EE569 Digital Image Processing

HOMEWORK #2

Issued: 01/31/2022 Due: 11:59PM, 02/20/2022

Problem 1: Edge Detection (50 %)

1.1 Motivation

Edge detection is the process of calculating regions of varying brightness in a picture using matrix algebra. Areas with substantial variances in pixel brightness generally represent the boundary of an item. We detected all of the edges in an image after discovering all of the major changes in intensities.

The Sobel method is concerned with determining boundaries by evaluating the degree of change on 2-dimensional pictures, with a specific emphasis on regions with a high frequency of change. The exact amount of change at each point in the supplied grayscale image is a common application.

The Canny operator is a multilevel edge detection technique invented by Australian computer scientist John F. Canny in 1986 with the purpose of finding the best edge.

The Structured Edge (SE) detector is a quick edge detector with high accuracy. It may be used as a source of data for any vision technique that requires high-quality edge maps.

The distinction between edge detection and other filtering algorithms is the filter used.

1.2 Approach

1.2.1 Sobel Edge Detector

A pair of 3×3 convolution kernels make up the operator. One kernel is merely a 90° rotation of the other.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

These two matrices are used to detect the boundary with the largest change associated with each point in the horizontal and vertical directions, respectively. They can be used separately for the input image to measure the change in each direction, and then combined to see the absolute amount of change and the direction of change. The formula is as follows:

$$G = \sqrt{{G_x}^2 + {G_y}^2}$$

Then, using the cumulative histogram of the normalized Gradient map, establish the edge map threshold. G_x, G_y are normalized to 0~255 for output.

1.2.2 Canny Edge Detector

Basic Steps:

Filter image with Gaussian. A Gaussian matrix is multiplied by each pixel point and its neighbors, and its average with weights is taken as the final gray value.

Determine the gradient value and direction. The filter created by applying a Gaussian filter for gradient computation yields results comparable to the Sobel operator, i.e., the larger the weight, the closer the pixel point to the center.

Suppression of non-extreme values in gradient pictures. It's conceivable that the Gaussian filtering method magnifies the edges. This phase employs a criteria to filter points that aren't edges, allowing an edge to be as broad as feasible by 1 pixel point: if a pixel point belongs to an edge, its gradient value in the gradient direction is the highest. Otherwise, the gray value is set to 0 because it is not an edge.

Use two thresholds for edge connections. Points less than the low threshold are considered false edges with a value of 0, whereas points more than the high threshold are considered strong edges with a value of 1. The pixel positions in between must be double-checked. The edges are linked into contours based on the high threshold image, and when the contour's endpoint is reached, the algorithm searches for the point that satisfies the low threshold among the 8 neighboring points of the breakpoint, and then collects new edges based on this point until the entire image is closed.

1.2.3 Structured Edge

A data-driven approach for finding an edge is Structured Edge. There is a structure to each edge. To determine if a pixel is an edge, a random forest classifier is utilized. A random forest is made up of several decision trees that have been trained using random data and feature sets. To determine if a pixel is an edge, a 16 × 16 patch segmentation mask is utilized. A contour can be detected using alternative 0s and 1s.

1.2.4 Performance Evaluation

Precision: The accuracy rate in edge detection indicates the probability that the machine generated boundary pixels are true boundary pixels.

$$P = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Recall: Recall in edge detection represents the probability of detecting a true boundary pixel over all true boundary pixels.

$$R = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

F score: The F-score may be calculated using the accuracy and recall. The harmonic mean of accuracy and recall is the F-score. Any noticeable discrepancies in the classifier will be highlighted.

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

1.3 Results

(a)

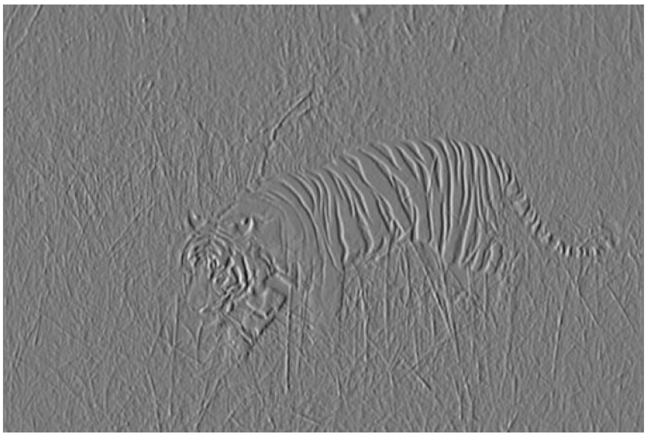
Original Image

Original Image

Figure 1: Original Image

(1)

x-gradient



x-gradient

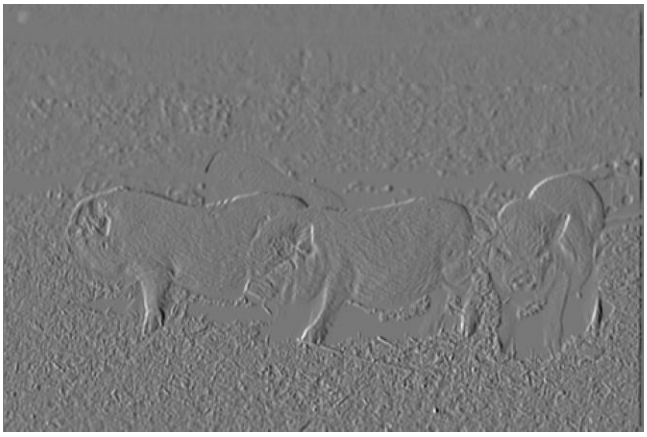
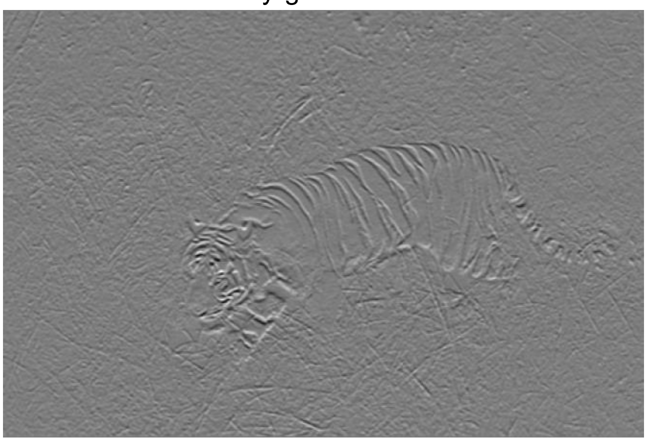


Figure 2: X-gradient

y-gradient



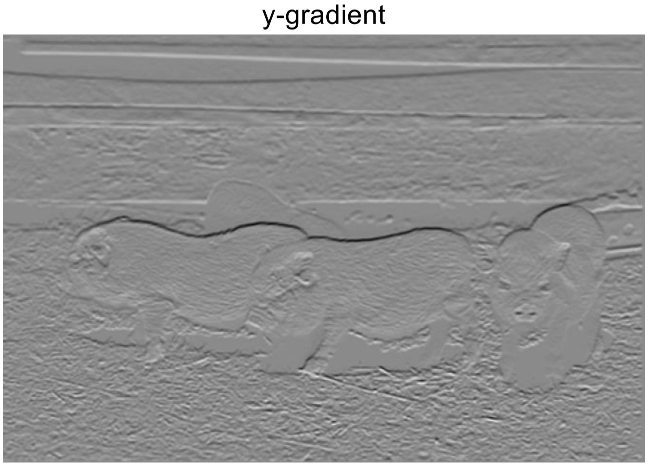
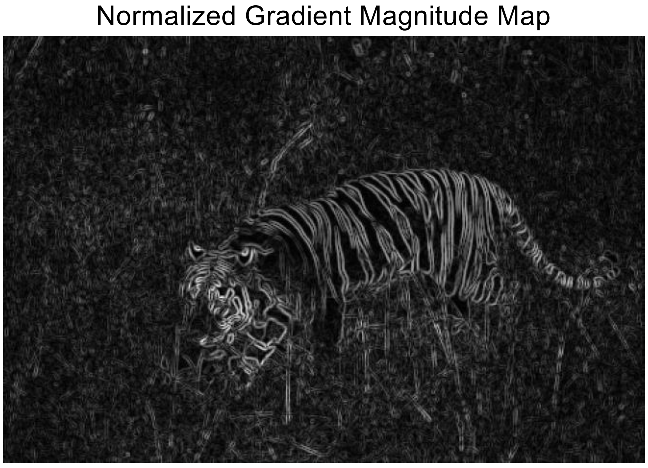


Figure 3: Y-gradient

(2)



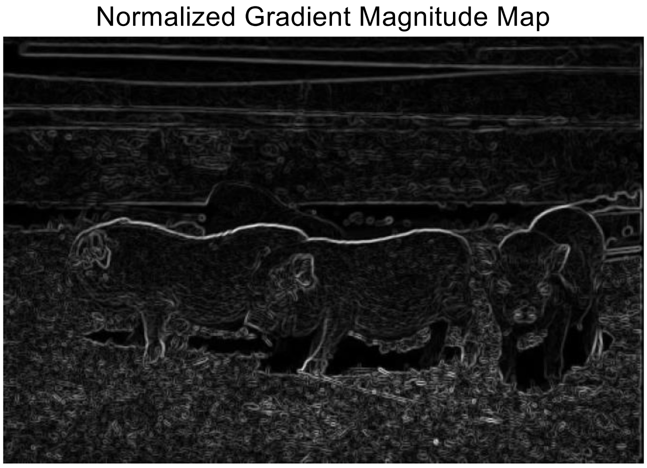
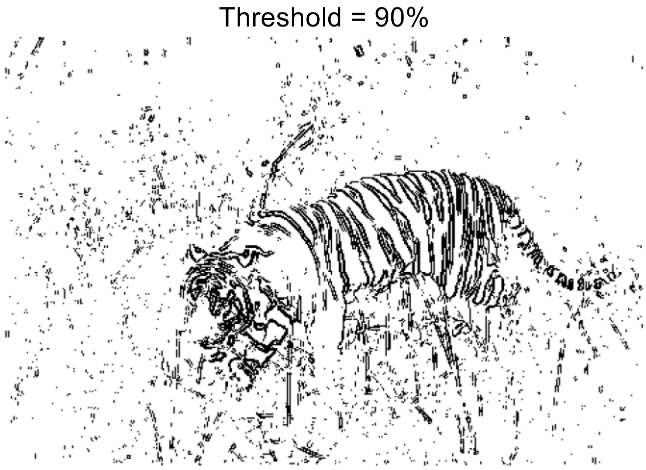


Figure 4: The normalized gradient magnitude map

(3)



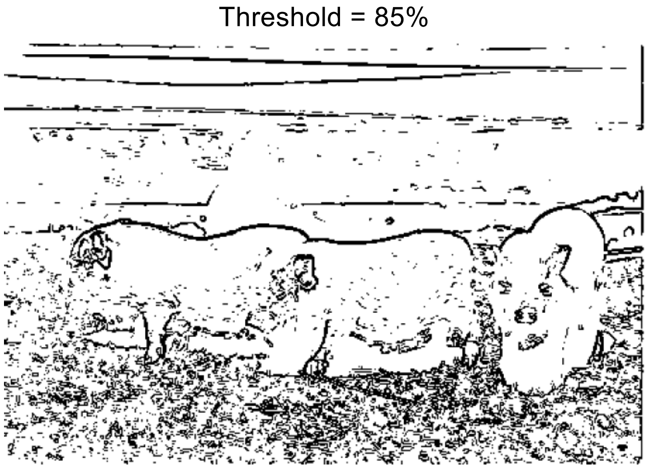


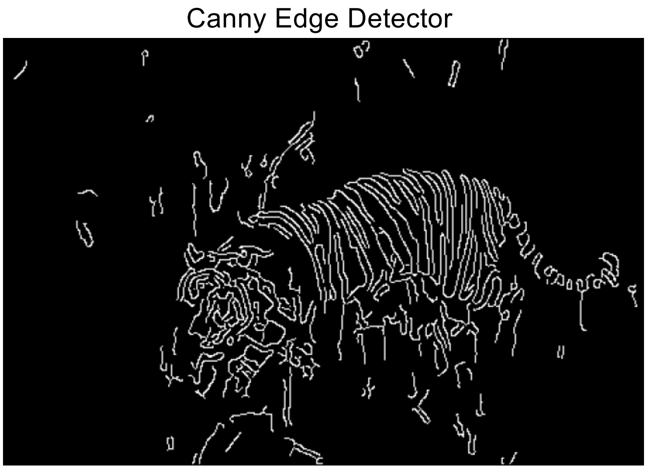
Figure 5: Edge map

(b)

(1) After computing the gradient image, the obtained gradient image has numerous problems such as thick and wide edges, weak edge interference, etc. Non-extreme value suppression may now be used to filter non-edge points so that the width of the edge is as broad as feasible 1 pixel point: if a pixel point belongs to an edge, then the gradient value of this pixel point in the gradient direction is the maximum. Otherwise, there is no edge and the gray value is set to 0. This removes a significant number of non-edge pixel points.

(2) Two thresholds are chosen, and points less than the low threshold are deemed false edges with a value of 0, while points more than the high threshold are regarded strong edges with a value of 1. If the center pixel is close to a pixel specified as an edge, it is considered an edge; otherwise, it is considered a non-edge.

(3)



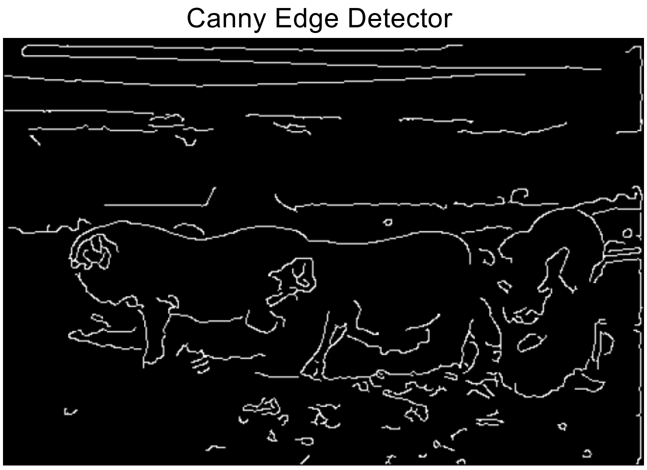
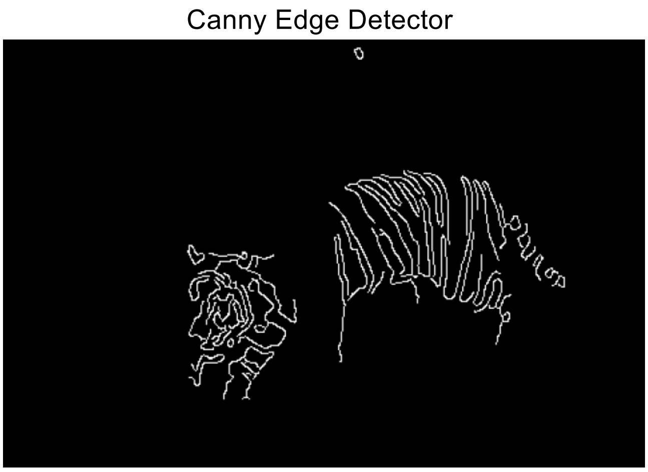


Figure 6: Low Threshold = 0.15 and High Threshold = 0.3



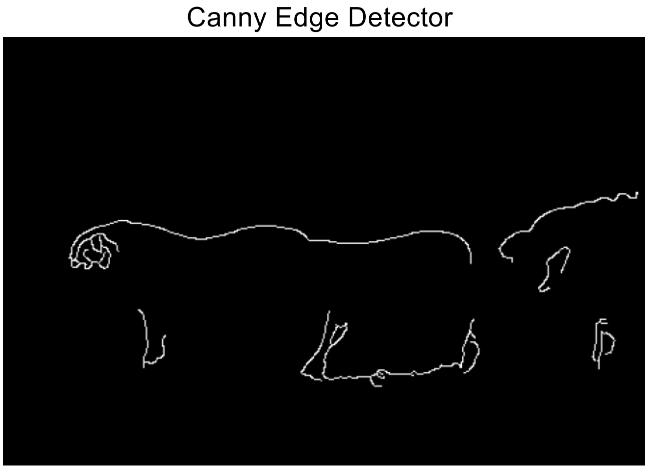


Figure 7: Low Threshold = 0.15 and High Threshold = 0.6

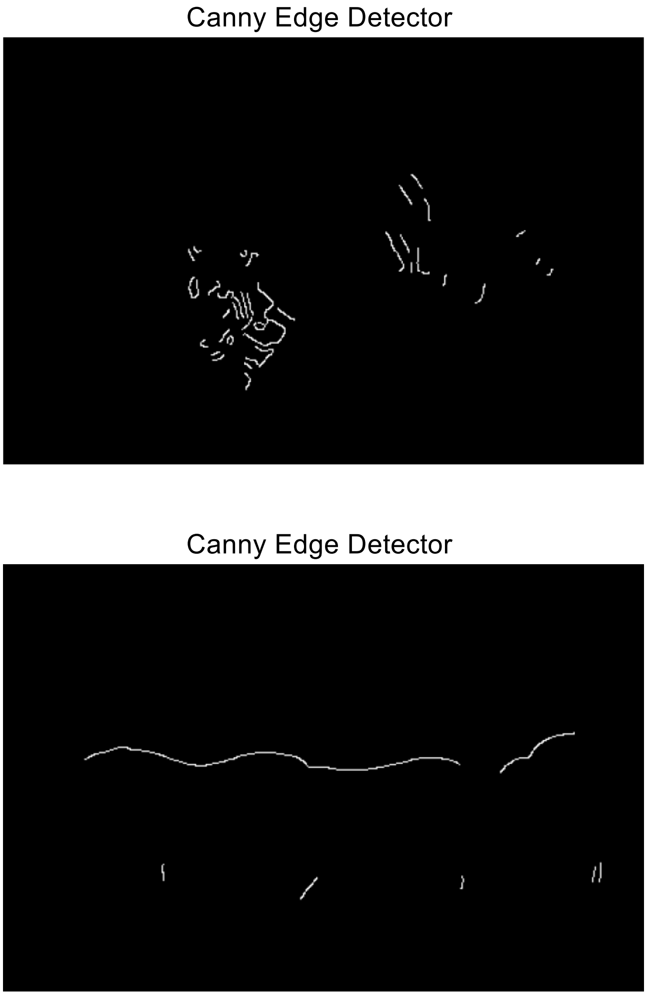
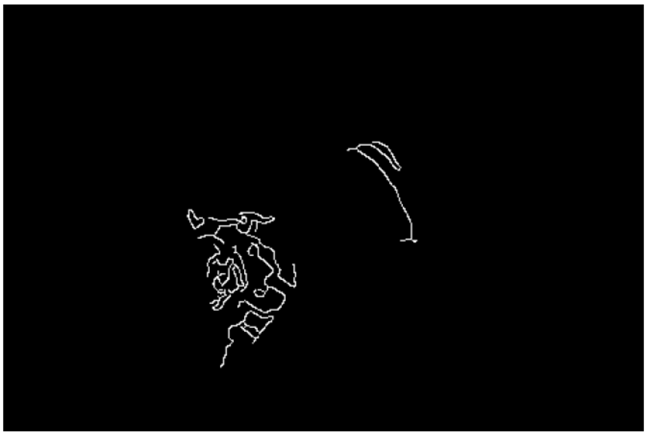


Figure 8: Low Threshold = 0.5 and High Threshold = 0.7

Canny Edge Detector



Canny Edge Detector

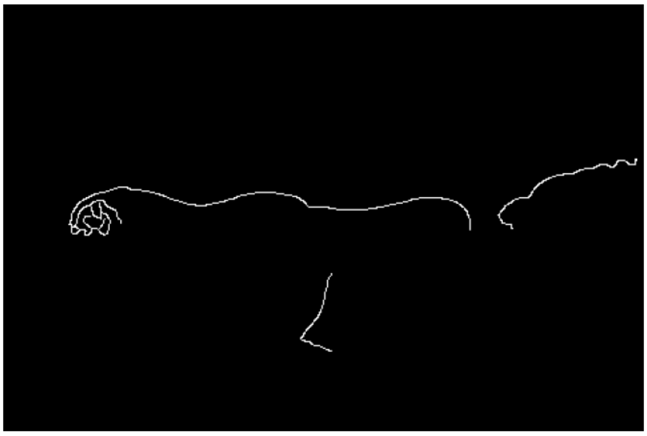


Figure 9: Low Threshold = 0.1 and High Threshold = 0.8

After experimenting with several low-high combinations, I discovered that for the same low threshold and different high threshold, the number of edges is significantly higher in the case of lower high threshold in both Tiger and Pig edge maps. Furthermore, when the difference between the two thresholds is greater, the edge connection performs better. Furthermore, if the low threshold is too low and the high threshold is too high, the recall will be low.

(c)
(1)

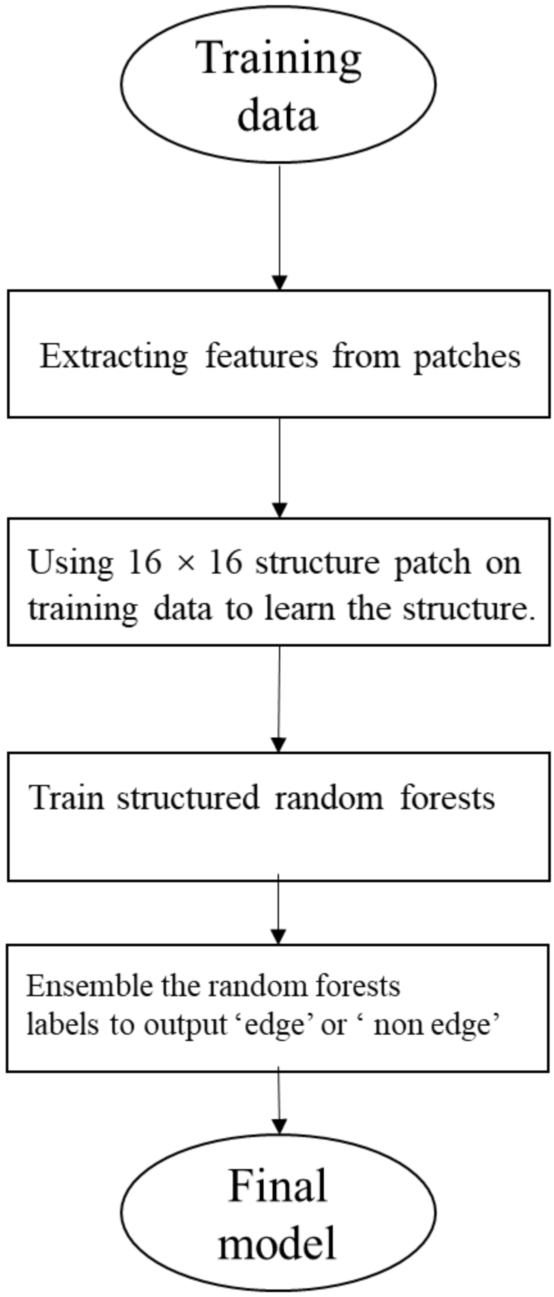


Figure 10: SE detection algorithm

We must first load training data before extracting patches' features. Relevant characteristics are retrieved in such a way that the Gini impurity or purity is high, resulting in a separate categorization. The decision trees in a random forest classifier are a collection of them. To learn the structure, we utilize a 16-16 structure patch using training data. After passing through each decision tree, the test data is given a label. The pixel is classified as 'edge' or 'not edge' by a random forest classifier based on voting or linear regression.

(2)
Decision trees function by picking the greatest characteristics to divide over and over

again. These characteristics are chosen by an information gain maximization technique. Entropy, which estimates the average degree of information in the various outcomes of a variable, may be used to assess information gain. The lesser the entropy, the less unpredictable the situation.

$$Entropy = \sum_{i} -p_i \log_2 p_i$$

The probability of class I is given by p_i. If we opt to divide on a feature by feature, we can now compute the information gain.:

$$IG(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{S} Entropy(S_v)$$

S_v is a subset of S where feature A's value is v, and Values(A) is a subset of all possible feature A values. As can be observed, the first term indicates our original set's total entropy, whereas the second term represents the predicted value of entropy. The predicted value of the entropy A after partitioning S according to feature A is the second term. We decide to partition based on the characteristic that improves this function the most.

A random forest is a decision tree that has been combined. It has a minimal computing cost and a strong ability to pick out the most useful characteristics. It is unaffected by feature normalization. The overfit problem might be avoided if tree diversity is sufficient..

(3)

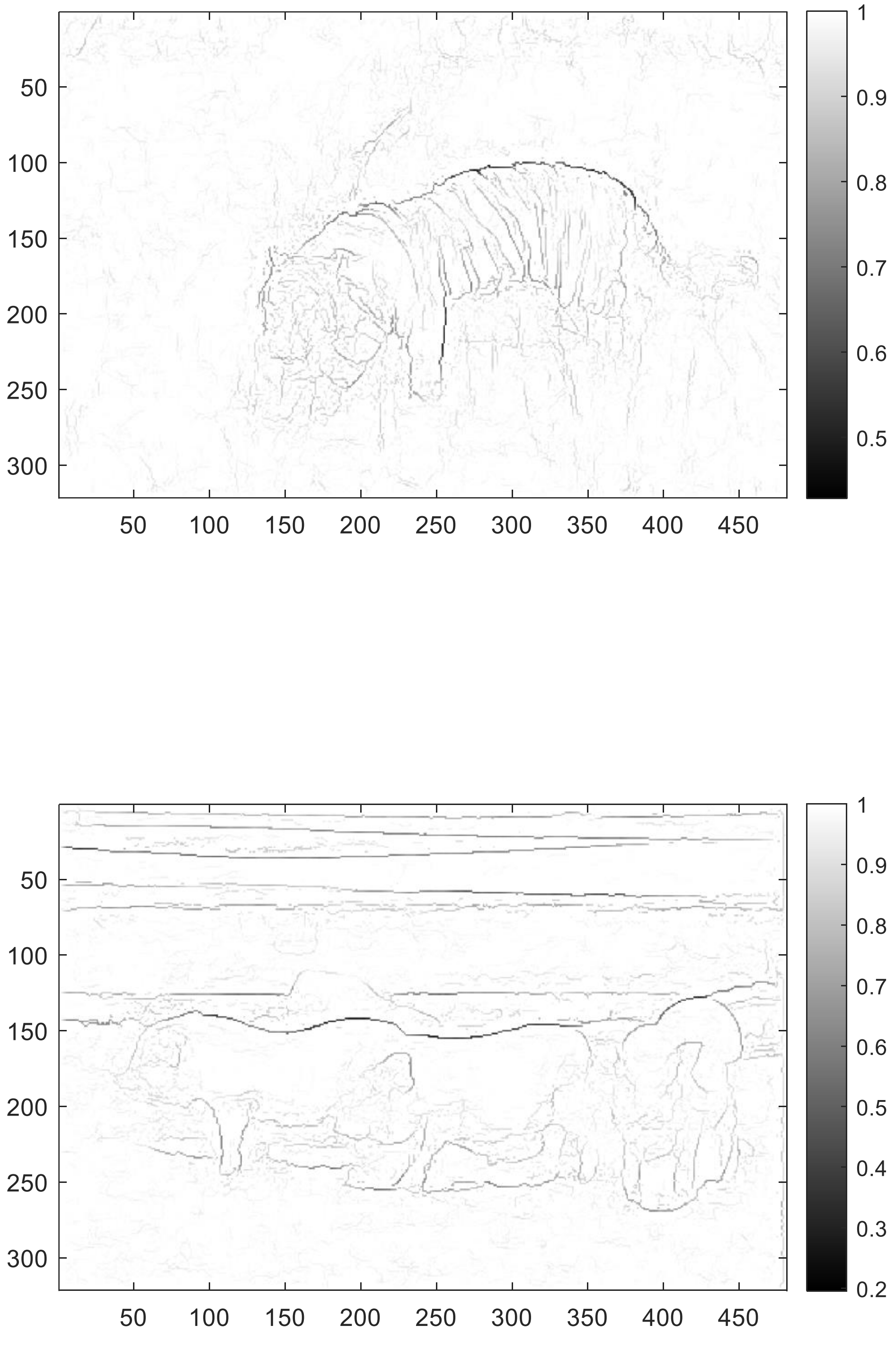
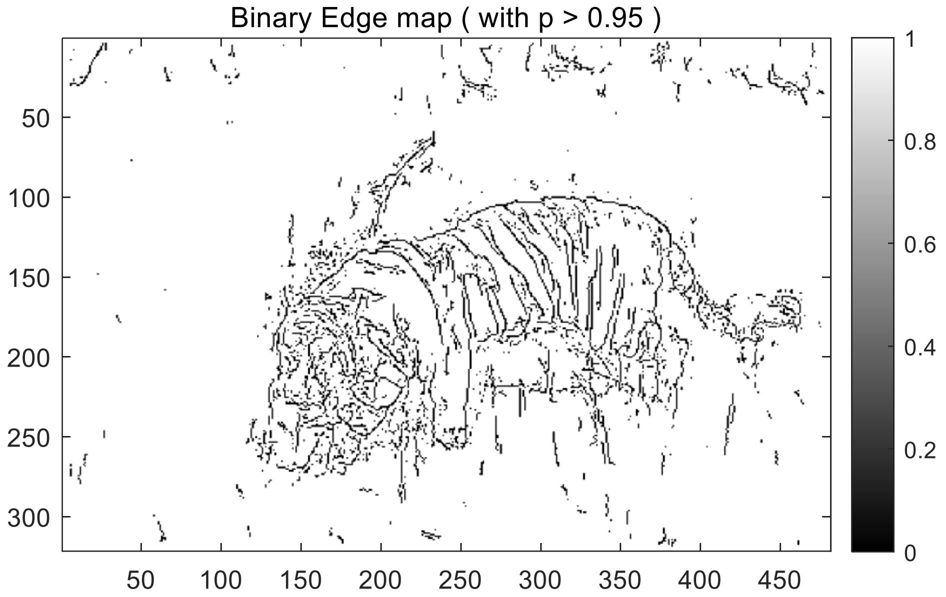


Figure 11: Probability edge map



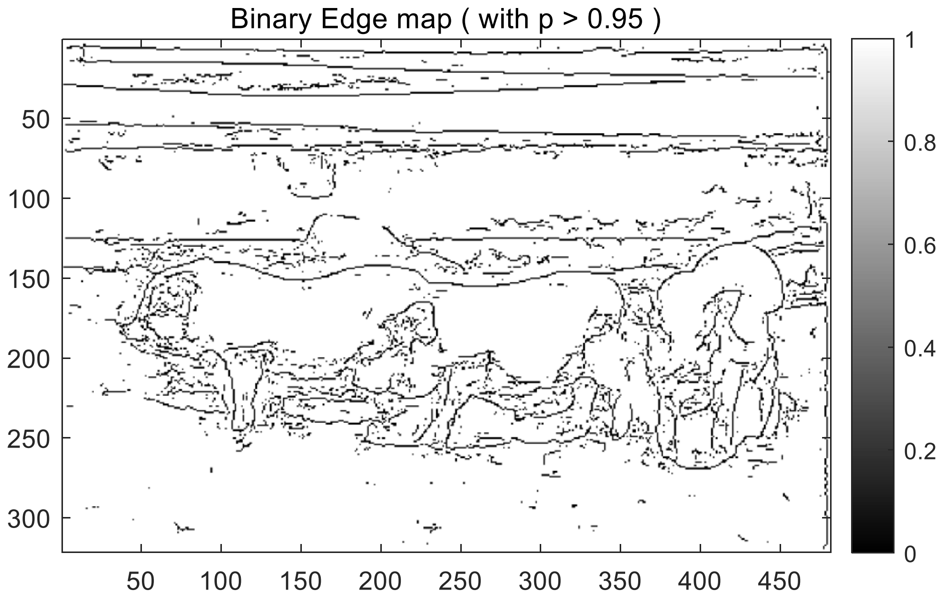


Figure 12: Binary edge map

The chosen parameters:
multiscale=1, for top accuracy.
sharpen=1, so that the edges are made sharp.
nTreesEval=1, for top speed.
nThreads=4.
nms=1, set to true to enable nms, To speed up the process, more threads are employed

for review. When nms = 1, a threshold of 0.95 is a decent choice for highlighting the key edges.

Compared to Canny, the output of the SE detector is finer. The SE detector utilizes many features to come up with an "edge" judgment, while Canny only uses Gaussian filtering to detect edge intersections. The Canny detector uses two thresholds for edge detection, possibly culling desired edges. Compared to Canny, the SE detector is more capable of identifying the edges of important objects. The output of SE can be parallelized, while Canny method cannot.

(d)
(1)

Sobel	Precision	Recall	F measure
Tiger GT	0.1836	0.9831	0.3094

Sobel	Precision	Recall	F measure
Pig GT	0.1862	0.7829	0.3008
Canny	Precision	Recall	F measure
Tiger GT	0.7124	0.8185	0.7618

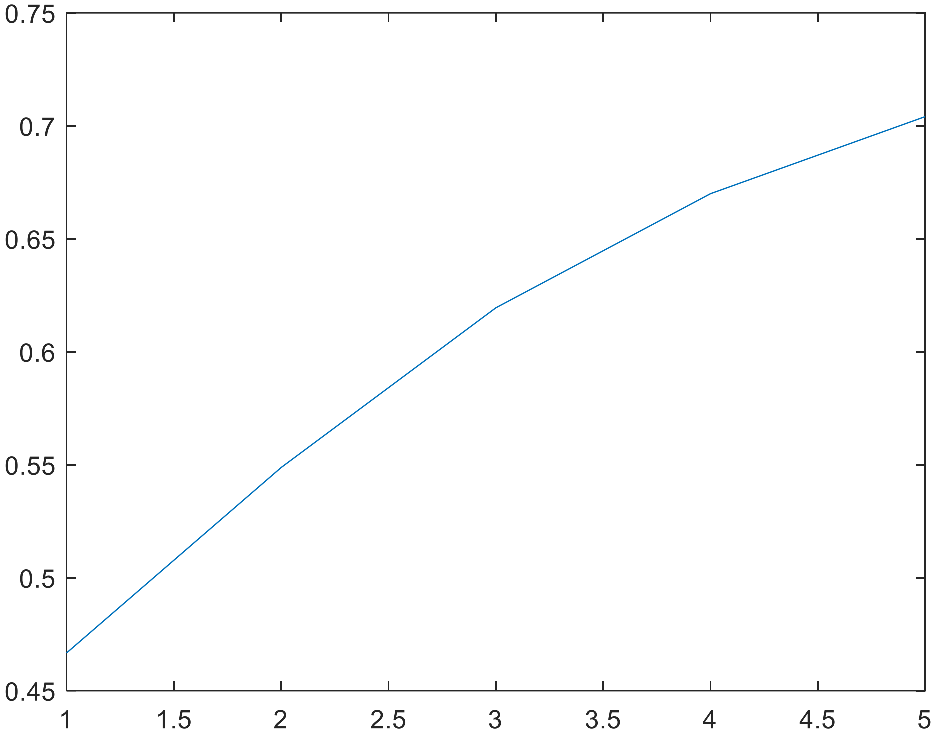
Canny	Precision	Recall	F measure
Pig GT	0.6015	0.6194	0.6103

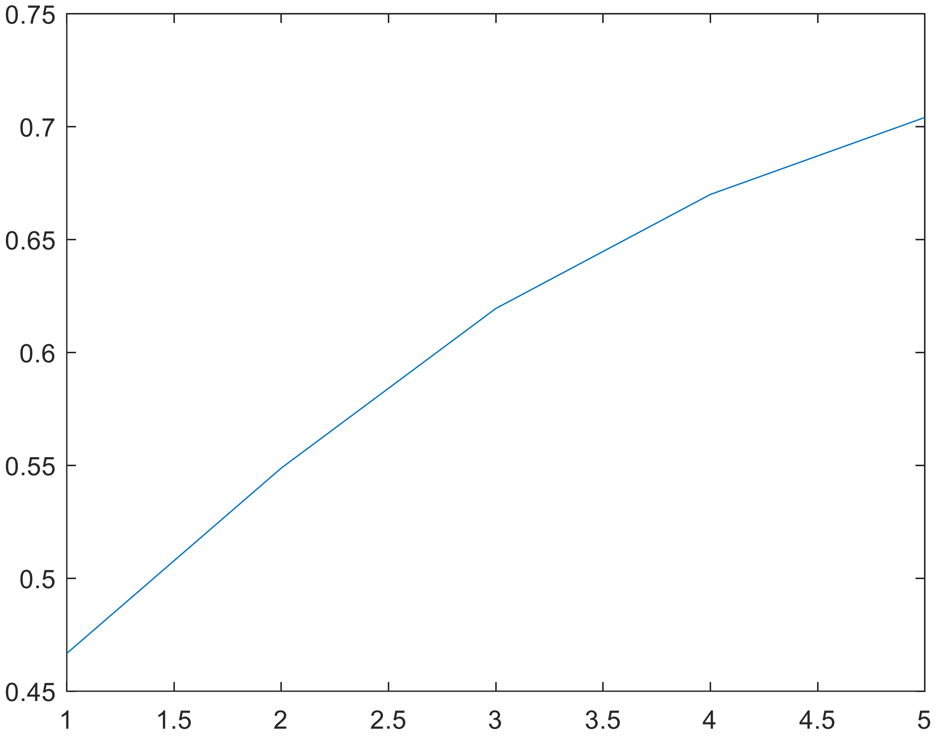
SE	Precision	Recall	F measure
Tiger GT	0.9446	0.1652	0.2812

	Precision	Recall	F measure
Pig GT	0.8964	0.2785	0.4250

We can find that the structured edge method outperforms Sobel and is very close to Canny.

(2)


0.75
0.7
0.65
0.6
0.55
0.5
0.45
1
1.5
2
2.5
3
3.5
4
4.5
5


0.75
0.7
0.65
0.6
0.55
0.5
0.45
1
1.5
2
2.5
3
3.5
4
4.5
5

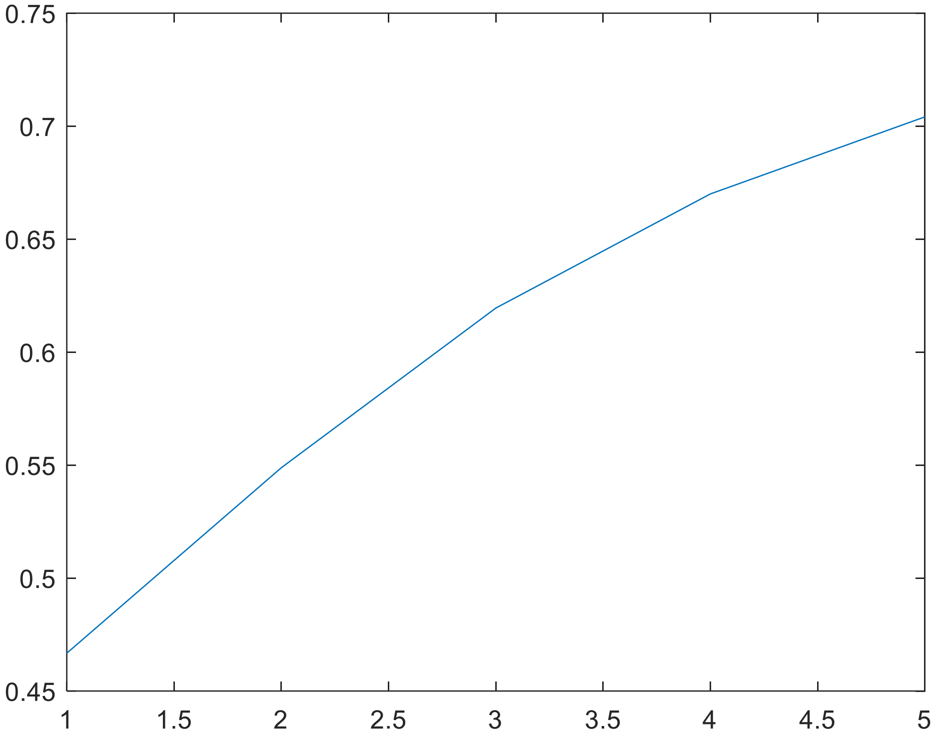
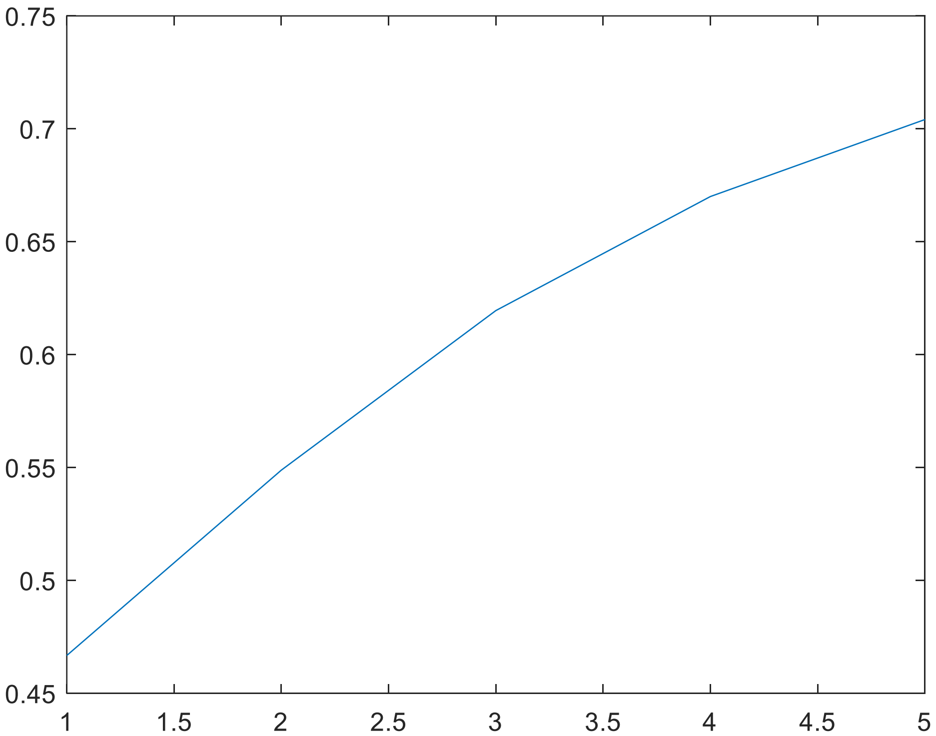


Figure 13: Tiger (Sobel, Canny, SE)



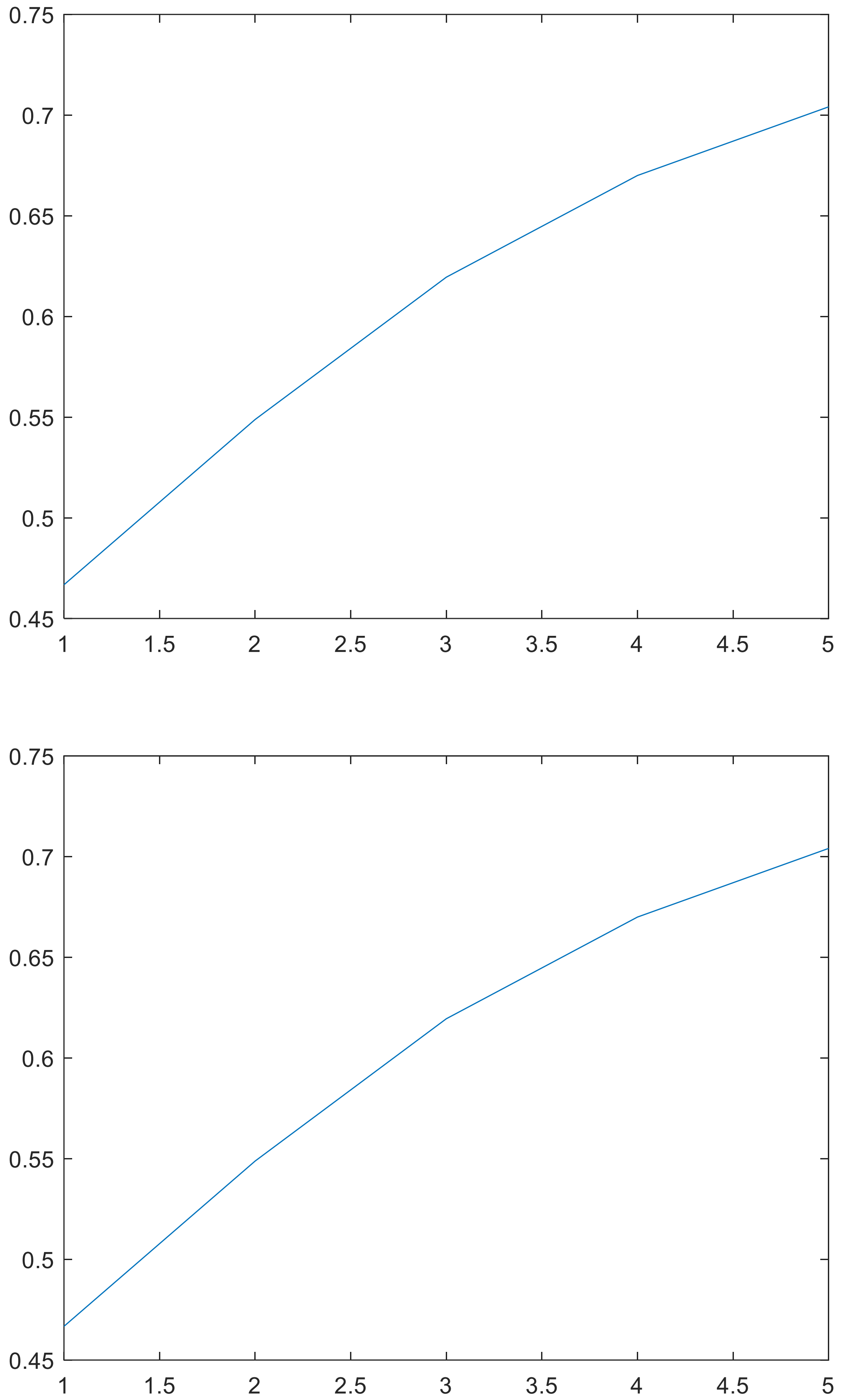


Figure 14: Pig (Sobel, Canny, SE)

Similar to previous results, the structured edge method outperforms Sobel, and is very close to Canny once we apply thresholding. Within the threshold range, both Sobel and structured edges achieve the best results.

(3)
By comparison, tiger image yielded higher F-values on average. The reason may be that there is only one tiger in the center position in the tiger image, but there are three pigs in the pig image, which affects the edge detection.

(4)
Because it is the harmonic mean of accuracy and memory, the F measure penalizes high precision and recall levels. When one is significantly higher or lower than the other, it is difficult to acquire a high F measurement. For instance, if the accuracy is 0 and the recall is 1, the normal mean will offer 0.5 points but the F value will be 0.

If Precision + Recall = constant, $F = \frac{2 \times P \times (constant - P)}{constant}$, when the equation is differentiated and the value is set to 0, the function maximizes P=R because P=constant-P=R. As a result, both accuracy and recall must have the same values in order to maximize the F measure.

1.4 Discussion
The edge map acquired directly by a simple edge operator like Sobel has a number of flaws, including noise contamination and edge lines that are overly thick and broad..

The first step of Canny edge detection uses Gaussian blur to remove the noise, but it also soothes the edges, making the edge information weaker and potentially allowing some needed edges to be missed in later steps, especially weak and isolated edges that may be eliminated in the double threshold calculation. It is naturally predictable that if the radius of the Gaussian blur is increased, the smoothing of the noise is increased, but it also makes the edges in the final edge map obtained significantly less.

Structured Edge has very fast computation. It is more accurate than other method. Texture elements can be omitted. It is appropriate for any type of local structure..

Overall, there is a wide variety of edge detection algorithms with a wide range of applications across many domains.

Problem 2: Digital Half-toning (30%)

2.1 Motivation

Continuous tone image usually refers to an image, the light to dark or light to dark hue change is composed by the density of imaging material particles per unit area, its depth, intensity is showing no polar changes, such as photo negatives, photos, various drawings; while halftone usually refers to the special processing of the print from light to dark or light to dark hue change is expressed by the size of the dot, because the dot in Because the dot is distributed in space at a certain distance and is discrete, and because there is always a limit to the number of levels of screening, it is not possible to achieve the same level of change in the image as the continuous tone image, so it is called halftone image.

Halftone technology is the process of converting a continuous tone image (such as a grayscale or color image) to a binary picture or a color image with a limited number of colors and color quantity, such that the visual impression is comparable to the original image from a distance. As we all know, digital halftone technology is a method of enhancing picture reproduction on monochrome/multi-color binary presentation systems using mathematics and computers, based on visual features of the human eye and image color presentation properties. The low-pass feature of the human eye, which interprets spatially near components of an image as a whole when examined at a given distance, is used in digital halftoning. Using this feature, the average local grayscale of the halftone image viewed by the human eye approaches the original image's average local grayscale value, resulting in an overall continuous tone impression.

The most common taxonomy of halftone technique is classified according to its processing method: dithering method, error diffusion method.

The dithering method is a typical algorithm of the point processing class of methods, mainly divided into two categories: random dithering and ordered dithering. Both techniques involve the usage of a template, also known as a dithering matrix or threshold matrix, which defines the order in which the dots become black as the brightness or grayscale value lowers, as well as the order in which the dots turn black as the luminance or grayscale value decreases. The dithering matrix's design is critical to the dithering process since it determines the quality of the halftone image. The technique is contrasted to the dithering matrix, which takes a range of values between the image's highest and minimum gray values for each threshold value. In 1976 Floyd and Steinberg proposed the error diffusion algorithm, which transitions the halftone screening from "point processing" to "neighborhood processing". The proposed error diffusion algorithm revolutionized halftone screening and was a milestone in halftone technology, and has contributed to the rapid development of halftone technology.

2.2 Approach

2.2.1 Dithering

2.2.1a Fixed thresholding

Assume the image has a value between 0 and 255. T is the value you should choose. If the pixel is smaller than T, set it to 0, else set it to 255. A basic threshold is the least squared error quantizer:

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

This produces a poor-quality rendering of a continuous tone image.

2.2.1b Random thresholding

Threshold that changes with space In the case of Random thresholding, T is a random number generator from a uniform distribution. If a pixel is smaller than random noise, set it to 0, else set it to 255.

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

2.2.1c Dithering Matrix

Turn the pixel "on" in a specific order for a continuous gray level patch. This gives the impression of continuous gray changes. The order to utilize is specified by a N N index matrix.

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

We may build iteratively on this basic matrix to obtain:

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

The index matrix can be converted to a “threshold matrix” or “screen” using the following operation.

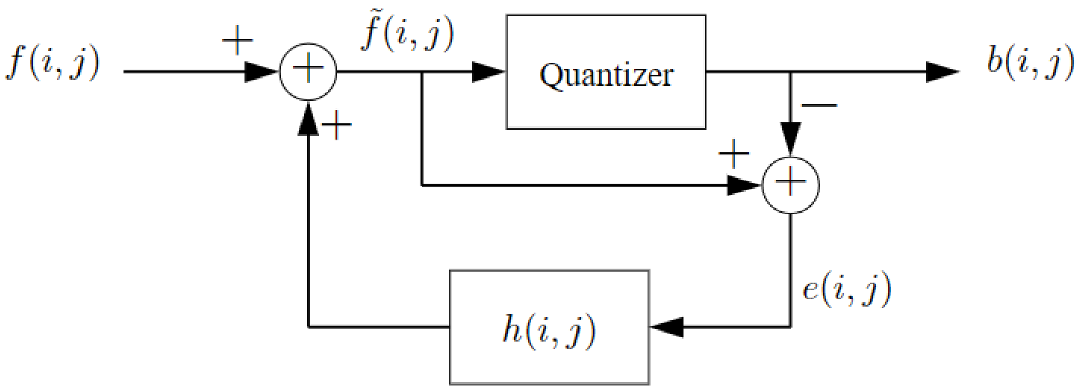
$$T(x,y) = \frac{I_N(x,y) + 0.5}{N^2} \times 255$$

The N × N matrix can then be “tiled” over the image using periodic replication: T (I mod N, j mod N).

The ordered dither algorithm is then applied via thresholding.

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

2.2.2 Error Diffusion



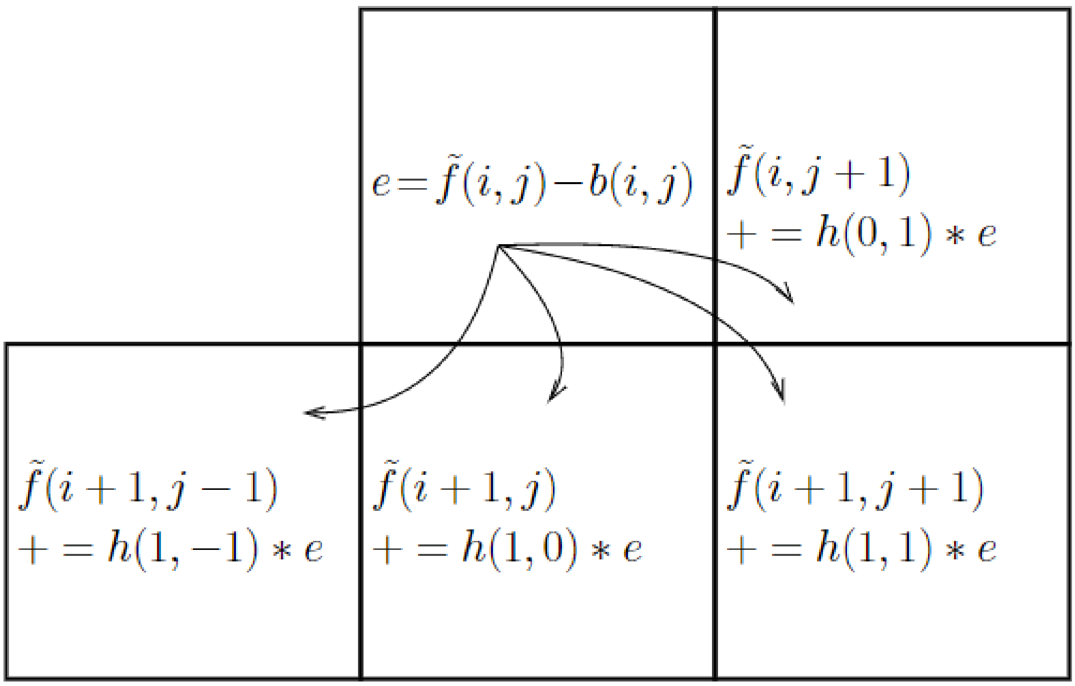
Equations are:

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

$$e(i,j) = \tilde{f}(i,j) - b(i,j)$$

$$\tilde{f}(i,j) = f(i,j) + \sum_{k,l \in S} h(k,l)e(i-k, j-l)$$

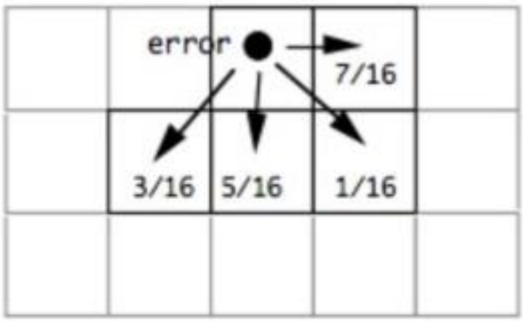
Diffuse error forward using the following scheme:



Here are some commonly used error diffusion Weights:

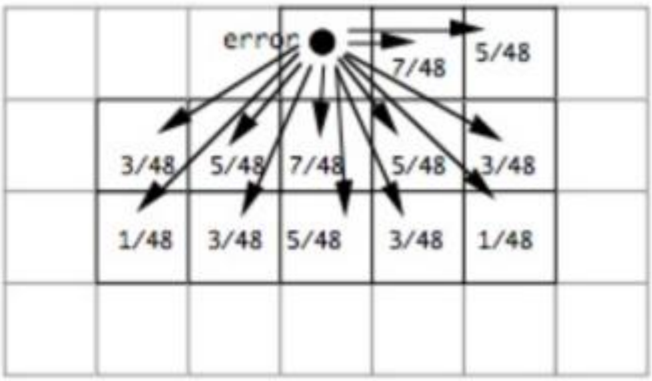
a. Floyd-Steinberg

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



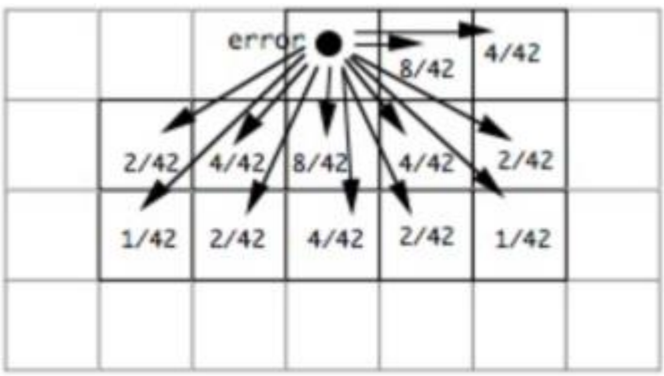
b. JJN

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



c. Stucki

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



2.3 Result

(a)

(1)

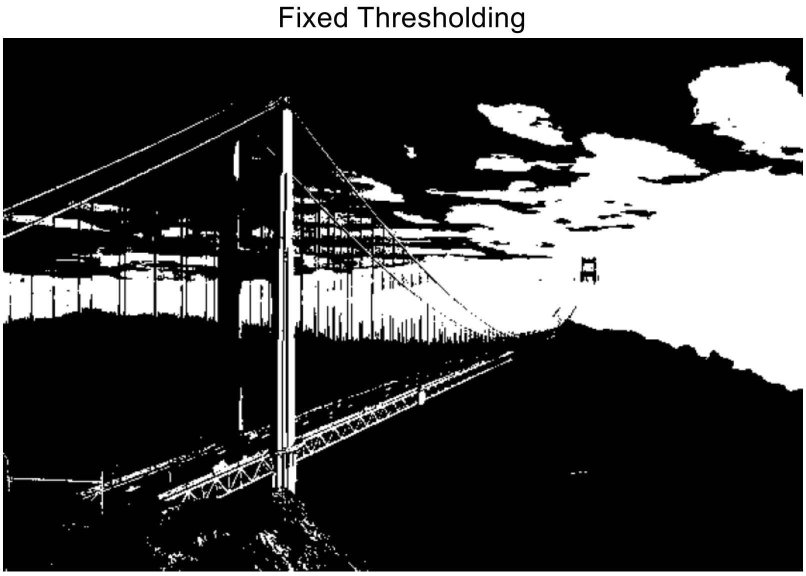


Figure 15: Fixed thresholding

(2)

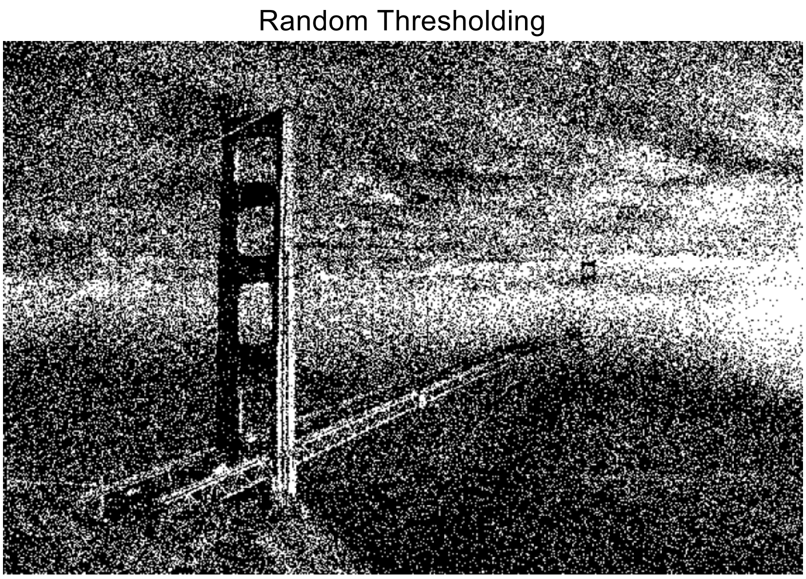


Figure 16: Random thresholding

(3)

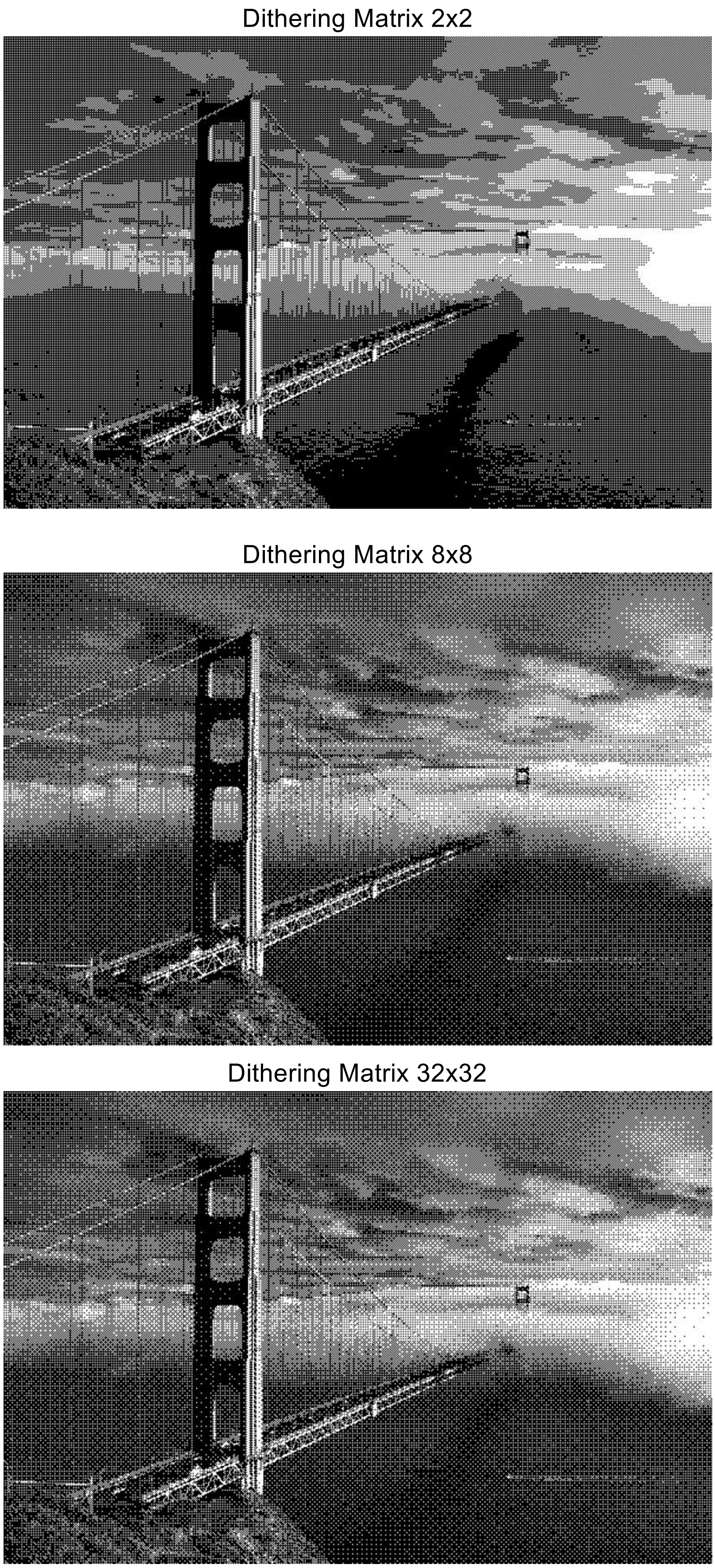


Figure 17: Dithering Matrix

Fixed thresholding algorithm is lack of depth as it is very obvious black and white. Random thresholding algorithm seems to be some variation in the edges and intensities of the input image. However, since we are randomly switching different pixel values on and off, the resulting image is very noisy. Dithering Matrix algorithm can output different results due to different size dithering matrixes. Because a smaller dither matrix allows less error to spread across the image, a bigger 8x8 or 32x32 dither matrix produces somewhat better results.

(b)

(1)

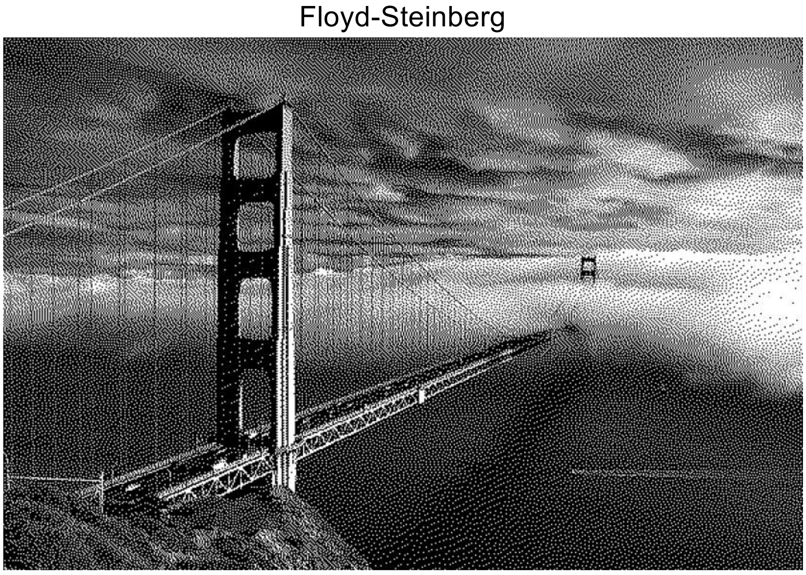


Figure 18: Floyd-Steinberg

(2)

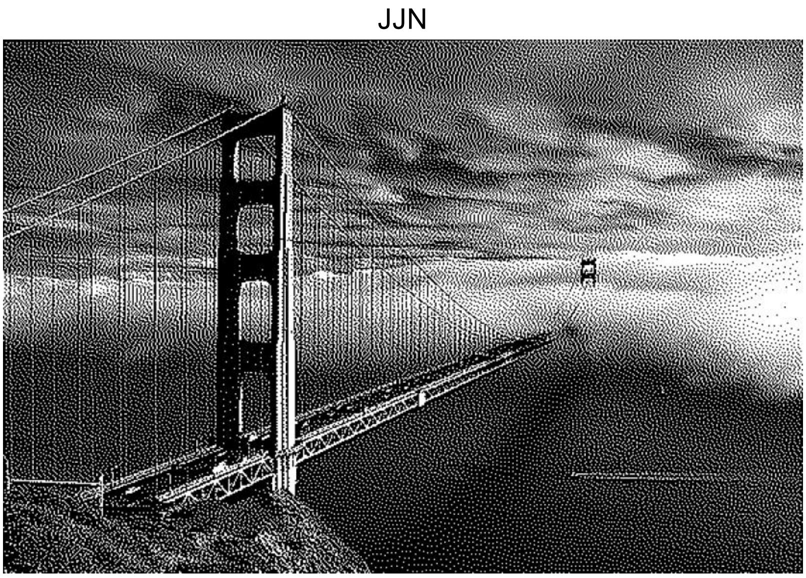


Figure 19: Jarvis, Judice, and Ninke (JJN)

(3)

Figure 20: Stucki

Compare by observation, JJN and Stucki is better than Floyd-Steinberg error diffusion method. But we can not get results with significant gaps when using larger dithering matrices. We can transfer the error to a larger number of neighbors because we can pass the error to a larger number of neighbors. We not only increase a pixel's range, but we also reduce its influence on flipping its neighbors, which might result in additional noise.

Even though this strategy cannot greatly enhance the outcomes, we may utilize a larger matrix to achieve better results. I also believe that walking across a picture utilizing a more complex scanning sequence is a more natural method to do it.

2.4 Discussion

Because the random dithering matrix is created at random, the image quality after halftone is frequently bad, and it is rarely utilized in practice. The ordered dithering matrix, on the other hand, is a regular dithering matrix that is employed by major printer makers for its good visual effect and fast processing speed.

Although the ordered dithering technique is straightforward and produces higher halftone image quality, it has the fatal flaw of incorporating considerable fake textures that repeat. Even if the dithering matrix is precisely built, the output halftone picture still includes faults, and the quality of the halftone image produced by the error diffusion technique is worse.

On the whole, the error diffusion effect is very good. The best halftone results available are still based on error diffusion methods. Error diffusion also has shortcomings. In terms of speed, the median threshold method does only one comparison operation for each pixel, which is fast. In contrast, error diffusion has to perform a large number of multiplication and division operations, which is relatively slow.

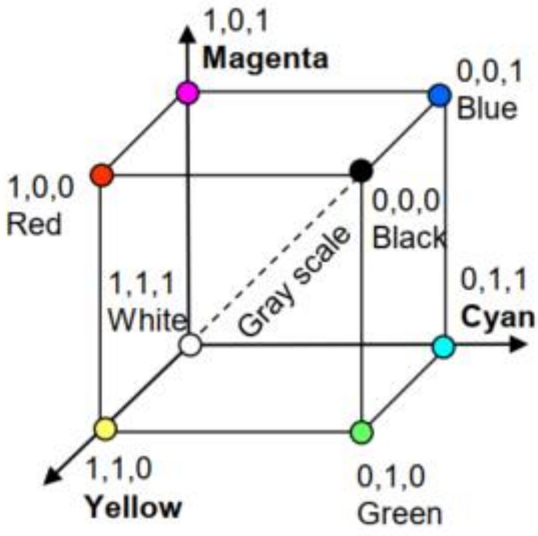
Problem 3: Color Half-toning with Error Diffusion (20%)

3.1 Motivation

Halftoning in grayscale diminishes the resolution of a continuous-tone image for printing or display. Grayscale halftoning by error diffusion forms quantization noise at high frequencies, where the Human Visual System (HVS) is least sensitive. When using a grayscale error diffusion technique on a single colorant plane in color halftones, the HVS response to color noise is not utilized. Ideally, the quantization error should extend to the frequencies and colors to which the HVS is least sensitive. Furthermore, color quantization works best in perceptual space, with the colorant vector chosen as the output color being perceptually closest to the color vector being quantized.

3.2 Approach

CMY Color Space:



3.2.1 Separable Error Diffusion

Do halftoning for each of the CMY channels independently.

(1) Separate an image into CMY three channels.

$$\begin{vmatrix} C \\ M \\ Y \end{vmatrix} = \begin{vmatrix} 1 \\ 1 \\ 1 \end{vmatrix} - \begin{vmatrix} R \\ G \\ B \end{vmatrix}$$

(2) Quantize each channel independently using the Floyd-Steinberg error diffusion technique.

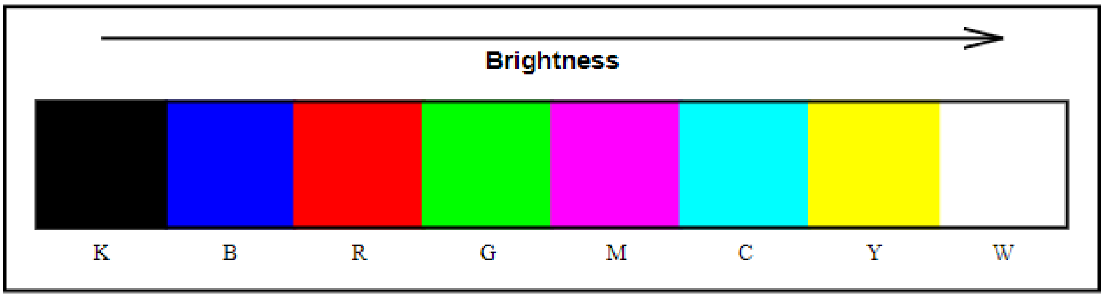
$$\frac{1}{16}\begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 7 \\ 3 & 5 & 1 \end{vmatrix}$$

(3) Error has been corrected Separately, the C, M, and Y channels are converted to R, G, and B channels.

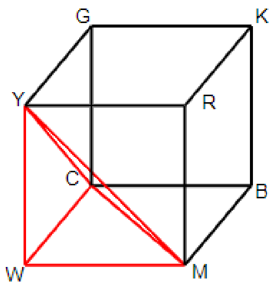
$$\begin{vmatrix} R \\ G \\ B \end{vmatrix} = \begin{vmatrix} 1 \\ 1 \\ 1 \end{vmatrix} - \begin{vmatrix} C \\ M \\ Y \end{vmatrix}$$

3.2.2 MBVQ-based Error diffusion

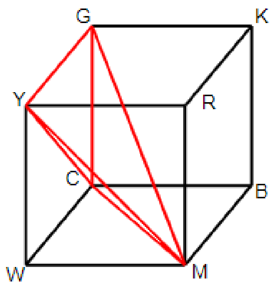
Quantize the color but preserve the energy (brightness):



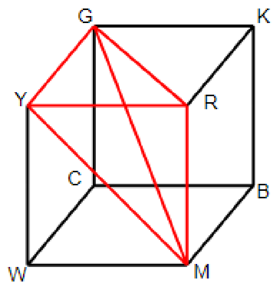
Begin by segmenting the RGB color space into quadrants with the least amount of brightness fluctuation (MBVQs). Each input color should be rendered using one of six complementary quadruples, with minimal brightness fluctuation in each: RGBK, WCMY, MYGC, RGMY, RGBM, or CMGB are all examples of color spaces.



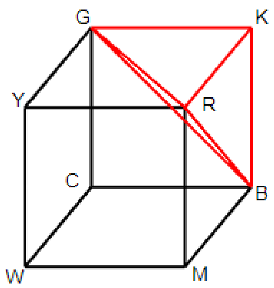
CMYW



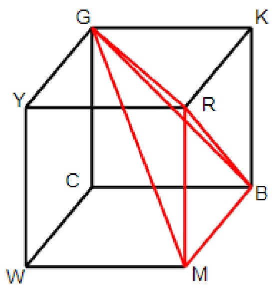
MYGC



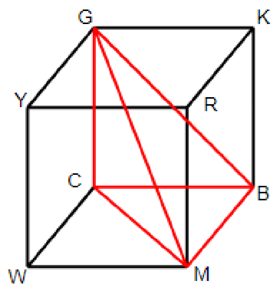
RGMY



KRGB



RGBM



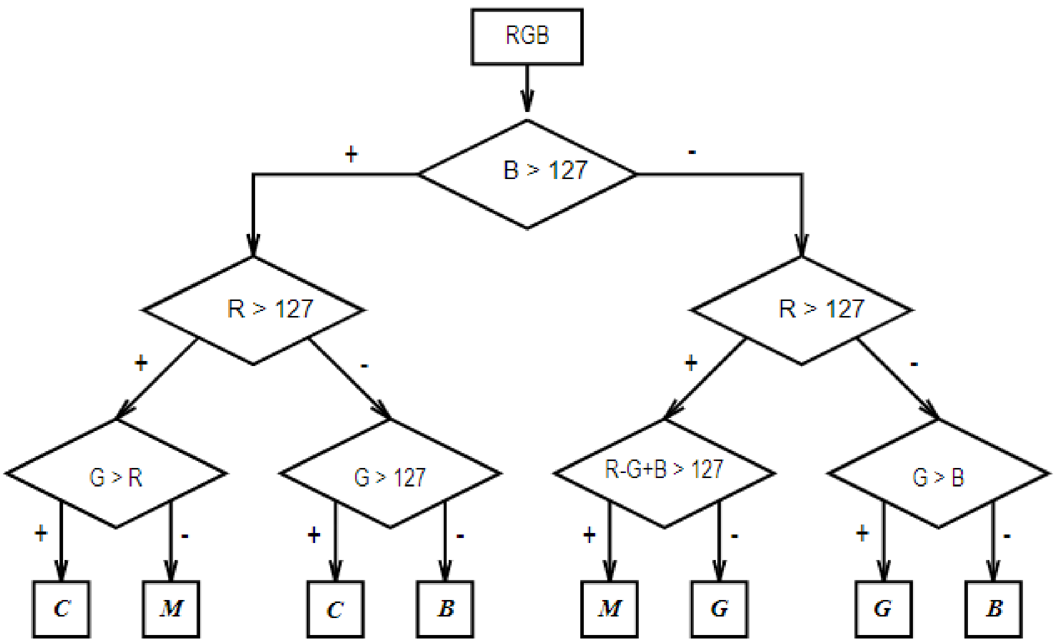
CMGB

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

(1) Calculate MBVQ (RGB (i, j)).

```
pyramid MBVQ(BYTE R, BYTE G, BYTE B)
{
  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;
}
```

(2) Locate the vertex v∈MBVQ that is closest to RGB (i, j) + e (i, j).



(3) Determine the quantization error by RGB (i, j) + e (i, j) - v.

(4) Spread the error through later pixels.

3.3 Results

(a)

Original Image

Figure 21: Original Image

Seperable Error Diffusion

Figure 22: Separable Error Diffusion

The quality of the image does not look good. Its half-toning noise is high. The color dots aren't all the same brightness. It ignores the spatial domain association between RGB planes.

(b)

(1)

To decrease halftone noise, it chooses the halftone set with the least amount of brightness variation from among all halftone sets that may provide the specified color. The spatial domain correlation is used to compute MBVQ quadrilaterals and vertices, since all red, green, and blue pixel intensities at the same various positions are examined simultaneously.

(2)

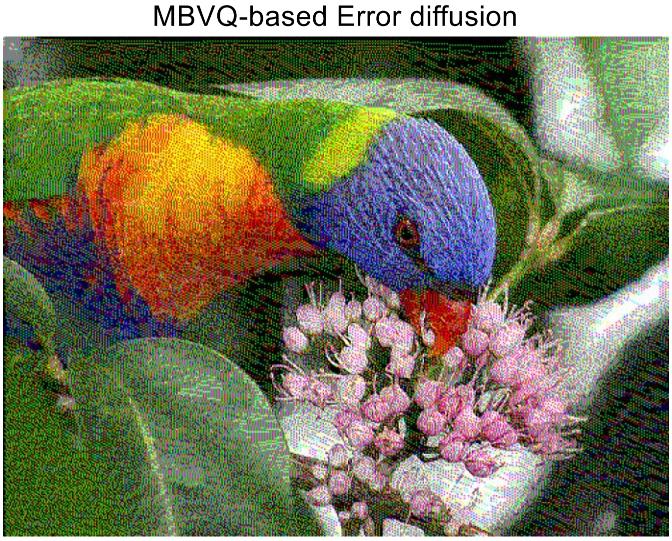


Figure 23: MBVQ-based Error diffusion

MBVQ-based Error Diffusion improves image quality over Separable Error Diffusion. The texture of the bird's feathers has a minor variation in brightness in MBVQ, however the separable error diffusion technique has a considerable difference due to black spots in the separable error diffusion method. The diffusion method differs significantly due to the black dots in the blue area.

3.4 Discussion

From the results we can find that separable error diffusion has its shortcomings. Its image quality is poor, because it operates on each channel separately. MBVQ-based Error diffusion is a superior approach than separable error diffusion. Color and brightness are preserved with MBVQ. The reason for this is that we assume the brightness won't fluctuate greatly in tiny parts of the image. Visual quality is increased because brightness shifts in tiny regions are decreased. Because just four colors are utilized for rendering, halftone noise is reduced. Spatial domain correlation is used because all red, green, and blue pixel intensities at the same locations are examined concurrently to generate MBVQ quads and vertices.