

# **Some Applications of First and Second Derivative Operators in Image Enhancement and Clinical Diagnosis**

*Thesis submitted in the partial fulfillment of the  
requirements for the degree of*

**Master of Technology**

in

**Computer Science and Engineering**

Submitted by

**Kiran Purohit**

Roll Number - 18CS4135

Under the supervision of

**Dr. Dakshina Ranjan Kisku**

Associate Professor

NIT Durgapur



Department of Computer Science and Engineering

National Institute of Technology, Durgapur

M.G. Avenue, A-Zone, Durgapur - 713209

West Bengal, India

June 2020

*I dedicate this thesis to my beautiful family. Especially to my parents for inspiring me to become a better person and teach me the importance of hard work and higher education. To my brother and my project guide for their constant encouragement.*



## Declaration

I certify that

- I. The work presented in this thesis is original and has been done by me under the guidance of my supervisor.
- II. The work has not been submitted to any other Institute for any degree or diploma.
- III. I have followed the guidelines provided by the institute in writing the thesis.
- IV. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
- V. Whenever I have used materials (data, theoretical analysis and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

Date:

Kiran Purohit  
18CS4135



## Certificate of Recommendation

This is to certify that the thesis entitled "**Some Applications of First and Second Derivative Operators in Image Enhancement and Clinical Diagnosis**", submitted by **Kiran Purohit** for the partial fulfilment of the requirements for the award of the degree of **Master of Technology in Computer Science and Engineering with specialization in Computer Science and Engineering**, is a bonafide research work under the guidance of **Dr. Dakshina Ranjan Kisku**. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma. In our opinion, this thesis is of the standard required for the partial fulfilment of the requirements for the award of the degree of **Master of Technology**.

---

**Dr. Dakshina Ranjan Kisku**  
Associate Professor and Project Guide  
Department of Computer Science and  
Engineering National Institute of Technology,  
Durgapur

---

**Dr. Tandra Pal**  
Professor and Head of the Department Department  
of Computer Science and Engineering National  
Institute of Technology, Durgapur



## Certificate of Approval

The foregoing thesis is hereby approved as a creditable study of technological subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite with degree for which it has been submitted. It is to be understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn but approve the thesis only for the purpose for which it has been submitted.

BOARD OF THESIS EXAMINERS

---

---

---

---

---

---

## **Acknowledgement**

I am thankful to numerous local and global peers who have donated towards shaping this thesis. I would like to express my heartfelt thanks to Dr Dakshina Ranjan Kisku for his advice during my thesis work. As my supervisor, he has constantly motivated me to remain concentrated on attaining my aim. His observations and remarks helped me to build the overall direction of the research and to move ahead with an examination in depth. He has assisted me greatly and been a source of knowledge.

I wish to acknowledge Prof. Tandra Pal, HOD of CSE, for providing such a supportive environment where I was offered such opportunities to learn and apply my knowledge in practical fields.

I am thankful to my friends who have helped me with new ideas, useful criticism, their precious time, a welcome ear or kind words.

I would also like to express my gratitude to my parents and brother for standing beside me all the time and supporting me morally and ethically. I must acknowledge the resources that I have got from the institute. I would like to thank the technical and administrative staff members who have been kind enough to help and advice in their respective roles.

At last, I would like to dedicate this thesis to my family, for their understanding, love and patience..

## **Abstract**

Unlike image restoration, image enhancement techniques are found to be subjective in nature as the appropriateness of the appearance of output image depends upon human perception. Hence, it is very difficult to determine the appropriateness of the image enhancement techniques including edge detection prior to an application. This thesis makes use of regression models to determine the suitability of edge detection operators. With the existing operators, a novel Hybrid operator is used in the evaluation. The novel detector is made of combining Canny and Sobel operators with the gradient of the texton image. With this approach, an estimation model as an objective function is determined and further, it is used to determine the degree of proximity (suitability) of the edge operators on two publicly available databases, viz. the BSDS300 and the Multi-cue. The experimental results exhibit that the proposed edge detector outperforms other operators. Sharpening Filters can also be used for data augmentation in detecting Coronavirus using deep learning CNN model. Coronavirus is rapidly increasing and threatening the health of millions of humans. Clinical study shows that it affects the lungs. So, lung infections can be diagnosed with the help of X-Ray and CT Scan images. As deep learning is the most effective and reliable AI technique to classify the COVID-19 screening, we proposed a model which uses Convolutional Neural Network (CNN) fused with the image processing based data augmentation. This application makes use of multiple representations of the same X-Ray and CT scan images, produced through sharpening filters viz. Sobel, Prewitt, Roberts, Scharr, Laplacian, Canny, and Hybrid, are mixed up with visible X-Ray and CT scan images for training the convolutional neural network (CNN) based deep learning model. Our proposed AI application has been tested on publicly available databases of both chest X-Ray and CT Scan images.

***Keywords: Edge detection; Regression models; Filter bank; Pair disk mask; Image gradient, COVID-19, CNN, Deep Learning.***

# Contents

<b>Dedication</b>	ii
<b>Declaration</b>	iii
<b>Certificate of Recommendation</b>	iv
<b>Certificate of Approval</b>	v
<b>Acknowledgement</b>	vi
<b>Abstract</b>	vii
<b>List of Figures</b>	x
<b>List of Tables</b>	xii
<b>List of Abbreviations</b>	xiii
<b>1 Introduction</b>	1
1.1 Sharpening filters and Edge Detection Operators . . . . .	3
1.1.1 Prewitt . . . . .	3
1.1.2 Sobel . . . . .	7
1.1.3 Roberts . . . . .	9
1.1.4 Scharr . . . . .	10
1.1.5 Robinson compass mask . . . . .	11
1.1.6 Kirsch Compass Mask . . . . .	12
1.1.7 Laplacian . . . . .	15
1.1.8 Canny . . . . .	16
1.2 Advantages and Disadvantages of Edge Detection Operator . . . . .	18
1.3 Regression Models . . . . .	19
1.3.1 Linear Regression . . . . .	20
1.3.2 Support Vector Regression . . . . .	20
1.3.3 Multiple Regression . . . . .	21
1.4 Potential applications of Edge Detection Operators . . . . .	22

1.5	Potential uses of Regression Models . . . . .	26
1.6	Thesis Contribution . . . . .	28
1.7	Thesis Organisation . . . . .	28
<b>2</b>	<b>Literature Survey</b>	<b>30</b>
2.1	Comparison of different Edge Detection Operators . . . . .	30
2.2	Detection of Covid-19 on X-Ray and CT Scan images . . . . .	35
2.3	Summary . . . . .	37
<b>3</b>	<b>Determining the Suitability of Edge Detectors using Regression Models</b>	<b>38</b>
3.1	Motivation . . . . .	38
3.2	Proposed Edge Detection Operator . . . . .	39
3.3	Measuring the Appropriateness of Edge Detection Operators . . . . .	41
3.4	Experimental Results and Analysis . . . . .	43
3.4.1	Results . . . . .	44
3.5	Conclusion . . . . .	47
<b>4</b>	<b>COVID-19 Detection on Chest X-Ray and CT Scan Images of Suspected Individuals Using CNN and Image Processing based Data Augmentation</b>	<b>48</b>
4.1	Motivation . . . . .	48
4.2	Proposed Model . . . . .	49
4.3	Experimental Results and Analysis . . . . .	52
4.3.1	Results . . . . .	52
4.4	Conclusion . . . . .	55
<b>5</b>	<b>Conclusion and Future Scope</b>	<b>56</b>
5.1	Conclusion . . . . .	56
5.2	Future Work . . . . .	56
<b>References</b>		<b>58</b>

# List of Figures

1.1	Sample Image [11]	5
1.2	Vertical Edges after applying Prewitt	6
1.3	Horizontal Edges after applying Prewitt	6
1.4	Vertical Edges after applying Sobel	8
1.5	Horizontal Edges after applying Sobel	9
1.6	Horizontal Edges after applying Roberts	10
1.7	Vertical Edges after applying Roberts	10
1.8	Horizontal Edges after applying Scharr	11
1.9	Vertical Edges after applying Scharr	11
1.10	Robinson Compass Mask	14
1.11	Kirsch Compass Mask	16
1.12	Edges after applying Laplacian	17
1.13	Non-maximum Suppression	18
1.15	Edges after applying Canny	19
1.16	Simple Linear Regression [14]	20
1.17	Support Vector Regression [15]	21
1.18	Multiple Regression [16]	22
1.19	Remote Sensing [18]	24
1.20	transmission [19]	24
1.21	Hurdle Detection [20]	25
1.22	Line follower robot [21]	26
2.1	Comparison of Edge Detection operators on Noisy Image (a) Original Image with Noise (b) Sobel (c) Robert (d) Canny [23]	31
2.2	A simulated gradient image. (b) Edges attained by the nonmaxima suppression of Canny. (c) The minor edges. (d) Edges achieved by the amended Canny edge detector [24]	32
2.3	Edge Transformative Derivation [26]	32
2.4	Edge detection from a grayscale representation of an image. [27]	33
2.5	Edge detection using the Canny edge detection operator. (a) original image; (b) edge image [28]	34

2.6	Types of edge detector [29] . . . . .	34
2.7	Diagram of proposed model by Jianpeng Zhang, Yutong Xie [32] . . . . .	35
3.1	Design Process of the Proposed Edge Detection Operator . . . . .	39
3.2	RMSE plot on BSDS300 and Multi-cue datasets (a) Linear Regression on BSDS300 dataset (b) Support Vector Regression on BSDS300 dataset (c) Multiple Regression on BSDS300 dataset (d) Linear Regression on Multi-cue dataset (e) Support Vector Regression on Multi-cue dataset (f) Multiple Regression on Multi-cue dataset . . . . .	47
4.1	Chest X-Ray and its Multiple Representations of a COVID-19 Infected Individual. . . . .	50
4.2	Chest CT scan and its Multiple Representations of a COVID-19 Infected Individual. . . . .	50
4.3	Convolutional Neural Network of Layer Size 32, 64 and 128. . . . .	51
4.4	Confusion Matrix of our model when train and test ratio is 70:30 . . . . .	53
4.5	ROC curves . . . . .	55

# List of Tables

1.1	Prewitt Filter . . . . .	4
1.2	Sobel Filter . . . . .	7
1.3	Weighted Sobel . . . . .	9
1.4	Roberts Filter . . . . .	10
1.5	Scharr Filter . . . . .	11
1.6	Robinson Compass Mask . . . . .	13
1.7	Kirsch Compass Mask . . . . .	15
1.8	Commonly used Laplacian Kernels . . . . .	15
1.9	Advantages and Disadvantages of Edge Detectors . . . . .	19
3.1	RMSE using Linear Regression on BSDS300 dataset . . . . .	45
3.2	RMSE using Support Vector Regression on BSDS300 dataset . . . . .	45
3.3	RMSE using Multiple Regression on BSDS300 dataset . . . . .	45
3.4	RMSE using Linear Regression on Multi-cue dataset . . . . .	46
3.5	RMSE using Support Vector Regression on Multi-cue dataset . . . . .	46
3.6	RMSE using Multiple Regression on Multi-cue dataset . . . . .	46
4.1	Results of proposed models with Original images and Data Augmentation with CT Scan and X-Ray images datasets . . . . .	54

# List of Abbreviations

<b>RMSE</b>	Root Mean Square Error
<b>LBP</b>	Local Binary Pattern
<b>RBF</b>	Radial Basis Function
<b>SVR</b>	Support Vector Regression
<b>UV</b>	UltaViolet
<b>BSDS</b>	The Berkeley Segmentation Dataset and Benchmark
<b>AI</b>	Artificial Intelligence
<b>CT</b>	X-ray Computed Tomography
<b>CNN</b>	Convolutional Neural Network
<b>COVID-19</b>	CoronaVirus Disease

# **Chapter 1**

## **Introduction**

Unlike image restoration, image enhancement techniques are found to be subjective in nature as the appropriateness of the appearance of output image depends upon human perception. Hence, it is very difficult to determine the appropriateness of the image enhancement techniques including edge detection prior to an application. Regression models are used here to determine the suitability of edge detection operators.

Edge detection [1] plays a vital role in image enhancement techniques. Such techniques are usually subjective to the user. In simple words, the output provided by the edge detection operators will be dependent on the perception of the viewer. Different users will judge the output provided by the edge detection operators variably and have different views. We want to avoid this difference in opinion and obtain a basis where perception of an output provided by the edge detection operators is uniform even with different users. Edge detection operators are subjective in nature, and there is no such basis function available to determine the best suitable operator for a particular application. Due to the subjective nature of edge detection operators, it is very difficult to predict the optimal edge detector while the appropriateness of edge detection output to be decided visually as the perception varies from user to user. Moreover, a user perceives the outcome of edge detection function differently and decides the suitability after having some cognitive responses [2] generated from the human vision system. Hence, it is cumbersome to obtain the same edged effect from a set of edge detection operators used for some specific application.

Edge detection is an essential part of the image processing technique. It plays a vital role in low-level feature extraction or finding shape information about objects. In edge detection, an edge is a sharp discontinuity change across the gray level boundaries. This discontinuity may include a line, an isolated point, and an edge. It consists of various image processing techniques that are used to identify edge points in an image where the brightness of the image changes more sharply and has discontinuities. Edges occur at image locations that represent the object boundaries, which are by any change in the gray levels at these object boundaries. These object boundaries also play a vital role in many computer vision algorithms such as Edge Detection based face/object recognition, Image Compression, etc. Edges are detected to filter out comparatively less critical and smaller details, to enhance the processing speed,

lowering the complexity without losing the required data. Generally, the performance of the edge detection is dependent on how well edge detector markings match with the visual perception of object boundaries.

In order to determine the suitability of edge detection operators [1] such as Sobel, Canny, Prewitt, Roberts, Scharr, Laplacian, and Hybrid edge detectors on variety images prior to an application using regression models [3–6]. The subjective nature of edge detection operators are transformed to objective functions while regression models compute best-fit function for each operator.

A robust methodology is presented to measure the degree of proximity of the edge detection operators using regression models [3–6]. In order to achieve the objective, the root mean square error (RMSE) [7] can be calculated for the respective edge detector. RMSE is directly related to the strength of the discontinuity in the image caused by the changes in the gray levels. In simple words, more the RMSE for a edge detector, more sharp the discontinuity to be found in the image. This helps us to choose mathematically the most optimal edge detector prior to an application.

Barely, comparing the RMSE of different edge detection operators does not give the correct proximity about appropriateness of an edge detection operator. Now, in this situation a mechanism needs to be introduced to find an optimal edge detection operator that measures the sharp discontinuity present in the image more accurately without overlapping with irrelevant information. In this work, a set of regression models such as Linear, Support Vector, and Multiple regression models are used.

These regression models find an optimal best-fit function in the cloud of response values determined from individual edge detection operator. This optimal best-fit function is referred as regression line, regression curve, and regression plane in case of Linear, Support Vector, and Multiple regression models respectively. This mechanism would be found useful when the suitability of the edge detection operator would be verified prior to an application. Moreover, this mechanism facilitates the entire process to transform a subjective image processing technique to objective function where the dependency on the user's perspective is minimized.

Besides, determining the suitability of edge detection operators, a novel edge detection method is proposed called Hybrid edge detector. It is a combination of weighted Canny image and weighted Sobel image combined with the texton gradient image [8–10]. The proposed edge detection operator is used along with the existing operators for experiments while keeping the experimental protocol uniform for all operators. As these edge detection operators exhibit sharp discontinuities after applying on an image, therefore it is assumed to obtain the edged effect related to structural information, which would make the difference in the appearance of the image.

Another application of Sharpening Filters is that, it can be used for data augmentation in case of detecting Coronavirus. Coronavirus disease or COVID-19 is an infectious

disease which came to light on December 31, 2019 when China informed to World Health Organization (WHO) about a pneumonia like infection due to unknown cause observed among people in Wuhan city of Hubei province in China. The coronavirus outbreak has so far infected millions of people and deaths are increasing day by day. Due to deadly infectious nature of coronavirus, it is spreading rapidly among people who are exposed to COVID-19 infected individuals. The virus spreads through droplets of saliva or discharge of swab from the nose while a COVID-19 infected person coughs or sneezes. A COVID-19 infected person may experience dry cough, fever, headache, muscle pain, sore throat and mild to moderate respiratory illness. However, older people and those having underlying medical conditions like cardiovascular disease, diabetes, chronic respiratory disease and cancer are more exposed to develop serious illness.

Coronavirus is rapidly increasing and threatening the health of millions of humans. Clinical study shows that it affects the lungs. So, lung infections can be diagnosed with the help of X-Ray and CT Scan images. As deep learning is the most effective and reliable AI technique to classify the COVID-19 screening, we proposed a model which uses Convolutional Neural Network (CNN) fused with the image processing based data augmentation. This application makes use of multiple representations of the same X-Ray and CT scan images, produced through sharpening filters viz. Sobel, Prewitt, Roberts, Scharr, Laplacian, Canny, and Hybrid, are mixed up with visible X-Ray and CT scan images for training the convolutional neural network (CNN) based deep learning model.

## 1.1 Sharpening filters and Edge Detection Operators

Sharpening of an image increases the contrast between dark and bright regions to bring out the features. The sharpening process is basically an application of a high pass filter to an image. Sharpening is just opposite to the blurring. In case of blurring, we reduce the edge content and in Sharpening, we increase the edge content.

Most of the information of the shape of an image is enclosed in edges. So first, edges are detected in an image by using filters given below and then by enhancing those edges, image sharpness will increase and image will become clearer. Here are some of the edge detection operators.

### 1.1.1 Prewitt

Prewitt edge detector is applied for detection of edges in a picture. Following kinds of edges are identified by Prewitt Operator:

- Vertical
- Horizontal

Edges are computed by the difference among the corresponding pixel values of a picture. The filters applied for edge detection are termed as derivative filters. A picture is a signal, so any differences in a signal can be computed using differentiation. Therefore the edge detectors are known as derivative filters or derivative operators.

The derivative filters must obey the below characteristics:

- The filter should contain opposite sign.
- Filter's sum should result zero.
- Higher weight symbolizes higher edge detection.

Prewitt edge detector consists of two filters one for edge detection vertically and another for edge detection horizontally.

-1	-1	-1
0	0	0
1	1	1

(a)  $G_x$  (Horizontal)

-1	0	1
-1	0	1
-1	0	1

(b)  $G_y$  (Vertical)

Table 1.1: Prewitt Filter

### 1.1.1.1 Horizontal Mask

Above mask in Table 1.1a finds edges horizontally as the zeros column lies in the horizontal direction. Convoluting this filter with a picture highlights the edges in the horizontal direction in the picture.

- **How it works**

This filter highlights the edges in the horizontal direction in a picture. It operates on the law of the above filter, and it computes the difference among the pixel values of a specific edge. The central row of the filter comprises of zeros. Hence it does not constitute the actual values of an edge in a picture. Instead, it computes the difference of below and above pixel values of the specific edge. Hence it increments the swift change of the value of intensities and produces the edge more noticeable. Both the filters obey the law of derivative filter, consists an opposite sign and both filters sum results zero. The third provision will not be applicable in this filter as both are standard filters, and we can not modify the content in them.

### 1.1.1.2 Vertical Mask

Above mask in Table 1.1b finds the edges vertically as the column containing zero is in the vertical direction. Convoluting this filter with a picture gives the edges in the vertical direction in a picture.

- **How it works**

Applying this filter on a picture will highlight edges in a vertical direction. It merely works as a first-order derivative and computes the difference of pixel values in an edge region. The central column consists of zero, so, it does not constitute the actual values of a picture. Instead, it computes the difference between left and right pixel intensities about that edge. It enhances the intensity of the edge, and it becomes improved compared to the original picture.

### 1.1.1.3 Gradient Magnitude

The gradient magnitude is given in Equation 1.1.

$$G = \sqrt{G_x^2 + G_y^2} \quad (1.1)$$

Generally, an estimated magnitude is calculated with the help of Equation 1.2 which computes faster.

$$G = |G_x| + |G_y| \quad (1.2)$$

The edge's angle of orientation is shown in Equation 1.3

$$\Theta = \tan^{-1} \left( \frac{G_y}{G_x} \right) \quad (1.3)$$

### 1.1.1.4 Sample Image

Sample picture is given in Fig 1.1 on which we will apply Prewitt edge detection operator.

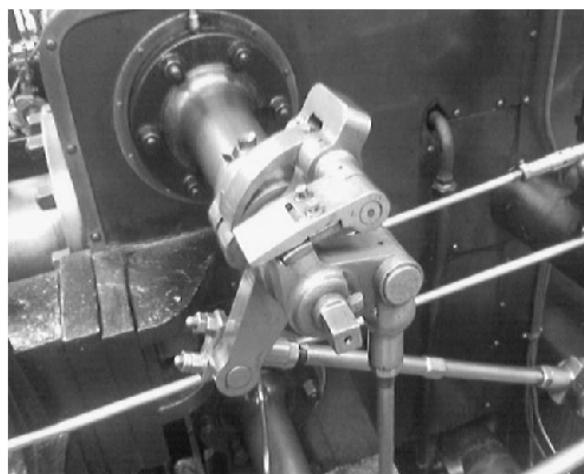


Figure 1.1: Sample Image [11]

- **After applying Vertical Mask**

Applying a vertical filter on the sample picture, we will get the below picture in

Fig 1.2. This picture includes edges in a vertical direction. We can see it more accurately by contrasting with edges in the horizontal direction in the image.



Figure 1.2: Vertical Edges after applying Prewitt

- **After applying Horizontal Mask**

After convolving the horizontal filter with the sample image, we will obtain the below picture in Fig 1.3.



Figure 1.3: Horizontal Edges after applying Prewitt

#### 1.1.1.5 Comparison

After using a vertical filter on the input picture, all the edges in the vertical direction are more noticeable than the original picture. Likewise, using a horizontal filter, all the edges in the horizontal direction are noticeable. Hence we can find both edges in a horizontal and vertical direction from a picture.

## 1.1.2 Sobel

The Sobel mask is quite comparable to the Prewitt mask. Sobel filter is also a derivate filter, and it is applied for detecting edges. Same as Prewitt mask, Sobel mask is also applied to find two types of edges in a picture:

- Vertical Mask
- Horizontal Mask

### 1.1.2.1 Difference with Prewitt Operator

The critical distinction is that in Sobel edge detector the coefficients of the filter are not fixed and it can be set according to our necessity unless they do not disrupt any characteristic of derivative filters.

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

(a)  $G_x$  (Horizontal)

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

(b)  $G_y$  (Vertical)

Table 1.2: Sobel Filter

### 1.1.2.2 Vertical Mask

Above mask in Table 1.2b operates identically as the Prewitt vertical filter. The unique distinction is that in the middle of the third and first column, it contains “2” and “-2” values. When this filter is convolved with a picture, it highlights the vertical edges.

- **How it works**

Using this filter on a picture highlights the vertical edges. It operates as a first-order derivative, and it determines the difference between the value of the pixel in an edge region.

As the middle column holds zero, so it does not cover the real values of a picture, instead, it computes the difference of right and left pixel intensities about that edge. The value of the centre pixel of the first and third column contains 2 and -2, respectively.

It provides more considerable weightage to the value of the pixel about the edge region. It enhances the intensity of the edge and it gets more enhanced compared to the original picture.

### 1.1.2.3 Horizontal Mask

Above mask in Table 1.2a detects edges horizontally as the column containing zeros lies in the horizontal direction. Convolving this filter with a picture highlight the edges in the

horizontal direction in a picture. The unique distinction is that it contains 2 and -2 as a central element of first and the third row.

- **How it works**

This filter will highlight the edges in the horizontal direction in a picture. It operates on the law of the derivative filters and computes the difference between the pixel values of a specific edge. It does not contain the real values of an edge in a picture as the centre row of filter comprises of zeros. Instead, it computes the difference of below and above pixel values of the specific edge. Therefore improving the immediate change of intensity values and obtaining the edge extra noticeable.

#### 1.1.2.4 Sample Image

Sample picture is given in Fig 1.1 on which we will apply Sobel edge detection operator.

- **After applying Vertical Mask**

Applying vertical filter on the sample Fig 1.1, we will get the below picture in Fig 1.4.



Figure 1.4: Vertical Edges after applying Sobel

- **After applying Horizontal Mask**

Applying horizontal filter on the sample figure, we will get the below picture in Fig 1.5.

#### 1.1.2.5 Comparison

You can view that in the image we applied a vertical filter all the edges in the vertical direction are more noticeable. Likewise, in the next picture, we have applied the horizontal filter. Hence all the horizontal edges are noticeable.

Hence, we can find both vertical and horizontal edges from a picture. If we examine the output of the Sobel filter with the Prewitt filter, we will notice that the Sobel filter detects more edges or produce edges more noticeable as contrasted to Prewitt filter.

It is of the fact that the Sobel filter allocates more weight to the pixel values about the edges.



Figure 1.5: Horizontal Edges after applying Sobel

#### 1.1.2.6 Applying more weight to mask

We can notice that if we give more weight to the filter, more edges will be shown. As stated in the beginning that there are no specific coefficients in the Sobel filter. Therefore one another weighted mask is given in Table 1.3.

-1	0	1
-5	0	5
-1	0	1

Table 1.3: Weighted Sobel

Comparing the output of this filter with the Prewitt vertical filter, it is evident that this filter will provide more edges as contrasted to Prewitt filter as more weight in the mask is allotted in Sobel.

#### 1.1.3 Roberts

The Roberts Cross edge detector does an easy, fast to calculate, 2 Directional spatial gradient estimation on a picture. Values of the pixel at every point in the result signify the calculated absolute measure of the spatial gradient of the input picture at that position. The mask contains a pair of convolution masks of  $2 \times 2$  size as given in Table 1.4. One mask is just the other turned by  $90^\circ$ . It is quite comparable to the Sobel mask. These masks are made to react maximally to the edges at  $45^\circ$  to the grid of pixels, each mask for one of the perpendicular orientations. The masks can be used independently to the input picture, to generate different amounts of the gradient component per orientation ( $G_x$  and  $G_y$ ). These can be merged to determine the absolute magnitude of the gradient and orientation of that gradient at every point. The horizontal and vertical edges after applying Roberts is given in Fig 1.6 and 1.7

respectively.

-1	0
0	-1

(a)  $G_x$ 

0	-1
1	0

(b)  $G_y$ 

Table 1.4: Roberts Filter



Figure 1.6: Horizontal Edges after applying Roberts



Figure 1.7: Vertical Edges after applying Roberts

### 1.1.4 Schar

The Scharr edge detection operator derives all the edges in a picture, despite their direction. It is applied using the sum of the two-directional edge improvement operations. The mask contains a set of  $3 \times 3$  convolution mask as given in Table 1.5. One mask is just the other turned by  $90^\circ$ . The masks can be used independently to the input picture, to provide different

measurements of the gradient component in both orientation ( $G_x$  and  $G_y$ ). The horizontal and vertical edges after applying Scharr is shown in Fig 1.8 and 1.9 respectively.

-3	-10	-3
0	0	0
3	10	3

(a)  $G_x$ 

-3	0	3
-10	0	10
-3	0	3

(b)  $G_y$ 

Table 1.5: Scharr Filter



Figure 1.8: Horizontal Edges after applying Scharr



Figure 1.9: Vertical Edges after applying Scharr

### 1.1.5 Robinson compass mask

Robinson compass filters are also a type of derivate filter used for detecting edges. This filter is also called as the direction filter. In this filter, one filter is taken and then rotated in every eight directions of the compass given below:

- North
- North West
- West
- South West
- South
- South East
- East
- North East

Here no standard filter is there. One can take any filter, and it is rotated to detect edges in the directions mentioned above. Every filter is rotated depending on the zero columns' direction.

For instance, let's examine the below filter which lies in North Direction and then it is rotated to obtain the filter in every direction shown in Table 1.6.

We can observe that every direction is obtained based on the direction of zeros. Every filter provides us with the edges on their direction. Let's examine the output of the above filters.

We can observe in Fig 1.10 that by using all the above filters, we will obtain edges in every direction. The output also relies on the picture. If there is a picture, which does not possess edges in North-East direction so then that filter will be useless.

### 1.1.6 Kirsch Compass Mask

Kirsch Compass filter is a derivative filter applied for finding edges. It is same as Robinson compass in terms of detecting edges in every eight directions of a compass. The single distinction in Robinson and kirsch compass filter is that in Kirsch, we have a conventional mask, but in Robinson, we switch the filter according to our necessities.

Using Kirsch Compass Filter, one can detect edges in the below eight directions.

- North
- North West
- West
- South West
- South

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

(a) North Direction Mask

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

(b) North West Direction Mask

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

(c) West Direction Mask

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

(d) South West Direction Mask

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

(e) South Direction Mask

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

(f) South East Direction Mask

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

(g) East Direction Mask

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

(h) North East Direction Mask

Table 1.6: Robinson Compass Mask

- South East
- East
- North East

Here a conventional filter is taken which supports all the features of a derivative filter and then it is rotated to detect the edges.

For instance, let's observe the below filter which lies in North Direction and then it is rotated to obtain the masks in every eight directions in Table 1.7.

Here one can observe that every direction is included, and every filter will provide you with the edges of their direction. To completely get the idea of these filters, it is applied to an actual picture.

As you can see in Fig 1.11 that by applying all the above filters, one will obtain edges in every eight directions. The output also relies on the picture. Let us assume there is a picture,

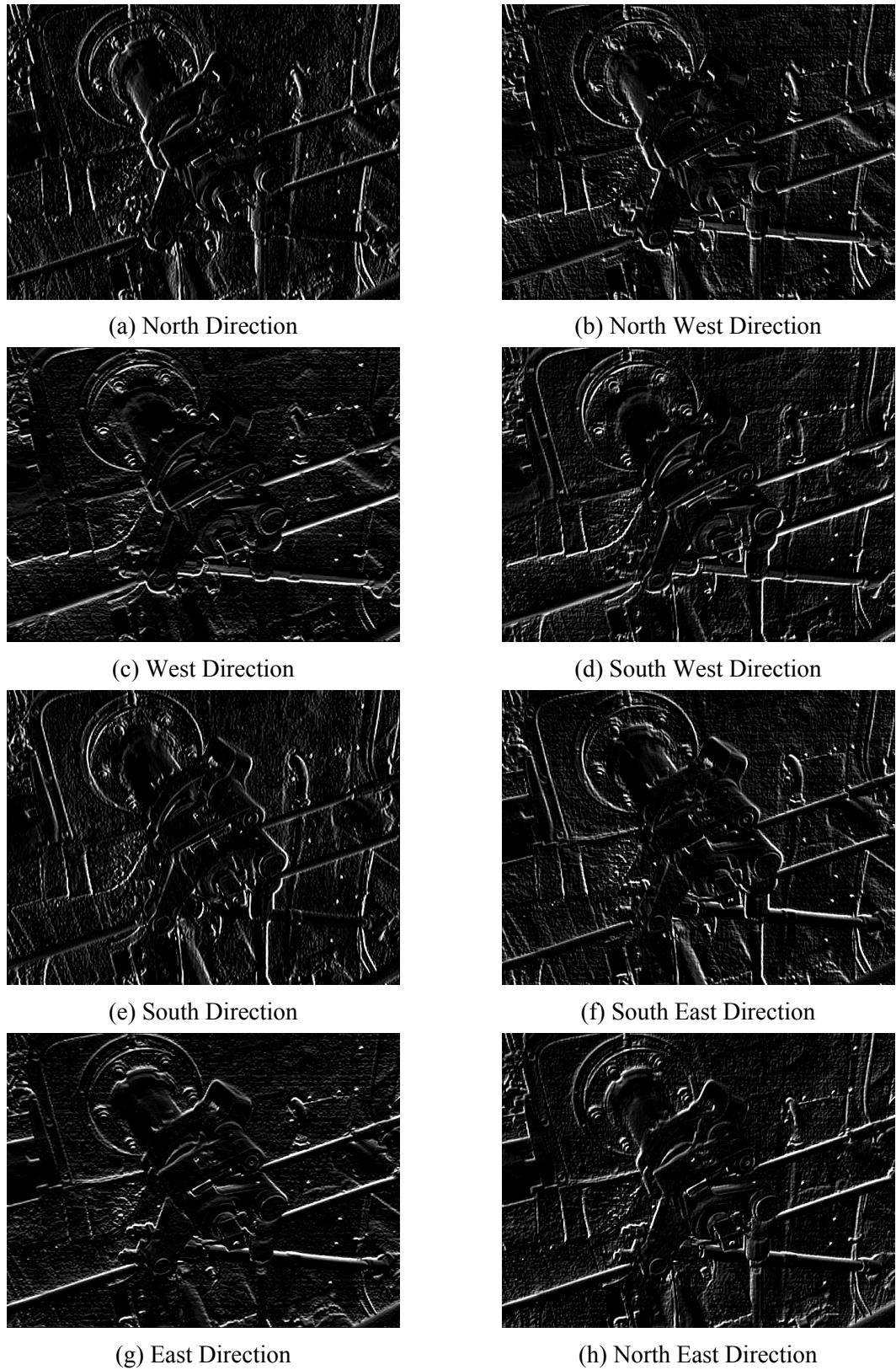


Figure 1.10: Robinson Compass Mask

having no edges in North-East direction then that filter will be useless.

-3	-3	5
-3	0	5
-3	-3	5

(a) North Direction Mask

-3	5	5
-3	0	5
-3	-3	-3

(b) North West Direction Mask

5	5	5
-3	0	-3
-3	-3	-3

(c) West Direction Mask

5	5	-3
5	0	-3
-3	-3	-3

(d) South West Direction Mask

5	-3	-3
5	0	-3
5	-3	-3

(e) South Direction Mask

-3	-3	-3
5	0	-3
5	5	-3

(f) South East Direction Mask

-3	-3	-3
-3	0	-3
5	5	5

(g) East Direction Mask

-3	-3	-3
-3	0	5
-3	5	5

(h) North East Direction Mask

Table 1.7: Krisch Compass Mask

### 1.1.7 Laplacian

The laplacian edge detector is a derivative operator applied to detect edges in an image. It is a second-order derivative filter. It can then be divided as positive laplacian or negative laplacian. These filters detect edges. Few detect vertically and horizontally, few detect in a single direction only, whereas few detect in every direction. Some generally applied small kernels are displayed in Table 1.8. Edges after applying Laplacian is shown in Fig 1.12.

1	1	1
1	-8	1
1	1	1

-1	2	-1
2	-4	2
-1	2	-1

0	1	0
1	-4	1
0	1	0

Table 1.8: Commonly used Laplacian Kernels

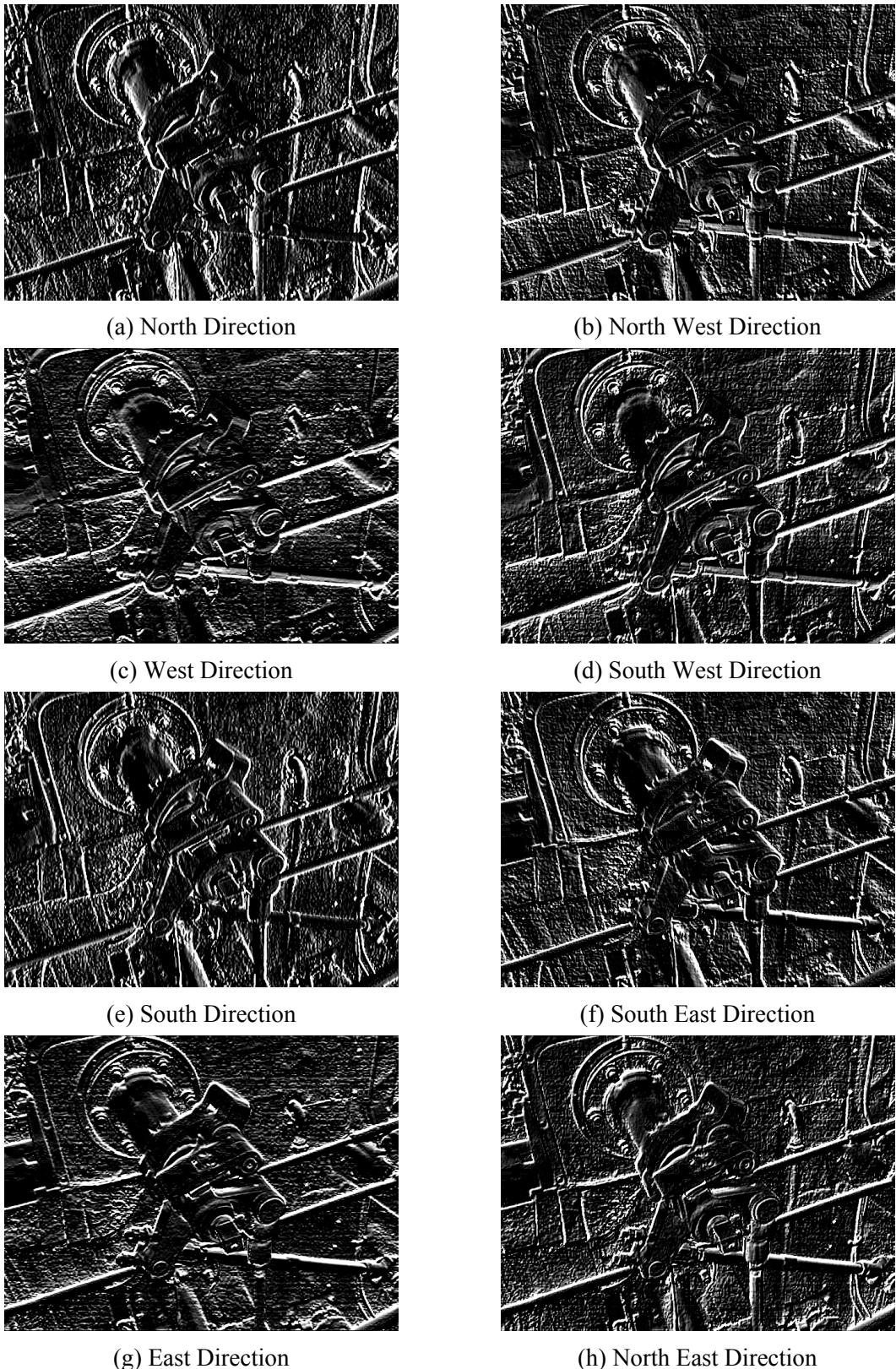


Figure 1.11: Kirsch Compass Mask

### 1.1.8 Canny

Canny edge detection operator is used to find edges in pictures. It is a multi-stage algorithm. This edge detection operator detects edges by searching for local maxima of the gradient of



Figure 1.12: Edges after applying Laplacian

$f(x, y)$ . A gradient is computed with the help of Gaussian filter's derivative. The technique takes two thresholds to find weak and strong edges. It keeps the weak edges in the result only if there is a connection between them and strong edges.

Canny edge detection operator is a multi-stage algorithm. Each step is described below.

### 1. Reduction of noise

As edge detection is sensitive to noise in the picture, the initial action is to eliminate the noise in the picture using a Gaussian filter of 5x5 size.

### 2. Determining Gradient of an Image

After smoothening of an image, it is next convolved with Sobel filter in both vertical ( $G_y$ ) and horizontal ( $G_x$ ) direction.

$$\text{EdgeGradient}(G) = \sqrt{G_x^2 + G_y^2} \quad (1.4)$$

$$\text{Angle}(\Theta) = \tan^{-1} \left( \frac{G_y}{G_x} \right) \quad (1.5)$$

### 3. Non-maximum Suppression

A complete scan of a picture is done to eliminate any undesired pixels which do not create the edge. Every pixel is examined whether, in its neighbourhood, it is a local maximum in the direction of the gradient. It is shown in Figure 1.13. In the vertical direction, a Point 'A' is shown on edge. Gradient direction lies normal to edge. In gradient directions, Point B and C are shown. Point A is examined with point B and C to watch if it makes a local maximum. If it makes, it is reviewed for the following stage else, it is discarded ( assigns zero to it). The outcome we perceive is a binary image having "thin edges".

### 4. Hysteresis Thresholding

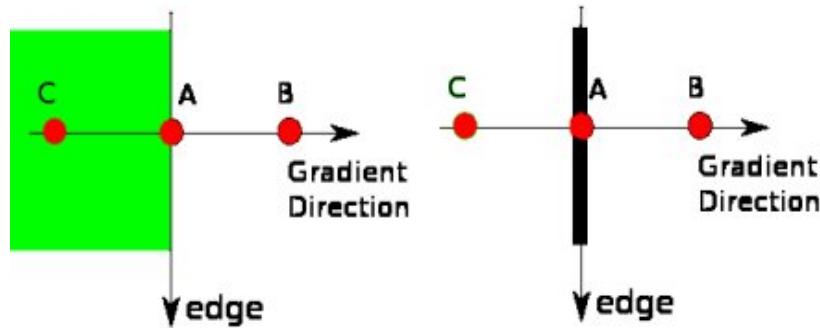


Figure 1.13: Non-maximum Suppression [12]

- This stage determines which edges are actually edges and which are not.
- Two threshold values are required, which are  $\text{minVal}$  and  $\text{maxVal}$ .
- Edges having intensity gradient larger than  $\text{maxVal}$  are surely edges, and those lesser than  $\text{minVal}$  are surely non-edges, and hence rejected.
- The edges which lie within the two thresholds are considered edges or non-edges depending upon their connectivity. If edges are attached with “sure-edge” pixels, they are classified as a part of edges else they are rejected. It is shown in Figure 1.14.

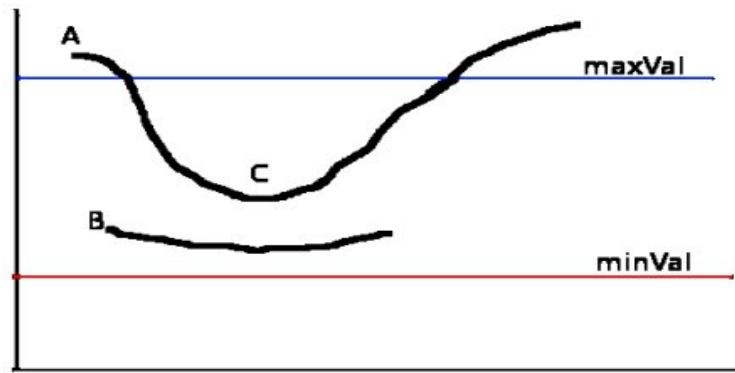


Figure 1.14: Hysteresis Thresholding [13]

The edges after applying Canny edge detector is presented in Fig 1.15.

## 1.2 Advantages and Disadvantages of Edge Detection Operator

Edge detection is a crucial step in the field of computer vision. It is required to highlight the actual edges to get the desired output from the process. Therefore it is essential to select edge detection operators that accurately fits the application. Here, we initially show some of the Edge Detection Operators's advantages and disadvantage in Table 1.9.



Figure 1.15: Edges after applying Canny

Operator	Advantages	Disadvantages
Classical (Sobel, Roberts, Prewitt, Kirsch,...)	Simple, Edge Detection and their orientations	Sensitive towards noise, Not accurate
Zero Crossing(Laplacian and Second derivative)	Edge Detection and their orientations. Have fixed features in every directions	Responds to specific existing edges, Sensitive towards noise
Laplacian of Gaussian(LoG)	Finds the exact location of edges, Tests a broader area on sides of the pixel	Fails at the curves, corners, and where the grey level intensity differs. Does not finds the orientation of edge as applying the Laplacian filter
Gaussian(Canny)	Uses probability for calculating error rate, response, and Localization. Improves signal to noise ratio, detects better especially in noise conditions	Complicated Calculations, time consumption, false zero crossings

Table 1.9: Advantages and Disadvantages of Edge Detectors

### 1.3 Regression Models

The result of the dependent variables using independent variables is predicted by Regression models. Most complex problems are handled by regression analysis. Few of the standard regression models are described below:

### 1.3.1 Linear Regression

Linear Regression [3, 4] is the most usual predictive model to recognize the relationship between the variables. In Linear Regression, one variable is considered as a dependent ( $y$ ), and the other as the independent variable ( $x$ ). The equation for linear regression is given in Eq 1.6.

$$y = mx + c \quad (1.6)$$

where,  $x$  is an independent variable with a continuous value,  $y$  is the dependent variable, having a continuous or a categorical value,  $m$  is the slope of the line, and  $c$  is the  $y$ -intercept. Linear Regression is shown in Fig 1.16. We are trying to minimize the squared distance

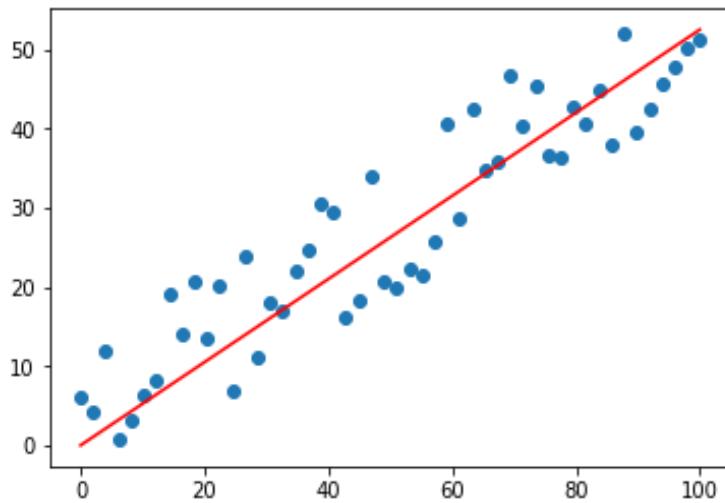


Figure 1.16: Simple Linear Regression [14]

(hence the “least squares”) between the observations themselves and the predicted values given in Eq 1.7.

$$\text{Difference}_i^2 = (y_i - (mx_i + c))^2 \quad (1.7)$$

### 1.3.2 Support Vector Regression

Support Vector Machine is an example of supervised learning models for classification and regression. [5]. Support Vector Machine, in case of regression, is explicitly said to be Support Vector Regression. Support Vector Regression may be linear or non-linear, applying specific kernel functions. One significant property of Support Vector Regression(SVR) is, rather than reducing the observed training error, it tries to minimize the generalized error bound to attain a generalized performance. Support Vector Regression is shown in Fig 1.17. The purpose of SVR is to compute a linear regression function on a high dimensional feature space in which the input data are mapped through a non-linear function.

We have used RBF (Radial Basis Function), which is a non-linear kernel. The kernel functions modify the data into a higher dimensional feature space so that we can perform the

linear separation. The RBF kernel is defined in Eq 1.8.

$$K(x^i, x^j) = \exp(-\|x^i - x^j\|^2 / (2\sigma^2)) \quad (1.8)$$

where,

- $\|x^i - x^j\|^2$  = Squared Euclidean distance between vectors  $x_i$  &  $x_j$
- $\sigma$  is a free parameter
- $\gamma$  is a parameter that sets the “spread” of the kernel
- $\gamma = 1/(2\sigma^2)$

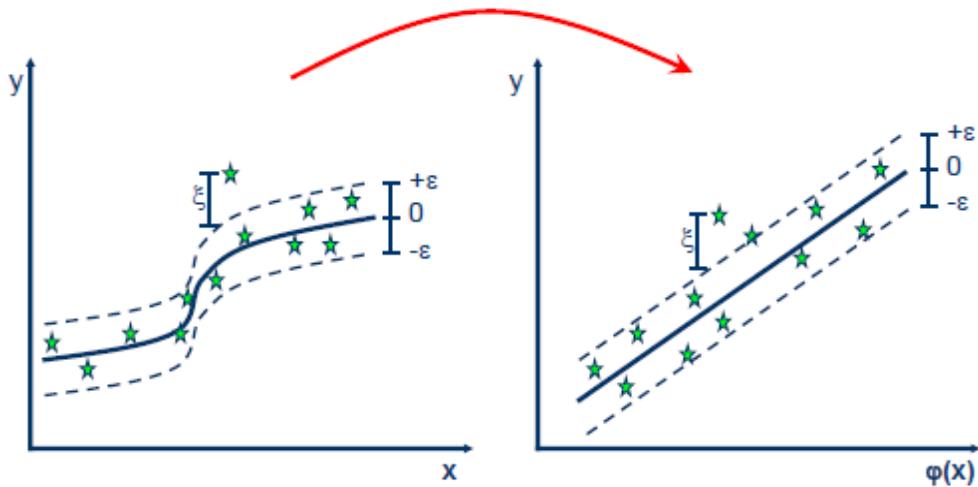


Figure 1.17: Support Vector Regression [15]

### 1.3.3 Multiple Regression

Multiple regression [6] can be defined as an extended version of linear regression. It is useful when the variable's value depends on two or more variables value. The variable which is predicted is known as the dependent variable or the outcome. The variables which predict the value of the dependent variable are known as independent variables or the predictor. The Multiple Regression is described in Eq 1.9.

$$y = b_1x_1 + b_2x_2 + \dots + b_nx_n + c \quad (1.9)$$

Here,  $b_i$ 's ( $i=1,2,\dots,n$ ) are known as the regression coefficients, which represent the value at which the criterion variable changes along with the changes in predictor variable. Multiple Regression is shown in Fig 1.18.

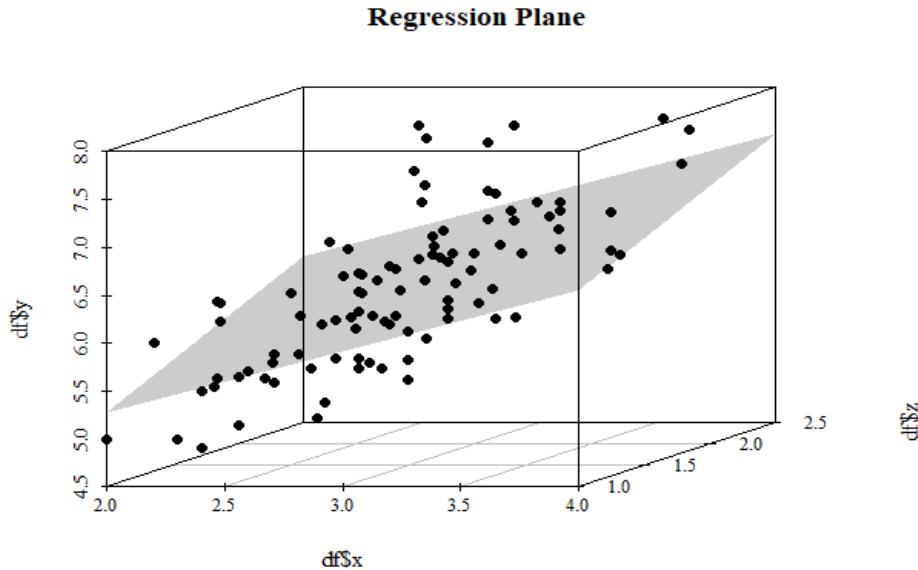


Figure 1.18: Multiple Regression [16]

Here, two independent variables and one dependent variable are taken. Dependent variable is represented by the response values obtained after convolution of input image using the edge detectors. The first independent variable is represented by the pixel number and the second independent variable as the local binary pattern (LBP) [17] of the input image. LBP is used to calculate a local texture representation.

LBP is used to calculate a local representation of texture. Every central pixel is checked with its eight neighbouring pixels. The neighbours who have lesser value than the central pixel will be assigned the bit 0. The remaining neighbours whose value is equal or higher than the central pixel will be assigned the bit 1. For every central pixel, a binary number is calculated by appending all these binary bits in a clockwise manner, beginning with one of its top-left neighbours. The final decimal value of the binary number formed is used as the central pixel value.

## 1.4 Potential applications of Edge Detection Operators

Few of the significant fields in which Edge Detection Operators is majorly used are described below:

- Microscopic Imaging
- Video processing
- Pattern recognition

- Color processing
- Machine/Robot vision
- Transmission and encoding
- Remote sensing
- Medical field
- Image sharpening
- Others

### **1. Image sharpening**

Image sharpening refers here to synthesize images clicked from the modern-day camera to convert them into a better picture or to operate those images to get the required result.

### **2. Medical Areas**

Few applications of Edge Detection Operators in the area of medical is:

- Gamma ray imaging
- PET scan
- X Ray Imaging
- Medical CT
- UV imaging

### **3. UV imaging**

In remote sensing, the earth area is examined by a satellite or from a very high level and then inspected to get details about it. One specific use of digital image processing in the case of remote sensing is to find infrastructure destruction by an earthquake.

A considerable amount of time is taken to grab damage, even if severe accidents are to be focused. As the region affected by the earthquake is infrequently so vast, it is impossible to check it with the human eye to figure out the damages. Although, then it is a very chaotic and gradual process. Digital image processing provides a solution to this. The affected area image is taken from the above and later examined to find the earthquake's several types of damage as shown in Fig 1.19.



Figure 1.19: Remote Sensing [18]

The main steps included in the examination are:

- The production of edges
- Examination and improvement of several types of edges

#### 4. Transmission and encoding

The initial image sent through the wire was from London to New York through a submarine cable. The image that was sent is shown in Fig 1.20.



Figure 1.20: transmission [19]

The image that was transmitted took three hours to make from first place to another.

Nowadays, we can watch live video feeds or live CCTV footage sitting in one continent or the country in just a few seconds. It signifies that plenty of jobs has been finished in this field. This area is not only related to transmission but also with encoding. Various

arrangements have been made for low or high bandwidth to encode images and then send it on the internet or e.t.c.

## 5. Machine/Robot vision

Besides the various problems that a robot encounter today, one of the main challenges is enhancing the robot's vision. Making robots intelligent enough to see objects, identify objects, identify the obstacles, e.t.c. This area has given many works, and the computer vision field is initiated to work on it.

## 6. Hurdle detection

Hurdle detection is one of the tasks that can be completed through image processing, by recognizing various objects in the picture and then the distance is calculated between robot and hurdles as shown in Fig 1.21.

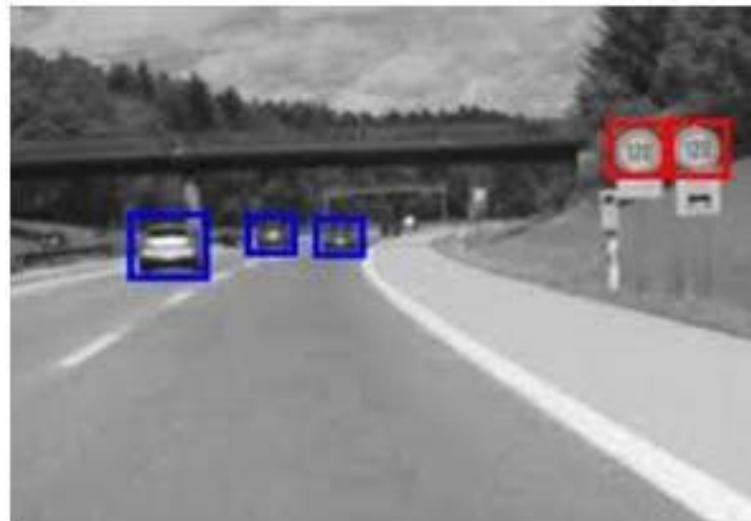


Figure 1.21: Hurdle Detection [20]

## 7. Line follower robot

Today robots perform by following the line and therefore called line follower robots. It assists a robot in walking on its path and performing specific tasks. It is also attained through image processing shown in Fig 1.22.

## 8. Color processing

Color processing refers to the refining of color images and various color spaces, such as the RGB color model, YCbCr, HSV, etc. It also includes understanding the storage, transmission, and encoding of the color images.



Figure 1.22: Line follower robot [21]

## 9. Pattern recognition

Pattern recognition includes learning from image processing and different other fields, which include machine learning (an area of artificial intelligence). Image processing is used in pattern recognition, for recognizing the items in a picture, and then it is applied to train the machine for changing the pattern. Various application of Pattern recognition includes identifying images, recognition of handwriting, computer-aided diagnosis, e.t.c

## 10. Video processing

A video is the swift movement of images. The video quality relies on the number of images/frames per minute and the standard of each frame included. Video processing includes color space conversion, aspect ratio conversion, frame rate conversion, detection of motion, detection of motion, specific enhancement, reducing noise, e.t.c.

# 1.5 Potential uses of Regression Models

## 1. Predictive Analytics

Predictive analytics deals with predicting future chances and prospects, which are the most crucial implementation of regression analysis in the case of business. The examination of demand forecasts the number of products that a customer might purchase. There are other dependent variables also other than demand in case of business. Regression analysis can perform various tasks other than predicting the effect on direct revenue. We can predict the number of customers who will go ahead of a specific billboard and use it to calculate the highest bid for an advertisement. To calculate the available number of claims in a given period and credit standing of policyholders, insurance companies majorly depend on regression analysis.

## **2. Operation Efficiency**

Regression models may be used to improve business processes. For example, a plant manager can generate a statistical model that recognizes the effect of temperature of the oven on the shelf life of baked cookies on ovens. We can examine the connection between the callers' waiting time and the number of complaints in a call center. Decision-making, with data-driven help, reduces politics from decision making, hypothesis, and guesswork. Thus, it improves performance in business by pointing out fields that have the highest effect on revenues and operational efficiency.

## **3. Supporting Decisions**

Nowadays, Businesses are filled with operations, finance data, and purchase of the customer. To make an informed decision on business progressively, executives are nowadays relying on data analytics, therefore reducing the instinct and gut feeling. To present a scientific angle to any business management regression analysis can be used. Regression analysis takes the path to accurate and smarter decisions by lowering a massive amount of raw data into actionable information. It is not that regression analysis is an end to the manager's analytical and creative thinking. This method proved to be a perfect tool for testing a hypothesis before jumping into execution.

## **4. Correcting Errors**

Regression is significant for giving empirical support to the decision of the management and spotting judgmental errors. An example could be, a manager of a retail store can believe that increasing shopping hours will significantly increase his sales. Regression analysis can show that the advance in revenue could not be enough to support the increase in operating expenses because of higher hours of working (such as an extra charge for employee labor). Therefore, regression analysis gives quantitative support for preventing mistakes and making decisions due to the manager's instinct.

## **5. New Insights**

With the increase in time, businesses have collected a massive volume of disordered data that can produce beneficial insights. Still, this data is pointless without actual analysis. Such techniques can find a link between various variables by detecting patterns that were earlier unseen. For example, data analysis in the view of purchase accounts and sales systems can feature trends like rising demand on specific weekdays or at fixed times of the year. One can keep personnel, and optimal stock prior to a spike in demand appears by recognizing these insights.

## **1.6 Thesis Contribution**

We proposed an estimation model as an objective function and further, it is used to determine the degree of proximity (suitability) of the edge operators. Motivated by the need of correctly detecting the edges, the following objectives were undertaken.

1. To detect the edges correctly and accurately.
2. To avoid the difference in opinion of users about the output provided by the edge detection operators and obtain a basis where perception of output provided by the edge detection operators is uniform even with different users.
3. To design and develop a novel edge detection operator called Hybrid operator in order to strengthen the process of detecting edges and sharp discontinuities for various image processing applications.
4. To develop a model to determine the degree of proximity (suitability) of the edge operators on basis of Regression Models.
5. To determine the best suitable edge detection operator for a particular application.
6. To obtain the same edged effect from a set of edge detection operators used for some specific application.
7. To compute the Root Mean Square Error for the respective edge detector. RMSE is directly related to the strength of the discontinuity in the image caused by the changes in the gray levels.
8. To detect COVID-19 on chest X-Ray and CT Scan images of suspected individuals.
9. To develop a model which uses CNN fused with the image processing based data augmentation.
10. To resolve the issue of smaller number of images for training the deep-model, which does not give the desired accuracy.
11. To increase the data by using multiple representations of the same X-Ray and CT scan images, produced through sharpening filters.

## **1.7 Thesis Organisation**

This thesis will consist of three more chapters. Chapter 2 presents the literature survey, which focuses on recognizing various researches about the suitability of edge detection operators prior to an application and how they will apply to our work. It will also comprise of

preliminary sections and exhaustive descriptions of various types of edge detection operators and its applications. Some papers related to detection of Covid-19 on Chest X-Ray and CT Scan images are also studied. Chapter 3 contains a thorough explanation of the project determining the suitability of edge detector in detail, Chapter 4 presents the proposed work for detecting Covid-19 in Chest X-Ray and CT Scan images, and Chapter 5 draws the conclusion and mentions possible future works that can be done.

## Chapter 2

# Literature Survey

Edge detection operators are subjective in nature, and there is no such basis function available to determine the best suitable operator for a particular application. Due to the subjective nature of edge detection operators, it is extremely difficult to predict the best edge detector while the appropriateness of edge detection output to be decided visually as the perception varies from user to user. Moreover, a user perceives the outcome of edge detection function differently and decides the suitability after having some cognitive responses generated from the human vision system. Hence, it is cumbersome to obtain the same edged effect from a set of edge detection operators used for some specific application. A number of papers were studied during this thesis work, which helps to understand the basic concept of edge detectors, functioning of different operator, comparison of different edge detectors and main idea of thesis work. Some papers are discussed below:

### 2.1 Comparison of different Edge Detection Operators

**John Canny** [22], in this paper, a computational method to edge-detection was described. This method's achievement depends on the meaning of a comprehensive set of goals for calculating edge points. This paper defines detection and localization criteria for edges and introduce a mathematical form for these norm as functionals on the operator impulse response then added to confirm that the detector has only one response to a single edge. After analysis to step boundaries, find a natural uncertainty principle between localization and detection performance, which are the two main primary goals. This principle derives a single operator shape, which is optimal at any scale. The optimal detector has a simple approximate implementation in which edges are marked at maxima in gradient magnitude of a Gaussian-smoothed image and extend this simple detector using operators of various widths to handle with different signal-to-noise ratios in the picture.

This paper presents a general method, known as feature synthesis, for the fine-to-coarse integration of information from operators at a variety of scales. Finally, it shows that the step edge detector's performance is considerably enhanced as the operator point spread

function is extended along the edge.

**Raman Maini, Dr. Himanshu Aggarwal** [23] are focused on the comparison and study of various edge detection operators. According to this paper, Edges identify boundaries and are thus a problem in image processing. Edge detection notably reduces the amount of data and filters out impractical information, while preserving the crucial structural quality in an image. As edge detection is at the forefront of image processing for object detection, it is necessary to have a good grasp of edge detection operators. A comparative analysis of different edge detection operators is presented in this paper.

It has exhibited that the Canny edge detection operator performs better than all the following operators under all conditions- Canny, LoG(Laplacian of Gaussian), Robert, Prewitt, Sobel. The visual comparison is made in this paper to compare which edge detection operator performs better than all the other edge detection operators.

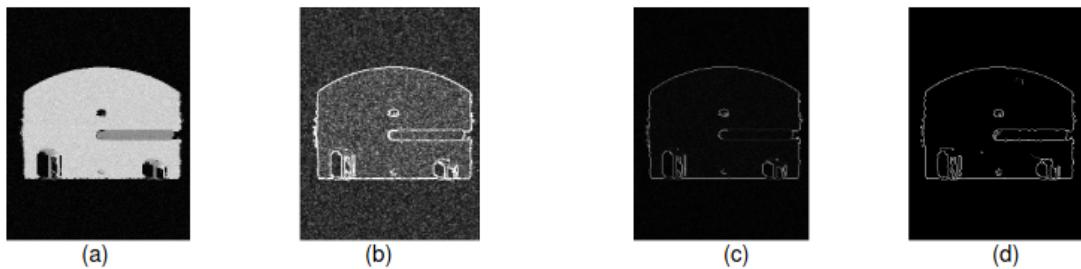


Figure 2.1: Comparison of Edge Detection operators on Noisy Image (a) Original Image with Noise (b) Sobel (c) Robert (d) Canny [23]

**Lijun Ding, Ardeshir Goshtasby** [24] work on the Canny edge detection operator and explain that in computer vision, the Canny edge detection operator is widely used to find sharp gray levels intensity changes and to locate boundaries of the object in an image. If the magnitude of the gradient of the pixel is higher than the pixels at both sides in the direction of change of maximum intensity, then the Canny edge detection operator classifies a pixel as an edge. They show that determining edges in this order causes some evident edges to be missed out. They also show how to amend the Canny edge detection operator to enhance its edge detection accuracy.

**Yu Hongshan Wang Yaonan** [25], Edge detection is the starting step to obtain an image feature. An enhanced Canny edge detection algorithm is presented to achieve robust and thin edges. Compared with the conventional Canny method, there are four improvements to reduce computation time and ensure detection accuracy. Firstly, a 2 Dimensional Gaussian kernel is decomposed into two independent 1 Dimensional filters, i.e., row filter and column filter, which allows calculating image gradient in a parallel way. As a result, the computation time is reduced highly. Secondly, the method uses two thresholds, to find sharp and weak edges, and includes the fragile edges in the final output only if they are attached to sharp

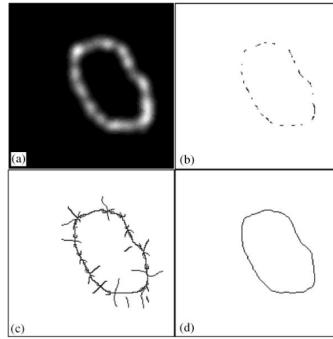


Figure 2.2: A simulated gradient image. (b) Edges attained by the nonmaxima suppression of Canny. (c) The minor edges. (d) Edges achieved by the amended Canny edge detector [24]

edges. Therefore, this method is improbable than the others to be deceived by noise and more likely to find actual weak edges. Thirdly, the on-maximum suppression principle is adopted to detect actual edges.

Finally, a thin edge operation is conducted based on a morphological operator to achieve a single-pixel level edge. The effectiveness of the proposed algorithm is shown through practical experiments.

**D. Poobathy and Dr. R. Manicka Chezian** [26], In this paper, they describe the Edge detection methods, which are accessible for pre-processing in computer vision. Sobel, Roberts, Prewitt Canny, Laplacian of Gaussian (LoG), are the most used algorithms. They compare these edge detection operators using Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR) of the resultant image.

The set of four universally standardized test images is adopted for the experiment. The MSE and PSNR results are numeric values; the performance of algorithms are identified based on that. The time required to detect edges for each algorithm is also noted down and documented. After the experiment, the Canny edge detection operator was found the best among others in terms of edge detection accuracy.

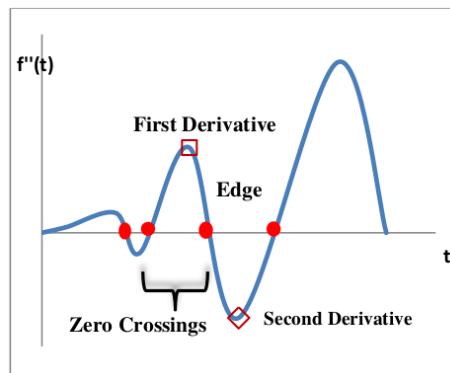


Figure 2.3: Edge Transformative Derivation [26]

**Tamilselvi Nagasankar and B. Ankaryarkanni** [27], Their objective is to analyze the various edge detection operators to obtain the best and worst performance of the edge detection operator. Edge detection is an essential and necessary operation to be completed for any image processing application, image analysis, pattern recognition on various images such as satellite images, medical images, etc. The performance of the edged image is measured with the help of a signal-noise ratio and entropy. High SNR and entropy values determine the high quality of the edged image, and the low values represent the picture's low quality. Making an in-depth analysis of various edge detection operators is worth enough in Image processing. Five majorly used edge detection operators, such as Sobel, Roberts, Prewitt, Canny, and Log, are examined for analysis. From the study, it is recognized that the Canny edge detection operator is performing best among the other edge detection operators. Out of the five image information, the Canny edge detection operator on Dither binary image information produces high SNR and entropy values.

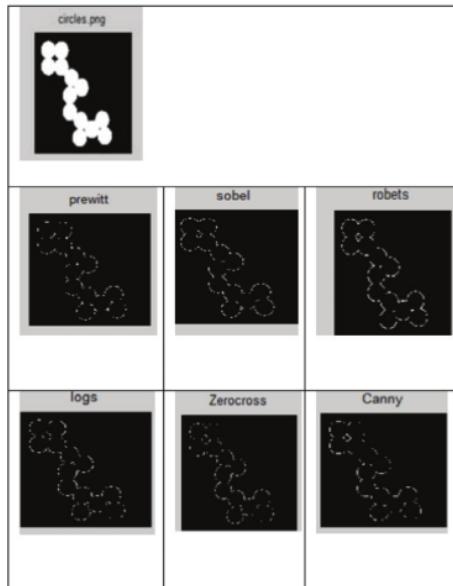


Figure 2.4: Edge detection from a grayscale representation of an image. [27]

**Shuai Wan , Fuzheng Yang and Mingyi He** [28], In spite of the ubiquity of traditional methods for the detection of edges using gradient-threshold, the fundamental global threshold is not flexible to the picture content concerning human perception. In this paper, local limits are fittingly selected to mark the underlying challenges, taking into account the activity-masking feature of the human visual system. The local thresholds are then selected and then utilized for labeling the edges in the gradient image. Considerable experimental results have shown the efficacy of the proposed method for detecting edges in visible quality.

**Rashmi, Mukesh Kumar and Rohini Saxena** [29], An edge is described as a set of attached pixels that makes a border between two disconnected regions. Edge-detection is a technique of segmentation of an image into areas of discontinuity. Edge-detection plays

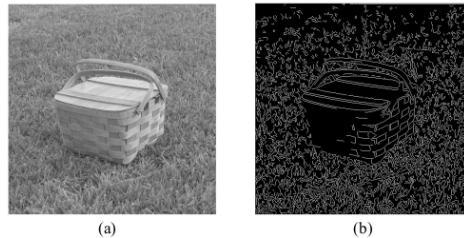


Figure 2.5: Edge detection using the Canny edge detection operator. (a) original image; (b) edge image [28]

a vital role in digital image processing and effective form of our life. In this paper, several edge-detection operators are used, such as Sobel, Roberts, Prewitt, Canny, and Marr Hildreth edge detection operators. After collating, the performance of the Canny edge detection operator is found to be better than the other edge detection operators on various sides, such as the low probability of detection of false edges, better for an image with noise, gives sharp edges, adaptive, etc.

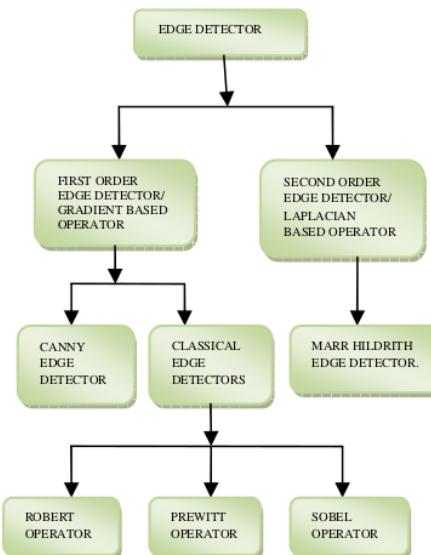


Figure 2.6: Types of edge detector [29]

**Geng Hao, Luo Min, Hu Fen** [30], They worked for an enhanced, flexible edge detection operator based on the Canny edge detection operator. In this paper, they describe edge detection is an essential topic in image recognition, image analysis, and digital image processing. The edges are not only the essential feature of an image and the critical basis for image segmentation but also a valuable data source of the feature of texture and the foundation of the analysis of quality of a shape. The assumption of obtaining the precise object contour in a conventional Canny edge detection operator is fixed to suitable parameters without the flexibility. They proposed a flexible Canny edge detection operator built on the Canny theory. The highest between-class variance technique is used to obtain low and high thresholds. With the diagram, they showed that their algorithm gave proper

edge detection.

**Mohamed Abo-Zahhad and et al,** [31] They presented edge detection with a preprocessing approach. Edge detection is the procedure of finding boundaries of objects which occur in an image. Various conventional operators based techniques have been extensively used for the detection of edges. Due to the intrinsic quality of images, these techniques prove useless if used without preprocessing. In this paper, preprocessing method has been used to get specific parameters that help carry out preferred edge detection with the conventional operator's based detection of edges. The presented preprocessing process involves calculating the histogram, determining the number of peaks, and abolishing inappropriate spikes. Threshold values are found from the intensity values equivalent to relative peaks. Ideal thresholds are determined from these threshold values, optimal limits are determined, Simulation results show that our pre-processed way, when used with a traditional edge detection approach, increases its performance. They have shown that using the wavelet edge detection approach to the segmented images obtained through preprocessing method produces the highest performance among other classical edge detection operators.

## 2.2 Detection of Covid-19 on X-Ray and CT Scan images

**Jianpeng Zhang, Yutong Xie,** [32] In the paper COVID-19 Screening on Chest X-ray Images Using Deep Learning-based Anomaly Detection, they have introduced a deep-learning model to detect COVID-19 against non-COVID-19 cases. As presented in Figure 2.7, the model is comprised of three segments, specifically, an anomaly detection head, a classification head, and a backbone network.

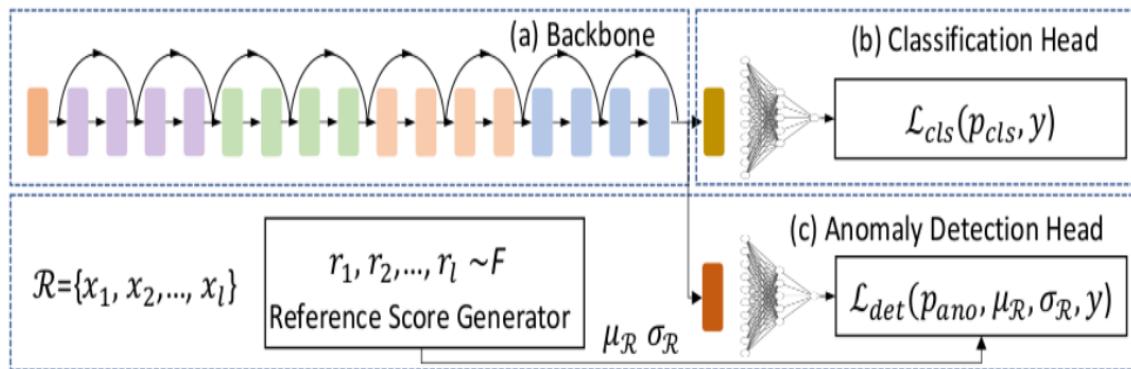


Figure 2.7: Diagram of proposed model by Jianpeng Zhang, Yutong Xie [32]

The backbone network is applied to obtain the high-level features of a given input chest X-ray image  $x$ , which is then given as input to the classification head and anomaly detection head, respectively. The classification head produces a classification score  $p_{cls}$ , and the

anomaly detection head produces a scalar anomaly score  $p_{\text{ano}}$ . Consequently, another scalar score  $\mu$  is computed by randomly choosing 1 normal X-ray images and computing the mean of 1 anomaly scores. Lastly, the model is optimized via reducing the binary cross-entropy loss for classification and the deviation loss for anomaly detection and, intending to give statistically significantly higher classification scores and anomaly scores to X-ray images with COVID-19 to those given to normal controls.

The dataset applied for their work consists of 100 chest X-ray images, all of which were infected with COVID-19, and 1431 chest X-ray images known as pneumonia (not COVID-19).

**Ophir et al.**, [33] have presented Rapid ai development cycle for the coronavirus (covid-19) pandemic: Initial results for automated detection & patient monitoring using deep learning ct image analysis. For tracking, quantification, and detection of COVID-19, an Artificial Intelligence-based automated CT image analysis tools are developed. It can be used to distinguish COVID-19 patients from others who do not have this disease.

Various global databases, including from Chinese disease-infected areas, were covered. They have introduced a system that employs a robust 2 Dimensional and 3 Dimensional deep learning models, transforming and conforming existing Artificial Intelligence models and merging them with clinical knowledge. They have performed various retrospective attempts to examine the performance of the system in detecting suspected COVID-19 thoracic CT features and to estimate the development of the virus in per patient over time using a 3 Dimensional volume review, producing a “Corona score”. The study comprises a testing set of 157 international patients (China and U.S).

**Shi et al.**, [34] They have proposed the paper Large-Scale Screening of COVID-19 from Community Acquired Pneumonia using Infection Size-Aware Classification. The global expanse of coronavirus disease (COVID-19) has become a dangerous risk for global health. It is of high interest to quickly and correctly check patients with COVID-19 from community-acquired pneumonia (CAP). In their research, a total of 1658 patients with COVID-19 and 1027 patients of CAP undergo thin-section CT. All images were preprocessed to get the segmentation of both viruses and lung fields, which were utilized to obtain location-specific traits. An infection Size Aware Random Forest method (iSARF) was developed, in which subjects were automatically divided into collections with varying ranges of infected lesion sizes, succeeded by random forests in per collection for classification. Experimental results exhibit that the given method produced a sensitivity of 0.907, a specificity of 0.833, and accuracy of 0.879 following five-fold cross-validation. High-performance margins on comparison techniques were produced mainly for the cases with virus size in the average range, from 0.01% to 10%. The additional addition of Radiomics features displays a slight improvement. They anticipated that their proposed structure could help in clinical decision making.

## **2.3 Summary**

In this chapter, various Edge Detection operators are explored, which helps to understand the basic concept of edge detectors, functioning of different operator, comparison of different edge detectors. Later some papers regarding CoronaVirus Infection and its detection on CTSCAN and X-RAY images are explored.

## **Chapter 3**

# **Determining the Suitability of Edge Detectors using Regression Models**

Unlike image restoration, image enhancement techniques are found to be subjective in nature as the appropriateness of the appearance of output image depends upon human perception. Hence, it is very difficult to determine the appropriateness of the image enhancement techniques including edge detection prior to an application. This paper makes use of regression models to determine the suitability of edge detection operators. With the existing operators, a novel Hybrid operator is used in the evaluation. The novel detector is made of combining Canny and Sobel operators with the gradient of texton image. With this approach, an estimation model as an objective function is determined and further, it is used to determine the degree of proximity (suitability) of the edge operators on two publicly available databases, viz. the BSDS300 and the Multi-cue.

### **3.1 Motivation**

Once the edge detection is finished, the further task of understanding the information present in the original picture can be significantly simplified. Still, attaining such ideal edges from quite complicated real-life photos can be more complex than it seems.

Traditionally, edge detection operators are compared by means of visible perception of their output image. In case of non-trivial images, it is nearly impossible and unreliable to conclude what edge detection operator is best just by looking at the output provided by it.

Also, the selection of an edge detector depends on the nature of edges. Therefore it is imperative to determine the suitability of an edge detector before using it on an image.

The edges obtained from non-trivial images are usually hindered by fragmentation. Therefore, the edge curves are disjoint because of absent edge segments and false edges (edges not corresponding to any essential phenomena in the image), making the next task of understanding the image data complex. As a result, visual perception to understand the edges can prove to be a difficult task.

## 3.2 Proposed Edge Detection Operator

In order to strengthen the process of detecting edges and sharp discontinuities for various image processing applications, a novel edge detection operator called Hybrid operator is proposed. It uses both Sobel [35] and Canny [36] detectors to provide high-frequency spatial information and at the same time, it normalizes the noise content in the image. This rare combination makes the proposed operator very much useful for edge detection as well as image segmentation operations [37] while texture properties of the image is enhanced. In this operator, a texton based texture framework which gives a unique texture is used. This texton image [10] is created with the use of derivative of Gaussian [38] filter bank. Then we find the gradient of the texton image by the use of paired disk masks. The output image produced in this process is a combination of Canny operator, Sobel operator, and gradient of texton image.

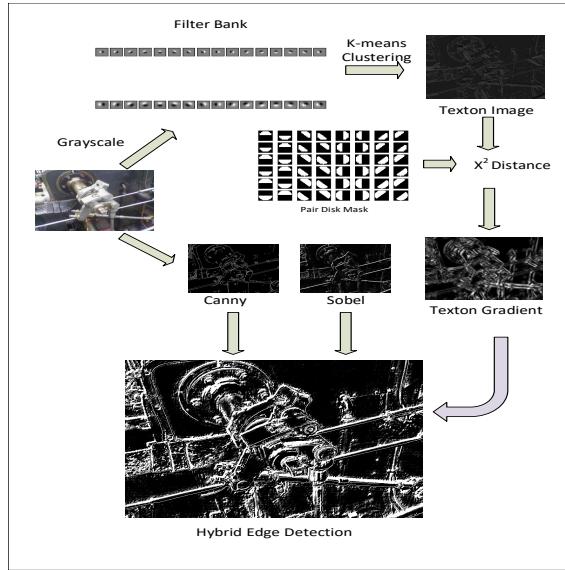


Figure 3.1: Design Process of the Proposed Edge Detection Operator

The design process of the proposed operator consists of the following steps: (a) The first step constructs a derivative of Gaussian filter bank by convolving the Gaussian function [1] with the Sobel operator. Here, the Sobel operator is used as derivative operator. This derivative of Gaussian filter is computed at different scales ( $\sigma_1 = 1.2$  and  $\sigma_2 = 1.4$ ) and orientations (16 bins). With two different scales and 16 bins a filter bank of 32 filters is obtained. This pipeline of filters is then convolved with the input image and it creates a set of texton images while clustering is performed using k-means [39] with response values. It has been observed that if the kernel size varies then the overall appearance of convolution effect either turns to be smooth or contains high-frequency sharp discontinuities. (b) Second step determines texton image and it is then quantized in terms of set of pixels having range of  $[1, k]$  where  $k$  represents the number of clusters in k-means clustering algorithm. To obtain

a better result, k is fixed to 64. The filter bank produces a three-dimensional vector where depth is equal to the number of filters used. Further, the gradient of texton image is obtained with the help of a pair of masks as described in Algorithm 1. Here, the pair of disk masks refers to a pair of binary images of half disk. The pair of masks, i.e., left mask and right mask are convolved with an image having 0 and 1 pixel values where the pixel values are decided to be 1 when equal number of bins are found at the corresponding pixel locations and rest of the pixel locations are assigned to be 0. Then we find a distance between this two convolved images using Chi-square metric [40] which is found to be faster as compared to aggregating count and pixel neighborhood for the histogram as described in Algorithm 2. The pair of masks is made to be determined with three scales (5,10,15) and 8 orientations that correspond to the radius of a disk. The size of a kernel varies with respect to the radius of a disk.

---

**Algorithm 1** PairDiskMask algorithm
 

---

```

1: procedure PairDiskMask(radius, hdmOrientation)
2:   pdMasks = [ ]
3:   for <radii = 1:radius> do
4:     mask = create matrix with all zero's having size (radii*2+1,
radii*2+1)
5:   end for
6:   for <i = 1:radii> do
7:     x = i - radii2
8:     for <j = 1:(radii * 2 + 1)> do
9:       if (x + (j - radii)2 < radii2) then
10:        mask[i, j] = 1
11:       end if
12:     end for
13:   end for
14:   rotateAngle = 360.0/hdmOrientation
15:   for <i = 1:hdmOrientation> do
16:     rotated=interpolate mask with -i*rotateAngle
17:     rotated[rotated>1]=1.0
18:     rotated[rotated<0]=0.0
19:     rotated_p=interpolate mask with -i*rotateAngle-180      ▷ Rotated Pair
20:     rotated_p[rotated_p>1]=1.0
21:     rotated_p[rotated_p<0]=0.0
22:     pdMask ← rotated, rotated_p
23:   end for
24:   return pdMask
25: end procedure
    
```

---

(c) Third step computes a pair of edged images which are obtained by convolving the Canny and Sobel edge detectors with the input image. Then weighted edged images are determined by multiplying a weight with each edged image. Weights are determined according to the output it produces accurately. Further, weighted edged images are combined with the gradient of texton image as given in Equation 3.1.

$$O = Tg \times (w1 \times \text{Canny\_image} + w2 \times \text{Sobel\_image}) \quad (3.1)$$

where,  $O$  is output image,  $Tg$  is gradient of texton image,  $w1$  and  $w2$  are weights ( $w1+w2 = 1$ ). The design process of the proposed egde detection operator is shown in Fig 3.1.

---

**Algorithm 2** GetGradient algorithm
 

---

**Input :** Image,PairDiskMask

**Output :** Gradient image

```

1: chi_sqr_dist=img*0
2: for <i=1:num_bins> do
3:   temp = 1 where img is in bin i and 0 elsewhere
4:   gi = convolve temp with left_mask
5:   hi = convolve temp with right_mask
6:   update chi_sqr_dist+= 0.5 ×  $\sum_{i=1}^{\text{num\_bins}} \frac{(gi-hi)^2}{(gi+hi)}$ 
7: end for
```

---

### 3.3 Measuring the Appropriateness of Edge Detection Operators

This section presents a robust methodology to measure the degree of proximity of the edge detection operators using regression models [3–6]. In order to achieve the objective, the root mean square error (RMSE) [7] can be calculated for the respective edge detector. RMSE is directly related to the strength of the discontinuity in the image caused by the changes in the gray levels. In simple words, more the RMSE for a edge detector, more sharp the discontinuity to be found in the image. This helps us to choose mathematically the most optimal edge detector prior to an application. Barely, comparing the RMSE of different edge detection operators does not give the correct proximity about appropriateness of an edge detection operator. Now, in this situation a mechanism needs to be introduced to find an optimal edge detection operator that measures the sharp discontinuity present in the image more accurately without overlapping with irrelevant information. In this work, a set of regression models such as Linear, Support Vector, and Multiple regression models are used. These regression models find an optimal best-fit function in the cloud of response values determined from individual edge detection operator. This optimal best-fit function is referred as regression line, regression curve, and regression plane in case of Linear, Support Vector, and Multiple regression models respectively. This mechanism would be found useful

when the suitability of the edge detection operator would be verified prior to an application. Moreover, this mechanism facilitates the entire process to transform a subjective image processing technique to objective function where the dependency on the user's perspective is minimized.

Traditionally, edge detection operators are compared by means of visible perception of their output image. In case of non-trivial images, it is nearly impossible and unreliable to conclude which edge detection operator is best just by looking at the output provided by it. Also, the selection of an edge detector depends on the nature of edges. Therefore, it is imperative to determine the suitability of an edge detector prior to an application.

Some commonly used parameters are not suitable to determine the degree of proximity of an edge detector prior to an application. For, e.g,

- Response value can't be used as a parameter to determine the suitability of an edge detector as it's not a correct measure because some kernel have high values whereas some have low values.
- Total number of edges is also not a correct parameter to determine the suitability of an edge detector as false edges may be occurred due to noise.
- Visual perception [41] is also not a correct measure to determine the suitability of an edge detector as it's nearly impossible and unreliable to conclude what edge detection operator is best just by looking at the output provided by it.

Next, the regression models are introduced briefly with their working principles.

**Linear Regression:** In Linear Regression [3, 4], one variable is considered as the independent variable ( $x$ ) and the other is considered as the dependent variable ( $y$ ). In this regression model, the square distance between the observations themselves and the predicted values are minimized as given in Equation 3.2.

$$diff_i^2 = (y_i - (mx_i + c))^2 \quad (3.2)$$

where,  $m$  is slope of a line and  $c$  is intercept of a line.

**Support Vector Regression:** Support Vector Regression (SVR) [5] is used while working with continuous values instead of discrete values. Here, RBF (Radial Basis Function) is used which is a non-linear kernel. The kernel functions transforms the data into a higher dimensional feature space so that the linear separation can be performed.

**Multiple Regression:** Multiple regression [6] is the modification of linear regression. The variable which is to be predicted is called the dependent variable ( $y$ ). The variables which are used to predict the value of the dependent variable are called the independent variables ( $x_1, x_2, \dots, x_n$ ). The equation of Multiple regression given in Equation 3.3.

$$y = b_1x_1 + b_2x_2 + \dots + b_nx_n + c \quad (3.3)$$

where,  $b_i$ 's ( $i = 1, 2, \dots, n$ ) are known as the regression coefficients, and  $c$  is the constant.

Here, two independent variables and one dependent variable are taken. Dependent variable is represented by the response values obtained after convolution of the input image with the edge detectors. The first independent variable is represented by the pixel number and the second independent variable as the local binary pattern (LBP) [17] of the input image. LBP is used to compute a local representation of texture.

As the regression models always try to give the least square error with the help of optimal best-fit function, therefore it would be appropriate to use these models to measure the degree of proximity of individual edge detection operator.

The proposed work consists of the following steps.

**Step 1:** Histogram Equalization [1] is initiated to correct the contrast of the given gray scale input image.

**Step 2:** Find the low -level features, edges, and sharp discontinuities by applying edge detection operators, viz. Sobel, Prewitt, Roberts, Scharr, Laplacian, Canny, and Hybrid.

**Step 3:** Plot the response values of the convolved image for each edge detection operator.

**Step 4:** Take independent variable  $x$  as the pixel number ( $1, 2, 3, \dots, n$ ) and the dependent variable  $y$  as the response values (value at each pixel) obtained after the convolution of the input image with the edge detector. For multiple regression take the other independent variable  $x_2$  as the local binary pattern of the input image.

**Step 5:** Apply Linear regression, Support vector regression and Multiple regression on the output image.

**Step 6:** Calculate root mean square error for the respective edge detector.

**Step 7:** Compare the root mean square error of the edge detectors. Edge Detector having highest root mean square error will be best suitable for the input image.

The overall idea behind the proposed methodology is reflected from the above steps and it would be advantageous to apply this mechanism while it would be difficult to measure the appropriateness of the appearance of an output image with human perception.

### 3.4 Experimental Results and Analysis

The proposed methodology has been evaluated with two publicly available databases, such as the Berkeley Segmentation Data Set and Benchmarks 300 (BSDS300) [42], and the Multi-cue boundary detection dataset [43]. The BSDS300 database consist of 300 images and it is basically used for image segmentation and object detection. To conduct the experiment, 200 images are selected randomly. On the other hand, the Multi-cue dataset consist of 100 scenes and each scene is having 20 frame instances including left and right view. For experiment, 10 scenes, each containing 20 images, are considered.

### 3.4.1 Results

The RMSE is calculated for each edge detector using three different regression models as shown in Table 3.1, 3.2, 3.3 and Table 3.4, 3.5, 3.6 on the BSDS and the Multicue dataset respectively. Table shows the results on restricted number of images as it is difficult to show the results due to space constraint. RMSE is directly related to the strength of the discontinuity. This helps us to choose mathematically the most optimal edge detector prior to an application. It has been noticed from the RMSE tables that the proposed edge detection operator is found dominating on both datasets for all three regression models. In addition, it has also been observed that both Canny and Scharr operators alternately found suitable for many images in terms of degree of proximity. Also, Laplacian and Sobel performs closely identical in most of the cases. However, the degree of proximity for Hybrid edge detector is found significantly better than other operators.

To show the regression enabled trade-off between RMSE and image using different sharpening filters, a pair of bar charts is made for each regression model determined on the BSDS300 and the Multi-cue datasets separately. To generate the bar chart, 10 images are taken from the BSDS300 dataset. Fig 3.2 (a-c) shows three bar charts determined on the BSDS300 dataset for Linear, Support Vector, and Multiple Regression respectively. From the Multi-cue dataset 200 images are taken from 10 scenes, i.e., 20 images from one scene. Fig 3.2 (d-f) shows another three bar charts for the Multi-cue dataset for Linear, Support Vector, and Multiple Regression respectively. Each bar chart appears to be a comparison between seven edge detectors that are validated through regression models. In order to have a better visualization of RMSE on different edge detectors two Y-axis are used. As the RSME for Hybrid filter (proposed one) is extremely high, all the edge detectors together in one axis cannot be observed. Therefore, two axes are required to show all the edge detectors in one bar chart. The left hand side Y-axis is for edge detectors except Hybrid operator whereas the right hand side Y-axis is for Hybrid operator. It is clear from the bar charts that the Hybrid Operator outperforms other edge detectors.

Edge Detection Operators	Linear Regression				
	RMSE values				
	Image-1	Image-2	Image-3	Image-4	Image-5
Sobel	67.16	79.24	89.44	78.56	87.45
Prewitt	59.66	69.56	79.58	67.77	78.61
Roberts	28.82	30.59	48.14	40.66	37.56
Scharr	102.34	111.46	107.01	107.84	113.31
Laplacian	67.12	81.20	91.79	86.74	92.42
Canny	92.72	117.33	120.65	119.45	121.71
Hybrid	1904.81	2420.90	3113.05	2294.91	2432.38

Table 3.1: RMSE using Linear Regression on BSDS300 dataset

Edge Detection Operators	Support Vector Regression				
	RMSE values				
	Image-1	Image-2	Image-3	Image-4	Image-5
Sobel	72.48	83.65	91.23	80.61	91.94
Prewitt	64.27	72.96	81.50	69.30	83.31
Roberts	31.25	31.73	52.54	44.10	38.83
Scharr	106.02	115.24	123.47	109.93	120.40
Laplacian	73.22	93.66	107.89	100.81	109.48
Canny	98.99	135.42	141.98	138.94	144.47
Hybrid	1982.51	2504.75	3282.15	2396.15	2535.72

Table 3.2: RMSE using Support Vector Regression on BSDS300 dataset

Edge Detection Operators	Multiple Regression				
	RMSE values				
	Image-1	Image-2	Image-3	Image-4	Image-5
Sobel	60.58	74.70	101.33	76.33	86.98
Prewitt	50.49	64.47	85.23	63.08	74.95
Roberts	21.71	29.53	37.09	29.31	19.68
Scharr	120.44	133.85	164.56	136.98	145.68
Laplacian	40.24	70.11	84.85	80.27	86.32
Canny	43.39	79.30	86.95	83.18	90.14
Hybrid	847.61	1273.96	1783.85	1130.62	1397.27

Table 3.3: RMSE using Multiple Regression on BSDS300 dataset

Edge Detection Operators	Linear Regression				
	Mean of RMSE values of 20 images				
	Multi-cue Subject-1	Multi-cue Subject-2	Multi-cue Subject-3	Multi-cue Subject-4	Multi-cue Subject-5
Sobel	80.22	81.19	82.02	82.29	78.54
Prewitt	70.40	73.58	70.41	73.14	68.84
Roberts	39.27	46.71	32.70	33.23	33.88
Scharr	110.60	112.83	112.96	110.67	108.35
Laplacian	86.82	93.62	80.08	83.48	75.49
Canny	121.03	122.44	118.64	116.22	109.73
Hybrid	2451.29	2189.57	2374.61	2429.39	2503.49

Table 3.4: RMSE using Linear Regression on Multi-cue dataset

Edge Detection Operators	Support Vector Regression				
	Mean of RMSE values of 20 images				
	Multi-cue Subject-1	Multi-cue Subject-2	Multi-cue Subject-3	Multi-cue Subject-4	Multi-cue Subject-5
Sobel	83.19	86.75	82.97	87.16	83.03
Prewitt	73.16	78.76	71.82	77.92	72.37
Roberts	41.29	50.45	33.59	34.83	35.79
Scharr	115.93	114.90	116.42	111.77	113.01
Laplacian	101.43	109.83	92.04	96.36	84.22
Canny	142.88	146.14	138.09	132.74	121.32
Hybrid	2419.95	2213.87	2618.78	2632.04	2652.86

Table 3.5: RMSE using Support Vector Regression on Multi-cue dataset

Edge Detection Operators	Multiple Regression				
	Mean of RMSE values of 20 images				
	Multi-cue Subject-1	Multi-cue Subject-2	Multi-cue Subject-3	Multi-cue Subject-4	Multi-cue Subject-5
Sobel	82.17	75.18	81.97	78.75	69.07
Prewitt	68.66	66.07	67.51	67.60	56.06
Roberts	47.46	37.48	27.05	27.83	26.05
Scharr	144.05	130.75	140.50	136.66	123.87
Laplacian	127.54	86.02	82.31	68.41	57.36
Canny	127.46	137.57	122.53	75.83	128.26
Hybrid	1444.44	1273.12	1317.54	1286.35	1114.34

Table 3.6: RMSE using Multiple Regression on Multi-cue dataset

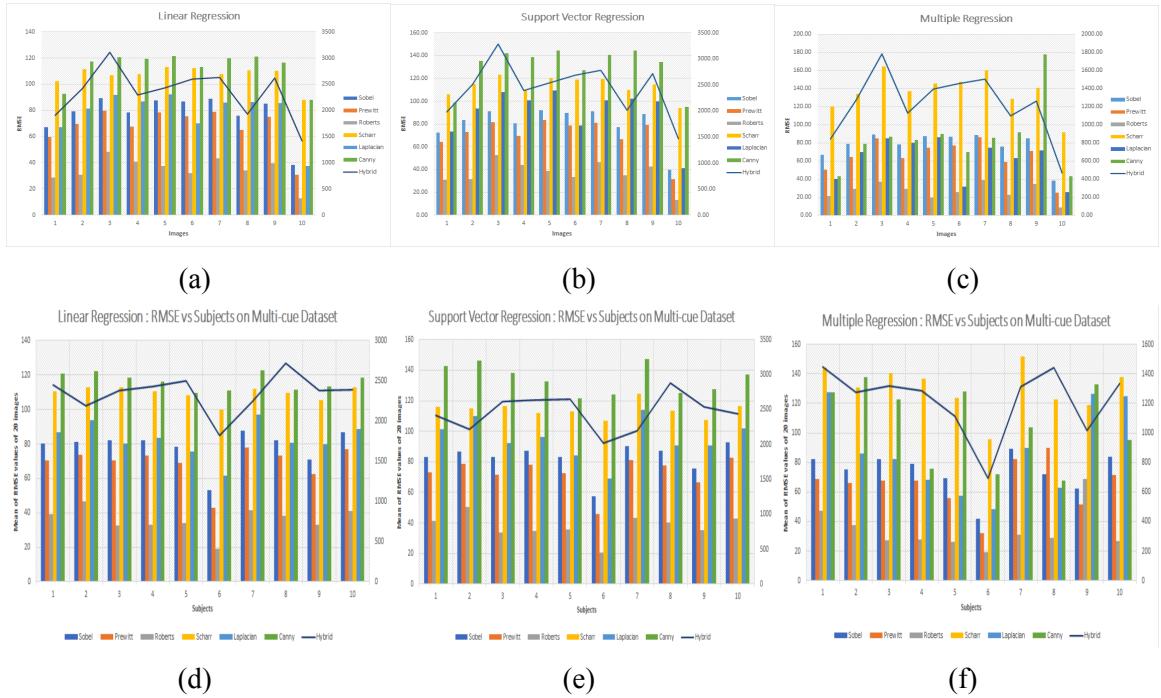


Figure 3.2: RMSE plot on BSDS300 and Multi-cue datasets (a) Linear Regression on BSDS300 dataset (b) Support Vector Regression on BSDS300 dataset (c) Multiple Regression on BSDS300 dataset (d) Linear Regression on Multi-cue dataset (e) Support Vector Regression on Multi-cue dataset (f) Multiple Regression on Multi-cue dataset

### 3.5 Conclusion

From the experiment, it has been noticed that the proposed edge detection operator i.e, Hybrid operator outperforms other edge detection operators with extremely high RMSE value on both datasets for all three regression models . RMSE is directly related to the strength of the discontinuity. This helps us to choose mathematically the most optimal edge detector prior to an application.

# **Chapter 4**

## **COVID-19 Detection on Chest X-Ray and CT Scan Images of Suspected Individuals Using CNN and Image Processing based Data Augmentation**

Coronavirus is rapidly increasing and threatening the health of millions of humans. Clinical study shows that it affects the lungs. So, lung infections can be diagnosed with the help of X-Ray and CT Scan images. As deep learning is the most effective and reliable AI technique to classify the COVID-19 screening, we proposed a model which uses Convolutional Neural Network (CNN) fused with the image processing based data augmentation. This application makes use of multiple representations of the same X-Ray and CT scan images, produced through sharpening filters viz. Sobel, Prewitt, Roberts, Scharr, Laplacian, Canny, and Hybrid, are mixed up with visible X-Ray and CT scan images for training the convolutional neural network (CNN) based deep learning model. Our proposed AI application has been tested on publicly available databases of both chest X-Ray and CT Scan images.

### **4.1 Motivation**

Coronavirus disease or COVID-19 is an infectious disease which has so far infected millions of people and deaths are increasing day by day. Due to deadly infectious nature of coronavirus, it is spreading rapidly among people who are exposed to COVID-19 infected individuals.

Due to unknown cause of pneumonia type infection and ability to generate new strain by mutation, it is almost impossible to have a cure in the form of vaccine or medicine or COVID-19 patients. Therefore, according to WHO more tests are recommended and social distancing is started in practice among people in high alert zones of different countries affected by corona pandemic. In the affected countries, reverse transcription polymerase chain reaction or RT-PCR has been adopted as standard diagnostic method to detect viral

nucleic acid as coronavirus infection in COVID-19 suspected individuals. The test takes 4-6 hours or even a whole day to give the results. As the test takes more time to generate the result compared to the time for spreading coronavirus among people and sometimes it gives false positive and false negative results, therefore, to test the COVID-19 infection rapidly and in more efficient way, chest X-Ray or/and CT scan images of COVID- 19 suspected individuals could be an answer. Moreover, the number of RT-PCR tests and shortage of test kits compared to coronavirus infected persons make it inefficient.

In contrast, X-Ray and CT scan images are widely accepted traditional form of diagnosing individuals for a number of diseases is a common practice adopted by radiologists and medics in healthcare and in medical imaging. The X-Ray and CT scan technologies have been using for several decades since its inception in medical diagnosis. In many highly affected regions or countries, it is difficult to provide sufficient number of RT-PCR test kits for testing COVID-19 infection for thousands of corona suspected people. Therefore, to address this issue, COVID detection can be made from chest X-Ray and CT scan images of corona suspected individuals who are suffering from COVID-19 symptoms.

## 4.2 Proposed Model

An AI-based software with image processing based data augmentation technique for detecting COVID-19 infection in corona suspected persons is proposed. With this integrated framework, both X-Ray and CT scan images of chest can be tested for virus detection. This application makes use of multiple representations of same X-Ray and CT scan images, produced through sharpening filters viz. Sobel, Prewitt, Roberts, Scharr, Laplacian, Canny, and Hybrid, are mixed up with visible X-Ray and CT scan images for training the convolutional neural network (CNN) based deep learning model. This deep learning model has the ability to learn the underlying pattern of COVID-19 infected X-Ray and CT scan images in a more effective way from representative images as well as original images of the same person used for training. The Figure 4.1 shows a chest X-Ray (top left) and its multiple representations of COVID-19 infected person whereas the Figure 4.2 shows a chest CT scan image (top left) and its multiple representations of COVID-19 infected person.

A convolution neural network (CNN) model is proposed to predict COVID and non-COVID cases from CT Scan and X-ray images. The proposed CNN based deep learning model uses three layers such as convolutional, pooling and fully connected layers. Two activation functions viz. RELU and sigmoid are used in the model. RELU is used after convolutional layer and sigmoid function is used for classification of test image into COVID and non-COVID classes. In training stage, the standard first-order stochastic gradient descent optimizer is used with a batch size of 32, maximum epochs 30 and binary cross-entropy based loss function. In order to alleviate overfitting of the model, data augmentation is used for training the model using image processing techniques.

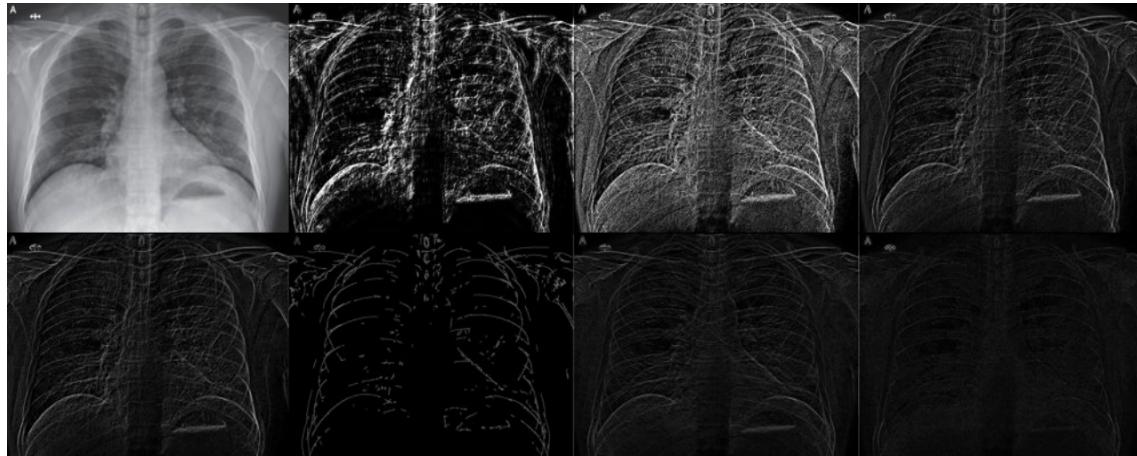


Figure 4.1: Chest X-Ray and its Multiple Representations of a COVID-19 Infected Individual.

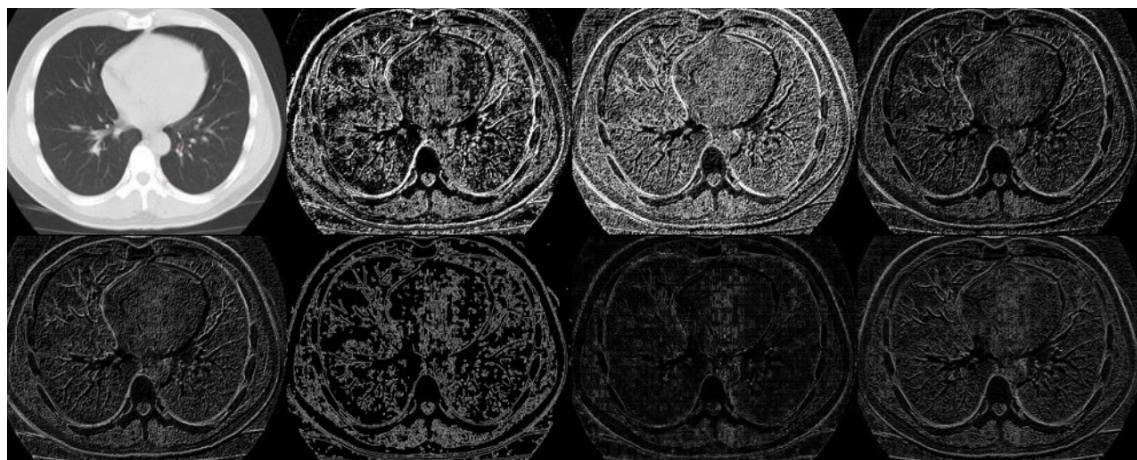


Figure 4.2: Chest CT scan and its Multiple Representations of a COVID-19 Infected Individual.

This augmentation generates large number of representative images carrying discontinuous information. Figure 4.3 shows the deep learning model with a number of parameters.

The X-Ray database contains 67 COVID images and the same number of non-COVID images whereas CT scan database contains 345 COVID images and the same number of non-COVID images. These databases contain a smaller number of images for deep-learning which does not give better accuracy. So, to resolve this issue we apply data augmentation by increasing the diversity of data available. For increasing the diversity of data, we apply seven different edge detection operators viz. Sobel, Prewitt, Roberts, Scharr, Laplacian, Canny, and Hybrid. Afterwards, we apply deep learning models for both CT Scan images and X-ray images which divided into different number of ratios. Steps for data augmentation and detection of COVID infection with the help of CT scan and X-ray images shown below:

Data Augmentation using Image Processing consists of the following steps:

**Step 1:** Accept the coloured input image from the data set.

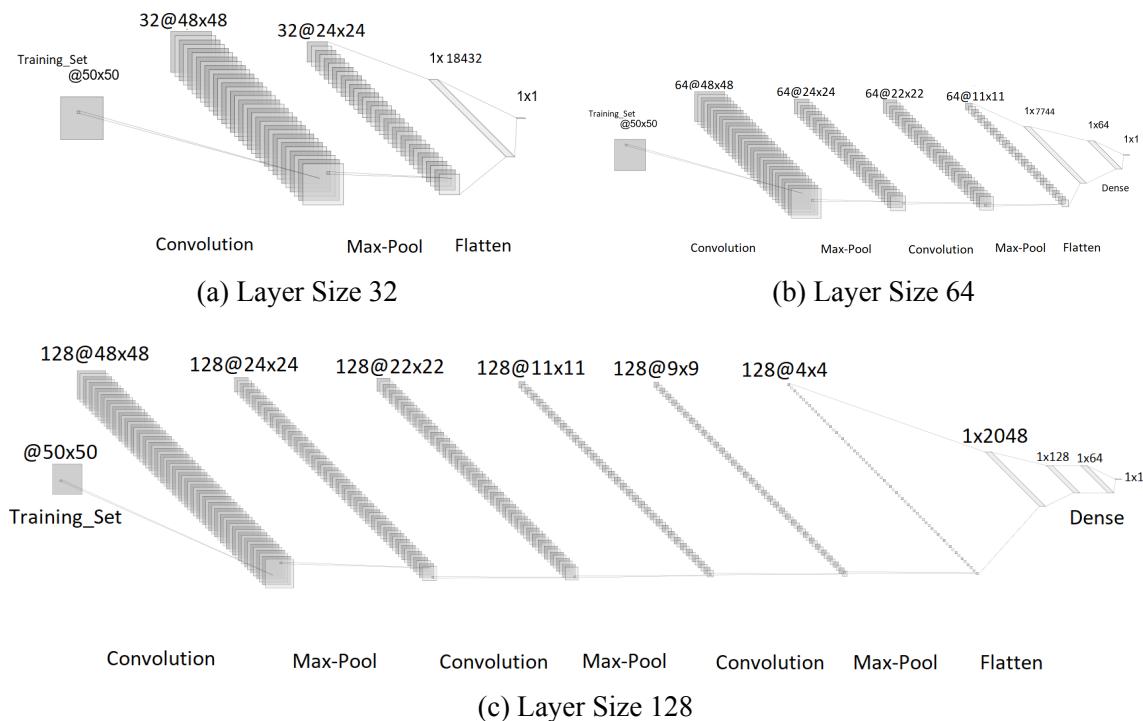


Figure 4.3: Convolutional Neural Network of Layer Size 32, 64 and 128.

**Step 2:** Convert the image into grayscale.

**Step 3:** Histogram Equalization is initiated to correct the contrast of the given grayscale input image.

**Step 4:** Find the low-level features, edges, and sharp discontinuities by applying edge detection operators, viz. Sobel, Prewitt, Roberts, Scharr, Laplacian, Canny, and Hybrid (combination of Canny and Sobel edge detector).

**Step 5:** Append/Mix the result obtained after applying Edge Detection Operators to the data set.

Steps for Detection of Coronavirus Infection in Xray and CT Scan Images of Suspected Individuals:

**Step 1:** Accept the coloured input images from the data set.

**Step 2:** Convert the image into grayscale.

**Step 3:** Down sample the images to  $50 \times 50$  dimension from their original size.

**Step 4:** In CNN based deep learning model we choose dense layers = [0, 1, 2], layer sizes = [32, 64, 128], conv layers = [1, 2, 3].

**Step 5:** Convolution with a  $3 \times 3$  filter size is applied.

**Step 6:** Activation function RELU is used after convolutional layer.

**Step 7:** Then Max Pooling is applied with  $2 \times 2$  filter size.

**Step 8:** Go to Step 5 if conv layer – 1 > 0

**Step 9:** Flatten the matrix.

**Step 10:** Activation function Sigmoid is used for classification of test image into COVID and non-COVID classes.

**Step 11:** In training stage, the standard first-order stochastic gradient descent optimizer is used with a batch size of 32, maximum epochs 30 and binary cross entropy-based loss function.

**Step 12:** The random subsampling or holdout method is adopted. Whole dataset is divided into a number of ratios like 80:20, 70:30 and 60:40 as training and testing samples.

**Step 13:** Various Evaluation Metrics are calculated such as classification accuracy, loss, area under ROC curve (AUC), precision, recall (Sensitivity), Specificity and F1 score.

**Step 14:** Once the model is trained, one can easily predict whether the individual is affected by Coronavirus infection or not.

In the training phase, we use the standard Stochastic gradient descent (SGD) algorithm with a batch size of 32 as the optimizer. We set the epochs to 30 and set the learning rate to 0.01, which is linearly decayed. Then, take training images to a fixed size of  $50 \times 50$  pixels. To alleviate the problem of overfitting to proposed model on our different set of training data, we use the data argumentation technique to enlarge the training datasets.

## 4.3 Experimental Results and Analysis

The proposed methodology has been evaluated with two publicly available databases, such as the X-Ray Dataset, and the CT Scan dataset. The databases of X-Ray [44] and CT scan [45] images are publicly available in GitHub repository for the purpose of scientific experiments. Both these datasets contain chest images of COVID-19 and non-COVID-19 individuals. The X-Ray database contains 67 COVID images and the same number of non-COVID images whereas CT scan database contains 345 COVID images and the same number of non-COVID images. To evaluate the framework in a robust and effective way, a number of evaluation metrics such as classification accuracy, loss, area under ROC curve (AUC), precision, sensitivity, specificity and F1 score have been used. The values of these metrics have been determined on different ratios of training and test samples considering a number of layers in deep model. The model is correctly able to classify the chest X-Ray and CT scan images of COVID-19 cases from non-COVID-19 cases.

### 4.3.1 Results

The whole dataset containing COVID positive and negative samples is divided into a number of ratios like 90:10, 85:15, 80:20, 75:25, 70:30, 65:35, 60:40, 55:45 and 50:50 as training and testing samples. It has been observed that when the number of training examples are increased, the model exhibits higher classification accuracy. For traditional train and test

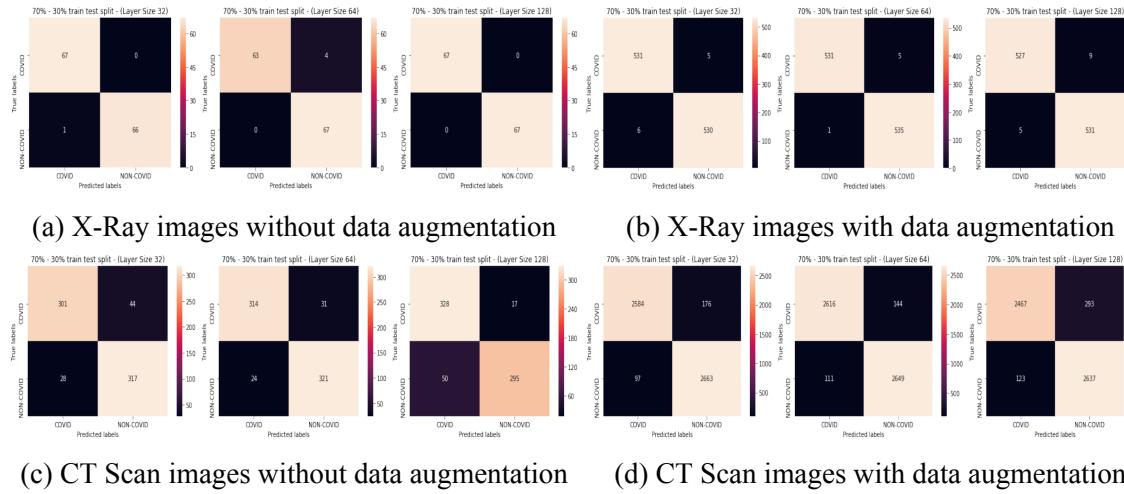


Figure 4.4: Confusion Matrix of our model when train and test ratio is 70:30

ratio of 70:30 we get classification accuracy around 95.38% and 98.97% for CT scan and X-Ray images respectively. Moreover, this result exhibits more consistency while layers are being changed in CNN based deep learning models. We randomly split the data into training and testing sets for evaluation. The experiment is conducted twice for every split and we notice that the accuracy of model comes similar on both evaluations. We have noticed while setting the layer sizes to different values F1 scores remains slightly similar and model loss (%) changes as we increase the layer size. For less training samples, smaller layer size gives good results whereas if the training samples are increasing then we require more layers for better accuracy.

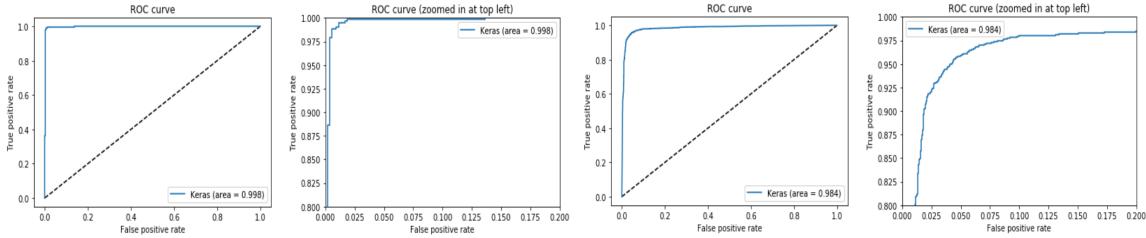
In this study, our model achieves the sensitivity of 99.07% when ratio of train and test is 70:30 & specificity of 98.88% when ratio of train and test is 70:30 and layer size 32 on the X-ray dataset that contains 536 images of COVID-19 and 536 images non-COVID-19. In the CT SCAN dataset that contains 2760 images of COVID-19 and 2760 images non-COVID-19, we achieve the sensitivity of 94.78 % & specificity of 95.98% when ratio of train and test is 70:30 and layer size 64. The results are shown in Table 4.1 for Data Augmented and original images applied on both CT Scan and X-Ray images. The model is correctly able to classify the chest X-Ray and CT scan images of COVID-19 cases from non-COVID-19 cases.

Figure 4.4 shows the confusion matrix for standard ratio of train and test i.e., 70:30 for both CT Scan and X-Ray images with data augmentation and without data augmentation and ROC curve for CT Scan and X-Ray images having higher accuracy when ratio of train and test is 70:30 as shown in Figure 4.5.

***COVID-19 Detection on Chest X-Ray and CT Scan Images of Suspected Individuals Using Chapter 4 CNN and Image Processing based Data Augmentation***

Train-Test	Original						Augmented					
	CT Scan Images			X Ray Images			CT Scan Images			X Ray Images		
	Layers		Layers		Layers		Layers		Layers		Layers	
<b>50%-50%</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>
Accuracy (%)	85.07	87.1	84.49	95.52	97.01	98.51	90.6	90.4	88.06	96.64	98.41	97.67
Loss (%)	37.15	53.79	60.73	12.35	14.82	14.38	30.58	58.41	62.73	14.49	9.04	14.63
AUC (%)	92	93.5	90.9	98.8	98.6	98.7	95.5	95.2	93.4	98.8	99.5	99.3
Precision	0.88	0.84	0.84	0.94	0.98	0.97	0.92	0.9	0.88	0.97	0.99	0.98
Sensitivity (%)	81.15	91.01	84.64	97.01	95.52	100	89.35	90.69	87.86	96.46	97.95	97.01
Specificity (%)	88.99	83.19	84.35	94.03	98.51	97.01	91.85	90.11	88.26	96.83	98.88	98.32
F1 Score	0.84	0.88	0.85	0.96	0.97	0.99	0.91	0.9	0.88	0.97	0.98	0.98
<b>55%-45%</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>
Accuracy (%)	83.19	87.54	88.55	98.51	98.51	98.51	91.92	90.62	90.47	98.69	98.79	97.67
Loss (%)	37.73	47.69	56.94	7.96	12.17	5.68	26.74	49.65	55.89	9.36	6.79	13.51
AUC (%)	92.9	93.1	94.7	99.9	98.8	99.6	96.4	95.8	95.6	99.6	99.7	99.1
Precision	0.78	0.87	0.87	0.97	0.97	0.99	0.93	0.91	0.89	0.99	0.99	0.98
Sensitivity (%)	91.88	88.7	90.72	100	100	98.51	90.25	90.04	81.96	98.5	99.44	97.39
Specificity (%)	74.5	86.38	86.38	97.01	97.01	98.51	93.59	91.19	88.12	98.88	98.13	97.95
F1 Score	0.85	0.88	0.89	0.99	0.99	0.99	0.92	0.91	0.9	0.99	0.99	0.98
<b>60%-40%</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>
Accuracy (%)	88.7	90.87	87.68	98.51	99.25	98.51	93.46	92.54	91.59	98.88	98.79	98.04
Loss (%)	31.45	36.13	65.75	6.49	2.59	13.16	22.82	48.46	65.33	8.92	10.38	12.12
AUC (%)	94.5	95.2	93.5	99.9	100	98.5	97.2	96.5	96	99.7	99.4	99.4
Precision	0.86	0.9	0.89	0.97	1	0.97	0.92	0.93	0.91	0.99	0.99	0.98
Sensitivity (%)	92.17	92.17	85.51	100	98.51	100	94.67	91.99	91.74	99.06	98.7	97.95
Specificity (%)	85.22	89.57	89.86	97.01	100	97.01	92.25	93.08	91.45	98.7	98.88	98.13
F1 Score	0.89	0.91	0.87	0.99	0.99	0.99	0.93	0.93	0.92	0.99	0.99	0.98
<b>65%-35%</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>
Accuracy (%)	88.99	89.86	90.72	97.76	97.01	97.76	93.64	93.59	92.34	98.79	98.79	98.23
Loss (%)	35.49	36.54	49.44	7.01	13.51	13.87	24.64	40.04	58.46	6.5	7.52	10.59
AUC (%)	93.4	95.2	96.1	99.8	99.7	98.7	97.1	97.4	96.3	99.7	99.6	99.6
Precision	0.9	0.92	0.91	0.98	1	0.98	0.95	0.94	0.93	0.99	0.99	0.98
Sensitivity (%)	87.83	87.54	90.14	97.01	94.03	97.01	91.66	93.37	91.74	98.51	99.07	98.32
Specificity (%)	90.14	92.17	91.3	98.51	100	98.51	95.62	93.8	92.93	99.07	98.51	98.13
F1 Score	0.89	0.9	0.91	0.98	0.97	0.98	0.94	0.94	0.92	0.99	0.99	0.98
<b>70%-30%</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>
Accuracy (%)	89.57	92.03	90.29	99.25	97.1	100	95.05	95.38	92.46	98.97	99.44	98.69
Loss (%)	29.79	26.81	50.84	5.21	12.09	2.22	19.44	26.97	37.33	7.4	1.91	8.05
AUC (%)	94.8	95.9	95.7	100	99.8	100	97.8	98.4	97.2	99.7	100	99.6
Precision	0.91	0.93	0.87	0.99	1	1	0.96	0.96	0.95	0.99	1	0.99
Sensitivity (%)	87.25	91.01	95.07	100	94.03	100	93.64	94.78	89.38	99.07	99.07	98.33
Specificity (%)	91.88	93.04	85.51	98.51	100	100	96.48	95.98	95.54	98.88	99.81	99.06
F1 Score	0.89	0.92	0.91	0.99	0.97	1	0.95	0.95	0.92	0.99	0.99	0.99
<b>75%-25%</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>
Accuracy (%)	91.16	93.19	88.26	97.76	100	97.76	95.99	95.76	94.38	99.53	99.72	98.41
Loss (%)	26.25	23.52	43.35	9.38	2.81	13.11	15.18	27.09	36.36	4.13	1.39	8.95
AUC (%)	97.1	97.1	92.6	99.8	100	100	98.3	98.4	98	99.9	100	99.8
Precision	0.96	0.93	0.91	0.96	1	0.98	0.96	0.97	0.95	0.99	1	0.98
Sensitivity (%)	86.38	93.91	85.22	100	100	97.01	96.16	94.96	93.37	99.62	99.63	98.5
Specificity (%)	95.94	92.46	91.3	95.52	100	98.51	95.83	96.56	95.4	99.44	99.81	98.32
F1 Score	0.91	0.93	0.88	0.98	1	0.98	0.96	0.96	0.94	0.99	1	0.98
<b>80%-20%</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>
Accuracy (%)	94.06	95.22	94.35	99.25	97.01	99.25	96.47	96.99	95.45	99.07	98.88	99.16
Loss (%)	18.93	17.16	41.09	3.84	6.27	12.05	13.79	16.39	21.58	7.63	5.18	4.87
AUC (%)	98.1	98	96.8	100	100	98.6	98.6	99	98.2	99.5	99.7	99.8
Precision	0.93	0.95	0.94	0.99	1	0.99	0.96	0.97	0.95	0.99	0.99	0.99
Sensitivity (%)	95.07	95.65	94.78	100	94.03	100	96.73	97.39	95.83	99.25	99.25	99.44
Specificity (%)	93.04	94.78	93.91	98.51	100	98.51	96.2	96.6	95.07	98.88	98.52	98.89
F1 Score	0.94	0.95	0.94	0.99	0.97	0.99	0.96	0.97	0.95	0.99	0.99	0.99
<b>85%-15%</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>
Accuracy (%)	92.61	96.23	97.1	99.25	99.25	100	97.68	97.9	96.23	99.63	99.35	99.25
Loss (%)	22.28	16.53	15.76	5.47	2.45	0.24	10.51	12.68	19.53	2.94	3.65	4.75
AUC (%)	97.8	98.4	99	99.4	100	100	99	99.5	98.9	100	99.9	99.9
Precision	0.9	0.97	0.97	0.99	1	1	0.98	0.98	0.95	1	0.99	0.99
Sensitivity (%)	96.23	95.65	97.68	100	98.51	100	97.17	97.83	97.28	99.63	99.44	99.25
Specificity (%)	88.99	96.81	96.52	98.51	100	100	98.19	97.97	95.18	99.63	99.25	99.25
F1 Score	0.93	0.96	0.97	0.99	0.99	1	0.97	0.98	0.96	1	0.99	0.99
<b>90%-10%</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>	<b>32</b>	<b>64</b>	<b>128</b>
Accuracy (%)	94.93	98.26	98.26	99.25	100	99.25	98.5	98.41	96.9	100	99.63	99.63
Loss (%)	20.65	9.47	11.68	3.21	1.08	2.2	7.27	8.75	13.84	3.66	2.57	2.5
AUC (%)	98.4	99.5	99	100	100	100	99.4	99.6	99	100	99.9	100
Precision	0.97	0.97	0.98	0.99	1	0.99	0.98	0.98	0.96	1	1	1
Sensitivity (%)	92.46	99.42	98.55	100	100	100	98.69	98.37	97.5	100	99.63	99.44
Specificity (%)	97.39	97.1	97.97	98.51	100	98.51	98.29	98.44	96.3	100	99.63	99.81
F1 Score	0.95	0.98	0.98	0.99	1	0.99	0.98	0.97	0.97	1	1	1

Table 4.1: Results of proposed models with Original images and Data Augmentation with CT Scan and X-Ray images datasets



(a) X-Ray images with data augmentation having 32 layers & 70:30 train-test ratio  
(b) CT Scan images with data augmentation having 64 layers & 70:30 train-test ratio

Figure 4.5: ROC curves

## 4.4 Conclusion

Fang et al. [46] shows the sensitivity up to 98% for CT Scan images of 51 patients. Further, Ophir et al. [33] implement the deep learning AI technology for the detection of COVID-19 through CT Scan images. Shi et al. [34] collects large-scale CT Scan images and implement machine learning based AI technique for COVID-19 screening by achieving the sensitivity of 90.70% and specificity of 83.30%. Whereas Jianpeng Zhang & Yutong Xie achieved the sensitivity of 90.00% & specificity of 87.84% (when  $T = 0.25$ ) or the sensitivity of 96.00% specificity of 70.65% (when  $T = 0.15$ ) on the X-ray dataset. Our model outperforms with the comparison of the above implementations. The proposed AI application has been tested on publicly available databases contributed by COVID and non-COVID individuals. The experimental results are found to be satisfactory and emerged as a useful software for COVID-19 detection on chest X-Ray and CT scan images of corona suspected population. The application can have the following usage:

1. Overcome the issues of shortage of RT-PCR kits
2. Minimize the cost of testing
3. Easy to use by diagnostic and medics persons
4. Can be used for rapid testing
5. The software application can be used both in offline and online mode

## **Chapter 5**

# **Conclusion and Future Scope**

### **5.1 Conclusion**

This thesis has proposed a novel methodology to determine the appropriateness of edge detection operators using regression models. From the experiment, it has been observed that the Hybrid operator outperforms other edge detection operators with extremely high RMSE value. It is well known that an edge detection operator which gives higher RMSE value, would be considered as efficient edge operator. Although, in practice, regression models give least square error for consideration, however in the proposed work higher RMSE values are considered to determine the suitability of edge detection operators.

Sharpening Filters can be used for data augmentation in case of detection of Coronavirus using deep learning CNN model. The main aim of using deep learning model is to attain higher accuracy of classification with chest X-Ray and CT scan images by distinguishing the COVID-19 cases from non-COVID-19 cases. For making the learning of the model about the patterns more effective, multiple representations of the same X-Ray and CT scan images are produced through sharpening filters viz. Sobel, Prewitt, Roberts, Scharr, Laplacian, Canny, and Hybrid, and then merged with the X-Ray and CT scan images to train the convolutional neural network (CNN) based deep learning model as deep learning is the most effective and reliable AI technique to classify the COVID-19 screening.

### **5.2 Future Work**

The project basically deals with measuring the suitability of an edge detection operator using Regression Models prior to an application. With a huge dataset containing similar type of images or scenes for e.g. Medical images such as X RAY and CT SCAN, similar projects can be carried out. This can have various possible applications, such as an application that can sort filters by RMSE values, filter having highest RMSE value will be best suitable for that particular database or application. For different applications, different filters can be best. It totally depends on the nature of edges. Further, this work can be extended using deep learning models with integrated framework, RNN, Autu Encoder etc so that it performs well.

The application for, detecting CoronaVirus on Chest X-Ray and CT Scan Images of Suspected Individuals Using CNN and Image Processing based Data Augmentation, overcomes the issues of shortage of RT-PCR kits, Minimizes the cost of testing, easy to use by diagnostic and medics persons, and can be used for rapid testing. The software application can be used both in offline and online mode. We are in the stage of further developing the application for, detecting CoronaVirus on Chest X-Ray and CT Scan Images of Suspected Individuals, for robustness and reliability. So that the software can be deployed at the earliest for commercial use in healthcare sectors.

# References

- [1] Rafael C Gonzalez, Richard E Woods, and Steven L Eddins. *Digital image processing using MATLAB*. Pearson Education India, 2004.
- [2] Richard Petty, Thomas M Ostrom, and Timothy C Brock. *Cognitive responses in persuasion*. Psychology Press, 2014.
- [3] George AF Seber and Alan J Lee. *Linear regression analysis*, volume 329. John Wiley & Sons, 2012.
- [4] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*, volume 821. John Wiley & Sons, 2012.
- [5] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.
- [6] Gülden Kaya Uyanık and Neşe Güler. A study on multiple linear regression analysis. *Procedia-Social and Behavioral Sciences*, 106(1):234–240, 2013.
- [7] Norman Levinson. The wiener (root mean square) error criterion in filter design and prediction. *Journal of Mathematics and Physics*, 25(1-4):261–278, 1946.
- [8] Qiang Song, Ruiqin Xiong, Dong Liu, Zhiwei Xiong, Feng Wu, and Wen Gao. Fast image super-resolution via local adaptive gradient field sharpening transform. *IEEE Transactions on Image Processing*, 27(4):1966–1980, 2018.
- [9] Parmeshwar Khurd, Claus Bahlmann, Peter Maday, Ali Kamen, Summer Gibbs-Strauss, Elizabeth M Genega, and John V Frangioni. Computer-aided gleason grading of prostate cancer histopathological images using texton forests. In *2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 636–639. IEEE, 2010.
- [10] Chu He, Shuang Li, Zixian Liao, and Mingsheng Liao. Texture classification of polsar data based on sparse coding of wavelet polarization textons. *IEEE Transactions on Geoscience and Remote Sensing*, 51(8):4576–4590, 2013.
- [11] Wikipedia. Steam engine image. [Online; accessed May 31, 2020], [https://upload.wikimedia.org/wikipedia/commons/f/f0/Valve\\_original\\_%281%29.PNG](https://upload.wikimedia.org/wikipedia/commons/f/f0/Valve_original_%281%29.PNG).

- [12] OpenCV-Python Tutorials. Non-maximum suppression, 2013. [Online; accessed May 31, 2020], [https://opencv-python-tutorials.readthedocs.io/en/latest/\\_images/nms.jpg](https://opencv-python-tutorials.readthedocs.io/en/latest/_images/nms.jpg).
- [13] OpenCV-Python Tutorials. Hysteresis, 2013. [Online; accessed May 31, 2020], [https://opencv-python-tutorials.readthedocs.io/en/latest/\\_images/hysteresis.jpg](https://opencv-python-tutorials.readthedocs.io/en/latest/_images/hysteresis.jpg).
- [14] Gilbert Tanner. Linear-regression, 2019. [Online; accessed May 31, 2020], [https://cms.gilberttanner.com/content/images/2019/09/1c0185c17f8ea09c2ef0df764bc369bfae2b5f89\\_linear\\_regression\\_explained\\_7.png](https://cms.gilberttanner.com/content/images/2019/09/1c0185c17f8ea09c2ef0df764bc369bfae2b5f89_linear_regression_explained_7.png).
- [15] Data Science. Support vector regression. [Online; accessed May 31, 2020], [https://www.saedsayad.com/images/SVR\\_5.png](https://www.saedsayad.com/images/SVR_5.png).
- [16] Stack Overflow. Multiple regression. [Online; accessed May 31, 2020], <https://i.stack.imgur.com/Tc3Y0.png>.
- [17] Mohd Safirin Karis, Nur Rafiqah Abdul Razif, Nursabillilah Mohd Ali, M Asyraf Rosli, Mohd Shahriel Mohd Aras, and Mariam Md Ghazaly. Local binary pattern (lbp) with application to variant object detection: A survey and method. In *2016 IEEE 12th International Colloquium on Signal Processing & Its Applications (CSPA)*, pages 221–226. IEEE, 2016.
- [18] TutorialsPoint. Remote sensing, 2020. [Online; accessed May 31, 2020], [https://www.tutorialspoint.com/dip/images/remote\\_sensing.jpg](https://www.tutorialspoint.com/dip/images/remote_sensing.jpg).
- [19] TutorialsPoint. Transmission, 2020. [Online; accessed May 31, 2020], <https://www.tutorialspoint.com/dip/images/transmission.jpg>.
- [20] TutorialsPoint. Hurdle detection, 2020. [Online; accessed May 31, 2020], [https://www.tutorialspoint.com/dip/images/hurdle\\_detection.jpg](https://www.tutorialspoint.com/dip/images/hurdle_detection.jpg).
- [21] TutorialsPoint. Line follower robot, 2020. [Online; accessed May 31, 2020], [https://www.tutorialspoint.com/dip/images/line\\_follow.jpg](https://www.tutorialspoint.com/dip/images/line_follow.jpg).
- [22] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.
- [23] Raman Maini and Himanshu Aggarwal. Study and comparison of various image edge detection techniques. *International journal of image processing (IJIP)*, 3(1):1–11, 2009.

- [24] Lijun Ding and Ardesir Goshtasby. On the canny edge detector. *Pattern Recognition*, 34(3):721–725, 2001.
- [25] Yu Hongshan Wang Yaonan. An improved canny edge detection algorithm [j]. *Computer Engineering and Applications*, 20, 2004.
- [26] D Poobathy and R Manicka Chezian. Edge detection operators: Peak signal to noise ratio based comparison. *IJ Image, Graphics and Signal Processing*, 6(10):55–61, 2014.
- [27] Tamilselvi Nagasankar and B Ankaryarkanni. Performance analysis of edge detection algorithms on various image types. *Indian Journal of Science and Technology*, 9(21):1–7, 2016.
- [28] Shuai Wan, Fuzheng Yang, and Mingyi He. Gradient-threshold edge detection based on perceptually adaptive threshold selection. In *2008 3rd IEEE Conference on Industrial Electronics and Applications*, pages 999–1002. IEEE, 2008.
- [29] Mukesh Kumar Rashmi and Saxena Rohini. Algorithm and technique on different edge detection technique: A survey. *SIPIJ*, 4(3), 2013.
- [30] Geng Hao, Luo Min, and Hu Feng. Improved self-adaptive edge detection method based on canny. In *2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics*, volume 2, pages 527–530. IEEE, 2013.
- [31] Mohamed Abo-Zahhad, Reda Ragab Gharieb, Sabah M Ahmed, Ahmed Abd El-Baset Donkol, et al. Edge detection with a preprocessing approach. *Journal of Signal and Information Processing*, 5(04):123, 2014.
- [32] Jianpeng Zhang, Yutong Xie, Yi Li, Chunhua Shen, and Yong Xia. Covid-19 screening on chest x-ray images using deep learning based anomaly detection. *arXiv preprint arXiv:2003.12338*, 2020.
- [33] Ophir Gozes, Maayan Frid-Adar, Hayit Greenspan, Patrick D Browning, Huangqi Zhang, Wenbin Ji, Adam Bernheim, and Eliot Siegel. Rapid ai development cycle for the coronavirus (covid-19) pandemic: Initial results for automated detection & patient monitoring using deep learning ct image analysis. *arXiv preprint arXiv:2003.05037*, 2020.
- [34] Feng Shi, Liming Xia, Fei Shan, Dijia Wu, Ying Wei, Huan Yuan, Huiting Jiang, Yaozong Gao, He Sui, and Dinggang Shen. Large-scale screening of covid-19 from community acquired pneumonia using infection size-aware classification. *arXiv preprint arXiv:2003.09860*, 2020.

- [35] Wenshuo Gao, Xiaoguang Zhang, Lei Yang, and Huizhong Liu. An improved sobel edge detection. In *2010 3rd international conference on computer science and information technology*, volume 5, pages 67–71. IEEE, 2010.
- [36] Weibin Rong, Zhanjing Li, Wei Zhang, and Lining Sun. An improved canny edge detection algorithm. In *2014 IEEE International Conference on Mechatronics and Automation*, pages 577–582. IEEE, 2014.
- [37] Anchal Kalra and Roshan Lal Chhokar. A hybrid approach using sobel and canny operator for digital image edge detection. In *2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMECTE)*, pages 305–310. IEEE, 2016.
- [38] Bob Zhang, Lin Zhang, Lei Zhang, and Fakhri Karray. Retinal vessel extraction by matched filter with first-order derivative of gaussian. *Computers in biology and medicine*, 40(4):438–445, 2010.
- [39] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.
- [40] Xiuwen Liu and DeLiang Wang. A spectral histogram model for texton modeling and texture discrimination. *Vision Research*, 42(23):2617–2634, 2002.
- [41] Tom Cornsweet. *Visual perception*. Academic press, 2012.
- [42] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.
- [43] Christian Wojek, Stefan Walk, and Bernt Schiele. Multi-cue onboard pedestrian detection. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 794–801. IEEE, 2009.
- [44] Joseph Paul Cohen, Paul Morrison, and Lan Dao. Covid-19 image data collection. *arXiv preprint arXiv:2003.11597*, 2020.
- [45] Jinyu Zhao, Yichen Zhang, Xuehai He, and Pengtao Xie. Covid-ct-dataset: a ct scan dataset about covid-19. *arXiv preprint arXiv:2003.13865*, 2020.
- [46] Yicheng Fang, Huangqi Zhang, Jicheng Xie, Minjie Lin, Lingjun Ying, Peipei Pang, and Wenbin Ji. Sensitivity of chest ct for covid-19: comparison to rt-pcr. *Radiology*, page 200432, 2020.