

KNOWLEDGE REPRESENTATION OF WIREFRAME MODEL OF 3D IMAGES USING LAYERED ARCHITECTURE

Thesis Submitted For the Award of the Degree of

DOCTOR OF PHILOSOPHY

in

Computer Science & Engineering

By

Shilpa Rani

Registration No. 41900568

Supervised By

Dr. Deepika Ghai

Co-Supervised by

Dr. Sandeep Kumar

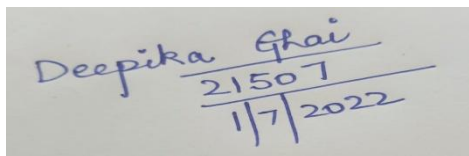


**LOVELY PROFESSIONAL UNIVERSITY
PUNJAB
2022**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
LOVELY PROFESSIONAL UNIVERSITY
Punjab, India-144411

CANDIDATE'S DECLARATION

I, hereby declared that the presented work in the thesis entitled “**Knowledge Representation of Wireframe Model Of 3D Images using Layered Architecture**” in fulfillment of degree of Doctor of Philosophy (Ph.D.) is outcome of research work carried out by me under the supervision Dr. Deepika Ghai, working as Assistant Professor, in the of Lovely Professional University, Punjab, India. In keeping with general practice of reporting scientific observations, due acknowledgements have been made whenever work described here has been based on findings of other investigator. This work has not been submitted in part or full to any other University or Institute for the award of any degree.



Deepika Ghai
21/5/22
11/7/2022

(Signature of Supervisor)



Shilpa
30/6/22

Name of the Scholar: Shilpa Rani

Registration No.: 41900568

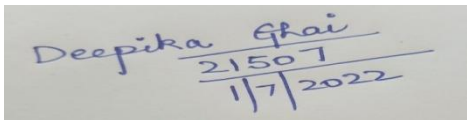
Department/school: Computer Science and Engineering

Lovely Professional University, Punjab, India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
LOVELY PROFESSIONAL UNIVERSITY
Punjab, India-144411

CERTIFICATE

This is to certify that the work reported in the Ph.D. thesis entitled **Knowledge Representation of Wireframe Model of 3D Images using Layered Architecture** submitted in fulfillment of the requirement for the reward of degree of Doctor of Philosophy (Ph.D.) in the Computer Science and Engineering, is a research work carried out Shilpa Rani, 41900568, is bonafide record of her original work carried out under my supervision and that no part of thesis has been submitted for any other degree, diploma or equivalent course.



(Signature of Supervisor)

Dr. Deepika Ghai

Assistant Professor,
Department of Electronics
& Electrical Engineering
Lovely Professional University
Punjab, India



(Signature of Co-Supervisor)

Dr. Sandeep Kumar

Professor, Department of CSE
Koneru Lakshmaiah Educational
Foundation, Vaddeswaram,
Andhra Pradesh, India

Place: Lovely Professional University, Punjab

Date: 30/06/2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
LOVELY PROFESSIONAL UNIVERSITY
Punjab, India-144411

ACKNOWLEDGEMENT

First and foremost, I would like to extend my sincerest gratitude to the almighty God without whom this work would not have been possible. I would like to express my deep sense of gratitude and sincere thanks to my supervisor Dr. Deepika Ghai and Dr. Sandeep Kumar for their invaluable guidance. Their enduring continual guidance and support motivated me in each and every phase of my research as well as in life. I am grateful to them for teaching me lots of things that not only led my thesis to fruition but also inculcated good researcher traits in me.

Prof. Dr. E. G. Rajan, President, Pentagram Research Centre Private Limited, Hyderabad and Director, Avatar MedVision US LLC NC USA, has been instrumental in bringing the research findings to light.

I would like to thank my parents, Sh. Ranveer Singh and Smt. Sunita Choudhary for their emotional support, love and blessings, who have worked hard day and night to make me what I am today. I am thankful to my in law's and my beloved Husband, Kumar Anand, for his persistent support, love, affection, understanding and care throughout this journey. The biggest thanks go to my younger brother Manish Kumar for his unconditional support, encouragement and motivation throughout all these years. A special thanks to my son, Reyansh Anand, who encouraged me to pursue excellence.

I would also like to thank my Research Committee Members for providing insightful and constructive comments. I would like to thank my friends and well-wishers who helped me directly or indirectly throughout the period of my Ph.D. research at Lovely Professional University, Punjab.

Place: Lovely Professional University, Punjab

(Shilpa Rani)

Date: 30/06/2022

SID: 41900568

LIST OF FIGURES

Fig. 1.1: Growing Areas of Computer Vision.....	1
Fig. 1.2: Categorization of Object Recognition	3
Fig. 1.3: Example of Appearance Based Recognition and Geometric Based Recognition	4
Fig. 1.4: Example of 3D Object Recognition	4
Fig. 1.5: Descriptor Based Method. Examples of Descriptive Strategy [5] a) Original Image, b) Interest of Regions are Detected Represented by Blue Circles, c) Region Descriptor for Each Region of Interest	5
Fig. 1.6: Number of Published Paper on 3D Object Recognition.....	6
Fig. 1.7: Object Recognition in Medical Imaging	9
Fig. 1.8: Object Recognition in Sports Tracking System	10
Fig. 1.9: Object Recognition in Manufacturing Industry.....	11
Fig. 1.10: Object Recognition in Automation.....	11
Fig. 1.11: Object Recognition in Battlefield.....	12
Fig. 1.12: Example of Reflection.....	12
Fig. 1.13: Less Textured Image Courtesy [30]	13
Fig. 1.14: Transparent Surface.....	13
Fig. 1.15: Illumination Variation Courtesy [31]	14
Fig. 1.16: Scale Variation in Shoes.....	14
Fig. 1.17: Rotation Variance.....	15
Fig. 1.18: Same Object with Different Background	15
Fig. 2.1: Steps of Local Feature Extraction	18
Fig. 2.2: Steps of Global Feature Extraction.....	19

Fig.3.1: Flow Chart of Proposed Methodology	48
Fig. 4.1: Cellular Logical Array Processing	52
Fig. 4.2: Sample Images of Wireframe Model [165].....	52
Fig.4. 3: Various Plane Representations of 3D Images	53
Fig. 4.4: Basic steps of proposed work	54
Fig. 4.5: Tabulation of Medical Images using Median Filter, Adaptive Mean Filter and Adaptive Median Filter	56
Fig. 4.6: Sixteen Convex Patterns in a 3x3x3 Windows.....	57
Fig. 4.7: Wireframe Model of the Input Images on ModelNet 40, PIX3D and PASCAL 3D+ Dataset	61
Fig. 4.8: Wireframe model of the input images on Fig-share and Kaggle Brain MRI Dataset.....	62
Fig. 4.9: Wireframe model and their different 3D View of the Input Images on Medical MRI Dataset.....	64
Fig.4. 10: Comparative Analysis of Wireframe Model of the Images on Example- Based Synthesis Dataset, Model-Net 40 Dataset and Medical MRI Dataset	65
Fig.4. 11: The Input Image and its Processed Results on Different Algorithms	66
Fig. 4.12: Processing Time of Different Algorithms on Cerebrix, Toutarix, Skull and GPR VSP Dataset	68
Fig. 5.1: Flow Chart of the Proposed Methodology	72
Fig. 5.2: Possible form of juxtaposition.....	75
Fig. 5.3: Pictorial Representation of a) Center Plane b) Back Plane and c) Front Plane with Corresponding Pixel Directions.....	76
Fig. 5.4: a) 27-Neighbourhood b) 7-Neighbourhood c) 9-Neighbourhood d) 15- Neighbourhood e) 19-Neighbourhood Structuring Elements	78
Fig. 5.5: a) 27-Neighborhood Structure of 3x3x3 Windows b) Representation of Co- ordinates.....	79

Fig. 5.6: Three Dimensional Visualizations of Convex Polyhedrons.....	92
Fig. 5.7: GUI for Creating a Knowledge Vector from Image.....	99
Fig. 5.8: GUI for displaying an original image from knowledge vector	102
Fig.5.9: Comparative Analysis of Accuracy and Execution Time	115
Fig. 6.1: Proposed Technique Diagram	117
Fig. 6.2: 3D Alex-Net Architecture	125
Fig. 6.3: 3D Convolution	126
Fig. 6.4: Convolution Operations	127
Fig. 6.5: 3D Max-pooling Operation	128
Fig. 6.6: Parametric ReLU	128
Fig. 6.7: GUI of Proposed Work.....	132
Fig.6. 8: Py-Plots of Accuracy, Loss, Val_Accuracy, Val_Loss on Figshare Dataset.....	135
Fig. 6.9: Comparative Analysis of Accuracy, Precision and Recall on Figshare Dataset.....	137
Fig. 6.10: Comparative Analysis of Specificity and F1-Score on Figshare Dataset..	137
Fig. 6.11: Comparative Analysis of Network Parameters of Existing Systems and Proposed System on Figshare Dataset	138
Fig. 6.12: Comparative Analysis of the Proposed Method with RCNN Based Approach on Figshare Dataset	139
Fig. 6.13: Py-Plots of Accuracy, Loss, Val_accuracy, Val_loss on Kaggle brain MRI dataset	140
Fig. 6.14: Comparative analysis of proposed Work with Existing State of Art Method on Kaggle Brain MRI dataset and Medical MRI Dataset	141
Fig. 6.15: Py-Plots of Accuracy, Loss, Val_Accuracy, Val_Loss on Medical MRI Dataset.....	142

Fig. 6.16: Py-Plots of Accuracy, Loss, Val_Accuracy, Val_Loss on BraTS 2019 Dataset.....	142
Fig. 6.17: Comparative Analysis of Accuracy on BraTS 2019 Dataset	144
Fig. 6.18: Comparative Analysis of Proposed Work on BraTS 2019 Dataset.....	145

LIST OF TABLES

Table 2.1: Comparative Analysis of Local Features Extraction Based Methods.	20
Table 2.2: Comparative Analysis of Global Features Extraction Based Methods.	24
Table 2.3: Study of Existing Methodology on Reconstruction of Images.....	34
Table 2.4: Comparative Analysis of Different Syntactic Pattern Recognition Methods.....	40
Table 2.5: Study of existing methodology on Brain Tumor Detection and Classification.....	43
Table 4. 1: PSNR (in dB) Comparison between Median Filter, Adaptive Mean Filter, Adaptive Median Filter	57
Table 4.2: Comparative Analysis of the Proposed Algorithm on Medical MRI Dataset.....	67
Table 4.3: Comparative Analysis of the Proposed Algorithm on Medical MRI Dataset.....	67
Table 4.4: Comparative Analysis of Time Complexity	68
Table 5.1: Set of Production Rules	75
Table 5.2: Representation of a Possible Combination of Convex Polyhedrons	80
Table 5.3: Description of Datasets.....	103
Table 5.4: Image reconstruction with Toutatrix Dataset	105
Table 5. 5: Image Reconstruction using Cetautomatix Dataset.....	106
Table 5.6: Image Reconstruction with Toutatrix Dataset	108
Table 5. 7: Image Reconstruction using Cerebrix Dataset	109
Table 5.8: Reconstruction of image on Toutatrix Dataset	111
Table 5.9: Image Reconstruction using Five Medical Datasets.....	112
Table 5.10: Values of PSNR and SSIM on Various Medical Datasets	114

Table 5. 11: Proposed and Current Approaches Comparison for Accuracy and Execution Time on Kaggle Brain MRI Dataset	114
Table 5.12: Proposed and Current Approaches Comparison for Accuracy and Execution Time on Five Medical Datasets	114
Table 6.1: Basic Patterns in a 3D Image	118
Table 6. 2: Details of Improved 3D Alex-Net Classifier	132
Table 6.3: List of Evaluation Parameters and their Formula	133
Table 6.4: Result Analysis of Proposed Work with other State-of-Art Methodology on Figshare dataset.....	136
Table 6.5: Result Analysis of Proposed Work with other State-of-Art Methodology on Figshare Dataset.....	137
Table 6.6: Performance Comparison of the Proposed Method with Existing RCNN Based Methods on Figshare Dataset	139
Table 6.7: Result Analysis of Proposed Work with other State-of-Art Methodology on Brain Tumor Kaggle Dataset and Medical MRI Dataset.....	141
Table 6.8: Result analysis of proposed work on BraTS 2019 Dataset.....	143
Table 6.9: Result Analysis of Proposed Work with other State-of-art Methodology on BraTS 2019 Dataset	143

LIST OF ABBREVIATIONS

3D-DWT	Three Dimensional Discrete Wavelets Transform
LBP	Local Binary Pattern
GLCM	Gray Level Co-occurrence Matrix
EDMS	Edge Direction Matrixes
CVFH	Clustered Viewpoint Feature Histogram
GAK	Global Alignment Kernels
GGFM	Global Geometric Feature Map
HOG-TOP	Histogram of Oriented Gradients from Three Orthogonal of Planes
S3DRGFs	Spatial 3-D Relational Geometric Features
3DCS-LBP	3D Center-Symmetric Local Binary Patterns
SLKP	Spatial Local Key Point
LDFA	Local Deep Feature Alignment
GOOD	Global Orthographic Object Descriptor
GASD	Globally Aligned Spatial Distribution
GFH	Global Fourier Histogram
VFH	Viewpoint Feature Histogram
OUR-CVFH	Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram
GSH	Global Structure Histograms
DA	Discriminant Analysis
SDVS	Shape Distribution on Voxel Surfaces
PCA	Principal Component Analysis
BPNN	Backpropagation Neural Network

GRNN	General Regression Neural Network
SSR	Scale Retinex Algorithm
PCANet	Palmprint Recognition using Unsupervised Convolutional Deep Learning Network
HSA	Hybrid Simulated Annealing
WLPCA	Widely Linear PCA
SVM	Support Vector Machine
3DDFA-d	3D Dense Face Alignment
CNN	Convolution Neural Network
Isomap-E-R	Isomap-Eigenanalysis-Regression
TLE	Trimmed Likelihood Estimator
HMC	Hidden Markov Chains
RMSEs	Root-Mean-Square Errors
RSEFNN	Recurrent Self-evolving Fuzzy Neural Network
SVR	Support Vector Regression
SONFIN	Self-Organizing Neural Fuzzy Inference Network
PR	Pattern Recognition
TP	True Positive
FN	False Negative
FP	False Positive
TN	True Negative
FPR	False Positive Rate
FNR	False Negative Rate
PSNR	Peak Signal to Noise Ratio
MSE	Mean Squared Error
AT	Average Time

AR	Average Recall
AP	Average Precision
AC	Average Accuracy
ANN	Artificial Neural Network
LDA	Linear Discriminant Analysis
NN	Neural Network
CNN	Convolution Neural Network
HOG	Histogram of Oriented Gradient
AMF	Adaptive Median Filter
3D- Alex-Net	Three Dimensional Alex-Net
R	Right
DR	Down Right
D	Down
DL	Down Left
L	Left
UL	Up Left
U	Up
UR	Up Right
B	Back
BR	Back Right
BDR	Back Down Right
BD	Back Down
BDL	Back Down Left
BL	Back Left
BUL	Back up Left
BU	Back up

BUR	Back up Right
F	Front
FR	Front Right
FDR	Front Down Right
FD	Front Down
FDL	Front Down Left
FL	Front Left
FUL	Front up Left
FU	Front up
FUR	Front up Right

TABLE OF CONTENTS

	Page No.
<i>Candidate's Declaration</i>	i
<i>Certificate</i>	ii
<i>Abstract</i>	iii
<i>Acknowledgment</i>	v
<i>Table of Contents</i>	vi
<i>List of Figures</i>	ix
<i>List of Tables</i>	xiii
<i>List of Abbreviations</i>	xv
Chapter 1: Introduction	
1.1 Introduction to Computer Vision	1
1.2 Introduction of Pattern Recognition	1
1.3 Object Recognition	2
1.3.1 Types of Object Recognition	2
1.4 3D Object Recognition	6
1.4.1 Motivation of 3D Object Recognition	7
1.4.2 Applications of 3D Recognition	8
1.4.3 Challenges of 3D Object Recognition	12
1.5 Summary	16
1.6 Thesis organization	16
Chapter 2: Literature Survey	
2.1 3D Object Recognition Based on Local and Global Feature-Based Methods	17
2.1.1 Local Feature Extraction	18

2.1.2 Global Feature Extraction	19
2.2 Literature Survey on Wireframe Model	31
2.3 Literature Survey on Reconstruction of Images	32
2.4 Reviews on Structural/ Syntactic Pattern Recognition Technique	37
2.5 Survey on Brain Tumor Classification Methods	42
2.6 Summary	44
Chapter 3: Research Objectives and Methodology	
3.1 Research Gap	46
3.2 Research Objectives	46
3.3 Research Methodology	47
3.3.1 Steps of Proposed Methodology	47
Chapter 4: 3D Wireframe Model of Images	
4.1 Introduction	51
4.2 Proposed Methodology	53
4.2.1 De-noising of the Image using Adaptive Median Filter	54
4.2.2 Image Enhancement	55
4.2.3 Construction of Wireframe Model	56
4.3 Analysis of Results	58
4.3.1 Datasets	58
4.3.2 Evaluation Parameters	60
4.3.3 Results on Different Datasets	60
4.4 Summary	69
Chapter 5: Construction and Reconstruction of 3D Images	
5.1 Introduction	70
5.2 Proposed Methodology	72
5.2.1 Knowledge Acquisition	72

5.2.2 Feature Extraction	92
5.2.3 Reconstruction of Original Three-Dimensional Image using Knowledge Vector	98
5.3 Analysis of Results	102
5.3.1 Datasets	102
5.3.2 Evaluation Parameters	103
5.3.3 Results	104
5.4 Summary	115
Chapter 6: 3D Object Recognition	
6.1 Introduction	116
6.2 Proposed Methodology	117
6.2.1 Knowledge Representation of the Image	118
6.2.2 Object Recognition using improved 3D Alex-Net	124
6.3 Analysis of Results	131
6.3.1 Datasets	131
6.3.2 Evaluation Parameters	133
6.3.3 Results	134
6.4 Summary	145
Chapter 7: Conclusion and Future Scope	
7.1 Conclusion	146
7.2 Future Scope	147
Bibliography	148
List of Publications	170

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
LOVELY PROFESSIONAL UNIVERSITY
Punjab, India-144411

ABSTRACT

In the past six decades, object identification has made significant progress. However, 3D object recognition remains a relatively less explored field. Due to the growing growth of three-dimensional data, it is vital to have a classification and recognition system to categorize three-dimensional objects accurately. This research's primary objective is to develop a novel technique for finding a wireframe model of three-dimensional images, which could be utilized for object classification. To achieve this objective, the suggested system's complete structure is organized into three modules: wireframe model, feature extraction, and classification.

In the first module, a Cellular logic array processing (CLAP) based algorithm is implemented to find the wireframe model of 3D images. To achieve this objective, preprocessing techniques are applied to the image to remove the noise from the image and enhance the quality of the input image. Further, a unique approach is developed based on cellular logic array processing (CLAP) to extract volumetric information from three-dimensional images and to identify the wireframe model of the 3D images. To analyze the performance of the proposed algorithm, various standard datasets like PASCAL 3D+, ModelNet 40, PIX 3D, Figshare dataset, Kaggle Brain MRI, Medical MRI and BraTS 2019 dataset are used. The proposed algorithm's results were compared to existing edge detection algorithms regarding processing time, temporal complexity, and information preservation. According to the result analysis, the suggested method is more accurate than existing state-of-the-art approaches.

The second module of this research proposes a novel syntactic pattern recognition-based approach to extract the features from the image. The proposed algorithm takes the wireframe model as an input image and extracts the features from the image and these features (Knowledge vector) consist of direction code and length. In addition, a

unique and novel algorithm is developed that can rebuild an image using a knowledge vector. Reconstruction allows us to investigate the interior details of 3D images, such as the object's size, form, and structure. In the world of clinical image processing, this could help us advance. The proposed algorithm performance is assessed on a medical imaging dataset, and the findings are outperformed. This method is superior to other methods since it involves only a few steps in reconstructing a 3D image.

The third contribution of this research is the invention of a novel method for identifying and classifying brain tumors using improved 3D Alex-Net architecture. According to the result analysis, the proposed framework is more efficient and accurate than existing state-of-the-art approaches. Four standard brain tumor datasets were considered to assess the performance of the proposed model. On the Figshare dataset, Kaggle Brain MRI dataset, Medical MRI dataset, and BraTS 2019 dataset, the proposed method's accuracy is 99.20%, 99.25%, 99.56%, and 97.12%, respectively, with recall values of 98.91%, 94.47%, 95.74%, and 93.8%. Experiments demonstrate that the proposed brain tumor classification model outperforms state-of-the-art approaches in accuracy and precision.

In the current research, 3x3x3 size convex polyhedron is used. In the Future, convex polyhedrons could be carried out in the neighborhood of 5x5x5, 7x7x7, 9x9x9, etc. Furthermore, the proposed system could combine with the existing cancer diagnosis tools to improve the robustness of the current CAD-based systems.

CHAPTER 1

INTRODUCTION

1.1 Introduction of Computer Vision

Computer vision has become one among the foremost appealing fields of Research. Computer vision is a field of computer science and it allows the computer to see, identify and recognize the image and provides the desired result. Computer vision is performing the same task which is done by human eyes. It's a technique of transmitting the human intelligence and intuitive way of thinking to computer. Automatic recognition, description, classification and grouping of patterns play an important role in the field of biological, biomedical, medicine, marketing, computer vision, artificial intelligence and remote sensing. **Fig.1.1** represents the various fields where computer vision plays an important role.

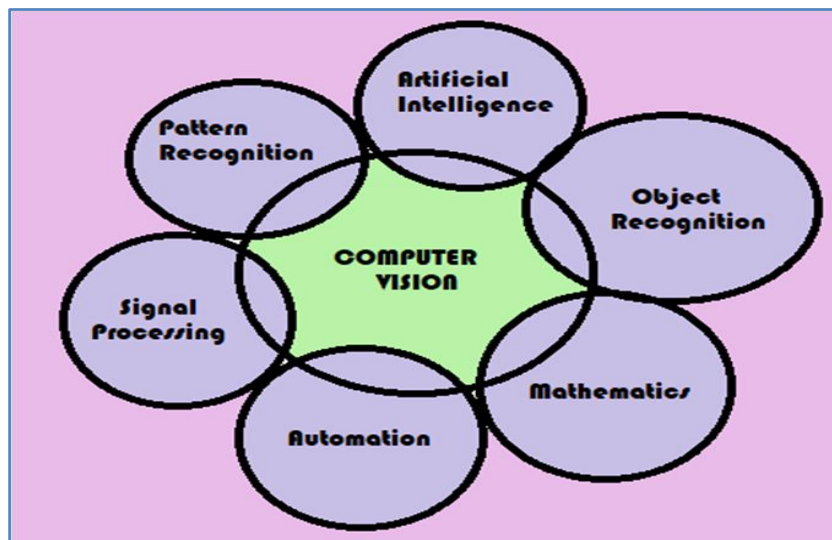


Fig. 1.1: Growing Areas of Computer Vision

1.2 Introduction of Pattern Recognition

Pattern recognition is a technique of recognizing the arrangement of patterns that gives specific information about the given dataset. The easiest way to understand pattern recognition is regular expression matching which is generally used in word processors and text editors. This technique is related to machine learning [1], artificial intelligence, data mining or in other words we can say that machine learning or

artificial intelligence are the approaches to solve the problems which are related to pattern recognition. Generally, Pattern recognition algorithms are applied to data, and these techniques will automatically recognize or uncover the similarity in data. With the help of these similarities, data can be easily classified into different categories. In other terms, pattern recognition is a technique that assigns a tag or label to the input dataset. According to the researcher's classification, clustering, regression analysis [1], sequence labeling and parsing [2] are examples of pattern recognition. Classification is more popular with the name of supervised machine learning and clustering, sometimes known as unsupervised machine learning. Pattern recognition is playing an important role in the field of medical, face recognition, speech recognition, text recognition, image recognition, handwriting recognition etc. [3].

1.3 Object Recognition

Object recognition is a method of finding the objects in the image or in the sequence of images. Human eyes can differentiate the objects easily that are different in size, shape, texture etc. Human brain can distinguish among the objects because human have better brain intelligence. Generalization of the objects is easy for human but it tough for the computer system because computer don't have intelligence like humans. Classification of the object which belongs to different category is always a challenging problem for computer and always this problem attracts the researcher to do something new in the field of object recognition.

1.3.1 Types of Object Recognition

In last three decades many object Recognition Techniques have been developed. These techniques can be classified into different categories according to their characteristics. **Fig. 1.2** represents the different categories of object recognition and description is given below:

- **Geometric Based Object Recognition**

Geometric based object Recognition method focus on recognition of the shape, color and texture of the image. The main agenda of this approach is to identify the contour the image. It involves the detection and capturing the boundary points of the objects which provides the information about the shape of the object [4].

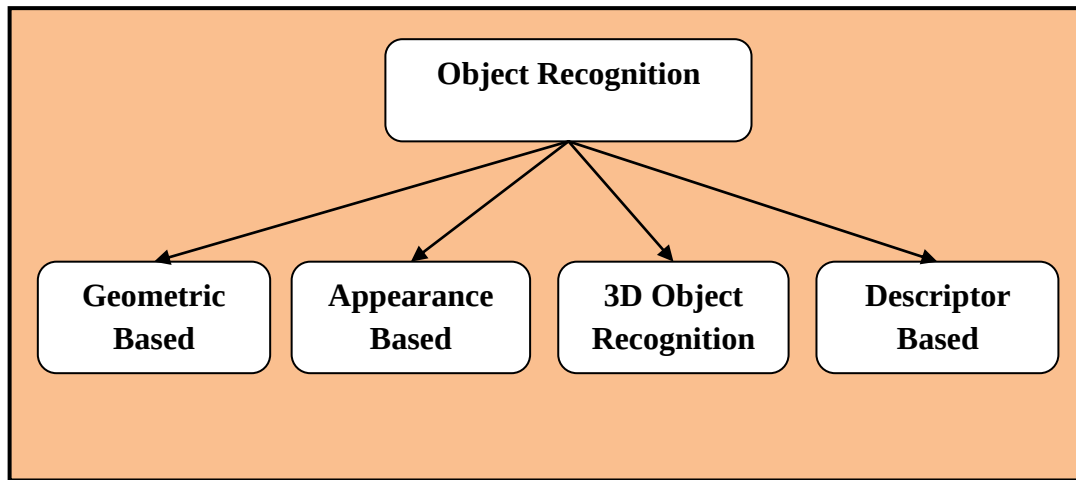


Fig. 1.2: Categorization of Object Recognition

- **Appearance Based Methods**

Most of the object recognition algorithm is based on gathering the information from a single image. But a single image can give only limited features which is not sufficient to discriminate one object with the other object. There is a possibility that a dataset can have similar type of objects or they have similar representation. The solution of this problem is to capture the image of same object with different viewpoints and this type of recognition comes under appearance based recognition [4]. In appearance based algorithm intensity of the pixel is considered as a feature of the object in the image. Pixel intensity is directly proportional to the brightness of the light emitted from the object along certain rays in space. Collection of all such radiance is known as light field. This technique will take pixel vector as an input and it will compute eigenvectors and each eigenvector represents one object. Each Eigen vector consists almost all the appearance variation of the object [4]. In other words, each face image can be represented by a linear combination of eigen-faces with minimum error, and this linear combination constitutes a compact reorientation and it is represented in the **Fig. 1.3**.

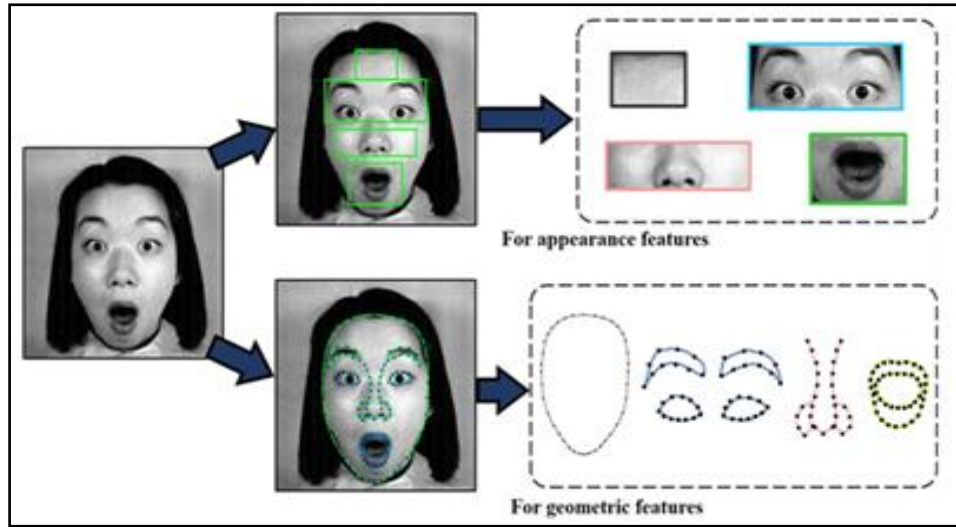


Fig. 1.3: Example of Appearance Based Recognition and Geometric Based Recognition

- **3D Object Recognition Method**

2D image can be represented on paper or it can display on the computer screen easily.

2D image can be represented with the help of two points (x,y) with function $f(x,y)$.

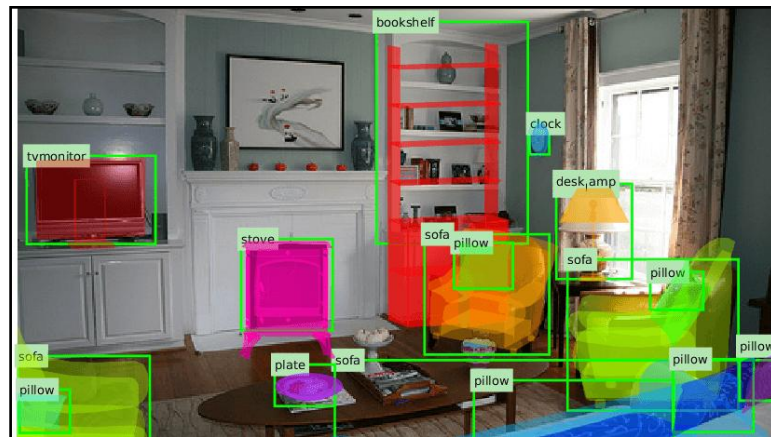


Fig. 1.4: Example of 3D Object Recognition

Here f is a real function which represents the density, transmissivity, reflectivity; intensity etc. 3D images are extension of 2D image where 3D image can be represented with real function $f(x,y,z)$. Here z represents the density of the image. 3D space can be represented through small a cube which is shown in the **Fig. 1.4**. Each cube will be having x,y,z axis and the value of the function f averaged over each cube is assigned to it. In the case of 2D each (x,y) coordinate is called as pixel but in the case of 3D (x,y,z) coordinates are referred as voxels or volumetric cells. So density

of 3D image will make it different from 3D image. In last few years 3D object recognition and classification is growing field of research. 3D object recognition method is used to recognize the 3D object from the image and it will retrieve the information from the object which is required for a particular application. Most of the 3D image processing applications use sensors. Most of the system uses range images or depth maps. Range image is having pixel values corresponds to the distance. In range image information about the z-direction is stored dependent on the (x,y) position in the image plane. This type of data representation is not 3D representation and therefore it is called 2.5D [5].

- **Descriptor Based Methods**

Sometimes geometric algorithms are not suitable for particular dataset. Because these techniques are helpful for identifying the shape of the object and sometimes shape information will not be enough when dataset is containing huge amount of same objects.

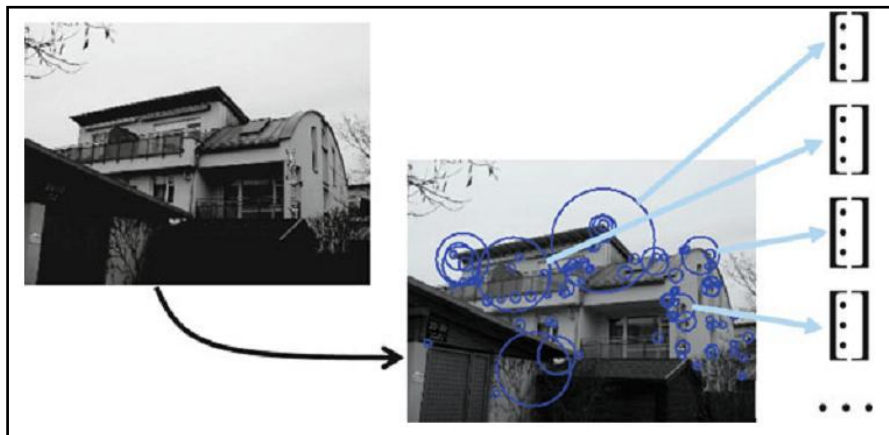


Fig. 1.5: Descriptor Based Method. Examples of Descriptive Strategy [5] a) Original Image, b) Interest of Regions are Detected Represented by Blue Circles, c) Region Descriptor for Each Region of Interest

There is possibility that images are having heavy background occlusion or clutter then appearance based algorithms are not suitable for such scenario.

To overcome from these problems Schmid and Mohr [5] suggested a new technique for the description of the image. This is divided into two steps:

- i. Detection of Region of interest from the image.
- ii. Creating a feature vector (region descriptor) for region of interest.

Each region descriptor characterizes the image information available in a local neighborhood around one interest point and it is shown in the **Fig. 1.5**.

1.4 3D Object Recognition

The main objective of the 3D object recognition system is to identify and locate the object in a given image. Tremendous work has been done by different researchers in the field of 2D object recognition. Now the whole world is talking about three-dimensional object recognition. As 3D Object recognition is a growing research area of computer vision and many researchers are working on this. According to our poll and the data from IEEE Explore, the number of papers published by the research community each year is rising, as seen in **Fig.1.6**.

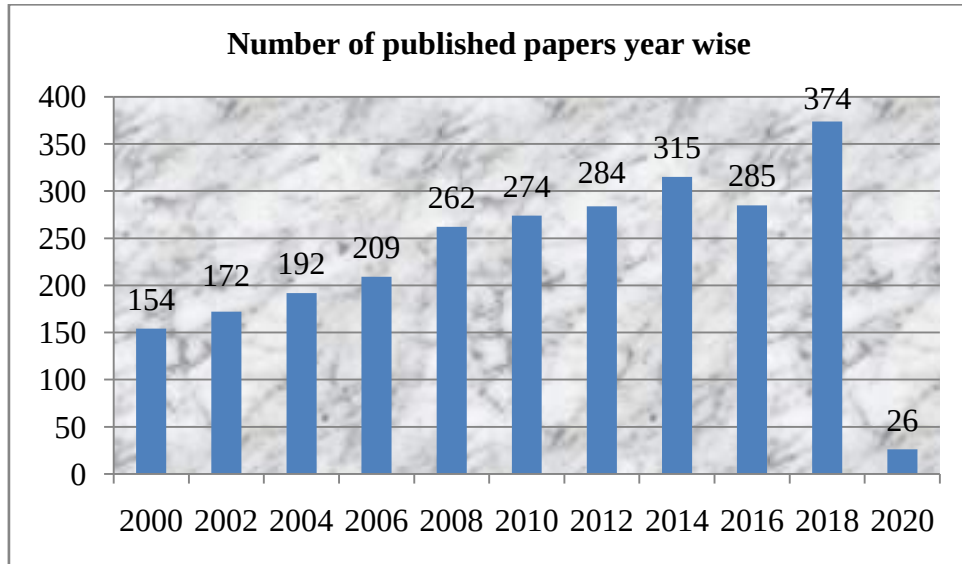


Fig. 1.6: Number of Published Paper on 3D Object Recognition

There are many applications where 3D images are playing a big role. These applications are mobile manipulation, robotics, medical field, biometric analysis and intelligence surveillance system etc. [6-10]. Compare to 2D images, 3D images (range image) will provide more geometric information. The features which are extracted from the 3D image are not affected by illumination, scale or rotation etc [11]. The pose which is estimated from the 3D image is a more accurate comparison to the pose which is estimated from the 2D image. We can conclude that the range / 3D images can overcome the problem, which is generally faced by the different researchers in 2D Object Recognition [12]. All the reasons, as mentioned earlier, make 3D object

recognition as open and active research area [13]. According to the extraction of the feature, this method is categorized into two parts:

- The global feature-based recognition method
- Local feature based recognition method

The global feature-based method considers the image as a whole [14], and the whole 3D object can be represented as a vector. These methods are more suitable for retrieving the 3D model and identifying the shape of the object [15-16] when the geometric structure of the object is fragile. This method helps identify the contour, shape descriptors, and texture feature of the image. But this method fails if the image is more cluttered or occlusion is present in the image [17]. This method includes geometric 3D moments [18], shape distribution [19], viewpoint feature histogram [20], and potential well space embedding [21].

Local feature based technique is more suitable for the real-world scene where clutter and occlusion [17] are present frequently. This method extracts only local features around specific key points. It includes surface signature [22], SIFT, SURF, LBP, BRISK, 3D Tensor [23], THRIFT [24] etc. even these methods have given better results in the field of 2D object recognition [25].

Fundamentally both representations are different, and both the representations will provide different information. Both methods require a different type of classifier for classification. Dimitri A. Lisin et al. [26] proposed a method for combining local and global features. In this proposed method, authors have performed the classification process using SVM & NPD. The local or global feature method alone is giving an accuracy of around 54%, whereas the performance of the combined method is 65.5% with a reduction of an error rate of 20%. Authors have applied this technique to marine science data [27-28] and the dataset consists of an 1826 grayscale image which belongs to 14 different classes.

1.4.1 Motivation of 3D Object Recognition

The Main motive of object recognition is to recognize and locate all known objects in a scene. The 3-D digital image data obtained with various scanners are essentially the characteristic reflections of various components within the scanning area of particular

interest. To make meaningful interpretations, one must be able to visualize the hidden parts of a 3-D image. So, processing 3-D image data is all about extracting physical, morphological, and structural properties from the image's hidden parts. There are many applications where 3D images are playing a big role. Some of the motivation behind 3D object recognition is discussed below:

- Preferably in 3D space, identifying pose of objects in 3D is very important for robotic control systems. The information obtained from the 3D object recognition system can be helpful for avoiding the obstacles and it provides an easy way to interact with the environment.
- Three-dimensional digital image processing is a vital topic of study with extensive applications in medical imaging, seismology, meteorology, and material sciences. In order to solve a certain problem, one must detect distinct features of three-dimensional images while employing appropriate algorithms.
- Extraction of volumes and forms of certain physical body parts is required for quantitative chemical analysis of anatomical structures in MRI/CT images. To do so, three-dimensional images must be analyzed for elements such as points, lines, contours, surfaces, wireframes, textures, and skeleton structures. Detecting 3D objects and surface and volumetric features in three-dimensional images has always been difficult, and little advances have been made and discovered in this domain.

Many academics are developing methodologies and algorithms to process 3-D digital photos, and it has been determined from a review of various research papers that no algorithm or methodology can be acknowledged as a universal tool for processing all sorts of 3-D image data. This field still has a lot of room for advancement because exact details regarding hidden portions can only be retrieved when processing methodologies are efficient and sturdy.

1.4.2 Applications of 3D Recognition

Human is having capability of recognition of the object with their eyes in different lightning condition, occlusion, viewpoints and illumination etc. Human is having

ability to recognize the object even if objects are different in size, shape and appearance. Human eyes can categorize more than 10000 objects at a time [4].but same problem will be difficult for the machine to solve. Machine should be able to see the 3D world same like human eyes. But processing a 3D data with existing technique is not so easy. We require a simple representation of 3D shape information to avoid the computationally difficult process of shape recovery, storage, and comparison; we also need a computer with a decent recognition algorithm. Even with the current state of recognition technology, there are several real-world uses for machine vision. Many of these applications would benefit tremendously from an enhancement in the resilience and accuracy of present recognition methods. Below is a discussion of a few of the applications.

- **Medical Imaging**

In order to help doctors with diagnosis, therapy planning, and image-guided therapies, medical image processing software is becoming more and more important. Accurate, reliable, and rapid tracking of deformable anatomical objects like the heart is essential in medical picture processing. Figure 1.8 depicts a visual representation of medical imaging.



Fig. 1.7: Object Recognition in Medical Imaging

- **Tracking of Different Sports**

Young generation loves sports activity and wants to learn the different sports.

With the help of video clips (shown in **Fig. 1.8**) it's easy to identify the winning



Fig. 1.8: Object Recognition in Sports Tracking System

pattern of a particular sports person Digging, analyzing and representing multidimensional information is a new source of learning the techniques of a particular sports person. For example the monitoring of ball movement is critical for extracting any information from ball-based sports video sequences, and we can automatically capture video frames depending on ball movement.

- **Manufacturing Industry**

Now a day's most of the manufacturing industries are using Robots for pick the object and place them at particular place as shown in **Fig. 1.8**. It's reducing man powers drastically. But it's mandatory for robots to recognize the object properly and place them at allocated place without any error. In the process of Quality management, sorting, assembly line Object detection is a part of the process.

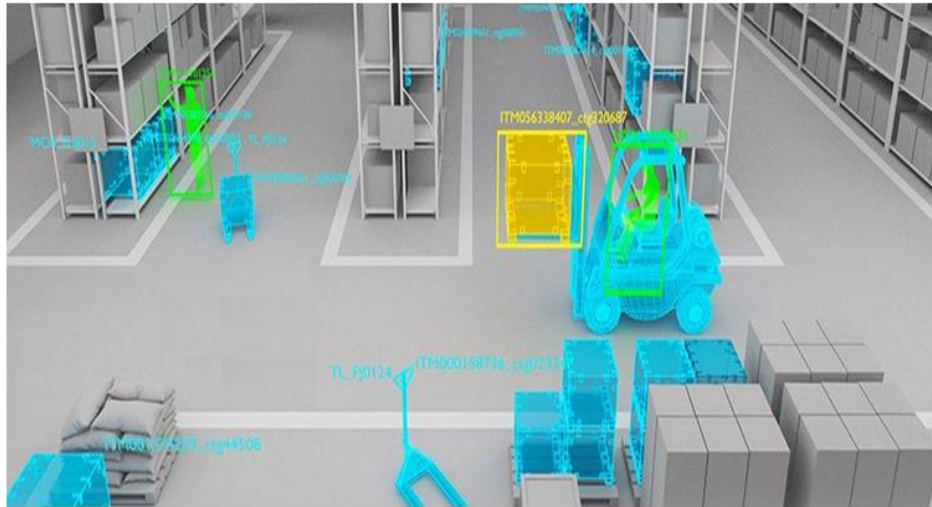


Fig. 1.9: Object Recognition in Manufacturing Industry

- **Robots in Automation**

Automation industries are using Robots for assembling of the parts of vehicle. If robots are able to recognize the objects then assembling will be less tedious. Here object recognition plays an important role and it shown in the **Fig. 1.10**.



Fig. 1.10: Object Recognition in Automation

- **Navigation System in Battlefield**

Robots can be helpful for the navigation purpose. Deploying the robots in the battle field or the area where human cannot survive and information can be retrieved with the help of robots. Robots can helpful in establishing navigation between human to the place where we cannot reach and same is shown in the **Fig. 1.11**.



Fig. 1.11: Object Recognition in Battlefield

1.4.3 Challenges of 3D Object Recognition

Designing a system which can be used for reorganization of different object is a cumbersome task. The fact that an object can be present in any shape, size, or position, rotated, mirrored, or transformed in any way that preserves distance ratios and co-linearity is a key difficulty in object recognition systems. Depending on the distance between the camera and the item, objects may have varying physical sizes or be geometrically portrayed differently. Different types of objects to be considered for the problem of object recognition. To evaluate an object recognition technique, each application imposes different requirements and constraints, such as [29]:

- **Reflection Property of the Surface**

Reflection is an important physical property of any object. The surface which is having high reflection will leads to specularity issue. This issue is difficult to solve with normal image based algorithm and structural based algorithm. Reflection property is shown in the **Fig. 1.12**.



Fig. 1.12: Example of Reflection

- **Lack of Pattern**

If image is having less texture or pattern on surface of the object then recognition process will be difficult to achieve. In this case identifying the shape of the object with the help of other methods will make whole process more challenging. As shown in the **Fig. 1.13** all the objects are having the same texture/ pattern hence it recognition is always a difficult task.



Fig. 1.13: Less Textured Image Courtesy [30]

- **Transparent and Dark Surface**

If the surface of the object is transparent then it's difficult to identify the shape of the object. Transparent surface can absorb the light totally or partially and makes object recognition always difficult. As shown in the **Fig. 1.14**, all the bottles are transparent and identifying the exact shape needs always an efficient recognition algorithm.



Fig. 1.14: Transparent Surface

- **Illumination**

Intensity of an object always depends on the angle, color and strength of the light. Recognition algorithm should be written in such a way that it can identify the object in any illumination changes. Mechanisms like edge extraction are sensitive to lighting conditions.

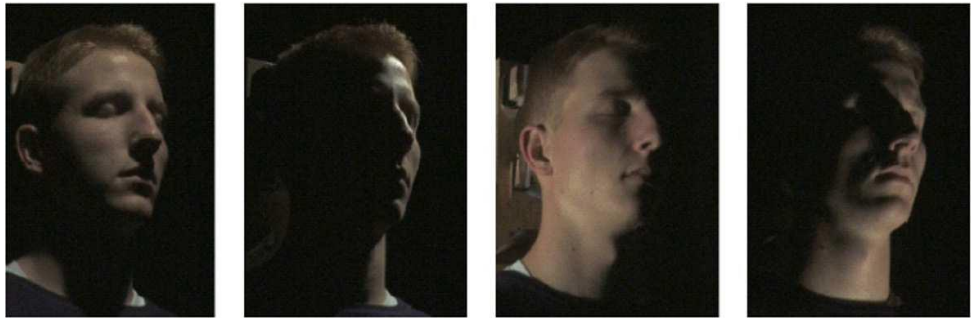


Fig. 1.15: Illumination Variation Courtesy [31]

With brighter images, slight color variations may become more obvious and cause the extraction of excessive edge information. When the lighting is poor, the difference in pixel brightness tends to decrease, occasionally making the edges disappear. Light intensity values vary across all images, as seen in Fig. 1.15, and each image emphasizes a distinct feature.

- **Scale**

The size of the object depends on distance between object and image acquisition system. As shown in the **Fig. 1.16**, all the shoes are of different size and recognition algorithm should be written in such a way that it can work with different size of objects.



Fig. 1.16: Scale Variation in Shoes

- **Rotation**

It is always possible that the same object can be present in the image at different angle as shown in the **Fig. 1.17**. Rotation of the object will not change the actual information about the image. Hence object recognition algorithm should be written in such a way that it works with different angles of the image.



Fig. 1.17: Rotation Variance

- **Background clutter**

A normal image will always contain the region of interest and the background image as shown in the **Fig.1.18**. For the same object we can have different backgrounds. Hence performance of the recognition algorithm should not depend on the background information.



Fig. 1.18: Same Object with Different Background

1.5 Summary

In today's age of technological growth, where machines play an active role in everyday life, giving them the ability to sense an object is a critical work. Object recognition is a fascinating problem with origins in artificial intelligence. Because we perceive the majority of the objects around us through our eyes, computer vision is one of the most promising techniques for solving this challenge. The human object identification mechanism is still partially understood, and researchers have explored a variety of techniques to replicate this incredible skill with machines. We have a long history of half-hearted and encouraging attempts to address this issue. Initial studies focused on classifying segmented objects based on structural descriptions. Approaches that explicitly retrieve the 3D information of the objects became possible as computing power developed. The use of structured lighting during imaging to promote shape recovery was one such strategy. In light of above discussion, we offer a unique and efficient method of 3D object recognition based on pattern recognition techniques that focuses on reducing computing complexity and memory space while remaining robust to variation in size and shape of the objects.

1.6 Thesis Organization

The thesis's outline is as follows:

Chapter 2 describes the existing works in the field of 3D objects recognition, wireframe model, and syntactic pattern recognition and brain tumor classification methods.

Chapter 3 describes the proposed methods for wireframe model, feature extraction and classification.

Chapter 4, describes the noise removal, image enhancement technique and a novel approach for extraction of the 3D surface /wire frame from the image using preprocessing techniques. Chapter 5, A novel algorithm is described for feature extraction using syntactic pattern recognition technique. All the experiments were performed on the standard benchmark datasets to prove the efficiency of the algorithm.

In Chapter 6, improved 3D Alex-Net architecture is implemented for brain tumor classification.

Chapter 7 concludes the thesis work and describes the future scope.

CHAPTER 2

LITERATURE SURVEY

In the realm of computer vision, 3D object recognition and pattern recognition are active and rapidly increasing research topics. Any pattern recognition system must define the pattern class, feature extraction, design classifiers, clustering, and selection of test datasets, as well as evaluate performance. The object is recognized by the pattern recognition system, thus the features must be extracted in such a way that they are suited for a specific recognition method. The two components of the object recognition methodology are the local feature extraction method and the global feature extraction method. Numerous researchers have produced excellent work in the area of local and global feature extraction. Real-world scenarios are more suited to local feature-based techniques. Global feature-based methods are preferable for extracting the 3D model and figuring out the shape of an object when the object's geometric structure is poor. Pattern recognition has been extensively studied during the past 50 years. However, no single technique can be used for all applications, including text detection, localization, speech recognition, remote sensing, bioinformatics, data mining, and multimedia applications. This chapter gives a systematic analysis of the popular local and global feature-based algorithms for 3D object detection. There are numerous studies that compare various techniques. The remainder of Section 2.2 compares and contrasts the various wireframe methodologies. In section 2.3, we evaluated the comparative study on 3D image reconstruction. An introduction of structural pattern recognition techniques and a comparison of brain tumor categorization algorithms are given in Sections 2.4 and 2.5, respectively. Section 2.6 provides a summary of the chapter.

2.1 3D Object Recognition Based on Local and Global Feature-Based Methods

The pixels in an image can be thought of as data, and the information that can be gleaned from them is called a feature or a feature vector. Feature extraction aims to reduce the number of features in the datasets by creating a new feature set. A new feature set should be able to represent most of the features which are present in the

original dataset. The main aim of feature extraction is to reduce the size of elements in the dataset [32]. According to the different approaches, two types of features can be extracted from the image, i.e. local and global features. The features can be extracted from the image as per the requirement of the application so they could be local, global, or both (local and global). Therefore there are two types of techniques for extracting features:

2.1.1 Local feature extraction based methods

2.1.2 Global feature extraction based methods

2.1.1 Local Feature Extraction

Local feature-based methods can be used for object detection and identification [17]. Generally, local feature-based approach is divided into three steps as shown in **Fig. 2.1**. The first step is 3D key point detection which describes the key points of the image. Key points are nothing but the region of interest where most of the features would be present [16]. In this step, the location information and size or distance from the neighborhood (scale) is also identified. The next step is to describe the local surface features. In this step, geometric information about the key points will be extracted and a feature vector will be created [16]. The last step is matching the surface. This step is comprised of matching features, creating hypotheses, and verifying the objects. Some examples of the local feature-based method are SIFT[17], SURF[33], LBP[34], BRISK, MSER, surface signature [22], 3D Tensor [23] and FREAK etc.

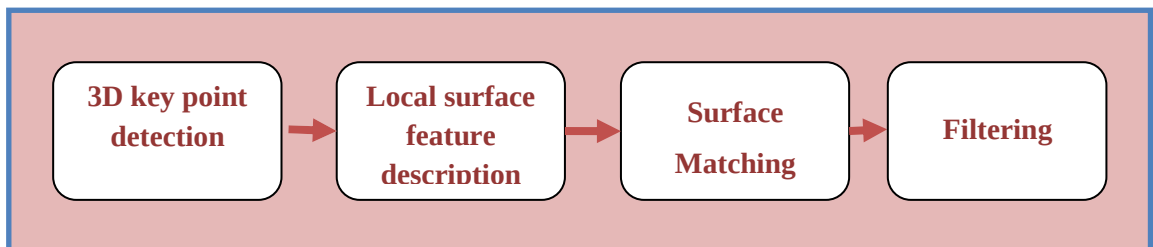


Fig. 2.1: Steps of Local Feature Extraction

In the last 20 years, various local feature extraction methods have been used, which has given new direction to 3D object recognition. A comparative study has been done,

and **Table 2.1** shows the comparative analysis of local feature extraction based methods.

2.1.2 Global Feature Extraction

Global feature-based methods consider the image as a whole. This method is used for image retrieval, object recognition, shape classification, contour representation, shape description, pose estimation and textual features. The techniques which can be used for global feature extraction are Shape Matrices, DCT, DRT Invariant Moments, HOG and Co-HOG are some examples of global descriptors [23-24]. Basic steps of global feature extraction are represented in the **Fig. 2.2**:

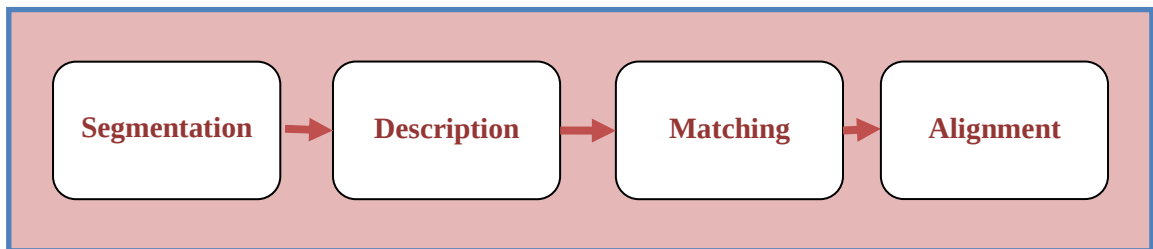


Fig. 2.2: Steps of Global Feature Extraction

Different methods have implemented in the last two decades, which has given the new direction to the 3D recognition field. A comparative study has been done on global feature extraction method, and a summary of the analysis is given in **Table 2.2**.

Table 2.1: Comparative Analysis of Local Features Extraction Based Methods.

Sr. No.	Authors	Technique name	Dataset	Performance analysis	PRE	KS	FS	Classifier/Matching	Future	Parameters/Challenges
1.	Y. Sun et al. [35]	3D surface representation for image registration	USGS DEM data	The authors have proposed a method to extract the surface point. Results show that the proposed method performance is suitable for real and synthetic data for pose estimation.	N	Y	Irregular local geodesic circles	Candidate point selection	Surface matching under occlusion & stability analysis	
2.	D. A. Kumar et al. [36]	S3DRGFs	BVC3DSL, HDM05, CMU, NTU RGBD	The proposed method can represent complex human signs with high precision for the development of a 3D sign language machine interpreter. This method improves the performance of classifier also	N	Joint area and joint perimeter features	joints, edges, angles, surfaces	GAK		Noise
3.	W. Maorui et al. [34]	3D CS-LBP	-	The proposed method can give a reduced sized dimension vector which is less sensitive to noise. Performance speed is comparatively better than traditional 3D LBP.	N	Y		N		Noise, recognition speed is better

Sr. No.	Authors	Technique name	Dataset	Performance analysis	PRE	KS	FS	Classifier/Matching	Future	Parameters/Challenges
4.	Ming Dong et al. [37]	Saliency detection & feature extraction	Synthetic & MRI data	The proposed method is working directly on 3D space because it's the ability to identify the 3D structure quickly.		Y		Descriptor Matching		Scaling, rotation, noise
5.	Z. Wang et al. [38]	SKLP	Confocal, Pollenmonitor, CHMontior	The proposed method can reduce the dimension, and results show that SKLP has the robustness to the rotation, illumination, transformation and scale variation.	N	Y		Classification		Illumination, Rotation, scaling & affine transformation
6.	Chen et al. [39]	LSP	OSU	The proposed method is applied to a single and two object scene. able to deal with a large dataset.	N	Y		Matching		Occlusion
7.	Mian et al. [8]	Depth values	UWA	The proposed method has 80% occlusion, and the recognition rate is 95%.	N	Y		matching		Scaling, occlusion, repeatability,
8.	Guo et al. [40]	TriSI	UWA	The average recognition rate of the proposed method is 98.4%.100% RR for 80% occlusion	Y	Y		Matching		Noise, mesh resolution, occlusion

Sr. No.	Authors	Technique name	Dataset	Performance analysis	PRE	KS	FS	Classifier/Matching	Future	Parameters/Challenges
9.	Bayramoglu et al. [17]	LSP,SIFT	1 http://www.vlfeat.org/vedaldi/ 2 http://sample.ece.ohiostate.edu/data/3DDB/RID/minolta/	Performance of LSP is high and SIFT performance is better for 2.5D images.	N	Y	N	---		Scaling, rotation, occlusion
10.	Knopp et al. [33]	3D SURF	KUL, Princeton, Tosca+Sumner, HREC'09	The proposed 3DSIFT performance is better.	N	N		classification		Texture, RR
11	Tang et al. [41]	HONV for detection & classification	RGB-D, own dataset	THE proposed HONV performance is better.	N	N		Classification	retrieval	Precision, recall
12.	Tombari et al. [42]	USC	http://graphics.stanford.edu/data/3Dscanrep	The proposed 3D-USC performance is better. And 3D descriptor occupies less memory	N	N		Matching		Precision, recall
13.	Li and Guskov et al. [43]	NBS	Stuttgart	The recognition rate is 98.02% for the Stuttgart dataset.		Y	Y	classifier		Occlusion. transformation

Sr. No.	Authors	Technique name	Dataset	Performance analysis	PRE	KS	FS	Classifier/Matching	Future	Parameters/Challenges
14.	Zhong et al. [44]	ISS	PSB	The proposed method has a recognition rate of 97% on the PSB dataset.	N	Y		Matching		noise, obscuration, cluttering, pose estimation
15.	Drost et al. [45]	Point Pair Feature	UWA	The proposed method has given 97.0% accuracy and 84% occlusion. Recognition speed is good.	N	Y		Matching	Suitable for pick and place applications.	Noise, clutter, partial occlusion
16.	Taati et al. [46]	VD-LSD	Queen's LIDAR, Queen's Stereo, UWA	Recognition accuracy is 98.5% on the UWA dataset and 80% on Queen's LIDAR dataset and; 74.7% RR on the Queen's Stereo database.	Y	Y	Y	Matching		Precision, occlusion, clutter
17.	Bariya et al. [47]	EM, registration	Queen's LIDAR, UWA	The proposed method is having Recognition accuracy: 97.5% on the UWA dataset and 82.43% on Queen's LIDAR dataset.	N	Y	N	Matching		Scaling, clutter

Sr. No.	Authors	Technique name	Dataset	Performance analysis	PRE	KS	FS	Classifier/Matching	Future	Parameters/Challenges
18.	Aldoma et al. [48]	SHOT	UWA, Kinect	Recognition accuracy is 100% on the UWA dataset and 79.5% on Kinect datasets.	Y	Y	N	classification		Recall, precision, Occlusion
19.	Guo et al. [49]	RoPS	Bologna 1, UWA, Queen's LIDAR, Ca' F. Venezia	The accuracy of RoPS is 100%, 98.9%, 95.4% and 96.0% on Bologna 1, UWA, Queen's LIDAR, Ca' Foscari Venezia dataset.	N	Y	N	classification		Noise and mesh resolution

Table 2.2: Comparative Analysis of Global Features Extraction Based Methods.

Sr. No	Authors	Technique Name	Dataset	Performance Analysis	Seg/Pre	FD	FS	Classifier/Matching	Future	Parameters/Challenges
1.	Zhongyu Lou et al. [50]	3D layout from a single image	SUN	Performance of the proposed method for 3D classification is 11.7% higher than the state of the art method.	Y	Y	N	Classification		Accuracy, precision, recall, and F-score.
2.	M. Naeimzaghiani et al. [51]	Hybrid GLCM and EDMS	Binary character dataset with 3520 image	The hybrid method with feature selection has given high performance compared to GLCM & EDMS alone.	N	N	Y	Multi-NN, Decision tree, Bayes net		Accuracy

Sr. No.	Authors	Technique name	Dataset	Performance analysis	PRE	KS	FS	Classifier/ Matching	Future	Parameters / Challenges
3.	Z. Fan et al. [52]	Extended CVFH	Washington University RGBD Dataset, Real dataset	The method is helpful for object recognition and grasping of the objects. The recognition rate is high for similar objects also.	S	N	N	SVM	Pose estimation	Recall, Precision, Accuracy, F-1 Score
4.	C. Fookes et al. [53]	The 3D global registration algorithm	MRI images	The proposed algorithm gave high performance compared to other pairwise registration techniques.	N		Y	Matching		Accuracy
5.	X. Pan et al. [54]	Sparse style Features	Liu & Lun dataset	CPU occupancy is 12.5% and the time cost is 8 hrs because 7threads can be executed at a time.	N	N	Y	CNN		Accuracy
6.	Donghui Wang et al. [55]	GGFM for similarity matching	-	Comparatively fast and less storage required because every feature of the model would be computed once in the case of GGFM.	N	N	N	Matching		Translation, rotation, scaling
7.	I. I. Ganapathi et al. [56]	global 3D descriptor	UND-J2, In house database	For proposed method recognition rate for global feature extraction is 88.53%, 91.76%, 94.84% for rank-1, 2 & 3.	Pre	N	N	Matching		Verification accuracy, EER

Sr. No.	Authors	Technique name	Dataset	Performance analysis	PRE	KS	FS	Classifier/ Matching	Future	Parameters / Challenges
8.	D. Sarwinda et al. [57]	HOG-TOP	ADNI	3D HOG-TOP had 95% accuracy and 96% sensitivity which is comparative good from other approaches.	N	N	Y(P PCA)	Random forest classifier	Focus on extracting the features from 3D PET and 3D SPECT brain images	Deformation, sensitivity, specificity
9.	K. Onishi et al. [58]	HOG	under various conditions,	Eextract the features without depending on orientation or clothes. Background Feature dimension can be reduced.	N	N	Y (PCA)	N	Detectio n of the human body without subtracti on from the backgrou nd.	Noise and used five features for implementati on: standing, open arm, hands-up, walking, running.
10.	B. Wang et al. [59]	2D-SIFT & 3D- HOG	Biwi Kinect Head Pose Database.	Proposed method have achieved low error rate & accurate head position estimation.	N	N	Y	RANDOM FOREST		Accuracy, noise, scaling

Sr. No.	Authors	Technique name	Dataset	Performance analysis	PR E	KS	FS	Classifier/ Matching	Future	Parameters / Challenges
11.	P. S. Kiran et al. [60]	3D Relational geometric features	KLU3DSL, NTU RGB-D, HDM05 and CMU	The proposed method has better accuracy in terms of five different parameters such as distance, angle surface, area, perimeters are 69.1%, 72.4%, 78.7%, 86.5%, 86.1%.	N	N	N	Global alignment kernel for matching	It would be helpful for mobile applications.	
12.	Rusu et al. [61]	GFPPFH	IKEA objects	The proposed algorithm can separate the scene and object into geometric form with an accuracy rate of 98.27% .	Y	N	N	CRF for annotation of the surface into geometric class & SVM for object labeling.	Boundary feature can be added to improve the classification accuracy.	Accuracy
13.	B. Drost et al. [45]	PPF(L+G)	Mian	Authors have focused on three evaluation parameters and algorithm has given better performance even in the presence of noise, partial occlusions and clutter.						Noise, clutter

Sr. No.	Authors	Technique name	Dataset	Performance analysis	PR E	KS	FS	Classifier/ Matching	Future	Parameters / Challenges
14.	Muja et al. [62]	ReIn for recognition & pose estimation	-	The authors have designed a model for object recognition and estimation of the pose. Recognition is better and robust, even when surface noise is present.	Y	Y	N	BIGGPY for 2D and VFH for 3D	Can extend the work for transparent, textured and un-textured images.	Recall & precision
15.	Madry et al. [63]	GSH for object representation & pose estimation	PSB, SOC	THE proposed GSH has performance is better than VFH in terms of 3D object classification.	N	N	N	Clustering	algorithm should work for less training data.	Pose & scaling noise.
16.	Wohlking et al. [64]	SDVS	-	The performance of the SDVS is fast and robust.						
17.	Wohlking et al. [65]	ESF	3D Net	Authors have combined angle, area and point distance functions and results show that recognition rate is better than other methods such as SDVS, VFH, CVFH, and GSHOT.	N	N	N	NN classifier	Occlusion, speed	Angle, point distance, shape features

Sr. No.	Authors	Technique name	Dataset	Performance analysis	PR E	KS	FS	Classifier/ Matching	Future	Parameters / Challenges
18.	Chen et al. [66]	GFH	public Sydney Urban Objects Dataset	The performance of 3D urban object recognition is better than global SI and LGPFH on open Sydney Urban Objects Dataset and their dataset.	N	N	N	SVM with RBF kernel	Compare the identification with other classifiers.	Accuracy, F-measure
19.	Lima et al. [67]	GASD for recognition & pose estimation	OCID	The proposed method is working nicely with pose estimation and recognition rate and performance are better than existing and less processing time.	N	N	N	matching	Pose refinement speed could be improved using GPU.	Pose estimation, mean and standard deviation, error rate.
20.	Aldoma et al. [68]	OURCVFH L+G	23 scenes captured through Kinect sensors	The performance of OURCVFH has a better compared to existing CVFH and SHOT.RR is 82% and recognition speed has been increased up to 91%	N	N	N	NN		Pose estimation, distortion
21.	Aldoma et al. [69]	CVFH	Kinect	This algorithm can recognize the object correctly and detection of a pose is also useful when training data has different properties.	S	N	N	NN	Deal with the vast dataset	Partial occlusion, noise

Sr. No.	Authors	Technique name	Dataset	Performance analysis	PR E	KS	FS	Classifier/ Matching	Future	Parameters / Challenges
22.	A. Ghasemzadeh et al.[70]	3D-DWT	PolyU-HSFD	Accuracy is high compared to spatio-spectra classification.	N	N	N	Classification (KNN,CRC)		Accuracy
23.	Kasaei et al. [71]	GOOD for recognition & pose estimation	-	Computation time and memory occupancy are less compared to other methods.	S	N	Y			Scale, pose variation, noise, and computation time and memory usage.

2.2 Literature Survey on Wireframe Model

Superficial and volumetric features are always important for the processing of any 3D image. The surface, points and lines of an image are called superficial features. Edges, Contour or a skeletal representation of an image are called volumetric features. In a 3D image, Edges are also two types. The edges which intersect the two planes or surfaces are called volumetric edges and the edges which separate the two regions are called superficial edges. The extended version of 3D edges from the 2D edges is introduced by H. Lui [72]. The author had given the approach which generalized the detection of 3D edges using a gradient-based method. Further, Zucker et. al [73] have considered edges as a line and constructed wireframe model. Hueckel et al. [74] extended the work of Zucker et. al [73] and considered edge as a plane that passes through the center of the particular voxel. This approach has given an optimal view of representing the image. Rosenfeld et al.[75] proposed a new method for finding the 3D edge using the magnitude of the surface. O. Monga et al. [76] proposed the canny detector algorithm for 3D images.. The algorithm for extracting the wireframe contour of a 3D object was introduced by Mukherjee et al. [77]. The author has considered a 3D input image as a binary array. The first thinning of the image had been done and then the segmentation method was applied. Further, each segment is polygonized using the Graham Polygonized and boundary traversal method. If segments are adjacent then they would be connected through vertex merging methods. Finally, the obtained results are processed to remove unwanted values. Xiaoyuan Ren et al. [78] have proposed a 3D viewpoint & shape estimation-based method and conducted the wireframe modeling using the maximum probability distribution method. CNN model is used to detect the viewpoint of the 3D objects. Simant Prakoonwit and Ralph Benjamin[79]proposed a method for the reconstruction of 3D surface points and wireframe using 2D images. All the images were taken from different viewpoints and the wireframe is created with the help of combining the contour of 2D images.

Ali et al. [76] resented a method for the recognition of Bengali digits. M. Rashid et al. [77] proposed a method in which eight directional codes are identified and these direction codes are used to convert the image into a feature vector, but the technique is only applicable for Bengali digits. U. Pal et al. [78] have also proposed a method

for the recognition of Bengali numerals. This technique was able to recognize the numerals without thinning and normalization operation. Sarkar et al. [79] proposed the POS (Part of speech) tagging and & parsing technique Method. This method is used to find the answers from the given text.

As per the literature survey, Most of the 3D edge detection algorithms are an extension of the 2D edge detection algorithm. In most of the cases researches have used CAD model for identifying the wireframe model of the image. There is no specific algorithm which is designed for finding the wireframe model of 3D images.

According to a survey of the literature, most 3D edge detection algorithms are extensions of 2D edge detection algorithms. Most of the time, researchers have used CAD models to figure out the wireframe model of the image. There is no particular algorithm for finding the wireframe model of a 3D image.

2.3 Reviews on Reconstruction of Images

In the past ten years, numerous methods have been proposed for the 3D reconstruction using a single 2D image. Jiang et al. [81] proposed an algorithm that reconstructed 3D face images using smooth coarse to adequate optimization. Xiaoguang et al. [82] proposed the 2DSAL method (Two-dimensional self-supervised learning method), which can construct 3D faces using 2D images. 3D face reconstruction gained more attention in the last three decades since Hu et al. [83] reconstructed the 3D image using shape and shading technique. It works on the concept of changing principle. Whenever there is a change in the surface orientation, there is a possibility of a shading change. This method is helpful to recover the character based on the intensity of the pixels in the image. Image formation must be known before applying the shape from the shading method. Queau et al. [84] also implemented the shape from the shading technique to reconstruct 3D images. However, these techniques have some drawbacks. These techniques can be applied on the Lambertian surface because SFS can deal with the face as Lambertian surface (LS). Finding the depth of the image is another difficulty. If the image has a specular surface, it is impossible to find the depth of the image due to the surface's reflectance behavior. The specular surface will have a mirror-like character, hence finding the depth is a complicated process and the

result would be more worst if dealing with synthetic data. Therefore, it is required to find a method that can work with all type of surfaces.

After the SFS, researchers have introduced a statistical learning-based method for the reconstruction of 3D images. These methods used the learning-based approach. In this direction, Zhang et al. [85] proposed a method on temporal-based correspondence. This method has excellent results for high-resolution images. However, this method requires a complex setup, controlled illumination environment and multiple input images for the reconstruction of 3D images. These methods require rigorous human involvement for finding the features of the face from the image, which makes them computationally expensive. On the other side, some methods [86-88] do not require human interference for tracking facial features. However, these methods required multiple views of the single image and controlled lighting environments. Kemelmacher Shlizerman and Seitz et al. [89] proposed 3D face reconstruction using multiple images. This method can produce high-quality 3D face models using photometric stereo. The neural network also follows the learning-based approach and these methods have better performance for regression problems. 3D face reconstruction-based neural network proposed by Blanz and Vetter [90]. This method is a popular parametric face model because of its simplicity and this method is a foundation of other face reconstructed methods. The neural network-based approach requires a massive amount of images for learning purposes and it will be computationally expensive.

In this section, we performed a brief literature survey on different 3D face reconstruction method and these methods can be categorized as optimization-based methods, supervised learning-based methods and shape and shading base face reconstruction. A complete comparative study is shown in **Table 2.3**.

Table 2.3: Study of Existing Methodology on Reconstruction of Images.

Sr. No.	Authors & Year	Methodology	Database	Remarks
1	Yajing Chen et al. [91], 2020	3D Morphable Model Regression	MICC Florence, FRGC2	Error = 2.41 ± 0.62 Error = 5.05 ± 1.44
2	Yangyu Fan et al. [92], 2020	Generative Adversarial Network	WildUV dataset	PSNR = 29.5 SSIM = 0.952
3	Yu-Hsuan Huang et al. [93], 2020	GLADNet	SoF Dataset, CASIA-WebFace	Accuracy Improved by 2.5%
4	H. M. Rehan Afzal et al. [94], 2020	Scale-Invariant Feature Transform (SIFT), Multivariate Gaussian Distribution, Basel face model (BFM)	LFW Database	RMSE = 0.8866 PSNR = 27.27 Error = 1.36 Mean Error = 0.4996 Std. Error = 0.3070 SSIM = 0.9864
5	Qing-Ming Liu et al. [95], 2019	Convolution Neural Network	Helen, CelebA face, CMU+MIT, WHU-SCF datasets	PSNR = 37.496 SSIM = 0.918
6	H. M. Rehan Afzal et al. [96], 2018	Supervised Descent Method (SDM), Multivariate Gaussian Distribution, PCA	LFW Dataset	Perform Better
7	Yudong Guo et al. [97], 2018	Convolution Neural Network (Single-image CoarseNet + Tracking CoarseNet and FineNet)	Self-Dataset	MSE = 0.56 Mean vertex distance = 4.56

Sr. No.	Authors & Year	Methodology	Database	Remarks
8	Luo Jiang et al. [98], 2018	Bilinear Face Model	Face Warehouse, Bosphorus database	3DRMSE = 1.75 ± 0.29 Run-Time CPU = 3min
9	Chen Cao et al. [99], 2014	Active Shape Model, PCA	Face Model Databases	Better Performance
10	Huy Tho Ho et al. [100], 2013	Markov Random Fields, Belief Propagation, Lucas-Kanade (LK) algorithm	FERET, CMU-PIE, Multi-PIE	Recognition Rate = 95.5% on FERET, 98.8% on CMU-PIE, 89.4% on Multi-PIE
11	Mingli Song et al. [101], 2012	Coupled Radial basis Function Network	BU3D database	MSE = 5.60%
12	Jian Lai et al. [102], 2012	Weighted Global Sparse Representation	Extended Yale B and AR database	Recognition Rate = 98.9% & 97.5%

Few authors have worked on reconstructing medical images, and they have taken different approaches, which are discussed below:

Yinsheng Li and Guang-Hong Chen [103] introduced DIM which is used to identify the inconsistency of the acquired projected data taken from different viewpoints. Further, the input data is divided into a subset according to the DIM value at each viewpoint angle and then the SMART-RECON algorithm is applied to the datasets for the reconstruction of the images. Practical utilization of the proposed method is tested on vivo human subject data and performance is better than existing methods. Aurelien Bustin et al. [104] proposed a method that combined the statistical and geometric 3D features. The main aim of the proposed method was to make a single isotropic volume by combining the multiple 3D volumetric data which is acquired from different orientations and recovery of sharp edges of the image and thin anatomical structure. These two aims are achieved using the Augmented Lagrangian scheme. The

performance of the proposed method is evaluated on numerical simulations and the MRI datasets. Benjamin Hou et al. [105] proposed a method in which intensity information of 2D image slices is used for the registration of the image and further CNN is used for transformation of the image and for the learning of the regression function which maps the 2D slices into three-dimensional atlas spaces. Performance of the algorithm is evaluated on MRI dataset, fetal imagery with synthetic motion, and real MRI data and achieved average spatial prediction error of 7mm. This method is a computationally better solution for 2D and 3D registration and it gives good results in real-time scenarios. Daniel p. Benalcazar et al. [106] built a 3D model by adding the depth information in 2D iris images. First, the smart phone camera captured the various 2D images from different angles and different lighting conditions, and then structure from motion algorithm is applied for reconstruction of the point cloud three-dimensional model. Finally, the best fitting three dimensional mesh models is obtained with the help of the screened Poisson surface reconstruction method. The performance of the proposed method is evaluated on different ten subjects and the model contained approx 11000 three-dimensional points. The performance of the proposed 3D iris scanning method is better than the existing methods.

Xuehang Zheng, Sai prasad Ravishankar [107] proposed a new PWLS based reconstruction method and the quality of the reconstructed 3D images is better than the existing PWLS-EP based method. This proposed method improved the quality of the image and image resolution and it's more efficient in terms of computational speed. Chen Zhang [108] proposed a method for volumetric surface reconstruction and assumed that every scene consists of a cuboids structure. The proposed method having reconstruction fidelity even if the object consists of a heavy curved and concave structure and the method can preserve the smoothness and cleanliness of the surface. This method is suitable for 3D printing of the images. Yuanhao Guo et al. [109] proposed a two-phase three-dimensional reconstruction method for light microscopy axis view images. In the first phase, an improved 3D volumetric representation is defined and in the 2nd phase, 3D reconstruction is achieved by searching the optimal surface over the confidence mapping. The performance of the proposed method is evaluated on three different datasets and the approach can represent 3D shape in a precise manner. This method can be used to determine the

shape of the objects. Lu Ding et al. [110] proposed a method for optoacoustic image systems that helps to find the shape and size of the image. These methods have significant improvement in the spatial resolution, a contrast to noise ratio and quality of the reconstructed image is reported better than existing techniques. Bin Li et al. [111] proposed a 3D-ReConstnet network that used a residual network for the extraction of the features from the 2D image. According to this method, if objects are occluded in the image then the Gaussian probability distribution method is used for the learning of the feature maps. The performance of the proposed method is evaluated on ShapeNet and Pix3D dataset and the results are better than other existing methods. Volumetric representation is an important approach to three-dimensional reconstruction. The main goal of volumetric representation is estimating a convex hull in three-dimensional spaces from each view [112-115]. In the same way, the space carving algorithm is used to recover the shape of the 3D objects by removing the voxels which are not visible in the particular view [116-117]. But these methods require the proper segmentation and it is not always possible in some scenarios. There are many well designed three-dimensional reconstruction algorithms are available. Some algorithms are only designed for the reconstruction of peculiar and transparent objects [118]. These methods outperform on a macroscopic scale because to collect the surface-related information they setup the various lightning while capturing the images [119-120]. In recent research [121] authors have presented a semantic reconstruction which is a combination of a data term and regularization constraints and results are outperformed on public datasets [122]. Now a day's researchers are focusing on deep learning-based methods for improving the matching quality [123-124] and for the reconstruction of 3D images but these methods require a large amount of training and testing data which are not available in our case.

2.4 Reviews on Structural/ Syntactic Pattern Recognition Technique

The word "parsing" comes from a Latin word that means "part of speech" or "specific object analysis." It is the method of analyzing the string of symbols in computer language, data structure, or natural language. The term parsing is having a different meaning in the field of computer science & linguistics [125]. In the field of computational linguistics, parsing is a method or technique in which a computer tries

to understand the meaning of a particular word or string of words and as a result, it will provide a parse tree.

In terms of computational linguistic parsing, it is a method /technique in which computer will try to understand the meaning of a particular word or string of words and as a result, it will provide a parse tree. A parse tree represents the syntactic relationship between each other. In terms of computer science, this term is used for analyzing the language of computers and it is referred as syntactic analysis. According to the syntactic technique, a pattern is a combination of simple sub-patterns [126-127]. The fundamental sub-patterns are called primitives and the relationship between the primitives is referred to as complex patterns [128-129].

Syntactic pattern recognition is a correlation between the structure of patterns and the theory of formal language. Patterns are the sentences or sequence of words which belongs to a particular language and the sentence can be generated according to the grammar rules. Therefore, Complex patterns could be represented through a small number of primitives and grammar rules. Every pattern class is associated with the grammar and it must be inferred from the training dataset. If the proper structure of the pattern is available then syntactic pattern recognition can be used instead of statistical pattern recognition [130]. The parsing technique supports supervised and unsupervised machine learning techniques for predicting the labels. Syntactic pattern recognition technology can be used in the field of medical science, data mining [131], speech recognition, language translation, data transmission, etc.

Quang et al. [80] have presented research on the A* algorithm for parsing techniques and proposed an HGM model (Hierarchical graph method) based on the same algorithm. This method can replace the existing VNM (Virtual node method), which was used to overcome the problem of Chomsky standard forms. The performance of HGM was better than VNM. For the last three decades, researchers have extensively used the machine learning approach in various areas, including medical diagnostics. According to our literature survey, the creation of a wireframe model from the volumetric-based method is not done so far and in most of the papers, authors have used different viewpoints of 3D images and constructed the wireframe model of 3D

images. In **Table 2.4**, several other parsing approaches utilized by various researchers are discussed.

As we progress through the literature review. We've gone through many object detection and pattern recognition techniques so far. We're currently narrowing down our literature search and intend to concentrate on brain tumor classification and syntactic pattern recognition. Further, this chapter explores various brain tumor classification approaches.

Table 2.4: Comparative Analysis of Different Syntactic Pattern Recognition Methods

Sr. No.	Authors & Year	Parsing Technique	Database	Results	Future work/ Suggestions
1.	S. Jaf et al. [132], 2019	MLDP	University dependency dataset	The authors have proposed an MLDP. The proposed parser is giving good results even for different languages. a parser is having better head attachment accuracy	Split MLP_{label} , MLP_{arc} can be helpful for other parsing architecture.
2.	Yue-Jie Zhang et al. [133], 2005	DOP based parsing	-	The performance of the proposed Chinese parsing method is satisfactory if the sentence structure is not very complex.	It should be tested for complex sentences also and need of the real and better storage methods.
3.	J. Sarker et al. [134], 2019	Semantic parsing	bAbI-10k English question answering dataset [135]	In proposed methodology authors have used parser which is based on lambda calculus. The accuracy of the proposed method is 95% and the overall mean accuracy is 83%.	In future Author can work with long and complex sentences. Machine learning techniques can be used for input and output.
4.	L.Q. Thang et al. [136], 2012	HGM	-	The authors have proposed the HGM method. This method is a replacement for VNM. The performance result of HGM is comparatively better than classical VNM.	The proposed method can be used to integrate the parsing system into the text to speech conversion.
5.	S. Mall et al. [137], 2014	CKY algorithm	-	PA is used for translation. Algorithm is used for analyzing the structure of the sentence. CKY has given better results i. e 91.4%.	For gender classification authors have used some linguistic rules.

Sr. No.	Authors & Year	Parsing Technique	Database	Results	Future work/ Suggestions
6.	Ogiela et al. [131], 2000			In this paper, the author has identified the scope of syntactic pattern recognition. Authors have proved that this method is helpful in the field of medical science for early diagnosis of a dangerous disease.	
7.	G. Badr et al. [138], 2006	CBS-LLP	-	Authors have proposed a new version of BS which is used in AI. It is searching for the graph efficiently in terms of no. of operations required.. A result shows the improvements up to 75% in terms of no. of required operations.	In future authors can work on optimization of computation and they can try to reduce the time and space complexity.
8.	K. Huang et al. [139], 2019	Syntactic pattern recognition		A new method is proposed to find the location of wavelets. To calculate the distance Levenshtein method is used. Proposed NPFS is used to identify the no. of clusters.	Each class property can reveal the changes in the continuous layer and
9.	M. N. Husen et al. [140], 2016	Syntactic approach	300 car driving samples	The proposed method is used to detect the intention of the driver using a syntactic approach. This method has a 90% recognition rate with less than 0.5 seconds.	Detection and recognition rate can be improved in the future.
10.	Y. Yu et al. [141], 2003	Syntactic approach for graphic symbols	15 vector-based datasets in the GREC 2003	The proposed method describes the geometric relation and provides the location information of primitives w.r.t. object. This model has given 100% accuracy for vector-based graphic symbols with different scale and rotation transformations.	In future authors can work with other patterns that can be represented in vector forms.

2.5 Survey on Brain Tumor Classification Methods

Deep neural networks have recently benefited a variety of industries, including natural language processing and picture analysis. Compared to older methods, DNNs can automatically learn discriminative features because hierarchical information has been incorporated into the learned characteristics. One of the most prominent 3D medical image analysis approaches is CNN, which has demonstrated excellent performance among the other methods mentioned.

Zhang et al.[142] suggested a method for segmenting brain tissues using multi-modality MRI images that utilized deep CNN. A 2D CNN-based method was reported by Li et al. [143] for automatically segmenting lesions from CT slices and the CNN model was compared to various classic machine learning techniques such as AdaBoost [144], random forests (RF) [145] and SVM [146]. An increasing amount of focus has been placed on attention-based image classification and semantic segmentation architectures [147-149]. The attention-related mechanism has been used to overcome the problems in real-world applications for several medical imaging jobs. Mzoughi et al. [150] have proposed employing volumetric CNNs to distinguish between high and low-grade tumors in T1 contrast-enhanced images. Pei et al. [151] performed a similar study on grading-based glioma classification, in which they first segmented the tumor before classifying it as either HGG or LGG. Ge et al. [152] employed a framework that combined T1 contrast-enhanced, T2 contrast-enhanced, and FLAIR images to simultaneously classify the tumor. The enhanced performance of these applications gives CNN an advantage. **Table 2.5** contains a discussion of existing brain tumor detection and classification methods.

Table 2.5: Study of existing methodology on Brain Tumor Detection and Classification

Sr. No.	Authors & Year	Methodology	Database	Remarks
1	L. Kuang et al. [153], 2019	Multi-CNN	MICCAI BraTS[55]	Accuracy = 84.3% Sensitivity = 82.5% Specificity = 97.7%
2	Ahmed H. et al. [154], 2020	Genetic Algorithm	Self Dataset	FOM = 89.87% Accuracy = 97.71% Sensitivity = 94.71% Specificity = 98.5 %
3	P. Kumar et al. [155], 2019	Deep Wavelet Autoencoder (DWA)	Brain Image Dataset[56]	Accuracy = 93.14% Sensitivity = 94.26% Specificity = 92.16% F-Score = 93.15%
4	Tarek E et al. [156], 2019	Adaboost Classifier + Back Propagation	Self Dataset	Accuracy = 93.33% Sensitivity = 71.42% Specificity = 88.88%
5	V. Anitha et al. [157], 2015	K-Means + DWT	Self Dataset	Accuracy = 92.43% Sensitivity = 95.53% Specificity = 50.6 %
6	S. Anand et al. [158], 2020	Swarm Intelligence and K-means	Figshare Dataset[33]	Accuracy = 98.7% Sensitivity = 93.4% Specificity = 65.1% F1-Score = 98.7% JAC = 66.46 %
7	Swati et al.[159], 2019	VGG-19	Figshare Dataset[33]	Accuracy=94.82%
8	Hassan et al.[160],2020	VGG-16	Brain MRI Dataset[34]	Accuracy=96% Precision=93% Recall=100% F1-score=97% Time=6846sec
9	Hassan et al.[160],2020	ResNet-50	Brain MRI Dataset[34]	Accuracy=89% Precision=87% Recall=93% F1-score=90% Time=9091sec

Sr. No.	Authors & Year	Methodology	Database	Remarks
10	Momina Masood et al.[161],2021	Mask-RCNN	Brain MRI Dataset[34]	Accuracy=98.34%
11	Milica M.Badza et al.[162],2020	CNN	Figshare Dataset[33]	Accuracy=95.40% F1-Score=94.94%
12	Vimal Kurup et al.[163],2019	Capsulenet	Figshare Dataset[33]	Accuracy=92.60% F1-Score=93.33%

The classification of the 3D objects in biology, cytology or medical field is always a difficult task. The objects can be discriminated against with their symbolic characteristics and a human being can train his brain to recognize various objects. But computers cannot simply differentiate between these objects since it lacks human intelligence. Computers require some sort of representation or method for object recognition. There are many approaches for 3D object recognition, but CNN is a popular one because it incorporated with surface characteristics of the 3D image.

2.6 Summary

3D object recognition and pattern recognition is a progressive and rapidly growing research area. Based on that, a complete study of 3D object recognition and pattern recognition methods has been presented in this chapter. We tried to summarize, the different object recognition and pattern recognition techniques perform in a different scenario. Most of the papers which are included in the survey are funded/ supported by government authority. It shows the authenticity of the work. An ample amount of work has been done in the field of feature extraction, classification, and clustering. Many authors have focused on specific challenges of 3D object recognition such as illumination, noise, and scaling. Still, there are some open challenging areas where researchers can focus. There is a need for improvement in the algorithms which can boost the recognition rate and it would be helpful in the field of robotics, CAD/ CAM system, medical imaging, data mining, remote sensing, etc. Another challenging area is the storage of features or feature vector as feature vectors of the 3D image takes lot of space in memory So, in the future, we will try to introduce a new method of feature

representation which takes less space in memory compared to the existing technique. Other areas of improvement could be computational speed where researchers can focus. After studying various papers on 3D reconstruction, we conclude that volumetric based approaches are giving more promising results and properly addressing the challenges of 3D reconstruction. Taking this into consideration, in this research we are considering syntactic pattern recognition technique for feature extraction and object recognition.

Most of the researchers used statistical pattern recognition techniques on medical images. As per our literature survey there is no evidence of applying the structural pattern recognition-based method in the medical field. Researchers have recently begun utilizing algorithms based on deep learning for the segmentation and identification of brain tumors. Therefore, syntactic patterns have been implemented for the classification of brain tumors. This is an effective method for early detection and classification of brain cancers.

As per the literature survey, we identified some of the research gaps. In the next chapter will discuss about the research gaps and objective of the research work.

CHAPTER 3

RESEARCH OBJECTIVES AND METHODOLOGY

3.1 Research Gaps

There have been many attempts in the past to recognize objects in 3D images, but no perfect solution to the problem has yet been proposed. The issue with 3D object recognition could be:

- A precise preprocessing method can play an important role in accuracy of object recognition. So there is a need of improvement in the existing preprocessing techniques.
- Most of the parsing algorithms are only applied for text analysis. Therefore, parsing algorithms should be explored for other applications also.
- Most of the researchers have focused on statistical pattern recognition technique for classification but very less work has been done in the field of structural pattern recognition.
- Till now, Syntactic pattern recognition techniques are applied to 2D images only. Therefore there is scope to understand the complex 3D images using syntactic pattern recognition.
- Representation of a 3D image in a computer will take lot of space. Reducing the data dimensionality can be another challenging task.
- Most of the object recognition algorithms are focused on illumination and cluttering. Still very less work has been done in the rotation and scaling of the image.
- Representation of a 3D image in a computer will take lot of space. Reducing the data dimensionality can be another challenging task which can also increase the transmission speed of data.

3.2 Research Objective

The research's major goal is to develop a broad methodology based on a novel theory of pattern recognition and a mechanism for item identification and recognition. The goal is to create a generic method to solving a broader range of issues. The fact that

generalization is one of the most difficult and valuable skills a computer may have is the major reason for our interest in this topic. Because generalization is perhaps more important in computer vision than in any other field and it is a difficult task. Therefore, this research has four key objectives:

- Remove the noise from the image and extraction of the 3D surface /wire frame from the image using preprocessing techniques.
- To propose an effective approach to represent the 3D surface/wire frame of the image as a feature vector.
- To propose an effective 3D object recognition approach using feature vector.
- To analyze the performance of proposed work on benchmark/standard datasets.

3.3. Research Methodology

The proposed method first takes the image as an input; further, the following operations mentioned in the **Fig. 3.1** are used for the implementation purpose.

3.3.1Steps of Proposed Methodology

Step 1: Provide an image as an input.

In this step, the 3D standard datasets like ModelNet 40, PIX 3D Dataset, PASCAL 3D + dataset, Figshare dataset, Kaggle brain MRI dataset, Medical MRI dataset and BraTS 2019 dataset are acquired from the different sources.

Step 2: Applying pre-processing techniques to remove the noise from the image and enhance the quality of the image

As there are many pre-processing and are available, there is a need to select the correct and appropriate method for pre-processing. Pre-processing helps to remove the noise from the image and it enhances the quality of the image. In the proposed method, adaptive median filter is used for removing the noise from the image. To enhance the quality of the image, contrast image enhancement technique is applied.

Step 3: Finding the wireframe model of the image

In this step wireframe model of the image is identified. In this research, first proposed cellular logic array processing technique is applied to find the edges of the given input

image. Proposed algorithm will try to locate those points where boundaries of the region over which gray level is not changing. This process is done for all the planes of the 3D image. The edges and corners which are discovered from the three dimensional structure using CLAP based method are now integrated. The tracked edge features are integrated to make a wireframe model of static objects.

Step 4: Feature Extrcation

There are different types of shape texture features are available in the image. The extraction of features from the image is an important step. In this step, we will design an algorithm that takes contour/ wireframe as an input and provides a feature vector or knowledge vector as output. In the proposed method we used picture description language for the representation of the features. The image is traced pixel by pixel which starts from pixel position (1,1,1) to find the initial foreground pixel (x_i, y_i, z_i) with the intensity value 1. This will identify the initial point of the first component present in the image. The proposed algorithm is written in such a way that it will find the next neighbor of the current pixel in all the preferred direction. Preference will be generally given to the right side of the pixel (from the current pixel). The naming conventions of the preferred direction are: Right(R), Down Right (DR), Down (D), Down Left (DL), Left (L), Up left (UL), Up (U), and Up Right (UR), Back(B), Back Right(BR), Back Down Right (BDR), Back Down (BD), Back Down Left (BDL), Back Left (BL), Back-Up left (BUL), Back-Up (BU), and Back-Up Right (BUR), Front(F), Front Right(FR), Front Down Right (FDR), Front Down (FD), Front Down Left (FDL), Front Left (FL), Front Up left (FUL), Front Up (FU), and Front Up Right (FUR). In this method, the previously recognized direction has given priority compared to other preferred directions. The tracing of each pixel will continue until the algorithm will not find any connected neighbor pixel to the current position (x,y) and will remove the pixel which is already traced. If there is no neighbor pixel then the algorithm will display the knowledge vector of that component and again algorithm will continue until it will not trace all the components which are present in the image. If all the components are traced then the final knowledge vector will be displayed.

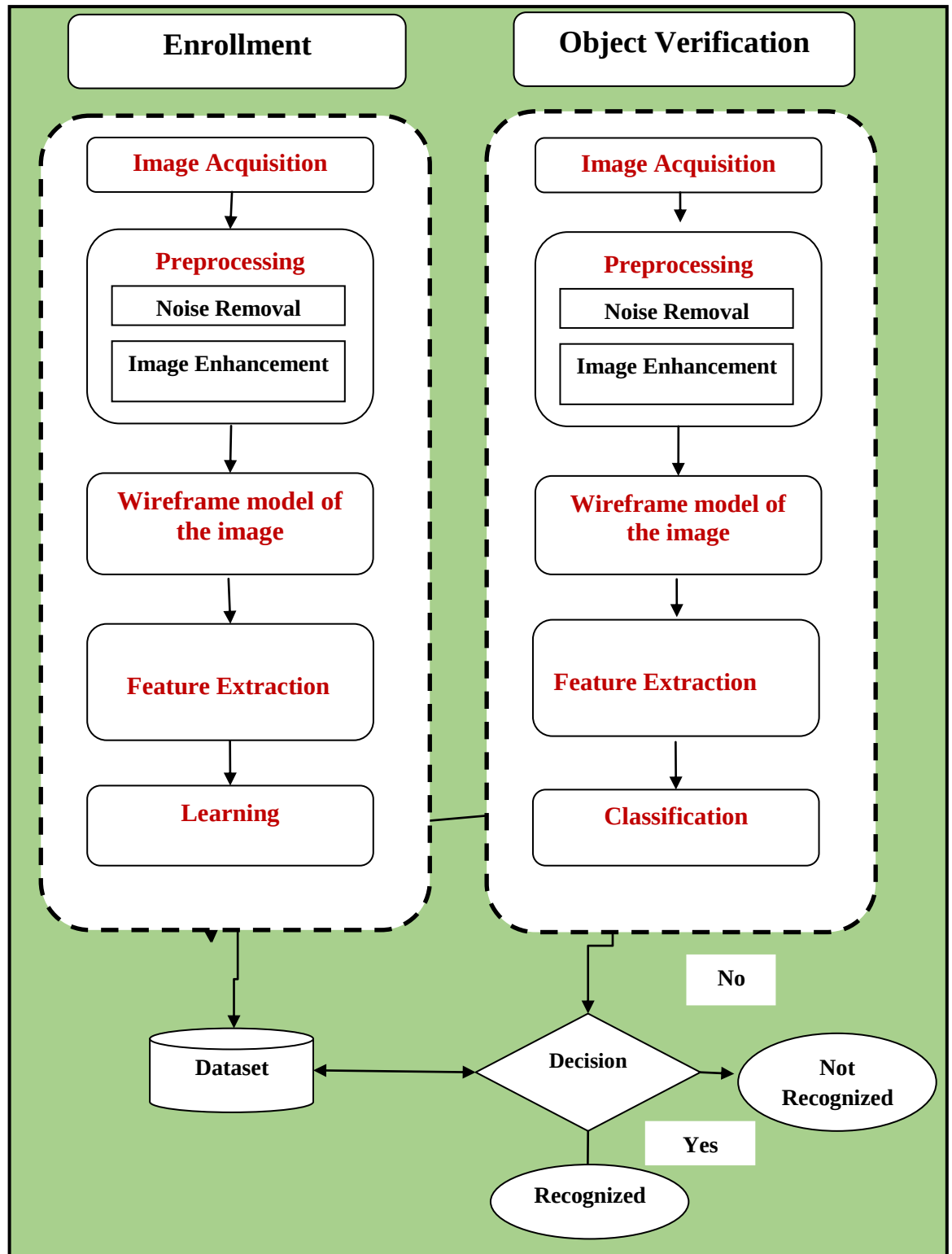


Fig.3. 1: Flow Chart of Proposed Methodology

This knowledge vector is nothing but a feature vector that consists of the direction and length code of each component and this knowledge vector will give the information

that some objects are present in the image. This feature vector would be the input for the next stage.

Step 5: Classification

The classification module accepts the input data and categorizes them as members of one of the specified classes based on the set of attributes they possess. The selection of a classifier and estimation of classification parameters are both part of the design process.

Step 6: Performance evaluation of the proposed technique.

In the final step, the proposed method is evaluated on standard dataset. The novel methodology is then tested against current methods on a standard dataset.

The following chapter will explain the preprocessing approach and proposed methodology for the creation of wireframe model of 3D images.

3D WIREFRAME MODEL OF IMAGES

4.1 Introduction

The surface of the 3D object contains a lot of information about the object and it's always an important and prominent research area of computer vision with a variety of applications. Several researchers have made efforts in this field [164-178]. A 3D image that is displayed on the computer screen contains different light intensities and it's difficult to visualize the hidden parts of the image to get some meaningful information. Therefore, the processing of the 3D image is all about the extraction of the physical, structural and morphological information from the hidden or invisible part of the image. The precise and useful information can be obtained only when the technique which we are applying on 3D image is more reliable in terms of accuracy. Superficial and volumetric features are always important for the processing of any 3D image. The surface, points, and lines of an image are called superficial features. Edges, Contour, a skeletal representation of an image are called volumetric features. In a 3D image, Edges are also two types. The edges which intersect the two planes or surfaces are called volumetric edges and the edges which separate the two regions are called superficial edges. Extraction of volumetric and superficial features from the image always remains a difficult task. These volumetric features are useful to construct the CAD model of different physical objects which is also called a wireframe model of 3D images. The volumetric intersection approach was first introduced by Martin et al. [165] and further investigated and modified by other researchers [166-171].

In this area, Rajan et al. [172] introduced a mathematical framework with the name Cellular Logic Array Processing (CLAP) method. This method can process 2D and 3D images as well. CLAP is a logical and mathematical framework which is developed with the help of the fundamental concept of "Markov's Normal Algorithms" and "Von Neumann's Cellular Automata". Markov's Normal Algorithms Works as Turing machine and cellular automata is a parallel processor and it is

represented in **Fig. 4.1**. Cellular Logic Array Processing is able to detect the different features of 3D images such as 3D points, 3D lines, 3D skeleton and 3D surface.

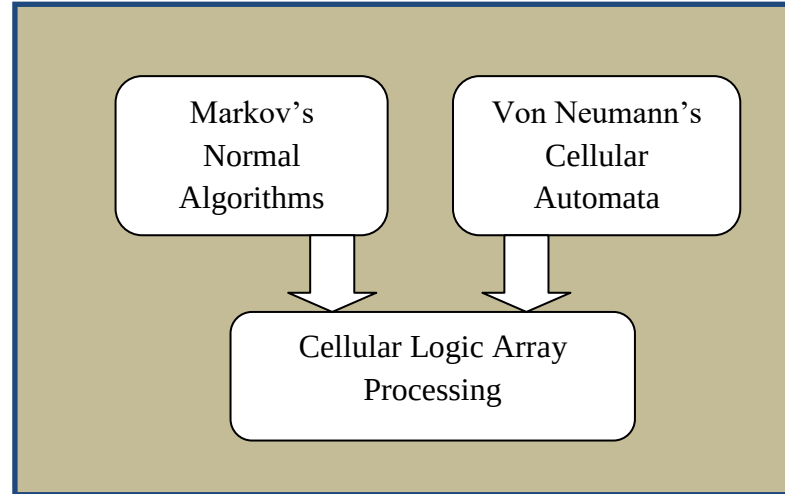


Fig. 4.1: Cellular Logical Array Processing

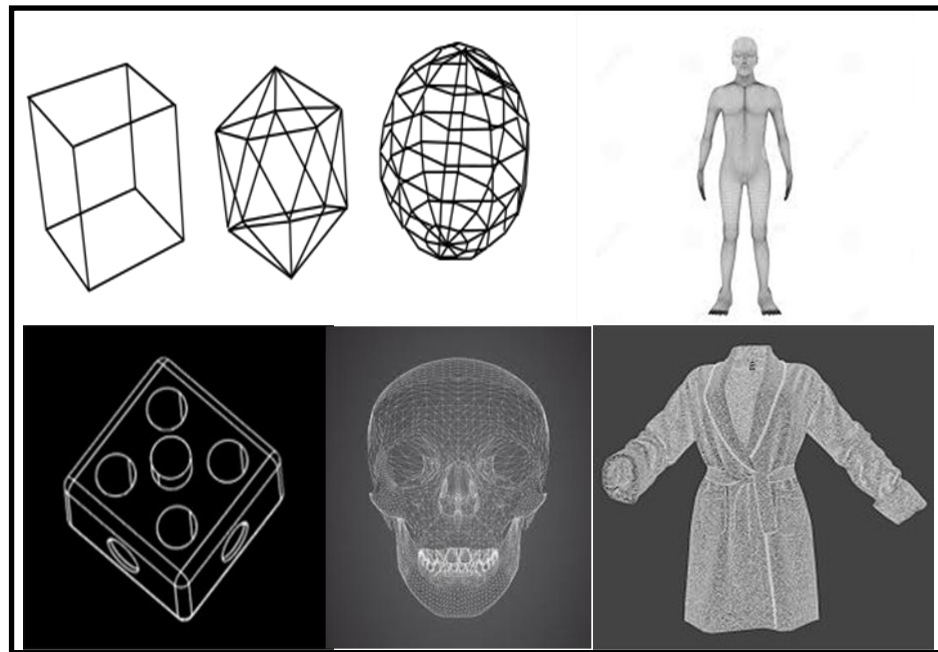


Fig. 4.2: Sample Images of Wireframe Model [165]

According to our literature survey, the creation of a wireframe model from the volumetric-based method is not done so far and in most of the papers, authors have used different viewpoints of 3D images and constructed the wireframe model of 3D images. The objective of this research is to find a feasible solution to extract the volumetric features of a 3D image in the form of a wireframe model. Wireframe model is a skeleton or visual representation of the three dimensional objects. The

sample images of the wireframe model are shown in **Fig. 4.2**. This chapter describes the proposed methodology for the construction of a wireframe model.

Image preprocessing is essential step for computer vision related applications. Taking this into consideration first image preprocessing has been performed to remove the noise from the image. Further contrast image enhancement technique is used to improve the quality of image. Lastly, a CLAP-based technique is employed to construct the wireframe model of a 3D image. The following is a detailed description of the proposed methods.

4.2 Proposed Methodology

The proposed method is designed for 3D images and an image could be a simple image that consists of several objects or it could be medical images that are more complex and contains a lot of hidden information. To understand and analyze a 3D image in a proper way, a 3D image can be represented using different view and plane that is represented in **Fig. 4.3**.

The proposed methodology consists of three parts:

- De-noising of the Image using Adaptive Median Filter
- Image Enhancement
- Construction of Wireframe Model

A further detailed description is explained below and workflow of the proposed method is represented in **Fig. 4.4**. A detailed description is given below

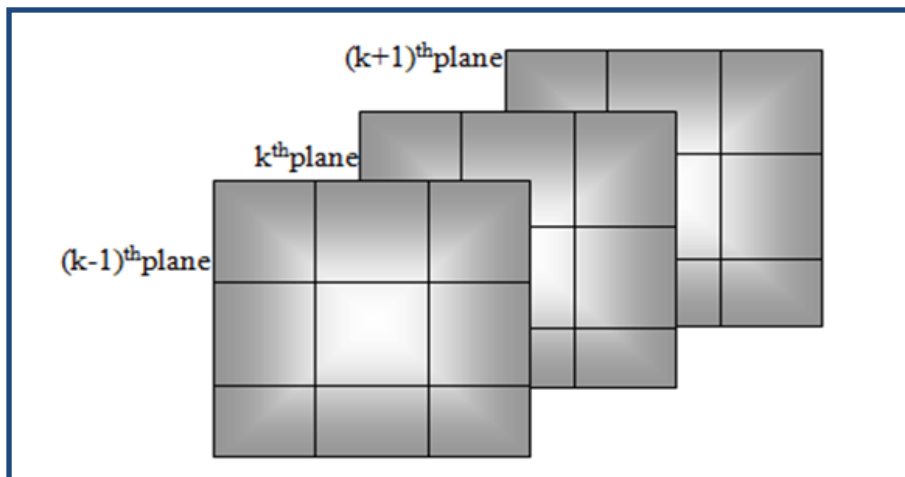


Fig.4. 3: Various Plane Representations of 3D Images

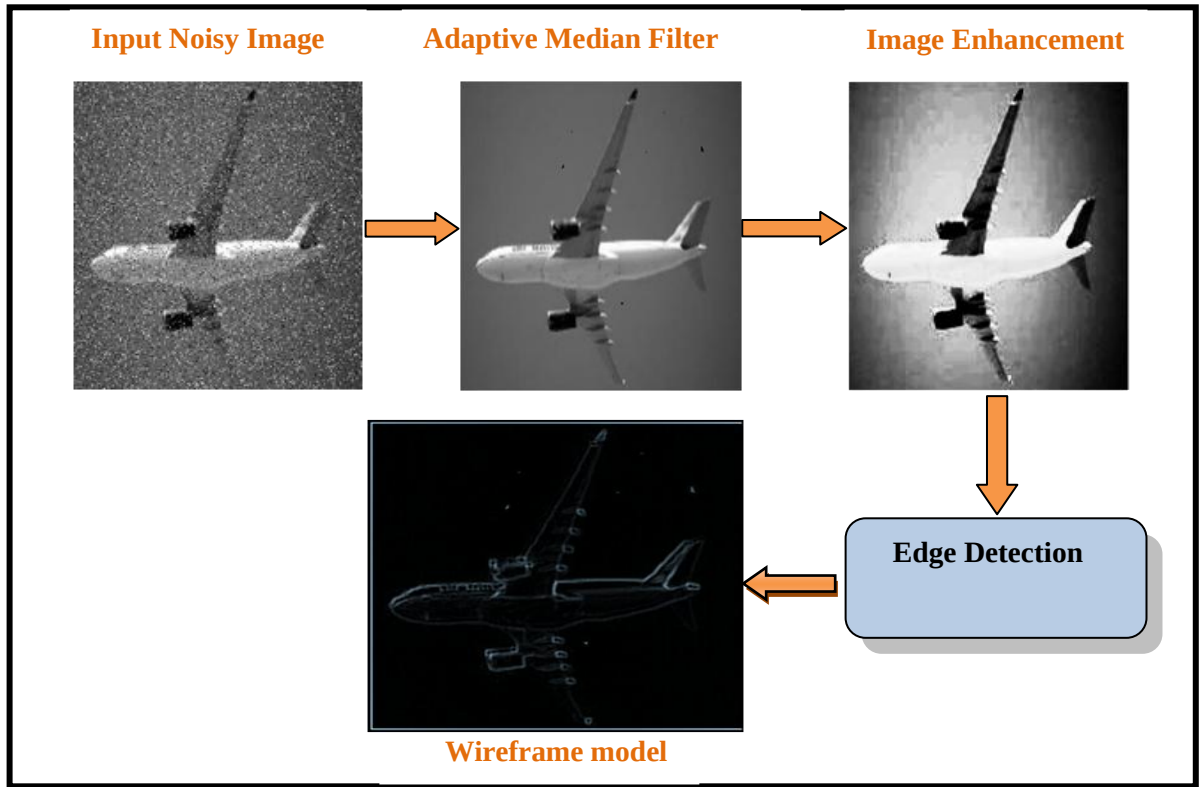


Fig. 4.4: Basic steps of proposed work

4.2.1 De-noising of the Image using Adaptive Median Filter

An adaptive median filter is used to remove the noise from the image. It compares each pixel of the image to the neighborhood pixel. If any one of the pixel values is drastically different from the neighborhood pixel, then that pixel will be considered as noise and the adaptive median filter will replace that noisy pixel

Algorithm 4.2: Adaptive Median Filter

Step 1: Start

Step 2: Find

$$P_{a1} = X_{\text{median}} - X_{\text{minimum}} \text{ \& } P_{a2} = X_{\text{median}} - X_{\text{maximum}}$$

Step 3: if ($P_{a1} > 0$ and $P_{a2} < 0$), then go to step 5; else, go to step 4.

Step 4: Increase the size of the window.

Step 4a. if window_size < I_{max} repeat Step 2 and 3 else the result is I_{xy} .

Step 5: Find

$$P_{b1} = I_{xy} - X_{\text{minimum}} \text{ \& } P_{b2} = I_{xy} - X_{\text{maximum}}$$

Step 6: If ($P_{b1} > 0$ and $P_{b2} < 0$) then result from I_{xy} else result is X_{median}

Step 7: Stop

value with the neighborhood pixel's median value. The algorithm of the adaptive median filter is as follows. As median filter can remove salt and pepper noise but if speckle noise is present in the image then median filter is not the good choice. Whereas adaptive median filter is able to remove the noise from the image and it can also deal with speckle noise. Adaptive median filter can also preserve the edges of the images and it sharpens the edges of the object. The comparative results of all the three filters are shown in the **Fig. 4.5**. PSNR value is also calculated on some random images of the medical dataset and shown in the **Table 4.1**. PSNR is used to calculate the peak error. It is a ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. If the PSNR value is higher it means the quality of the processed image is good. The formula for calculating the PSNR value is mentioned below:

$$PSNR = 10 \cdot \log_{10} \frac{MAX_I^2}{MSE} \quad (4.1)$$

Where MAX_i is the maximum pixel value of the image and as per the analysis the PSNR value is high when adaptive median filter is applied on the images. Adaptive median filter are applied on the random images of different dataset and it is observed that adaptive median filter is working well compared to the median and adaptive mean filter.

4.2.2 Image Enhancement

Image enhancement is the essential and second stage of the proposed work. After removing the noise from the image, contrast stretching is used to enhance the quality of the image [189]. This process will improve the pixel's value and sharpening the edges of the image.

In this process, firstly color image is converted into a grey-level image and the value of the pixels lies between 0 & 255. All the grey level values are remapped or structured through stretching function span over the whole image as shown in Equation

$$newimg[k] = \frac{(img(k) - v_{min}) \times 255}{(v_{max} - v_{min})} \quad (4.2)$$

After enhancing the quality of the image, in the next step edges are detected from the image which is explained in the next section.

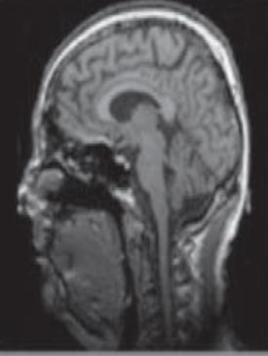
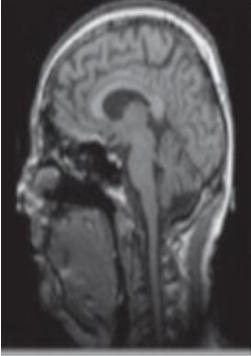
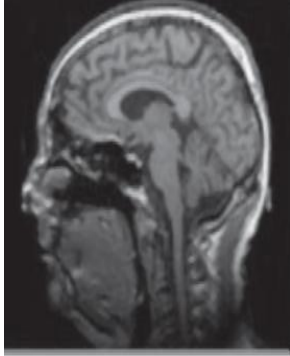
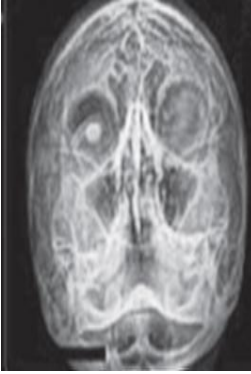
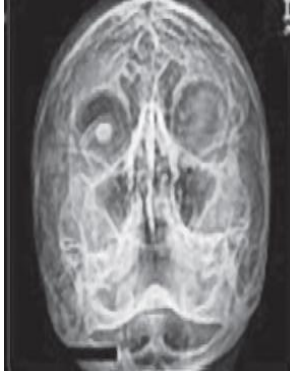
Original Image	Median Filter	Adaptive Mean filter	Adaptive Median filter
			
			
			

Fig. 4.5: Tabulation of Medical Images using Median Filter, Adaptive Mean Filter and Adaptive Median Filter

Table 4. 1: PSNR (in dB) Comparison between Median Filter, Adaptive Mean Filter, Adaptive Median Filter

Image	Median filter	Mean filter	Adaptive Median Filter
	PSNR (in dB)		
Brain Image	25.09	26.24	27.24
Skull Image	24.45	25.04	26.45
Lung Image	23.18	24.30	26.54

4.2.3 Construction of Wireframe

Many edge detection algorithms used the concept of the neighborhood principle for the detection of the edges. These algorithms work on the abrupt change in the gray level value of the surrounding pixels. Keeping the same concept in mind, our algorithm will try to locate those points where boundaries of the region over which gray level are not changing. This process is done for all three planes of the 3D image which is represented in **Fig.4.3** and it is done in the following manner:

For the scanning of the input image, the size of the sliding window is 3x3x3. In each movement of the sliding window 3x3x3 size of the input image is covered. In each movement, the algorithm will check that some or all pixels are forming convex polyhedrons. It means at least there should be at least one pixel in each row and column. The possibility of such type of convex region is 16 and it is shown in **Fig. 4.6**.

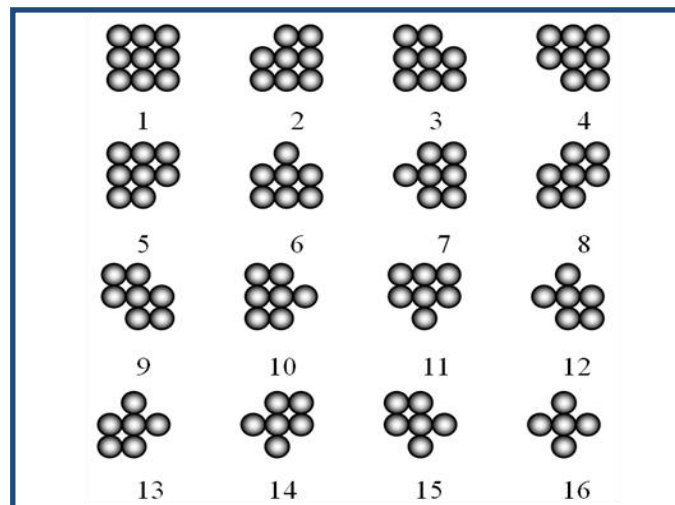


Fig. 4.6: Sixteen Convex Patterns in a 3x3 Windows

As shown in the figure pattern number sixteen is available in the rest of the other patterns so any one of the patterns has occurred or not, we need to check the sixteenth pattern alone. If sliding window pixels consist of pattern 16^{th} then in the next step, zero will be assigned to the center cell so that boundaries will be intact and it will erase all the interior parts of the sub-image. This procedure is continued until it covers all the neighborhood pixels of the front plane and then the same procedure will be repeated for the central plane and rear/backward plane. The effect of this method is that it gives the boundary of the given 3D gray level image. A detailed description of the procedure is mentioned in the **Algorithm 4.3**.

4.3 Analysis of Results

Proposed methodology is evaluated on different type of datasets to check the performance of the algorithm. In this research, we used seven datasets in which four datasets belongs to Brain MRI images and other four datasets consists of different objects.

4.3.1 Datasets

The first dataset is ‘Figshare brain tumor dataset’ which is publically available [173]. This dataset consists of a total of 3064 images of 233 different patients. The second dataset is again ‘Brain MRI dataset’ which is publically available on Kaggle[174].

This dataset consists of a total of 253 MRI images. The third dataset is the medical MRI dataset and it is collected from the pentagram research institute, Hyderabad. This dataset consists of five different folders such as: Cerebrix dataset, Toutarix dataset, Skull dataset and GPR dataset. The dataset contains 2D slices of MRI, which can be combined to make a 3D view of the images. The forth dataset is BraTS 2019 dataset [175]. This dataset consists the images of four different contrasts. The dataset contains four folders. The folder structure is composed of FLAIR (Fluid Attenuated Inversion Recovery), T1, T1-contrast enhanced and T2.

The performance of the proposed method is evaluated on three different standard datasets: ModelNet40 dataset [176], PIX 3D dataset [177], PASCAL 3D+ dataset [178] and Example Based Synthesis dataset [179]. ModelNet40 is the core dataset, which contains 12,311 CAD models divided into 40 different classes.

Algorithm 4. 3: Construction of wireframe model

Input: 3D image Dataset.

Output: Wireframe model.

1: Load Dataset

2: Sliding window (SW) having $3 \times 3 \times 3$ size is used to read 3D image's pixels.

3: While SW not reaches to all pixels of the plane.

{

3a: Read front plane's pixel.

{

If all pixel values read by SW contain at least one pixel in every column and row then replace central pixel with zero. Otherwise, move SW to read the next pixels.

}

3b: Read central plane's pixels.

{

If all pixel values read by SW contain at least one pixel in every column and row then replace central pixel with zero. Otherwise, move SW to read the next pixels.

}

3c: Read rear plane's pixel.

{

If all pixel values read by SW contain at least one pixel in every column and row then replace central pixel with zero. Otherwise, move SW to read the next pixels.

}

}

4: Concatenate all tracked edges that are obtained from step 3 and build a wireframe model

5: Display the result.

6: End.

There are 9,843 models in the training set and 2,468 models in the test set in this dataset. To build multi-orientation or view of each 3D model, we rotated the dataset equally in horizontal direction and finally produced the dataset on 4th and 7th depths and multi-orientation (3, 6, 9, 12, 18, 24, and 30 rotations) occurs only on depth 7th.

Each 3D model's multi-orientation is applied to both the training and testing datasets individually. PIX 3D dataset contains 395 3D shapes of 9 different categories of objects. For every shape, an actual image is present in the dataset. Pascal3D+ includes 12 rigid object categories drawn from the PASCAL VOC 2012 dataset. These objects have pose information marked on them (azimuth, elevation, and distance to camera). PASCAL 3D+ also includes pose-annotated photos from the ImageNet dataset for these 12 categories. The entire dataset for our research was created in the same manner as ModelNet40, as previously explained. Example-Based Synthesis of 3D Object Arrangements is Synthesis dataset which consists of different scenes of 3D objects, models, and texture related image.

4.3.2 Evaluation Parameters

The proposed method is evaluated on two different parameters. The first evaluation parameter is Accuracy. Accuracy means that how accurately our algorithm can find the wireframe model of the given image. The other parameter is the processing time. Here, the processing time is nothing but the time which is required for converting the input image into the wireframe model. To calculate the Processing time following formula is used:

$$\text{Processing Time} = 1/\text{Performance of the Processor} \quad (4.3)$$

The other evaluation parameter is the number of passes i.e. no. of passes required for detecting/ constructing the wireframe model of the 3D image.

4.3.3 Results on Different Datasets

Wireframe model of 3D objects are constructed on ModelNet40, PIX 3D and PASCAL 3D + dataset and results are shown in **Fig. 4.7**. In the same way wireframe model is also constructed for Figshare and Kaggle Brain MRI dataset and results are shown in the **Fig. 4.8**. As the algorithm is working on the 3D images therefore it is possible to see the different views of 3D images and the results are shown in **Fig. 4.9**.

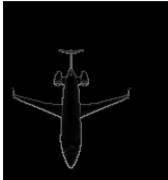
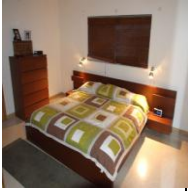
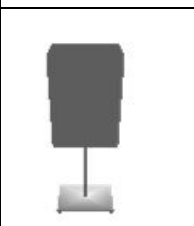
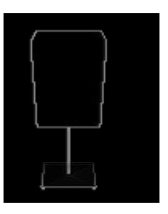
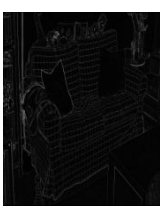
Model Net 40 Dataset		PIX 3D Dataset		PASCAL 3D + Dataset	
Original Image	Wireframe Model	Original Image	Wireframe Model	Original Image	Wireframe Model
					
					
					
					
					
					

Fig. 4.7: Wireframe Model of the Input Images on ModelNet 40, PIX3D and PASCAL 3D+ Dataset

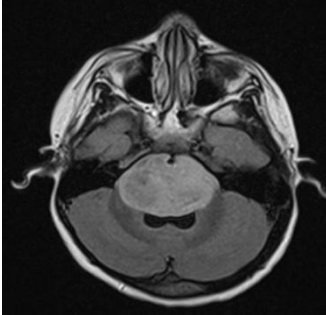
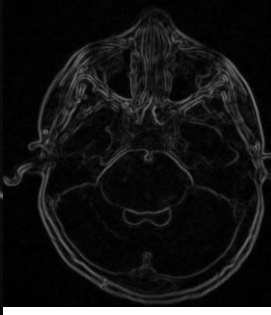
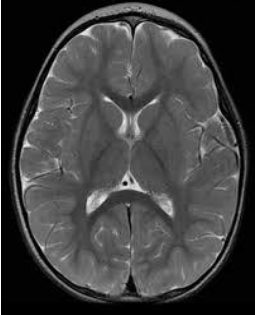
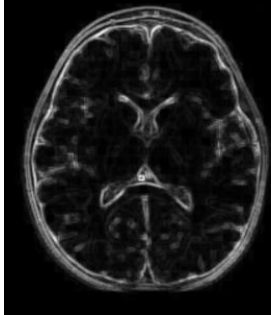
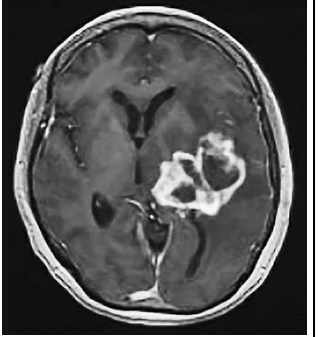
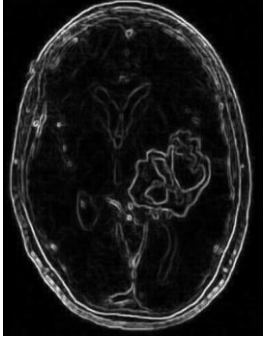
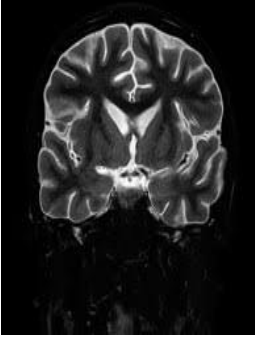
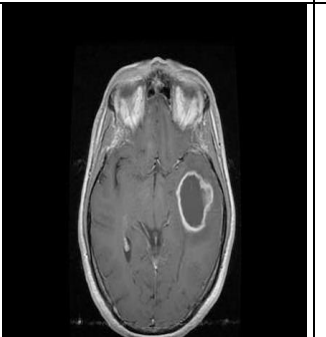
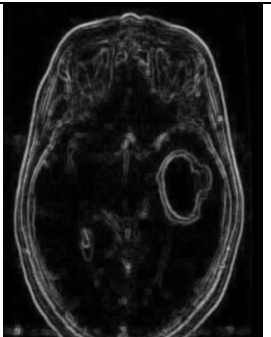
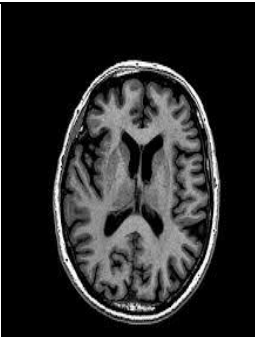
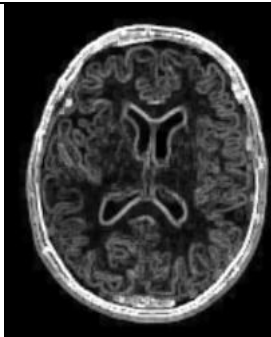
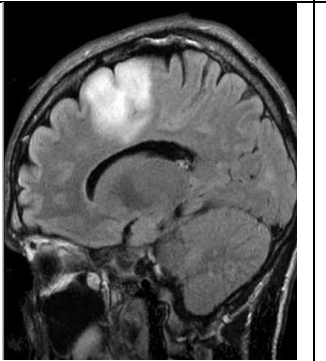
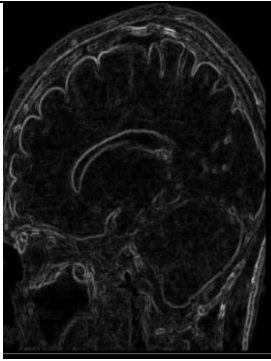
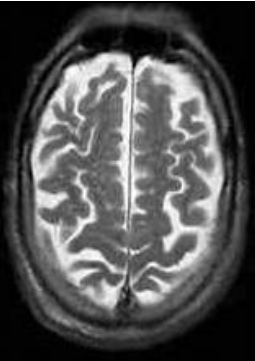
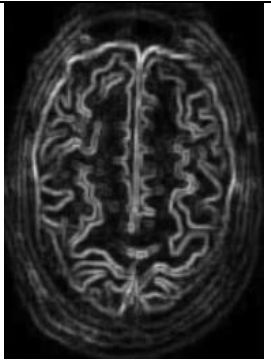
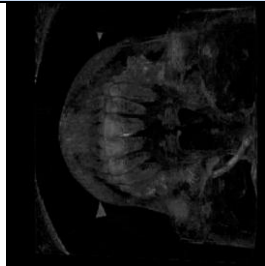
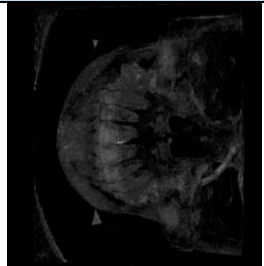
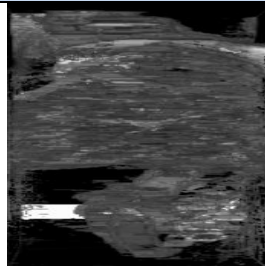
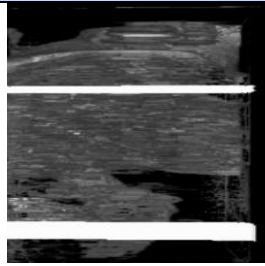
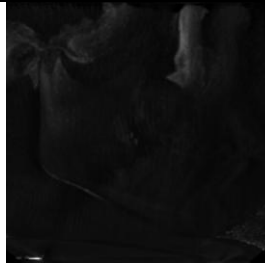
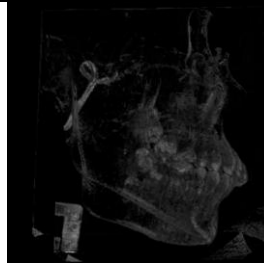
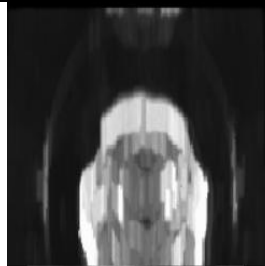
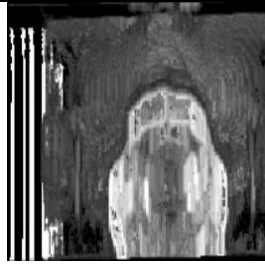
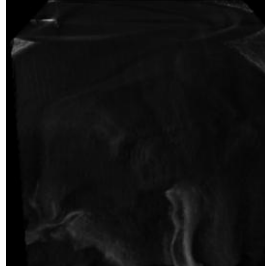
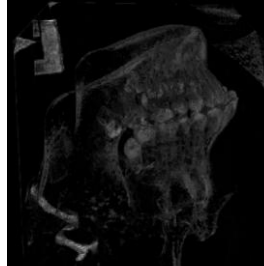
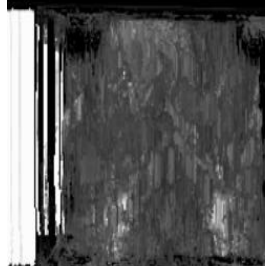
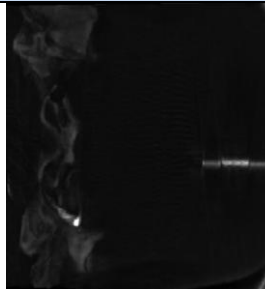
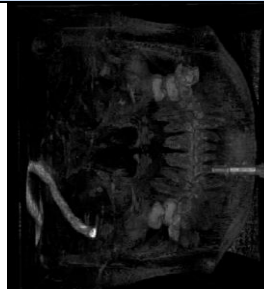
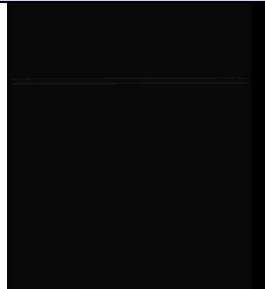
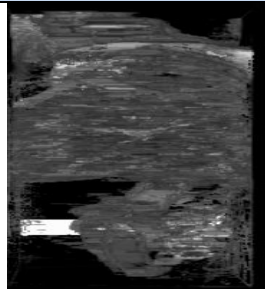
Figshare Dataset		Kaggle Dataset	
Original Image	Wireframe Model	Original Image	Wireframe Model
			
			
			
			

Fig. 4.8: Wireframe model of the input images on Fig-share and Kaggle Brain MRI Dataset

Front View			
Original Image	Wireframe	Original Image	Wireframe
			
Top View			
Original Image	Wireframe	Original Image	Wireframe
			
Bottom View			
Original Image	Wireframe	Original Image	Wireframe
			
Right View			
Original	Wireframe	Original Image	Wireframe
			

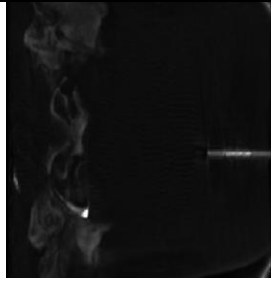
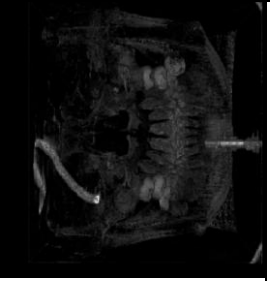
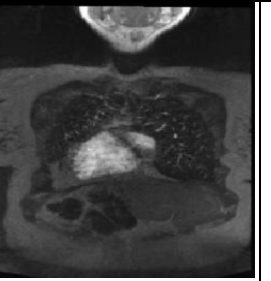
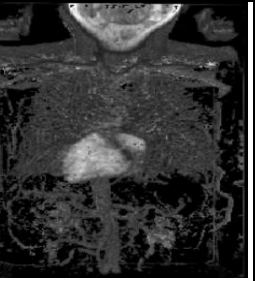
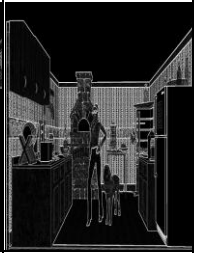
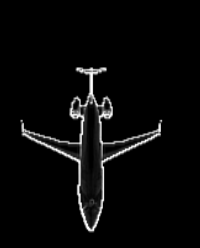
Left view			
Original	Wireframe	Original Image	Wireframe
			

Fig. 4.9: Wireframe model and their different 3D View of the Input Images on Medical MRI Dataset

Original	Sobel	Prewitt	Canny	Log	Proposed
					
					
					
					

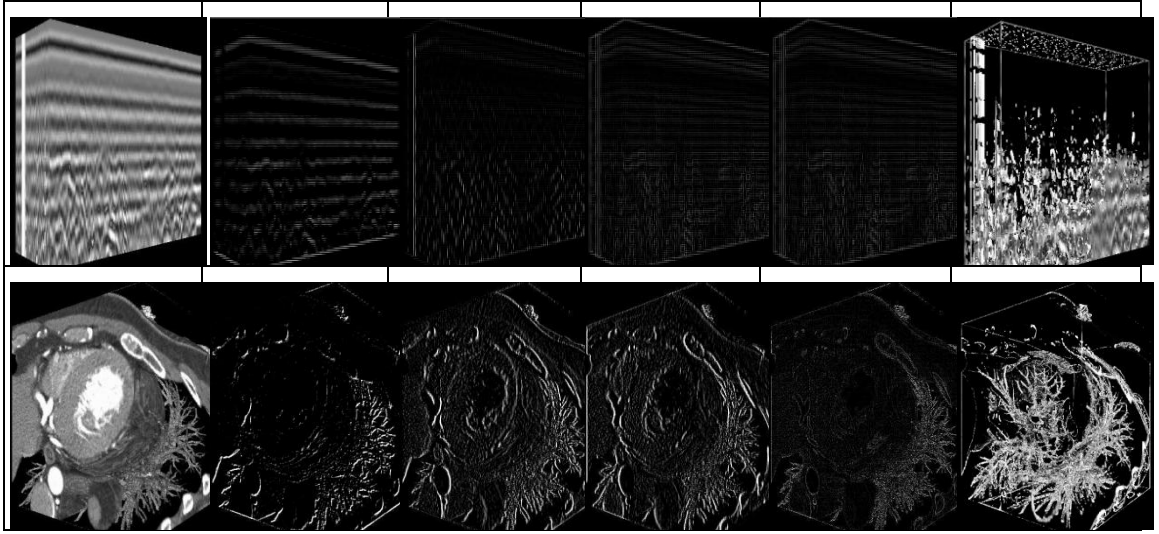


Fig.4. 10: Comparative Analysis of Wireframe Model of the Images on Example-Based Synthesis Dataset, Model-Net 40 Dataset and Medical MRI Dataset

The proposed algorithm can explore the internal details of the given input image. This technique is helpful for the medical practitioner for analyzing the images properly and identifying the diseases in the early stage. For the qualitative analysis of the proposed methodology, the results of the proposed algorithm are compared with the Sobel, Prewitt and Canny and log edge detection algorithms. The comparative analysis is performed on an example-based synthesis dataset, ModelNet 40 dataset and medical dataset. According to the experimental analysis, the proposed method gives better results compared to the existing state of art methods and results are shown in the **Fig. 4.10**. Further for qualitative and quantitative analysis the comparative analysis has been done with an edge detection algorithm. . The wireframe model has been utilized by majority of authors in order to analyze the various views and to reconstruct the 3D images hence the comparison is difficult with papers. Considering this, we have chosen some edge detection algorithms which are as follows:

- Dilation based 3-D edge detection (DBED)
- Erosion based 3-D edge detection (EBED)
- Morphological filtering of 3-D images for edge detection (MFED)
- Opening based 3-D edge detection (OBED)

The results of the above-mentioned algorithms are compared with the proposed method and these results are shown in **Fig 4.11**. As per the qualitative analysis, the quality of the wireframe model is far better than the existing edge detection method.

Table 4.2 and **4.3** also represent the comparative analysis and it includes the number of voxels present in the original image and the number of voxels present in the image after processing the image using various algorithms. As per the analysis number of voxels has been reduced after processing the image. DBED produced less number of voxels compared to the other algorithms but DBED fails as it is not able to provide internal details of the object properly.

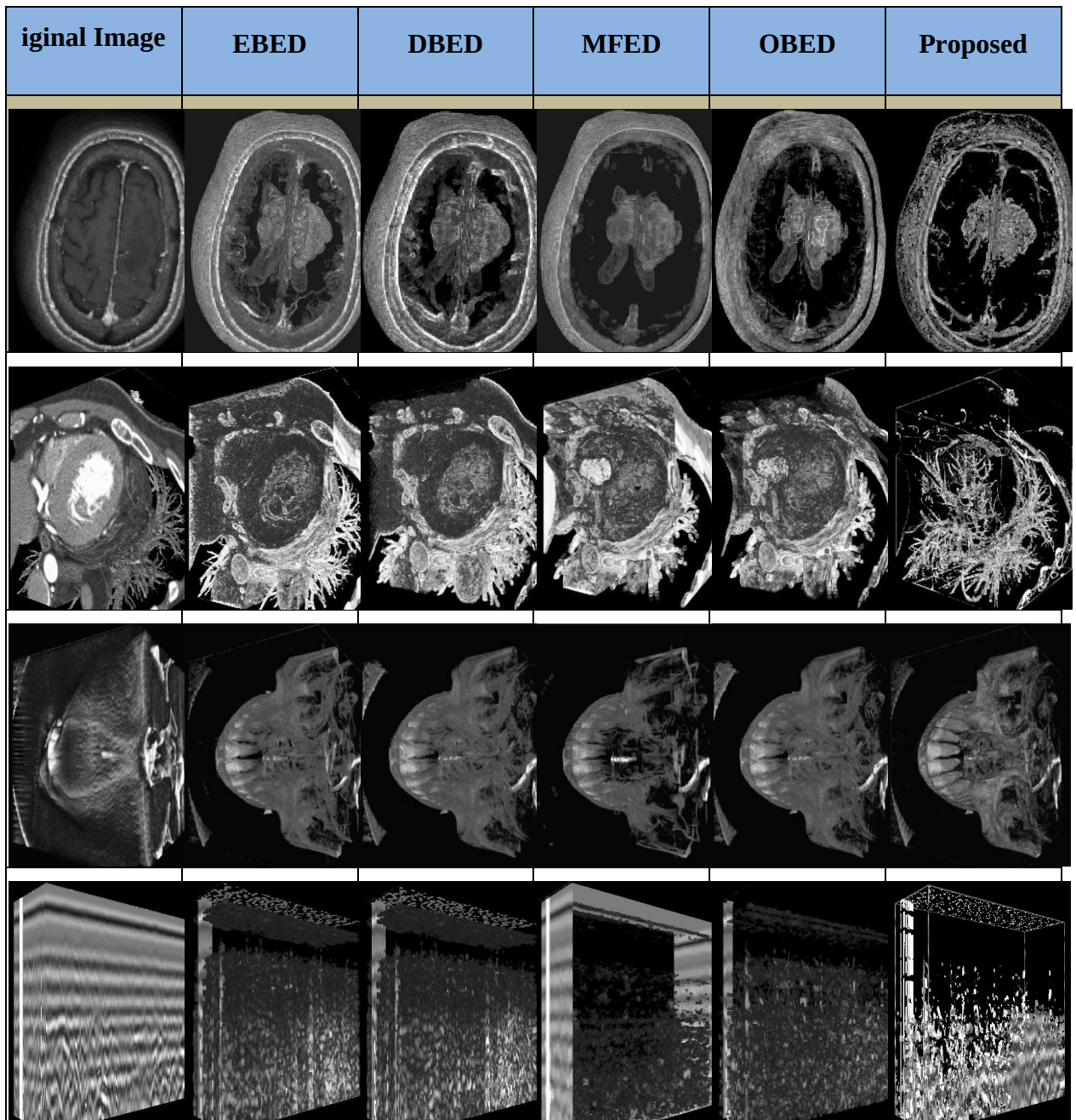


Fig.4. 11: The Input Image and its Processed Results on Different Algorithms

Table 4.2: Comparative Analysis of the Proposed Algorithm on Medical MRI Dataset

Comparative Analysis	Cerebrix Dataset			Toutarix Dataset		
Algorithm	Voxels Before Processing	Voxels After Processing	Processing Time In Seconds	Voxels Before Processing	Voxels After Processing	Processing Time In Seconds
EBED	1205998	327815	<1	9743894	3590022	8
DBED	1205998	360752	<1	9743894	3685942	8
MFED	1205998	107257	8.54	9743894	2032360	78.24
OBED	1205998	124252	2.85	9743894	2183937	25.99
Proposed	1205998	153059	1.12	9743894	3220742	5.03

Table 4.3: Comparative Analysis of the Proposed Algorithm on Medical MRI Dataset

Comparative Analysis	Skull Dataset			GPR VSP		
Algorithm	Voxels Before Processing	Voxels After Processing	Processing Time In Seconds	Voxels Before Processing	Voxels After Processing	Processing Time In Seconds
EBED	14834427	565475	16.00	5277502	2028118	3
DBED	14834427	614731	16.00	5277502	1817062	3
MFED	14834427	302602	139.70	5277502	382883	29.71
OBED	14834427	307344	46.93	5277502	464527	9.69
	14834427	405054	15.40	5277502	1025220	2.09

As shown in **Fig. 4.11**, proposed method can give minute and internal details of the object. The processing time of the proposed method is also compared with the existing algorithms and the processing time of the proposed methodology is comparatively lesser than existing algorithms because a single iteration is sufficient for converting an image into a wireframe model. The time complexity of each

algorithm is calculated and shown in **Table 4.4** and plotted in **Fig 4.12**. The time complexity of this approach is $O(n^3r^3) + O(n^3)$ where n represents the size of the image, r is the size of the structuring element. Here size of the structuring element is $3 \times 3 \times 3$, P -value is 9 and Q value is 3.

Table 4.4: Comparative Analysis of Time Complexity

Algorithm	Noise Handling	Preserve Important Information	Time Complexity
EBED	No	Yes	$O(n^3+r^3)$
DBED	No	Yes	$O(n^3+r^3)$
MFED	Yes	No	$P*O(n^3+r^3)$
OBED	Yes	No	$Q*O(n^3+r^3)$
Proposed Method	Yes	Yes	$O(n^3+r^3)+ O(n^3)$

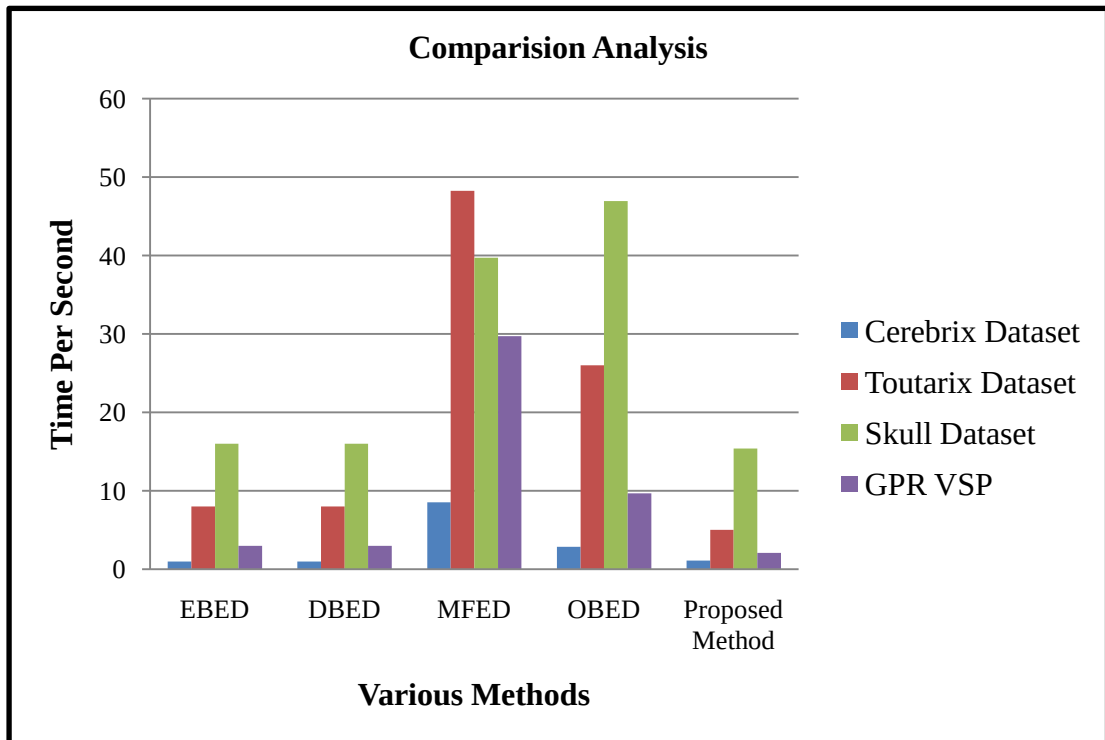


Fig. 4.12: Processing Time of Different Algorithms on Cerebrix, Toutarix, Skull and GPR VSP Dataset

4.3 Summary

The performance of the algorithm is compared with EBED, DBED, OBED, and MFED. It is observed that EBED and DBED algorithms take less time to process but these algorithms are not able to remove the noise from the image. MFED algorithm can remove the noise from the image also detects the edges but sometimes it also removes the sharp corners and small holes of the objects. The processing time of the MFED algorithm is 8 times more than the EBED and DBED and proposed methodology. OBED algorithm performs well as it can handle the noise, preserving the useful information and processing time is lesser than MFED, EBED, DBED but more than proposed methodology. The proposed method performs well because it removes the noise from the image, preserving the useful information, preserving all the boundary of the objects and taking less processing time compared to state of art methods. The proposed method can find the wireframe model in a single iteration and the processing time is less than the other existing algorithms. For a better analysis of the algorithm, the proposed method is evaluated on seven different types of datasets and the results are outperformed. The algorithm is able to find the wireframe model for a simple, complex image and the image which consists of more than one object.

To analyze the performance of the proposed algorithm, various types of datasets are taken into considerations. According to the analysis, algorithms work well with various types of datasets. However, we further focused our research and for subsequent studies only medical images are considered.

To achieve the 2nd objective, wireframe model is used as an input image and further a novel algorithm is implemented to extract the features from 3D image. The detailed description of feature extraction algorithm is discussed in the next chapter.

CHAPTER-5

CONSTRUCTION AND RECONSTRUCTION OF 3D IMAGES

5.1 Introduction

The two main tasks of pattern recognition are description and classification. The pattern recognition system describes the object in the image and then puts the object into a category based on the description. Pattern recognition can be implemented using two approaches: statistical and structural. These two techniques use different methods for description and classification of the objects. Statistical approaches [180–182] use decision-theory ideas to tell which items belong to which classes based on their quantitative characteristics. The Structural Pattern Recognition Approach [183–185] is sometimes called the Syntactic Pattern Recognition Approach because it uses the idea of syntactic grammars to tell the difference between objects that belong to different classes based on their shape, size, or structural features. Some researchers combined both structural and statistical approach for pattern recognition. The statistical approach is well explored by many researchers but structural pattern recognition still doesn't explore adequately because it requires domain knowledge for the description and classification purpose. Therefore, Structural pattern recognition techniques are domain-specific.

Three-dimensional image construction and reconstruction are important parts of the field of computer vision, which is used in many real-world situations. Researchers have been working in this area for the past 30 years because construction and reconstruction are important parts of medical imaging. By putting the 3D image back together, we can find information about the patient's lesion, which could be a new and accurate way to diagnose illness and add clinical value. In response to this, we came up with new ways to build and reconstruct images.

An actual 3D clay model was created to identify syntactic grammar, and the different possible patterns in the image were found. As per the domain knowledge, a pixel could be present in any of 26 possible directions in a 3D image, according to domain knowledge, and we included all possible directions in the proposed method. First,

using syntactic pattern recognition, features are extracted from the images which are further used for reconstruction of the image. The proposed methodology is capable of extracting in-depth features in all possible orientations and planes, as well as textual representation of the 3D image. This feature map is nothing more than a syntactic string containing direction and length information.

Similarly, a unique algorithm for image reconstruction has been proposed. The knowledge vector is used as an input in this technique, and the algorithm can reconstruct a 3D image. Reconstruction allows us to investigate the interior details of 3D images, such as the object's size, form, and structure, potentially advancing us in the field of medical image processing.

This is the first time in the field of image processing where a three dimensional image can be reconstructed using textual information and it could be a great contribution in the field of structural pattern recognition as well as adds a new value in the field of image processing. As we discussed already that structural pattern recognition techniques are domain-specific so that we considered only medical images for our research. Many researchers are working on the reconstruction of a 3D image because the minimally invasive procedure is getting popular nowadays. After all, it provides better accuracy and minimum recovery time. In the above-mentioned method, the surgical navigation system will always play an important and critical role. It helps the doctors to focus on a particular position, edge, or outline of the image. But Most of the surgical navigation systems are based on two-dimensional medical images but very little development in the field of three-dimensional medical images. Therefore we developed a platform for the reconstruction of the three dimensional images. This System takes the series of 2D/ 3D medical images as an input and converts these series of images into textual form first and then in the second stage this textual information/knowledge vector would be taken as an input and reconstruction of the 3D image would be done in this stage. It displays the human organs on the computer screen which gives convenience to the doctors for the diagnosis of the disease. This three dimensional reconstruction gives the facility to doctors to focus on a particular point and find the nature of the lesion and its surrounding tissues, rotation of the image, traverse the image from inside, zoom in and zoom out, thus it helps the doctors

to diagnose the diseases, its severity and improve the accuracy and decrease the diagnosis time.

5.2 Proposed Methodology

The proposed methodology is divided into three parts and the flow chart of the proposed method is shown in **Fig. 5.1**. A detailed explanation is discussed further:

5.2.1 Knowledge Acquisition

Knowledge acquisition is a method on which domain knowledge is acquired for extracting the features of a specific application or domain. This is the first step of structural pattern recognition based methods. Generally, Structural approach uses syntactic grammars concept for the discrimination between objects which belongs to different class based upon their shape, size, or structural features. Structural pattern recognition techniques are domain-specific.

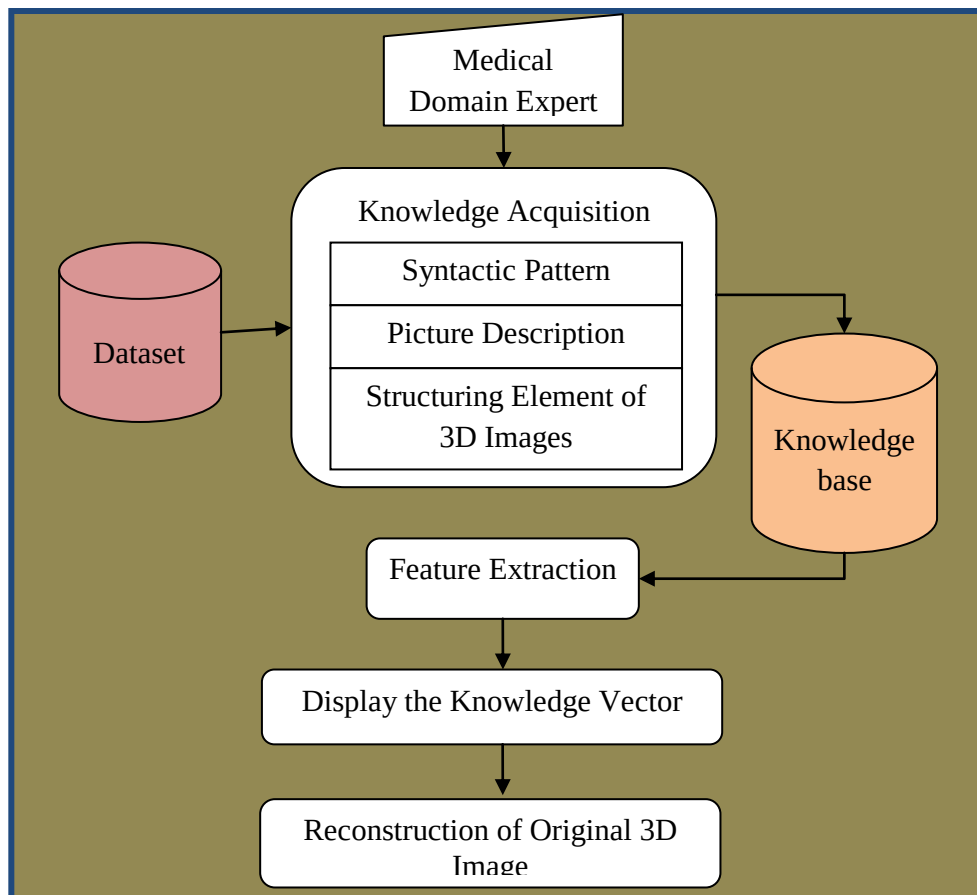


Fig. 5.1: Flow Chart of the Proposed Methodology

Further knowledge acquisition steps are explained in detail.

- **Syntactic Pattern Recognition**

The new and less explored approach of pattern recognition is Syntactic pattern recognition which consists of the formal language theory concepts. Syntactic pattern recognition term is a synonym of the linguistic, grammatical, and structural pattern recognition system. The formal language theory was originated by Noam Chomsky in the 1950's with the development of a mathematical model of grammar. The concept is described as follows:

Alphabet is a finite set of symbols. A word is any finite string that consists of symbols from the alphabets. For example a valid word of alphabets $\{a, b\}$ is $\{a, b, ab, ba, bb, aa \dots\}$. A word with no symbol is an empty word that is denoted by Λ . A language that consists of a set of words over a finite set of alphabets. Every natural language follows some grammar rules. The grammar of the formal language consists of four tuples.

$$G = \{P_N, P_T, P_r, R\}$$

Here

P_N = Set of variables or terminals.

P_T = Set of constants or non-terminals.

P_r = set of production rules.

R_s = Root symbol or start symbol.

P = Union of P_N and P_T

Here P_N and P_T are disjoint set and R always belongs to either P_N or P_T .

Set of empty words denoted by P^* which is also called free Monoid. P^+ is a set of sentences of $V^* - \Lambda$ which is called a free semi-group.

The language which is generated by these tuples G is called $L(G)$. It should satisfy two conditions: Each string should only consist of terminals. Every string should derive from the root R_s by applying the suitable production rules. Production rules consist of expressions in the form of $X \rightarrow Y$. It means string X can be replaced by string Y . Where X = String in P^+

$$Y = \text{String in } P^*$$

In the proposed methodology, the set of terminals or constants are denoted by:

$$P_T =$$

$$\{ R, DR, D, DL, L, UL, U, UR, B, BR, BDR, BD, BDL, BL, BUL, BU, BUR, F, FR, FDR, FD, FDL, FL, FUL, FU, FUR \}$$

Root R can be changed according to the object position. Here these symbols represent the direction in which a pixel could be present. Here R, D, L, U, B , and F are elementary symbols, and $DR, DL, UL, UR, BR, BDR, BD, BDL, BL, BUL, BU, BUR, FR, FDR, FD, FDL, FL, FUL, FU, FUR$ are composite symbols of the alphabet P_T . The set of production rules are generated through the Normal algorithm which was introduced by A.A. Markov. The Normal algorithm is used to recognize the angle of change in a particular direction during tracking of the contour. This is done with the help of the look-ahead tracing (LAT) method.

- **Picture Description Language (PDL)**

Picture description language is an application of semantic/linguistic concepts of pattern recognition. The pattern can be represented as string grammar, and this string grammar can be obtained by simply juxtaposing two strings. Juxtaposition is nothing but keeping the two objects together without losing their identities. This can be done with concatenation also but it involves some spatial arrangements and there is a possibility of losing the identity or losing some information related to the objects. Juxtaposition is easy and reliable because it only involves the head and tail position (shown in **Fig. 5.2**). Based on this permissible form of juxtaposition and because each primitive is abstracted as a directed line segment, it is evident that the structure of PDL are directed graphs and also that these structures can be handled by string grammars. The rules of a juxtaposition of PDL are as follows: Blank primitives must be used for generating disjoint structures. A null point primitive has an identical head and tail. The mechanics of PDL could be used to obtain the contour/ wireframe of a pattern. To illustrate this mechanism, consider the following PDL grammar: $G = \{P_N, P_T, P_r, R_s\}$

$$\text{Where } P_N = \{R_s, P_1, P_2, P_3, P_4, P_5, P_6, P_7\}$$

$P_T =$

{ R, DR, D, DL, L, UL, U, UR, B, BR, BDR, BD, BDL, BL, BUL, BU, BUR, F, FR, FDR, FD, FDL, FL, FUL, FU, FUR }

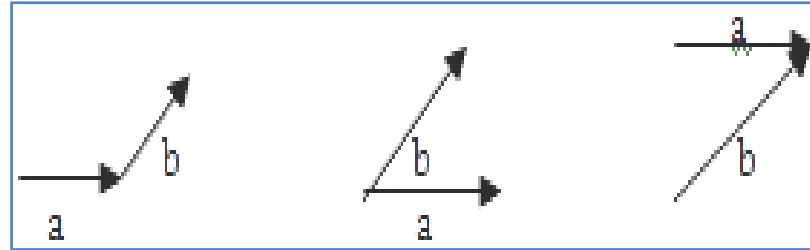


Fig. 5.2: Possible form of juxtaposition

A set of production rules are represented by P_r and production rules are shown in the **Table 5.1**.

Table 5.1: Set of Production Rules

R_s	$\rightarrow$	R+P1
P1	$\rightarrow$	DR+P2
P2	$\rightarrow$	D+P3
P3	$\rightarrow$	DL+P4
P4	$\rightarrow$	L+P5
P5	$\rightarrow$	UL+P6
P6	$\rightarrow$	U+P7
P7	$\rightarrow$	UR+P8
P8	$\rightarrow$	B+P9
P9	$\rightarrow$	BR+P10
P10	$\rightarrow$	BDR+P11
P11	$\rightarrow$	BD+P12
P12	$\rightarrow$	BDL+P13
P13	$\rightarrow$	BL+P14

P14	→	BUL+P15
P15	→	BU+P16
P16	→	BUR+P17
P17	→	F+P18
P18	→	FR+P19
P19	→	FDR+P20
P20	→	FD+P21
P21	→	FDL+P22
P22	→	FL+P23
P23	→	FUL+P24
P24	→	FU+P25
P26	→	FUR

The result obtained by applying all the productions is:

R+(DR+(D+(DL+(L+(UL+(U+(UR+(B+ BR+ (BDR+ (BD+ (BDL+ (BL+ (BUL+
 (BU+ (BUR+ (F+ (FR+ (FDR+ (FD+ (FDL+ (FL+(FUL+
 (FU+FUR))))))))))))))))))))))

The above production system creates an octagon as represented in the **Fig. 5.3**:

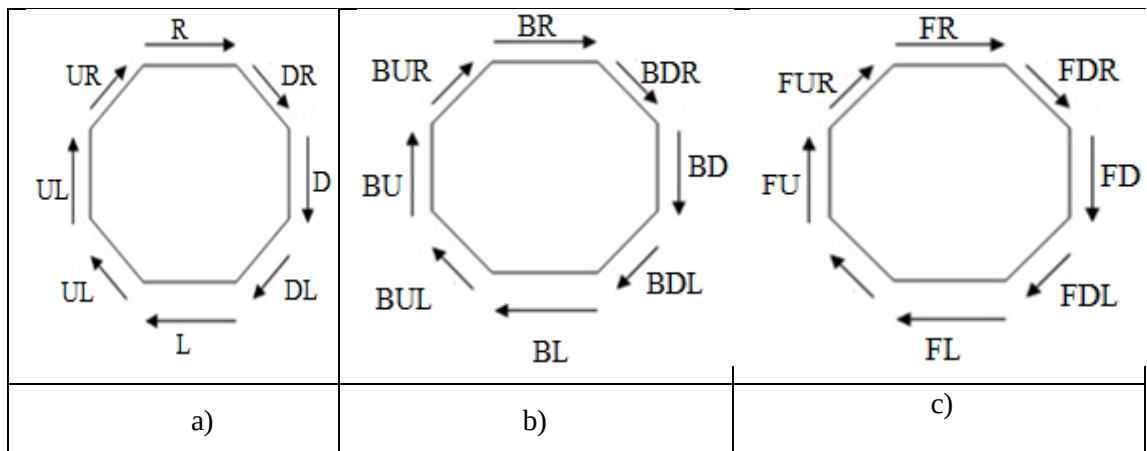


Fig. 5.3: Pictorial Representation of a) Center Plane b) Back Plane and c) Front Plane with Corresponding Pixel Directions

The directions of the pixels are defined with the help of production rules. Suppose an alphabet A is given then A^* could be constructed over A which consists of all the possible words containing A and null word also. Hence any subset of A^* is a formal language. According to this theory, any digital image consists of a set of patterns and the same digital image can be represented as a language that consist a finite array of vertex/ vertices. As mentioned, the PDL is a language of representing the digital image into the regular array of vertices.

- **Structuring Elements of 3D Images**

The size and shape of a 2D/ 3D processed image always depend on the structuring elements which have been chosen for processing the image. In the case of a 2D image, if the window size/ neighborhood structure is 3×3 then a maximum of 16 convex polygons can be present. For a 3D image if the window size/ neighborhood structure is $3 \times 3 \times 3$ then a maximum of 256 convex polyhedrons can be present. In the same way, we can find the number of concave polygons and polyhedrons. These polygons and polyhedrons are called a structuring element which can be used for the processing of 2D and 3D images. The number of structuring elements always depends on the size of the neighborhood. This concept was introduced by E.G Rajan and Jirawit Lerdsinmongkol [72]. Consider an image with dimensions of $(m \times n \times d)$, where m is the image's width, n is its height, and d is the dataset's depth. Because the structural element is $3 \times 3 \times 3$, there will be 27 neighborhoods in total, including the centre pixel. In the same way, we can have different structuring elements which are represented in **Fig. 5.4**. In our research, we used a neighborhood structure of size $3 \times 3 \times 3$ so we have 256 possible structuring elements. A detailed description of 3D convex polyhedrons is given below:

- **Three-Dimensional Rectangular Convex Polyhedrons**

The idea of constructing 3D convex polyhedrons is the same as 2D convex polygons. As we discussed earlier, a 3D image consists of voxels and each voxel itself represents a cube. So a 3D image is an array of cubes. Here we are considering the structuring element of size $3 \times 3 \times 3$ so it consists of 27 neighborhood structures.

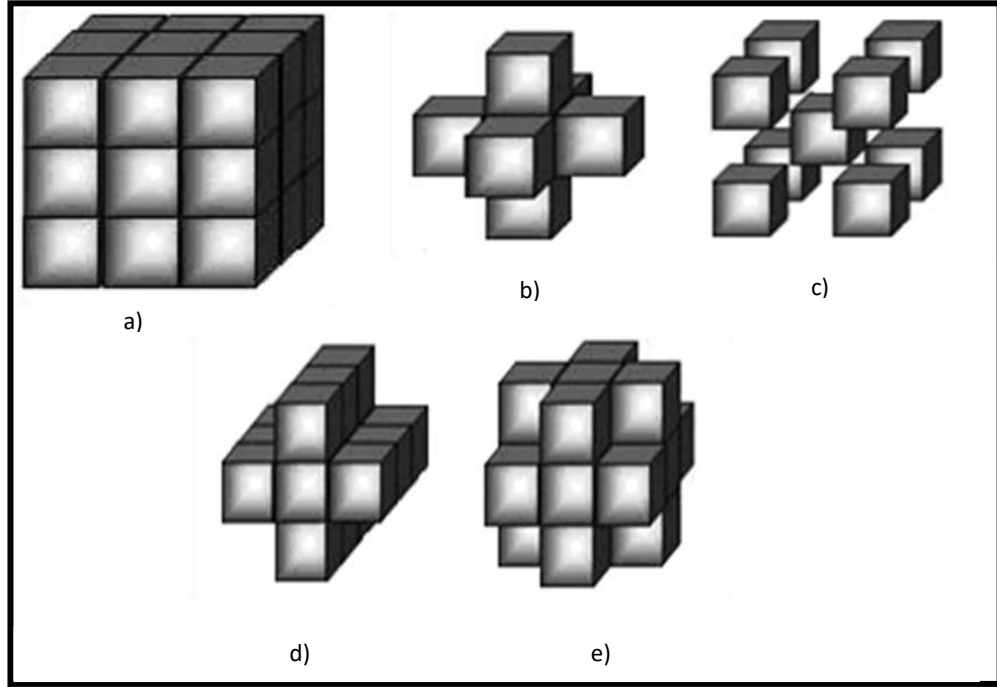


Fig. 5.4: a) 27-Neighbourhood b) 7-Neighbourhood c) 9-Neighbourhood d) 15-Neighbourhood e) 19-Neighbourhood Structuring Elements

The smallest size of possible convex polyhedrons consists of 7 neighborhoods. The labeling of the $3 \times 3 \times 3$ array and their possible coordinates are represented further in **Fig. 5.5** and three different planes are considered where the K^{th} plane is the central plane and $(k - 1)$ and $(K + 1)$ is the front and rear plane. In a $3 \times 3 \times 3$ size of array total of 8 corner pixels so there is a possibility of any corner pixel would not be available or any two corner pixels would not be available etc. Therefore, there are 256 different possible ways of constructing the convex polyhedrons w. r. t central pixel 14. The formula is given below:

$$\sum_{k=0}^{k=8} C_k = {}^8C_0 + {}^8C_1 + {}^8C_2 + {}^8C_3 + {}^8C_4 + {}^8C_5 + {}^8C_6 + {}^8C_7 + {}^8C_8 = 256 \quad (5.1)$$

Here, these convex polyhedrons are divided into 9 categories:

- When all the corner pixels are present then the number of possible convex polyhedrons will be ${}^8C_0$ which is equivalent to 1.

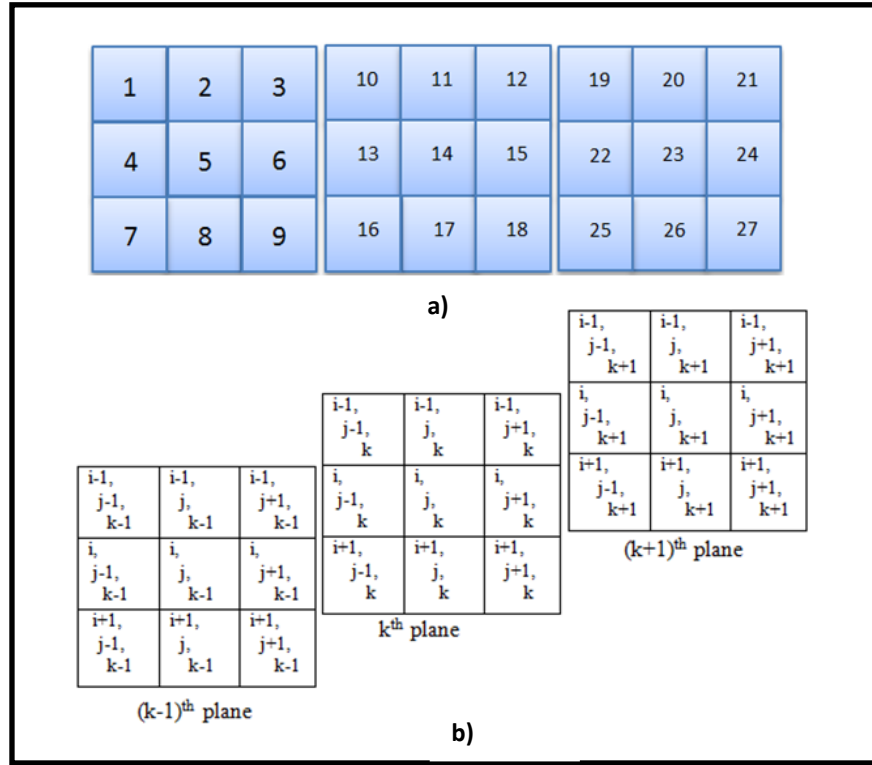


Fig. 5.5: a) 27-Neighborhood Structure of 3x3x3 Windows b) Representation of Coordinates

- When one of the corner pixels is not available then the number of possible convex polyhedrons will be ${}^8C_1$ which is equivalent to 8.
- When two of the corner pixels are not available then the number of possible convex polyhedrons will be ${}^8C_2$ which is equivalent to 28.
- When three of the corner pixels are not available then the number of possible convex polyhedrons will be ${}^8C_3$ which is equivalent to 56.
- When all the corner pixels are not available then the number of possible convex polyhedrons will be ${}^8C_4$ which is equivalent to 70.
- When all the corner pixels are not available then the number of possible convex polyhedrons will be ${}^8C_5$ which is equivalent to 56.
- When all the corner pixels are not available then the number of possible convex polyhedrons will be ${}^8C_6$ which is equivalent to 28.
- When all the corner pixels are not available then the number of possible convex polyhedrons will be ${}^8C_7$ which is equivalent to 8.
- When all the corner pixels are not available then the number of possible convex polyhedrons will be ${}^8C_8$ which is equivalent to 1.

Further, all the possible 256 convex polyhedrons are represented in **Table 5.2** and here A, B, C, D, E, F, G, H and I represent the Possible eliminated pixels, and group P, Q, R, S, T, U, V, W, and X represents the possibility of convex polyhedron after removing the pixels.

Table 5.2: Representation of a Possible Combination of Convex Polyhedrons

Group P contains 1 possible combination			
A={}			
$P_a = \{1,3,7,9,19,21,25,27\}$			
Group Q contains 8 possible combinations			
$B_a = \{1\}$ $Q_1 = \{3,7,9,19,21,25,27\}$	$B_b = \{3\}$ $Q_3 = \{1,7,9,19,21,25,27\}$	$B_c = \{7\}$ $Q_7 = \{1,3,9,19,21,25,27\}$	$B_d = \{9\}$ $Q_9 = \{1,3,7,19,21,25,27\}$
$B_e = \{19\}$ $Q_e = \{1,3,7,9,21,25,27\}$	$B_f = \{21\}$ $Q_f = \{1,3,7,9,19,25,27\}$	$B_g = \{25\}$ $Q_a = \{1,3,7,9,19,21,27\}$	$B_h = \{27\}$ $Q_a = \{1,3,7,9,19,21,25\}$
Group R contains 28 combinations			
$C_a = \{1,3\}$ $R_{1,3} = \{7,9,19,21,25,27\}$	$C_b = \{1,7\}$ $R_{1,7} = \{3,9,19,21,25,27\}$	$C_c = \{1,9\}$ $R_{1,9} = \{3,7,19,21,25,27\}$	$C_d = \{1,19\}$ $R_{1,19} = \{3,7,9,21,25,27\}$
$C_e = \{1,21\}$ $R_{1,21} = \{3,7,9,19,25,27\}$	$C_f = \{1,25\}$ $R_{1,25} = \{3,7,9,19,21,27\}$	$C_g = \{1,27\}$ $R_{1,27} = \{3,7,9,19,21,25\}$	$C_h = \{3,7\}$ $R_{3,7} = \{1,9,19,21,25,27\}$
$C_i = \{3,9\}$ $R_{3,9} = \{1,7,19,21,25,27\}$	$C_j = \{3,19\}$ $R_{3,19} = \{1,7,9,21,25,27\}$	$C_k = \{3,21\}$ $R_{3,21} = \{1,7,9,19,25,27\}$	$C_l = \{3,25\}$ $R_{3,25} = \{1,7,9,19,21,27\}$
$C_m = \{3,27\}$ $R_{3,27} = \{1,7,9,19,21,25\}$	$C_n = \{7,9\}$ $R_{7,9} = \{1,3,19,21,25,27\}$	$C_o = \{7,19\}$ $R_{7,19} = \{1,3,9,21,25,27\}$	$C_p = \{7,21\}$ $R_{7,21} = \{1,3,9,19,25,27\}$
$C_q = \{7,25\}$ $R_{7,25} = \{1,3,9,19,21,27\}$	$C_r = \{7,27\}$ $R_{7,27} = \{1,3,9,19,21,25\}$	$C_s = \{9,19\}$ $R_{9,19} = \{1,3,7,21,25,27\}$	$C_t = \{9,21\}$ $R_{9,21} = \{1,3,7,19,25,27\}$
$C_u = \{9,25\}$ $R_{9,25} = \{1,3,7,19,21,27\}$	$C_v = \{9,27\}$ $R_{9,27} = \{1,3,7,19,21,25\}$	$C_w = \{19,21\}$ $R_{19,21} = \{1,3,7,9,25,27\}$	$C_x = \{19,25\}$ $R_{19,25} = \{1,3,7,9,21,27\}$
$C_y = \{19,27\}$ $R_{19,27} = \{1,3,7,9,21,25\}$	$C_z = \{21,25\}$ $R_{21,25} = \{1,3,7,9,19,27\}$	$C_{za} = \{21,27\}$ $R_{21,27} = \{1,3,7,9,19,25\}$	$C_{zb} = \{25,27\}$ $C_{25,27} = \{1,3,7,9,19,21\}$
Group S contains 56 combinations			
$D_a = \{1,3,7\}$	$D_b = \{1,3,9\}$	$D_c = \{1,3,19\}$	$D_d = \{1,3,21\}$

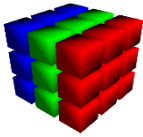
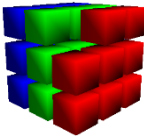
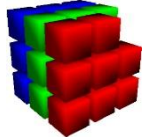
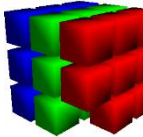
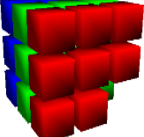
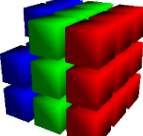
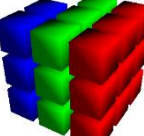
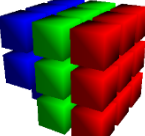
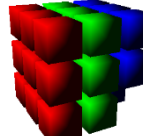
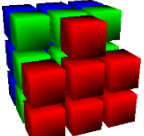
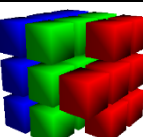
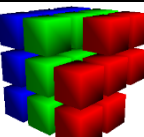
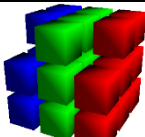
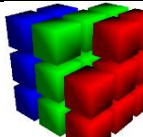
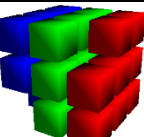
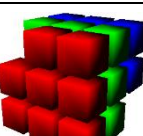
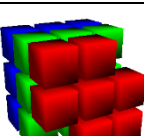
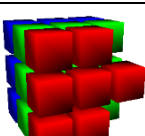
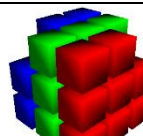
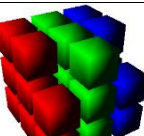
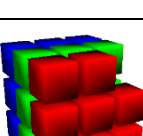
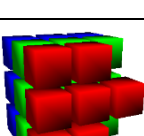
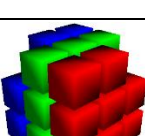
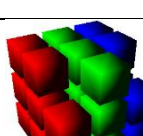
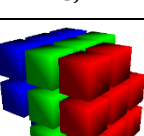
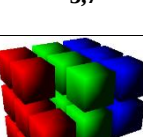
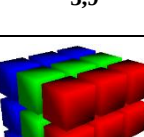
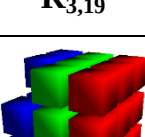
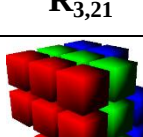
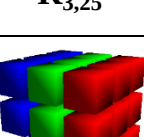
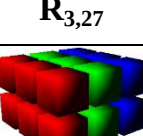
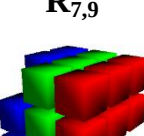
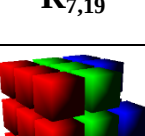
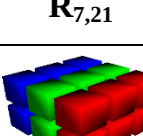
$S_{1,3,7}=\{9,19,21,25,27\}$	$S_{1,3,9}=\{7,19,21,25,27\}$	$S_{1,3,19}=\{7,9,21,25,27\}$	$S_{1,3,21}=\{7,9,19,25,27\}$
$D_e=\{1,3,25\}$ $S_{1,3,25}=\{7,9,19,21,27\}$	$D_f=\{1,3,27\}$ $S_{1,3,27}=\{7,9,19,21,25\}$	$D_g=\{1,7,9\}$ $S_{1,7,9}=\{3,19,21,25,27\}$	$D_h=\{1,7,19\}$ $S_{1,7,19}=\{3,9,21,25,27\}$
$D_i=\{1,7,21\}$ $S_{1,7,21}=\{3,9,19,25,27\}$	$D_j=\{1,7,25\}$ $S_{1,7,25}=\{3,9,19,21,27\}$	$D_k=\{1,7,27\}$ $S_{1,7,27}=\{3,9,19,21,25\}$	$D_l=\{1,9,19\}$ $S_{1,9,19}=\{3,7,21,25,27\}$
$D_m=\{1,9,21\}$ $S_{1,9,21}=\{3,7,19,25,27\}$	$D_n=\{1,9,25\}$ $S_{1,9,25}=\{3,7,19,21,27\}$	$D_o=\{1,9,27\}$ $S_{1,9,27}=\{3,7,19,21,25\}$	$D_p=\{1,19,21\}$ $S_{1,19,21}=\{3,7,9,25,27\}$
$D_q=\{1,19,25\}$ $S_{1,19,25}=\{3,7,9,21,27\}$	$D_r=\{1,19,27\}$ $S_{1,19,27}=\{3,7,9,21,25\}$	$D_s=\{1,21,25\}$ $S_{1,21,25}=\{3,7,9,19,27\}$	$D_t=\{1,21,27\}$ $S_{1,21,27}=\{3,7,9,19,25\}$
$D_u=\{1,25,27\}$ $S_{1,25,27}=\{3,7,9,19,21\}$	$D_v=\{3,7,9\}$ $S_{3,7,9}=\{1,19,21,25,27\}$	$D_w=\{3,7,19\}$ $S_{3,7,19}=\{1,9,21,25,27\}$	$D_x=\{3,7,21\}$ $S_{3,7,21}=\{1,9,19,25,27\}$
$D_y=\{3,7,25\}$ $S_{3,7,25}=\{1,9,19,21,27\}$	$D_z=\{3,7,27\}$ $S_{3,7,27}=\{1,9,19,21,25\}$	$D_{za}=\{3,9,19\}$ $S_{3,9,19}=\{1,7,21,25,27\}$	$D_{zb}=\{3,9,21\}$ $S_{3,9,21}=\{1,7,19,25,27\}$
$D_{zc}=\{3,9,25\}$ $S_{3,9,25}=\{1,7,19,21,27\}$	$D_{zd}=\{3,9,27\}$ $S_{3,9,27}=\{1,7,19,21,25\}$	$D_{ze}=\{3,19,21\}$ $S_{3,19,21}=\{1,7,9,25,27\}$	$D_{zf}=\{3,19,25\}$ $S_{3,19,25}=\{1,7,9,21,27\}$
$D_{zg}=\{3,19,27\}$ $S_{3,19,27}=\{1,7,9,21,25\}$	$D_{zh}=\{3,21,25\}$ $S_{3,21,25}=\{1,7,9,19,27\}$	$D_{zi}=\{3,21,27\}$ $S_{3,21,27}=\{1,7,9,19,25\}$	$D_{zj}=\{3,25,27\}$ $S_{3,25,27}=\{1,7,9,19,21\}$
$D_{zk}=\{7,9,19\}$ $S_{7,9,19}=\{1,3,21,25,27\}$	$D_{zl}=\{7,9,21\}$ $S_{7,9,21}=\{1,3,19,25,27\}$	$D_{zm}=\{7,9,25\}$ $S_{7,9,25}=\{1,3,19,21,27\}$	$D_{zn}=\{7,9,27\}$ $S_{7,9,27}=\{1,3,19,21,25\}$
$D_{zo}=\{7,19,21\}$ $S_{7,19,21}=\{1,3,9,25,27\}$	$D_{zp}=\{7,19,25\}$ $S_{7,19,25}=\{1,3,9,21,27\}$	$D_{zq}=\{7,19,27\}$ $S_{7,19,27}=\{1,3,9,21,25\}$	$D_{zr}=\{7,21,25\}$ $S_{7,21,25}=\{1,3,9,19,27\}$
$D_{zs}=\{7,21,27\}$ $S_{7,21,27}=\{1,3,9,19,25\}$	$D_{zt}=\{7,25,27\}$ $S_{7,25,27}=\{1,3,9,19,21\}$	$D_{zu}=\{9,19,21\}$ $S_{9,19,21}=\{1,3,7,25,27\}$	$D_{zv}=\{9,19,25\}$ $S_{9,19,25}=\{1,3,7,21,27\}$
$D_{zw}=\{9,19,27\}$ $S_{9,19,27}=\{1,3,7,21,25\}$	$D_{zx}=\{9,21,25\}$ $S_{9,21,25}=\{1,3,7,19,27\}$	$D_{zy}=\{9,21,27\}$ $S_{9,21,27}=\{1,3,7,19,25\}$	$D_{zz}=\{9,25,27\}$ $S_{9,25,27}=\{1,3,7,19,21\}$
$D_{ya}=\{19,21,25\}$ $S_{19,21,25}=\{1,3,7,9,27\}$	$D_{yb}=\{19,21,27\}$ $S_{19,21,27}=\{1,3,7,9,25\}$	$D_{yc}=\{19,25,27\}$ $S_{19,25,27}=\{1,3,7,9,21\}$	$D_{yd}=\{21,25,27\}$ $S_{21,25,27}=\{1,3,7,9,19\}$
Group T contains 70 combinations			
$E_a=\{1,3,7,9\}$ $T_{1,3,7,9}=\{19,21,25,27\}$	$E_b=\{1,3,7,19\}$ $T_{1,3,7,19}=\{9,21,25,27\}$	$E_c=\{1,3,7,9\}$ $T_{1,3,7,21}=\{9,19,25,27\}$	$E_d=\{1,3,7,25\}$ $T_{1,3,7,25}=\{9,19,21,27\}$

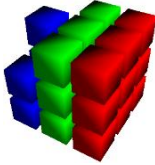
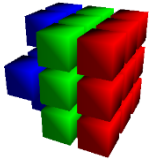
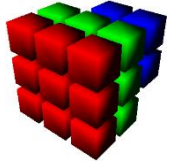
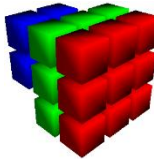
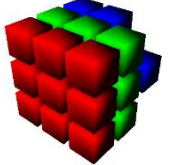
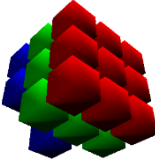
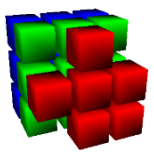
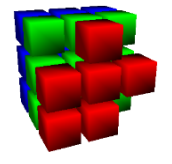
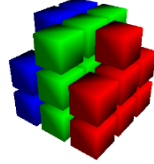
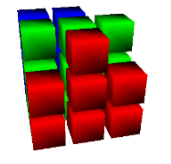
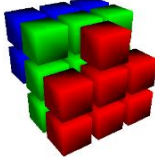
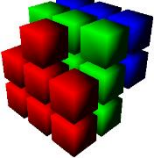
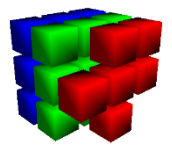
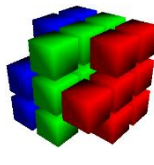
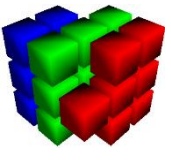
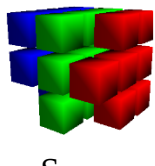
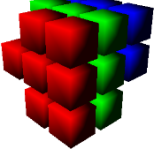
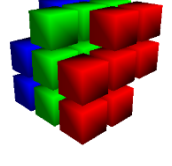
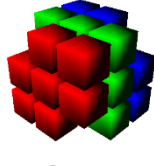
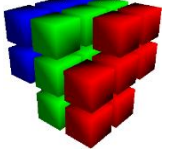
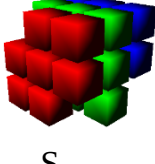
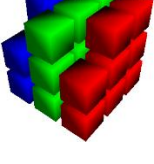
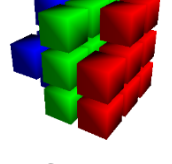
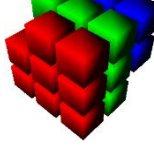
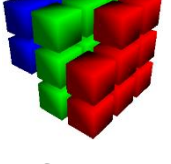
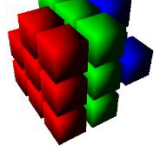
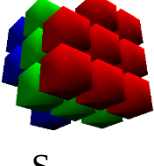
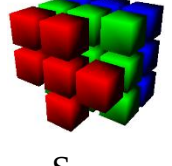
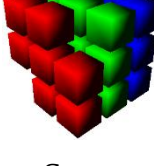
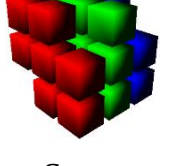
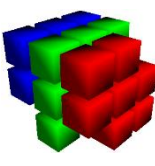
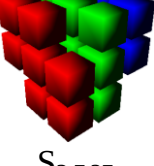
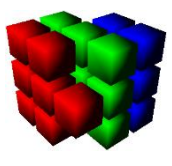
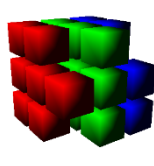
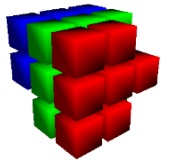
$E_f=\{1,3,7,27\}$ $T_{1,3,7,27}=\{9,19,21,25\}$	$E_g=\{1,3,9,19\}$ $T_{1,3,9,19}=\{7,21,25,27\}$	$E_h=\{1,3,9,21\}$ $T_{1,3,9,21}=\{7,19,25,27\}$	$E_i=\{1,3,9,25\}$ $T_{1,3,9,25}=\{7,19,21,27\}$
$E_k=\{1,3,9,27\}$ $T_{1,3,9,27}=\{7,19,21,25\}$	$E_l=\{1,3,19,21\}$ $T_{1,3,19,21}=\{7,9,25,27\}$	$E_m=\{1,3,19,25\}$ $T_{1,3,19,25}=\{7,9,21,27\}$	$E_n=\{1,3,19,27\}$ $T_{1,3,19,27}=\{7,9,21,25\}$
$E_o=\{1,3,21,25\}$ $T_{1,3,21,25}=\{7,9,19,27\}$	$E_p=\{1,3,21,27\}$ $T_{1,3,21,27}=\{7,9,19,25\}$	$E_q=\{1,3,25,27\}$ $T_{1,3,25,27}=\{7,9,19,21\}$	$E_r=\{1,7,9,19\}$ $T_{1,7,9,19}=\{3,21,25,27\}$
$E_s=\{1,7,9,21\}$ $T_{1,7,9,21}=\{3,19,25,27\}$	$E_t=\{1,7,9,25\}$ $T_{1,7,9,25}=\{3,19,21,27\}$	$E_u=\{1,7,9,27\}$ $T_{1,7,9,27}=\{3,19,21,25\}$	$E_v=\{1,3,7,9\}$ $T_{1,7,19,21}=\{3,9,25,27\}$
$E_w=\{1,7,19,25\}$ $T_{1,7,19,25}=\{3,9,21,27\}$	$E_x=\{1,7,19,27\}$ $T_{1,7,19,27}=\{3,9,21,25\}$	$E_y=\{1,7,21,25\}$ $T_{1,7,21,25}=\{3,9,19,27\}$	$E_z=\{1,7,21,27\}$ $T_{1,7,21,27}=\{3,9,19,25\}$
$E_{za}=\{1,7,25,27\}$ $T_{1,7,25,27}=\{3,9,19,21\}$	$E_{zb}=\{1,9,19,21\}$ $T_{1,9,19,21}=\{3,7,25,27\}$	$E_{zc}=\{1,9,19,25\}$ $T_{1,9,19,25}=\{3,7,21,27\}$	$E_{zd}=\{1,9,19,27\}$ $T_{1,9,19,27}=\{3,7,21,25\}$
$E_{ze}=\{1,9,21,25\}$ $T_{1,9,21,25}=\{3,7,19,27\}$	$E_{zf}=\{1,9,21,27\}$ $T_{1,9,21,27}=\{3,7,19,25\}$	$E_{zg}=\{1,9,25,27\}$ $T_{1,9,25,27}=\{3,7,19,21\}$	$E_{zh}=\{1,19,25,27\}$ $T_{1,19,21,25}=\{3,7,9,27\}$
$E_{zi}=\{1,19,21,27\}$ $T_{1,19,21,27}=\{3,7,9,25\}$	$E_{zj}=\{1,19,25,27\}$ $T_{1,19,25,27}=\{3,7,9,21\}$	$E_{zk}=\{1,21,25,27\}$ $T_{1,21,25,27}=\{3,7,9,19\}$	$E_{zl}=\{1,21,25,27\}$ $T_{3,7,9,19}=\{1,21,25,27\}$
$E_{zm}=\{3,7,9,21\}$ $T_{3,7,9,21}=\{1,19,25,27\}$	$E_{zn}=\{3,7,9,25\}$ $T_{3,7,9,25}=\{1,19,21,27\}$	$E_{zo}=\{3,7,9,27\}$ $T_{3,7,9,27}=\{1,19,21,25\}$	$E_{zp}=\{3,7,19,21\}$ $T_{3,7,19,21}=\{1,9,25,27\}$
$E_{zq}=\{3,7,19,25\}$ $T_{3,7,19,25}=\{1,9,21,27\}$	$E_{zr}=\{3,7,19,27\}$ $T_{3,7,19,27}=\{1,9,21,25\}$	$E_{zs}=\{3,7,21,25\}$ $T_{3,7,21,25}=\{1,9,19,27\}$	$E_{zt}=\{3,7,21,27\}$ $T_{3,7,21,27}=\{1,9,19,25\}$
$E_{zu}=\{3,7,25,27\}$ $T_{3,7,25,27}=\{1,9,19,21\}$	$E_{zv}=\{3,9,19,21\}$ $T_{3,9,19,21}=\{1,7,25,27\}$	$E_{zw}=\{3,9,19,25\}$ $T_{3,9,19,25}=\{1,7,21,27\}$	$E_{zx}=\{3,9,19,27\}$ $T_{3,9,19,27}=\{1,7,21,25\}$
$E_{zy}=\{3,9,21,25\}$ $T_{3,9,21,25}=\{1,7,19,27\}$	$E_{zz}=\{3,9,21,27\}$ $T_{3,9,21,27}=\{1,7,19,25\}$	$E_{ya}=\{3,9,25,27\}$ $T_{3,9,25,27}=\{1,7,19,21\}$	$E_{yb}=\{3,19,21,25\}$ $T_{3,19,21,25}=\{1,7,9,27\}$
$E_{yc}=\{3,19,21,27\}$ $T_{3,19,21,27}=\{1,7,9,25\}$	$E_{yd}=\{3,19,25,27\}$ $T_{3,19,25,27}=\{1,7,9,21\}$	$E_{ye}=\{3,21,25,27\}$ $T_{3,21,25,27}=\{1,7,9,19\}$	$E_{yf}=\{7,9,19,21\}$ $T_{7,9,19,21}=\{1,3,25,27\}$
$E_{yg}=\{7,9,19,25\}$ $T_{7,9,19,25}=\{1,3,21,27\}$	$E_{yh}=\{7,9,19,27\}$ $T_{7,9,19,27}=\{1,3,21,25\}$	$E_{yi}=\{7,9,21,25\}$ $T_{7,9,21,25}=\{1,3,19,27\}$	$E_{yj}=\{7,9,21,27\}$ $T_{7,9,21,27}=\{1,3,19,25\}$
$E_{yk}=\{7,9,25,27\}$ $T_{7,9,25,27}=\{1,3,19,21\}$	$E_{yl}=\{7,19,21,25\}$ $T_{7,19,21,25}=\{1,3,9,27\}$	$E_{ym}=\{7,19,21,27\}$ $T_{7,19,21,27}=\{1,3,9,25\}$	$E_{yn}=\{7,19,25,27\}$ $T_{7,19,25,27}=\{1,3,9,21\}$
$E_{yo}=\{7,21,25,27\}$	$E_{yp}=\{9,19,21,25\}$	$E_{yq}=\{9,19,21,27\}$	$E_{yr}=\{9,19,25,27\}$

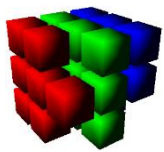
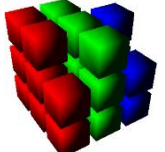
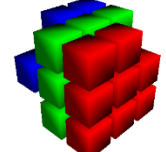
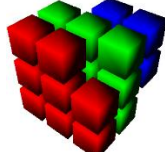
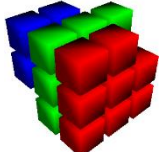
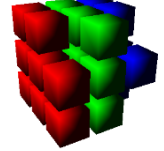
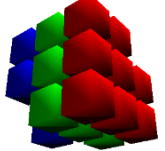
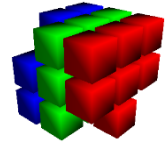
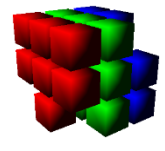
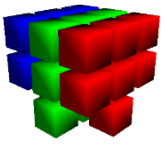
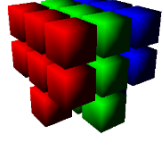
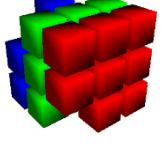
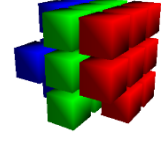
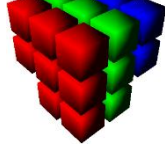
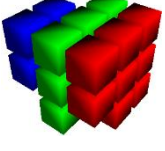
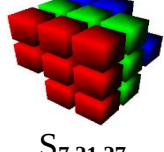
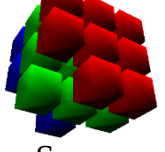
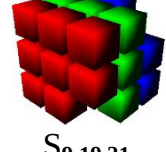
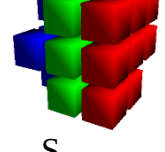
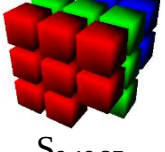
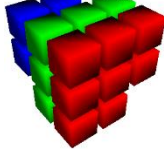
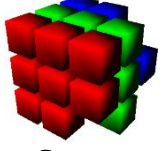
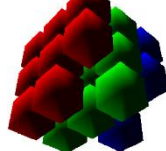
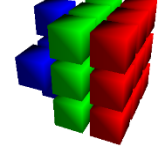
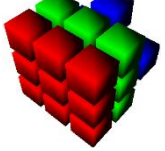
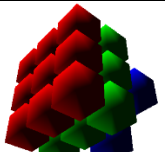
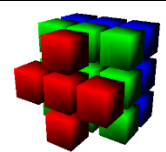
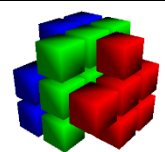
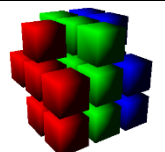
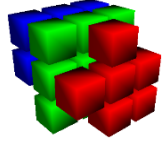
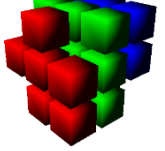
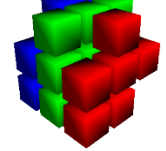
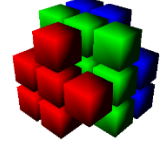
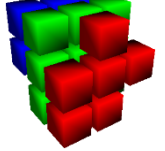
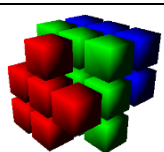
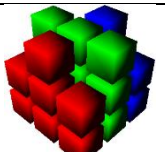
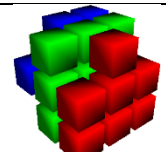
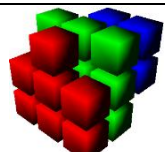
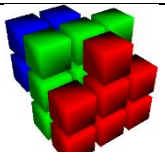
$T_{7,21,25,27}=\{1,3,9,19\}$	$T_{9,19,21,25}=\{1,3,7,27\}$	$T_{9,19,21,27}=\{1,3,7,25\}$	$T_{9,19,25,27}=\{1,3,7,21\}$
$E_{ys}=\{9,21,25,27\}$ $T_{9,21,25,27}=\{1,3,7,19\}$	$E_{yt}=\{19,21,25,27\}$ $T_{19,21,25,27}=\{1,3,7,9\}$		
Group U contains 56 combinations			
$F_a=\{1,3,7,9,19\}$ $U_{1,3,7,9,19}=\{21,25,27\}$	$F_b=\{1,3,7,9,21\}$ $U_{1,3,7,9,21}=\{19,25,27\}$	$F_c=\{1,3,7,9,25\}$ $U_{1,3,7,9,25}=\{19,21,27\}$	$F_d=\{1,3,7,9,27\}$ $U_{1,3,7,9,27}=\{19,21,25\}$
$F_e=\{1,3,7,19,21\}$ $U_{1,3,7,19,21}=\{9,25,27\}$	$F_f=\{1,3,7,19,25\}$ $U_{1,3,7,19,25}=\{9,21,27\}$	$F_g=\{1,3,7,19,27\}$ $U_{1,3,7,19,27}=\{9,21,25\}$	$F_h=\{1,3,7,21,25\}$ $U_{1,3,7,21,25}=\{9,19,7\}$
$F_i=\{1,3,7,21,27\}$ $U_{1,3,7,21,27}=\{9,19,25\}$	$F_j=\{1,3,7,25,27\}$ $U_{1,3,7,25,27}=\{9,19,21\}$	$F_k=\{1,3,9,19,21\}$ $U_{1,3,9,19,21}=\{7,25,27\}$	$F_l=\{1,3,9,19,25\}$ $U_{1,3,9,19,25}=\{7,21,27\}$
$F_m=\{1,3,9,19,27\}$ $U_{1,3,9,19,27}=\{7,21,25\}$	$F_n=\{1,3,9,21,25\}$ $U_{1,3,9,21,25}=\{7,19,27\}$	$F_o=\{1,3,9,21,27\}$ $U_{1,3,9,21,27}=\{7,19,25\}$	$F_p=\{1,3,9,25,27\}$ $U_{1,3,9,25,27}=\{7,19,21\}$
$F_q=\{1,3,19,21,25\}$ $U_{1,3,19,21,25}=\{7,9,27\}$	$F_r=\{1,3,19,21,27\}$ $U_{1,3,19,21,27}=\{7,9,25\}$	$F_s=\{1,3,19,25,27\}$ $U_{1,3,19,25,27}=\{7,9,21\}$	$F_t=\{1,3,21,25,27\}$ $U_{1,3,21,25,27}=\{7,9,19\}$
$F_u=\{1,7,9,19,21\}$ $U_{1,7,9,19,21}=\{3,25,27\}$	$F_v=\{1,7,9,19,25\}$ $U_{1,7,9,19,25}=\{3,21,27\}$	$F_w=\{1,7,9,19,27\}$ $U_{1,7,9,19,27}=\{3,21,25\}$	$F_x=\{1,7,9,19,27\}$ $U_{1,7,9,21,25}=\{3,19,27\}$
$F_y=\{1,7,9,21,27\}$ $U_{1,7,9,21,27}=\{3,19,25\}$	$F_z=\{1,7,9,25,27\}$ $U_{1,7,9,25,27}=\{3,19,21\}$	$F_{za}=\{1,7,19,21,25\}$ $U_{1,7,19,21,25}=\{3,9,27\}$	$F_{zb}=\{1,7,19,21,27\}$ $U_{1,7,19,21,27}=\{3,9,25\}$
$F_{zc}=\{1,7,19,25,27\}$ $U_{1,7,19,25,27}=\{3,9,21\}$	$F_{zd}=\{1,7,21,25,27\}$ $U_{1,7,21,25,27}=\{3,9,19\}$	$F_{ze}=\{1,9,19,21,25\}$ $U_{1,9,19,21,25}=\{3,7,27\}$	$F_{zf}=\{1,9,19,21,27\}$ $U_{1,9,19,21,27}=\{3,7,25\}$
$F_{zg}=\{1,9,19,25,27\}$ $U_{1,9,19,25,27}=\{3,7,21\}$	$F_{zh}=\{1,9,21,25,27\}$ $U_{1,9,21,25,27}=\{3,7,19\}$	$F_{zi}=\{1,19,21,25,27\}$ $U_{1,19,21,25,27}=\{3,7,9\}$	$F_{zj}=\{3,7,9,19,21\}$ $U_{3,7,9,19,21}=\{1,25,27\}$
$F_{zk}=\{3,7,9,19,25\}$ $U_{3,7,9,19,25}=\{1,21,27\}$	$F_{zl}=\{3,7,9,19,27\}$ $U_{3,7,9,19,27}=\{1,21,25\}$	$F_{zm}=\{3,7,9,21,25\}$ $U_{3,7,9,21,25}=\{1,19,27\}$	$F_{zn}=\{3,7,9,21,27\}$ $U_{3,7,9,21,27}=\{1,19,25\}$
$F_{zo}=\{3,7,9,25,27\}$ $U_{3,7,9,25,27}=\{1,19,21\}$	$F_{zp}=\{3,7,19,21,25\}$ $U_{3,7,19,21,25}=\{1,9,27\}$	$F_{zq}=\{3,7,19,21,27\}$ $U_{3,7,19,21,27}=\{1,9,25\}$	$F_{zr}=\{3,7,19,25,27\}$ $U_{3,7,19,25,27}=\{1,9,21\}$
$F_{zs}=\{3,7,21,25,27\}$ $U_{3,7,21,25,27}=\{1,9,19\}$	$F_{zt}=\{3,9,19,21,25\}$ $U_{3,9,19,21,25}=\{1,7,27\}$	$F_{zu}=\{3,9,19,21,2\}$ $U_{3,9,19,21,27}=\{1,7,25\}$	$F_{zv}=\{3,9,19,25,27\}$ $F_{3,9,19,25,27}=\{1,7,21\}$
$F_{zw}=\{3,9,21,25,27\}$ $U_{3,9,21,25,27}=\{1,7,19\}$	$F_{zx}=\{3,19,21,25,27\}$ $U_{3,19,21,25,27}=\{1,7,9\}$	$F_{zy}=\{7,9,19,21,25\}$ $U_{7,9,19,21,25}=\{1,3,27\}$	$F_{zz}=\{7,9,19,21,27\}$ $U_{7,9,19,21,27}=\{1,3,25\}$
$F_{ya}=\{7,9,19,25,27\}$	$F_{yb}=\{7,9,21,25,27\}$	$F_{yc}=\{7,19,21,25,27\}$	$F_{yd}=\{9,19,21,25,27\}$

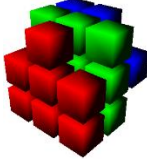
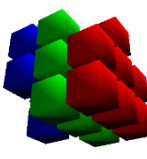
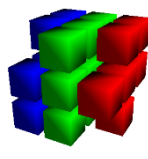
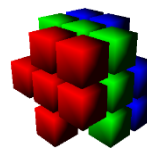
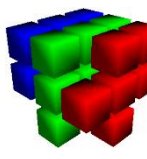
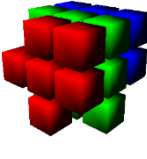
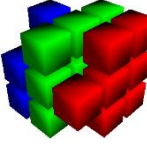
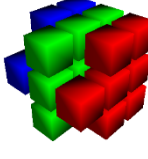
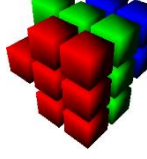
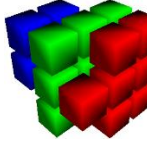
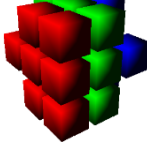
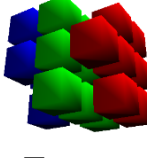
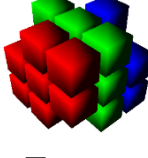
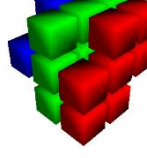
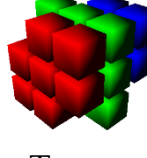
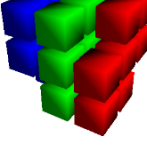
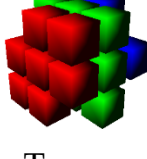
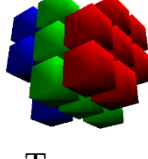
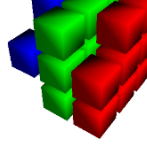
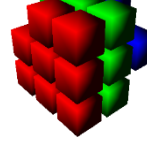
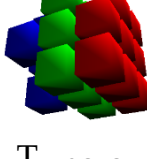
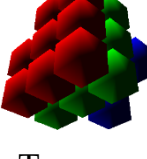
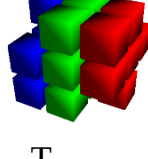
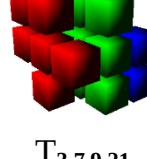
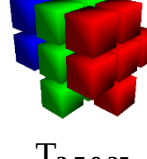
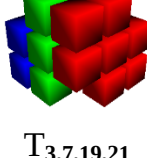
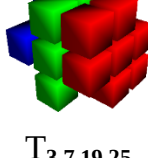
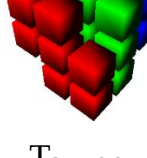
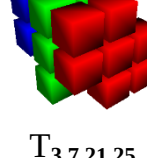
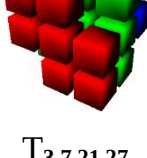
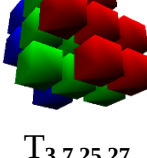
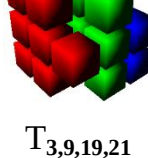
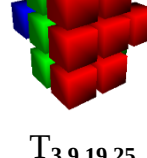
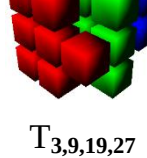
$U_{7,9,19,25,27}=\{1,3,21\}$	$U_{7,9,21,25,27}=\{1,3,19\}$	$U_{7,19,21,25,27}=\{1,3,9\}$	$U_{9,19,21,25,27}=\{1,3,7\}$
Group V contains 28 combinations			
$F_a=\{1,3,7,9,19,21\}$ $V_{1,3,7,9,19,21}=\{25,27\}$	$F_b=\{1,3,7,9,19,25\}$ $V_{1,3,7,9,19,25}=\{21,27\}$	$F_c=\{1,3,7,9,19,27\}$ $V_{1,3,7,9,19,27}=\{21,25\}$	$F_d=\{1,3,7,9,21,25\}$ $V_{1,3,7,9,21,25}=\{19,27\}$
$F_e=\{1,3,7,9,21,27\}$ $V_{1,3,7,9,21,27}=\{19,25\}$	$F_f=\{1,3,7,9,25,27\}$ $V_{1,3,7,9,25,27}=\{19,21\}$	$F_g=\{1,3,7,19,21,25\}$ $V_{1,3,7,19,21,25}=\{9,27\}$	$F_h=\{1,3,7,9,21,27\}$ $V_{1,3,7,19,21,27}=\{9,25\}$
$F_i=\{1,3,7,19,25,27\}$ $V_{1,3,7,19,25,27}=\{9,21\}$	$F_j=\{1,3,7,21,25,27\}$ $V_{1,3,7,21,25,27}=\{9,19\}$	$F_k=\{1,3,9,19,21,25\}$ $V_{1,3,9,19,21,25}=\{7,27\}$	$F_l=\{1,3,9,19,21,27\}$ $V_{1,3,9,19,21,27}=\{7,25\}$
$F_m=\{1,3,9,19,25,27\}$ $V_{1,3,9,19,25,27}=\{7,21\}$	$F_n=\{1,3,9,21,25,27\}$ $V_{1,3,9,21,25,27}=\{7,19\}$	$F_o=\{1,3,19,21,25,27\}$ $V_{1,3,19,21,25,27}=\{7,9\}$	$F_p=\{1,7,9,19,21,25\}$ $V_{1,7,9,19,21,25}=\{3,27\}$
$F_q=\{1,7,9,19,21,27\}$ $V_{1,7,9,19,21,27}=\{3,25\}$	$F_r=\{1,7,9,19,25,27\}$ $V_{1,7,9,19,25,27}=\{3,21\}$	$F_s=\{1,7,9,21,25,27\}$ $V_{1,7,9,21,25,27}=\{3,19\}$	$F_t=\{1,7,19,21,25,27\}$ $V_{1,7,19,21,25,27}=\{3,9\}$
$F_u=\{1,9,19,21,25,27\}$ $V_{1,9,19,21,25,27}=\{3,7\}$	$F_v=\{3,7,9,19,21,25\}$ $V_{3,7,9,19,21,25}=\{1,27\}$	$F_w=\{1,3,9,19,25,27\}$ $V_{3,7,9,19,21,27}=\{1,25\}$	$F_x=\{3,7,9,19,25,27\}$ $V_{3,7,9,19,25,27}=\{1,21\}$
$F_y=\{3,7,9,21,25,27\}$ $V_{3,7,9,21,25,27}=\{1,19\}$	$F_z=\{3,7,19,21,25,27\}$ $V_{3,7,19,21,25,27}=\{1,9\}$	$F_{za}=\{3,9,19,21,25,27\}$ $V_{3,9,19,21,25,27}=\{1,7\}$	$F_{zb}=\{7,9,19,21,25,27\}$ $V_{7,9,19,21,25,27}=\{1,3\}$
Group W Contains 8 combinations			
$G_a=\{1,3,7,9,19,21,25\}$ $W_{1,3,7,9,19,21,25}=\{27\}$	$G_b=\{1,3,7,9,19,21,27\}$ $W_{1,3,7,9,19,21,27}=\{25\}$	$G_c=\{1,3,7,9,19,25,27\}$ $W_{1,3,7,9,19,25,27}=\{21\}$	$G_d=\{1,3,7,9,21,25,27\}$ $W_{1,3,7,9,21,25,27}=\{19\}$
$G_e=\{1,3,7,19,21,25,27\}$ $W_{1,3,7,19,21,25,27}=\{9\}$	$G_f=\{1,3,9,19,21,25,27\}$ $W_{1,3,9,19,21,25,27}=\{7\}$	$G_g=\{1,7,9,19,21,25,27\}$ $W_{1,7,9,19,21,25,27}=\{3\}$	$G_h=\{3,7,9,19,21,25,27\}$ $W_{3,7,9,19,21,25,27}=\{1\}$
Group X 1 combination			
$H_a=\{1,3,7,9,19,21,25,27\}$ $X_{1,3,7,9,19,21,25,27}=\{ \}$			

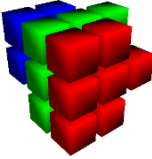
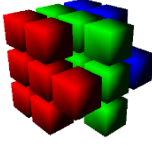
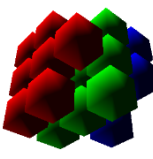
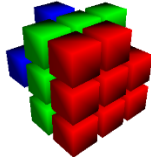
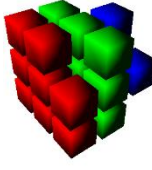
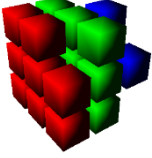
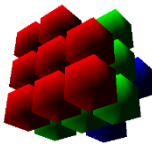
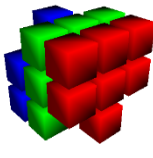
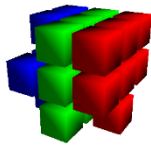
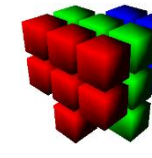
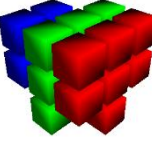
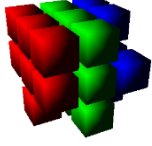
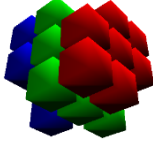
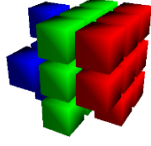
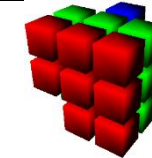
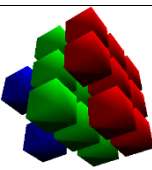
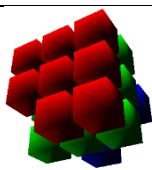
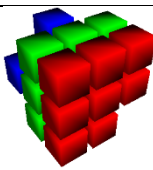
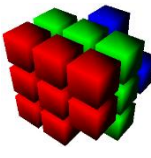
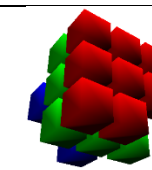
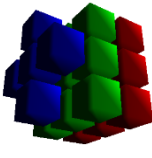
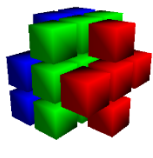
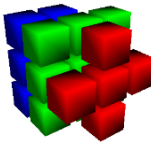
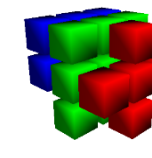
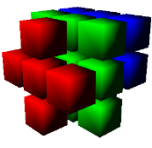
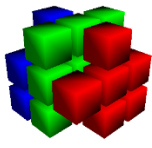
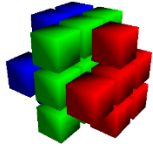
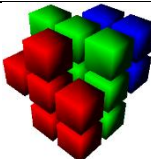
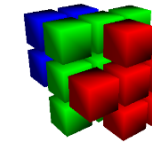
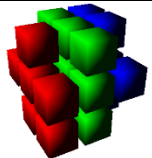
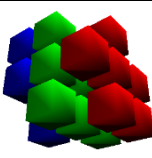
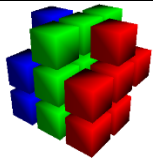
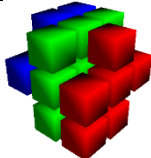
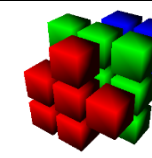
The visualization of 256 convex polyhedrons which are listed in the above table is shown in **Fig. 5.6**.

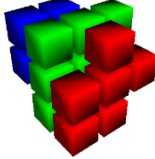
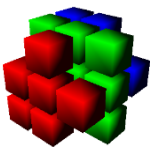
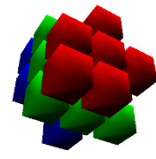
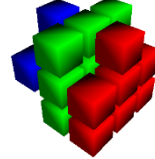
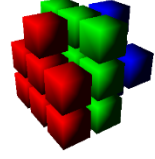
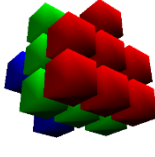
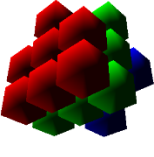
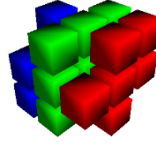
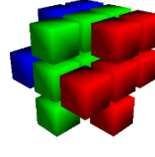
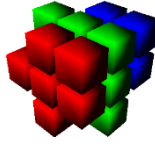
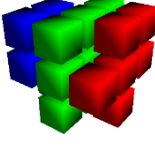
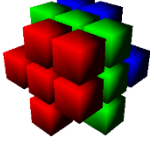
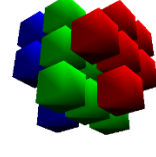
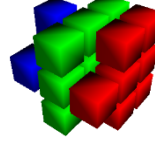
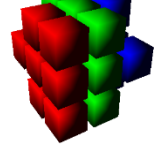
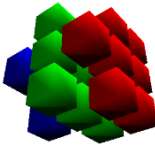
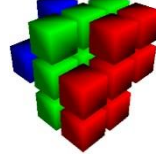
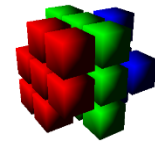
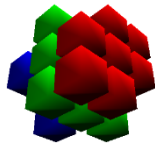
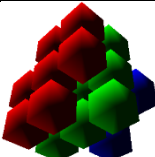
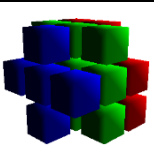
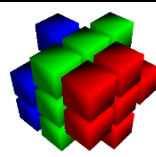
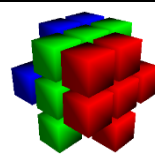
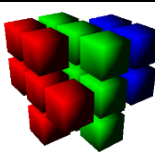
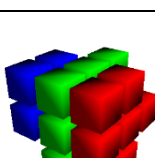
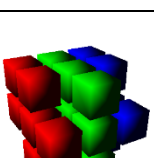
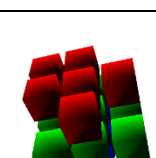
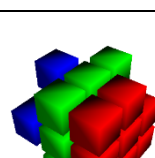
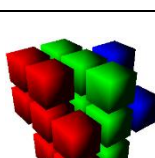
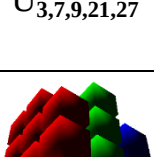
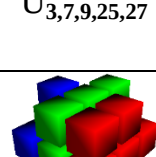
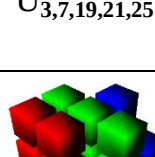
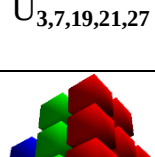
 P	 Q₁	 Q₃	 Q₇	 Q₉
 Q₁₉	 Q₂₁	 Q₂₅	 Q₂₇	 R_{1,3}
 R_{1,7}	 R_{1,9}	 R_{1,19}	 R_{1,21}	 R_{1,25}
 R_{1,27}	 R_{3,7}	 R_{3,9}	 R_{3,19}	 R_{3,21}
 R_{3,7}	 R_{3,9}	 R_{3,19}	 R_{3,21}	 R_{3,25}
 R_{3,27}	 R_{7,9}	 R_{7,19}	 R_{7,21}	 R_{7,25}
 R_{7,27}	 R_{9,19}	 R_{9,21}	 R_{9,25}	 R_{9,27}

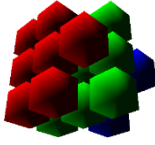
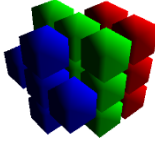
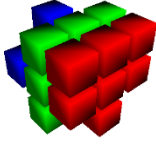
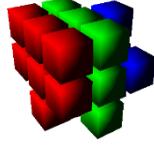
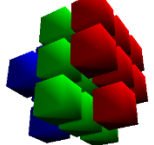
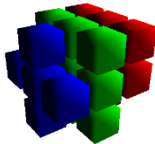
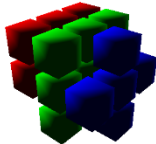
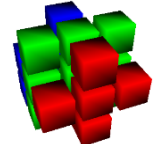
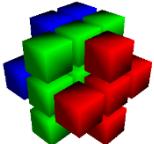
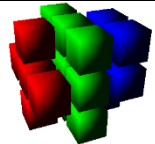
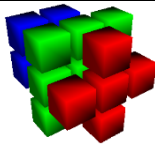
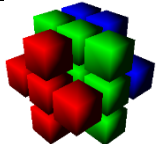
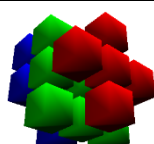
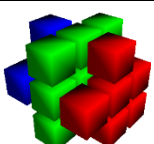
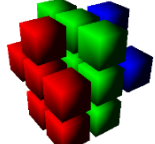
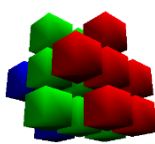
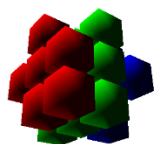
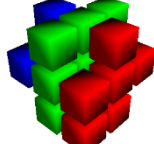
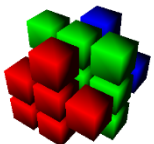
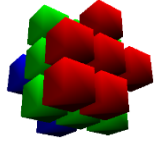
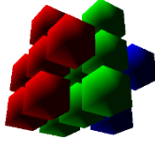
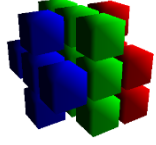
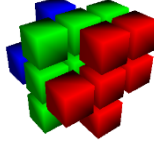
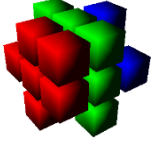
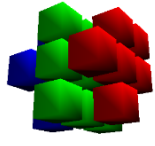
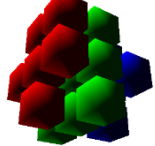
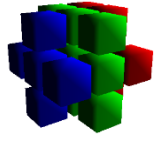
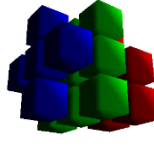
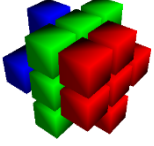
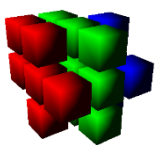
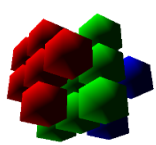
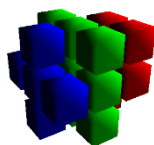
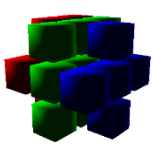
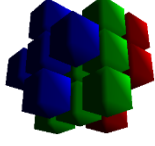
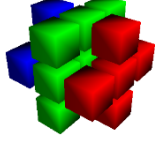
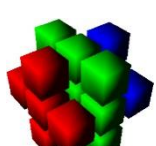
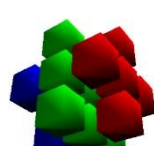
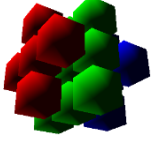
 $R_{19,21}$	 $R_{19,25}$	 $R_{19,27}$	 $R_{21,25}$	 $R_{21,27}$
 $R_{25,27}$	 $S_{1,3,7}$	 $S_{1,3,9}$	 $S_{1,3,19}$	 $S_{1,3,21}$
 $S_{1,3,25}$	 $S_{1,3,27}$	 $S_{1,7,9}$	 $S_{1,7,19}$	 $S_{1,7,21}$
 $S_{1,7,25}$	 $S_{1,7,27}$	 $S_{1,9,19}$	 $S_{1,9,21}$	 $S_{1,9,25}$
 $S_{1,9,27}$	 $S_{1,19,21}$	 $S_{1,19,25}$	 $S_{1,19,27}$	 $S_{1,21,25}$
 $S_{1,21,27}$	 $S_{1,25,27}$	 $S_{3,7,9}$	 $S_{3,7,19}$	 $S_{3,7,21}$
 $S_{3,7,25}$	 $S_{3,7,27}$	 $S_{3,9,19}$	 $S_{3,9,21}$	 $S_{3,9,25}$

 $S_{3,9,27}$	 $S_{3,19,21}$	 $S_{3,19,25}$	 $S_{3,19,27}$	 $S_{3,21,25}$
 $S_{3,21,27}$	 $S_{3,25,27}$	 $S_{7,9,19}$	 $S_{7,9,21}$	 $S_{7,9,25}$
 $S_{7,9,27}$	 $S_{7,19,21}$	 $S_{7,19,25}$	 $S_{7,19,27}$	 $S_{7,21,25}$
 $S_{7,21,27}$	 $S_{7,25,27}$	 $S_{9,19,21}$	 $S_{9,19,25}$	 $S_{9,19,27}$
 $S_{9,21,25}$	 $S_{9,21,27}$	 $S_{9,25,27}$	 $S_{19,21,25}$	 $S_{19,21,27}$
 $S_{19,25,27}$	 $S_{21,25,27}$	 $T_{1,3,7,9}$	 $T_{1,3,7,19}$	 $T_{1,3,7,21}$
 $T_{1,3,7,25}$	 $T_{1,3,7,27}$	 $T_{1,3,9,19}$	 $T_{1,3,9,21}$	 $T_{1,3,9,25}$
 $T_{1,3,9,27}$	 $T_{1,3,9,27}$	 $T_{1,3,9,27}$	 $T_{1,3,9,27}$	 $T_{1,3,9,27}$

$T_{1,3,9,27}$	$T_{1,3,19,21}$	$T_{1,3,19,25}$	$T_{1,3,19,27}$	$T_{1,3,21,25}$
				
$T_{1,3,21,27}$	$T_{1,3,25,27}$	$T_{1,7,9,19}$	$T_{1,7,9,21}$	$T_{1,7,9,25}$
				
$T_{1,7,9,27}$	$T_{1,7,19,21}$	$T_{1,7,19,25}$	$T_{1,7,19,27}$	$T_{1,7,21,25}$
				
$T_{1,7,21,27}$	$T_{1,7,25,27}$	$T_{1,9,19,21}$	$T_{1,9,19,25}$	$T_{1,9,19,27}$
				
$T_{1,9,21,25}$	$T_{1,9,21,27}$	$T_{1,9,25,27}$	$T_{1,19,21,25}$	$T_{1,19,21,27}$
				
$T_{1,19,25,27}$	$T_{1,21,25,27}$	$T_{3,7,9,19}$	$T_{3,7,9,21}$	$T_{3,7,9,25}$
				
$T_{3,7,9,27}$	$T_{3,7,19,21}$	$T_{3,7,19,25}$	$T_{3,7,19,27}$	$T_{3,7,21,25}$
				
$T_{3,7,21,27}$	$T_{3,7,25,27}$	$T_{3,9,19,21}$	$T_{3,9,19,25}$	$T_{3,9,19,27}$

 $T_{3,9,21,25}$	 $T_{3,9,21,27}$	 $T_{3,9,25,27}$	 $T_{3,19,21,25}$	 $T_{3,19,21,27}$
 $T_{3,19,25,27}$	 $T_{3,21,25,27}$	 $T_{7,9,19,21}$	 $T_{7,9,19,25}$	 $T_{7,9,19,27}$
 $T_{7,9,21,25}$	 $T_{7,9,21,27}$	 $T_{7,9,25,27}$	 $T_{7,19,21,25}$	 $T_{7,19,21,27}$
 $T_{7,19,25,27}$	 $T_{7,21,25,27}$	 $T_{9,19,21,25}$	 $T_{9,19,21,27}$	 $T_{9,19,25,27}$
 $T_{9,21,25,27}$	 $T_{19,21,25,27}$	 $U_{1,3,7,9,19}$	 $U_{1,3,7,9,21}$	 $U_{1,3,7,9,25}$
 $U_{1,3,7,9,27}$	 $U_{1,3,7,19,21}$	 $U_{1,3,7,19,25}$	 $U_{1,3,7,19,27}$	 $U_{1,3,7,21,25}$
 $U_{1,3,7,21,27}$	 $U_{1,3,7,25,27}$	 $U_{1,3,9,19,21}$	 $U_{1,3,9,19,25}$	 $U_{1,3,9,19,27}$

 $U_{1,3,9,21,25}$	 $U_{1,3,9,21,27}$	 $U_{1,3,9,25,27}$	 $U_{1,3,19,21,25}$	 $U_{1,3,19,21,27}$
 $U_{1,3,19,25,27}$	 $U_{1,3,21,25,27}$	 $U_{1,7,9,19,21}$	 $U_{1,7,9,19,25}$	 $U_{1,7,9,19,27}$
 $U_{1,7,9,21,25}$	 $U_{1,7,9,21,27}$	 $U_{1,7,9,25,27}$	 $U_{1,7,19,21,25}$	 $U_{1,7,19,21,27}$
 $U_{1,7,19,25,27}$	 $U_{1,7,21,25,27}$	 $U_{1,9,19,21,25}$	 $U_{1,9,19,21,27}$	 $U_{1,9,19,25,27}$
 $U_{1,9,21,25,27}$	 $U_{1,19,21,25,27}$	 $U_{3,7,9,19,21}$	 $U_{3,7,9,19,25}$	 $U_{3,7,9,19,27}$
 $U_{3,7,9,21,25}$	 $U_{3,7,9,21,27}$	 $U_{3,7,9,25,27}$	 $U_{3,7,19,21,25}$	 $U_{3,7,19,21,27}$
 $U_{3,7,19,25,27}$	 $U_{3,7,21,25,27}$	 $U_{3,9,19,21,25}$	 $U_{3,9,19,21,27}$	 $U_{3,9,19,25,27}$

 $U_{3,9,21,25,27}$	 $U_{3,19,21,25,27}$	 $U_{7,9,19,21,25}$	 $U_{7,9,19,21,27}$	 $U_{7,9,19,25,27}$
 $U_{7,9,21,25,27}$	 $U_{7,19,21,25,27}$	 $U_{9,19,21,25,27}$	 $V_{1,3,7,9,19,21}$	 $V_{1,3,7,9,19,25}$
 $V_{1,3,7,9,19,27}$	 $V_{1,3,7,9,21,25}$	 $V_{1,3,7,9,21,27}$	 $V_{1,3,7,9,25,27}$	 $V_{1,3,7,19,21,25}$
 $V_{1,3,7,19,21,27}$	 $V_{1,3,7,19,25,27}$	 $V_{1,3,7,21,25,27}$	 $V_{1,3,9,19,21,25}$	 $V_{1,3,9,19,21,27}$
 $V_{1,3,9,19,25,27}$	 $V_{1,3,9,21,25,27}$	 $V_{1,3,19,21,25,27}$	 $V_{1,7,9,19,21,25}$	 $V_{1,7,9,19,21,27}$
 $V_{1,7,9,19,25,27}$	 $V_{1,7,9,21,25,27}$	 $V_{1,7,19,21,25,27}$	 $V_{1,9,19,21,25,27}$	 $V_{3,7,9,19,21,25}$
 $V_{3,7,9,19,21,27}$	 $V_{3,7,9,19,25,27}$	 $V_{3,7,9,21,25,27}$	 $V_{3,7,19,21,25,27}$	 $V_{3,9,19,21,25,27}$
 $V_{7,9,19,21,25,27}$	 $W_{1,3,7,9,19,21,25}$	 $W_{1,3,7,9,19,21,25,27}$	 $W_{1,3,7,9,19,21,25,27}$	 $W_{1,3,7,9,19,21,25,27}$

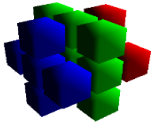
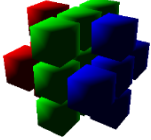
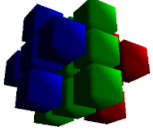
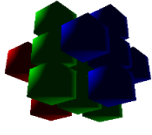
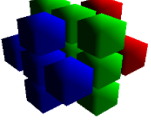
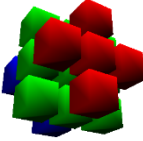
		$W_{1,3,7,9,19,21,27}$	$W_{1,3,7,9,19,25,27}$	
 $W_{1,3,7,19,21,25,27}$	 $W_{1,3,9,19,21,25,27}$	 $W_{1,7,9,19,21,25,27}$	 $W_{3,7,9,19,21,25,27}$	 $X_{1,3,7,9,19,21,25,27}$ Back view
 $X_{1,3,7,9,19,21,25,27}$ Front View				

Fig. 5.6: Three Dimensional Visualizations of Convex Polyhedrons

5.2.2 Feature Extraction

Extracting the features from an image is an important step in object recognition. In the proposed method we used picture description language for the representation of the features. The image is traced pixel by pixel which starts from pixel position (1,1,1) to find the initial foreground pixel (x_i, y_i, z_i) with the intensity value 1. This will identify the initial point of the first component present in the image. The proposed algorithm is written in such a way that it will find the next neighborhood pixel of the current pixel in all the preferred direction. Preference will be generally given to the right side of the pixel (from the current pixel). These 26 directions are already represented in **Fig. 5.3**. The naming conventions of the preferred direction are: Right(R), Down Right (DR), Down (D), Down Left (DL), Left (L), Up left (UL), Up (U), and Up Right (UR), Back(B), Back Right(BR), Back Down Right (BDR), Back Down (BD), Back Down Left (BDL), Back Left (BL), Back-Up left (BUL), Back-Up (BU), and Back-Up Right (BUR), Front(F), Front Right(FR), Front Down Right (FDR), Front Down (FD), Front Down Left (FDL), Front Left (FL), Front Up left (FUL), Front Up (FU), and Front Up Right (FUR).

In this method, the previously recognized direction has given priority compared to other preferred directions. The tracing of each pixel will continue until the algorithm will not find any connected neighbor pixel to the current position (x, y) and will

remove the pixel which is already traced. If there is no neighborhood pixel available then the algorithm will display the knowledge vector of that component and again algorithm will continue until it will not trace all the components which are present in the image. If all the components are traced then the final knowledge vector will be displayed. This knowledge vector is nothing but a feature vector that consists of the direction code and length code of each component and this knowledge vector will give the information that some objects are present in the image.

Algorithm 5. 1: Creating the Knowledge Vector from the Given Image.

Input: Given Image Empty loader Output: Knowledge vector
--

```

Begin
Initialize
    Dimension of the image
i,j,k=dim
length = []
    direction = []
arr= []
loader=[]
    image=[]
    arr2=[]
    arr_i=[]
    arr_j=[]
    arr_k=[]
Assign the dimension values to the
variable dim.
#Create an empty container of the same
size.
Create a 3D array (img) of the same size
as the dimension variable and initialize it
as zero.
Give the path of the dataset and read the
images from the dataset and store the
value in array im.
    for k in range(dim):
        for i in range(dim):
            for j in range(dim):
                if im[i][j][k]!=0
                    then                                assign
img[m][i+1][j+1]==255
store_value function(i, j, k)
    append
    i value in arr_i
    j value in arr_j
    k value in arr_k
Traversing function(x, y, z)
    Initialize the x, y, z value to i, j, k
    Initialize the img array as 0
    Call Store_value (i, j, k)
    Initialize the loader as [0 0]
While (image[k, i, j]!=255)
#Right direction
    if img(k, i, j+1)==255:
        if R\ is available in loader then
            image [k, i, j+1] = 0
            j=j+1
            call store(i, j, k)
            Loader[1]=loader[1]+1
        else:
            append the direction at loader[0]
            append the length with {\} at loader
[1]
            clear the loader

```

```

        append R\
        append 0 to the loader
#Down right direction
    else if img(k, i+1, j+1)==255:
        if DR\ is available in loader then
            image [k, i, j+1] = 0
            j=j+1
            call store(i, j, k)
            Loader[1]=loader[1]+1
        else:
            append the direction at loader[0]
            append the length with {\} at loader
[1]
            clear the loader
            append DR\
            append 0 to the loader
#Down direction
    else if img(k, i+1, j)==255:
        if D\ is available in loader then
            image [k, i, j+1] = 0
            j=j+1
            call store(i, j, k)
            Loader[1]=loader[1]+1
        else:
            append the direction at loader[0]
            append the length with {\} at loader
[1]
            clear the loader
            append D\
            append 0 to the loader
#Down left direction
    else if img(k, i+1, j-1)==255:
        if DL\ is available in loader then
            image [k, i, j+1] = 0
            j=j+1
            call store(i, j, k)
            Loader[1]=loader[1]+1
        else:
            append the direction at loader[0]
            append the length with {\} at loader
[1]
            clear the loader
            append DL\
            append 0 to the loader
#Left direction
    else if img(k, i, j-1)==255:
        if L\ is available in loader then
            image [k, i, j+1] = 0
            j=j+1
            call store(i, j, k)
            Loader[1]=loader[1]+1

```

```

else:
    append the direction at loader[0]
    append the length with {} at loader
[1]
    clear the loader
    append L at loader
    append 0 to the loader
#up left direction
else if img(k, i-1, j-1)==255:
    if UL is available in loader then
        image [k, i, j+1] = 0
        j=j+1
        call store(i, j, k)
        Loader[1]=loader[1]+1
    else:
        append the direction at loader[0]
        append the length with {} at loader
[1]
    clear the loader
    append UL at loader
    append 0 to the loader
#Up direction
else if img(k, i-1, j)==255:
    if U is available in loader then
        image [k, i, j+1] = 0
        j=j+1
        call store(i, j, k)
        Loader[1]=loader[1]+1
    else:
        append the direction at loader[0]
        append the length with {} at loader
[1]
    clear the loader
    append U at loader
    append 0 to the loader
#Up Right direction
else if img(k, i-1, j+1)==255:
    if UR is available in loader then
        image [k, i, j+1] = 0
        j=j+1
        call store(i, j, k)
        Loader[1]=loader[1]+1
    else:
        append the direction at loader[0]
        append the length with {} at loader
[1]
    clear the loader
    append UR at loader
    append 0 to the loader
#Back direction
else if img(k+1, i, j)==255:
    if B is available in loader then

```

```

        image [k, i, j+1] = 0
        j=j+1
        call store(i, j, k)
        Loader[1]=loader[1]+1
    else:
        append the direction at loader[0]
        append the length with {} at loader
[1]
    clear the loader
    append B at loader
    append 0 to the loader
#Back right direction
else if img(k+1, i, j+1)==255:
    if BR is available in loader then
        image [k, i, j+1] = 0
        j=j+1
        call store(i, j, k)
        Loader[1]=loader[1]+1
    else:
        append the direction at loader[0]
        append the length with {} at loader
[1]
    clear the loader
    append BR at loader
    append 0 to the loader
#Back down right direction
else if img(k+1, i+1, j+1)==255:
    if BDR is available in loader then
        image [k, i, j+1] = 0
        j=j+1
        call store(i, j, k)
        Loader[1]=loader[1]+1
    else:
        append the direction at loader[0]
        append the length with {} at loader
[1]
    clear the loader
    append BDR at loader
    append 0 to the loader
# Back down direction
else if img(k+1, i+1, j)==255:
    if BD is available in loader then
        image [k, i, j+1] = 0
        j=j+1
        call store(i, j, k)
        Loader[1]=loader[1]+1
    else:
        append the direction at loader[0]
        append the length with {} at loader
[1]
    clear the loader
    append BD at loader

```

```

    append 0 to the loader
# Back down left direction
    else if img(k+1, i+1, j-1)==255:
        if BDL\□ is available in loader then
            image [k, i, j+1] = 0
            j=j+1
            call store(i, j, k)
            Loader[1]=loader[1]+1
        else:
            append the direction at loader[0]
            append the length with {}□ at loader
[1]
            clear the loader
            append BDL\□
            append 0 to the loader
# Back left direction
    else if img(k+1, i, j-1)==255:
        if BL\□ is available in loader then
            image [k, i, j+1] = 0
            j=j+1
            call store(i, j, k)
            Loader[1]=loader[1]+1
        else:
            append the direction at loader[0]
            append the length with {}□ at loader
[1]
            clear the loader
            append BL\□
            append 0 to the loader
# Back up left direction
    else if img(k+1, i-1, j-1)==255:
        if BUL\□ is available in loader then
            image [k, i, j+1] = 0
            j=j+1
            call store(i, j, k)
            Loader[1]=loader[1]+1
        else:
            append the direction at loader[0]
            append the length with {}□ at loader
[1]
            clear the loader
            append BUL\□
            append 0 to the loader
# Back up direction
    else if img(k+1, i-1, j)==255:
        if BU\□ is available in loader then
            image [k, i, j+1] = 0
            j=j+1
            call store(i, j, k)
            Loader[1]=loader[1]+1
        else:
            append the direction at loader[0]

```

```

    append the length with {}□ at loader
[1]
    clear the loader
    append BUR\□
    append 0 to the loader
# Back up right direction
    else if img(k+1, i-1, j+1)==255:
        if BUR\□ is available in loader then
            image [k, i, j+1] = 0
            j=j+1
            call store(i, j, k)
            Loader[1]=loader[1]+1
        else:
            append the direction at loader[0]
            append the length with {}□ at loader
[1]
            clear the loader
            append BUR\□
            append 0 to the loader
# Forward direction
    else if img(k-1, i, j)==255:
        if F\□ is available in loader then
            image [k, i, j+1] = 0
            j=j+1
            call store(i, j, k)
            Loader[1]=loader[1]+1
        else:
            append the direction at loader[0]
            append the length with {}□ at loader
[1]
            clear the loader
            append F\□
            append 0 to the loader
# Forward Right direction
    else if img(k-1, i, j+1)==255:
        if FR\□ is available in loader then
            image [k, i, j+1] = 0
            j=j+1
            call store(i, j, k)
            Loader[1]=loader[1]+1
        else:
            append the direction at loader[0]
            append the length with {}□ at loader
[1]
            clear the loader
            append FR\□
            append 0 to the loader
#Forward down right direction
    else if img(k-1, i+1, j+1)==255:
        if FDR\□ is available in loader then
            image [k, i, j+1] = 0
            j=j+1

```

```

call store(i, j, k)
Loader[1]=loader[1]+1
else:
append the direction at loader[0]
append the length with F\ at loader
[1]
clear the loader
append FDR\
append 0 to the loader
# Forward down direction
else if img(k-1, i+1, j)==255:
if FD\ is available in loader then
image [k, i, j+1] = 0
j=j+1
call store(i, j, k)
Loader[1]=loader[1]+1
else:
append the direction at loader[0]
append the length with F\ at loader
[1]
clear the loader
append FD\
append 0 to the loader
# Forward down left direction
else if img(k-1, i+1, j-1)==255:
if FDL\ is available in loader then
image [k, i, j+1] = 0
j=j+1
call store(i, j, k)
Loader[1]=loader[1]+1
else:
append the direction at loader[0]
append the length with F\ at loader
[1]
clear the loader
append FDL\
append 0 to the loader
# Forward left direction
else if img(k-1, i, j-1)==255:
if FL\ is available in loader then
image [k, i, j+1] = 0
j=j+1
call store(i, j, k)
Loader[1]=loader[1]+1
else:
append the direction at loader[0]
append the length with F\ at loader
[1]
clear the loader
append FL\
append 0 to the loader
# Forward up left direction

```

```

else if img(k-1, i-1, j-1)==255:
if FUL\ is available in loader then
image [k, i, j+1] = 0
j=j+1
call store(i, j, k)
Loader[1]=loader[1]+1
else:
append the direction at loader[0]
append the length with F\ at loader
[1]
clear the loader
append FUL\
append 0 to the loader
# Forward up direction
else if img(k-1, i-1, j)==255:
if FU\ is available in loader then
image [k, i, j+1] = 0
j=j+1
call store(i, j, k)
Loader[1]=loader[1]+1
else:
append the direction at loader[0]
append the length with F\ at loader
[1]
clear the loader
append FU\
append 0 to the loader
# Forward up Right direction
else if img(k-1, i-1, j+1)==255:
if FUR\ is available in loader then
image [k, i, j+1] = 0
j=j+1
call store(i, j, k)
Loader[1]=loader[1]+1
else:
append the direction at loader[0]
append the length with F\ at loader
[1]
clear the loader
append FUR\
append 0 to the loader
# Back down left direction
else if img(k+1, i+1, j-1)==255:
if BDL\ is available in loader then
image [k, i, j+1] = 0
j=j+1
call store(i, j, k)
Loader[1]=loader[1]+1
else:
append the direction at loader[0]
append the length with F\ at loader

```

```

[1]
    clear the loader
    append BDL\□
    append 0 to the loader
else
    Break
    append the direction at loader[0])
append the length with {}\□ at loader [1]
clear the loader
pop up the direction
pop up the length
add the direction to the arr
append "*" to arr
add the length to arr
Append the i, j, k value in this format
<{},{},{}>
Clear the direction array
Clear the length array

```

```

for k in range (dim):
    for i in range (dim):
        for j in range(dim):
            if img[k][i][j]=255:
                Then append < {}, {}, {}> in
arr call traversing function by
                passing the value of i, j and
                k. (traversing(i,j,k))
            Initialize str=""
        for element in arr:
            str=str+ element
        open a text file and write the content of
        str into the text file
        Close the text file.
    end

```

The above algorithm is applied on a simple cube and knowledge vector is extracted and it is shown below:

```

< 1,1,1 > R□D□L□U□B□U□B□R□D□L□U□
    * 99□99□99□98□1□98□99□99□99□98 < 2,1,100 >
    < 1,100,2 > B□ * 97□ < 1,100,99 > < 100,1,2 > B□ * 97□
    < 100,1,99 > < 100,100,2 > B□ * 97□ < 100,100,99 >

```

The part before the * symbol is called Direction Code (DC) and the following part is called Length Code (LC). To make the model user friendly GUI is created in which user has to select the dataset folder. Once the folder is selected then user has to click on the 'Create knowledge vector' then knowledge vector will created and it display on the screen. User can save the knowledge vector for reconstruction of the original image. **Fig. 5.7** shows the same illustration.

5.2.3 Reconstruction of Original Three-Dimensional Image using Knowledge Vector

In the previous step, the knowledge vector of the image is identified which is nothing but features of the image. In this step, the feature vector is considered as an input and a novel algorithm is written which converts the knowledge vector into wireframe model. Moreover, the reconstruction algorithm is explained in detail:

Assign a[2] value to **the Length** array.
Assign a[3] value to the **End** array.
Call **Printer Function** bypassing Direction, Length, and **Start** Parameter.
Display the 3D Image on Screen Using graph plot.

Printer function(Direction, Length, Start):
Split the **Start** array on basis of “,” and store the resultant array in **the Start** array.
Split the **Length** array on basis of “,” and store the resultant array in **the Length** array.
Split the **Direction** array on basis of “,” and store the resultant array in **the Direction** array.
Store the Start [0] value in variable i.
Store the Start [1] value in variable j.
Store the Start [2] value in variable k.
Now append the value of i, j, k in p, q, r.
Display the value of **Length** array and **Direction** array.
for m in range(Length(Direction)-1):
Calculate the Length[m] and store the result in variable L.
#Right Direction
If Direction[m]==”R”
z=j+L
while(j<z):
Increment j value by 1.
append the value of j, i, k in p, q, r.
#Down Direction
Else If Direction[m]==”D”
z=i+L
while(i<z):
Increment i value by 1.
Append the value of j, i, k in p, q, r.
#Left Direction
Else If Direction[m]==”L”
z=j-L
while (j>z):
Decrement j value by 1.
Append the value of j, i, k in p, q, r.
#up Direction
Else If Direction[m]==”U”
z=i-L
while(i>z):
Decrement i value by 1.
Append the value of j, i, k in p, q, r.
#Down right Direction
Else If Direction[m]==”DR”
z=j+L
while(j<z):
Increment i & j value by 1.
Append the value of j, i, k in p, q, r.
#Down Left Direction

Else If Direction[m]==”DL”
z=i+L
while(i<z):
Increment i value by 1.
Decrement j value by 1.
Append the value of j, i, k in p, q, r.
#up Left Direction
Else If Direction[m]==”UL”
z=j-L
while(j>z):
Decrement i & j value by 1.
Append the value of j, i, k in p, q, r.
#up right Direction
Else If Direction[m]==”UR”
z=i-L
while(i>z):
Increment j value by 1.
Decrement i value by 1.
Append the value of j, i, k in p, q, r.
#Back Direction
Else If Direction[m]==”B”
z=k+L
while(k<z):
Increment k value by 1.
Append the value of j, i, k in p, q, r.
#Back right Direction
Else If Direction[m]==”BR”
z=j+L
while(j<z):
Increment j and k value by 1.
Append the value of j, i, k in p, q, r.
#Back down Direction
Else If Direction[m]==”BD”
z=i+L
while(i<z):
Increment i and k value by 1.
Append the value of j, i, k in p, q, r.
#Back Left Direction
Else If Direction[m]==”BL”
z=j-L
while(j>z):
Increment k value by 1.
Decrement j value by 1
Append the value of j, i, k in p, q, r.
#Back up Direction
Else If Direction[m]==”BU”
z=i-L
while(i>z):
Increment k value by 1.
Decrement i value by 1
Append the value of j, i, k in p, q, r.
#Back down right Direction
Else If Direction[m]==”BDR”
z=j+L

```

while(j<z):
    Increment i, j and k value by 1.
    Decrement i value by 1
    Append the value of j, i, k in p, q, r.
#Back down left Direction
Else If Direction[m]=="BDL"
    z=i+L
while(i<z):
    Increment i and k value by 1.
    Decrement j value by 1
    Append the value of j, i, k in p, q, r.
#Back up left Direction
Else If Direction[m]=="BUL"
    z=j-L
while(j>z):
    Increment k value by 1.
    Decrement i and j value by 1
    Append the value of j, i, k in p, q, r.
#Back up Right Direction
Else If Direction[m]=="BUR"
    z=i-L
while(i>z):
    Increment j and k value by 1.
    Decrement i value by 1.
    Append the value of j, i, k in p, q, r.
#Forward Direction
Else If Direction[m]=="F"
    z=k-L
while(k>z):
    Decrement k value by 1
    Append the value of j, i, k in p, q, r.
#Forward Right Direction
Else If Direction[m]=="FR"
    z=j+L
while(j<z):
    Increment j value by 1.
    Decrement k value by 1
    Append the value of j, i, k in p, q, r.
#Forward Down Direction
Else If Direction[m]=="FD"
    z=i+L
while(i<z):
    Increment i value by 1.
    Decrement k value by 1
    Append the value of j, i, k in p, q, r.
#Forward left Direction
Else If Direction[m]=="FL"
    z=j-L
while(j>z):
    Decrement j and k value by 1
    Append the value of j, i, k in p, q, r.
#Forward up Direction
Else If Direction[m]=="FU"
    z=i-L

```

```

while(i>z):
    Decrement i and k value by 1
    Append the value of j, i, k in p, q, r.
#Forward down right Direction
Else If Direction[m]=="FDR"
    z=j+L
while(j<z):
    Increment i and j value by 1.
    Decrement k value by 1
    Append the value of j, i, k in p, q, r.
#Forward down left Direction
Else If Direction[m]=="FDL"
    z=i+L
while(i<z):
    Increment i value by 1.
    Decrement j and k value by 1
    Append the value of j, i, k in p, q, r.
#Forward up left Direction
Else If Direction[m]=="FUL"
    z=j-L
while(j>z):
    Decrement i, j and k value by 1
    Append the value of j, i, k in p, q, r.
#Forward up right Direction
Else If Direction[m]=="FUR"
    z=i-L
while(i>z):
    Increment j value by 1.
    Decrement i and k value by 1
    Append the value of j, i, k in p, q, r.
END

```

To make the model easy to use, a graphical user interface (GUI) was made where the text file can be chosen as an input. Once the text file is chosen, the original image is shown on the screen. The same image is shown in Fig.5.8.

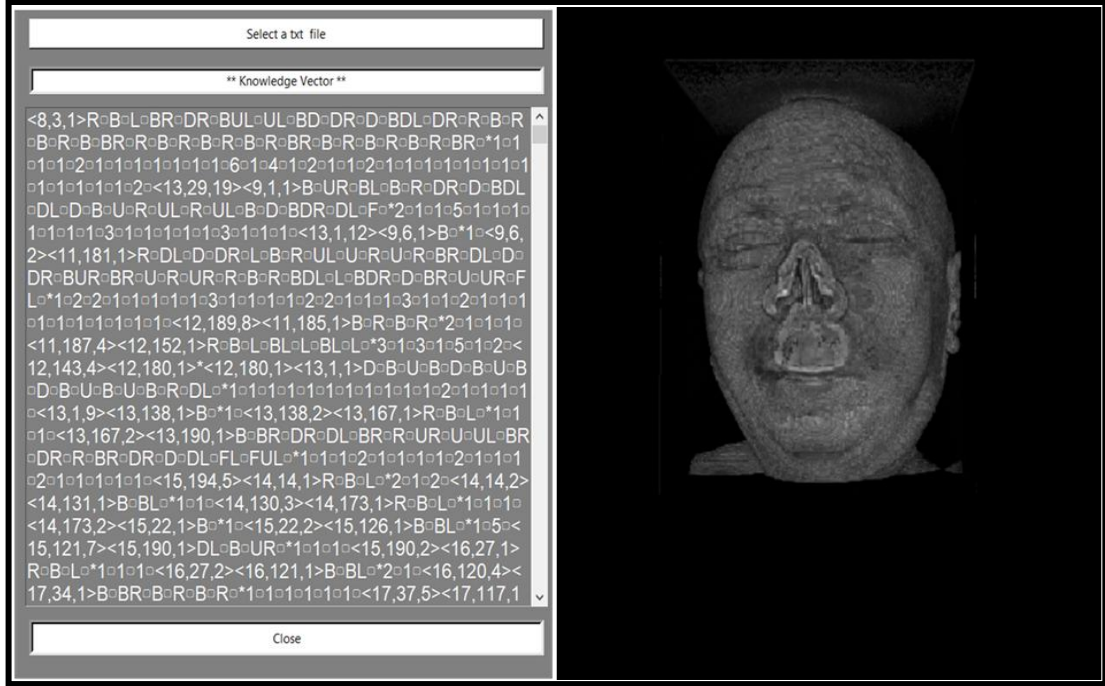


Fig. 5.8: GUI for displaying an original image from knowledge vector

5.3 Analysis of Results

5.3.1 Datasets

The proposed approach is evaluated on two datasets. The first dataset consists of five different folders. All five datasets are medical images and a detailed description is given in **Table 5.3**. This dataset is collected from Pentagon Research Centre Private Limited, Hyderabad. The second dataset is ‘Kaggle Brain MRI dataset’ which is publicly available on Kaggle[174].

Table 5.3: Description of Datasets

Dataset Name	Details of MRI Image	
Medical MRI Dataset	Toutatix	No. of Images = 256 Width = 256 Height = 256 Depth = 256 Max Gray level: 256
	Cerebrix	No. of Images = 43 Width = 256 Height = 256 Depth = 43 Max Gray level: 256
	Cetautomatix	No of Images = 256 Width = 256 Height = 256 Depth = 256 Max Gray level:114
	MRI	No of Images = 256 Width = 256 Height = 256 Depth = 256 Max Max Gray level: 181
	Cell Farct	No of Images = 256 Width = 256 Height = 256 Depth = 256 Max Gray level: 256
Kaggle Brain MRI Dataset		No. of Images = 253 Width = 845 Height = 845 Depth = 253 Max Gray level: 256

5.3.2 Evaluation Parameters

Some common image processing assessment metrics are used to assess the performance of the proposed methodology. The First parameter is accuracy which means how accurately the algorithm can find the feature vector of the image and at the same time how accurately the algorithm can reconstruct the 3D image using knowledge vector. The other most common parameters of evaluating the quality of image reconstruction are Peak signal to noise ratio (PSNR), Mean squared error (MSE), Structure similarity index(SSIM). These parameters work on the pixel-wise comparison between the original image and the reconstructed image. PSNR (based on

the MSE measure) is a ratio that influences the quality of a signal's representation by comparing its maximum achievable power to the power of corrupting noise. The power of corrupting noise is measured by MSE and SSIM is considered as the mean and variance of the image intensities. PSNR is defined as:

$$PSNR = 10 \cdot \log_{10} \frac{MAX_I^2}{MSE} \quad (5.2)$$

Where MAX_I is the maximum pixel value of the image and

$$SE(f, g) = \frac{1}{mn} \sum_i^{m-1} \sum_j^{n-1} (f(i, j) - g(i, j))^2 \quad (5.3)$$

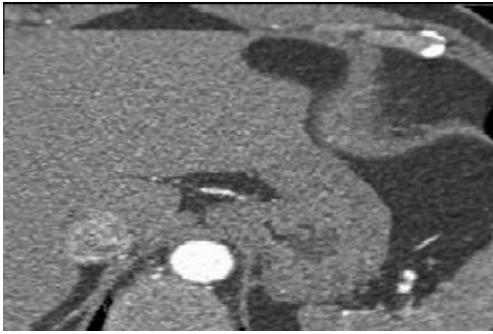
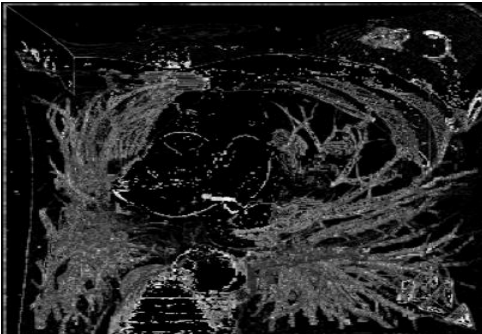
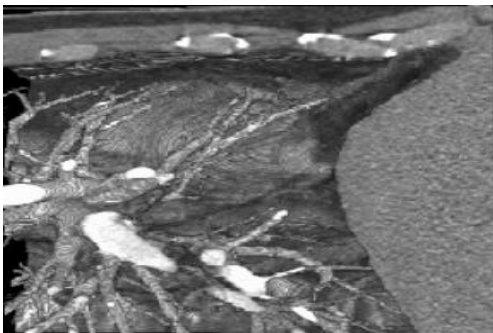
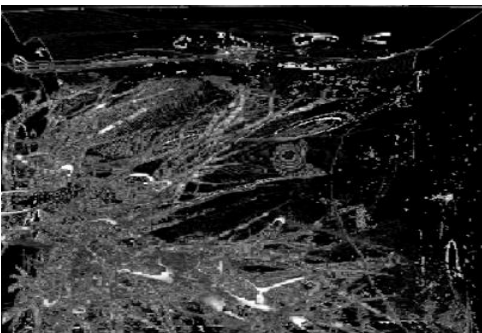
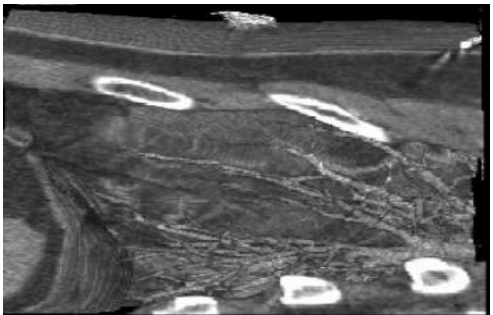
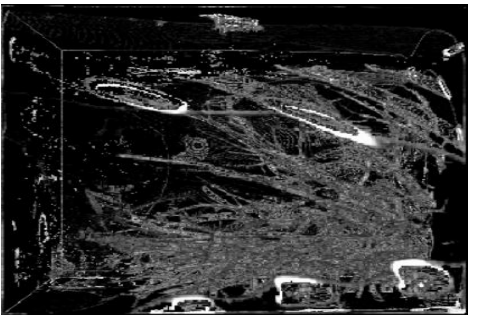
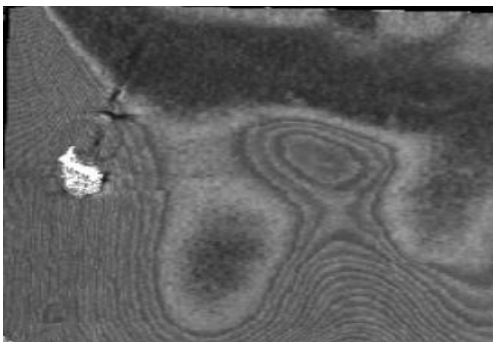
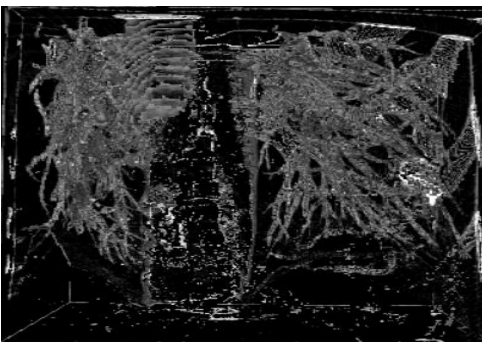
The other parameter is the execution time. The execution time is nothing but the time which is required for the reconstruction of the original image using a knowledge vector. To calculate the execution time following formula is used.

$$\text{Execution time} = 1/(\text{Performance of the Processor}) \quad (5.4)$$

5.3.3 Results

The performance of any algorithm always depends on the input dataset. Different datasets are used for the evaluation of the algorithm and different parameters are chosen for the evaluation purpose. The reconstructed images of all datasets are represented in the **Table 5.4-5.9** and it is visible in some images that reconstructed images are more clear and having sharp edges because the first algorithm works based on convex polyhedrons and our algorithm can extract the features in such a way that it is not affecting the quality of the image while reconstruction. PSNR and SSIM are calculated on all the datasets and it is represented in **Table 5.10**. The other evaluation parameters are execution time and accuracy and it is represented in **Table 5.11, Table 5.12** and **Fig. 5.9**.

Table 5.4: Image reconstruction with Toutatrix Dataset

Front View	
	
Left View	
	
Right View	
	
Top View	
	

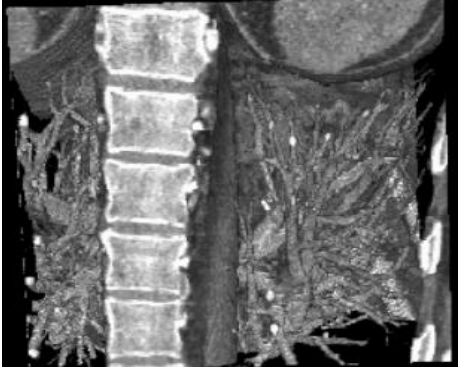
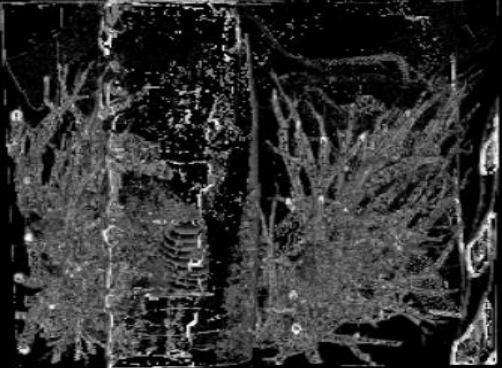
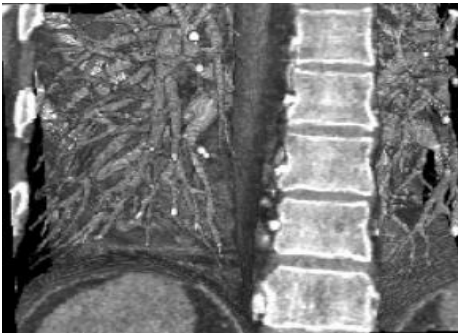
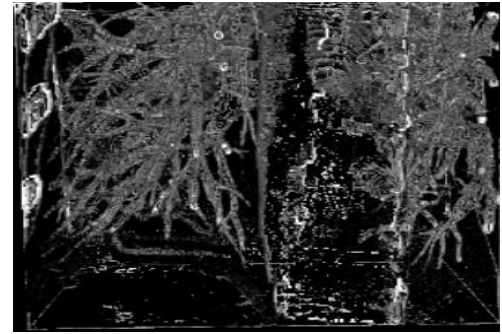
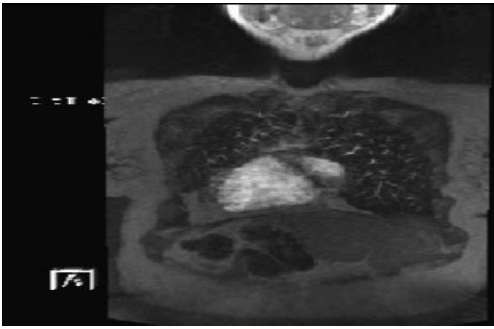
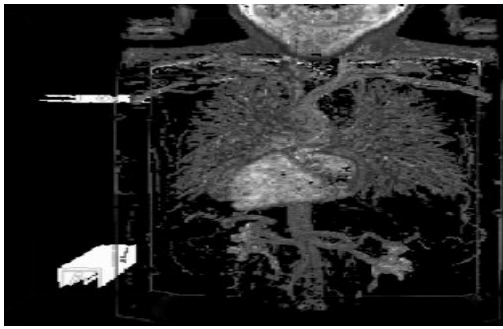
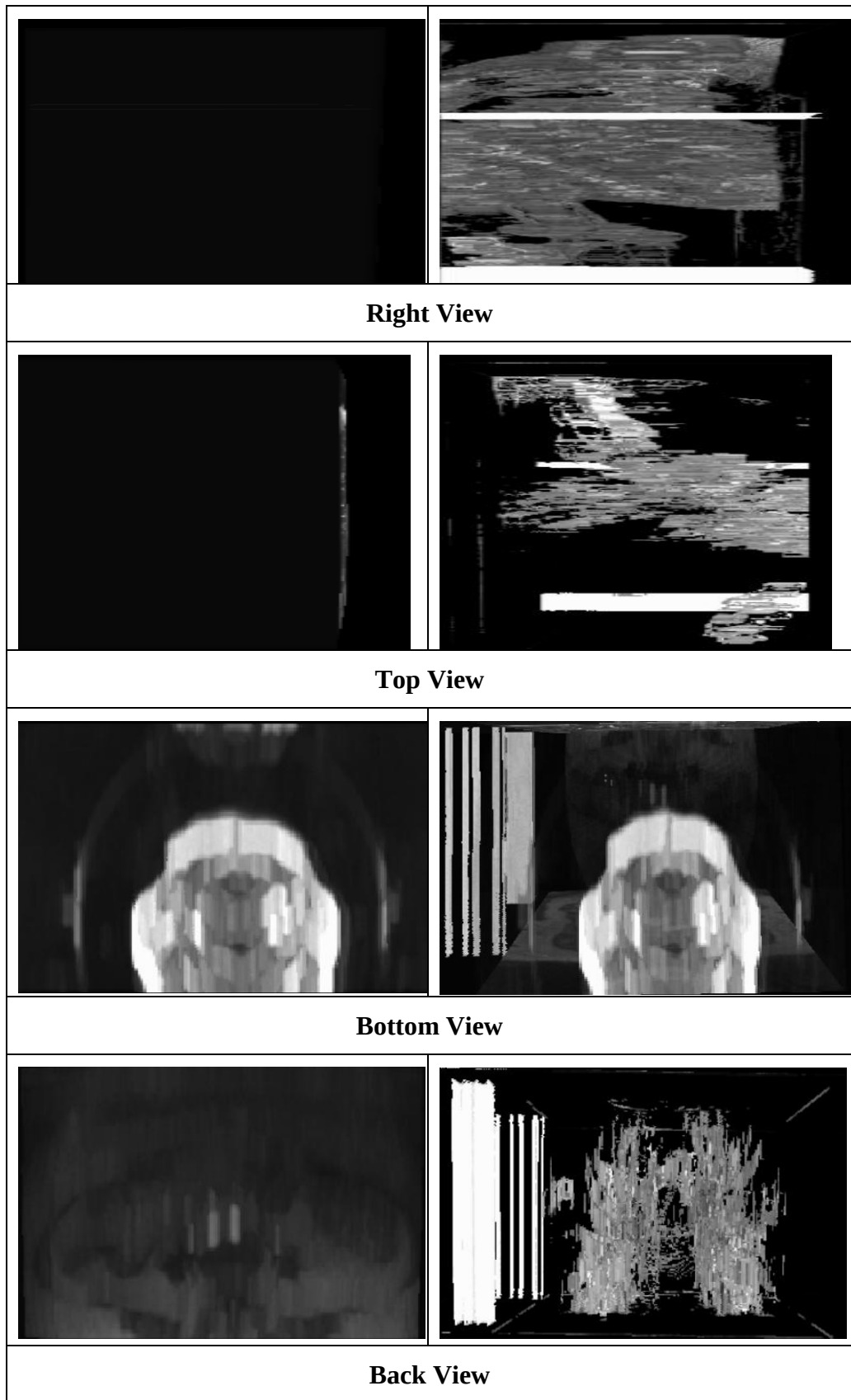
Bottom View	
	
Back View	
	

Table 5. 5: Image Reconstruction using Cetautomatix Dataset

Front View	
	
Left View	



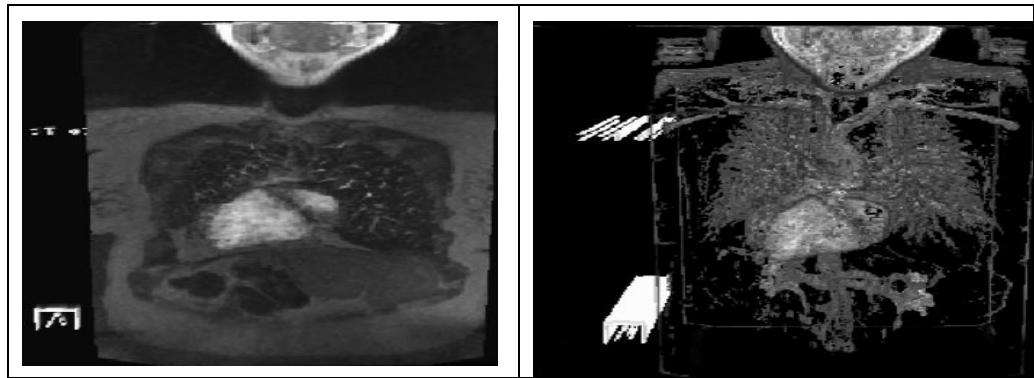
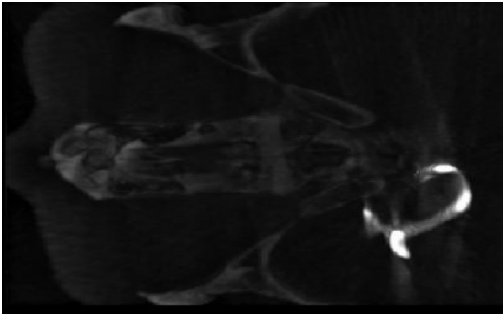
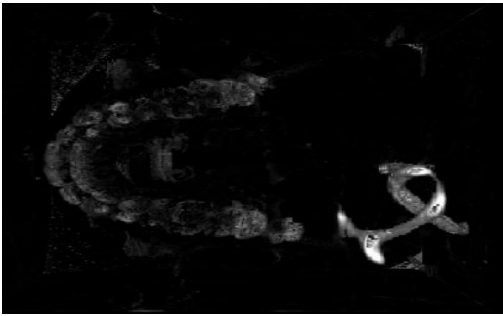
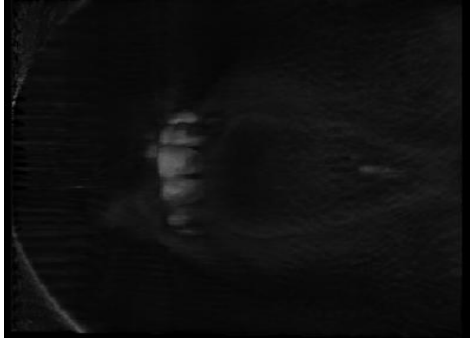
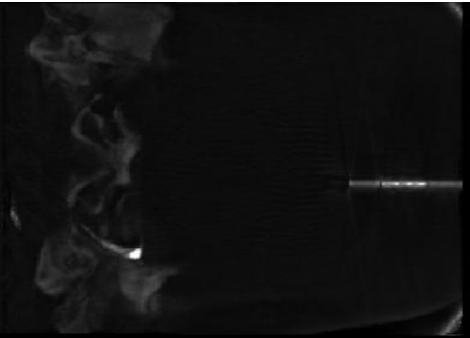
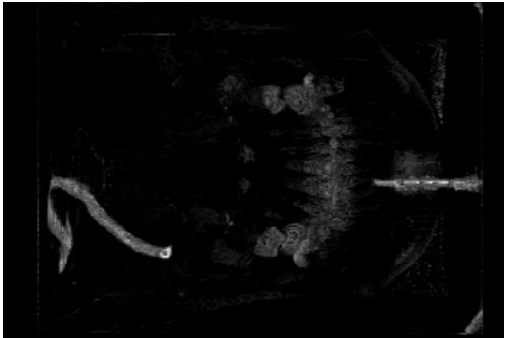


Table 5.6: Image Reconstruction with Toutatrix Dataset

Front View	
	
Left View	
	
Right View	
	
Top View	

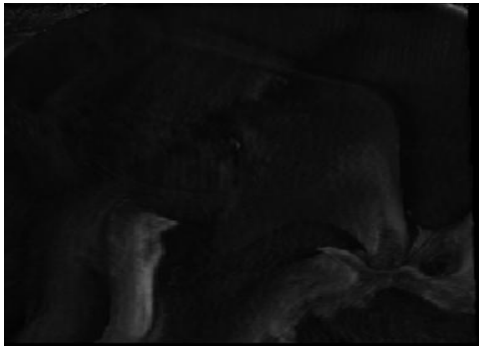
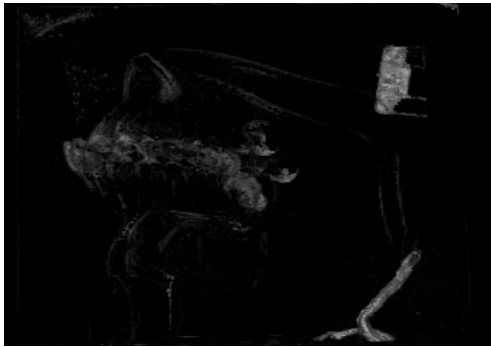
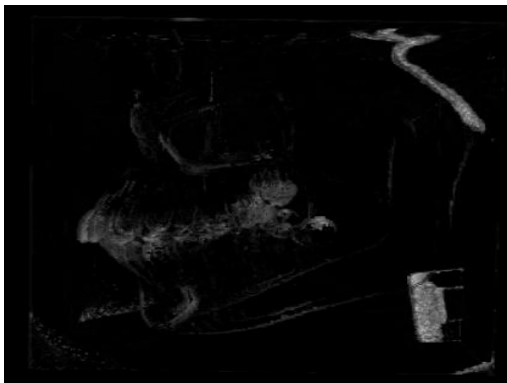
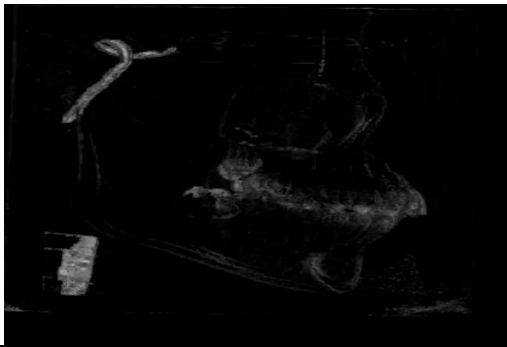
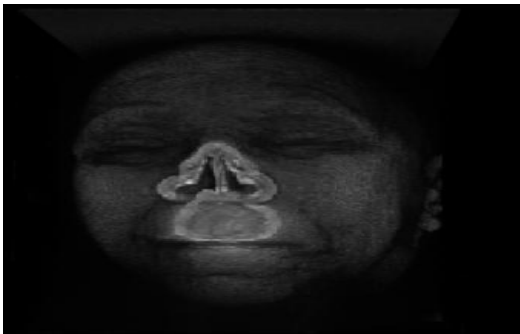
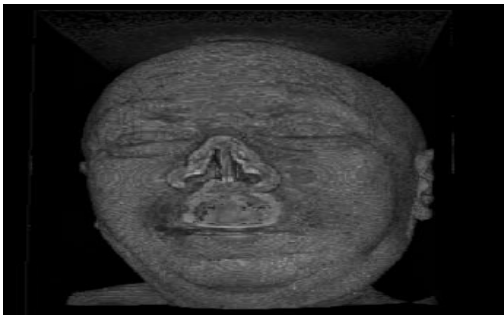
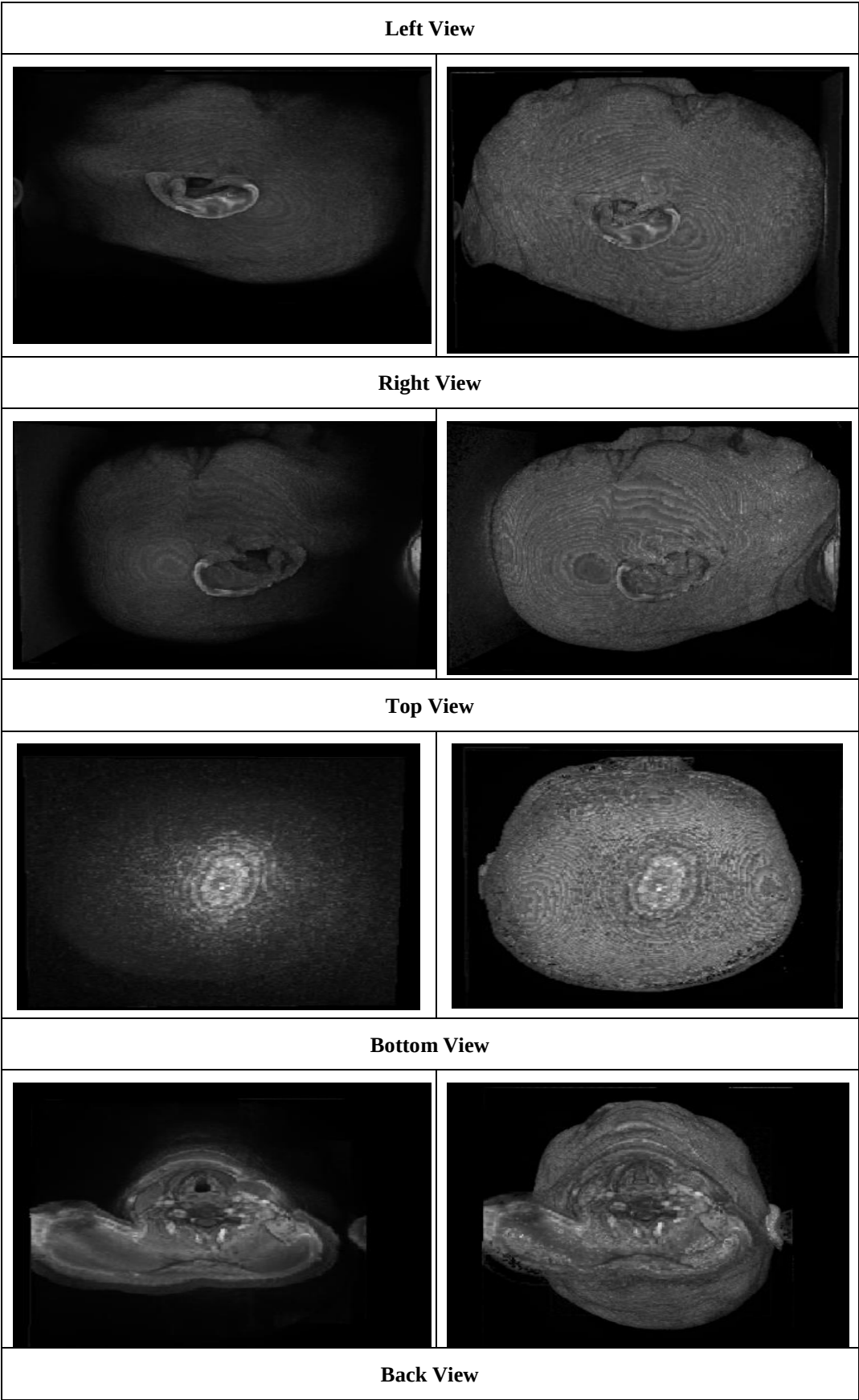
	
Bottom View	
	
Back View	
	

Table 5. 7: Image Reconstruction using Cerebrix Dataset

Front View	
	



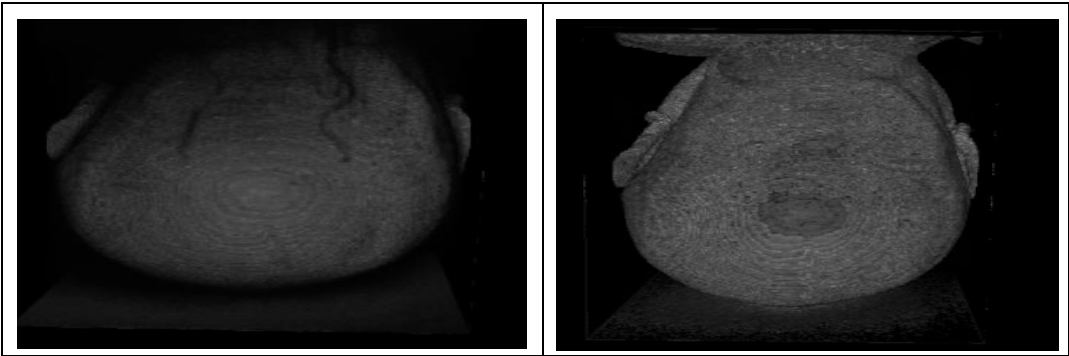


Table 5.8: Reconstruction of image on Toutatrix Dataset

Front View	
Left View	
Right View	

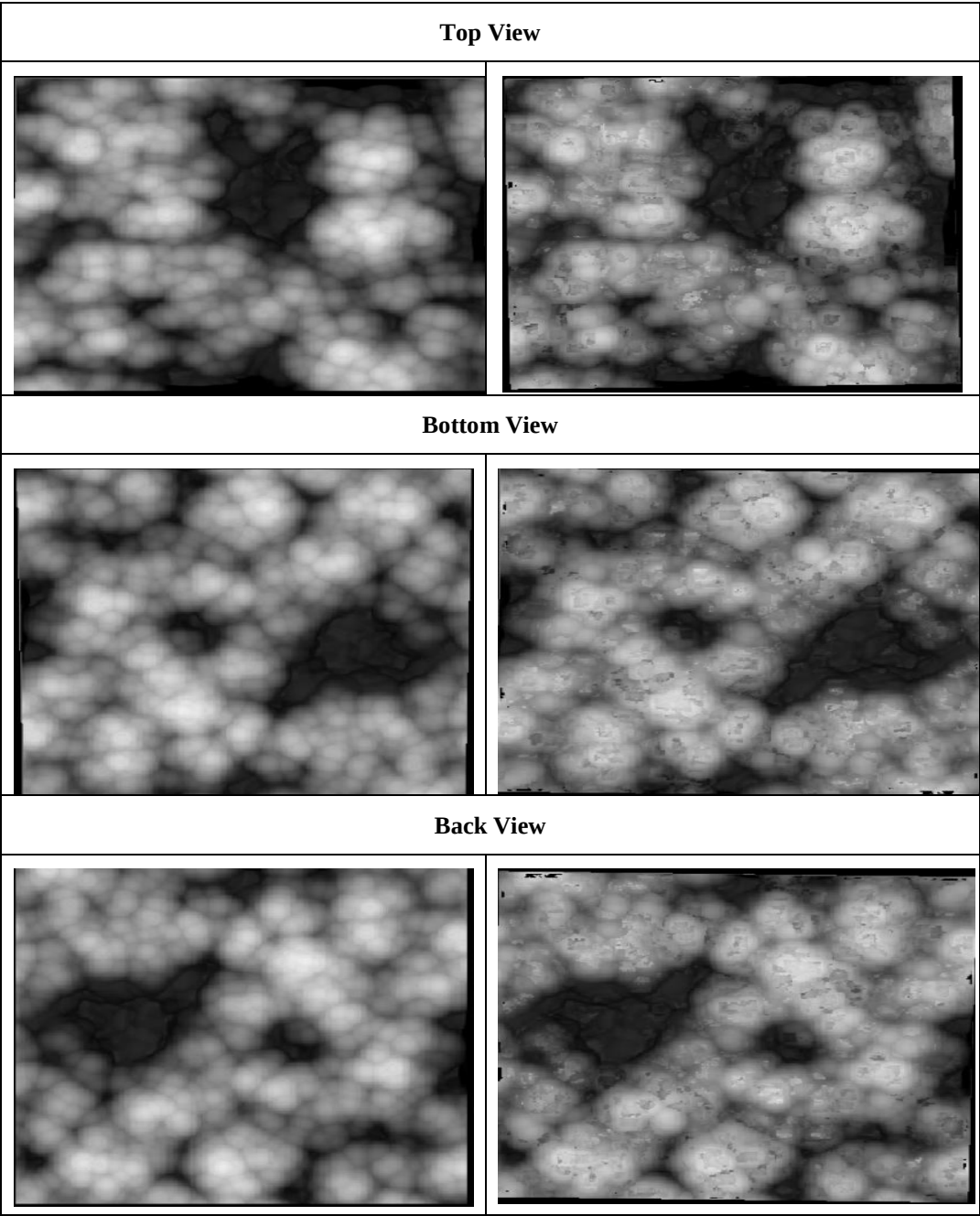
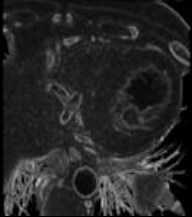
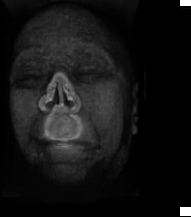
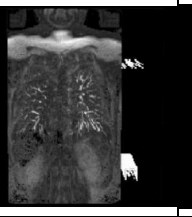
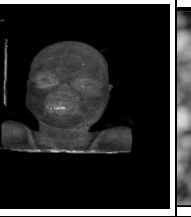


Table 5.9: Image Reconstruction using Five Medical Datasets

	Reconstructed Image				
	Toutatix	Cerebrix	Cetautomatix	MRI	Cell Farct
Front View					

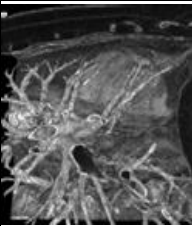
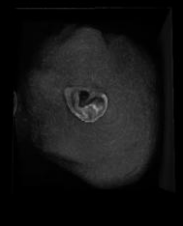
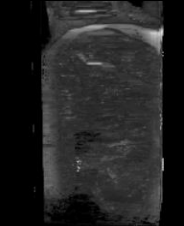
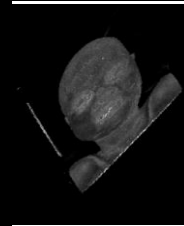
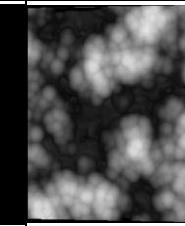
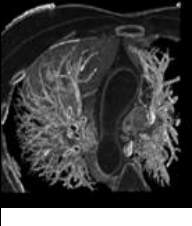
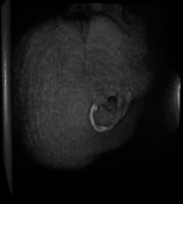
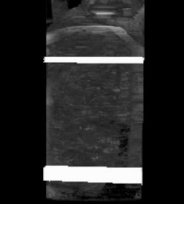
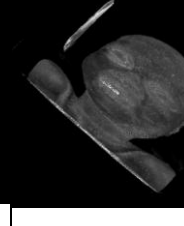
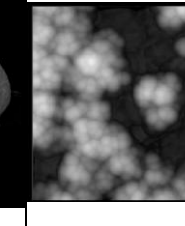
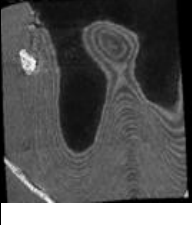
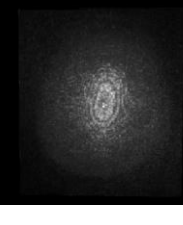
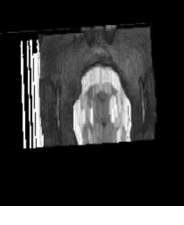
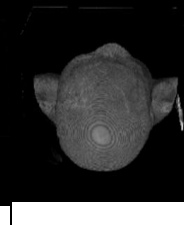
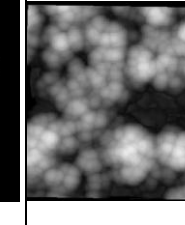
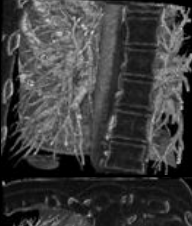
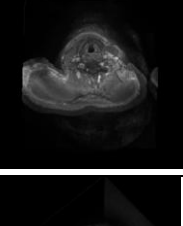
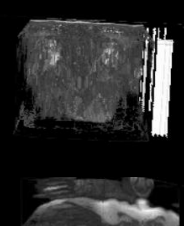
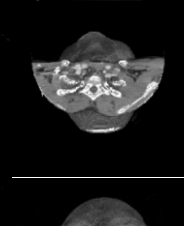
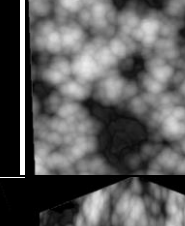
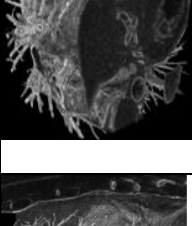
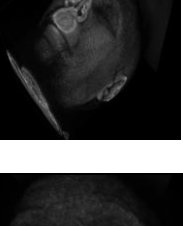
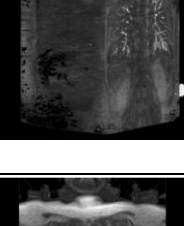
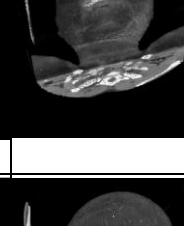
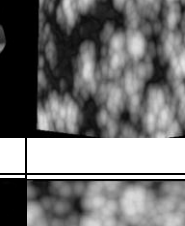
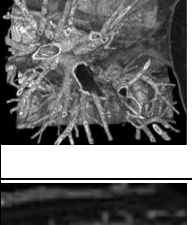
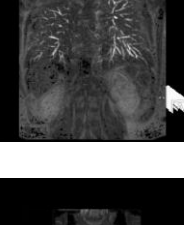
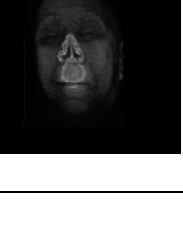
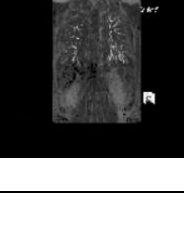
Left View					
Right View					
Top View					
Bottom View					
Side View					
Zoom In					
Zoom Out					

Table 5.10: Values of PSNR and SSIM on Various **Medical Datasets**

Dataset	PSNR	SSIM
Toutatix	32.59±1.20	0.93±0.01
Cerebrix	31.67±1.47	0.91±0.03
Cetautomatix	29.86±1.20	0.87±0.03
MRI	32.57±1.20	0.90±0.01
Cell Farct	31.26±1.80	0.89±0.01
Kaggle Brain MRI Dataset	32.50±1.17	0.90±0.02

Table 5. 11: Proposed and Current Approaches Comparison for Accuracy and Execution Time on **Kaggle Brain MRI Dataset**

Methods	Accuracy	Execution Time(Sec.)
FCM clustering & Seeded Region Growing Technique [186]	62.84%	7.12
Region Based Growing Algorithm[187]	90.52%	12.51
3D-AlexNet [188]	92.11%	-
Inception-V4 [188]	88.01%	-
ResNet 50 [188]	85.80%	-
Proposed Work	94.78%	6.76

Table 5.12: Proposed and Current Approaches Comparison for Accuracy and Execution Time on Five **Medical Datasets**

	Toutatix	Cerebrix	Cetautomatix	MRI	Cell Farct
Accuracy %	96.20	97.15	96.80	97.42	98.65
Execution Time(Sec.)	1.10	1.15	1.13	1.15	1.11

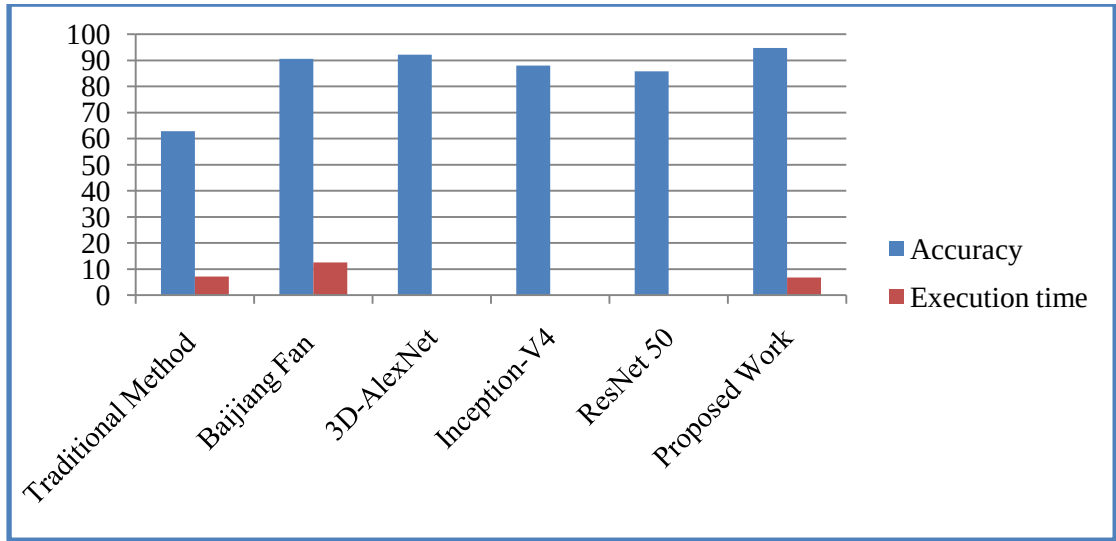


Fig.5.9: Comparative Analysis of Accuracy and Execution Time

5.4 Summary

In this chapter, we have designed a syntactic pattern recognition based approach that can represent the image into textual form. Another novel algorithm is proposed for the reconstruction of three-dimensional images. Reconstruction allows us to investigate the interior details of 3D images, such as the object's size, form, and structure, potentially advancing us in the field of medical image processing. GUI is set up so that the image can be seen correctly. The image can be rotated in all directions, and you can also zoom in and out of the image. The performance of the proposed algorithms is tested on a set of medical imaging data, and the results are better than state of art methods. On the Kaggle Brain MRI dataset, the proposed approach is 94.78% accurate and takes 6.76 seconds to run. This is better than the current state-of-the-art method. This method is better than others because it only takes a few steps to reconstruct a 3D image, which makes it unique.

Reconstruction is simply carried out to evaluate the novelty of our feature extraction approach. With the aid of the reconstruction algorithm, we can easily determine how accurately our feature extraction method can extract the features.

Further these features are used for classification of the objects which could be present in the image. For the classification purpose we have choose brain tumor as an application. We will go into more detail regarding 3D Alex-Net architecture-based object recognition and classification in the following chapter.

6.1 Introduction

In computer vision and medical image processing, object recognition is an open-ended problem. Humans don't need to do much to recognize any thing because our brains have various processing systems and billions of neurons. To recognize an object, we just need to be visually stimulated for a few moments; however the computer requires more data in order to do so. Although an entire image can be provided as input, there will be computational restrictions. Therefore, a generalized representation of different objects in one domain can give different identities to the same domain's objects. This generalized representation of objects can be further used for the identification of different diseases such as lung cancer, brain tumors etc. As Image processing techniques play an essential role in the diagnosis and monitoring of patients. The biomedical field has gained much attention from researchers and many advance artificial intelligence-based techniques are available for the early detection of the disease. However, due to the unpredictable nature of cancer, it still needs more advanced techniques.

Brain cancer is a life-threatening disease and approximately every year, 80000 new cases are reported [189]. The treatment of brain cancer always depends on the early and accurate detection of the tumor and detection is always depends on tumor type, type of pathology and the time of the first investigation of the disease. Generally, brain tumors will irregularly propagate the brain cells [154] [190] and according to the research, three types of brain tumors are identified: Glioma, Meningioma, and Pituitary. Most neurologists will use CAD-based models to detect and classify brain tumors [191]. According to our literature survey, most of the researchers used segmentation-based techniques to classify brain tumors [192]. In the last decade, statistical-based classification and detection methods are satisfactory but not a hundred percent accurate. We proposed an efficient approach based on structural pattern recognition and a deep neural network. Deep learning-based algorithms play

an essential role in the medical field and according to the literature survey Alex-Net-based architecture detects cancer in the early stage more accurately [153-158]. Therefore 3D Alex-Net architecture is used for the classification of tumor type. This research's primary motivation is to give a new insight into the structural pattern recognition technique because structural pattern recognition techniques can deal with complex medical images. Keeping this into mind, we targeted our few objectives, which are as follows:

- Proposed an effective 3D object recognition approach using feature vector and 3D Alex-Net architecture.
- The performance of the proposed work is validated on four benchmark brain tumor datasets such as Figshare dataset, Kaggle Brain MRI dataset, Medical MRI dataset and BraTS 2019 dataset.

This chapter is further divided into four sections to achieve these objectives mentioned above. The second section covers the proposed methodology. The third and fourth section covers the result section and summary.

6.2 Proposed Methodology

The proposed methodology is summarized in this section. The feature vector is the initial input used in the proposed technique, which uses the subsequent procedures (listed in Fig. 6.1) to classify the brain tumor:

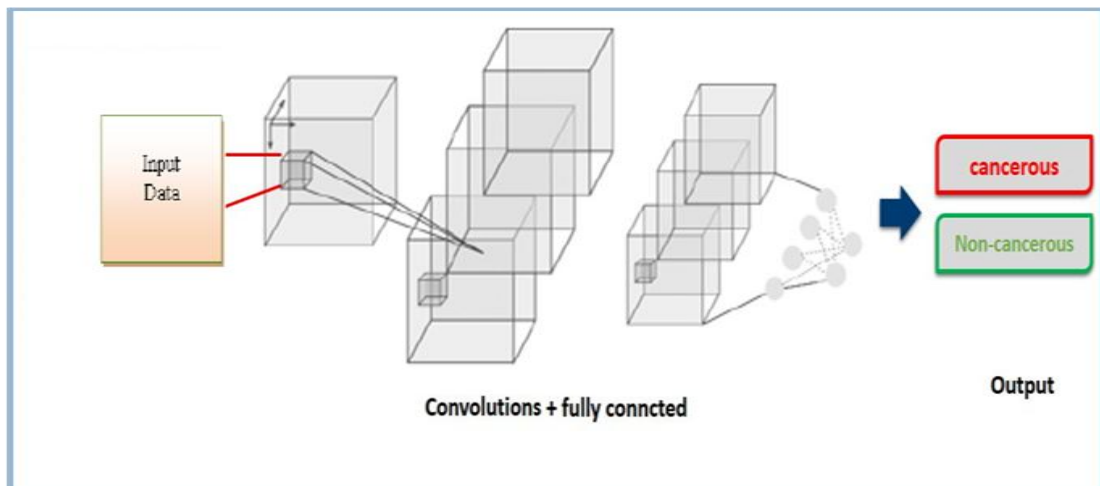


Fig. 6.1: Proposed Technique Diagram

6.2.1 Knowledge Representation of the Image

There are different types of shape and texture features are available in the image. The knowledge vector representation of an image provides the representation of the basic 3D patterns which gives enough information to classify an object. In this step the basic pattern of the 3D image are identified and these 3D patterns will give enough information to classify an object. The following methodology is taken into consideration to identify the pattern in the image using syntactic pattern recognition technique.

- **Finding out the basic patterns in a 3x3x3 3D space**

At the centre of a 3D image, there are 26 surrounding pixels. Combinations of 27 pixels at a time in a 3x3x3 region have revealed the basic patterns. Only the corner pixels are taken into account here and it is possible that none of the corner pixels are present in the image. Thus, there are 256 unique patterns in the image. In this chapter, all the conceivable patterns are depicted in a visual form and It's shown in **Table 6.1**. These patterns must be identified in the image, and then the frequency of these patterns in the image must be calculated.

Table 6.1: Basic Patterns in a 3D Image

NULL	<0,0>DR,R,U,*1,1,1,<0,2>	<0,2>D,*1,<1,2><1,0>D,*1,<2,0>
<0,1>*<0,1>	<0,0>R,D,L,*2,1,1,<1,1>	<0,1>R,D,*1,1,<1,2><1,0>D,*1,<2,0>
<0,0>*<0,0>	<1,0>R,*2,<1,2>	<0,0>D,*2,<2,0><0,2>D,*1,<1,2>
<0,0>R,*1,<0,1>	<0,1>DR,L,*1,2,<1,0>	<0,0>R,D,*2,1,<1,2><1,0>D,*1,<2,0>
<0,2>*<0,2>	<0,0>DR,R,*1,1,<1,2><1,0>*<1,0>	<1,1>R,*1,<1,2><2,0>*<2,0>
<0,1>R,*1,<0,2>	<0,0>R,DR,L,*1,1,2,<1,0>	<0,1>DR,L,DL,*1,1,1,<2,0>
<0,0>*<0,0><0,2>*<0,2>	<0,2>D,L,*1,2,<1,0>	<0,0>DR,R,*1,1,<1,2><2,0>*<2,0>
<0,0>R,*2,<0,2>	<0,1>R,D,L,*1,1,2,<1,0>	<0,0>R,DR,L,DL,*1,1,1,1,<2,0>
<1,0>*<1,0>	<0,0>DR,R,U,*1,1,1,<0,2><1,0>*<1,0>	<0,2>D,L,DL,*1,1,1,<2,0>
<0,1>DL,*1,<1,0>	<0,0>R,D,L,*2,1,2,<1,0>	<0,1>R,D,L,DL,*1,1,1,1,<2,0>
<0,0>D,*1,<1,0>	<2,0>*<2,0>	<0,0>DR,R,U,*1,1,1,<0,2><2,0>*<2,0>
<0,0>R,DL,*1,1,<1,0>	<0,1>*<0,1><2,0>*<2,0>	<0,0>R,D,L,DL,*2,1,1,1,<2,0>
<0,2>*<0,2><1,0>*<1,0>	<0,0>*<0,0><2,0>*<2,0>	<1,0>R,*2,<1,2><2,0>*<2,0>
<0,1>R,*1,<0,2><1,0>*<1,0>	<0,0>R,*1,<0,1><2,0>*<2,0>	
<0,0>D,*1,<1,0><0,2>*<0,2>		

>	<0,2>*<0,2><2,0>*<2,0>	<0,1>DR,L,DL,U,*1,1,1,1,<1,0>
<0,0>R,*2,<0,2><1,0>*<1,0>	<0,1>R,*1,<0,2><2,0>*<2,0>	<0,0>DR,R,*1,1,<1,2><1,0>D,*1,<2,0>
>	<0,0>*<0,0><0,2>*<0,2><2,0>*<2,0>	<0,0>R,DR,L,DL,U,*1,1,1,1,1,<1,0>
<1,1>*<1,1>	<0,0>R,*2,<0,2><2,0>*<2,0>	<0,2>D,L,DL,U,*1,1,1,1,<1,0>
<0,1>D,*1,<1,1>	<1,0>D,*1,<2,0>	<0,1>R,D,L,DL,U,*1,1,1,1,1,<1,0>
<0,0>DR,*1,<1,1>	<0,1>DL,D,*1,1,<2,0>	>
<0,0>R,D,*1,1,<1,1>	<0,0>D,*2,<2,0>	<0,0>DR,R,U,*1,1,1,<0,2><1,0>D,*1,<2,0>
<0,2>DL,*1,<1,1>	<0,0>R,DL,D,*1,1,1,<2,0>	<0,0>R,D,L,DL,U,*2,1,1,1,1,<1,0>
<0,1>R,DL,*1,1,<1,1>	<0,2>*<0,2><1,0>D,*1,<2,0>	>
<0,0>DR,UR,*1,1,<0,2>	<0,1>R,*1,<0,2><1,0>D,*1,<2,0>	<2,1>*<2,1>
<0,0>R,DL,*2,1,<1,1>	<0,0>D,*2,<2,0><0,2>*<0,2>	<0,1>*<0,1><2,1>*<2,1>
<1,0>R,*1,<1,1>	<0,0>R,*2,<0,2><1,0>D,*1,<2,0>	<0,0>*<0,0><2,1>*<2,1>
<0,1>D,L,*1,1,<1,0>	<1,1>DL,*1,<2,0>	<0,0>R,*1,<0,1><2,1>*<2,1>
<0,0>DR,L,*1,1,<1,0>	<0,1>D,DL,*1,1,<2,0>	<0,2>*<0,2><2,1>*<2,1>
<0,0>R,D,L,*1,1,1,<1,0>	<0,0>DR,DL,*1,1,<2,0>	<0,1>R,*1,<0,2><2,1>*<2,1>
<0,2>DL,L,*1,1,<1,0>	<0,0>R,D,DL,*1,1,1,<2,0>	<0,0>*<0,0><0,2>*<0,2><2,1>*<2,1>
<0,1>R,DL,L,*1,1,1,<1,0>	<0,2>DL,*2,<2,0>	<0,0>R,*2,<0,2><2,1>*<2,1>
<0,0>DR,L,*1,1,<1,0><0,2>*<0,2>	<0,1>R,DL,*1,2,<2,0>	<1,0>DR,*1,<2,1>
<0,0>R,DL,L,*2,1,1,<1,0>	<0,0>DR,DL,*1,1,<2,0><0,2>*<0,2>	<0,1>DL,DR,*1,1,<2,1>
<1,2>*<1,2>	<0,0>R,DL,*2,2,<2,0>	<0,0>D,DR,*1,1,<2,1>
<0,1>DR,*1,<1,2>	<1,0>R,DL,*1,1,<2,0>	<0,0>R,DL,DR,*1,1,1,<2,1>
<0,0>*<0,0><1,2>*<1,2>	<0,1>D,DL,U,*1,1,1,<1,0>	<0,2>*<0,2><1,0>DR,*1,<2,1>
<0,0>R,DR,*1,1,<1,2>	<0,0>DR,DL,U,*1,1,1,<1,0>	<0,1>R,*1,<0,2><1,0>DR,*1,<2,1>
<0,2>D,*1,<1,2>	<0,0>R,D,DL,U,*1,1,1,1,<1,0>	>
<0,1>R,D,*1,1,<1,2>	<0,2>DL,U,*2,1,<1,0>	<0,0>D,DR,*1,1,<2,1><0,2>*<0,2>
<0,0>*<0,0><0,2>D,*1,<1,2>	<0,1>R,DL,U,*1,2,1,<1,0>	>
>	<0,0>DR,DL,U,*1,1,1,<1,0><0,2>*<0,2>	<1,1>D,*1,<2,1>
<0,0>R,D,*2,1,<1,2>	<0,0>R,DL,U,*2,2,1,<1,0>	<0,1>D,*2,<2,1>
<1,0>*<1,0><1,2>*<1,2>	<1,2>*<1,2><2,0>*<2,0>	<0,0>DR,D,*1,1,<2,1>
<0,1>DR,*1,<1,2><1,0>*<1,0>	<0,1>DR,*1,<1,2><2,0>*<2,0>	<0,0>R,D,*1,2,<2,1>
<0,0>D,*1,<1,0><1,2>*<1,2>	<0,0>*<0,0><1,2>*<1,2><2,0>*<2,0>	<0,2>DL,D,*1,1,<2,1>
>	<0,0>R,DR,*1,1,<1,2><1,0>*<1,0>	<0,1>R,DL,D,*1,1,1,<2,1>
<0,2>D,*1,<1,2><1,0>*<1,0>	<0,0>R,DR,*1,1,<1,2><2,0>*<2,0>	<0,0>DR,D,*1,1,<2,1><0,2>*<0,2>
>	<0,2>D,*1,<1,2><2,0>*<2,0>	>
<0,1>R,D,*1,1,<1,2><1,0>*<1,0>		<0,0>R,DL,D,*2,1,1,<2,1>

<0,0>D,*1,<1,0><0,2>D,*1,<1,2> <0,0>R,D,*2,1,<1,2><1,0>*<1,0> <1,1>R,*1,<1,2> <0,1>DR,L,*1,1,<1,1> <0,0>DR,R,*1,1,<1,2> <0,0>R,DR,L,*1,1,1,<1,1> <0,2>D,L,*1,1,<1,1> <0,1>R,D,L,*1,1,1,<1,1> <0,0>R,DR,DL,*1,1,1,<2,1> <0,2>D,DL,*1,1,<2,1> <0,1>R,D,DL,*1,1,1,<2,1> <0,0>*<0,0><0,2>D,DL,*1,1,<2,1> <0,0>R,D,DL,*2,1,1,<2,1> <1,0>DR,UR,*1,1,<1,2> <0,1>DR,DL,UL,*1,1,1,<1,0> <0,0>D,DR,UR,*1,1,1,<1,2> <0,0>*<0,0><1,2>DL,*1,<2,1> <0,0>R,DR,DL,UL,*1,1,1,1,<1,0> <0,2>D,DL,UL,*1,1,1,<1,0> <0,1>R,D,DL,UL,*1,1,1,1,<1,0> <0,0>D,DR,UR,U,*1,1,1,1,<0,2> <0,0>R,D,DL,UL,*2,1,1,1,<1,0> <1,1>R,DL,*1,1,<2,1> <0,1>DR,DL,U,*1,1,1,<1,1> <0,0>DR,R,DL,*1,1,1,<2,1> <0,0>R,DR,DL,U,*1,1,1,1,<1,1> <0,2>D,DL,U,*1,1,1,<1,1> <0,1>R,D,DL,U,*1,1,1,1,<1,1> <0,0>DR,R,DL,*1,1,1,<2,1> <0,2>*<0,2> <0,0>R,D,DL,U,*2,1,1,1,<1,1>	<0,1>R,D,*1,1,<1,2><2,0>*<2,0> <0,0>*<0,0><0,2>D,*1,<1,2><2,0>*<2,0> <0,0>R,D,*2,1,<1,2><2,0>*<2,0> <1,0>D,*1,<2,0><1,2>*<1,2> <0,1>DR,*1,<1,2><1,0>D,*1,<2,0> <1,2>DL,*1,<2,1> <0,1>R,DL,DR,*1,1,1,<2,2> <0,0>DR,*2,<2,2><0,2>*<0,2> <0,0>R,DL,DR,*2,1,1,<2,2> <1,0>R,DR,*1,1,<2,2> <0,1>D,DR,*1,1,<2,2><1,0>*<1,0> <0,0>DR,*2,<2,2><1,0>*<1,0> <0,0>R,D,DR,*1,1,1,<2,2><1,0>*<1,0> <0,2>DL,DR,*1,1,<2,2><1,0>*<1,0> <0,1>R,DL,DR,*1,1,1,<2,2><1,0>*<1,0> <0,0>DR,*2,<2,2><0,2>*<0,2> <1,0>*<1,0> <0,0>R,DL,DR,*2,1,1,<2,2><1,0>*<1,0> <1,2>D,*1,<2,2> <0,1>DR,D,*1,1,<2,2> <0,0>*<0,0><1,2>D,*1,<2,2> <0,0>R,DR,D,*1,1,1,<2,2> <0,2>D,*2,<2,2> <0,1>R,D,*1,2,<2,2> <0,0>*<0,0><0,2>D,*2,<2,2> <0,0>R,D,*2,2,<2,2> <1,0>*<1,0><1,2>D,*1,<2,2> <0,1>DR,D,*1,1,<2,2><1,0>*<1,0> <0,0>D,*1,<1,0><1,2>D,*1,<2,2> <0,0>R,DR,D,*1,1,1,<2,2><1,0>*<1,0>	<1,0>R,D,*1,1,<2,1> <0,1>D,UL,*2,1,<1,0> <0,0>DR,D,UL,*1,1,1,<1,0> <0,0>R,D,UL,*1,2,1,<1,0> <0,2>DL,D,UL,*1,1,1,<1,0> <0,1>R,DL,D,UL,*1,1,1,1,<1,0> <0,0>DR,D,UL,*1,1,1,<1,0><0,2>*<0,2> <0,0>R,DL,D,UL,*2,1,1,1,<1,0> <0,2>DL,D,L,*1,1,1,<2,0> <0,1>R,DL,D,L,*1,1,1,1,<2,0> <0,0>DR,D,L,*1,1,1,<2,0><0,2>*<0,2> <0,0>R,DL,D,L,*2,1,1,1,<2,0> <1,0>R,D,L,*1,1,1,<2,0> <0,1>D,L,U,*2,1,1,<1,0> <0,0>DR,D,L,U,*1,1,1,1,<1,0> <0,0>R,D,L,U,*1,2,1,1,<1,0> <0,2>DL,D,L,U,*1,1,1,1,<1,0> <0,1>R,DL,D,L,U,*1,1,1,1,1,<1,0> <0,0>DR,D,L,U,*1,1,1,1,<1,0><0,2>*<0,2> <0,0>R,DL,D,L,U,*2,1,1,1,1,<1,0> <1,2>DL,L,*1,1,<2,0> <0,1>DR,DL,L,*1,1,1,<2,0> <0,0>*<0,0><1,2>DL,L,*1,1,<2,0> <0,0>R,DR,DL,L,*1,1,1,1,<2,0> <0,2>D,DL,L,*1,1,1,<2,0> <0,1>R,D,DL,L,*1,1,1,1,<2,0> <0,0>*<0,0><0,2>D,DL,L,*1,1,1,<2,0> <0,0>R,D,DL,L,*2,1,1,1,<2,0> <1,0>DR,L,*1,1,<2,0><1,2>*<1,2> <0,1>DR,DL,L,U,*1,1,1,1,<1,0> <0,0>D,DR,L,*1,1,1,<2,0><1,2>*<1,2> <0,0>R,DR,DL,L,U,*1,1,1,1,1,<1,1>
--	---	---

1>	<0,2>D,*2,<2,2><1,0>*<1,0>	0>
<1,0>R,DL,*2,1,<2,1>	<0,1>R,D,*1,2,<2,2><1,0>*<1,0>	<0,2>D,DL,L,U,*1,1,1,1,<1,0>
<0,1>DR,DL,UL,R,*1,1,1,1,<1,1>	<0,0>D,*1,<1,0><0,2>D,*2,<2,2>	<0,1>R,D,DL,L,U,*1,1,1,1,1,<1,0>
<0,0>DR,R,DL,UL,*1,1,1,1,<1,0>	<0,0>R,D,*2,2,<2,2><1,0>*<1,0>	<0,0>D,DR,L,*1,1,1,<2,0><0,2>D,*1,<1,2>
<0,0>R,DR,DL,UL,R,*1,1,1,1,1,<1,1>	<1,1>R,D,*1,1,<2,2>	<0,0>R,D,DL,L,U,*2,1,1,1,1,<1,0>
<0,2>D,DL,UL,R,*1,1,1,1,<1,1>	<0,1>DR,D,UL,*1,1,1,<1,1>	<1,1>R,DL,L,*1,1,1,<2,0>
<0,1>R,D,DL,UL,R,*1,1,1,1,1,<1,1>	<0,0>DR,R,D,*1,1,1,<2,2>	<0,1>DR,DL,L,UR,*1,1,1,1,<1,1>
<0,0>DR,R,DL,UL,*1,1,1,1,<1,0><0,2>*<0,2>	<0,0>R,DR,D,UL,*1,1,1,1,<1,1>	<0,0>DR,R,DL,L,*1,1,1,1,<2,0>
<0,0>R,D,DL,UL,R,*2,1,1,1,1,<1,1>	<0,2>D,UL,*2,1,<1,1>	<0,0>R,DR,DL,L,UR,*1,1,1,1,1,<1,1>
<2,0>R,*1,<2,1>	<0,1>R,D,UL,*1,2,1,<1,1>	<0,2>D,DL,L,UR,*1,1,1,1,<1,1>
<0,1>*<0,1><2,0>R,*1,<2,1>	<0,0>DR,R,D,*1,1,1,<2,2><0,2>*<0,2>	<0,1>R,D,DL,L,UR,*1,1,1,1,1,<1,1>
<0,0>*<0,0><2,0>R,*1,<2,1>	<0,0>R,D,UL,*2,2,1,<1,1>	<0,0>DR,R,DL,L,*1,1,1,1,<2,0><0,2>*<0,2>
<0,0>R,*1,<0,1><2,0>R,*1,<2,1>	<1,0>R,D,*2,1,<2,2>	<0,0>R,D,DL,L,UR,*2,1,1,1,1,<1,1>
<0,2>*<0,2><2,0>R,*1,<2,1>	<0,1>DR,D,UL,L,*1,1,1,1,<1,0>	<1,0>R,DL,L,*2,1,1,<2,0>
<0,1>R,*1,<0,2><2,0>R,*1,<2,1>	<0,0>DR,R,D,*1,1,1,<2,2><1,0>*<1,0>	<0,1>DR,DL,L,U,R,*1,1,1,1,1,<1,1>
<0,0>*<0,0><0,2>*<0,2><2,0>R,*1,<2,1>	<0,0>R,DR,D,UL,L,*1,1,1,1,1,<1,0>	<0,0>DR,R,DL,L,U,*1,1,1,1,1,<1,0>
<0,0>R,*2,<0,2><2,0>R,*1,<2,1>	<0,2>D,UL,L,*2,1,1,<1,0>	<0,0>R,DR,DL,L,U,R,*1,1,1,1,1,1,<1,1>
<1,0>DR,L,*1,1,<2,0>	<0,1>R,D,UL,L,*1,2,1,1,<1,0>	<0,2>D,DL,L,U,R,*1,1,1,1,1,<1,1>
<0,1>DL,DR,L,*1,1,1,<2,0>	<0,0>DR,R,D,*1,1,1,<2,2><0,2>*<0,2><1,0>*<1,0>	<0,1>R,D,DL,L,U,R,*1,1,1,1,1,1,<1,1>
<0,0>D,DR,L,*1,1,1,<2,0>	<0,0>R,D,UL,L,*2,2,1,1,<1,0>	<0,0>DR,R,DL,L,U,*1,1,1,1,1,<1,0><0,2>*<0,2>
<0,0>R,DL,DR,L,*1,1,1,1,<2,0>	<2,0>*<2,0><2,2>*<2,2>	<0,0>R,D,DL,L,U,R,*2,1,1,1,1,1,<1,1>
<0,2>*<0,2><1,0>DR,L,*1,1,<2,0>	<0,1>*<0,1><2,0>*<2,0><2,2>*<2,2>	<2,2>*<2,2>
<0,1>R,*1,<0,2><1,0>DR,L,*1,1,<2,0>	<0,0>*<0,0><2,0>*<2,0><2,2>*<2,2>	<0,1>*<0,1><2,2>*<2,2>
<0,0>D,DR,L,*1,1,1,<2,0><0,2>*<0,2>	<0,1>R,*1,<0,2><2,0>*<2,0><2,2>*<2,2>	<0,0>R,*1,<0,1><2,2>*<2,2>
<0,0>R,*2,<0,2><1,0>DR,L,*1,1,<2,0>	<0,0>*<0,0><0,2>*<0,2><2,0>*<2,0><2,2>*<2,2>	<0,2>*<0,2><2,2>*<2,2>
	<0,0>R,*2,<0,2><2,0>*<2,0><2,2>*<2,2>	<0,1>R,*1,<0,2><2,2>*<2,2>
		<0,0>*<0,0><0,2>*<0,2><2,2>*<2,2>*<

0>	<0,0>DR,L,*2,1,<2,1><0,2>*<0,2>	<0,2><1,0>D,*1,<2,0>
<0,0>R,D,DR,*1,1,1,<2,2><2,0>*<2,0>	<0,0>R,DL,DR,L,*2,1,1,1,<2,1>	<0,1>R,D,L,U,*1,2,1,1,<1,1>
<0,2>DL,DR,*1,1,<2,2><2,0>*<2,0>	<1,0>R,DR,L,*1,1,1,<2,1>	<0,0>DR,R,D,L,*1,1,1,1,<2,1><0,2>*<0,2>
<0,1>R,DL,DR,*1,1,1,<2,2><2,0>*<2,0>	<0,1>D,DR,L,UL,*1,1,1,1,<1,0>	<0,0>R,D,L,U,*2,2,1,1,<1,1>
<0,0>DR,*2,<2,2><0,2>*<0,2><2,0>*<2,0>	<0,0>DR,L,UL,*2,1,1,<1,0>	<1,0>R,D,L,*2,1,1,<2,1>
<0,0>R,DL,DR,*2,1,1,<2,2><2,0>*<2,0>	<0,0>R,D,DR,L,UL,*1,1,1,1,1,<1,0>	<0,1>DR,D,L,UL,R,*1,1,1,1,1,<1,1>
<1,0>R,DR,*1,1,<2,2><2,0>*<2,0>	<0,2>DL,DR,L,UL,*1,1,1,1,<1,0>	<0,0>DR,R,D,L,UL,*1,1,1,1,1,<1,0>
<0,1>D,DR,*1,1,<2,2><1,0>D,*1,<2,0>	<0,1>R,DL,DR,L,UL,*1,1,1,1,1,<1,0>	<0,0>R,DR,D,L,UL,R,*1,1,1,1,1,1,<1,1>
<0,0>DR,*2,<2,2><1,0>D,*1,<2,0>	<0,0>DR,L,UL,*2,1,1,<1,0><0,2>*<0,2>	<0,2>D,L,UL,R,*2,1,1,1,<1,1>
<0,0>R,D,DR,*1,1,1,<2,2><1,0>D,*1,<2,0>	<0,0>R,DL,DR,L,UL,*2,1,1,1,1,<1,0>	<0,1>R,D,L,UL,R,*1,2,1,1,1,<1,1>
<0,2>DL,DR,*1,1,<2,2><1,0>D,*1,<2,0>	<1,2>D,L,*1,1,<2,1>	<0,0>DR,R,D,L,UL,*1,1,1,1,1,<1,0><0,2>*<0,2>
<0,1>R,DL,DR,*1,1,1,<2,2><1,0>D,*1,<2,0>	<0,1>DR,D,L,*1,1,1,<2,1>	<0,0>R,D,L,UL,R,*2,2,1,1,1,<1,1>
<0,0>DR,*2,<2,2><0,2>*<0,2><2,0><1,0>D,*1,<2,0>	<0,0>*<0,0><1,2>D,L,*1,1,<2,1>	<2,0>R,*2,<2,2>
<0,0>R,DL,DR,*2,1,1,<2,2><1,0>D,*1,<2,0>	<0,0>R,DR,D,L,*1,1,1,1,<2,1>	<0,0>DR,R,D,L,*1,1,1,2,<2,0>
<1,2>D,*1,<2,2><2,0>*<2,0>	<0,2>D,L,*2,1,<2,1>	<0,0>R,DR,D,L,UR,*1,1,1,2,1,<1,1>
<0,1>DR,D,*1,1,<2,2><2,0>*<2,0>	<0,1>R,D,L,*1,2,1,<2,1>	<0,2>D,L,UR,*2,2,1,<1,1>
<0,0>*<0,0><1,2>D,*1,<2,2><2,0>*<2,0>	<0,0>*<0,0><0,2>D,L,*2,1,<2,1>	<0,1>R,D,L,UR,*1,2,2,1,<1,1>
<1,1>R,D,L,*1,1,1,<2,1>	<0,0>R,D,L,*2,2,1,<2,1>	<0,1>*<0,1><2,0>R,*2,<2,2>
<0,1>DR,D,L,U,*1,1,1,1,<1,1>	<1,0>DR,R,U,*1,1,1,<1,2>	<0,0>*<0,0><2,0>R,*2,<2,2>
<0,0>DR,R,D,L,*1,1,1,1,<2,1>	<0,1>DR,D,L,UL,*1,1,1,1,<1,0>	<0,0>R,*1,<0,1><2,0>R,*2,<2,2>
<0,0>R,DR,D,L,U,*1,1,1,1,1,<1,1>	<0,0>D,DR,R,U,*1,1,1,1,<1,2>	<0,2>*<0,2><2,0>R,*2,<2,2>
<0,2>D,L,U,*2,1,1,<1,1>	<0,0>R,DR,D,L,UL,*1,1,1,1,1,<1,0>	<0,1>R,*1,<0,2><2,0>R,*2,<2,2>
<0,2>D,UL,DL,*2,1,1,<2,0>	<0,2>*<0,2><1,0>DR,R,*1,1,<2,2><2,0>*<2,0>	<0,0>*<0,0><0,2>*<0,2><2,0>R,*2,<2,2>
<0,0>R,D,DR,L,U,*1,1,1,2,1,<1,0>	<0,1>R,*1,<0,2><1,0>DR,R,*1,1,<2,2><2,0>*<2,0>	<0,0>R,*2,<0,2><2,0>R,*2,<2,2>
	<0,0>R,D,L,UL,*2,2,1,1,<1,0>	<1,0>DR,R,*1,1,<2,2><2,0>*<2,0>
	<0,1>D,DR,L,*1,1,2,<2,0>	<0,1>DL,DR,R,*1,1,1,<2,2><2,0>*<2,0>
	<0,0>DR,L,*2,2,<2,0>	<0,0>D,DR,R,*1,1,1,<2,2><2,0>*<2,0>
	<0,0>R,D,DR,L,*1,1,1,2,<2,0>	<0,0>R,DL,DR,R,*1,1,1,1,<2,2><

$\langle 0,2 \rangle \text{DL,DR,L,U,*1,1,2,1}, \langle 1,0 \rangle$ $\langle 0,1 \rangle \text{R,DL,DR,L,U,*1,1,1,2,1}, \langle 1,0 \rangle$ $\langle 0,0 \rangle \text{DR,L,U,*2,2,1}, \langle 1,0 \rangle \langle 0,2 \rangle * \langle 0,2 \rangle$ $\langle 0,0 \rangle \text{R,DL,DR,L,U,*2,1,1,2,1}, \langle 1,0 \rangle$ $\langle 1,2 \rangle \text{D,L,*1,2}, \langle 2,0 \rangle$ $\langle 0,1 \rangle \text{DR,D,L,*1,1,2}, \langle 2,0 \rangle$ $\langle 0,0 \rangle * \langle 0,0 \rangle \langle 1,2 \rangle \text{D,L,*1,2}, \langle 2,0 \rangle$ $\langle 0,0 \rangle \text{R,DR,D,L,*1,1,1,2}, \langle 2,0 \rangle$ $2,0 \rangle * \langle 2,0 \rangle$ $\langle 0,0 \rangle \text{R,DR,D,L,U,*1,1,1,2,1}, \langle 1,0 \rangle$ $\langle 0,2 \rangle \text{D,L,U,*2,2,1}, \langle 1,0 \rangle$ $\langle 0,2 \rangle \text{D,L,UL,*2,1,1}, \langle 1,0 \rangle$ $\langle 0,1 \rangle \text{R,D,L,UL,*1,2,1,1}, \langle 1,0 \rangle$ $\langle 0,0 \rangle \text{D,DR,R,U,*1,1,1,2}, \langle 0,2 \rangle$	$\langle 0,2 \rangle \text{DL,DR,L,*1,1,2}, \langle 2,0 \rangle$ $\langle 0,1 \rangle \text{R,DL,DR,L,*1,1,1,2}, \langle 2,0 \rangle$ $\langle 0,0 \rangle \text{DR,L,*2,2}, \langle 2,0 \rangle \langle 0,2 \rangle * \langle 0,2 \rangle$ $\langle 0,0 \rangle \text{R,DL,DR,L,*2,1,1,2}, \langle 2,0 \rangle$ $\langle 1,0 \rangle \text{R,DR,L,*1,1,2}, \langle 2,0 \rangle$ $\langle 0,1 \rangle \text{D,DR,L,U,*1,1,2,1}, \langle 1,0 \rangle$ $\langle 0,0 \rangle \text{DR,L,U,*2,2,1}, \langle 1,0 \rangle * \langle 2,0 \rangle$ $\langle 0,0 \rangle \text{R,*2}, \langle 0,2 \rangle \langle 1,0 \rangle \text{DR,R,*1,1}, \langle 2,2 \rangle \langle 2,0 \rangle * \langle 2,0 \rangle$ $\langle 1,1 \rangle \text{DR,L,*1,2}, \langle 2,0 \rangle$ $\langle 0,1 \rangle \text{R,D,L,U,*1,2,2,1}, \langle 1,0 \rangle$ $\langle 0,0 \rangle \text{D,DR,R,U,*1,1,1,2}, \langle 0,2 \rangle \langle 2,0 \rangle * \langle 2,0 \rangle$ $\langle 0,0 \rangle \text{R,D,L,U,*2,2,2,1}, \langle 1,0 \rangle$ $\langle 1,1 \rangle \text{R,D,L,*1,1,2}, \langle 2,0 \rangle$ $\langle 0,1 \rangle \text{DR,D,L,UR,*1,1,2,1}, \langle 1,1 \rangle$	$2,0 \rangle * \langle 2,0 \rangle$ $\langle 0,0 \rangle \text{D,DR,R,*1,1,1}, \langle 2,2 \rangle \langle 0,2 \rangle * \langle 0,2 \rangle \langle 2,0 \rangle$ $\langle 0,0 \rangle \text{DR,R,D,L,*1,1,1,2}, \langle 2,0 \rangle \langle 0,2 \rangle * \langle 0,2 \rangle$ $\langle 0,0 \rangle \text{R,D,L,UR,*2,2,2,1}, \langle 1,1 \rangle$ $\langle 1,0 \rangle \text{R,D,L,*2,1,2}, \langle 2,0 \rangle$ $\langle 0,1 \rangle \text{DR,D,L,U,R,*1,1,2,1,1}, \langle 1,1 \rangle$ $\langle 0,0 \rangle \text{DR,R,D,L,U,*1,1,1,2,1}, \langle 1,0 \rangle$ $\langle 0,0 \rangle \text{R,DR,D,L,U,*1,1,1,2,1,1}, \langle 1,1 \rangle$ $\langle 0,2 \rangle \text{D,L,U,R,*2,2,1,1}, \langle 1,1 \rangle$ $\langle 0,1 \rangle \text{R,D,L,U,R,*1,2,2,1,1}, \langle 1,1 \rangle$ $\langle 0,0 \rangle \text{DR,R,D,L,U,*1,1,1,2,1}, \langle 1,0 \rangle \langle 0,2 \rangle * \langle 0,2 \rangle$ $\langle 0,0 \rangle \text{R,D,L,U,R,*2,2,2,1,1}, \langle 1,1 \rangle$ $\langle 0,2 \rangle \text{D,L,*2,2}, \langle 2,0 \rangle$ $\langle 0,1 \rangle \text{R,D,L,*1,2,2}, \langle 2,0 \rangle$ $\langle 0,0 \rangle * \langle 0,0 \rangle \langle 0,2 \rangle \text{D,L,*2,2}, \langle 2,0 \rangle$ $\langle 0,0 \rangle \text{R,D,L,*2,2,2}, \langle 2,0 \rangle$ $\langle 1,0 \rangle \text{DR,R,U,*1,1,1}, \langle 1,2 \rangle \langle 2,0 \rangle * \langle 2,0 \rangle$ $\langle 0,1 \rangle \text{DR,D,L,U,*1,1,2,1}, \langle 1,0 \rangle$ $\langle 0,0 \rangle \text{D,DR,R,U,*1,1,1,1}, \langle 1,2 \rangle \langle$
--	--	---

6.2.2 Object Recognition using 3D Alex-Net

The three-dimensional knowledge vector is a vector of 256 values that saves an image's overall syntactic pattern. This image can be two or three-dimensional. The purpose of such a vector is to get a succinct representation of a 2-D or 3-D image. These vectors, therefore, hold essential information regarding the shape and morphology. Therefore, theoretically, these vectors hold information about an image and thus can be used to represent it. The corresponding vector is generated using the research shown in the previous section. For each syntactic pattern, the image feature vector is calculated and fed as an integer to the neural network. In the proposed methodology, 3D Alex-Net is used for classification purpose. Alex-Net is implemented in 2012 by Alex Krizhevsky[188]. In the Alex-Net, eight layers are

present, in which 5 are convolution layers and 3 are fully connected layers. 1st convolution layer is followed by the first max-pooling layer and the same for the second convolution layer. 3rd, 4th and 5th layers are connected on consecutive positions and then the 5th convolution layer is connected to the 3rd max pooling layer.

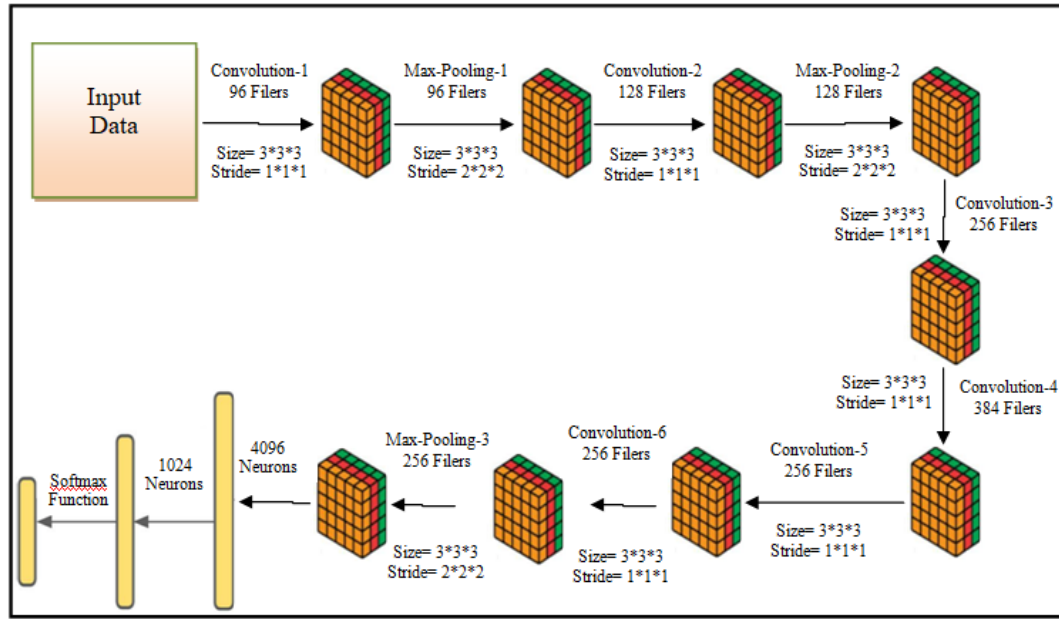


Fig. 6.2: 3D Alex-Net Architecture

The output generated by the 3rd max pooling layer will become the input to the next two fully connected layers. The third fully connected layer will now be input for the soft-max classifier and the Softmax layer consists of three class labels. As we have done modification in the traditional Alex-Net architecture and added one conventional layer in the architecture. The architecture of improved 3D Alex-Net is represented in **Fig. 6.2**. To implement 3D Alex-Net the minimum computer configuration must contain python tensor flow libraries with Keras and NumPy packages. We used the 3.7 version of Python that supports the GPU version of tensor flow. We have done some modifications in the existing AlexNet architecture to improve the performance of the model. The detailed description of 3D Alex-Net architecture is given below:

- Convolution
- Max-pooling
- Dropouts

- Parametric ReLU
- Padding and Stride
- Fully Connected Layer
- Softmax Function
- ADAM Optimizer

The description of improved 3D Alex-Net architecture is explained below:

- **Convolution**

The proposed system consists of repeated application of 3x3x3 of the convolution operation. Each followed by a parametric ReLU operation.

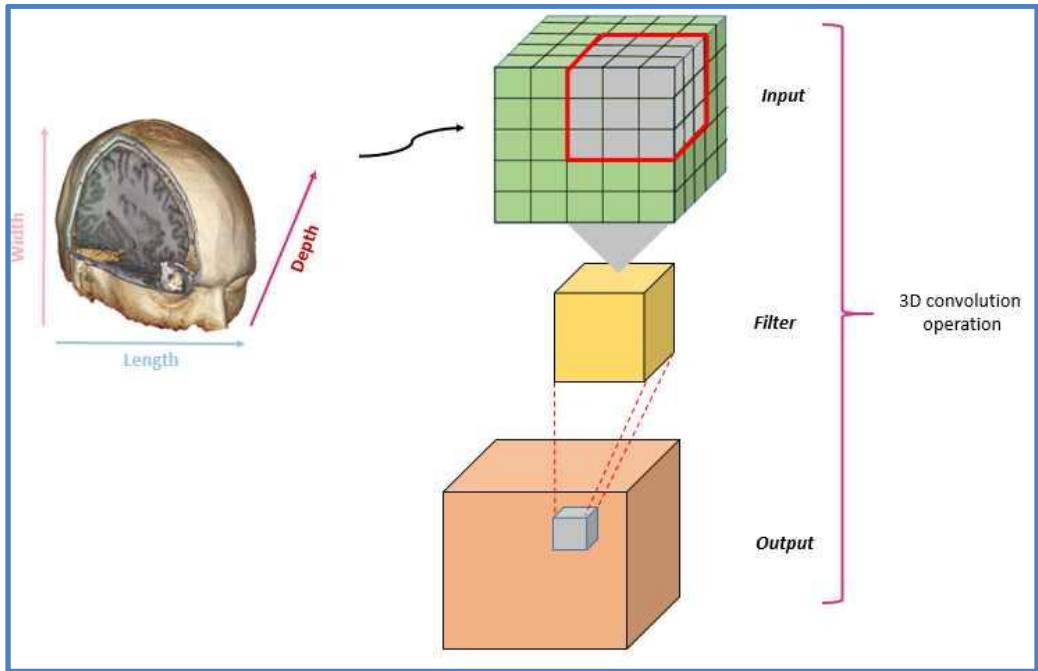


Fig. 6.3: 3D Convolution

The number of filters in the proposed network will change from 96-384. As we move down in the down-sampling, the depth of the layers will increase. The convolution operation of f and g is defined as below.

$$(f * g)(t) = \int_{-\infty}^{\infty} f(T)g(T - r)dr \quad (6.1)$$

The 3D convolution and convolution operation is depicted in **Fig. 6.3** and **Fig. 6.4**

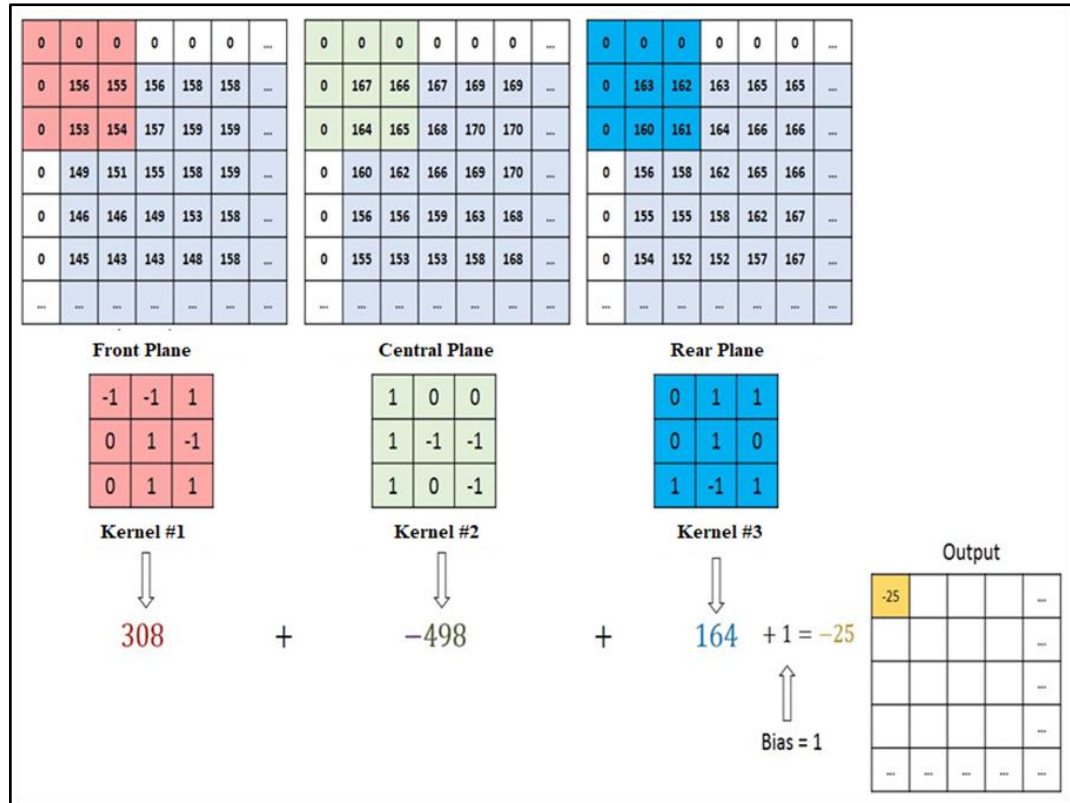


Fig. 6.4: Convolution Operations

- **Max Pooling Operation**

Max pooling operation is performed after convolution, which picks the maximum value from the feature map. For example, $\text{Max}(0.0, 0.0, 1.0, 2.0) = 2.0$. It highlights the most present value in the feature map. We applied the 'max-pooling 3D' operation to perform max pooling. With the stride of 2x2x2, the max-pooling 3D operation returns the horizontal max-pooling operation, which returns half of the feature map. This operation works on every depth slice of the input feature map and resizes it. The pictorial representation of 3D max-pooling is represented in **Fig. 6.5**. The spatial max pooling can be derived as,

$$Y_{x,y}(X) = \max_{i,j=0}^1 x_{2x+i,2y+j} \quad (6.2)$$

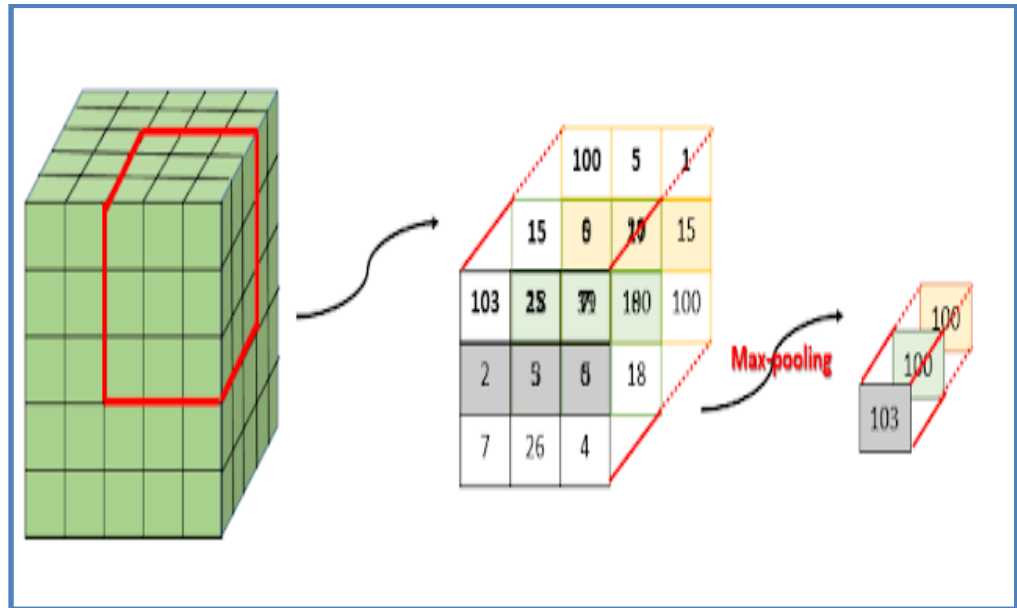


Fig. 6.5: 3D Max-pooling Operation

- **Parametric ReLU**

Instead of taking ReLU, we used parametric ReLU to challenge the dying ReLU problem. The parameterized ReLU function is employed when the leaky ReLU function fails to solve the problem of dead neurons and the relevant data is not successfully passed to the next layer. As we are working only on edges of the images hence the neurons of few layers are pushed on to an inactive state. Because the proposed methodology's learning rate is too low, most neurons become locked in the dead state. This lowers the model's performance even more.

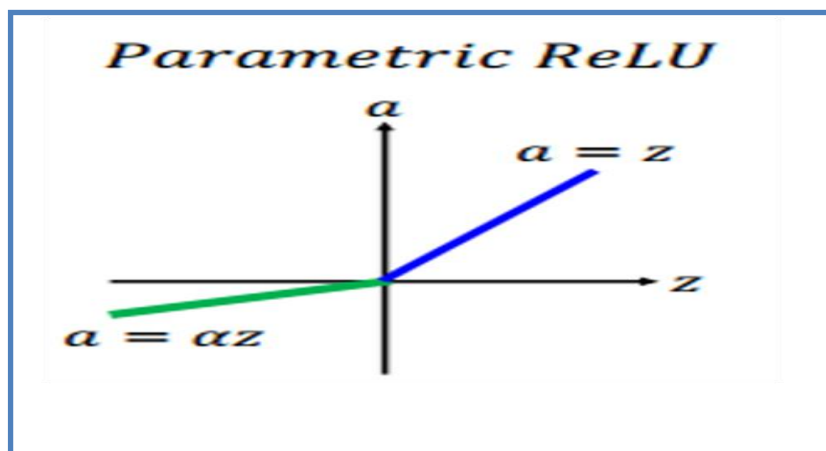


Fig. 6.6: Parametric ReLU

Parametric ReLU is a modified version of ReLU that aims to address the problem of the gradient of the left half of the axis becoming zero. As its argument a , this function returns the slope of the negative portion of the function. Through back propagation, the optimal value of a is determined. It will allow a slight gradient when the unit is inactive and it is shown in the **Fig. 6.6**. The parametric ReLU can be denoted as,

$$f(x) = \max(ax, x) \quad (6.3)$$

Here a is a learning parameter whose will dynamically change based on the activation function.

- **Padding**

When numerous convolution procedures are performed simultaneously, the resulting image may become vanishingly small due to the image shrinking with each operation. Padding is the process of including empty pixels around the edges of an image. Padding is used to maintain the original size of an image while applying a convolutional filter on the image. It allows the filter to execute full convolutions on the image's edge pixels and it also preserves the corner pixels of the image. The formula for calculating the output image size is as follows:

$$N_{out} = \frac{N_{in} + 2P - F}{S} + 1 \quad (6.4)$$

Where

N_{out} = Size of output features

N_{in} = Size of input features

P = Size of padding

F = Size of Filter

S = Convolution stride size

In this research, in order to make sure the output has the same shape as the input image, the padding value of 'same' is used as a parameter in Conv3D and it calculates and adds the amount of padding needed to make sure that the output image has the same shape as its input.

- **Dropout**

Dropout is implemented per convolution operation except for the output convolution operation. It drops the hidden layers in the network to overcome over-fitting data in a neural network. The dropout value considered as 0.5 here.

- **Fully Connected Layer**

The fully connected layer takes a vector as its input. There are a lot of filters that are made by convolution. Each one is made up of a grid. Flattening specifies a function that maps these filters to a vector; hence we can go back through the convolutional layers and see if there are any errors. It takes the pooled feature map that was made in the pooling step and turns it into a one-dimensional vector.

- **Softmax Function**

Softmax function is said to be a combination of several sigmoid. It is used to calculate the relative probability and softmax function gives the probability of each class. Most often, it is used as an activation function for the last layer of a neural network when it is used for multi-class classification. In the proposed work, softmax function is used as multiclass classification. Mathematically it can be represented as:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (6.5)$$

Where

σ = Softmax function

Z = Input vector

e^{z_i} = exponential function for input vector

K = No. of classes

e^{z_j} = exponential function for output vector

- **ADAM Optimizer**

Adaptive moment optimizer employs squared gradients similar to the RMS, but generates moving averages. Due to the fact that the neural network's learning rate may be chosen at will, the technique is easy to implement. In the proposed system, the

starting moving averages for the mini batch and beta of Equation (6.6) and Equation (6.7) were set to 0.9 and 0.999, respectively:

$$m_i = \beta m_{i-1} + (1 - \beta) g_i \quad (6.6)$$

$$v_i = \beta_2 v_{i-1} + l' - \beta_2 g_i^2 \quad (6.7)$$

Where m and v is mean and variance of the moving averages of mini-batch and beta, respectively. These values represent the initial and secondary moments. The weights of the corrected moments can be determined using the equation 6.8.

$$w_t = w_{t-1} - \eta \frac{m_i}{\sqrt{v_i + \epsilon}} \quad (6.8)$$

6.3 Analysis of Results

6.3.1 Datasets

In this research, we used four different brain MRI datasets. The first dataset is ‘Figshare brain tumor dataset’ which is publicly available [173]. This dataset consists of a total of 3064 images of 233 different patients. The second dataset is again ‘Kaggle brain MRI dataset’ which is publicly available on Kaggle[174]. This dataset consists of a total of 253 MRI images. The third dataset is the medical MRI dataset and it is collected from the pentagram research institute, Hyderabad. The dataset contains 2D slices of MRI, which can be combined to make a 3D view of the images. The fourth dataset is BraTS 2019 dataset[175]. This dataset consists of images of four different contrasts. The dataset contains four folders. The folder structure is composed of FLAIR (Fluid Attenuated Inversion Recovery), T1, T1-contrast enhanced and T2 .

To build multi-orientation or view of each 3D model, we rotated the dataset equally in horizontal direction and finally produced the dataset on 4th and 7th depths and multi-orientation (3, 6, 9, 12, 18, 24, and 30 rotations) occurs only on depth 7th. Each 3D model's multi-orientation is applied to both the training and testing datasets individually. To train the deep learning model, Keras with TensorFlow libraries has been used. The specification of the system is windows operating system, GeForce GTX TITAN X. We developed a GUI based system for better analysis purpose. GUI of the proposed methodology is shown in **Fig. 6. 7**.

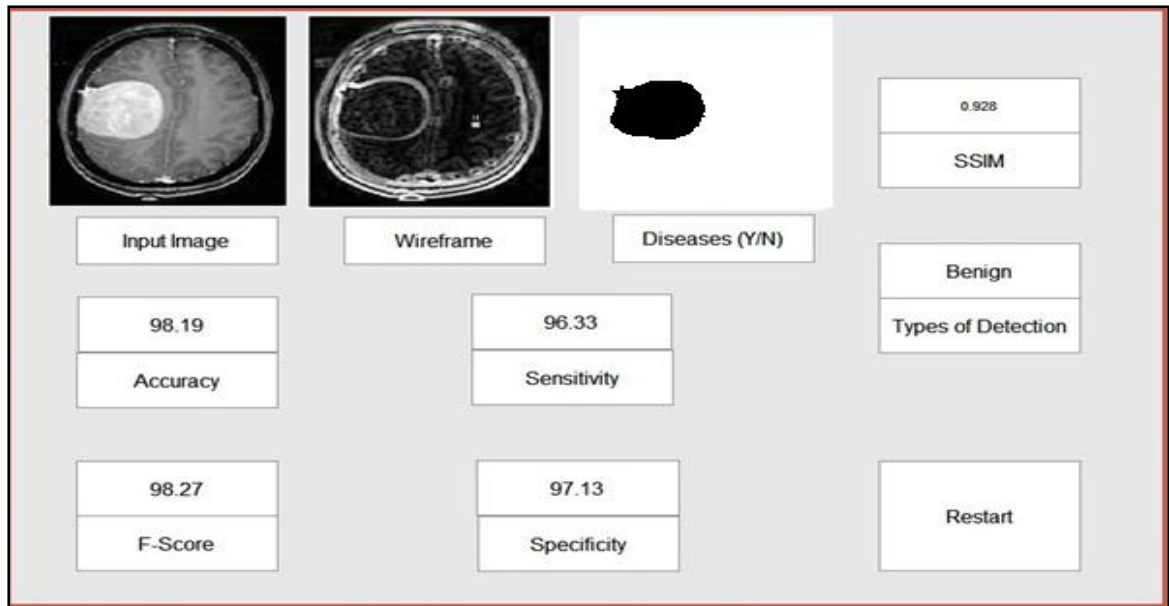


Fig. 6.7: GUI of Proposed Work

During the brain tumor classification, 3DAlex-Net architecture is used, which requires values of different hyper-parameters to be set. This is required to obtain the optimum performance of the architecture. These hyper-parameters are epochs, learning rate, and dropout, and batch size. The optimized values of these hyper-parameters for each problem are mentioned in **Table 6.2**. The epoch is 50 and batch size is considered 128. The early stopping property has been introduced in the training process, which stops the learning process at certain epochs, in case the accuracy and loss are constant for continuous four epochs. The validation split is set to 0.1. This parameter allows the model to handle train/test data splitting on its own. 0.1 splits the data by splitting the sample dataset into 90% of training and 10% to test the samples. This allows plotting the loss, accuracy values correctly while plotting them.

The dropout technique is used to ignore a random set of neurons during the training is given as 0.50. The dropout set to 50% means three in 6 inputs is excluded from each cycle. The learning rate ranges from 0 to 1. An excessively high learning rate can cause the model to converge too quickly to a suboptimal solution. When the learning rate is very low, the model becomes stuck in the training phase. At the same time, a slow learning rate can lead the process to halt

Table 6. 2: Details of Improved 3D Alex-Net Classifier

Parameter Name	Values
Epochs	50
Dropout	0.50
Batch Size	128
No. of Convolution	6
No. of Fully Connected Layer	3
Pooling Layer	Max Pooling
Pooling Layer Window Size	2

6.3.2 Evaluation Parameters

Accuracy, sensitivity, recall, specificity, and F1-score parameters are used to assess the efficacy of the proposed methodology (**Table 6.3**). Accuracy cannot be the only factor used to evaluate the performance of the proposed work. As a result, we also took additional factors into account. Recall is crucial in the classification of brain tumours. Recall indicates how accurately a model can identify a positive class as such.

Table 6.3: List of Evaluation Parameters and their Formula

S. No.	Parameters Name	Formula
1	Accuracy	$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)}$
2	Precision	$Precision = \frac{TP}{TP + FP}$
3	Recall	$Recall = \frac{TP}{TP + FN}$
4	Specificity	$Specificity = \frac{TN}{TN + FP}$
5	F1- Score	$F1\ Score = 2 * \frac{precision * Recall}{Precision + Recall}$
TP= True Positive, TN= True Negative, FP=False Positive, FN = False Negative		

6.3.3 Results

This section represents the comparative analysis of the classification results of the proposed method with the existing methods over the above mentioned datasets. For the classification of different types of tumors, various hyper-parameters were used for the training of the proposed model. The obtained results are evaluated using different quantitative measures which are mentioned in **Table 6.3**. As we worked on the validation accuracy and loss hence the results of validation accuracy, validation loss, accuracy and loss are shown in the **Fig 6.8**, **Fig. 6.13**, **Fig. 6.15** and **Fig. 6.16** on all four datasets. As the number of epochs goes up, validation accuracy goes up and validation loss goes down. This shows that the model was set up correctly. The performance of the proposed method is evaluated on the Figshare dataset, and Brain MRI Kaggle dataset, Medical MRI dataset and BraTS 2019 dataset. The evaluation parameters are accuracy, sensitivity, specificity, precision, recall and F1-score. Most of the authors have worked on the average accuracy and they have not identified class wise accuracy. In this research, we identified classwise accuracy and later we identified average accuracy. The class-wise accuracy of the Figshare dataset is 99%, 99.34%, 99.26% on glioma, meningioma, and pituitary classes. In the similar way, we calculated precision, recall and F1-score values.

Table 6.4 shows the comparative analysis of the classification results obtained by existing methods on the Figshare dataset [173]. As per the comparative analysis, till date the highest accuracy is obtained using Fine-tune VGG-16[193] model. Authors have used fine-tuning transfer learning to improve the efficiency of the architecture. Proposed method achieved highest accuracy on Meningioma class, highest recall and specificity is achieved on glioma class and highest precision and F1-score value on pituitary class. As per comparative analysis, proposed method performed better than existing methods in terms of accuracy, precision, recall, specificity and F1-score. **Fig. 6.9** and **Fig. 6.10** shows the graphical representation of the proposed and existing method results on the Figshare dataset. Some of the authors have only worked on the accuracy parameter. **Table 6.5** and **Fig. 6.11** shows the same result analysis of proposed work with other state-of-art methodology on Figshare dataset. Some authors

have used RCNN based method for the classification of the brain tumor and **Table 6.6** shows the comparative analysis of the existing method on the Figshare dataset. The evaluation parameters are Accuracy and sensitivity. As per the comparative study, the proposed method gives better results in terms of accuracy on the Figshare dataset. **Fig. 6.12** shows the graphical representation of the proposed and existing method results on the Figshare dataset. For a better evaluation of the proposed method, we performed the experiments on three other datasets.

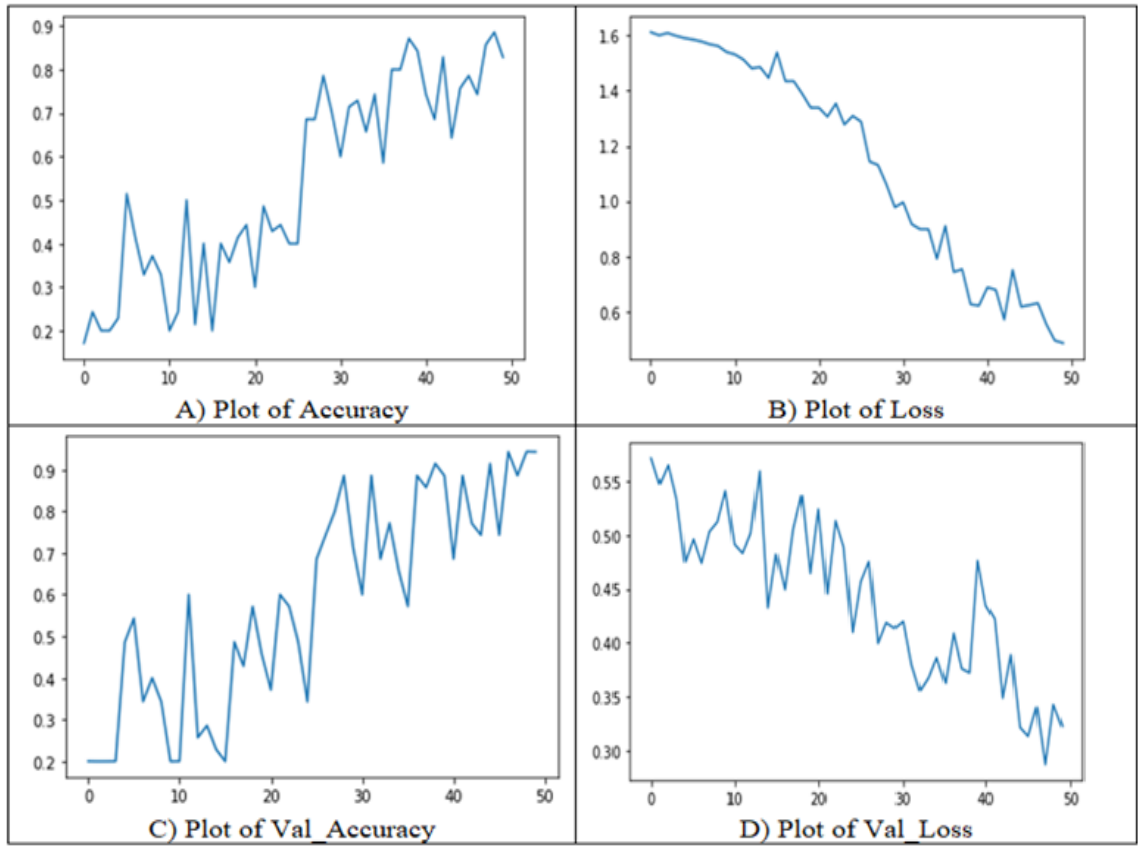


Fig.6. 8: Py-Plots of Accuracy, Loss, Val_Accuracy, Val_Loss on Figshare Dataset

The accuracy of the proposed method on the Kaggle Brain MRI dataset is 99.25%. As per the comparative study, Mask RCNN performs better with an accuracy of 98.00% but our proposed system has 1.25% higher accuracy rate than the existing system. Some other evaluation parameters are used for the validation of the proposed work. The result of the comparative analysis is shown in **Table 6.7**. The graphical representation of comparative analysis are shown in **Fig. 6.14**. The accuracy of the proposed method is also evaluated on the Medical MRI dataset and it is 99.56%. Results are shown in the **Table 6.7**.

Table 6.4: Result Analysis of Proposed Work with other State-of-Art Methodology on Figshare dataset.

Authors	Method	Classes	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	F1-Score(%)
Arisha Rehman et al.[193]	Freeze AlexNet	-	91.21		90.48	95.48	-
Arisha Rehman et al.[193]	Freeze Google Net	-	93.46		95.38	97.74	-
Arisha Rehman et al.[193]	Freeze VGG16	-	89.76		87.81	94.64	-
Pashaei et	CNN	-	93.68		-	-	93.00
Milica M.Badza et al.[162]	CNN	-	95.40		-	-	94.94
Vimal Kurup et al.[163]	Capsulenet	-	92.60		-	-	93.33
Srinivasan et al.[195]	GLCM& Wavelet packets	-	93.30		-	-	72.00
Proposed Method	3D AlexNet	Glioma	99.00	99.45	98.99	98.91	99.21
		Meningioma	99.34	99.69	98.78	99.45	99.23
		Pituitary	99.26	99.57	98.97	98.47	99.26
		Average	99.20	99.57	98.91	98.94	99.23

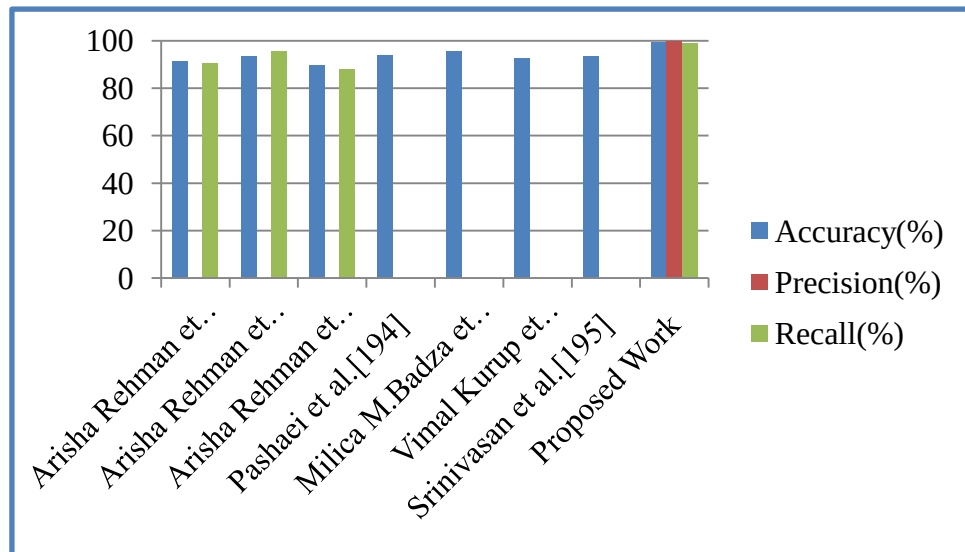


Fig. 6.9: Comparative Analysis of Accuracy, Precision and Recall on Figshare Dataset

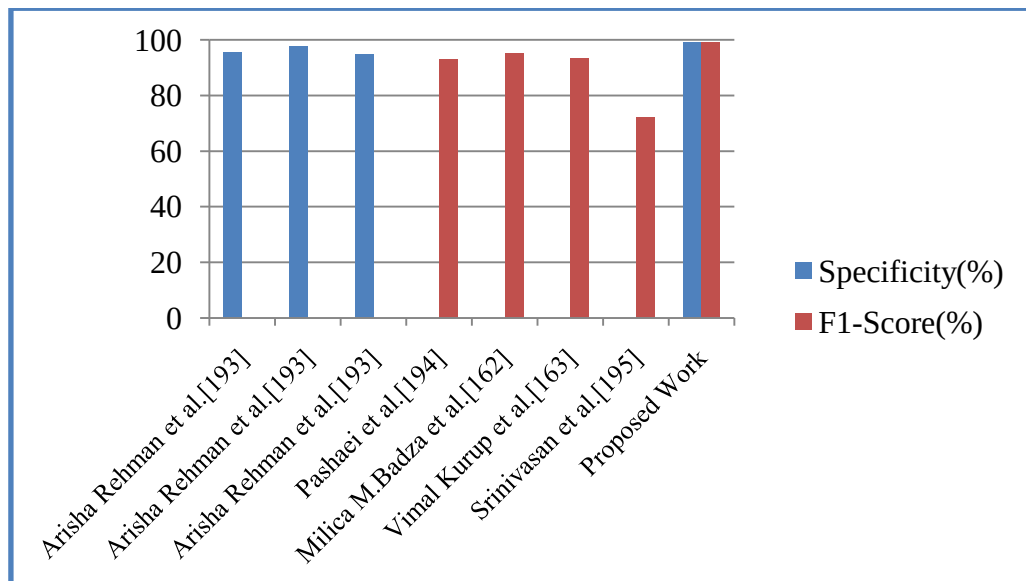


Fig. 6.10: Comparative Analysis of Specificity and F1-Score on Figshare Dataset

Table 6.5: Result Analysis of Proposed Work with other State-of-Art Methodology on Figshare Dataset

Authors(Year)	Method	Accuracy(%)
Cheng et al.[197] (2015)	SVM	87.54
Dheng et al.[198] (2009)	SVM	94.68
Abir et al.[199](2018)	PNN	83.33
WIdhiarso et	CNN	82.00

Afshar et al.[201] (2018)	CapsNet	86.56
Abiwinanda et al.	CNN	84.19
Ismael and	MPNN	91.90
Pashaei et al. [194](2018)	ELM	93.68
Deepak et al.[204](2019)	GoogleNet and SVM	97.10
Swati et al.[159](2019)	VGG 19	94.82
Gumaei et al.[47] (2019)	CNN based on complex networks	94.93
Huang et al.[46] (2020)	GIST descriptor and ELM	95.49
Arisha Rehman et al.[193] (2020)	Attention module, Hyper column technique, Residual block	97.39
Arisha Rehman et al.[193] (2020)	Fine-tune AlexNet	98.04
Arisha Rehman et al.[193] (2020)	Fine-tune GoogleNet	98.69
N. Chakrabarty et al.[174] (2021)	Fine-tune VGG16	97.69
Proposed Methodology	3D AlexNet	99.20

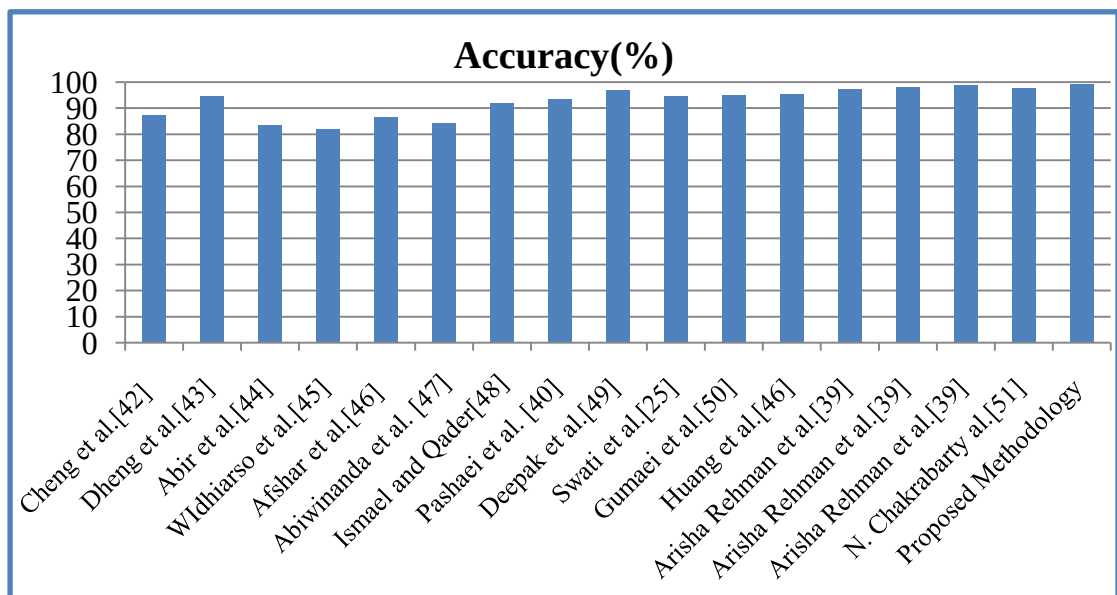


Fig. 6.11: Comparative Analysis of Network Parameters of Existing Systems and Proposed System on Figshare Dataset

Table 6.6: Performance Comparison of the Proposed Method with Existing RCNN Based Methods on Figshare Dataset

Authors	Evaluation Metrics	
	Accuracy %	Recall %
Rayahan et al.[205]	92.0	95.0
S. Ren et al. [206]	94.0	94.0
Momina Masood et al. [161]	95.9	95.3
Momina Masood et al.[161]	96.3	95.3
Proposed Method	99.20	98.91

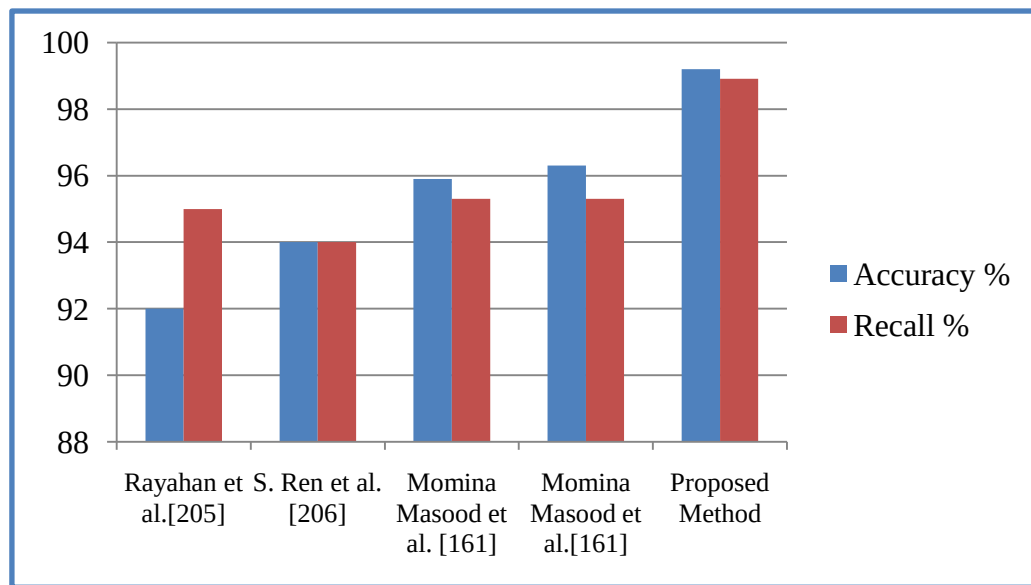


Fig. 6.12: Comparative Analysis of the Proposed Method with RCNN Based Approach on Figshare Dataset

The proposed work performance was assessed using the BraTS 2019 dataset. The average accuracy of the proposed approach on the BraTS 2019 dataset is shown in **Table 6.8** and **Fig. 6.17**. The proposed method is compared to seven other current methods in this part, and all of the researchers focused on two types of tumors: LGG and HGG tumors. Zhuge et al. [62] used Deep CNN for the classification of tumor using T1, T2, T2-FLAIR images and achieved the impressive accuracy of 97.1%. The proposed method utilized 3D Alex-Net architecture for tumor classification and used

T1,T1ce,T2,T2-FLAIR images and achieved a 97.12% accuracy, which was 0.02% higher than the previous method [210].

Table 6.9 displays the precision, recall, specificity and F1-score performance of the various methods by class. The proposed 3D Alex-Net had the maximum precision of 99.89% for healthy glioma. Proposed 3D Alex-Net approach attained the maximum precision of 94.87% and 94.64% for low grade glioma and low grade glioma, respectively, which is greater than the existing ResNet mixed convolution method. Similarly, the new technique outperforms the existing one in terms of Recall for LGG and HGG classes, but in the case of healthy participants, proposed method's recall value is 96.28%, which is 3.22% higher than the pre-trained ResNet Mixed convolution method. The best model, as with LGG, HGG, and healthy individuals, is ResNet Mixed convolution, which has the highest precision, recall, specificity and F1-score values. The proposed method improves the average recall rate. The suggested approach's F1-score value for LGG and HGG classes is higher than the existing method, while ResNet 3D achieved the highest F1-score value in the categorization of healthy participants. The detailed representation on the BraTS 2019 dataset is graphically represented in **Fig. 6.18**.

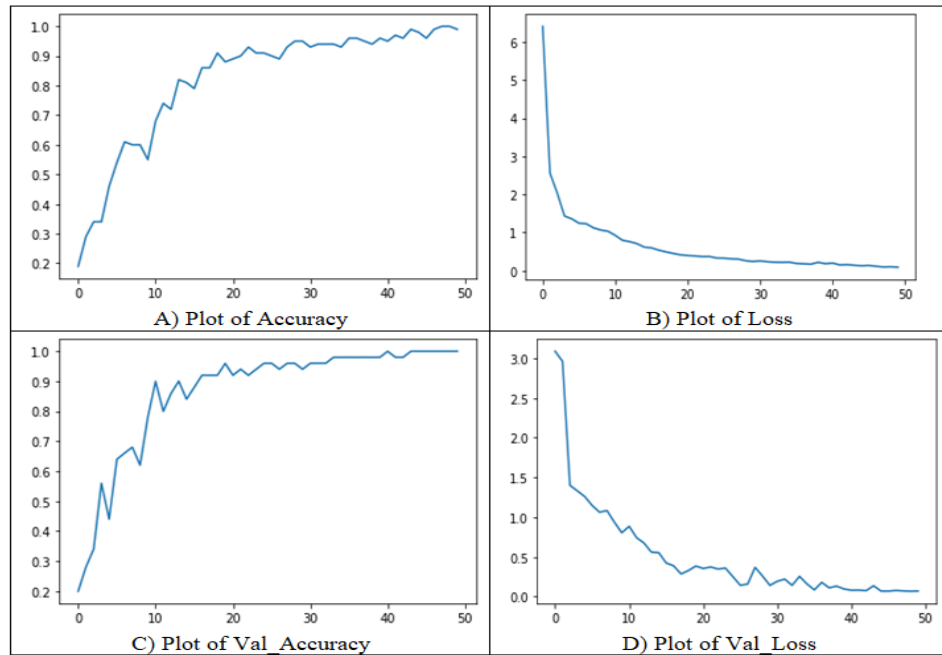


Fig. 6.13: Py-Plots of Accuracy, Loss, Val_accuracy, Val_loss on Kaggle brain MRI dataset

Table 6.7: Result Analysis of Proposed Work with other State-of-Art Methodology on Brain Tumor Kaggle Dataset and Medical MRI Dataset

Authors	Method	Accuracy %	Precision %	Recall %	Specificity %	F1-Score %
Swati et al.[159]	VGG-19	94.0	-	-		-
Momina Masood et al. [161]	Mask-RCNN	98.0	-	-		-
Hassan et al.[160]	ResNet-50	89.0	87.0	93.0		90.0
Hasan et al.[160]	VGG-16	75.0	77.0	71.0		74.0
Proposed work on Kaggle Brain	Inception-v3	99.25	97.38	94.47	95.52	95.90
Proposed work on Medical MRI	3D Alex-Net	99.56	98.51	95.74	96.63	97.10

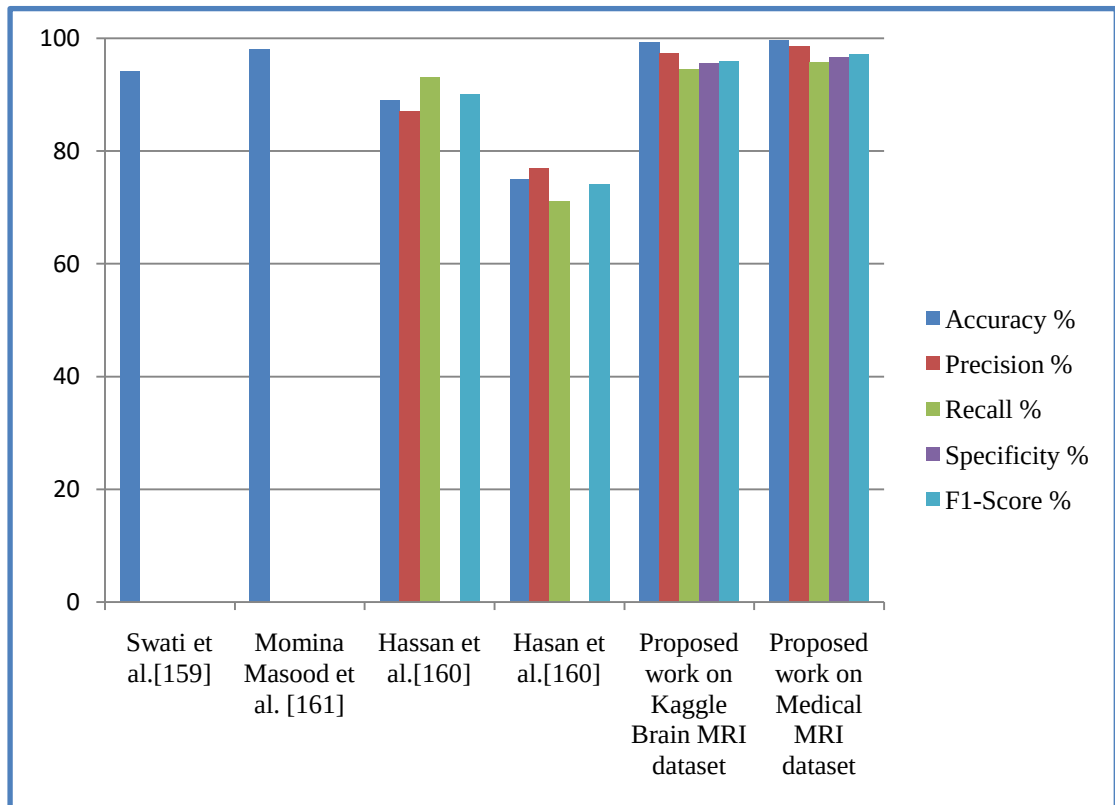


Fig. 6.14: Comparative analysis of proposed Work with Existing State of Art Method on Kaggle Brain MRI dataset and Medical MRI Dataset

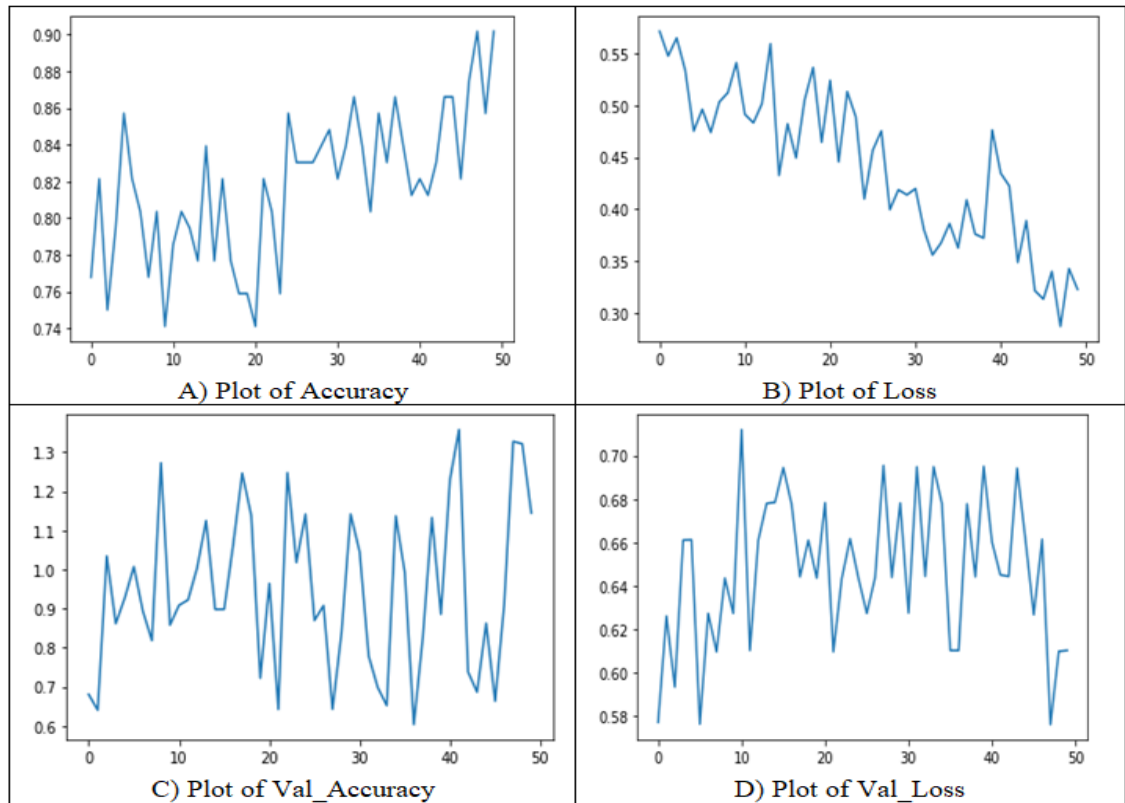


Fig. 6.15: Py-Plots of Accuracy, Loss, Val_Accuracy, Val_Loss on Medical MRI Dataset

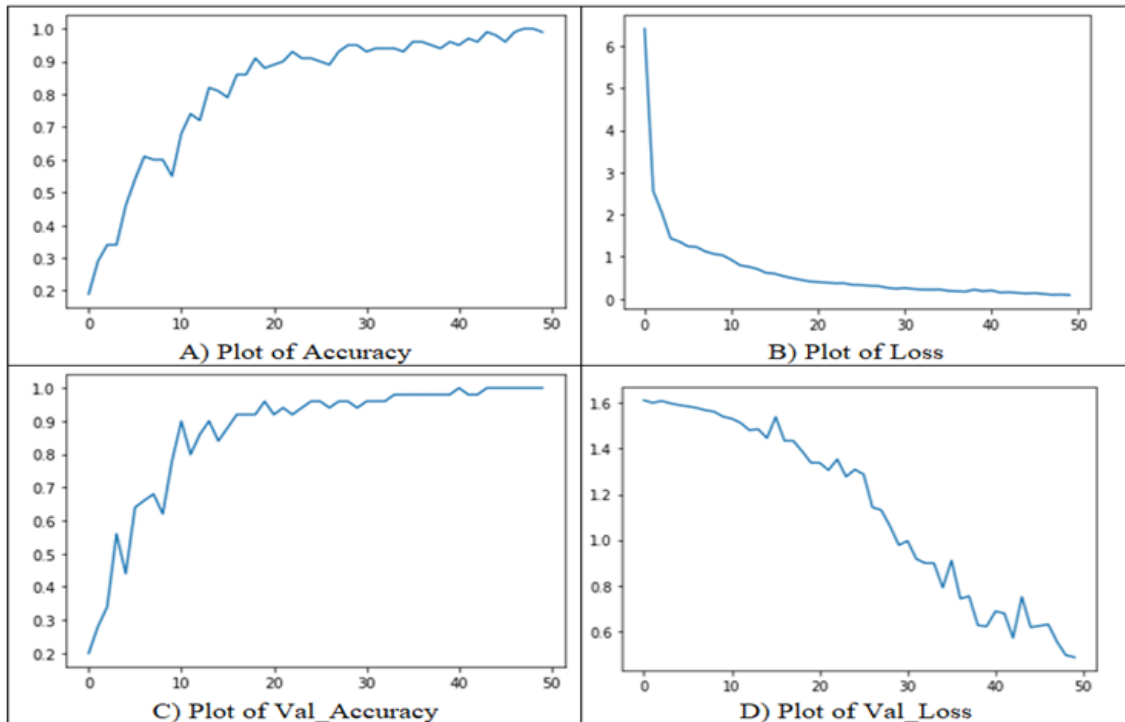


Fig. 6.16: Py-Plots of Accuracy, Loss, Val_Accuracy, Val_Loss on BraTS 2019 Dataset

Table 6.8: Result analysis of proposed work on BraTS 2019 Dataset

Authors	Method	Contrast	Accuracy(%)
Shahzadi et al.[58]	CNN with LSTM	T2-FLAIR	84.00
Pei et al.[151]	CNN	T1,T1ce,T2,T2-FLAIR	74.90
Ge et al.[152]	Deep CNN	T1,T2,T2-FLAIR	90.87
Mzoughi et al.[150]	Deep CNN	T1,T1ce,T2,T2-FLAIR	96.59
Ouerghi et al.[208]	Random Forest	T1,T2,T2-FLAIR	96.50
Soumick Chatterjee et al.[209]	Pre-trained ResNet Mixed Convolution	T1ce	95.98
Zhuge et al.[210]	Deep CNN	T1, T2, T2-FLAIR	97.10
Proposed	3D AlexNet	T1,T1ce,T2,T2-FLAIR	97.12

Table 6.9: Result Analysis of Proposed Work with other State-of-art Methodology on BraTS 2019 Dataset

Authors	Classes	Precision %	Recall %	Specificity %	F1-Score %
ResNet 3D[209]	Low grade Glioma	79.80	92.0	90.70	85.40
	High grade Glioma	93.30	82.80	96.20	87.70
	Healthy subjects	99.30	99.00	99.00	99.0
Pre-trained ResNet 3D [209]	Low grade Glioma	78.10	85.50	91.10	81.40
	High grade Glioma	89.60	83.50	93.70	86.30
	Healthy subjects	100	99.90	100	99.90
ResNet(2+1)D [209]	Low grade Glioma	78.60	91.40	90.20	84.40
	High grade Glioma	93.0	82.20	96.20	87.30
	Healthy subjects	99.4	99.00	99.70	99.20

Pre-trained ResNet(2+1)D [209]	Low grade Glioma	84.10	91.00	93.10	87.30
	High grade Glioma	92.80	87.00	95.90	89.70
	Healthy subjects	99.90	99.80	99.90	99.90
ResNet Mixed Convolution [209]	Low grade Glioma	74.70	88.60	85.50	77.30
	High grade Glioma	91.10	75.00	96.30	82.30
	Healthy subjects	99.40	97.60	99.70	98.50
Pre-trained ResNet Mixed Convolution [209]	Low grade Glioma	86.30	93.10	91.20	89.40
	High grade Glioma	94.40	88.30	93.60	91.20
	Healthy subjects	99.70	99.50	99.60	99.60
3D AlexNet	Low grade Glioma	94.87	94.85	96.35	93.53
	High grade Glioma	94.64	90.48	99.89	92.51
	Healthy subjects	99.89	96.28	98.05	98.05

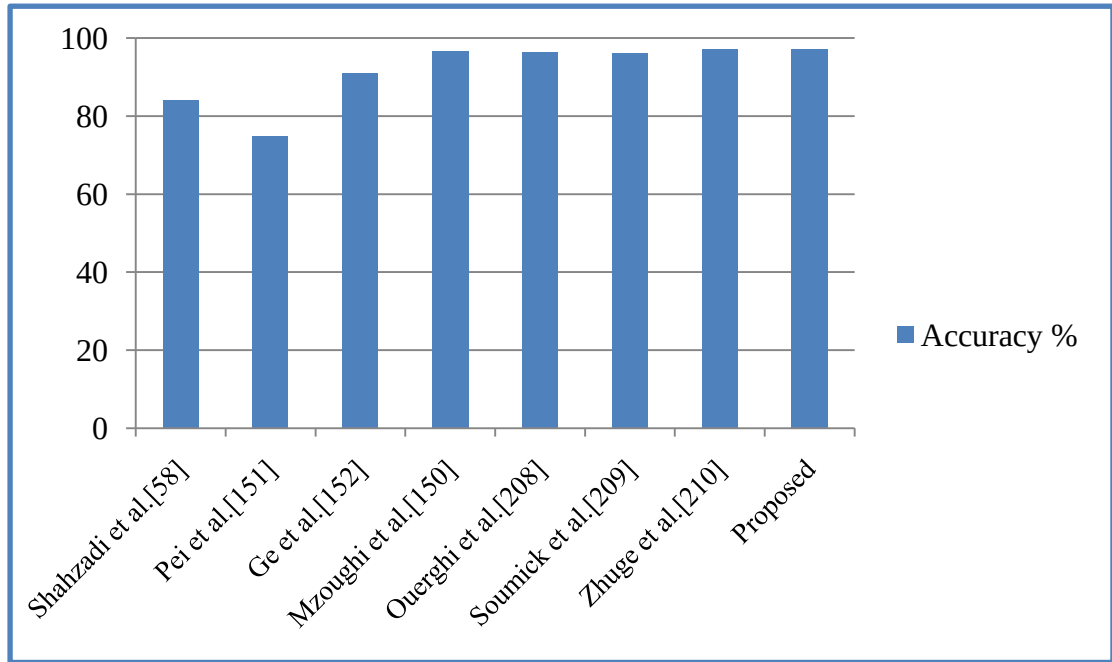


Fig. 6.17: Comparative Analysis of Accuracy on BraTS 2019 Dataset

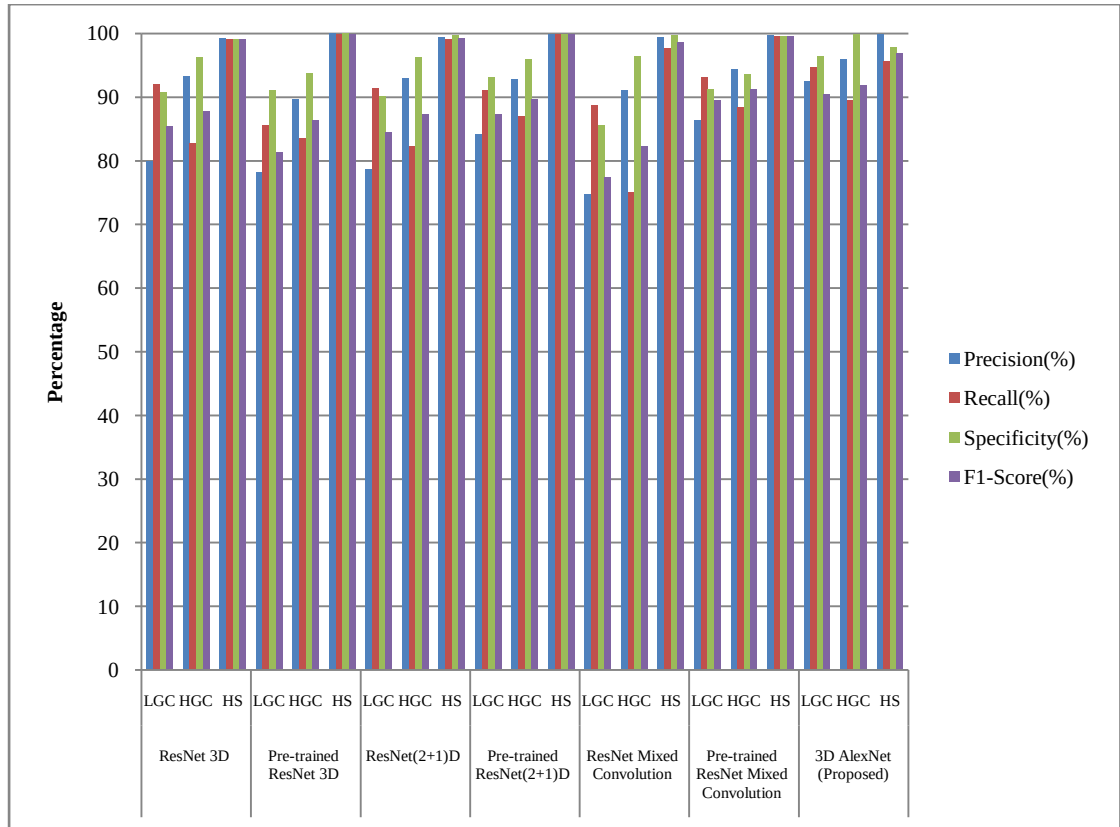


Fig. 6.18: Comparative Analysis of Proposed Work on BraTS 2019 Dataset

6.4 Summary

In this study, we used 3D Alex-Net architecture to detect and classify brain tumors. A wireframe model and a syntactic pattern recognition approach were used to create a tumor detection and classification system. Four standard brain tumor datasets were considered to assess the performance of the proposed model. On the Figshare dataset, Kaggle Brain MRI dataset, Medical MRI dataset, and BraTS 2019 dataset, the proposed method's accuracy is 99.20%, 99.25%, 99.56%, and 97.12%, respectively, with recall values of 98.91%, 94.47%, 95.74%, and 93.87%. Recall is important parameter to evaluate the performance of the proposed work. As per the comparative analysis, the Recall value is improved on the Figshare, Kaggle, and BraTS 2019 datasets. But there are still ways to improve the recall value. The results of the experiments show that the proposed Brain tumor classification model is better than the state of art methods.

CHAPTER-7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

This chapter summarized the major contributions of the overall research project and made recommendations for further research. The fact that generalization is one of the most difficult and valuable skills a computer may have is the major reason for our interest in this topic. It is a challenging task because generalization is likely more crucial in computer vision than in any other field. As a result, this work made three major contributions.

The first contribution of this study is the creation of an innovative and reliable algorithm for recognizing volumetric information in 3D images. The entire study focuses on the representation of the 3D image wireframe model. To begin with, image noise is removed and image quality is improved using pre-processing techniques. A unique approach is developed on the basis of cellular logic array processing to extract volumetric information from three dimensional images and to identify the wireframe model of the 3D images. The suggested algorithm's results were compared to existing edge detection techniques in terms of processing time, time complexity, and information preservation for the analysis. According to the results's analysis, the proposed methodology is more reliable, fast, and accurate than state-of approaches.

The research's second contribution is the introduction of a syntactic pattern recognition-based approach for representing images in textual form. A novel algorithm is introduced that takes an image as input and extracts the image's volumetric properties before converting it to textual form. In addition, a unique approach is shown that can rebuild an image using a knowledge vector. Reconstruction allows us to investigate the interior details of 3D images, such as the object's size, form, and structure and in the world of clinical image processing, this could help us advance. The suggested algorithm performance is assessed on a medical imaging dataset, and the findings are out performed in real-time. This method is superior to other methods since it involves only a few steps in the reconstruction of a 3D image, making it unique.

The research's third contribution is the development of a unique tumor classification method using improved 3D Alex-Net architecture. The proposed framework is more efficient and robust than existing state-of-the-art approaches, according to the results analysis.

Four standard brain tumor datasets were considered to assess the performance of the proposed model. On the Figshare dataset, Kaggle Brain MRI dataset, Medical MRI dataset, and BraTS 2019 dataset, the proposed method's accuracy is 99.20%, 99.25%, 99.56%, and 97.12%, respectively, with recall values of 98.91%, 94.47%, 95.74%, and 93.8%. Experiments demonstrate that the proposed brain tumor classification model outperforms existing state-of-the-art approaches in terms of accuracy and precision.

7.2 Future Scope

The following points could be considered for the future work in this direction:

- In future, wireframe model could be used for Object localization and pose estimation.
- In the current research, the convex polyhedrons is carried out in the neighborhood generation of convex polyhedrons could be carried out in the neighborhood of 5x5x5, 7x7x7, 9x9x9 and so on.
- 3-D processing algorithms presented in this report can be suitably modified for processing 3-D video i.e., streaming sequence of 3-D data.
- Furthermore, the proposed system could combine with the existing cancer diagnosis tools to improve the robustness of the existing CAD based systems.
- The proposed system can be used to investigate the changes in the volume of intracranial structures in fetuses with congenital heart disease (CHD) by three-dimensional volumetric MRI in the earlier stages of pregnancy.
- The proposed system could be used for data transmission purpose.

BIBLIOGRAPHY

- [1]. Bishop, Christopher M., and Nasser M. Nasrabadi, "Pattern recognition and machine learning," New York, Springer, 2006, vol. 4, no. 4, pp. 738.
- [2]. Kilmister, C. W. "Mathematical logic by Ian Chiswell and Wilfrid Hodges," Oxford University Press, Oxford texts in logic, 2009, vol. 93, no. 528, pp. 574-575.
- [3]. Jain, Anil K., Robert P. W. Duin, and Jianchang Mao., "Statistical pattern recognition: A review," IEEE Transactions on pattern analysis and machine intelligence, vol. 22, no. 1, pp.4-37, 2000.
- [4]. Abiodun OI, Kiru MU, Jantan A, Omolara AE, Dada KV, Umar AM, Linus OU, Arshad H, Kazaure AA, Gana U, "Comprehensive review of artificial neural network applications to pattern recognition," IEEE Access vol. 7, no.1, pp.158820–158846, 2019.
- [5]. Ahmad T, Jameel A, Ahmad B, "Pattern recognition using statistical and neural techniques," In International Conference on Computer Networks and Information Technology, 2011 vol. 1, no. 1, pp. 87–91.
- [6]. Guo Y, Wan J, Lu M, Niu W, "A parts-based method for articulated target recognition in laser radar data," Optik-International Journal for Light and Electron Optics vol. 124, no. 17, pp. 2727-2733, 2013.
- [7]. Lei Y, Bennamoun M, Hayat M, Guo Y, "An efficient 3D face recognition approach using local geometrical signatures," Pattern Recognition, vol. 47, no.2, pp.502–524, 2013.
- [8]. Mian A, Bennamoun M, Owens R, "On the repeatability and quality of keypoints for local feature based 3D object retrieval from cluttered scenes," International Journal of Computer Vision, vol. 89, no. 2, pp. 348-361, 2010.
- [9]. Sukno FM, Waddington JL, Whelan PF, "Comparing 3D descriptors for local search of craniofacial landmarks," In International symposium on visual computing, 2012, vol. 1, no. 1, pp. 92–103.
- [10]. Tombari F, Salti S, Di Stefano L, "Unique signatures of histograms for local surface description, " In European Conference on Computer Vision, 2010, vol. 2, no. 1, pp. 356–369.

- [11]. Creusot C, Pears N, Austin J, “A machine-learning approach to keypoint detection and land marking on 3D meshes,” *International journal of computer vision*, vol. 102, no. 1, pp. 146-179, 2013.
- [12]. Lai K, Bo L, Ren X, Fox D, “A large-scale hierarchical multi-view RGB-D object dataset,” In *IEEE International Conference on Robotics and Automation*, 2011, vol. 2, no. 4, pp. 1817–1824.
- [13]. B. Drost, M. Ulrich, N. Navab, and S. Ilic, “Model globally, match locally: Efficient and robust 3D object recognition,” In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, vol. 4, no. 2, pp. 998–1005.
- [14]. Castellani U, Cristani M, Fantoni S, Murino V, “Sparse points matching by combining 3D mesh saliency with statistical descriptors,” *Computer Graphics Forum*, vol. 27, no.2, pp. 643–652, 2008.
- [15]. Bronstein, Alexander M., Michael M. Bronstein, and Ron Kimmel, “Numerical geometry of non-rigid shapes,” *Springer Science & Business Media*, 2008.
- [16]. Bustos, Benjamin, Daniel A. Keim, Dietmar Saupe, Tobias Schreck, and Dejan V. Vranić, “Feature-based similarity search in 3D object databases,” *ACM Computing Surveys* , vol. 37, no. 4, pp. 345-387, 2005.
- [17]. Bayramoglu N, Alatan A, “Shape index SIFT: Range image recognition using local feature,” In *20th International Conference on Pattern Recognition*, 2010, vol. 1, no. 1, pp. 352–355.
- [18]. Paquet E, Rioux M, Murching A, Naveen T, Tabatabai, “A Description of shape information for 2-D and 3-D objects,” *Signal Process Image Commun* vol. 16, no. 1, pp.103–122, 2000.
- [19]. Osada R, Funkhouser T, Chazelle B, Dobkin D, “Shape distributions. *ACM Trans Graph*,” vol. 21, no. 4, pp.807– 832, 2002.
- [20]. Rusu R, Bradski G, Thibaux R, Hsu J, “Fast 3D recognition and pose using the viewpoint feature histogram,” In *IEEE International Conference on Intelligent Robots and Systems*, 2010, vol. 1, no. 1, pp. 2155–2162
- [21]. Shang L, Greenspan M, “Real-time object recognition in sparse range images using error surface embedding,” *International Journal of Computer Vision* , vol. 89, no. 2, pp. 211–228, 2010.

- [22]. Yamany SM, Farag AA, "Surface signatures: an orientation independent free-form surface representation scheme for objects registration and matching," IEEE transactions on pattern analysis and machine intelligence, vol. 24, no. 8, pp. 1105–1120, 2002.
- [23]. Mian A, Bennamoun M, Owens R, "Three-dimensional model-based object recognition and segmentation in cluttered scenes," IEEE transactions on pattern analysis and machine intelligence, vol. 28, no.10, pp. 1584–1601, 2006.
- [24]. Flint A, Dick A, Hengel A, "THRIFT: Local 3D structure recognition," In 9th International Conference on Digital Image Computing Techniques and Applications, 2007, vol. 1, no. 1, pp. 182–188.
- [25]. Lowe, David G., "Distinctive image features from scale-invariant keypoints," International journal of computer vision, vol. 60, no. 2, pp. 91-110, 2004.
- [26]. Lisin DA, Mattar MA, Blaschko MB, Learned-Miller EG, "Benfield MC Combining Local and Global Image Features for Object Class Recognition," In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops, San Diego, 2005, vol. 1, no. 1, pp. 47–47.
- [27]. Davis CS, "The video plankton recorder (VPR): design and initial results," Arch Hydrobiol Beih Ergebn Limnol, 1992, vol. 36, no. 1, pp. 67–81.
- [28]. Bariya P, Nishino K, "Scale-hierarchical 3D object recognition in cluttered scenes," In IEEE computer society conference on computer vision and pattern recognition, San Francisco, 2010, vol. 1, no. 1, pp 1657-1664.
- [29]. Alhamzi, Khaled, Mohammed Elmogy, and Sherif Barakat, "3D object recognition based on image features A survey," International Journal of Computer and Information Technology, vol. 3, no. 03, pp.651-660, 2014.
- [30]. R. Ramamoorthi, "Modeling Illumination Variation with Spherical Harmonics," In Face Processing: Advanced Modeling Methods, 2006, vol. 1, no. 1, pp. 385-424.
- [31]. P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar, "Acquiring the reactance field of a human face," In SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and

- interactive techniques, New York, NY, USA, ACM Press/Addison-Wesley Publishing Co., 2000, pp. 145-156.
- [32]. Choudhary S, Lakhwani K, Agrawal S, “An efficient hybrid technique of feature extraction for facial expression recognition using AdaBoost Classifier,” *International Journal of Engineering Research & Technology*, vol. 8, no. 1, 2012.
 - [33]. Knopp J, PrasadM, Willems G, Timofte R, Van Gool L, “Hough transform and 3D SURF for robust three-dimensional classification,” In *11th European Conference on Computer Vision*, 2010, pp. 589–602.
 - [34]. W. Maorui and M. Zhichun, “Research of 3D ear recognition based on local feature matching,” *Proceedings of the 31st Chinese Control Conference*, 2012, pp. 3779–3782.
 - [35]. Sun Y, Abidi MA, “Surface matching by 3D point's fingerprint,” In *8th IEEE International Conference on Computer Vision*, 2001, pp. 263–269.
 - [36]. Kumar DA, Sastry ASCS, Kishore PVV, Kumar EK, Kumar MTK, “S3DRGF: spatial 3-D relational geometric features for 3-D sign language representation and recognition,” *IEEE Signal Process Letters* vol. 26, no. 1, pp. 169–173, 2019.
 - [37]. Dong M, Chen Y, “Salient region detection and feature extraction in 3D visual data,” In *15th IEEE International Conference on Image Processing*, 2008, pp. 185–188.
 - [38]. Wang Z, Bao W, Lin D, Wang Z, “A local feature descriptor based on SIFT for 3D pollen image recognition,” *IEEE Access*, vol. 7, no. 1, pp. 152658–152666, 2019.
 - [39]. Chen H, Bhanu B, “3D free-form object recognition in range images using local surface patches,” *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1252–1262, 2007.
 - [40]. Guo Y, Soheli F, Bennamoun M, Lu M, Wan J, “TriSI: A distinctive local surface descriptor for 3D modeling and object recognition,” In *8th International Conference on Computer Graphics Theory and Applications*, 2013, pp. 86–93.

- [41]. Tang S, Wang X, Lv X, Han TX, Keller J, He Z, Skubic M, Lao S, “Histogram of oriented normal vectors for object recognition with a depth sensor,” In 11th Asian Conference on Computer Vision, 2012, pp. 525–538.
- [42]. Tombari F, Salti S, Di Stefano L, “Unique shape context for 3D data description,” In ACM Workshop on 3D Object Retrieval, 2010, pp. 57–62.
- [43]. Li X, Guskov I, “3D object recognition from range images using pyramid matching,” In 11th IEEE International Conference on Computer Vision, 2007, pp. 1–6.
- [44]. Zhong Y, “Intrinsic shape signatures: A shape descriptor for 3D object recognition,” In IEEE International Conference on Computer Vision Workshops, 2009, pp. 689–696.
- [45]. Drost B, Ulrich M, Navab N, Ilic S, “Model globally, match locally: Efficient and robust 3D object recognition,” In IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 998–1005.
- [46]. Taati B, GreenspanM, “Local shape descriptor selection for object recognition in range data,” Computer Vision Image Understanding, vol. 115, no. 5, pp. 681–694, 2011.
- [47]. Bariya P, Novatnack J, Schwartz G, Nishino K, “3D geometric scale variability in range images: features and descriptors,” International Journal of Computer Vision, vol. 99, no. 2, pp. 232–255, 2012.
- [48]. Aldoma A, Tombari F, Di Stefano L, Vincze M, “A global hypotheses verification method for 3D object recognition,” In European Conference on Computer Vision, 2012, pp.511–524.
- [49]. Guo Y, Sohel F, Bennamoun M, Lu M, Wan J, “Rotational projection statistics for 3D local surface description and object recognition,” In International Journal of Computer Vision, vol.105, no.1, pp. 63–86, 2013.
- [50]. Lou Z, Gevers T, Hu N, “Extracting 3D layout from a single image using global image structures,” IEEE Transaction Image Processing, vol. 24, no. 10, pp. 3098–3108.
- [51]. Naeimizaghiani M, Abdullah SNHS, Bataineh B, PirahanSiah F, “Character recognition based on global feature extraction,” In Proceedings of the

- International Conference on Electrical Engineering and Informatics, 2011, pp. 1–4.
- [52]. Fan Z, Li Z, Li W, You Y, Chen W, Li C, “A Combined Texture-Shape Global 3D Feature Descriptor for Object Recognition and Grasping,” In International Conference on Industrial Informatics Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), 2017, pp. 47–54.
 - [53]. Fookes C, Williams J, Bennamoun M, “Global 3D rigid registration of medical images. Proceedings,” In International Conference on Image Processing, 2000, vol. 2, pp. 447–450.
 - [54]. Pan X, Lu J, Liu F, “3D patch-based sparse learning for style feature extraction,” IEEE Access, vol. 7, no. 1, pp. 172403–172412, 2019.
 - [55]. Wang D, Cui C, Wu Z, “Matching 3D models with global geometric feature map,” In 12th International Multi-Media Modelling Conference, 2006, pp. 4–11.
 - [56]. Ganapathi II, Prakash S, 3D ear recognition using global and local features. IET Biometrics, vol. 7, no.3, pp. 232–241, 2018.
 - [57]. Sarwinda D, Bustamam A, “3D-HOG Features –Based Classification using MRI Images to Early Diagnosis of Alzheimer’s Disease,” In IEEE 17th International Conference on Computer and Information Science (ICIS), 2018, pp. 457–462.
 - [58]. Onishi K, Takiguchi T, Ariki Y, “3D human posture estimation using the HOG features from the monocular image,” In 19th International Conference on Pattern Recognition, 2008, pp. 1–4.
 - [59]. Wang B, Liang W, Wang Y, Liang Y, “Head Pose Estimation with Combined 2D SIFT and 3D HOG Features,” In 7th International Conference on Image and Graphics, Qingdao, 2013, pp. 650–655.
 - [60]. Kiran PS, Kumar DA, Kishore PVV, Kumar EK, Kumar MTK, Sastry ASCS, “Investigation of 3-D Relational Geometric Features for Kernel- Based 3-D Sign Language Recognition,” In IEEE International Conference on Intelligent Systems and Green Technology (ICISGT), 2019, pp. 31–313.

- [61]. Rusu RB, Holzbach A, Beetz M, Bradski G, “Detecting and segmenting objects for mobile manipulation,” In IEEE International Conference on Computer Vision Workshops, 2009, pp. 47–54.
- [62]. Muja M, Rusu RB, Bradski G, Lowe DG, “REIN - A fast, robust, scalable REcognition Infrastructure,” In IEEE International Conference on Robotics and Automation, 2011, pp. 2939–2946.
- [63]. Madry M, Ek CH, Detry R, Hang K, Kragic D, “Improving generalization for 3D object categorization with Global Structure Histograms,” In IEEE International Conference on Intelligent Robots and Systems, 2012, pp. 1379–1386.
- [64]. Wohlking W, Vincze M, “Shape distributions on voxel surfaces for 3d object classification from depth images,” In IEEE International Conference on Signal and Image Processing Applications, 2012, pp. 115–120.
- [65]. Wohlking W, Vincze M, “Ensemble of shape functions for 3d object classification,” In IEEE International Conference on Robotics and Biomimetics, 2012, pp. 2987–2992.
- [66]. Chen T, Dai B, Liu D, Song J, “Performance of global descriptors for Velodyne-based urban object recognition,” In IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, 2014, pp. 667–673.
- [67]. Lima JPSdM, Teichrieb V, “An Efficient Global Point Cloud Descriptor for Object Recognition and Pose Estimation,” In 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Sao Paulo, 2016, pp. 56–63.
- [68]. Aldoma A, Tombari F, Rusu RB, VinczeM, “OUR-CVFH-oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6DOF pose estimation,” In Joint DAGM (German Association for Pattern Recognition) and OAGM Symposium, 2012, pp. 113–122.
- [69]. Aldoma, Aitor, Markus Vincze, Nico Blodow, David Gossow, Suat Gedikli, Radu Bogdan Rusu, and Gary Bradski, “CAD-model recognition and 6DOF pose estimation using 3D cues,” In IEEE international conference on computer vision workshops (ICCV workshops), 2011 pp. 585-592.

- [70]. Ghasemzadeh A, Demirel H, “3D discrete wavelet transform-based feature extraction for hyperspectral face recognition,” *IET Biometrics* vol. 7, no. 1, pp. 49–55, 2018.
- [71]. Kasaei SH, Tom AM, Lopes LS, Oliveira M, “Good: a global orthographic object descriptor for 3d object recognition and manipulation,” *Pattern Recognition Letters*, vol. 83, no. 1, pp. 312–320, 2016.
- [72]. Rajan, E. G., “Cellular logic array processing for high throughput image processing systems”, *Sadhana*, vol. 18, no. 2, pp. 279-300, 1993.
- [73]. H.K. Liu; “Two and Three-dimensional boundary detection,” *Computer Graphics Image Processing*, vol. 6 pp.123-134, 1977.
- [74]. S.W. Zucker, R.A. Hummel, “A Three Dimensional Edge Operator,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 3, no. 3, 1981.
- [75]. M.F. Heuckel, “An operator which locates edges in digitized pictures,” *J. Association for computing machinery*, vol. 18, pp. 113-125, 1971.
- [76]. M. Morgenthaler, A. Rosenfeld, “Multidimensional edge detection by hyper surface fitting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 3, no. 4, July, 1981.
- [77]. O. Monga and R. Deriche, “A new three-dimensional boundary detection,” In *Proceeding of International Conference on Pattern Recognition*, Paris, 1986.
- [78]. Mukherjee, J., Das, P.P., and Chatterji, B.N., “An algorithm for the extraction of the wireframe structure of a three-dimensional object,” *Pattern recognition*, vol. 23, no. 9, pp.999-1010, 1990.
- [79]. Ren, Xiaoyuan, Libing Jiang, Xiaolan Tang, and Weichun Liu, “3D Wireframe Modeling and Viewpoint Estimation for Multi-Class Objects Combining Deep Neural Network and Deformable Model Matching,” *Applied Sciences*, vol. 9, no. 10, 2019.
- [80]. Prakoonwit, Simant, and Ralph Benjamin, “3D surface point and wireframe reconstruction from multiview photographic images,” *Image and Vision Computing*, vol. 25, no. 9, pp. 1509-1518, 2007.

- [81]. Jiang, Luo, Juyong Zhang, Bailin Deng, Hao Li, and Ligang Liu, “3D face reconstruction with geometry details from a single image,” *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 4756-4770, 2018.
- [82]. Tu, Xiaoguang, Jian Zhao, Mei Xie, Zihang Jiang, Akshaya Balamurugan, Yao Luo, Yang Zhao, Lingxiao He, Zheng Ma, and Jiashi Feng, “3D face reconstruction from a single image assisted by 2D face images in the wild,” *IEEE Transactions on Multimedia*, vol. 23, pp. 1160-1172, 2020.
- [83]. J.-F. Hu, W.-S. Zheng, X. Xie, and J. Lai, “Sparse transfer for facial shape-from-shading,” *Pattern Recognition*, vol. 68, pp. 272–285, 2017.
- [84]. Y. Quéau, J. Mélou, F. Castan, D. Cremers, and J. D. Durou, “A variational approach to shape-from-shading under natural illumination,” In *Proceeding of International Workshop on Energy Minimization Method in Comput. Vision and Pattern Recognition*, 2017, pp. 342–357.
- [85]. Jeni, L.A., Cohn, J.F. and Kanade, T., “Dense 3D face alignment from 2D videos in real-time,” In *11th IEEE international conference and workshops on automatic face and gesture recognition*, 2015, vol. 1, pp. 1-8.
- [86]. A. Maghari, I. Liao, and B. Belaton, “Quantitative analysis on PCA-based statistical 3D face shape modeling,” In *Computational Modelling of Objects Represented in Images III: Fundamentals, Methods and Applications*, vol. 13, 2012.
- [87]. D. Bradley, W. Heidrich, T. Popa, and A. Sheffer, “High resolution passive facial performance capture,” *ACM Trans. Graph.*, vol. 29, no. 4, pp. 1–10, 2010.
- [88]. J. Jo, H. Choi, I.-J. Kim, and J. Kim, “Single-view-based 3D facial reconstruction method robust against pose variations,” *Pattern Recognition*, vol. 48, no. 1, pp. 73–85, 2015.
- [89]. I. Kemelmacher-Shlizerman and S. M. Seitz, “Face reconstruction in the wild,” In *Proceeding of International Conference on Computer Vision*, 2011, pp. 1746–1753.
- [90]. Blanz, Volker, and Thomas Vetter, “A morphable model for the synthesis of 3D faces,” In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999, pp. 187-194.

- [91]. Chen, Yajing, Fanzi Wu, Zeyu Wang, Yibing Song, Yonggen Ling, and Linchao Bao., "Self-supervised learning of detailed 3d face reconstruction," IEEE Transactions on Image Processing vol. 29, pp. 8696-8705,2020.
- [92]. Fan, Yangyu, Yang Liu, Guoyun Lv, Shiya Liu, Gen Li, and Yanhui Huang, "Full Face-and-Head 3D Model with Photorealistic Texture," IEEE Access vol. 8, pp. 210709-210721, 2020.
- [93]. Huang, Yu-Hsuan, and Homer H. Chen., "Face Recognition Under Low Illumination Via Deep Feature Reconstruction Network," In IEEE International Conference on Image Processing (ICIP), 2020, pp. 2161-2165.
- [94]. Afzal, HM Rehan, Suhuai Luo, M. Kamran Afzal, Gopal Chaudhary, Manju Khari, and Sathish AP Kumar, "3D Face Reconstruction From Single 2D Image Using Distinctive Features," IEEE Access, vol. 8, pp. 180681-180689, 2020.
- [95]. Liu, Qing-Ming, Rui-Sheng Jia, Chao-Yue Zhao, Xiao-Ying Liu, Hong-Mei Sun, and Xing-Li Zhang, "Face super-resolution reconstruction based on self-attention residual network," IEEE Access, vol. 8, pp. 4110-4121, 2019.
- [96]. Afzal, HM Rehan, Suhuai Luo, and M. Kamran Afzal, "Reconstruction of 3D facial image using a single 2D image," In International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2018, pp. 1-5.
- [97]. Guo, Yudong, Jianfei Cai, Boyi Jiang, and Jianmin Zheng, "CNN-based real-time dense face reconstruction with inverse-rendered photo-realistic face images," IEEE transactions on pattern analysis and machine intelligence, vol. 41, no. 6, pp. 1294-1307, 2018.
- [98]. Jiang, Luo, Juyong Zhang, Bailin Deng, Hao Li, and Ligang Liu, "3D face reconstruction with geometry details from a single image," IEEE Transactions on Image Processing , vol. 27, no. 10, pp. 4756-4770, 2018.
- [99]. Cao, Chen, Yanlin Weng, Shun Zhou, Yiyong Tong, and Kun Zhou, "Face warehouse: A 3D facial expression database for visual computing," IEEE Transactions on Visualization and Computer Graphics, vol. 20, no. 3, pp. 413-425, 2013.

- [100]. Ho, Huy Tho, and Rama Chellappa, "Pose-invariant face recognition using markov random fields," *IEEE transactions on image processing*, vol. 22, no. 4, pp.1573-1584, 2012.
- [101]. Song, Mingli, Dacheng Tao, Xiaoqin Huang, Chun Chen, and Jiajun Bu, "Three-dimensional face reconstruction from a single image by a coupled RBF network," *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2887-2897, 2012.
- [102]. Lai, Jian, and Xudong Jiang, "Modular weighted global sparse representation for robust face recognition," *IEEE Signal processing letters*, vol. 19, no. 9, pp. 571-574, 2012.
- [103]. Li, Yinsheng, and Guang-Hong Chen, "An Empirical Data Inconsistency Metric (DIM) Driven CT Image Reconstruction Method," *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, pp. 337-348, 2018.
- [104]. Bustin, Aurelien, Damien Voilliot, Anne Menini, Jacques Felblinger, Christian de Chillou, Darius Burschka, Laurent Bonnemains, and Freddy Odille, "Isotropic reconstruction of MR images using 3D patch-based self-similarity learning," *IEEE Transactions on Medical Imaging*, vol. 37, no. 8, pp. 1932-1942, 2018.
- [105]. Hou, Benjamin, Bishesh Khanal, Amir Alansary, Steven McDonagh, Alice Davidson, Mary Rutherford, Jo V. Hajnal, Daniel Rueckert, Ben Glocker, and Bernhard Kainz, "3-D reconstruction in canonical coordinate space from arbitrarily oriented 2-D images," *IEEE transactions on medical imaging*, vol. 37, no. 8, pp. 1737-1750, 2018.
- [106]. Benalcazar, Daniel P., Diego Bastias, Claudio A. Perez, and Kevin W. Bowyer, "A 3D Iris Scanner from Multiple 2D Visible Light Images," *IEEE Access*, vol. 7, pp. 61461-61472, 2019.
- [107]. Zheng, Xuehang, Saiprasad Ravishankar, Yong Long, and Jeffrey A. Fessler, "PWLS-ULTRA: An efficient clustering and learning-based approach for low-dose 3D CT image reconstruction," *IEEE transactions on medical imaging*, vol. 37, no. 6, pp.1498-1510, 2018.

- [108]. Zhang, Chen, “CuFusion2: Accurate and Denoised Volumetric 3D Object Reconstruction Using Depth Cameras,” *IEEE Access*, vol. 7, pp. 49882-49893, 2019.
- [109]. Guo, Yuanhao, Yunpeng Zhang, and Fons J. Verbeek, “A two-phase 3D reconstruction approach for light microscopy axial-view imaging,” *IEEE Journal of Selected Topics in Signal Processing* vol. 11, no. 7, pp.1034-1046, 2017.
- [110]. Ding, Lu, Xosé Luís Deán-Ben, and Daniel Razansky, “Efficient 3-D model-based reconstruction scheme for arbitrary optoacoustic acquisition geometries,” *IEEE Transactions on Medical Imaging*, vol. 36, no. 9, pp. 1858-1867, 2017.
- [111]. Li, Bin, Yonghan Zhang, Bo Zhao, and Hongyao Shao, “3D-ReConstnet: A Single-View 3D-Object Point Cloud Reconstruction Network,” *IEEE Access* vol. 8, pp.83782-83790, 2020.
- [112]. A. Fitzgibbon, G. Cross, and A. Zisserman, “Automatic 3d model construction for turn-table sequences,” In *European Workshop on 3D Structure from Multiple Images of Large-Scale Environments*. Springer, 1998, pp. 155–170.
- [113]. J. Franco and E. Boyer, “Exact polyhedral visual hulls,” In *British Machine Vision Conference*, 2003, vol. 1, pp. 329–338.
- [114]. S. Lazebnik, Y. Furukawa, and J. Ponce, “Projective visual hulls,” *International Journal of Computer Vision*, vol. 74, no. 2, pp. 137–165, 2007.
- [115]. A. Djelouah, J. Franco, E. Boyer, F. Le Clerc, and P. Perez, “N-tuple color segmentation for multi-view silhouette extraction,” In *European Conference on Computer Vision*, Springer, 2012, pp. 818–831.
- [116]. K. Kutulakos and S. Seitz, “A theory of shape by space carving,” *International Journal of Computer Vision*, vol. 38, no. 3, pp. 199–218, 2000.
- [117]. Y. Furukawa and J. Ponce, “Carved visual hulls for image-based modeling,” *International Journal of Computer Vision*, vol. 81, no. 1, pp. 53–67, 2009.
- [118]. I. Ihrke, K. Kutulakos, H. Lensch, M. Magnor, and W. Heidrich, “Transparent and specular object reconstruction,” In *Computer Graphics Forum*, 2010, vol. 29, no. 8, pp. 2400–2426.

- [119]. D. Liu, X. Chen, and Y. Yang, "Frequency-based 3d reconstruction of transparent and specular objects," In IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 660–667.
- [120]. Y. Qian, M. Gong, and Y. Yang, "3D reconstruction of transparent objects with position-normal consistency," In IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4369–4377.
- [121]. Savinov, Nikolay, Christian Hane, Lubor Ladicky, and Marc Pollefeys, "Semantic 3d reconstruction with continuous regularization and ray potentials using a visibility consistency constraint," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5460–5469.
- [122]. S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," In IEEE Conference on Computer Vision and Pattern Recognition, 2006, vol. 1 , pp. 519–528.
- [123]. X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. Berg, "Matchnet: unifying feature and metric learning for patch-based matching," In IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3279–3286.
- [124]. J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," Journal of Machine Learning Research, vol. 17, pp. 1–32, 2016.
- [125]. Tomita, Masaru, and See-Kiong Ng. "The generalized LR parsing algorithm." In Generalized LR parsing, Springer, Boston, pp. 1-16, 1991.
- [126]. Durbin, Richard, Sean R. Eddy, Anders Krogh, and Graeme Mitchison, "Biological sequence analysis: probabilistic models of proteins and nucleic acids," Cambridge university press, pp. 9-10, 1998.
- [127]. Christopher, D. Manning, and Schütze Hinrich, "Foundations of statistical natural language processing," pp. 128-129, 1999.
- [128]. Thomason, Michael G, "Syntactic methods in pattern recognition," Pattern Recognition Theory and Applications, pp. 119-137, 1982.
- [129]. Pavlidis, T, "Structural pattern recognition Springer Verlag," Berlin, FRG 1977.

- [130]. Cui, Lixin, Lu Bai, Yue Wang, Xiao Bai, Zhihong Zhang, and Edwin R. Hancock, "P2P lending analysis using the most relevant graph-based features," In Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), Springer ,Cham, 2016, pp. 3-14.
- [131]. Ogiela MR, Tadeusiewicz R, "Application of syntactic methods of pattern recognition for data mining and knowledge discovery in medicine," In International Society for Optics and Photonics, Data Mining and Knowledge Discovery: Theory, Tools, and Technology , 2000, pp. 308–318.
- [132]. Jaf S, Calder C, "Deep learning for natural language parsing," IEEE Access vol. 7, no. , pp. 131363–131373, 2019.
- [133]. Zhang Y-J, Zhang T, Zhu J-B, Yao T-S, "Research on DOP- based Chinese parsing," In International Conference on Machine Learning and Cybernetics, 2005, pp. 3840–3845.
- [134]. Sarker J, Billah M, Mamun MA, "Textual Question Answering for Semantic Parsing in Natural Language Processing," In 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), 2019, pp. 1–5.
- [135]. Seo, Minjoon, Sewon Min, Ali Farhadi, and Hannaneh Hajishirzi, "Query-reduction networks for question answering," arXiv preprint arXiv:1606.04582 , 2016.
- [136]. Hang, Le Quang, "A Hierarchical Graph Method Using in A* Algorithm for Vietnamese Parsing Technique," In Proceedings of the International Conference on Asian Language Processing, 2012, pp. 25-28.
- [137]. Mall S, Jaiswal UC, "Resolving issues in parsing technique in machine translation from Hindi language to English language," In International Conference on Computer and Communication Technology (ICCCT), 2014, pp. 55–58.
- [138]. Badr G, Oommen BJ, "On optimizing syntactic pattern recognition using tries and AI-based heuristic-search strategies," IEEE Transactions on Systems, Man, and Cybernetics, vol. 36, no. 3, pp. 611–622, 2014.

- [139]. Huang K, Leu D, “Syntactic pattern recognition for wavelet clustering in seismogram,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 7, pp. 2453–2461, 2019.
- [140]. Husen MN, Lee S, “Continuous Car Driving intention recognition with syntactic pattern approach,” In *International Conference on Information and Communication Technology (ICICTM)*, 2016, pp. 71–76.
- [141]. Yu Y, Zhang W, Liu W, “A New Syntactic Approach to Graphic Symbol Recognition,” In *9th International Conference on Document Analysis and Recognition*, 2007, pp. 516–520.
- [142]. W. Zhang, R. Li, H. Deng, L. Wang, W. Lin, S. Ji, and D. Shen, “Deep convolutional neural networks for multi-modality isointense infant brain image segmentation,” *NeuroImage*, vol. 108, pp. 214–224, 2015.
- [143]. W. Li, F. Jia, and Q. Hu, “Automatic segmentation of liver tumor in CT images with deep convolutional neural networks,” *Journal of Computer and Communications*, vol. 3, no. 11, pp. 146–151, 2015.
- [144]. M. Collins, R. E. Schapire, and Y. Singer, “Logistic regression, AdaBoost and Bregman distances,” *Machine Learning*, vol. 48, no. 1-3, pp. 253–285, 2002.
- [145]. L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [146]. T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, “Support vector machine classification and validation of cancer tissue samples using microarray expression data,” *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.
- [147]. F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, “Residual Attention Network for Image Classification,” In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6450–6458.
- [148]. L. C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, “Attention to scale: Scale-aware semantic image segmentation,” In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3640–3649.
- [149]. J. Schlemper, O. Oktay, L. Chen, J. Matthew, C. Knight, B. Kainz, B. Glocker, and D. Rueckert, “Attention-Gated Networks for Improving Ultrasound Scan Plane Detection,” *arXiv preprint arXiv:1804.05338*, 2018.

- [150]. Mzoughi, Hiba, Ines Njeh, Ali Wali, Mohamed Ben Slima, Ahmed BenHamida, Chokri Mhiri, and Kharedine Ben Mahfoudhe, "Deep multi-scale 3D convolutional neural network (CNN) for MRI gliomas brain tumor classification," *Journal of Digital Imaging*, vol. 33, no. 4, pp. 903-915, 2020.
- [151]. Pei, Linmin, Lasitha Vidyaratne, Wei-Wen Hsu, Md Monibor Rahman, and Khan M. Iftekharruddin, "Brain tumor classification using 3D convolutional neural network," In *International MICCAI brainlesion workshop*, pp. 335-342. Springer, Cham, 2019.
- [152]. Ge, Chenjie, Irene Yu-Hua Gu, Asgeir Store Jakola, and Jie Yang, "Deep learning and multi-sensor fusion for glioma classification using multistream 2D convolutional networks," In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2018, pp. 5894-5897.
- [153]. Li, Ming, Lishan Kuang, Shuhua Xu, and ZhanguoSha, "Brain Tumor Detection Based on Multimodal Information Fusion and Convolutional Neural Network," *IEEE Access*, vol. 7, pp. 180134-180146, 2019.
- [154]. Abdel-Gawad, Ahmed H., Lobna A. Said, and Ahmed G. Radwan, "Optimized Edge Detection Technique for Brain Tumor Detection in MR Images," *IEEE Access*, vol. 8, pp. 136243-136259, 2020.
- [155]. Mallick, Pradeep Kumar, Seuc Ho Ryu, Sandeep Kumar Satapathy, Shruti Mishra, GiaNhu Nguyen, and Prayag Tiwari, "Brain MRI image classification for cancer detection using deep wavelet autoencoder-based deep neural network," *IEEE Access*, vol. 7, pp. 46278-46287, 2019.
- [156]. Shakeel, P. Mohamed, Tarek E. El Tobely, Haytham Al-Feel, GunasekaranManogaran, and S. Baskar, "Neural network-based brain tumor detection using wireless infrared imaging sensor," *IEEE Access*, vol. 7, pp. 5577-5588, 2019.
- [157]. Anitha, V., and S. J. I. C. V. Murugavalli, "Brain tumor classification using two-tier classifier with adaptive segmentation technique," *IET computer vision*, vol. 10, pp. 9-17, 2016.
- [158]. Mano, Abhisha, and Swaminathan Anand, "Method of multi-region tumour segmentation in brain MRI images using grid-based segmentation and

- weighted bee swarm optimization,” *IET Image Processing*, vol. 14, no. 12, pp. 2901-2910, 2020.
- [159]. Swati, Zar Nawab Khan, Qinghua Zhao, Muhammad Kabir, Farman Ali, Zakir Ali, Saeed Ahmed, and Jianfeng Lu, “Brain tumor classification for MR images using transfer learning and fine-tuning,” *Computerized Medical Imaging and Graphics*, pp. 34-46, 2019.
- [160]. Hassan Ali Khan, Wu Jue, Muhammad Mushtaq, Muhammad Umer Mushtaq, “Brain tumor classification in MRI image using convolutional neural network,” *Mathematical Biosciences and Engineering*, vol. 17, no.5, pp. 6203-6216, 2020.
- [161]. Masood, Momina, TahiraNazir, Marriam Nawaz, AwaisMehmood, Junaid Rashid, Hyuk-Yoon Kwon, Toqeer Mahmood, and Amir Hussain, “A Novel Deep Learning Method for Recognition and Classification of Brain Tumors from MRI Images,” *Diagnostics*, vo. 11, no. 5, pp. 744, 2021.
- [162]. Badža, Milica M., and Marko Č. Barjaktarović, “Classification of brain tumors from MRI images using a convolutional neural network,” *Applied Sciences*, vol. 10, no. 6, pp. 1999, 2020.
- [163]. Vimal Kurup, R., V. Sowmya, and K. P. Soman, “Effect of data pre-processing on brain tumor classification using capsulenet,” In *International Conference on Intelligent Computing and Communication Technologies*, Springer, Singapore, 2019, pp. 110-119..
- [164]. Eisert, Peter, Eckehard Steinbach, and Bernd Girod, “Automatic reconstruction of stationary 3-D objects from multiple uncalibrated camera views,” *IEEE Transactions on Circuits and Systems for Video Technology* vol. 10, no. 2, pp. 261-277, 2000.
- [165]. Martin, Worthy N, and Jagdishkumar Keshoram Aggarwal, “Volumetric descriptions of objects from multiple views,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 2, pp. 150-158, 1983.
- [166]. Zhu X, Liu X, Lei Z, Li SZ, “Face alignment in full pose range: a 3D Total solution,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 78–92, 2019.

- [167]. Boyer, Edmond, "Object models from contour sequences," In European Conference on Computer Vision, Berlin, Heidelberg, Springer, 1996, pp. 109-118.
- [168]. S. Sullivan, J. Ponce, "Automatic model construction, pose estimation and object recognition from photographs using triangular splines," In International Conference on Computer Vision (ICCV), Bombay, India, 1998, pp. 90–95.
- [169]. W. Niem, J. Wingbermuhle, "Automatic reconstruction of 3D objects using a mobile monoscopic camera," In International Conference on Recent Advances in 3D Imaging and Modeling, Ottawa, Canada, 1997, pp. 173–180.
- [170]. Szeliski, Richard, "Rapid octree construction from image sequences," CVGIP: Image understanding, vol. 58, no. 1, pp.23-32, 1993.
- [171]. A.W., Fitzgibbon, G. Cross, A. Zisserman, "Automatic 3D model construction for turn-table sequences," In ECCV'98 Workshop on 3D Structure from Multiple Images of Large-Scale Environments, Freiburg, Germany, 1998, pp. 154–170.
- [172]. Jasmin, G. D., and E. G. Rajan. "Normalized vector codes for object recognition using artificial neural networks in the framework of picture description languages," Global Journal of Computer Science and Technology, 2013.
- [173]. Cheng, J. Brain tumor dataset, Available online: https://figshare.com/articles/brain_tumor_dataset/1512427, accessed on 21 December 2020.
- [174]. N. Chakrabarty, Brain MRI images dataset for brain tumor detection, Kaggle, 2019, Available online: <https://www.kaggle.com/navoneel/brain-mri-images-forbrain-tumor-detection> , accessed on 1 April 2021.
- [175]. Available online: <https://www.kaggle.com/aryashah2k/brain-tumor-segmentation-brats-2019>
- [176]. <https://modelnet.cs.princeton.edu/>
- [177]. <http://pix3d.csail.mit.edu/>
- [178]. Mottaghi, Roozbeh, Yu Xiang, and Silvio Savarese, "Beyond pascal: A benchmark for 3D object detection in the wild," In IEEE Winter Conference on Applications of Computer Vision. 2014.

- [179]. Fisher, Matthew, Daniel Ritchie, Manolis Savva, Thomas Funkhouser, and Pat Hanrahan, "Example-based synthesis of 3D object arrangements," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, pp. 1-11, 2012.
- [180]. Duda, Richard O., and Peter E. Hart. *Pattern classification*, "John Wiley & Sons, pp. 97-98, 2006.
- [181]. Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Boston, second edition, pp. 95, 1990.
- [182]. Anil K. Jain, Robert P. W. Duin, and Jianchang Mao, "Statistical Pattern Recognition: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4 –37, 2000.
- [183]. Liu, H.H. and Fu, K.S., "A syntactic pattern recognition approach to seismic discrimination Geoexploration," vol. 20, no. 1, pp.183-196, 1982.
- [184]. Rafael C. Gonzalez and Michael G. Thomason, "Syntactic Pattern Recognition: An Introduction," Addison-Wesley, Reading, Massachusetts, pp. 5-7, 1978.
- [185]. T. Pavlidis, "Structural Pattern Recognition," Springer-Verlag, Berlin, pp. 9-10, 1977.
- [186]. K. Thiruvankadam, P Nagaraja, "Brain tumor segmentation of MRI brain images through FCM clustering and seeded region growing technique," *International Journal of Applied Engineering Research*, vol. 10, no.7, pp. 427-43, 2015.
- [187]. Fan, Baijiang, Yunbo Rao, Wei Liu, Qifei Wang, and Huaiyu Wen, "Region-based growing algorithm for 3D reconstruction from MRI images," In 2017 2nd International Conference on Image, Vision, and Computing (ICIVC), 2017, pp. 521-525.
- [188]. Chen, Jun, Zhechao Wan, Jiacheng Zhang, Wenhua Li, Yanbing Chen, Yuebing Li, and Yue Duan, "Medical image segmentation and reconstruction of prostate tumor based on 3D AlexNet," *Computer Methods and Programs in Biomedicine*, 105878, 2020.
- [189]. Li, Ming, LishanKuang, Shuhua Xu, and ZhanguoSha, "Brain Tumor Detection Based on Multimodal Information Fusion and Convolutional Neural Network," *IEEE Access*, vol. 7, pp. 180134-180146, 2019.

- [190]. Han, Changhee, Leonardo Rundo, Ryosuke Araki, Yudai Nagano, Yujiro Furukawa, Giancarlo Mauri, Hideki Nakayama, and Hideaki Hayashi, "Combining noise-to-image and image-to-image GANs: Brain MR image augmentation for tumor detection," *IEEE Access*, vol. 7, pp. 156966-156977, 2019.
- [191]. Wang, Weiguang, Fanlong Bu, Ziyi Lin, and Shuangqing Zhai, "Learning Methods of Convolutional Neural Network Combined With Image Feature Extraction in Brain Tumor Detection," *IEEE Access*, vol. 8, pp. 152659-152668, 2020.
- [192]. Liu, Hsun K, "Two-and three-dimensional boundary detection," *Computer Graphics and Image Processing*, vol. 6, no. 2, pp. 123-134, 1977.
- [193]. Rehman, Arshia, SaeedaNaz, Muhammad Imran Razzak, FaizaAkram, and Muhammad Imran, "A deep learning-based framework for automatic brain tumors classification using transfer learning," *Circuits, Systems, and Signal Processing*, vol. 39, no. 20, pp. 757-775, 2020.
- [194]. Pashaei, A Sajedi, H., Jazayeri, N, "Brain Tumor Classification via Convolutional Neural Network and Extreme Learning Machines," In *Proceedings of the 2018 8th International Conference on Computer and Knowledge Engineering (ICCKE)*, Mashhad, Iran, 2018; pp. 314–319.
- [195]. Srinivasan, K.; Nandhitha, N.M., "Development of Deep Learning algorithms for Brain Tumor classification using GLCM and Wavelet Packets," *Caribbean Journal of Science*, no. 53, pp. 1222–1228, 2019.
- [196]. Zhuge, Ying, Holly Ning, Peter Mathen, Jason Y. Cheng, Andra V. Krauze, Kevin Camphausen, and Robert W. Miller, "Automated glioma grading on conventional MRI images using deep convolutional neural networks," *Medical physics*, vol. 47, no. 7, pp. 3044-3053, 2020.
- [197]. J. Cheng,W. Huang, R. ShuangliangCao,W.Y.Yang, Z.Yun, Z.Wang, Q. Feng, "Enhanced performance of brain tumor classification via tumor region augmentation and partition," *PLoS ONE*, vol. 10, no. 10, pp. e0140381.
- [198]. J. Deng, W.Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, "ImageNet: a large-scale hierarchical image database," In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2009)*, 2009, pp. 248–255.
- [199]. and Pattern Recognition (CVPR-2009), 2009, pp. 248–255.

- [200]. T.A. Abir, J.A. Siraji, E. Ahmed, B. Khulna, “Analysis of a novel MRI based brain tumour classification using probabilistic neural network,” *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 4, no. 8, pp. 65–79, 2018.
- [201]. W. Widhiarso, Y. Yohannes, C. Prakarsah, “Brain tumor classification using gray level co-occurrence matrix and convolutional neural network,” *Indonesian Journal of Electronics and Instrumentation Systems*, vol. 8, no. 2, pp. 179–190, 2018.
- [202]. Afshar, A.Mohammadi, K.N Plataniotis, “Brain tumor type classification via capsule networks,” *arXiv preprint, arXiv: 1802.10200*, 2018.
- [203]. N. Abiwinanda, M. Hanif, S.T. Hesaputra, A. Handayani, T.R. Mengko, “Brain tumor classification using convolutional neural network,” In *World Congress on Medical Physics and Biomedical Engineering*, pp. 183–189, 2019.
- [204]. M.R. Ismael, I. Abdel-Qader, “Brain tumor classification via statistical features and back-propagation neural network,” In *IEEE International Conference on Information Technology*, 2018, pp. 0252–0257.
- [205]. Deepak, S., Ameer, P., “Brain tumor classification using deep CNN features via transfer learning,” *Computers in Biology and Medicine*, vol 111, pp. 103345, 2019.
- [206]. Rayhan, F., “FR-MRInet: A Deep Convolutional Encoder-Decoder for Brain Tumor Segmentation with Relu-RGB and Sliding window,” *International Journal of Computer Applications*, vol. 975, pp. 8887, 2018.
- [207]. Ren, S.; He, K.; Girshick, R.B.; Sun, J., “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, 2017.
- [208]. Shahzadi, Iram, Tong Boon Tang, Fabrice Meriadeau, and Abdul Quyyum, “CNN-LSTM: cascaded framework for brain tumour classification,” In *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, 2018, pp. 633-637.
- [209]. Ouerghi, Hajer, Olfa Mourali, and Ezzeddine Zagrouba, “Glioma classification via mr images radiomics analysis,” *The Visual Computer*, vol. 38, no. 4, pp. 1427-1441, 2022.

- [210]. Chatterjee, S., Nizamani, F.A., Nürnberger, A. and Speck, O., “Classification of brain tumours in MR images using deep spatiotemporal models,” *Scientific Reports*, vol. 12, no. 1, pp.1-11, 2022.
- [211]. Zhuge, Ying, Holly Ning, Peter Mathen, Jason Y. Cheng, Andra V. Krauze, Kevin Camphausen, and Robert W. Miller, “Automated glioma grading on conventional MRI images using deep convolutional neural networks,” *Medical physics*, vol. 47, no. 7, pp. 3044-3053, 2020.
- [212]. J. Smith, R. Jones, and K. Trello, “Adaptive filtering in data communications with self improved error reference,” In *Proceeding of IEEE International Conference on Wireless Communications* , 2004, pp. 65-68.

LIST OF PUBLICATIONS

Journal Publication

1. **Shilpa Rani**, Deepika Ghai and Sandeep Kumar, “Reconstruction of Simple and Complex Three Dimensional Images Using Pattern Recognition Algorithm” in **Journal of Information Technology Management. (Scopus Journal Published).DOI: 10.22059/JITM.2022.87475**
2. **Shilpa Rani**, Kamlesh Lakhwani and Sandeep Kumar, “Three Dimensional Objects Recognition & Pattern Recognition Technique; Related challenges: A Review” in Multimedia Tools and Application, pp. 1-44, 2022. **(SCI Published, IF-2.757 & Q-1 Journal). DOI:10.1007/s11042-022-12412-2**
3. **Shilpa Rani**, Deepika Ghai, Sandeep Kumar, “Efficient 3D Alex-Net Architecture for Object Recognition using Syntactic Patterns from Medical Images”, in Computational Intelligence and Neuroscience, vol. 2, no. 1, pp. 1-16, 2022. **(SCI-Published, IF-3.63 & Q-1 Journal) DOI:10.1155/2022/7882924.**
4. **Shilpa Rani**, Deepika Ghai and Sandeep Kumar, “Object Detection and Recognition using Contour based Edge Detection and Fast R-CNN” in Multimedia Tool and Application, **Springer Journal (SCI-Accepted, IF-2.757 & Q-1 Journal).**
5. **Shilpa Rani**, Kamlesh Lakhwani and Sandeep Kumar “Syntactic approach to reconstruct simple and complex medical images” in International Journal of Signal and Imaging Systems Engineering, 2022. **(E-SCI & Scopus Accepted, I.F-1.31).**
6. **Shilpa Rani**, Deepika Ghai and Sandeep Kumar, “Knowledge Vector Representation of Three Dimensional Convex Polyhedrons and Reconstruction of Medical Images using Knowledge Vector”, ” in Multimedia Tools and Application **(SCI, Communicated).**

Conference Publication

7. **Shilpa Rani**, Kamlesh Lakhwani, and Sandeep Kumar, "Three Dimensional Wireframe Model of Medical and Complex Images Using Cellular Logic Array Processing Techniques." In 12th International Conference on Soft Computing and Pattern Recognition, pp. 196-207, Springer 2020. **(Scopus Indexed Published, MIR Lab-USA). DOI: 10.1007/978-3-030-73689-7_20**
8. **Shilpa Rani**, Deepika Ghai and Sandeep Kumar, "Reconstruction of Wire Frame Model of Complex Images using Syntactic Pattern Recognition" in 4th Smart Cities Symposium, pp. 8-13, November 2021. **(Scopus Indexed Published, University of Bahrain). DOI: 10.1049/icp.2022.0304**
9. **Shilpa Rani**, Deepika Ghai, Sandeep Kumar, "Automatic Detection of Brain Tumor from CT and MRI Images using Wireframe model and 3D Alex-Net", in 3rd International Conference on Decision Aid Sciences and Applications (DASA), pp. 1132-1138, 2022. **(Scopus Indexed Published, Mae Fah Luang University, Thailand). DOI: 10.1109/DASA54658.2022.9765114.**

Book Chapter

10. **Shilpa Rani**, Deepika Ghai and Sandeep Kumar, "Construction and Reconstruction of 3D Facial and Wireframe Model using Syntactic Pattern Recognition" in "Cognitive Behavior & Human Computer Interaction" in Scrivener & Willey Publishing House **(Scopus Indexed Book Chapter Published). DOI:10.1002/9781119792109.ch6**