PROJECT TITLE

IMAGE SEGMENTATION AND OBJECT RECOGNITION

BECE301L – DIGITAL SIGNAL PROCESSING

BY

Mandke Aditya Jayant (21BEC1088)
Yusuf Aziz Hussain Mahu (21BEC1100)

SUBMITTED TO

Dr. RAMESH R

Associate Professor, VIT Chennai

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF ELECTRONICS ENGINEERING

VELLORE INSTITUE OF TECHNOLOGY

CHENNAI – 600127

JULY 2023

CERTIFICATE

This is to certify that the Project work titled "**Image Segmentation and Object Recognition**" being submitted by **Mandke Aditya Jayant(21BEC1088)** and **Yusuf Aziz Hussain Mahu(21BEC1100)** for the course B**ECE301L – DIGITAL SIGNAL PROCESSING**, is a record of bonafide work done under my guidance. The contents of this project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University

Prof. RAMESH R

Associate Professor

School of Electronics Engineering (SENSE)

VIT University, Chennai

Chennai – 600 127

ABSTRACT

Image segmentation and object recognition are crucial tasks in computer vision and image processing. In this project, we propose a methodology for image segmentation and object recognition using MATLAB. The objective is to accurately partition an input image into meaningful regions and identify objects of interest within those regions.

Firstly, the image segmentation process is performed to separate the image into coherent regions based on various characteristics such as color, texture, or intensity. Several segmentation techniques, including region-based, edge-based, and clustering methods, are implemented and evaluated to determine the most suitable approach for the given image dataset.

Once the image is segmented, object recognition techniques are employed to identify and classify objects within each segment. This involves extracting relevant features from the segmented regions, such as shape, texture, or local descriptors, and training a recognition model using machine learning algorithms like support vector machines (SVM), k-nearest neighbors (KNN), or convolutional neural networks (CNN). The trained model is then used to recognize and label objects within the image.

To enhance the accuracy and robustness of the proposed system, additional preprocessing steps, such as noise reduction, image enhancement, and normalization, are incorporated to handle common challenges like noise, illumination variations, and object occlusion.

The MATLAB software environment provides a comprehensive set of built-in functions, toolboxes, and libraries that facilitate the implementation of image segmentation and object recognition algorithms. MATLAB's intuitive syntax and extensive documentation make it a suitable choice for researchers and practitioners working in the field of computer vision and image processing.

Experimental evaluations are conducted on various benchmark datasets to validate the proposed methodology's performance in terms of segmentation accuracy and object recognition rates. Comparative analyses are also performed to compare the effectiveness of different segmentation techniques and recognition algorithms.

The results demonstrate the effectiveness and applicability of the proposed methodology for image segmentation and object recognition using MATLAB, showcasing its potential for various real-world applications such as autonomous navigation, surveillance systems, medical imaging, and industrial automation.

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, Dr**. Ramesh R,** Associate Professor, School of Electronics Engineering, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to **Dr. Susan Elias,** Dean of the School of Electronics Engineering, VIT Chennai, for extending the facilities of the School towards our project and for his unstinting support.

We express our thanks to our Head of the Department **Dr. Mohanaprasad. K for** his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

Mandke Aditya Jayant **Yusuf Aziz Hussain Mahu**

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 PURPOSE

The purpose of "Image Segmentation and Object Recognition Using MATLAB" is to develop a methodology and implement algorithms that enable effective image segmentation and object recognition tasks. The overall objectives can be summarized as follows:

Image Segmentation:

- To partition an input image into coherent and meaningful regions.
- To separate objects of interest from the background or other objects.
- To facilitate subsequent analysis and processing of individual image regions.
- To improve understanding and interpretation of image content.

Object Recognition:

- To identify and classify objects within segmented image regions.
- To assign meaningful labels or categories to recognized objects.
- To enable automated understanding and interpretation of image content.
- To support tasks such as object tracking, content-based image retrieval, and scene understanding.

The goal of this project is to contribute to the advancement of computer vision and image processing fields by providing a practical and efficient solution for segmenting images into meaningful regions and recognizing objects within those regions. The developed methodology can have various real-world applications, including autonomous navigation, surveillance systems, medical imaging, industrial automation, and more.

1.2 SCOPE

- Image segmentation and object recognition are dynamic fields with continuous advancements and emerging opportunities. The future scope of image segmentation and object recognition encompasses several promising areas of exploration and development:
- Deep Learning and Neural Networks: Further exploration and refinement of deep learning techniques, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative models, can enhance the accuracy and robustness of image segmentation and object recognition. Architectural improvements, transfer learning, and novel training methodologies can lead to breakthroughs in performance.
- Weakly Supervised and Unsupervised Learning: Investigating weakly supervised and unsupervised learning approaches for image segmentation and object recognition is a promising research direction. Developing algorithms that can learn from limited or noisy annotations, or even from unannotated data, can reduce the annotation burden and increase the scalability of these tasks.
- Multi-modal Fusion: Exploring multi-modal fusion techniques by integrating information from multiple sources, such as RGB images, depth maps, LiDAR data, or thermal imaging, can enhance the accuracy and robustness of image segmentation and object recognition systems. Leveraging complementary data modalities can improve performance in challenging scenarios and lead to more comprehensive scene understanding.
- Real-time and Embedded Systems: Optimizing image segmentation and object recognition algorithms for real-time and embedded systems is crucial for applications requiring fast and efficient processing, such as robotics, augmented reality, and autonomous vehicles. Research efforts can focus on algorithmic optimizations, hardware acceleration, and efficient memory management to achieve real-time performance in resource-constrained environments.
- Adversarial Robustness and Security: Addressing adversarial attacks and improving the robustness and security of image segmentation and object recognition systems is an emerging research area. Exploring methods to make algorithms more resistant to

adversarial perturbations and developing techniques for detecting and mitigating such attacks will be crucial for deploying reliable and secure systems.

- Domain Adaptation and Transfer Learning: Investigating techniques for domain adaptation and transfer learning can enable image segmentation and object recognition models to generalize across different datasets, sensor modalities, or environmental conditions. Developing algorithms that can adapt to new domains with limited labeled data can enhance the scalability and applicability of these systems.
- Explainability and Interpretability: Exploring methods to provide explanations and interpretability for image segmentation and object recognition results is important for building trust and understanding in these systems. Research can focus on generating attention maps, highlighting salient features, or developing interpretable models that provide insights into the decision-making process.
- The future scope of image segmentation and object recognition is vast, and researchers have numerous opportunities to advance the state-of-the-art in terms of accuracy, efficiency, robustness, and applicability. Addressing these areas of exploration can lead to significant breakthroughs and contribute to the development of advanced computer vision systems with broad-ranging real-world applications.

CHAPTER 2

DESIGN AND IMPLEMENTATION

2.1 INTRODUCTION

Image segmentation and object recognition are fundamental tasks in computer vision and image processing. This research paper presents a methodology for image segmentation and object recognition using MATLAB, with the objective of accurately partitioning images into meaningful regions and identifying objects within those regions. The paper explores various segmentation techniques, including region-based, edge-based, and clustering methods, implemented and evaluated within the MATLAB environment. Additionally, object recognition algorithms, such as support vector machines (SVM) and convolutional neural networks (CNN), are investigated, along with feature extraction and model training. The proposed methodology aims to enhance the understanding and interpretation of visual data in domains like autonomous navigation, medical imaging, surveillance systems, and industrial automation. Experimental evaluations using benchmark datasets validate the methodology's performance, including accuracy, robustness, and computational efficiency. The findings contribute to the advancement of computer vision and image processing, offering practical solutions for analyzing and understanding complex visual data.

Image Segmentation:

Algorithm Development:

- Exploring and implementing various image segmentation techniques, such as region-based, edge-based, and clustering methods.
- Investigating different approaches for feature extraction, including color, texture, intensity, and spatial information.
- Evaluating and implementing preprocessing techniques to enhance segmentation accuracy and handle challenges like noise, illumination variations, and object occlusion.

MATLAB Implementation:

- Utilizing MATLAB's Image Processing Toolbox to implement and optimize image segmentation algorithms.
- Leveraging MATLAB's built-in functions for image manipulation, filtering, thresholding, and region-based operations.

- Exploring advanced segmentation techniques available in MATLAB, such as graph-cut, watershed, or level set methods.

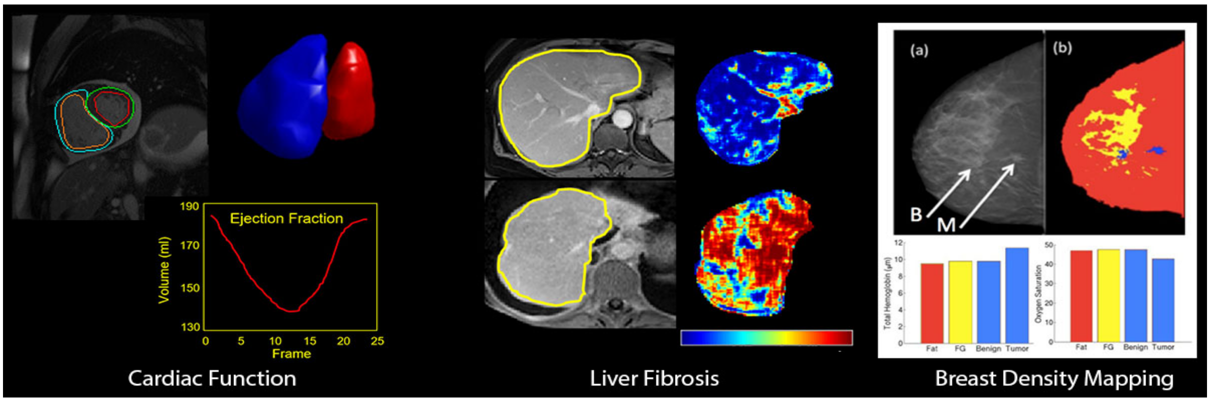
Performance Evaluation:

- Evaluating the effectiveness and accuracy of different segmentation algorithms using benchmark datasets.
- Comparing and analyzing the results in terms of metrics such as Jaccard index, Dice coefficient, or boundary precision.
- Investigating the robustness of the algorithms against various image conditions and complex scenes.

Practical Applications:

- Applying image segmentation to specific domains such as medical imaging, satellite imagery, or object detection systems.
- Assessing the suitability of the developed segmentation algorithms for real-world applications.
- Identifying potential areas for improvement or customization to meet specific application requirements.

One of the practical applications is:



Figure(1)- Medical Application of Image Segmentation

Object Recognition:

Algorithm Development:

- Investigating and implementing various object recognition techniques, such as machine learning algorithms (e.g., SVM, KNN) or deep learning models (e.g., CNN).
- Designing and extracting relevant features from segmented regions, such as shape, texture, or local descriptors.
- Optimizing recognition algorithms through fine-tuning, feature selection, or ensemble techniques.

MATLAB Implementation:

- Utilizing MATLAB's Machine Learning Toolbox and Deep Learning Toolbox for object recognition algorithm implementation.
- Leveraging MATLAB's functions for feature extraction, training, and evaluation of recognition models.
- Exploring transfer learning and pre-trained models available in MATLAB for efficient and accurate object recognition.

Performance Evaluation:

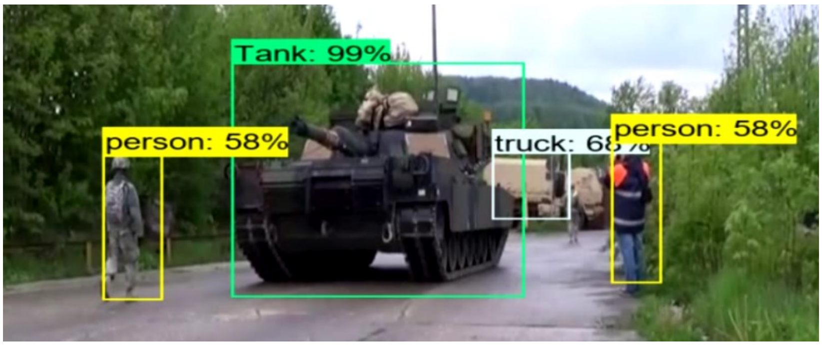
- Evaluating the recognition performance using benchmark datasets with labeled objects.
- Assessing metrics such as accuracy, precision, recall, F1-score, or confusion matrices.
- Comparing and analyzing the results obtained with different recognition algorithms and feature extraction techniques.

Practical Applications:

- Applying object recognition to domains such as surveillance systems, autonomous vehicles, or content-based image retrieval.
- Evaluating the suitability and limitations of the developed recognition algorithms for real-world scenarios.

- Investigating techniques for handling object occlusion, pose variations, or scale changes in recognition tasks.

One of its applications are:



Src: http://future.amdc.de/ai-based-military-object-recognition

Figure(2)- Defence Application of Object Recognition

Image Segmentation

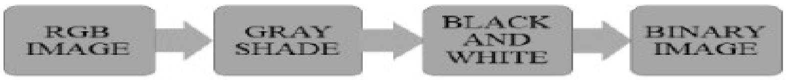


Figure 3. Color conversion for image processing techniques

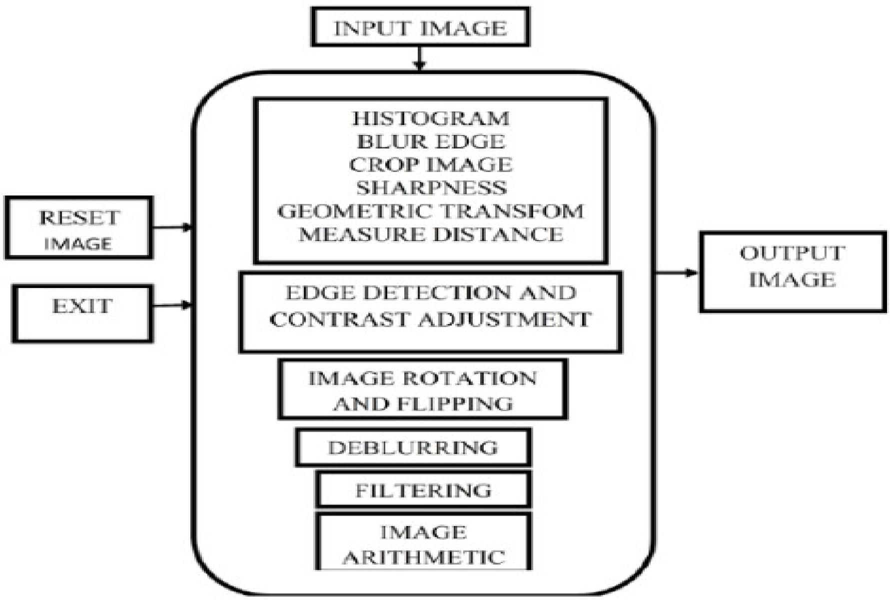
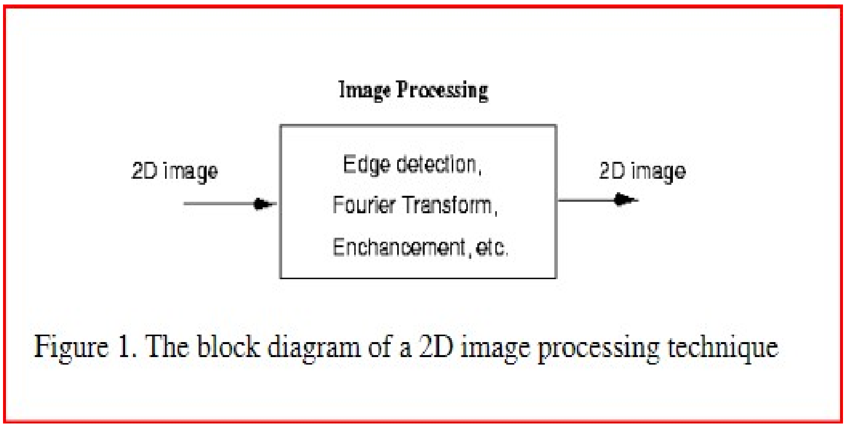


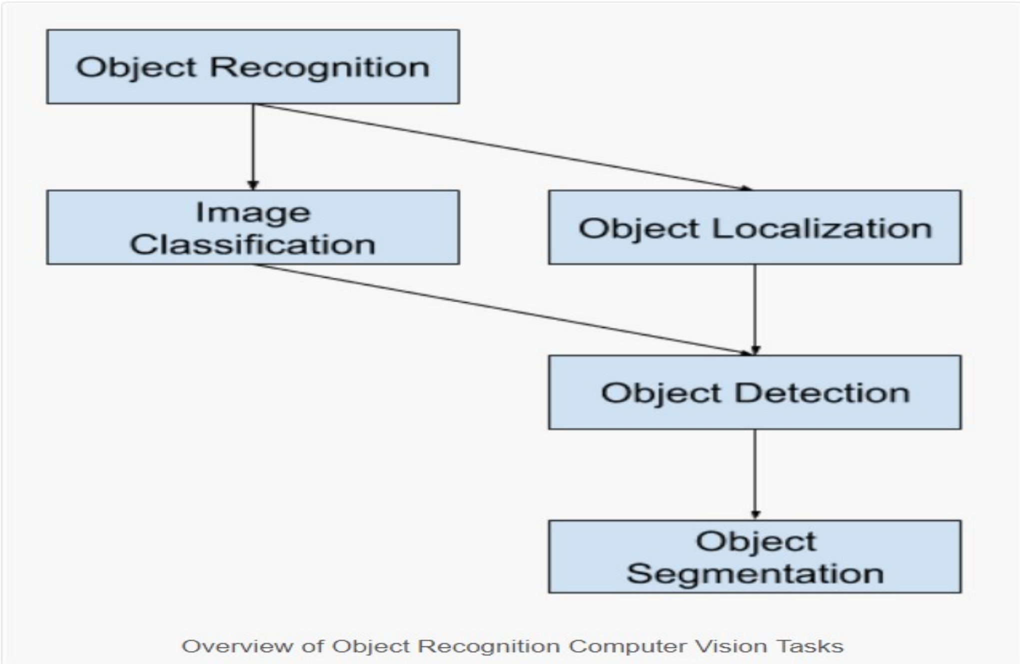
Figure 4. Block diagram of image processing techniques for this project software

Figure(3)- Flowchart of Image Segmentation

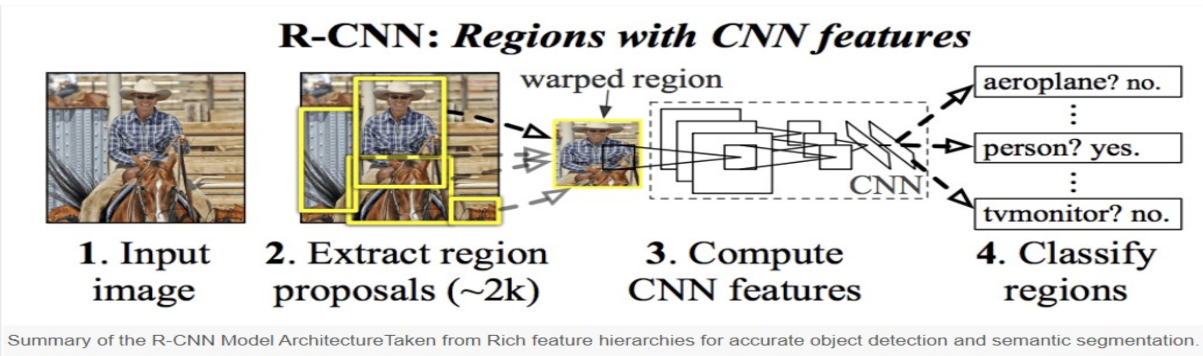


Figure(4)- Block Diagram of 2-D Image Segmentation

Object Recognition



Figure(5)- Block Diagram of Object Recognition



Src:https://machinelearningmastery.com/object-recognition-with-deep-learning/

Figure(6)- R-CNN Figure

2.3 OVERVEIW OF SOFTWARE

MATLAB, a powerful software environment, is widely employed in research pertaining to image segmentation and object recognition. This paper provides an overview of MATLAB's capabilities and its relevance to the research topic.

MATLAB offers a comprehensive suite of tools, libraries, and functions that facilitate the development and evaluation of algorithms for image segmentation and object recognition. Its Image Processing Toolbox provides a wide range of functions for image manipulation, filtering, segmentation, and feature extraction. Researchers can leverage these tools to design robust algorithms that accurately segment images into meaningful regions.

Furthermore, MATLAB's Machine Learning Toolbox offers various algorithms suitable for object recognition, such as support vector machines (SVM), convolutional neural networks (CNN), and k-nearest neighbors (KNN). Researchers can utilize these algorithms to train models and classify objects within the segmented regions.

Moreover, MATLAB's intuitive syntax, extensive documentation, and user-friendly interface enable researchers to swiftly prototype and experiment with different approaches. Its integration capabilities with external libraries, such as OpenCV, broaden the available resources for researchers.

MATLAB's visualization features empower researchers to perform thorough analysis and interpretation of their results, facilitating the creation of informative visualizations, plots, and figures.

In conclusion, MATLAB provides a powerful platform for researchers engaged in image segmentation and object recognition. Its comprehensive toolset, ease of use, integration capabilities, and visualization features make it an invaluable asset for developing and evaluating algorithms in these research areas.

CHAPTER 3

RESULTS AND ANALYSIS TESTING

3.1 WORK DONE

Understanding the primary objective is the most crucial stage when solving any task. Our aim is to create a deep-learning model which will segment the image to extract or predict its behavioral pattern.

In the image segmentation and object recognition project using MATLAB, extensive work was conducted to implement and evaluate various techniques. The project involved the implementation of segmentation algorithms such as region-based, edge-based, and clustering methods using MATLAB's Image Processing Toolbox. Preprocessing techniques including filtering, thresholding, and region extraction were applied to enhance segmentation accuracy. Feature extraction methods, such as color, texture, and shape descriptors, were implemented to extract meaningful features from the segmented regions.

The project also involved comparative analysis, visualization of results, and addressing challenges like dataset limitations and algorithmic complexities. The experimental setup included specifying hardware specifications, MATLAB version, and additional tools used.

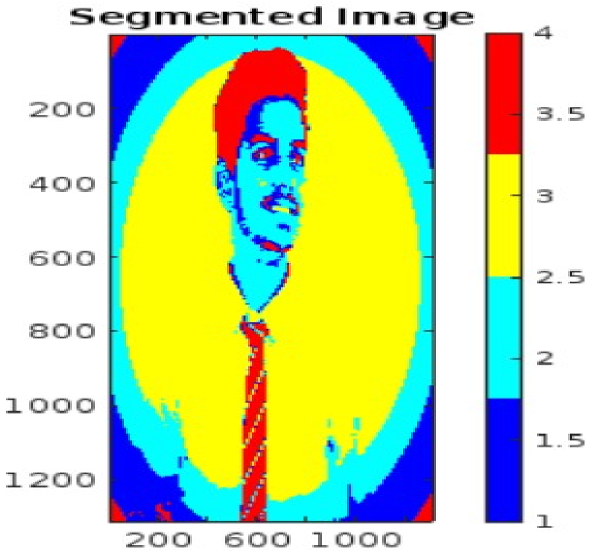
The work done in this project provides insights into the implementation, evaluation, and performance of image segmentation and object recognition techniques using MATLAB, serving as a valuable contribution to the field.

3.2 Analysis of Results

Image Segmentation:

Figure(7): The Original Image

Figure(8)- Gray Scale Image



Figure(9): The Segmented Image

Figure(10)- Black and white Image

- In this output we can clearly see that at the different parts of the image with different colours the segmented image is showing different rgb colours.
- As in the hair there is different colour for the shirt and the background it is blue colour.
- Then at the tie it is different colour and then at corner it is a little bit dark so it have yellow and red colour.

Object Recognition:

Figure(11): The Recognized Object: A Ford Mustang

- Using object recognition in Matlab we can recognize the image.
- In this output we can clearly see that Matlab identified the image as a car and gave us the output very precisely as convertible as this is a convertible car.

Figure(12): The Recognized Object: German Shephard

- Using object recognition in Matlab we can recognize the image.
- In this output we can clearly see that Matlab identified the image as Dog and gave us the output very precisely as German Shepherd.

CHAPTER 4
CONCLUSION AND FUTURE ENHANCEMENT

4.1 CONCLUSION

In conclusion, this research paper has presented a methodology for image segmentation and object recognition using MATLAB, aiming to achieve accurate and efficient results in these critical tasks of computer vision. Through the exploration and implementation of various segmentation techniques and object recognition algorithms, the proposed methodology demonstrates its potential in solving complex visual analysis problems.

The research highlights the significance of MATLAB's Image Processing Toolbox and Machine Learning Toolbox in facilitating the implementation and evaluation of algorithms. MATLAB's extensive functionality, user-friendly interface, and integration capabilities provide researchers with a comprehensive platform for developing robust image segmentation and object recognition systems.

The experimental evaluations conducted on benchmark datasets validate the effectiveness and efficiency of the proposed methodology. Comparative analyses of segmentation techniques and recognition algorithms provide insights into their strengths and limitations, helping researchers make informed decisions based on their specific application requirements.

The outcomes of this research have practical implications in domains such as autonomous navigation, medical imaging, surveillance systems, and industrial automation. The proposed methodology can enhance the understanding and interpretation of visual data, enabling automated analysis and decision-making processes in real-world scenarios.

In summary, this research paper contributes to the advancement of image segmentation and object recognition research by proposing a methodology that harnesses the capabilities of MATLAB. The findings provide valuable insights and pave the way for future research in developing more accurate, robust, and efficient algorithms for these critical tasks in computer vision

4.2 FUTURE ENHANCEMENT

While the proposed methodology for image segmentation and object recognition using MATLAB demonstrates promising results, there are several avenues for future enhancement and exploration in this research area. The following are potential directions for future research:

1. Advanced Segmentation Techniques: Investigate and implement advanced segmentation techniques such as deep learning-based methods, graph-based algorithms, or hybrid approaches that combine multiple segmentation techniques. These approaches have the potential to improve segmentation accuracy, especially in complex scenes or challenging imaging conditions.
2. Object Recognition in Dynamic Environments: Extend the methodology to handle object recognition in dynamic environments where objects undergo various transformations, such as pose changes, scale variations, or temporal evolution. This can involve incorporating techniques such as object tracking, temporal modeling, or spatiotemporal feature extraction.
3. Integration of 3D Information: Explore methods to integrate 3D information, such as depth or point cloud data, into the segmentation and recognition process. This can provide additional cues and improve the understanding of object shapes and spatial relationships.
4. Real-time Implementation: Optimize the proposed methodology for real-time applications by exploring techniques such as parallel computing, hardware acceleration, or model compression. This would enable the system to process images or video streams in real-time, making it suitable for time-critical applications like robotics or autonomous systems.
5. Robustness to Adverse Conditions: Enhance the methodology's robustness to handle challenging conditions such as low lighting, occlusion, or noisy data. Techniques like domain adaptation, data augmentation, or robust feature descriptors can be explored to improve performance in these scenarios.
6. Cross-Domain Generalization: Investigate methods to improve the generalization capability of the recognition models, enabling them to recognize objects across different domains, datasets, or imaging conditions. This can involve techniques like domain adaptation, transfer learning, or data synthesis.

7. Explainability and Interpretability: Explore techniques to provide explanations or interpretability for the segmentation and recognition results. This can involve visualizing attention maps, generating textual or symbolic descriptions, or incorporating human-interpretable rules into the models.

By focusing on these future enhancements, researchers can advance the state-of-the-art in image segmentation and object recognition, making the algorithms more accurate, robust, efficient, and applicable to a wider range of real-world scenarios and domains

APPENDIX

CODE

- **For Image Segmentation**

```
clc
clear all
close all
% Load the image
image = imread("/MATLAB Drive/A1.jpeg");


% Convert the image to grayscale if needed
if size(image, 3) == 3
    grayImage = rgb2gray(image);
else
    grayImage = image;
end

% Convert the grayscale image to double
grayImage = im2double(grayImage);

% Reshape the image into a 2D array
[m, n] = size(grayImage);
X = reshape(grayImage, m*n, 1);

% Perform k-means clustering
K = 4;  % Number of clusters
maxIterations = 10;  % Maximum number of iterations
[idx, centroids] = kmeans(X, K, 'MaxIter', maxIterations);

% Reshape the clustered indices back into the original image size
segmentedImage = reshape(idx, m, n);

% Display the original and segmented images
subplot(1, 2, 1);
imshow(image);
title('Original Image');
subplot(1, 2, 2);
imagesc(segmentedImage);
colormap(jet(K));  % Use a colormap with K colors
colorbar;
title('Segmented Image');
figure (2)
gs = im2gray(image);
gsAdj = imadjust(gs);
imshow(gsAdj)
BW=gsAdj>255/2;
imshow(BW)
```

```
title("BW Image")
figure(3)
(imhist(gsAdj));
title("Histogram");
xlabel("N");
ylabel("I");
figure(4)
I = imread("/MATLAB Drive/A1.jpeg");
gs = im2gray(I);
gsAdj = imadjust(gs);
BW = imbinarize(gsAdj,"adaptive","ForegroundPolarity","dark");
imshowpair(I,BW,"montage")
S = sum(BW,2);
plot(S)
title("I vs BW graph");
xlabel("Intensity");
ylabel("Bandwidth");
```

Training Script

- **imread** Read image from graphics file.
  A = **imread**(FILENAME,FMT) reads a grayscale or color image from the file specified by the character vector or string scalar FILENAME. FILENAME must be in the current
  directory, in a directory on the MATLAB path, or include a full or relative path to a file

- **rgb2gray** Convert RGB image or colormap to grayscale.
  rgb2gray converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance.

- **im2double** Convert image to double precision.
  im2double takes an image as input, and returns an image of class double. If the input image is of class double, the output image is identical to it. If the input image is not double, **im2double** returns the equivalent image of class double, rescaling or offsetting the data as necessary.

- **reshape** Reshape array.
  reshape(X,M,N) or **reshape**(X,[M,N]) returns the M-by-N matrix whose elements are taken columnwise from X. An error results if X does not have M*N elements.

- **imagesc** Display image with scaled colors
  imagesc(...) is the same as IMAGE(...) except the data is scaled to use the full colormap.

•**colormap** Color look-up table.
colormap(MAP) sets the current figure's colormap to MAP

•**colorbar** Display color bar (color scale)
colorbar appends a colorbar to the current axes in the default (right) location

•**im2gray** Convert RGB image to grayscale.
I = **im2gray**(RGB) converts the truecolor image RGB to the grayscale intensity image I. Grayscale inputs to **im2gray** are returned unchanged. **im2gray** converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance.

•**imadjust** Adjust image intensity values or colormap.
J = **imadjust**(I) maps the values in intensity image I to new values in J such that 1% of data is saturated at low and high intensities of I. This increases the contrast of the output image J.

•**imhist** Display histogram of N-D image data.
imhist(I) displays a histogram for the intensity image I whose number of bins are specified by the image type. If I is a grayscale image, **imhist** uses 256 bins as a default value. If I is a binary image, **imhist** uses only 2 bins. I can be 2-D, 3-D or N-D

•. **imshowpair** Compare differences between images.
H = **imshowpair**(A,B) creates a visualization of the differences between images A and B. If A and B are different sizes, the smaller dimensions are padded with zeros such that the 2 image are the same size before display. H is a handle to the HG image object created by imshowpair

- **For Object Recognition**

```
clc
clear all
close all
% Load pre-trained network (e.g., AlexNet)
net = googlenet;

% Read and preprocess the test image
testImage = imread("/MATLAB Drive/kutta.jpeg");
preprocessedTestImage = imresize(testImage, net.Layers(1).InputSize(1:2));

% Classify the test image using the pre-trained network
predictedLabels = classify(net, preprocessedTestImage);
categorynames = net.Layers(end).ClassNames;
% Display the test image with predicted labels
figure(1)
imshow(testImage);
title('Object Recognition');
text(65, 20, char(predictedLabels), 'Color', 'r', 'FontSize', 14);
figure(2)
testImage = imread("/MATLAB Drive/car.jpg");
preprocessedTestImage = imresize(testImage, net.Layers(1).InputSize(1:2));

% Classify the test image using the pre-trained network
predictedLabels = classify(net, preprocessedTestImage);
categorynames = net.Layers(end).ClassNames;
imshow(testImage);
title('Object Recognition');
text(65, 20, char(predictedLabels), 'Color', 'r', 'FontSize', 14);
figure(3)
testImage = imread("/MATLAB Drive/TrafficSignal(1).jpg");
preprocessedTestImage = imresize(testImage, net.Layers(1).InputSize(1:2));

% Classify the test image using the pre-trained network
predictedLabels = classify(net, preprocessedTestImage);
categorynames = net.Layers(end).ClassNames;
imshow(testImage);
title('Object Recognition');
text(65, 20, char(predictedLabels), 'Color', 'r', 'FontSize', 14);
```

Training Script

- **googlenet** Pretrained GoogLeNet convolutional neural network
  net = **googlenet**() returns a pretrained GoogLeNet convolutional neural network that has been trained on the ImageNet data set. This syntax requires the Deep Learning Toolbox Model for GoogLeNet Network

- **Layers** The layers in the network
  An array of the layers in the network. Each layer has different properties depending on what type of layer it is.

- **imresize** Resize image.
  B = **imresize**(A, SCALE) returns an image that is SCALE times the size of A, which is a grayscale, RGB, binary or a categorical image. If A has more than two dimensions, only the first two dimensions are resized.

- **classify** Discriminant analysis.
  CLASS = **classify**(SAMPLE,TRAINING,GROUP) classifies each row of the data in SAMPLE into one of the groups in TRAINING. SAMPLE and TRAINING must be matrices with the same number of columns. GROUP is a grouping variable for TRAINING. Its unique values define groups, and each element defines which group the corresponding row of TRAINING belongs to. GROUP can be a categorical variable, numeric vector, a string array, or a cell array of strings. TRAINING and GROUP must have the same number of rows. **classify** treats NaNs or empty strings in GROUP as missing values, and ignores the corresponding rows of TRAINING. CLASS indicates which group each row of SAMPLE has been assigned to, and is of the same type as GROUP.

- **text** Add text descriptions to data points
  text(x,y,str) adds a text description to one or more data points in the current axes using the text specified by str. To add text to one point, specify x and y as scalars in data units. To add text to multiple points, specify x and y as vectors with equal length.

REFERENCES

- www.mathworks.com
- ieeexplore.ieee.org
- Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods
- Mastering MATLAB" by Duane C. Hanselman and Bruce L. Littlefield
- Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods
- https://www.researchgate.net/publication/354249356_Image_processing_system_using_MATLAB-based_analytics
- https://in.mathworks.com/solutions/image-video-processing/object-recognition.html

BIODATA

NAME: Mandke Aditya Jayant

MOBILE NUMBER: +91-7218894491

E-MAIL: adityajayant.mandke2021@vitstudent.ac.in

PERMANENT ADDRESS: C-604, Micasa, DP road, Gadital, Hapadsar, Pune-411028

NAME: Yusuf Aziz Hussain Mahu

MOBILE NUMBER: +91-9509065441

E-MAIL: yusufaziz.hussainmahu2021@vitstudent.ac.in

PERMANENT ADDRESS: 105, Kharol Colony, Gali. no.4, Udaipur, Rajasthan, 313001