Augmented Map Based Intelligent Navigation System

A Thesis

submitted in partial fulfillment of the requirements for the award of the degree of

Doctor of Philosophy

in

Department of Computer Science and Engineering

by

Baljit Kaur
(Reg no: 901403016)

Thapar Institute of Engineering and Technology
Patiala-147004, Punjab, India
July, 2019

Certificate

I hereby certify that the work, which is being presented in the thesis, entitled **Augmented Map Based Intelligent Navigation System**, in partial fulfillment of the requirements for the award of the degree of **Doctor of Philosophy** and submitted to the institution is an authentic record of my own work carried out during the period **July, 2014** to **July, 2019** under the supervision of **Dr. Jhilik Bhattacharya**. I have also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

The matter presented in this thesis has not been submitted elsewhere for the award of any other degree or diploma from any institution.

Date: 28.02.20

Baljit Kaur
Candidate

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

Date: 28.02.20

Dr. Jhilik Bhattacharya
Supervisor

Abstract

The presented research work aims at augmenting maps with scene information such that it can provide intelligent capabilities. The work hence exploits existing state-of-the-art methods and evolves them to create a robust information representation framework, used for task specific augmentation purposes. For instance, the augmented map will be able to provide a visually impaired user (or a user new to the environment for example) a scene localization information, the amount of traffic or human presence information in the scene, and guidance to destination points. The scene maps are hence augmented with localization features, traffic density information and distance maps for this purpose. In this regard, the primary computer vision tasks like map generation, scene localization, and object detection and classification are revisited . Substantial attention is given to suitable algorithm developments for the scene localization and object detection sub-tasks. Also, the representation framework is created in a manner such that it can be re-used for different sub-tasks with minimal re-computational resource requirements. Throughout this work there are two supplementary goals. While on one hand the primary task is to generate augmented maps for achieving intelligent capabilities, equal amount of attention has been given in reusing and developing the tools for facilitating higher prediction confidence, lower inference time, greater robustness.

Using maps, scene recognition is an important aspect for robot navigation and localization. A trajectory based map has been generated by navigating a mobile robot. The addition of new nodes in this map is based on an user interface. This map is further populated with deep CNN features extracted from the scenes. This work particularly explores scene localization problem using state of the art deep learning models. The success of these deep networks rely on large datasets. Replicating performance on domain specific data using pre-trained networks is quite difficult in most cases primarily due to unavailability of large domain specific datasets. Also pre-trained models incur larger memory requirements and greater inference time. Currently, a 2-step approach has been used for scene localization. In step 1, a zone matching based on deep features classified with integrated set based approach is used. This is further iterated with a capsule based landmark detection in the second step. Particular emphasis has been given on factors like reduced inference time, maximum reuse of information and greater prediction confidence. For facilitating this, practices like finetuning compressed networks, using soft target based training and extending a single background class to GAN based multiple dustbin classes are adapted.

Vehicle detection and classification is another important task for street surveillance and

scene perception for robot navigation or autonomous vehicles. This research work further focuses on traffic detection for real time applications using three components. The first component includes designing convolutional feature map-based classifiers. The second component encourages use of multimodal feature fusion with edge features, scale space features and optical flow features. The third component focuses on training mechanisms and utilizes an effective adaptive learning rate technique to deal with saddle points; and proposes an average covariance matrix based pre-conditioning approach. Special attention has also been given to accommodate blur features in real time.
Generated maps have also been augmented with traffic density estimation. For augmentation, obtaining traffic density information for an area particularly focuses on itinerary perception subject to different environmental conditions. This refers to extraction of traffic related information. The problem is modeled as a machine learning technique where the traffic distribution at different times (including same days,different days,different weather) are observed continuously using a service robot. This data is posed as a gaussian process for post estimation where a Region Of Interest(ROI) input, queried to a database of traffic density distributions, learned from the scenes at different points of time will generate an information pertaining to the region conditioned on environmental and timing events. Finally, case studies related to visually impaired navigation, traffic density estimation on particular routes at different instances, tourist guide for guiding an unfamiliar person to conveniently navigate from one place to another in particular organization and cow detector to detect and locate the cow's position, have been done. It should be noted that these are sample studies and can be extended as individual ones. The feature augmentation framework can be applied for similar activities such as surveillance etc.

Keywords: Augmented Maps, Feature extraction, CNN, vision based navigation Intelligent systems, Deep Learning.

Acknowledgements

First, I would like to express my deep gratitude to my supervisor **Dr. Jhilik Bhattacharya** for her invaluable advice and encouragement at every step of my PhD program. Without her unfailing support and belief in me, this thesis would not have been possible. Her contribution to this thesis goes well beyond her role as an academic supervisor and includes constant support on a personal level without which this journey may never have been completed. And for this, I am truly grateful.

I would like to express my gratitude to our HOD **Prof. Maninder Singh** for his constant motivation and encouragement. I also wish to thank the my research committee members and non-teaching staff of the institute for their help and support. I would also like to thanks to all the teachers and friend from whom I learn the art of happiness and never give up approach. A special thanks to my dearest friend; Ms. Priya Arora, who helped a lot in submission of my thesis.

Finally, I would like to express my sincere and deep gratitude to my parents and family member for their love, encouragement, care and support. Finally thanks to my husband Mr. Randeep Singh for having faith on me and supporting me at every step. Without his support, I could not complete my Ph.D program and finally lot of love to my sons Tanmehar Singh and Eashmehar Singh, they have cooperated and did lot of prayers for the completion of my PhD.

Baljit Kaur

Table of Contents

List of Figures

List of Tables

List of Notations

$0C3fc$	No spatial convolutional but 3 fc layers
$1C3fc$	1 spatial convolutional followed by 3 fc layers
$1M1$	2 convolutional layers
$DS1, DS2, DS3$	Three parts of dataset
$FDS1, FDS2, FDS3$	Extracted features of RPNs of dataset DS1,DS2,DS3
CTS	Caltech dataset
PTS	PASCAL VOC dataset
OTS	Own created dataset
$1LR, 2LR, 3LR$	Three different learning rates for training NoC
I_1C3fc_nLR	Models trained with different learning rates with Intensity (RGB) image features as inputs, n={1,2,3}
$CNN - Int$	Networks used for intensity image
$CNN - Edges_C$	Networks used for edge image using canny detector)
$CNN - Edges_S$	Networks used for edge image using sobel detector)
$CNN - Edges_P$	Networks used for edge image using prewitt detector)
$CNN - Gauss_3$	Networks used for Gaussian image (t=3)
$CNN - Gauss_5$	Networks used for Gaussian image (t=5)
$CNN - Opt$	Networks used for optical flow image
$conv5\_F_{Int}$	Features from $conv5$ of $CNN - Int$
$conv5\_F_{Edges_C}$	Features from $conv5$ of $CNN - Edges_C$
$conv5\_F_{Edges_S}$	Features from $conv5$ of $CNN - Edges_S$
$conv5\_F_{Edges_P}$	Features from $conv5$ of $CNN - Edges_P$
$conv5\_F_{Opt}$	Features from $conv5$ of $CNN - opt$
$conv5\_F_{Gauss_3}$	Features from $conv5$ of $CNN - Gauss_3$
$conv5\_F_{Gauss_5}$	Features from $conv5$ of $CNN - Gauss_5$
F_{Edges}	Fusion of intensity and edge features
F_{Gauss}	Fusion of intensity and scale space features
F_{opt}	Fusion of intensity and optical flow features

List of Abbreviations

ADAS	Advanced Driver Assistance Systems
AR	Augmented Reality
AOA	Angle Of Arrival
BoW	Bag-Of-Words
BPSK	Binary-Phase-Shift Keying
CDL	Covariance Discriminative Learning
CIR	Channel Impulse Response
CNN	Convolutional Neural Network
CSI	Channel State Information
DCNN	Deep Convolutional Neural Network
DRL	Deep Reinforcement Learning
GIS	Geographic Information System
GNSS	Global Navigation Satellite System
GP	Gaussian process
GPS	Global Positioning System
GUI	Graphical User Interface
HERML	Hybrid Euclidean-and-Riemannian Metric Learning
HOG	Histogram of Oriented Gradients
KNN	K-Nearest Neighbors
LBP	Local Binary Patterns
LDA	Linear Discriminant Analysis
LED	Log-Euclidean Distance
LRF	Laser Range Finder
LOS	Los Of Signal
NoC	Network on Convolutional
PCA	Principal Component Analysis
PLS	Partial Least Squares
RCNN	Region based CNN
RFID	Radio-Frequency Identification
RSSI	Received Signal Strength Information
SGM	Single Gaussian Model
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localization And Mapping
SURF	Speeded Up Robust Features

TDU	Tongue Display Unit
TOA	Time Of Arrival
TDOA	Time Difference Of Arrival
UGV	Unmanned Ground Vehicle
UNB	Ultra Narrow band
UWB	UltraWide band
VS	Visual Servoing
WLAN	Wireless Local Area Network
WNN	Weightless Neural Networks
YOLO	You Look Only Once

Chapter 1

Introduction

Technology has now made such a level of advancement that human experts often prefer to involve intelligent systems in decision-making processes of most complex situations. Intelligent systems are technologically advanced software often supported by sophisticated hardware which recognize as well as respond to the surrounding world. Intelligent systems have the capacity to learn from experience and the ability to adapt according to current data. Examples of such systems vary from automated vacuums, facial recognition programs to Amazon's personalized shopping suggestions. These systems exist all around us such as in traffic lights, digital televisions, smart meters, digital signage, airplane controls, automobiles and many more. There are examples of intelligent systems for smart classrooms [1] used by teachers in distant education. The educator can write straight on a wall-size media board or use voice and gesture to engage remote learners in the class conversation. This can be possible using multi-agent system such as OAA (Open Agent Architecture), hand-tracking system, speech recognition system and many more. Tutoring systems [2–4] are other examples aimed at providing learners with instant and tailored instruction or feedback, generally without the need for human teacher intervention. These Intelligent tutoring systems used reinforcement learning to customize their instructions according to the student's need. One of these is shown in Figure 1.1c. Some of the popular intelligent personal assistants include Apple's Siri, Amazon Alexa, Samsung Bixby etc. The female voice-activated assistant interacts with the users to assist them to send messages, find information, add events to the calendar, make voice calls, get directions and open applications. They uses machine-learning technology to understand natural language questions and requests. Among automobiles based examples, electric car developed by tesla [5] [6] is one of the optimum automobiles available till date. This particular car is able to achieve many accolades as well as has features like absolute technological innovation, predictive capabilities and self-driving.

Shifting our focus to vision based examples, which indeed is our main focus. The OrCam MyEye[7] is a portable, artificial vision device that allows visually impaired people to understand text and identify objects through audio feedback, describing what they are unable to see. Gesture based systems are available for appliances in smart homes[8] in which there are power saving systems to switch ON the lights only in presence of

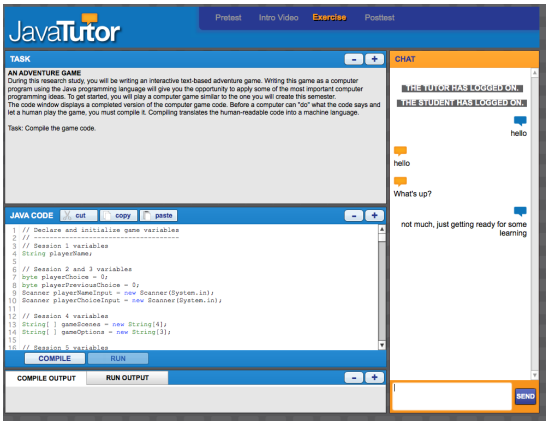
people or based on light intensity[9]. Augmented reality (AR), an important application of intelligent systems, and is used for various business purposes (like shopping, tourism etc.) through mobile applications. MekaMon: an AR robot[10] is one of the first gaming robot which is a real life battle-bot with next-gen augmented reality gameplay. L'Oreal makeup[11] gives an impression of what the retailer's products look on the face. ARGON4[12] is a full-featured web browser that includes the ability to display augmented reality content. AR games like Pokemon Go[12] is an augmented reality mobile game in which pokemon appears to be in the player's real-world location. AR is an interactive experience where the real world objects are augmented by computer-generated information as shown in Figure 1.1a in which AR app has been used to get information about buildings located at a particular place. Augmented map is another significant example which presents a framework which is much more comprehensive to the users. Some notable augmented Maps include augmented paper maps [13] which contain geographic data of the city, transit maps[14] containing flood information and location of the river shown on the map. Figure 1.1b depicts an example of augmented map.

Another important intelligent system which is an addition to the social cause is Guided navigation. It is of paramount concern primarily for visually impaired people. They generally use canes in order to detect obstacles, elevation changes, and memorize the layout or topography while travelling. They rely solely on this crude survey when subjected to unknown areas. With technological advancements and sophisticated developments in the field of sensors, many guidance systems have been developed and employed for the visually impaired, complementing canes, and giving alarms. Some of these include a "smartphone-based navigation system" (ARIANNA)[15] for both indoor and outdoor environment, harness-vest[16], belt[17], helmet[18], Ultrasonic smart glasses[19], etc. Many other types of wearable navigation aids have been developed for assisting the visually impaired. Some of these are shown in Figure 1.4.

As mentioned above, in this study, we aim towards vision based intelligent systems. Of these, we particularly focus on the use of augmented maps for the purpose. At a fundamental level this refers to a representation of information from a mapping of low level features obtained using visual sensors to its high level semantic classes. We begin by reviewing map generation and navigation , particularly focusing on the types of features used for the map generation algorithms and the type of intelligent activity these conduct. We further move towards augmented maps.

(a) AR

(b) Augmented Map



(c) Intelligent tutoring system

Figure 1.1: Examples of Intelligent Systems

1.1 Navigation

Map building and Navigation systems, use different kinds of vision as well as non-vision sensors. Till now GNSS (Global Navigation Satellite System) devices have been widely used as navigation aid to move from one place to another. Although they perform well in localization, for specified environment within particular range of distance; yet, they lack in various conditions. For example, energy requirements of good quality GNSS is very high, which is troublesome for battery-powered devices, such as sensor nodes. In addition, it may take even minutes for GNSS receivers to capture and lock sufficient satellite signals to predict user place when the GNSS receiver is switched on, which may not be appropriate for situations that require no delay. Further, these devices were not able to provide good accuracy in urban and indoor environments. List of methods that can be used for Indoor and outdoor navigation has been shown in Figure 1.2. Some of the

methods listed under indoor navigation can be used for outdoor navigation also. However, the methods listed under outdoor navigation can not be used for indoor navigation.

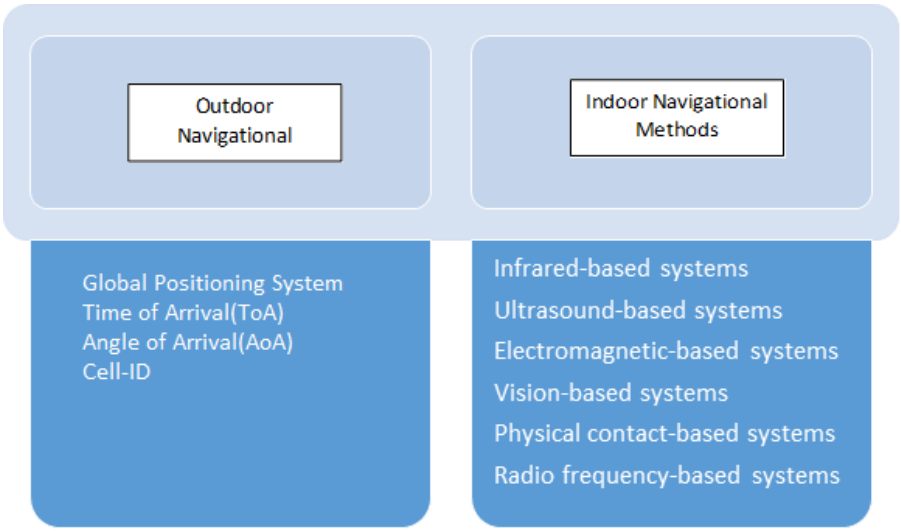


Figure 1.2: List of indoor and outdoor methods for navigation

In early 2000s, object detection for navigation was done using sensors such as ultrasonic, infrared, Received Signal Strength Information (RSSI), UltraWide band (UWB), RFID and many more. Although these used to give only absence/presence of object and/or its distance; however these are unable to give type of object which is very important. The other drawback, owing to obstacles along their route, was the uneven radiation patterns. These obstacles resulted in various routes of radiation, reflections of radiation and attenuations of signal. This resulted in the generation of false signal receptions from the receivers. Hence, the accurate position detection got degraded. Hence, to be concluded, integrated systems for example; RFID in combination with other technologies ie. Infrared, ultrasonic, etc. can give better results for indoor positioning systems [20]. There are many systems that provide autonomous navigation which is the process of roaming safely following a path from start and end point. Different sensors were used for this objective in mobile robotics, which resulted in a variety of alternatives. Sensors such as laser range finders[21], sonars[22] and radars[23] have been utilized in autonomous navigation methods. These sensors are very suited for detecting obstacles and are easy to use due to the reason that they evaluate the distances directly from the robot to obstacles. Many such sensors and their works have been presented in Table 2.1 in chapter 2. However, these sensors are costly as well as they do not provide any information about the objects' type. Hence vision based sensors are preferred in such cases.

1.1.1 Vision based Navigation

Vision sensor based map building and navigation require quality cameras which are now affordable, small and provide real-time high-resolution information. They are passive and do not interfere with other sensors. They are used to detect obstacles and use human-defined rules to identify forbidden areas and navigate mobile robots which range sensors cannot do. Such forbidden areas are in the same plane as the path and are not obstacles but should be considered as non-traversable path. A detailed literature review for vision based navigation systems has been conducted and provided in appendix in Table 6.1. A categorical summary of the same has been provided here in form of sensors used, features extracted, hardware involved, types of maps and application area.

Type of sensors used: Some researchers have used fusion of camera with other sensor such as laser [24] [25] [26] [27] for finding distance whereas others have used only cameras of different types such as omnidirectional cameras [28] [29] [26], pinhole camera [30] [31], fisheye camera [32], stereo camera [33] [25] [34], RGBD camera [35] [36] [37] etc for their research.

Type of features: A lot of map building has been reported with handcrafted features. Some examples include use of SIFT [38] features to encode visual maps for estimation of robot's position further used for landmark based navigation, canny edge detector for extracting quadrangles for the sake mobile robot navigation in open cluttered and corridor-like spaces[39] [40]; and HOG features for mobile robot navigation[41], FAST features for creating map which was further used for navigation system for blind[37]. Later, with the increased use of deep neural networks, deep features such as features from RESNET[42] have been used to train CNN for specified goals, such as "go to a chair". Others have used deep reinforcement learning [43], supervised learning [44] and deep learning for target driven navigation [45] [36] . Figure 1.3 shows the chart for switching from handcrafted features to deep features with the passage of time.

Hardware involved: Now a days, Consumer hardware are available with computational power required by image processing techniques. For these reasons, vision-based navigation for mobile robots has been a widely researched topic in recent years. Researchers have used Vision-based navigation in mobile robots such as CyCab[30], RobuCab [32], LEFKOS: the RWI B21r[46], Nomadic XR4000[40], ExaBot[47] while others used simulation [48] [38] [49] [42] [50] [51]for the purpose. Some of the them have considered indoor environment [28] [52] [40] [33] [29] [37] [53] [54] [36] [35], while some worked for outdoor [55] [56] [57]. Both indoor and outdoor environment are reported in [26] [58] [47].

Type of maps: For various environments, different kinds of maps such as 3D maps, topological maps etc., have been generated and further used for the navigation purpose.

3D Local Voxel Grid Maps (LVGM) have been used to detect obstacles in 3D space and also classified into occupied, free, and unknown states of the navigational environment [37], 3-D map for a mobile robot to navigate in an unidentified environment supervised by a CCD camera[27] mounted on the ceiling, 3D map has also been generated for tested area to analyse the position accuracy of the navigation system[59]. Topological maps have been built in different ways such as considering edges as navigable paths and the nodes as open areas [47], considering every image as one node of the map, a topological map composed of a set of vies obtained from omnidirectional images acquired and organized autonomously by the robot in its visual memory[28], to generate topological maps, occupancy information was extracted directly from the noisy sparse point cloud.

Applications: The application areas of these vision based navigation systems range from autonomous vehicles in urban environments[60] [61] [62] [59], agricultural robots [63] [25] to map generation for Intelligent transport system [64] [65]. It is very helpful in designing devices for visually impaired people [66] [37] [54] [35].

Many vision-based navigation devices are incorporated with augmented maps for better understanding of the surrounding area. Rather augmented maps is the need of devices developed for visually impaired as they can feel environment friendly by getting the complete information of the area they are moving in. Hence, the applications such as augmented maps and blind aid systems have been detailed further in next section.

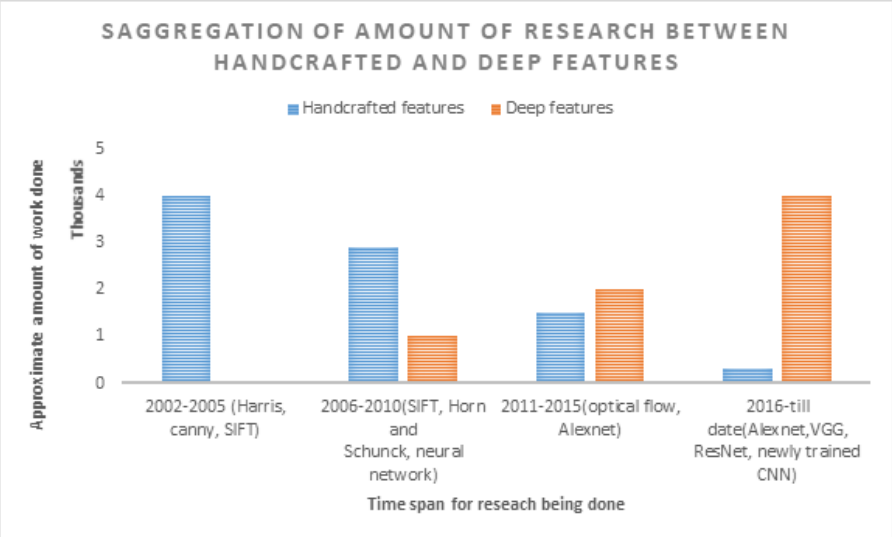


Figure 1.3: From handcrafted features to deep CNN features

1.1.2 Applications

There are various applications of Vision based navigation. Work done on two of them have been discussed below:

1.1.2.1 Augmented Maps

Generation of Augmented Maps has evolved in the last few years in tandem to the development of Augmented Reality devices required for interfacing. These maps presents a representation framework which is much more comprehensive to the users. Some notable research work on Augmented Maps presented in the last few years include augmented paper maps with GIS, cartographic maps, transit maps and augmented areal earth maps. Yang et al. [67] presented a framework for augmenting different paper maps of a city with GIS(Geographic Information System) data for that city. Paper augmented maps have been enhanced to paper based cartographic maps which can be used with a pointing device for augmenting information. For example, a river shown on the map will provide exact location and different stages of the flood information[14]. Paelke et al.[68] have also explored the augmented paper maps' design where maps were augmented with extra functionality by means of mobile device to accomplish significant integration between device and map combining their respective strengths. They followed a user-driven approach that collaborate usability engineering methods with agile software development methodologies. In the context of outdoor use, they highlighted the comparison between paper maps and maps on electronic portable handheld devices in terms of overall characteristics, content presentation and use. Interaction properties indicated that they have strengths and weaknesses complementing each others'. Paper maps are inexpensive and useful for different functions, but lack vibrant and interactive characteristics such as electronic maps. Handheld maps have a strong potential for presenting up-to-date dynamic content, specifically tailored to the user, his present position, and the task at hand, but often have issues with resolution, ease of control, and reliability. An integration of paper maps and electronic devices combining their respective benefits are very attractive. Likewise, Morrison et al.[69] have also used augmented reality with paper maps collaboratively using MapLens. Work has also been done on generating transit maps that allows people to interact in public transit stations and vehicles. Matei et al. [13] presented the application that recognizes a map picture acquired with a phone camera from the database and overlays relevant realtime navigation information such as the user's current location and the time to reach the destination. Kihwan et al. [70] presented augmented aerial earth maps that provide live information of the city which may include weather

details, ongoing festival, matches played and many more. Research has also been done on generating human augmented maps. Here, a robot moves to different locations and collects topography. This is augmented at various levels with room interior information[71]. Chien et al.[72] proposed an approach to join local maps to a global map. This work aims to utilize the concept of map augmentation for developing a system which can store the traffic density of any region at various instants and later provide an estimation of the density statistics. This will help user to analyze the general gatherings of different regions. The brief of research work done on augmented maps has been presented in Table 1.1.

Table 1.1: Some highlights of research on augmented maps

Ref Year	**Sensors/Devices**	**Type**	**Application**
[14] 2005	Pointing device	Paper based cartographic maps using digital graphical information	Flood control
[71] 2006	Service robot	Human augmented map using Topographical data	localization
[70] 2009	Camera	Augmented Aerial Earth Maps	Surveillance
[68] 2010	Augmented paper maps device	Augmented paper maps	Location-Based Services (LBS) and tourism
[69] 2011	MapLens	Collaboration of paper map and Mobile application	Navigation Gaming
[13] 2011	Mobile	Real time maps	Navigation
[72] 2013	Robot	Combination of local and global maps	Localization
[67] 2015	Camera	Paper maps using Geographic Information System	Road Intersection and tracking

1.1.2.2 Blind Aid

Another most important application of vision-based navigation is blind aid systems. It is of utmost necessity to know the surrounding environment and have the knowledge about the probable obstacles around one's self. Many researchers have worked on vision-based navigation for visually impaired people. A number of devices are developed for visually impaired people to provide them information about presence of obstacles, types of obstacle, their distances etc. This information is further utilized to assist visually impaired people for navigating safely, both indoors and outdoors. A smartphone based navigation system(ARIANNA)[15] for both indoor and outdoor environment is available for visually impaired. Work contributed by different researchers in this area can be discussed in terms of sensors used for input; output representation type; and hardware gear, used for either or both. In most of the cases,the scene is perceived using ultrasonic sensors[73] or by extracting images/videos using vision sensors[74]. The output of the above systems(provided in Table 1.2) can be in the form of tactile image[74]; tongue display through voltage pulse[75]; sound patterns/musical auditory information[76, 77] etc. Common wearable helping aids for visually impaired include:

1. Wearable tactile harness-vest display[16] to give instructions about directional navigation using six vibrating motors.

2. A belt[17] associated to a computer along with ultrasonic sensors gives acoustic response in guidance mode, where the system knows about the target and user is guided using tactile signal. In image mode, the user is informed about the environment using tactile image. It translates visuals of the scene into tactile or acoustic information to facilitate safe and swift foot steps.

3. Helmet[18] mounted with ultrasonic chips and speakers. It amplifies echoes produced by ultrasonic sounds for locating objects in space.

4. Ultrasonic smart glasses[19] use ultrasonic waves to detect obstacle.

Many other type of wearable navigation aids have been developed for assisting the visually impaired. Some of these are shown in Figure 1.4. A schematic representation displaying various types of work done for the visually impaired during various time periods is given in Figure 1.5. It is observed that since $1970's$, scene perception via TDU(Tongue Display Unit), tactile images, sound patterns has been done. While in the last decade the commercialization of a lot of wearable devices like Google glass, helmet, finger reader and many more has been witnessed.

Table 1.2: Blind aid technologies using different sensor inputs

Ref. Year	Results	Device used	Method
[74] 2003	Voice message about visual scene	Braille	Convert camera image to tactile image.
[75] 2011	Creates real-time tactile images on the tongue	Tongue display unit (TDU)	Generate programmable pulse to deliver message through a matrix of surface electrodes.
[78] 2008	Allow communication between deaf-blind persons.	Mechanical fingers used to transmit Braille symbols	Finger Braille recognition system
[16] 2006	Navigation Assistance	harness-vest	Convert navigation information into tactile inputs.
[17] 1993	Detect Obstacles	• Ultrasonic sensors • Stereophonic headphones	The acoustic signals are transmitted as discrete beeps or continuous sounds.
[18] 2015	Navigation aid and object perception	Ultrasonic chirps	Amplifies echoes produced by ultrasonic sounds to locate objects.
[73] 2000	Navigation aid	• Ultrasonic transmitter • Two microphones	Translate echoes in sounds for scanning objects.
[76] 2007	Audio output for visual input	Webcam	The Vibe device converts a video stream into a stereophonic sound stream.
[77] 2014	Visual information via musical auditory experience.	Camera	Shape, location and color information was given using sound.
[79] 2012	Detect walls, openings, and vertical roads	• IR • LED • Photodiode	Pulses emitted by LED, retro diffused light detected by the photodiode.
[80] 2015	Obstacle(not type)	Glasses-type vision camera	Deformable Grid
[81] 2015	Information about • Moving objects • Static objects • Audio warning	• Electrode matrix • Mobile Kinect • RF transmitter	The color image, depth image, and accelerometer information provided by Kinect
[82] 2017	Give voice message for facebook feeds to the blind user.	Facebook	Artificial intelligence
[83] 2018	Helps to avoid obstacles and give its approximate distance	Ultrasonic sensors	Ultrasonic signals
[19] 2018	Indication of obstacle with distance <=300cm using buzzer	Ultrasonic Smart glasses	Ultrasonic waves.
[84] 2018	Detect obstacle with distance	• Sonar belt • IR sensors	Infrared light

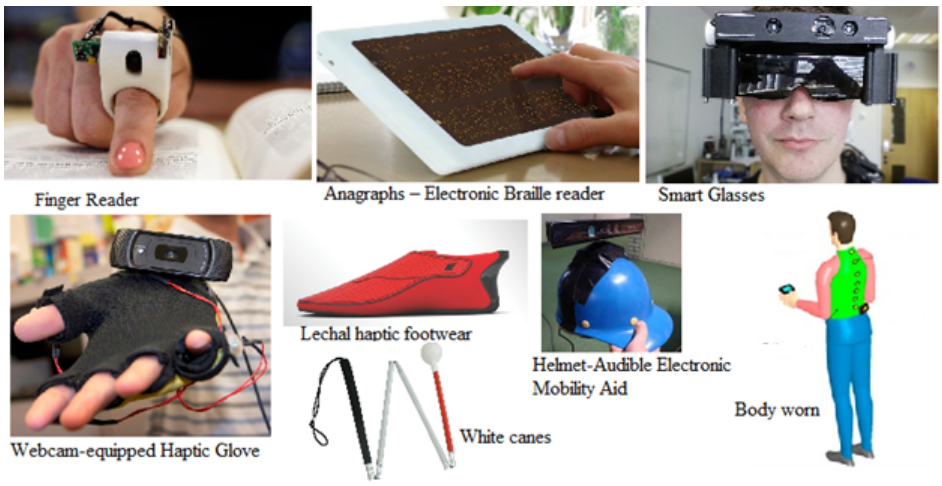


Figure 1.4: A few assistive technological devices for blind aid

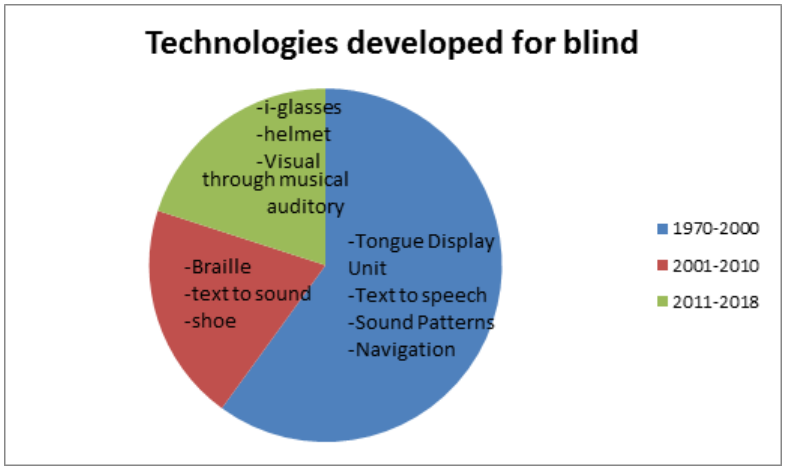


Figure 1.5: Time span representation of usage of particular technologies

1.2 Research Motivation

The increased use of deep learning for different applications motivated us to use deep features for generating augmented maps which can be reused for a wide range of purposes. We covered the following points in our study.

- Information reuse: Extracted and represented information in a way such that it could be efficiently and easily reused for varying purposes. For example, the data extracted from a scene such as vehicles, pedestrians, buildings, etc can be used for localization, traffic detection, density detection for disabled navigation, tourist guide, etc.

- Hybrid Maps: Various map representation such as 3D and topological maps have been reported in literature. Motivated from the methods of generating maps and their accurate usage, we have opted for a hybrid map generation approach using a mobile robot. The hybrid nature of the map is influenced by its different sources of information, i.e. trajectory based path information and deep features based scene information.

- Deep Learning Architecture: There is lot of success reported in literature for the use of convolutional neural network. Critical success particularly depends on the type of architecture. The design of a deep network for object classification plays an important role. While exploring the network design for classification, it was analyzed that not only the deep feature maps, but also a deep and convolutional per region classifier [85] was taken into consideration due to its special importance for object detection. Thus, for the purpose of this research work also, various per region classifier architectures were explored. Further, network compression and finetuning options for faster inference time were considered.

- Features: The features provided to the network for classification are also important. Some of the researchers have used edge images or depth data along with RGB images separately. We explored the scope of optimizing the feature representation of a network by focusing on multimodal and multiscale features. Fusion of RGB features with other features such as edge, scale and optical flow features are explored.

- Data Preparation and Outlier Handling: From a huge amount of data that has been collected while data acquisition, usage of data in an optimal way is required to represent the model. Less amount of data will not result in good accuracy. Also for using video data, a lot of similar frames could be captured which might have resulted in non-homogeneity. In such cases low gradients are generated during learning which lead to slow or no convergence. All these factors are considered by exploiting suitable pre-processing based on FFT , key frame selection techniques on data-set. Further adaptive learning rates were used to deal with saddle points. Further blurred images captured due to random movement of camera can have a negative impact on the network. So, we need to carefully eliminate blurred images or remove blur effect from images [86] [87] [88] or train the network using blurred images. Also, alternatives to single background class to improve network verification ability are explored.

1.3 Objectives

The following research objectives are formulated:

1. To better understand and explore various concepts, techniques available for vision based navigation of a robot, considering application specific improvement scopes.
2. To generate models which map traffic patterns on particular routes at different instances.
3. To build vision and trajectory log based maps using the mobile robot.
4. To augment the maps with the generated models for imparting human experience based learnings to the intelligent system.
5. To develop intelligent vehicle navigational capabilities using the augmented maps.

To achieve first objective Various concepts and techniques have been studied and reviewed for vision based navigation. This has been used for various purposes such as agricultural, underwater surveillance, for visually impaired, for generating augmented maps and many more. Different feature extraction techniques utilized in each case were particularly observed. A note of basic assumptions, constraints was made to understand the scope of adaption, reuse and development of these techniques.

To achieve second objective The process has been implemented using 2 components that are object detection and scene localization. Both these components require robustness in terms of different environmental and structured conditions. Hence, these focus on variant data collection and unique feature representation to incorporate the same. A thorough study of successful object detectors and scene localizers was carried out and suitably adapted and enhanced for our purpose. The object detector has 3 sub-parts i.e. (1)designing convolution feature map based classifier. (2) utilizing multimodal features for deep feature extractors with suitable fusion techniques, (3)utilizing an effective adaptive learning rate technique to deal with saddle points; and proposing an average covariance matrix based pre-conditioning approach and The second component includes 2 sub-parts i.e. (1)engaged integrated distance classifiers for set based image classification technique for detection of particular location, (2)the location information was further fine-tuned with a landmark classifier build using a capsule network.

To achieve third objective Vision and trajectory log based maps have been gener-

ated by utilizing XS80 WiRobot. Data has been collected by navigating the robot and simultaneously capturing the images during different days and times reflecting weather variations. While navigating, the trajectory map of the robot is updated with nodes. Each node tags the visual data along with it. Point cloud maps have been generated to represent the visual data. The node adding process is done via human interaction with the robot control module.

To achieve fourth objective Obtained itinerary perception subject to different environmental conditions. This refers to extraction of traffic related information from an augmented map. Maps generated in the previous step are equipped with the object detector for traffic estimation and scene localizer for region-wise information modeling. The problem was modeled as a machine learning technique where the traffic distribution at different times (including same days,different days,different weather) were observed continuously using a service WiRobot. This data was posed as a gaussian process for post estimation of traffic density distributions, learned from the scenes at different points of time.

To achieve fifth objective Sample case studies have been performed for developing intelligent vehicle navigational capabilities using the augmented maps. Various modules implemented for this work such as object detection, scene localization, map generation & density augmentation have been used for sample applications through these case studies. These include navigation system for blind, traffic density estimation, cow tracker and tour guide.

1.4 Chapters' Outline

The thesis is organized into 6 chapters. A brief outline is given below:

Chapter 1: Introduction This chapter provides an overview of Augmented Maps along with vision based navigation systems.Various concepts and techniques have been studied and reviewed for vision based navigation including feature extraction techniques, scene localization techniques and map generation & augmentation techniques. Applications of vision based navigation have also been highlighted.

Chapter 2: Vision and trajectory based Map Generation In this chapter, map generation for scene localization is discussed in details along with explanation of robot specifications. WiRobot X80 has been used for the purpose. WiRobot has been navigated in Thapar Institute of Engg. and Technology (TIET) campus. Instructions set having distance and directions has been used to generate the tra-

jectory map and point cloud maps have been used to represent the scenes. The node addition process is done in an interactive manner while navigating the robot. The details have been shown in Figure 1.6.

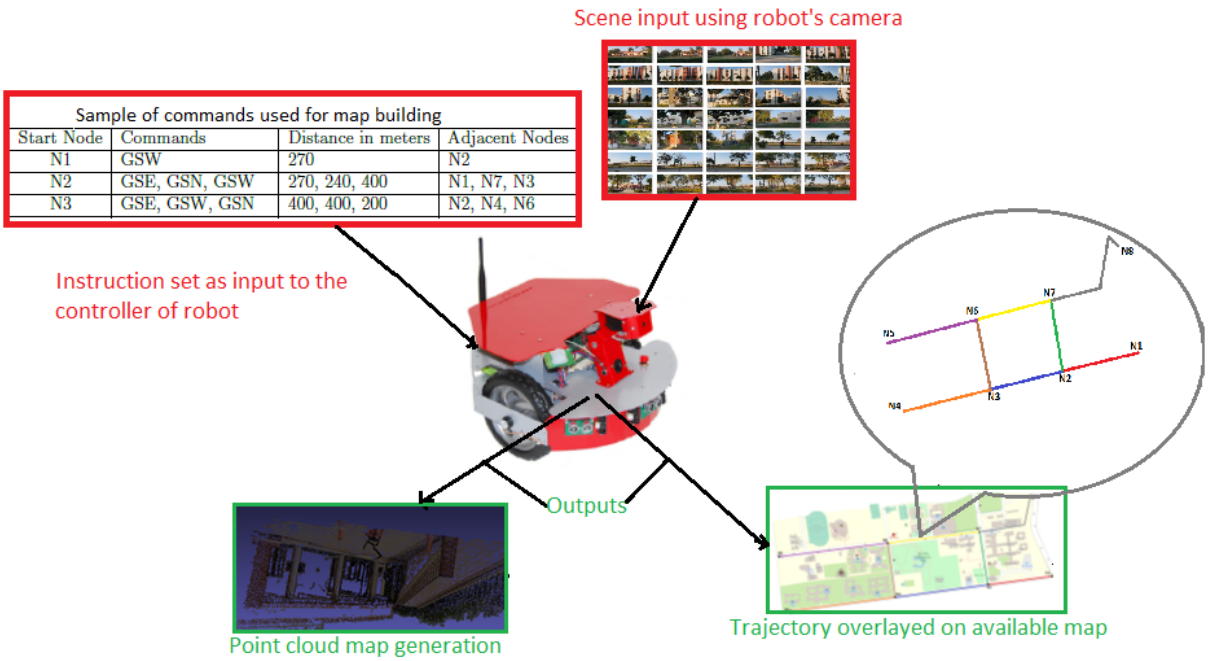


Figure 1.6: Process of map generation using robot

Chapter 3: Scene Localization Figure 1.7 presents the overview of this chapter. It includes scene localization that has been done using set based zone classification and landmark detection. For set-based zone classification, deep features from bottleneck layer of places CNN have been extracted. Further using set based difference technique, matching of scenes has been done. An integrated distance classifier for set based image classification for the detection of particular location has been proposed. This classification has been done using aggregate distance of three distances obtained from three algorithms that are COV+LDA, COV+PLS and HERML. Further landmark was detected by training a capsule network with building regions extracted from a compressed building detector. This detector is a compressed VGG16 network trained with six classes such as Byke, Bicycle, Person, car, tree, background along with building classes. Further to improve the confidence of the landmark detection, a second capsule network was used. This network uses positive and negative images of the biased classes. The negative images referred as dustbin classes are generated with Generative Adversarial Networks (GANs).

Chapter 4: CNN based Object Detection and Classification This chapter describes experiments and observations about building an object detector. These experiments

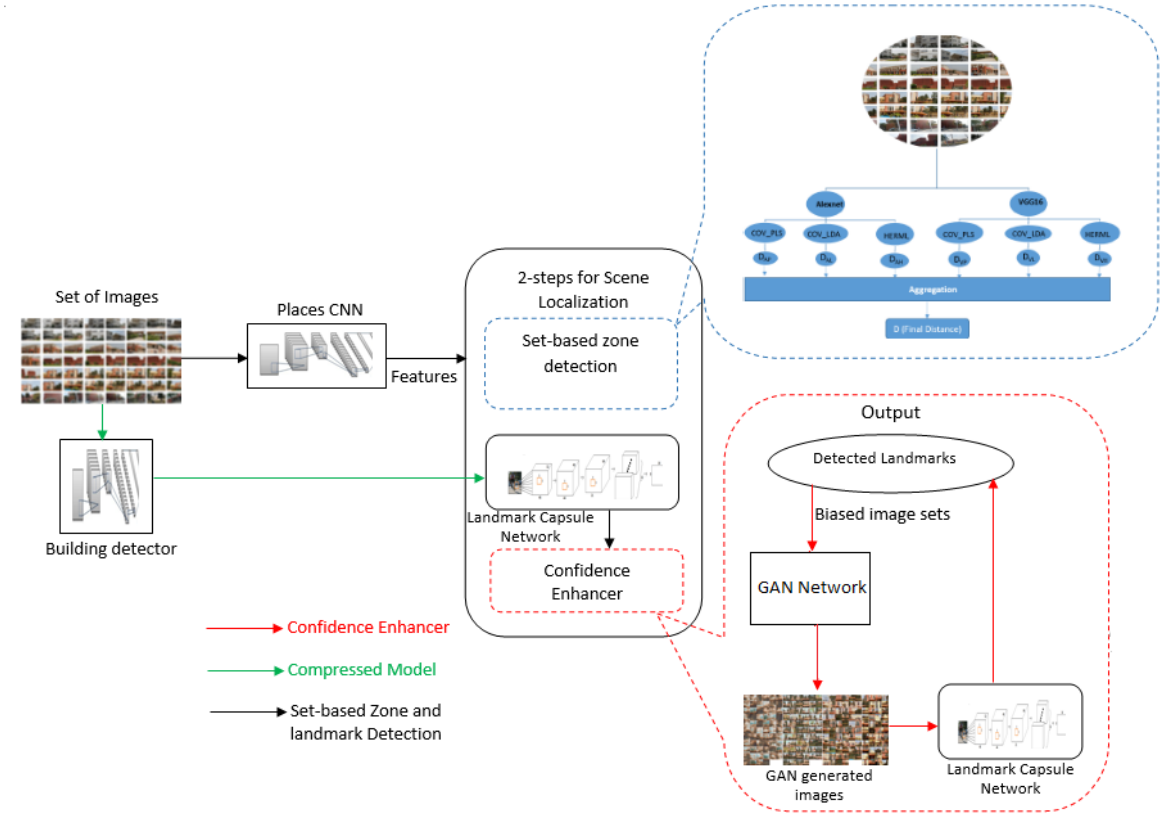


Figure 1.7: Block Diagram of work done in Chapter3

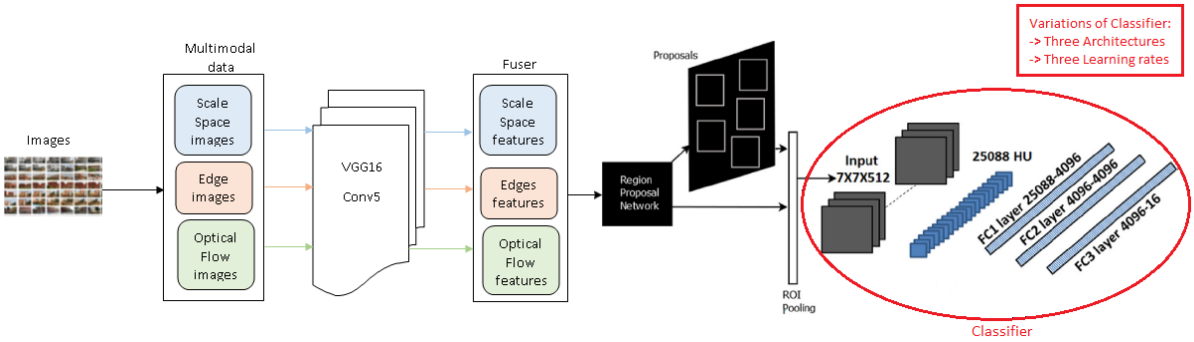


Figure 1.8: Block Diagram of work done in Chapter4

have been performed using normal and blur image features. Different architectures of Network on Convolutional (NoC) feature map classifier have been presented. The convolutional feature map-based classifier was based on multimodal features, fusion using edge features, scale space features and optical flow features. The use of an effective adaptive learning rate technique to deal with saddle points proposing an average covariance matrix based pre-conditioning approach was proved beneficial. The block diagram has been presented in Figure 1.8. Results have been compared with various pre-trained networks such as R-CNN[89], Fast R-CNN[90], Faster R-CNN[91] and YOLO[92]. Various experiments were performed with different learning rates, architectures using benchmark datasets such as Apollo[93], KITTI[94], Cityscapes[95], Berkeley[96], Caltech, PASCAL VOC and self created. Experimental results demonstrate that in comparison to fully connected network based classifier, Network on Convolutional (NoC) feature map classifier provided high classification accuracy.

Chapter 5: Case Studies The scene localizer, object detector and augmented map were applied to some sample studies to verify the robustness of these techniques. Four different case studies have been conducted. These case studies are the applications of work done such as:

Scene perception for visually impaired The object detector model has been utilized for the navigation of blind. An odroid board; running a torch code on a linux mint platform; integrated with an USB camera and USB laser was utilized for the purpose. The software implements a multiscale fusion based object detector with distance information from laser. To reduce utility problems, a user-centered design approach has been acquired in which feedback from various individuals was obtained to understand their problems and requirements. The valuable insights gained from the feedback were then used to

modify the system to best suit the requirements of the user.

Density estimation The model has been generated for mapping traffic patterns on particular routes at different instances. The traffic pattern is treated as an object detector problem whereas particular routes are reffered as scene localizer component. The problem was modeled as a machine learning technique where the traffic distribution at different times (including same days,different days,different weather) were observed continuously using a service robot. Gaussian model has been used for density estimation model generation. Augmented maps were helpful for estimating density in indoor areas.

Cow tracking The object detector and scene localizer modules have been exploited for building tracking system for cows. The mobile robot will navigate autonomously, continuously capturing scenes which are being processed by the object detector. When cows are detected, they are also processed by scene localizer to provide necessary information. According to survey [97], it has been observed that cow tracker system can be very helpful for farmers with the large land where cows can be lost and many farmers are using these kind of devices[98].

Tour guide Scene localizer and maps have been utilized for guiding an unfamiliar person to conveniently navigate from one place to another in particular organization. A software has been developed which takes source and destination as input and give path from source to destination overlayed on a map as output to the user. Even source can be taken as user's location using scene localizer model.

Chapter 6: Conclusions and Future scope Thesis concludes with this chapter by making a brief statement of proposed research work. It give the significance of work done, contributions and also provides an insight into the future directions for working in this area.

Chapter 2

Vision Based Map Generation

Mobile robot mapping and localization has generally been done using two types of maps that are geometric and topological maps. Geometric maps represent the total navigation space in a global coordinate system. On the other hand, topological methods represent the environment as a graph. In the environment, locations are vertices and path between them are edges [99], [100]. Topological maps are used for mobile robot localization tasks. These maps are compact, simple, require less computer memory, and also expedite the robot navigation processes [101]. Hybrid methods have also been presented in [99] [100] due to the reason that they hold the advantages of multiple methods. In this chapter, we discuss a hybrid map generation using trajectory data and further populating it with visual scene features. Work on map generation using various sensors done by number of researchers has been reviewed in section 2.1.1. Further, Section 2.2 describes trajectory map building with vision augmentation including robot description(section 2.2.1), data collection for map generation(section 2.2.2) and point clouds(section 2.2.3) of the scenes of generated maps.

2.1 Navigation

Navigation system is easy in outdoor environment due to Global Positioning System (GPS) which provide exact positions and is generally used for mobile robot localization. However, GPS is not able to provide accurate positioning information in an indoor environment. For indoor environments, the positioning methods that have been proposed are Wi-Fi, ultrasound, Bluetooth, image recognition and inertial navigation [102]; [103]. Ultrasonic signals [104] were used for calculating the position of mobile robot in an indoor environment. The combination of ultrasonic sensors and a laser range finder (LRF) data for mobile robot localization was proposed for a semi-unknown indoor environment [105]. Radio-Frequency Identification (RFID) was used for positioning system [106]. The sensors used in indoor environment have many disadvantages such as (1) Many of these sensors are expensive, (2) their cover range is limited, (3) get affected by other surrounding sensors and (4) give approximate positions. Inspite of all the problems, these

are various reasons that they were popular for quite a long period of time. Table 2.1 shows mostly used technologies for localization in indoor environment along with their advantages and disadvantages. Many researchers preferred using vision based techniques due to the many reasons such as cameras are less disruptive and cheaper to install than automatic vehicle detectors, visual information extracted is used to get high accuracy of positioning even when there is situation of heavy traffic and in case when type of object information is required.

Table 2.1: Indoor Technologies used for localization

Technology	**Common measurement methods**	**Advantage**	**Disadvantage**	**References**
Bluetooth	Proximity, RSS	Present in common devices such as smartphones [107]	The uses must to do a scan The users must do a scan to detect the beacons in the area [107]; cover range is limited [108]	[109][110][111]
Zigbee	RSS, Phase Shift Measurements	Low cost and energy consumption	Low transmission rate, complexity and cost [112].	[113]
WLAN	RSS	The hardware is cost effective and easy to install; LOS is not required [114]	Affected by multipath and fast fading effects [114].	[115][116][117]
UWB	ToA, TDOA	UWB signals have greater penetration of obstacles; it has a desirable direct resolvability of direct multipath components [112].	High financial cost	[118][119][120]
RFID	Proximity, RSS	Suitable for dense environments; does not require LOS [107].	High cost of readers, complex infrastructure	[121][122]
Ultrasonic	ToA, TDOA	Low cost; does not suffer interference from electromagnetic waves [107].	Affected by high frequency sounds; large scale implementation is complex [107]	[104][123]
Infrared	Proximity, Differential Phase-shift, AoA	Absence of radio electromagnetic interference [114].	Expensive system hardware and maintenance cost [114].	[124][125]
Celluar Based Dead Reckoning	RSS, Tracking	Can work at the same frequency as other devices [107], Extra hardware is not required.	Low precision [107], Computes an approximate position [112]	[126][127][128]
Channel State Information (CSI)	frequencies	The Channel Impulse Response (CIR) has a greater granularity than the RSS because it can record the channel's amplitude and phase response at distinct frequencies as well as between distinct transmitter-receiver antenna combinations.	On off - the-shelf NICs, it's not readily accessible.	[129]
Fingerprinting	Features	Simple to use	New fingerprints are needed even if the room varies slightly	[130]
Visible Light	LED emitters	Wide-scale availability, potential to provide high accuracy, multipath free	Comparatively greater power utilization, range is impaired, requiring mainly LoS	[131]

Table 2.1 continued from previous page

Technology	Common measurement methods	Advantage	Disadvantage	References
SigFox	SigFox works in unauthorized ISM radio bands and utilizes proprietary radio Ultra Narrow Band (UNB) and binary-phase-shiftkeying (BPSK) modulation to deliver ultra-low data rate and reliable long-range communication.	Wide reception range, low energy consumption, high reliability	Great distance among the base station and device, severe signal attenuation outdoor to indoor owing to construction walls	[132]
Acoustic Signal	Microphone sensors	High localization precision has been demonstrated by acoustic-based systems,	The transmission energy should be sufficiently small to cause noise pollution	[130]

Table 2.2 shows categories of the vision based schemes inclusive of their advantages and disadvantages.

Table 2.2: Categorical representation of Vision Based Topological Schemes

Categories	Advantages	Disadvantages	Descriptor	References
Global descriptors	Easy to calculate and save storage space ; decrease mapping and localization tasks computational requirements [133]	Low robustness to the impacts of occlusion and lighting, lowering the strength of discrimination [133]	PCA, Omni-gist, DP-FACT	[134][135][136][137][138][139]
Handcrafted Local features	High strength of discrimination ; modifications in scale and occlusion, illumination and rotation are generally more robust [140]	Computer costs and storage requirements are greater than for worldwide descriptors [141]	SIFT, Wavelets, 3D-PIRF, (SURF), Color features	[142][143][144][145]
BoW schemes	Satisfactory to process with a large amount of pictures [141]	there is a presence of loud phrases because of the coarseness of the vocabulary building technique and the loss of the spatial relationships between the phrases [140]	FAST/BRIE, SURF, SIFT, ORB	[146][147][148][149][150]
Deep Features	More accuracy due to deep features, fast processing for large datasets.	Require high end system for computation.	ConvNet, RCNN, Faster RCNN	[151][152][153]
Vision + other sensors	- Give distance along with classification, verify the classification - Gaussian Mixture: Able to handle harsh weather and poorly textured roadways	- Need extra hardware - Require lot of computation	LIDAR, Laser, GPS	[154][155][156][157]

2.1.1 Map Generation Background

A lot of work reported in literature discusses various techniques used for generating maps and navigating using them. Researchers have generated maps in many ways such as

using trajectory data, combination of camera and range sensors and using camera only. Following are some of the evidences of each of the categories:

Trajectory data based techniques: Jo et al.[158] proposed a map generation algorithm for autonomous car. The algorithm for roadway map generation is consisted of three phases for wit, "data acquisition, data processing, and road modelling". In the data acquisition phase, raw trajectory and motion information for map generation were obtained using a probe car, fitted with GPS and on-board sensors, through geographic exploration. The data processing phase then processes the trajectory and movement information obtained into traffic geometry information Optimal smoothing technique and a B-spline road modeling technique that could also produce a precise and reliable road map for riding autonomous cars. The authors provided significant road map criteria for autonomous car applications, namely a requirement for geometry and a requirement for execution. The authors proposed a gradual correction approximation algorithm to use the correct number of control points and knots to approximate the geometry of the road. Kock et al.[159] presented "Digital Road Maps" validation methods in Predictive Control. "Digital road maps with slope, curve and other road data provide a chance to implement model-based predictive control strategy" that can assist save fuel, boost security and convenience, and decrease wear in car operation. The authors presented a technique that finds an appropriate map for predictive control apps with accurate slope data and validates the efficiency of the model. Global Positioning System (GPS) and GLONAS satellites were used to determine road altitude and slope profiles. To determine altitude and slope profiles of roads, Global Positioning System (GPS) and GLONAS satellites were used. The driving experiment was carried out using the same control algorithm & the car with a poor map and a good map with predictive control implementation and the test was also carried out on the same highway with the same weather and traffic conditions. The authors discussed how poor maps impact real performance and how a very excellent map can enhance a predictive control algorithm's performance.

Hybrid techniques using combination of multiple sensors: For generating map using low-cost sensors, Guo et al.[160] worked on "Automatic Lane-level Map Generation" for roadways as well as intersections for Advanced Driver Assistance Systems. Digital maps at the lane level can make driving tasks easy for robotic cars along with improving accuracy and reliability for advanced driver assistance systems (ADAS) by providing strong driving environment priorities. Modules such as the generation of road orthographic images and the construction of lane graphs were discussed. The authors first "divided the global map into fixed local segments based on the road network topology" which was imported from open street map. By fusing GPS, INS, and visual odometry, the bird's eye view images of the road surface were accumulated and then with the ref-

erence of the local map segments, integrated into synthetic orthographic images . Next, the data on the path driven was obtained from the orthographic pictures of the highway and a big number of vehicle trajectories. They modeled the lane centerlines as smooth curves. A number of curve depictions, such as polylines, circular arc splines, clothoid splines, etc., were used for map generation. They used clothoid spline to map ordinary roads in a connection section and used cubic splines to produce "virtual" transition routes for junctions in the node section. Such data was then used to build a map lane graph based on the advanced lane models suggested by authors without manual processing. Guan et al.[161] have discussed the development of automated algorithms for extracting road features from Mobile Laser Scanning (MLS) point cloud data. They researched on Automated Road Information Extraction from vehicle-borne MLS system. They used a laser beam to scan a visible surface and record the beam travel time and reflected energy from the surface to extract their geometry and intensity information in the form of 3-D point clouds. On the street, corner points of objects and white road marks on road surfaces were selected as they were easily identified in point clouds. Based on the surface area of the highway, they created Geo-Referenced Feature (GRF) images and developed curb detection algorithm to extract high retroreflectivity road markings and cracks with low contrast to their surroundings, low signal-to-noise ratio and bad continuity.John et al.[162] proposed the off-line step in which a convolutional neural network has been used to detect and recognize the traffic lights (TL) in the picture using on-board GPS sensor providing region-of-interest data. The detected traffic light information was then used with a modified multi-dimensional density-based spatial clustering of noise applications (M-DBSCAN) algorithm to generate saliency maps. Using vehicle GPS data, the produced saliency maps were indexed. Further, saliency maps were used to estimate the location of traffic light in the image for the map-based real-time TL detection. For developing high-precision road map, Gwon et al.[163] focused on three intelligent vehicle system roadmap criteria: centimeter precision, storage effectiveness, and usability. By obtaining information using a mobile 3D laser scanner, the road data acquisition and processing system recorded precise 3D road geometry information. For a road having lane markings, the Cartesian coordinates of the lane markings were obtained as the road geometry data from the 3D laser scanning data. For a road having no lane markings, such as when going off road, the trajectory of a probe vehicle driving along centerline of a road is used. The data of road geometry was then edited to obtain meta information, and the refined data was depicted as sets of piecewise polynomials in the street modeling scheme to guarantee the map's storage efficiency and usability.

Tan et al.[164] proposed a method for generating the Radio Map (RM) for the particular site. For this, the inertial smartphone measurements were employed to produce trajecto-

ries using the dead-reckoning (PDR) algorithm for pedestrians. PDR trajectories, some landmark points along with the collected fingerprints have been adapted to form a factor graph that is a graphical depiction of a trajectory estimation issue. The trajectories generated inertially could be considered as an original trajectory guess. The trajectories generated inertially could be considered as an original trajectory guess. A raw RM with the estimated positions was created by optimizing the graph. This technique did not require any additional hardware, much additional information such as elaboratedindoor maps or additional survey process. The only additional data was some of the foreknown landmarks locations that were frequently achieved because many apps for indoor positioning need some reference points to start with. Luo et al.[165] integrated metric maps and topological maps together for the navigation of a mobile robot. To record accessible routes and useful metadata, they used topological maps. The robot created a topological node after some constant distance, the node was linked with an edge to the prior node.The robot continued to compare its present view with the images stored in topological nodes during navigation. They used a neural network to compare images and create a similarity value. They proposed an image-based particle filter that could deliver a more flexible estimated robot pose. In addition, hybrid metric-topomap also record the images of the environment, that could give information with more semantic meaning. Gunasekaran et al.[166] proposed system consists of map generation, path planning and path tracking for autonomous mobile robot. Ultrasonic sensor information were gathered by shifting the robot across the surrounding perimeter while the occupancy grid was incrementally updated utilizing range sensors to evaluate the distance from the surrounding edge to the obstacle. This dataset was converted to a map used by Genetic Algorithms (GAs) to generate an ideal path from a predefined beginning and goal point. By pursuing a path defined by way points estimated using GAs, the independent robot system traveled between these points. The pure pursuit (PP) algorithm was used for trajectory following. Sock et al.[167] proposed an approach for the use of 3D-LIDAR and camera to estimate terrain traversability and create a 2D probabilistic grid map online. In many robotic applications, the combination of LIDAR and camera is favored because they provide complementary data. Separate traversability maps were created, each with data recorded from a single sensor. Vision sensor traversability estimation autonomously collectedtraining information and updated classifier without human interference as the vehicle traversed the terrain. Camera-based traversability map was produced by diving the picture into blocks and developed as a binary classification of every other block, assuming that the car is always placed on a traversable flat ground. They used linear SVM classifier for detection using RGB Colour means, Lab Colour means, Entropy of Intensity and Normalized Positions. They have done traversability map with LIDAR by building

2.5D elevation grid map and derive traversability map utilizing difference in height among non-neighboring cells, however separate cells at a certain range. They utilized exponential function to map the slope value to the score for traversability. In order to enhance the detection efficiency, two separately constructed probabilistic maps were combined using the rule of Bayes. They introduced the algorithm on a UGV (Unmanned Ground Vehicle) and tested their strategy to assess the detection efficiency on a rough terrain. They have a more stable algorithm against sensor failure or environmental change. Many researchers defined a topological map and further populated the nodes and vertices using image features which were obtained by using LBP [168] or Bag-Of-Words (BoW) [140]. They defined topological maps as graphs, where nodes are distinctive places of the environment and edges represent topological relationships between them.

Camera based techniques: The task of performing autonomous driving by tracking a preset route created in a manual mapping trip has been focused by Vivacqua et. al [169]. They proposed an accurate low-cost localization approach, combining techniques of visual lane marking detection, dead reckoning, map-matching, and data fusion. For building the map, the vehicle was manually driven by a person along the desired path, and a proper map was built online without any kind of manual adjustment. The map contained the lane markings detected and a reference path that corresponds to the path described by the center of mass of the vehicle. Once the vehicle was precisely localized in the map, a reference path would guide the vehicle through the sections of missing lane markings detected using lane marking detector algorithm optimized for short range operation. The localization system has the function to estimate the vehicles pose in the map reference system using two techniques of absolute pose measurement, one based on global navigation satellite System (GNSS) data and other based on Back Lane Marking Registry (BLMR) data. This information and dead reckoning were fused by a filter to produce continuous and more robust pose estimation. In [170] SURF features are applied to extract the points which are used for both mapping and localization. This approach is described as topometric, because it is a fusion between topological and metric approaches. A topometric map is developed by once riding the path and recording a visual feature database. Then by matching features to this database at runtime, the vehicle gets localized. The localization is made on a topological map, but the map is geo-localized in order to achieve a metric localization. Further 3D features have been used by researchers for creating maps. Lategahn and Stiller [171] extracted 3D features using stereo vision, were used to create a 3D map in which it is possible to self-localize using both the real-time computed landmarks and the ones stored in the map. The map was developed during an offline computation using the vehicle cameras . Each stereo image was filtered using a blob and corner filter during map construction. Non-maximum suppression was

then employed to filter reactions resulting in a set of salient points. Subsequently, each point was defined by a 256 dimensional feature vector consisting of gray values of the main point area. The search space for matching features was spatially restricted and used SIMD instructions to perform matching effectively. Outliers were heuristically removed using a voting system after delauny triangulation. Park et al.[172] constructed 3D global of robot's surroundings using 3D feature maps extracted from SURF algorithm using stereo camera and information extracted from LiDAR for localization. Even topological maps are being generated for navigation purposes. Xu et al. [173] utilized the stereo camera and 3D point cloud map for presenting an algorithm for vehicle localization. They generated maps using Mobile Mapping System (MMS). The 3D PCL map included geometry information (latitude, longitude and altitude) and intensity information. They transformed the latitude, longitude and altitude to the real-world coordinate system. Further obtained 6 DOF transformation from real-world coordinate system to the camera coordinate. They adopted particle filter based framework for estimating vehicle position. To reconstruct the vehicle's pose, they synthesize the intensity images and virtual depth from PCL map and match them with stereo vision depth and the ouput of live camera. They defined a framework to match the information of the live camera and the offline map. Further by lyrio et al. [174], an image based localization scheme was presented applying Virtual Generalizing Random Access Memory (VG-RAM) [175]. A neural map is built from 3D landmarks, detected by a stereo vision system, and used for localization. Firstly, VG-RAM Image-Based Mapping (VIBM) received images of the environment, captured by the robots stereo camera, along with the robots global poses where the images were captured. Subsequently, it identified characteristic points on the acquired pictures and used the range data acquired from depth maps to calculate their 3D positions (3D landmarks). A stereo matching algorithm calculated these depth maps. Finally, using VG-RAM architecture, VIBM learns images, associated global poses and positions of landmarks. Recently, 3D model of the scene has been generated using photogrammetric software [59]. This provides a scene's sparse 3D model as well as positions and orientations of all images. Image matching activities were performed to achieve a dense 3D point cloud which were further used for navigation system. They used single-view geometry approach which is based on the Space Resection Algorithm to estimate the position of the camera.

2.2 Trajectory Map Building with vision Augmentation

For the proposed research work, Robot has been used for data collection. The details of robot are described below:

2.2.1 Robot Description

WiRobot is an embedded robotic system of electronics and software expanded from the extensive humanoid robot of Dr. Robot set up with a Multimedia Controller (PMB5010), a Sensing Motion Controller (PMS5005) and multiple electronic peripheral modules. The software component will be mounted on a PC and is accountable for connecting and exchanging information with the robot via wireless connection. All the details about robot given below have been taken from Figure2.2.

2.2.1.1 Mechanics

The X80 has two 12V DC engines , of which ,each supply 300 oz from the wheel-based platform. -Inches of torque on the 18 cm (7 in.) wheels of the X80 with a maximum velocity of more than 1 m / s (3.3 ft / s). Two high-resolution quadrature encoders (1200 counts per wheel cycle) installed on each wheel provide high-precision wheel motion measurement and control. The weight of the scheme is only 3.5 kg (7.7 lb.), but it can carry an extra 10 kg (22 lb.) payload. With its WiFi 802.11 wireless module embedded high bandwidth (11Mbps), the system can upload all sensor information (including encoder sensor measurements) to a PC or server at prices above 10Hz. Similarly, it is a snap to stream audio (8Hzx 8bits) and video (up to 14 fps) for either direct surveillance or high-level Al systems. Commands and instructions sent to the X80 also pass at rates exceeding 10Hz via the same wireless connection, offering real-time control and access.

2.2.1.2 Sensors

Figure2.3 shows the sensor details of robot. X80 provides complete WiFi (802.11b) wireless, multimedia, sensing and movement capabilities and is equipped with a broad range of sensor, camera and audio modules that can be used suitably, Of these camera, IR and Ultrasonic sensors are used in most cases. The human proximity sensors, temperature sensors have been used in certain cases. The embedded camera head can pan and tilt

separately, powered by distinct RC servo motors and is easily used to capture images for panoramic views. A Hokuyo URG-04LX-UG01 scanning laser is used to get the distance of the detected object from the user. Laser scanner is able to report ranges from $20mm$ to $5600mm$ in a 240 degree arc with 0.36 degree angular resolution. Its power consumption, 5V, allows it to be used on battery operated platforms.

Calibration: Camera and laser calibration has been done for the proposed work. The extrinsic calibration of laser range finder and camera has been done using the traditional calibration method which has been used in Ref. 176 also. The world coordinates of image have been projected to image coordinates using extrinsic parameters(E), orientation(O) and position(d) of camera. Further the transformation from camera coordinate system to laser coordinate system using laser's orientation(ϕ) and position(δ) has been done. The world coordinates $P = [x, y, z]^T$ can be projected to the image coordinates $\rho = [\upsilon, \nu]^T$ as follows:

$$\rho \sim E(OP + d) \tag{2.1}$$

where E is the camera intrinsic matrix, O a $3X3$ orthonormal matrix representing the cameras orientation, and d a 3-vector representing its position. The camera can show important lens distortion in actual instances, which can be modelled as a 5-vector parameter composed of coefficients of radial and tangent distortion. At the laser range finder, a laser coordinate system is defined with an origin, and the laser scanning plane is $y = 0$. Presume a point P in the camera coordinate scheme is located at a P^f point in the laser coordinate scheme, and the conversion from the camera coordinate scheme to laser coordinate scheme can be defined as :

$$P^f = \phi P + \delta \tag{2.2}$$

where Φ is a $3X3$ orthonormal matrix representing the cameras orientation relative to the laser ranger finder and is Δ a 3-vector corresponding to its relative position.

In the proposed work, calibration has been done by using the experimental setup where small objects such as bottle are used to obtain 3D and 2D coordinates [177]. Both the captured scenes from camera and laser sensor, are divided into grids for the mapping purpose. One grid of camera correspond to one grid of laser considering common centre point. Some of the instances of the process of getting data is shown in Figure 2.1. In camera coordinate system, the calibration plane can be parameterized by 3-vector $\aleph$ which can be calculated as in equation 2.3.

$$\aleph = -O_3({O_3}^T.d) \tag{2.3}$$

where O_3 is the third column of matrix O and d the center of the camera, in world coordinates. Since the laser points must be on the camera's estimated calibration plane, we get a geometric constraint on the rigid transformation between the system of camera coordinates and the system of laser coordinates. Given a laser point P^f in the laser coordinate system, from equation 2.2, we can determine its coordinate P in the camera reference frame as $P = \phi^{-1}(P^f - \delta)$. Since the point P is on the calibration plane defined by $\aleph$, it satisfies that $\aleph.P = \| \aleph \|^2$. Then we have

$$\aleph.\phi^{-1}(P^f - \delta) = \| \aleph \|^2 \tag{2.4}$$

For a measured calibration plane parameters N and laser point P^f , this gives a constraint on ϕ and δ. For solving the extrinsic calibration problem for the system of a camera and laser range finder, assuming that all the laser points are on the plane $y = 0$. So the laser point can be represented as $\hat{P^f} = [x, z, 1]^T$ The calibration plane parameter ($\aleph$) shown in equation 2.5 can be obtained by using equation 2.3.

$$\aleph.\Re P^f = \| \aleph \|^2 \tag{2.5}$$

where $\Re$ is a $3X3$ transform matrix from the laser coordinate system to the camera coordinate system. For each pose of the calibration plane. We have several linear equations in the unknown parameters of $\Re$, which we solve with standard linear least squares. After knowing H; ϕ and δ can be calculated using the equation 2.6.

$$\begin{aligned} \phi &= [\Re_1, -\Re_1 \times \Re_2, \Re_2]^T \\ \delta &= -[\Re_1, -\Re_1 \times \Re_2, \Re_2]^T \Re_3 \end{aligned} \tag{2.6}$$

where $\Re_i$ is the i^{th} column of matrix $\Re$. Equation 2.4 gives the Euclidean distance between a laser point and the calibration plane.

The equation 2.7 shows the calculation for the whole data set that is for different positions of bottle.

$$\Sigma_i \Sigma_j (\frac{\aleph_i}{\| \aleph_i \|}.(\phi^{-1}(P_{ij}^f - \delta)) - \| \aleph_i \|)^2 \tag{2.7}$$

where $\aleph_i$ is the bottle plane in the i^{th} position. The whole process has been further summarized below:

1. Different objects (bottles in this case) were taken and placed them in front of the camera-laser range finder system in the different orientations.

2. For each position of bottle, extracted the laser points in the laser reading, and

detected the bottle grid points in the image. Estimated the camera orientation O_i and position d_i with respect to the bottle, and then computed the calibration plane parameter $\aleph_i$.

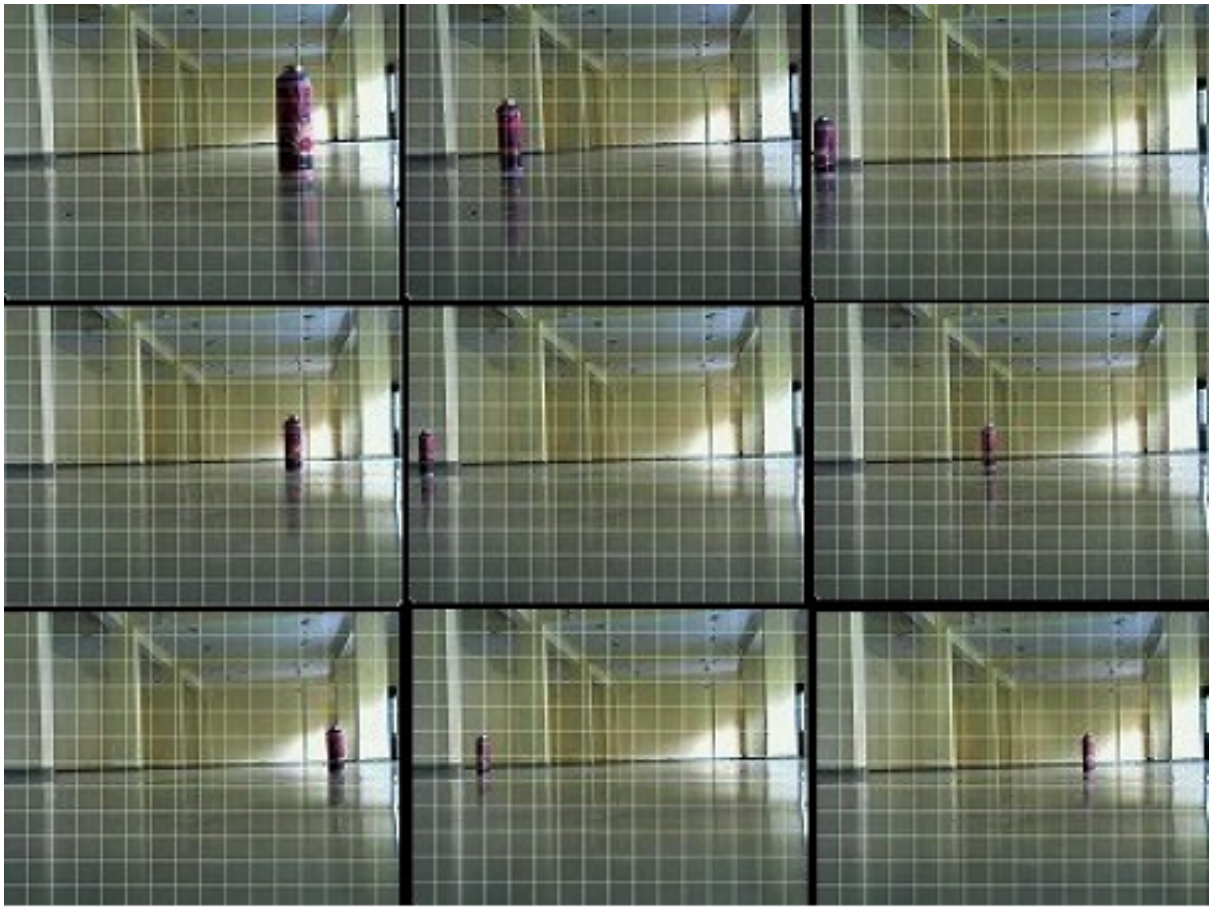
3. Estimated the parameter ϕ and δ using the equation 2.5.

4. Refined ϕ and δ using the equation 2.7.

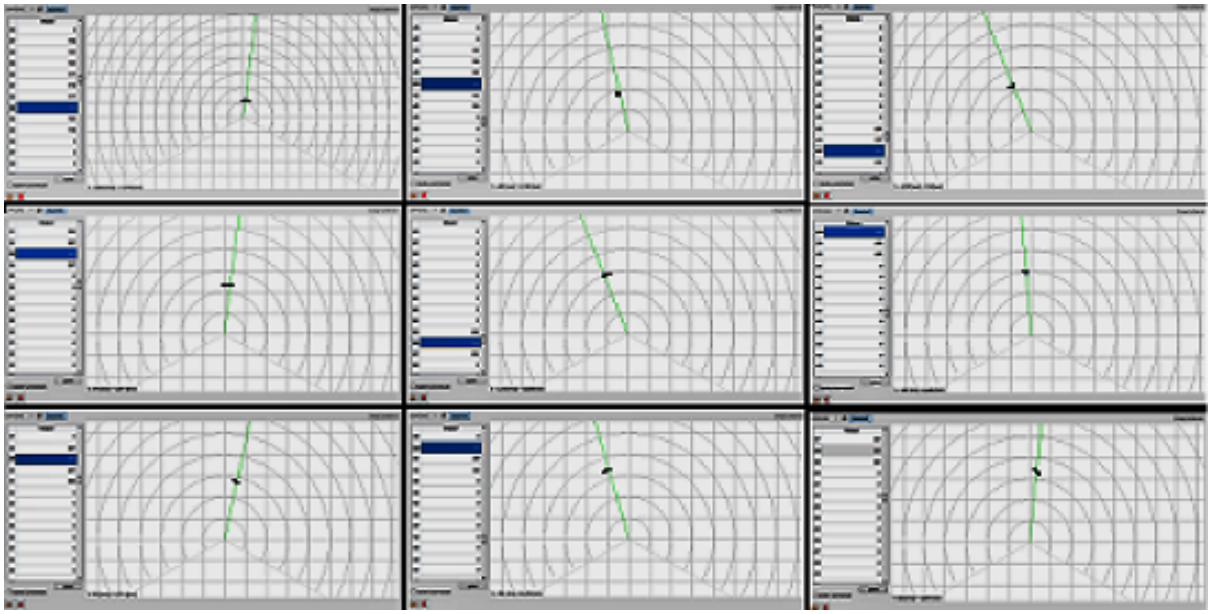
The laser gives the data in the form of polar value ($\wp$), distance(γ) and angle(α) of the object from the center point of laser. From the polar value, Cartesian coordinates of the object have been calculated. A mapping from camera to laser coordinates has been accomplished by using a function represented in equation 2.8 from which the distance of the object has been figured out.

$$f : A_i \rightarrow B_i \tag{2.8}$$

where A represents Data of camera and B represents laser data. Equation 2.8 has been solved using inverse mapping of equation 2.1 and 2.2



(a) Instances of images(captured via camera) in grid style



(b) Instances of images(generated via laser) in grid style

Figure 2.1: Mapping of laser data with images capture by camera

2.2.1.3 Mechanical and Control Highlights

- Two 12V motors with over 300oz.-inch torque each
- 7 inch driving wheel
- Max speed of 1 m/sec
- Dimensions:
  - 38.0 cm (15.0 inch) diameter
  - 25.5 cm (10.0 inch) height
- Weight: 3.5 kg
- Large top mounting deck for additional devices such as a notebook computer
- Additional carrying payload: 10 kg
- Pre-programmed fine speed and position control achieved by an integrated PMS5005 module employing two 1200 count per wheel -cycle quadrature encoders.

2.2.1.4 Electronic System Highlights

- Fully integrated WiFi (802.11b) system with dual serial communication channels (max of 912.6 Kbps per channel), supporting both UDP and TCP/IP protocol.
- Full color video and two-way audio capability. (CMOS color image module and audio module are fully integrated.)
- Battery: 3700mAh with over 3 hours for nominal operation.
- Collision detection sensors include 3 sonar range sensors and 7 IR range sensors
- Two pyroelectric sensors for human motion detection
- Additional sensors such as supplementary sonar sensors, temperature sensors, acceleration / tilting sensor, or customized sensors can be added.

2.2.2 Data Collection

Data for map generation of university campus has been collected using WiRobot X80. Robot has been navigated in the campus using the path in the google map as shown in Figure 2.3a. Continuous navigation instructions are provided to the robot via an user

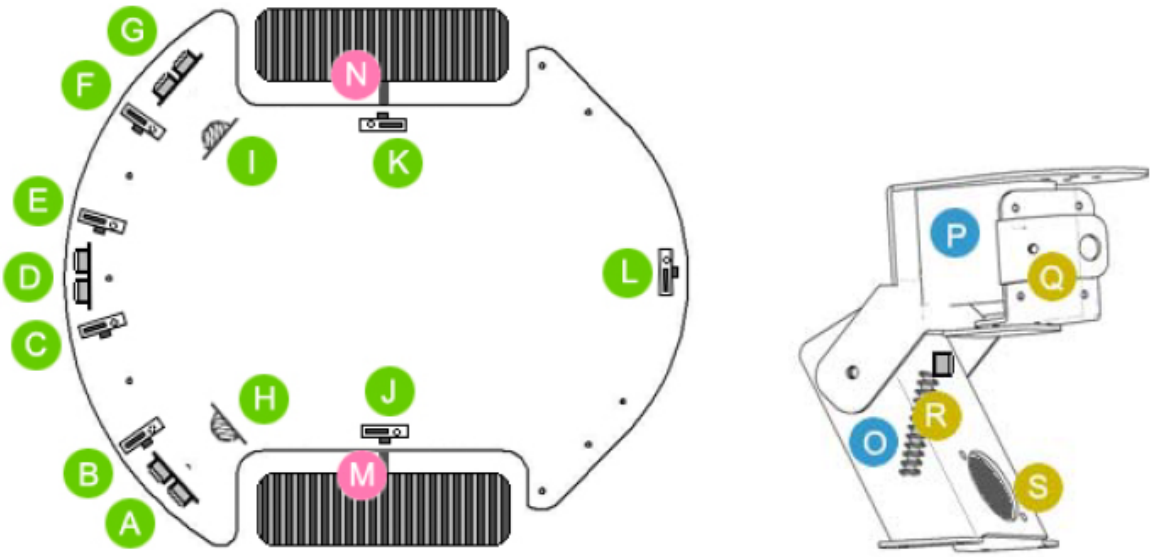


Figure 2.2: Structure of Robot

Electrical Module	X80 Location! Setting
Ultrasonic #1	A) Left front
Ultrasonic #2	D) Middle front
Ultrasonic #3	G) Right front
Human Sensor #1	H) Left front, upper lever
Human Sensor #2	I) Right front, upper lever
Infrared Range Sensor #1	B) Front
Infrared Range Sensor #2	C) Front
Infrared Range Sensor #3	E) Front
Infrared Range Sensor #4	F) Front
Infrared Range Sensor #5	K) Right side
Infrared Range Sensor #6	L) Rear
Infrared Range Sensor #7	J) Left side
Servo #1	P) To control the left/right movement of the neck (use channel1)
Servo #2	O) To control the up/down movement of the neck (use channel2)
DC Motor #1 with quadrature encoder	M) Left, use channel 1
DC Motor #2 with quadrature encoder	N) Right, use channel 2
Camera	Q) Middle front
Speaker	S) Middle front, under the camera
Microphone	R) Beside the speaker

Table 2.3: Table shows location of sensors on robot

(a) Map extracted from Google map (b) Path generated on selected area

Figure 2.3: Map considered for path generation

interface for moving along this path. During the movement, the robot avoids obstacle using its ultrasonic and IR sensors. The user interface is also used to add nodes at desired points during the navigation process. The trajectory data of the robot is used to add the distances between the nodes. The process has been repeated for the path shown in Figure 2.3b. The steps for collection of data has been represented in Algorithm 2.1. Once the trajectory map with node information is built, the robot can be instructed to navigate using the map. At this time the Robot also continuously captures images of both sides using the pan-tilt camera and records the data. The data set includes images of various effects like sunny, cloudy, evening time etc.

Algorithm 2.1 Algorithm for map building

1. Start the robot.
   Repeat Steps 2-4 till end point:
2. While navigating, add nodes N_i at specified locations.
3. Save the distance from previous node where $d = N_i \rightarrow N_{i+1}$.
4. Save the directions to move from one node to another.

Output: Trajectory map is built.

Table 2.4 presents the distance and directions to follow while moving from every node to their adjacent nodes. In this table, GS stands for Go Straight and E, W, N, S have been abbreviated for East, West, North, South. As an example while navigating from N1 to N5, the commands can be followed as shown in equation 2.9 in which RT represents Right Turn whereas LT stands for Left Turn. The nodes are shown in Figure 2.3b.

$$N1 \xrightarrow{GSW(270)} N2 \xrightarrow{RTGS(240)} N7 \xrightarrow{LTGS(450)} N6 \xrightarrow{GS(500)} N5 \tag{2.9}$$

The whole area has been divided into 16 zones as depicted in Figure 2.4. The images captured by robot have been separated into different folders of zones(Z1 to Z16) while navigating the robot. They are further reviewed and images at the zone boundaries are labelled using human annotation. Few of the images of zones have been demonstrated in

Start Node	Commands	Distance in meters	Adjacent Nodes
N1	GSW	270	N2
N2	GSE, GSN, GSW	270, 240, 400	N1, N7, N3
N3	GSE, GSW, GSN	400, 400, 200	N2, N4, N6
N4	GSE	400	N3
N5	GSE	500	N6
N6	GSS, GSW, GSE	200, 500, 450	N3, N5, N7
N7	GSS, GSW, GSE	240, 450, 450	N2, N6, N8
N8	GSS, TLGS, TRGS	450	N7

Table 2.4: Commands for navigating from a node to its adjacent node

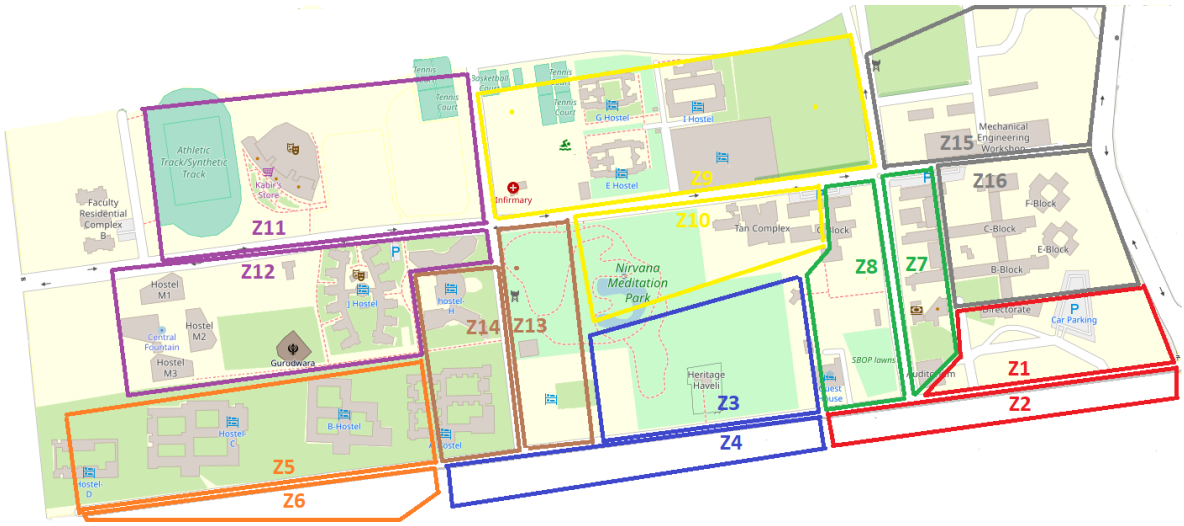


Figure 2.4: Map with Zones using notations Z1 to Z16

Figure 2.5.

These images are used in two different ways: a) Generating deep features for scene localization and augmenting the trajectory maps with scene information, this is discussed in the next chapter. b) The collected images are also used to create zone wise scene panoramas and point clouds. In this work point clouds of scenes have been generated to provide virtual tours as demonstrated in section 2.2.3.

2.2.3 Point Cloud

The Graphical User Interface(GUI) has been prepared to provide a virtual tour demonstration. At the launch of GUI, it shows the basic map of university campus taken from openstreetmap [178]. It has various buttons, giving variety of information such that path taken into consideration for map augmentation, Zones information and other information related to zones. The user can click on any location of the map. This click is mapped to

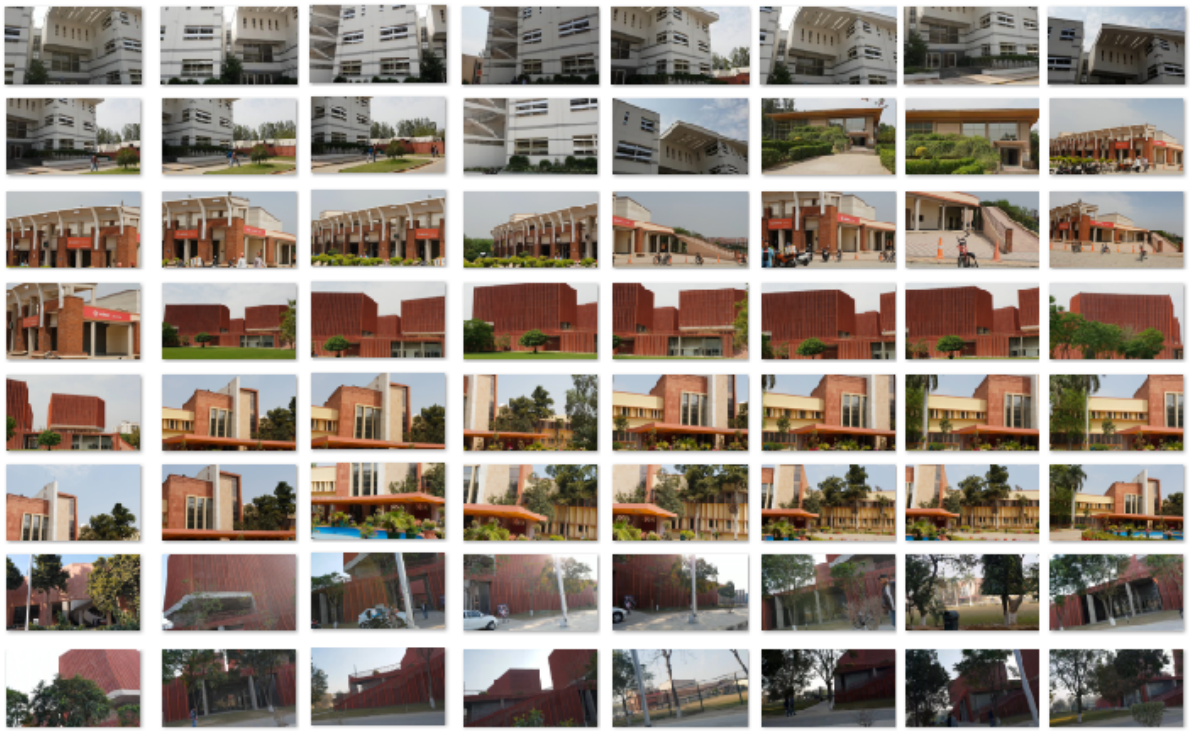


Figure 2.5: Instances of Zones

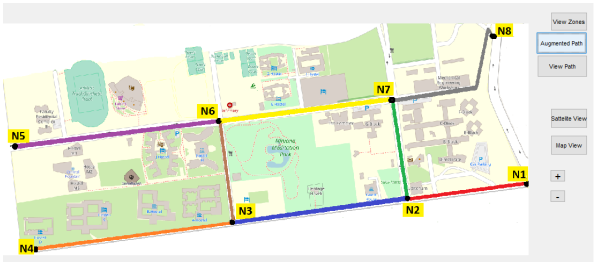
the respective zone and the corresponding scene is showed. Moreover, different zoomed views of the selected scene can be viewed based on the users clicks. Views of GUI has been presented in Figure 2.6 in which map view(2.6a), satellite view(2.6b), Zones on map(2.6c), path selected on map(2.6d), path used for the task(2.6e), view of building when clicked on map(2.6f), left(2.6g)-right(2.6h) views of selected building. The sub-image views from the scene panorama are reconstructed from the point cloud as depicted in figures 2.7c and 2.8c.

Point cloud is defined as the feature representation of a scene or any kind of area. Point clouds produced by 3D scanners and 3D imaging are visualized for the ease of measurement. Point cloud based map generation have been used for localization and reconstruction. Multiple point clouds of urban environment were generated by Hanmoudi et al. [179] for generating 3D maps. Various researchers used point cloud maps for different purposes such as for estimating vehicle position [180], for generating 3D map of kitchen environment[181], for indoor scenes[182] and for shape detection[183]. There are many devices available for generating point cloud maps of the surrounding which are useful for map building. Devices that has camera and depth estimator are the most popular for generating point clouds such as Kinect. In the proposed work, point cloud has been generated using feature extraction, feature matching technique and dense reconstruction using multiple images of one scene. A point cloud is a set of data points in a 3D coordi-

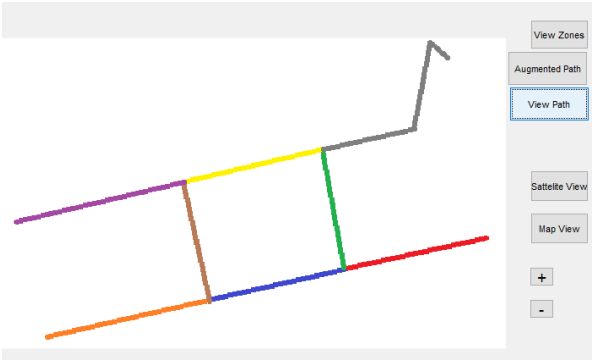
(a) Map View

(b) Satellite View

(c) Zones on map

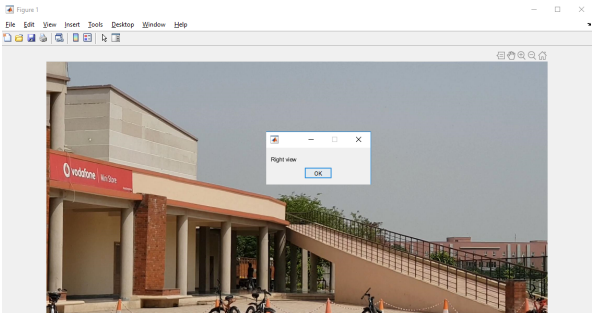


(d) Path selected on map

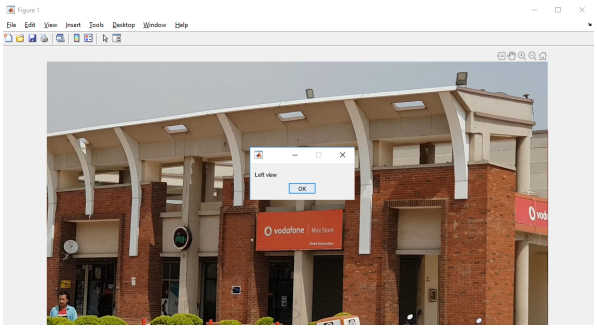


(e) Path used for the task

(f) View of building when clicked on map



(g) Left View



(h) Right View

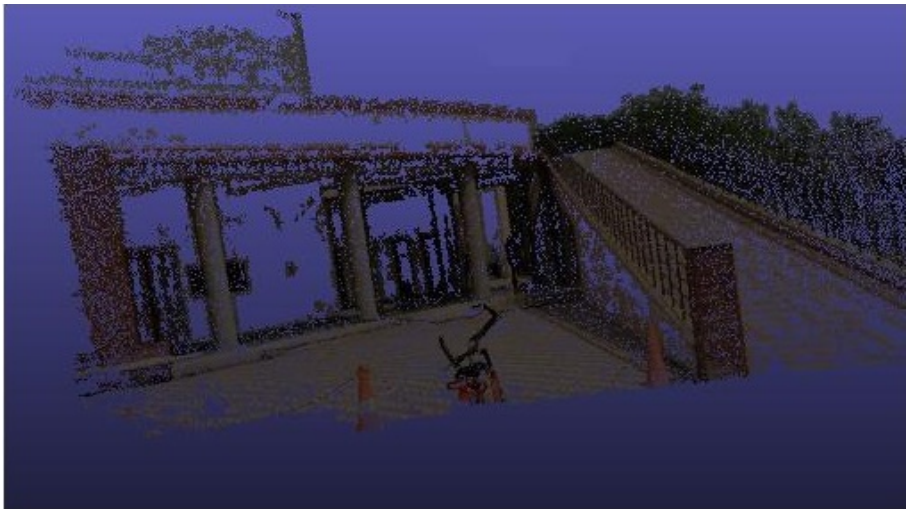
Figure 2.6: Various views of GUI

nate system, normaly defined by x, y, and z coordinates. They are used to represent and object's surface and contain no data of any internal features, color, materials etc. For generating point cloud, the camera and range sensors of robot scanned the surrounding environment. The scan was converted to a point cloud where a tuple containing the 3D position in world coordinates (x,y,z), intensity and distance values (i,d) were depicted at each point. Point clouds are good starting point for 3D modeling and can be useful when placing 3D items in a scene. Two examples have been shown in Figure 2.7 in which the whole scene has been captured in panoramic view via robot as depicted in Figure 2.7a and 2.8a. The 3D features have been calculated and point cloud maps have been generated from the scenes as shown in Figures 2.7b and 2.8b to have 3D view of the scenes. Further, Figure 2.7c and 2.8c contain the subset images of the scenes. The point clouds of the scenes captured during map generation have been referred further for scene localization explained in next chapter.

2.3 Conclusion

This chapter discusses the trajectory based map generation procedure. It gives a brief description about the robot and its sensors used for the purpose, including calibration steps. The data from the university campus has been collected using this robot. The whole area has been divided into zones. For populating the topological maps with visual features, scene data has been collected from the different zones(Z1 to Z16) in the generated map. Finally, point cloud maps of captured scenes have been generated so that it can be used for scene reconstruction for virtual touring.

(a) Scene1

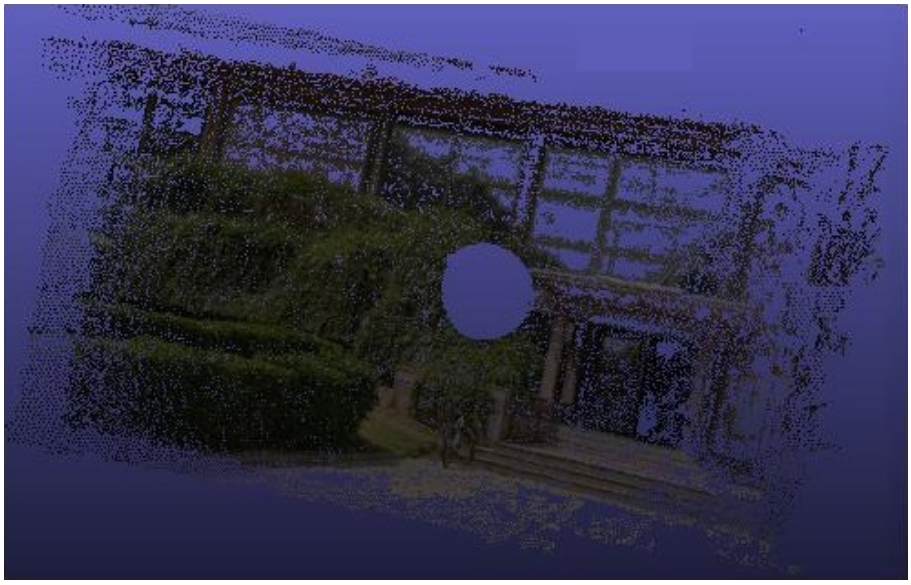


(b) Point cloud from scene1

(c) Subset images from scene1

Figure 2.7: Example1: Point cloud and subset of panorama of the scene of zone captured by robot

(a) Scene2



(b) Point cloud from scene2

(c) Subset images from scene2

Figure 2.8: Example2: Point cloud and subset of panorama of another scene of different zone captured by robot

Chapter 3

Scene Localization

Scene recognition and robot localization plays an important role in computer vision based research in robotics community. In this case, the system first has to learn the places very carefully and remember them for later recognition just like a human brain does. This is very helpful for robot localization for autonomous navigation in long routes. In this chapter, we discuss scene visual feature extraction for populating the topological map generated in the last chapter such that it can be used for scene localization. A 2-step approach has been used for the purpose allowing a zone level localization as well as a landmark detection. Zone detection using CNN features with set-based image classification has been implemented in section 3.2.1. Further landmark detection has been implemented by training a capsule network in section 3.2.2. Results have been provided in section 3.3. During implementing the different steps, we specifically looked at particular issues concerning use of deep networks for real-time tasks for domain specific data. Details regarding network compression for faster inference time, soft target training for better transfer learning with smaller data size and GAN based multiple dustbin classes to improve verification in case of outliers have been discussed.

3.1 Background

For global localization, it is required to know the robot position and orientation. Forechi et al. [184] proposed a 2 step solution for this. Firstly, a WNN is used to solve place recognition as a classification problem which returns the most similar image of live image and its pose. Secondly a CNN solves the issue of visual localization by comparing the image taken by live camera with the recollected image to obtain a 6D relative pose. A Convolutional Neural Network (CNN) was trained with the two images as input which is giving output of a relative position vector; to estimate the relative camera position between the current and the recollected images. Both the approaches solved the issue of global localization by utilizing the topological and metric data to approximate the current position of the vehicle. Robots will require training for recognizing landmarks using features so that while autonomous navigation, it can localize itself. Many researchers

have performed scene recognition and applied the same for robot localization. Ishikoori et al. [185] presented a visual landmark based semantic scene recognition method and described visual landmark features using AKAZE on saliency map after excluding human regions by Histogram of Gradient. For the position recognition method, they used self-organization maps to create codebooks as visual words and counter propagation networks to map features into a low dimension space as a category map based on neighborhood and competitive learning. In [186], 3D point cloud was reduced via sampling the depth image points and grouping them into points sets either belonging to 3D planes or those that did not correspond to planes within specified error margin. Then a localized algorithm that down projected plane filtered points on to a 2D and assigned correspondences for each point was used to enable obstacle avoidance. Both the plane filtered, as well as the outlier point clouds utilized depth information and not RGB data. A place recognition algorithm based on 3D-laser was proposed [187] to accomplish loop closure detection for SLAM. In this case, only 3D laser was used. 3D laser points were converted to 2D images using Bearing Angle (BA) image model. Further, scene matching was performed using ORB(oriented fast and rotated brief) features, extracted from BA images. However finding a query BA image from the set of BA images was too costly in terms of computation. Hence, to improve search efficiency in real-time place recognition, a visual Bag of Words (BoW) approach was used. Furthermore, a 3D-geometry-based verification algorithm and a speed normalization algorithm were proposed to complete the proposed place recognition algorithm. Xu et al. [188] used a hybrid map based localization method which consisted of two steps. First a monocular vision based rough global localization method and second laser range finder precise localization method. Combining the above methods gave better localization results. Some other handcrafted feature extraction techniques reported for scene classification include:

- Antonio et al. [189] explored and demonstrated that with huge data set, even simple nonparametric methods such as simple nearest neighbour can perform well in object recognition. They showed that images having enough information inspite of very low-resolution can be used for scene recognition, segmentation and object detection.

- SIFT features were used for scene recognition by van et al. [190]. Their work adopted an approach for invariance to light intensity, color and shifts. These included histograms in various color moments, color spaces and moment invariants and color extensions of SIFT. The SIFT points were obtained using Harris-Laplace point detector. Further SVM classifier was used and kernel function was used for distance between features.

- Xiao et al. [191] introduced the idea of Scene Detection recognizing the scene type inside image regions instead of whole images. They used SUN database and extracted features using SSIM (Self-similarity descriptors), GIST, SIFT, HOG and LBP features.

There are significant scene classification results using state of the art CNN features. Some interesting observations are reported below. Sunderhauf et al. [192] presented place categorization trained network which when confronted with severe appearance changes provides a stable performance. They further clarify which networks and layers are ideal for which aspects (appearance and viewpoint) of the place recognition problem. They extracted features from various layers of alexnet and compared the performances. The experiments showed that the middle layer features were more robust in terms of appearance changes than any other layer's features. The performance of both the higher and lower layers in the feature hierarchy got inadequate robustness and presented bad results for place recognition. 2 order magnitude speed up was obtained using the cosine distance approximation between features over the bit vectors found by Locality Sensitive Hashing, compressing the features data. Herranz et al. considered two problems in [193], viz biasing of dataset in patch-based CNNs with various scaling variants, and productively combining ImageNet with Places got the fact that they are related. They depicted that the accuracy of recognition was dependant on the multi-scale combinations of places and imagenet. So ImageNet-CNN as well as Places-CNN were implicitly tuned for different scale ranges (scene and object scales). Four variations were compared: original masked, original with background, canonical masked and canonical with background each with the combination of ImageNet and Places network features. While entire scene classification was considered by researchers, some work also focused on identifying scene based on a particular landmark in it. In [194], researchers used a method to extract candidate landmarks from the images. ConvNet features were extracted from each landmark proposals which were of huge size and could create difficulty in matching process. So dimensionality reduction was applied to make it more efficient. They showed that the ConvNet features are very reliable in terms of appearance and viewpoint change. They also emphasized that landmark proposals require no training, their system is training-free in that there was no requirement of task or site-specific training. They also highlighted that the system did not need sequences of images but only single images were required for matching. Bolei et al. [195] worked on scene classification and recognition using image database with deep scene understanding. They fine-tuned pretrained networks Alexnet, vgg16 and resnet with places database. "Places-CNNs perform much better than the ImageNet feature+SVM baseline" while, Places205- GoogLeNet and Places205-VGG outperformed Places205-AlexNet with a great margin. The reason might be their deeper structures.

For AlexNet and VGG, they used the feature vector of 4096-dimensional from the fc7 layer. For GoogLeNet, they used the feature vector of 1024-dimensional from the global average pooling layer before softmax producing the output of the class. The input was given as scene and question was asked that: Is this an art school? The result was given in the form of yes or no. Bolei et al. [196] Proposed a new method for comparing the density and variety of image datasets and showing that Places is as dense and more diverse as other scene datasets. They learned deep features for identification of sceneusing CNN and established the latest state-of-the-art outcomes on multiple scene-centered datasets. They used ImageNet-CNN already trained to extract features. They used the mean image method to visualize the units of the higher layers.

3.2 2-step Localization

A 2-step approach has been used for scene localization, motivated from the idea of [184]. For step 1, deep features of zones mentioned in section 2.2.2 from last layer of places CNN have been extracted. State of the art CNN algorithms for this purpose have been reviewed in [185–196]. The 2nd step in the proposed work is an extra refinement of the positioning using landmark detection via Capsule networks. In the 1st step, after extracting features using CNN, set based difference technique was used for scene matching. The results have been shown in Table 3.1. To further pin the identified scene with an associated landmark a capsule network based landmark detection is used as a 2nd Step using particular buildings for it. Here we used the class data along with dustbin classes to remove the biasness of result in case of outliers. We generated dustbin classes using GAN for greater confidence of network for known classes and lower confidence for unknown classes.

3.2.1 Zone Detection (places CNN+set based)

We begin by extracting zone wise features from last fully connected layer of places365 CNN and proceed with a Set based Classification. For a Set based image classification a set of test images is passed to the algorithm and it gives the result in the form of difference of the test set from each set of training image classes. The set with the minimum difference represent the class of test image set. In this work, the features of each zone images comprise of a set. These techniques have been used extensively by researchers for face and object recognition tasks. Locally Grassmannian Discriminant Analysis has applied for face recognition using set of images by Xu et al.[197]. They continued to use cooperative exemplary representation as convex hull. The sample set is depicted

collaboratively across all the samples from separate gallery sets. For the probe set, the remaining error wasestimated and each gallery was set after the representation coefficient has been resolved. The gallery set with the minimum mistake was considered to be the consequence of the classification. Wang et al. [198] Proposed a novel approach for image-set classification i.e. Covariance Discriminative Learning (CDL) by modeling the image with its covariance matrix. They derived a kernel function that links the covariance matrix to an Euclidean space from the Riemannian manifold. They used two methods that are Partial Least Squares (PLS) and Linear Discriminant Analysis (LDA). Huang et al. [199] introduced multiple statistics for set modeling followed by embedding them into multiple heterogeneous spaces which includes one Euclidean space and two different Riemannian manifolds. Also they proposed the Hybrid Euclidean-and-Riemannian Metric Learning (HERML) for fusing such statistics lying in heterogeneous spaces. In this work, the methods used for finding set based differences are described below. Let $S = [S_1, S_2, ..., S_N]$ be the training set having N feature sets containing cl zones where $S_i = [s_1, s_2, ..., s_n]$ represents i^{th} feature set and n is the number of samples in each set.

COV+LDA "The covariance matrix is mapped from the Riemannian manifold to an Euclidean space via a kernel function using Log-Euclidean Distance (LED)". Any learning technique dedicated to vector space can be utilized in its linear or kernel formulation with this explicit mapping. In this, LDA is considered and in next point PLS is considered. For this, the feature set is represented using its covariance matrix from equation 3.1 [198].

$$C = \frac{1}{n-1}\sum_{i=1}^{n}(\mu_i - \overline{\mu})(\mu_i - \overline{\mu})^T \tag{3.1}$$

where $\overline{\mu}$ is the mean of the feature samples. The kernel function effectively maps the points from the manifold to a Euclidean space through equation 3.2

$$\Psi_{log} : M \mapsto T_I, C = log(C) \tag{3.2}$$

By computing the inner product in the Euclidean space T_I, we obtain a Riemannian kernel function on the manifold M using equation 3.3

$$\kappa_{log(C_1,C_2)} = tr[log(C_1.C_2)] \tag{3.3}$$

The training samples and Riemannian kernel were then fed to kernal variant of

LDA using equation 3.4.

$$\alpha_o pt = argmax \frac{\alpha^T KWK\alpha}{\alpha^T KK\alpha} \tag{3.4}$$

where $\alpha = [a_1, ..., a_n]^T$, K is the kernel Gram matrix:$K_{ij} = k(s_i, s_j)$, and W is defined as: $W_{ij} = \frac{1}{number of samples in k^{th} class}$,if s_i, s_j are both in the k^{th} class; otherwise $W_{ij} = 0$.

Classification was done using equation 3.5

$$\tau = A^T K_t \tag{3.5}$$

where $K_t = [k(s_1, s_t), ..., (k(s_n, s_t)]^T$ and $A = [\alpha_1, ..., \alpha_{cl}]$.

COV+PLS "PLS creates score/latent vectors by utilizing the existing correlations among various sets of variables and also keeping best of the variance of both sets". Taking Riemannian kernel, kernal variant PLS is was used to learn the regression model using equation 3.6 [198]

$$\beta_\phi = \Phi^T U(T^T KU)^{-1} T^T Y \tag{3.6}$$

Equation 3.7 was used for testing phase.

$$\gamma_t^T = [\phi(S_t)]^T B_\phi = K_t^T U(T^T KU)^{-1} T^T Y \tag{3.7}$$

where t and u are the column vectors of T and U respectively

HERML In Hybrid Euclidean-and-Riemannian Metric Learning method, after finding mean, covariance; the data distribution set can be modelled as Single Gaussian Model(SGM). The distances between training pairs $d_{B_z^t}(K_{.i}^z, K_{.j}^z)$ have been computed where $z = 1, ..., Z$. Calculated α using equation 3.8 and set $\alpha \leftarrow min(\alpha, \eta_{ij})$ and $\eta_{ij} \leftarrow \eta_{ij} - \alpha$ [199].

$$\frac{\delta_{ij}}{Z} \sum_{z=1}^{Z} \frac{d_{B_z^t}(K_{.i}^z, K_{.j}^z)}{1 - \delta_{ij}\alpha d_{B_z^t}(K_{.i}^z, K_{.j}^z)} - \frac{\gamma \xi_{ij}^t}{\gamma + \delta_{ij}\alpha\xi_{ij}^t} = 0. \tag{3.8}$$

where $d_{B_z}(K_{.i}^z, K_{.j}^z)$ indicates the distance between the i_{th} and j_{th} samples under the learned metric B_z for the z_{th} statistic mapping in the Hilbert space as shown in equation 3.9.

$$d_{B_z}(K_{.i}^z, K_{.j}^z) = tr(B_z(K_{.i}^z - K_{.j}^z)(K_{.i}^z - K_{.j}^z)^T) \tag{3.9}$$

Update B_z^{t+1} using equation 3.10 for $z = 1, ..., Z$.

$$B_z^{t+1} = B_z^t + \beta_z B_z (K_{.i}^z - K_{.j}^z)(K_{.i}^z - K_{.j}^z)^T B_z \tag{3.10}$$

ξ_{ij}^{t+1} is updated using equation 3.11 until convergence. The output is in the form of Mahalanobis matrices $B_1, ..., B_Z$

$$\xi_{ij}^{t+1} = \frac{\gamma \xi_{ij}^t}{\gamma + \delta_{ij} \alpha \xi_{ij}^t} \tag{3.11}$$

These algorithms have been used for obtaining differences between the set of images and further used for zone detection. The features of self created places data set from two networks that are Alexnet and VGG16 pre-trained on places data set, have been extracted. These networks are pre-trained on places data-set. Each zone with feature vector extracted from VGG16 is of size $500X4096$ and from Alexnet is of size $500X4096$. Further these differences were normalized using min-max and fused(added) motivated from [200] to improve the recognition rate using following formula shown in equation 3.12

$$D = D *_{AP} + D *_{AL} + D *_{AH} + D *_{VP} + D *_{VL} + D *_{VH} \tag{3.12}$$

where D_{AP}, D_{AL} and D_{AH} are the distances obtained from training and testing feature sets in which features were extracted from Alexnet pre-trained with places dataset and distances were obtained using $COV + PLS, COV + LDA$ and $HERML$ respectively. D_{VP}, D_{VL} and D_{VH} are the distances obtained from training and testing feature sets in which features were extracted from VGG16 pre-trained with places dataset and distances were obtained using $COV + PLS, COV + LDA$ and $HERML$ respectively. $D*_{AP}, D*_{AL}, D*_{AH}, D*_{VP}, D*_{VL}$ and $D*_{VH}$ are the normalized version of D_{AP}, D_{AL}, D_{AH}, D_{VP}, D_{VL} and D_{VH} respectively using min-max technique. Figure 3.1 shows the methodology used in this work for image set classification.

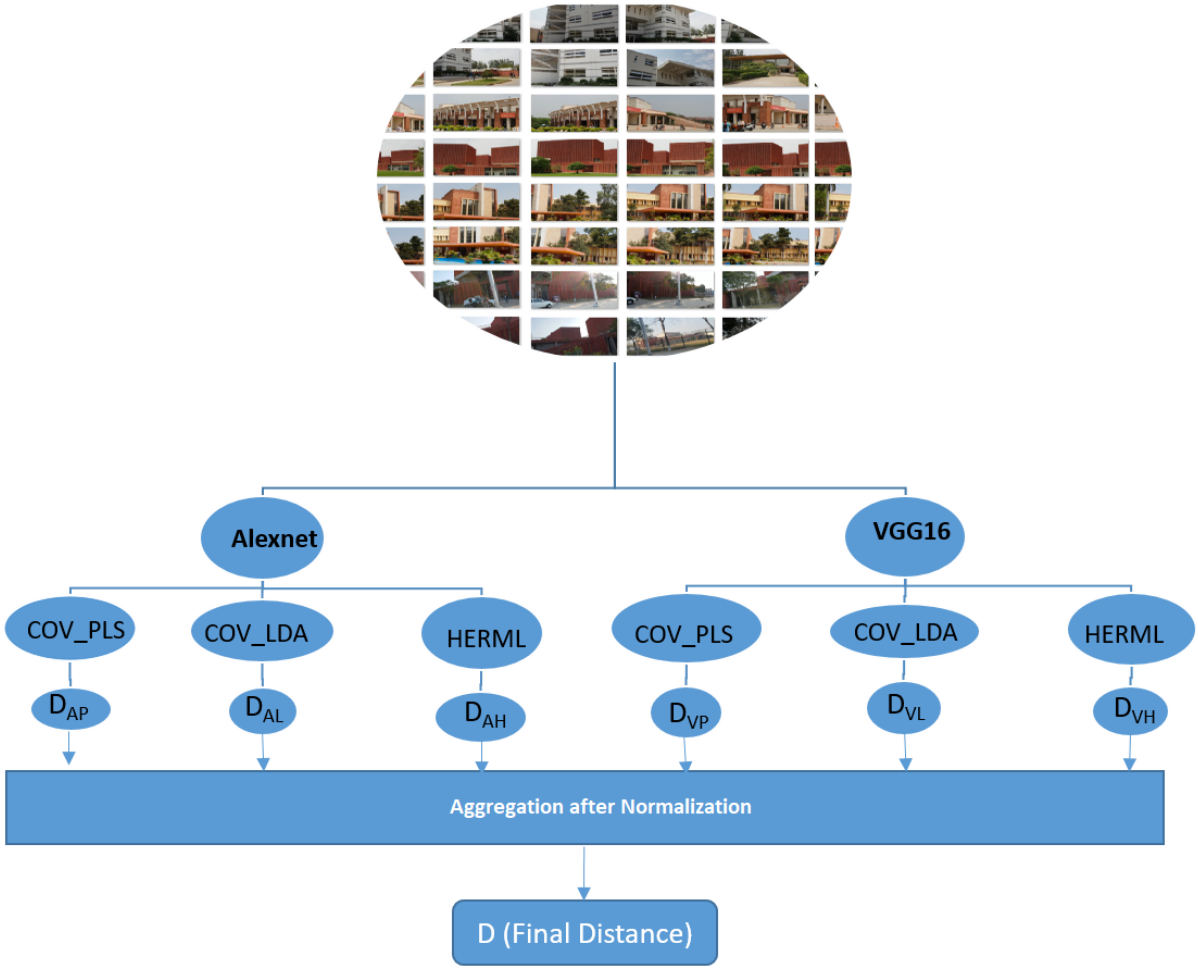


Figure 3.1: Methodology used for image set classification

3.2.2 Land Mark Detection

In this particular section we focused on obtaining a second level landmark recognition by using distinct landmarks in each zone. This was done by training a capsule network. Although very limited amount of work is reported using capsule network based classifications, we were motivated by some results in [201] [202]. The steps followed for landmark recognition are:

- **We trained a compressed network for landmark detection using (i) weight projection (ii) soft targets. A network (named NoC) is created by taking input from the region pool features of VGG16 and adding 1 convolutional and 3 fully connected layers. We initially fine tuned the NoC by freezing the lower layers. We next compressed the entire network and fine tuned the reduced NOC network by adding an extra building class.**

- **A 2 step approach is used for training capsule network model for fine-grained classification. First, a capsule network is trained for 5 building classes. Following this a second capsule network is trained using classes which result in more number of false positives. Here we use the class data along with dustbin classes to remove the biasness of result in case of outliers.**
- **We generated dustbin classes using GAN for greater confidence of network for known classes and lower confidence for unknown classes.**

3.2.2.1 Compressed soft target based detection model

For scene localization, we primarily detected buildings. These are then classified into specific landmarks. Our primary objective is hence to detect buildings. Available object detection networks e.g. YOLO, faster RCNN do not have a building class. We can start by training a detection network for buildings/not buildings. However, this can result in overfitting due to limited data and may not give reliable results. Hence we added another class to the pretrained available detection network for better classification.

Accuracies for detection networks are improved in many ways. A very basic approach includes training different models with the same data and taking an average of their predictions. This obviously involves huge resource engagement as training large models is cumbersome. Also inference time using these models are equally high. Not to mention the memory requirement and inability to use these in embedded system. As a solution to this a lot of small networks either retrained from scratch or compressed version of the original, further fine-tuned to improve the performance is available [203]. However there is a significant loss in performance. A runtime compression approach introduced in [204] suggests using the same number of parameters but computing them in a resource intensive manner. A weight matrix W_{MXN} with M previous nodes and N next layer nodes is decomposed as shown in equation 3.13.

$$W_{MXN} = U_{MXM} \sum_{NXN} V^T_{NXN} \tag{3.13}$$

This is further approximated by keeping say C values leading to equation 3.14.

$$\overline{W_{MXN}} = U_{MXC} \sum_{CXC} V^T_{CXN} = U_{MXC} N^T_{CXN} \tag{3.14}$$

Thus MXN number of weight parameters drop down to $(M + N)C$. We utilized lower

layers of VGG and made modification in the higher layers. The reason behind choosing VGG16 was that it is very deep network with 41 layers. Hence, it is better from many other nets in terms that it has $3X3$ sized kernel filters one after the other. With a given effective area size of input image on which output depends, couple of smaller sized kernel is better than large sized kernel because more than one non-linear layers result the deep network which makes it possible to learn more complex features at a lower cost and gives better accuracy. The fact whether deeper networks provide better results is however a much broader exploration aspect and has been investigated individually for domain specific applications [205].

We created a new network with features extracted from region pooling layer of VGG16 as input and 1 convolutional layer(512, 512, 3, 1, 1) and 3 fully connected layers(25088-4096), (4096-4096) and (4096-7) are added, referred as NoC shown in Figure 3.2 and fine tuned with shared weights of VGG16. Only NoC is fine tuned. We then reduced the parameters of entire fine tuned network. We analysed two different reduction sizes that are reducing the parameters size of the original network by 1.6 and 1.2 times as shown in Table 3.2 without altering the layer size and named them as R1 and R2 respectively. We conclude that using reduced network not only took less inference time but also do not make much difference in accuracy. So we proceed further with reduced network R2.

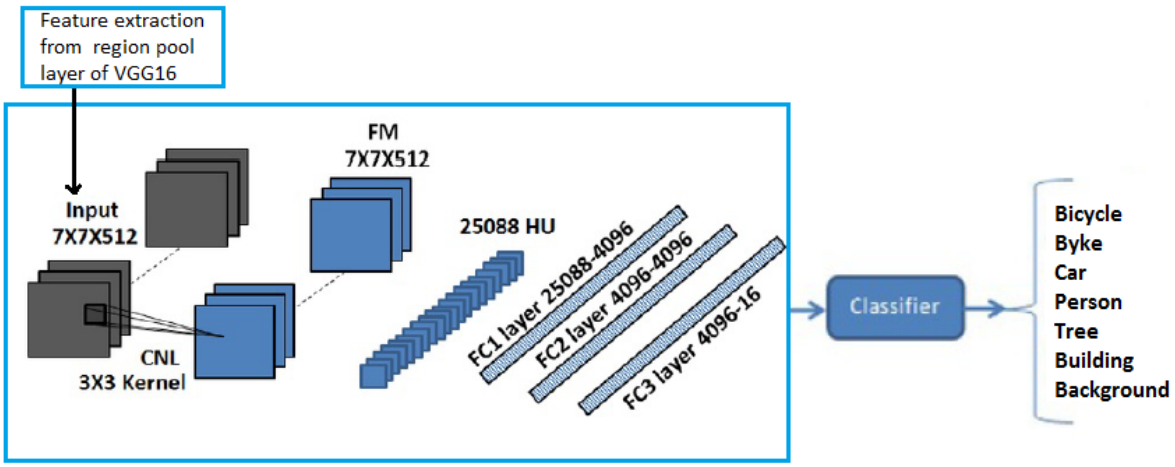


Figure 3.2: Procedure of NoC trained for object detection and classification

To reuse the learning of the old network, fine-tuning may lead to catastrophic forgetting. Also in our case the network do not see the building class at all. To deal with this issue we used the concept of soft targets. The learning of the original network is transferred to this compressed one by using the soft targets. When the soft targets have high entropy they provide much lower gradients hence the smaller model can be trained with lower amount

of data. We trained our small networks using weight updation as given in equation 3.15.

$$W_t = W_{t-1} - \alpha(\beta_1(\frac{1}{T}(\frac{e^{\frac{Z_i}{T}}}{\sum_j e^{\frac{Z_j}{T}}} - \frac{1}{2}(\frac{e^{\frac{v_i^1}{T}}}{\sum_j e^{\frac{v_i^1}{T}}} + \frac{e^{\frac{v_i^2}{T}}}{\sum_j e^{\frac{v_i^2}{T}}}))) + \beta_2(\frac{1}{T}(\frac{e^{\frac{Z_i}{T}}}{\sum e^{\frac{Z_j}{T}}} - 1))) \quad (3.15)$$

α is learning rate, T is the temperature, Z_i is the current prediction, $v^{\frac{1}{i}}$ and $v^{\frac{2}{i}}$ are the soft targets of the original and compressed network. β_1 and β_2 are the weight of the hard and soft targets. It is seen that the hard targets require a lower weight [206] for better performance. The detected building regions are further passed to a Capsule network for building classification so that we can localize the scene.

3.2.2.2 Basic Capsule Architecture

Building a special CNN with fine-grained categories of a particular class is a difficult task. First training this without over fitting is a crucial problem to solve. Second managing intraclass variations require careful selection of suitable loss functions which are capable of handling intraclass variations. [206] proposes a solution to use a 2 step classification where the 1st step is classification results using a large generalized net. In the 2nd step the output is checked with all the specialized nets which mainly consist of classes confused together. We use capsule architecture [207] for implementing specialized classes. We do not have a fixed measure of the contributions of multiple fixations over single fixation for relevant scene understanding. However we assume that in a multilayered classification system, the capsules (group of neurons) in each layer will choose its parent in the higher layers to match parts to whole. As in the case of overlapping digits, the problem of building architecture classification is equally; perhaps more critical. It seems that application of dynamic routing by agreement will prove much more effective compared to max-pooling for segmenting overlapping regions.
Capsules are a vector specifying the features of the object and its likelihood. These features can be any of the instantiation parameters like position, size and orientation, deformation, velocity, albedo (light reflection), hue, texture, etc. The activity vector of a capsule represents the instantiation parameters for a part or whole of an object. The length of the vector represents the probability of object existence. A squashing function is used to ensure that short vectors shrink to zero length while long vectors retains a

value slightly less than 1.

$$v_j = \frac{\| S_j \|^2}{1+ \| S_j \|^2} \frac{S_j}{\| S_j \|} \tag{3.16}$$

v_j in equation 3.16 is capsule output and S_j shown in equation 3.17 is input.

$$S_j = \Sigma_i c_{ij} \widehat{u}, \widehat{u} = w_{ij} u_i \tag{3.17}$$

u_i gives the output vector for each capsule in the layer below connected to $'j'$,c_{ij} and b_{ij} are presented in equation 3.18.

$$\begin{aligned} c_{ij} &= \frac{exp(b_{ij})}{\Sigma_k exp(b_{ik})} \\ b_{ij} &= b_{ij} + \widehat{u}.v_j \end{aligned} \tag{3.18}$$

The loss function is computed as given in equation 3.19

$$L = T_k max(0, m^+ - \| v_k \|)^2 + \lambda(1 - T_k) max(0, \| v_k \| - m^-)^2 \tag{3.19}$$

Where T_k is 1 or soft target of the class sample.
The architecture of capsule model is depicted in Figure 3.3. The input to the network are images with resolution $48X48$. It has 3 convolutional layer of 32, 48 and 64 filters,1 primary capsule layer and 1 classification capsule layer as listed below;

- First layer is conv layer with 32 3*3 filters, stride 1
- 48 3X3 filters stride 1
- 64 3X3 filters stride 2
- 32 3X3 filters stride 2 8D
- 16D capsules for each class.

The main properties of Capsule Network are as follows:

- It is more robust to changes in the orientation and size of the input
- It needs much less data which is often hard to get.
- It can identify new, unseen variations of the class without ever being trained on them.

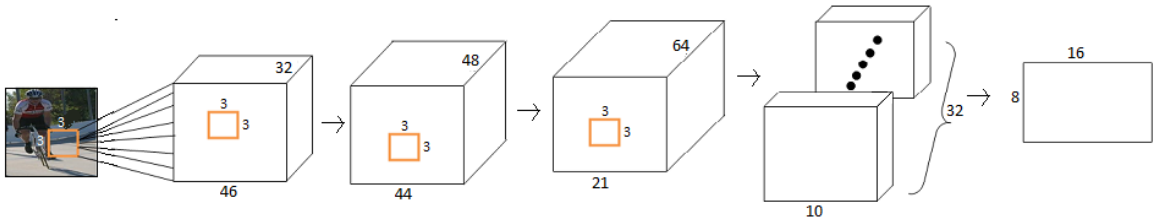


Figure 3.3: Capsule Model Architecture

3.3 Experimental Results

Following experiments have been performed for zone and landmark detection:

3.3.1 EXP: Zone Detection

For experiments done in this work, videos of different zones were captured. Further frames were extracted from the collected videos. Approximately, 20,000 frames were extracted for each zone. Out of them, 500 frames (per zone) were randomly selected for training purpose. The experimental results shown in table 3.1 are obtained using 100 frames (per zone). The data for each zone is annotated automatically while capturing them via manual node addition in the robot as mentioned in section 2.2.2. The data for each zone is captured under various environmental condition like sunny, cloudy, evening time, occlusion etc. The features of self created places data set from two networks that are Alexnet and VGG16 pre-trained on places data set, have been extracted. These networks are pre-trained on places data-set having 1.8 million images from 365 scene categories. Each zone with feature vector extracted from VGG16 is of size $500X4096$ and from Alexnet is of size $500X4096$. These features were further passed to three methods $COV + LDA, COV + PLS$[198] and $HERML$[199]. Further these differences were normalized using min-max and fused(added) motivated from [200] to improve the recognition rate using following formula shown in equation 3.12 The results of with and without fusion of differences have been shown in Table 3.1 which depicts that using fusion method, the results have been improved.

Table 3.1: Recognition Rate of self created places data set

Network	Methods	Top1	Top2	Top4
Alexnet[195]	**COV+LDA**[198]	0.6234	0.6925	0.7815
	COV+PLS[198]	0.6501	0.7000	0.8125
	HERML[199]	0.4512	0.5250	0.7000
VGG16[208]	**COV+LDA**[198]	0.6725	0.7435	0.8225
	COV+PLS[198]	0.7015	0.7500	0.8655
	HERML[199]	0.5135	0.5715	0.7515
Proposed		**0.8515**	**0.8925**	**0.9884**

3.3.2 EXP: Landmark Detection

For scene classification, we currently utilized 5 different scenes each having approximate 10000 images. In our scenario, we have 5 landmarks named audi,cos,hostel,main and tan as shown in Figure 3.4.

Figure 3.4: Building Dataset Samples

EXP:Compression of the network(NoC)

As discussed earlier, we created NoC network (added 1 convolutional and 3 fully connected layers) whose input is the features extracted from region pool layer of VGG16. We fine tuned the NoC using RPNs of PASCAL VOC 2007 dataset. The dataset being very large(1200000) is divided into 3 parts. We tried training the network using different learning rates and different training options(average of 3 networks trained on 3 parts of dataset individually). The best network is chosen for the detection and classification purpose on the basis of loss. Minimum loss value obtained is 0.0017. The loss graph is shown in Figure 3.5. Details can be referred from [209].

We reduced parameters of fine tuned NoC as shown in Table 3.2. We use different ways to present the original vs compressed performances. Firstly we extracted features from 36th linear layer of the original, R1 and R2 networks individually. Using these features:

- We trained and then testing SVM, further calculated the accuracy of original, R1 and R2 networks as shown in Table 3.3

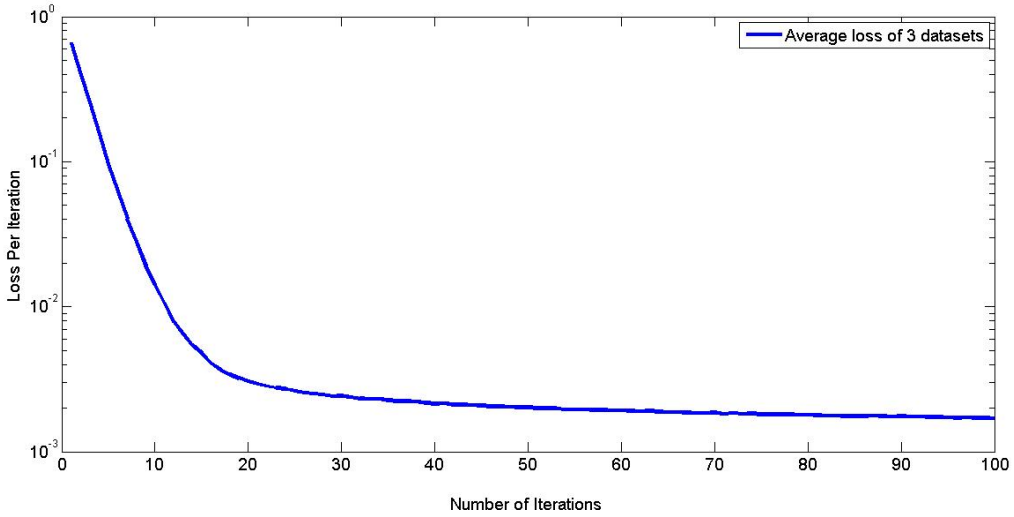


Figure 3.5: Training Loss of NoC

- Mean square error between features of original and two reduced networks using 20 sets each having 900 images approximately is calculated as shown in Table 3.4.
- tsne graphs with different distance options are plotted as shown in Figures 3.6, 3.7 and 3.8. Random classes are selected for showing tsne graphs. We used only 6 classes for better clarity of the diagram. In all these graphs 16 is background class. It can be seen that an acceptable performance was obtained even with reduced networks as tsne show prominent features clusters

Table 3.2: Table Showing original and reduced weights

Layers	Original	Total	R1	Total	R2	Total
3rd Conv	64x64	4096	64x30+30x64	3840	64x25+25x64	3200
5th Conv	64x128	8192	64x30+30x128	5760	64x25+25x128	4800
8th Conv	128x128	16384	128x60+60x128	15360	128x55+55x128	14080
11th Conv	128x256	32768	128x60+60x256	23040	128x55+55x256	21120
13th Conv	256x256	55696	256x100+100x256	51200	256x90+90x256	46080
15th Conv	256x256	55696	256x100+100x256	51200	256x90+90x256	46080
18th Conv	256x512	131072	256x128+128x512	98304	256x110+110x512	84480
20th Conv	512x512	262144	512x230+230x512	235520	512x200+200x512	204800
22nd Conv	512x512	262144	512x230+230x512	235520	512x200+200x512	204800
25th Conv	512x512	262144	512x230+230x512	235520	512x200+200x512	204800
27th Conv	512x512	262144	512x230+230x512	235520	512x200+200x512	204800
29th Conv	512x512	262144	512x230+230x512	235520	512x200+200x512	204800
33rd Linear	25088x4096	102760448	25088x3000+3000x4096	87552000	25088x2000+2000x4096	58368000
36th Linear	4096x4096	16777216	4096x2000+2000x4096	16384000	4096x1500+1500x4096	12288000
39th Linear	4096x7	28672	4096x6+6x7	24618	4096x4+4x7	16412

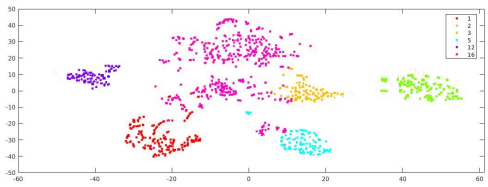
Further, We fine-tuned the compressed network using images of buildings(50000) captured from real time environment. Byke, Bicycle, Person, car, tree and background data(each class having 50000 images approx) is taken from PASCAL VOC dataset. Figure 3.9 depicts the results obtained when query images are given to the fine tuned compressed network.

Table 3.3: Top 1 recognition rate of original and reduced weight networks

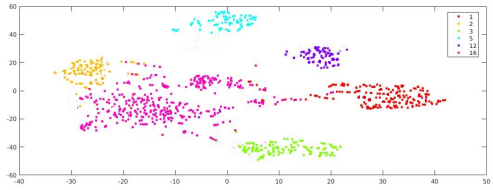
Networks	Accuracy(%)
Original	75
R1	72
R2	70

Table 3.4: Mean square error between original and 2 reduced networks

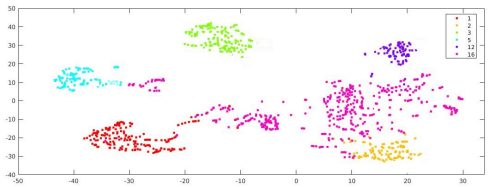
Set No. each having 900 images approx.	Original-R1	Original-R2
Set 1	0.127672	0.242223
Set 2	0.129059	0.250102
Set 3	0.12596	0.245335
Set 4	0.128723	0.247677
Set 5	0.124173	0.243574
Set 6	0.129403	0.244215
Set 7	0.123881	0.251462
Set 8	0.127262	0.233262
Set 9	0.12968	0.244878
Set 10	0.122108	0.243051
Set 11	0.125097	0.23933
Set 12	0.126167	0.242962
Set 13	0.12646	0.236947
Set 14	0.124044	0.246558
Set 15	0.132908	0.235323
Set 16	0.126561	0.2479
Set 17	0.125089	0.245463
Set 18	0.132261	0.240963
Set 19	0.130622	0.245204
Set 20	0.127854	0.241175



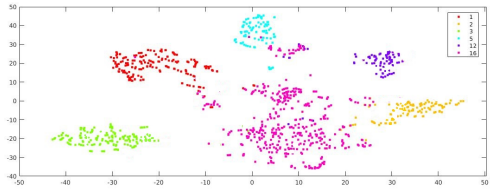
(a) Euclidean



(b) Chebychev

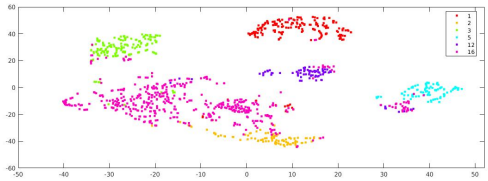


(c) Cosine

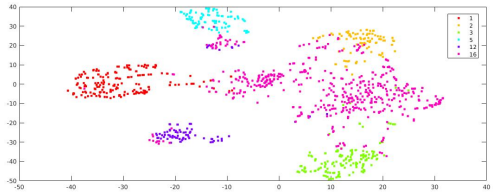


(d) Minkowski

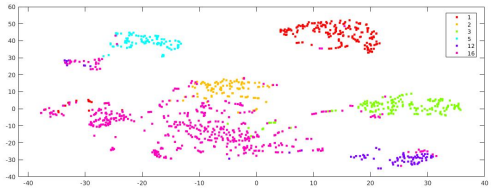
Figure 3.6: tsne plots of dataset from original model using features of 6 classes. Clusters were formed using Euclidean, Chebychev, Cosine and Minkowski distances.



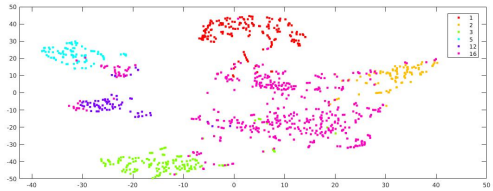
(a) Euclidean



(b) Chebychev

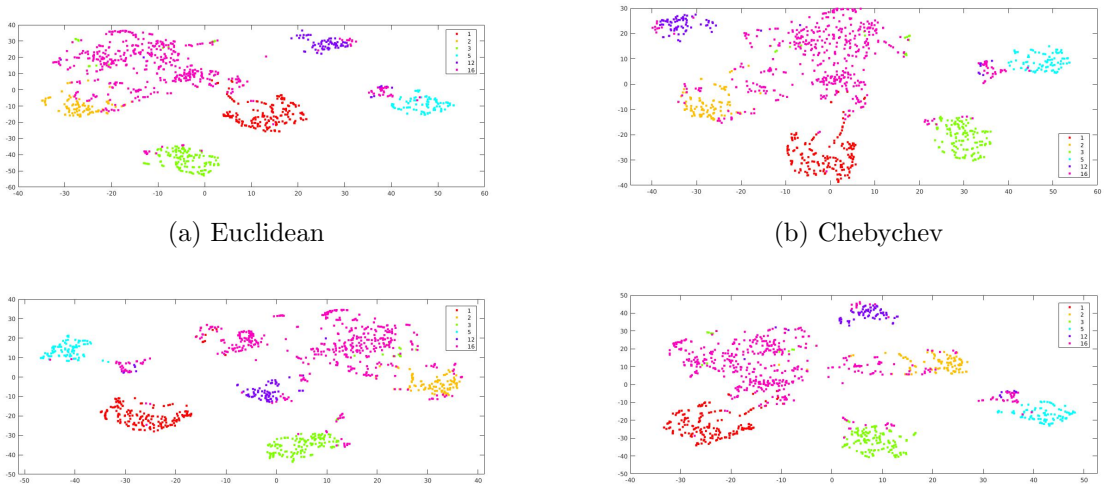


(c) Cosine



(d) Minkowski

Figure 3.7: tsne plots of dataset from R1 model using features of 6 classes. Clusters with R1 is almost as distinct as obtained using original network in Figure3.6



(a) Euclidean

(b) Chebychev

(c) Cosine

(d) Minkowski

Figure 3.8: tsne plots of dataset from R2 model using features of 6 classes. Clusters are still distinct but seem slightly less than that of R1.

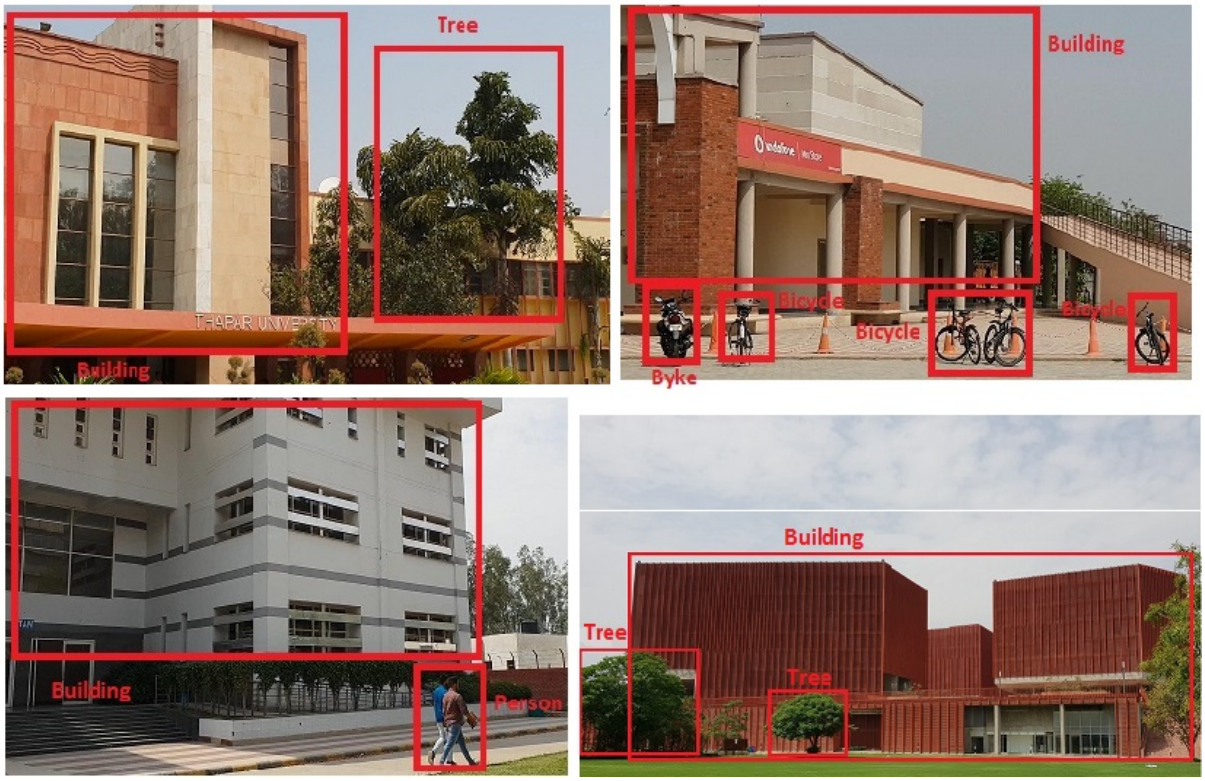


Figure 3.9: Detection result using fine-tuned NoC

EXP: Capsule Network

We trained capsule network for building classification. Three types of networks were experimented: (i) One is mentioned in Section 3.2.2.2 and written as (48-64-32 8D-5 16D), referred as C1. Its input image size is 48x48. It has 3 convolutional layer of 32,

Table 3.5: Results of SVM and Capsule networks in the form of accuracy

Networks	Accuracy(%)
SVM	96
C1	99
C2	99
C3	99

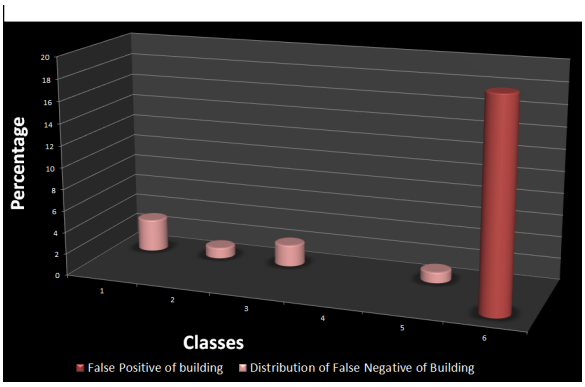
48 and 64 filters,1 primary capsule layer and 1 classification capsule layer (ii) (64-128-32 8D - 5 16D) referred as C2 having 2 convolutional layers with 48 and 128 filters, 1 primary capsule layer and 1 classification capsule layer. Its input image size is 227x227. (iii)(128-32 8D - 5 16D) referred as C3 has 1 convolutional layer with 64 filters, 1 primary capsule layer and 1 classification capsule layer. Its input image size is 28x28. The results presented in Table 3.5 shows that all the networks have almost equal accuracy. It is seen that input image size or number of convolutional layers effected the training time(more layers, greater size, more training time but did not have much changes on the accuracy or converged loss). For performance comparison, we also used SVM classifier. We first extracted features from the last layers of fine tuned network. Then we trained SVM classifier with these features of approximately 20000 samples. We include 10 classes that are: Byke, Bicycle, Person, car, tree, audi, cos, hostel, main and tan. We tested trained SVM with around 1000 samples.

EXP: Network Confidence using GAN

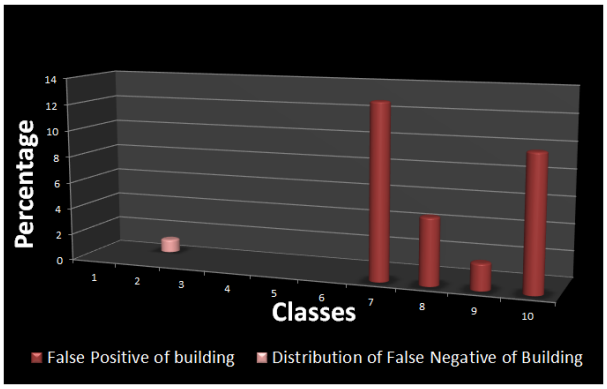
Table 3.5 shows that we obtained a high accuracy for both SVM and for capsule networks. However we are now more concerned about false positives,errors and confidence level. We compared the false positives and false negatives of buildings in 3 cases as shown in Figure 3.10 :

- When we have only one building class (Figure 3.10a).
- Having different types of building classes (Figure 3.10b).
- When the buildings are not amongst the classes, we call them as outliers (Figure 3.10c).

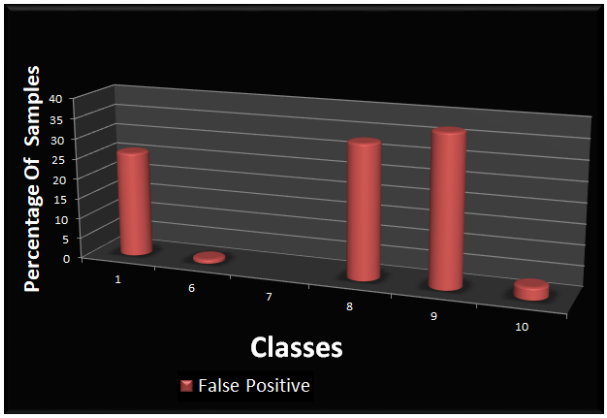
We trained capsule network on approximately 50000 dataset of buildings of different categories. After training capsule network for five classes of building, we obtained 99 percent top 1 recognition accuracy. We named this model as $O5$ that is original capsule network with 5 classes. However, using the trained capsule model, we evaluated two outliers testsets, we obtained the results biased to 2 classes (3 and 4 in this case) as seen in Figure 3.11. Considering this issue, we generated these dustbin classes using GAN



(a) Results of SVM trained with 6 classes, 5 other classes and 1 building class when tested with mix(contain all the 6 classes) testset



(b) Results of SVM trained on 10 classes includes 5 other classes and 5 different buildings classes when tested with mix testset



(c) Results of different buildings that are not amongst the classes i.e. outliers of building classes

Figure 3.10: Class-wise False Positive and False Negative Rate of different SVMs. Results (Figure 3.10c) shows that when tested with outlier building classes, all false positive concentrate on 2 specific building classes

[210]. Dustbin classes of 3 and 4 are generated using images class 3 and 4 respectively in GAN. Error losses of GAN for classes 3 and 4 are shown in Figure 3.12. Results of GAN for classes 3 and 4 is shown in Figures 3.13 and 3.14. We used 2 binary classifiers referred as $1C1D$ where we use 1 original class and 1 dustbin class. For example, 3 and 'not3' (generated from class 3 images using GAN) , 4 and 'not4' (generated from class 4 images using GAN). The marginal and reconstructional losses obtained from GAN network are represented in graphs as depicted in Figure 3.15. Further all the networks are tested using samples from the classes and outliers. Testing of $O5$ using classes and outliers are referred as $O5\_C$ and $O5\_Ot$ respectively. Testing of $1C1D$ using classes and outliers are referred as $1C1D\_C$ and $1C1D\_Ot$ respectively. We calculated the confidence using highest and 2nd highest(highest/2ndhighest) probability score of each sample. The

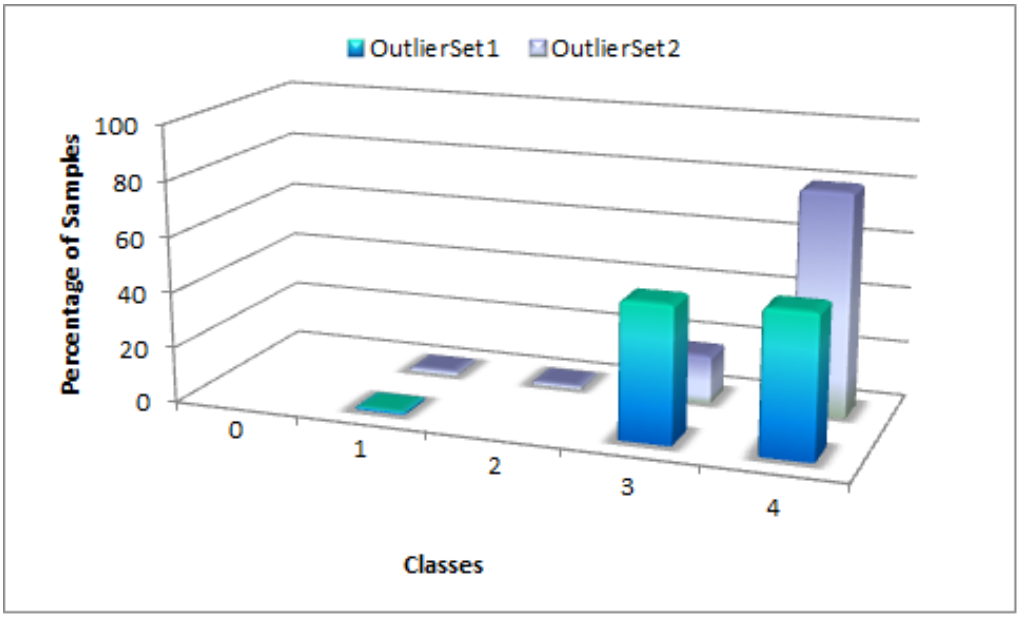
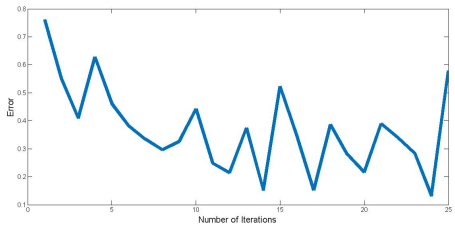


Figure 3.11: Capsule Network trained with 5 building classes when tested with different sets of outliers building classes show biasness of result towards 2 classes i.e. 3 and 4

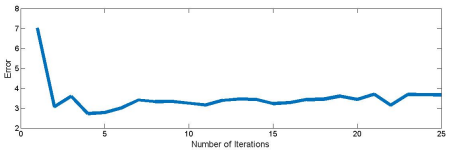
confidence obtained from all trained capsule networks is shown in Figure 3.16. Table 3.6 depicts that minimum confidence obtained from $O5$ for samples is 77.579% and maximum confidence obtained from $O5$ for outliers is 65.753%. This concluded that if we set the confidence level of $O5$ to be 75%, we will achieve a 100% successful outliers detection. The confidence level of $1C1D$ can be set as 60% to get 100% detection of outliers as minimum confidence obtained from $1C1D$ for samples is 75.588% and maximum confidence obtained from $1C1D$ for outliers is 56.101%. In both of these networks, confidence of most of the samples are lying between 90 to 95%.

3.4 Conclusion

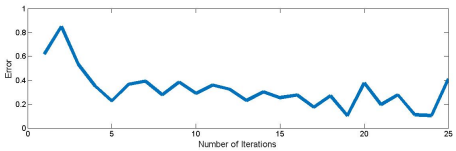
In this work, a scene localization system is developed using a 2 step procedure. Firstly, zone detection has been done by extracting zone wise features from last fully connected layer of places365 CNN and proceed with a Set based Classification. Three set-based difference extraction algorithms have been used for obtaining set-based differences, further aggregation of all the three normalized differences have been finally used for zone detection. Secondly, landmark is detected by training a compressed network derived from VGG detection model with PASCAL VOC dataset using weight projection and soft targets. This results in a network which is 1.6 times faster than the original. We achieved 99% accuracy for landmark recognition using a capsule network. We further improved



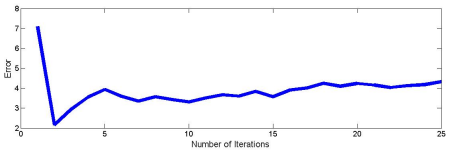
(a) Error loss of class 3 Discriminator



(b) Error loss of class 3 Generator



(c) Error loss of class 4 Discriminator



(d) Error loss of class 4 Generator

Figure 3.12: Error Loss graphs

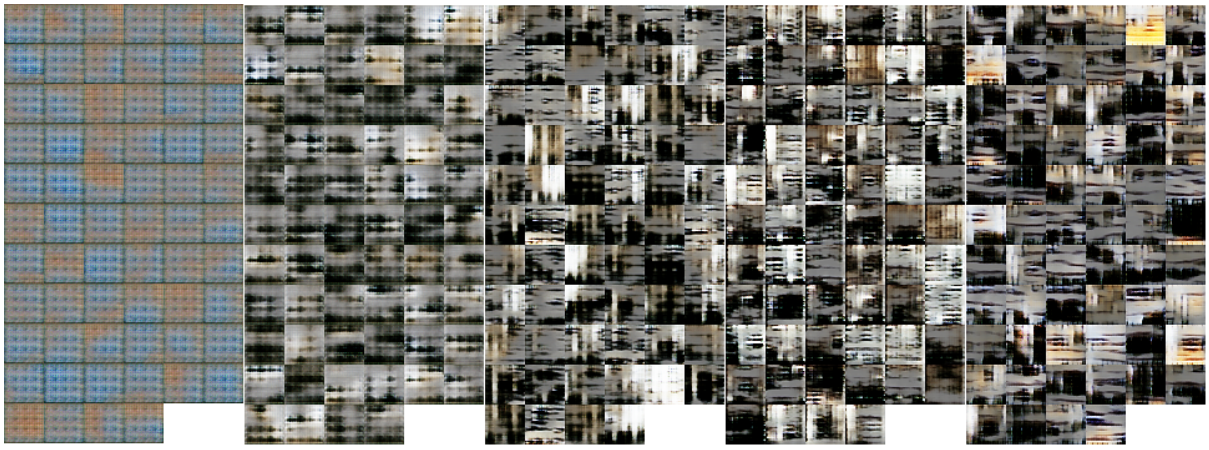


Figure 3.13: Dustbin class of 3 generated using GAN(4^{th},8^{th}, 16^{th} and 25^{th} iterations from left to right)

Table 3.6: Confidence obtained from different capsule networks

Network	Confidence Range	O5 (%)	1C1D (%)
Samples	70-80	2	5
	80-90	40	5
	90-95	56	90
Min Value		77.579	75.588
Outliers	0-10	12	27
	10-20	22	39
	20-30	6	24
	30-40	14	7
	40-50	13	0
	50-60	14	2
	60-70	10	0
Max Value		65.753	56.101

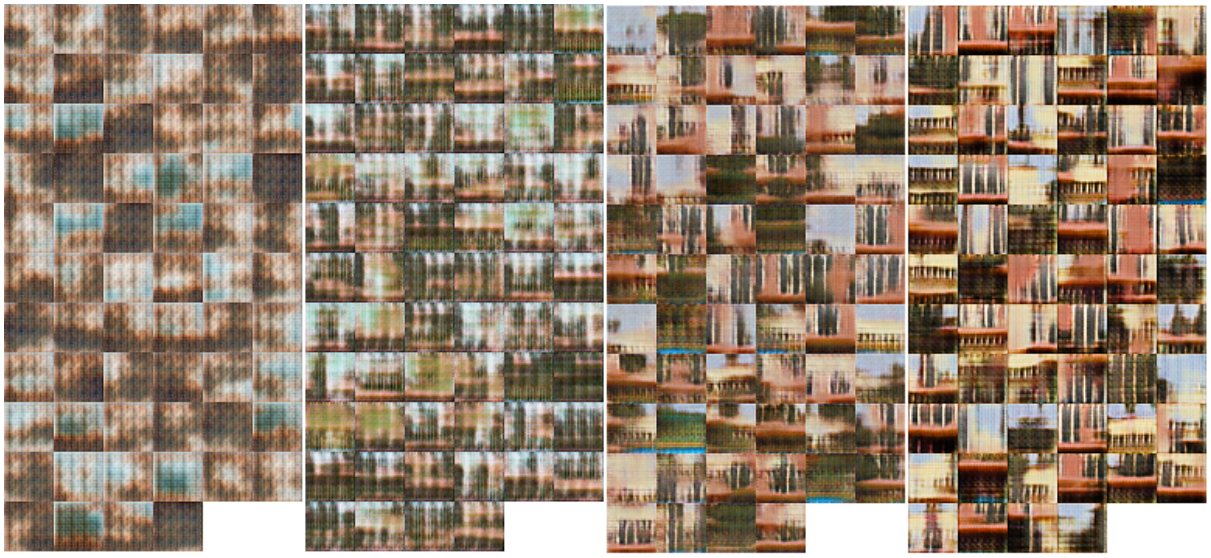
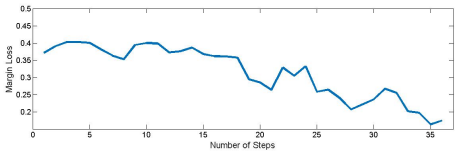
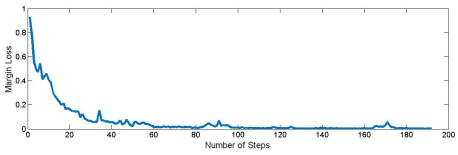


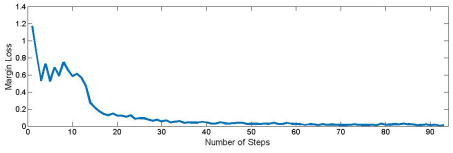
Figure 3.14: Dustbin class of 4 generated using GAN(4^{th},8^{th}, 16^{th} and 25^{th} iterations from left to right)



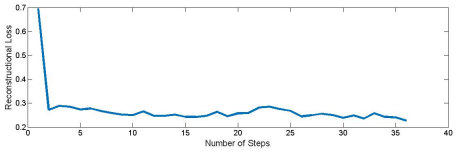
(a) Marginal Loss of 1C1D



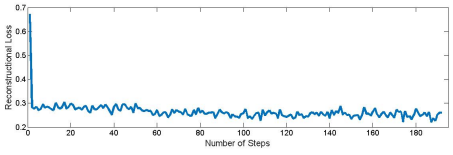
(b) Marginal Loss of O5_Ot



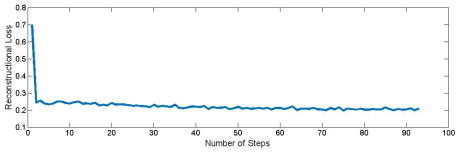
(c) Marginal Loss of O5



(d) Reconstruction loss of 1C1D



(e) Reconstruction loss of O5_Ot



(f) Reconstruction loss of O5

Figure 3.15: Marginal and Reconstruction Loss

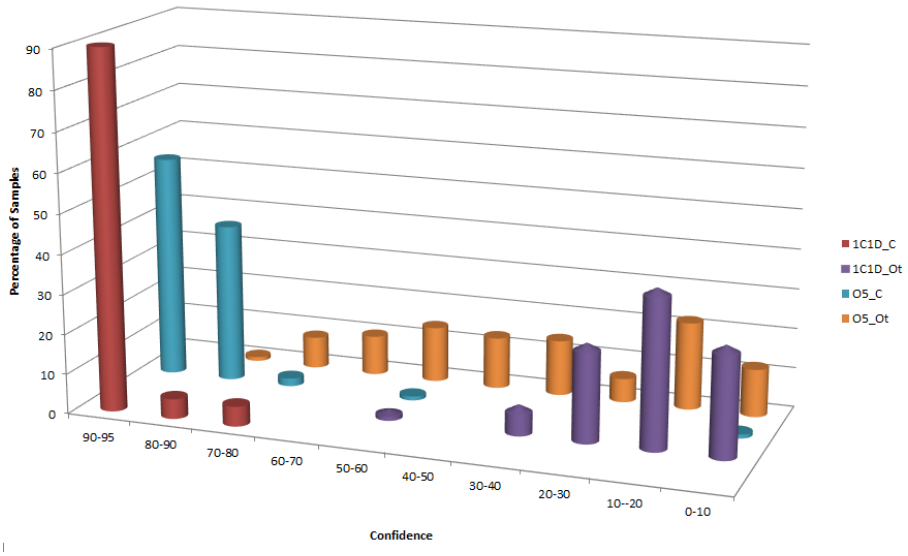


Figure 3.16: Comparison of Confidence based on probability calculated using highest and 2^{nd} highest probability score of each sample. It can be seen that confidence of majority of the samples is between 90 to 95% for $1C1D\_C$, while testing outliers $1C1D\_Ot$, maximum samples lie between 10-20% of confidence

network confidence to detect outliers by training it with dustbin classes obtained using GAN. We report confidence of (75%-56%) which is minimum and maximum confidence obtained from binary classifier network(1original+1dustbin class) using samples and outliers respectively.

Chapter 4

CNN based Object Detection and Classification

Vehicle detection and classification is an important task for street surveillance and scene perception for robot navigation or autonomous vehicles. A lot of work has been reported for real time traffic detection by [211], [212] and [213]. The choice of particular algorithms for effective scene classification depends on speed requirements [214] and availability of multicue data [215]. While some of these utilized handcrafted feature extraction techniques such as HOG [211] and Haar-like features [216] for feature extraction, most of the researchers have now shifted to deep learning based object detectors such as R-CNN[89], fast R-CNN[90], faster R-CNN[91] and YOLO [92]. Different vehicle detection systems reported in literature also apply fine-tuning to pre-trained models [217], [218] with their own data to enhance the performance. Apart from concentrating on network architectures and features, many of these also aim towards making the detection system more robust by specifically accommodating techniques to deal with distortions like blur [219] and noise [211] in the input. Particular traffic detection systems dealing with varying weather conditions [220] and night time environments [221] have also been reported. This chapter focuses on traffic detection for real time applications using three components. The first component includes designing convolutional feature map-based classifier based on multimodal optical flow features. The second component is to utilize an effective adaptive learning rate technique to deal with saddle points; and propose an average covariance matrix based pre-conditioning approach. The third component is to separately train multimodal model using blur data which caters blur effect of real time data. Extensive experimental results with different learning rates, architectures are reported using benchmark datasets such as Apollo, KITTI, Cityscapes, Berkeley, Caltech, PASCAL VOC and self created. For the implementation of this, dataset has been prepared as described in section 4.2.1 to achieve good accuracy. Further various experiment set-up have been discussed in section 4.2.2. And then results for the experiments performed have been presented in section 4.3.

4.1 Background

Considerable amount of work has been reported on candidate region detection as well as classification on different categories of images. The research works undertaken for review have been classified into two categories. The works included in the category (I) focus on application specific detection tasks, while the category (II) includes studies which aim at improving detections with respect to speed, accuracy and false alarm detections.
(I) CNN-based detection and classification techniques have been implemented for detecting pedestrian, cyclist, vehicle type, animals and many more. A combined framework for concurrent detection of a pedestrian and a cyclist has been proposed by [222] using R-CNN on upper body regions detected with ACF, LCDF [223]. [224] classified different vehicle types(car, truck, bus and van) from different views using a multi-task R-CNN. An animal detection technique using multilevel graph cut for combination motion with spatial context has been presented in the work of [225]. The feature description used for animal detection was a combination of deep learning (pre-trained caffe CNN) and oriented gradient histogram features encoded with Fisher vectors. [220] fine-tuned their own vehicle dataset using GoogLeNet, pre-trained with ILSVRC-2012 data to obtain vehicle classification results. [226] detected vehicles using Bayesian probability model and classified multiple types of vehicle by adopting AlexNet as classifier pre-trained with ILSVRC 2012 ImageNet data set. Along with classification, their framework also detected vehicle location. [227] detected vehicles using pre-trained fast-RCNN network and classified them into types through VGG_CNN_M_1024 model. A K-means algorithm has been utilized for clustering the vehicle data prior to training of VGG_CNN_M_1024.
(II) While most of the available research focused on using different networks and classifiers for specific applications, some researchers have particularly focused on increasing speed and accuracy while reducing probability of false alarms. For example, [228] presented an accelerating method that proved to be effective for very deep models. They proposed a response reconstruction method that takes into account the non-linear neurons and a low-rank constraint. A solution based on Generalized Singular Value Decomposition (GSVD) has been developed for this non-linear problem, without the need of stochastic gradient descent (SGD). Their method has been evaluated under whole-model speedup ratios. It could effectively reduce the accumulated error of multiple layers due to the non-linear asymmetric reconstruction. A method to reduce false alarms has been introduced by [229] where detection results have been propagated to adjacent frames according to motion information. The resultant duplicate boxes have been removed by non-maximum suppression (NMS). Another effective approach to reduce false alarms including context-based CNN object detection model has been introduced by [230]. [221] have proposed a

NOSCOPE system for the purpose of accelerating neural network for video with the help of inference-optimized model search. Table 4.1 provides the summary of existing object detection and classification algorithms in terms of following parameters: A-whether the algorithm considered blur or noisy image data during training or testing, B-robustness of the algorithm at different weather conditions particularly night time environment, C- use of high end hardware, particularly for testing, D- processing time, E- Number of objects considered for detection and classification, F- whether the detection decision is influenced by the presence of other objects in the frame or the detection in successive frames is dependent on the availability of object in the previous frame, G- whether multispectral or multimodal features have been taken into account, H- data has been pre-processed or not. It can be observed from the table that in most of the works, fast and faster R-CNN have been used for detection. However, some of them used handcrafted feature extraction techniques [211]. In general, each of them focused towards a specific goal. For example, [228] particularly focused on enhancing time and did not consider other parameters like data preprocessing, multimodal/multispectral features, blur/noise, etc. Multispectral/multimodal features have been considered for pedestrian detection [218] and for predicting the presence or absence of an object [221], [217] and [214].

Table 4.1: Summary of Related works (A-Blur/Noise, B-Weather/Night, C-Hardware, D-Processing time, E-Number of Objects, F-Tracking/ Contextual, G-Multimodal/ Multispectral, H-Pre-processing)

Author(s)	**Features**	**Work done**	**Data set used**	**A**	**B**	**C**	**D**	**E**	F	G	H
[216]	Adaboost, Haar-like features	Vehicle tracking	TME motorway	√					√		
[218]	Faster R-CNN	Pedestrian detection	KAIST pedestrian	√				1		√	√
[221]	NN+ yolo v2	Presence or absence of a given class of object	MS-COCO	√	√	√	√	yes/ no	√		√
[228]	VGG16+Fast R-CNN	Detection and classification	PASCAL VOC 2007				√				
[229]	T-CNN	Detection and classification	ImageNet VID, YTO				√	30	√		
[220]	GoogLeNet	Vehicle classification	Own constructed	√	√		√	6			
[230]	AC-CNN	Proposed Attention to Context CNN object detection model	PASCAL VOC 2007, 2010, 2012				√	20	√		
[223]	HOG, Decision tree	Pedestrian Detection	INRIA, Caltech pedestrian				√	1			√
[222]	Fast R-CNN	Pedestrian, cyclist detection	Own created pedestrian and cyclist			√	√	2			

Table 4.1 continued from previous page

Author	Features	Work done	Data set used	A	B	C	D	E	F	G	H
[224]	R-CNN	multi-view classification of vehicles	Own created		√			4			
[226]	-Bayesian framework -CNN	Classify multi vehicle types	Own created	√	√		√	3			
[227]	-Fast R-CNN, K-means	Vehicle type classification	Own created		√	√		3			√
[211]	HOG, KNN, SVM	object classification and tracking (pedestrian, motorcycle, car and van)	Real time	√				4	√		
[231]	Object detection in Adaptive Cruise Control using Closest in Path Vehicle (CIPV),SVM	Object Detection	Collected from vehicles radar	√			√	2			
[215]	Faster R-CNN	Cyclist detection	KITTI					1			
[212]	Aggregated channel features (ACF), AdaBoost classifier	Pedestrian detection	Real time			√	√	1	√		
[232]	Particle based filter	On road vehicle tracking	Beijing highway				√	1	√		
[213]	Scaled Unscented Kalman Filter (SUKF)	Vehicle Tracking	Real time		√		√	1	√		√
[233]	context-based feature descriptor in combination with a SVM classifier	Pedestrian Detection	Own created				√	1			
[217]	CNN(ResNet)	Pedestrian Detection	KITTI				√	1		√	√
[214]	CNN	person, bicycle, motorbike, tricycle, car, truck, van and bus	KITTI			√	√	8	√		

Tracking method has been applied by some of the researchers like [214], [213], [232], [211], [212]etc. Most of them did not consider explicit data pre-processing techniques with the exception of [218], [221], [223], [227], [213] and [217]. [217] reported use of special hardware for detection purposes. Some of them considered blur/noisy image data and/or different weather conditions [221] [220] [226] [231] . [216], [230] and [229] focused on frame based information for detection and classification. However, robustness in terms

of different weather conditions and blur/noise has not been considered. [221] considered most of the parameters other than multimodal/multispectral data for detecting presence or absence of class of objects.

In the present research work, the objective is to detect traffic regions from a scene and further use findings for a dedicated vehicle classification CNN [209]. In this chapter, the major factors which would help enhance the performance of existing detection architectures for example faster R-CNN when reused, have been initially identified .

Dataset preparation: Data preparation is an essential factor to be considered for fine tuning the network. Data augmentation [234], [235] is the widely adapted solution when available data is limited. Also, while preparing data from video frames, special care needs to be taken to avoid homogeneity. Evidences of data pre-processing using selective search and decorrelation can be found in work reported by [227] and [223] respectively. In this work, the dataset has been pre-processed using FFT and key frame selection techniques.

Architecture: Fine tuning of existing architectures has been widely used by [236], [217]. In this process, particular care should be taken for low gradients generated during learning which led to slow or no convergence. Use of adaptive learning rates [237] [238] [239] proved to be beneficial in this case. The design of a deep network plays an important role in object classification . While exploring the network design for classification, it has been analyzed that not only the deep feature maps, but also a deep and convolutional per region classifier has to be taken into consideration due to its special importance for object detection. In their study, [85] presented that the detection accuracy of GoogLeNets [240] and ResNets [241] was further improved after the use of a per region classifier. Hence, for the purpose of this research work, various per region classifier architectures have been used for NoCs to measure and compare the performance of object detection and classification.

Features: The features provided to the network for classification are also important. It has been observed that for different object detection challenges, deep neural networks improve the performance by averaging over different crops or scale of a particular image. PCA and whitening of pixels have been used to reduce the over-fitting problem in Imagenet [242]. This caters for intensity variations in the training image. Some researchers found the use of RGB images with depth data for improving the accuracy of object detection [243][244]. These include training a network from scratch using RGB, depth and/or LIDAR data or fine tuning pre-trained nets like VGG/Alex-net with depth/LIDAR data for improving object detection performance. In the current work, orientation value obtained from Optical Flow has been

used with the consideration that it will encapsulate pose information. Optical features have been used for pedestrian detection [245], Occlusion detection [246] [247], and visual odometry [248], while the former used HOG and LUV features, and the later two used CNN features. However, CNN trained with multimodal features for vehicle detection and classification is yet to be found.

Blur effect: There are various reasons [249] of blur effect in images or frames acquired from videos. Blurred images captured due to random movement of camera or any other reason, can have a negative impact on the network. So, care has to be taken to eliminate blurred images or remove blur effect from images [86] [87] [88]. The network could also be trained using blurred images. An example of the same is available in the work undertaken by [236] where GoogleNet was fine tuned using blurred data to improve the results. [219] presented the performance of VGG 16 network tested with different degree and type of blur images and subsequently fine-tuned the state-of-the-art network with the same type of data. In the current work, the NoC architecture has been trained using multimodal features obtained from normal as well as blurred data.

4.2 Multimodal Object Detector

The training and testing phases have been briefly discussed using Figures 4.1 and 4.2. NoC is defined as the network or classifier that has been generated using shared weights of VGG16 and adding fully connected layers to it. The input to the architecture of NoC is the features extracted from last convolutional layer of VGG16 and it has been further fine-tuned with the features of selected dataset. The three architectures of NoCs that have been used to develop the CNN-based classifier are as follows: First one is the widely used multiple fully connected (fc) layers for classification (referred as $0C3fc$). The second architecture used 1 spatial convolutional, followed by 3 fc layers (referred as $1C3fc$). The third architecture used 2 convolutional layers which has been henceforth called $1M1$. As is evident from Figure 4.1, FDS1,FDS2,FDS3 are the extracted features of RPNs of dataset DS1,DS2,DS3 respectively. These features are passed to three architectures separately. This particular design has been motivated from the study conducted from [85] where it was seen that the design of deep network played an important role for object classification. While exploring the network design for classification, it has been analyzed that not only the deep feature maps, but also a deep and convolutional per region classifier was taken into consideration due to its special importance for object detection. In their study, [85] presented that the detection accuracy of GoogLeNets and ResNets further improved

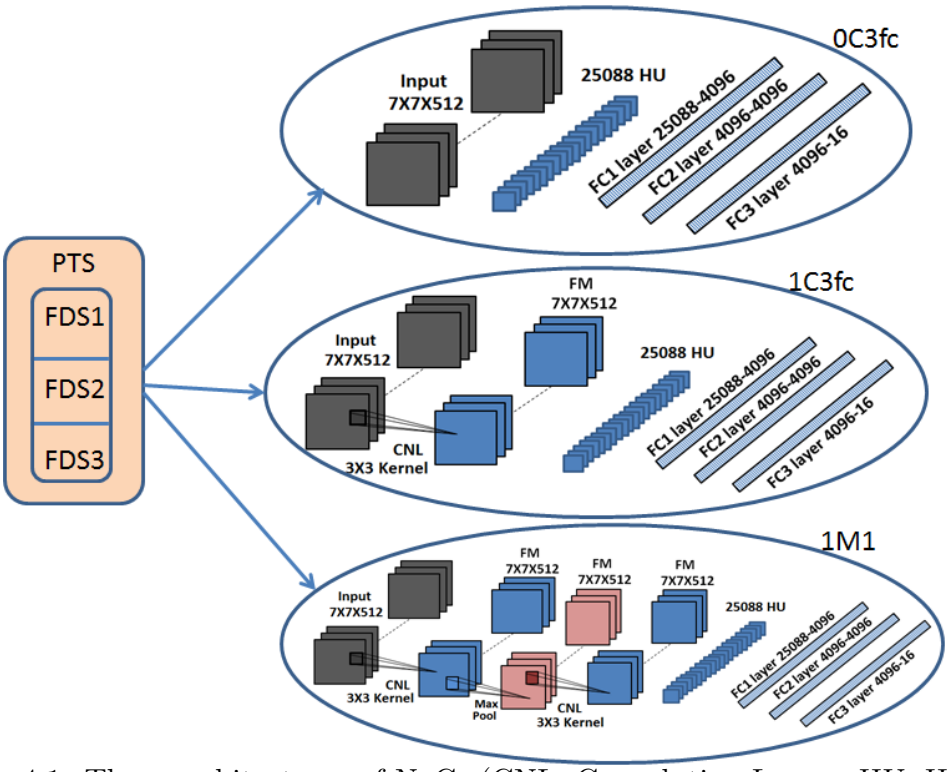


Figure 4.1: Three architectures of NoCs (CNL: Convolution Layers, HU: Hidden Units, FM: Feature Map)

after the use of a per region classifier. Thus, for the purpose of this research work as well, various per region classifier architectures have been used for NoCs to measure and compare the performance of object detection and classification. Algorithm 4.1 shows the steps for training the network. The training has been repeated for RGB intensity-based features and multimodal features. Further, these features have been computed using normal images as well as blurred images. The experiments have been performed using three different learning rates referred as 1LR, 2LR and 3LR. These have been further elaborated in section 4.2.2.1. Algorithm 4.1 particularly shows the flow of training using adaptive learning rate (3LR). As shown in Figure 4.2, various features have been extracted individually and passed to the trained NoCs separately. Experimental details have been provided in Section 4.2.2.

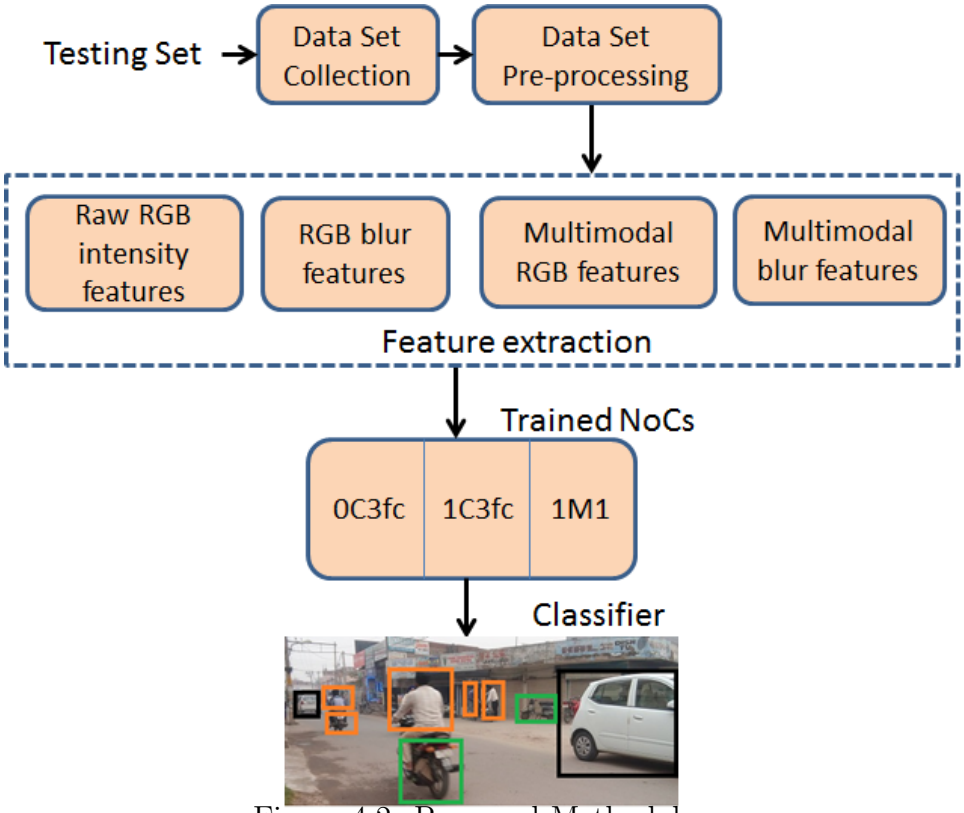


Figure 4.2: Proposed Methodology

Algorithm 4.1 Training NoC classifier

Input: Data Set DS $=\{(x_1, y_1), \cdot \cdot \cdot, (x_n, y_n)\}$,$x_i \epsilon X$, number of classes C, $y_i \epsilon Y = y_1, y_2, \cdots, y_c$

Output: $R = [O \epsilon X, y \epsilon Y]$, O is object detected, y is its label.

1: Divide DS into 3 equal parts DS_j, where j=1,2,3.

2: Extract features (F) of DS_j from last max pool("$pool5$",32^{nd}) layer of VGG16, where $F \epsilon$ {RGB, RGB blur, Multimodal RGB, Multimodal blur}

3: Pass DS_j for ROI pooling using standard Region Proposal Algorithm [91].

4: For A=1,2,3, where A ϵ $\{0C3fc, 1C3fc, 1M1\}$

Train these networks with RPN features of DS_j using weight updation as given in equation 4.7.

EndFor

4.2.1 Dataset Preparation

Number of datasets are available for object detection and classification. [250] have discussed details regarding usage of datasets. In this work, NoC has been evaluated using

various datasets such as Caltech [251], PASCAL VOC [252], Apollo [93], Berkeley [96], Cityscapes [95], KITTI [94] and self created.

- The Caltech 256 object categories dataset has 30607 images. It has 256 classes out of these classes, objects required for the proposed work for example; pedestrians, motorbikes, bicycles, etc have been used. It is referred as CTS.

- Berkeley is a large dataset of natural images manually segmented. Berkeley subset used for experimental results in this work corresponds to 300 images available online.

- The KITTI dataset was recorded from a moving platform while driving in and around Karlsruhe, Germany. Up to 15 cars and 30 pedestrians have been visible per image. Raw dataset of various categories are available. Sequences from categories such as city, road and person have been used for purposed work.

- Cityscapes is a large-scale dataset that contains a diverse set of stereo video sequences recorded in street scenes from 50 different cities, with high quality pixel-level annotations of 5000 frames in addition to a larger set of 20000 weakly annotated frames. Left images of 18 cities have used for our purpose.

- ApolloScape contains large and rich labelling including holistic semantic dense point cloud for each site, stereo, per-pixel semantic labelling, lanemark labelling, instance segmentation, 3D car instance, high accurate location for every frame in various driving videos from multiple sites, cities and daytimes. Apollo data subset taken from camera 5 and 6 have been used for purposed work.

- PASCAL VOC dataset is available in various versions such as 2007, 2009 and 2012. PASCAL VOC 2007 has been used for purposed work and is referred as PTS. It would be pertinent to note that for the different experiments, every NoC has been trained using 1300000 RPNs of PTS. Around 300 region proposals have been extracted from each image using Region Proposal Network (RPN).

- For purposed work, a new dataset was created (referred as OTS) by capturing videos using "Sony Cyber-shot DSC-T77" 10.1 MP camera with resolution of 640 x 480. Frames have been extracted from the videos, a few of the frames had to be discarded manually as they did not contain the required objects. Data was collected at different spots and timings with a camera mounted on a tripod which was periodically moved at different pan and tilt angles for posture variations. While extracting data from videos, data non-homogeneity has been maintained by using FFT [253] and key frame selection techniques [254].

4.2.2 Experimental Set-up

The following experiments have been set up using different hyper parametric variations, architectural variations, blur network and comparison with benchmark datasets, as listed below:

A. *Learning Rates*: The learning rate hyper parameter has been set by performing three different learning rate experiments using NoC architecture having 1 convolutional and 3 fully connected layers (1C3fc). Intensity(RGB) image features (F_{Int}) extracted from VGG(last max pool($'pool5'$,32^{nd}) layer from 41 layered network) have been passed as input. Various models prepared have been named as I_1C3fc_1LR, I_1C3fc_2LR and I_1C3fc_3LR. The most suitable (I_1C3fc_3LR) has been further used for the remaining set of experiments. This has been explained in 4.2.2.1.

B. *NoC Models*: Three different architectural variations ($0C3fc$, $1C3fc$ and $1M1$) with input as intensity image features, have been used . Models prepared have been named as I_0C3fc_3LR, I_1C3fc_3LR and I_1M1_3LR. The best performing architecture ($1C3fc$) with best performing learning rate ($3LR$) has been utilized for other experiments. NoC models have been elaborated in 4.2.2.2.

C. *Features*: Multimodal features have been used for various experiments. Fusion techniques of other features with RGB features have also been discussed in 4.2.2.3.

D. *Blur Network*: Blur network has been trained using normal as well as multimodal features. These have been further tested with blurred as well as normal data. The set up has been discussed in detail in 4.2.2.4.

4.2.2.1 Learning Rates

Learning rate plays an important role for the convergence of training loss. RMSProp used Hessian-based pre-conditioning with first order gradients for adaptive learning rates. However, it is important to effectively handle noise included in first order gradients during stochastic optimization (mini batch settings). Other variants of RMSProp such as AdaDelta and Adane have also been considered superior to SGD in terms of training speed based on the fact that saddle points slow-down the progress of first order gradients. SGD iteratively updates the parameter θ as shown in equation 4.1:

$$\theta_t = \theta_{t-1} - \alpha \nabla f(\theta_{t-1}) \tag{4.1}$$

α is the learning rate; and $\nabla f(\theta_{t-1})$ is the first order gradient. The updating value of RMSProp is given in equation 4.2:

$$\begin{aligned} \theta_t &= \theta_{t-1} - \frac{\alpha}{\sqrt{\psi_t} + \varepsilon} \nabla f(\theta_{t-1}) \\ \psi_t &= \beta \psi_{t-1} + (1-\beta) \nabla f(\theta_{t-1})^2 \end{aligned} \tag{4.2}$$

Where, β is the decay rate for computing ψ_t. In Hessian-based conditioning, the training efficiency is increased by reducing the Hessian condition number by transforming the parameters as represented in equation 4.3:

$$\theta_t = \theta_{t-1} - \alpha D^{-1} \nabla f(\theta_{t-1}) \tag{4.3}$$

Here, $D = \sqrt{diag(H^2)}$, which worked well even when H was indefinite as in the case of saddle points. [237] verified that $\sqrt{\psi_t}$ can be used as D. [238] used a covariance matrix-based pre-conditioning to deal with noisy gradients in mini-batches. They argued that if covariance $c[i, j]$ has a large value, then the gradient strongly oscillates leading to inefficient progress of updating directions. The gradients have been pre-conditioned as shown in equation 5.12. Covariance c_t^2 and Mean μ_t are given in equation 4.5.

$$\theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{c_t^2} + \varepsilon} \nabla f(\theta_{t-1}) \tag{4.4}$$

$$\begin{aligned} c_t^2 &= \gamma c_{t-1}^2 + \gamma(1-\gamma)(\nabla f(\theta_{t-1}) - \mu_{t-1})^2 \\ \mu_t &= \gamma \mu_{t-1} + (1-\gamma) \nabla f(\theta_{t-1}) \end{aligned} \tag{4.5}$$

where, γ is the hyper-parameter of the decay rate. For the purpose of this work, PASCAL VOC 2007 dataset has been divided into three parts, i.e., DS1, DS2 and DS3. The following methods have been used to train the networks:

1. DS1 has been trained for 100 iterations. The trained net has been then used to train DS2 which has been further used for DS3. In all the cases, learning rate of linear decay from 0.01 to 0.005 has been used; and weights have been updated according to equation 4.1. This is named as 1LR.

2. In this case, net trained with DS1 has been used for DS2 and similarly net trained with DS2 has been used for DS3. However, weight updation in this case has been done according to equation 4.2. It has been named as 2LR.

3. In this case, DS1,DS2 and DS3 have been trained and the weights have been updated

as discussed in equation 4.3, rewritten as equation 4.6. Final weights θ_t^f have been updated as shown in equation 4.7. This process has been abbreviated as 3LR.

$$\theta_t^j = \theta_{t-1}^j - \frac{\alpha}{\sqrt{c_t^2 + \varepsilon}} \nabla f(\theta_{t-1}^j) \tag{4.6}$$

$$\theta_t^f = \frac{1}{3} \sum_{j=1}^{3} (\theta_t) \tag{4.7}$$

The results for the same are exhibited in Table 4.2 in section 4.3.1.

4.2.2.2 NoC Model

Features extracted from last convolutional layer of VGG16 have been passed to Region Proposal Network for extracting region proposals. They have been further passed to NoC models. Samples of the region proposals with their scores obtained from detection algorithm are shown in Figure 4.3.

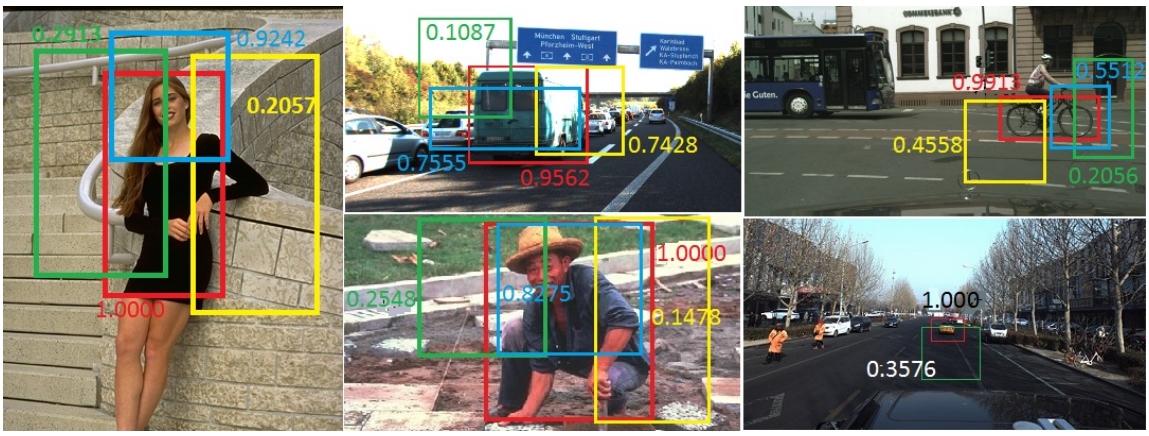


Figure 4.3: Region proposals using object detection algorithm

Experiments have been performed using three different architectures ($0C3fc$, $1C3fc$ and $1M1$) with the best performing learning rate 3LR. The best performing architecture has been utilized for further experiments. The results of experiments performed using three NoCs with different test sets are shown in Table 4.3 in section 4.3.2.

4.2.2.3 Features:Multimodal CNN based feature extraction techniques

For getting multimodal features, intensity images features extracted from 5th convolutional layer $Conv5\_F_I$ have been fused to other features extracted using edge, optical flow and scale space representations. Further these features were applied to Region Proposal Network (RPN). It includes edges obtained from canny, sobel and prewitt edge detection algorithms; scales values of $t = 3$ and 5; orientation data of optical flow for extracting pertinent features through different scale and orientation information of an object. The motivation behind each has been provided below:

- Improving the visibility through sharpening techniques often lead to better feature detection accuracy due to increased depth and detail clarity tidiness and depth [255, 256] . In the proposed work, the images have been sharpened by fusing (adding) various edge features which results in improved feature extraction by neural network than using normal intensity images.
- Optical flow allows us to retain the benefits of motion information. Sun et al.[257] proved that CNN fine-tuned with optical flow features in combination with RGB features performed better than using only RGB features. Moreover, calculation of the motion dynamics at the feature level is faster as well as more robust. This is evident as deep features are capable of noise elimination from the raw input and providing more semantic and discriminative representations. Joe et al.[258] also proved that using optical flow features for training a CNN can greatly benefit the classification accuracy. Since our networks process video frames at 1fps, they do not use any apparent motion information. Therefore, we additionally train our model on optical flow images.
- A single-scale representation blurs salient information (useful in object matching) of different scales. Features extracted from various scales will contain more pertinent information which improves the training of CNN. This has been used for various purposes using CNN [85, 259] since a long time. The reason for using this in proposed work is that Multi-scale[85] visual information includes feature representations at both global contextual and local saliency scales.

The PASCAL VOC dataset is used for extracting the features from 5th convolutional layer ($conv5$) of VGG 16. The steps of training and testing are shown in Algorithm 4.2. Networks used for intensity image, edge image (canny, sobel and pewitt), Gaussian image (t=3,5) and optical flow image are named as $CNN - Int$, $CNN - Edges(CNN - Edges_C, CNN - Edges_S, CNN - Edges_P)$, $CNN - Gauss(CNN - Gauss_3, CNN - Gauss_5)$ and $CNN - Opt$ respectively.

Features from $conv5$ of $CNN-Int$ named as $conv5\_F_{Int}$ are fused with features of $conv5$ of ($CNN-Edges_C(conv5\_F_{Edges_C}), CNN-Edges_S(conv5\_F_{Edges_S}), CNN-Edges_P(conv5\_F_{Edges_P})$) separately for resulting features map (F_{Edges}). In the same manner, feature map (F_{Opt}) is obtained by fusing $conv5\_F_{Int}$ and $conv5\_F_{Opt}$. Feature map (F_{Gauss}) is obtained by fusing $conv5\_F_{Int}$, $conv5\_F_{Gauss_3}$ and $conv5\_F_{Gauss_5}$. These feature maps are further passed for ROI pooling and classifies using two different classification networks CNN_1C and CNN_0C as shown in Figure 4.4. CNN_0C is a brief name for I_0C3fc_3LR has three fully connected layers while CNN_1C is a brief name for I_1C3fc_3LR has one convolutional layer with three fully connected layers.

Algorithm 4.2 Obstacle predictor and classifier along with distance for blind

Input: Training Set P $=\{(x_1, y_1), \cdot\ \cdot\ \cdot, (x_n, y_n)\}$,$x_i \epsilon X$, number of classes C, $y_i \epsilon Y = y_1, y_2, \cdots, y_c$

Output: $R = [O \epsilon X, y \epsilon Y]$, O is object detected, y is its label.

1: Divide P into 3 equal parts P_j where j=1,2,3.

2: For j=1 to 3

(i) Extract edges, scale space and optical features of intensity images(I) of P_j.

(ii) Fuse convolutional feature maps of I with A, where $A \epsilon \{E, G, O\}$ in which E is for edges(canny(E_C), sobel(E_S), prewitt(E_P)), G is for Gaussian(G_3,G_5),O is for optical flow.

(iii)Pass the fused feature maps for ROI pooling using Region Proposal Algorithm as given in Algorithm 4.3

(iv) Create two networks having 1convolution and 3 fully connected layers(CNN_1C) and only three fully connected layers(CNN_0C) using shared weights of VGG16

(v) Train this network with $P_j \bigcup \{x_i, y_i\}$

(vi) Output the trained net which can predict and classify the obstacle.

EndFor

3: Using trained net, features of test set are extracted and SVM is trained to classify.

4: Calculate accuracy by comparing the predicted and actual output.

Algorithm 4.3 Region Proposal Algorithm

(*a*) The first step is that image is given as input to a convolution network which will output a set of convolutional feature maps on the last convolutional layer

(*b*) Then a sliding window of size n x n is run spatially on these feature maps. A set of anchors are generated which all have the same center but with different aspect ratios and different scales. All these coordinates are computed with respect to the original image.

(*c*) For each of these anchors, a value p* indicated how much these anchors overlap with the ground-truth(GT) bounding boxes. p* is computed as shown in equation below:

$$P^* = max \begin{cases} 1 & \text{if } IU > 0.7 \\ -1 & \text{if } IU < 0.3 \\ 0 & \text{otherwise} \end{cases}$$

where IU is intersection over union and is defined below in equation:

$$IU = \frac{Anchor \bigcap GTbox}{Anchor \bigcup GTbox}$$

Fusion: Fusion at sensor level, feature level and decision level is a long practiced technique for elevating the recognition performance and also complementing information in some cases. We particular look at feature fusion where multiple features are extracted and fused via a linear [260] or a non-linear classifier [261]. While these reported handcrafted feature fusion, CNN feature fusion has also been explored [262]. The authors use a concatenation based fusion, which leads to the enlarged size of feature map and take lot of computational time and space. Hence in this work, addition and/or maximum of features are used which retain the size of feature map. There are three cases of fusion of feature extraction:

(a) In case of edges, features are fused by adding them ($conv5\_F_{Int}$ and $conv5\_F_{Edges_C}$), ($conv5\_F_{Int}$ and $conv5\_F_{Edges_S}$) and ($conv5\_F_{Int}$ and $conv5\_F_{Edges_P}$). Further taking the maximum of these three gives the final feature map (F_{Edges}) as shown in equation 4.8. The process using different network classifiers (CNN_1C and CNN_0C) is shown in Figure4.4.

$$F_{Edges} = max \begin{cases} conv5\_F_{Int} + conv5\_F_{Edges_C} \\ conv5\_F_{Int} + conv5\_F_{Edges_S} \\ conv5\_F_{Int} + conv5\_F_{Edges_P} \end{cases} \tag{4.8}$$

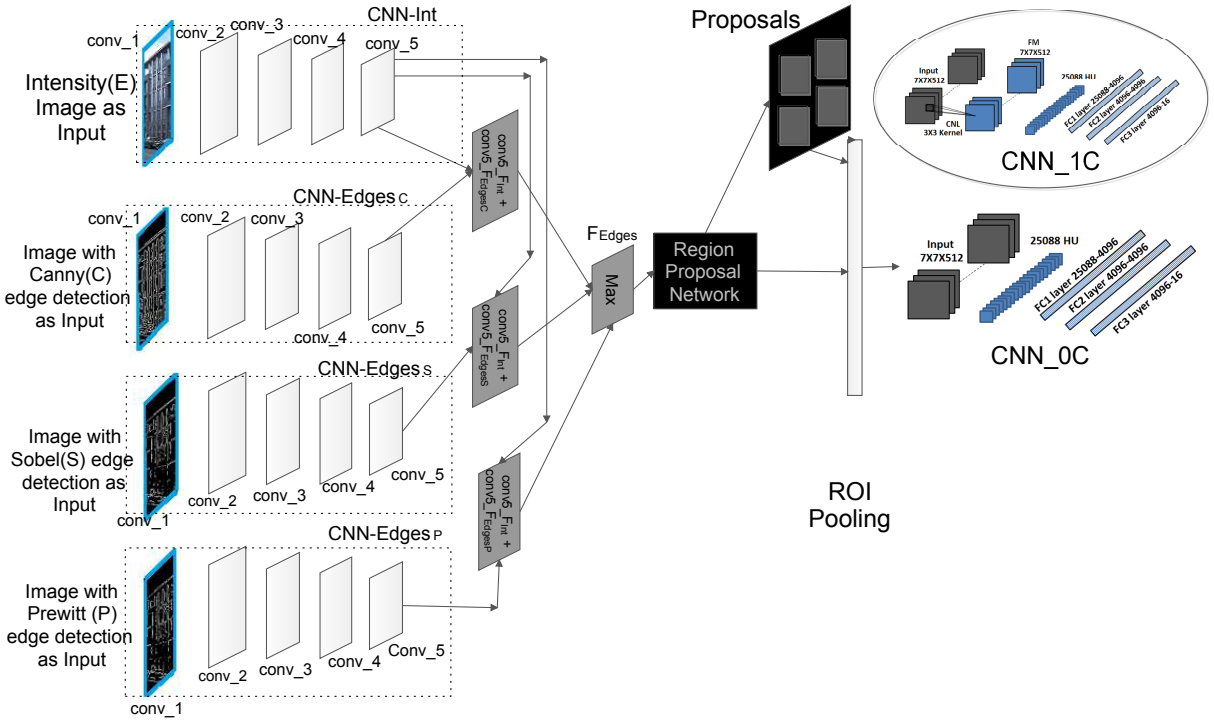


Figure 4.4: Multimodal object detection and classification using CNN_F_{Int} and F_{Edges}

(b) For optical flow, $Conv5\_F_{Int}$ are fused (added) to orientation features ($conv5\_F_{Opt}$). Feature map F_{Opt} is obtained as shown in equation 4.9. The whole process is shown in Figure4.5 with classifier network CNN_0C. It is also done with CNN_1C.

$$F_{Opt} = Conv5\_F_{Int} + conv5\_F_{Opt} \tag{4.9}$$

(c) For scaled images, the fusion is done by taking maximum of the features of $Conv5\_F_{Int}, conv5\_F_{Gauss_3}$ and $conv5\_F_{Gauss_5}$. The feature map (F_{Gauss}) is obtained using equation 4.10. The process is shown in Figure4.6.

$$F_{Gauss} = max \begin{cases} Conv5\_F_{Int} \\ conv5\_F_{Gauss_3} \\ conv5\_F_{Gauss_5} \end{cases} \tag{4.10}$$

The results of experiments performed with various multimodal features are shown in Table 4.4 in section 4.3.3.

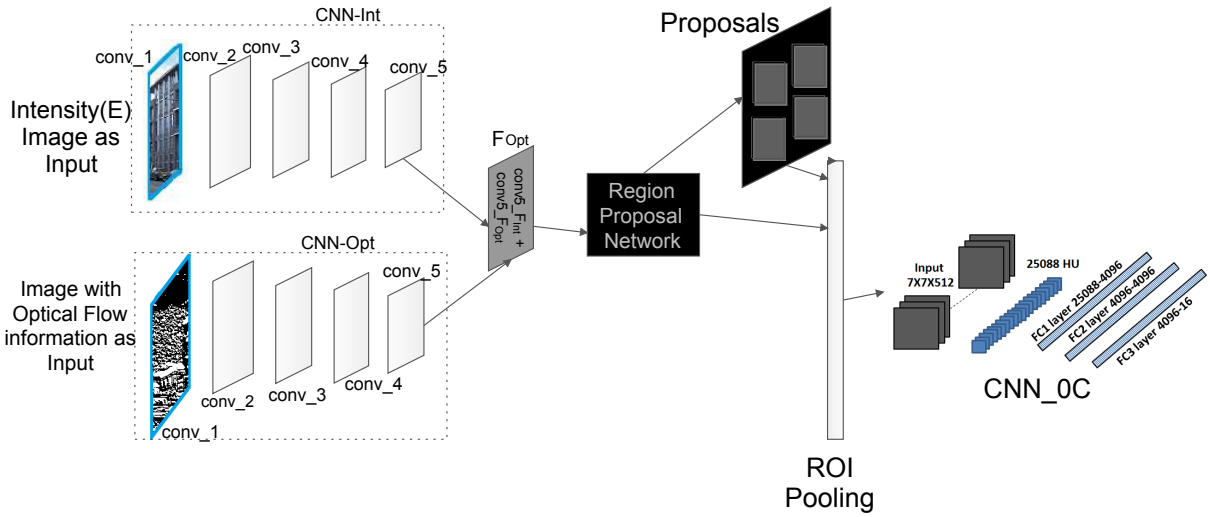


Figure 4.5: Multimodal object detection and classification using CNN_F_{Int} and F_{Opt} features

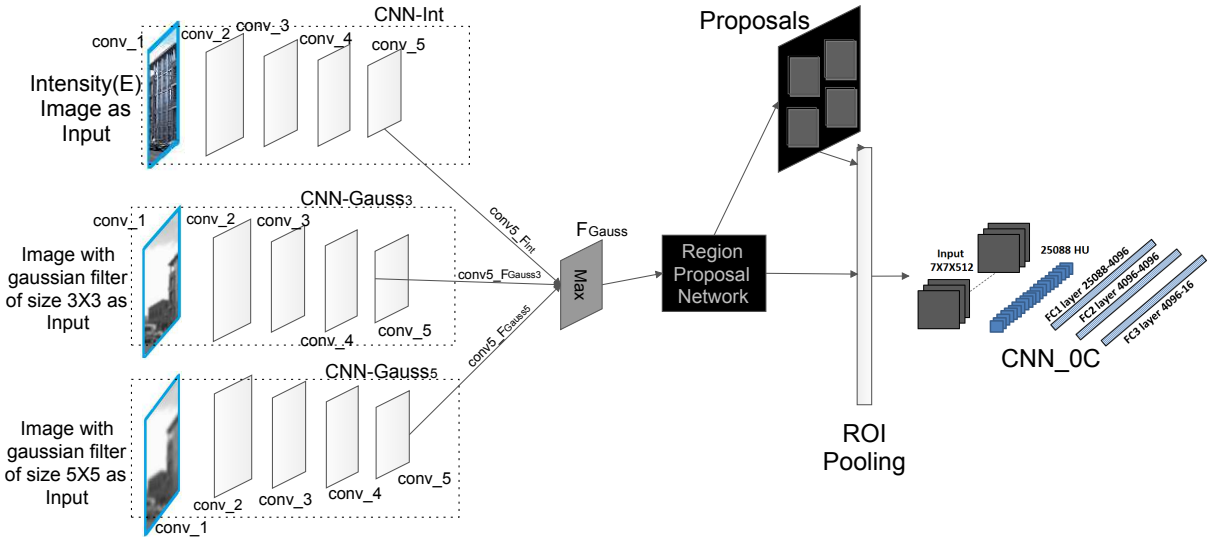


Figure 4.6: Multimodal object detection and classification using CNN_F_{Int} and F_{Gauss} features

4.2.2.4 Blur Network

When blurred data has been given to the networks trained on normal images, they gave poor results of classification accuracies as shown in Table 4.5. Hence, the networks have been trained using blurred images. Different combinations of results (accuracy) are presented in Table 4.5 in section 4.3.4. All these results have been obtained by using $1C3fc$. Features of unblurred (referred to as Normal)/blurred data have been extracted from last layer of net trained with Normal/blurred data. These extracted features have been used for testing purpose by feeding them as

input to SVM trained on normal/blurred data. The various combinations are listed below:

1. Normal data, I_1C3fc_3LR and SVM trained on normal data (N-N-N).
2. Normal data, $I_{blur}\_1C3fc\_3LR$ and SVM trained on blurred data (N-B-B).
3. Blurred data, I_1C3fc_3LR and SVM trained on normal data (B-N-N).
4. Blurred data, $I_{blur}\_1C3fc\_3LR$ and SVM trained on blurred data (B-B-B).

The results for above combinations have been presented in Table 4.5 in section 4.3.4

4.3 Experiments and Results

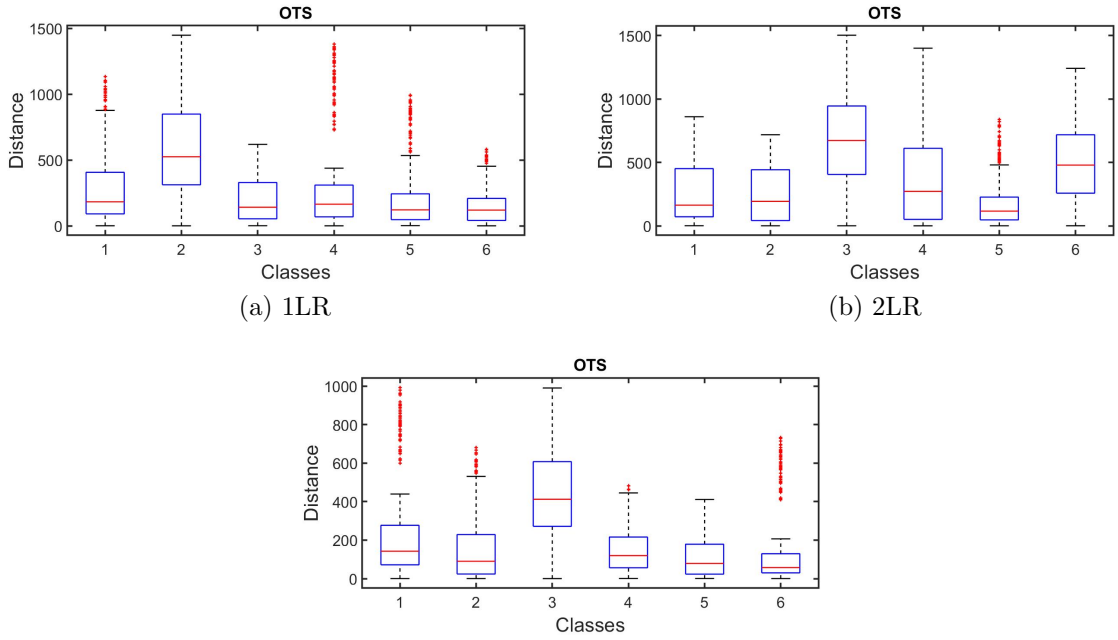
4.3.1 EXP:Learning Rates

As per the description in section 4.2.2.1, experiments on DS1,DS2 and DS3 are performed. Results in Table 4.2 shows that I_1C3fc_3LR NoC provided better results for all the data sets using RGB features. Test results shown in Table 4.2 as well as Tables 4.3 to 4.6 have been obtained using CTS, OTS and PTS with 1200 images of each set. To be at par with results presented in different papers, SVM classification of features extracted from the second last layer of trained nets has been used.

Table 4.2: Accuracy of Different Test Sets with Different NoCs Trained with Different Learning Rates

Model/Datasets	Accuracy		
	CTS	OTS	PTS
I_1C3fc_1LR	79.4	80.0	73.0
I_1C3fc_2LR	80.0	80.0	73.0
I_1C3fc_3LR	83.0	81.0	75.0

Key outcomes of the experiments performed above represent that wile training NoC with 3 different learning rates, I_1C3fc_3LR provided almost (4 ± 2)% and (3 ± 1)% more accuracy when compared to I_1C3fc_1LR and I_1C3fc_2LR respectively. This specifically highlighted that adaptive learning rates (3LR) performed better by dealing with saddle points particularly in the case of fine tuning. Box plots shown in Figure 4.7 have been presented to verify the results that networks trained using 3LR provide better results.



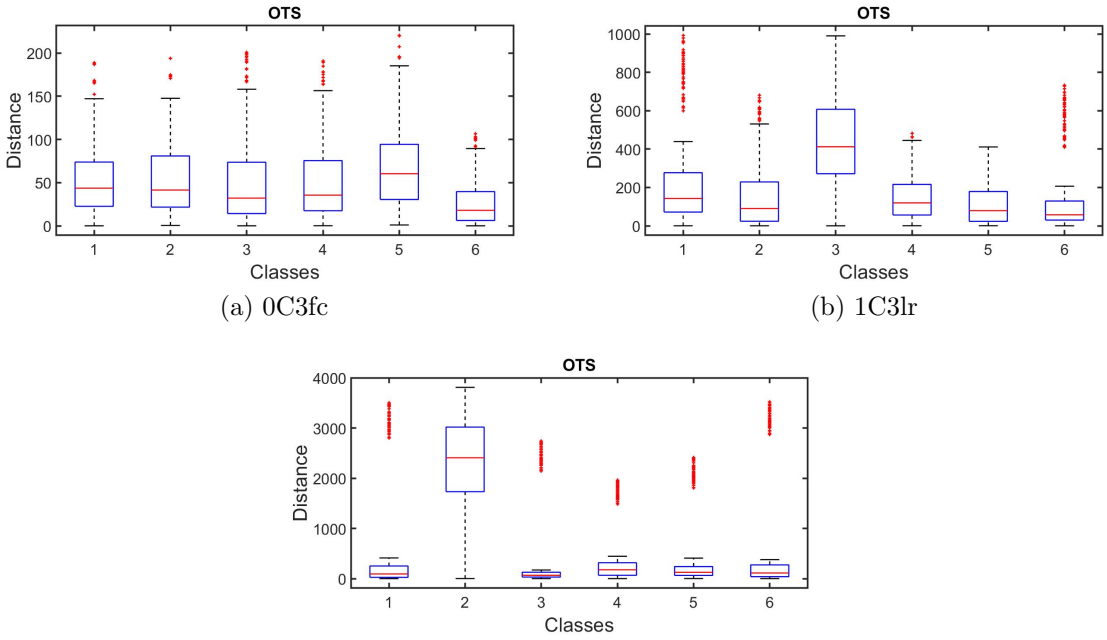
(a) 1LR (b) 2LR

(c) 3LR

Figure 4.7: Box plots developed using network (1C3fc) trained with all the three learning rates. Calculated mean square error of samples from cluster centre in the feature space. It is seen that the mean square error using 3LR is lower than that of 1LR and 2LR

4.3.2 EXP:NOC

Using best performing learning rates, various architectures of NoCs mentioned in section 4.2.2.2 have been trained and tested using RGB features. Results shown in Table 4.3 depict that I_1C3fc_3LR performed (12 ± 3)% more accurately in comparison to I_0C3fc_3LR. The deviation shown is with respect to the results obtained from the datasets CTS, OTS and PTS. Although I_1M1_3LR performed better than I_0C3fc_3LR by (8 ± 3)%, but its performance has been relatively low in comparison to I_1C3fc_3LR by (5 ± 2)% due to small amount of training data. However, addition of more data will require increasing the number of layers in the net which will increase the cost. Hence, I_1C3fc_3LR has been obtained as a suitable solution without increasing data size. Hence, the best performing model is I_1C3fc_3LR which has been verified by analyzing box plots presented in Figure 4.8.



(a) 0C3fc (b) 1C3lr

(c) 1M13lr

Figure 4.8: Box plots developed using various networks architectures trained with 3LR. Calculated mean square error of samples from cluster centre in the feature space. It is seen that the mean square error using 1C3LR is lower than that of 0C3LR and 1M13LR

Table 4.3: Accuracy of Different Test Sets with Different NoCs

Method	Accuracy		
	CTS	OTS	PTS
I_0C3fc_3LR	70.0	65.0	66.0
I_1C3fc_3LR	83.0	81.0	75.0
I_ 1M1_3LR	78.0	76.6	73.8

4.3.3 EXP:Features

While using normal(RGB) data, accuracies obtained from CNN_1C have 12% to 14% rise than obtained from CNN_0C. We also observed that an average of 10% to 12% improvement is obtained while using multimodal features in place of normal features. Out of all the multimodal features mentioned above, scale space features with CNN_1C outperform the others. Accuracies for different network architectures in terms of object detection and classification for normal(F_{Int}), edges(F_{Edges}), scaled (F_{Gauss}) and Optical flow (F_{Opt}) image features are shown in Table 4.4. Results depict that scaled, optical and

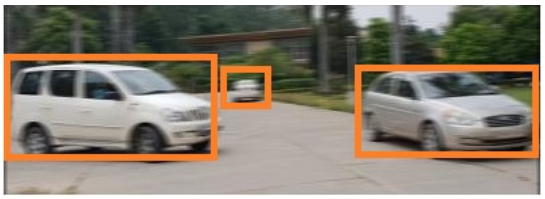
edge image features provide higher detection accuracies compared to raw RGB intensity image features (named as normal). Also, accuracies obtained using CNN_1C are higher than those of CNN_0C. For instance, using intensity data; accuracies obtained for T1, T2 and T3 are 65.0, 64.0 and 63.5 for CNN_0C and accuracies for CTS, OTS and PTS are 79.2, 81.0 and 78.0 for CNN_1C; which is approximately 12-14% rise. It is also observed that an average of 10-12% improvement is obtained while using multimodal features inplace of normal intensity features for CNN_0C. So, using multimodal features is almost as beneficial as adding another convolutional layer.

Table 4.4: Top-1 recognition rate (accuracy) of various networks using different architectures trained with as well as multimodal features

Architectures	**CNN_0C**				**CNN_1C**			
Networks/ Test Sets	**CNN-Int-0C**	**CNN-Edges-0C**	**CNN-Gauss-0C**	**CNN-Opt-0C**	**CNN-Int-1C**	**CNN-Edges-1C**	**CNN-Gauss-1C**	**CNN-Opt-1C**
CTS	70.0	79.4	77.8	77.5	83.0	82.0	81.7	**84.0**
OTS	65.0	78.5	77.0	77.0	81.0	82.0	**83.3**	79.8
PTS	66.5	72.9	74.4	73.0	75.0	78.9	77.4	74.8

Performance of multimodal CNNs with different types of features like edges, gaussian and optical flow has been predented in Figure 4.9. These features perform better as compared to raw intensity images even in the situations when the objects are far away as seen in Figure 4.9a or there is comparative motion in the image or image gets blurred as seen in Figure 4.9b.

(a) Cars detected with yellow represent their detection from networks trained with F_{Gauss},F_{Edges},F_{Opt} and $conv5\_F_Int$ whereas car detected with black represents their detection from networks trained with F_{Gauss},F_{Edges} and F_{Opt} only



(b) Cars detected with orange represent their detection from networks trained with F_{Gauss} and F_{Opt}

Figure 4.9: Object detection in case of scenes containing distant objects and blur

4.3.4 EXP:Blur Network

The results obtained(shown in Table 4.5) clarify that when blurred data has been tested on CNN trained with blur features of RGB data(I_{blur}_$1C3fc$_$3LR$), it provided approximately (12 $\pm$ 2)% better classification accuracy than the case when blurred data has been tested using non-blur i.e. only RGB features (I_$1C3fc$_$3LR$).

Table 4.5: Accuracies of NoCs from Nets Trained with Normal and Blurred data

Test Sets	CTS	OTS	PTS
N-N-N	83.0	81.0	75.0
B-N-N	59.0	69.0	62.0
N-B-B	73.0	70.0	71.6
B-B-B	73.7	74.6	72.0

The losses obtained from networks having 1 and 2 convolutional layers trained with normal and blurred data with different learning rates are given in supplementary data. It has been observed that the training loss converges better for 1LR and 3LR as compared to 2LR for both $1C3fc$ and $1M1$. The training losses and t-SNE plots along with the test accuracies also indicate towards the inference that 3LR with $1C3fc$ is the most suitable among the different options considered here. Test data are represented using t-SNE, i.e. t-distributed Stochastic Neighbor Embedding which is defined as an algorithm for dimensionality reduction and is adapted to visualize high-dimensional data in a scatter plot. The idea is to embed high-dimensional points into 2 or 3 dimensions in a manner that similarities among points are retained. Nearby points in the high-dimensional space correspond to nearby embedded low-dimensional points; and distant points in high-dimensional space correspond to distant embedded low-dimensional points. To show the data distribution graphically, t-SNE for subset of training data is presented in Figure 4.10 and for NoC with different datasets in Figure 4.11. t-SNE for OTS in the case of I_$1C3fc$_$3LR$ gave good and separated clusters for every class. To validate the results, box plot for OTS has been presented in Figure4.12.

4.3.5 Results

Proposed algorithms have been trained on a PC with Xeon Processor with 64 GB RAM, two 8 GB NVIDIA GPU (P 4000) and tested on a CPU with 12 GB RAM and tested with different hyper parametric variations and architectural variations. Metrics like Top-1 accuracy, F1 score and mAP using various datasets have been demonstrated. For all the experiments reported below, seven different image representations have been used, such

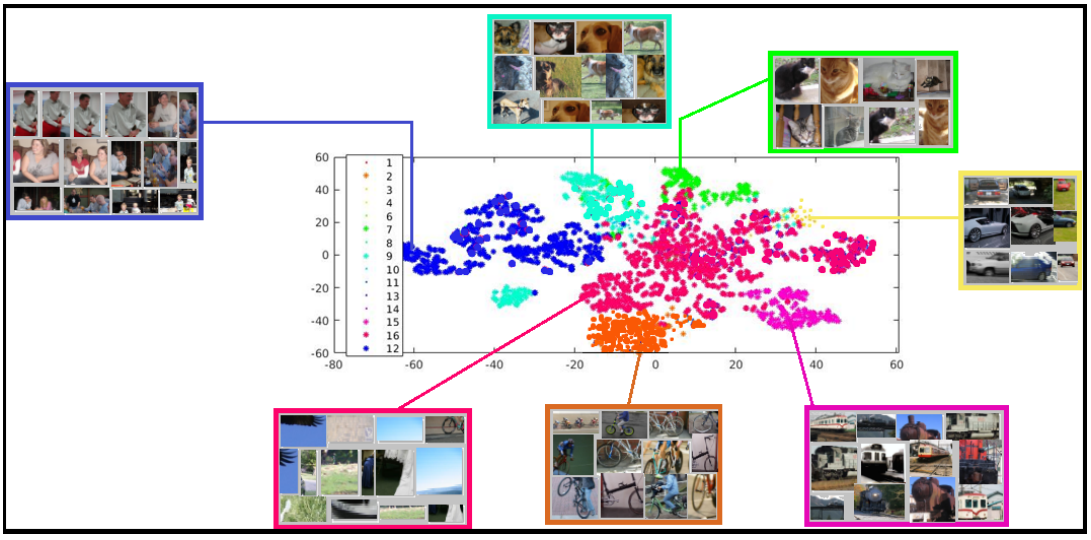


Figure 4.10: t-SNE distribution for the subset of training data extracted from $1C3fc$ with (3LR) shows that the data has been clustered according to different classes.

as: (1)intensity or RGB image, (2) canny edge, (3) sobel edge, (4) prewitt edge, (5) scale space using gaussian filter $3X3$, (6) scale space using gaussian filter $5X5$ and (7) optical flow features. Further fusions of features have been done using addition and maximum operation which results in 4 types of input for training the networks i.e (1) F_{Int} (features of RGB intensity image), (2) F_{Edges} (Fused features of RGB, canny, sobel and prewit) , (3) F_{Gauss} (fused features of RGB, 3X3 and 5X5) and (4) F_{Opt} (fused features of RGB and optical flow). In training phase, network is trained using PASCAL dataset (with these image representation features); which is divided into three subsets($P1, P2, P3$). The 2 different network architectures(CNN_1C and CNN_0C) have been used for training using 4 mentioned image features. Hence, 3 networks for each image representation using each architectures have been trained on different subsets of data ($P1, P2, P3$) individually with 3 learning rates. Using features such as F_{Int}, F_{Edges}, F_{Gauss} and F_{Opt}), 8 trained networks have been obtained namely (1) $CNN - Int - 1C$, (2) $CNN - Int - 0C$, (3) $CNN - Edges - 1C$, (4)$CNN - Edges - 0C$, (5)$CNN - Gauss - 1C$, (6)$CNN - Gauss - 0C$, (7)$CNN - Opt - 1C$ and (8)$CNN - Opt - 0C$.

Further testing with self created dataset is done using these trained nets. During testing, a person is blind folded to realistically simulate the experience of visually impaired people. Testing is done at different spots and timings. Three testsets are taken for testing that are subsets of CTS, OTS and PTS. All the results are presented in the form of Vehicle Detection and Classification using deep neural network and The results have been further presented in the following ways:

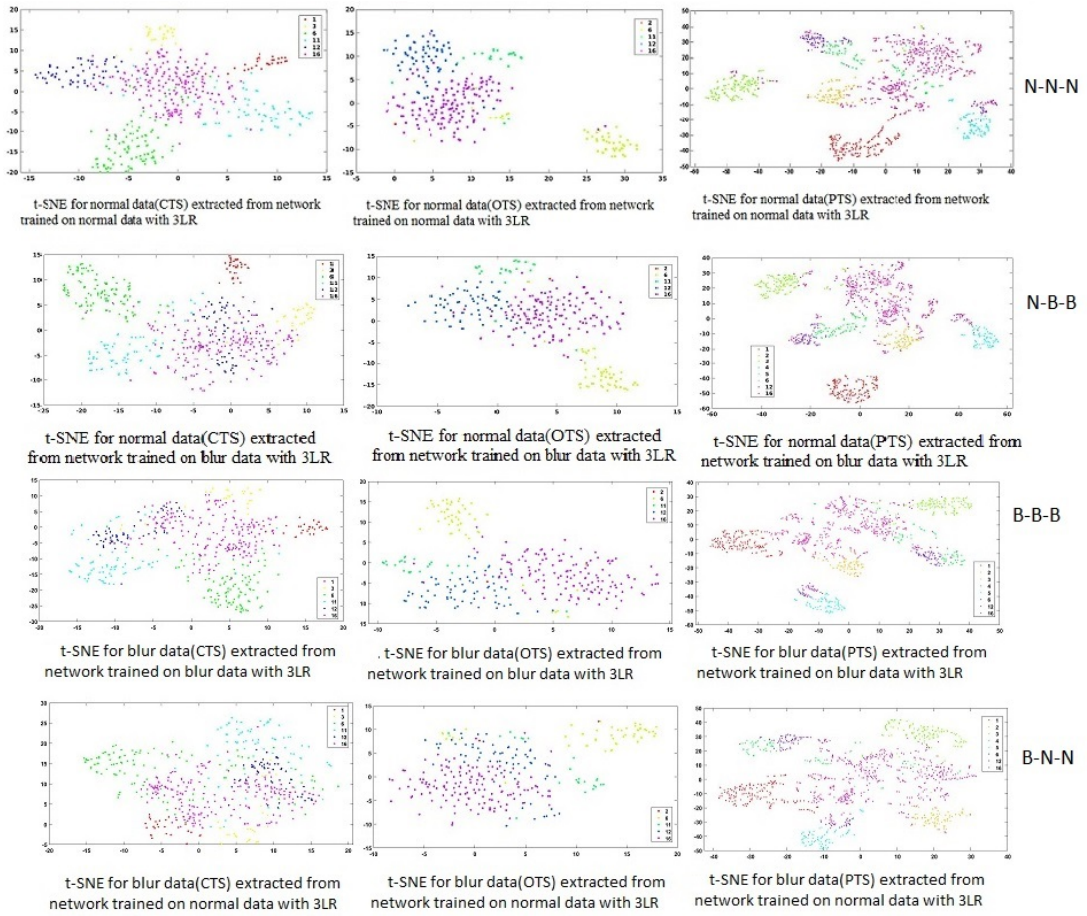
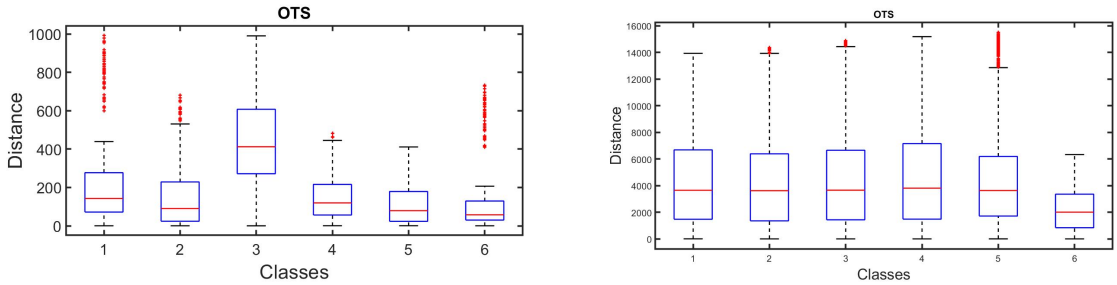


Figure 4.11: t-SNE for all datasets extracted from NoC ($1C3fc$) trained on normal or blur data with 3LR showing the clustered data from trained NoC. Here N-N-N(normal data tested with NoC trained with normal data) and B-B-B(Blurred data test with NoC trained with Blurred data) are giving better results.

Exp(A): Comparison with existing Techniques.
Exp(B): Various Sample sizes chosen for assessing the performance of networks with various metrics.
Exp(C): Mean and Standard Deviation calculated for network weights.
Exp(D): The performance of proposed algorithm has been compared with various benchmark datasets.

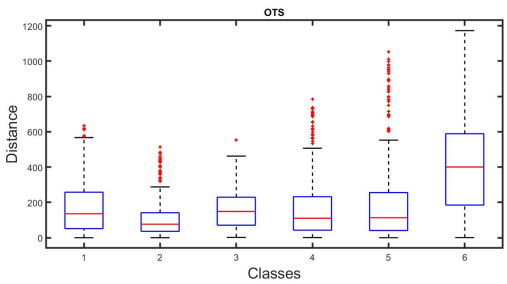
Exp(A): Existing techniques have been tested using own created testsets. The top-1 detection accuracy has been presented in Table 4.6.

Figure 4.13 depicts object detection using R-CNN, fast R-CNN, faster R-CNN, YOLO and NoC(I_1C3fc_3LR). Table 4.6 highlights the results showing accuracy of detection with all these methods which depicts that using NoC on pre-processed data increased the classification accuracy by approximately 5% as compared to NoC without pre-processing

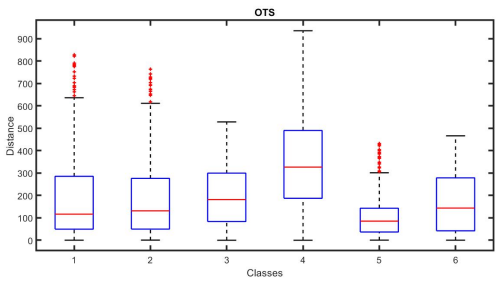


(a) N-N-N

(b) N-B-B



(c) B-B-B



(d) B-N-N

Figure 4.12: Box plots for testset OTS are validating the results. Calculated mean square error of samples from cluster centre in the feature space. It is seen that the mean square error for N-N-N and B-B-B is lower than that of N-B-B and B-N-N

data in all the three datasets. While comparing NoCs with other CNNs, NoCs showed $(20 \pm 5)\%$ higher classification accuracy without pre-processing of data, and provided $(25 \pm 5)\%$ higher classification accuracy with pre-processed data of CTS, OTS and PTS abbreviated as T1, T2 and T3 respectively for further tables and graphs. It has also been observed that while using PTS; YOLO and proposed NoC performed the same without pre-processing. Also, results of faster R-CNN and I_0C3fc_3LR have been almost similar. Hence, addition of a convolutional layer and data pre-processing proved to be quite helpful in improving the classification accuracy.

Table 4.6: Comparison of Accuracies of NoC with Other Object Detection Methods

Method/Test Sets	T1	T2	T3
R-CNN [89]	40.0	54.0	60.8
Fast R-CNN [90]	54.0	61.0	68.7
Faster R-CNN [91]	60.0	66.0	69.9
Yolo [92]	65.0	68.0	74.0
NoC without Pre-processing	**76**	**75**	**74**
NoC with Pre-processing	**83**	**81**	**75**

Exp(B): Figure4.14 shows accuracy of $T2$ with the different methods and highlights that $CNN - Gauss - 1C$ has the highest accuracy for different number of samples per class. Also, unlike other methods, accuracy of optical flow features is considerably higher with even less number of samples per class. Figure 4.15 depicts recall, precision and F1 score

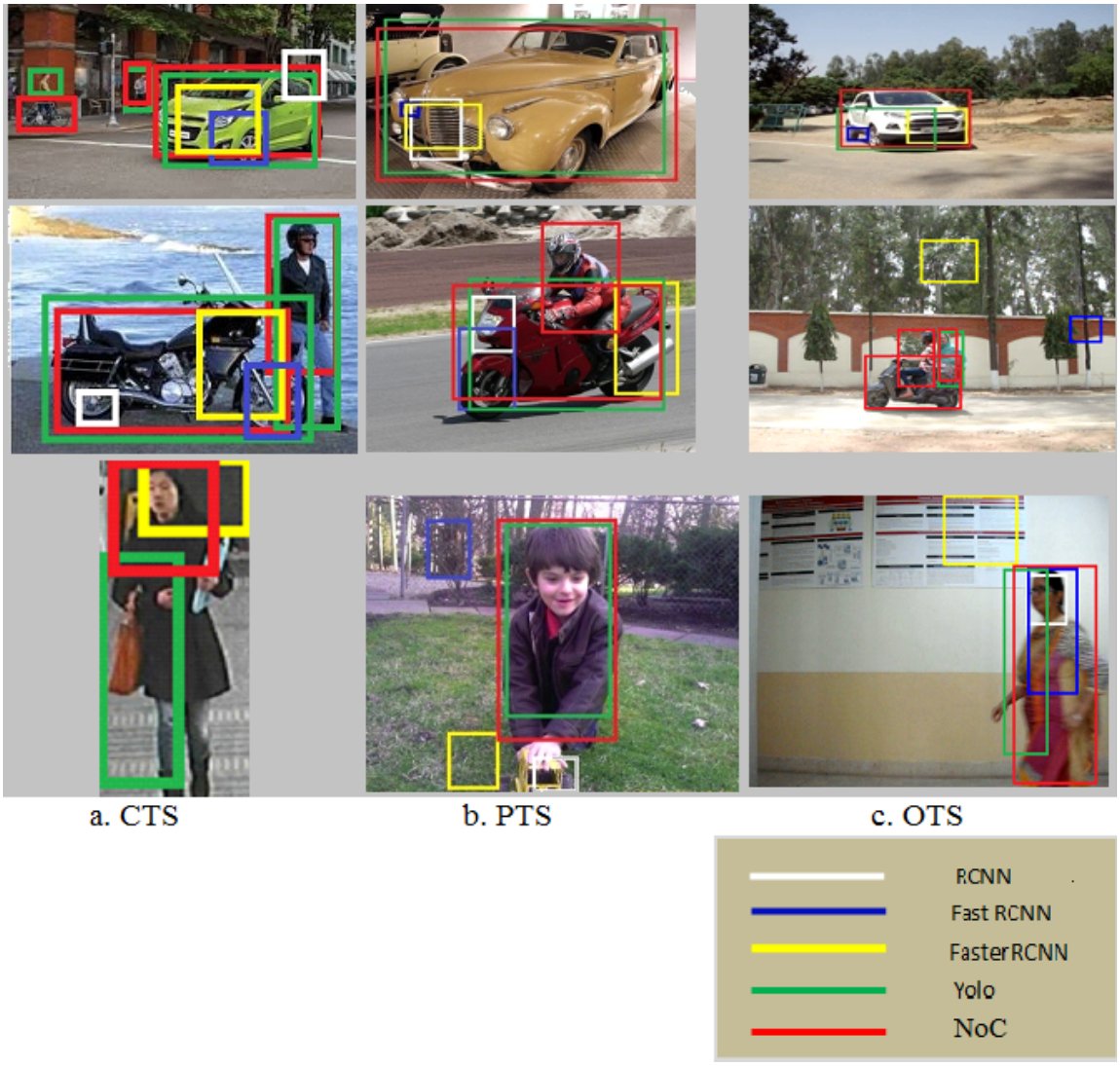


Figure 4.13: Comparison of our NoC with other object detection method

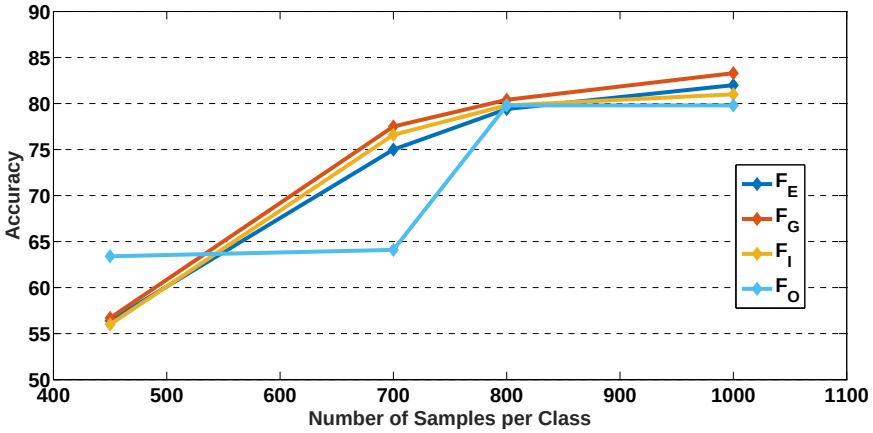
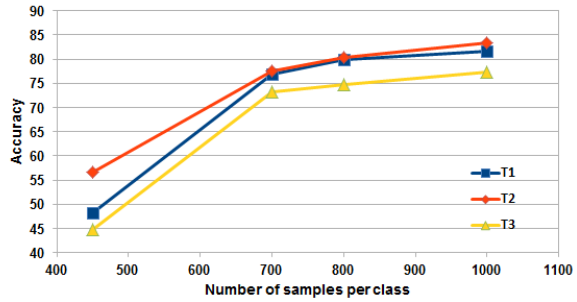
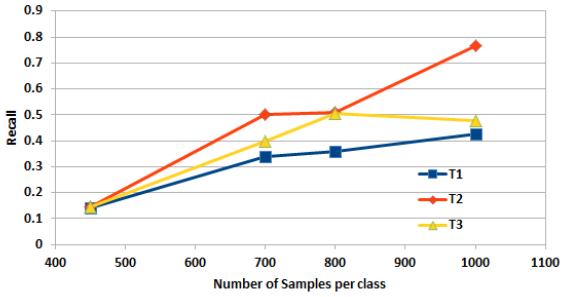


Figure 4.14: Accuracy of T2 testset with all methods

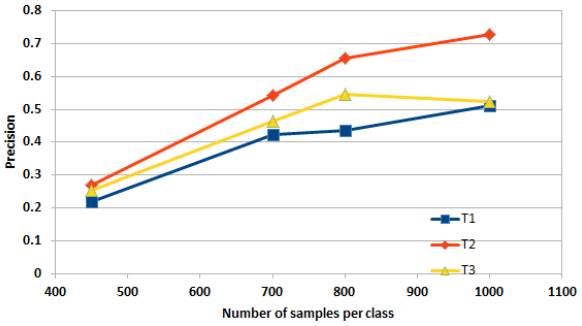
for T1, T2 and T3 using $CNN-Gauss-1C$ with different number of training samples per class. In all the graphs, $CNN-Gauss-1C, CNN-Edges-1C, CNN-Opt-1C$ and $CNN-Int-1C$ have been abbreviated as $F_G, F_E, F_O and F_I$ respectively.



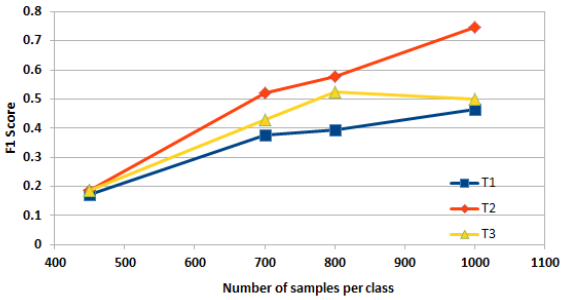
(a) Accuracy of all testsets with $CNN-Gauss-1C$



(b) Recall of all testsets with $CNN-Gauss-1C$



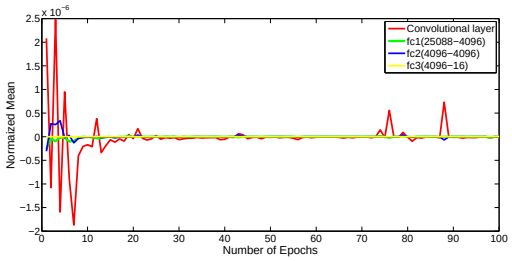
(c) Precision of all testsets with $CNN-Gauss-1C$



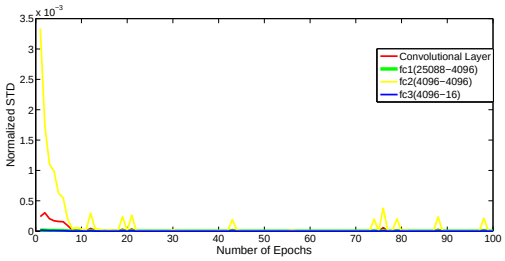
(d) F1 Score of all testsets with $CNN-Gauss-1C$

Figure 4.15: Comparison of testsets based on different metrics

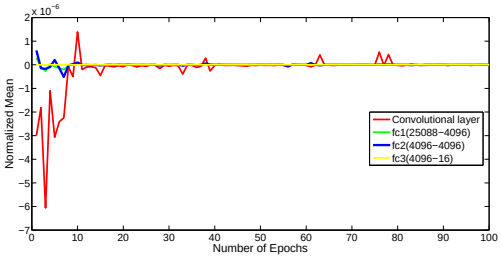
Exp(C): In general, the performance of Deep CNN is measured in terms of detection or classification accuracies obtained using individual networks. Schwartz-Ziv and Tish [263] opened the black box of deep neural networks by using mutual information and, mean and standard deviation of weights learnt across the layers for measuring the performance of networks. According to them over the iteration, they obtained large mean and small STD during the fitting phase where as during compression, there is large fluctuation with small mean and large STD. However the difference between the mean and STD becomes constant to denote network convergence. With these parameters, they recorded layer wise performance to get clear view of results. In this work, Figures 4.16 and 4.17 shows the norm of the means and standard deviations of the weight gradients for each layer of network CNN_1C and CNN_0C respectively as function of the number of training epochs. The values are normalized by the L2 norms of the weights for each layer. It is observed



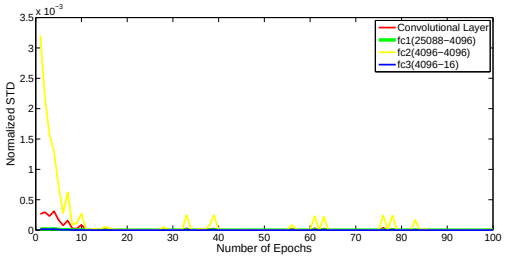
(a) Normalized Mean for sharpened images



(b) Normalized Standard Deviation for sharpened images



(c) Normalized Mean for scaled images

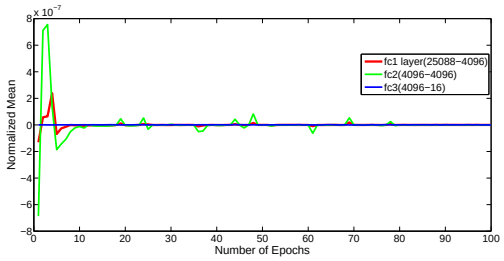


(d) Normalized Standard Deviation for scaled images

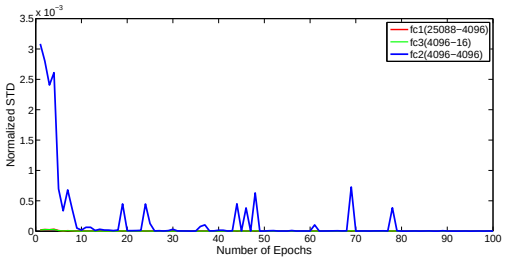
Figure 4.16: The norm of the means and standard deviations of the weights gradients for each layer of network CNN_1C as function of the number of training epochs. The values are normalized by the L2 norms of the weights for each layer.

that for CNN_0C, mean converges in the order fc3,fc1,fc2 while STD converges in the order fc1,fc3,fc2 whereas for CNN_0C, mean converges in the order fc3,fc1,fc2,convolutional layer while STD converges in the order fc1,fc3,convolutional layer,fc2. It should be noted that as the networks are initialized with pre-trained weights, an early convergence is observed. Graphs represent that mean of convolutional layer in network CNN_1C reduce to zero in the end where as standard deviation (STD) of fc2 layer converges in later epochs when compared to other layers. In case of network CNN_0C, mean of fc2 converges in the end and STD of fc3 converges in the last as compared to other layers. Finally the graphs conclude that as par with accuracy, network CNN_1C performs better than CNN_0C in terms of mean and standard deviation also. Fluctuating values of STD of CNN_1C shows that it performs better than CNN_0C.

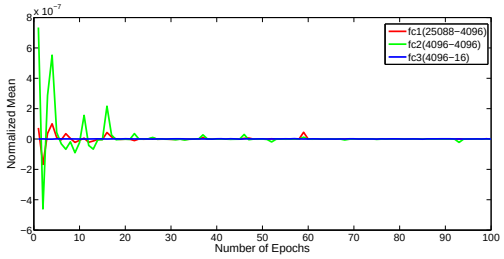
Exp(D): The trained NoC has been tested with various benchmark datasets as mentioned in section 4.2.1. Table 4.7 presents the F1 score, mAP(Mean Average Precision) and Top-1 Accuracy of proposed NoCs with various datasets. It has been concluded that for almost all the data sets multimodal features(I+G) are giving approximately 2 to 4 % better accuracy and also showing better F-score and mAP than intensity features.



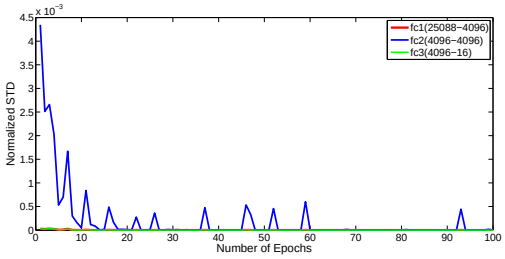
(a) Normalized Mean for sharpened images



(b) Normalized Standard Deviation for sharpened images



(c) Normalized Mean for scaled images



(d) Normalized Standard Deviation for scaled images

Figure 4.17: The norm of the means and standard deviations of the weights gradients for each layer of network CNN_0C, as function of the number of training epochs. The values are normalized by the L2 norms of the weights for each layer

4.3.6 Discussion

In comparison to vehicle classification technology used by [220] which focused on effects of illumination, noise, etc. The current work (1) dealt with illumination using FFT-based technique, (2) considered blur instead of noise, (3) used multimodal NoC instead of GoogleNet (used by them) for feature extraction and classification . While testing proposed algorithm with KITTI test set, an improvement of more than 5% has been observed as compared to algorithm used by [214]. [230] tested their algorithm with PASCAL VOC dataset and observed mAP of 72.4 which gets improved by approximately 2% when the proposed algorithm has been tested with PASCAL VOC dataset. The work undertaken by [218] dealt with noisy and/or blurred images by using multispectral features for the purpose, but considered only pedestrians for classification. However, the current work (1) used multimodal (RGB+OF) features along with blur data for fine tuning NoCs, (2) fine-tuned network for classifying multiple classes, i.e., 16 number of classes in this case. By the use of multiple classes, the number of false alarms have been reduced which helped in classifying more number of objects. Contrary to previous research contributions which considered weather conditions [220] [227], night time environment [226] [224] [221] [227]; the present work captured images during different times of day and on different

days. Hence, the proposal implicitly comprises features of images captured during different weather conditions. However, the performance for each weather type has been not explicitly analyzed. Also, night time environment is yet to be considered. [228] and [220] considered fine tuning-based object detection where VGG16 and GoogleNet have been used with fixed learning rate. This work differs from theirs in 3 ways: (1) In the present work, per region classifier architecture for fine tuning has been used. (2) Three types of learning rate including adaptive learning rate to deal with saddle points have been used. (3) When proposed algorithm has been used for testing PASCAL VOC test set, mAP has been improved by almost 10% in comparison to the value of mAP shown by them for PASCAL VOC. [216] and [229] used related frame information from the consecutive frames which is planned to be incorporated in future work. However, the former work extracted the features using handcrafted techniques in contrast to NoC used in the current work. Moreover, these researchers did not consider the blurred data.

Table 4.7: Comparison of NoC while using various test sets

Parameters	**Top-1**					**F1 Score**					**mAP**				
Dataset/ Methods	I	I_{blur}	$I+O$	$I+E$	$I+G$	I	I_{blur}	$I+O$	$I+E$	$I+G$	I	I_{blur}	$I+O$	$I+E$	$I+G$
Apollo	81.25	80.26	82.89	81.00	83.10	0.7636	0.7689	0.7795	0.7777	0.7810	79.73	83.79	85.01	83.45	85.50
Berkeley	94.73	76.30	96.01	78.94	95.50	0.9439	0.7690	0.9692	0.7873	0.9519	97.36	81.95	98.56	88.45	91.25
CityScapes	88.15	80.26	89.47	81.50	90.15	0.8901	0.8304	0.8818	0.8591	0.8859	89.08	85.84	90.13	91.19	90.91
KITTI	77.63	53.94	80.26	67.10	81.89	0.8022	0.5800	0.8419	0.6700	0.8510	76.92	65.96	83.81	74.23	84.09
Caltech	83.00	73.70	84.00	82.00	81.70	0.8385	0.7524	0.8512	0.8128	0.8635	87.46	80.72	89.68	83.93	90.17
OTS	81.00	74.60	79.80	82.00	83.30	0.8377	0.7647	0.8111	0.7724	0.8429	87.74	81.99	82.79	80.06	88.51
PASCAL	75.00	72.00	74.80	78.90	77.40	0.7567	0.7312	0.7426	0.7288	0.7616	77.79	74.43	74.40	71.81	78.16

4.4 Conclusion

In this chapter, data collection and sampling tricks prior to training have been discussed. Extensive experiments have been performed on different convolutional classification architectures with various learning rates. The results showed that the performance of I_1C3fc_3LR has been relatively better. Blurred data has also been used to train these NoCs. It has been observed that $I_{blur}\_1C3fc\_3LR$ can be used for blurred as well as unblurred data, whereas I_1C3fc_3LR fails to tackle blurred data. Furthermore, multimodal features computed for training normal as well as blurred NoCs proved to be beneficial.

Chapter 5

Case Studies: To develop intelligent vehicle navigational capabilities using the augmented maps

The multimodal fusion based NOC classifier for feature extraction, 2-step scene perception & localization, map generation can be used for developing many vision based intelligent capabilities. We present a few sample applications for which related case studies have been performed.

5.1 Modules' Description

For any typical application the 3 modules are required a.Map based scene-localization b. Object detection c. Density estimation.

5.1.1 Map based Scene localization module

Map generation using mobile trajectory data has been discussed in chapter 2. Scene localization has been discussed in detail in chapter 3 in which a 2-step approach has been used. Zone detection using places CNN and set-based image classification has been implemented in section 3.2.1. Further landmark detection has been implemented by fine tuning capsule network in section 3.2.2. A single background class typically used in verification cases is replaced by GAN based dustbin classes for greater confidence of network for known classes and lower confidence for unknown classes as shown in section 3.3.

5.1.2 Object detection and classification module

Object detection and classification has been implemented using depp CNN features in chapter 4. In this, after dataset preparation, extensive experiments have been performed on different convolutional classification architectures($0C3fc, 1C3fc, 1M1$) with various learning rates ($1LR, 2LR, 3LR$) as detailed in section 4.2. Also, multimodal features have been used to train the NoC (Network on convolutional features) as discussed in section 4.2.2.3. The rest of the sample studies shown have been performed using best performing model $CNN - Gauss - 1C$ trained on $1C3fc$ with $3LR$.

5.1.3 Density estimation module

Density estimation has been implemented using feature extraction from object detection module. A class of augmented map application has been introduced which can provide detailed knowledge about any area, to a user. This brief particularly focuses on obtaining itinerary perception subject to different environmental conditions. This refers to extraction of traffic related information from an augmented map. The problem is modelled as a machine learning technique where the traffic distribution at different times (including same days, different days and different weather) are observed continuously using a service robot. This data is posed as a Gaussian process for post-estimation. Our system consists of a vision sensor which will acquire the region of interest input, queried to a database of traffic density distributions, learned from the scenes at different points of time. The user interacting with the system will obtain an information pertaining to the region conditioned on environmental and timing events.

The system can be depicted as shown in equation 5.1

$$\begin{aligned} \partial &= f(\mathbf{X}, c, d, t) + \epsilon, \text{ where} \\ \epsilon &\sim N(0, \Sigma_\epsilon) \end{aligned} \tag{5.1}$$

$\mathbf{X}$ in this work encompasses the visual features obtained from the continuous scene capture using vision sensor. These are represented as treelet decomposition features. Any other descriptors like Curvelets[264], HOG-HOF of STIP[265] or 3D corners[266] can be effectively used as well. Extracting these features for detected objects via CNN in a scene instead of the whole scene acts as a preliminary filter to pick objects of interest only. Alternative the scene could be segmented into individual components [267] and segments of interests could be analysed. Further climatic, day and time representations c, d, t being intractable are computed as a function $\phi(\mathbf{X})$ of the input space imaging features. Each

image I is represented with its treelet decomposition features $\mathbf{x}$, using set of patch data I_t [268]. The computation is further depicted as in equation 5.2 and 5.3

$$\mathbf{x} = sgn(W^\top \mathbf{y} - \bar{\mathbf{y}}) \tag{5.2}$$

where W is the top level basis matrix and $\mathbf{y}$ is computed as

$$\mathbf{y} = \sum_{i=1}^{p=L} s_{L,i} V_{L,i} + \sum_{i=1}^{L} d_i \mathbf{w}_i \tag{5.3}$$

Here s_L and d_L are the sum and difference variables where as V_L and $\mathbf{w}_L$ are the scaling and detail functions of basis matrix. It is assumed that I_t for each independent observations of the same ROI follows $P(\mathbf{x}^{o_1} \neq \mathbf{x}^{o_2}) < \epsilon$, where o_1 and o_2 are two observations for the same region. The feature space $\mathbf{x}$ is projected to $\phi(\mathbf{x})$ using c, d, t information as given in equation 5.4

$$\begin{aligned} f(\mathbf{x}) &= \phi(\mathbf{x}).w, \text{where} \\ \phi(\mathbf{x}) &= \frac{e^{\alpha \mathbf{x}} - 1}{\alpha} + \alpha \end{aligned} \tag{5.4}$$

In the above equation $\alpha = c, d, t$. Estimating the density for any conditioned distribution is a complex problem when $\mathbf{X}$ is high dimensional. In order to handle missing data as well as outliers the density estimation is viewed as a function approximation problem.

Distribution Model Generation from Augmented Map The problem is modeled as gaussian process as shown in equation 5.5 in which the mean is given in equation 5.6 and covariance is represented in equation 5.7. The task is to estimate the density of a test region given the current environmental information and past experiences. Inference is thus drawn in two steps which include computation of the distribution corresponding to any test sequence.

$$f(\phi(\mathbf{x})) \sim GP(m(\phi(\mathbf{x})), k(\phi(\mathbf{x}), \phi(\mathbf{x}'))) \tag{5.5}$$

$$m(\phi(\mathbf{x})) = E(f(\phi(\mathbf{x}))) \tag{5.6}$$

$$k(\phi(\mathbf{x}), \phi(\mathbf{x}')) = E[(f(\phi(\mathbf{x})) - m(\phi(\mathbf{x})))(f(\phi(\mathbf{x}')) - m(\phi(\mathbf{x})))] \tag{5.7}$$

$\phi(\mathbf{x})$ and $\phi(\mathbf{x}_*)$ are henceforth referred to as ϕ and ϕ_* for simplicity. Predictions for

Gaussian Process regression are shown in equations 5.8, 5.9 and 5.10.

$$f_*|\phi, \partial, \phi_* \sim N(\bar{f}_*, cov(f_*)), \tag{5.8}$$

where mean in the above equation is represented by $\bar{f}_*$ as shown in equation 5.8 and covariance is given in equation 5.10.

$$\begin{aligned} \bar{f}_* &\triangleq E[f_*|\phi, \partial, \phi_*] \\ &= K(\phi_*, \phi)[K(\phi, \phi) + \sigma_n^2 I]^{-1}\partial, \end{aligned} \tag{5.9}$$

$$cov(f_*) = K(\phi_*, \phi_*) - K(\phi_*, \phi)[K(\phi, \phi) + \sigma_n^2 I]^{-1}K(\phi, \phi_*) \tag{5.10}$$

The linear preliminary policy can be defined as shown in equation 5.11 when the state $\mathbf{x}$ is gaussian distributed.

$$\partial(\mathbf{x}_*) = w\phi_* \tag{5.11}$$

where w is the parameter matrix. This can be extended for the nonlinear case as represented in equation 5.12

$$\begin{aligned} \partial(\mathbf{x}_*) &= \sum_{i-1}^{N} k(m_i, \phi_*)(\phi^\top \Sigma_p \phi + \sigma_\partial^2 I)^{-1}\partial \\ &= k(M, \phi_*)^\top \alpha \end{aligned} \tag{5.12}$$

where $a = 1, 2, 3,\ M = [m_1 ... m_N]$ are the centres of the Gaussian basis functions, Σ_p is covariance and $\alpha = (\phi^\top \Sigma_p \phi + 0.001I)^{-1}\partial$ The predictive mean of $\partial(\mathbf{x}_*)$ can be obtained as shown in equation 5.13

$$\begin{aligned} E[\partial(\mathbf{x}_*)] &= \alpha_a^\top E_{\phi_*}[k(M, \phi_*)] \\ &= \alpha_a^\top \int k(M, \phi_*)p(\phi_*)d(\phi_*) \\ &= \alpha_a^\top r_a, \end{aligned} \tag{5.13}$$

The predictive covariance for a, b=1,2,3 is depicted as given in equation 5.14

$$\begin{aligned} cov_{\mathbf{x}_*}[\partial_a(\mathbf{x}_*), \partial_b(\mathbf{x}_*)] &= E_{\phi_*}[\partial_a(\phi_*), \partial_b(\phi_*)] \\ &- E_{\phi_*}[\partial_a(\phi_*)]E_{\phi_*}[\partial_b(\phi_*)] \end{aligned} \tag{5.14}$$

The model used here is referred to as a deterministic gaussian process equivalent to a

regularized Radial Basis Function (RBF) network. The choice of covariance function dictates the motion of learning for a supervised model. The most common covariance function used for any machine learning problem is the square exponential kernel.
The current model prefers the use of matern class kernel to avoid the strong smoothness assumptions of the former. The covariance function is represented in following equations,

$$K(\phi, \phi') = \begin{cases} \sigma_{ab} M(h|v, \beta), & if a = b, \\ \rho_{ab}\sigma_{ab} M(h|v, \beta), & otherwise. \end{cases} \tag{5.15}$$

where

$$M(h|v, \beta) = \frac{2^{1-v}}{\Gamma(v)} (\beta \parallel h \parallel)^v M_v(\beta \parallel h \parallel) \tag{5.16}$$

v=3/2, 5/2 are the most preferred parameter choices for machine learning. In this $\parallel h \parallel$ denotes the Euclidean distance between ϕ and ϕ' The loss function which specifies the penalty for guessing a value is considered as

$$\Pi(\partial_*|\mathbf{x}_*) \quad \varepsilon \quad E[c_{f_*}] \tag{5.17}$$

where c_{f_*} is the cost of guessing ∂_*. We minimize this loss by averaging over the models as shown in equation 5.18 [269] .

$$\begin{aligned} E_{f_*}[c(f_*)] &= \int c(f_*) p(f_*) df_* \\ &= 1 - \int \exp(-\frac{1}{2}(f_* - f'_*)^\top T^{-1}(f_* - f'_*)) p(f_*) d(f_*) \end{aligned} \tag{5.18}$$

where T^{-1} is the precision matrix of the unnormalized Gaussian. Case studies related to these 3 modules have been presetend below:

5.2 Case Study 1: Augmented map based Assistive device for Visually Impaired

5.2.1 Problem

Navigation is of paramount concern for visually impaired people. A number of devices are developed for visually impaired people to provide them information about presence

of obstacles, types of obstacle, their distances etc. This information is further utilized to assist visually impaired people for navigating safely, both indoors and outdoors. Carrato et al.[270] have focused on detecting and recognizing people and their facial expression for assisting visually impaired people. Ruxandra et al.[271] proposed a distance-based navigation system, categorising the object as urgent or normal, depending on its distance from the user. Aravinda et al.[272] used vision based system along with Laser patterns for detecting potholes and uneven Surfaces. Kanwal et al.[273] provides information about wall-like obstacles, using the Kinect both as a camera and a depth estimator. Aladren et al.[274] proposed visual and range information-based Navigation assistance system for visually impaired. They used a consumer RGB-D camera, and took advantage of both range and visual information about floor, walls and obstacle for indoor environment. Their system gives voice commands about the obstacle to the user. Sarfraz and Rizvi [275] developed navigation assistance for indoor environment providing depth and object type information including presence of humans, doors, hallway or corridors, staircases, elevators, moving objects,etc. They used camera vision input and text-to-speech synthesized output to provide navigation aid.

Jafri et al. [276] presented an indoor navigation system based on visual and IR sensor data. They used tango tablet development kit for creating a 3-D reconstruction of the surrounding environment and associated a user with the Unity collider component and utilized it to check user's interaction with the reconstructed mesh for detection of obstacles. Message was conveyed to the user through voice alerts and beep patterns. Jiang et al. [277] proposed a wearable binocular system for object detection using a convolutional neural network on high quality images only. Li et al. [278] proposed a real-time holistic vision-based system called ISANA for blind navigation and wayfinding. ISANA runs on google tango mobile device containing vision and depth sensors. Maps were generated with the help of CAD model from different architectural layouts. They used text-to-audio on priority basis to convey feedback of system such as waypoint guidance, obstacle alerts and location awareness information.

Besides the development of different navigational aid systems for the visually impaired, there is also an ongoing research about suitability and acceptability of these systems. For example, it is argued [279] that there are many problems associated with these devices, such as (i)being invasive i.e. covering ears, blocking the tongue, requiring use of hands etc.,thus obstructing full range of body motion and function for the users. These devices are also cumbersome. (ii)Increasing the users' cognitive load as these devices may require lot of attention, causing distraction from the primary task. (iii)These devices require lot of training for their usage, which is difficult especially for children and those unfamiliar to technology. (iv) Cost is generally high. Affordability, serviceability and maintenance

is not easy. (v)Lastly, many of these devices are still in the early stages of development or are being tested at pre-clinical levels only.

5.2.2 System

In this work we use our augmented maps to facilitate safe navigation of visually impaired people. The user will have access to a scene perception system providing information about objects(object detection and classification from images) in the scene and their relative depth(using laser) for any unknown scene. For known scenes, additionally, the user can also know where he is via the scene localization component, get directions to a destination via the topographical map. The augmented model is used on an Odroid board making this system low-cost, light weight, simple and easily wearable emphasising that no explicit training is required to use the system. At the same time, retrieval of maximum information about the environment is ensured. The system currently works on restricted voice output which implies that the user can select how much information he wants. The system provides functionality such that he can choose to be informed about scene changes at (a)fixed intervals or (b)a single time only (c)when obstacles are too close.
The flow chart of system developed is shown in Figure 5.1. This system is capable of fulfilling these requirements:

1. Acquisition of a video stream and laser data from webcam (HD resolution 1920x1080, 25 fps) and laser device respectively.

2. Detection and classification of multiple objects from the scene, even if the position of object or vehicle is not perfectly in front of the camera via the object detection and classification module.

3. Mapped distance estimation of individual objects via the laser distance module.

4. Recognition of the surrounding environment via scene localization module.

5. Generation of a voice output telling the name of the recognized object or vehicle, its distance from the user and information about surrounding.

The proposed system is portable however the size will be reduced in future versions.

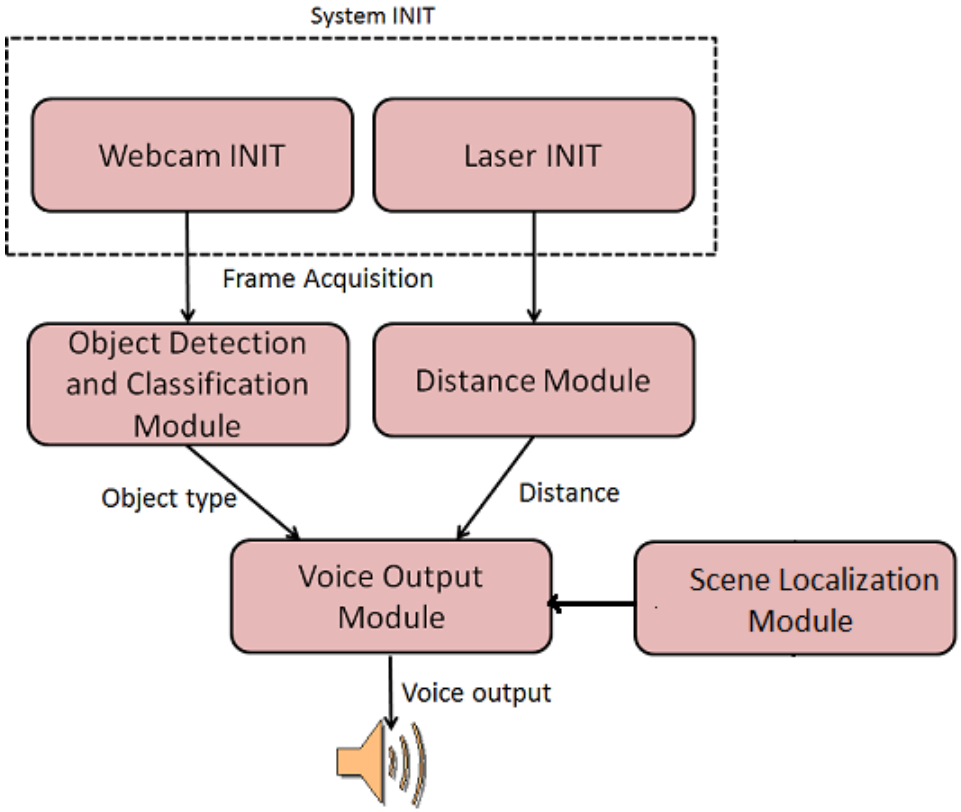


Figure 5.1: Flow chart for the proposed scheme

The system is compared with some existing systems in Table 5.1.

Table 5.1: Comparison of proposed system with other systems based on various properties

System	Light weight	CNN	Multimodal	Fast	Feedback to User	Outdoor environment	Non-vexatious
Tango tablet development kit [276]	×	×	×	√	√	×	×
Kinect based system [280]	×	×	×	√	√	√	×
Binocular vision sensor [277]	×	×	×	×	×	√	×
White cane tango mobile device [278]	√	×	×	√	√	×	×
ALICE [271]	√	×	×		√	√	√
Assistance system [272]	×	×	×	×	×	√	×
Assistance system [274]	–	×	×	×	√	×	–
Proposed System	√	√	√	√	√	√	√

5.2.3 Sample Examples

In the proposed system, the best performing network i.e. $CNN - Gauss - 1C$(CNN having 1 convolutional and 3 fully connected layers trained with multimodal features including RGB and gaussian features for object detection and classification) has been installed. The interaction of the system with the user is through voice messages. It informs user about the type of obstacle, its distance and information about the scene

Table 5.2: Time taken by the network ($CNN - Gauss - 1C$) to recognize the object and by espeak module for voice message.

Scenarios	Detected Object	Distance	Detection Time		Message Time	
			In 1st change	After 1st change	In 1st change	After 1st change
1	Person	700 mm	10.0 ms	10.0 ms	7.0 ms	2.5 ms
2	Bus	100 mm	9.0 ms	9.0 ms	6.5 ms	1.8 ms
3	Person	1000 mm	10.0 ms	10.0 ms	7.1 ms	2.5 ms
	Bike	950 mm	9.5 ms	9.5 ms	6.8 ms	2.0 ms
	Car	1500 mm	9.7 ms	–	6.7 ms	–
	Person	3000 mm	9.9 ms	–	7 ms	–
	Bike	2500 mm	9.5 ms	–	6.9 ms	–

as shown in Figure 5.2. The voice message is given through ear phones. However, large number of voice messages can confuse the user. In that regard, vibrations or some beep systems can also be incorporated for better clarity. For testing, the proposed system was repeatedly provided to user groups and then, according to the feedback, some changes were done in the system. Initially, the system was detecting and classifying all objects in all the frames acquired by the camera. Their information, along with their distance from the user, was conveyed with the help of voice messages. However for overcrowded scenes, there were lot of voice messages generated per frame which were taking lot of time to convey the messages, thereby confusing the user.

Certain changes were done after getting feedback, as shown in Figure 5.3 which includes setting of frame rate, information of object according to its distance and classification confidence of object. For details, refer our paper [281].

Table 5.2 shows different scenarios along with time taken by the system to recognize and inform the user. " − " in the table represents that no information about these objects was given in 2^{nd} change.

(a) Sample scene identified by scene localizer as Zone2

(b) Sample scene identified by scene localizer as Zone3

(c) Sample scene identified by scene localizer as Zone3

(d) Sample scene identified by scene localizer as Zone1

(e) Sample scene identified by scene localizer as Unknown scenes

(f) Sample scene identified by scene localizer as Unknown scenes

Figure 5.2: Results of object detection and classification along with their distance from the user

Figure 5.3: Changes according to the feedback of users

5.3 Case Study2: Density estimation using gaussian model

5.3.1 Problem

In this application, gaussian model has been used for estimating density at a particular place or area [282]. While navigating in university campus or any indoor areas, this model can help the person to navigate from one place to another through a less or non traffic(rushy) area in order to save time. Augmented maps have been generated for the density estimation including traffic and persons. Generation of Augmented Maps has evolved in the last few years in tandem to the development of Augmented Reality devices required for interfacing. These maps presents a representation framework which is much more comprehensive to the users. Some notable research work on Augmented Maps presented in the last few years include augmented paper maps with GIS, cartographic maps, transit maps and augmented areal earth maps. Number of researchers [283–286] have worked on density estimation using deep CNN features from networks such as Alexnet, VGG, GoogleNet, FCNN etc with variety of datasets. However all of them have covered persons only not vehicles or other objects.

5.3.2 System

The main objective of this model is to estimate the density ∂ of traffic conditioned on the gathered data and current weather, timing information.The term traffic here refers to vehicle and human traffic. In short, the user will have a general understanding of the different sectors under vigilance to an extent that clarifies which regions are more occupied by pedestrians or vehicles at particular instants. This will allow the user to

choose less crowded or more crowded paths for moving from source to destination based on his requirements. For example, at late night user will prefer a relatively crowded path for safety purposes. Figure 5.4 depicts the training procedure of the system. Each image post Objects of interest detection, undergoes treelet feature extraction and density estimation . The former is squashed using the time, climate and day parameters to represent the training input. Hence the same region(same treelet features) with different parametric conditions will result in different squashing output. The database thus has an entry of density distribution pertaining to each region at different parametric conditions.

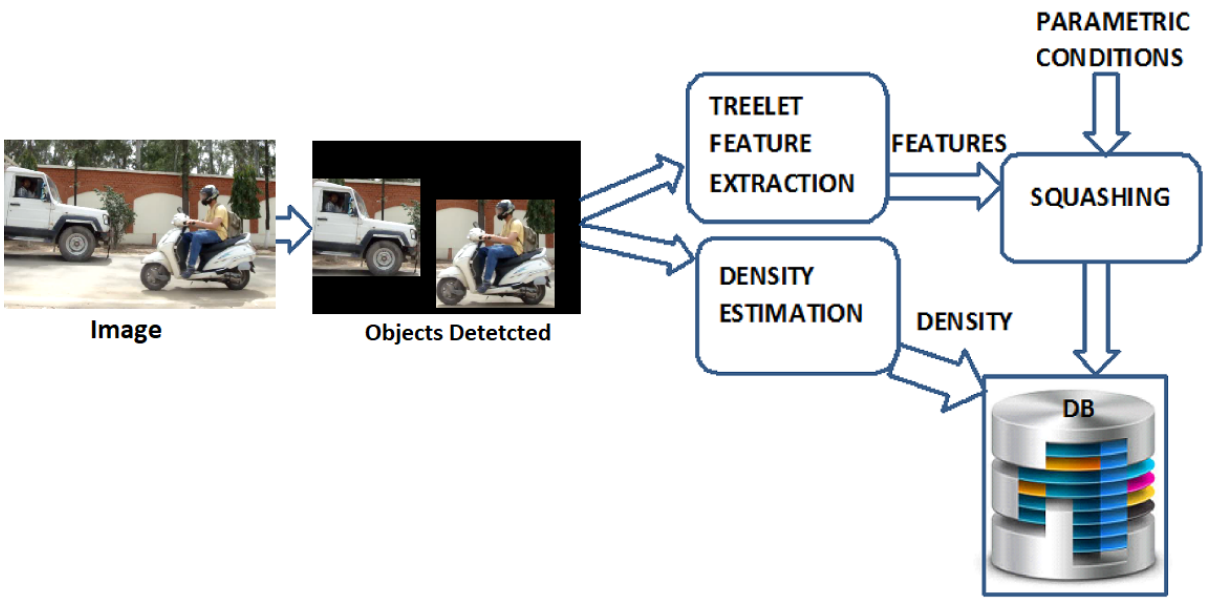


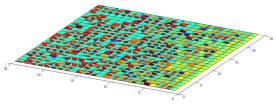
Figure 5.4: Generation of training database

5.3.3 Sample Examples

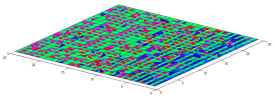
Figure 5.5 demonstrates the user experience with the system. User provide the regions in the form of images as input to the system. He intends to know the density distribution of that particular region at 11:15 am and 3:00 pm of a particular day in cold weather, Henceforth referred to as first and second scenario.

In this work, currently 23040 input-output pairs have been used. Of these 21000 were used as training set and 2040 were used for testing. This data has been collected by continuous vigilance of a region for 3 months in which six time slots have been taken in a day for seven days a week during different weather conditions. The time slots are taken

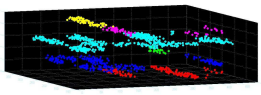
(a) Image as input by the user



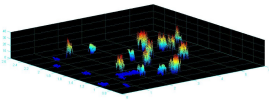
(b) Squashed features for first scenario



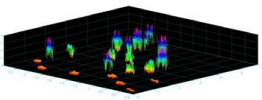
(c) Squashed features for second scenario



(d) Database



(e) Density distribution of first scenario



(f) Density distribution of second scenario

Figure 5.5: **Testing the Database** (Figure 5.5a is given by the user as input for an estimation of its density distribution during the second and third slot. The squashed features $\phi(\mathbf{x})$ for both scenarios are shown in figure 5.5b and 5.5c, Figure 5.5d demonstrates the database which is used to obtain the density distribution. x and y axes in the database represent the area while the distributions for different sequences are spanned across the z axis. The estimated density distribution for the input sequences are shown in Figure 5.5e and 5.5f in which x and y axes represent area and z axis shows the density range.)

as 07:00 to 10:00 am, 10:00 am to 1:00 pm and so on. The performance of the system is evaluated using negative log probability of the prediction model as shown in equation 5.19 [287]. A standardized mean loss of 0.009 is obtained.

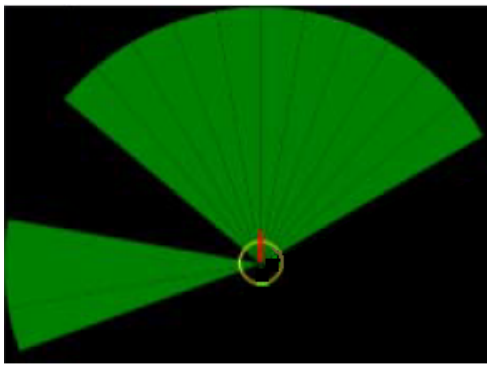
$$-logp(\partial_*|D\phi_*) = \frac{1}{2}log(2\pi\sigma_*^2) + \frac{(\partial_* - \overline{f}(\phi_*))^2}{2\sigma_*^2} \tag{5.19}$$

This work explores the possibility of building a system which can reflect the role of a human in observation and inferential capabilities. The model is yet to undergo huge amount of iterative updation for extending the inferential domain using the same database. For example common group of people interacting with each other, their behavior. This may also require implementation of estimation techniques other than gaussian process. The research on the whole has extensive application capabilities and utilities.

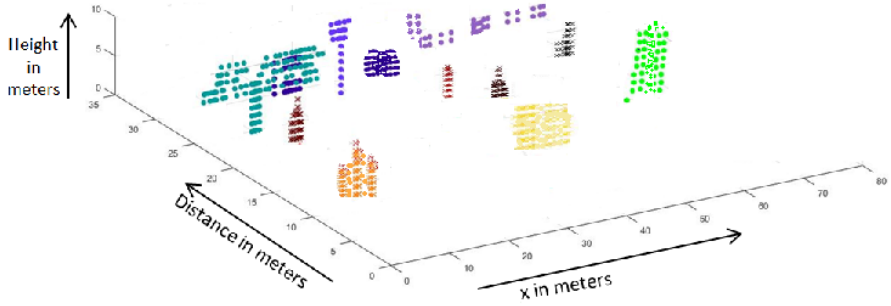
5.3.3.1 Validation of trend density with real-time density

Trend model has been generated using above density estimation model. It represents the trend of traffic density i.e. amount of traffic usually found at particular place during particular time. The system will check through this model for the trend map after scene segmentation and density estimation. The scene captured, its ultrasonic view and features are shown in Figure 5.6. During run time when the robot is navigated, it is given start

(a) panoramic view captured by the camera of robot



(b) Ultrasonic output of the scene



(c) Feature output of the scene along with distance of objects

Figure 5.6: Density estimation of scene along with distance

and end point. There can be multiple routes between start and end point. It checks the trend for traffic density according to time, day and weather condition; choose the route from start to end point with less density and follow the same. Further it captures the scene on its path; detect and classify objects from the scene; calculate traffic density in that real time scene using classified objects and change or follow the route accordingly. The process is shown in Figure 5.7.

Results of the proposed work are presented in the form of (1)Scene segmentation and classification using deep neural network. (2) Augmented map for robot navigation giving the traffic density on a particular scene with the distance of vehicle. 84 videos are captured at 3 time slots of 4 different locations for a week which are used to set the trend about the traffic scenario of the particular road or area. Out of these videos, 3400 vehicles including cars, trucks, pedestrians etc are classified and saved in different folders with the help of trained classifier. After training and testing the model generated 98 % accuracy for the recognition of various objects. Some of the instances of the results for estimating traffic density are shown in Figure 5.8

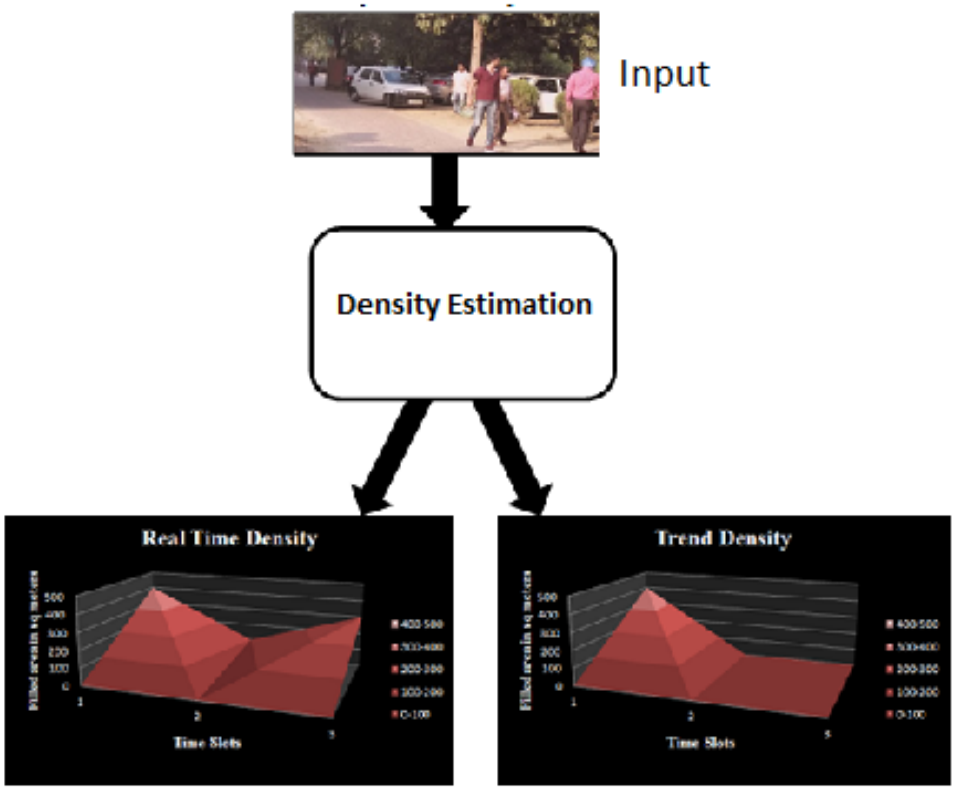


Figure 5.7: Navigation using trend(Real time and trend density estimation. Real time density map shows the density of that particular time when the scene is captured and trend density map shows the density of that whole day trend for three time slots. Map is generated between time slots and approximate area covered with traffic)

5.4 Case Study3: Object detection & classification based Cow Tracking system

5.4.1 Problem

Another application we sought to perceive with our augmented maps was a cattle tracking system. As cattle have the freedom of movement, they can move up to several miles every day. A farmer has to send people looking for them at various places. While this seems trivial for small farms, bigger farms have adapted RFID based cow tracking options. There are many GPS enabled devices available for cow tracker but they have the limitation of a low signal problem. There are many companies who provide cow trackers in various forms such as CowManager, Vence etc. However, they need to be tied on cows body part

The work for animal tracker has been done using wavelet features passing to Complex-

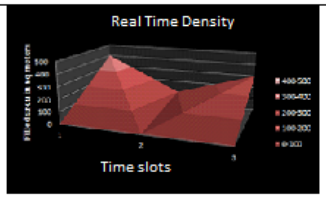
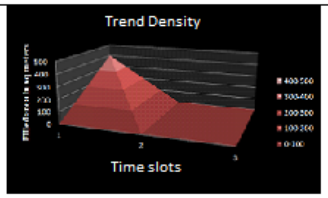
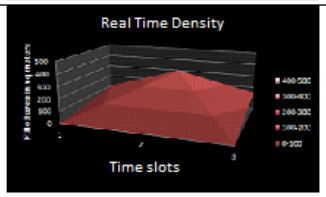
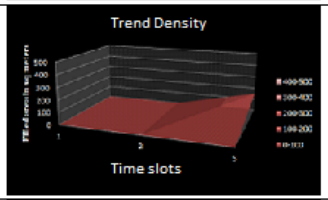
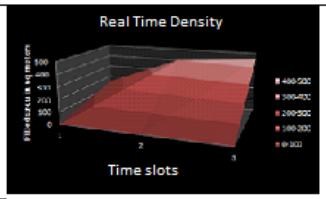
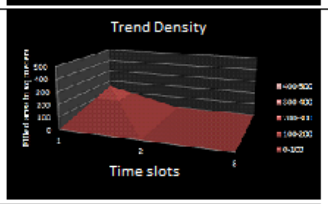
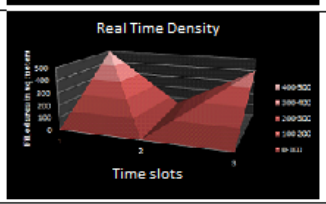
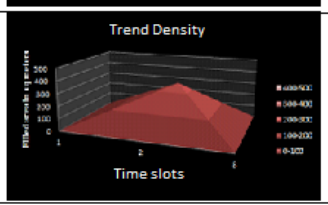
S.No.	Image	Real Time Density	Trend Density	Conclusion
1				Real time observation matched with trend density. The path is very crowded. So, blind user has to change the route
2				Real time observation matched with trend density. The path is not much crowded. So, blind user can follow the same route
3				Real time observation matched with trend density. The path is less crowded. So, blind user can follow the same route
4				Real time observation didnot match with trend density. The path had to be very less crowded according to trend. But it was crowded.

Figure 5.8: Real time scene is captured for giving an input where the output is shown in the form of real time and trend density maps. Maps are prepared with Time slots on x axis and area filled in square meter on y axis. Different colours in graphs represent density level from darker(less crowded) to lighter(highly crowded). Real time maps show all the three slots of the particular day when the scene is captured and trend map show the density map of that particular time only. Conclusion is made on the basis of density map, whether the real time map matches with the trend map or not. Accordingly robot navigation can be planned.

Valued Neural Network (CVNN). This has been done for five classes, wildebeest, zebra, grass, tree, and rock[288]. Specifically cow detection has been implemented by many researchers for various purposes such as implementation of an automatic 3D vision monitor for dairy cow locomotion in a commercial farm[289], implementation of machine vision for detecting behavior of cattle[290], convolutional Neural Network based cow tracker has been developed for tracking the cow in case of their entry and exit from their shelter[291], open source architecture has been developed for long range monitoring in order to track cows[292] where large number of sensors were used to accomplish this accurately, automatic cattle location tracking has been done using cameras[293].

5.4.2 System

In this case study, the application of object detection and classification, and scene localization has been presented. The robot has been used in place of human. The robot will be autonomously navigated to the field. Its camera will continuously capture the scene while navigating. Every frame is passed for object detection and classification using $CNN - Gauss - 1C$ model referred in section 4.3.5. When cows are detected, the scene is also passed to the localizer for scene detection using 2-step approach includes zone detection and landmark detection as referred in section 3.2.1. The information to the farmer will be in the form of frames with detected cows and a map with source to destination route overlayed. This application is very helpful for big farmers who have lot number of cows. Block diagram for the cow tracker has been shown in Figure 5.9

5.4.3 Sample Examples

In order to test our system, we opted to visit a small town named Mandour in Punjab. It has around 800 houses. The dataset of cows has been collected from the farmers of this place. Figure 5.10 shows some of the sample images of collected dataset of farms having many number of cows. The maps of particular place with assigned nodes and zones has been shown in Figure 5.11. These maps have been taken from google maps. Figure 5.12 shows the example of fields where cow goes alone. In this, firstly the map of the place has been generated using map generation module by adding nodes (Node1 to Node9), setting nodes and calculating distance and directions from one node to another as done in section 2.2.2. Further, scene localization model embedded in robot can detect the surrounding scene and information can be given to the user. Map generation model can be used for giving path information to the user i.e path from the home to where the cow

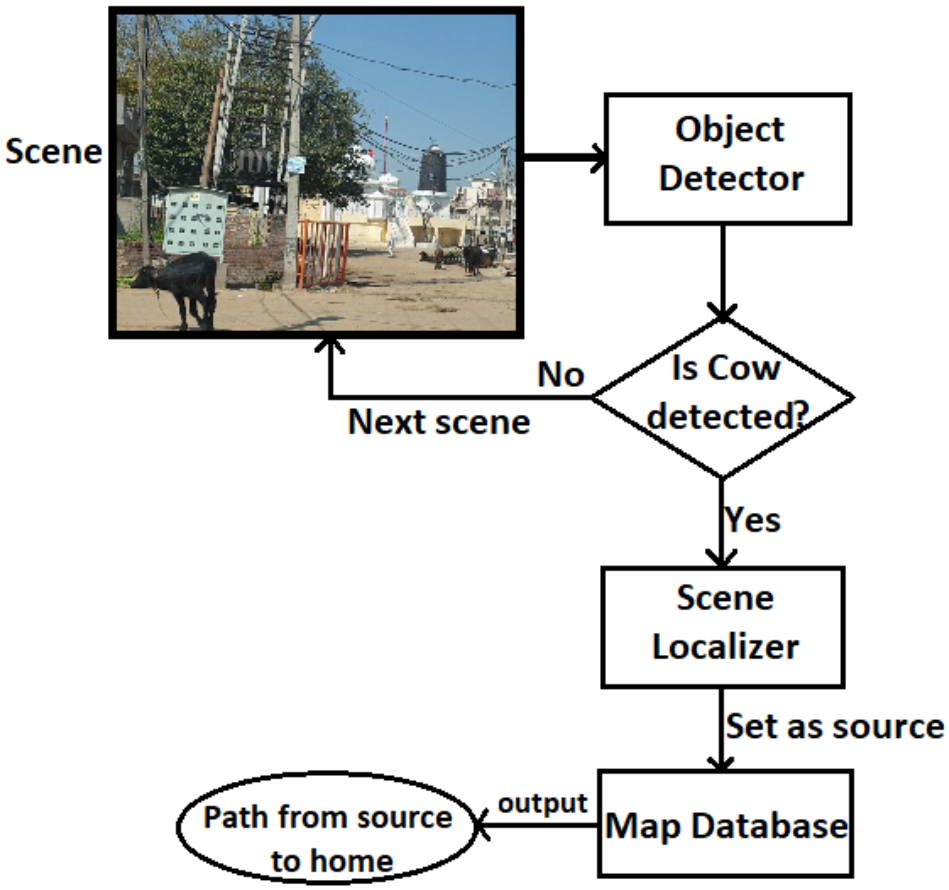
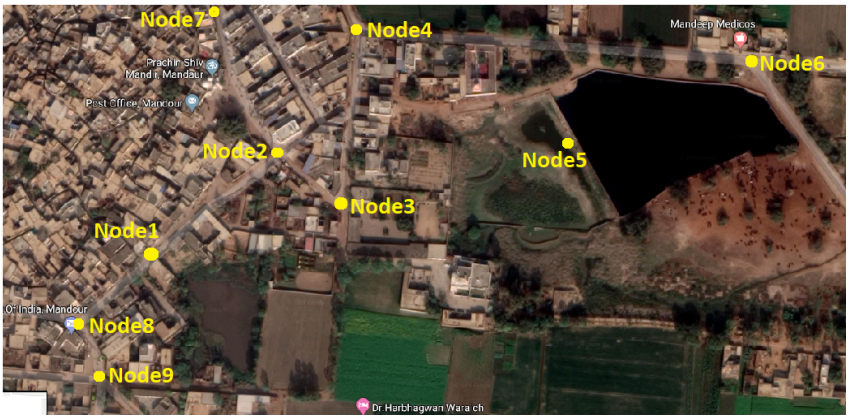
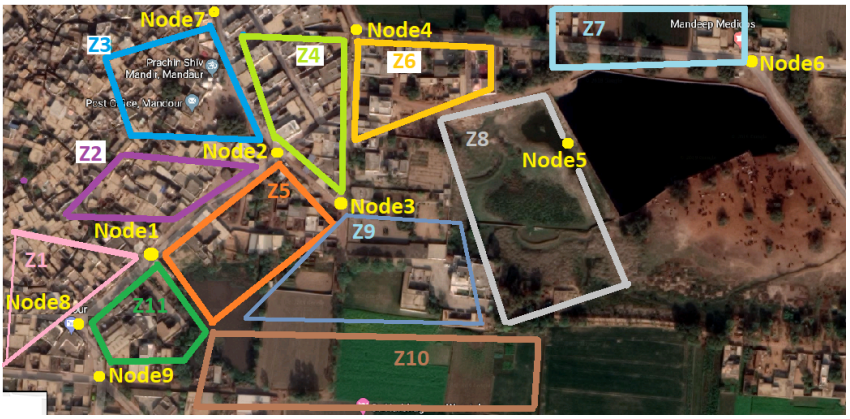


Figure 5.9: Block diagram of cow tracker

Figure 5.10: Sample images of cow dataset



(a) Map with nodes



(b) Map with Zones

Figure 5.11: Map of area where cow tracking has been implemented.

detected. As an example shown in Figure 5.13, the path of cow1 will be "from Node1, go to North-East: 500m, turn to South-East:100m, turn to North: 180m and then turn East 650m to Node5" and path for cow2 will be "from Node1, go to South-West: 150m, turn to South-East: 50m, turn to South: 50m, turn to East: 900m".

(a) Cow detected in Zone Z8

(b) Cow detected in Zone Z3

(c) Cow detected in Zone Z4

(d) Cow detected in Zone Z10

Figure 5.12: Cows lost in fields

Figure 5.13: Path from user's home (mentioned with green dot) to the place where the cow detected. White dot represent 1^{st} cow's location detected in zone8(as shown in Figure5.12a) and yellow dot represents 2^{nd} cow's location detected in zone10(as shown in Figure5.12d). Green line represents path of cow1 and red line represents path of cow2 to home.

5.5 Case Study4: Scene localization based Tour Guide system

5.5.1 Problem

There are many robotic tour guides used in large buildings and museums. Audio guides are available in many historic places where the places to see on the site are numbered. Every tourist is provided with a ear piece which provides the direction from the first site to the next and also relays information about each site. While this serves the purpose of guiding the tourist, it misses out on the interaction part. In order to make the experience more interactive, guides in the form of mobile applications or autonomous mobile robots are available. We implemented a tour guide exploiting our augmented map framework. A single robot is available for touring through the campus outdoors as well as specific indoor buildings. Also, in the absence of the robot an application can be used.
An autonomous indoor tour guide robot has been developed capable in assisting visitors by giving them a tour of the Engineering Labs and its facilities in Asia Pacific University. However this system recognize and understand users request based on keywords only [294]. A tourist guide using kinect has been developed for guiding visitors [295]. A Social Planning and Navigation for Tour-Guide Robot in Human Environment has been proposed in which the robot uses a laser sensor for building environments maps, localization, and detection of new obstacles, and an RGB-D camera (Kinect sensor) for social avoidance [296]. Vasquez et al.[297] have focused on sensor fusion with a semantic map for tour guide robot localization. However, the cost of sensors is high to afford by everyone. There is a robot available named TritonBot, a long-term autonomy robot working as a building receptionist and a tour guide. It recognizes peoples face, talks to them, and guides people to the labs and facilities in an office building. Topological maps have been used to guide the robot. However, this type of robots require lot of maintenance[298]. Another prototype of tour guide has been implemented by Pawade et al. [299] which is AR based application called ARCampusGo. This provide the name and details about the scanned structure or monument from the surrounding area. User has to select the particular structure or monument, it will render the route to the selected monument or structure from the current location.

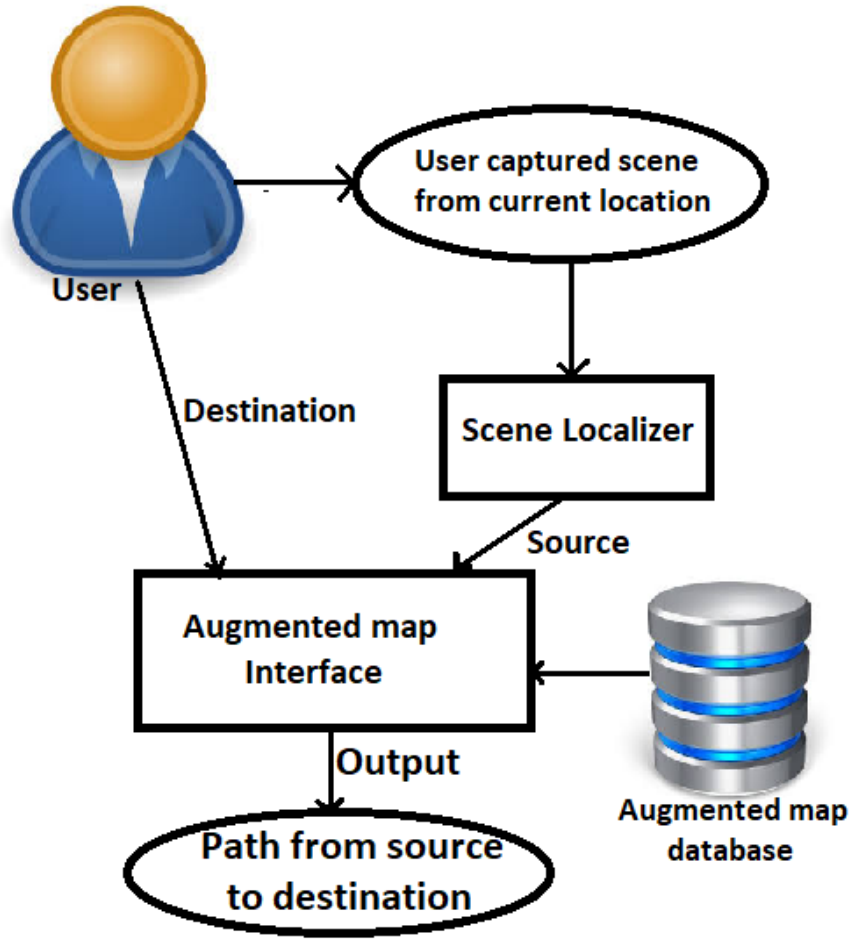


Figure 5.14: Working of tour guide

5.5.2 System

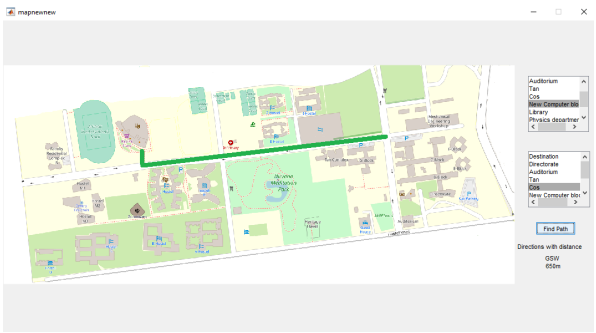
For using the augmented maps as a tour guide, a map developed for university will be used. In chapter2, table for paths from one zone to another has been presented. A GUI has been developed in which source and destination can be entered and the user will get the whole path from source to destination along with its distance. The user can also provide a real time picture of the scene as source and select the destination. The scene localizer will recognize the scene and the map will be used to generate the path to the destination. This will work for indoor locations as well as outdoor locations (even without GPS options). In this particular case study, virtual tour option has been generated from scene panorama using point clouds. Figure 5.14 shows the process of tour guide used in university campus.

5.5.3 Sample Examples

The GUI or software can also be incorporated in robot for the purpose. Otherwise using generated maps, way to destination can be found easily. Figure 5.15 demonstrates the glimpses of tour guide GUI . Figure 5.15a shows the map of university campus. It has sources and destinations lists with "Find path" button. Firstly, the user has to select one source and one destination from the lists and click on the button "Find path". It will show the distance on map and directions along with distance written on right bottom. The voice message can also be used for the directions which can be helpful for visually impaired visitors. Different maps for different entries have been presented in Figures 5.15b to 5.15e. Figure 5.16a depicts small sample of indoor map for navigation. Figure 5.16b presents the path from CITM lab i.e. node N1 to entrance of electronics department i.e. node N7. The directions of going N1 to N7 will be "GSE(Go straight to East) 200m, GSN 800m, GSE 100m GSN 100m".

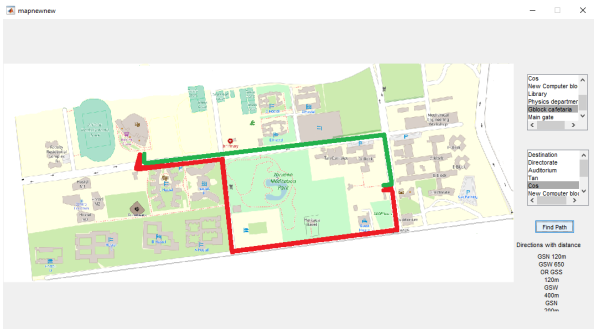
(a) Map of University campus

(b)



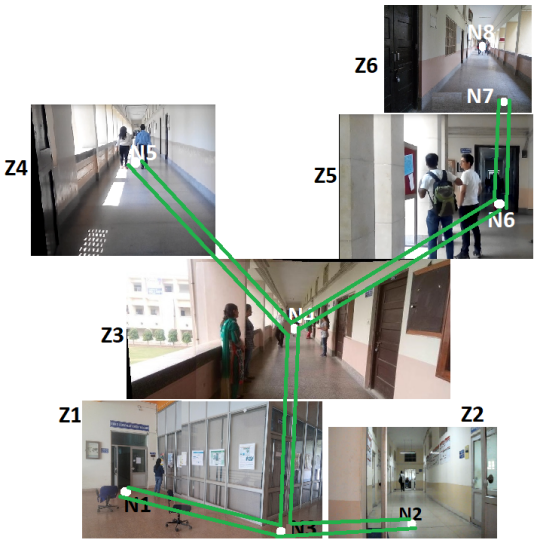
(c)

(d)

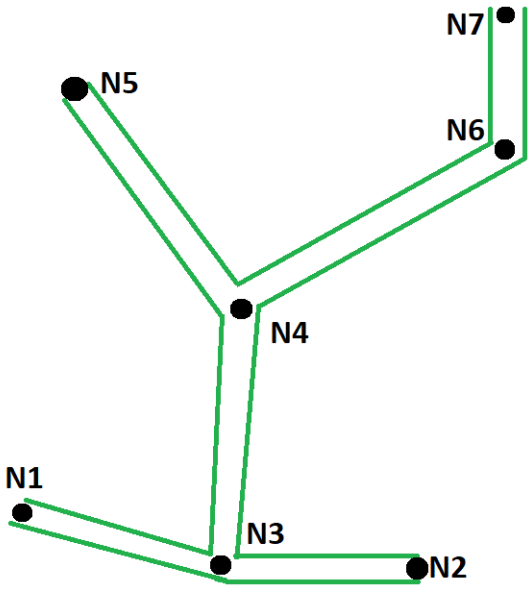


(e) Showing 2 alternate paths(green and red) from source to destination

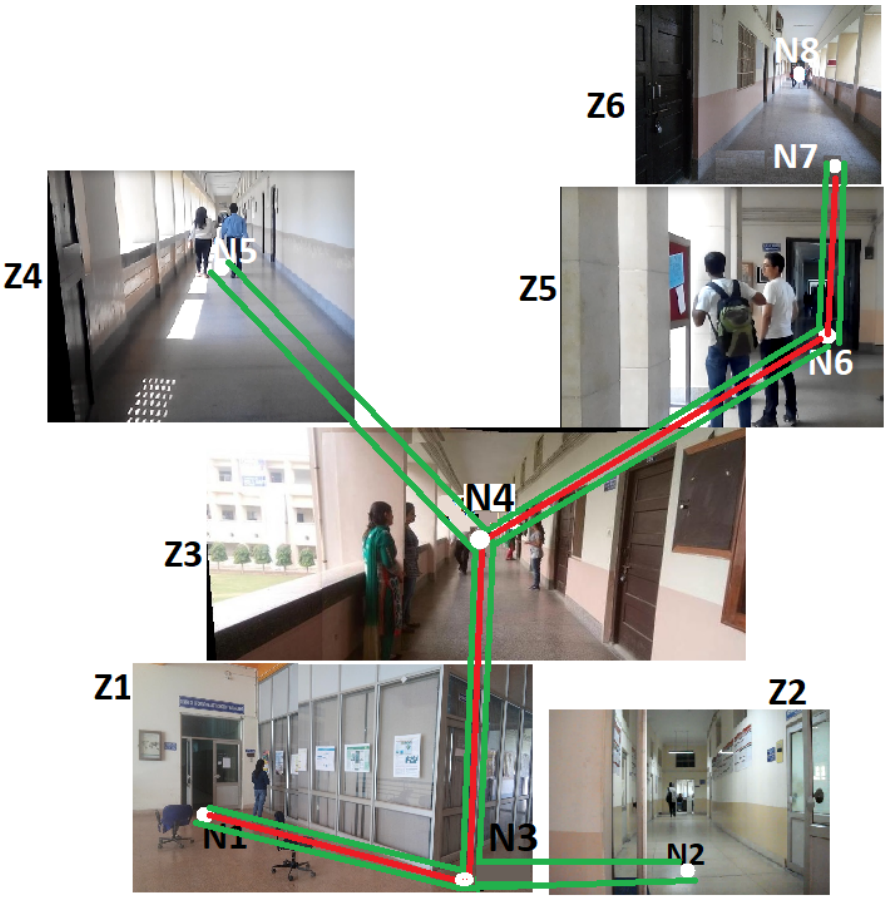
Figure 5.15: Few of the glimpses of GUI tour guide from source to destination along with the directions and distances. Four instances of app usage by tourist shown in Figure 5.15b 5.15c 5.15d 5.15e



(a) Inside View of CSE department with nodes, zones and pathway(green) from N1 to N7



(b) Pathway showing from N1 to N7 in CSE department



(c) Path(in red) from N1 to N7 embedded on map

Figure 5.16: Sample example of Tour guide for inside the buildings

Chapter 6

Conclusions and Future Works

This chapter is the concluding part of the thesis and also proposes some suggestions towards which the present work can be further extended. Section 6.1 presents the contributions for this particular research. Section 6.2 reports some future research directions and possible extensions of the work presented in the thesis.

6.1 Research Contribution

The work aims towards developing augmented maps which can be used for navigation by a human tourist, visually impaired or autonomous robot. At the same time it can be viewed as a scene perception system which provide information about objects in the scene and their distances from the subject, localization of the scene, density prediction in the scene. These involve 3 essential modules as discussed below:

1. Object Detector Module

   (a) For traffic or object classification, a Network on Convolutional Feature Maps(NoC) has been trained. The experiments for the current work have been performed with three different architectures to know the importance of adding or removing the layers.

   (b) Three different learning rates have been used to train NoCs in order to understand how the learning of the network is affected. This work particularly proposes an average covariance based pre-conditioning approach to deal with saddle points in deep networks.

   (c) NoCs have also been trained with blurred dataset and multimodal features have been used to accommodate blurred scenes during real time processing.

   (d) For improving the performance of deep learning-based object detector, fusion of features of normal RGB and features extracted from image edges, optical flow as well as scale space representation has been used.

2. Scene Localization Module

(a) A 2 step approach was used for scene localization i.e zones have been detected & recognized using set-based image classification. Three different set-based distance algorithms(COV+LDA, COV+PLS and HERML) have been used for calculating distance and aggregation of distances obtained from three algorithms has been utilized for considering final distance. The input to these algorithms were the features extracted from last convolutional layer of places alexnet and VGG nets. Further, capsule network model was trained for landmark classification.

(b) Transfer learning information from soft target training was used.

(c) We generated multiple dustbin classes using GAN to replace single background class training for greater confidence of network for known classes and lower confidence for unknown classes. It helped to improve the confidence of capsule network to detect outliers.

3. Density Estimation Module

(a) Suitable pre-processing (based on FFT and key frame selection)technique has been applied on data-set to remove homogeneity.

(b) Obtained itinerary perception subject to different environmental conditions. This refers to extraction of traffic related information from an augmented map. The problem was modeled as a machine learning technique where the traffic distribution at different times (including same days,different days,different weather) were observed continuously using a service WiRobot. This data was posed as a gaussian process for post estimation of traffic density distributions, learned from the scenes at different points of time.

6.2 Future Scope

This work focuses on integrating different modules like scene localization, object detection, density estimation to build an augmented map. Simultaneously it discusses some design and training specifics like soft target training, multimodal feature fusion, multiple dustbin classes for example. All the deep learning models are build in torch/pytorch environments. A development level next step will be to convert these into tensorflowlite formats such that they can be easily used on Coral boards. On a more research level perspective, an important extension of this work is to engage multispectral data for map building. This will not only improve the localization and detection accuracies but will

also make the system more robust in different cloudy and foggy environments where one sensor will complement the other. Another important aspect is to deal with temporal data. For all the modules reported in the work, each frame is processed individually. Taking temporal information into consideration will further improve the realtime performance in terms of false alarm avoidance, when the detection in successive frames depend on the availability of the object in the previous frame.

References

[1] Weikai Xie, Yuanchun Shi, and Guanyou Xu. Smart classroom–an intelligent environment for distant education. *PCM2001, Springer LNCS2195*, pages 662–668, 2002.

[2] Roger Nkambou, Jacqueline Bourdeau, and Riichiro Mizoguchi. Introduction: what are intelligent tutoring systems, and why this book? In *Advances in intelligent tutoring systems*, pages 1–12. Springer, 2010.

[3] Vasile Rus, Nobal Niraula, and Rajendra Banjade. Deeptutor: An effective, online intelligent tutoring system that promotes deep learning. In *29th AAAI Conference on Artificial Intelligence*, pages 4294–4295, 2015.

[4] Ed Sykes. Developmental process model for the java intelligent tutoring system. *Journal of Interactive Learning Research*, 18(3):399–410, 2007.

[5] Martin Eberhard and Marc Tarpenning. The 21 st century electric car tesla motors. *Tesla Motors*, pages 1–10, 2006.

[6] Michel Parent Charles E. Thorpe Alberto Broggi, Alexander Zelinsky. Intelligent vehicles. *Springer Handbook of Robotics*, pages 1175–1198, 2016.

[7] Amnon Shashua and Ziv Aviram. About orcam. `https://www.orcam.com/en/about/`, 2017 (Accessed June 17, 2019).

[8] Robert Neßelrath, Chensheng Lu, Christian H Schulz, Jochen Frey, and Jan Alexandersson. A gesture based system for context–sensitive interaction with smart homes. In *Ambient Assisted Living*, pages 209–219. Springer, 2011.

[9] Sagar B Vinay Raj KM Chinmaya HG, Nithin kumar N. Gesture based smart home automation system using real time inputs. *International Journal of Latest Research in Engineering and Technology*, ISSN 2454-5031:108–112, 2016.

[10] Thuy Ong. Mekamon is an ar robot that you control using your smartphone. `https://www.theverge.com/circuitbreaker/2017/11/15/16642774/mekamon-ar-robot`, 2017 (Accessed June 17, 2019).

[11] Business Wire. Loral joins youcam makeup, perfect corp.s augmented reality makeover app. `https://www.businesswire.com/news/home/20170710005692/en/LOreal-20Joins-YouCam-20Makeup-Perfect-20Corp's-Augmented`, 2017 (Accessed June 17, 2019).

[12] Newgen. 8 examples of augmented reality apps and their successful uses. `https://www.newgenapps.com/blog/augmented-reality-apps-ar-examples-success`, 2017 (Accessed June 17, 2019).

[13] Matei Stroila, Joe Mays, Bill Gale, and Jeff Bach. Augmented transit maps. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 485–490, 2011.

[14] Gerhard Reitmayr, Ethan Eade, and Tom Drummond. Localisation and interaction for augmented maps. In *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 120–129, 2005.

[15] Daniele Croce, Pierluigi Gallo, Domenico Garlisi, Laura Giarré, Stefano Mangione, and Ilenia Tinnirello. Arianna: A smartphone-based navigation system with human in the loop. In *22nd IEEE Mediterranean Conference of Control and Automation (MED)*, pages 8–13, 2014.

[16] Lynette A Jones, Brett Lockyer, and Erin Piateski. Tactile display and vibrotactile pattern recognition on the torso. *Advanced Robotics*, 20(12):1359–1374, 2006.

[17] Shraga Shoval, Johann Borenstein, and Yoram Koren. The navbelt-a computerized travel aid for the blind. In *Proceedings of the RESNA Conference*, pages 13–18, 1993.

[18] Jascha Sohl-Dickstein, Santani Teng, Benjamin M Gaub, Chris C Rodgers, Crystal Li, Michael R DeWeese, and Nicol S Harper. A device for human ultrasonic echolocation. *IEEE Transactions on Biomedical Engineering*, 62(6):1526–1534, 2015.

[19] Rohit Agarwal, Nikhil Ladha, Mohit Agarwal, Kuntal Kr Majee, Abhijit Das, Subham Kumar, Subham Kr Rai, Anand Kr Singh, Somen Nayak, Shopan Dey, et al. Low cost ultrasonic smart glasses for blind. In *8th IEEE Annual; Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 210–213, 2017.

[20] Hakan Koyuncu and Shuang Hua Yang. A survey of indoor positioning and object locating systems. *IJCSNS International Journal of Computer Science and Network Security*, 10(5):121–128, 2010.

[21] Hartmut Surmann, Kai Lingemann, Andreas Nüchter, and Joachim Hertzberg. A 3D laser range finder for autonomous mobile robots. In *Proceedings of the 32nd International Symposium on Robotics (ISR)*, volume 19, pages 153–158, 2001.

[22] Johann Borenstein and Yoram Koren. Obstacle avoidance with ultrasonic sensors. *IEEE Journal on Robotics and Automation*, 4(2):213–218, 1988.

[23] Kesav Kaliyaperumal, Sridhar Lakshmanan, and Karl Kluge. An algorithm for detecting roads and obstacles in radar images. *IEEE Transactions on Vehicular Technology*, 50(1):170–182, 2001.

[24] Jiali Shen and Huosheng Hu. Visual navigation of a museum guide robot. In *6th IEEE World Congress on Intelligent Control and Automation*, volume 2, pages 9169–9173, 2006.

[25] David Ball, Ben Upcroft, Gordon Wyeth, Peter Corke, Andrew English, Patrick Ross, Tim Patten, Robert Fitch, Salah Sukkarieh, and Andrew Bate. Vision-based obstacle detection and navigation for an agricultural robot. *Journal of Field Robotics*, 33(8):1107–1130, 2016.

[26] Olivier Koch, Matthew R Walter, Albert S Huang, and Seth Teller. Ground robot navigation using uncalibrated cameras. In *IEEE International Conference on Robotics and Automation*, pages 2423–2430, 2010.

[27] Wen-Chung Chang and Chun-Yi Chuang. Vision-based robot navigation and map building using active laser projection. In *IEEE/SICE International Symposium on System Integration (SII)*, pages 24–29, 2011.

[28] Emanuele Menegatti, Takeshi Maeda, and Hiroshi Ishiguro. Image-based memory for robot navigation using properties of omnidirectional images. *Robotics and Autonomous Systems*, 47(4):251–267, 2004.

[29] Afroza Begum, Minkyoung Lee, and Young J Kim. A simple visual servoing and navigation algorithm for an omnidirectional robot. In *3rd International Conference on Human-Centric Computing*, pages 1–5, 2010.

[30] Andrea Cherubini, François Chaumette, and Giuseppe Oriolo. An image-based visual servoing scheme for following paths with nonholonomic mobile robots. In *10th IEEE International Conference on Control, Automation, Robotics and Vision*, pages 108–113, 2008.

[31] Andrea Cherubini and François Chaumette. Visual navigation of a mobile robot with laser-based collision avoidance. *The International Journal of Robotics Research*, 32(2):189–205, 2013.

[32] Jonathan Courbon, Youcef Mezouar, and Philippe Martinet. Autonomous navigation of vehicles from a visual memory using a generic camera model. *IEEE Transactions on Intelligent Transportation Systems*, 10(3):392–402, 2009.

[33] Genci Capi, Hideki Toda, and Takuya Nagasaki. A vision based robot navigation and human tracking for social robotics. In *IEEE International Workshop on Robotic and Sensors Environments*, pages 1–6, 2010.

[34] Ebrahim Mattar, Khalid Al Mutib, Mansour Al Sulaiman, and Hajar Ramdane. Mobile robot intelligence based slam features learning and navigation. *International Journal of Computing and Digital Systems*, 7(01):23–34, 2018.

[35] Bing Li, Juan Pablo Munoz, Xuejian Rong, Qingtian Chen, Jizhong Xiao, Yingli Tian, Aries Arditi, and Mohammed Yousuf. Vision-based mobile indoor assistive navigation aid for blind people. *IEEE Transactions on Mobile Computing*, 18(3):702–714, 2019.

[36] Kaichun Mo, Haoxiang Li, Zhe Lin, and Joon-Young Lee. The adobeindoornav

dataset: Towards deep reinforcement learning based real-world indoor robot visual navigation. *arXiv preprint arXiv:1802.08824*, 2018.

[37] Young Hoon Lee and Gerard Medioni. Wearable RGBD indoor navigation system for the blind. In *European Conference on Computer Vision*, pages 493–508. Springer, 2014.

[38] Pablo Sala, Robert Sim, Ali Shokoufandeh, and Sven Dickinson. Landmark selection for vision-based navigation. *IEEE Transactions on Robotics*, 22(2):334–349, 2006.

[39] Enis Bayramoglu, Nils Axel Andersen, Niels Kjolstad Poulsen, Jens Christian Andersen, and Ole Ravn. Mobile robot navigation in a corridor using visual odometry. In *IEEE International Conference on Advanced Robotics*, pages 1–6, 2009.

[40] Jean-Bernard Hayet, Frédéric Lerasle, and Michel Devy. A visual landmark framework for mobile robot navigation. *Image and Vision Computing*, 25(8):1341–1351, 2007.

[41] Marco A Luna, Julio F Moya, Wilbert G Aguilar, and Vanessa Abad. Mobile robot with vision based navigation and pedestrian detection. *Ingenius*, 17:67–72, 2017.

[42] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2017.

[43] Charles Richter and Nicholas Roy. Safe visual navigation via deep learning and novelty detection. *Robotics: Science and Systems Foundation*, pages 1–23, 2017.

[44] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3357–3364, 2017.

[45] Justin S Smith, Jin-Ha Hwang, Fu-Jen Chu, and Patricio A Vela. Learning to navigate: Exploiting deep networks to inform sample-based planning during vision-based navigation. *arXiv preprint arXiv:1801.05132*, 2018.

[46] Antonis A Argyros, Kostas E Bekris, Stelios C Orphanoudakis, and Lydia E Kavraki. Robot homing by exploiting panoramic vision. *Autonomous Robots*, 19(1):7–25, 2005.

[47] Pablo De Cristóforis, Matias Nitsche, Tomáš Krajník, Taihú Pire, and Marta Mejail. Hybrid vision-based navigation for mobile robots in mixed indoor/outdoor environments. *Pattern Recognition Letters*, 53:118–128, 2015.

[48] Ranga Rodrigo and Jagath Samarabandu. Monocular vision for robot navigation. In *IEEE International Conference Mechatronics and Automation*, volume 2, pages

707–712, 2005.

[49] Gayan D Illeperuma and Upul J Sonnadara. An autonomous robot navigation system based on optical flow. In *6th IEEE International Conference on Industrial and Information Systems*, pages 489–492, 2011.

[50] Arsalan Mousavian, Alexander Toshev, Marek Fiser, Jana Kosecka, Ayzaan Wahid, and James Davidson. Visual representations for semantic target driven navigation. *arXiv preprint arXiv:1805.06066*, 2018.

[51] Axel López, François Chaumette, Eric Marchand, and Julien Pettré. Vision-based robot navigation through combining unsupervised learning and hierarchical reinforcement learning in dynamic environments based on optical flow. In *Computer Graphics Forum*, volume 38, pages 181–192, 2019.

[52] Nguyen Xuan Dao, Bum-Jae You, and Sang-Rok Oh. Visual navigation for indoor mobile robots using a single camera. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1992–1997, 2005.

[53] Wang Yuan, Zhijun Li, and Chun-Yi Su. Rgb-d sensor-based visual slam for localization and navigation of indoor mobile robot. In *IEEE International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 82–87, 2016.

[54] Hsueh-Cheng Wang, Robert K Katzschmann, Santani Teng, Brandon Araki, Laura Giarré, and Daniela Rus. Enabling independent navigation for visually impaired people through a wearable vision-based feedback system. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6533–6540, 2017.

[55] Eric Royer, Jonathan Bom, Michel Dhome, Benoit Thuilot, Maxime Lhuillier, and François Marmoiton. Outdoor autonomous navigation using monocular vision. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1253–1258, 2005.

[56] Eric Royer, Maxime Lhuillier, Michel Dhome, and Jean-Marc Lavest. Monocular vision for mobile robot localization and autonomous navigation. *International Journal of Computer Vision*, 74(3):237–260, 2007.

[57] Andrea Cherubini, Fabien Spindler, and François Chaumette. Autonomous visual navigation and laser-based moving obstacle avoidance. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):2101–2110, 2014.

[58] Kaushik Das Sharma, Amitava Chatterjee, and Anjan Rakshit. A pso–lyapunov hybrid stable adaptive fuzzy tracking control approach for vision-based robot navigation. *IEEE Transactions on Instrumentation and Measurement*, 61(7):1908–1914, 2012.

[59] S Gaglione, S Del Pizzo, S Troisi, and A Angrisano. Position accuracy analysis of a robust vision-based navigation system. *The International Archives of the*

Photogrammetry, Remote Sensing and Spatial Information Sciences, 42(2):355–361, 2018.

[60] Danilo Alves de Lima and Alessandro Corrêa Victorino. A visual servoing approach for road lane following with obstacle avoidance. In *17th IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 412–417, 2014.

[61] G Capi, S Kaneko, and B Hua. Neural network based guide robot navigation: an evolutionary approach. *Procedia Computer Science*, 76:74–79, 2015.

[62] Danilo Alves de Lima and Alessandro Corrêa Victorino. A hybrid controller for vision-based navigation of autonomous vehicles in urban environments. *IEEE Transactions on Intelligent Transportation Systems*, 17(8):2310–2323, 2016.

[63] Andrew English, Patrick Ross, David Ball, and Peter Corke. Vision based guidance for robot navigation in agriculture. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1693–1698, 2014.

[64] Kiyosumi Kidono, Jun Miura, and Yoshiaki Shirai. Autonomous visual navigation of a mobile robot using a human-guided experience. *Robotics and Autonomous Systems*, 40(2-3):121–130, 2002.

[65] Xiaomao Zhou, Tao Bai, Yanbin Gao, and Yuntao Han. Vision-based robot navigation through combining unsupervised learning and hierarchical reinforcement learning. *Sensors*, 19(7):1576–1599, 2019.

[66] DK Liyanage and MUS Perera. Optical flow based obstacle avoidance for the visually impaired. In *IEEE Business, Engineering & Industrial Applications Colloquium (BEIAC)*, pages 284–289, 2012.

[67] Liming Yang, Jean-Marie Normand, and Guillaume Moreau. Augmenting off-the-shelf paper maps using intersection detection and geographical information systems. In *14th IEEE/IAPR International Conference on Machine Vision Applications (MVA)*, pages 190–193, 2015.

[68] Volker Paelke and Monika Sester. Augmented paper maps: Exploring the design space of a mixed reality system. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(3):256–265, 2010.

[69] Ann Morrison, Alessandro Mulloni, Saija Lemmelä, Antti Oulasvirta, Giulio Jacucci, Peter Peltonen, Dieter Schmalstieg, and Holger Regenbrecht. Collaborative use of mobile augmented reality with paper maps. *Computers & Graphics*, 35(4):789–799, 2011.

[70] Kihwan Kim, Sangmin Oh, Jeonggyu Lee, and Irfan Essa. Augmenting aerial earth maps with dynamic information. In *8th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 35–38, 2009.

[71] Elin Topp, Henrik Christensen, et al. Topological modelling for human augmented

mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2257–2263, 2006.

[72] Chien-Hung Liu and Kai-Tai Song. A new approach to map joining for depth-augmented visual slam. In *9th IEEE Asian Control Conference (ASCC)*, pages 1–6, 2013.

[73] Leslie Kay. Auditory perception of objects by blind persons, using a bioacoustic high resolution air sonar. *The Journal of the Acoustical Society of America*, 107(6):3266–3275, 2000.

[74] Paul Bach-y Rita and Stephen W Kercel. Sensory substitution and the human–machine interface. *Trends in Cognitive Sciences*, 7(12):541–546, 2003.

[75] KA Kaczmarek. The tongue display unit (tdu) for electrotactile spatiotemporal pattern presentation. *Scientia Iranica*, 18(6):1476–1485, 2011.

[76] Malika Auvray, Sylvain Hanneton, and J Kevin O'Regan. Learning to perceive with a visuoauditory substitution system: localisation and object recognition with the voice. *Perception*, 36(3):416–430, 2007.

[77] Sami Abboud, Shlomi Hanassy, Shelly Levy-Tzedek, Shachar Maidenbaum, and Amir Amedi. Eyemusic: Introducing a visual colorful experience for the blind using auditory sensory substitution. *Restorative Neurology and Neuroscience*, 32(2):247–257, 2014.

[78] Yasuhiro Matsuda, Ichiro Sakuma, Yasuhiko Jimbo, Etsuko Kobayashi, Tatsuhiko Arafune, and Tsuneshi Isomura. Finger braille recognition system for people who communicate with deafblind people. In *IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 268–273, 2008.

[79] Joselin Villanueva and René Farcy. Optical device indicating a safe free path to blind people. *IEEE Transactions on Instrumentation and Measurement*, 61(1):170–177, 2012.

[80] Mun-Cheon Kang, Sung-Ho Chae, Jee-Young Sun, Jin-Woo Yoo, and Sung-Jea Ko. A novel obstacle detection method based on deformable grid for the visually impaired. *IEEE Transactions on Consumer Electronics*, 61(3):376–383, 2015.

[81] Van-Nam Hoang, Thanh-Huong Nguyen, Thi-Lan Le, Thi-Thanh Hai Tran, Tan-Phu Vuong, and Nicolas Vuillerme. Obstacle detection and warning for visually impaired people based on electrode matrix and mobile kinect. In *2nd IEEE National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS)*, pages 54–59, 2015.

[82] Shaomei Wu, Jeffrey Wieland, Omid Farivar, and Julie Schiller. Automatic alt-text: Computer-generated image descriptions for blind users on a social network service. In *CSCW*, pages 1180–1192, 2017.

[83] Shweta Jaiswal, Jyothi Warrier, Vineet Sinha, Rajesh Kumar Jain, and ME Student. Small sized vision based system for blinds. *International Journal of Engineering Science*, 8:15968–1572, 2018.

[84] Robert Katzschmann, Brandon Araki, and Daniela Rus. Safe local navigation for visually impaired users with a time-of-flight and haptic feedback device. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26:583–593.

[85] Shaoqing Ren, Kaiming He, Ross Girshick, Xiangyu Zhang, and Jian Sun. Object detection networks on convolutional feature maps. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 39(7):1476–1481, 2017.

[86] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural l0 sparse representation for natural image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1107–1114, 2013.

[87] Hojin Cho, Jue Wang, and Seungyong Lee. Text image deblurring using text-specific properties. In *European Conference on Computer Vision*, pages 524–537. Springer, 2012.

[88] Shicheng Zheng, Li Xu, and Jiaya Jia. Forward motion deblurring. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1465–1472, 2013.

[89] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[90] Ross Girshick. Fast R-CNN. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[91] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[92] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint*, 1612, 2016.

[93] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The apolloscape dataset for autonomous driving. *arXiv preprint arXiv:1803.06184*, 2018.

[94] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 32:1231–1237, 2013.

[95] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. In *CVPR*, pages 3213–3221, 2017.

[96] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented

natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th International Conference on Computer Vision*, volume 2, pages 416–423, July 2001.

[97] Hugo FM Milan, Kristen M Perano, and Kifle G Gebremedhin. Survey and future prospects in precision dairy farming. In *10th International Livestock Environment Symposium (ILES X)*, page 1. American Society of Agricultural and Biological Engineers, 2018.

[98] Derek Magee and Roger Boyle. Feature tracking in real world scenes (or how to track a cow). 1999.

[99] Sherine Rady. Vision-based hybrid map-building and robot localization in unstructured and moderately dynamic environments. In *Novel Applications of Intelligent Systems*, pages 231–250. Springer, 2016.

[100] Hao Wu, Guo-hui Tian, Yan Li, Feng-yu Zhou, and Peng Duan. Spatial semantic hybrid map building and application of mobile service robot. *Robotics and Autonomous Systems*, 62(6):923–941, 2014.

[101] Hongtai Cheng, Heping Chen, and Yong Liu. Topological indoor localization and navigation for autonomous mobile robot. *IEEE Transactions on Automation Science and Engineering*, 12(2):729–738, 2015.

[102] Rui Xu, Wu Chen, Ying Xu, and Shengyue Ji. A new indoor positioning system architecture using gps signals. *Sensors*, 15(5):10074–10087, 2015.

[103] D Wu, A Yang, L Zhu, and C Zhang. Life system modeling and simulation. In *International Conference on Life System Modeling and Simulation and International Conference on Intelligent Computing for Sustainable Energy and Environment*, volume 461, 2014.

[104] Ugur Yayan, Hikmet Yucel, et al. A low cost ultrasonic based positioning system for the indoor navigation of mobile robots. *Journal of Intelligent & Robotic Systems*, 78(3-4):541–552, 2015.

[105] Nak Yong Ko and Tae-Yong Kuc. Fusing range measurements from ultrasonic beacons and a laser range finder for localization of a mobile robot. *Sensors*, 15(5):11050–11075, 2015.

[106] Xiang Song, Xu Li, Wencheng Tang, and Weigong Zhang. A fusion strategy for reliable vehicle positioning utilizing rfid and in-vehicle sensors. *Information Fusion*, 31:76–86, 2016.

[107] Gabriel Deak, Kevin Curran, and Joan Condell. A survey of active and passive indoor localisation systems. *Computer Communications*, 35(16):1939–1954, 2012.

[108] Luca Mainetti, Luigi Patrono, and Ilaria Sergi. A survey on indoor positioning systems. In *22nd IEEE International Conference on Software, Telecommunications*

and Computer Networks (SoftCOM), pages 111–120, 2014.

[109] Andrzej Kwiecień, Michał Maćkowski, Marek Kojder, and Maciej Manczyk. Reliability of bluetooth smart technology for indoor localization system. In *International Conference on Computer Networks*, pages 444–454. Springer, 2015.

[110] Honggui Li. Low-cost 3D bluetooth indoor positioning with least square. *Wireless Personal Communications*, 78(2):1331–1344, 2014.

[111] Jiří Kárník and Jakub Streit. Summary of available indoor location techniques. *IFAC-PapersOnLine*, 49(25):311–317, 2016.

[112] Abdulrahman Alarifi, AbdulMalik Al-Salman, Mansour Alsaleh, Ahmad Alnafessah, Suheer Al-Hadhrami, Mai A Al-Ammar, and Hend S Al-Khalifa. Ultra wideband indoor positioning technologies: Analysis and recent advances. *Sensors*, 16(5):1–36, 2016.

[113] Shih-Hau Fang, Chu-Hsuan Wang, Ting-Yu Huang, Chin-Huang Yang, and Yung-Sheng Chen. An enhanced zigbee indoor positioning system with an ensemble approach. *IEEE Communications Letters*, 16(4):564–567, 2012.

[114] Tashnim JS Chowdhury, Colin Elkin, Vijay Devabhaktuni, Danda B Rawat, and Jared Oluoch. Advances on localization techniques for wireless sensor networks: A survey. *Computer Networks*, 110:284–305, 2016.

[115] Michał Nowicki and Piotr Skrzypczyński. Indoor navigation with a smartphone fusing inertial and wifi data via factor graph optimization. In *International Conference on Mobile Computing, Applications, and Services*, pages 280–298. Springer, 2015.

[116] Rafael Saraiva Campos, Lisandro Lovisolo, and Marcello Luiz R de Campos. Wi-fi multi-floor indoor positioning considering architectural aspects and controlled computational complexity. *Expert Systems with Applications*, 41(14):6211–6223, 2014.

[117] Hyuk Lim, Lu-Chuan Kung, Jennifer C Hou, and Haiyun Luo. Zero-configuration indoor localization over ieee 802.11 wireless infrastructure. *Wireless Networks*, 16(2):405–420, 2010.

[118] Reza Zandian and Ulf Witkowski. Performance analysis of small size and power efficient UWB communication nodes for indoor localization. In *Conference Towards Autonomous Robotic Systems*, pages 371–382. Springer, 2016.

[119] Stefania Monica and Gianluigi Ferrari. A swarm intelligence approach to 3D distance-based indoor uwb localization. In *European Conference on the Applications of Evolutionary Computation*, pages 91–102. Springer, 2015.

[120] Eva Arias-de Reyna. A cooperative localization algorithm for uwb indoor sensor networks. *Wireless Personal Communications*, 72(1):85–99, 2013.

[121] Luca Calderoni, Matteo Ferrara, Annalisa Franco, and Dario Maio. Indoor localization in a hospital environment using random forest classifiers. *Expert Systems with Applications*, 42(1):125–134, 2015.

[122] M Suruz Miah and Wail Gueaieb. A fuzzy logic approach for indoor mobile robot navigation using ukf and customized rfid communication. In *International Conference on Autonomous and Intelligent Systems*, pages 21–30. Springer, 2011.

[123] Hong-Shik Kim and Jong-Suk Choi. Advanced indoor localization using ultrasonic sensor and digital compass. In *IEEE International Conference on Control, Automation and Systems (ICCAS)*, pages 223–226, 2008.

[124] Marios Sioutis and Yasuo Tan. User indoor location system with passive infrared motion sensors and space subdivision. In *International Conference on Distributed, Ambient, and Pervasive Interactions*, pages 486–497. Springer, 2014.

[125] Grant Schindler, Christian Metzger, and Thad Starner. A wearable interface for topological mapping and localization in indoor environments. In *International Symposium on Location-and Context-Awareness*, pages 64–73. Springer, 2006.

[126] Alex Varshavsky, Eyal De Lara, Jeffrey Hightower, Anthony LaMarca, and Veljo Otsason. Gsm indoor localization. *Pervasive and Mobile Computing*, 3(6):698–720, 2007.

[127] Nisarg Kothari, Balajee Kannan, Evan D Glasgwow, and M Bernardine Dias. Robust indoor localization on a commercial smart phone. *Procedia Computer Science*, 10:1114–1120, 2012.

[128] Antonio R Jimenez, Fernando Seco, Carlos Prieto, and Jorge Guevara. A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu. In *IEEE International Symposium on Intelligent Signal Processing*, pages 37–42, 2009.

[129] Zheng Yang, Zimu Zhou, and Yunhao Liu. From rssi to csi: Indoor localization via channel response. *ACM Computing Surveys (CSUR)*, 46(2):1–25, 2013.

[130] Faheem Zafari, Athanasios Gkelias, and Kin Leung. A survey of indoor localization systems and technologies. *arXiv preprint arXiv:1709.01015*, 2017.

[131] Bemri. Visible light communication (vlc/li-fi) systems. *http://bemri.org/visible-light-communication.html.*, 2016.

[132] Marco Centenaro, Lorenzo Vangelista, Andrea Zanella, and Michele Zorzi. Long-range communications in unlicensed bands: The rising stars in the iot and smart city scenarios. *IEEE Wireless Communications*, 23(5):60–67, 2016.

[133] Francisco Bonin-Font, Alberto Ortiz, and Gabriel Oliver. Visual navigation for mobile robots: A survey. *Journal of Intelligent and Robotic Systems*, 53(3):263–296, 2008.

[134] Jessyca Almeida Bessa, Darlan Almeida Barroso, Ajalmar Rego da Rocha Neto,

and Auzuir Ripardo de Alexandria. Global location of mobile robots using artificial neural networks in omnidirectional images. *IEEE Latin America Transactions*, 13(10):3405–3414, 2015.

[135] Alejandro Rituerto, AC Murillo, and JJ Guerrero. Semantic labeling for indoor topological mapping using a wearable catadioptric system. *Robotics and Autonomous Systems*, 62(5):685–695, 2014.

[136] Ming Liu and Roland Siegwart. DP-FACT: Towards topological mapping and scene recognition with color for omnidirectional camera. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3503–3508, 2012.

[137] Marcin Dymczyk, Marius Fehr, Thomas Schneider, and Roland Siegwart. Long-term large-scale mapping and localization using maplab. *arXiv preprint arXiv:1805.10994*, 2018.

[138] Florian Walch, Caner Hazirbas, Laura Leal-Taixe, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers. Image-based localization using lstms for structured feature correlation. In *International Conference on Computer Vision (ICCV)*, pages 627–637, 2017.

[139] Eyasu Zemene, Yonatan Tariku Tesfaye, Haroon Idrees, Andrea Prati, Marcello Pelillo, and Mubarak Shah. Large-scale image geo-localization using dominant sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):148–161, 2019.

[140] Emilio Garcia-Fidalgo and Alberto Ortiz. Vision-based topological mapping and localization methods: A survey. *Robotics and Autonomous Systems*, 64:1–20, 2015.

[141] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, 43(1):55–81, 2015.

[142] Edward Johns and Guang-Zhong Yang. Global localization in a dense continuous topological map. In *ICRA*, volume 11, pages 1032–1037, 2011.

[143] Mana Saedan, Chee Wang Lim, and Marcelo H Ang. Appearance-based slam with map loop closing using an omnidirectional camera. In *IEEE/ASME international Conference on Advanced Intelligent Mechatronics*, pages 1–6, 2007.

[144] Hiroshi Morioka, Sangkyu Yi, and Osamu Hasegawa. Vision-based mobile robot's slam and navigation in crowded environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3998–4005, 2011.

[145] Risako Aoki, Shun Aoki, Yakumo Ohtagaki, and Ryusuke Miyamoto. Key point localization for 3D model generation from facial illustrations using surf and color features. In *7th IEEE International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*, pages 55–56, 2017.

[146] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

[147] Mark Cummins and Paul Newman. Appearance-only slam at large scale with fab-map 2.0. *The International Journal of Robotics Research*, 30(9):1100–1123, 2011.

[148] Ananth Ranganathan and Frank Dellaert. Online probabilistic topological mapping. *The International Journal of Robotics Research*, 30(6):755–771, 2011.

[149] Al-Hussein A El-Shafie, Mohamed Zaki, and Serag El-Din Habib. Fast CNN-based object tracking using localization layers and deep features interpolation. *arXiv preprint arXiv:1901.02620*, 2019.

[150] Sathya Narayanan Kasturi Rangan, Veera Ganesh Yalla, Davide Bacchet, and Izzy Domi. Improved localization using visual features and maps for autonomous cars. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 623–629, 2018.

[151] Carlos Guindel, David Martín, and José María Armingol. Traffic scene awareness for intelligent vehicles using convnets and stereo vision. *Robotics and Autonomous Systems*, 112:109–122, 2019.

[152] Francis Baek, Inhae Ha, and Hyoungkwan Kim. Augmented reality system for facility management using image-based indoor localization. *Automation in Construction*, 99:18–26, 2019.

[153] Song Xu, Wusheng Chou, and Hongyi Dong. A robust indoor localization system integrating visual localization aided by CNN-based image retrieval with monte carlo localization. *Sensors*, 19(2):249, 2019.

[154] Shuji Oishi, Yohei Inoue, Jun Miura, and Shota Tanaka. Seqslam++: View-based robot localization and navigation. *Robotics and Autonomous Systems*, 112:13–21, 2019.

[155] Xiaqing Ding, Yue Wang, Dongxuan Li, Li Tang, Huan Yin, and Rong Xiong. Laser map aided visual inertial localization in changing environment. *arXiv preprint arXiv:1803.01104*, 2018.

[156] Maurice Fallon. Accurate and robust localization for walking robots fusing kinematics, inertial, vision and lidar. *Interface Focus*, 8(4):1–9, 2018.

[157] Hao Cai, Zhaozheng Hu, Gang Huang, Dunyao Zhu, and Xiaocong Su. Integration of gps, monocular vision, and high definition (hd) map for accurate vehicle localization. *Sensors*, 18(10):32–70, 2018.

[158] Kichun Jo and Myoungho Sunwoo. Generation of a precise roadway map for autonomous cars. *IEEE Transactions on Intelligent Transportation Systems*, 15(3):925–937, 2013.

[159] Peter Kock, Ralf Weller, Andrzej W Ordys, and Gordana Collier. Validation meth-

ods for digital road maps in predictive control. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):339–351, 2014.

[160] Chunzhao Guo, Jun-ichi Meguro, Yoshiko Kojima, and Takashi Naito. Automatic lane-level map generation for advanced driver assistance systems using low-cost sensors. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3975–3982, 2014.

[161] Haiyan Guan, Jonathan Li, Yongtao Yu, Michael Chapman, and Cheng Wang. Automated road information extraction from mobile laser scanning data. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):194–205, 2014.

[162] Vijay John, Keisuke Yoneda, Zheng Liu, and Seiichi Mita. Saliency map generation by the convolutional neural network for real-time traffic light detection using template matching. *IEEE Transactions on Computational Imaging*, 1(3):159–173, 2015.

[163] Gi-Poong Gwon, Woo-Sol Hur, Seong-Woo Kim, and Seung-Woo Seo. Generation of a precise and efficient lane-level road map for intelligent vehicle systems. *IEEE Transactions on Vehicular Technology*, 66(6):4517–4533, 2016.

[164] Jian Tan, Xiangtao Fan, Shenghua Wang, and Yingchao Ren. Optimization-based wi-fi radio map construction for indoor positioning using only smart phones. *Sensors*, 18(9):3095, 2018.

[165] Ren C Luo and Wei Shih. Topological map generation for intrinsic visual navigation of an intelligent service robot. In *IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–6, 2019.

[166] Karthikeyan U Gunasekaran, Evan Krell, Alaa Sheta, and Scott A King. Map generation and path planning for autonomous mobile robot in static environments using ga. In *8th IEEE International Conference on Computer Science and Information Technology (CSIT)*, pages 91–96, 2018.

[167] Juil Sock, Jun Kim, Jihong Min, and Kiho Kwak. Probabilistic traversability map generation using 3D-lidar and camera. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5631–5637, 2016.

[168] Leandro B Marinho, Jefferson S Almeida, João Wellington M Souza, Victor Hugo C Albuquerque, and Pedro P Rebouças Filho. A novel mobile robot localization approach based on topological maps using classification with reject option in omnidirectional images. *Expert Systems with Applications*, 72:1–17, 2017.

[169] Rafael Peixoto Derenzi Vivacqua, Massimo Bertozzi, Pietro Cerri, Felipe Nascimento Martins, and Raquel Frizera Vassallo. Self-localization based on visual lane marking maps: An accurate low-cost approach for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 19(2):582–597, 2018.

[170] Hernán Badino, D Huber, and Takeo Kanade. Visual topometric localization. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 794–799, 2011.

[171] Henning Lategahn and Christoph Stiller. City gps using stereo vision. In *ICVES*, pages 1–6. Citeseer, 2012.

[172] Jongeun Park, Jin Yong Kim, BaekCheon Kim, and Sungshin Kim. Global map generation using lidar and stereo camera for initial positioning of mobile robot. In *IEEE International Conference on Information and Communication Technology Robotics (ICT-ROBOT)*, pages 1–4, 2018.

[173] Yuquan Xu, Vijay John, Seiichi Mita, Hossein Tehrani, Kazuhisa Ishimaru, and Sakiko Nishino. 3D point cloud map based vehicle localization using stereo camera. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 487–492, 2017.

[174] Lauro J Lyrio, Thiago Oliveira-Santos, Claudine Badue, and Alberto Ferreira De Souza. Image-based mapping, global localization and position tracking using vg-ram weightless neural networks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3603–3610, 2015.

[175] Guilherme N DeSouza and Avinash C Kak. Vision for mobile robot navigation: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 24(2):237–267, 2002.

[176] Qilong Zhang and Robert Pless. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *IROS*, volume 3, pages 2301–2306, 2004.

[177] Baljit Kaur and Jhilik Bhattacharya. Augmented map based traffic density estimation for robot navigation. In *IEEE Region Ten Symposium (Tensymp)*, pages 254–259, 2018.

[178] OpenStreetMap contributors. *https://www.openstreetmap.org/way/395256540/map=16/30.3538/76.3679*, [accessed on 6th january 2019].

[179] Karim Hammoudi, Fadi Dornaika, Bahman Soheilian, Bruno Vallet, John McDonald, and Nicolas Paparoditis. Recovering occlusion-free textured 3D maps of urban facades by a synergistic use of terrestrial images, 3D point clouds and area-based information. *Procedia Engineering*, 41:971–980, 2012.

[180] Adi Sujiwo, Eijiro Takeuchi, Luis Yoichi Morales, Naoki Akai, Yoshiki Ninomiya, and Masato Edahiro. Localization based on multiple visual-metric maps. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 212–219, 2017.

[181] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Andreas Holzbach, and Michael Beetz. Model-based and learned semantic object labeling in 3D point cloud maps of kitchen environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3601–3608, 2009.

[182] Hema S Koppula, Abhishek Anand, Thorsten Joachims, and Ashutosh Saxena. Semantic labeling of 3D point clouds for indoor scenes. In *Advances in Neural Information Processing Systems*, pages 244–252, 2011.

[183] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. In *Computer Graphics Forum*, volume 26, pages 214–226. Wiley Online Library, 2007.

[184] Avelino Forechi, Thiago Oliveira-Santos, Claudine Badue, and Alberto F De Souza. Visual global localization with a hybrid WNN-CNN approach. *arXiv preprint arXiv:1805.03183*, 2018.

[185] Yuuji Ishikoori, Hirokazu Madokoro, and Kazuhito Sato. Semantic position recognition and visual landmark detection with invariant for human effect. In *System Integration (SII), 2017 IEEE/SICE International Symposium on*, pages 657–662. IEEE, 2017.

[186] Joydeep Biswas and Manuela Veloso. Depth camera based indoor mobile robot localization and navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1697–1702, 2012.

[187] Fengkui Cao, Yan Zhuang, Hong Zhang, and Wei Wang. Robust place recognition and loop closing in laser-based slam for ugvs in urban environments. *IEEE Sensors Journal*, 18:4242 – 4252, 2018.

[188] Xiandong Xu, Bingrong Hong, and Yi Guan. Humanoid robot localization based on hybrid map. In *Security, Pattern Analysis, and Cybernetics (SPAC), 2017 International Conference on*, pages 509–514. IEEE, 2017.

[189] Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

[190] Koen Van De Sande, Theo Gevers, and Cees Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010.

[191] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE conference on*, pages 3485–3492, 2010.

[192] Niko Sünderhauf, Sareh Shirazi, Feras Dayoub, Ben Upcroft, and Michael Milford. On the performance of convnet features for place recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 4297–4304, 2015.

[193] Luis Herranz, Shuqiang Jiang, and Xiangyang Li. Scene recognition with CNNs:

objects, scales and dataset bias. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 571–579, 2016.

[194] Niko Sünderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Proceedings of Robotics: Science and Systems XII*, pages 1–10, 2015.

[195] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:1452 – 1464, 2017.

[196] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, pages 487–495, 2014.

[197] Zhi Xu, Guoyong Cai, Yimin Wen, Dongdong Chen, and Liyao Han. Image set-based classification using collaborative exemplars representation. *Signal, Image and Video Processing*, pages 1–9, 2018.

[198] Ruiping Wang, Huimin Guo, Larry S Davis, and Qionghai Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2496–2503, 2012.

[199] Zhiwu Huang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Hybrid euclidean-and-riemannian metric learning for image set classification. In *Asian Conference on Computer Vision*, pages 562–577. Springer, 2014.

[200] Li Hou, Kang Han, Wanggen Wan, Jenq-Neng Hwang, and Haiyan Yao. Normalized distance aggregation of discriminative features for person reidentification. *Journal of Electronic Imaging*, 27(2):023006, 2018.

[201] Yu Li, Meiyu Qian, Pengfeng Liu, Qian Cai, Xiaoying Li, Junwen Guo, Huan Yan, Fengyuan Yu, Kun Yuan, Juan Yu, et al. The recognition of rice images by uav based on capsule network. *Cluster Computing*, pages 1–10, 2018.

[202] Parnian Afshar, Arash Mohammadi, and Konstantinos N Plataniotis. Brain tumor type classification via capsule networks. *arXiv preprint arXiv:1802.10200*, 2018.

[203] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *Advances in Neural Information Processing Systems*, pages 1269–1277, 2014.

[204] Nicholas D Lane, Sourav Bhattacharya, Petko Georgiev, Claudio Forlivesi, Lei Jiao, Lorena Qendro, and Fahim Kawsar. Deepx: A software accelerator for low-power deep learning inference on mobile devices. In *Information Processing in Sensor Networks (IPSN), 2016 15th ACM/IEEE International Conference on*, pages 1–12,

2016.

[205] Bappaditya Chakraborty, Bikash Shaw, Jayanta Aich, Ujjwal Bhattacharya, and Swapan Kumar Parui. Does deeper network lead to better accuracy: A case study on handwritten devanagari characters. In *13th IEEE, IAPR International Workshop on Document Analysis Systems (DAS)*, pages 411–416, 2018.

[206] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[207] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3859–3869, 2017.

[208] Grigorios Kalliatakis. Keras-vgg16-places365. `https://github.com/GKalliatakis/Keras-VGG16-places365`, 2017.

[209] Baljit Kaur and Jhilik Bhattacharya. A convolutional feature map-based deep network targeted towards traffic detection and classification. *Expert Systems with Applications*, 124:119 – 129, 2019.

[210] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[211] Ipek Baris and Yalin Bastanlar. Classification and tracking of traffic scene objects with hybrid camera systems. In *20th IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2017.

[212] Ivana Shopovska, Ljubomir Jovanov, Peter Veelaert, Wilfried Philips, Merwan Birem, and Kris Lehaen. A hybrid fusion based frontal-lateral collaborative pedestrian detection and tracking. In *20th IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2017.

[213] Carsten Fries and Hans-Joachim Wuensche. Monocular template-based vehicle tracking for autonomous convoy driving. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2727–2732, 2014.

[214] Chaowei Hu, Yunpeng Wang, Guizhen Yu, Zhangyu Wang, Ao Lei, and Zhehua Hu. Embedding CNN-based fast obstacles detection for autonomous vehicles. Technical report, SAE Technical Paper, 2018.

[215] Khaled Saleh, Mohammed Hossny, Ahmed Hossny, and Saeid Nahavandi. Cyclist detection in LIDAR scans using faster R-CNN and synthetic depth images. In *20th IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2017.

[216] Fernando García, A Prioletti, P Cerri, and A Broggi. Phd filter for vehicle tracking based on a monocular camera. *Expert Systems with Applications*, 91:472–479, 2018.

[217] Damien Matti, Hazım Kemal Ekenel, and Jean-Philippe Thiran. Combining lidar space clustering and convolutional neural networks for pedestrian detection. *arXiv preprint arXiv:1710.06160*, 2017.

[218] Jong Hyun Kim, Ganbayar Batchuluun, and Kang Ryoung Park. Pedestrian detection based on faster R-CNN in nighttime by fusing deep convolutional features of successive images. *Expert Systems with Applications*, 114:15–33, 2018.

[219] Igor Vasiljevic, Ayan Chakrabarti, and Gregory Shakhnarovich. Examining the impact of blur on recognition by convolutional networks. *arXiv preprint arXiv:1611.05760*, 2016.

[220] Li Zhuo, Liying Jiang, Ziqi Zhu, Jiafeng Li, Jing Zhang, and Haixia Long. Vehicle classification for large-scale traffic surveillance videos using convolutional neural networks. *Machine Vision and Applications*, pages 1–10, 2017.

[221] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia. Optimizing deep CNN-based queries over video streams at scale. *arXiv preprint arXiv:1703.02529*, 2017.

[222] Xiaofei Li, Lingxi Li, Fabian Flohr, Jianqiang Wang, Hui Xiong, Morys Bernhard, Shuyue Pan, Dariu M Gavrila, and Keqiang Li. A unified framework for concurrent pedestrian and cyclist detection. *IEEE transactions on Intelligent Transportation Systems*, 18(2):269–281, 2017.

[223] Woonhyun Nam, Piotr Dollár, and Joon Hee Han. Local decorrelation for improved pedestrian detection. In *Advances in Neural Information Processing Systems*, pages 424–432, 2014.

[224] Zhuoqun Huo, Yizhang Xia, and Bailing Zhang. Vehicle type classification and attribute prediction using multi-task rcnn. In *IEEE International Conference on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 564–569, 2016.

[225] Zhi Zhang, Zhihai He, Guitao Cao, and Wenming Cao. Animal detection from highly cluttered natural scenes using spatiotemporal object region proposals and patch verification. *IEEE Transactions on Multimedia*, 18(10):2079–2092, 2016.

[226] Yanjie Yao, Bin Tian, and Fei-Yue Wang. Coupled multivehicle detection and classification with prior objectness measure. *IEEE Transactions on Vehicular Technology*, 66(3):1975–1984, 2017.

[227] Shu Wang, Feng Liu, Zongliang Gan, and Ziguan Cui. Vehicle type classification via adaptive feature clustering for traffic surveillance video. In *8th IEEE International Conference on Wireless Communications & Signal Processing (WCSP)*, pages 1–5, 2016.

[228] Xiangyu Zhang, Jianhua Zou, Kaiming He, and Jian Sun. Accelerating very deep

convolutional networks for classification and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):1943–1955, 2016.

[229] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, et al. T-CNN: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 2896 – 2907, 2017.

[230] Jianan Li, Yunchao Wei, Xiaodan Liang, Jian Dong, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. Attentive contexts for object detection. *IEEE Transactions on Multimedia*, 19(5):944–954, 2017.

[231] Hyun Soo Park, Dae Jung Kim, Chang Mook Kang, Seok Cheol Kee, and Chung Choo Chung. Object detection in adaptive cruise control using multi-class support vector machine. In *20th IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2017.

[232] Yongkun Fang, Chao Wang, Huijing Zhao, and Hongbin Zha. On-road vehicle tracking using part-based particle filter. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3755–3761, 2017.

[233] Friederike Schneemann and Patrick Heinemann. Context-based detection of pedestrian crossing intention for autonomous driving in urban environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2243–2248, 2016.

[234] Jun Ding, Bo Chen, Hongwei Liu, and Mengyuan Huang. Convolutional neural network with data augmentation for sar target recognition. *IEEE Geoscience and remote sensing letters*, 13(3):364–368, 2016.

[235] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3D lidar using fully convolutional network. *arXiv preprint arXiv:1608.07916*, 2016.

[236] QIAOJIN GUO, ZHONGYAN LIANG, and JIE HU. Vehicle classification with convolutional neural network on motion blurred images. *DEStech Transactions on Computer Science and Engineering*, pages 40–45, 2017.

[237] Yann Dauphin, Harm de Vries, and Yoshua Bengio. Equilibrated adaptive learning rates for non-convex optimization. In *Advances in Neural Information Processing Systems*, pages 1504–1512, 2015.

[238] Yasutoshi Ida, Yasuhiro Fujiwara, and Sotetsu Iwamura. Adaptive learning rate via covariance matrix based preconditioning for deep neural networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1923–1929. AAAI Press, 2017.

[239] U Bhattacharya and SK Parui. Self-adaptive learning rates in backpropagation algorithm improve its function approximation performance. In *Proceedings of ICNN-*

International Conference on Neural Networks, volume 5, pages 2784–2788, 1995.

[240] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[241] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[242] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[243] Yuanzhouhan Cao, Chunhua Shen, and Heng Tao Shen. Exploiting depth from single monocular images for object detection and semantic segmentation. *IEEE Transactions on Image Processing*, 26(2):836–846, 2017.

[244] R Niessner, H Schilling, and B Jutzi. Investigations on the potential of convolutional neural networks for vehicle classification based on rgb and lidar data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4:115, 2017.

[245] Rabia Rauf, Ahmad R Shahid, Sheikh Ziauddin, and Asad Ali Safi. Pedestrian detection using hog, luv and optical flow as features with adaboost as classifier. In *6th IEEE International Conference on Image Processing Theory Tools and Applications (IPTA)*, pages 1–4, 2016.

[246] Danut Ovidiu Pop, Alexandrina Rogozan, Fawzi Nashashibi, and Abdelaziz Bensrhair. Incremental cross-modality deep learning for pedestrian recognition. In *IEEE Intelligent Vehicles Symposium*, 2017.

[247] Soumik Sarkar, Vivek Venugopalan, Kishore Reddy, Julian Ryde, Navdeep Jaitly, and Michael Giering. Deep learning for automated occlusion edge detection in rgb-d frames. *Journal of Signal Processing Systems*, 88(2):205–217, 2017.

[248] Peter Muller and Andreas Savakis. Flowdometry: An optical flow and deep learning based approach to visual odometry. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 624–631, 2017.

[249] Francesca Dardi, Leonardo Abate, Jeroen Stessen, and Giovanni Ramponi. Causes and subjective evaluation of blurriness in video frames. *Signal Processing: Image Communication*, 28(3):209–221, 2013.

[250] Hang Yin and Christian Berger. When to use what data set for your self-driving car algorithm: An overview of publicly available driving datasets. In *20th IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8,

2017.

[251] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *PAMI*, 34, 2012.

[252] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[253] B Srinivasa Reddy and Biswanath N Chatterji. An fft-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing*, 5(8):1266–1271, 1996.

[254] Zhen Gao, Guoliang Lu, and Peng Yan. Key-frame selection for video summarization: an approach of multidimensional time series analysis. *Multidimensional Systems and Signal Processing*, pages 1–21, 2017.

[255] Kanika Lakhani, Bhawna Minocha, and Neeraj Gugnani. Analyzing edge detection techniques for feature extraction in dental radiographs. *Perspectives in Science*, 8:395–398, 2016.

[256] Mohamed Ali and David Clausi. Using the canny edge detector for feature extraction and enhancement of remote sensing images. In *IEEE International Conference on Geoscience and Remote Sensing Symposium (IGARSS)*, volume 5, pages 2298–2300, 2001.

[257] Shuyang Sun, Zhanghui Kuang, Lu Sheng, Wanli Ouyang, and Wei Zhang. Optical flow guided feature: a fast and robust motion representation for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1390–1399, 2018.

[258] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015.

[259] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, pages 1–4, 2017.

[260] Mandeep Singh, Sukhwinder Singh, and Savita Gupta. An information fusion based method for liver classification using texture analysis of ultrasound images. *Information Fusion*, 19:91–96, 2014.

[261] Chandan Singh, Ekta Walia, and Kanwal Preet Kaur. Enhancing color image retrieval performance with feature fusion and non-linear support vector machine classifier. *Optik*, 158:127–141, 2018.

[262] Jingjing Liu, Shaoting Zhang, Shu Wang, and Dimitris N Metaxas. Multispectral

deep neural networks for pedestrian detection. *arXiv preprint arXiv:1611.02644*, 2016.

[263] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[264] Dengsheng Zhang, M Monirul Islam, Guojun Lu, and Ishrat Jahan Sumana. Rotation invariant curvelet features for region based image retrieval. *International Journal of Computer Vision*, 98(2):187–201, 2012.

[265] Debapratim Das Dawn and Soharab Hossain Shaikh. A comprehensive survey of human action recognition with spatio-temporal interest point (stip) detector. *The Visual Computer*, 32(3):289–306, 2016.

[266] Jiao Tian and Derek Molloy. 3D corner detection and matching for manmade scene/object structure cognition. In *VISAPP (1)*, pages 477–480, 2013.

[267] Paul F. Whelan Dana E. Ilean. Image segmentation based on the integration of colourtexturedescriptorsa review. *Pattern Recognition*, 44:24792501, 2011.

[268] Ann B Lee, Boaz Nadler, and Larry Wasserman. Treelets: an adaptive multi-scale basis for sparse unordered data. *The Annals of Applied Statistics*, pages 435–471, 2008.

[269] Marc Peter Deisenroth, Dieter Fox, and Carl Edward Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):408–423, 2015.

[270] Sergio Carrato, Gianfranco Fenu, Eric Medvet, Enzo Mumolo, Felice Andrea Pellegrino, and Giovanni Ramponi. Towards more natural social interactions of visually impaired persons. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 729–740. Springer, 2015.

[271] Ruxandra Tapu, Bogdan Mocanu, Andrei Bursuc, and Titus Zaharia. A smartphone-based obstacle detection and classification system for assisting visually impaired people. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 444–451, 2013.

[272] Aravinda S Rao, Jayavardhana Gubbi, Marimuthu Palaniswami, and Elaine Wong. A vision-based system to detect potholes and uneven surfaces for assisting blind people. In *IEEE International Conference on Communications (ICC)*, pages 1–6, 2016.

[273] Nadia Kanwal, Erkan Bostanci, Keith Currie, and Adrian F Clark. A navigation system for the visually impaired: a fusion of vision and depth sensor. *Applied Bionics and Biomechanics*, pages 1–16, 2015.

[274] Aitor Aladren, Gonzalo López-Nicolás, Luis Puig, and Josechu J Guerrero. Navigation assistance for the visually impaired using rgb-d sensor with range expansion.

IEEE Systems Journal, 10(3):922–932, 2016.

[275] M Sarfraz and SM Ali J Rizvi. Indoor navigational aid system for the visually impaired. In *IEEE conference on Geometric Modeling and Imaging (GMAI)*, pages 127–132, 2007.

[276] Rabia Jafri, Rodrigo Louzada Campos, Syed Abid Ali, and Hamid R Arabnia. Visual and infrared sensor data-based obstacle detection for the visually impaired using the google project tango tablet development kit and the unity engine. *IEEE Access*, 6:443–454, 2018.

[277] Bin Jiang, Jiachen Yang, Zhihan Lv, and Houbing Song. Wearable vision assistance system based on binocular sensors for visually impaired users. *IEEE Internet of Things Journal*, 6:1375–133, 2018.

[278] Bing Li, Juan Pablo Munoz, Xuejian Rong, Qingtian Chen, Jizhong Xiao, Yingli Tian, Aries Arditi, and Mohammed Yousuf. Vision-based mobile indoor assistive navigation aid for blind people. *IEEE Transactions on Mobile Computing*, 18:702–714, 2018.

[279] Monica Gori, Giulia Cappagli, Alessia Tonelli, Gabriel Baud-Bovy, and Sara Finocchietti. Devices for visually impaired people: High technological devices with low user acceptance and no adaptability for children. *Neuroscience & Biobehavioral Reviews*, 69:79–88, 2016.

[280] Laurindo Britto Neto, Felipe Grijalva, Vanessa Regina Margareth Lima Maike, Luiz Cesar Martini, Dinei Florencio, Maria Cecilia Calani Baranauskas, Anderson Rocha, and Siome Goldenstein. A kinect-based wearable face recognition system to aid visually impaired users. *IEEE Transactions on Human-Machine Systems*, 47(1):52–64, 2017.

[281] Baljit Kaur and Jhilik Bhattacharya. Scene perception system for visually impaired based on object detection and classification using multimodal deep convolutional neural network. *Journal of Electronic Imaging*, 28(1):013031, 2019.

[282] B Kaur and J Bhattacharya. Predictive hierarchical human augmented map generation for itinerary perception. *Electronics Letters*, 52(16):1381–1383, 2016.

[283] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *Advances in neural information processing systems*, pages 1324–1332, 2010.

[284] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[285] Di Kang, Zheng Ma, and Antoni B Chan. Beyond counting: Comparisons of density maps for crowd analysis tasks-counting, detection, and tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 29:1408–1422, 2018.

[286] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–546, 2018.

[287] Carl Edward Rasmussen. Gaussian processes for machine learning. 2006.

[288] Elly Matul Imah, Atik Wintarti, R Sulaiman, and Manuharawati Manuharawati. Automation animal tracker using complex value neural network. In *MATEC Web of Conferences*, volume 197, page 03020. EDP Sciences, 2018.

[289] Tom Van Hertem, Andrés Schlageter Tello, Stefano Viazzi, Machteld Steensels, Claudia Bahr, Carlos Eduardo Bites Romanini, Kees Lokhorst, Ephraim Maltz, Ilan Halachmi, and Daniel Berckmans. Implementation of an automatic 3D vision monitor for dairy cow locomotion in a commercial farm. *Biosystems Engineering*, 173:166–175, 2018.

[290] Abozar Nasirahmadi, Sandra A Edwards, and Barbara Sturm. Implementation of machine vision for detecting behaviour of cattle and pigs. *Livestock Science*, 202:25–38, 2017.

[291] Oleksiy Guzhva, Håkan Ardö, Mikael Nilsson, Anders Herlin, and Linda Tufvesson. Now you see me: Convolutional neural network based tracker for dairy cows. *Frontiers in Robotics and AI*, 5:107, 2018.

[292] Nicholas Zinas, Sotirios Kontogiannis, George Kokkonis, Stavros Valsamidis, and Ioannis Kazanidis. Proposed open source architecture for long range monitoring. the case study of cattle tracking at pogoniani. In *Proceedings of the 21st Pan-Hellenic Conference on Informatics*, page 57. ACM, 2017.

[293] Trung-Kien Dao, Thi-Lan Le, David Harle, Paul Murray, Christos Tachtatzis, Stephen Marshall, Craig Michie, and Ivan Andonovic. Automatic cattle location tracking using image processing. In *23rd IEEE European Signal Processing Conference (EUSIPCO)*, pages 2636–2640, 2015.

[294] Alpha Daye Diallo, Suresh Gobee, and Vickneswari Durairajah. Autonomous tour guide robot using embedded system control. *Procedia Computer Science*, 76:126–133, 2015.

[295] Asraa Al-Wazzan, Rawan Al-Farhan, Farah Al-Ali, and Mohammed El-Abd. Tour-guide robot. In *IEEE International Conference on Industrial Informatics and Computer Systems (CIICS)*, pages 1–5, 2016.

[296] Abir Bellarbi, Souhila Kahlouche, Nouara Achour, and Noureddine Ouadah. A social planning and navigation for tour-guide robot in human environment. In *8th IEEE International Conference on Modelling, Identification and Control (ICMIC)*, pages 622–627, 2016.

[297] Biel Piero E Alvarado Vasquez, Ruben Gonzalez, Fernando Matia, and Paloma De La Puente. Sensor fusion for tour-guide robot localization. *IEEE Access*, 6:78947–78964, 2018.

[298] Shengye Wang and Henrik I Christensen. Tritonbot: First lessons learned from deployment of a long-term autonomy tour guide robot. In *27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 158–165, 2018.

[299] Dipti Pawade, Avani Sakhapara, Maheshwar Mundhe, Aniruddha Kamath, and Devansh Dave. Augmented reality based campus guide application using feature points object detection. 5:76–85, 2018.

[300] H-M Gross, Alexander Koenig, H-J Boehme, and Ch Schroeter. Vision-based monte carlo self-localization for a mobile service robot acting as shopping assistant in a home store. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 256–262, 2002.

[301] Amy J Briggs, Carrick Detweiler, Daniel Scharstein, and Alexander Vandenberg-Rodes. Expected shortest paths for landmark-based robot navigation. *The International Journal of Robotics Research*, 23(7-8):717–728, 2004.

[302] Hiroshi Ishida, Hidenao Tanaka, Haruki Taniguchi, and Toyosaka Moriizumi. Mobile robot navigation using vision and olfaction to search for a gas/odor source. *Autonomous Robots*, 20(3):231–238, 2006.

[303] Wesley H Huang, Brett R Fajen, Jonathan R Fink, and William H Warren. Visual navigation and obstacle avoidance using a steering potential function. *Robotics and Autonomous Systems*, 54(4):288–299, 2006.

[304] Brett R Fajen, William H Warren, Selim Temizer, and Leslie Pack Kaelbling. A dynamical model of visually-guided steering, obstacle avoidance, and route selection. *International Journal of Computer Vision*, 54(1-3):13–34, 2003.

[305] Amy Briggs, Yunpeng Li, Daniel Scharstein, and Matt Wilder. Robot navigation using 1d panoramic images. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 2679–2685, 2006.

[306] Kahlouche Souhila and Achour Karim. Optical flow based robot obstacle avoidance. *International Journal of Advanced Robotic Systems*, 4(1):2, 2007.

[307] Jin Zhou and Baoxin Li. Exploiting vertical lines in vision-based navigation for mobile robot platforms. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 1–465, 2007.

[308] D Santosh, Supreeth Achar, and CV Jawahar. Autonomous image-based exploration for mobile robot navigation. In *IEEE International Conference on Robotics and Automation*, pages 2717–2722, 2008.

[309] N Nirmal Singh, Avishek Chatterjee, Amitava Chatterjee, and Anjan Rakshit. A two-layered subgoal based mobile robot navigation algorithm with vision system and ir sensors. *Measurement*, 44(4):620–641, 2011.

[310] Ji Zhang, George Kantor, Marcel Bergerman, and Sanjiv Singh. Monocular visual navigation of an autonomous vehicle in natural scene corridor-like environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3659–3666, 2012.

[311] Naoya Ohnishi and Atsushi Imiya. Appearance-based navigation and homing for autonomous mobile robot. *Image and Vision Computing*, 31(6-7):511–532, 2013.

[312] Yudin Dmitriy Aleksandrovich, Postolsky Grigoriy Gennadievich, Kizhuk Alexander Stepanovich, and Magergut Valeriy Zalmanovich. Mobile robot navigation based on artificial landmarks with machine vision system. *World Applied Sciences Journal*, 24(11):1467–1472, 2013.

[313] Julio Delgado-Galvan, Alberto Navarro-Ramirez, Jose Nunez-Varela, Cesar Puente-Montejano, and Francisco Martinez-Perez. Vision-based humanoid robot navigation in a featureless environment. In *Mexican Conference on Pattern Recognition*, pages 169–178. Springer, 2015.

[314] Chris J Ostafew, Angela P Schoellig, Timothy D Barfoot, and Jack Collier. Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking. *Journal of Field Robotics*, 33(1):133–152, 2016.

[315] Yan Lu and Dezhen Song. Visual navigation using heterogeneous landmarks and unsupervised geometric constraints. *IEEE Transactions on Robotics*, 31(3):736–749, 2015.

[316] Fabian Blochliger, Marius Fehr, Marcin Dymczyk, Thomas Schneider, and Rol Siegwart. Topomap: Topological mapping and navigation based on visual slam maps. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9, 2018.

[317] Muhammad Raees, Sehat Ullah, and Sami Ur Rahman. VEN-3DVE: vision based egocentric navigation for 3D virtual environments. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 13(1):35–45, 2019.

[318] Mohamed Aladem, Stanley Baek, and Samir A Rawashdeh. Evaluation of image enhancement techniques for vision-based navigation under low illumination. *Journal of Robotics*, pages 1–16, 2019.

Appendix

Table 6.1: Brief of navigation systems using vision sensors

Ref Year	Applications	Hardware	Algorithm	pros	Cons	Working	Assumption
[64] 2002	Autonomous navigation	3 wheeled electric scooter	Map generation, Robot localization, Shortest path generation	Also select shortest path for navigation	Only for fixed environment	Robot utilized the map and the past experience to select viewing directions.	
[300] 2002	Shopping Assistant in a Home Store	PERSES	Monte Carlo Localization, Bayesian filtering technique	Reliable, robust	Area is fixed	"Presented particle filters in combination with a graph-based representation of the operation area by local panoramic views, performed navigation in a home store with a maze-like topology"	During training, the robot's original location and direction of heading were known.
[28] 2004	Robot navigation	Not mentioned	Monte Carlo Localization technique	Self-organise its visual memory	Not for outdoor environment	Created a topological map and organises into robot's visual memory. Proposed a method in which every image is represented by the components of its Fourier transform. Also defined a similarity function that can assess the degree of similarity between two images using the Fourier signatures.	
[301] 2004	Landmark based robot navigation	Not mentioned	Dijkstras shortest-path algorithm, breadth-first search algorithm	Fast and reliable	Not used Natural landmarks, navigation not done in arbitrary environments.	"To construct robust and efficient navigation plans, formulated the expected shortest paths problem as a Markov decision process"	The robot could only travel to landmarks that were visible.
[55] 2005	Outdoor autonomous navigation	Electric vehicle: Cycab	Harris corner detector, Grunerts method with RANSAC	Works well even in area where there are obstacles that prevents from satellite receiving	Robot have to stay on reference trajectory	A 3D reconstruction of the learning video sequence was built followed by real time localization. From the video sequences, map was built which was further used for localization.	
[52] 2005	Indoor navigation	Home service robot: ISSAC	Adaptive Canny operator and Lucas-Kanade algorithm	Proposed algorithm can be applied for real-time application	Not for outdoor environment	"Proposed a novel visual navigation method by combining visual localization with the extraction of valid planar region"	"Ground floor is flat and the image plane is perpendicular to the ground floor"
[46] 2005	Robot homing	LEFKOS, the RWI B21r mobile robot	KLT algorithm	By exploiting very easy sensory data, a complicated conduct such as homing is accomplished.	Only for homing. Can not go to other locations.	Proposed method was based on tracking image corners in panoramic views of the environment. "By memorizing and processing the life-cycle of the tracked corners, the robot was able to define area that should be revisited sequentially to achieve homing"	The robot mounted with a panoramic camera, "tracks visual features in panoramic views of the surroundings that it acquires while moving"
[48] 2005	Robot localization and navigation	Simulation	Feature tracking and 3D reconstruction	Less cost	Occlusion has not been handled	The landmark based localization was performed as each new image was acquired at each subsequent robot position. The method consists of feature tracking, reconstruction, upgrading to true coordinates, propagating the landmarks and the new robot position.	"The initial pose of the robot is known" and five or more landmarks could be identified.
[24] 2006	Navigation of a museum guide robot	Museum guide robot ATLAS	ENZCC	The robot was designed to take the homing action in low batteries & Visitor-free environment.	Only for museum kind of environment.	Multi-sensor based people detection and navigation have been implemented. The fast visual appearance based navigation, EZNCC, was also developed.	

Table 6.1 continued from previous page

Ref Year	Applications	Hardware	Algorithm	pros	Cons	Working	Assumption
[302] 2006	Searching for a source of gas / odor	Assembled Mobile robot	Algorithm 1: Cooperation of vision and olfaction. Algorithm 2: Cooperation of vision, olfaction, and airflow sensing	Proposed navigation strategies were able to accomplish the source-localization even if the near-floor gas concentration is too small to detect.	Take too much time	The robot firstly looked for a visible object that was placed on the ground and dropped to verify the object with the gas sensor. Secondly, when gas was detected, the robot turned toward the upwind direction and looked for any visible object.	The floor is flat and the object is on the floor.
[303] 2006	Navigation for non-holonomic robots	MagellanPro robot	Modified Fajen and Warrens model [304]	"Effective for real-time navigation and obstacle avoidance"	"Angular width of obstacles and their distance was not used"	Approach used the relative headings to the goal and to obstacles, "the distance to the target and angular width of obstacles to calculate a potential field over the heading of the robot"	Obstacles were circular, goal heading and distance were known
[305] 2006	Navigation	Not mentioned	Feature extraction and feature matching, Dijkstras algorithm	Robust to occlusion, repeating patterns and lighting variations.	Not tried for outdoor environment.	The proposed technique used characteristics of scale-space and used circular dynamic programming to match the image. The decreased dimensionality enabled thick sampling of opinions of reference and real-time views analysis.	
[38] 2006	Landmark based Navigation	Simulator	SIFT, markovs chain	Experimented on both synthetic and real data	Environmental change was not handled	"Presented a novel graph theoretic formulation of the problem" of automatically extracting an optimal set of landmarks from an environment for visual navigation.	
[40] 2007	Mobile robot navigation	Nomadic XR4000	Canny-Deriche edge detector, landmark recognition algo	Method remains efficient despite ambient brightness variations or viewing changes.	Only for indoor environment	A method for extracting quadrangles in open cluttered and corridor-like spaces. These quadrangles can correspond to planar objects.	Landmarks have to be easily detected in the image signal, they can be locally characterized to distinguish them from others.
[306] 2007	Visual obstacle avoidance of autonomous mobile robot	RWI-B21r	Horn and Schunck technique, navigation algorithm	No cost of other sensors	Accuracy is very less	"Detected the presence of objects close to the robot based on the information of the movement of the image brightness"	
[56] 2007	Navigation using trajectory	CyCab	Grunerts method with RANSAC	"Reduced the cost and size of the localization system"	Map of the environment need to be modified	Showed that autonomous navigation is possible in outdoor situation with the use of a single camera and natural landmarks applying three step approach.	Ground is locally planar and horizontal at the current position of the robot.
[307] 2007	Robot navigation	Not mentioned	Hough transformation, normalized-homography-based ground plane detection approach	Reasonably well even in the presence of inaccuracies of the vertical line detection	Lack of prior calibration	Proposed an approach to estimate the camera orientation based on detected vertical lines. Further used the estimated camera orientation in one key problem of mobile robot navigation: ground/obstacle detection	Mobile robot is navigating on a plane
[30] 2008	Visual Servoing	CyCab	A Lyapunov-based stability analysis	"A stability analysis of nonholonomic image-based path following is carried out"	Only for fixed targets	Due to a switching strategy between two primitive controllers, the method can be used in general initial conditions and a Lyapunov-based stability assessment was performed.	
[308] 2008	Navigation	Differential drive robot	Planning and servoing algorithms, exploration algorithm	Approach facilitated the robot to autonomously expand its workspace and memorise newly discovered information	Not for outdoor environments	Proposed algorithm that detected the frontier boundaries from the images and utilised them to explore the unknown regions of the environment. A topological graph was used for modelling the explored workspace.	No overhanging obstacles, region directly in front of the robot was open space

Table 6.1 continued from previous page

Ref Year	Applications	Hardware	Algorithm	pros	Cons	Working	Assumption
[32] 2009	Autonomous navigation in urban areas	RobuCab	RANSAC, Dijkstras algorithm	The navigation strategy is robust to changes between the learning and autonomous navigation steps.	Only for urban areas	Presented a framework for autonomous navigation that enables a vehicle to follow a visual path obtained during a learning stage using a single camera and natural landmarks.	Memorizing key views along the performed path to use the references as checkpoints for future navigation missions.
[39] 2009	Robot Navigation in a Corridor	Not mentioned	Hough transform, Kalman filter, Canny edge detector		Done for only straight corridor	Incorporated computer vision into mobile robot localization which included the generation of localization information from raw images and its fusion with the odometric pose estimation.	The mobile robot has a single camera mounted on it without any ability to turn or move w.r.t the robot itself. The active wheels of the robot also have encoders available for dead reckoning
[33] 2010	Indoor robot navigation	TateRob	Neural network	Performance is good	Only for indoor environments	Convert the captured image to a binary one that was used as the neural controller input after the partition. The neural control system was developed online using real robots, which maps visual data to motor commands.	
[29] 2010	Omnidirectional vision-based robot navigation.	WowWeeTMs Rovio	Vision algorithm	Robot could successfully navigate through building environment and finds the goal point within a reasonable amount of time.	Only for indoor environment	The proposed algorithm detected red or green cones from the images captured by Rovios camera and employed those to find goal state.	Intermediate path positions were also provided with red cones, which were assumed to be visible from each other
[26] 2010	Robot navigation	Small rover fitted with wheel encoders, an inertial measuring unit at low cost, an omnidirectional camera rig, and a laser range scanner.	Rao-Blackwellized particle filter algorithm.	Successful navigation through dynamic and unknown environments	Specific environment not mentioned	The technique constructed a topological representation of the route of the robot. The technique located the robot in the graph after revision and supplied coarse yet robust navigation advice in the body frame of the robot	The robot is able to recognize directional orders and has a fundamental ability to avoid reactive obstacles.
[309] 2011	Robot navigation	KOALA robot	Dijkstras algo, navigation algo	Satisfying results, for static environments as well as dynamic environments.	Not for outdoor environment	Hybridized the shortest path algorithm with camera-based image processing to improve the quality of vision-based mobile robots navigation in the true globe, enabling the robot to achieve its objective by pursuing the shortest practical route to avoid obstacles	The surface is uniform
[49] 2011	Robot navigation	Simulator VRML 97, Micro mouse bot	Horn-Schunck method	Used optical flow to avoid obstacles in autonomous robot navigation.	"The camera view did not capture the objects on the two sides"	The video stream captured through a virtual camera as seen by the robot was used to calculate the optical flow to determine the direction and the speed of the robot for the next step.	Neither the robot nor the obstacles are scale invarient
[27] 2011	Robot navigation and map building	Custom mobile robot	Proposed navigation and map building rule	The system is low-cost, flexible and capable of building a 3-D map in a laboratory environment	Fixed environment	Map building was performed by reconstructing the projection of the line from the actively-controlled laser projector. By exploring within the field of view of a ceiling mounted CCD camera, the local map was built by the onboard laser-vision system	The world frame was assumed to be aligned with initial base frame of the mobile robot

Table 6.1 continued from previous page

Ref Year	Applications	Hardware	Algorithm	pros	Cons	Working	Assumption
[31] 2012	Outdoor navigation	CyCab	Grunerts method with RANSAC	Outdoor visual navigation with obstacle avoidance has been carried out in urban environments	Occlusion problem, moving obstacles cannot be handeled	The camera detects the features required for navigating, while the scanner senses the obstacles.	Only static obstacles are there.
[310] 2012	Driving a vehicle autonomously along the tree rows	Toro MDE eWorkman electric vehicle	Extended Kalman filter.	The robot drives on off-road irregular terrains"	Only tree row for navigation	"A monocular visual navigation approach is developed"	The robot followed the Ackermann steering model, the driving speed was known and the camera followed the pinhole camera model
[66] 2012	Visually impaired	Not mentioned	Horn and Shunck algorithm	Useful for visually impaired	No real time implementation	Evaluated the feasibility of utilizing optical flow estimation methods for the purpose of providing a feedback mechanism using auditory and tactile mechanisms.	Local brightness pattern is invariant over short time intervals
[58] 2012	Autonomous mobile robot navigation	RC servo robot	Hybrid-adaptation-strategy-based approach (HASBA), PSOBA method	Design have been implemented for robot navigation in unknown environments.	Comparison with existing methods have not been done	"Utilized stable adaptive fuzzy tracking controllers designed by Lyapunov theory and PSO-based hybrid methodologies"	"The reference point lies at the midpoint of the two drive wheels"
[311] 2013	Robot navigation	Not mentioned	Guidance algorithm	Allowed the mobile robot to return from the destination to the initial position		The direction to the target is given at the robot's original position. Without collision with obstacles, the robot dynamically chooses a local route to the target. The destination guidance algorithm enables the mobile robot to return to the original position from the destination	The ground plane is the planar area, the camera mounted on the mobile robot looks downward, the robot observes the world using the camera mounted on itself for navigation, the robot camera captures a sequence of images since the robot moves and the planar area occupies more than 1/2 of the image.
[312] 2013	Warehouse automated guided vehicles, service robots, security robots, mobile robots operating in hostile and dangerous to human health environments	Wheeled mobile robot	Hough transform	Flexibility to change the route, undemanding to the quality of flooring, low equipment cost and ease of installation	Accuracy was not shown	Developed a machine vision system (MVS), which identifies artificial landmarks on images from a video camera with pan-tilt mechanism and allows to calculate the deviation of the robot from the set course.	
[63] 2014	"Guide a robot along the crop rows and weed between crops"	"John Deere Gator TE electric utility vehicle"	Row tracking algorithm	"The method is able to track crop rows across fields with extremely varied appearance during day and night"	Only for agricultural purpose	This technique operates by extracting and monitoring the direction and lateral offset of the dominant parallel texture in a Simulated overhead perspective of the scene and thus abstracting crop-specific information such as color, spacing and periodicity.	The method assumed crops were planted in approximately straight rows on reasonably flat ground;
[57] 2014	Outdoor navigation	CyCab	Kalman filter, Markov model		Not for indoor environments	"Merged a reactive tentacle-based technique with visual servoing to guarantee path following, obstacle bypassing, and collision avoidance by deceleration"	"All objects are rigid, the translational velocities of all points on an object are identical and equal to that of its centroid"

Table 6.1 continued from previous page

Ref Year	Applications	Hardware	Algorithm	pros	Cons	Working	Assumption
[60] 2014	Robot navigation in urban environment	Electrical carlike robot from the ROBOTEX	Image-Based Visual Servoing, Dynamic window approach, RANSAC	For real time environments	Fixed obstacles were considered	"Combined an Image-Based Visual Servoing with an Image-Based Dynamic Window Approach in a hybrid controller called VS+IDWA"	
[37] 2015	Indoor navigation for visually impaired	Wearable device, smartphone	RANSAC, real-time navigation algorithms	Fast and absolute	Real time testing is not done	The navigation algorithm performs real-time 6-DOF feature based visual odometry using a glass-mounted RGBD camera as an input device. It also builds a 3D voxel map of the environment and analyzes 3D traversability	Blind was assumed to move on the ground plane.
[47] 2015	Indoor and outdoor navigation	Caterpillar-tracked robot (ExaBot) A four-wheeledP3-AT from Mobile Robots	Combination of segmentation and feature-based navigation algorithm	Method is robust and easy to implement and does not require sensor calibration or structured environment, and its computational complexity is independent of the environment size.	Night time is not considered	The technique allowed the use of navigation based on both segmentation and feature. A topological map of the setting was described in which the edges were navigable routes and the nodes were open spaces.	
[313] 2015	To navigate in featureless environment	NAO humanoid robot	Watershed algorithm, k-means algorithm, wall follower algorithm	Reliable	Distance from walls are not found	A topological vision based approach for navigating through a featureless maze-like environment	Environment was assumed to be like maze.
[61] 2015	Urban environments navigation	Guide robot	Navigation algorithm	Robot responded fast in front of moving obstacles.	Only for urban environments	Trained neural networks to control the robot to reach the target location in urban dynamic environments	
[314] 2015	Path tracking	DMRV, ROC6, Clearpath Husky	"Learning-based Nonlinear Model Predictive Control (LB-NMPC) algorithm	Flexible and effective for decreasing path-tracking errors"	Robustness	The algorithm LB-NMPC used a model of a vehicle and a model of learned disruption. Disturbances were modeled as a Gussian process (GP) based on the state, input, and other relevant variables of the system. The GP has been updated based on prior trial experience.	Disturbances are uncorrelated
[315] 2015	Navigation	PackBot	"Heterogeneous landmark-based visual navigation (HLVN) algorithm"	Performed well on datasets where rectilinear structures dominate the scene.	Environment constraints are there and robustness need to be enhanced	Utilize heterogeneous visual features and their inner geometric constraints managed by a novel multilayer feature graph (MFG). The method extended the "SLAM framework by explicitly exploiting the heterogeneous features and their inner geometric relationships in an unsupervised manner"	The camera is precalibrated with its radial distortion removed
[62] 2016	Urban environment navigation	Car-like robot APACHE	Image based dynamic window approach	Robust and viable, methodology can be applied with low-cost sensors	Obstacles are fixed	"Combined a Visual Servoing (VS) approach, as deliberative control, and the Image-based Dynamic Window Approach (IDWA), as reactive control"	
[25] 2016	Agricultural purpose	Ackermann-steered electric John Deere TE Gator,	Visual crop row-tracking method	appropriate for detecting obstacles in cropping fields	The method was unable to prevent the vehicle from driving on the crop rows	Proposed a visually aided localization system using inexpensive GPS, inertial, and camera sensors that allowed the robot to continue to follow crop rows during GPS outages. Also an obstacle detection system using a stereo vision sensor has been presented.	The ground is flat

Table 6.1 continued from previous page

Ref Year	Applications	Hardware	Algorithm	pros	Cons	Working	Assumption
[53] 2016	Robot navigation and localization	DX CAN-bus	Extended kalman filter	Effective for indoor environment	Only for indoor environment	Presented a landmark recognition and a "localization method based on RGB-D vision. An EKF method was introduced to estimate robot position basing on the landmarks. The path planning approach based on a differential flatness point-to-point trajectory planning were proposed"	
[54] 2017	Indoor obstacle detection and recognition for visually impaired	Wearable device	Stixel World, Object detection algorithm	Rapid acclimation, readily interpretable signals, and moderate comfort with the haptic belt interface.	Outdoor is missing	Used depth information from a camera, the system distinguishes walkable free space from obstacles and could identify a few important types of objects. These descriptions of the surroundings were communicated to the person wearing the device and translated into safe navigation directions	Only indoor environment
[42] 2017	Navigation	Simulator	RESNET features, CNN	Achieved semantically specified goals, such as go to a chair.	Only static environments were considered	"Proposed architecture learns to map from first-person views and plans a sequence of actions towards goals in the environment"	Mapper predicts free space
[44] 2017	Target-driven visual navigation.	SCITOS	Deep Reinforcement Learning (DRL) approach	Generalized method to fresh goals and scenes not used during the model's end-to-end practice. Methods worked in discrete and ongoing areas	Environmental constraints	Proposed a deep reinforcement learning (DRL) framework for visual navigation having known targets.	
[41] 2017	Vision based navigation and pedestrian detection	Ackerman steering configuration for the robot	HOG cascade classifier, LBP cascade Classifier, Adaboost	A smooth braking control to keep a safe distance between the mobile robot and the pedestrian	Slow response	Evaluated two pedestrian detection algorithms: HOG cascade classifier and LBP cascade classifier off-line and onboard the robot. Also implemented a communication system between the robot and the ground station	
[43] 2017	Safe Visual Navigation	Autonomous RC car	Deep learning	Provide safe, high-speed visual navigation of an autonomous mobile robot	Tested for limited environments using small convolutional network architectures.	"Demonstrated a safe, self-supervised method of learning a visual collision prediction model online"	Ability to accurately infer dense environment geometry through the used "sensors is limited to 5 m or less, which was characteristic of many monocular, stereo and RGBD methods"
[36] 2018	Vision based indoor navigation	Siamese A3C model	Deep reinforcement learning algo	Tried unseen targets also	Lack of practicality	Proposed the AdobeIndoorNav dataset to fill the gap of synthetic 3D scene datasets and 3D reconstructed scene datasets.	
[316] 2018	Path planning	Turtlebot	Topomap algorithm	Achieved good performance with lower computation times and storage requirements.	Topological maps do not include sematic information	Transformed into a three-dimensional topological map a sparse feature-based map from a visual SLAM scheme. Directly from the noisy sparse point cloud extracted occupancy information. Then created a set of convex clusters of free space, which were the topological map's vertices.	Vertices were assumed to be convex
[59] 2018	Navigation	Not mentioned	Vision based algorithm	In an urban canyon scenario	Scenario is limited	A monocular vision-based navigation system with a single camera and 3-D map is proposed.	

Table 6.1 continued from previous page

Ref Year	Applications	Hardware	Algorithm	pros	Cons	Working	Assumption
[34] 2018	Intelligent robot system	POWERROB - KSU-IMR	A* algo, EKF, Dijkstras algo, Iterative Closest Point (ICP) localization algo	Navigate even at difficult localities.	The mobile robots performance varies "depending on the area, region, and zone of navigation"	"The adopted learning paradigm was a five layers Neuro-Fuzzy learning architecture, with to ability to create an FIS inference for enhanced navigation. To capture the enormous visual and non-visual sensory data, the mobile robot platform has fully computer-interfaced stereo vision, and reliable 3D perception system onboard the mobile platform."	Robots initial position
[45] 2018	Vision-Based Navigation	Turtlebot	Deep learning method	the number of trajectories needed to find a path is significantly reduced	Problem arises from limited field of view associated to dense visual sensors	Use of deep networks to inform sample-based navigation planning	A constant forward velocity was assumed
[317] 2018	3D gaming, engineering,medical and simulation.	Not mentioned	Navigation technique using camera	Has reliable recognition and accuracy rates	The system is not capable to be used in complex virtual collaborative environment.	Implemented in a visual studio project navigation by index tracing (NBIT). With NBIT, parallel processing was performed for backend index tracking and frontend rendering of virtual scene.	
[35] 2019	Indoor navigation assistance for visually impaired	ISANA system runs on Google Tango mobile device	Time-stamped map Kalman filter algorithm	Easy to use	Need cognitive understanding and navigation should be done in more complex and cluttered environment	"Presented a mobile computing ISANA with SmartCane prototype to assist blind individuals with independent indoor travel. ISANA functionalities included indoor semantic map construction, navigation and wayfinding, obstacle avoidance, and a multimodal user interface"	
[50] 2019	Target driven visual navigation	Simulator	Markov model	CNN is used for training.	Real time robot is not used.	Used semantic segmentation and detection masks as observations obtained by state-of-the-art computer vision algorithms and used a deep network to learn the navigation policy.	Model has collision detection module
[51] 2019	To simulate the collective motion of animal species, such as birds or fishes	Simulator	vision based algorithm using optical flow features	Performed well in scenes having static as well as dynamic obstacles	The environment lighting conditions were not considered	The agents viewed optical flow with static objects as a result of their relative movement. In order to extract visual features, the optical flow was then segmented and processed. Then, the relationship between these visual features and the movement of the agent was established and used to design a set of control functions that would allow characters to perform object-dependent tasks.	An accurate estimate of the optical flow is available
[318] 2019	Navigation under Low Illumination	Not mentioned	AGAST/BRIEF, histogram equalization	Operate under low-illumination conditions at night	Lack of automatic activation of methods when transition happens between no image preprocessing and when preprocessing is activated due to scene illumination conditions	"Presented and investigated four methods to enhance images under challenging night conditions. The findings were relevant to a wide range of feature-based vision systems, such as tracking for augmented reality, image registration, localization, and mapping, as well as deep learning-based object detectors"	
[65] 2019	For generating maps	RatLab	modified Slow Feature Analysis (SFA) algorithm	Efficient	Real time implementation is not done	Presented a vision-based navigation system that involves generating place and HD cells through learning from visual images, building topological maps based on learned cell representations and performing navigation using hierarchical reinforcement learning	

List of Publications

Journals

1. Baljit Kaur, Jhilik Bhattacharya, *"A Convolutional Feature Map-based Deep Network targeted towards Traffic Detection and Classification"*, Expert Systems with Applications, Volume 124, 15 June 2019, Pages 119-129, 2019.
2. Baljit Kaur, Jhilik Bhattacharya, *"Scene perception system for visually impaired based on object detection and classification using multimodal deep convolutional neural network"*, J. Electron. Imaging 28(1), 013031 (2019), doi: 10.1117/1.JEI.28.1.013031.
3. Baljit Kaur, Jhilik Bhattacharya, *"Predictive hierarchical human augmented map generation for itinerary perceptionx"*, IET Electronic Letters, Volume 52, Issue 16, 04 August 2016, p. 1381 1383.

International Conference

1. Baljit Kaur, Jhilik Bhattacharya, *"Augmented Map based Traffic Density Estimation for Robot Navigation"*, IEEE TENSYMP 2018.