.713513494	0.126064736	0.999999973	0.14591	0.80595	0.247082
Threshold 43	0.17377261	0.77298848	0.083618278	0.772988484	0.119840213	0.775862047	0.19414484	0.72413791	0.118568995	0.999999971	0.13799	0.8092	0.23577
Threshold 44	0.16602067	0.7859327	0.079888094	0.785932698	0.114070129	0.785932698	0.18644068	0.740061139	0.111413969	0.999999969	0.13157	0.81957	0.226733
Threshold 45	0.15826873	0.79804558	0.076157911	0.798045577	0.108743897	0.798045577	0.18027735	0.762214959	0.104599659	0.999999967	0.12561	0.83127	0.218239
Threshold 46	0.1505168	0.82332153	0.072427728	0.823321526	0.103417665	0.823321526	0.17180277	0.787985838	0.096422487	0.999999965	0.11892	0.85159	0.208691
Threshold 47	0.14405685	0.82899625	0.069319242	0.828996252	0.098979139	0.828996252	0.16409861	0.791821532	0.09165247	0.999999963	0.11362	0.85576	0.200605
Threshold 48	0.13501292	0.84959346	0.064967361	0.849593461	0.092321349	0.845528421	0.1540832	0.813008097	0.083816014	0.999999959	0.10604	0.87154	0.189074
Threshold 49	0.124677	0.85777774	0.059993783	0.85777774	0.08566356	0.85777774	0.1440678	0.831111074	0.076660988	0.999999956	0.09821	0.88089	0.17672
Threshold 50	0.11692506	0.88725486	0.0562636	0.887254858	0.080337328	0.887254858	0.13636364	0.867647016	0.069505963	0.999999951	0.09188	0.90588	0.166835
Threshold 51	0.11046512	0.90476186	0.053155113	0.904761857	0.075898802	0.904761857	0.13020031	0.894179847	0.06439523	0.999999947	0.08682	0.92169	0.158695
Threshold 52	0.10142119	0.92899403	0.048803233	0.928994028	0.069684865	0.928994028	0.1201849	0.923076868	0.05758092	0.999999941	0.07954	0.94201	0.146684
Threshold 53	0.09496124	0.93630567	0.045694747	0.936305673	0.065246338	0.936305673	0.11325116	0.936305673	0.053492334	0.999999936	0.07453	0.94904	0.138204
Threshold 54	0.09237726	0.94078941	0.044451352	0.940789412	0.063470928	0.940789412	0.11016949	0.940789412	0.051788756	0.999999934	0.07245	0.95263	0.13466
Threshold 55	0.08527132	0.94964022	0.041032017	0.949640219	0.058588549	0.949640219	0.10169492	0.949640219	0.047359455	0.999999928	0.06679	0.95971	0.124886
Threshold 56	0.07751938	0.94488182	0.037301834	0.944881815	0.053262317	0.944881815	0.09244992	0.944881815	0.043270869	0.999999921	0.06076	0.95591	0.114258
Threshold 57	0.07105943	0.94017086	0.034193348	0.94017086	0.048823791	0.94017086	0.08474576	0.94017086	0.039863714	0.999999915	0.05574	0.95214	0.105309
Threshold 58	0.0620155	0.95049496	0.029841467	0.950494955	0.042609854	0.950494955	0.07395994	0.950494955	0.034412266	0.999999901	0.04857	0.9604	0.092459
Threshold 59	0.05490956	0.94444434	0.026422132	0.94444434	0.037727474	0.94444434	0.06548536	0.94444434	0.030664395	0.999999889	0.04304	0.95556	0.082372
Threshold 60	0.04909561	0.93827149	0.023624495	0.938271489	0.033732801	0.938271489	0.05855162	0.938271489	0.027597956	0.999999877	0.03852	0.95062	0.07404
Threshold 61	0.04005168	0.95384601	0.019272614	0.953846007	0.027518864	0.953846007	0.04776579	0.953846007	0.022146508	0.999999846	0.03135	0.96308	0.060725
Threshold 62	0.03165375	0.9999998	0.015231582	0.999999796	0.021748779	0.999999796	0.03775039	0.999999796	0.01669506	0.999999796	0.02462	1	0.048049
Threshold 63	0.02971576	0.99999978	0.014299036	0.999999783	0.020417221	0.999999783	0.03543914	0.999999783	0.015672913	0.999999783	0.02311	1	0.045173
Threshold 64	0.02131783	0.9999997	0.010258004	0.999999697	0.014647137	0.999999697	0.02542373	0.999999697	0.011243612	0.999999697	0.01658	1	0.032615
Threshold 65	0.01744186	0.99999963	0.008392913	0.99999963	0.011984021	0.99999963	0.02080123	0.99999963	0.009199319	0.99999963	0.01356	1	0.026764
Threshold 66	0.0122739	0.99999947	0.005906124	0.999999474	0.0084332	0.999999474	0.0146379	0.999999474	0.006473595	0.999999474	0.00954	1	0.018909
Threshold 67	0.00839793	0.99999923	0.004041032	0.999999231	0.005770084	0.999999231	0.01001541	0.999999231	0.004429302	0.999999231	0.00653	1	0.012977
Threshold 68	0.00581395	0.99999889	0.002797638	0.999998889	0.003994674	0.999998889	0.00693374	0.999998889	0.00306644	0.999998889	0.00452	1	0.009002
Threshold 69	0.00387597	0.99999833	0.001865092	0.999998333	0.002663116	0.999998333	0.0046225	0.999998333	0.002044293	0.999998333	0.00301	1	0.00601
Threshold 70	0.00258398	0.9999975	0.001243394	0.9999975	0.001775411	0.9999975	0.00308166	0.9999975	0.001362862	0.9999975	0.00201	1	0.004011
Threshold 71	0.00193798	0.99999667	0.000932546	0.999996667	0.001331558	0.999996667	0.00231125	0.999996667	0.001022147	0.999996667	0.00151	1	0.00301
Threshold 72	0.00064599	0.99999	0.000310849	0.99999	0.000443853	0.99999	0.00077042	0.99999	0.000340716	0.99999	0.0005	0.99999	0.001004
Threshold 73	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 74	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 75	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 76	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 77	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 78	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 79	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 80	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 81	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 82	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 83	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 84	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 85	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 86	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 87	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 88	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 89	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 90	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 91	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 92	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 93	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 94	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 95	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 96	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 97	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 98	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 99	0	0	0	0	0	0	0	0	0	0	0	0	0
Means	0.258287	0.48900619	0.155182537	0.511131529	0.191488789	0.496844227	0.26636161	0.459840661	0.223543785	0.625784809			
Mean F scores	0.33802597		0.238078411		0.276432236		0.33732258		0.329409981				0.3039
Best F scores													0.565364

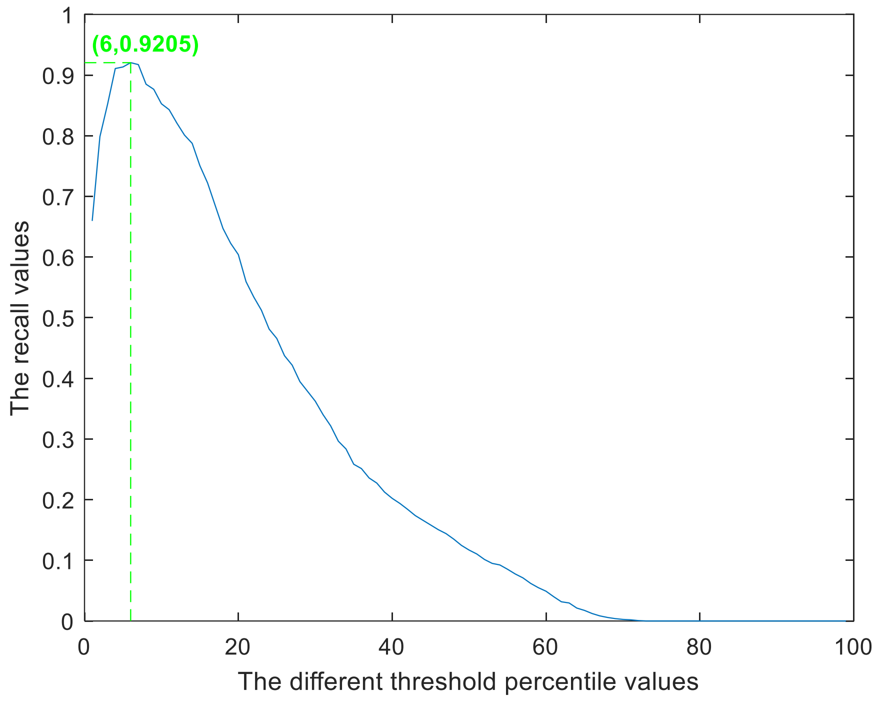


Figure. 1.61 The recall values under different threshold values
for the “Dogs” ’s Structured-Edge result with the 1st ground truth image

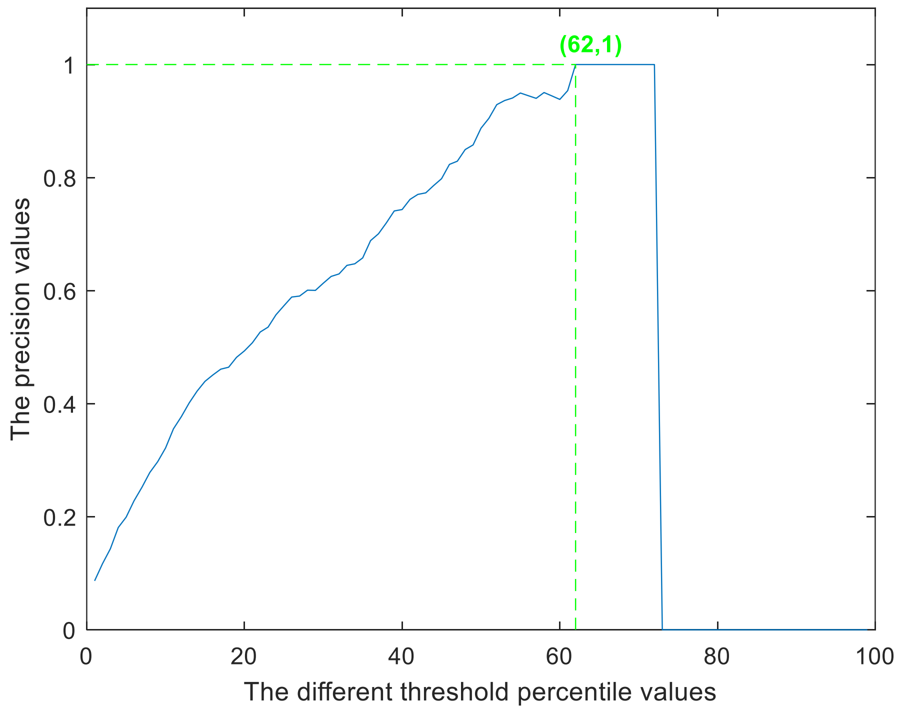


Figure. 1.62 The precision values under different threshold values
for the “Dogs” ’s Structured-Edge result with the 1st ground truth image

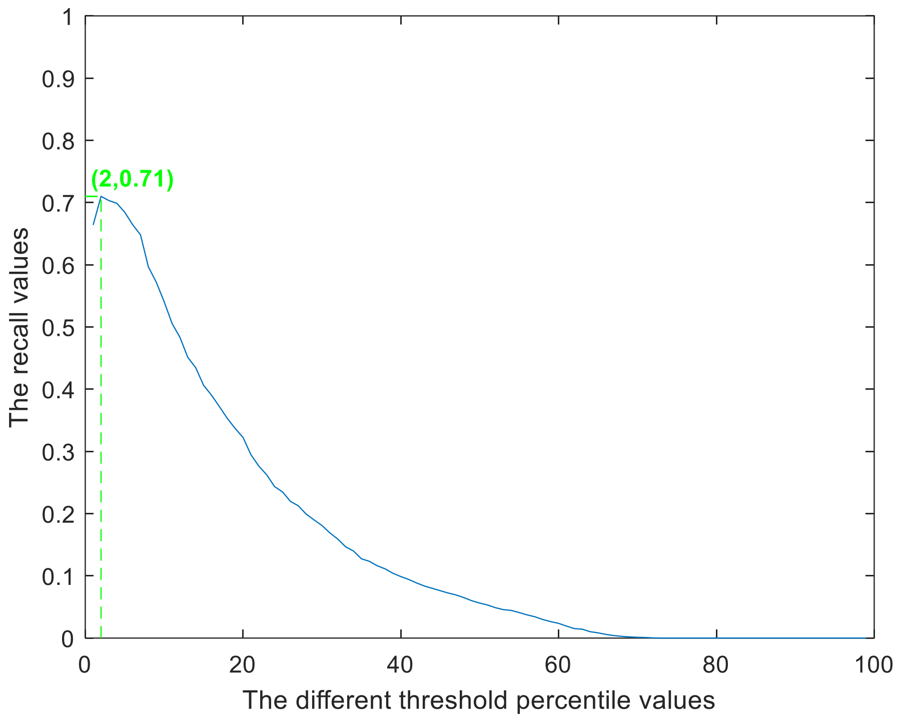


Figure. 1.63 The recall values under different threshold values
for the “Dogs” ’s Structured-Edge result with the 2nd ground truth image

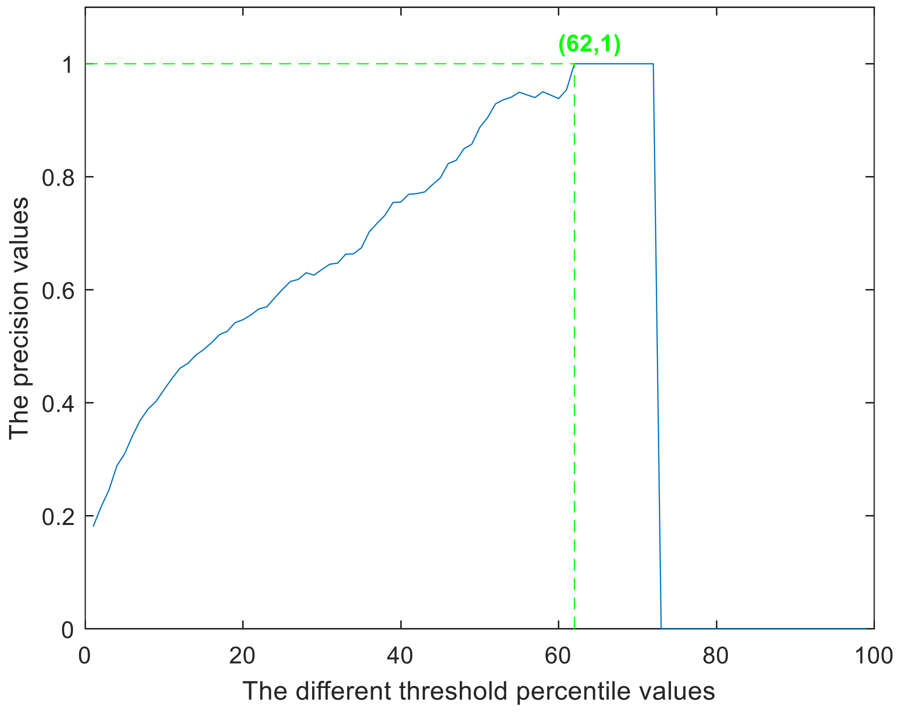


Figure. 1.64 The precision values under different threshold values
for the “Dogs” ’s Structured-Edge result with the 2nd ground truth image

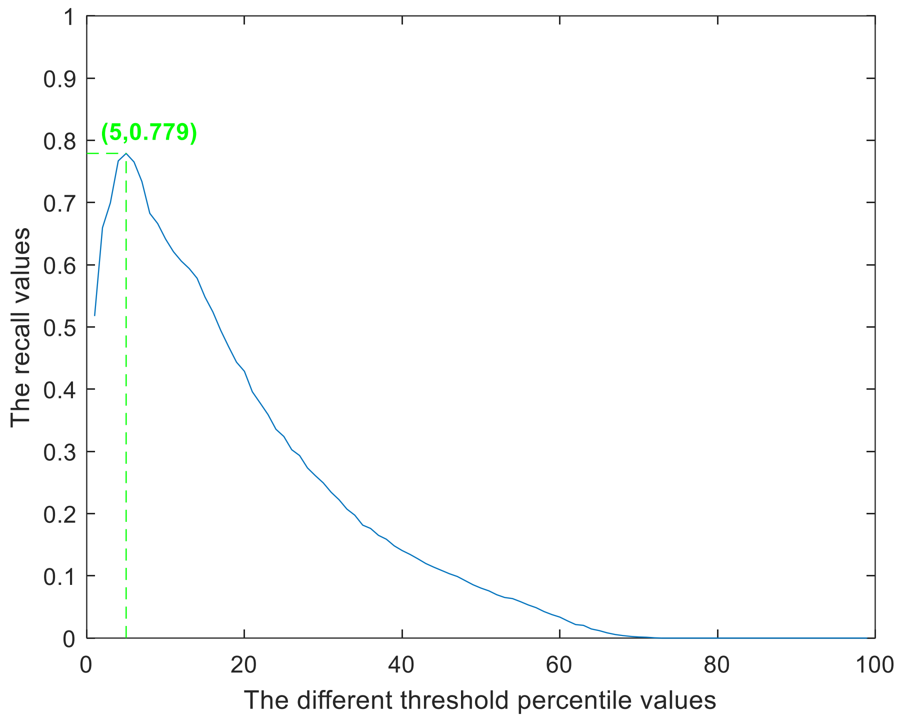


Figure. 1.65 The recall values under different threshold values
for the “Dogs” ’s Structured-Edge result with the 3rd ground truth image

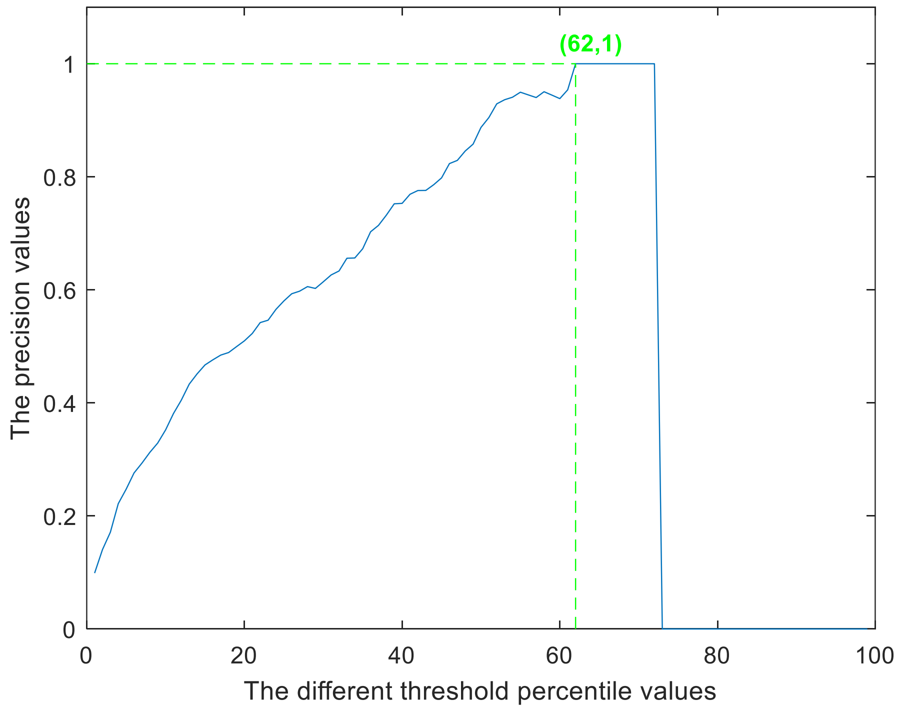


Figure. 1.66 The precision values under different threshold values
for the “Dogs” ’s Structured-Edge result with the 3rd ground truth image

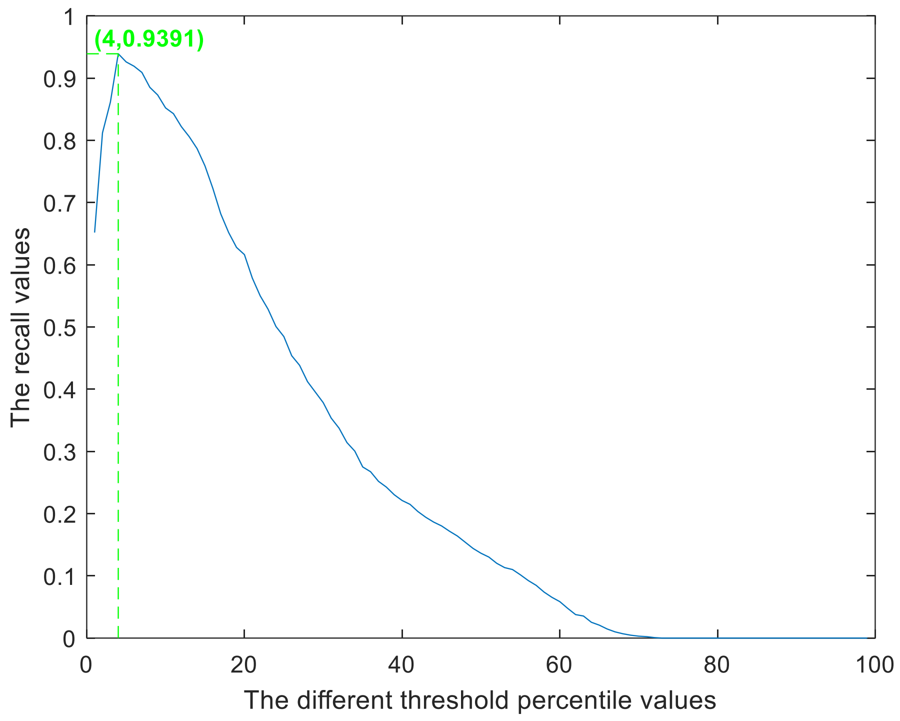


Figure. 1.67 The recall values under different threshold values
for the “Dogs” ’s Structured-Edge result with the 4th ground truth image

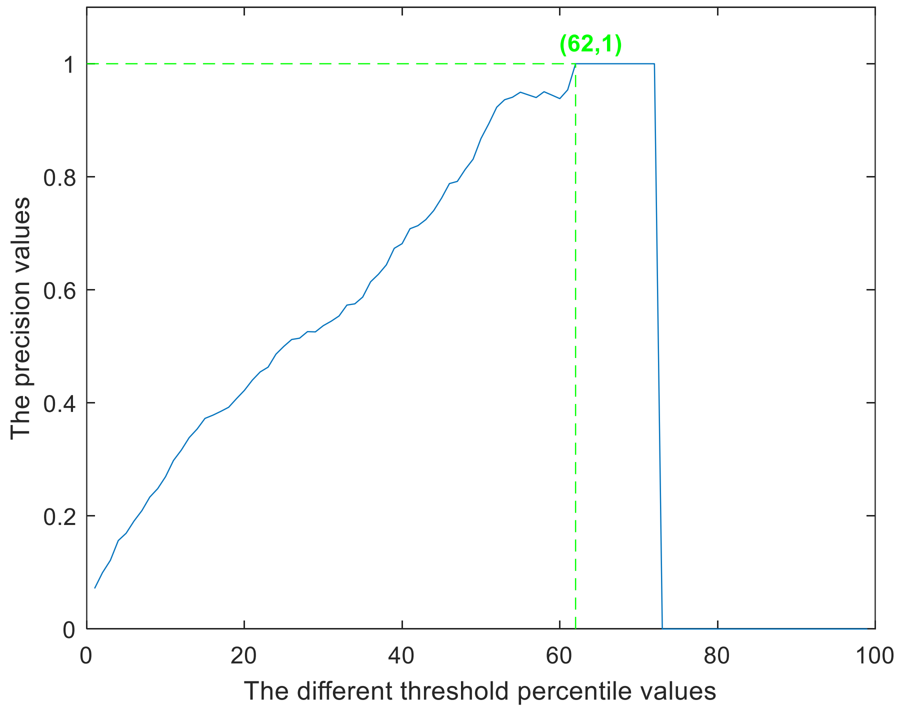


Figure. 1.68 The precision values under different threshold values
for the “Dogs” ’s Structured-Edge result with the 4th ground truth image

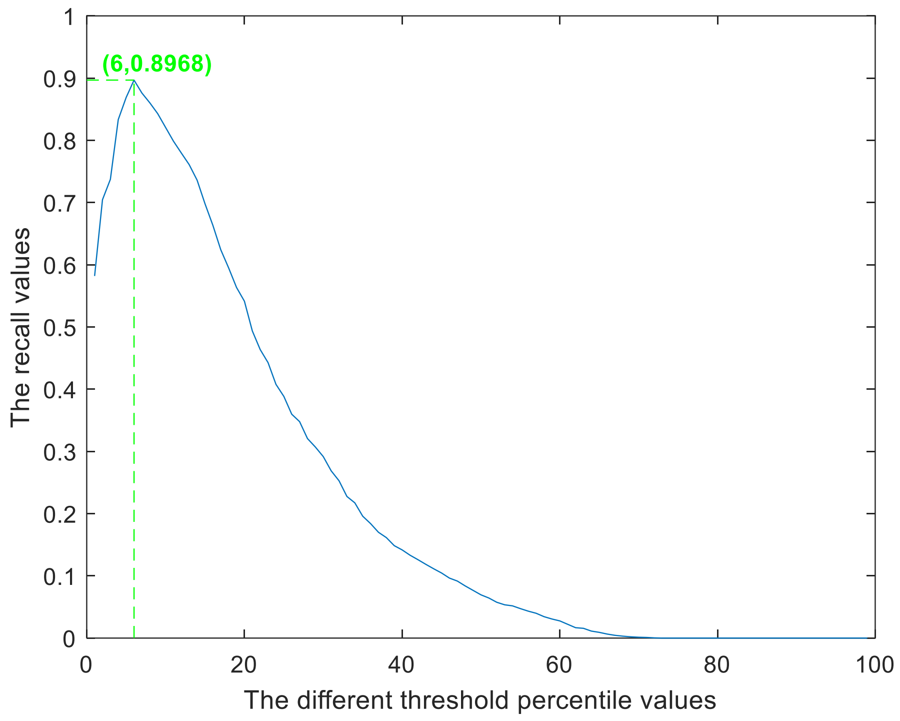


Figure. 1.69 The recall values under different threshold values for the “Dogs” ’s Structured-Edge result with the 5th ground truth image

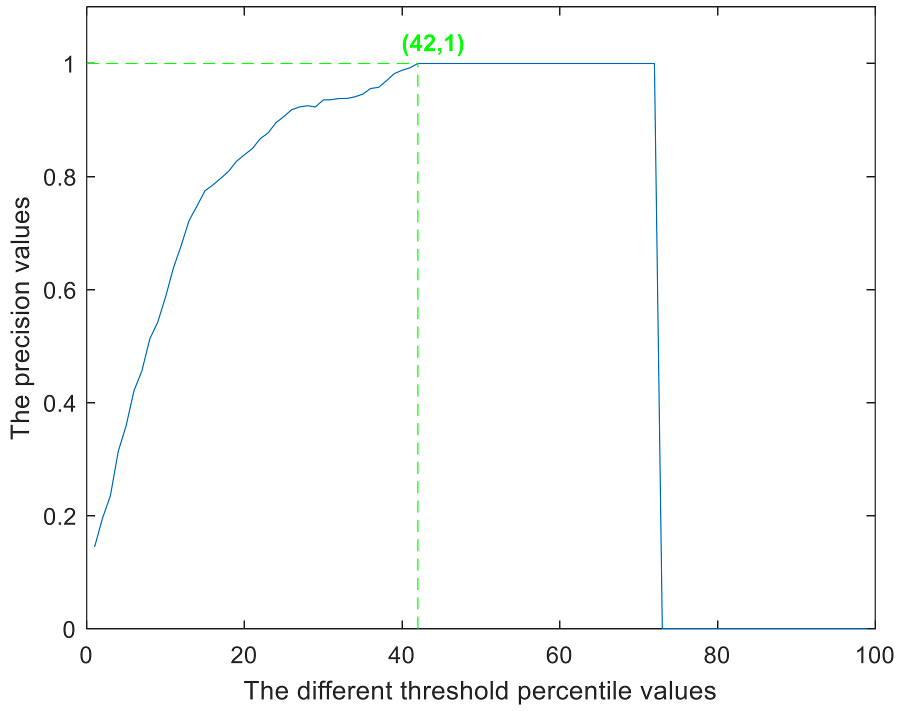


Figure. 1.70 The precision values under different threshold values for the “Dogs” ’s Structured-Edge result with the 5th ground truth image

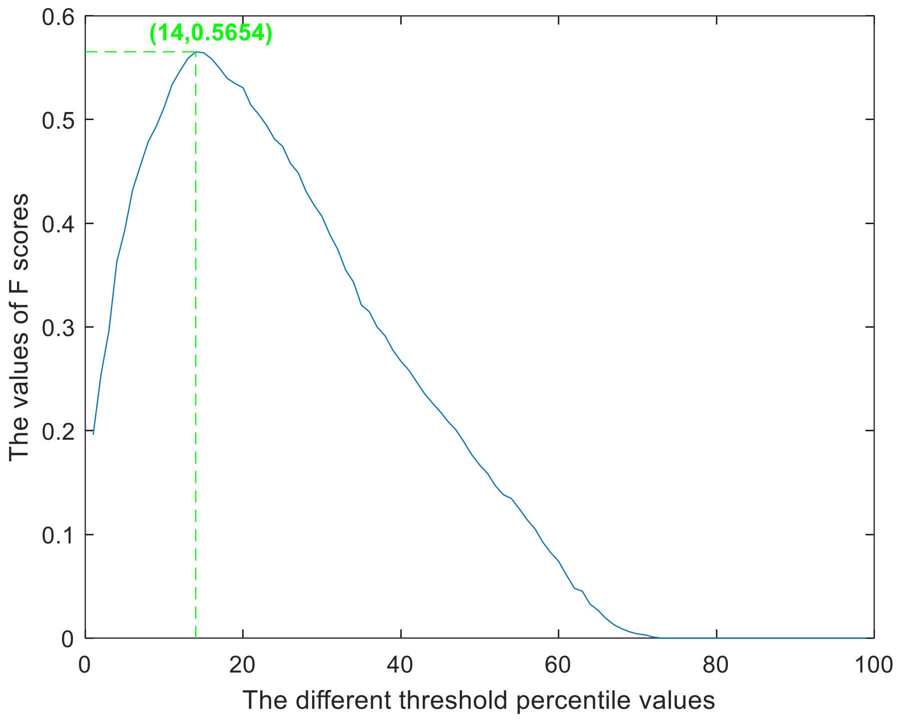


Figure. 1.71 The F values under different threshold values
for the “Dogs” ’s Structured-Edge result among 5 ground truth images

Table. 1.6 The performance evaluation for the structured edge map of “ Gallery”

	The 1st GT_R	The 1st GT_P	The 2nd GT_R	The 2nd GT_P	The 3rd GT_R	The 3rd GT_P	The4th GT_R	The 4th GT_P	The 5th GT_R	The 5th GT_P	Means_R	Means_P	F scores
Threshold 1	0.4603638	0.318322981	0.451685959	0.278726708	0.468447176	0.307763975	0.46510443	0.283540372	0.480239521	0.435869565	0.465168	0.465234	0.382539
Threshold 2	0.562991242	0.408972267	0.560895823	0.363621533	0.566769085	0.391190864	0.56800815	0.363784665	0.545423439	0.520065252	0.560818	0.520644	0.473373
Threshold 3	0.65528857	0.478753076	0.631605435	0.411812961	0.652564406	0.452994257	0.63652573	0.410008203	0.597091531	0.572600491	0.634615	0.562824	0.536877
Threshold 4	0.73950146	0.535273081	0.71716155	0.463263978	0.745686599	0.512841352	0.71523179	0.45643693	0.668776732	0.63540312	0.717272	0.598751	0.603337
Threshold 5	0.780148215	0.570443349	0.76673377	0.500328406	0.787993382	0.547454843	0.79597555	0.513136288	0.711377246	0.68275862	0.768446	0.613689	0.64975
Threshold 6	0.850437907	0.606308036	0.840966281	0.535062439	0.86338927	0.584854306	0.8555782	0.537784181	0.779811805	0.729747037	0.838037	0.623774	0.698464
Threshold 7	0.890410959	0.618178983	0.882989431	0.547084502	0.901914441	0.594948549	0.90601121	0.554568131	0.827031651	0.753663859	0.881672	0.631071	0.723663
Threshold 8	0.91084662	0.623712132	0.898087569	0.548823619	0.926494918	0.602798707	0.93683138	0.565585114	0.865526091	0.777948638	0.907557	0.637699	0.739365
Threshold 9	0.914664271	0.629618178	0.904630096	0.555727314	0.930512881	0.608594836	0.94396332	0.572886071	0.872711719	0.788529911	0.913296	0.645291	0.746391
Threshold 10	0.907028969	0.631291027	0.903371917	0.561112847	0.92696762	0.613004063	0.95160469	0.583932478	0.874764756	0.799155985	0.912748	0.65863	0.750822
Threshold 11	0.914888839	0.638457921	0.915702063	0.570286788	0.93027653	0.616831217	0.96102904	0.591286631	0.883832335	0.809590972	0.921146	0.663262	0.758924
Threshold 12	0.913316865	0.651136726	0.918721691	0.584534101	0.929567478	0.629682996	0.95695364	0.601504962	0.882976903	0.826288824	0.920307	0.670211	0.76778
Threshold 13	0.904558724	0.655065863	0.917715148	0.593104569	0.915622784	0.630021141	0.94625573	0.604163278	0.877331052	0.83395674	0.912297	0.677316	0.768093
Threshold 14	0.900516506	0.662153235	0.91419225	0.599900924	0.910895769	0.636393658	0.94370861	0.611789959	0.871171942	0.840819021	0.908097	0.682844	0.771222
Threshold 15	0.891982933	0.66789978	0.914947157	0.611400705	0.897423777	0.638473179	0.93785023	0.619135698	0.864499572	0.849672102	0.901341	0.691481	0.773428
Threshold 16	0.885245902	0.672237379	0.914443885	0.619713505	0.892696762	0.64409959	0.93224656	0.624147339	0.856800684	0.854024555	0.896287	0.697185	0.775136
Threshold 17	0.86772962	0.678728262	0.903120282	0.630423326	0.877806665	0.652380115	0.91696383	0.632355523	0.841060736	0.863516598	0.881336	0.706272	0.774945
Threshold 18	0.858073209	0.685135376	0.897081027	0.639232561	0.867407232	0.658059888	0.90524707	0.637260175	0.826518392	0.866236326	0.870865	0.70766	0.774402
Threshold 19	0.836290141	0.693095104	0.878711626	0.649916247	0.848971874	0.668527823	0.88537952	0.646938394	0.80239521	0.872882932	0.85035	0.71531	0.771638
Threshold 20	0.822816079	0.694728857	0.865374937	0.652066741	0.83479083	0.669700416	0.87162506	0.648843381	0.787681779	0.872961697	0.836458	0.719315	0.766682
Threshold 21	0.79429598	0.700534758	0.84046301	0.661517131	0.804301584	0.673994849	0.85048395	0.661319072	0.759452524	0.879183995	0.809799	0.725746	0.759623
Threshold 22	0.765326746	0.70326042	0.815299446	0.668592652	0.771685181	0.673751546	0.82832399	0.671068921	0.729512404	0.879900947	0.78203	0.731964	0.749357
Threshold 23	0.747810465	0.709718669	0.799446402	0.677109973	0.752540771	0.678601874	0.80871116	0.676683716	0.711719418	0.886615514	0.764046	0.734813	0.744398
Threshold 24	0.721760611	0.717410713	0.771766482	0.684598213	0.722760577	0.682589284	0.7804381	0.68392857	0.68314799	0.891294641	0.735975	0.743164	0.733959
Threshold 25	0.707388278	0.720988783	0.75691998	0.688487066	0.706688726	0.684367131	0.76668365	0.688944837	0.666210436	0.891279467	0.720778	0.752997	0.727723
Threshold 26	0.68852459	0.732791585	0.735530951	0.698613765	0.684235405	0.691921604	0.74172185	0.695984702	0.641745081	0.896510514	0.698352	0.762287	0.720056
Threshold 27	0.67190658	0.744093507	0.716406643	0.708032826	0.669581659	0.704551105	0.71854305	0.701566773	0.623781009	0.906739615	0.680044	0.769057	0.714658
Threshold 28	0.654390299	0.753750645	0.696779064	0.716244178	0.653509809	0.715209517	0.69943963	0.710294877	0.605816938	0.915933779	0.661987	0.777324	0.7086
Threshold 29	0.63664945	0.761482673	0.680422748	0.726295996	0.634365398	0.720923984	0.67829852	0.715283372	0.586826347	0.921300024	0.643312	0.787813	0.700582
Threshold 30	0.611273299	0.771541948	0.652994464	0.735544216	0.60954857	0.731009068	0.64875191	0.721938773	0.559281437	0.926587299	0.61637	0.792269	0.687548
Threshold 31	0.584100606	0.784615382	0.624559638	0.748717946	0.582604585	0.743589741	0.61589404	0.729411763	0.528999145	0.932730012	0.587232	0.797581	0.672888
Threshold 32	0.568605435	0.789276806	0.609713135	0.75529925	0.566296384	0.746882791	0.59984717	0.734102242	0.513601369	0.935785533	0.571613	0.801607	0.664087
Threshold 33	0.548394341	0.798300095	0.587820835	0.763648249	0.545497518	0.754494931	0.57463067	0.737495911	0.488793841	0.933965345	0.549027	0.809481	0.650361
Threshold 34	0.538288794	0.802477399	0.577503775	0.768329425	0.536043489	0.759290255	0.56444218	0.741881484	0.478357571	0.936056241	0.538927	0.812071	0.644526
Threshold 35	0.526835841	0.811764703	0.566683442	0.779238752	0.525171354	0.768858129	0.5501783	0.747404842	0.464841745	0.940138405	0.526742	0.818787	0.638194
Threshold 36	0.51223894	0.814642854	0.55183694	0.783214283	0.509572205	0.769999997	0.53616913	0.751785712	0.450641574	0.940714282	0.512092	0.823855	0.628098
Threshold 37	0.497866607	0.819896447	0.538751887	0.791789938	0.49846372	0.779955618	0.52190525	0.757766269	0.436954662	0.944526624	0.498788	0.827083	0.619924
Threshold 38	0.490231305	0.826580837	0.531202818	0.799318437	0.489246041	0.783794014	0.51375446	0.763725859	0.427373824	0.94585384	0.490362	0.829805	0.61479
Threshold 39	0.472265888	0.830240818	0.514343231	0.806948279	0.470574332	0.786024474	0.49388691	0.765495457	0.410265184	0.94670351	0.472267	0.836724	0.601225
Threshold 40	0.459690097	0.833469052	0.503271263	0.814332244	0.45710234	0.78745928	0.48038716	0.767915306	0.397433704	0.945846902	0.459577	0.842806	0.591533
Threshold 41	0.438580732	0.841810341	0.482385506	0.8262931	0.435830773	0.794827583	0.45720835	0.773706893	0.375876818	0.946982755	0.437976	0.847776	0.574978
Threshold 42	0.423534696	0.848021579	0.46829391	0.836780572	0.419995273	0.799010788	0.44268976	0.781474817	0.360992301	0.948741003	0.423101	0.85229	0.563374
Threshold 43	0.417920503	0.853278309	0.463764469	0.845025214	0.413850154	0.802842729	0.43683138	0.786336539	0.355004277	0.951398437	0.417474	0.856552	0.55945
Threshold 44	0.403323602	0.856870225	0.450931052	0.854961828	0.398251004	0.80391221	0.42180336	0.790076332	0.342686056	0.955629766	0.403399	0.865628	0.547604
Threshold 45	0.393218055	0.862561572	0.44011072	0.86157635	0.38737887	0.807389159	0.41034131	0.793596055	0.332591959	0.957635463	0.392728	0.870506	0.538533
Threshold 46	0.377498316	0.87279335	0.424760946	0.876427825	0.37130702	0.815680162	0.39378502	0.802699892	0.316509837	0.960539974	0.376772	0.877339	0.525019
Threshold 47	0.366719066	0.87890204	0.414695521	0.886975237	0.359489482	0.81862217	0.38155884	0.806243268	0.305731394	0.961786862	0.365639	0.882278	0.514969
Threshold 48	0.354143274	0.888951517	0.399345747	0.894588496	0.34601749	0.825253659	0.36805909	0.8145434	0.292386655	0.963359634	0.35199	0.889216	0.502408
Threshold 49	0.346283404	0.895990698	0.391041772	0.902963388	0.338217915	0.831493313	0.35812532	0.816966875	0.283832335	0.963974428	0.3435	0.899421	0.494478
Threshold 50	0.332360207	0.90686274	0.373427277	0.90931372	0.323091468	0.837622544	0.34284259	0.824754897	0.270145423	0.967524504	0.328373	0.90267	0.479624
Threshold 51	0.321805524	0.921543402	0.359838953	0.919614142	0.311746632	0.848231506	0.32985227	0.832797422	0.25936698	0.974919608	0.316522	0.910384	0.468253
Threshold 52	0.311250842	0.925233639	0.348263714	0.923898525	0.301347199	0.851134841	0.31940907	0.837116149	0.250128315	0.975967951	0.30608	0.916255	0.457145
Threshold 53	0.295081967	0.934566138	0.332159034	0.938833564	0.285511699	0.859174958	0.30183393	0.842816495	0.234901625	0.976529154	0.289898	0.921082	0.439757
Threshold 54	0.282281608	0.941573027	0.319073981	0.949812727	0.273221461	0.865917597	0.28782476	0.846441941	0.22326775	0.977528083	0.277134	0.924541	0.42555
Threshold 55	0.267909275	0.949085116	0.303724207	0.960222745	0.258804065	0.871121711	0.27228732	0.850437543	0.209580838	0.974542554	0.262461	0.929676	0.408513
Threshold 56	0.256680889	0.954090142	0.291645697	0.967445735	0.247459229	0.873956587	0.25980642	0.851419025	0.2	0.97579298	0.251118	0.932008	0.394957
Threshold 57	0.240736582	0.964028768	0.273779567	0.978417257	0.229969274	0.874999992	0.24223128	0.85521582	0.185628743	0.975719416	0.234469	0.933684	0.374487
Threshold 58	0.229957332	0.969696961	0.261701057	0.984848476	0.218151737	0.874053022	0.23051452	0.857007568	0.176047904	0.974431809	0.223275	0.934821	0.360244
Threshold 59	0.208174265	0.975789463	0.237040765	0.991578937	0.195462066	0.870526307	0.20682629	0.854736833	0.158597092	0.975789463	0.20122	0.933085	0.331084
Threshold 60	0.196945879	0.978794632	0.224458983	0.995535703	0.18411723	0.869419633	0.19485481	0.853794633	0.149700599	0.976562489	0.190016	0.927401	0.315831
Threshold 61	0.178082192	0.982651785	0.203069955	0.999999988	0.164263767	0.861214364	0.17345899	0.843866161	0.134987169	0.977695155	0.170772	0.923721	0.288703
Threshold 62	0.155625421	0.978813545	0.178158027	0.999999986	0.141810447	0.847457615	0.15002547	0.831920892	0.118562874	0.978813545	0.148836	0.921254	0.256504
Threshold 63	0.142375926	0.982945721	0.162304982	0.999999984	0.127629402	0.837209289	0.13474274	0.820155026	0.107955518	0.978294558	0.135002	0.916858	0.235572
Threshold 64	0.127105322	0.986062701	0.144438853	0.999999983	0.112030253	0.825783958	0.11844116	0.810104516	0.096663815	0.98432054	0.119736	0.911409	0.211925
Threshold 65	0.115652369	0.986590019	0.1313538	0.999999981	0.100212716	0.812260521	0.10621498	0.798850559	0.088109495	0.986590019	0.108309	0.90936	0.19373
Threshold 66	0.099708062	0.993288568	0.112481127	0.999999978	0.083668164	0.791946291	0.08863984	0.778523473	0.075962361	0.993288568	0.092092	0.901377	0.167279
Threshold 67	0.090949921	0.997536921	0.102164066	0.999999975	0.074923186	0.780788158	0.0794702	0.768472887	0.069461078	0.999999975	0.083394	0.893009	0.152775
Threshold 68	0.081518078	0.999999972	0.091343734	0.999999972	0.064996455	0.757575737	0.06928171	0.749311274	0.062104363	0.999999972	0.073849	0.879021	0.136512
Threshold 69	0.073882776	0.99999997	0.082788123	0.99999997	0.05719688	0.735562288	0.06113092	0.729483261	0.056287425	0.99999997	0.066257	0.865339	0.12336
Threshold 70	0.064226364	0.999999965	0.071967791	0.999999965	0.047270149	0.699300675	0.05068772	0.695804171	0.04893071	0.999999965	0.056617	0.860251	0.10638
Threshold 71	0.056366494	0.99999996	0.063160544	0.99999996	0.039470574	0.665338619	0.04228222	0.661354555	0.042942686	0.99999996	0.048845	0.850909	0.092469
Threshold 72	0.053671682	0.999999958	0.060140916	0.999999958	0.036870716	0.652719638	0.03948039	0.648535538	0.040889649	0.999999958	0.046211	0.849515	0.087709
Threshold 73	0.049180328	0.9954545	0.055359839	0.999999955	0.032852753	0.631818153	0.03515028	0.627272699	0.037639008	0.999999955	0.042036	0.838202	0.080114
Threshold 74	0.046260948	0.999999951	0.05183694	0.999999951	0.030489246	0.626213562	0.03260316	0.621359193	0.035243798	0.999999951	0.039287	0.834146	0.0751
Threshold 75	0.039973052	0.999999944	0.044791142	0.999999944	0.025053179	0.595505585	0.02699949	0.595505585	0.030453379	0.999999944	0.033454	0.824113	0.064339
Threshold 76	0.036829104	0.999999939	0.041268244	0.999999939	0.022689671	0.585365818	0.02445237	0.585365818	0.028058169	0.999999939	0.03066	0.827642	0.059144
Threshold 77	0.031664047	0.999999929	0.035480624	0.999999929	0.018671709	0.560283648	0.02012226	0.560283648	0.024123182	0.999999929	0.026012	0.825532	0.050432
Threshold 78	0.027621828	0.999999919	0.030951183	0.999999919	0.016544552	0.569105645	0.01782985	0.569105645	0.021043627	0.999999919	0.022798	0.837681	0.044374
Threshold 79	0.021109364	0.999999894	0.023653749	0.999999894	0.012526589	0.563829727	0.01349975	0.563829727	0.016082121	0.999999894	0.017374	0.819608	0.034032
Threshold 80	0.015495172	0.999999855	0.017362859	0.999999855	0.009690381	0.594202812	0.0104432	0.594202812	0.011804962	0.999999855	0.012959	0.851852	0.025523
Threshold 81	0.011452953	0.999999804	0.012833417	0.999999804	0.006617821	0.5490195	0.00713194	0.5490195	0.008725406	0.999999804	0.009352	0.942856	0.018493
Threshold 82	0.006063328	0.99999963	0.006794162	0.99999963	0.004017963	0.629629396	0.00433011	0.629629396	0.004619333	0.99999963	0.005165	0.999999	0.010268
Threshold 83	0.003143948	0.999999286	0.003522899	0.999999286	0.002836209	0.857142245	0.00305655	0.857142245	0.00239521	0.999999286	0.002991	0.999995	0.005963
Threshold 84	0.001571974	0.999998571	0.001761449	0.999998571	0.001654455	0.999998571	0.00178299	0.999998571	0.001197605	0.999998571	0.001594	0.99999	0.003182
Threshold 85	0.000449135	0.999995	0.000503271	0.999995	0.000472701	0.999995	0.00050942	0.999995	0.000342173	0.999995	0.000455	0	0.00091
Threshold 86	0.000224568	0.99999	0.000251636	0.99999	0.000236351	0.99999	0.00025471	0.99999	0.000171086	0.99999	0.000228	0	0.000455
Threshold 87	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 88	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 89	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 90	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 91	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 92	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 93	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 94	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 95	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 96	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 97	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 98	0	0	0	0	0	0	0	0	0	0	0	0	0
Threshold 99	0	0	0	0	0	0	0	0	0	0	0	0	0
Means	0.258287004	0.489006193	0.155182537	0.511131529	0.191488789	0.496844227	0.26636161	0.459840661	0.223543785	0.625784809			
Mean F scores	0.489969123		0.5031239		0.463664854		0.47235505		0.473211402				0.4805
Best F scores													0.775136

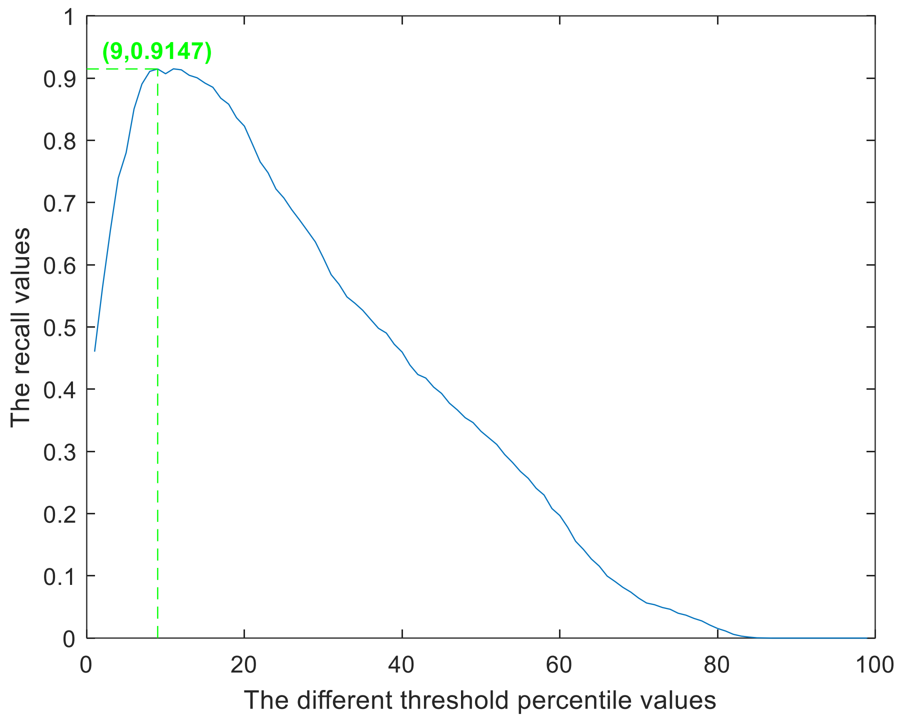


Figure. 1.72 The recall values under different threshold values
for the “Gallery” ’s Structured-Edge result with the 1st ground truth image

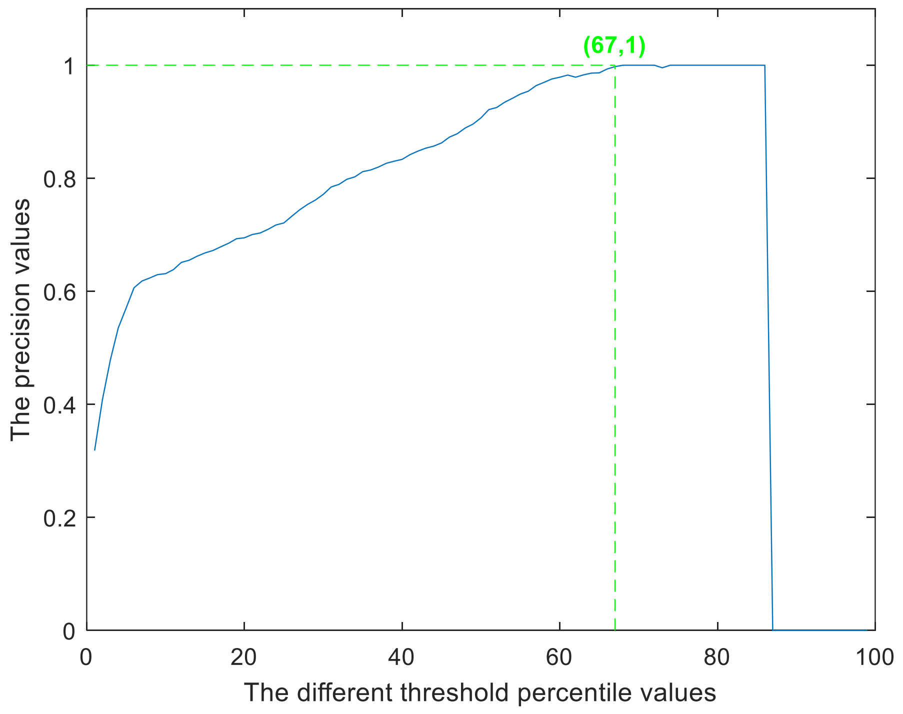


Figure. 1.73 The precision values under different threshold values
for the “Gallery” ’s Structured-Edge result with the 1st ground truth image

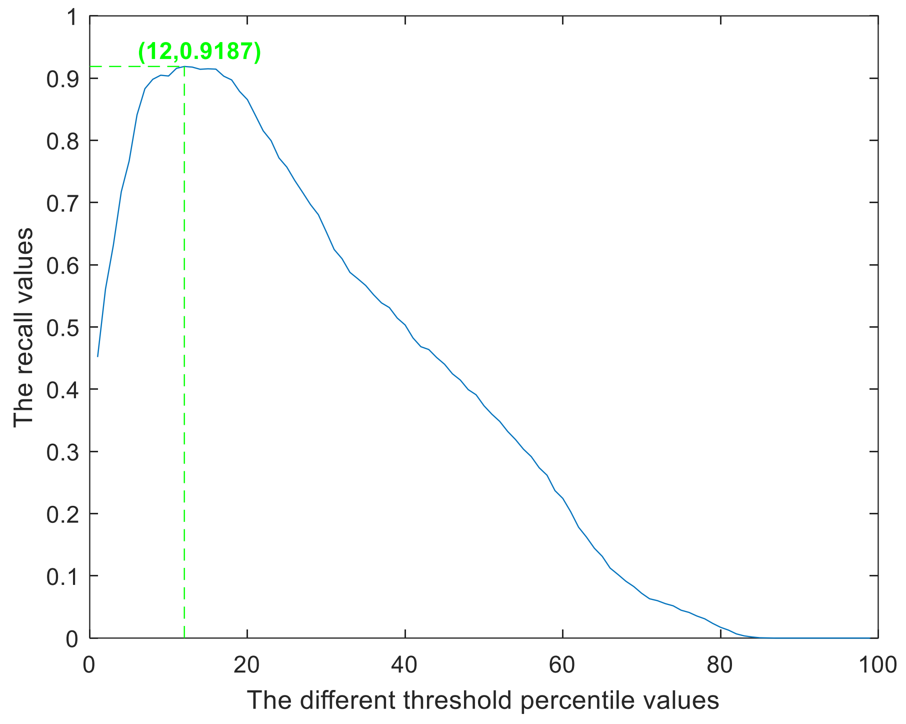


Figure. 1.74 The recall values under different threshold values
for the “Gallery” ’s Structured-Edge result with the 2nd ground truth image

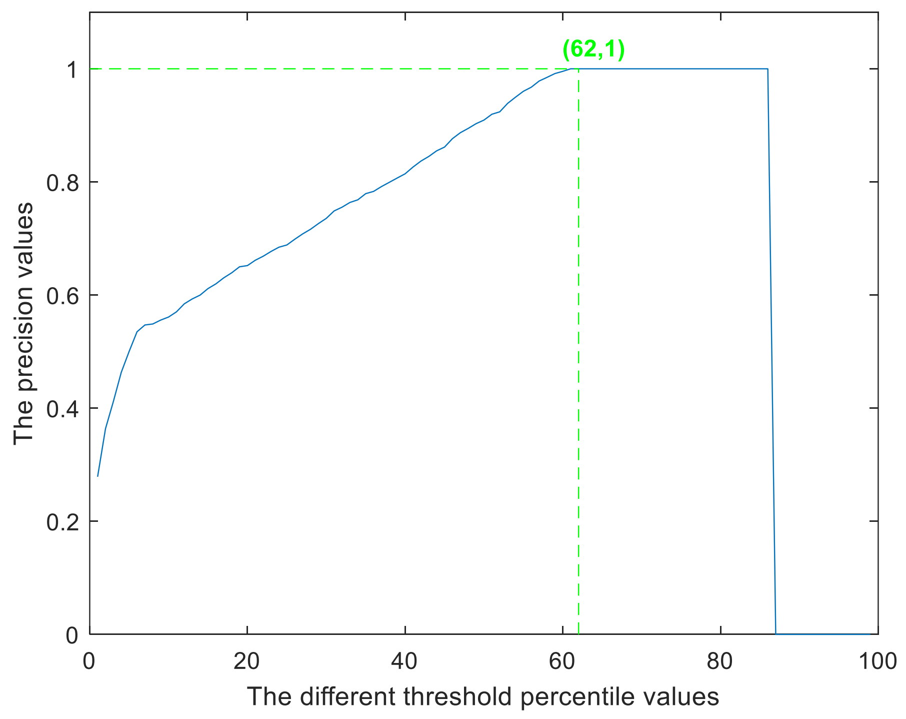


Figure. 1.75 The precision values under different threshold values
for the “Gallery” ’s Structured-Edge result with the 2nd ground truth image

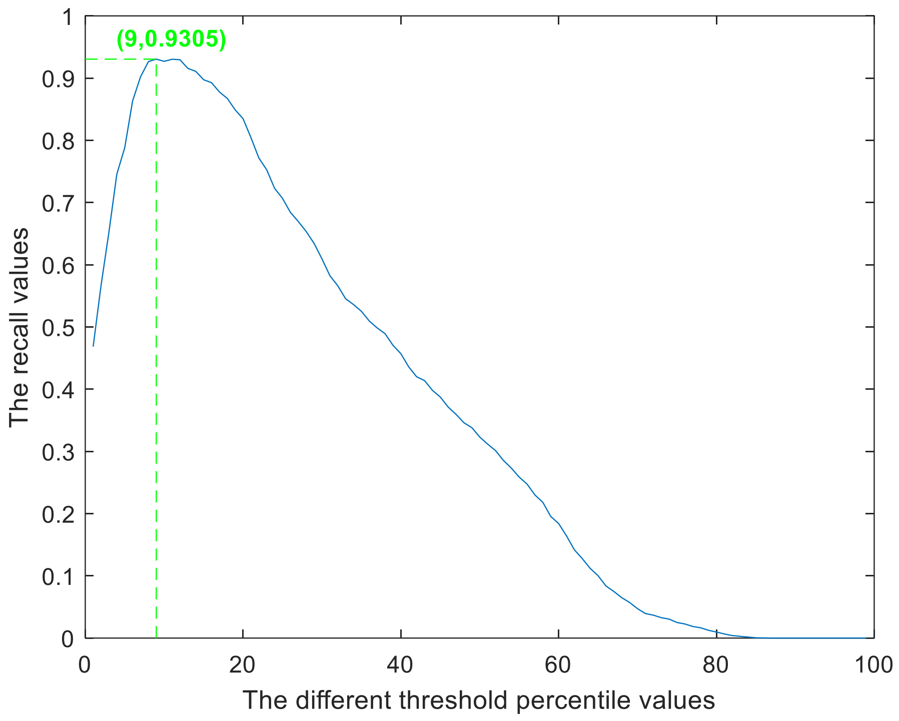


Figure. 1.76 The recall values under different threshold values
for the “Gallery” ’s Structured-Edge result with the 3rd ground truth image

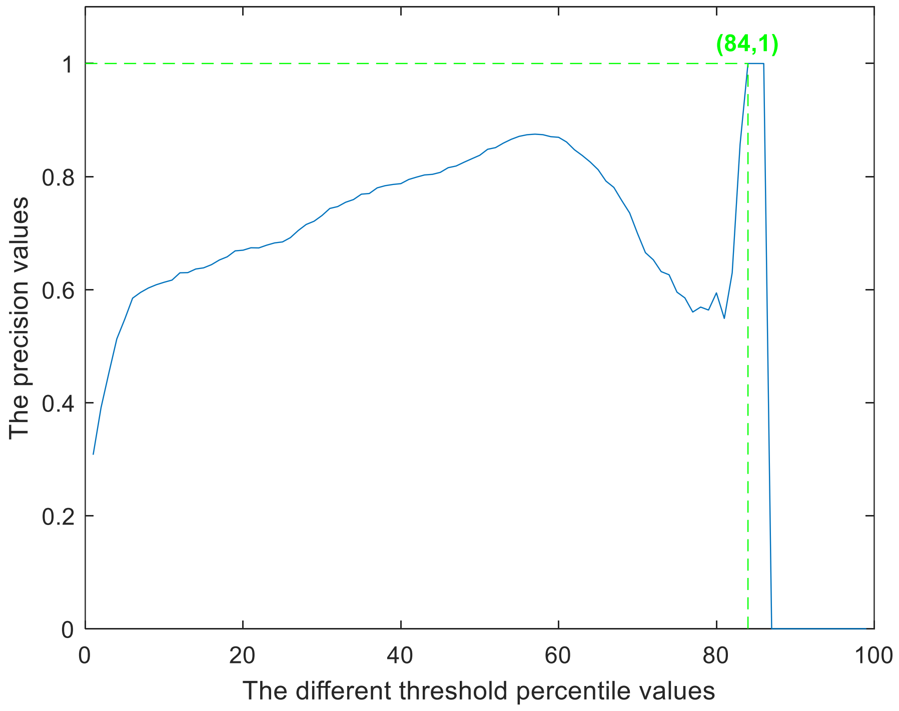


Figure. 1.77 The precision values under different threshold values
for the “Gallery” ’s Structured-Edge result with the 3rd ground truth image

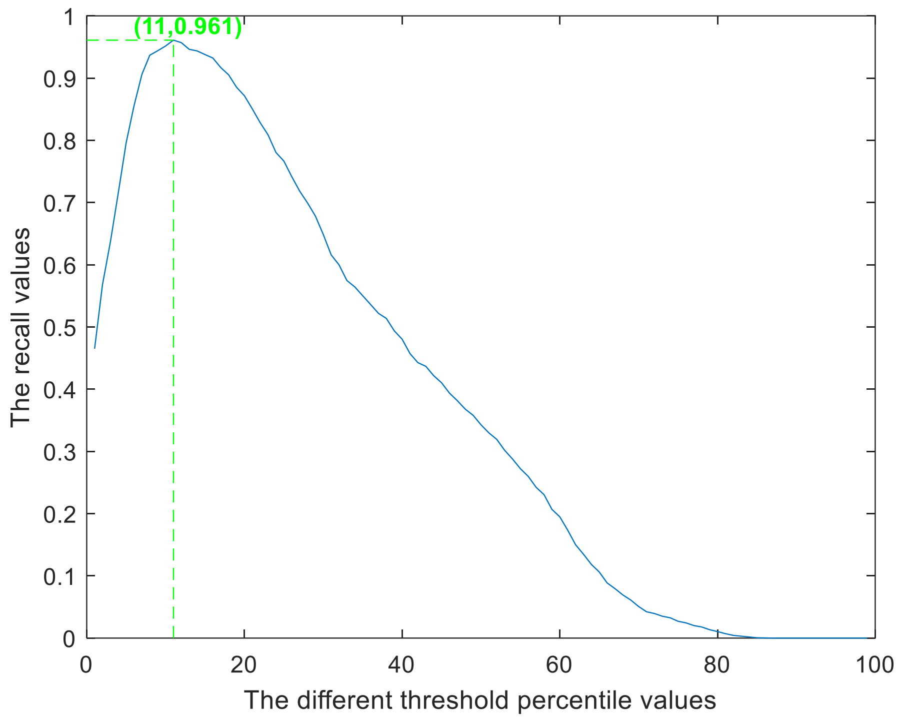


Figure. 1.78 The recall values under different threshold values
for the “Gallery” ’s Structured-Edge result with the 4th ground truth image

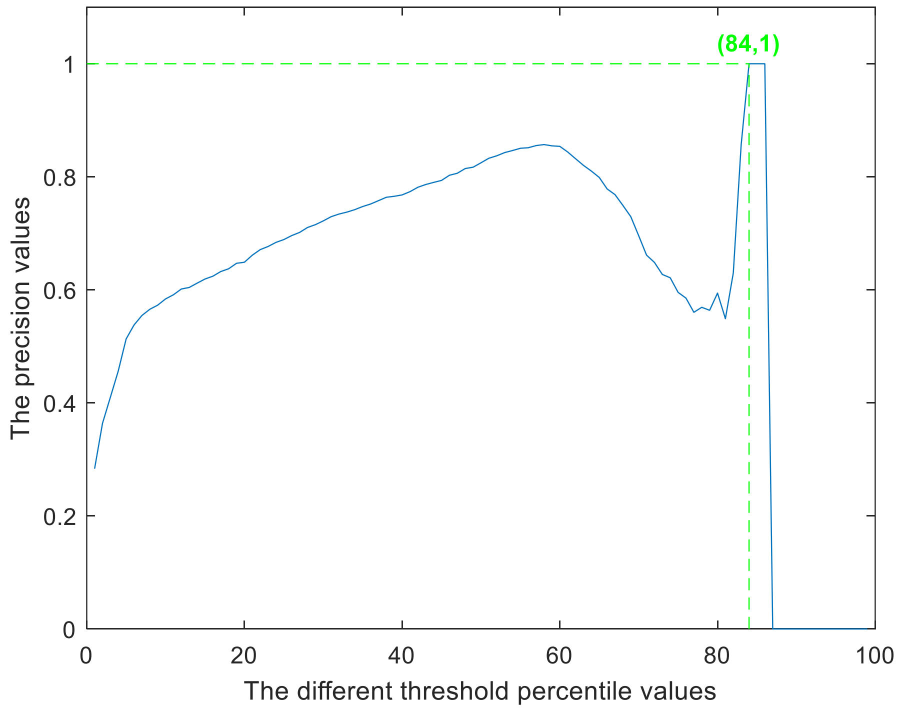


Figure. 1.79 The precision values under different threshold values
for the “Gallery” ’s Structured-Edge result with the 4th ground truth image

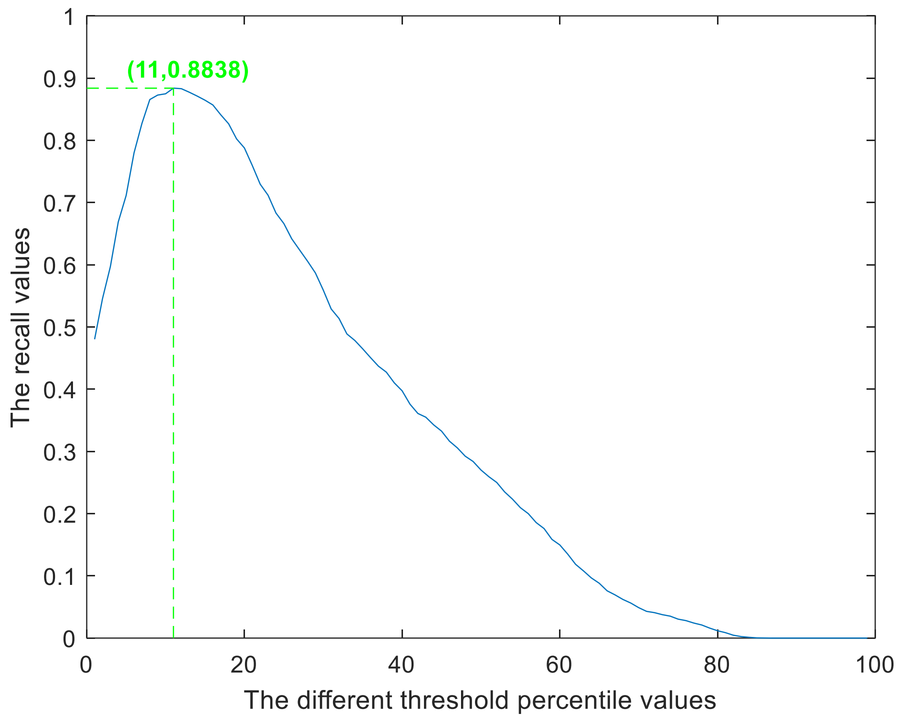


Figure. 1.80 The recall values under different threshold values
for the “Gallery” ’s Structured-Edge result with the 5th ground truth image

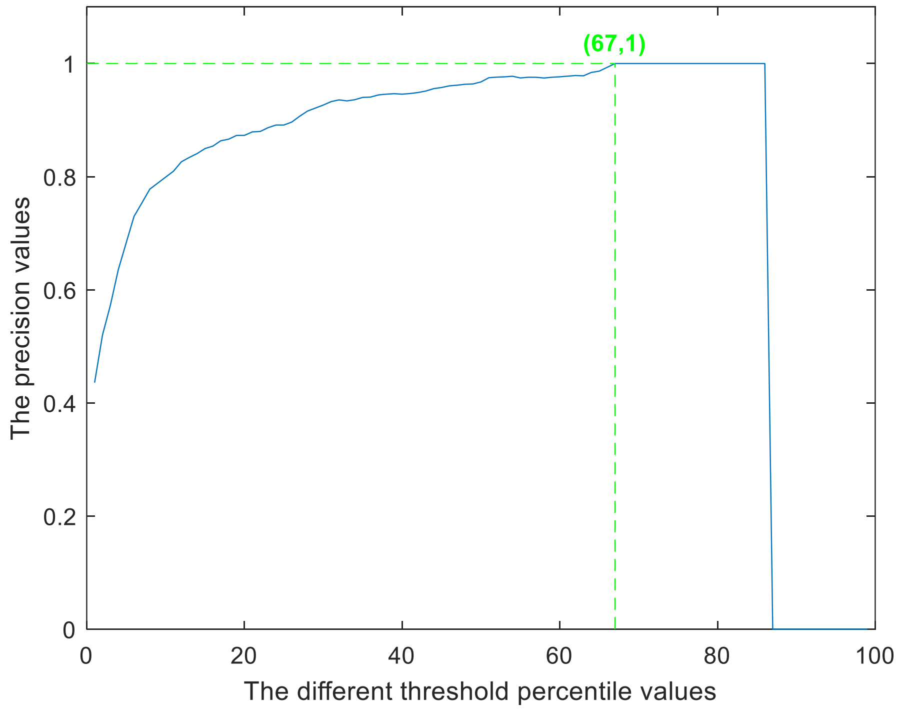


Figure. 1.81 The precision values under different threshold values
for the “Gallery” ’s Structured-Edge result with the 5th ground truth image

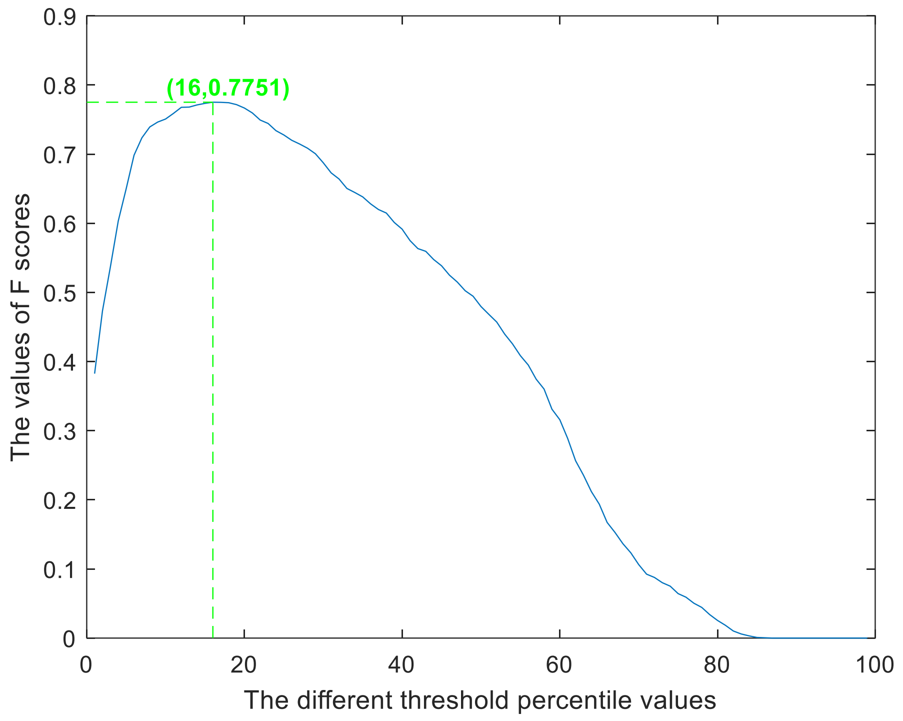


Figure. 1.82 The F values under different threshold values
for the “Gallery” ’s Structured-Edge result among 5 ground truth images

1.4 Discussion

As I have mentioned in the previous report, how to evaluate images' visual quality is always a huge challenge to people, even to image scholars themselves, because the process of that evaluation is so subjective that it's almost impossible for individuals to reach a consensus about whether certain images' visual quality is good enough. It's not unapparent that in order to evaluate an image's quality objectively, we have to ask a number of people's opinions.

However, in this report, I will merely use my own judgement to evaluate the visual quality of resulting images. My discussion of experiment images of Problem 1 is stated as below.

In part (a), I use C++ to implement the Sobel Edge Detector. For the image "Dogs", from the table.1.1, the best threshold value is 0.34 that can get a F measure of 0.2751 and generate a final edge map as shown in the Fig.1.5. For the image "Gallery", from the table.1.2, the best threshold value is 0.22 that can get a F measure of 0.6113 and generate a final edge map as shown in the Fig.1.10.

In part (b), I use OpenCV in Visual Studio 2019 to implement the Canny Edge Detector. Every image I set 13 different combination of parameters to generate edge maps. For the image "Dogs", from the table.1.3, among 13 samples, the best threshold value is low one is 0.31 and the high one is 0.94 that can get a F measure of 0.2963 and generate a final edge map as shown in the Fig.1.23. For the image "Gallery", among 13 samples, the best threshold value is low one is 0.35 and the high one is 0.701 that can get a F measure of 0.6459 and generate a final edge map as shown in the Fig.1.30.

In part (c), I use a MATLAB source code from [4] to implement the Structured Edge Detector. The two resulting images are shown in the Fig.1.37 and the Fig.1.38 respectively. For the image "Dogs", from the table.1.5, the best threshold value is 0.14 that can get a F measure of 0.5654. For the image "Gallery", from the table.1.6, the best threshold value is 0.16 that can get a F measure of 0.7751. Frankly speaking, the SE detector's visual results are better than the Canny detector because , from my point of view, the resulting images of the SE detector can make lots of unnecessary details disappear without losing the desirable pixels that human beings want.

To sum up, in term of the F measure, the Canny detector is better than the Sobel detector and the Structured Edge Detector is better than the Canny detector. From my angle, we definitely don't need the Sobel detector anymore due to its inefficiency in detecting edges. Although the final performance of edge maps of SE is better than Canny, the running time for Canny is faster than the SE. We can still utilize those two techniques. Also, we can improve the Canny detector's performance by applying more advanced denoised techniques such as the BM3D.

What's more, I also notice that the "Gallery" image is likely to get a higher F score than the image "Dogs". I believe this is largely due to a fact that the "Gallery" image has more regular contours, vertical lines or horizontal lines that can make individuals easily reach a consensus whether this pixel belongs to an edge point or not when letting them draw their own ground truth images.

Finally, I have to mention that we cannot make both recall values and precision values high simultaneously, thus forcing us to make a trade-off between two indices. Hence, F measure can be a balanced index that both takes precision and recall into account, which is conductive to the evaluation of the performance of different edge detectors. We cannot get a high F measure if precision is significantly higher than recall, and vice versa. If the sum of precision and recall is a constant A, then

$$F = 2 \cdot \frac{(A - P) \cdot P}{A} = 2 \cdot \frac{-\left(P - {}^{A}\!/_{2}\right)^{2} + {}^{A^{2}}\!/_{4}}{A}$$

So, when precision is equal to recall, the F measure reaches the maximum.

Problem 2: Digital Half-toning (50%)

(a) Dithering (15%)

(b) Error Diffusion (15%)

(c) Color Halftoning with Error Diffusion (10%)

2.1. Abstract and Motivation

In the academic field of digital image processing, halftone is the reprographic technique simulating continuous-tone imagery through the use of dots, which might have different kinds of sizes or spacing. Normally speaking, the continuous-tone imagery contains an infinite range of colors or grays, the halftone process reduces visual reproductions to an image that is printed with only one color of ink, whose dots are of different sizes or spacing. This idea plays an important role in the printing industry based on a basic optical illusion: when halftone dots are tremendously small, the humans' eyes will interpret the patterned areas as if they were smooth tones.

Therefore, in order to see the power of halftone, in this report, I am about to use C++ to implement two algorithms of digital half-toning: one is dithering, the other is error diffusion. For dithering , I will use the fixed thresholding, random thresholding, and dithering matrix to implement digital half-toning. And for error diffusion, I will use Floyd-Steinberg, Jarvis, Judice, and Ninke (JJN), and Stucki's three approaches to implement digital half-toning respectively. Moreover, I will also realize color halftoning with error diffusion by using separable error diffusion and MBVQ-based error diffusion.

After acquiring raw data files from Visual Studio 2019, I will use Matlab 2019b to show images and analyze their effects of digital half-toning with different methods.

2.2. Approach and Procedures

(a) Dithering

i. Fixed thresholding

In this part, I will use C++ to write an algorithm to realize the operation of fixed thresholding. The basic thought is shown as follows:

$$G(i,j) = \begin{cases} 0, & if\ 0 \le F(i,j) \prec T \\ 255, & if\ T \le F(i,j) < 256 \end{cases}$$

ii. Random thresholding

In this part, I will use C++ to write an algorithm to realize the operation of random thresholding. As for random numbers, I will use the "rand()" built-in function to generate them. The basic thought is shown as follows:

$$G(i,j) = \begin{cases} 0, & if\ 0 \le F(i,j) \prec rand(i,j) \\ 255, & if\ rand(i,j)\ \le F(i,j) < 256 \end{cases}$$

iii. Dithering Matrix

In this part, dithering parameters will be specified by an index matrix, whose values located at each entry can render some indication how likely a dot of ink will be turned on. In my code, the first index matrix is

$$I_2(i,j) = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix}$$

where 3 indicates the pixel that is the least likely to be turned on while 0 is the most likely one. In order to have larger Bayer index matrices, I will write a function to recursively use the formula:

$$I_{2n}(i,j) = \begin{bmatrix} 4 \times I_n(i,j) + 1 & 4 \times I_n(i,j) + 2 \\ 4 \times I_n(i,j) + 3 & 4 \times I_n(i,j) \end{bmatrix}$$

After I have acquired the index matrix with 2 by 2, 8 by 8, or 32 by 32, I am about to get the relevant threshold matrix T by the following formula

$$T(x,y) = \frac{I_N(x,y) + 0.5}{N^2} \times 255$$

where N^2 denotes the total number of pixels in different threshold matrices, and (x,y) is the matrix's location. In order to make threshold matrices operate the full image, I will implement the following formula:

$$G(i,j) = \begin{cases} 0, & if\ F(i,j) \le T(i\ mod\ N, j\ mod\ N) \\ 255, & otherwise \end{cases}$$

In my program, I will create I_2, I_8, I_{32}threshold matrices and see the results of halftoning the image "Light House".

(b) Error Diffusion

Step1: Initialize a matrix $\tilde{f}(i,j)$ by copying the original image $f(i,j)$.

Step2: For each pixel, binarize it by using the following formula:

$$b(i,j) = \begin{cases} 255, & if\ \tilde{f}(i,j) > T \\ 0, & otherwise \end{cases}$$

Then diffuse error $(e = \tilde{f}(i,j) - b(i,j))$ forward with the serpentine scanning by utilizing the following three matrices:

i. Floyd-Steinberg

$$\frac{1}{16}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 7 \\ 3 & 5 & 1 \end{bmatrix}$$

ii. Jarvis, Judice, and Ninke (JJN)

$$\frac{1}{48}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 5 \\ 3 & 5 & 7 & 5 & 3 \\ 1 & 3 & 5 & 3 & 1 \end{bmatrix}$$

iii. Stucki

$$\frac{1}{42}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 4 \\ 2 & 4 & 8 & 4 & 2 \\ 1 & 2 & 4 & 2 & 1 \end{bmatrix}$$

In Visual Studio 2019, I will use C++ to implement the algorithm to realize the three kinds of error diffusion respectively.

(c) Color Halftoning with Error Diffusion

i. Separable Error Diffusion

In this part, I will separate an image into CMY three channels and apply the Floyd-Steinberg error diffusion algorithm of part(b) to handle each channel separately, thus making it possible to achieve color halftoning.

ii. MBVQ-based error diffusion

At pixel (i, j), I denote its three R, G, and B values by RGB (i, j) and the RGB value and the accumulated error by e(i, j) . The key idea of color diffusion can be formalized as follows:

For each pixel (i; j) in the image, the algorithm will do:

Step 1: Determine MBVQ for each RGB (i, j).

```
pyramid MBVQ(R value, G value, B value)
{
if ((R+G) > 255)
   if ((G+B) > 255)
      if ((R+G+B) > 510)  return CMYW;
      else return MYGC;
   else return RGMY;
else
   if (!((G+B) > 255))
      if (!((R+G+B) > 255))  return KRGB;
      else return RGBM;
   else return CMGB;
}
```

Step 2: Find the vertex v ∈ MBVQ which is closest to RGB(i, j)+e(i, j).

This step is the only difference between separable error diffusion and color Diffusion. In this step, we intend to enable the algorithm to look for the closest vertex in the MBVQ of the color, rather than the closest of the eight vertices of the cube.

Notice: Because I use C++ language programming language, I write my MBVQ decision tree on my own with the help of the TA’s hint.

Step 3: Compute the quantization error RGB(i; j) + e(i; j) - v.

Step 4: Distribute the error to “future" pixels just as the separable error diffusion does.

2.3. Experimental Results

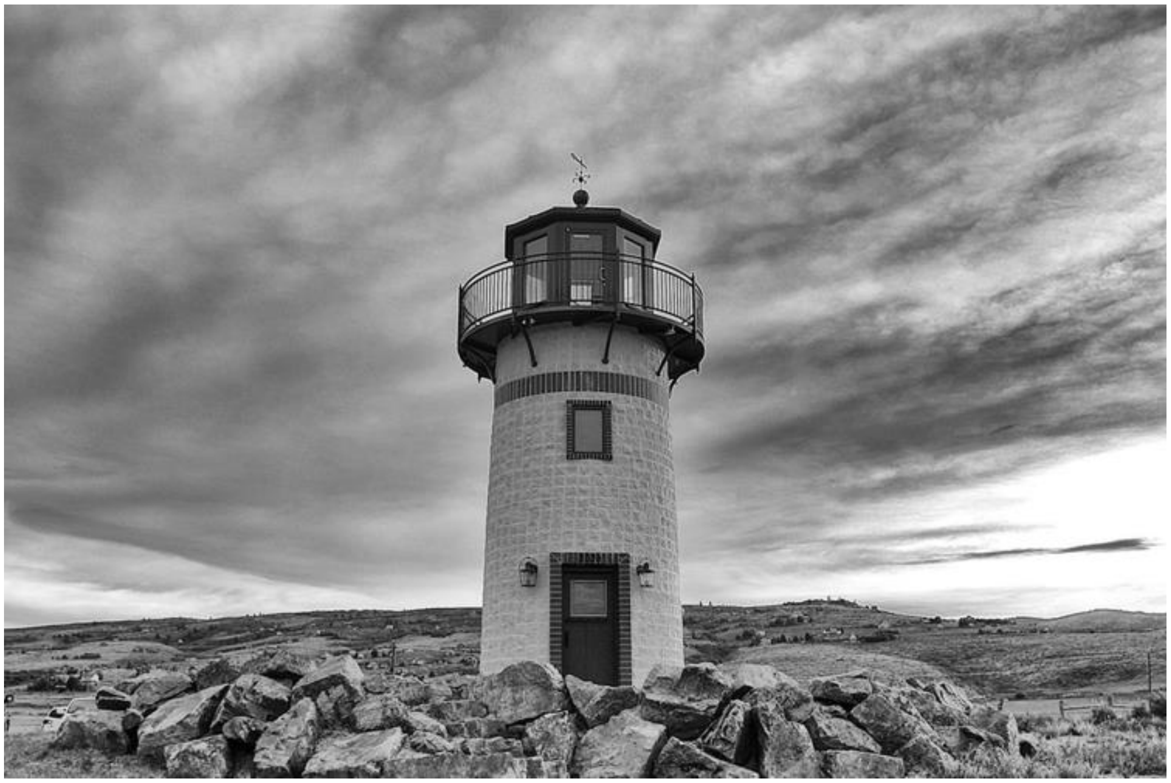


Figure. 2.1 The original “Light House” image

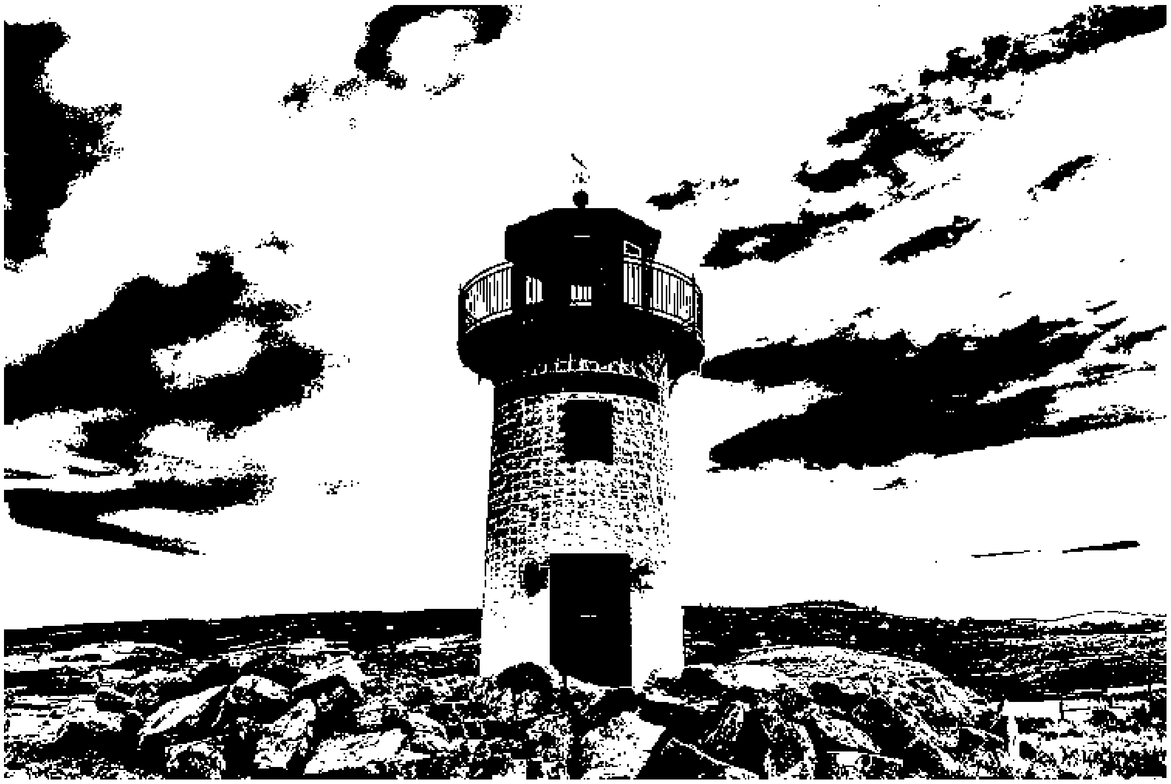


Figure. 2.2 The resulting “Light House”
Image after fixed thresholding

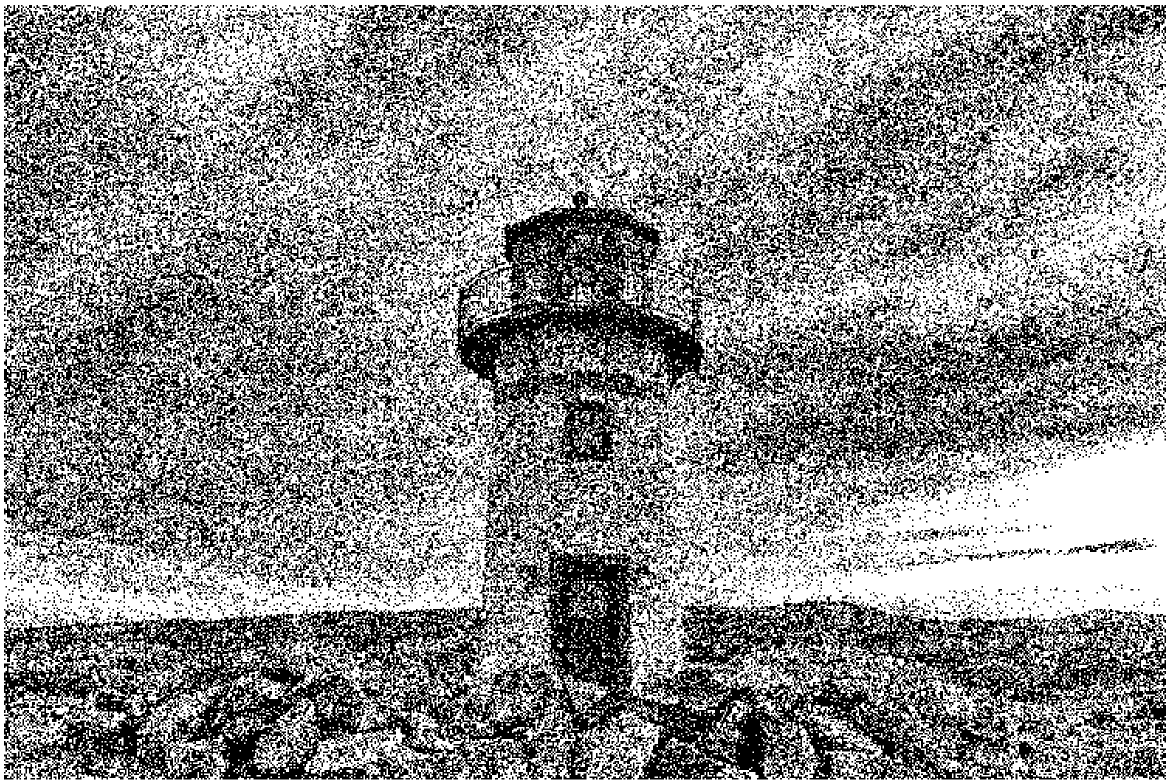


Figure. 2.3 The resulting “Light House” Image after random thresholding

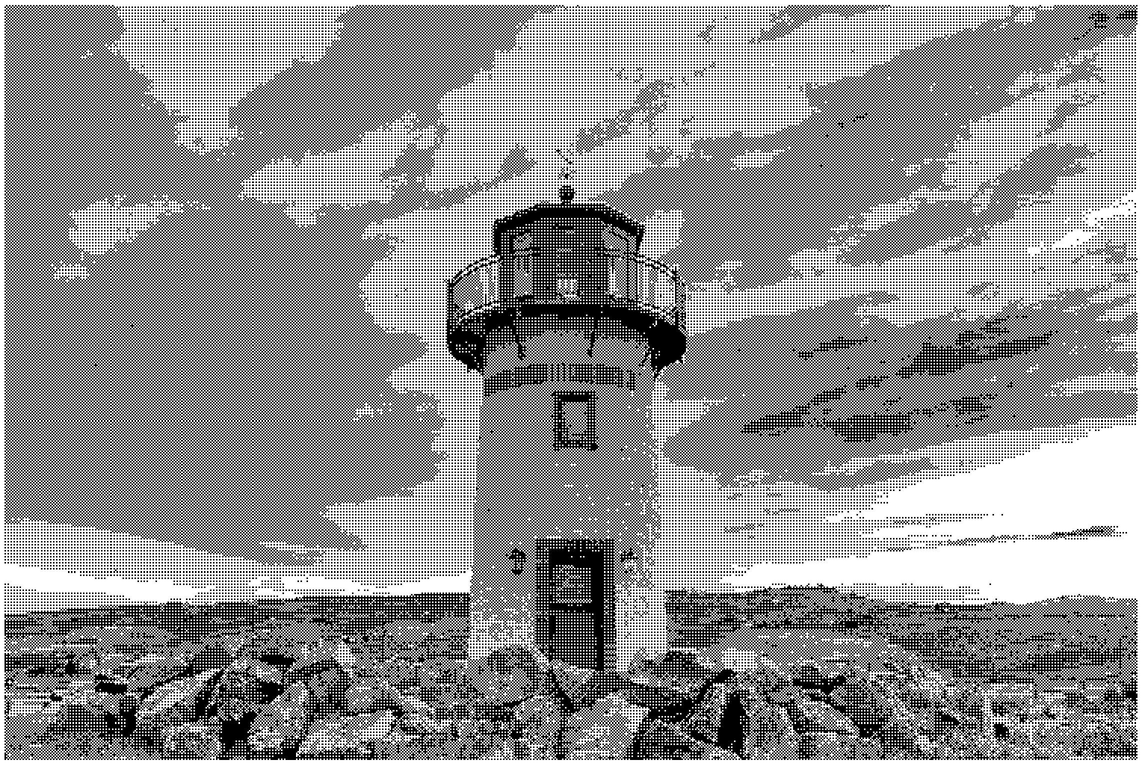


Figure. 2.4 The resulting “Light House” Image after using the dithering matrix I_2

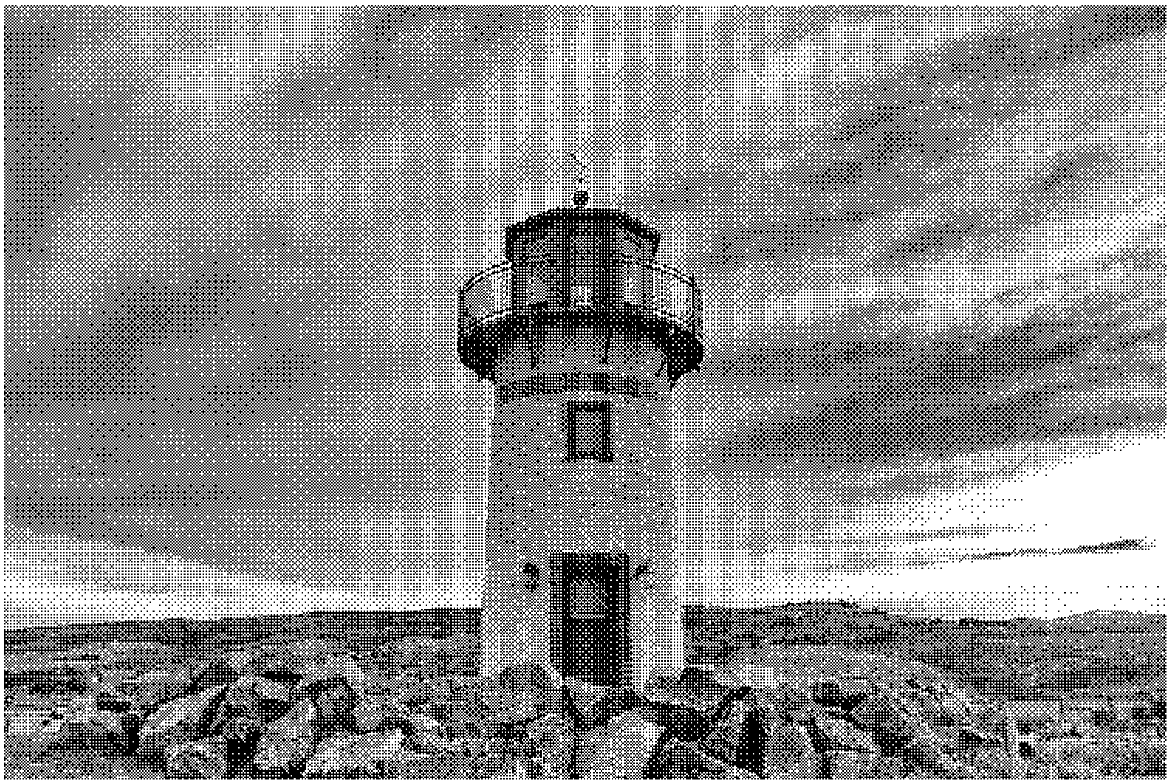


Figure. 2.5 The resulting "Light House"
Image after using the dithering matrix I_8

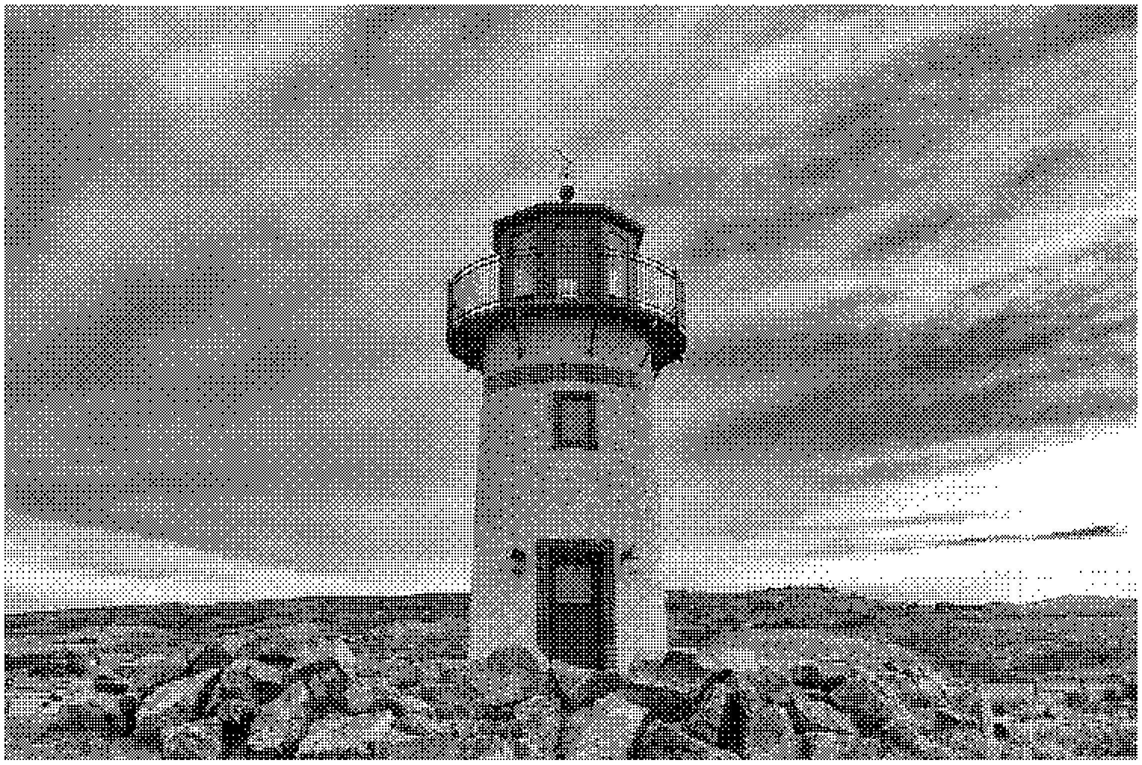


Figure. 2.6 The resulting "Light House"
Image after using the dithering matrix I_{32}

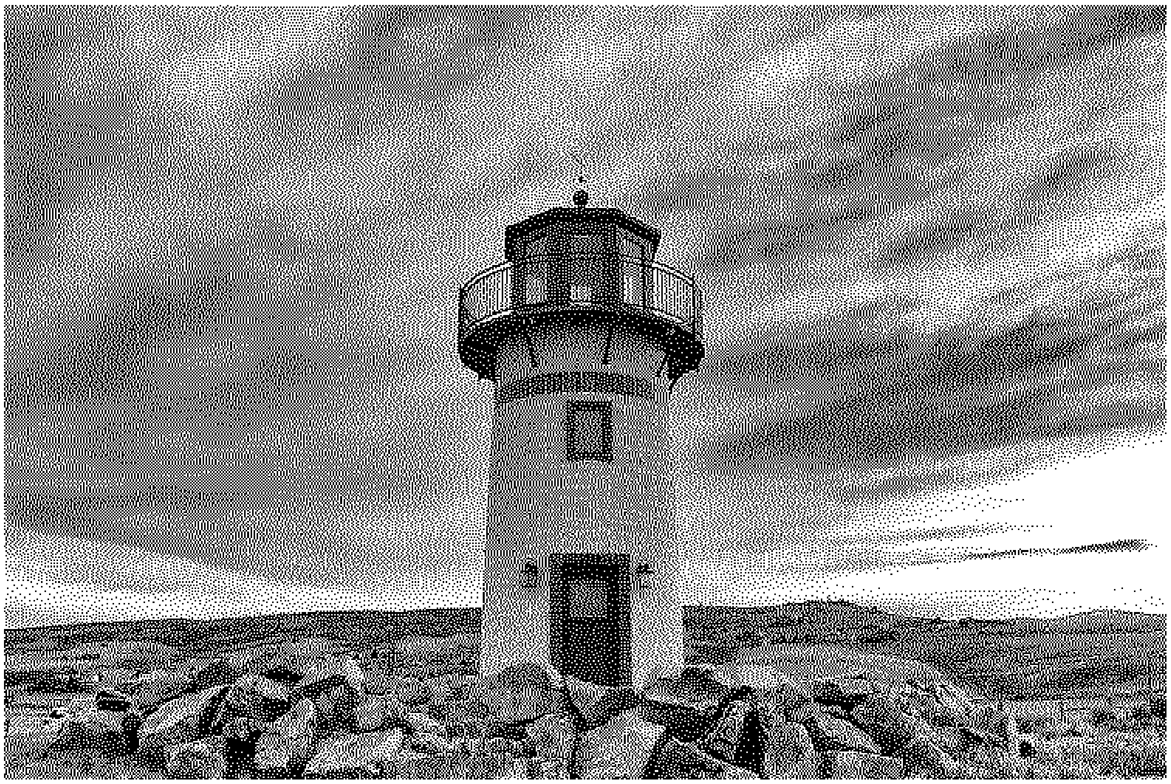


Figure. 2.7 The resulting “Light House”

Image after using Floyd-Steinberg’s error diffusion

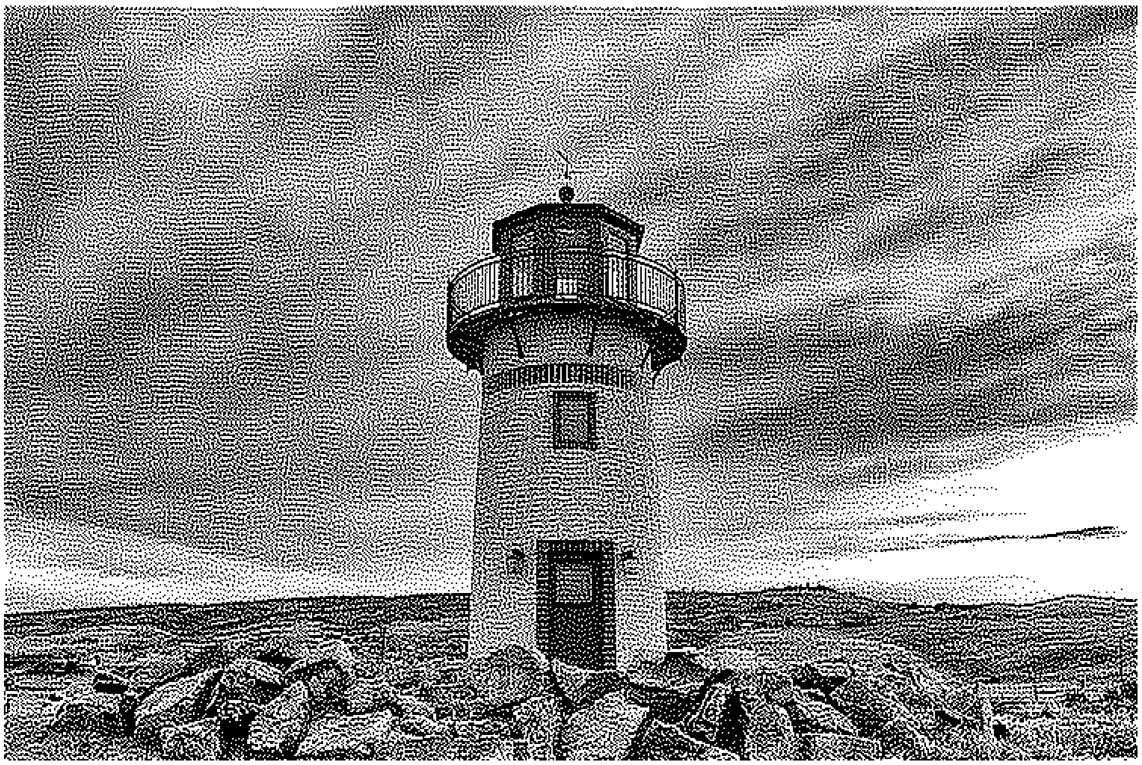


Figure. 2.8 The resulting “Light House”

Image after using Jarvis, Judice, and Ninke (JJN)’s error diffusion

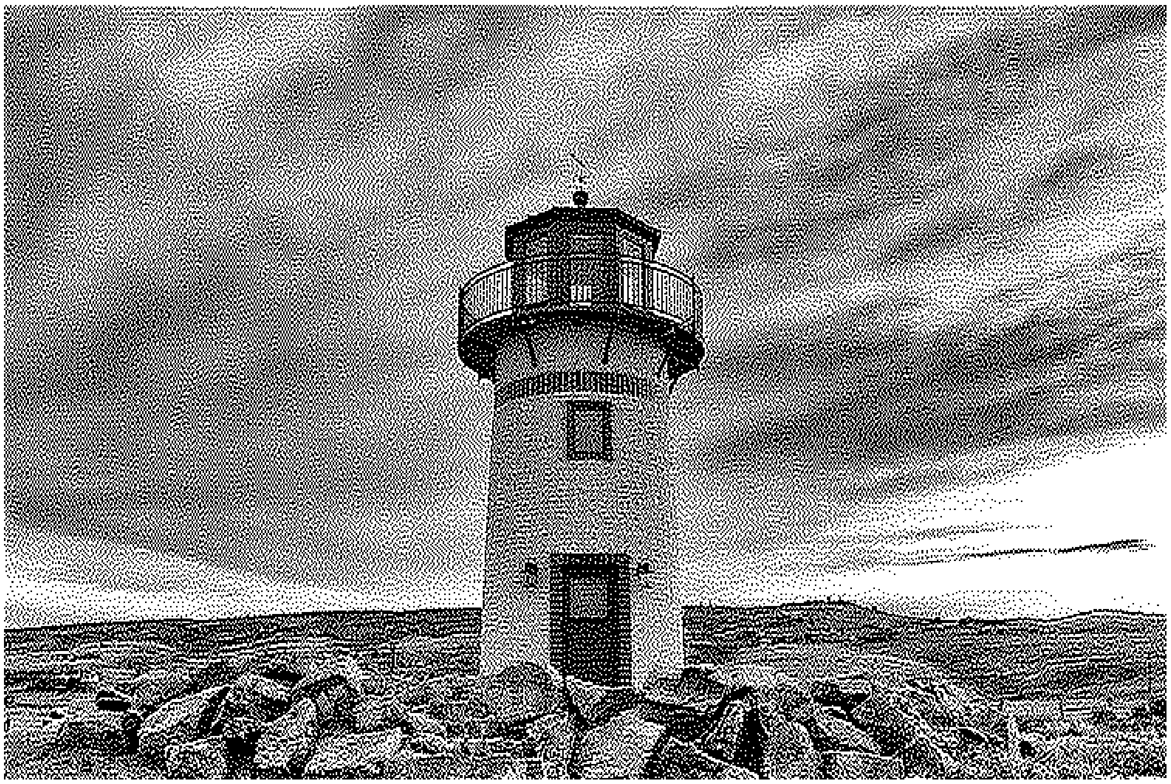


Figure. 2.9 The resulting “Light House”
Image after using Stucki’s error diffusion

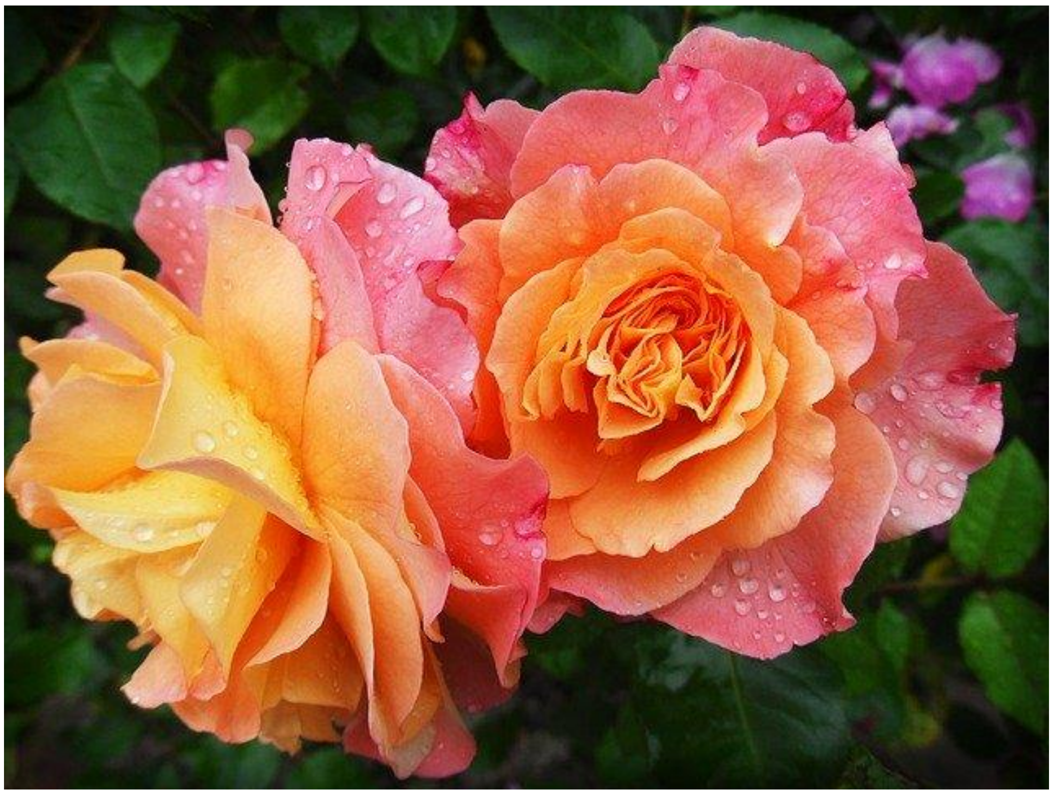


Figure. 2.10 The original “Rose” image

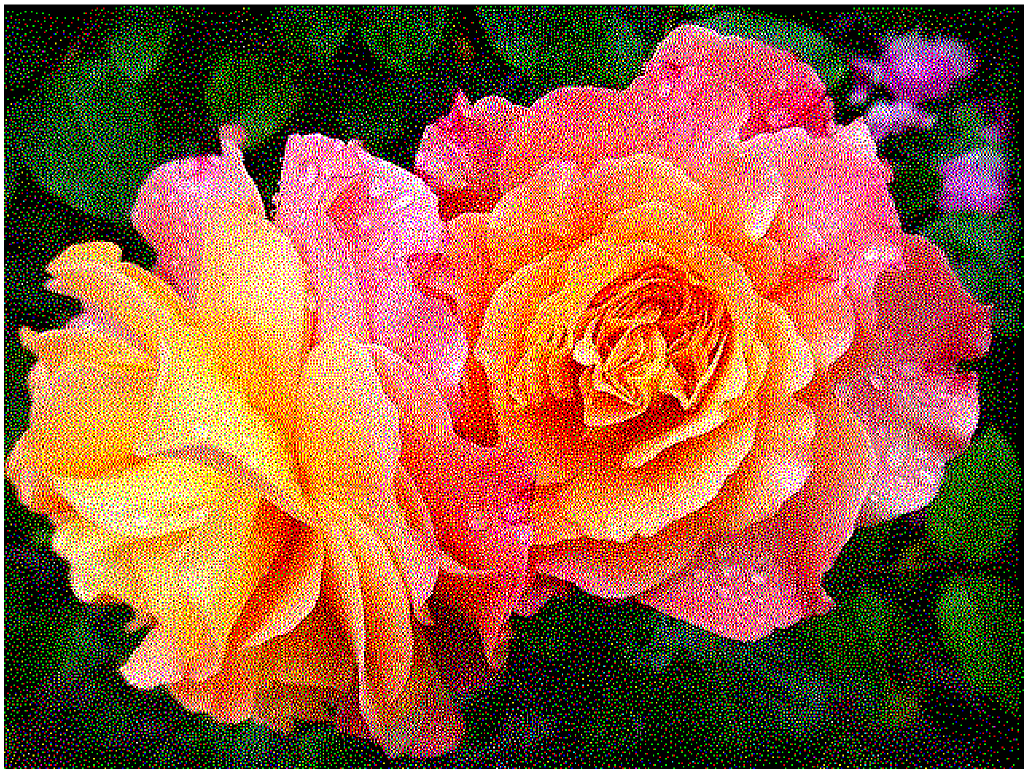


Figure. 2.11 The resulting "Rose" image
after using separable error diffusion

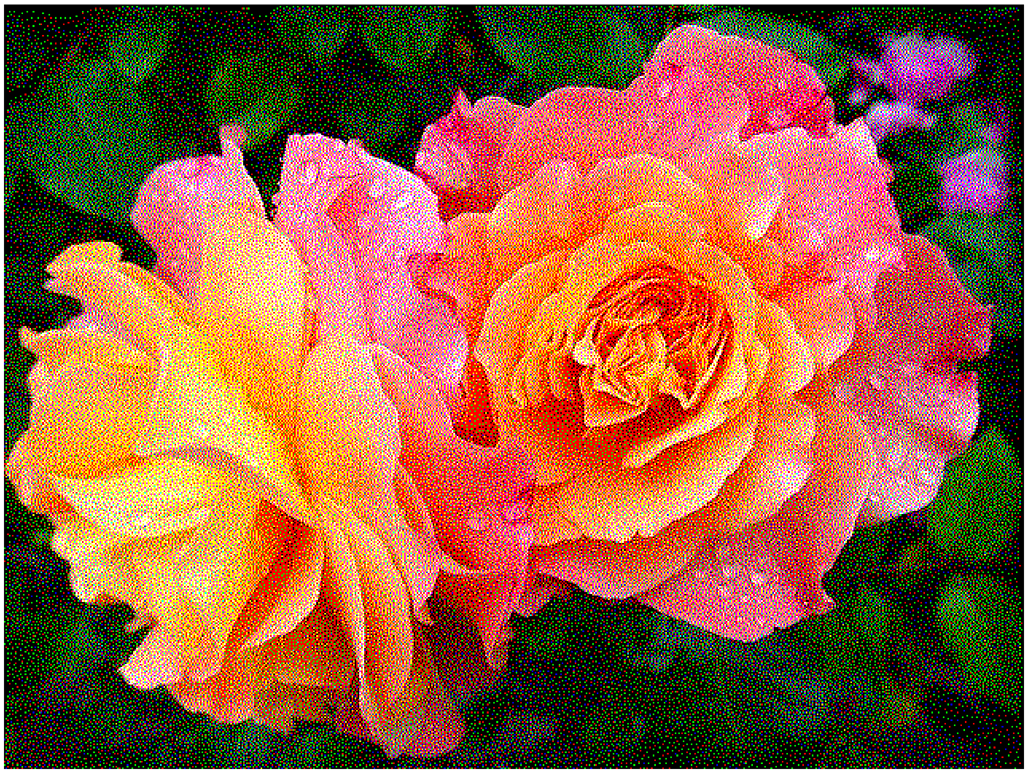


Figure. 2.12 The resulting "Rose" image
after using MBVQ-based error diffusion

2.4. Discussion

As stated before, in this report, I will merely use my own judgement to evaluate the visual quality of resulting images. My discussion of experimental results of Problem 2 is stated as below.

In part (a), according to the Fig.2.2, frankly speaking, if I print that image on a white paper, I guess individuals will have a good visual performance, meaning that people can recognize the original image as shown in the Fig.2.1 from the printed image. However, if humans in industry always choose this way to print something, it will be definitely a huge waste of ink. Taking environment-friendly economy into account, we ought to find more advanced methods. From the Fig.2.3, it's not hard for us to reach a conclusion that random thresholding is totally unacceptable, which will cause lots of noise that is a great disadvantage to people's visual performance. After using the dithering matrix I_2, as shown in the Fig.2.4, the printed image can save more ink than the fixed thresholding way without losing so much characteristics. When I increase the size of the dithering matrix, as shown in the Fig.2.5 and the Fig.2.6, at first glance I am so satisfied because I believe those two images have better visual quality than the Fig.2.4 in terms of the sketch's beauty. However, after careful thought, I believe there are still some shortcomings of the Dithering Matrix technique because the resulting images have texture-like and periodic visual patterns. One main factor, I believe, is that as the matrix's size becomes larger, the process of implementing the algorithm will make images have a number of boundaries due to non-overlapping and independent blocks, therefore resulting in the periodicity of artifacts.

In part (b), I use three different matrices (Floyd-Steinberg, JJN, and Stucki) to implement error diffusion. From the Fig.2.7, the Fig.2.8, and the Fig.2.9, I believe the three images' visual performance is better than the images resulting from the Dithering Matrix technique because those three images don't have many crossing or periodic patterns with the help of the feedback of neighborhood pixels, which enables the system to become self-correcting while implementing the algorithm. Hence, I prefer the error diffusion method. However, from the paper [5], even if causal filtering as I use in my algorithm is an attractive feature, it is exactly the reason for one disadvantage of error diffusion known as directional hysteresis. In order to resolve that problem, we can use a new digital halftoning technique based on multiscale error diffusion as shown in the paper[5].

In part (c), I use the Floyd-Steinberg matrix to implement color separable error diffusion as shown in the Fig.2.11. This approach's main shortcoming, according to the paper [6], I believe, is that the colors used to reduce the noticeability of the pattern cannot be satisfied by a simple Cartesian product generalization of monochrome halftoning. The MBVC can characterize a set of participating halftone colors for given input color. Because of the MBVC, we are able to derive ink relocation, a postprocess to arbitrary halftoning algorithms, thus improving images' visual quality as shown in Fig.2.12. By making two images small enough, I can notice that the MBVQ way can make images' color more evenly distributed.

References

[1] H.G. Barrow and J.M. Tenenbaum "Interpreting line drawings as three-dimensional surfaces", Artificial Intelligence (1981): vol 17, issues 1–3, pages 75–116.

[2] [Online] Available: https: //docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_ detector/canny_detector.html

[3] Dollar, Piotr, and C. Lawrence Zitnick. "Fast Edge Detection Using Structured Forests." IEEE Transactions on Pattern Analysis and Machine Intelligence 37.8 (2015): 1558–1570. Web.

[4] [Online] Available: https://github.com/pdollar/edges

[5] Ioannis Katsavounidis and C.-C. Jay Kuo "A Multiscale Error Diffusion Technique for Digital Halftoning", IEEE Transactions on Image Processing (1997): vol. 6, no. 3.

[6] Doron Shaked, Nur Arad, Andrew Fitzhugh, Irwin Sobel "Color Diffusion: Error-Diffusion for Color Halftones", HP Laboratories Israel (1999): HPL-96-128(R.1).