Understanding Environmental Cues Using Deep Learning-Based Image Analysis

Elhassan Mohamed

School of Engineering

University of Kent

A thesis submitted for the degree of

Doctor of Philosophy in Electronic Engineering

March 2022

Pages: 232

May Allah accept this work purely for his sake . . .

“Say: My prayers and sacrifice, my life and death, are all for God, Lord of all the Worlds; He has no partner. This is what I am commanded, and I am the first to devote myself to Him.”

(Al-An’am 162-163)

Acknowledgements

I would like to express my gratitude to my supervisors, Dr Konstantinos Sirlantzis and Prof Gareth Howells, for their help and support throughout the years. I could not have reached this level without their guidance and knowledge. I have learned from their experiences, and I will remember their words all of my life.

I would like to thank the head of the Intelligent Interactions research group, Kent Assistive RObotics Laboratory, Dr Konstantinos Sirlantzis, and its members for the logistic support.

A special thanks to my wonderful team, whom we travelled the journey together, Paul Oprea, Sotirios Chatzidimitriadis, Yankun Yang, Jihad Dib, and Odysseas Doumas. I appreciate the time we spent together, the discussions we had, and the memorable moments.

I would like to pay the honour, appreciation, and sincere gratitude to my family, my father and backbone (Prof Nazir), my beloved mother (Amira), my beautiful sisters (Safa and Esraa), and my elder brother (Dr Mohamed), for their support and prayers. May Allah bless them and grant them the highest rank in paradise.

Lastly, I would like to thank everyone who helped me throughout these years, every single person who supported me, stood by my side, believed in me, and treated me well.

This work is supported by the Assistive Devices for empowering dis-Abled People through robotic Technologies (ADAPT) project. ADAPT is selected for funding by the INTERREG VA France (Channel) England Programme which is co-financed by the European Regional Development Fund (ERDF). The European Regional Development Fund (ERDF) is one of the main financial instruments of the European Unions (EU) cohesion policy.

Abstract

The recent advances in Artificial Intelligence (AI) motivate its ubiquitous use for computer vision with applications in several fields, such as autonomous and assisted navigation. Deep learning, a branch of artificial intelligence, is shown to be useful, for example, in object detection and semantic segmentation tasks. Such algorithms achieved high performance in terms of accuracy and computational time compared to conventional techniques. However, new application areas, such as providing information for arbitrary environments to address, for example, indoor navigation or simultaneous indoor and outdoor navigation, introduce several challenges that should be overcome. For these challenges, novel deep learning-based methods should be introduced, implemented, and tested in realistic scenarios such as assistive driving of mobile platforms, e.g., powered wheelchairs.

This thesis introduces and explores novel deep learning techniques for object detection and semantic segmentation to enable intelligent systems which aid scene understanding and human-system interaction and could be used in the navigation of any robotic platform. A prominent area in which these types of systems are needed, i.e., aiding with the driving of powered wheelchairs for users with visual disabilities, is chosen as the realistic application to test the performance of the algorithms and methods introduced. Extensive investigations of their characteristics are performed, including using explainable AI (XAI) to justify corresponding system outputs.

A review of relevant literature reveals a number of distinct challenges that need to be addressed to develop a system able to operate in realistic environments:

The first challenge our proposed systems aim to address is being able to perform well with small and large size objects simultaneously. State-of-the-art object detection systems struggle with the localisation of small size objects. These systems are usually trained on large size objects containing abundant information and large numbers of pixels to be utilised by the model during the training and inference processes. Our research investigates the performance of these detectors on a proposed dataset that mainly contains small size objects. Furthermore, the study discusses the means of enhancing the detector's performance on tasks that involve the detection of multi-size objects using multi-head detectors to make predictions on different feature maps. The introduced multi-head detector has achieved mean Average Precision (mAP) of 0 .818 on the proposed dataset. Finally, our investigation findings proposed a roadmap to help the scientific community to choose the best detector for a given application.

The second important challenge to be addressed is the requirement to provide information about the elements of the scene on which the system's decisions were based. System transparency ensures the reliability of deep learning-based computer vision systems. It is not only important to attain an accurate system but also a system that can explain its predictions. A black box system should provide insights into what is happening inside the system to be approved and used in real-life applications. Policymakers and legislators require a certain level of system transparency to approve such technologies. Therefore, in this thesis, we investigate the robustness of systems in terms of their abilities to explain the reason behind a specific decision by introducing novel explanation techniques, thus contributing to the so-called XAI field. Also, novel explanation techniques that can visualise the two main characteristics of robust explanation maps, i.e., fine-grained details and discriminative regions in a single representation (in the form of a "heatmap"), are introduced, implemented, and tested for well-known AI methods. Unlike standard visualisation methods used currently, the introduced ones can identify multiple important image characteristics upon which the system decisions are based.

The third main challenge addressed in this work is providing information about objects of the environment at the image pixels level. Semantic segmentation at the pixel level is explored to better utilise the available images of the dataset used. Pixel classification can better define the boundaries and the geometric shape of the target object than object detection, which provides bounding boxes containing the detected objects. Classifying every pixel in an image, consequently identifying the object boundary, size, and location, facilitate subsequent tasks and human-system interactions at higher accuracy. Also, novel semantic segmentation architectures that can process images from both indoor and outdoor environments are introduced, implemented, and tested in this work. Analysing and understanding data from two different distributions (indoor and outdoor) with a variety of object sizes is challenging due to the difference in the images' contexts and the limited number of datasets currently publicly available, which our systems were shown to be able to handle with significant accuracy and processing speed.

Finally, the proposed systems are tested in a realistic scenario drawn from the field of assistive robotics in powered mobility (Electrical Powered Wheelchairs - EPWs). Visually impaired persons with comorbidities are not prescribed a powered wheelchair due to their sight condition. This is an ideal setting to test our system in real conditions. The proposed semantic segmentation system aims to provide visual cues to aid with the navigation process and increase the user's independence. As these systems are meant to be installed on moving platforms such as mobile robots (EPWs in our case), they are susceptible to mechanical vibrations caused by different terrains. These could negatively impact the performance of our smart deep learning-based computer vision systems. Vibration effects on these systems are examined in detail, where the implication on performance and prospective solutions are highlighted. Our final results indicated that there is a deterioration of 4% in the performance due to these vibrations.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This thesis is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and acknowledgements.

Elhassan Mohamed
March 2022

List of Publication

Articles

1. E. Mohamed, K. Sirlantzis and G. Howells, “A review of visualisation-as-explanation techniques for convolutional neural networks and their evaluation,” in *Displays*, vol. 73, pp. 102239, 2022, doi: 10.1016/j.displa.2022.102239.

2. E. Mohamed, K. Sirlantzis, G. Howells and J. Dib, "Investigation of Vibration and User Comfort for Powered Wheelchairs," in *IEEE Sensors Letters*, vol. 6, no. 2, pp. 1-4, Feb. 2022, Art no. 2500204, doi: 10.1109/LSENS.2022.3147740.

3. E. Mohamed, K. Sirlantzis and G. Howells, "A pixel-wise annotated dataset of small overlooked indoor objects for semantic segmentation applications," in *Data in Brief*, vol. 40, pp. 107791, 2022, doi: 10.1016/j.dib.2022.107791.

4. E. Mohamed, K. Sirlantzis and G. Howells, "Indoor/Outdoor Semantic Segmentation Using Deep Learning for Visually Impaired Wheelchair Users," in *IEEE Access*, vol. 9, pp. 147914-147932, 2021, doi: 10.1109/ACCESS.2021.3123952.

Conferences

1. E. Mohamed, K. Sirlantzis and G. Howells, “Analysing the Impact of Vibration on Smart Wheelchair Systems and Users,” *2022 3rd International Conference on Pattern Recognition and Artificial Intelligence (ICPRAI).* (Accepted).

2. E. Mohamed, K. Sirlantzis and G. Howells, "Incorporation Of Rejection Criterion - A Novel Technique For Evaluating Semantic Segmentation Systems," *2021 14th International Conference on Human System Interaction (HSI)*, 2021, pp. 1-7, doi: 10.1109/HSI52170.2021.9538787.

3. E. Mohamed, K. Sirlantzis and G. Howells, "Application of transfer learning for object detection on manually collected data," *2019 In Proceedings of SAI Intelligent Systems Conference*, 2019, pp. 919–931, doi:10.1007/978-3-030-29516-5_69.

4. E. Mohamed, J. Dib, K. Sirlantzis and G. Howells, “Integrating ride dynamics measurements and user comfort assessment to smart robotic wheelchairs,” *2019 15th Conference on Global Challenges in Assistive Technology: Research, Policy & Practice*, IOS Press. Available at: https://aaate2019.eu/aaate-2019-proceedings/.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

Acc	Accuracy
ADAPT	Assistive Devices for empowering disAbled People through robotic Technology
AI	Artificial Intelligent
AP	Average Precision
ASPP	Atrous Spatial Pyramid Pooling
BB_IoU	Bounding Box Intersection over Union
BF score	Boundary F1 score
BN	Batch Normalisation
CAM	Class Activation Map
CamVid	Cambridge-driving Labeled Video Database
CNN	Convolutional Neural Network
CRC	Cascade Rejection Classifier
CRF	Conditional Random Field

CSPNet	Cross Stage Partial Network
DAG	Directed Acyclic Graph
DCNN	Deep Convolutional Neural Network
DL	Deep Learning
DLV1	Deep Lab Version 1
DLV2	Deep Lab Version 2
DLV3	Deep Lab Version 3
DLV3+	Deep Lab Version 3 plus
EPW	Electrical powered wheelchair
FCL	Fully Connected Layer
FCN	Fully Convolutional Network
FPN	Feature Pyramid Network
FPS	Frame Per Second
GA	Global Accuracy
GAN	Generative-Adversarial Networks
GAP	Global Average Pooling
GBD-Net	Gated Bi-Directional Network
GI	element-wise products of Gradients and Input
gTruth	Ground Truth

HOG	Histogram of Oriented Gradients
IG	Integrated Gradient
IMU	Inertial Measurement Unit
ION	Inside-Outside Net
IoU	Intersection over Union
IS	Instance Segmentation
LIDAR	LIght Detection And Ranging
LIME	Local Interpre Model-agnostic Explanations
MA	Mean Accuracy
mAP	mean Average Precision
Mask_IoU	Mask Intersection over Union
mIoU	mean Intersection over Union
MR-CNN	Multi-Region Convolutional Neural Network
MS-CNN	Multi-Scale Convolutional Neural Network
MSE	Mean Square Error
MTGAN	Multi-Task Generative Adversarial Network
NMS	Non-Maximum Suppression
OC	Object Classification
OD	Object Detection

P	Precision
PQ	Panoptic Quality
PS	Panoptic Segmentation
R	Recall
R-CNN	Region-based Convolutional Neural Network
RMS	Root Mean Square
RNN	Recurrent Neural Network
RoI	Region of Interest
RPN	Region Proposal Network
SDP	Scale-Dependent Pooling
SDR	Smallest Destroying Region
SGDM	Stochastic Gradient Descent with Momentum
SmoothGrad	Smooth Gradients
SNIP	Scale Normalization for Image Pyramids
SPP	Spatial Pyramid Pooling
SS	Semantic Segmentation
SSD	Single Shot Detectors
SSR	Smallest Sufficient Region
SVM	Support Vector Machine

XAI	eXplainable Artificial Intelligence
YOLO	You Only Look Once

Chapter 1

Introduction

Artificial intelligence technologies and applications have recently witnessed exponential growth in many sectors. The growth in the adaption of artificial intelligence in general and Deep Learning (DL) in specific is attributed to the efficiency and competency of the produced systems compared to conventional systems. However, DL techniques have introduced many challenges and raised many questions. This thesis aims to adopt these technologies to help disabled users who rely on powered wheelchairs for commuting. The developed systems are not bounded by their application for powered wheelchairs but can be used with any robotic platform.

The thesis aims to investigate the limits of DL in terms of usability and adaptation in real-life applications. ADAPT (Assistive Devices for empowering disAbled People through robotic Technologies) project aspires to develop smart systems that can be integrated into Electrical Powered Wheelchair (EPW) to help and assist disabled people in their daily lives. These smart devices can help with autonomous driving, environmental understanding, physiological measurements, speed adaptation to terrain types, etc.

The thesis focuses on DL for computer vision tasks such as object classification, object detection, and semantic segmentation (pixels classification). The objective is to build a computer vision system based on DL that can help visually impaired disabled users to navigate safely.

This can be achieved by displaying environmental cues on a display fitted on the powered wheelchair and customised upon the user's need and level of disability. The user can use the displayed information, including the estimated distance to the target object, to interact with the surrounding environment.

Many challenges have been encountered throughout the research journey. For instance, the performance of a DL system trained on a specific dataset for a specific environment can negatively impact when deployed in a different environment. The proposed solutions try to mitigate this issue by using shared systems with no need for retraining. This can help to build a system that can be used in different setups, such as indoors and outdoors simultaneously.

Through the thesis, the challenges and the proposed solutions are explored. The introduction chapter is organised as follows: research motivations are presented in section 1.1. Aims and objectives are highlighted in section 1.2. Research questions are introduced in section 1.3. Section 1.4 highlights the contributions of the thesis. The applications of the proposed systems are presented in section 1.5. Section 1.6 outlines the scope of the thesis. Lastly, the thesis structure is outlined in section 1.7.

1.1 Research motivation

The motivation for this research stems from the advent of novel Artificial Intelligent (AI) technologies and their ability to form part of computer vision systems which could be "trained" to sense and understand their environment, thus providing context information to their users. Software implementation of such systems and corresponding appropriate hardware have reached maturity levels which allow deployment in mobile platforms such as robotic devices. One significant example is mobility aid devices such as powered wheelchairs. Thus, the application example of the research reported in this thesis is motivated by the need for assistive devices that can help visually impaired users with mobility impairments who could use powered wheelchairs for their daily activities. Visual impairment, however, precludes this category of users from being prescribed and being able to use wheelchairs, thus restricting their wellbeing. For this reason, we believe this to be an important application area and testing framework for the AI computer vision systems mentioned previously. Powered wheelchairs form a realistic mobile platform that addresses arbitrary environments in a user's everyday life. However, these systems need to be robust and efficient to satisfy the strict requirement of this type of use. Furthermore, they should be trusted by the users and the relevant regulators.

1.2 Aim and objectives

This thesis aims to explore the feasibility of using deep learning-based computer vision systems on mobile platforms to understand arbitrary realistic environments.

The main objectives of the thesis are:

- To investigate the feasibility of developing smart deep learning-based computer vision systems that can be used to comprehend the surrounding environment by providing visual cues and geometric information of the target objects.

- To investigate the reliability and robustness of the proposed systems and means of enhancing their transparency and explainability.

- To implement and test the developed systems for a real-life application, such as integrating the system on a powered wheelchair to guide visually impaired disabled users, to practically assess the proposed systems and their ability to facilitate the navigation process.

- To investigate the impact of the induced vibrations in a realistic environment on the detection performance of the developed systems that are installed on the powered wheelchair and the anticipated deterioration in performance.

1.3 Research questions

Subsequently, the research questions investigated in this thesis are:

- How do the objects of a dataset, in terms of size and distribution, impact the trained deep learning models? Given the wide variety of deep learning-based detection systems, how can we choose the best detector for a given application?

- In terms of the applicability of these systems, can we avoid long training times spent by computer vision systems based on DL? Can a DL system process images from two different distributions simultaneously (e.g. large and small size objects, or indoor and outdoor environment objects)? What are the necessary modifications that can achieve this purpose?

- What are the challenges in adopting such technologies in real-life scenarios? How reliable are the proposed systems? What level of system transparency can be attained? For example, could explainable AI (XAI) offer levels of transparency and decision justification required in realistic implementations?

- What level of environmental understanding can AI-based computer vision systems achieve, i.e. could pixel classification be achieved, and could semantic information be extracted from arbitrary video sequences? Are there available semantic segmentation evaluation metrics accurately assessing system performance? Could pixels' confidence scores enhance the validity of the evaluation process and provide better insights into the operating characteristics of the assessed systems?

- Can a smart computer vision system based on deep learning help visually impaired users to navigate safely? What level of assistance can it offer?

- What performance degradation should we expect in a deep learning-based computer vision system installed on a mobile platform (e.g. a powered wheelchair) due to mechanical vibrations caused by different types of terrains traversed in everyday use?

1.4 Main contributions

- We introduced, implemented, and tested shared systems that can simultaneously process different size objects from different distributions. The developed systems do not require retraining and can achieve high performance in both environments.

- We introduced a roadmap to choose the optimal deep learning-based detector for a given application based on the size of the objects of the dataset.

- Two datasets with multi-size objects were collected and annotated for ground truth. These serve in training and evaluation for object detection and semantic segmentation tasks. There are no similar datasets publicly available to the best of our knowledge.

- We introduced explanation methods that can help to understand the intuition behind a specific decision or prediction.

- A novel technique to assess semantic segmentation systems is introduced. Unlike other metrics, the proposed one incorporates pixels' confidence scores in the evaluation process.

- The impact of mechanical vibrations on the performance of the smart computer vision systems installed on robotic platforms, such as powered wheelchairs, is investigated, and related results are analysed.

1.5 Applications

The project aims to develop assistive devices that can be integrated into powered wheelchairs. A semantic segmentation system has been proposed to process images from two different distributions simultaneously. This system can help visually impaired disabled powered wheelchair users to understand their surrounding environment. Besides, it allows interaction with the environment by providing visual cues and distance to the target object when needed.

The introduced system has been deployed and tested for indoor scenarios. It can be easily adjusted to the user's needs. Results show the ability of the semantic segmentation system to estimate the distance to the target object accurately.

1.6 Scope of the thesis

The scope of the thesis focuses on creating smart computer vision solutions and systems that can help with navigation and scene understanding. These systems can be installed on mobile platforms, such as powered wheelchairs and used by disabled users, such as visually impaired users, to comprehend the surrounding environment and interact with objects, such as door handles and switches. Thus, the developed systems need to be adaptable and customised to the users' needs. Furthermore, the proposed systems need to be transparent and reliable.

The scope of the thesis covers:

- DL techniques for object detection and semantic segmentation. Conventional computer vision systems are beyond the scope of this thesis.
- Autonomous systems based on collision avoidance and path tracking are beyond this thesis's scope.

- DL explanation and visualisation techniques to understand the system's predictions. Model approximation methods are beyond the scope of this thesis as we are interested in the direct explanation of pretrained models using visualisation methods.
- The vibration impact on the performance of detection systems installed on the powered wheelchair and mitigation means. However, powered wheelchair dynamics and suspension system redesigning are beyond the scope of this thesis.

1.7 Structure of the thesis

The thesis structure is organised as follows:

- **Chapter 2** reviews state-of-the-art systems for object detection, semantic segmentation, and explanation methods.
- **Chapter 3** introduces two application-specific datasets for indoor objects of interest. One is annotated on the bounding box level for object detection tasks, while the second is annotated on the pixel level for semantic segmentation (pixel classification) tasks. These datasets are important not only for powered wheelchair applications but also for any robotic platform. Similar to their usage with powered wheelchair applications to understand the surrounding environment, service robots in specific and computer vision systems, in general, can benefit from these datasets. The majority of the datasets' objects can be categorised as small size objects, making them challenging to detect. However, they are essential for many applications.
- **Chapter 4** proposes novel explanation techniques to visualise Convolutional Neural Network (CNN) predictions. The proposed explanation methods offer two benefits over state-of-the-art ones. First, they can produce comprehensive maps with fine-grained details and discriminative regions, unlike individual methods that can produce either

fine-grained details, such as gradient-based methods or discriminative regions, such as Class Activation Map (CAM) based methods.

Second, the proposed methods visualise all the features contributing to the model's prediction in different colours. Each colour represents a different method, Unlike individual methods that generate monotonic specific features based on the explanation technique. Consequently, the proposed methods are descriptive and easy to understand.

- **Chapter 5** investigates the performance of state-of-the-art object detection systems on the proposed dataset. The impact of feature extraction networks, feature extraction layers, number of anchor boxes, and training data on the produced systems is investigated. This chapter's findings can help the scientific community to expect the detector performance based on the application data. In addition, the optimal system can be selected for a specific application.
- **Chapter 6** proposes novel semantic segmentation systems that can process images from two different environments. The developed systems do not require retraining and can achieve high performance in both environments. Semantic segmentation systems can classify every pixel in a given image. Consequently, information such as geometric shape, centre of gravity, and distance to a specific object can be acquired. This provides a remote mapping of accessibility features in the surrounding environments. Based on this information, further interaction can be performed, such as object manipulation, which enhances the usability of the proposed systems.
- **Chapter 7** proposes a novel technique that incorporates pixels' confidence scores in the evaluation process of semantic segmentation systems. Pixels' confidence scores reflect the certainty of the system. Consequently, their contribution to the assessment process is vital.
- **Chapter 8** concludes the thesis, demonstrates the findings, and highlights future work.

Chapter 2

Explainable Deep Learning: A Review of Literature

2.1 Introduction

Deep learning techniques for object detection [1, 2] and semantic segmentation [3, 4] have achieved state-of-the-art performance on object localisation and pixel classification tasks compared to traditional methods that use hand features or algorithms to detect corners. Since then, deep learning approaches have been adopted in many real-life applications. Artificial intelligence (AI) in general and deep learning in specific have seen a great leap that boosts their ubiquitous use in many fields such as autonomous [5], industrial [6–8] and medical applications [9–12].

Electrical Powered Wheelchair (EPW) users can benefit from these technologies, especially people with visual impairments. Also, people with cognitive and physical issues can use computer vision systems based on deep learning to better understand their surroundings and better interact with environments. Currently, visually impaired wheelchair users cannot be prescribed powered wheelchairs due to their conditions. Some solutions based on semantic segmentation have been proposed to help non-disabled visually impaired users to navigate safely by identifying the terrain type using smart glasses [13] or identifying the most walkable direction using a hand-held camera [14].

Scene understanding approaches are widely used in the autonomous driving industry. Compared to autonomous vehicles, powered wheelchairs have limited speeds and operate on specific routes. Moreover, visually impaired disabled users cannot fully utilise their bodies which limits their vision and increases the risks. Adopting scene understanding technologies to help EPW users to drive safely in indoor and outdoor environments is a trending research topic that needs novel contributions and pioneer solutions.

Deep learning solutions must be reliable and robust to be approved by authorities and adopted by industry. This cannot be achieved unless the model is able to explain its decisions (why does the model take a specific action in a given situation?). Deep learning systems are treated as black boxes, where the reason for an action or a prediction is unclear. The explanation methods of AI systems or so-called XAI are currently of significant concern and interest because they help to ensure the reliability and robustness of deep learning models.

In this chapter, we focus on the need for such a system that can help users with visual impairments to use an EPW as some of these users are not prescribed a powered wheelchair due to their disabilities [15]. Also, scene understanding techniques used by powered wheelchairs are discussed (2.2). Then, the literature of deep learning methods for object detection (2.3) and semantic segmentation (2.4) tasks are reviewed. After that, explanation methods for Convolutional Neural Networks (CNNs) are presented (2.5). Lastly, we highlight the research directions and the prospective systems that can achieve the research objectives (2.6).

2.2 Assistive devices and motivations

There are many motivations for disabled people to utilise EPWs. Apart from the main reason, which is mobility, other factors such as productivity, leisure and independence are involved [16]. That is why assistive devices should enable users to master their objectives independently and enhance their quality of life. At the same time, poor design and faulty assistive devices have a negative influence on the users' experience [16].

Clinicians report that they saw almost the same number of patients who cannot use a powered wheelchair as who can [15]. Patients find it extremely difficult to manoeuvre an EPW indoors, especially in small areas and while negotiating doorways. Clinicians also report that 40% of their powered wheelchair users find steering tasks difficult. At the same time, five to nine percent find them impossible. On the other hand, the percentage of those who cannot use a powered wheelchair due to visual impairment, cognitive disorder or motor skills is 85%. An automated navigation system is believed to half this percentage [15].

Navigation systems based on computer vision offer semi-autonomous and fully autonomous driving capabilities for EPWs' users. For instance, driving a wheelchair using face tracking [17] or eye and iris movement [18, 19]. Moreover, technologies such as collision detection and avoidance can be used to assist the EPW driver in negotiating obstacles [20]. Viswanathan et al. [21] introduce a 3D stereo-vision navigation-based system that can detect potential object collisions by stopping the movement towards that object, plan paths towards a specific goal using visual odometry and prompts to assists the user in navigation based on the user's level of awareness.

A comprehensive review of smart wheelchairs is presented in [22]. Although these systems provide great help, they can be unsatisfactory or faulty. For example, consider the case when a user wants to approach an object that has been detected as an obstacle by the system. In this case, the autonomous system wants to avoid the object, while the user needs to reach that object. The only possible solution, in this case, is to disable the system.

In contrast, we propose to build smart systems based on deep learning for computer vision tasks to act as a guide for the user. They do not interfere with the navigation process. They are non-intrusive systems, which classify the environment into different classes to lead and smooth the user's navigation process. Computer vision techniques based on deep learning proved high efficiency in terms of speed and accuracy compared to conventional methods [23].

A closely related system to the proposed one is presented in [24]. A wheelchair system to guide people with severe disabilities is used to track manually taught paths (reference paths stored on a memory) using optical encoders mounted on the wheelchairs' wheels and visual beacons (passive cues) placed throughout the wheelchair surrounding environment (on walls, stationary objects, etc.). Relying only on the optical encoders to estimate the wheelchair's position may introduce errors because of inaccurate initial conditions, wheel slippage, etc.. Hence, environmental cues that are captured by the two cameras installed on the powered wheelchair are used to correct and update the wheelchair's position and orientation using Kalman filter algorithm. The system uses the difference between the reference path and the estimated position to drive the wheelchair automatically. However, the system does not override the control from the user to follow a path until the user requests so.

The main disadvantages of such a system are as follows: it needs visual cues to be deployed in the wheelchair environment. It needs a manually taught reference path. Most importantly, the system needs a different setup for different environments. This means that if the environment changes, new reference paths are needed to teach the system. Although we do not use our proposed systems for path tracking, our systems are capable of detecting visual cues automatically using deep learning methods, which means no need to add physical visual cues to the environment. Moreover, the proposed systems can detect the distance to a specific object using the Intel®RealSense depth camera (video). Besides, our systems provide all the information to the user on a screen from which the user can take full control of the EPW (video).

In contrast to the fully autonomous EPWs systems that take full control away from the users, which are undesirable, our proposed methods provide environmental cues to help, guide and keep the users in full control. EPWs systems that provide collision avoidance support such as [25, 26] may not be suitable for drivers who are unable to determine their location and cannot navigate to a specific location. Our proposed systems allow the users to understand their surroundings and provide them with the distance to the target object when needed. Consequently, the proposed systems can be seen as in-between systems that can provide environmental cues (scene understanding) while giving full control to the users to avoid both limitations of non-autonomous and fully autonomous systems. Though they can be integrated with autonomous ones, and the users can decide the level of assistance.

Object detection and segmentation deep learning-based methods are reviewed in the following sections to find the best system for our application. Although small size object detection is an important topic for real-life applications, it is not well-covered by state-of-the-art detection systems. Thus, we highlight state-of-the-art systems and techniques that deal with small size object detection challenges. Also, scene understanding using semantic segmentation techniques are investigated. Lastly, visualisation methods to ensure the reliability and the robustness of deep learning models are discussed.

2.3 Object detection

Object detection is different from object recognition and classification. For object classification, the task is to classify a given image to one class from a set of predefined classes. Usually, the image has one object. Suppose the image has many objects of different categories. In that case, the classification task will assign a single category to the image with a confidence probability or a classification score that reflects the classification confidence.

For object detection tasks, it is not only important to classify the objects in a given image but also to determine their locations. This means multiple objects of different classes can be classified and localised in a single image.

Several methods have been proposed for object detection using deep learning [1, 2]. This literature review concentrates on state-of-the-art ones that are deeply investigated and documented, open-source, easily implemented and tested on different platforms such as MATLAB.

Most importantly, methods that can achieve high performance under the challenging conditions of the proposed dataset. The proposed dataset contains predominantly small objects. Small object detection is challenging even for state-of-the-art methods that can achieve high performance on large size objects. In short, we focus on methods that can satisfy and achieve our objectives.

Object detection literature can be divided into two categories: one-stage and two-stage object detection systems. Generally, object detection systems based on deep learning consist of CNNs with regression and classification layers. One-stage systems use regression and classification layers to output the bounding boxes and the class of the object in one shot. In comparison, two-stage systems introduce an initial step to generate object proposals before bounding box regression and object classification. Thus, the pipeline of object detection systems has three components (Fig. 2.1): region selection, feature extraction, and classification and regression.

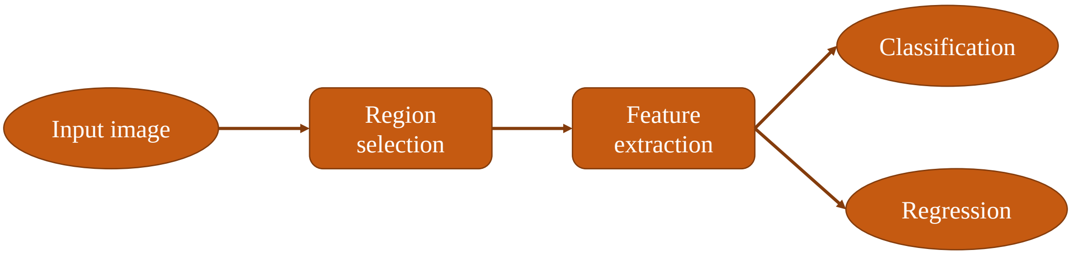


Fig. 2.1 The main blocks for a typical object detection system.

Region selection:

To predict the possible positions of objects in an image, the whole image needs to be scanned. Different objects may have different sizes and aspect ratios which require multi-scale sliding windows. Scanning the whole image to find object proposals is a time and resource-consuming process. It represents the bottleneck for modern object detection systems. Nevertheless, it is a very important stage in the pipeline of object detection systems for accurate detection. Using a limited number of scanning windows can negatively impact the system's performance as some objects may be overlooked. Traditional region selection methods such as selective search [27] and edge boxes [28] are time-consuming. However, regional proposal networks [29] overcome this disadvantage by using a CNN to predict these regions.

Feature extraction:

Object features that distinguish different categories are required for object classification. Histogram of Oriented Gradients (HOG) [30, 31] and Haar-like [32] are examples of traditional feature extraction algorithms. It is a challenging process to manually design a robust feature extraction system that can extract features from different objects because of the variety of object sizes, illumination conditions, occlusions, and backgrounds. Besides, some objects share similar geometric information such as different kinds of door handles, making the extraction of their distinct features a difficult operation. CNNs trained on large datasets can efficiently and automatically learn complex feature representations that can overcome the limitations of conventional methods.

Classification and regression:

The extracted features are then fed into an algorithm to classify the objects. Also, the location of the objects need to be determined. Traditional methods such as Support Vector Machines (SVM) [33] and Deformable Part-based Models (DPM) [34] can be used. However, state-of-the-art models usually utilise a classification layer and a regression layer for classification and bounding box regression tasks, respectively.

Advantages of CNN over traditional methods:

- Hierarchical representation interconnected relations between image pixels and high representation such as features patterns can be learned automatically. These features can be more descriptive and achieve high performance with deep architectures compared to shallow ones.

- Several related tasks can be optimised jointly, such as classification and regression using a combined loss function.

2.3.1 Two-stages object detectors (detectors based on region proposals)

Region-based Convolutional Neural Network (R-CNN) [35] achieved a significant improvement in performance compared to conventional object detection methods such as DPM [34]. It uses selective search [27] to generate 2000 region proposals for each image. The selective search approach uses saliency cues in a bottom-up manner to provide proposals. The generated regions are cropped or wrapped into a fixed size (227x227 pixels which the CNN requires). A CNN based on AlexNet [36] is then used to extract feature vectors for each region with a fixed dimension of 4096. Lastly, class-specific SVMs trained for each class is used to score the feature vectors, and a regression layer is used to fit a bounding box for each region. Multiple bounding boxes for an object are then filtered using the Non-Maximum Suppression (NMS) technique.

R-CNN uses AlexNet that has been trained on the ImageNet dataset [37]. Then, the model is fine-tuned on the domain-specific object detection task using the transfer learning [38, 39] technique that proves its efficiency compared to training from scratch, especially when domain-specific data are limited.

R-CNN has some drawbacks: first, selective search is a slow process representing a bottleneck for fast processing. Second, wrapping and cropping the region proposals to fit a fixed size suitable for the CNN input can introduce distortion. Third, R-CNN training is a computationally expensive multi-stage pipeline. The CNN, the SVM and the bounding box regressors need to be trained separately. The extracted features need a large disk size to be stored. Finally, the system cannot achieve real-time inference.

Spatial Pyramid Pooling Network (SPP-Net) modified R-CNN with a Spatial Pyramid Pooling (SPP) layer [40]. The SPP layer uses the convolutional layer's feature map to generate fixed-size vectors for object proposals. The constraint of having a fixed-size input image to the CNN (by wrapping or cropping) is avoided using the SPP layer to aggregate the information between the last convolutional layer and the subsequent fully connected layers. SPP-Net reduces the computational cost by computing a convolutional feature map for the entire image and then classify each object proposal. SPP-Net introduces some flexibility and better performance over R-CNN. However, it is still a multi-stage pipeline approach that requires large storage. Moreover, the convolutional layers before the SPP layer cannot be fine-tuned [40].

Fast R-CNN [41] is proposed to overcome R-CNN and SPP-Net problems. Without caching features, it can be trained end-to-end on a multi-task loss function for classification and bounding box regression. Fast R-CNN uses a special case of SPP layer called Region of Interest (RoI) pooling layer, which has one pyramid level.

Like SPP-Net, Fast R-CNN process the whole image to produce a feature map. The RoI pooling layer extracts a fixed-size feature vector from the feature map for each region proposal. The extracted feature vectors are then fed into a sequence of fully connected layers connected to two output layers: a softmax layer to produce the classes probabilities and a regression layer to produce the bounding box positions.

Softmax classifier used in Fast R-CNN outperforms linear SVMs classifiers used in R-CNN and SPP-Net. It does not require a disk storage. Moreover, it improves both accuracy and efficiency. Increasing region proposals can enhance the accuracy of the system. However, the accuracy will increase to a certain level then decrease with the increase of the object proposals. Further, the extra object proposals require more computations and resources that can negatively impact the inference speed of two-stage object detection networks.

Object detection models such as SSP-Net and Fast R-CNN suffer from bottleneck computations due to region proposals. Ren et al. [29] introduce Region Proposal Network (RPN), a fully convolutional network that shares full-image convolutional features with the detection network. It can be trained end-to-end to generate region proposals at nearly cost-free computations. Faster R-CNN is a merge between fast R-CNN and RPN where the training process is alternated between fine-tuning the RPN and the detection network. Consequently, the bottleneck region proposals step of the previous object detectors is replaced by a CNN that simultaneously predicts object bounding boxes and classes scores at each region proposal. RPN can process input images with arbitrary sizes. Furthermore, it reduces the number of region proposals from 2000 in the previous approaches to 300 region proposals per image without compromising the accuracy.

Unlike SPP-Net [40], Fast R-CNN [41], DPM [34], and Overfeat [42], which use pyramids of images or filters, Faster R-CNN introduced the concept of anchor boxes. RPN adapts anchors (reference boxes) with three different scales and three different aspect ratios. The regression towards the output Bounding Box (BB) is achieved by comparing the proposed BB with the anchors. RPN guides the Fast R-CNN network to the places where it is most likely to detect objects. Region proposals are generated by sliding a small network over the feature map of the last shared convolutional layer. The small network takes as input $n \times n$ spatial window of the input convolutional feature map. Each sliding window is mapped to a lower-dimensional feature, which is fed into two fully connected layers: box-regressor and box classifier layers. Multiple objects for each sliding window location can be detected using anchors. A total of nine anchors are centred at each sliding window location (3 scales × 3 aspect ratios).

Anchor boxes that cross the image boundaries represent a challenge for Faster R-CNN. These anchors are ignored during the training step (they do not contribute to the loss function) as the training can not converge without doing so [29]. This may introduce cross-boundary proposals boxes that will be clipped to the image boundary. Alternating training is also time-consuming as it first trains the RPN and uses the proposals to train the Fast R-CNN. Then, the network tuned by Fast R-CNN is used to initialise the RPN in an iterative process. However, Faster R-CNN with RPN can achieve an adequate frame rate with high accuracy compared to other region proposal methods.

2.3.2 One-stages object detectors (detectors based on classification/ regression)

Lenc et al. [43] introduce one of the early attempts to accelerate the two-stage object detection networks. The study suggests to drop the region proposal section from the R-CNN [35] as it represents the bottleneck of the architecture. Instead of using selective search for region proposals, the proposed system uses an image-independent list of candidate regions sampled from the distribution of the bounding boxes in the dataset. The investigation found that the CNN architecture by itself, without the fully connected layers, contains sufficient geometric information (spatial information) for accurate object detection. However, the accuracy of the proposed system without the region proposals network was negatively impacted.

YOLO [44] (You Only Look Once) is a unified real-time object detection system with a design resembling GoogleNet [45]. Unlike previous detection approaches that modify classifier networks to perform detection, YOLO deals with the object detection task as a regression problem at which bounding boxes are spatially separated and associated with class probabilities. YOLO approach uses a CNN to predict both the class probabilities and the bounding boxes from an image. It is a one-stage process that needs one evaluation (forward pass) for predictions. Moreover, the network can be trained end-to-end.

YOLO process the entire image at once to extract contextual information, unlike sliding window and region proposals techniques where each area in the image is processed individually. Being able to process small areas in an image increases the system's accuracy, which is an advantage of multi-stage approaches. At the same time, it increases the processing time, which has a negative impact on the system's speed. On the other hand, YOLO has better performance when it comes to the detection of large context images, such as backgrounds. However, it struggles to localise small objects precisely.

YOLO system divides the input image into an $S \times S$ grid cells. Each grid cell is responsible for detecting an object if that object falls in its center. In addition, each grid cell predicts B

bounding boxes and the associated confidence scores. Confidence score can be defined as $P_r($ object $) \times$ IoU $^{\text{truth}}_{\text{pred}}$. If no object exists in the grid cell, then $P_r(object)$ equals to zero. However, if an object is detected by the grid cell, then $P_r(object)$ equals to one, and the confidence score equals to the IoU between the predicted and the ground truth box. Thus, each bounding box has five attributes: (x, y) are the coordinates of box's center with respect to the grid cell, (w, h) are the width and height of the object with respect to the image, and the confidence score [44].

The network only predicts one box for each object within a grid cell. If an object happens to occupy more than one cell, like a large object or near border objects, multiple cells can localise the object efficiently. For multiple detections, NMS can be used to remove boxes with low IoU. Unlike classifier-based approaches, YOLO network is very fast at test time. It requires a single forward propagation through the network to produce the predictions (Table 2.1).

On the other hand, YOLO approach can only predict two bounding boxes for each grid cell. In addition, each grid cell can only have one class. Because of this spatial constraint, the detection of close objects is limited. Consequently, the network struggle with objects that appear in groups.

Another limitation of YOLO model is the treatment of errors in the loss function. The main source of errors is incorrect localisations. The loss function treats errors of small bounding boxes in the same way as of large bounding boxes. While a small error in large boxes may be acceptable, it may greatly impact the IoU of small boxes.

Table 2.1 **Performance of different detector systems on PASCAL dataset [44].**

Detector	Training data	mAP (%)	FPS
Fastest DPM [46]	2007	30.4	15
R-CNN Minus R [43]	2007	53.5	6
Fast R-CNN [41]	2007+2012	70.0	0.5
Faster R-CNN VGG-16 [29]	2007+2012	73.2	7
Faster R-CNN ZF [29]	2007+2012	62.1	18
YOLO VGG-16 [44]	2007+2012	66.4	21

YOLO V2 [47], V3 [48] and V4 [49] are introduced to solve some of the challenges of YOLO V1 [44] and to enhance the detector performance. For instance, YOLO V2 uses Batch Normalisation (BN) [50], anchor boxes, and multi-scale training. Whereas YOLO V3 uses residual connections [51] and Feature Pyramid Network (FPN) [52] with three predictions at different layers to process the image at different spatial resolutions. YOLO V4 [49] uses a different backbone network called CSPDarknet53. The introduced backbone network uses Cross Stage Partial Network (CSPNet) strategy to partition the feature map of the base layer into two parts and then merges them through a cross-stage hierarchy. Split and merge strategy allows for more gradient flow through the network. Also, YOLO V4 uses PANet [53] instead of FPN to aggregate and accurately preserve spatial information. Comparisons of YOLO detectors performances on different datasets are presented in [54–56].

Single Shot Detectors (SSD) [57], inspired by RPN, MultiBox and multi-scale representation, adopts anchor boxes to overcome YOLO limits. Instead of a fixed grid adapted by YOLO, SSD use anchor boxes with different scales and aspect ratios to extract the bounding boxes of the objects. In addition, the predictions from multiple feature maps are fused to detect objects of various sizes. SSD adds several feature layers to the end of the network to predict the offset of the default anchor box and its confidence. With the usage of anchor boxes and data augmentation, SSD is able to outperform YOLO in terms of accuracy and speed. However, it struggles with small size objects, similar to YOLO. Thus, enhancements have been introduced to overcome this issue, such as using skip connection with deconvolutional layers [58] and improving the network structures [59].

RetinaNet [60] introduces two improvements over previous single-stage detectors. It uses FPN [52] and a novel focal loss instead of cross-entropy loss. RetinaNet has four main components: a) a bottom-up ResNet as a feature extraction network, b) a top-down feature pyramid network, c) a classification subnetwork, and d) a regression subnetwork.

The feature extraction network extracts feature maps at different scales. FPN up-samples the coarser feature maps from the pyramid's top to the pyramid's bottom and laterally merge the maps with the corresponding same spatial size maps from the feature extractor network. It augments the CNN to build a rich and multi-scale feature map from a single resolution input image. The classification and regression subnetworks predict the class scores and the bounding boxes, respectively.

Focal loss is introduced to deal with the foreground-background class imbalance. Class imbalance constraints the performance of single-stage detectors from surpassing two-stage detectors. Focal loss reduces the loss contribution from easy examples (examples the produce high detection probabilities), as small loss values from easy examples can misguide the model during training. Moreover, it increases the loss contribution towards correcting the misclassified examples. This can be done by adding a modulating term to the cross-entropy loss to boost the learning of hard examples. RetinaNet achieved state-of-the-art performance in terms of speed and accuracy compared to one- and two-stage detectors.

2.3.3 Small object detection

Small object detection represents a challenge for state-of-the-art detectors. These detectors are fine-tuned on datasets that contain large size objects. Moreover, the base networks of these detectors are trained on general datasets such as ImageNet [37]. Studies show that state-of-the-art models [61, 62, 23, 1, 63], in addition to standard datasets such as PASCAL [64] and Microsoft COCO [65], do not give much consideration to small object detection. The performance of these models on small size objects is not clearly investigated as the evaluation of these models with the focus on the detection of small size objects is limited [23]. Small objects occupy a few pixels of an image, resulting in few features to be utilised. Also, it is challenging to distinguish small size objects from the background due to the lack of distinctive shape information. Small size objects also require high localisation precision than large size ones. Further, the large stride steps of the convolutional filters in the CNN architecture may skip the small objects.

Moreover, the definition of small object size is not unified, which presents another challenge for researchers. Chen et al. [61] classified objects from the PASCAL dataset to be small if the ratio between the bounding box area to the image area, averaged over all the instances of that class, is in the range of 0.08% to 0.58%. This corresponds to 16×16 to 42×42 pixels. Small objects can vary in size according to the image size, which is not constant for the PASCAL dataset. To compare, the median relative areas of the PASCAL dataset objects are between 1.38% to 46.4% [61].

Torralba et al. [66] introduced a dataset for tiny images with 32×32 pixels. Zhu et al. [62] followed the definition of the Microsoft COCO dataset for small size objects to be equal to or less than 32×32 pixels. Microsoft COCO contains small objects, but they occupy large parts of the images. The variation in small size objects definition is attributed to the dataset image size. For the PASCAL dataset, the image size varies. Whereas for the Microsoft COCO dataset, the image size is fixed and is equal to 640×480. In the light of the previously mentioned definitions, the thesis follows a new combined definition for small size objects. An object is categorised as small if its size equals to or is less than 42×42 pixels. We adopted this definition as the image size in the proposed object detection dataset is 512×512 pixels and in the proposed semantic segmentation dataset is 960×540 pixels.

Chen et al. [61] study is one of the initial works that tries to enhance the performance of R-CNN on small size objects. The study introduces ContextNet, at which the region proposals and the context of the regions are forward propagated through two CNNs. Then the results of the two networks are concatenated. A limitation for the proposed system is that the two CNNs do not share any weights. Consequently, the system requires more training time and resources.

R-CNN is tested using the small object dataset [61], a subset of images from both Microsoft COCO [65] and SUN datasets [67]. The choice of R-CNN over Fast R-CNN is attributed to the

RoI pooling layer used in Fast R-CNN that maps the region proposals to further small maps. While this works with large objects and enhances the overall performance, it can negatively impact small ones. On the other hand, R-CNN resizes all the region proposal boxes to a fixed size, allowing the generation of full feature maps even for small objects. However, resizing region proposals may introduce artefacts.

The smallest anchor box used in the original implementation of Faster R-CNN is too large to accommodate the objects of the small size dataset. Thus, anchor boxes of the proposed R-CNN are modified with respect to the statistics of the small object size in the dataset [61]. Anchor boxes modification to accommodate small size objects is a well-known technique to enhance detectors' performance on small size objects [23]. In addition, anchor boxes can be attached to different convolution layers with smaller strides to avoid skipping small size objects. Consequently, the proposed technique has improved the performance of Faster R-CNN on the small size object dataset.

Several strategies have been introduced to enhance the detector performance on small size objects, such as feature learning, context-based detection, data augmentation, training strategies. In addition, Generative-Adversarial Networks (GAN) [68] achieved good results on the task of small object detection. Tong et al. [2] review deep learning methods for small object detection. The review highlights the following remarks: multi-scale feature learning, context modelling, and data augmentation can enhance the performance of state-of-the-art detection methods on the detection of small size objects. Input image resolution and base networks have a great impact on detection performance. The combination of multiple techniques to enhance object detectors can further improve the performance [69, 70]. Lastly, large datasets and the combination of multiple datasets can boost the detector to learn better representation of small size objects.

Data augmentation refers to increasing the number of images or instances of small size objects by image transformation that includes flipping, cropping, scaling, etc. The main idea is to extend the dataset with a large amount of data by increasing the representation of small size objects, which can help to boost the performance of detectors on small size objects [71].

The training strategy named Scale Normalization for Image Pyramids (SNIP) [69] can selectively backpropagate the gradient of object instances based on their sizes. This can help to focus the training on the object of interest.

Multi-scale feature representation combines the activations from multiple layers to aggregate the spatial resolution of different size objects. Multi-Scale Convolution Neural Network (MS-CNN) [72] is proposed with multiple scale-independent output layers to tackle the inconsistency between the objects sizes and the receptive fields. To better use the scale-independent convolutional features for small object detection, Scale-Dependent Pooling (SDP) and layer-wise Cascade Rejection Classifiers (CRCs) are introduced [73]. Aggregating and compressing the hierarchical feature maps are used by HyperNet [74] to calculate the shared features between RPN and object detection networks.

The detection performance can be improved by exploiting features from and around the RoI to deal with occlusions and small objects. Gated Bi-Directional Network (GBD-Net) [75] proposes the idea of gates that control the transmission of messages between different support regions. SegDeepM [76] uses Markov random field to exploit object segments and reduce the dependencies on initial candidate boxes. Multi-Region CNN (MR-CNN) [77] is proposed to capture multiple aspects of objects such as distinct parts and semantic features. Inside-Outside Net (ION) [70] is introduced to capture contextual and multi-scale representation features from inside and outside the RoI with spatial Recurrent Neural Network (RNN) [78]. The MulitPath architecture [79] is proposed by modifying Fast R-CNN to have a multi-scale skip connection [70], a modified foveal structure [77], and a novel loss function that sums different IoU losses.

GANs introduced by Goodfellow et al. [68] has shown great potential in small object detection. A typical GAN consists of two networks: a generator and a discriminator. The generator creates new samples, and the discriminator distinguishes between the generated data and the true one. The two networks aim to reach an optimised network that is immune to adversarial data. Perceptual GAN [80] boosts the detection of small size objects by generating super-resolved representation for small objects to narrow the representation difference between small size and large size objects. The discriminator competes with the generator to recognise the generated images with an additional requirement that the generated representation needs to benefit the process of detecting small size objects. Like the generator of the Perceptual GAN, the Multi-Task Generative Adversarial Network (MTGAN) [81] generator upsamples small blurred images into high-resolution clear images. The discriminator of the MTGAN, on the other hand, is a multitasking network that describes each image patch by a real or fake score, a category score and a regression offset.

Faster R-CNN struggles with small object detection and localisation due to the coarseness of its feature maps and limited information provided in candidate boxes. To tackle this issue, complementary information from multiple sources is needed to contribute to the network decision. Cao et al. [82] try to enhance Faster R-CNN performance on small object detection by proposing a new loss function based on the IoU, an improved NMS to avoid the losses of the overlapping objects, and a bilinear interpolation to improve the RoI pooling operation.

In chapter 5, we investigate the performance of state-of-the-art object detection systems on the proposed dataset. The detailed investigation of different detector architectures and different training strategies gives a roadmap to choose the best system for a given application.

2.4 Semantic segmentation

2.4.1 Series architecture

Fully Convolutional Network (FCN) [83] represents the fundamental of many state-of-the-art deep learning techniques for semantic segmentation. Besides, it represents the base of full scene understanding using deep learning. Semantic segmentation architectures can be divided into two main categories: series architectures and encoder-decoder architectures. Though, the latter architectures stem from the series ones.

FCN is considered the first work to train a network end-to-end for pixel-wise predictions using supervised pre-trained networks. It adapts state-of-the-art classification networks such as AlexNet [36], VGG [84] and GoogleNet [45] to make use of the learned features by these networks on classification tasks and transfer them to semantic segmentation tasks through transfer learning techniques [38, 39] and architecture modifications. Architecture modifications include replacing all the fully connected layers with convolutional ones and in-network up-sampling to the original input image size. FCN does not make use of pre- or post-processing complications such as super-pixels, region proposals or post-hoc refinement by random field or local classifiers [85, 86].

Although FCN architecture has achieved a high score on standard metrics (mean pixel Intersection over Union), the produced semantic segmentation output is unrefined. Spatial details are not accurate, and object boundaries are not well-defined. It does not comprise useful global context information, instance awareness is not presented, and performance is far from real-time execution. Also, it is not entirely suited for unstructured data such as 3D point cloud [3, 4].

The main challenge facing semantic segmentation is the tension between semantics and locations (global and local information). Many solutions have been proposed to integrate context knowledge, such as Conditional Random Fields (CRFs), dilated convolutions and multi-scale predictions. DeepLab [87, 88] makes use of CRFs to refine segmentation results and object boundaries as a separate post-processing stage. Dilated convolution, also known as atrous convolution, is used in DeepLab [87–90] to enhance the output resolution. Also, multi-scale context aggregation [91] makes use of dilated convolution. Dilated convolutions support expanding receptive fields without trading-off the resolution. They allow efficient dense feature extraction on any arbitrary resolution. Besides, multi-scale sub-networks with different output resolution are proposed to refine the coarse prediction progressively [92].

Skip architecture is introduced in FCN [83] to overcome the global/local information dilemma. Skip design combines 'fuses' semantic information from deep, coarse layers with appearance 'context' information from shallow, fine layers to produce detailed segmentation. By doing so, the model becomes capable of making local predictions in the sense of the global structure. Skip connections convert the series architecture of the FCN into a DAG one (Directed Acyclic Graph). Skip architecture weights are learned end-to-end to refine the semantics and spatial accuracy of the output [83].

2.4.2 Encoder-decoder architecture

On the other hand, there is the encoder-decoder network architecture. Many state-of-the-art semantic segmentation architectures follow this design such as U-Net [93], SegNet [94] and DeepLab version 3 plus (DLV3+) [90]. U-Net [93] is inspired by FCN [83] with some modifications to yield precise segmentation with few training images. The main architecture modification is in the addition of the decoder part (up-sampling), where a large number of feature channels allow the network to propagate context information to high resolution layers.

U-Net is trained end-to-end and outperforms the sliding window based convolutional network [95, 35] in terms of accuracy and inference speed. The system has achieved high performance on biomedical image segmentation applications using a few annotated images thanks to the data augmentation techniques. It is also promised to provide high-quality results on other segmentation applications.

Both DeconvNet [96] and SegNet [94] use VGG-16 [84] as their feature extraction network (encoder part). Unlike DeconvNet, SegNet discards the fully connected layers of the VGG-16 architecture. The decoder part of the DeconvNet consists of deconvolution and un-pooling layers [96]. However, the SegNet decoder part recalls max-pooling indices from the corresponding encoder layer during the up-sampling process, unlike U-Net [93] which transfers the entire feature maps from the encoder to the decoder. This makes SegNet fast in both training and testing with a small model size and memory footprint.

DLV3+ [90] follows the encoder-decoder structure. It uses DeepLabV3 (DLV3) [89] as the encoder attached to it a simple yet effective decoder module. DLV3 and DLV3+ avoid using CRF as it is a post-processing stage that obstructs the network from end-to-end training, unlike their ancestor systems DeepLabV1 (DLV1) [87] and DeepLabV2 (DLV2) [88] which can be considered as two cascade modules systems (Deep Convolution Neural Network (DCNN) then CRFs). DLV3+ introduces atrous separable convolution, which is composed of a depthwise convolution (spatial convolution for each input channel) followed by a pointwise convolution (1×1 convolution to combine the output from depthwise convolution). This leads to a significant reduction in computation complexity. Atrous separable convolution is applied to both Atrous Spatial Pyramid Pooling (ASPP) and decoder modules. ASPP is introduced in DLV2 inspired by the spatial pyramid pooling method [40] to capture objects and context at multiple scales.

The decoder part of DLV3+ is simpler than that of U-Net [93] and SegNet [94]. Encoder features are first bilinearly up-sampled by a factor of 4 and then concatenated with the corresponding low-level features. A 1×1 convolution reduces the number of channels of the low-level features before concatenation. After concatenation, a few 3×3 convolutions are applied to refine the features, followed by another bilinear up-sampling by a factor of 4. This strategy is better than directly up-sampling the features by a factor of 16 as it reduces the required computations (the number of trainable parameters). Besides, it allows multi-scale features to propagate through multiple layers of the decoder part. Consequently, better information can be extracted from the images.

In chapter 6, scene understanding systems based on semantic segmentation techniques are proposed. Further, novel shared systems that can process data from different distributions are introduced. The proposed systems are practically implemented and tested with real users to evaluate the level of assistants which can be offered to visually impaired disabled users.

2.5 Explainable artificial intelligence

Visualisation techniques are powerful tools to explain the behaviour of AI systems. They can be used to identify important features that contribute to decisions, investigate biases in datasets, and find mistakes in structural elements of the system (e.g., network architectures). This is vital for safety-critical applications such as autonomous navigation and operation systems (e.g., autonomous trains or cars), where prediction errors may have serious implications. Lawmakers and regulators may not allow the use of such systems if they cannot explain the logic underlying a decision or an action taken. These systems are required to offer a high level of 'transparency' to be approved for deployment. Thus, being highly accurate without being able to explain the basis of their performance will not satisfy the regulatory requirements [97–99]. The lack of system interpretability is a major obstacle to the wider adoption of AI in safety-critical applications. Explainable AI (XAI) techniques to visualise CNN predictions, so that the system can reason about its decisions, offer possible solutions.

Explanation methods, such as decision trees [100], are powerless to explain very deep CNN behaviour. Our focus is on visualisation methods because CNN visualisation is the direct way to investigate network decisions and representations. Also, visualising CNN filters can be performed at different network locations. In the next subsections, hidden layers and post-hoc visualisations are discussed.

2.5.1 Hidden layers features visualisation

High-layer filters in traditional CNNs can describe a mixture of patterns. Consequently, it is challenging to understand the contribution of each part of the object. Zhang et al. [101] propose a method to modify a CNN to be more interpretable by training high convolutional layer filters to represent a specific part of an object without any additional object-specific annotated data.

A software tool [102] is introduced to enable the visualisation of the channel's activations of convolutional layers in the same spatial layout as the input where each filter is activated by a specific feature or pattern such as edges, faces, eyes, etc. Using the proposed software, pooling and normalization layers can be visualised, which can reflect their impact on the model's behaviour. Real-time visualisation of all filters of a specific layer on one screen is a very informative approach as it can display all the data propagating through a CNN.

Visualising a trained model using DeconvNet [103] can help to select better architectures. For example, by visualising the first and second layers of AlexNet architecture [36], it was noticed that the first layer filters are a mixture of high and low-frequency information. Whereas the visualisation of the second layer filters show aliasing artefacts caused by a large stride $(s = 4)$ that is used in the first convolutional layer. A new architecture is proposed to overcome these problems by reducing the filter size in the first convolutional layer from 11×11 to 7×7 and reducing the stride to 2 instead of 4. Consequently, the new system retained more information in the first and second convolutional layers and achieved better accuracy.

2.5.2 Visualisation of output layer activations (post-hoc visualisation)

As an input pattern cause a given activation in the feature maps, Zeiler et al. [104] map this activation back to the input pixel space using deconvolutional networks [103]. The steps can be explained as follows: an input image is presented to the CNN, and the features are computed through the networks' layers. To analyse a given activation, all other activations in that layer are set to zero. Then the feature maps are passed to the attached deconvolutional layer. Finally, the input pixel space is reached through successive un-pooling, rectifying, and filtering operations to reconstruct the layer's activity.

The DeconvNet approach recalls the position of the max-pooling layers' values during a forward pass by storing these values in switches. The activations are then copied into the positions indicated by these switches during the deconvolutional process, while other lower layer activations are set to zero. Switches are introduced as the max-pooling operation is non-invertible.

A drawback of the DeconvNet method is that the image-specific information comes from max-pooling layers (switches). The absence of pooling layers will result in non-image-specific explanations. Also, negative pieces of evidence are discarded during the backpropagation process due to ReLU units which may result in less informative heatmaps [105].

Occlusion sensitivity [103] is introduced to make sure that the object itself is the element that activates the network and not the context or the background, as well as, to show the ability of the model to locate the object in an image. This can be attained by occluding different portions of the input image with a grey square in a sliding window manner and monitoring the classifier's output. The system clearly shows its ability to localise the target object within an image as the correct class probability dropped significantly when the object of interest is occluded.

Gradients approach [106], also known as backpropagation or saliency method, visualises the derivatives of the target object score with respect to the input image. Saliency maps are generated for the trained network and not during the training process (i.e., the networks' weights are constant). Backpropagation is the process of increasing or decreasing networks' weights to minimise the loss function during the training process [107]. Saliency maps return the spatial discriminative pixels locations of a particular class in an image.

Saliency maps can be computed as follows: through backpropagation, the class score derivatives are calculated w.r.t the input image. Then, the saliency map values are arranged in the same order as the input image pixels, i.e., $m \times n$ derivatives matrix will have the same indices as $m \times n$ input images pixels where m and n represent rows and columns of a grey-scale image, respectively. Suppose the input is a multi-channel image such as RGB images. In that case, the maximum derivative magnitude is selected across all the channels to produce a single class saliency value for each pixel. Finally, the derivatives matrix is plotted to produce the saliency map. Saliency maps need one backpropagation pass to be produced. They can be considered a weakly supervised approach for object localisation.

Backpropagation ‘saliency’ approach can be considered a generalisation of DeconvNet [103] as it can be used to visualise any layers’ features and not only the convolutional ones. DeconvNet is equivalent to the gradient approach through a CNN except for the backpropagation through the ReLU layers.

Although Gradient heatmaps are computationally faster than Occlusion as it only needs one backward propagation through the network, they do not fully explain the image prediction. The calculated map measures pixels change that would make an image belong to a specific category. However, it does not explain the classifier decision as argued by [105] or the direct relation to the variation of the output [108, 109].

DeconvNet approach [103], which zeros negative values of the top gradients, and backpropagation [106], which zeros negative values from the bottom inputs, are then combined to produce Guided Backpropagation (GBP) [110] which zeros both negative values. The signal from higher layers guides the backpropagation; hence the name is derived. It works as the switches of the DeconvNet approach [103]. Doing so prevents negative gradients from flowing back, which can undesirably impact the activations visualisation.

Similar to GBP, DeSaliNet [111] combines both advantages of DeconvNet, which can accurately reproduce image boundaries, with the saliency method, which can localise objects efficiently. It can be noticed that DeconvNet [103], Backpropagation [106], and GBP [110] use almost the same steps to produce visualisation maps, although they are described in different ways. The main difference is in how they handle the gradients through the ReLU layers. DeconvNet allows only positive derivatives to backpropagate (i.e., applying ReLU operation to the gradients). Backpropagation passes only the positive elements corresponding to the preceding feature map (from the lower layer). GBP combines both techniques. Figure 2.2 depicts the difference.

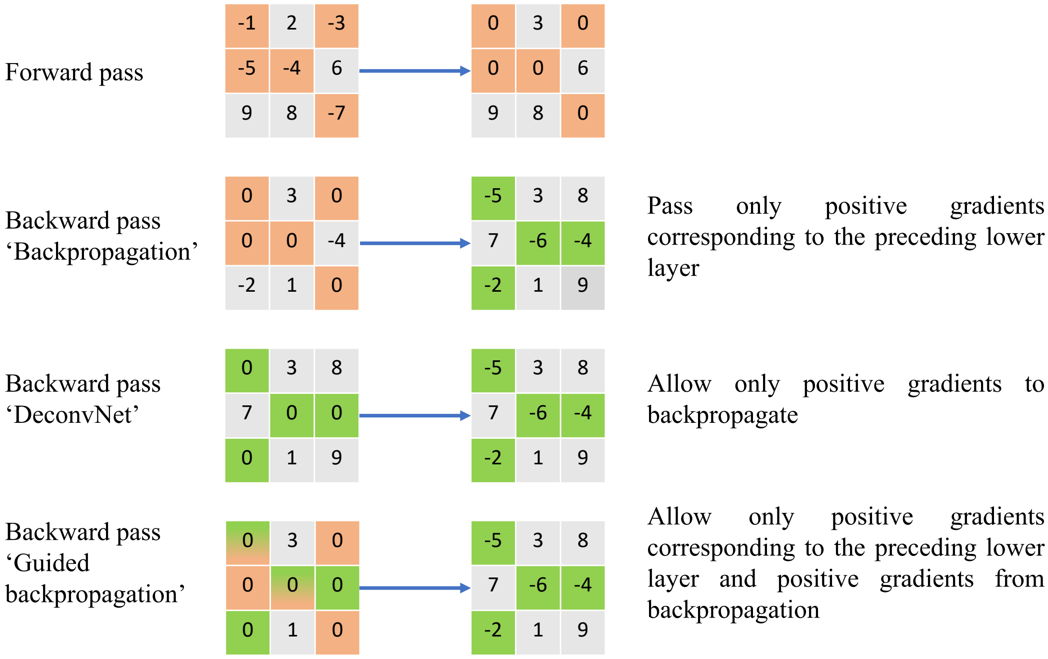


Fig. 2.2 Main differences between backpropagation, DeconvNet and Guided backpropagation approaches (reproduced from [110]).

Many approaches based on Gradients (eq. (2.1)) are proposed, such as element-wise products of gradients and input (GI) [112] (eq. (2.2)), Integrated Gradients (IG) [113] (eq. (2.3)), Smooth Gradients (SmoothGrad) [114] (eq. (2.4)), etc. Gradients of the output score are calculated w.r.t input and then multiplied by the input to enhance the heatmap resolution [112]. Moreover, GI can be used to address the gradient saturation problem [112]. Although this technique can visually enhance the produced maps, this may be attributed to the original image's quality rather than the visualisation technique [115].

$$\frac{\partial Y^c(x)}{\partial x} \tag{2.1}$$

$$x \odot \frac{\partial Y^c(x)}{\partial x} \tag{2.2}$$

$$(x-\bar{x})\int_{\alpha=0}^{1} \frac{\partial Y^c(\bar{x}+\alpha(x-\bar{x}))}{\partial x} d\alpha \tag{2.3}$$

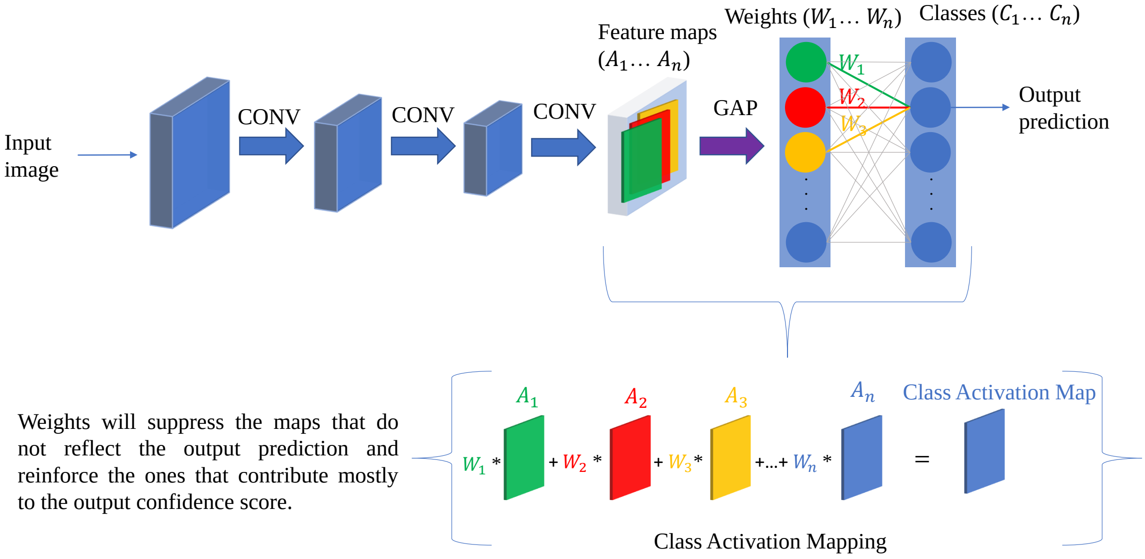
$$\frac{1}{n}\sum_{1}^{n} M_c\left(x+g\left(0,\sigma^2\right)\right) \tag{2.4}$$

$$\begin{aligned}
x &: Input \\
\bar{x} &: Baseline \\
Y^c &: Output\ prediction\ for\ class\ c \\
n &: Number\ of\ samples \\
M_c &: Class\ activation\ map\ for\ class\ c \\
g\left(0,\sigma^2\right) &: Gaussian\ noise\ with\ standard\ deviation\ \sigma
\end{aligned}$$

The Integrated Gradients [113] approach accumulates gradients over scaled-up versions of the input that follow a baseline defined by the user, i.e. it integrates the gradients of all points that fall on the straight-line path from the baseline to the input. The Smooth Gradients [114] approach uses added noise to enhance heatmap sharpness by averaging the explanations of noisy input copies. As Gradient sensitivity maps tend to be noisy due to the noisy gradients, SmoothGrad reduces visual noise by sampling similar images with added noise and then taking the average of the resulting sensitivity maps.

The term Class Activation Map (CAM) has been used to refer to the weighted activation maps generated for an image. Global Average Pooling (GAP) layer is introduced to generate accurate discriminative localisation. Though GAP is not a novel technique, its utilisation to produce heatmaps is a major contribution [116]. The intuition behind using GAP is that it helps the network to identify the complete scope of the object [116]. Unlike global max pooling where the localisation is limited to a point lying on the object's boundary [111].

CAM technique displays a heatmap representation that highlights image pixels which trigger the CNN to categorise an image to a specific class. Primarily, the approach maps the predicted class score back to the previous convolutional layer. GAP layer outputs the spatial average of the feature map of each unit at the last convolutional layer. A weighted sum of these values is used to generate the final output. The process can be summarised as follows: after the last convolutional layer of a typical CNN, the GAP layer takes the convolutional layer channels as an input and return their average as an output. Each output per category is assigned a weight. Then, a heatmap is generated per class output, and the weighted sum is calculated for all the heatmaps. Finally, the CAM is up-sampled to the image input size (Fig. 2.3).



Weights will suppress the maps that do not reflect the output prediction and reinforce the ones that contribute mostly to the output confidence score.

Fig. 2.3 Class Activation Map (CAM) generation process (reproduced from [116]).

Grad-CAM uses the gradient information that is passed to the last convolutional layer to assign importance weights to each neuron. The main difference between CAM [116] and Grad-CAM [117] is in the way of generating the weights for the feature maps. In CAM, heatmaps are generated by taking the weighted average sum of the last convolutional layer activations using the Fully Connected (FC) layer's weights, which is connected to the output class. Whereas in Grad-CAM, the gradients of any layer are used to generate these weights.

The Grad-CAM approach can be summarised as follows: first, gradients of the score for a specific class are computed w.r.t. feature map activations of a convolutional layer. Then, to obtain significance weights for each feature map, the computed gradients are global average pooled. Finally, the forward activation maps are weighted and combined 'weighted summed' followed by a ReLU operation (Fig. 2.4). ReLU is used to highlight features that have a positive effect on the class of interest. These features represent pixels intensities that contribute to the class gradients. Negative influence pixels usually belong to a different class; that is why they need to be suppressed using a ReLU function to obtain better localisation. The final result is a coarse heatmap of the same size as the final convolutional layer feature map.

Grad-CAM can be considered as a generalisation of CAM, or CAM is a special case of Grad-CAM. On the other hand, Grad-CAM cannot highlight fine-grained details. Pixel space gradient visualisation methods such as GBP [110] produce higher resolution visualisations than Grad-CAM [117]. To counter this problem Guided Grad-CAM technique is introduced. Grad-CAM and GBP techniques are combined by point-wise multiplication to produce high-resolution (fine-grained details) and class discriminative (class regions) maps (Fig. 2.4).

Localisation error is argued to be a descriptive metric for assessing saliency methods. Table 2.2 shows the performance of different saliency methods on the ImageNet validation dataset for localisation.

Table 2.2 **Localisation error of state-of-the-art saliency methods on the ImageNet validation dataset.**

Approach	Localisation error (%)
Gradients [106]	41.7
Guided Backpropagation [110]	42.0
CAM [116]	48.1
Grad-CAM [117]	47.5
Occlusion [103]	48.6

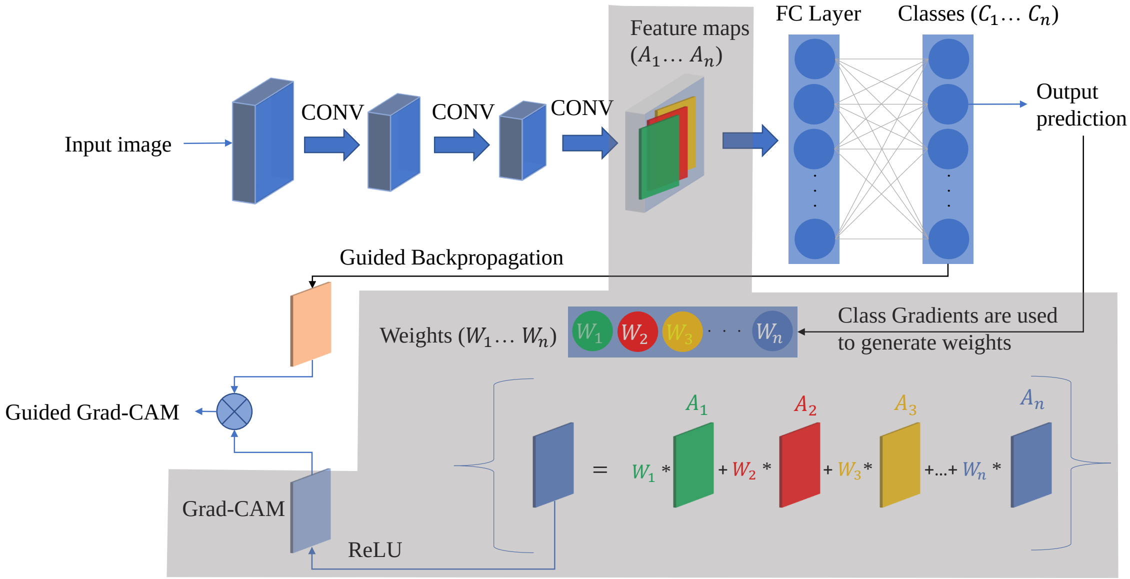


Fig. 2.4 Grad-CAM (grey-shaded) and Guided Grad-CAM approaches (reproduced from [117]).

Results in Table 2.2 are reported in [118–121], following the same evaluation protocol as [118] and using the same CNN (GoogleNet). The evaluation process on the localisation task is as follows: given an image, the class of interest, and the corresponding saliency map, the object segmentation mask is computed by thresholding the foreground area to cover 95% energy out of the produced saliency map. Then, the tightest bounding box that contains the whole object in the saliency map is calculated as the result localisation bounding box. This localisation box is only considered valid if the Intersection over Union (IoU) with the ground truth bounding box is greater than 0.5.

Fig. 2.5 summarises all the reviewed visualisation techniques. It can be seen that all the methods are spin-off two main methods: Gradients and CAM.

Visualisation techniques are essential tools to understand CNN behaviours. Reliable systems based on deep learning techniques need to reason about their predictions. For this reason, we ensure the transparency of the proposed systems to accelerate their approval for real-life applications.

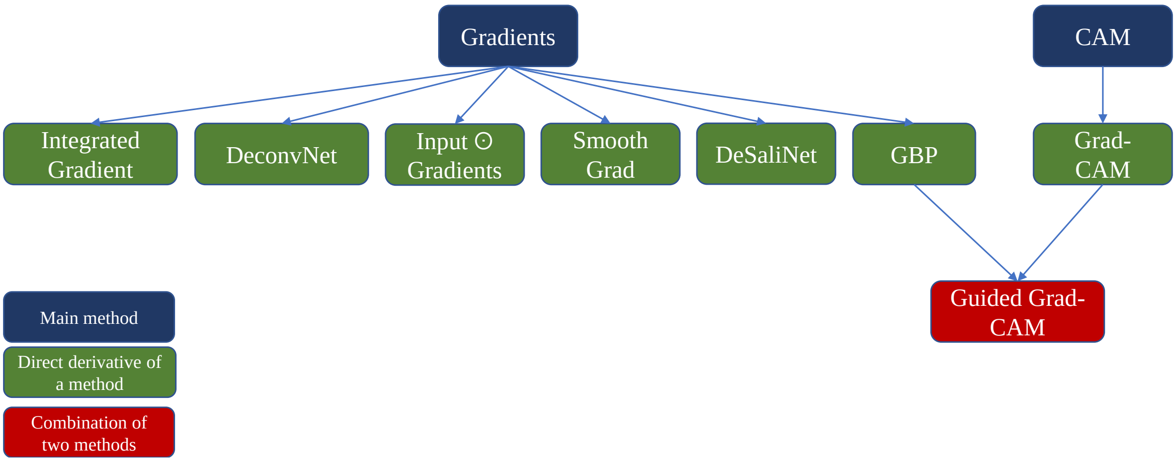


Fig. 2.5 A chart of post-hoc visualisation techniques.

In chapter 4, we introduce novel visualisation techniques that inherit the advantages of both gradient-based and CAM-based techniques. The proposed techniques are applied to the developed object detection systems to ensure their reliability. Applying visualisation techniques to detection tasks is a novel contribution as visualisation techniques are usually applied to classification tasks.

2.6 Discussion and conclusion

The background studies highlight the need for such a system that can enhance the independence of visually impaired disabled users. The system needs to act as a guide to assist when needed and not as a caretaker who controls the whole navigation process. Computer vision systems based on deep learning techniques could be optimal solutions because of their efficiency and high performance.

It is challenging to decide which detector architecture is best suited for a given application. Standard detection metrics do not tell the complete story. Real-time processing, accuracy, and memory usage are critical for all applications. State-of-the-art systems, however, cannot achieve all criteria at once. There will always be a trade-off between accuracy and speed [122]. Thus, researchers and users need to decide the best system for a given application.

For the object detection task, we follow the recommendations from the previous studies by redesigning the anchor boxes, using feature pyramid networks, using residual blocks, using high-resolution input images and following different training strategies to achieve the best results on the proposed dataset. We also use explanation methods to ensure the proposed systems' transparency and robustness.

Moreover, we increase the resolution of the images and introduce a second dataset for semantic segmentation to achieve better results and enhance human-system interaction. We conduct the experiments on the pixel level to better utilise every pixel in the image. This allows the extraction of accurate information, such as the distance to the target object. Also, geometric information can be obtained, which facilitates the interaction with the surrounding environment. Further, we propose novel systems that can simultaneously segment indoor and outdoor images to solve the challenge of processing data from different distributions. The proposed systems can process different context images without the need to retrain a new system on a new combined dataset.

Chapter 3

Semantic Segmentation and Object Detection Datasets

3.1 Introduction

The main purpose of the proposed datasets is to provide ground truth annotated data for objects of interest to powered wheelchair users who may need to interact with on a daily-life basis. Since the widely available datasets contain only general objects, these datasets are introduced to cover the missing pieces. Although the proposed datasets can be considered application-specific, the introduced objects are not only important for powered wheelchair users but also for indoor navigation and environmental understanding. For example, indoor assistive and service robots need to comprehend their surroundings for effective navigation and interaction with different size objects.

The dataset's objects are chosen because they represent a typical indoor environment that a powered wheelchair user needs to interact with during daily activities. The types of the objects could, of course, be expanded or customised based on the user's needs and the surrounding environment, either by utilising objects from publicly available datasets if they exist or by collecting new data. Many studies and experiments concerned with powered wheelchair navigation and training of new powered wheelchair drivers used the same classes of objects that we introduced in the dataset, such as doors, door handles, light switches, etc., in real and simulated environments [123–125]. However, the proposed dataset has a specific advantage as it introduces three different types of door handles, unlike the generic door handle class presented in the standard datasets. This could be important information to define how a wheelchair user should approach a door to open it. Also, other indoor objects such as fire extinguishers, key slots and push buttons are added to enrich the realistic applicability of the proposed dataset.

The proposed object detection dataset is collected in different indoor environments. A handheld camera is used for the data collection process to enhance the collected images' diversity and perspective. Moreover, the object detection dataset has been collected from four different indoor environments. The collected images are annotated on the bounding box level for the object detection tasks.

On the other hand, the proposed semantic segmentation dataset is recorded using a camera installed on a powered wheelchair. The camera is installed beneath the joystick to have a clear vision with no obstructions from the user's body or legs. The powered wheelchair is then driven through the corridors of the indoor environment, where videos are recorded. Then, the collected videos are annotated on the pixel level for the semantic segmentation (pixel classification) task. Pixel level annotations allow the extraction of detailed information of the target object, such as the geometric shape. Moreover, it facilitates the interaction with the target object, for example, by estimating the distance to the object's centre.

MATLAB software is used to annotate the datasets. The datasets have various object sizes (small, medium, and large), which can explain the variation of the bounding boxes and pixels distribution in the object detection and the semantic segmentation datasets. Usually, Deep Convolutional Neural Networks (DCNNs) that perform well on large size objects fail to produce accurate results on small size objects. Whereas training a DCNN on a multi-size objects dataset can build more robust systems.

Although the recorded objects are vital for many applications, more images of different kinds of door handles with different angles, orientations, and illuminations are included because they are rare in the publicly available datasets.

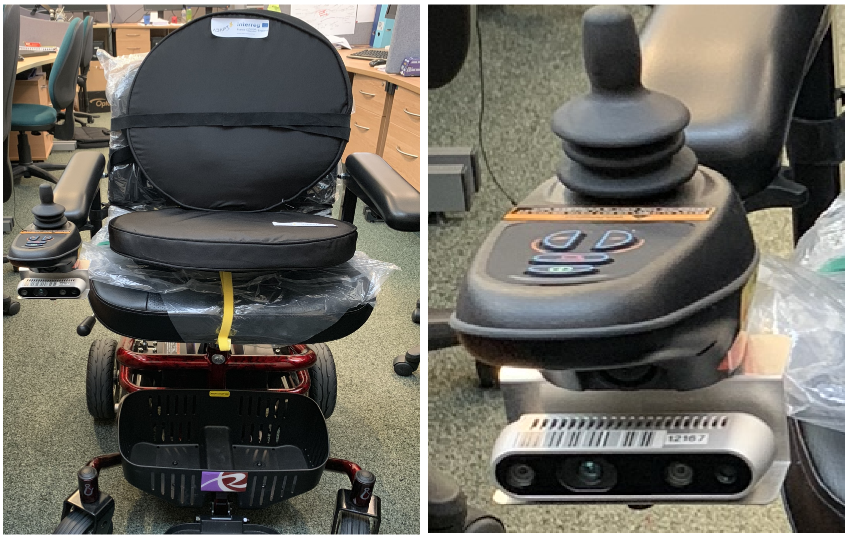
The proposed semantic segmentation dataset has 1549 images and covers nine different classes. In comparison, the proposed object detection dataset has 3292 images and covers eight classes. We used the datasets to train and test semantic segmentation and object detection systems to aid and guide visually impaired disabled users by providing visual cues. The Semantic segmentation dataset is made publicly available [126]. At the same time, the object detection dataset is under preparation for public release.

This chapter introduces the two datasets, highlights their importance, and discusses their differences. Section 3.2 presents the data collection setup. Section 3.3 discuss the motivation behind the proposed datasets. Semantic segmentation and object detection datasets are presented in sections 3.4 and 3.5, respectively. Lastly, the chapter is concluded in section 3.6.

3.2 Data collection

For the semantic segmentation dataset, an Intel® RealSense depth camera is installed on the Roma Reno II powered wheelchair for data collection and inference (Fig 3.1). Electrical Powered Wheelchairs (EPWs) have limited positions where a camera can be integrated or placed. The size of the EPW constrains these positions. Also, placing a camera on an EPW should not be obscured by the driver's body, legs, or hands. We proposed two locations that can be used for this purpose. The first option is a camera installed below the joystick controller, as shown in Fig 3.1b. The second one is a camera installed on a stick or a holder that can be extended above the driver's head. There might be other places depending on the EPW type and design. For each case, a video has been collected. Each of them is recorded in two different environments to capture a different perspective and trajectory. We annotated and used the first video.

On the other hand, we used a web camera connected to a laptop to record the object detection dataset. The simple setup helps to collect more images at different locations with different perspectives. It is also more convenient for other collaborators to use the same setup to capture more images from different environments.

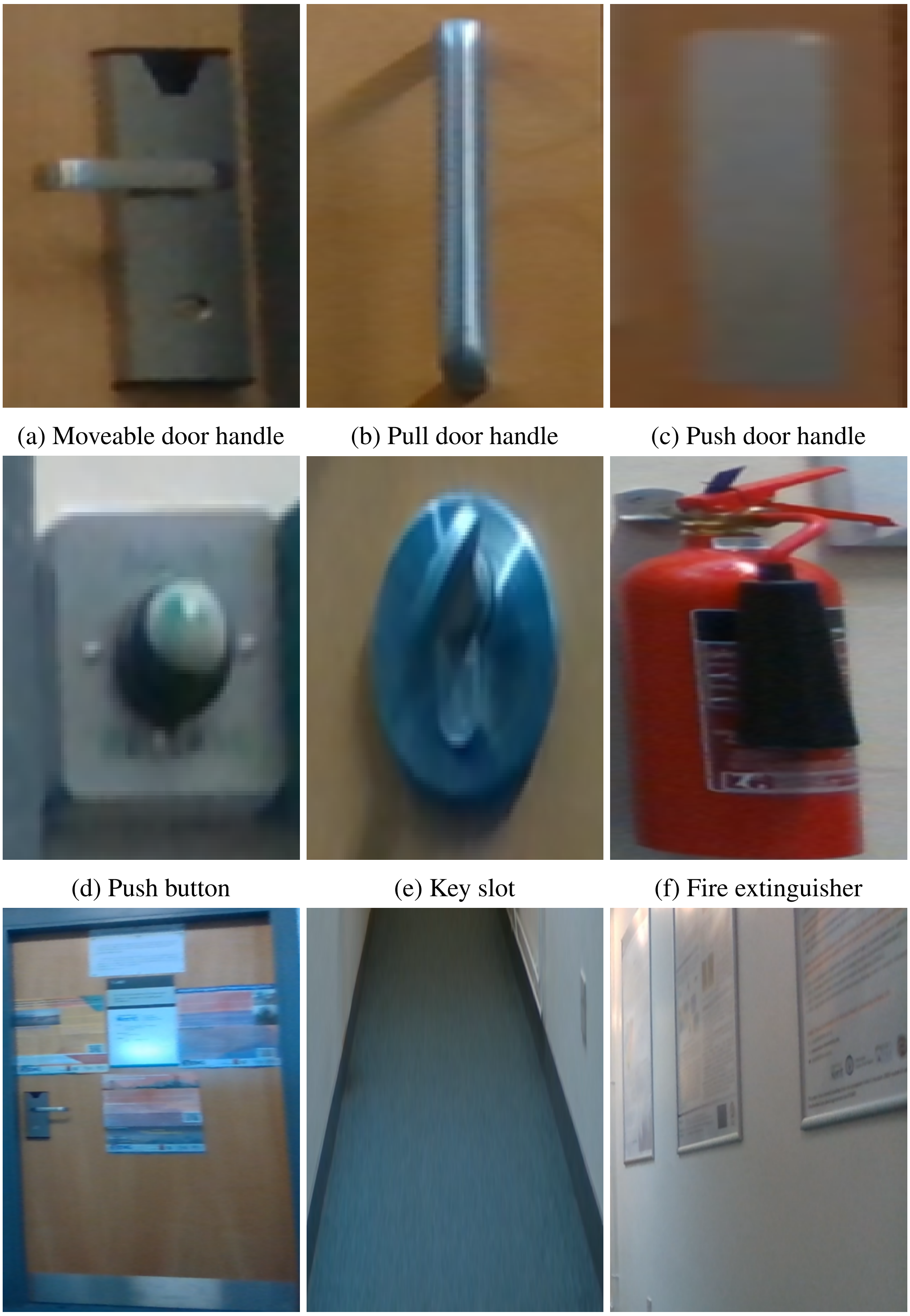


(a) Roma Reno II EPW (b) Intel® RealSense Camera

Fig. 3.1 The camera is installed beneath the EPW's joystick so that no interference with the users' legs can obstruct vision.

3.3 Motivation

Available standard datasets [127–131] contain general objects of indoor environment but lack objects related to the proposed application. For example, the door handle class in the aforementioned datasets is generic. Whereas the proposed indoor datasets contain different kinds of door handles for better perception and human-system interaction. So, collecting and annotating a task-specific dataset is a non-avoidable requirement. Objects of interest are doors, floors, walls, fire extinguishers, key slots, switches, and different kinds of door handles such as moveable, pull and push door handles. Fig 3.2 shows the classes of interest of the proposed datasets.



(a) Moveable door handle (b) Pull door handle (c) Push door handle

(d) Push button (e) Key slot (f) Fire extinguisher

(g) Door (h) Carpet floor (i) Background wall

Fig. 3.2 **The classes of the proposed dataset.**

These objects are not only important for EPWs users but also for any robotic platform. Any robotic platform which uses particular actuators to open a door would require information about the type of the door handle in order to engage a suitable strategy for opening the door. For example, pull door handles require different actuation than moveable door handles. These classes represent the main objects an EPW user may need to interact with or utilise on a daily life basis. Other classes of interest can be added later depending on the user's ability and the surrounding environment.

The proposed indoor datasets can be augmented using some classes from the ADE20K [127, 128], NYU depth [129, 130] and SceneNN [131] datasets which have objects of the indoor environment. However, specific classes, such as door handle types, do not exist in these datasets. These classes, besides key slot and switch classes, are infrequent. Nevertheless, they are important for many applications, especially indoor navigation. To keep a class distribution balance, abundant classes such as doors, floors and walls are not included from the standard datasets. However, more objects from standard datasets may be included to create a customised implementation of the system upon the user's need and the adequate distribution of important task-oriented labels. This may require a system retrain to tune the system's weights on the extended dataset.

Unlike the well-known datasets [132, 65], which usually have one big object per image or contain small objects but occupy large parts of the image, the proposed dataset has many objects per image; some of them can be categorised as small objects such as door handles and switches. In addition, these small objects are not available in the aforementioned datasets. This needs novel approaches that can produce high accuracy and precise edges. Using high resolution and large size images may help to tackle this problem as many pixels can be utilised and contribute to the object's classification. However, this would require higher computations than smaller and fewer resolution images.

3.4 Semantic segmentation dataset

While driving the EPW through the indoor environment, a one-minute video is recorded and annotated manually on the pixel level. Images extracted from the video are shuffled and split randomly into 70% for training (1084 images), 15% for validation (232 images) and 15% for testing (233 images). The resolution of the extracted images is 960×540×3 pixels. We train the semantic segmentation systems on high resolution and large size images, unlike the original implementation of DLV3+ [90] which crops patches of 513×513 size from the PASCAL VOC dataset [132] images during training and testing. Examples of the collected data with ground truth annotations are shown in Fig 3.3. Pixels that do not fit into any of the eight predefined classes are assigned to an extra class called the ‘Background Wall’ class. At the same time, small areas between two different classes, such as door frames or cupboards, are kept unannotated (void pixels). These pixels cannot fit in one class, such as the ‘Background Wall’ class, as they belong to different categories of objects.

The proposed dataset images might look homogeneous as it has been collected from one trajectory. Many factors can affect the perspective of the captured dataset, such as the camera installation, which is constrained by the available space on the EPW. However, we captured different angles, directions and orientations of small and rare objects of interest under different illuminations. Fig 3.4 shows the front, side, and partial views of moveable door handles captured during the data collection. Moreover, data augmentation is employed during training which gives another dimension for the dataset. Data augmentation techniques help to enhance the model’s robustness and increase the model’s ability to generalise to other environments. Furthermore, the dataset will be extended along with the study and future work.

It can be noticed from Fig 3.5 that categories such as Doors and Background walls dominate the distribution of the pixels. Whereas door handles have fewer pixels. This can be attributed to the objects’ sizes. Though, the dataset has many object instances of all classes (Table 3.1).

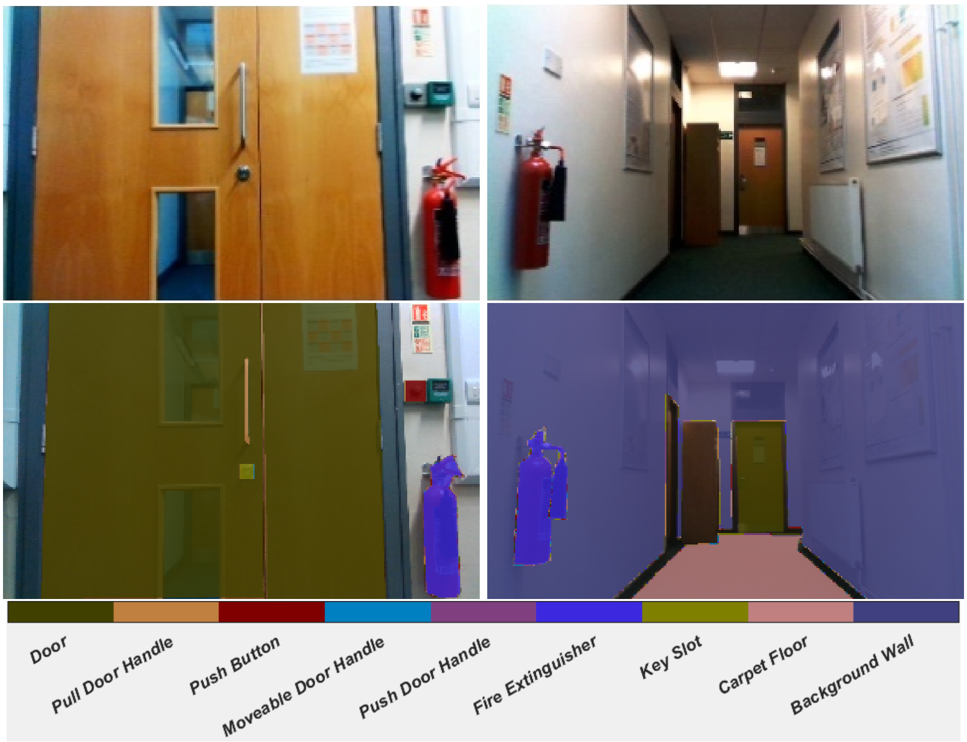


Fig. 3.3 Examples from the collected indoor dataset with the first row represents the original images and the second one represents the annotated ones.

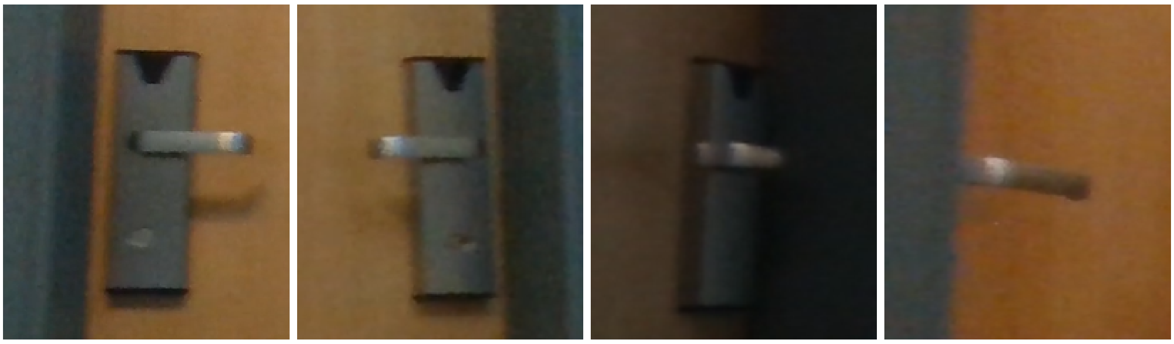


Fig. 3.4 **Moveabe door handles.** Although the dataset objects might look similar, we collected different angles and orientations of rare classes under different light conditions.

3.5 Object detection dataset

The object detection dataset contains 3292 images. Unlike the semantic segmentation dataset collection setup, the simple setup of the data collection system for object detection allows more images to be captured from various environments. However, the perspective from a handheld camera is different from that of a camera installed on an EPW. Images of the object detection

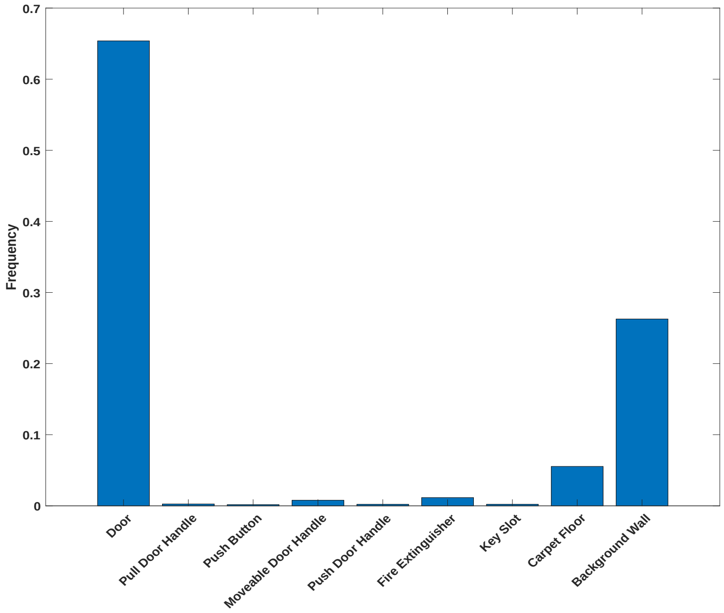


Fig. 3.5 **Pixels distribution of the proposed semantic segmentation dataset objects.**

Table 3.1 **The number of annotated pixels per class and object instances.**

Class	Pixel count (Million)	Number of instances
Door	239.87	1742
Pull door handle	0.95	173
Push button	0.63	159
Moveable door handle	2.87	1134
Push door handle	0.78	262
Fire extinguisher	4.25	486
Key slot	0.78	216
Carpet floor	20.32	698
Background wall	96.40	398

dataset are shuffled and split randomly into 60% for training (1975), 10% for validation (330 images), and 30% for testing (987 images). The split ratio is different from that of the semantic segmentation dataset, as the object detection dataset contains more images. The image resolution is 512×512×3 pixels. The reason for using low resolution compared to the semantic segmentation dataset is attributed to the available resources of the training environment. Still, the used resolution is better or comparable to the standard datasets.

The object detection dataset is annotated on the bounding boxes level. Examples of the collected data are shown in Fig. 3.6. Objects are categorised into eight classes, similar to the semantic segmentation dataset. However, the object detection dataset does not contain the 'Background wall' or the 'Carpet floor' classes. Nevertheless, it contains the 'ID reader' class. Objects that do not fit into any of the predefined classes are kept unannotated.

The number of object instances per class and the number of images which contain that object are shown in Table 3.2. The highest number of instances is for the 'Door' class. Whereas the lowest is for the 'Push button'.

Table 3.2 **The number of instances per class and the image count.**

Class	No# of instance	No# of images
Door	3443	2548
Pull door handle	362	227
Push button	92	92
Moveable door handle	2035	1797
Push door handle	437	367
Fire extinguisher	536	536
Key slot	826	763
ID reader	500	485

3.6 Conclusion

In conclusion, we propose semantic segmentation and object detection datasets to fill the gap for important objects that a powered wheelchair or any robotic platform may need to interact with or utilise. These objects do not exist or are rare in standard datasets, which motivates the collection of these datasets. Moreover, the semantic segmentation dataset is made publicly available for benchmarking by other researchers.

We believe that the proposed datasets cover important classes. Segmentation and detection of such objects for manipulation or scene understandings can enhance human-system interaction and improve the independence of disabled users.

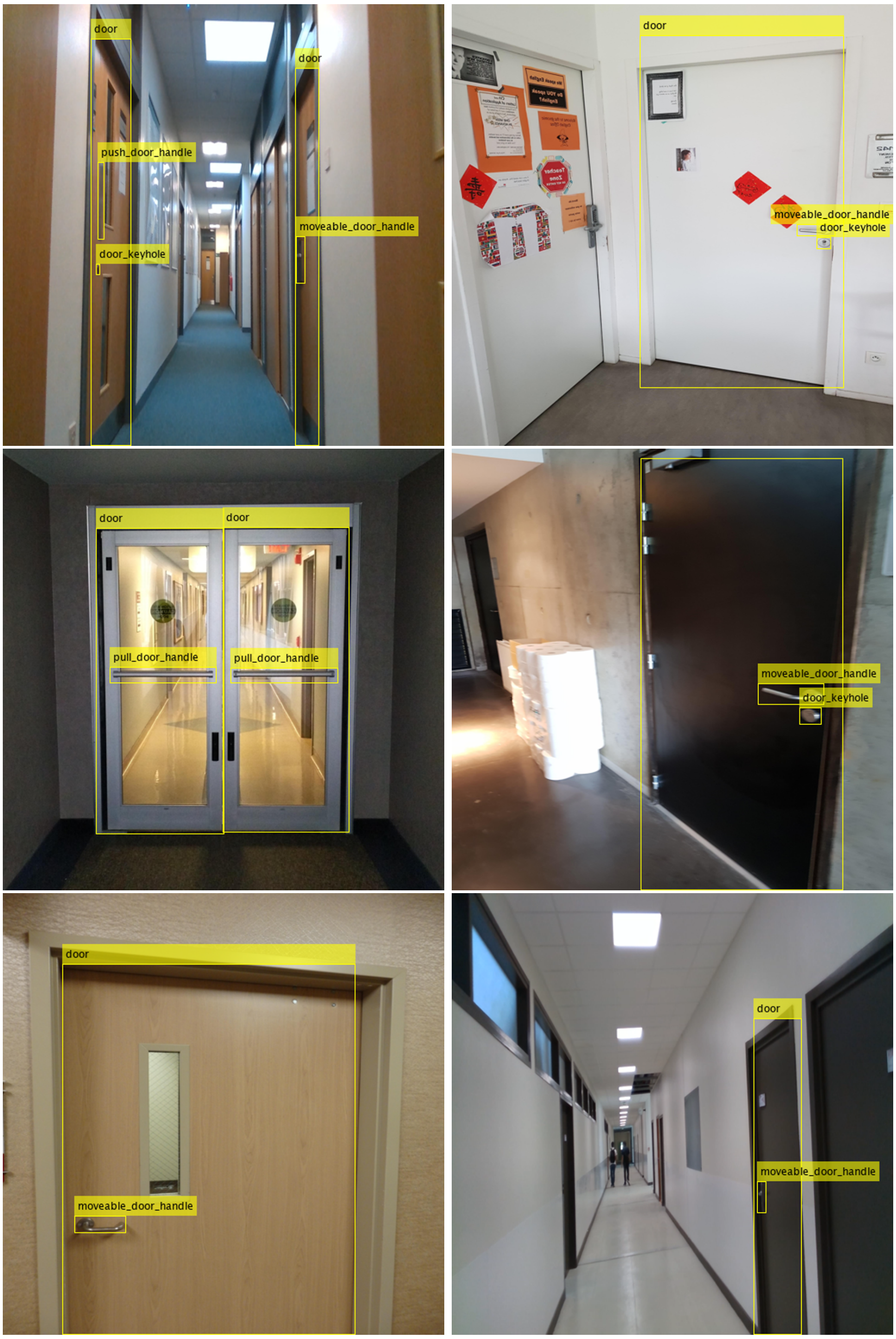


Fig. 3.6 Examples from the collected object detection dataset with the bounding box annotations.

Chapter 4

Aspects of Explainable Artificial Intelligence

4.1 Introduction

The ubiquitous utilisation of Deep Convolutional Neural Networks (DCNN) in many applications because of their accurate performance has put a significant responsibility on the regulators to approve and legalise them [97–99]. Systems based on DCNN can be seen as black boxes where the reason behind a particular decision is ambiguous. The internal operation of DCNN is decentralised as many neurons can contribute to the final output. It is not a straightforward process that can produce a definite output for a predefined input but a possibility that it belongs to a specific class. This decision needs to be justified. In other words, the system needs to explain the reason for its predictions. Consequently, understanding DCNN behaviour is essential, especially for critical industries such as medicine and automation, where the tolerance for errors should be zero.

Clear reasoning and interpretation of DCNN's predictions can accelerate the approval process and boost the trust in black box systems [97–99]. A transparent system that can justify its predictions is the ultimate endeavour. A system that explains its behaviour can help the developers to debug in case of errors, enhance in case of bias, and trust in case of critical predictions.

Activation maps (also called attribution, heat or saliency maps) visualisation is one of the main methods of eXplainable Artificial Intelligence (XAI) to understand CNNs decisions. Heatmaps highlight the main features in an image that stimulate a CNN to classify an image to a particular class. Eliminating these features can result in a significant decrease in the output accuracy.

Many methods and approaches are introduced to generate CNN heatmaps. CNN visualisations techniques can be split into three main categories: visualising what stimulates a specific unit by mathematically synthesise images that maximally activate that unit [133, 134], visualising the filters of hidden layers, and visualising the features that induce a network to assign an image to a specific class (post-hoc activation visualisation). Post-hoc activation visualisation methods produce heatmaps that define the contribution of each input feature to the output prediction. Visualising CNN predictions using post-hoc techniques is the focus of this chapter.

This chapter proposes visualisation methods that aggregate the maps of gradient-based approaches and combine them with the CAM map [116]. The intuition is to obtain both fine-grained details using gradient-based approaches and regional contribution using CAM approach in one comprehensive map. The proposed methods are different from Guided Grad-CAM [117] at which Grad-CAM is combined with Guided backpropagation (GBP) [110]. However, the introduced methods aggregate several gradient heatmaps (Gradients [106], GBP [110], and Integrated Gradients (IG) [113]). These gradient-based heatmaps are chosen because they focus on various activation features due to the differences in their implementations. Consequently, the produced heatmap is comprehensive. Moreover, the resultant heatmap can be combined with the CAM heatmap to add a further localisation dimension.

Each of the combined maps has a weighting parameter that can be adjusted to highlight specific map features. This can help to emerge particular features or details depending on the application, as some applications require high-resolution maps while others require localisation capabilities. The choice of these visualisation methods can be attributed to their high-resolution maps and accurate localisation capabilities. Thus, the resultant heatmap visualise the main contributing features to the CNN decision. Therefore, it can be used to understand the main motive for the model's prediction. Consequently, the model can justify and explain its decision, which is the main objective of the proposed methods. Extensive qualitative and quantitative experiments are conducted to compare the proposed techniques with state-of-the-art ones. Results show the ability of the proposed methods to explain CNN predictions. In addition, the proposed methods can accurately localise the target object.

This chapter is organised as follows: the new visualisation approaches are explained in section 4.2. Section 4.3 shows applications of visualising CNN attribution maps. Experiments and results are discussed in sections 4.4 and 4.5, respectively. Lastly, the chapter is concluded, and the future work is highlighted in section 4.6.

4.2 Methodology

Gradient-based methods such as saliency [106], GBP [110], and IG [113] approaches can produce high-resolution heatmaps. However, the CAM [116] approach is better in creating class discriminative heatmaps. We propose two techniques to attain the benefits of both directions: weighted sum of gradients approach (WS-Grad) and concatenation of gradients approach (Concat-Grad). The produced heatmaps from both techniques are then aggregated with the CAM one.

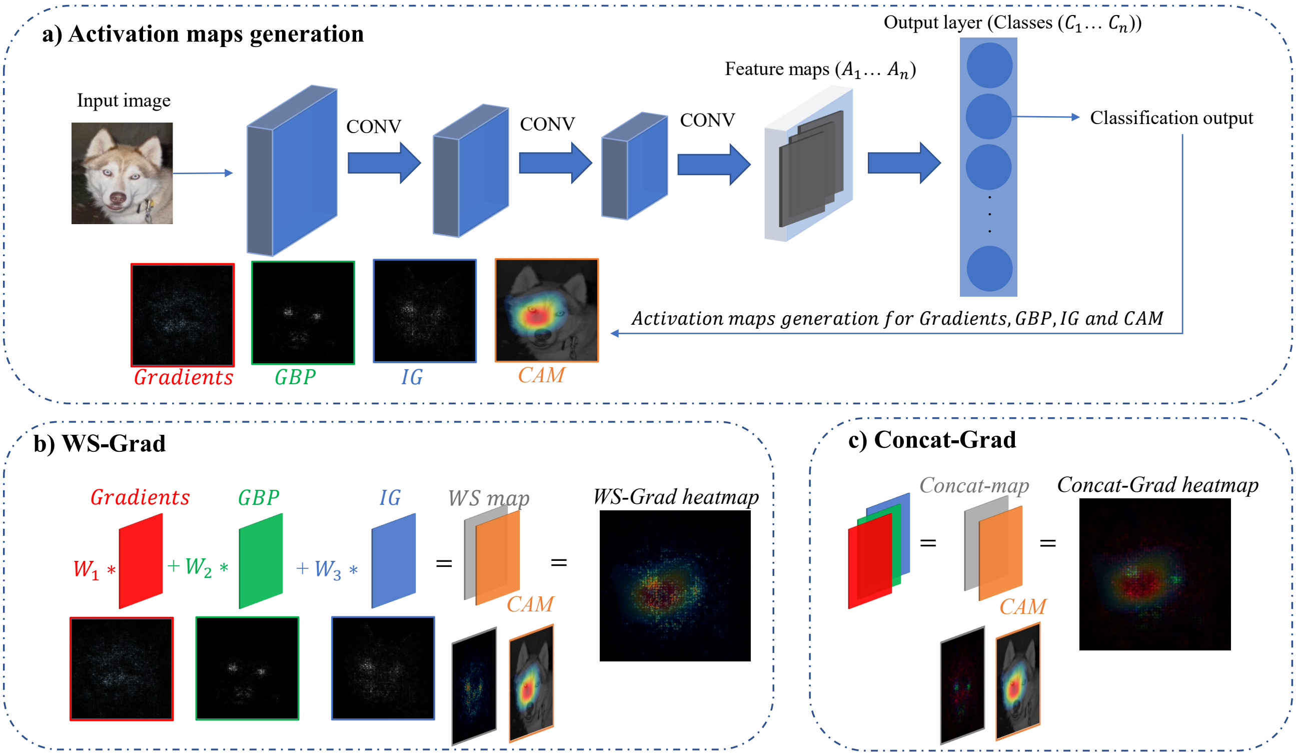


Fig. 4.1 a) activation maps generation, b) weighted sum gradients (WS map) aggregated with CAM to produce WS-Grad heatmap, and c) concatenated gradients (Concat map) aggregated with CAM to produce Concat-Grad heatmap.

4.2.1 Weighted sum of gradients approach (WS-Grad)

Fig. 4.1 (b) shows the weighted sum gradients aggregated with CAM. First, gradients-based heatmaps are generated (Fig. 4.1 (a)). Then each map is multiplied by an integer weight that the user can determine to highlight specific features as different maps can highlight various elements. For example, saliency maps (Gradient approach) highlight all the features that contribute equally to the prediction. However, GBP focuses on the most discriminative features and ignores supplementary ones. IG approach accumulates gradients over scaled-up versions of the input that follow a baseline defined by the user. The flexibility of choosing weights is a powerful tool that can be utilised differently according to the applications. Lastly, the weighted maps are summed and aggregated with the CAM one.

The produced heatmap is more expressive than the individual ones, where the most important features are strongly highlighted with high resolution. Also, CAM approach provides an additional dimension by highlighting the discriminative region, which is very helpful for localisation tasks.

4.2.2 Concatenation of gradients approach (Concat-Grad)

Fig. 4.1 (c) shows the concatenated gradients aggregated with CAM. Similar to WS-Grad, Gradient-based heatmaps are generated. Then, the generated maps are weighted and concatenated as a single image with three channels. Lastly, the concatenated map is aggregated by the CAM one.

The Concat-Grad approach produces high-resolution heatmaps where all the important features using different visualisation approaches can be seen and identified in one image. The generated map has three channels (similar to an RGB image). Consequently, the produced heatmap reflects each map using a different colour. This means Gradient, GBP, and IG features are depicted in red, green, and blue colours, respectively. The novelty in this method can be seen in the ability to distinguish various features of different approaches using distinctive colours in one map, which is very informative as it gives better insights into the important features and their corresponding approach. Additionally, the produced heatmap is easy to interpret and understand. Furthermore, aggregating the produced map with CAM creates a comprehensive heatmap. The attained map can be characterised by a high-resolution multi-channel heatmap that can highlight the most discriminative region.

In section 4.5, we qualitatively and quantitatively compare our approaches with state-of-the-art ones. Our techniques produce high-resolution comprehensive heatmaps that individual ones cannot produce.

4.3 Applications

Applications of explanation techniques are vast and vital. Generally, they ensure the reliability and trustworthiness of black box systems. More specifically, it can be used to debug models, evaluate performance, enhance training, and analyse data. Solely depending on the validation section of a dataset to assess the model's performance is inadequate as the validation dataset can be biased or limited. That is why visual inspection can add another dimension to the validation process.

As a case study, Layer-wise Relevance Propagation (LRP) [135] explanation technique is applied to two models that produced the same test accuracy on a document classification task [136]. It is noticed that the Support Vector Machine (SVM) has based its decision on word count while CNN has assigned more relevance to keywords. Similarly, two models in the image classification domain, a Fisher vector classifier trained on PASCAL images [132] and a CNN trained on ImageNet [137], are visually compared using LRP explanation technique [138]. Both systems have produced the same classification accuracy for the horse category. However, they use different cues to attribute their decision. The Fisher Vector classifier has assigned high relevance to copyrights tags usually present in horse images. However, the CNN uses the horse edges and contours to make predictions. Using explanation approaches can help to mitigate the system's weakness (in this case, the Fisher Vector classifier) by identifying the bias. To overcome this problem, untagged images can be introduced to retrain the system.

LRP explanation technique [135] is used to analyse data of face images to identify which pixels are responsible for age and gender characteristics [139]. It is also used to visualise electroencephalogram (EEG) heatmaps after a CNN is trained to map EEG patterns to a set of movements to understand which part of the brain contribute to a specific decision [140]. Guided Backpropagation [110] is used as a part of a system to highlight features corresponding to shadow pixels in 2D ultrasound images[141]. The method is applied to two 2D ultrasound datasets acquired during foetal screening and can generate shadow-focused confidence maps that can be used for biometric measurements [141].

Haofan et al. [142] used Score-CAM (an enhanced version of CAM) to debug different systems. Score-CAM can achieve adequate object localisation even with poor classification models. However, the noise in the saliency map decreases as the model's performance increases. This can be considered as an indication of a model convergence [142]. Also, Score-CAM can help to identify dataset biases and reasons for wrong predictions.

The aforementioned methods (LRP and Score-CAM) are similar to those presented and used to create the proposed novel methods. They are being used for the same purposes, such as visualising CNN decisions and identifying biases in datasets. However, LRP and Score-CAM use different techniques to produce the heatmaps. LRP [135] uses backpropagation to compute relevance. It is a generalised approach to visualise the contributions of non-linear classifiers by pixel-wise decomposition of the output prediction. Whereas Score-CAM [142] is a gradient-free CAM-based visualisation method. It uses the scores of the forward pass of the element-wise multiplication of the input image with the extracted feature map as channel weights. Channel weights (scores) are then multiplied by the activation maps and combined linearly to generate the heatmap of the target class.

4.4 Experiment setup

A DCNN trained on a simple binary classification task is used to investigate the performance of different visualisation techniques. Kaggle dataset for cat vs dog competition[1] is used to evaluate different visualisation techniques qualitatively and quantitatively. The train-validate dataset contains 25,000 images divided equally into two classes: cat and dog. The train-validate dataset is randomly split into 80% for training and 20% for validation. Another 12,500 images are kept aside for testing. Transfer learning technique is used to train a DCNN model for

[1] https://www.kaggle.com/c/dogs-vs-cats/data

classification with some modifications. SqueezeNet [143] model is chosen as it is one of the smallest DCNN in terms of size and number of trainable parameters. Nevertheless, it is very efficient. Results show that it can achieve AlexNet [36] level of accuracy with 50× fewer parameters. SqueezeNet is pre-trained on ImageNet [137] dataset to classify 1000 different objects. To perform transfer learning, the final convolutional layer with 1000 channels is replaced with a new one that has two channels (equivalent to the number of classes in the dataset). Also, the classification layer is replaced with a new one that reflects the number of classes in the dataset. The weights and biases learning rate factors in the added convolutional layer are set to 10, which means they are 10× the global learning rate (the learning in the new layer is ten times faster than the transferred layers).

To obtain the best model, Bayesian optimisation algorithm [144] is used. Bayesian optimisation algorithm can be used to optimise the hyperparameters of classification models. The algorithm maintains a Gaussian process model of the objective function internally. The objective function evaluations are used to train the Gaussian process model. In our case, the objective function is used to train a DCNN. The function returns the classification error on the validation dataset. Then, Bayesian optimisation is used to minimise the classification error on the validation dataset. The objective function has two variables to manipulate: mini-batch size and section depth. Mini-batch size can take any integer value between 1 (online training) and 128. Whereas section depth is the number of additional convolutional blocks that can be added to the end of the SqueezeNet and before the new convolutional layer. The section depth can take one of three values 0, 1, or 2. Each block consists of a convolutional layer (with a 3×3 filter size and 11 channels) and an average pooling layer (with a 2×2 filter size). The best-achieved model is then used to visualise the network's layers.

Each iteration of the Bayesian optimisation trains the proposed network using the following parameters: Stochastic Gradient Descent (SGD) with 0.9 Momentum is used as the optimisation algorithm. A starting learning rate of 0.0001 is used. Then, the learning rate is dropped by a

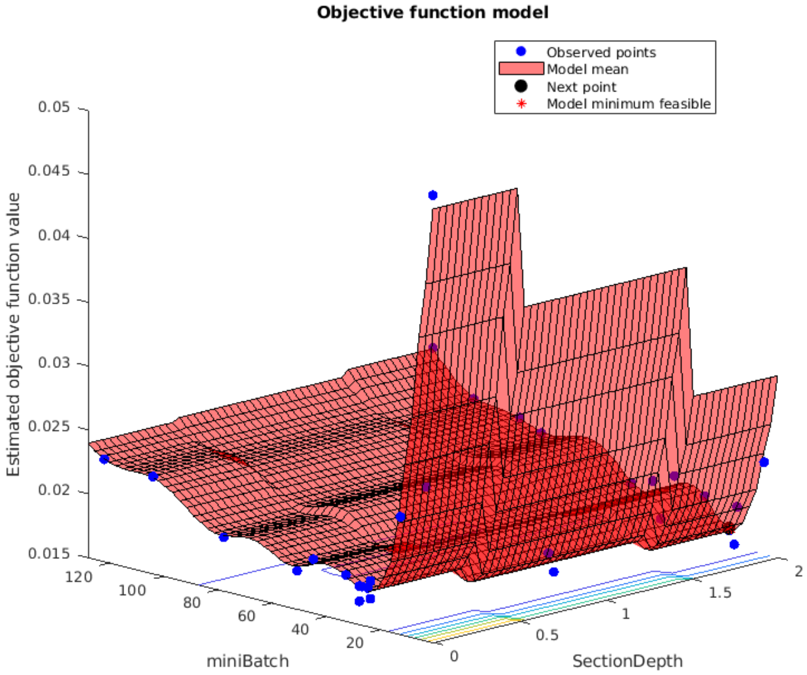


Fig. 4.2 **Objective function model.** The model minimum point is at 0.0152.

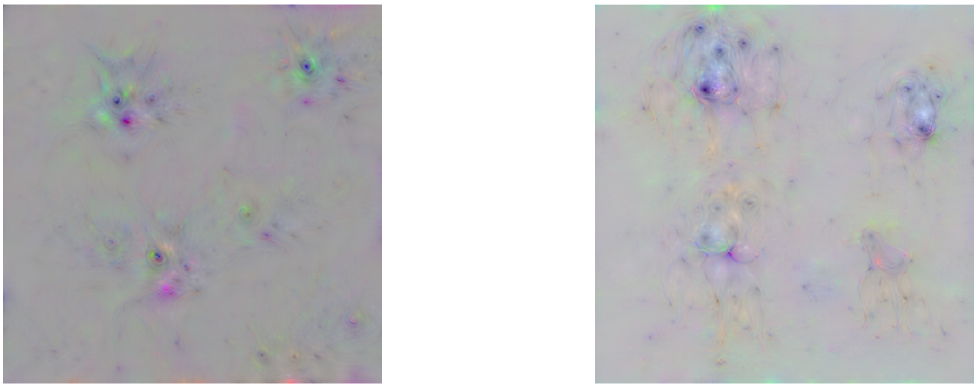
factor of 0.1 every six epochs. The maximum number of epochs is 20. Training examples are shuffled every epoch. The setup with the aforementioned parameters is chosen experimentally to achieve the best performance.

The objective function is evaluated 30 times to best utilise the power of Bayesian optimisation. Fig. 4.2 shows that the best DCNN achieved by the objective function has two more convolutional blocks than SqueezeNet. The mini-batch size used to achieve a 0.0152 validation error (98.48% accuracy) is 17. The lowest error is achieved at the 23rd iteration of evaluating the objective function.The obtained system (a modified version of SqueezeNet) is used to investigate different visualisation techniques. The system's accuracy on the validation dataset for the cat and dog classes are 98.8% and 98. 2% respectively.

4.5 Results and discussion

4.5.1 Qualitative results

Fig. 4.3 shows the synthesised images that maximally activate class-specific neurons using Deep Dream [145]. We visualise the neurons of the last layer before the softmax layer of the trained network. It can be noticed that the images are unclear and unnatural. They consist of scattered parts of faces, ears, and legs. However, it is a useful diagnostic tool to understand the network behaviour and the learned representations.



(a) cat.

(b) dog.

Fig. 4.3 Synthetically generated images that maximally activate cat (a) and dog (b) neurons.

Fig. 4.4 shows examples of the learned features by the hidden layers of the model. For example, filters (channels) of the middle layers can learn to detect specific colours (the filter in Fig. 4.4b detects red colours), patterns (the filter in Fig. 4.4c detect cat strips), and eyes (the filter in Fig. 4.4d detect cat's eyes) where white pixels represent the strongest features. Filters of Fig. 4.4 are extracted from 'fire6-squeeze 1×1' convolutional layer, which has 48 channels. Every channel is responsible for a specific colour or pattern. Hidden layers give insights into information learned during the training process. Each channel learns to capture a specific feature. Earlier layers learn simple features. At the same time, deeper ones learn more complex and composite features. By investigating networks' channels, network performance can be enhanced by strengthening the low represented features.

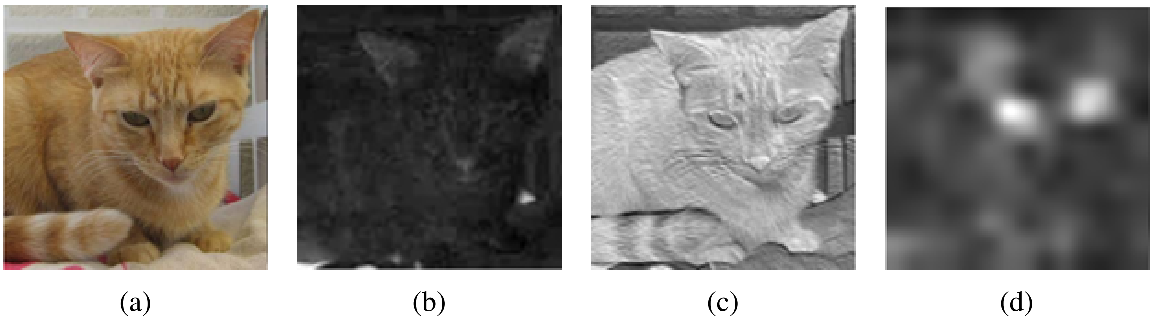
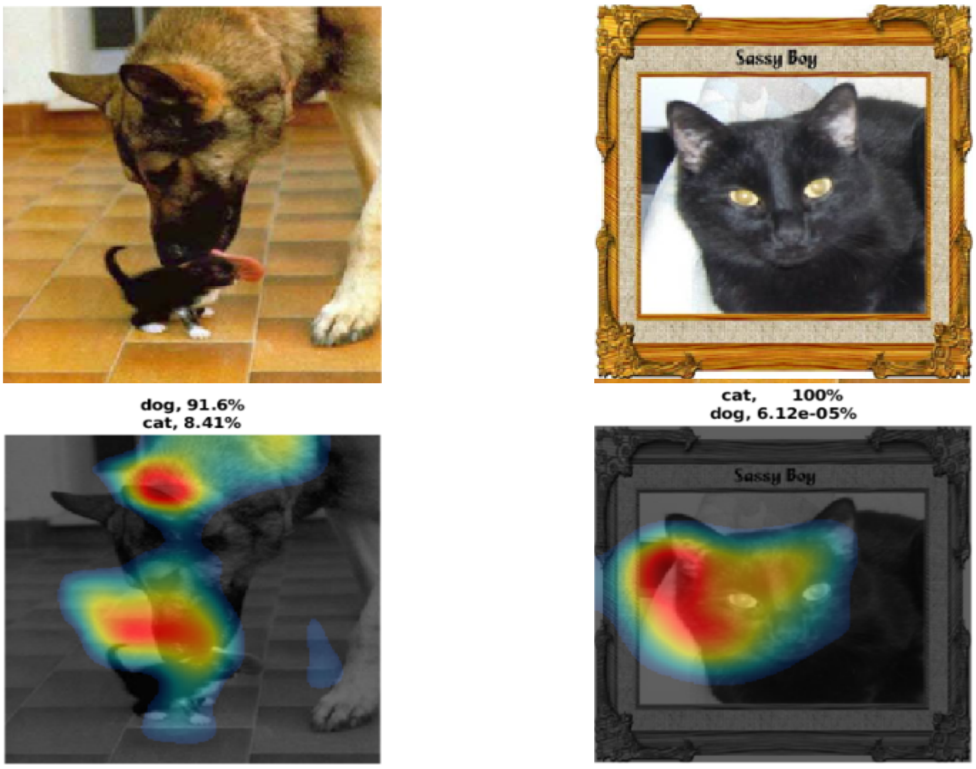

(a) (b) (c) (d)

Fig. 4.4 Examples of hidden layer filters visualisation: the channel in (b) captures red colours, the channel in (c) captures strips, and the channel in (d) captures eyes.

Different visualisation techniques have been used to understand the network's behaviour. The ReLU layer that follows the last convolutional layer in the trained network is used to compute the activations for CAM [116]. CAM technique can be visualised in Fig. 4.5, which shows the main regions that positively contribute to the network's predictions. Fig. 4.5a shows that the model is mostly focused on the dog's ear, nose, and mouth (red regions). Although there is a cat in the image, it does not grab the network's attention. This clearly explains the reason behind the classification of that image as a dog. Similarly, the network focused on the cat's face in Fig. 4.5b, which emphasises the intuition behind the correct prediction.

Gradient method (shown in Fig. 4.6, second column) computes the gradients of the output class w.r.t the input image. The generated heatmaps using sensitivity analysis methods are sharper than that of CAM [117]. Nevertheless, they tend to be noisy. Gradients are calculated for each channel of the RGB image. However, direct plotting of the RGB attribution map results in messy visualisations. Consequently, each pixel's absolute value across the RGB channels is summed and scaled in the range of zero to one. Still, the attribution maps are not clear enough, but the main contributing object to the network prediction can be easily determined. Guided Backpropagation [110] (Fig. 4.6, third column) provides sharper maps by zeroing both elements of the gradients that are less than zero (during the backward pass) and elements of the input that are less than zero (during the forward pass) using ReLU units. A black image is chosen as the baseline image for the IG method (Fig. 4.6, last column). Still, the



(a) (b)

Fig. 4.5 CAM visualisation of randomly selected images from the validation dataset.

reliability of GBP and IG is questionable [115]. However, the Gradient method [106] is reliable for explaining network decisions [115], even though the produced heatmaps are unclear.

4.5.2 Proposed approaches qualitative analysis

Fig. 4.7 and Fig. 4.8 compare the proposed approaches with the individual gradient-based methods and CAM technique. Gradient-based methods produce high-resolution heatmaps. Unlike CAM technique which produces regional features heatmaps. On the other hand, the proposed approaches can produce coherent fine-grained and class discriminative heatmaps.

Qualitatively, the proposed approaches produce more interpretable and understandable maps. Unlike individual approaches, weighted sum gradient-based (WS-Grad) maps highlight all the important features with more focus on the most discriminative ones. Besides, the generated maps are more dynamic and lively where the object of interest emerges. This clearly

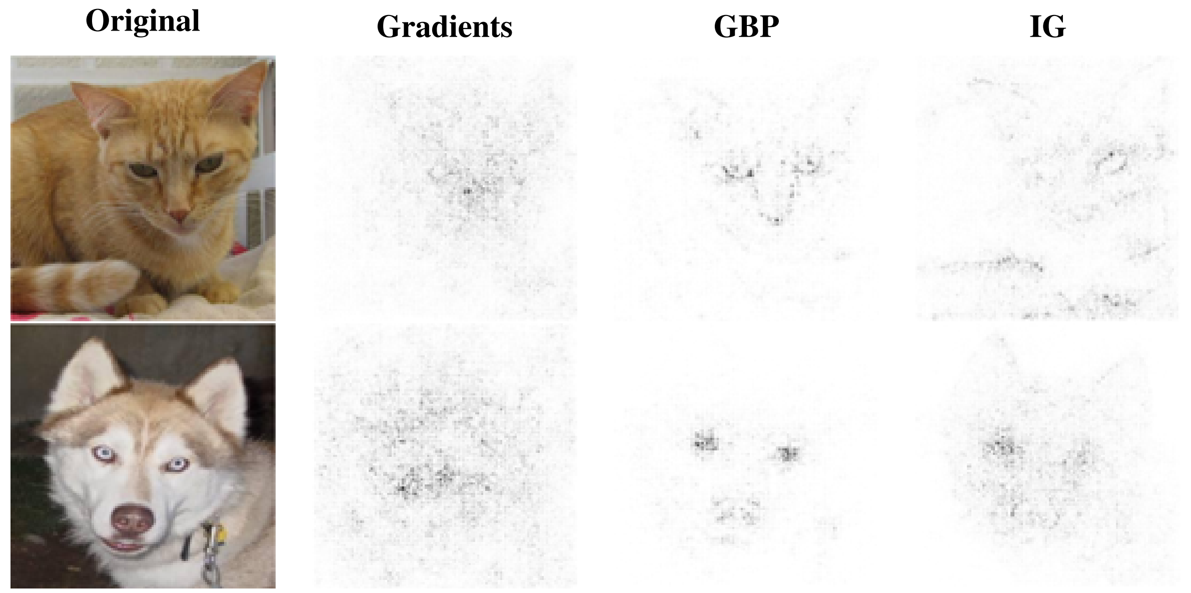


Fig. 4.6 Gradients methods produces unclear attributions maps while DeconvNet, Guided Backpropagation, and Integrated Gradients generate better quality and more descriptive maps.

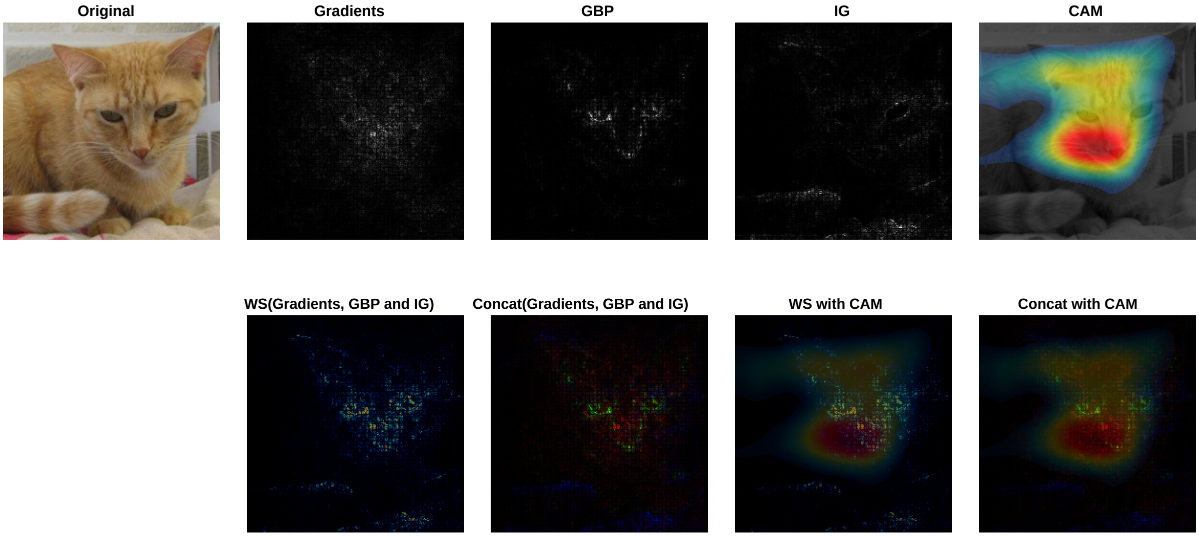


Fig. 4.7 Weighted sum gradient-based proposed approach solely and with CAM technique. Also, Concatenation gradient-based proposed approach solely and with CAM technique for an example image of a cat.

indicates how the CNN is motivated to classify an object to a specific category. An important step that can be used to achieve the transparency condition of XAI models. Consequently, trust can be put in such systems.

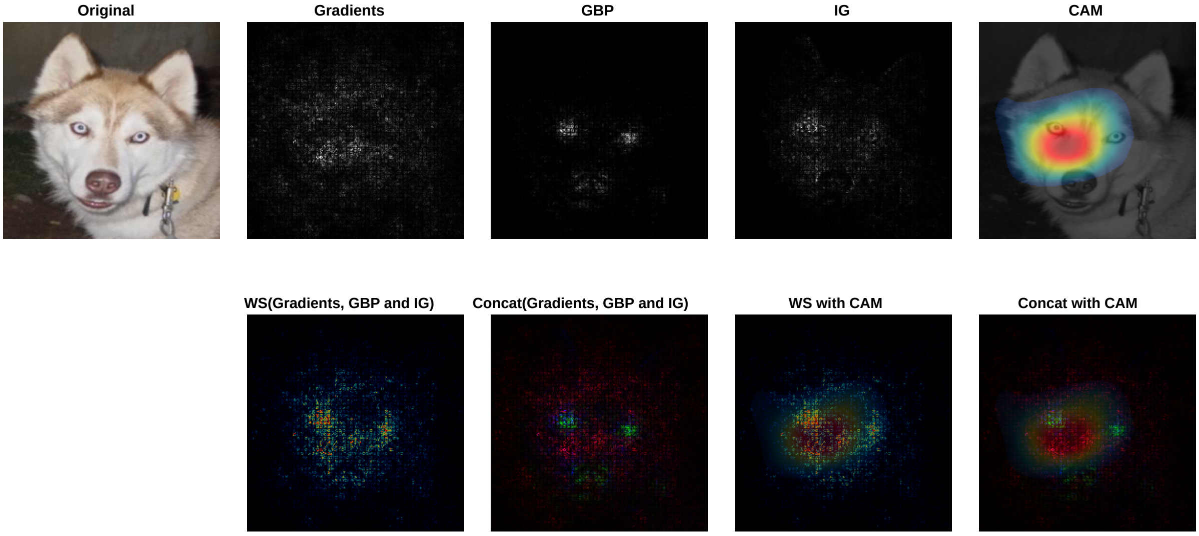


Fig. 4.8 Weighted sum gradient-based proposed approach solely and with CAM technique. Also, Concatenation gradient-based proposed approach solely and with CAM technique for an example image of a dog.

Similarly, concatenation gradient-based (Concat-Grad) maps combine the most important features and present them in a colour interpretable way that facilitates the reading and understanding of such heatmaps. For example, the red colour in the Concat-Grad shown in both Fig. 4.7 and Fig. 4.8 represents the Gradients map important features, the green colour represents the GBP map important features, and the blue colour represents the IG map important features. The three maps capture different features. However, they all contribute to the network decision as depicted in the two figures (Fig. 4.7 and Fig. 4.8). The Gradients method captures the face and skin, GPB focuses more on the eyes and nose, and IG highlights the contours. The powerful novel technique can present all this information for investigation and analysis in one comprehensive heatmap. Combining both of the proposed techniques with CAM maps adds an extra dimension of information (discriminative regions).

4.5.3 Quantitative results

To quantitatively assess the proposed techniques' localisation abilities, we set up an intuitive experiment to measure the IoU between the ground truth (gTruth) Bounding Boxes (BBs) and the BBs that encompass all of the detected features of different visualisation techniques. IoU reflects the ability of the visualisation method to localise the target object. Consequently, it can achieve better results on unsupervised and semi-supervised learning tasks.

The BBs for the gTruth data are annotated manually to encompass the whole object. The BBs for the visualisation techniques are generated using the minimum eigenvalue algorithm [146] which finds the strongest 100 feature points. Then, the smallest BB that can comprise all of the generated points is constructed. It worth mentioning that the proposed model, which is used to generate the heatmaps for all methods, has not been trained on any BB annotated data. Detecting an object with a model that has not been trained on object detection tasks is very useful in many applications because BBs are not available for many datasets.

Two images (a cat and a dog) are randomly selected from the test set for the evaluation process. The proposed WS-Grad and Concat-Grad approaches achieved better localisation with 0.660 and 0.609 IoU, respectively, compared to vanilla Gradients and GBP, which achieved 0.540 and 0.320, respectively. Whereas IG achieved the highest IoU of 0.694 (Fig. 4.9).

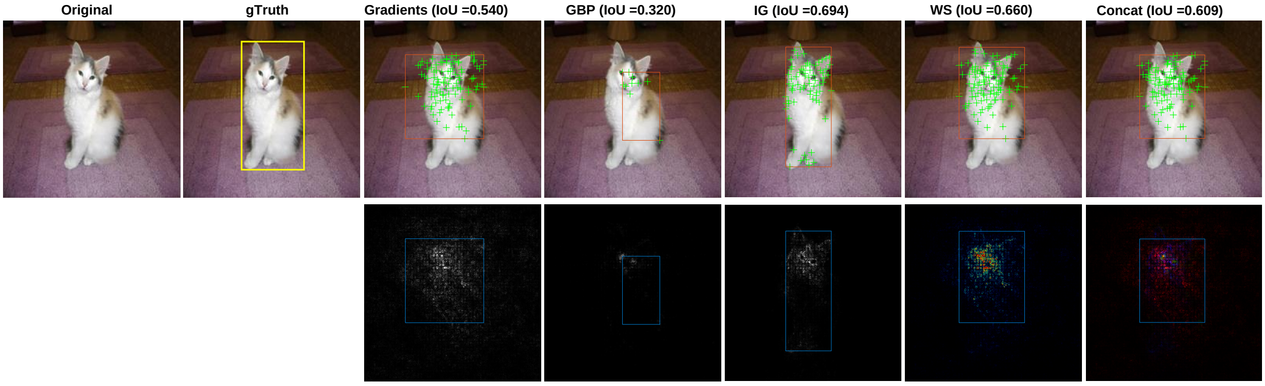


Fig. 4.9 Localisation capabilities for Gradients, GPB, IG, WS-Grad, and Concat-Grad approaches (cat).

Object localisation capabilities of the proposed approaches are on the same level of accuracy as IG. However, WS-Grad and Concat-Grad can highlight specific heatmaps that might contain important information using the weighting parameters. Fig. 4.10 shows that if we emphasise the IG heatmap, as it has the highest IoU, by multiplying it by a factor of 3, then the produced aggregated heatmaps can achieve the highest IoU with 0.887 and 0.810 in the cases of using the WS-Grad and Concat-Grad approaches, respectively.

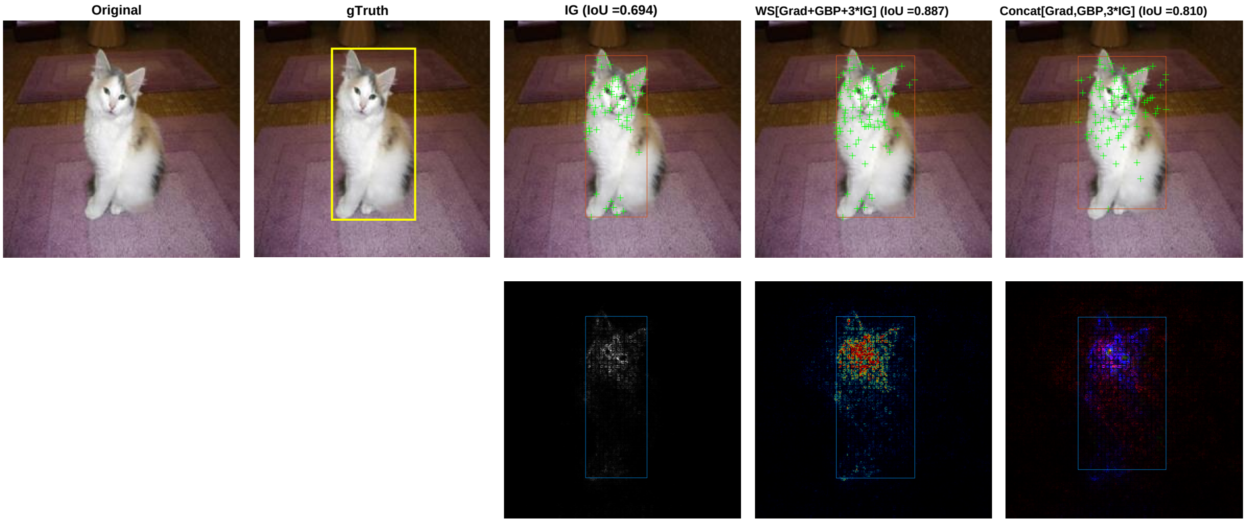

Original
gTruth
IG (IoU =0.694)
WS[Grad+GBP+3*IG] (IoU =0.887)
Concat[Grad,GBP,3*IG] (IoU =0.810)

Fig. 4.10 Using the weighting parameters of the WS-Grad and Concat-Grad approaches to boost the important features of IG heatmap results in the highest IoU (cat).

The weighting parameters (multiplying factor) are selected by experimentally trying different integer values in the range of 1 to 10. By increasing the multiplying factor, the WS-Grad approach could produce better IoU (0.887) until it reaches the factor of 3. After that, the IoU keeps decreasing until it reaches the multiplying factor of 5, where it plateaus at 0.697 IoU.

On the other hand, the IoU of the Concat-Grad approach keeps improving until it reaches the multiplying factor of 6, where it achieves 0.893 IoU. IoU starts to slightly decrease when multiplying by a factor of 7. The multiplying factor of 3 is chosen because it achieves the highest IoU in both cases of WS-Grad and Concat-Grad. Optimising the weighting parameters is a matter of trial and error. However, increasing the weightings arbitrary may result in noisy heatmaps where the important features could be lost.

Similar observations can be extracted from Fig. 4.11 where our approaches achieved 0.662 and 0.694 for WS-Grad and Concat-Grad, respectively. The achieved IoUs beat the localisation capabilities of vanilla gradients and IG, which achieved 0.642 and 0.609, respectively. At the same time, GBP achieved the highest IoU of 0.702.

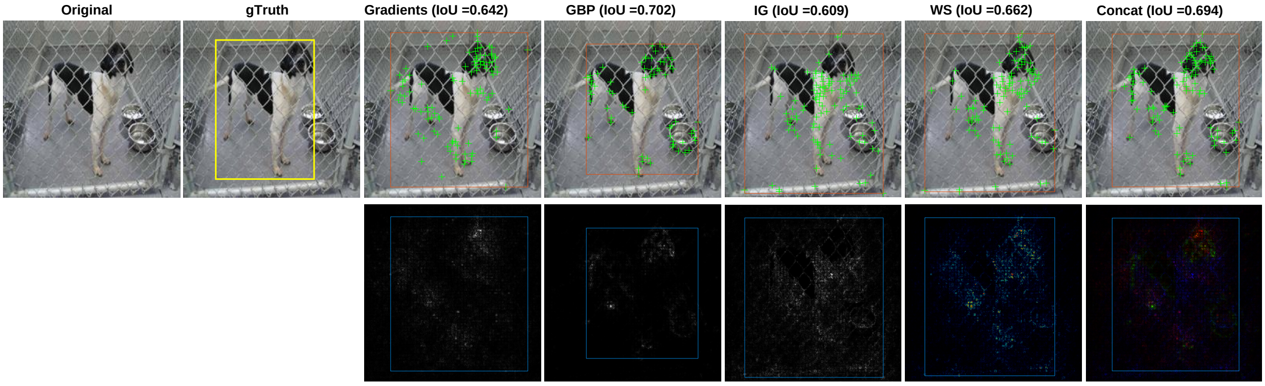


Fig. 4.11 Localisation capabilities for Gradients, GPB, IG, WS-Grad, and Concat-Grad approaches (dog).

Amplifying the detected features of the GBP approach using the weighting parameters of the proposed approaches results in the highest IoU of 0.728 in the case of WS-Grad (Fig. 4.12). Different multiplying factors (weighting parameters) are tried starting from 1 to 10. The IoU of WS-Grad and Concat-Grad keeps improving until the multiplying factor of 3, at which they achieve 0.728 and 0.700 IoU, respectively. After that, the IoU of the WS-Grad approach starts to slightly decrease, reaching 0.709 when multiplying by a factor of 6. At the same time, the Concat-Grad approach plateaus at IoU of 0.700 until the multiplication factor of 6. This concludes the flexibility of the proposed methods and their high performance in object localisation tasks. Manipulating the weighting parameters is a trial-and-error process. However, it is a great tool to generate heatmaps that suppress unwanted features and boost important ones.

4.5.4 ImageNet qualitative and quantitative results

For further investigation, the proposed visualisation techniques are tested on a multi-class object classification task using off-the-shelf AlexNet [36] which has been trained on ImageNet dataset [137]. Test images are randomly selected from the dataset with the corresponding BB annotations.

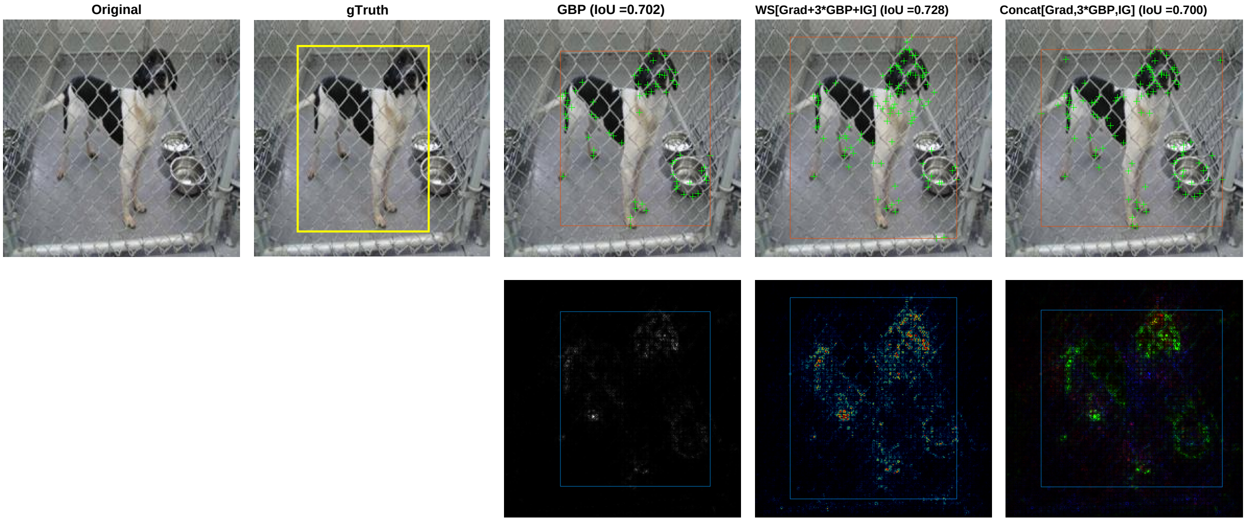

Original
gTruth
GBP (IoU =0.702)
WS[Grad+3*GBP+IG] (IoU =0.728)
Concat[Grad,3*GBP,IG] (IoU =0.700)

Fig. 4.12 Using the weighting parameters of the WS-Grad and Concat-Grad approaches to boost the important features of GBP heatmap results in the highest IoU (dog).

Table 4.1 shows the ability of the proposed methods to localise the target objects. The produced BB of the introduced methods demonstrated competitive abilities to detect the whole objects, knowing that the model was not trained on the BB level.

Table 4.1 **IoU of the proposed methods on randomly selected images from ImageNet dataset.**

Method / **Image**	**Gradients**	**GBP**	**IG**	**WS-Grad**	**Concat-Grad**
Spider	0.908	**0.909**	0.783	0.889	0.905
Robin	0.521	0.443	0.578	0.543	**0.645**
Bolete	0.436	0.503	0.553	**0.656**	0.523
$Pay-phone$	0.723	0.775	0.770	0.763	**0.798**
Pomegranate	0.420	0.633	0.588	0.671	**0.673**

Fig. 4.13 shows that the proposed methods generate better heatmaps than Gradients, GBP, and IG methods. For example, the first two rows in Fig 4.13 depicted the features that contribute to the CNN's decision for the 'Spider' class using different explanation methods. Gradients map highlights all the features that contribute to the output prediction equally, which makes the heatmap noisy. GBP map focuses on the spider body and legs. IG map shows that the spider head is the main motive for the CNN prediction. The features of different explanation

methods can be seen clearly in one heatmap using the WS-Grad and the Concat-Grad proposed approaches, which can justify the CNN decision.

The collective features that activate the CNN are shown by the WS-Grad proposed approach in one image. Whereas the Concat-Grad approach depicted each of the individual maps in different colours as follows: the extracted features of the spider and its web using the Gradients method are shown using the red colour. The extracted features of the spider's body and legs using the GBP method are shown using the green colour. The extracted features of the spider's head from the IG map is shown using the blue colour. The produced heatmap using the Concat-Grad is not only visually appealing but also comprehensive and inherently interpretable.

Quantitatively, the proposed methods achieve better IoU in most cases compared to the state-of-the-art methods and without using the weighting parameters (Fig 4.13). Even in the worst cases, the proposed methods are competitive with the state-of-the-art ones. It is believed, as shown in the binary classification quantitative analysis, that using the weighting parameters can further enhance the IoU results of the WS-Grad and the Concat-Grad approaches.

4.6 Conclusion

In this chapter, two novel techniques for visualising CNN predictions are proposed. The proposed techniques incorporate gradient-based methods and aggregate them with the CAM method. Also, the introduced weighting parameter, which can be manipulated to highlight or suppress specific features, is a powerful tool that can be utilised to achieve better localisation and visualisation results. Qualitatively and quantitatively, the proposed methods outperform individual gradient-based ones. Moreover, the obtained information using the proposed approaches cannot be obtained using a single method. Furthermore, the generated heatmaps are informative and human interpretable.

The highlighted features using the proposed methods indicate the motive for the CNN decision. Consequently, they can be used for XAI applications, where detecting and classifying

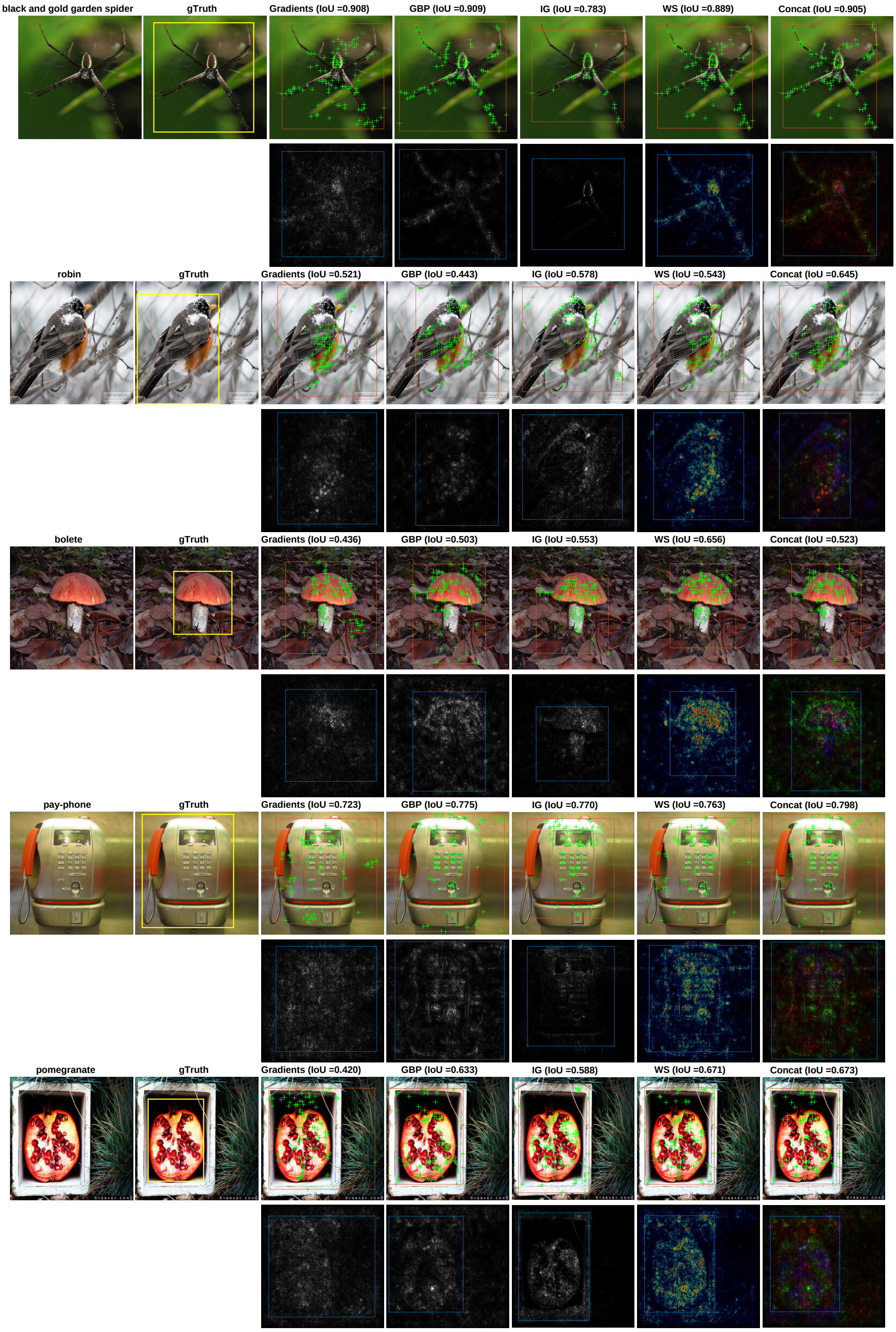


Fig. 4.13 Examples from the ImageNet dataset where our proposed approaches perform at least as good as state-of-the-art methods. In many cases, they surpass them in terms of their ability to localise objects and visually explain what motivates the CNN to produce a specific decision.

the object is not enough, but extra information regarding the main reason for this prediction should be illustrated. Understanding the strengths and weaknesses of different visualisation approaches can give indications on how and when these methods can be utilised to justify the predictions of CNNs. Moreover, it assists the regulators to understand the behaviour of the black-box system. Consequently, it facilitates the deployment of such systems in real-life applications.

The proposed methods overcome the challenging task of producing a single heatmap that contains both fine-grained details and regional discriminative information. We understand that visually appealing methods might not be reliable. That is why we performed quantitative analyses that showed the high abilities of the proposed methods to localise objects. The next step will concentrate on assessing the robustness of the proposed techniques (sanity checks) and applying the techniques to different tasks such as object detection.

Chapter 5

Investigation of Object Detection Methods

5.1 Introduction

State-of-the-art object detector systems based on deep learning and convolutional neural networks have shown significant performance on standard datasets in terms of accuracy and processing speed. Standard datasets mainly contain large objects that occupy a large area of an image. This helps the detector to exploit more pixels during the training and inference stages. Consequently, more information can be used in the training and prediction steps. The case becomes more challenging for small size objects as the information utilised by the system for training or inference is limited. Furthermore, small size objects may appear in groups which complicate their detection. When it comes to small size objects, the performance of state-of-the-art systems has not been systematically investigated. Redesigning the model's architecture and anchor boxes may help to tackle this problem.

Small anchor boxes are efficient with small objects but fail to capture large ones and vice versa. This limits the capabilities of state-of-the-art object detectors to detect small and large size objects simultaneously. Also, increasing the number of anchor boxes to fit both object sizes can negatively impact the detector's speed and accuracy. Speed-accuracy trade-off is another major challenge for object detectors. Therefore, object detectors are considered application-oriented.

This chapter investigates the impact of different object detection architectures on the detector performance. State-of-the-art object detectors are tested on an application-specific dataset. The proposed dataset is challenging as its major components are small size objects. Nevertheless, it contains large and medium size objects, which introduce more challenges. Questions such as the minimum number of anchor boxes that can accommodate the proposed datasets, which base network to use, which feature extraction layer to use, the impact of different detector architectures on accuracy are investigated in this chapter. Besides, the training strategies and data augmentation implications are sought to be discussed.

In the light of the previously mentioned definitions of small size objects in Chapter 3, the thesis follows a new combined definition of small size objects. An object is categorised as 'small' if its size equals or less than 42×42 pixels. This definition is adopted because the size of the images of the proposed object detection dataset is 512×512 pixels.

This chapter is organised as follows: the methodology is introduced in section 5.2, where the system architecture, the training parameters, and the evaluation process are discussed. In section 5.3, results are presented and discussed. Lastly, the chapter is concluded in section 5.4.

5.2 Methodology

5.2.1 Challenges

The proposed object detection dataset mainly contains small size objects. Objects with sizes less than or equal to 42×42 pixels are categorised as small size objects. Objects bigger than 42×42 pixels and less than 96×96 pixels are categorised as medium size objects. Objects greater than 96×96 pixels are categorised as large size objects. The definition of object sizes follows that of the Traffic [62] and COCO [65] datasets except for small size objects because the proposed images are larger than that of the traffic dataset. Consequently, the definition of small size objects is 42×42 pixels instead of 32×32 pixels.

Figure 5.1 shows the sizes and the aspect ratios of the Bounding Boxes (BB) of the object detection dataset. For the ease of understanding and to better distinguish between objects, a set of randomly selected 100 objects from each category is displayed with a different colour. Plotting all the dataset objects can produce a chaotic figure that is difficult to understand. The majority of the object sizes can be categorised as small and medium size objects. Consequently, the performance of state-of-the-art detection systems trained on large size objects can differ due to the different nature of the proposed dataset. State-of-the-art detectors are designed with anchor boxes to accommodate general dataset objects (mainly large objects). However, task-specific datasets need different designs for the anchor boxes and different techniques to capture small size objects along with medium and large size objects.

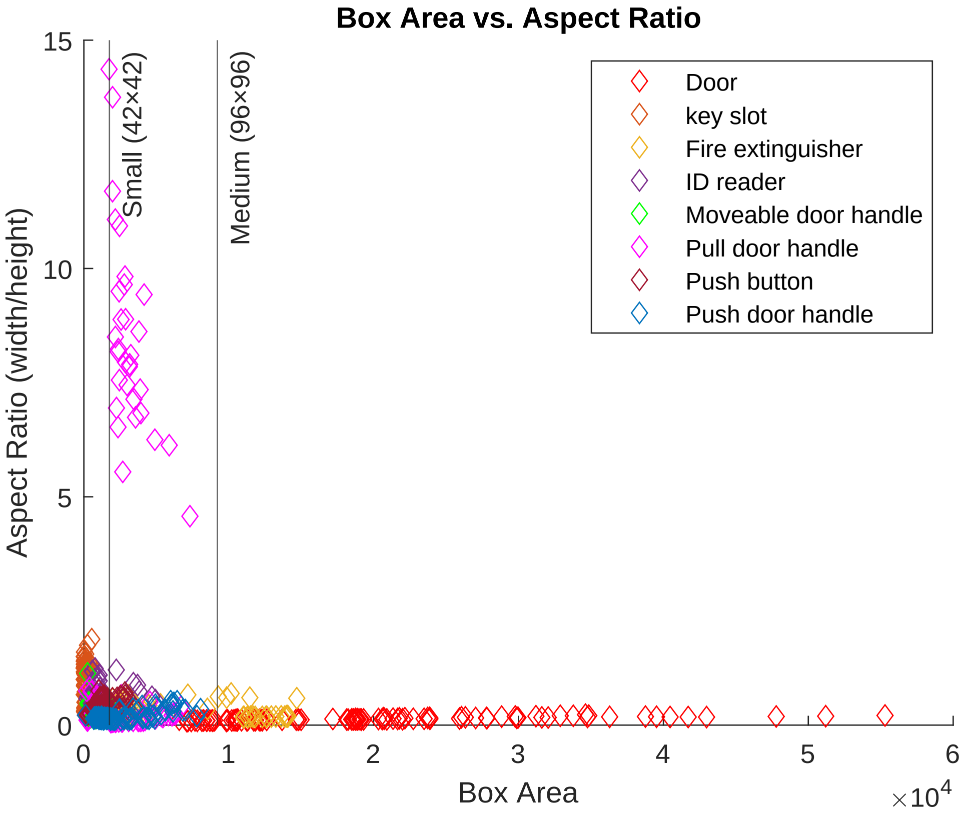


Fig. 5.1 The sizes and aspect ratios of the BBs of the object detection dataset.

5.2.2 System architecture

Two of the most widely adopted object detection systems are used for performance investigation and comparison. The first detector is the one stage YOLO V3, while the second is the two-stage Faster R-CNN. Both detectors use ResNet-18 as the base network.

Feature extraction network

The pipeline of the Faster R-CNN network consists of a feature extraction network, a Region Proposal Network (RPN), and two sub-networks for class prediction and bounding box regression. The feature extraction network is a pretrained network that extracts the features from the input image. The RPN is trained to extract region proposals from the feature maps produced by the feature extraction network. Lastly, the classification and regression networks predict the class and the bounding box of each region proposal.

The detection of small size objects is a delicate task. The spatial information and features of small size objects are limited. Consequently, these objects can get lost as the feature maps are down-sampled through the network layers. The choice of the feature extraction network is based on the application requirements. A deep network results in high accuracy but low processing speed and vice versa. Thus, the choice of the base network is a trade-off between accuracy and speed.

ResNet-18 [51] has been used as the feature extraction network in our experiments. ResNet-18 is the smallest version of the ResNet family. Nevertheless, it is a powerful network that uses residual blocks. It can achieve adequate processing speed with high accuracy. Residual blocks help to overcome deep network problems of vanishing and exploding gradients [147, 148]. Residual blocks reuse the activations from previous layers until the adjacent layer learns its weights [51].

The choice of the feature extraction layer that feeds into the RPN is also a trade-off between the strength of the extracted features and the spatial resolution. High feature extraction layers (deep layers) down the network result in strong encoded features, but the object's spatial information is lost. However, feature extraction layers up the network (early layers) have a better spatial resolution but weak encoded features. Empirical analysis can identify the optimal feature extraction layer for a specific application.

Four different feature extraction layers are used in our experiments to investigate the trade-off between spatial resolution and discriminative features: 'res4a_relu', 'res4b_relu', 'res5a_relu', and 'res5b_relu'. These are the ReLU layers after the last four residual blocks of the ResNet-18 network.

Anchor boxes

Anchors are a set of predefined boxes with different sizes and aspect ratios that represent the objects of the dataset. They are estimated from the training data and used as initial priors to enhance the predicted bounding boxes.

Anchor boxes are used to eliminate the need to scan the whole image using different sizes and aspect ratios sliding windows to make a prediction at each potential position. Consequently, the whole image can be processed in a single propagation through the network, which enhances the overall prediction speed. Different sizes of anchor boxes enable the detection of multi-scale objects. The model predicts the offsets of the anchor boxes to refine boxes' locations and sizes.

The final detector output is produced by removing the anchor boxes that belong to the background. Also, other anchor boxes with confidence scores below a specific threshold are ignored. Lastly, the multiple detections of the same object are refined using the Non-Maximum Suppression (NMS) technique. Anchor boxes enable the prediction of multiple objects with different sizes and scales, besides overlapping objects.

Manually selecting the anchor boxes for the dataset is challenging as objects groups are scattered with varying sizes and aspect ratios (Figure 5.1). A clustering algorithm, such as *k*-means [149], can group boxes of similar aspect ratios and sizes based on a specific metric.

The Intersection over Union (IoU) distance metric is used to estimate the anchor boxes that better represent the dataset objects. IoU distance metric based clustering algorithm can produce anchor boxes that fit the dataset objects efficiently as it is invariant to the boxes' sizes [47]. Whereas other metrics such as Euclidean distance can lead to large errors as the boxes' sizes increase [47].

The number of anchor boxes is a hyper-parameter that can be selected empirically. However, the mean IoU (mIoU) between the training data boxes and the estimated anchor boxes can be used to assess the number and validity of the estimated boxes. Figure 5.2 shows the estimated number of anchor boxes w.r.t the training data bounding boxes and the corresponding mIoU. The maximum number of anchor boxes is set to 30 as the mIoU plateaus or degrades after this point. Arbitrary increasing the number of anchor boxes can negatively affect the detector performance. Many anchor boxes can result in training data overfitting problem. Moreover, the computation cost is directly proportional to the number of anchor boxes. Consequently, it is a trade-off process where the lowest number of anchor boxes that can achieve the highest mIoU is the objective.

A large number of anchor boxes results in slow detectors. Thus, a mIoU greater than 0.5 indicates adequate overlap between the training boxes and the estimated anchor boxes. Usually, marginal improvement can be achieved with many anchor boxes (mIoU start to oscillate between 0.6 and 0.75 after 15 anchor boxes).

Three data points are selected to understand the impact of the anchor boxes on the detector performance (Fig.5.2). First, the point at which the mIoU is more than or equal to 0.5 with the lowest number of anchor boxes (number of anchor boxes = 3, mIoU = 0.518). Second, the point at which the highest mIoU can be achieved (number of anchor boxes = 23, mIoU = 0.757). Third, the point with the highest number of anchor boxes (number of anchor boxes = 30, mIoU = 0.756).

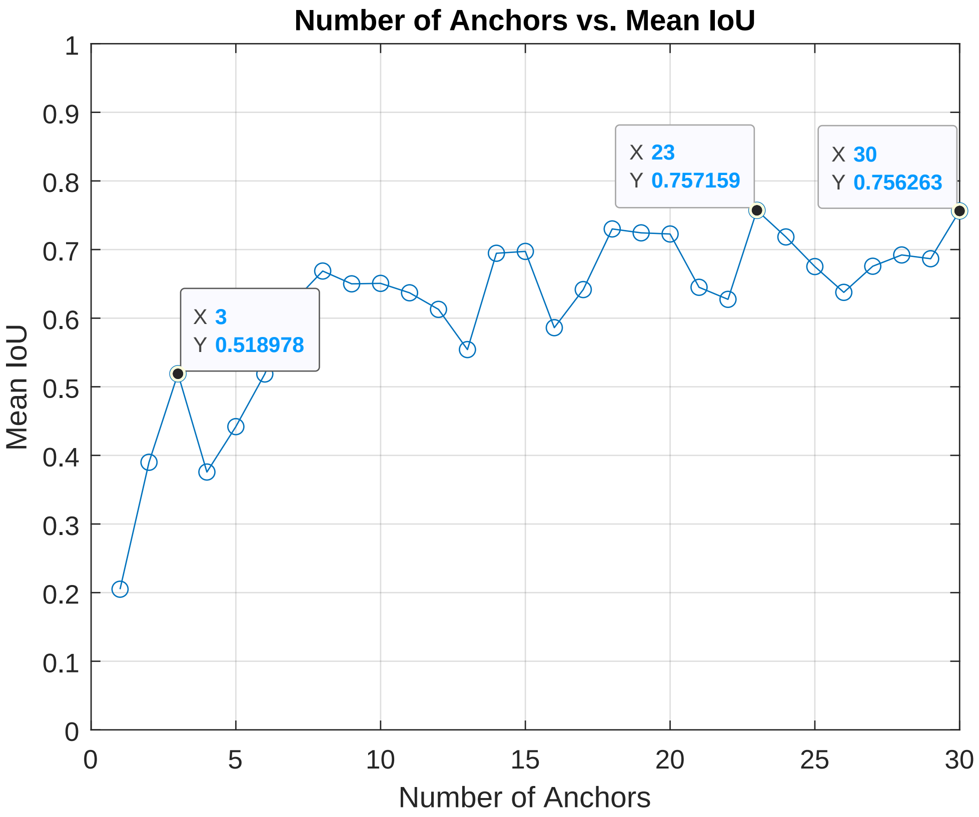


Fig. 5.2 The estimated number of the anchor boxes with the corresponding mIoU.

The adequate number of anchor boxes to achieve high accuracy, fast processing speed, or a trade-off between both metrics can be attained by analysing the dataset objects. Nevertheless, the application requirements are the main motive for choosing the number of anchor boxes.

5.2.3 Training

Training parameters

Several training parameters are tried to find the optimal ones that can achieve the highest performance. The training parameters for both detectors (Faster R-CNN and YOLO V3) are as follows: Stochastic Gradient Descent with Momentum (SGDM) is used as the training optimiser with 0.9 momentum. The Learning rate starts at 0.001 and then drops by a factor of 0.1 every six epochs. L2 regularisation of 0.005 is utilised to avoid overfitting. Training examples are shuffled every epoch to limit sequence memorising and avoid computing the gradients for the same batch of images.

Data augmentation

Data augmentation techniques, such as image flipping, can be employed to increase the variations and the number of training samples. Augmentation techniques can result in improved accuracy and enhanced model generalisation. Data augmentation techniques are usually applied to the training data only to produce a robust model and to avoid evaluation bias. In our experiments, data augmentation techniques are applied to the training dataset by horizontal flipping of the images and associated boxes to investigate their impact on the produced detectors.

Training details

The detectors are trained on a personal computer with a NVIDIA GeForce RTX 2080. Training time varies as the training process can be stopped early when the loss of the validation dataset plateaus or when the training reaches the maximum number of epochs (30 epochs). The largest mini-batch size that can accommodate the available memory is sought. The largest mini-batch sizes are 2 and 16 in the case of the Faster R-CNN and YOLO V3 detectors, respectively. Table 5.1 shows the training time of each model, the used mini-batch size, the stopping epoch, the trained model size, and the number of layers.

Table 5.1 **Training details.**

Metrics **Model**	**Feature network**	**Feature layer**	**No# of anchors**	**Training time (≈hours)**	**Mini-batch size**	**Stopping epoch**	**Size (MB) / No# of layers**
Faster R-CNN	ResNet-18	res4a_relu	30	16.3	2	15	42/82
Faster R-CNN*	ResNet-18	res4a_relu	30	18.6	2	15	42/82
Faster R-CNN	ResNet-18	res4b_relu	30	14	2	17	42/82
Faster R-CNN*	ResNet-18	res4b_relu	30	17.5	2	18	42/82
Faster R-CNN	ResNet-18	res5a_relu	30	5	2	15	48.5/82
Faster R-CNN*	ResNet-18	res5a_relu	30	4	2	11	48.5/82
Faster R-CNN	ResNet-18	res5b_relu	30	2.5	2	14	48.5/82

Faster R-CNN*	ResNet-18	res5b_relu	30	3	2	17	48.5/82
Faster R-CNN	ResNet-18	res4a_relu	23	34.7	2	9	42/82
Faster R-CNN*	ResNet-18	res4a_relu	23	56	2	14	42/82
Faster R-CNN	ResNet-18	res4b_relu	23	41.5	2	19	42/82
Faster R-CNN*	ResNet-18	res4b_relu	23	32	2	15	42/82
Faster R-CNN	ResNet-18	res5a_relu	23	4.5	2	13	48.4/82
Faster R-CNN*	ResNet-18	res5a_relu	23	5	2	13	48.4/82
Faster R-CNN	ResNet-18	res5b_relu	23	2.7	2	15	48.4/82
Faster R-CNN*	ResNet-18	res5b_relu	23	2.3	2	14	48.4/82
Faster R-CNN	ResNet-18	res4a_relu	3	17.6	2	10	41.9/82
Faster R-CNN*	ResNet-18	res4a_relu	3	23.3	2	14	41.9/82
Faster R-CNN	ResNet-18	res4b_relu	3	12	2	10	41.9/82
Faster R-CNN*	ResNet-18	res4b_relu	3	11.1	2	11	41.9/82
Faster R-CNN	ResNet-18	res5a_relu	3	1	2	9	48.2/82
Faster R-CNN*	ResNet-18	res5a_relu	3	1.1	2	12	48.2/82
Faster R-CNN	ResNet-18	res5b_relu	3	0.5	2	6	48.2/82
Faster R-CNN*	ResNet-18	res5b_relu	3	0.4	2	6	48.2/82
YOLO V3	ResNet-18	res4a_relu	30	0.9	16	25	10.6/48
YOLO V3*	ResNet-18	res4a_relu	30	1.1	16	29	10.6/48
YOLO V3	ResNet-18	res4b_relu	30	1	16	24	14.8/55
YOLO V3*	ResNet-18	res4b_relu	30	1.1	16	28	14.8/55
YOLO V3	ResNet-18	res5a_relu	30	1.1	16	25	41.1/64
YOLO V3*	ResNet-18	res5a_relu	30	1.2	16	28	41.1/64
YOLO V3	ResNet-18	res5b_relu	30	1.3	16	27	57.9/71
YOLO V3*	ResNet-18	res5b_relu	30	1.5	16	30	57.9/71
YOLO V3	ResNet-18	res4a_relu	23	0.8	16	24	10.4/48
YOLO V3*	ResNet-18	res4a_relu	23	1.1	16	28	10.4/48
YOLO V3	ResNet-18	res4b_relu	23	0.9	16	23	14.6/55
YOLO V3*	ResNet-18	res4b_relu	23	1.1	16	27	14.6/55
YOLO V3	ResNet-18	res5a_relu	23	1.2	16	27	40.8/64
YOLO V3*	ResNet-18	res5a_relu	23	1.3	16	30	40.8/64
YOLO V3	ResNet-18	res5b_relu	23	1.4	16	29	57.6/71
YOLO V3*	ResNet-18	res5b_relu	23	1.5	16	30	57.6/71
YOLO V3	ResNet-18	res4a_relu	3	0.8	16	24	10/48
YOLO V3*	ResNet-18	res4a_relu	3	0.9	16	27	10/48
YOLO V3	ResNet-18	res4b_relu	3	0.9	16	26	14.2/55
YOLO V3*	ResNet-18	res4b_relu	3	0.9	16	25	14.2/55
YOLO V3	ResNet-18	res5a_relu	3	1.3	16	30	39.8/64
YOLO V3*	ResNet-18	res5a_relu	3	1.2	16	28	39.8/64
YOLO V3	ResNet-18	res5b_relu	3	1.2	16	25	56.6/71

YOLO V3*	ResNet-18	res5b_relu	3	1.5	16	30	56.6/71
YOLO V3	ResNet-18	res4a&5a_relu	3	1.2	16	24	43.2/73
YOLO V3*	ResNet-18	res4a&5a_relu	3	1.2	16	24	43.2/73
YOLO V3	ResNet-18	res4a&5b_relu	3	1.5	16	27	60/80
YOLO V3*	ResNet-18	res4a&5b_relu	3	1.5	16	28	60/80
YOLO V3	ResNet-18	res4b&5a_relu	3	1.3	16	27	43.2/73
YOLO V3*	ResNet-18	res4b&5a_relu	3	1.5	16	28	43.2/73
YOLO V3	ResNet-18	res4b&5b_relu	3	1.5	16	27	60/80
YOLO V3*	ResNet-18	res4b&5b_relu	3	1.6	16	27	60/80
YOLO V3	ResNet-18	res4a&4b&5a_relu	3	1.4	16	16	46.4/82
YOLO V3*	ResNet-18	res4a&4b&5a_relu	3	1.8	16	30	46.4/82
YOLO V3	ResNet-18	res4a&4b&5b_relu	3	1.4	16	23	63.2/89
YOLO V3*	ResNet-18	res4a&4b&5b_relu	3	1.7	16	28	63.2/89
YOLO V3	ResNet-18	res4a&5a&5b_relu	3	1.5	16	24	72.9/89
YOLO V3*	ResNet-18	res4a&5a&5b_relu	3	1.7	16	27	72.9/89
YOLO V3	ResNet-18	res4b&5a&5b_relu	3	1.9	16	30	72.9/89
YOLO V3*	ResNet-18	res4b&5a&5b_relu	3	1.9	16	30	72.9/89

* System trained on augmented data.

MB = Megabyte.

Generally, Faster R-CNN detectors take significant training time compared to YOLO V3 detectors. The long training time is attributed to the detector architecture, which comprises a RPN attached to a Fast R-CNN. This is translated into many layers and large footprints. In contrast, the footprints and number of layers of YOLO V3 detectors vary depending on the feature extraction layer and the number of prediction heads. The smallest YOLO V3 detector has 48 layers and occupies a memory of 10 MB (Table 5.1).

5.2.4 Evaluation

Average Precision (AP) that can be computed from the Precision (P) Recall (R) curves is the standard metric of evaluating object detectors. Precision can be calculated using (5.1) as the ratio between True Positive (TP) instances to all positive instances. Whereas Recall can be calculated using (5.2) as the ratio between TP instances to the sum of TP and False Negative (FN) instances (ground truth positives).

$$P = \frac{TP}{TP+FP} \tag{5.1}$$

$$R = \frac{TP}{TP+FN} \tag{5.2}$$

Intersection over Union is used to determine which detection is TP, False Positive (FP), or FN. If there is an overlap between the detected object bounding box and the ground truth bounding box above a certain threshold (in our experiments, the threshold is 0.5), the detection is considered a TP. If the IoU is less than the threshold, the detection is FP. Lastly, if there is an object, but it has not been detected, or the object is detected with a wrong category, then it is a FN.

AP is then calculated as the area under the Precision/Recall curve for a specific class of objects using (5.3). A high AP value indicates the ability of the model to detect a specific class of objects efficiently and vice versa.

$$AP = \int_0^1 P(R)dR \tag{5.3}$$

Mean Average Precision (mAP) is used to assess the detector's abilities over all the dataset objects. mAP can be calculated using (5.4), where AP_k is the AP for class K and N is the total number of classes. The metric reflects the detector's performance over the whole dataset objects. In our experiments, we reported the AP for each class and the mAP for all classes.

$$mAP = \frac{1}{N}\sum_{K=1}^{N} AP_K \tag{5.4}$$

5.3 Results and discussion

The loss functions that have been used in the training process of Faster R-CNN and YOLO V3 are different, which can explain the difference in the results of Table 5.2. The objective function of Faster R-CNN follows the multi-task loss function of Fast R-CNN. However, it is minimised to a combination between the object classification loss and the bounding box regression loss (5.5). The classification loss is a *log* loss over two classes, while the regression loss is the smooth $L1$ loss [41]. Smooth $L1$ loss is less sensitive to outliers compared to $L2$ loss, especially when regression targets are unbounded, which may cause exploding gradients when $L2$ loss is used.

$$\mathscr{L}_{\text{Faster R-CNN}} = \mathscr{L}_{\text{cls}} + \mathscr{L}_{\text{box}} \tag{5.5}$$

On the other hand, the YOLO V3 loss function optimises the training process over three different losses. Like Faster R-CNN, the classification loss is the binary cross-entropy loss. Unlike Faster R-CNN, Mean Square Error (MSE) is used for the bounding box loss. Besides, YOLO V3 introduces the bounding boxes objectness loss [48] which is an additional binary cross-entropy loss for the overlapping between the predicted and the ground truth boxes. Ideally, the objectness score should equal one when the best overlapping anchor box among all anchor boxes overlaps with the ground truth box. The predictions are ignored when other anchor boxes (not the best overlapping anchor box) overlaps with the object box. This means that there is one anchor box assigned for each ground truth object [48].

$$\mathscr{L}_{\text{YOLO V3}} = \mathscr{L}_{\text{cls}} + \mathscr{L}_{\text{box}} + \mathscr{L}_{\text{obj}} \tag{5.6}$$

Table 5.2 shows that the lowest validation loss achieved using Faster R-CNN is 0.192 using res5a_relu as the feature extraction layer with three anchor boxes. The lowest validation losses achieved using YOLO V3 with single, double, and triple heads are 2.52, 2.35, and 2.38, respectively. Like Faster R-CNN, the best achieved validation loss YOLO V3 detectors use three anchor boxes. Unlike Faster R-CNN, YOLO detectors are trained using augmented data and with different feature extraction layers.

Table 5.2 **Validation loss.**

Model	**Feature network**	**Feature layer**	**No# anchors**	**Validation loss**
Faster R-CNN	ResNet-18	res4a_relu	30	0.681
Faster R-CNN*	ResNet-18	res4a_relu	30	0.719
Faster R-CNN	ResNet-18	res4b_relu	30	0.575
Faster R-CNN*	ResNet-18	res4b_relu	30	0.627
Faster R-CNN	ResNet-18	res5a_relu	30	0.369
Faster R-CNN*	ResNet-18	res5a_relu	30	0.399
Faster R-CNN	ResNet-18	res5b_relu	30	0.396
Faster R-CNN*	ResNet-18	res5b_relu	30	0.479
Faster R-CNN	ResNet-18	res4a_relu	23	0.639
Faster R-CNN*	ResNet-18	res4a_relu	23	0.639
Faster R-CNN	ResNet-18	res4b_relu	23	0.521
Faster R-CNN*	ResNet-18	res4b_relu	23	0.529
Faster R-CNN	ResNet-18	res5a_relu	23	0.303
Faster R-CNN*	ResNet-18	res5a_relu	23	0.311
Faster R-CNN	ResNet-18	res5b_relu	23	0.318
Faster R-CNN*	ResNet-18	res5b_relu	23	0.342
Faster R-CNN	ResNet-18	res4a_relu	3	0.366
Faster R-CNN*	ResNet-18	res4a_relu	3	0.380
Faster R-CNN	ResNet-18	res4b_relu	3	0.310
Faster R-CNN*	ResNet-18	res4b_relu	3	0.328
Faster R-CNN	ResNet-18	res5a_relu	3	**0.192**
Faster R-CNN*	ResNet-18	res5a_relu	3	0.201
Faster R-CNN	ResNet-18	res5b_relu	3	0.336
Faster R-CNN*	ResNet-18	res5b_relu	3	0.264
YOLO V3	ResNet-18	res4a_relu	30	7.91

YOLO V3*	ResNet-18	res4a_relu	30	6.40
YOLO V3	ResNet-18	res4b_relu	30	6.89
YOLO V3*	ResNet-18	res4b_relu	30	5.23
YOLO V3	ResNet-18	res5a_relu	30	4.68
YOLO V3*	ResNet-18	res5a_relu	30	3.93
YOLO V3	ResNet-18	res5b_relu	30	4.20
YOLO V3*	ResNet-18	res5b_relu	30	3.57
YOLO V3	ResNet-18	res4a_relu	23	7.64
YOLO V3*	ResNet-18	res4a_relu	23	6.25
YOLO V3	ResNet-18	res4b_relu	23	6.31
YOLO V3*	ResNet-18	res4b_relu	23	5.17
YOLO V3	ResNet-18	res5a_relu	23	4.37
YOLO V3*	ResNet-18	res5a_relu	23	3.90
YOLO V3	ResNet-18	res5b_relu	23	3.96
YOLO V3*	ResNet-18	res5b_relu	23	3.40
YOLO V3	ResNet-18	res4a_relu	3	6.22
YOLO V3*	ResNet-18	res4a_relu	3	4.73
YOLO V3	ResNet-18	res4b_relu	3	4.67
YOLO V3*	ResNet-18	res4b_relu	3	3.75
YOLO V3	ResNet-18	res5a_relu	3	3.27
YOLO V3*	ResNet-18	res5a_relu	3	2.85
YOLO V3	ResNet-18	res5b_relu	3	2.74
YOLO V3*	ResNet-18	res5b_relu	3	**2.52**
YOLO V3	ResNet-18	res4a&5a_relu	3	3.56
YOLO V3*	ResNet-18	res4a&5a_relu	3	2.95
YOLO V3	ResNet-18	res4a&5b_relu	3	2.95
YOLO V3*	ResNet-18	res4a&5b_relu	3	**2.35**
YOLO V3	ResNet-18	res4b&5a_relu	3	3.52
YOLO V3*	ResNet-18	res4b&5a_relu	3	2.97
YOLO V3	ResNet-18	res4b&5b_relu	3	2.93
YOLO V3*	ResNet-18	res4b&5b_relu	3	2.51
YOLO V3	ResNet-18	res4a&4b&5a_relu	3	3.79

YOLO V3*	ResNet-18	res4a&4b&5a_relu	3	**2.38**
YOLO V3	ResNet-18	res4a&4b&5b_relu	3	3.28
YOLO V3*	ResNet-18	res4a&4b&5b_relu	3	2.94
YOLO V3	ResNet-18	res4a&5a&5b_relu	3	3.27
YOLO V3*	ResNet-18	res4a&5a&5b_relu	3	2.46
YOLO V3	ResNet-18	res4b&5a&5b_relu	3	3.32
YOLO V3*	ResNet-18	res4b&5a&5b_relu	3	2.57

* System trained on augmented data.

Earlier feature extraction layers in the network have higher spatial resolutions but may extract less semantic information compared to layers further down the network. High spatial resolution features are better for small and medium size objects but not for large size ones. In contrast, strong semantic information is important for large size objects. However, due to the successive downsampling of the feature maps as the network goes deep, this information is lost for small objects. This makes the choice of the feature extraction layer a challenging task. As an example from Table 5.3, the AP of the smallest object in the proposed dataset (key slot) using earlier feature extraction layers such as res4a_relu or res4b_relu is significantly better than the AP when later layers such as res5a_relu or res5b_relu are used.

On the other hand, using res5a_relu or res5b_relu as the feature extraction layers on the largest size object in the proposed dataset (door) produce better AP than using res4a_relu or res4b_relu. This can be clearly seen from YOLO results. In contrast, Faster R-CNN results do not reflect this fact.

YOLO V3 detector can make predictions using multiple prediction heads over different scale feature maps in a similar approach to Feature Pyramid Network (FPN) [52]. The first head makes predictions over the first feature map. The second head makes predictions over a concatenation of the current feature map, after up-sampling, and the previous feature map. The same approach is followed for the other heads. Thus, semantic information and fine-grained details can be obtained from the up-sampled and high-resolution feature maps, respectively.

This approach allows the prediction of different scale objects, where small size objects can be detected from the high-resolution maps and large size objects can be extracted from strong semantic feature maps.

On the other hand, using a single feature map for prediction is less efficient than predictions over multiple feature maps, even with multiple scale anchor boxes (pyramid of anchors) that are used in Faster R-CNN [29]. Overall, the performance of YOLO V3 using single or multiple prediction heads are significantly better than that of Faster R-CNN for all object sizes.

Table 5.3 **Detector performance in terms of AP for each class and mAP for all classes using different training parameters.**

Model	**Feature network**	**Feature layer**	**No# of anchors**	**AP for each class**								**mAP**
				Door	*Key slot*	*Fire extinguisher*	*ID reader*	*Moveable door handle*	*Pull door handle*	*Push button*	*Push door handle*	
Faster R-CNN	ResNet-18	res4a_relu	30	0.911	0.061	0.587	0.315	0.330	0.354	zero	0.205	0.345
Faster R-CNN*	ResNet-18	res4a_relu	30	0.916	0.066	0.639	0.285	0.379	0.340	zero	0.250	0.359
Faster R-CNN	ResNet-18	res4b_relu	30	0.911	0.089	0.666	0.274	0.404	0.415	zero	0.289	0.381
Faster R-CNN*	ResNet-18	res4b_relu	30	0.905	0.088	0.709	0.215	0.394	0.379	zero	0.238	0.366
Faster R-CNN	ResNet-18	res5a_relu	30	0.810	zero	0.407	0.071	0.081	zero	zero	0.029	0.174
Faster R-CNN*	ResNet-18	res5a_relu	30	0.799	zero	0.396	0.088	0.056	zero	zero	zero	0.167
Faster R-CNN	ResNet-18	res5b_relu	30	0.504	zero	0.154	zero	0.011	zero	zero	zero	0.083
Faster R-CNN*	ResNet-18	res5b_relu	30	0.432	zero	0.097	zero	0.002	zero	zero	zero	0.066
Faster R-CNN	ResNet-18	res4a_relu	23	0.883	0.092	0.673	0.190	0.298	0.317	zero	0.280	0.341
Faster R-CNN*	ResNet-18	res4a_relu	23	0.890	0.086	0.633	0.203	0.411	0.351	zero	0.244	0.352
Faster R-CNN	ResNet-18	res4b_relu	23	**0.924**	0.068	0.742	0.426	0.427	0.384	0.279	0.222	**0.434**
Faster R-CNN*	ResNet-18	res4b_relu	23	0.910	0.109	0.642	0.285	0.409	0.344	0.197	0.242	0.392
Faster R-CNN	ResNet-18	res5a_relu	23	0.780	zero	0.388	0.132	0.069	zero	zero	0.071	0.180
Faster R-CNN*	ResNet-18	res5a_relu	23	0.777	zero	0.361	zero	0.041	zero	zero	zero	0.147
Faster R-CNN	ResNet-18	res5b_relu	23	0.505	zero	0.242	0.068	0.035	zero	zero	zero	0.106
Faster R-CNN*	ResNet-18	res5b_relu	23	0.431	zero	0.088	zero	0.009	zero	zero	zero	0.066
Faster R-CNN	ResNet-18	res4a_relu	3	0.873	0.018	0.692	0.211	0.390	0.202	0.000	0.082	0.308
Faster R-CNN*	ResNet-18	res4a_relu	3	0.869	0.066	0.729	0.177	0.361	0.356	0.000	0.108	0.333
Faster R-CNN	ResNet-18	res4b_relu	3	0.909	0.105	0.602	0.481	0.396	0.338	zero	0.190	0.378
Faster R-CNN*	ResNet-18	res4b_relu	3	0.889	0.034	0.754	0.180	0.418	0.354	zero	0.233	0.357
Faster R-CNN	ResNet-18	res5a_relu	3	0.745	zero	0.415	zero	0.028	zero	zero	zero	0.148
Faster R-CNN*	ResNet-18	res5a_relu	3	0.749	zero	0.257	zero	0.033	zero	zero	zero	0.130
Faster R-CNN	ResNet-18	res5b_relu	3	0.123	zero	0.116	zero	0.004	zero	zero	zero	0.030

Faster R-CNN*	ResNet-18	res5b_relu	3	0.502	zero	0.127	zero	0.002	zero	zero	zero	0.079
YOLO V3	ResNet-18	res4a_relu	30	0.516	0.462	0.714	0.657	0.632	0.536	0.500	0.458	0.559
YOLO V3*	ResNet-18	res4a_relu	30	0.561	0.556	0.787	0.739	0.744	0.543	0.566	0.470	0.620
YOLO V3	ResNet-18	res4b_relu	30	0.573	0.510	0.788	0.751	0.716	0.566	0.666	0.538	0.638
YOLO V3*	ResNet-18	res4b_relu	30	0.670	0.546	0.797	0.807	0.779	0.575	0.689	0.648	0.668
YOLO V3	ResNet-18	res5a_relu	30	0.721	0.156	0.843	0.618	0.593	0.519	0.833	0.437	0.590
YOLO V3*	ResNet-18	res5a_relu	30	0.786	0.230	0.838	0.670	0.616	0.510	0.739	0.412	0.600
YOLO V3	ResNet-18	res5b_relu	30	0.760	0.192	0.856	0.604	0.621	0.482	0.797	0.437	0.593
YOLO V3*	ResNet-18	res5b_relu	30	0.809	0.196	0.826	0.634	0.617	0.462	0.774	0.400	0.589
YOLO V3	ResNet-18	res4a_relu	23	0.563	0.424	0.720	0.687	0.633	0.578	0.566	0.407	0.572
YOLO V3*	ResNet-18	res4a_relu	23	0.609	0.474	0.754	0.680	0.712	0.539	0.493	0.505	0.596
YOLO V3	ResNet-18	res4b_relu	23	0.632	0.445	0.766	0.731	0.675	0.658	0.600	0.455	0.620
YOLO V3*	ResNet-18	res4b_relu	23	0.699	0.496	0.825	0.760	0.743	0.567	0.622	0.515	0.653
YOLO V3	ResNet-18	res5a_relu	23	0.771	0.194	0.806	0.623	0.622	0.517	0.740	0.405	0.584
YOLO V3*	ResNet-18	res5a_relu	23	0.791	0.176	0.753	0.645	0.580	0.454	0.800	0.405	0.575
YOLO V3	ResNet-18	res5b_relu	23	0.809	0.192	0.826	0.617	0.592	0.477	0.833	0.527	0.609
YOLO V3*	ResNet-18	res5b_relu	23	0.821	0.166	0.777	0.654	0.591	0.471	0.866	0.362	0.588
YOLO V3	ResNet-18	res4a_relu	3	0.652	0.502	0.805	0.762	0.721	0.643	0.758	0.555	0.674
YOLO V3*	ResNet-18	res4a_relu	3	0.726	0.523	0.882	0.817	0.767	0.572	0.849	0.673	0.726
YOLO V3	ResNet-18	res4b_relu	3	0.745	0.558	0.846	0.804	0.750	0.641	0.755	0.686	0.723
YOLO V3*	ResNet-18	res4b_relu	3	0.815	0.582	0.922	0.868	0.777	0.665	0.755	0.736	**0.765**
YOLO V3	ResNet-18	res5a_relu	3	0.834	0.195	0.859	0.712	0.692	0.562	0.861	0.511	0.653
YOLO V3*	ResNet-18	res5a_relu	3	0.855	0.188	0.877	0.725	0.728	0.501	0.691	0.483	0.631
YOLO V3	ResNet-18	res5b_relu	3	0.873	0.212	0.874	0.766	0.736	0.523	0.695	0.581	0.657
YOLO V3*	ResNet-18	res5b_relu	3	0.887	0.189	0.881	0.768	0.732	0.528	0.833	0.522	0.667
YOLO V3	ResNet-18	res4a&5a_relu	3	0.834	0.499	0.914	0.792	0.754	0.599	0.800	0.512	0.713
YOLO V3*	ResNet-18	res4a&5a_relu	3	0.856	0.595	0.911	0.843	0.805	0.517	0.866	0.682	0.759
YOLO V3	ResNet-18	res4a&5b_relu	3	0.879	0.636	0.915	0.856	0.821	0.617	0.769	0.688	0.772

YOLO V3*	ResNet-18	res4a&5b_relu	3	0.902	0.614	0.908	0.865	0.843	0.630	0.777	0.743	0.785
YOLO V3	ResNet-18	res4b&5a_relu	3	0.835	0.537	0.904	0.825	0.767	0.613	0.798	0.574	0.731
YOLO V3*	ResNet-18	res4b&5a_relu	3	0.862	0.582	0.914	**0.904**	0.812	0.606	0.850	0.682	0.776
YOLO V3	ResNet-18	res4b&5b_relu	3	0.888	0.573	0.922	0.855	0.807	0.684	0.881	0.679	**0.786**
YOLO V3*	ResNet-18	res4b&5b_relu	3	0.895	0.604	0.915	0.892	0.844	0.621	0.753	0.707	0.778
YOLO V3	ResNet-18	res4a & 4b &5a_relu	3	0.847	0.510	0.896	0.803	0.766	0.695	0.793	0.667	0.747
YOLO V3*	ResNet-18	res4a & 4b &5a_relu	3	0.907	0.**673**	0.937	0.862	**0.869**	0.638	**0.893**	**0.766**	**0.818**
YOLO V3	ResNet-18	res4a & 4b &5b_relu	3	0.892	0.454	0.896	0.773	0.759	**0.724**	0.757	0.607	0.732
YOLO V3*	ResNet-18	res4a & 4b &5b_relu	3	0.917	0.571	0.941	0.887	0.831	0.655	0.775	0.730	0.788
YOLO V3	ResNet-18	res4a & 5a &5b_relu	3	0.878	0.593	0.921	0.792	0.775	0.616	0.775	0.651	0.750
YOLO V3*	ResNet-18	res4a & 5a &5b_relu	3	0.892	0.561	**0.954**	0.877	0.833	0.612	0.749	0.641	0.764
YOLO V3	ResNet-18	res4b & 5a &5b_relu	3	0.877	0.539	0.902	0.787	0.770	0.552	0.782	0.535	0.718
YOLO V3*	ResNet-18	res4b & 5a &5b_relu	3	0.900	0.516	0.932	0.847	0.837	0.601	0.870	0.709	0.776

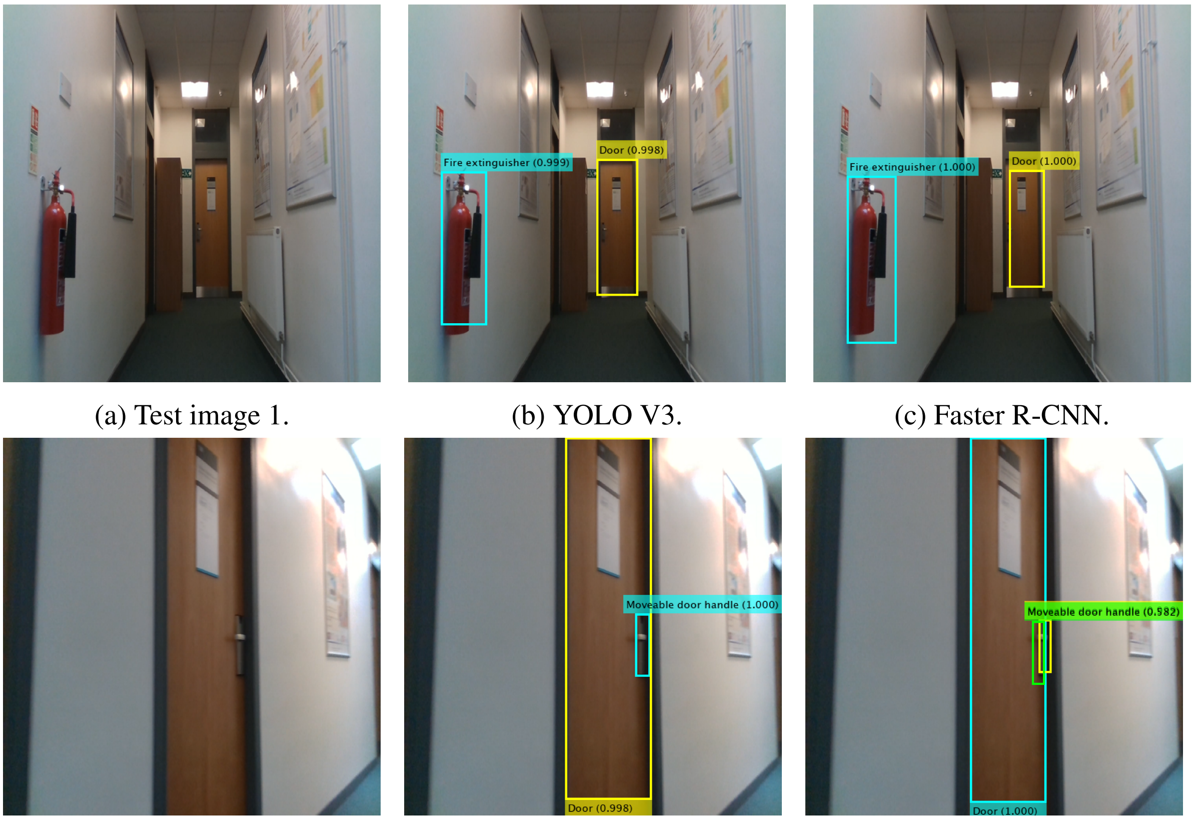
* System trained on augmented data.

Table 5.3 shows that the best Faster R-CNN detector achieved a mAP of 0.434. Whereas the best single head YOLO V3 detector achieved a mAP of 0.765. Faster R-CNN detector uses res4b_relu as the feature extraction layer with 23 anchor boxes. Similarly, YOLO V3 detector uses res4b_relu as the feature extraction layer but with only three anchor boxes. The best double and triple heads YOLO V3 detectors achieved a mAP of 0.786 and 0.818, respectively. Both of them use only three anchor boxes. Lastly, the overall best performance detector is YOLO V3 with triple heads and trained on augmented data.

Fig 5.3 shows detection examples of the best performing Faster R-CNN (23 anchor boxes and res4b_relu feature extraction layer) and YOLO V3 single head (3 anchor boxes, res4b_relu feature extraction layer, and trained using augmented training data) detectors on two test images. The two networks can predict the class categories and the bounding boxes with high confidence score. However, Faster R-CNN predicts two bounding boxes for the same object (the moveable door handle in Fig 5.3f) one with a high confidence score of 0.99, while the second one confidence score is relatively low (0.58). The used confidence score threshold value in the experiments is 0.5. A higher threshold value can discard the second box. Other Faster R-CNN detectors find it challenging to detect all the objects in the test images.

In contrast, other YOLO V3 detectors can localise all the objects in the test images with minor differences in the confidence scores. The predicted bounding box using Faster R-CNN for the fire extinguisher in Fig 5.3c covers the whole object, unlike the produced bounding box from the YOLO V3 detector (Fig 5.3b). In comparison, the bounding box for the door object in the same images is fully covered by the YOLO V3 detector and partially covered by the Faster R-CNN detector.

Detection examples of the best performance three heads YOLO V3 detector are shown in Fig 5.4. The detector can localise small size objects, such as ID reader and pull door handle, along with large and medium size objects.


Fire extinguisher (0.999)
Door (0.998)
Fire extinguisher (1.000)
Door (1.000)
Moveable door handle (1.000)
Door (0.998)
Moveable door handle (0.582)
Door (1.000)

(a) Test image 1. (b) YOLO V3. (c) Faster R-CNN.

(d) Test image 2. (e) YOLO V3. (f) Faster R-CNN.

Fig. 5.3 The prediction results of YOLO V3 and Faster R-CNN on two test images.

Detailed investigation of different detectors using different parameters can give insights into the suitable detector for a given application. YOLO V3 with three detection heads and three anchor boxes has achieved the best performance using the augmented data for training on the proposed dataset. Also, the system needs less time for training compared to Faster R-CNN detectors.

It can be concluded from the results that increasing the number of anchor boxes is not enhancing the detector's accuracy. However, feature extraction layers significantly impact the detector's performance. Feature extraction layers greatly affect the detector's ability to capture different size objects. An earlier layer can better localise small size objects, while a deeper layer can better encode large size objects. Consequently, multi-head detectors can capture different size objects efficiently because the predictions are made over several feature maps.

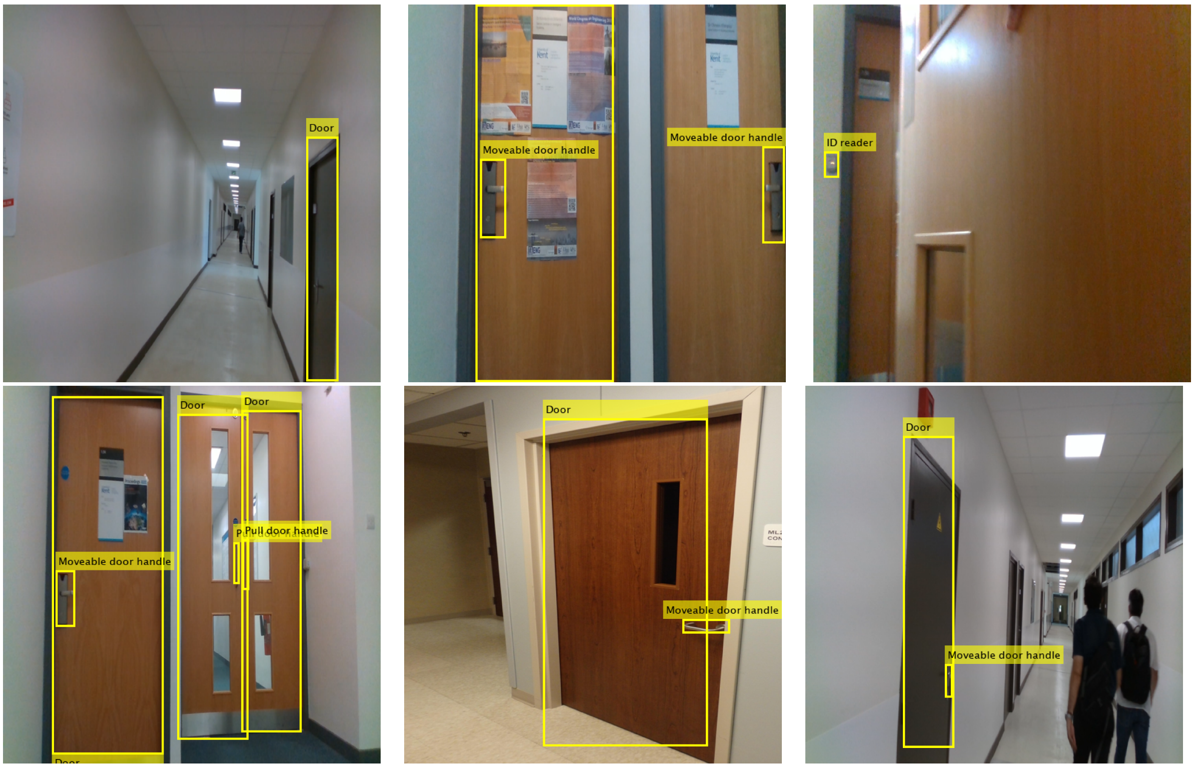


Fig. 5.4 The prediction results of the best performing YOLO V3 detector with three heads.

5.3.1 Visualisation of detector predictions

To validate the reliability of the best single head YOLO V3 system, the proposed techniques for visualising the network decisions are used. However, the proposed techniques (WS-Grad and Concat-Grad) are applied to classification tasks as shown in Chapter 4. Thus, applying them to different tasks, such as object detection, is a novel contribution.

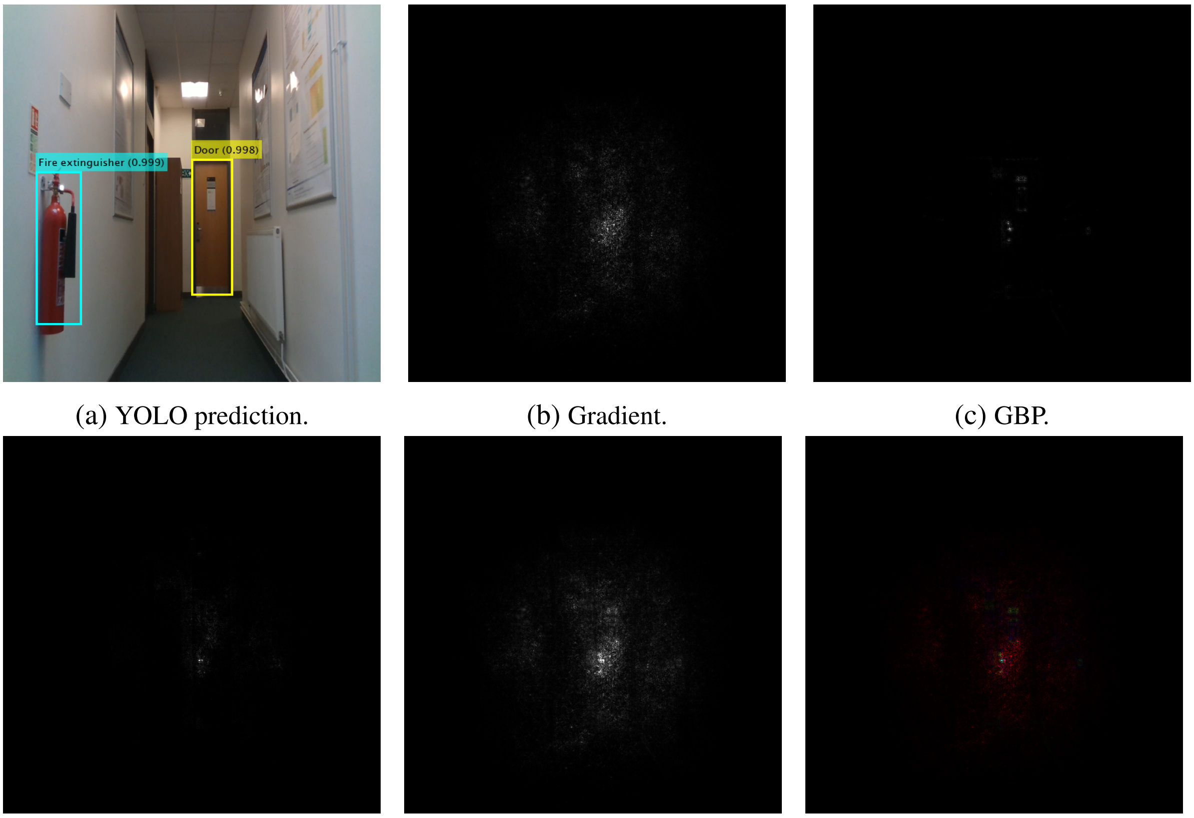
The output of a YOLO V3 single head detector is $N \times N \times [3 \times (4+1+8)]$ where N represents the convolution filter size, 3 represents the number of anchor boxes, 4 represents the bounding box offsets, 1 represents the objectness score, and 8 represents the scores of classes. The convolutional filter size is 16, resulting in an output tensor of $16 \times 16 \times 39$ size. The target object score needs to be tracked back through the network layers to find the contributing input pixels to the target object score. However, the final output scores are obtained by multiplying the confidence score and the objectness score.

Tracking the final output scores through the output tensor is challenging because of the post-processing step that extracts the confidence and objectness score from the output tensor and multiples them. This post-processing step needs to be reversed. Besides, the two values that produce the final output score need to be tracked back (using backpropagation) through the network to find the corresponding contributing object. In contrast, the final output score can be directly tracked in classification tasks that use softmax layer in the output, whereas YOLO V3 does not have any softmax layers. The locations of the confidence and objectness scores are identified by analysing the output tensor. Consequently, the output target score has become feasible to be tracked through the YOLO network. In other words, by identifying the location of the confidence and objectness scores in the output tensor, the output score of the target object has become trackable.

Fig 5.5 and Fig 5.6 show the contributing pixels to the detector predictions of the door and the fire extinguisher, respectively. Gradient, GBP, IG, WS-Grad and Concat-Grad attribution maps are compared. Gradient, GBP and IG show specific individual features. However, our proposed methods show comprehensive maps that contains all of the contributing pixels. Furthermore, the Concat-Grad method shows the features of each individual method in different colours in a single map which enriches the output and makes it more descriptive and understandable.

Generally, the research area of visualising the network decision for object detection tasks is limited. The proposed techniques can help researchers in the field of object detection to understand and trust the decisions of their systems. Besides, visualising the detector prediction can help to debug the system in the case of bias or error.

One limitation of this study is in the visualisation of Faster R-CNN predictions. Faster R-CNN detectors contain some layers that are challenging to reverse (backpropagate the output through them) such as region proposal and region of interest pooling layers.



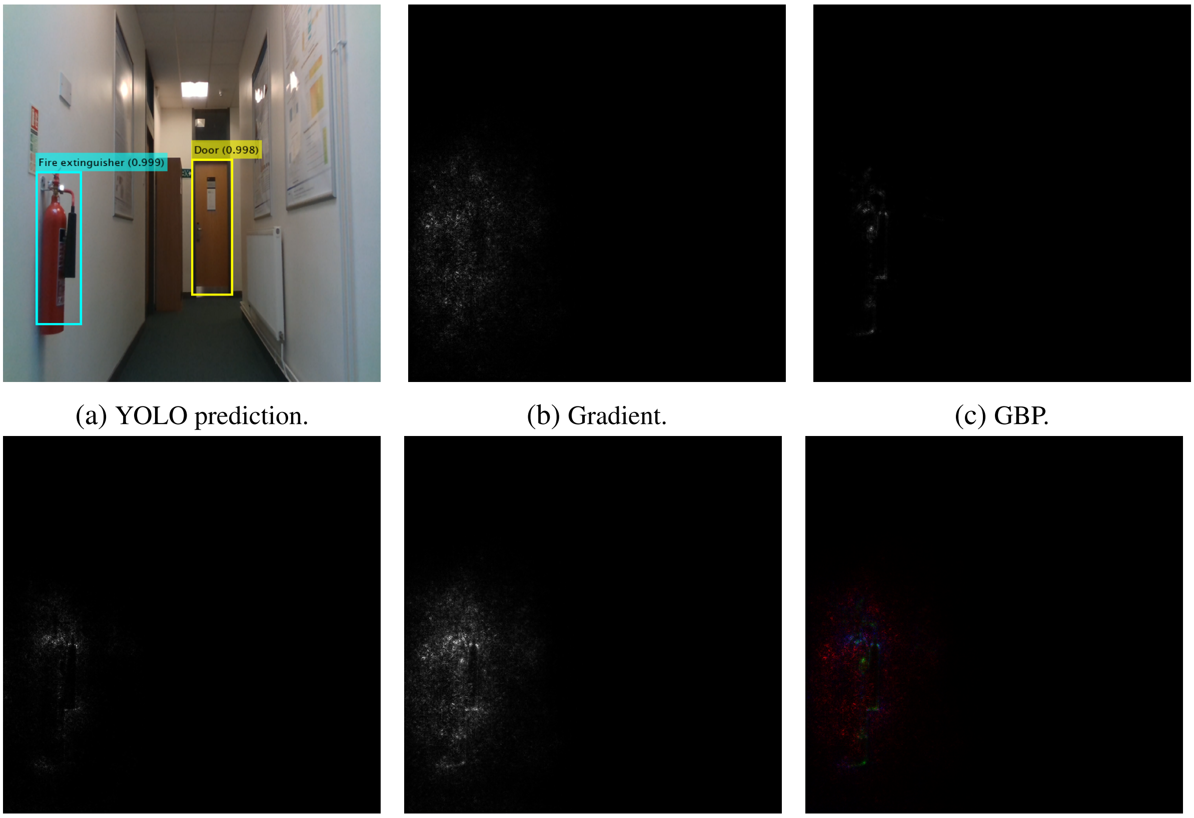
(a) YOLO prediction.
(b) Gradient.
(c) GBP.
(d) IG.
(e) WS-Grad (proposed).
(f) Concat-Grad (proposed).

Fig. 5.5 Gradients based attribution methods and the proposed methods to visualise the door prediction using YOLO V3.

5.4 Conclusion

In this chapter, a comprehensive investigation of object detectors performance is presented. Detectors performance using different feature extraction layers, different number of anchor boxes is investigated. Moreover, the impact of training the detectors using data augmentation techniques is highlighted.

Data augmentation positively impacts the produced detectors and results in lower validation loss compared to detectors trained on data without augmentation techniques. However, increasing the number of anchor boxes do not enhance the detector's performance. In contrast, it can negatively impact the performance.



(a) YOLO prediction.

(b) Gradient.

(c) GBP.

(d) IG.

(e) WS-Grad (proposed).

(f) Concat-Grad (proposed).

Fig. 5.6 Gradients based attribution methods and the proposed methods to visualise the fire extinguisher prediction using YOLO V3.

YOLO detectors with multi-prediction heads have achieved the best performance. Furthermore, YOLO detectors has less layers, less footprint, and train faster than Faster R-CNN detectors.

Feature extraction layer can significantly impact the ability of the detector to localise different size objects. An earlier feature extraction layer can better detect small size objects, as it can preserve spatial information. Whereas later layers are better with large size objects because it better encodes the object's features. Successive down-sampling of features as the object propagates through the network layers strengthen the encoded features, but spatial information of small size object can be lost. Consequently, earlier feature extraction layer is advisable with small size object detection applications.

This chapter greatly contributes to the visualisation techniques applied to object detection tasks as this research area is very limited. It is important to attain not only an accurate system but also a system that can explain its predictions. Black box systems, like deep convolutional networks, must provide adequate insights into the system's predictions. Developers, policymakers and legislators often require a certain level of system transparency to approve/appreciate such technologies and can self-assuredly conclude that the underlying system is robust and reliable. Consequently, these kinds of transparent systems can be approved and used for critical real-life applications. The proposed explanation techniques help achieve this by providing high-resolution and sharp heatmaps for the contributing features to the network decision compared to the state-of-the-art ones. This can greatly help to understand and explain the detector's behaviour.

Chapter 6

Semantic Segmentation with Practical Applications

6.1 Introduction

Electrical Powered Wheelchair (EPW) users may find navigation through indoor and outdoor environments a significant challenge due to their disabilities. Moreover, they may suffer from near-sightedness or cognitive problems that limit their driving experience. Developing a system that can help EPW users to navigate safely by providing visual feedback and further assistance when needed can have a significant impact on the user's wellbeing. Many accidents and injuries have been reported for users injuring themselves or falling from EPW as they could not distinguish between pavement edges and car routes or walls and doors [150–152]. Moreover, some users cannot be prescribed an EPW because of their disability [15].

A computer vision system that can help disabled users to distinguish between different components of a complex environment can significantly impact the user's experience, specifically if a visual feedback can be presented. This chapter presents computer vision systems based on deep learning, with an architecture based on residual blocks that can semantically segment high-resolution images. The systems are modified versions of DeepLab version 3 plus that can process high-resolution input images. Besides, they can simultaneously process images from indoor and outdoor environments, which is challenging due to the difference in data distribution and context. The proposed systems replace the base network with a smaller one and modify the encoder-decoder architecture. Nevertheless, they produce high-quality outputs with fast inference speed compared to the systems with deeper base networks.

EPWs' users who disfavour fully autonomous or semi-autonomous navigation (shared control) or who want to be in full control of the EPW can benefit from such a system that provides environmental cues for guidance. Autonomous systems can be frustrating to some users when they try to approach an object or a door, but the collision avoidance system prevents them from doing so. One of the main requirements for an EPW system is not to act as a caretaker but instead as an assistant, and the user can override the system control at any point [153]. Visually and cognitively impaired users can benefit from such a system that guides them while giving them full control over the EPW. However, the proposed systems can be combined with autonomous ones or used as standalone systems, depending on the user's condition.

Two individual systems based on deep learning for pixel classification are presented. A manually collected dataset for indoor environments [126] and an outdoor dataset (Cambridge-driving Labeled Video Database (CamVid) [154]) are used to train the two systems. The systems' architectures are based on DeepLab3 plus [90] (hereafter DLV3+ for simplicity) for semantic segmentation and ResNet-18 is used as the feature extraction backbone network [51]. ResNet-18 is an adequate choice as it has a smaller footprint and a fewer number of layers when compared to other versions of the same family (ResNet50 and ResNet101).

In addition, three novel shared systems are proposed. The shared systems can semantically segment images from both indoor and outdoor environments simultaneously. The novelty of the proposed three shared systems is not only in the architecture but also in the elegance of reusing the learned information and weights by the individual systems without the need to retrain the shared systems. Most importantly, the shared systems can process two different environments with almost the same accuracy as the individual systems, which is challenging as the data (images) being processed comes from two different distributions (indoor and outdoor).

Results show the ability of the proposed systems to detect objects with sharp edges and high accuracy for indoor and outdoor environments. The developed systems are deployed on a GPU based board and then integrated on an EPW for practical usage and evaluation.

The impact of the vibrations on the performance of the smart computer vision systems installed on the EPW, such as the semantic segmentation system, is investigated. Vibrations due to ramps, damaged terrains, and uneven tarmac are not only impacting the health and comfort of disabled users (Appendix B) but also can impact the accuracy of the smart systems installed on the powered wheelchair. We set up an experiment to assess the impact of vibrations on the semantic segmentation system installed on the powered wheelchair to provide the users with environmental cues. Environmental cues can help visually impaired users to locate objects in their surroundings [155].

This chapter is organised as follows: challenges, systems architectures and training strategies are presented in section 6.2. Results are discussed in section 6.3. Limitations of the systems and possible solutions are illustrated in section 6.4. Section 6.5 presents the practical implementation of the system. Vibration impact on the semantic segmentation system is investigated in section 6.6. Lastly, the chapter is concluded, and the future work is highlighted in section 6.7.

6.2 Methodology

6.2.1 Challenges

The majority of the objects in the proposed indoor dataset can be classified as small size objects. Small size objects do not possess enough pixels to be utilized for feature extraction. Also, distinguishing between different door handles represents a great challenge because of their common colour and location in the dataset's images. Consequently, conventional object detection and semantic segmentation (SS) techniques traditionally employed to detect objects occupying a large portion of images cannot be used [156]. In particular, object boundaries and intersections between objects are very poorly detected or segmented using conventional deep learning methods [83].

A semantic segmentation system that can process images of two different contexts, such as indoor and outdoor images, is another major challenge, not only because the images of the datasets are limited but also the type of images is different. A system that is trained to semantically segment indoor images can not perform well on outdoor scenarios and vice versa. This is because datasets' images have different distributions and contexts. We introduce shared systems that incorporate both scenarios with adequate accuracy and processing speed.

There will always be a trade-off between the system's speed and accuracy. As the proposed systems are meant to be deployed on an EPW for environmental parsing, we propose using a relatively small backbone network (ResNet-18) to achieve better Frame Per Second (FPS) than ResNet-50 and Xception without sacrificing accuracy thanks to the residual block architecture. Table 6.1 shows the number of layers and trainable parameters of the tested systems with different base networks.

6.2.2 System architecture

The proposed systems are based on DLV3+ architecture for semantic segmentation [90] with some modifications. The architecture's base network uses residual blocks, which help the system to process high-resolution images (960×540×3 pixels) using a deep network (many layers) without losing information because of the vanishing gradients problem. In the original implementation of DLV3+, ResNet-50, ResNet-101 [51], and Xception [157] networks are used as the system's feature extraction network. Various feature extraction networks are tested as the backbone of the systems, besides those used in the original implementation. However,

Table 6.1 **The number of layers and trainable parameters of the tested systems with different base networks.**

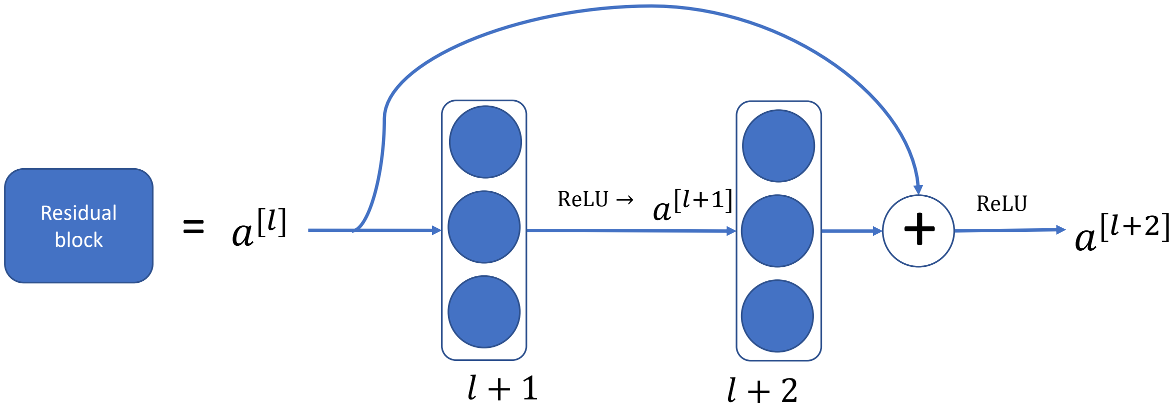
Metrics Model	Trainable parameters	Layers
$FCN-8s$	134.3 M	51
$FCN-16s$	134.3 M	47
$FCN-32s$	134.6 M	43
$SegNet\ (VGG-16)$	29.4 M	91
$SegNet\ (VGG-19)$	42.4 M	109
$U-Net$	30.9 M	58
$DLV3+(ResNet-18)$	20.6 M	100
$DLV3+(ResNet-50)$	44.1 M	206
$DLV3+(Xception)$	27.8 M	205
$Shared\ system\ 1$	30.0 M	133
$Shared\ system\ 2$	41.2 M	198
$Shared\ system\ 3$	41.2 M	200

M = Million.

ResNet-18 is the choice due to its small size and few parameters compared to networks form the same category (ResNet-50 and ResNet-101). Also, it can produce fast processing speed and comparable accuracy.

Very deep networks suffer from vanishing and exploding gradients [147, 148]. Residual blocks help to mitigate this problem by reusing the activations from previous layers until the adjacent layer learns its weights [51]. This allows the network to learn more low-level features without being worried about performance degradation as it goes deep. The architecture elegance is attributed to the short-cut connections that do not add either extra parameters or computational complexity [51]. A residual block structure can be seen in Fig 6.1.

Unlike FCN [83], DLV3+ uses the encoder-decoder structure [90]. The encoder part uses the same design of DLV3 [89], which uses dilated convolution ‘atrous’ to increase the receptive field of the layers. Atrous convolution is used to control the resolution by enlarging the field of view to incorporate a large context without increasing the number of parameters or computation. At the same time, a simple but effective design is used as a decoder network. Combined, they



$$z^{[l+i]} = W^{[l+i]}a^{[l]} + b^{[l+i]}$$
$$a^{[l+i]} = \sigma(z^{[l+i]})$$

σ: Rectified Linear Unit (ReLU)
$a^{[l]}$: activation of layer l.
$a^{[l+1]}$: activation of layer $l+1$ after applying ReLU.
$a^{[l+2]}$: activation of the linear output of layer $l+2$ ($z^{[l+2]}$)added to the activation of layer l then applying ReLU.

$$\boldsymbol{a}^{[l+2]} = \boldsymbol{\sigma}(\boldsymbol{z}^{[l+2]} + \boldsymbol{a}^{[l]})$$

Fig. 6.1 **Residual block.** The main building block for ResNet-18, ResNet-50, and ResNet-101.

represent the DLV3+ network. The encoder-decoder approach has proved its efficiency to refine object edges, resulting in better accuracy and Intersection over Union (IoU).

We adopt DLV3+ design but using ResNet-18 as a backbone feature extraction network (Fig 6.2). Besides, the input layer is modified to accept large size image inputs with 960×540×3 pixels. The indoor, outdoor and shared proposed systems are used to semantically segment images of both indoor and outdoor datasets.

Creating a system that can semantically segment indoor and outdoor environments simultaneously is challenging as data distribution and context differ. Also, the size of the datasets in both cases is limited, which means combing both datasets to train a single system can result in a non-robust model, a model that might not converge (reaching the global minima of the

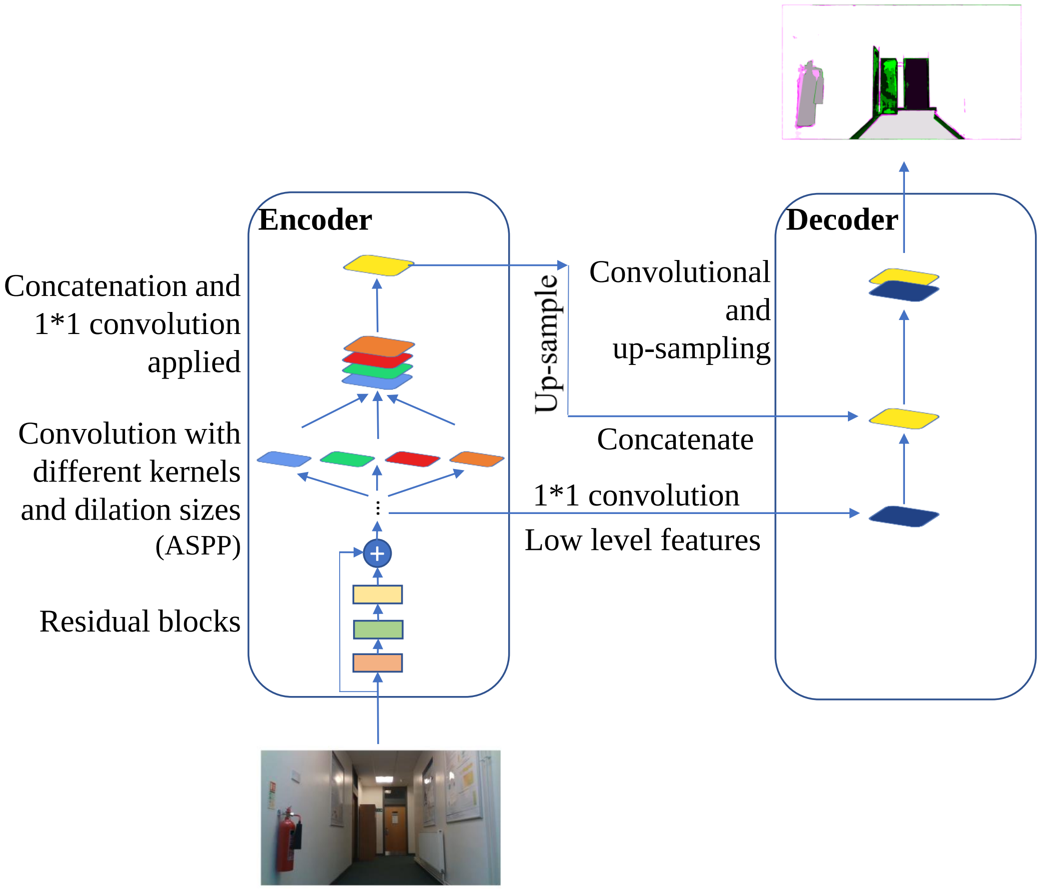

Encoder
Decoder
Concatenation and 1*1 convolution applied
Convolution with different kernels and dilation sizes (ASPP)
Residual blocks
Up-sample
Convolutional and up-sampling
Concatenate
1*1 convolution
Low level features

Fig. 6.2 **System architecture.** The encoder part with ASPP and the decoder part with simple bilinear upsampling.

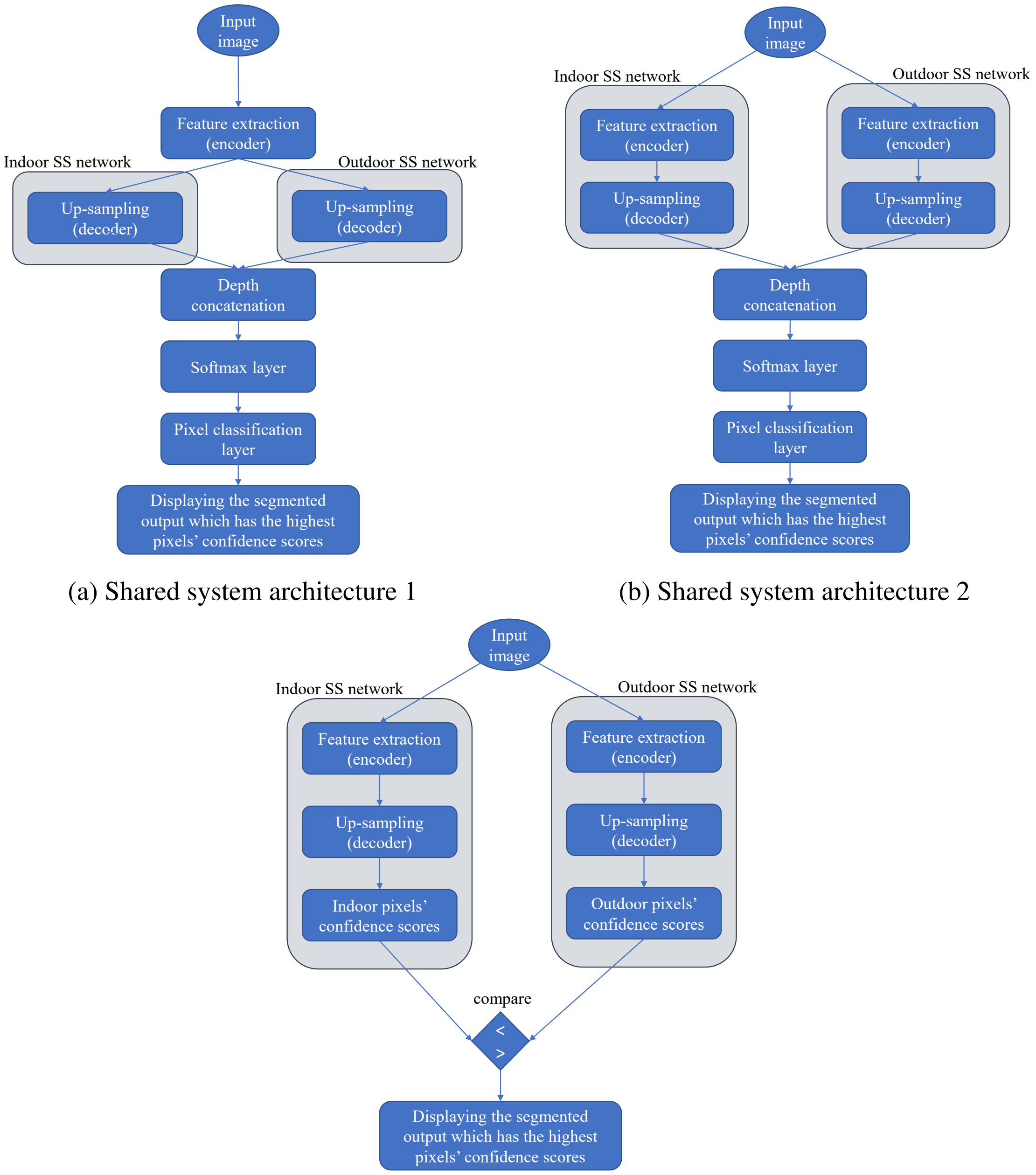
cost function). At the same time, if it is trained for a long time, the model can overfit the data. Consequently, it is challenging for any model to fit both scenarios. A very important research area where the proposed solutions can result in saved time and resources. It can save time because there is no need to retrain a new system. Resources also can be saved because one model can process multiple scenarios. Moreover, the proposed approach can help to create general models that can operate in multi-context situations.

Novel approaches are introduced by merging both the indoor and the outdoor systems after the training process of each system individually (Fig 6.3). The intuition is to make use of the learned information and weights by both individual systems (indoor and outdoor) without retraining a new system on a new combined dataset. The proposed techniques can help to combine systems from different domains, save training time and resources, and achieve adequate results in different environments simultaneously.

Different techniques of merging the two networks are explored. Each individual network consists of an encoder with a decoder attached to it. Thus, in the first trial, the encoder of one network is connected to the decoder of both networks. As the function of the encoder is to extract the features from the input image, it is dependent on the learned weights from the training data. Consequently, an encoder trained on indoor data can better extract features from indoor images than outdoor images and vice versa. Motivated by this fact, the encoders of both networks are included in the second trial. Although this technique increases the number of layers and the size of the proposed model, it can better encode input images from indoor and outdoor datasets compared to the usage of a single encoder from either network.

Our first trial is depicted in Fig 6.3a, which resulted in the proposed shared system 1. The system is constructed as follows: after training both systems (an indoor system on the indoor dataset and an outdoor system on the outdoor dataset), we extracted the feature extraction network (encoder) from one of the systems. Then, we connect this encoder to both decoders of the indoor and the outdoor systems. After that, we concatenate both outputs of both decoders. Lastly, the concatenated output is propagated through a softmax and pixel classification score layers that output the annotated image with the highest confidence score among all of the indoor and the outdoor classes.

The proposed shared system 1 is an end-to-end system that does not need any further post-processing steps. However, the system performance is highly impacted by the encoder part. This means that if we use the encoder part of the indoor system, the overall shared system performance on the indoor dataset will be better than that on the outdoor dataset and vice versa. Consequently, this system is biased by its encoder part. This leads to the second and third trials which are depicted in Fig 6.3b and Fig 6.3c, respectively.


Input image
Feature extraction (encoder)
Indoor SS network
Outdoor SS network
Up-sampling (decoder)
Up-sampling (decoder)
Depth concatenation
Softmax layer
Pixel classification layer
Displaying the segmented output which has the highest pixels' confidence scores
Input image
Indoor SS network
Outdoor SS network
Feature extraction (encoder)
Up-sampling (decoder)
Feature extraction (encoder)
Up-sampling (decoder)
Depth concatenation
Softmax layer
Pixel classification layer
Displaying the segmented output which has the highest pixels' confidence scores
Input image
Indoor SS network
Outdoor SS network
Feature extraction (encoder)
Up-sampling (decoder)
Indoor pixels' confidence scores
Feature extraction (encoder)
Up-sampling (decoder)
Outdoor pixels' confidence scores
compare
<
>
Displaying the segmented output which has the highest pixels' confidence scores

(a) Shared system architecture 1

(b) Shared system architecture 2

(c) Shared system architecture 3

Fig. 6.3 **Shared network architectures.** Shared system 1 uses either the trained feature extraction network (encoder) of the indoor or the outdoor semantic segmentation systems. Shared system 2 uses both feature extraction networks of the indoor and the outdoor systems. Shared system 3 uses the indoor and the outdoor semantic segmentation system simultaneously with an added post-processing step to display the annotated output that has the highest pixels' confidence scores.

In the second trial (Fig 6.3b), the encoders and the decoders of the trained indoor and outdoor semantic segmentation systems are integrated. The outputs of both decoders are up-sampled to the original image size. Then, both images are concatenated using the depth concatenation layer. After that, the concatenated output is propagated through softmax and pixel classification score layers. Lastly, the displayed output is the segmented image with the highest pixels' confidence scores across all of the 20 classes (9 indoor and 11 outdoor classes).

Shared system 2 performs well on both the indoor and the outdoor datasets. It is an end-to-end system that does not need any post-processing steps. However, scoring the pixels with respect to the 20 indoor and outdoor classes of the shared system 2 is more challenging than scoring 9 or 11 classes of the individual indoor and outdoor systems, respectively. It is a highly competitive scoring process between the 20 classes where the uncertainty increases, especially between dominant classes such as 'Background Wall' and 'Sky' from the indoor and the outdoor datasets, respectively. Consequently, the system's performance is adequate but not as good as the individual systems.

The main intuition behind shared system 3 (Fig 6.3c) is to make use of the high performance of the individual systems. We use both of the individual indoor and outdoor semantic segmentation systems to parse the same image. Then, we display the highest pixels' confidence scores annotated output to the user. The detailed process of shared system 3 is as follows: the encoders of both the indoor and the outdoor systems are included. Similarly, the decoder parts of both systems are included. Using the proposed shared system 3, we obtain two outputs from the two parallel systems. We then apply one post-processing step to determine which output from the two individual systems should be displayed. The mean of each row of the output pixels confidence scores is calculated for both individual systems, resulting in two vectors of means with the same height as the input image. Then, the maximum values of each vector are compared. If the indoor system achieves a maximum value that is higher than that of the outdoor system, then the input image is assumed to be an indoor image and vice versa. Accordingly, the system's output that achieves the highest maximum value is displayed.

Different comparison techniques of the pixels' confidence scores are tried, for example, comparing pixel by pixel and displaying the system's output that has the highest number of pixels with the highest pixels' confidence scores. However, the 'max(mean(score)' approach has achieved the best performance.

The proposed shared system 3 needs a post-processing step for the pixels comparison of the two individual systems. As the encoders and decoders of both systems are included, the system inference speed has slowed, which negatively impacts the system's real-time operation. On the bright side, the system can produce better results in both indoor and outdoor environments. One of the study's future work is to explore different system architectures that can enhance the system's inference speed while achieving high performance on two or more scenarios.

6.2.3 Training

The indoor and the outdoor systems are trained end-to-end with the following parameters: Stochastic Gradient Descent with Momentum (SGDM) is used as the training optimiser with 0.9 momentum. The learning rate starts at 0.001 and then drops by a factor of 0.3 every ten epochs. The aforementioned training parameters are chosen after several experiments with different parameters to achieve the best performance. To avoid overfitting, L2 regularisation is used. Training examples are shuffled every epoch to limit sequence memorising and avoid computing the gradients for the same batch of images. Image normalisation is employed to rescale all the pixels' values in the range of zero to one. Lastly, data augmentation with X and Y translations is employed to enhance model generalisation, which can increase the overall system accuracy. To avoid bias in favour of dominant classes, inverse frequency weighting is used to balance the classes weightings, where the class weights are the inverse of the class frequencies. This method increases class weights for under-represented classes. Additionally, different hyper-parameters and optimisation algorithms are tried to achieve the highest performance. Moreover, systems are trained several times under the same configurations to ensure reproducibility.

The proposed systems are trained on a personal computer with a NVIDIA GeForce RTX 2080. Training time varies as the training process can be stopped early when the loss of the validation dataset plateaus or when it reaches the maximum epochs of the training process (30 epochs). For the indoor dataset, the model's loss is validate every 200 iterations. However, the model's loss is validated every 50 iterations for the outdoor dataset. The difference in the two cases is attributed to the mini-batch size, the sizes of the datasets, and the model's size. The largest mini-batch size that can accommodate the available memory is sought. The largest mini-batch sizes are 8 and 4 in the case of the outdoor and the indoor datasets, respectively. The mini-batch size is reduced if the available memory can not accommodate the model size with a large mini-batch size. Consequently, the number of iterations per epoch varies. Table 6.2 shows the training time of each model, the mini-batch size, the stopping epoch and the trained model size.

Table 6.2 **Training details.**

Metrics Indoor/Outdoor **Model**	**Training time (≈hours)**	**Mini-batch size**	**Stopping epoch**	**Model size (MB)**
$FCN-8s$	3.5/0.75	2/2	14/6	477
$FCN-16s$	2/1.25	2/2	8/8	477
$FCN-32s$	2.25/2.25	2/2	9/14	478
$SegNet\ (VGG-16)$	5.5/8.5	2/2	19/23	104
$SegNet\ (VGG-19)$	9/8.75	2/2	26/21	142
$U-Net$	3/2.25	2/2	5/6	110
$DLV3+\ (ResNet-18)$	1.5/1.25	4/8	20/26	58.3
$DLV3+\ (ResNet-50)$	1/2	4/8	7/14	141
$DLV3+\ (Xception)$	9/3.5	4/8	17/18	83.4
$Shared\ system\ 1$	-	-	-	76.8
$Shared\ system\ 2$	-	-	-	116
$Shared\ system\ 3$	-	-	-	116.6

MB = Megabyte.

Systems are trained end-to-end using high-resolution and large-size training images of 960×540×3 pixels from the indoor and the outdoor datasets, unlike the original implementation of DLV3+, which crops patches of 513×513 size from the PASCAL VOC dataset [132] images during the training and testing processes. The proposed training approach enhances the system's ability to semantically segment small size objects alongside medium and larger size ones. Also, this boosts the effectiveness of large rate atrous convolutions as their weights can be applied to actual pixels and not to zero paddings.

6.3 Results and discussion

Standard metrics, such as mean Intersection over Union (mIoU), Accuracy, and BF score, are used to evaluate the performance of the systems. Each metric reflects a specific quality of the system, such as the ability of the system to classify pixels correctly (Accuracy), the alignment of the predicted pixels with the gTruth ones (IoU), and the alignment of the predicted object boundaries with the gtruth boundaries (BF score). Table 6.3 shows the detailed results of the trained systems. The proposed DLV3+ with ResNet-18 systems have achieved mIoU of 0.572/0.696 and mean BF scores of 0.673/0.772 for the indoor and outdoor datasets.

Table 6.3 **Results of running the trained individual models on the test set of the indoor and the outdoor datasets.**

Metrics Indoor/Outdoor **Model**	**Global Accuracy**	**Mean Accuracy**	**Mean IoU**	**Weighted IoU**	**Mean BF Score**
$FCN-8s$	0.963/0.808	0.801/0.771	0.552/0.518	0.953/0.732	0.661/0.621
$FCN-16s$	0.961/0.845	0.785/0.836	0.549/0.600	0.952/0.783	0.652/0.684
$FCN-32s$	0.953/0.813	0.766/0.775	0.538/0.523	0.944/0.740	0.583/0.619
$SegNet\ (VGG-16)$	0.960/0.697	0.804/0.680	0.551/0.453	0.950/0.609	0.658/0.451
$SegNet\ (VGG-19)$	0.956/0.783	0.796/0.755	0.528/0.499	0.946/0.686	0.657/0.501
$U-Net$	0.807/0.535	0.505/0.359	0.314/0.207	0.717/0.418	0.358/0.323
$DLV3+\ (ResNet-18)$	0.970/0.915	0.791/0.874	0.572/0.696	0.963/0.860	0.673/0.772
$DLV3+\ (ResNet-50)$	0.965/0.934	0.788/0.906	0.562/0.748	0.957/0.889	0.622/0.825
$DLV3+\ (Xception)$	0.966/0.911	0.808/0.883	0.560/0.692	0.958/0.856	0.621/0.769

Both indoor and outdoor DLV3+ with ResNet-18 systems have achieved high global and mean accuracy (0.970/0.915 and 0.791/0.874 for the indoor and the outdoor datasets, respectively). Global accuracy is the ratio between correctly classified pixels, regardless of the class, to the total number of pixels. In comparison, mean accuracy represents the correctly classified pixels for each class averaged over all classes.

To ensure the reproducibility of our results, we trained both the indoor and the outdoor systems three times. Images are shuffled and randomly split to guarantee that different images are used for training and testing at each time. Table 6.4 shows the mean and the standard deviation of systems' metrics. It can be seen that the proposed systems are robust and can reproduce the results under different conditions.

Table 6.4 **Mean and standard deviation of three trained models on the indoor and the outdoor datasets.**

Model / Metrics	DLV3+ (ResNet-18) indoor	DLV3+ (ResNet-18) outdoor
Global Accuracy	0.970 ± 0.003	0.919 ± 0.004
Mean Accuracy	0.799 ± 0.007	0.888 ± 0.013
Mean IoU	0.570 ± 0.016	0.703 ± 0.008
Weighted IoU	0.963 ± 0.004	0.868 ± 0.007
Mean BF Score	0.680 ± 0.024	0.781 ± 0.010

The detailed results for each class of the indoor dataset are shown in Table 6.5. It can be observed that objects with bigger sizes and larger numbers of pixels have achieved the highest IoU and BF scores, such as doors and background walls, while smaller objects have achieved the lowest IoU, such as pull and push door handles. This is understandable due to the few instances and pixels per object for small size objects in the proposed indoor dataset. Besides, it is challenging for any tested systems to align the predicted segments with the ground truth ones, reflected by the IoU metric, as these objects are tiny (for example, DLV3+ with ResNet-50 has achieved 0.102 IoU for the push door handle class (Table A.7). Detailed results for different models are shown in the appendix (Semantic Segmentation Supplementary Material). However, small size objects have achieved satisfactory accuracy and BF score. An adequate BF score is vital to our application as it reflects the system's ability to define object boundaries effectively. This is very important for visually impaired users and human-system interaction (Fig A.1).

Table 6.5 **Per-class metrics of the indoor system using DLV3+ with ResNet-18 on the test set.**

Metrics / Classes	Accuracy	IoU	Mean BF Score
Door	0.983	0.983	0.870
Pull Door Handle	0.593	0.150	0.593
Push Button	0.790	0.338	0.571
Moveable Door Handle	0.786	0.665	0.543
Push Door Handle	0.533	0.090	0.341
Fire Extinguisher	0.909	0.889	0.650
Key Slot	0.654	0.186	0.488
Carpet Floor	0.901	0.889	0.751
Background Wall	0.967	0.962	0.778

The outdoor system has achieved similar results (Table 6.6) to the indoor one as small size objects such as pole has achieved the lowest IoU. Whereas medium and large size objects have achieved better IoU and BF scores.

The three-stream model (FCN-8s), which adds two skip connections at layers pool3 and pool4, has achieved better overall results compared to FCN-16s, which add one skip connection at pool4 layer, and the series version of FCN (FCN-32s). In contrast, the deeper version of SegNet (SegNet with VGG-19) is not as accurate as the shallower version (SegNet with VGG-16), similar to DLV3+ with ResNet-18 that can achieve better performance compared to the deeper version (DLV3+ with ResNet-50). It can be concluded that deeper versions of semantic segmentation models do not ensure high performance. U-Net performance is the lowest amongst the tested systems (Table 6.3).

The achieved FPS for DLV3+ with ResNet-18 is better than that of DLV3+ with ResNet-50 and with Xception base networks (Table 6.7). The accuracy and speed can be enhanced further by increasing the number of small object instances in the proposed dataset and using a newer version of a GPU based board such as the Jetson AGX Xavier board. Also, the proposed

Table 6.6 **Per-class metrics of the outdoor system using DLV3+ with ResNet-18 on the test set.**

Metrics / Classes	Accuracy	IoU	Mean BF Score
Sky	0.958	0.937	0.932
Building	0.859	0.835	0.745
Pole	0.765	0.275	0.680
Road	0.952	0.939	0.934
Pavement	0.920	0.783	0.837
Tree	0.919	0.823	0.842
SignSymbol	0.722	0.432	0.592
Fence	0.810	0.624	0.709
Car	0.931	0.820	0.799
Pedestrian	0.873	0.483	0.640
Bicyclist	0.909	0.699	0.732

shared systems have achieved adequate speed. The most accurate shared system (shared system 3) has achieved the lowest speed among the proposed ones. However, the lowest accurate shared system (shared system 1) has achieved the highest speed amongst the proposed shared systems. Interestingly, the proposed shared systems have achieved higher FPS than state-of-the-art systems such as FCN, SegNet and U-Net. Although the proposed shared systems have more layers, they have less trainable parameters and smaller footprints. Moreover, they utilise residual blocks, which can explain their fast inference speed.

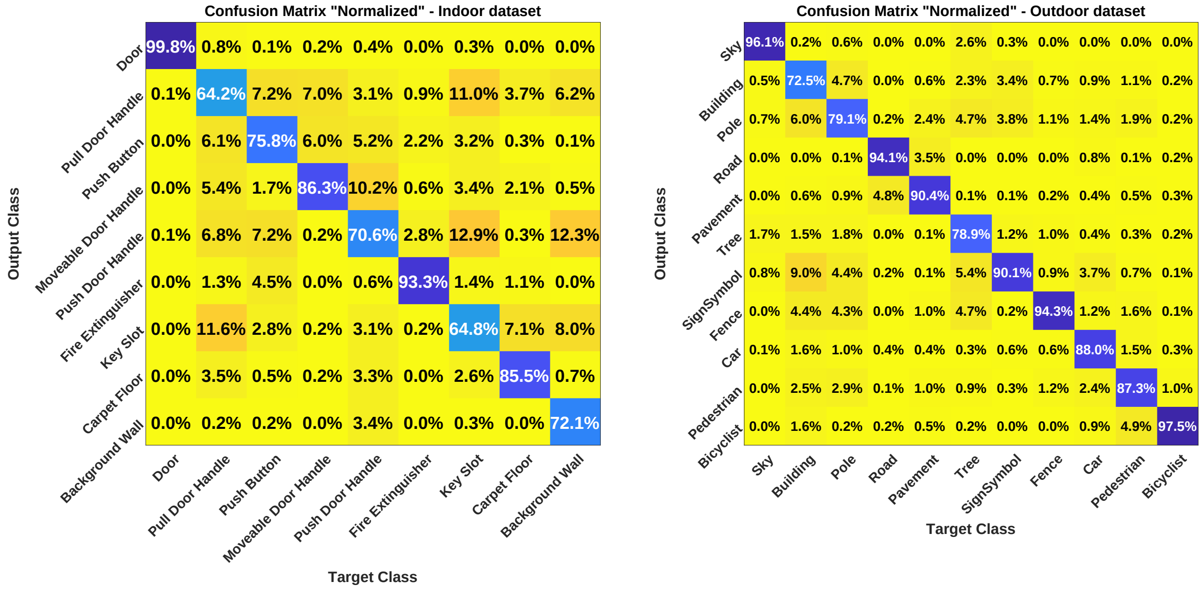
Similar observations can be extracted from the confusion matrices shown in Fig 6.4. It can be seen that the indoor model is slightly confused to distinguish between pixels of different door handles and key slot. The analogous silver colour and orientation of the door handles can represent the reason for that problem. This can be alleviated by increasing these object instances in the proposed dataset. There is a slight confusion between the sign symbol and the pole classes for the outdoor confusion matrix, which can be attributed to the similarity of the objects' structures.

Fig 6.5 and Fig 6.6 show some examples of the indoor and the outdoor systems in action where it can segment the scenery with good accuracy and sharp edges. Three rows of images

Table 6.7 **Average speed of the tested models in FPS when deployed on a Jeston TX2 GPU based board.**

Model	Speed in FPS
$FCN-8s$	0.86
$FCN-16s$	0.86
$FCN-32s$	0.86
$SegNet\ (VGG-16)$	0.89
$SegNet\ (VGG-19)$	0.72
$U-Net$	0.75
$DLV3+(ResNet-18)$	2.65
$DLV3+(ResNet-50)$	1.57
$DLV3+(Xception)$	2.00
$Sharedsystem1$	1.49
$Sharedsystem2$	1.30
$Sharedsystem3$	1.16

Systems are tested on a never seen before prerecorded video of the indoor environment (from the same distribution of the indoor dataset used for training) and on the CamVid video. TensorRT has been used to optimize systems' inference. The performance of FCN, SegNet, and U-Net is far from real-time execution.



	Door	Pull Door Handle	Push Button	Moveable Door Handle	Push Door Handle	Fire Extinguisher	Key Slot	Carpet Floor	Background Wall
Door	99.8%	0.8%	0.1%	0.2%	0.4%	0.0%	0.3%	0.0%	0.0%
Pull Door Handle	0.1%	64.2%	7.2%	7.0%	3.1%	0.9%	11.0%	3.7%	6.2%
Push Button	0.0%	6.1%	75.8%	6.0%	5.2%	2.2%	3.2%	0.3%	0.1%
Moveable Door Handle	0.0%	5.4%	1.7%	86.3%	10.2%	0.6%	3.4%	2.1%	0.5%
Push Door Handle	0.1%	6.8%	7.2%	0.2%	70.6%	2.8%	12.9%	0.3%	12.3%
Fire Extinguisher	0.0%	1.3%	4.5%	0.0%	0.6%	93.3%	1.4%	1.1%	0.0%
Key Slot	0.0%	11.6%	2.8%	0.2%	3.1%	0.2%	64.8%	7.1%	8.0%
Carpet Floor	0.0%	3.5%	0.5%	0.2%	3.3%	0.0%	2.6%	85.5%	0.7%
Background Wall	0.0%	0.2%	0.2%	0.0%	3.4%	0.0%	0.3%	0.0%	72.1%

	Sky	Building	Pole	Road	Pavement	Tree	SignSymbol	Fence	Car	Pedestrian	Bicyclist
Sky	96.1%	0.2%	0.6%	0.0%	0.0%	2.6%	0.3%	0.0%	0.0%	0.0%	0.0%
Building	0.5%	72.5%	4.7%	0.0%	0.6%	2.3%	3.4%	0.7%	0.9%	1.1%	0.2%
Pole	0.7%	6.0%	79.1%	0.2%	2.4%	4.7%	3.8%	1.1%	1.4%	1.9%	0.2%
Road	0.0%	0.0%	0.1%	94.1%	3.5%	0.0%	0.0%	0.0%	0.8%	0.1%	0.2%
Pavement	0.0%	0.6%	0.9%	4.8%	90.4%	0.1%	0.1%	0.2%	0.4%	0.5%	0.3%
Tree	1.7%	1.5%	1.8%	0.0%	0.1%	78.9%	1.2%	1.0%	0.4%	0.3%	0.2%
SignSymbol	0.8%	9.0%	4.4%	0.2%	0.1%	5.4%	90.1%	0.9%	3.7%	0.7%	0.1%
Fence	0.0%	4.4%	4.3%	0.0%	1.0%	4.7%	0.2%	94.3%	1.2%	1.6%	0.1%
Car	0.1%	1.6%	1.0%	0.4%	0.4%	0.3%	0.6%	0.6%	88.0%	1.5%	0.3%
Pedestrian	0.0%	2.5%	2.9%	0.1%	1.0%	0.9%	0.3%	1.2%	2.4%	87.3%	1.0%
Bicyclist	0.0%	1.6%	0.2%	0.2%	0.5%	0.2%	0.0%	0.0%	0.9%	4.9%	97.5%

Fig. 6.4 **Confusion matrices for the indoor and the outdoor systems.**

are shown where the first row represents the ground truth data, the second one shows the model's prediction, and the third one demonstrates the difference between the prediction and the ground truth data. The intense green and magenta colours that are shown in the third row indicate these differences. These pixels are unannotated or misclassified. The green colour shows the unannotated pixels which do not belong to objects of interest. Whereas the magenta one shows the misclassification of some parts of an object.

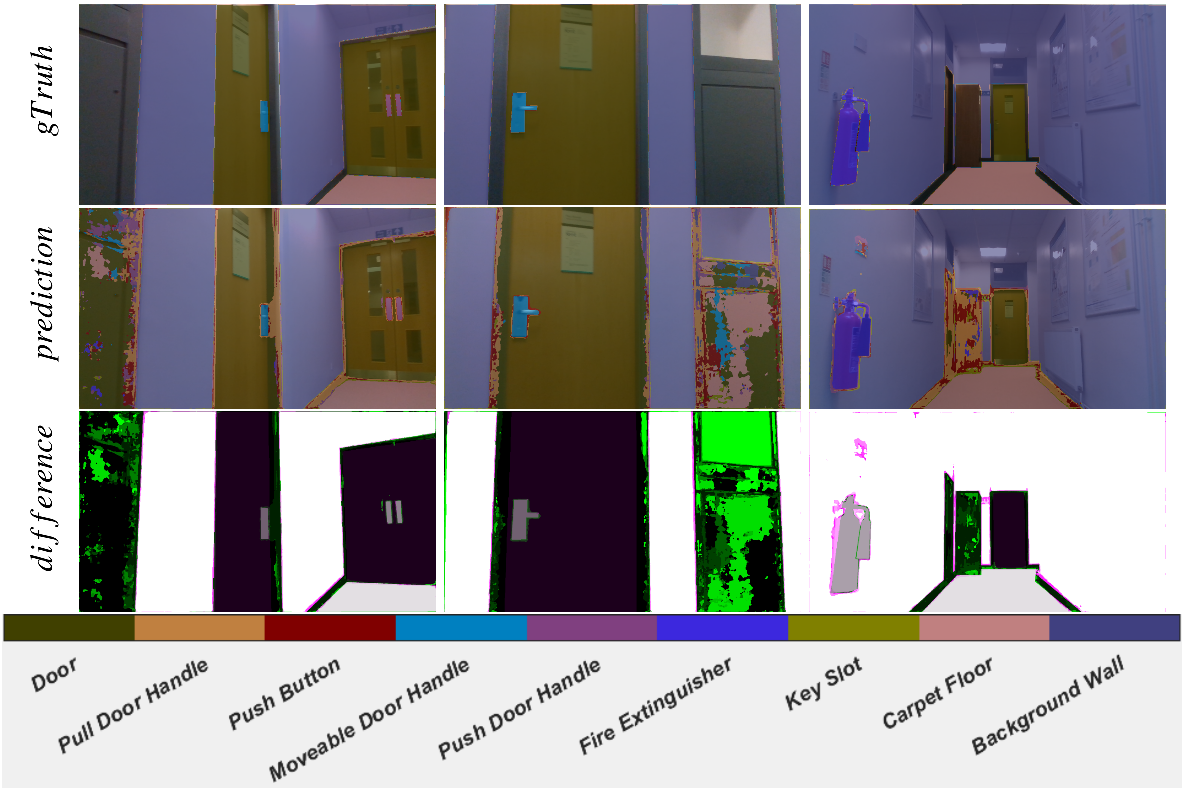


Fig. 6.5 **Results visualisation using the proposed indoor system on the test set.** The first row represents the ground truth data, the second row represents the system's output and the third row represents the difference between the ground truth and the prediction.

It can be seen from Fig 6.5 that the unannotated pixels in-between two annotated objects, which do not belong to either object, can represent a challenge to the proposed network. For instance, the indoor system struggles to classify door frame pixels as they do not belong to the door or the wall. Moreover, they are not annotated in the proposed dataset, which represent a challenge during inference.

One solution is to annotate door frames as a separate class. Training a semantic segmentation system on a dataset with some unannotated pixels increases the system's uncertainty. However,

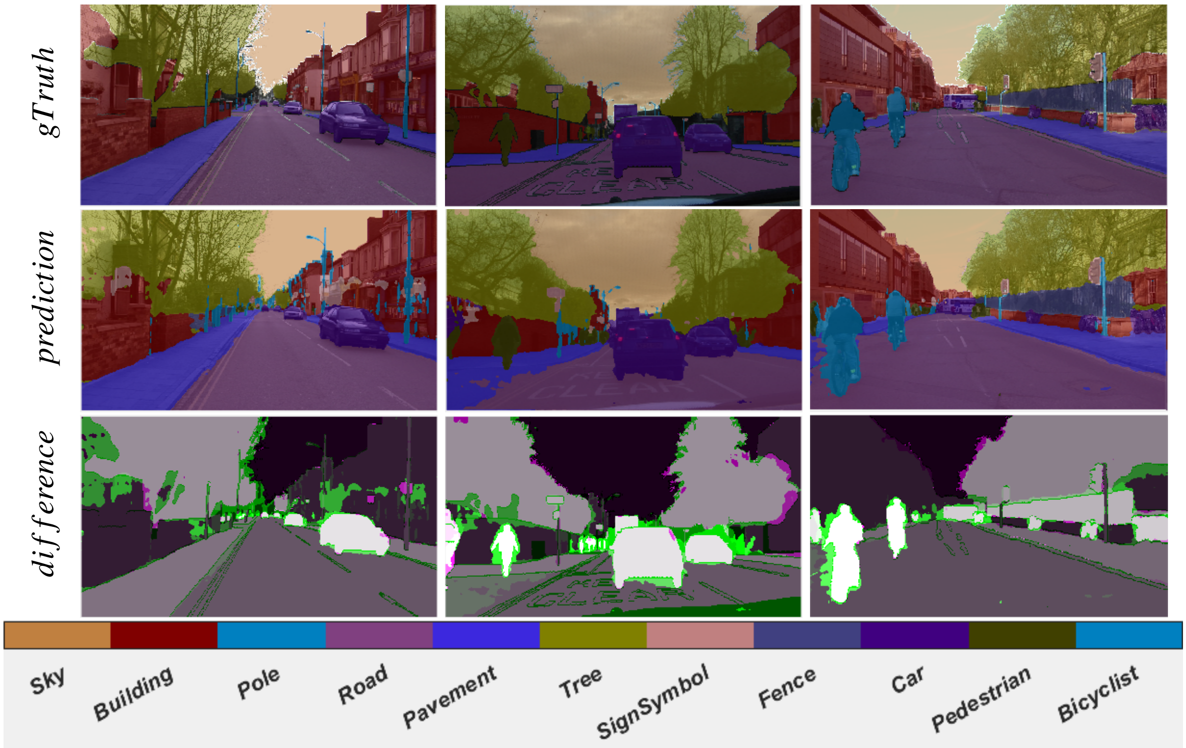


Fig. 6.6 **Results visualisation using the proposed outdoor system on the test set.** The first row represents the ground truth data, the second row represents the system's output and the third row represents the difference between the ground truth and the prediction.

annotating every pixel, even if it does not belong to any class of interest, reduces the system's uncertainty because they act as a false positive for the objects of interest. This can be done by annotating all the pixels in an image. This can increase the overall system accuracy and enhance the detection of the object's boundaries. However, the process of annotating every pixel is extremely labouring intense.

Another solution is to ignore the predicted pixels below a predefined threshold [158]. As the systems need to assign each pixel in an image to one of the predefined classes, unannotated pixels, which belong to classes of non-interest, will be assigned to one of the predefined classes. Comparing predicted pixels with the unannotated ones of the ground truth data can result in inaccurate metrics. Usually, these predicted pixels have low confidence scores. We propose to assign the predicted pixels below a specific threshold to a 'Reject' class [158]. Consequently, they can not be included in the evaluation process, resulting in quantitatively and qualitatively accurate predictions.

Fig 6.7 shows the qualitative segmentation comparison between the proposed and state-of-the-art systems. FCN-32s is the series version of FCN with an up-sampling stride of 32 and no skip connections. It is demonstrated that DLV3+ can define object boundaries better than FCN. At the same time, the segmentation of FCN can be seen as patches with fuzzy boundaries. For example, it is challenging to distinguish the moveable door handle grip from the body in the segmentation of FCN. Similarly, U-Net could not predict all the pixels correctly, especially small objects such as door handles. Although the predicted boundaries of SegNet are well defined, there are high uncertainties in the pixels around the correctly predicted ones.

On the other hand, the grip in the DLV3+ segmentation is well defined, which facilitates further interactions if needed, such as manipulating the door handle using a robotic arm. Table 6.3 emphasises the qualitative assessment. Compared to state-of-the-art systems, the proposed DLV3+ models have better mIoU and mean BF scores (contour matching score).

Shared systems 1, 2, and 3 have achieved adequate performance but are not as good as the individual ones (Table 6.8). For shared system 1 (Fig 6.3a), when the encoder of the indoor semantic segmentation system is used, the system has achieved a mean accuracy of 0.676 and 0.456 on the indoor and outdoor datasets, respectively. Also, it has achieved mIoU of 0.591 and 0.300 on the indoor and the outdoor datasets, respectively. Whereas when the encoder of the outdoor semantic segmentation system is used, the system has achieved a mean accuracy of 0.185 and 0.852 on the indoor and the outdoor datasets, respectively. Also, it has achieved mIoU of 0.182 and 0.689 on the indoor and the outdoor datasets, respectively.

Results show that the used encoder has a direct impact on the overall system performance. The encoder of shared system 1, which has been trained on the indoor dataset, can produce better results on the indoor images compared to the outdoor ones and vice versa. This indicates the bias of shared system 1 to the used encoder.

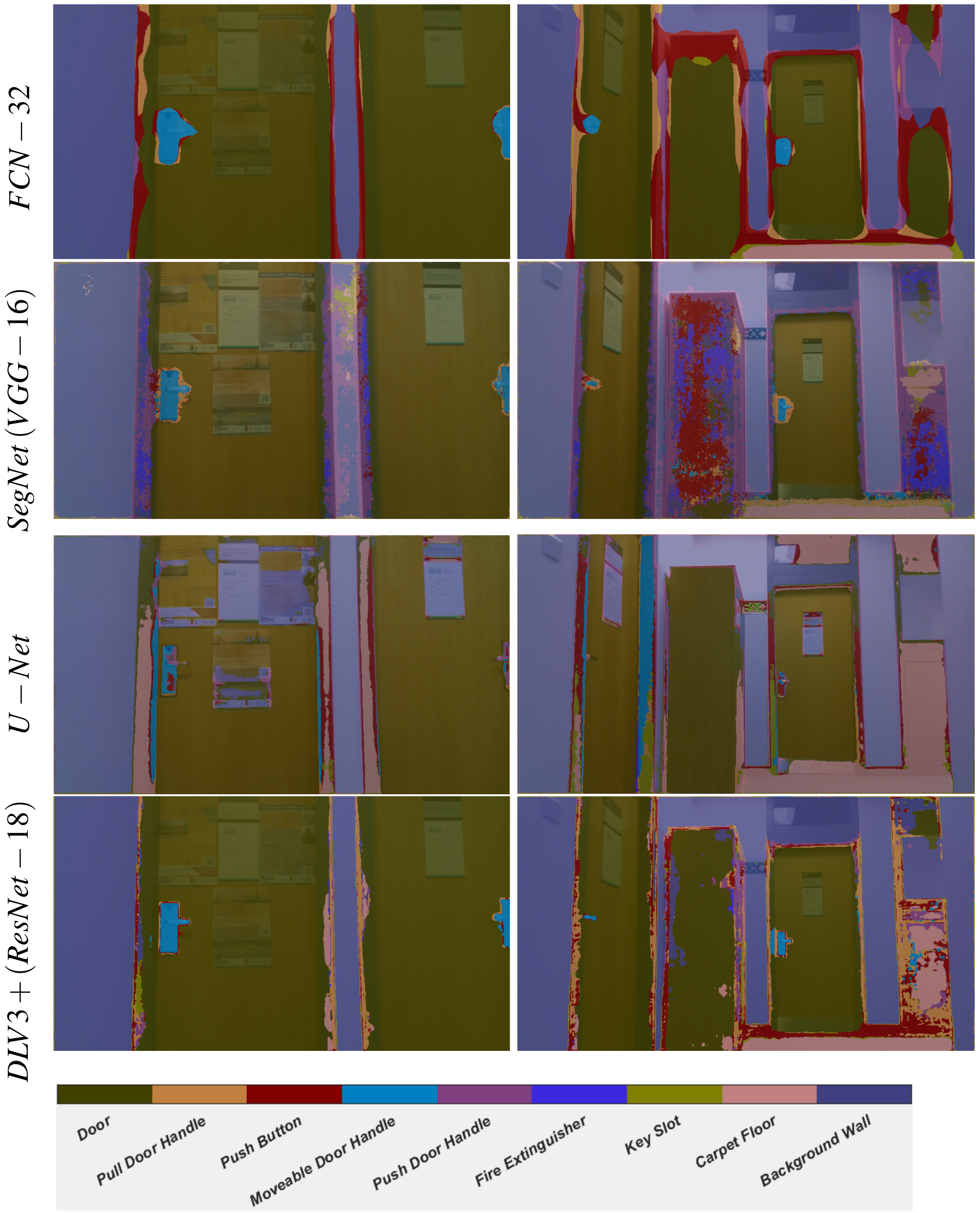


Fig. 6.7 **Qualitative comparison between the proposed indoor system based on DLV3+ and state-of-the-art systems.**

Shared system 2 (Fig 6.3b) has achieved mean accuracy and mIoU of 0.594 and 0.555 on the indoor dataset. Whereas it has achieved mean accuracy and mIoU of 0.830 and 0.657 on the outdoor dataset. The performance of shared system 2 is acceptable. However, the individual

Table 6.8 **Shared systems 1 and 2 detailed metrics.**

Metrics Indoor/Outdoor **Model**	**Global Accuracy**	**Mean Accuracy**	**Mean IoU**	**Weighted IoU**	**Mean BF Score**
Shared system 1 (indoor encoder)	0.920/0.582	0.676/0.456	0.591/0.300	0.915/0.506	0.601/0.360
Shared system 1 (outdoor encoder)	0.384/0.892	0.185/0.852	0.182/0.689	0.376/0.845	0.365/0.659
Shared system 2	0.725/0.838	0.594/0.830	0.555/0.657	0.725/0.790	0.540/0.608

systems produce better results. Detailed results of the shared systems are shown in Table 6.8, where both shared systems 1 and 2 have achieved acceptable Mean BF scores.

As shared system 3 (Fig 6.3c) propagates the images through both the individual indoor and outdoor semantic segmentation systems, the shared system's metrics are similar to the individual ones, which are the best-achieved metrics in terms of accuracy, IoU and BF score. However, as shared system 3 compares the pixels' scores of the individual systems (post-processing step), the annotated output image is dependant on that comparison. Table 6.9 shows the ability of the system to classify the input images as indoor or outdoor depending on the pixels' confidence scores using different comparison techniques.

To test the ability of the system to correctly classify the input images as indoor or outdoor ones, we propagate the indoor and the outdoor test sets images through the system. Shared system 3 is able to classify all of the images correctly using Max(Mean(score)) comparison technique described in the system architecture subsection. To obtain more robust results, we shuffled the indoor and the outdoor datasets. Then, the mixed dataset is split randomly into 70% for training, 15% for validation, and 15% for testing. This results in a mix (In+Out) test set with 337 images (232 indoor images and 105 outdoor images). Shared system 3 miss-classified 11 images form the (In+Out) test set as outdoor ones using the Max(Mean(score)) comparison technique (Table 6.9).

The system's inference speed is dependant on many factors such as the number of trainable parameters, the system's footprint and whether any post-processing techniques are applied.

Table 6.9 **Classification capabilities of shared system 3 using different techniques for comparing pixels' scores.**

	Dataset	**Classification**	
Method		**Indoor**	**Outdoor**
Pixel by pixel	Indoor (233)	216	17
	Outdoor (105)	Zero	105
	In+Out (232+105)	206	131
Max(Mean(score))	Indoor (233)	233	Zero
	Outdoor (105)	Zero	105
	In+Out (232+105)	221	116

Table 6.7 shows the speed of the proposed shared systems. Shared system 1 has fewer layers and footprint (Table 6.1) compared to Shared systems 2 and 3. Consequently, it has achieved the fastest inference speed among the proposed shared systems with 1.49 FPS. Shared system 3 is the slowest with 1.16 FPS. It has the largest footprint and a post-processing step. However, the proposed shared systems' inference speeds are higher than FCN, SegNet and U-Net systems.

Choosing the right system for the right application is a trade-off process between accuracy, inference speed, and the application domain. The deployment of the proposed indoor system can be seen in Fig 6.8. The user is controlling the EPW while the information is being displayed on the screen.

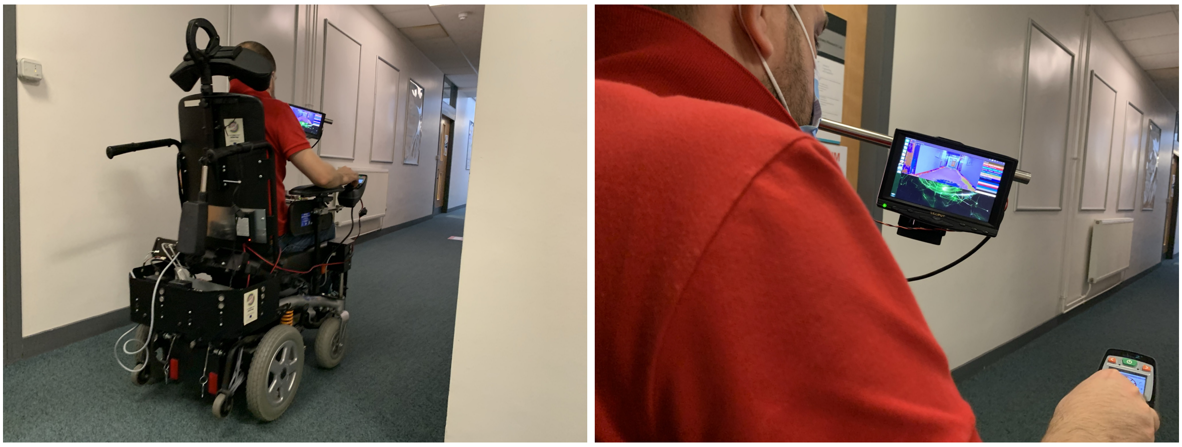


Fig. 6.8 **System deployment.** The proposed systems are deployed on an EPW with a display, a Nvidia Jetson TX2 board, and a depth camera.

6.4 Limitations of the systems

In this section, the limitations of the study and means of mitigation are discussed. Model choice is dependant on the application. The system's speed and accuracy are the main concerns of this application. More precisely, the ability of the system to clearly define objects boundaries. It is a challenging task to develop a model that can achieve significant accuracy with high inference speed. Tolerating high inference rates is acceptable as disabled users do not drive fast due to the speed limitation of the EPW. Consequently, the performance of the proposed system is adequate for the application.

One of the major problems facing semantic segmentation tasks is the ability of the systems to process data from two different distributions. The proposed shared systems offer solutions for this problem by merging the learned features of the two models (the indoor and the outdoor systems). However, solving the multi-context data processing issue has negatively impacted the system's speed and accuracy. Thus, the application should determine its needs and compromises to employ the best model for a given application.

Unannotated pixels of the ground truth data represent a challenge for the proposed semantic segmentation systems. As the systems need to assign each pixel in an image to one of the predefined classes, unannotated pixels which belong to classes of non-interest will be assigned to one of the predefined classes. Comparing predicted pixels with the unannotated ones of the ground truth data can result in inaccurate metrics. Usually, these predicted pixels have low confidence scores. We propose to assign the predicted pixels below a specific threshold to a 'Reject' class [158]. Consequently, they can not be included in the evaluation process, resulting in quantitatively and qualitatively accurate predictions.

The future work of the research will concentrate on expanding the proposed dataset, especially small size objects, which can positively impact the overall system accuracy. Besides, investigating different shared system architectures that can process multi-context data at high inference speed.

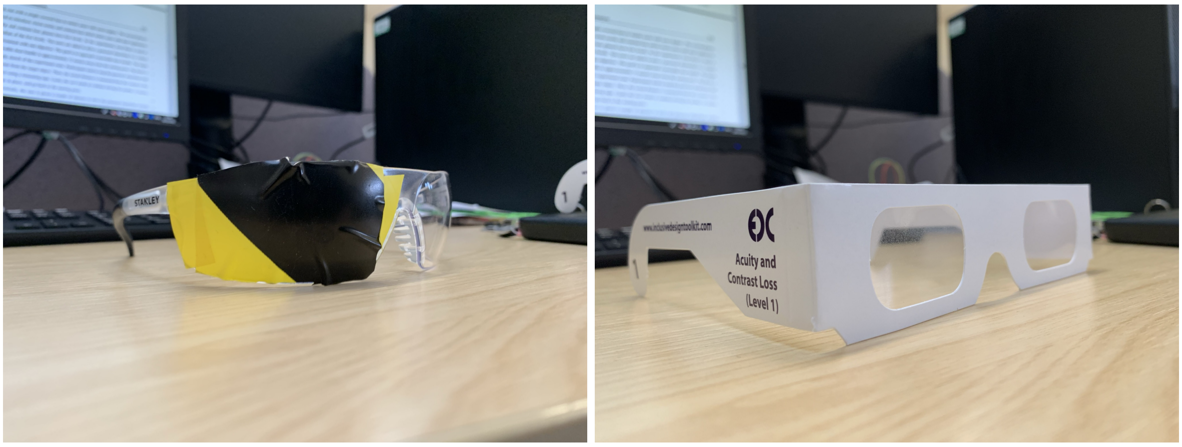
6.5 Practical implementation

6.5.1 Experiment setup

An experiment has been conducted to evaluate the proposed SS systems with healthy users. Visual impairment is simulated using two glasses (Fig. 6.9) to mimic typical use-case scenarios. The first pair of glasses has a single covered lens to simulate users with semi-neglect (Fig. 6.9a). The second pair of glasses are used to introduce vision acuity and contrast loss which simulate users with short-sightedness (Fig. 6.9b).The Cambridge simulation glasses[1] used in the experiments provide insights into the effects of vision loss on product use [159, 160]. Also, the glasses can be used to examine the visual accessibility of products and services. Acuity and contrast loss glasses have five levels. In the experiments, level three is used as a median of the five levels. Ten users are asked to drive a powered wheelchair in a controlled environment with one objective. The goal is to approach a moveable door handle. The target moveable door handle is approximately 12 meters away from the starting point. Fig 6.10 shows a simple sketch of the experiment route. Participants need to stop the powered wheelchair half a meter from the target object. Then, the actual distance is measured from the camera to the object using a measuring tape. Lastly, the users need to continue driving for another 12 meters, rotate in place, and return to the starting point.

Initially, the user is given a couple of minutes to try the powered wheelchair without any objective to familiarise and understand the controlling process. Then, each user needs to repeat the experiment five times. First, users need to approach the target object with their bare eyes. Second, users need to approach the target object using the one covered lens glasses with and without the SS system. Third, the user needs to use the acuity and contrast loss glasses to approach the target with and without the SS system. The semantic segmentation system is used in the second and third trials, where the user is asked to drive the powered wheelchair to the

[1] http://www.inclusivedesigntoolkit.com/csg/csg.html



(a) Glasses to simulate semi-neglect condition. (b) Glasses to simulate short-sightedness condition.

Fig. 6.9 **Visual impairment simulation.**

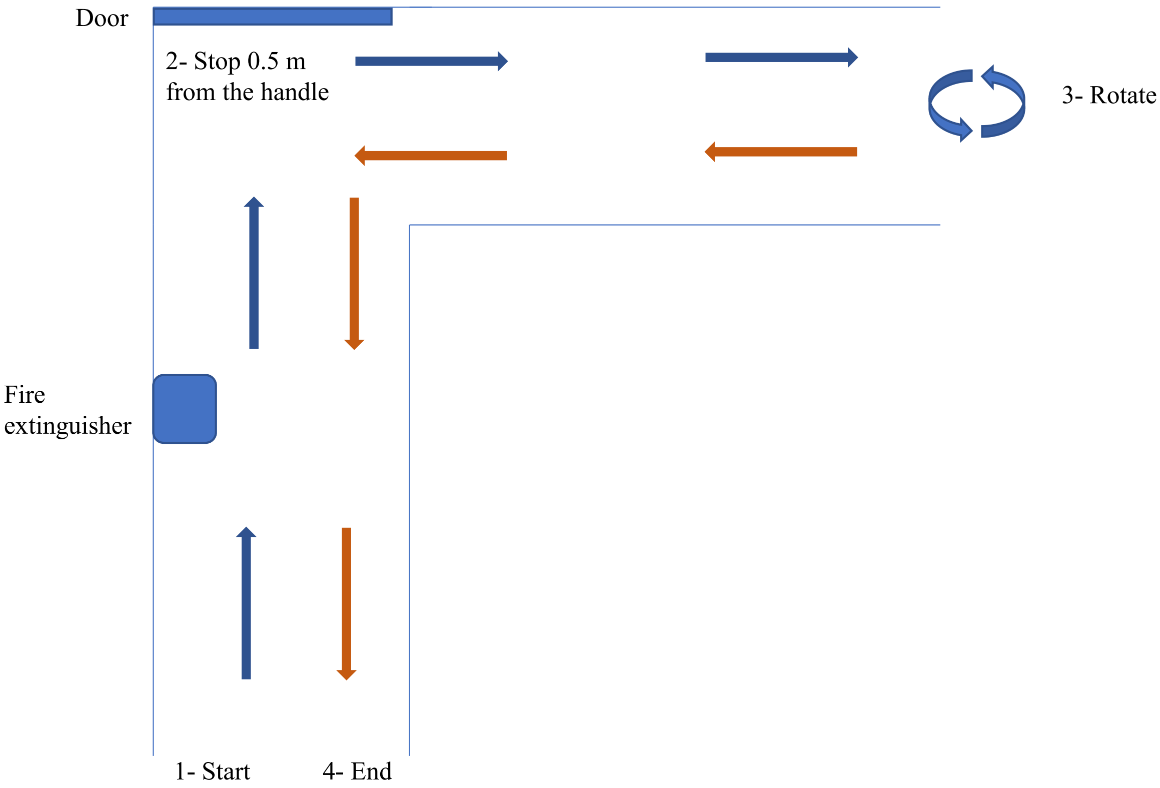
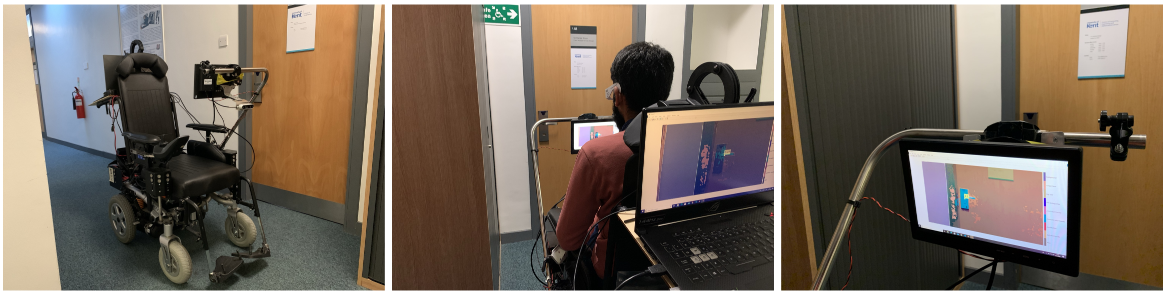


Fig. 6.10 **Experiment route.**

target object depending only on the display. Fig 6.11 shows the system setup used in the second and third trials. The display and the SS systems are not used in the first trial. The display is customised to the user's need, meaning if the user is using the one covered lens to simulate semi-neglect disorder on the right side of the environment, the display is fitted on the user's left side. The display is placed 30 *cm* from the user's face at all times. The display shows the indoor environment classified into objects on the pixel level, where the boundaries of the objects are well-defined. Furthermore, the system shows the distance to the target object (Fig 6.11).



(a) System installed on the EPW. (b) System in use. (c) Distance to the target object.

Fig. 6.11 **Practical implementation of the SS system.**

Users' information, such as user ID, age, powered wheelchair driving experience, and use of glasses, are recorded prior to the experiments. During the experiments, trial time, the actual distance to the target object, and the number of collisions are recorded. Users' details are shown in Table 6.10.

Table 6.10 **Users' details.**

User ID	**Age**	**Gender**	**Height** (cm)	**Weight** (Kg)	**Use of glasses**	**Experience***
1	40	M	170	95	Yes	1
2	22	F	160	48	No	1
3	24	F	170	70	No	1
4	23	F	164	52	No	1
5	30	M	178	75	No	4
6	32	M	175	100	No	1
7	29	M	178	61	No	1
8	26	M	180	80	Yes	1
9	32	M	180	78	No	1
10	28	M	187	115	Yes	2

* EPW driving experience on a scale from 1 to 5, where 1 represents the least experience.

After the experiments, users are asked three questions: Did you feel any stress or screen fatigue? Does the semantic segmentation system help in the navigation process, and How? Lastly, if they have any comments or concerns?

6.5.2 Results and discussion

Table 6.11 shows the experiment results for all users. User 3 could not participate in the second experiment. It can be seen that the completion time of all experiments under different conditions for all users is very close, which means that the semantic segmentation system does not slow down the navigation process. There is a slight increase in collisions while simulating visual impairment and using the SS system. All the collisions happen when the users rotate before going back to the starting point. The slight increase in the total number of collisions while simulating visual impairment conditions (4 and 3 collisions for semi-neglect and short-sightedness, respectively, with the SS system compared to 2 and 3 collisions without the SS system) can be attributed to the constraint view shown on the display. While a normal user can rotate his head to check all the sides of the powered wheelchair, it is challenging to check the sides and the back of the powered wheelchair from the display. Also, all users except user 5 have limited experience in driving powered wheelchairs.

Interestingly, the SS system has helped the users to approach the target object accurately. While simulating the semi-neglect condition, users could approach the target object with a mean distance of 59.9 *cm* compared to 73.9 *cm* without the SS system. For the short-sightedness case, the mean distance using the SS system is 59.2 *cm* compared to 52.8 *cm* without the SS system. The estimated distance of the short-sightedness case without the SS system is better than that with the SS system and even better than the estimated distance in the normal case. It was observed that the users try to approach the object until it becomes distinguishable with the acuity and contrast loss glasses, which results in better distance estimation.

Table 6.11 **Experiment results.**

User ID	**Normal**			**Semi-neglect**						**Short-sightedness**					
				Without SS system			**With SS system**			**Without SS system**			**With SS system**		
	T	C	D	T	C	D	T	C	D	T	C	D	T	C	D
1	2:13	1	80	2:00	1	64	2:24	zero	53	2:12	3	62	2:18	1	60
2	2:13	zero	60	2:10	zero	75	2:27	1	50	1:58	zero	41	1:53	zero	28
3	2:08	1	80	1:54	zero	75	2:11	1	45	-	-	-	-	-	-
4	2:05	zero	90	1:54	zero	92	2:13	zero	73	1:49	zero	65	1:52	zero	75
5	1:45	zero	75	1:42	zero	70	1:44	zero	66	1:45	zero	62	1:39	zero	55
6	2:25	zero	88	2:07	zero	103	2:21	1	80	1:52	zero	83	2:03	1	64
7	2:00	zero	41	1:49	zero	62	1:57	zero	60	1:44	zero	46	1:49	zero	66
8	2:01	zero	92	2:05	zero	67	2:14	zero	42	1:42	zero	33	1:51	zero	60
9	1:50	zero	81	1:45	zero	70	2:00	zero	55	1:51	zero	36	1:46	1	60
10	1:47	zero	77	1:59	1	61	1:56	1	75	1:41	zero	48	1:51	zero	65
Mean	2:02	0.2	76.4	1:56	0.2	73.9	2:08	0.4	59.9	1:52	0.3	52.8	1:56	0.33	59.2
Std	± 12	0.42	15.3	± 9	0.42	13.5	± 13	0.51	13.1	± 11	1	16.2	± 12	0.5	12.9

T : Time $(min : sec)$
C : No# of collisions
D : Actual distance to the target (cm)
Std: Standard deviation in seconds for time measurement.

The consistency in the results of the SS system is an indication of the system's robustness. In addition, the SS system results are significantly better than the normal case without any glasses to simulate visual impairment, which indicates the system's efficiency in estimating the accurate distance to the target object.

The post-experiment interviews reveal some interesting observations:

Simulation of semi-neglect condition

- 50% of the users felt some level of stress when they use the one-eye covered glasses, which is normal as they are not used to that situation. As a result, they could not estimate the distance correctly without the SS system. One-eye covered glasses have also introduced limited vision.

- 80% of the users found the SS system with the display helpful in estimating the distance because the system displayed the distance to the object in real-time and highlighted the target object (semantically segmenting the target object). In addition, the system helped the users to see the whole scene on the display, especially the blind spots due to the usage of the one-eye covered glasses.

- User 8 and 9 suggested to use a high-resolution display and a faster processing rate in terms of FPS to achieve fast estimations.
- User 2 suggested that the limited vision shown on the display in tight places may result in collisions.
- Without the SS system, user 10 was worried about hitting any object on the covered side by the glasses. However, the SS system greatly helps by showing these blind spots and highlighting the components of the environment in different colours. In contrast, user 9 suggests highlighting only the target object and not all the environmental cues.

Simulation of short-sightedness condition

- 40% of the users felt stress because of the blurred and foggy vision due to the usage of the acuity and contrast loss glasses.
- 8 out of 10 users found the system helpful in understanding the surroundings and estimating the distance to the target object. One user said the experience was easy with and without the system because the user became familiar with the path (User 4). In contrast, user 9 said the system was not helpful in distance estimation because the refresh rate is slow but was helpful for segmentation.
- 6 out of 10 users found the display hard to use while turning due to the limited vision of the sides and back of the powered wheelchair.
- User 1 found the system very helpful in understanding the surroundings. In addition, the real-time update of the distance to the target object is advantageous. With the blurred vision simulation, the user can recognise the door but not the handle, the wall but not the fire extinguisher. However, the SS system can give accurate information and allow the recognition of the door handle and the fire extinguisher.

- Three users highlighted that the distance estimation would disappear from the display if they closely approach the target object. This is understandable because the camera has a theoretical minimum depth of 28 *cm*.

It can be concluded from the quantitative and qualitative results that the proposed system can help visually impaired users to estimate the distance to the target object accurately. Moreover, it can help the users to understand the surrounding environment. However, the system's display limits the vision of the powered wheelchair sides in narrow areas. This can be compensated using proximity sensors to alert the user when turning. Also, increasing the system's speed in terms of FPS can further enhance the system's overall performance.

6.6 Vibration impact on computer vision systems

Rough terrains can negatively impact the performance of smart computer vision systems, which may result in inaccurate human-system interaction. Mobile robots and smart vehicles are susceptible to mechanical vibration due to traversing rough and uneven terrains. The ability to estimate the impact of vibration on the system performance is the first step to mitigating undesirable vibrations. This can enhance the system performance in challenging conditions; consequently, better human-system interaction can be attained.

Marichal et al. [161] investigated the impact of vibrations produced by a helicopter on a vision system. It is concluded that the quality of the captured images can be negatively impacted due to the camera's undesirable movement. The proposed semi-active frequency isolation technique has proved its efficiency in improving the captured images with low vibrations. Consequently, the subsequent utilisation of the captured images can be enhanced. However, the proposed technique needs prior knowledge of the vibration frequency in order for the system to be able to isolate it.

Periu et al. [162] studied the impact of the vibrations on the performance of obstacle detection using a LIght Detection And Ranging (LIDAR) sensor. The LIDAR sensor is installed on a tractor for obstacle detection and guidance purposes. Generally, agriculture vehicles do not have a suspension system, similar to the EPW used in our experiments. The measurement of the LIDAR sensor can be significantly impacted by mechanical vibrations induced during the vehicle's operation on rough and bumpy terrains [162]. The study proposes supporting bars and stabilising systems to counteract the vibrations impact. It is concluded that with the increase in the tractor speed, the accuracy of the LIDAR decreases due to the high level of mechanical vibrations; consequently, the position estimation error increase. Thus, the mean error distance and the standard deviation between the actual and the detected position increase.

This section investigates the impact of vibrations on a smart semantic segmentation system used by visually impaired EPW users to understand their surroundings by providing environmental cues. Environmental cues can help visually impaired users to locate objects in their surroundings. It also investigates the vibration impact on users' health and comfort and how it can be related to the impact on the computer vision systems of the powered wheelchair.

6.6.1 Experiment setup

A powered wheelchair is driven for 11 meters on a carpet floor with and without artificial bumps in a controlled indoor environment. The chosen distance represents the maximum straight route of the corridor without turnings. The bumps, which are used to introduce vibrations, are installed 1.5 meters apart (Fig. 6.12) to keep the seven bumps equally distanced throughout the route length and to provide enough space for the powered wheelchair to stabilise before the next bump. Two kinds of data are collected: the accelerations using an IMU sensor installed on the powered wheelchair seat and videos using a camera installed beneath the joystick (Fig. B.2). The acceleration data has been processed for the two scenarios (with and without bumps) to quantify whole-body vibrations impact on user's health and comfort with respect to the ISO-2631 standard [163]. The two 21-seconds videos are annotated on the pixel level for the assessment of the semantic segmentation system, with around 26.8 million pixels annotated for each video.

The extracted 65 ground truth images from each video are compared with the corresponding predictions using a semantic segmentation system trained on data from the same distribution (the same indoor environment) [155]. The proposed system is based on DLV3+ [90] with some modifications [155], such as the usage of ResNet-18 [51] as a feature extraction network.

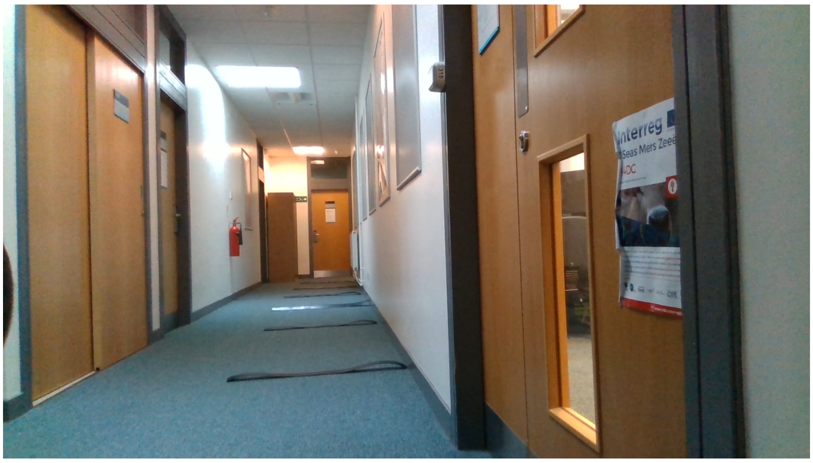


Fig. 6.12 **Artificial bumps to introduce vibrations fixed 1.5 meters apart.**

6.6.2 Results

Results show the impact of undesirable vibration on both the semantic segmentation systems and the user's health. Table 6.12 shows the impact of vibrations on the user's health and comfort (a detailed investigation of the impact of vibrations on user's health and comfort is presented in Appendix B). The calculations are made according to the ISO-2631 standard [163]. Driving the powered wheelchair on the carpet floor presents neither health risk nor discomfort to the user. The user of the powered wheelchair weighs 95 Kg and is 184 cm tall.

Table 6.12 **Vibration impact on user's health and comfort.**

State	No vibration					With vibration				
Assessment	Health				Comfort	Health				Comfort
Metric	$a_w(m/s^2)$	$MTVV(m/s^2)$	$MTVV/a_w$	$eVDV(7.5h)$	$a_v(m/s^2)$	$a_w(m/s^2)$	$MTVV(m/s^2)$	$MTVV/a_w$	$eVDV(7.5h)$	$a_v(m/s^2)$
Values	0.07	0.09	1.22	1.39	0.12	1.11	1.32	1.19	19.91	1.30
Result	No health risk				Not uncomfortable	Potential health risk				Uncomfortable

On the other hand, the introduced bumps make the situation a potential health risk and uncomfortable to the user. Fig. 6.13 shows the vertical acceleration of both scenarios (with and without the introduced vibrations). The vertical accelerations of the seven bumps can be clearly seen from the sudden change in the signal's amplitude (blue signal). In contrast, the red signal, which represents the 'no vibration' scenario, does not have sudden changes in amplitude.

The analysis of the whole-body vibration of the two scenarios is comparable with user 3 in Table B.2, for which the user drives the powered wheelchair on the carpet floor for the no vibration case, and the tiled concrete for the vibration case.

Fig. 6.14 shows the detection performance of the system in the absence (first column) and the presence (second column) of the introduced vibrations. It can be seen that the vibration has dramatically impacted the detection of objects such as the movable door handle, which the semantic segmentation system could not detect due to the sudden vibrations. Generally, the ability of the semantic segmentation system to classify the image pixels has degraded due to the introduced vibrations. Qualitatively, the degradation can be seen in the fourth row, where the intense green and magenta colours indicate these differences between the ground truth data and the system predictions. These pixels are unannotated or misclassified. The green colour shows the unannotated pixels which do not belong to objects of interest. At the same time, the magenta colour shows the misclassified objects.

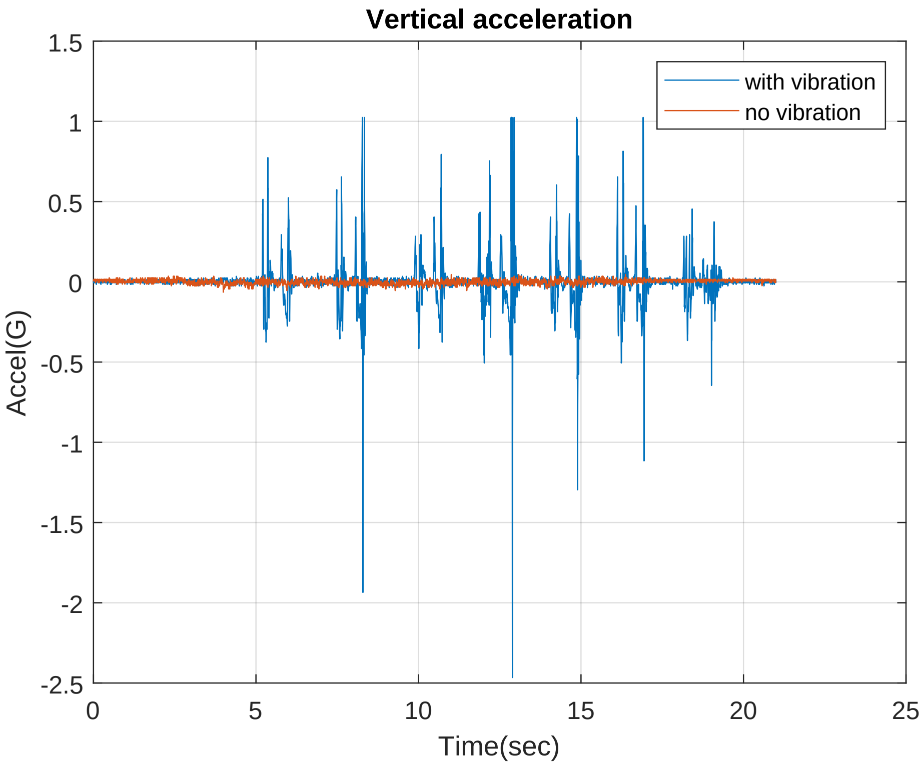


Fig. 6.13 **Vertical accelerations with and without the introduced vibrations.**

Quantitatively, Table 6.13 shows the evaluation metrics of the two scenarios (with and without the introduced vibrations). It can be observed that the performance of the semantic segmentation system degrades as a result of the introduced vibrations. Thus, it can be concluded from the results that the performance of the semantic segmenting system can be negatively impacted by the vibration encountered while driving the powered wheelchair. In addition, the change in performance is directly proportional to the amount of vibration.

To further investigate the vibration impact on the semantic system accuracy, we segregate the images of the vibration dataset (when artificial vibrations are introduced, the second row of Table 6.13) into two categories (last two rows of Table 6.13): images during the vibration incident and images before or after the vibration incident. The first group of images represents the times when the powered wheelchair encountered a bump, such as sub-figures b and d in Fig. 6.15. The second group of images represents the times when the powered wheelchair does not encounter a bump, such as sub-figures a and c in Fig. 6.15. Then, the mean and the standard

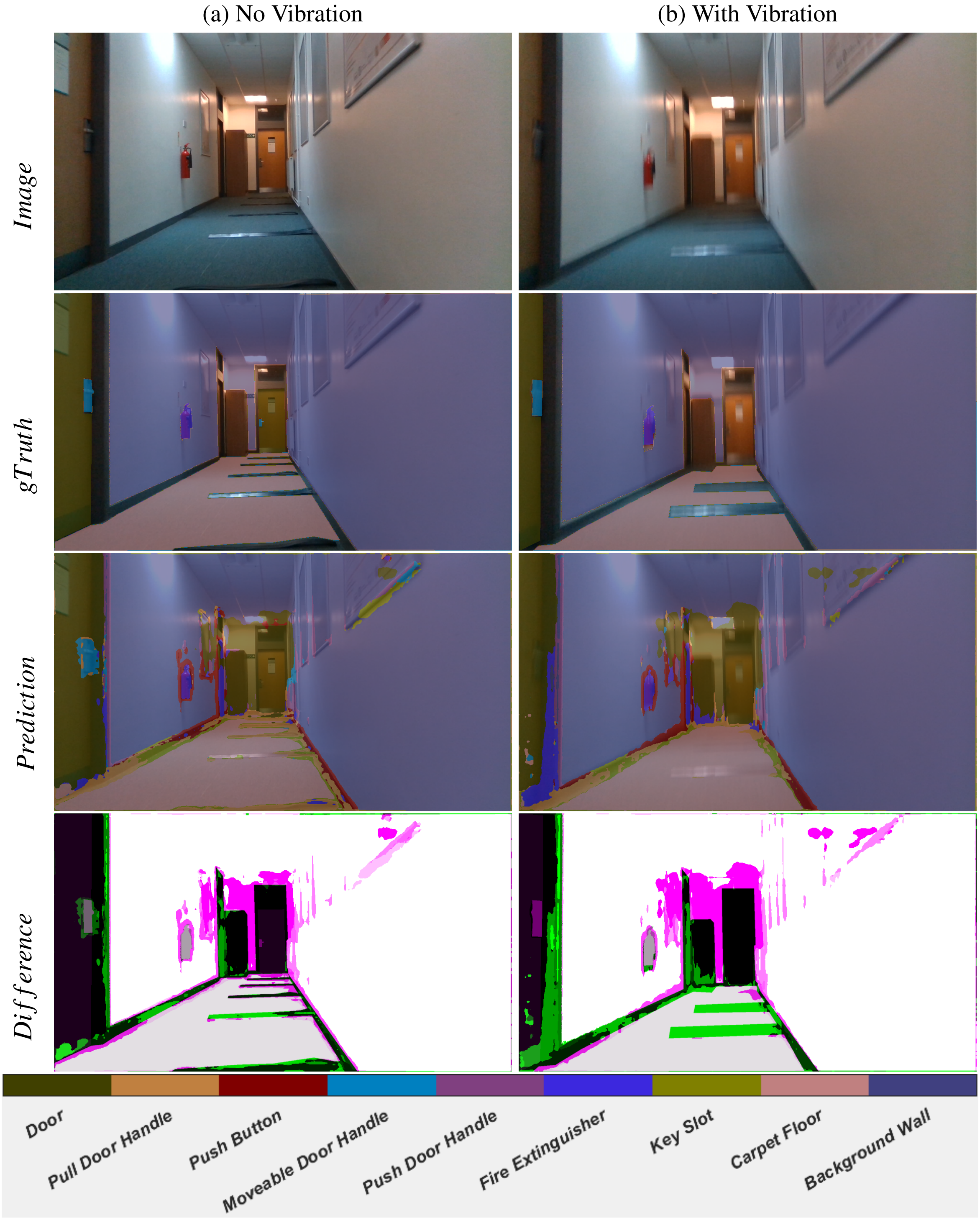


Fig. 6.14 **The impact of vibrations on the performance of the semantic segmentation system.**

Table 6.13 **Evaluation metrics with and without the introduced vibration on the images level.**

State	Metrics	Global Accuracy	Mean Accuracy	Mean IoU	Weighted IoU	Mean BF score
Without vibration (65 images)	Mean	0.914	0.492	0.340	0.889	0.508
	Std	0.040	0.061	0.034	0.051	0.075
With vibration (65 images)	Mean	0.877	0.475	0.309	0.842	0.472
	Std	0.062	0.054	0.041	0.081	0.075
Without vibration incident (50 images)	Mean	0.882	0.485	0.315	0.847	0.484
	Std	0.057	0.051	0.038	0.078	0.071
During vibration incident (15 images)	Mean	0.863	0.444	0.287	0.826	0.435
	Std	0.078	0.056	0.047	0.092	0.080

deviation of accuracy, IoU, and Mean BF score on the level of the images are calculated. The number of captured images during a vibration incident due to a bump is 15. The remaining images (50) are considered as images without vibration incident, although the total 65 images are captured together. Table 6.13 shows the metrics of the two groups of images (last two rows).

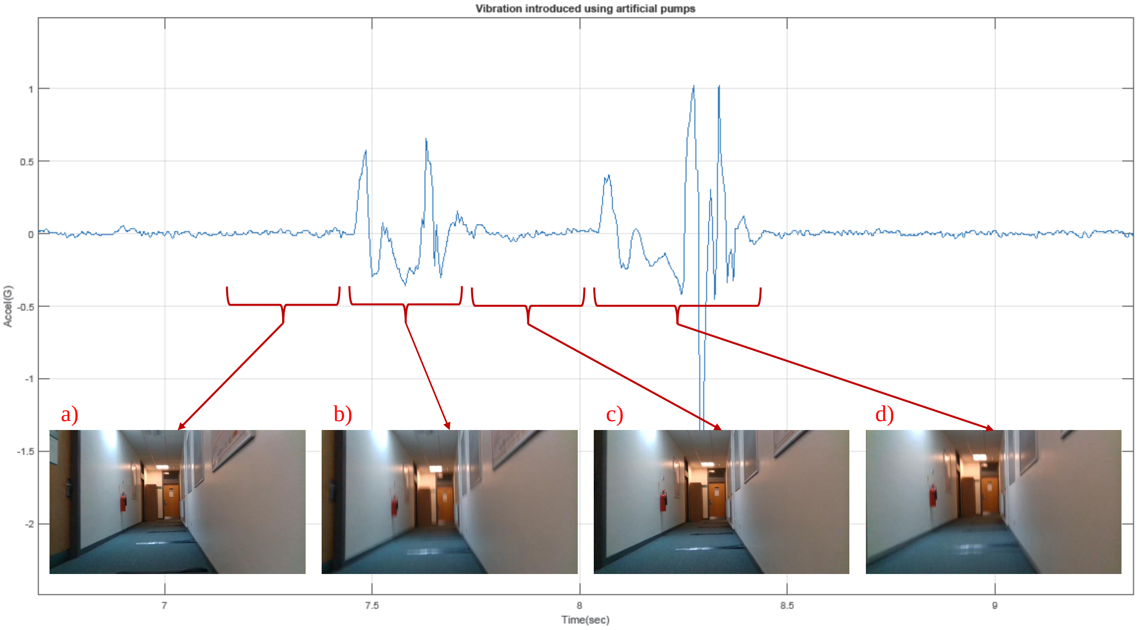


Fig. 6.15 **Segregation method for the images of the dataset with vibration.**

It can be noticed that the portion of images from the vibration dataset that is collected without the incident of external vibrations (before or after the bumps) has convergent metrics to the dataset which has been collected without any external vibrations. At the same time, the portion of images that are captured during the incident of vibration has been significantly impacted by the vibrations resulting in the lowest accuracy among all datasets.

From the results, we can anticipate a deterioration in the semantic segmentation system performance when driving a powered wheelchair on types of terrains that can cause health risks or uncomfortably to the users. Therefore, we recommend that the developers and researchers consider the impact of vibrations on their computer vision systems. A shock absorption system or a camera stabiliser holder can reduce the negative effects of the vibrations on the system's performance. Producing an accurate semantic segmentation system is beneficial for visually impaired disabled users to increase their independence. Moreover, it can allow the approval of using EPWs for those users who currently are not permitted to use powered wheelchairs due to their disabilities.

This study has been conducted on a Roma wheelchair that does not have a suspension system, similar to the tractor used in [162]. A wheelchair suspension, mainly used to dampen vibrations, may negatively impact the system's performance by introducing more vibrations to counter the external ones. This will be investigated in the future work.

6.7 Conclusion

This chapter presents semantic segmentation systems for indoor and outdoor environments. The proposed pixel classification systems have demonstrated high efficiency with adequate accuracy and BF scores. These systems are intended to help visually impaired EPWs users navigate safely and interact with the environment, as shown in the practical implementation section. Results show the proposed systems' abilities to precisely localise target objects compared to state-of-the-art semantic segmentation techniques. The proposed indoor system has achieved better mean BF scores with 9% and 5% higher than FCN-32s and DLV3+ with ResNet-50, respectively. At the same time, the outdoor system is 15% better than the FCN-32s system in terms of mean BF score. Also, the proposed DLV3+ with ResNet-18 system has achieved a processing speed of 2.65 FPS compared to 1.57 FPS and 2 FPS that the DLV3+ with ResNet-50 and with Xception have achieved, respectively.

The proposed shared systems that can process indoor and outdoor images simultaneously have achieved adequate performance on both tasks. Though, the inference speed and the overall performance are lower than that of the individual systems. Trading-off accuracy and speed with multi-distribution data processing are desirable in many applications. Moreover, the introduced shared systems do not require any retraining. This makes the proposed systems flexible and adaptable in many domains. Being able to segment images from two different data distributions simultaneously is challenging. Nevertheless, the proposed shared systems provide solutions to this challenging task, which is significantly important to many applications. The proposed systems are deployed on a GPU based board and integrated on an EPW for practical usage.

The practical implementation of the semantic segmentation system has shown the effectiveness of the proposed system in estimating the distance to the target object. Also, it can help visually impaired users to understand the surrounding environment. However, increasing the processing speed of the proposed system can further enhance the overall experience of the system.

In addition, we recommend that the researchers need to consider the impact of the vibrations on the smart systems installed on powered wheelchairs, especially computer vision systems, due to the negative implications of the vibrations on the performance of the semantic segmentation system. Our results indicated that there is a deterioration of 4% in the performance due to these vibrations.

The study has been conducted on a powered wheelchair that does not have a suspension system. A powered wheelchair suspension, mainly used to dampen vibrations, may negatively impact the system's performance by introducing more vibrations to counter the external ones. One of the future steps of this study is to investigate the impact of vibrations on the performance of semantic segmentation systems using powered wheelchairs with suspension systems.

Chapter 7

A Novel Semantic Segmentation Evaluation Method

7.1 Introduction

Pixel classification is the process of assigning a label to each pixel in an image from a predefined set of labels. It is a deep learning technique to semantically segment, hence the Semantic Segmentation (SS) name, an image into an annotated scene where each pixel has a label. A group of pixels with the same label represents an object. SS tasks are treated as supervised learning problems for which classifiers are trained to fit the training data on the pixels level. SS systems have been used in many human-system interaction applications. Consequently, enhancing the evaluation process of SS systems can better reflect the performance of these systems and enhance the human-system interaction.

SS systems have seen rapid progress in the past few years, not only from the performance perspective but also in the processing speed. Pixel classification systems have many applications such as autonomous driving [164], medical applications [93], and general scenes understanding [165]. Different Convolutional Neural Network (CNN) architectures for SS tasks have been proposed. These systems follow two main categories: the series architecture such as Fully

Convolutional Networks (FCN) [83] and the encoder-decoder architecture such as U-Net [93], and DeepLab [90]. The efficiency of a SS system can be measured by its performance on a target application. However, this kind of benchmarking is flawed because of the inability to compare algorithms due to the subjectivity of the measure. This leads to the introduction of other general application-independent metrics.

Many metrics are introduced to evaluate different deep learning tasks, for example. Accuracy (Acc) is used to evaluate Object Classification (OC) tasks. Average Precision (AP) and Bounding Box Intersection over Union (BB_IoU) are used to evaluate Object Detection (OD) tasks. BF score is used to evaluate SS tasks. Mask Intersection over Union (Mask_IoU) is used to evaluate Instance Segmentation (IS) tasks. Other metrics, such as Panoptic Quality (PQ) [166], is introduced to unify the evaluation of semantic and instance segmentations. PQ can be used to assess the performance of a system on both stuff and things classes (stuff classes such as sky, grass, etc., while things classes such as cars, people, etc.) in a simple and informative manner. Unlike Panoptic Segmentation (PS) and SS, IS incorporates Object (segment) confidence score in the AP metric calculation. Confidence scores are essential elements in the evaluation process of any system. These scores add a further informative dimension to the downstream systems. Consequently, utilising pixels' confidence scores to evaluate SS systems' performance can help to better assess these systems.

This chapter focuses on SS tasks, as we want to incorporate the pixels' confidence scores in the evaluation process of SS systems to better understand their behaviours. Results show that pixels' confidence scores can dramatically affect system performance evaluation. It can also provide a deep understanding of the system's operation. The main contributions of this chapter are as follows: a new evaluation technique that can be incorporated with the existing SS evaluation metrics is proposed. This technique is based on a well-known idea of thresholding that has been used in many applications over the past years. However, introducing this technique to evaluate SS tasks can be considered a novel contribution. Thresholding of pixels' confidence scores can contribute to the SS overall output. Consequently, it has a significant impact on the evaluation metrics. Pixels thresholding is distinct from Mask_IoU or BB_IoU as it is computed on the pixel level and not on the mask or the bounding box level as in the case of IS or OD tasks, respectively. Two datasets have been used to investigate the contribution of the new element (pixels' confidence scores) on the system's output and the evaluation metrics: a) the standard Cambridge-driving Labeled Video Database (CamVid) [154], b) a manually collected dataset for indoor environment [126].

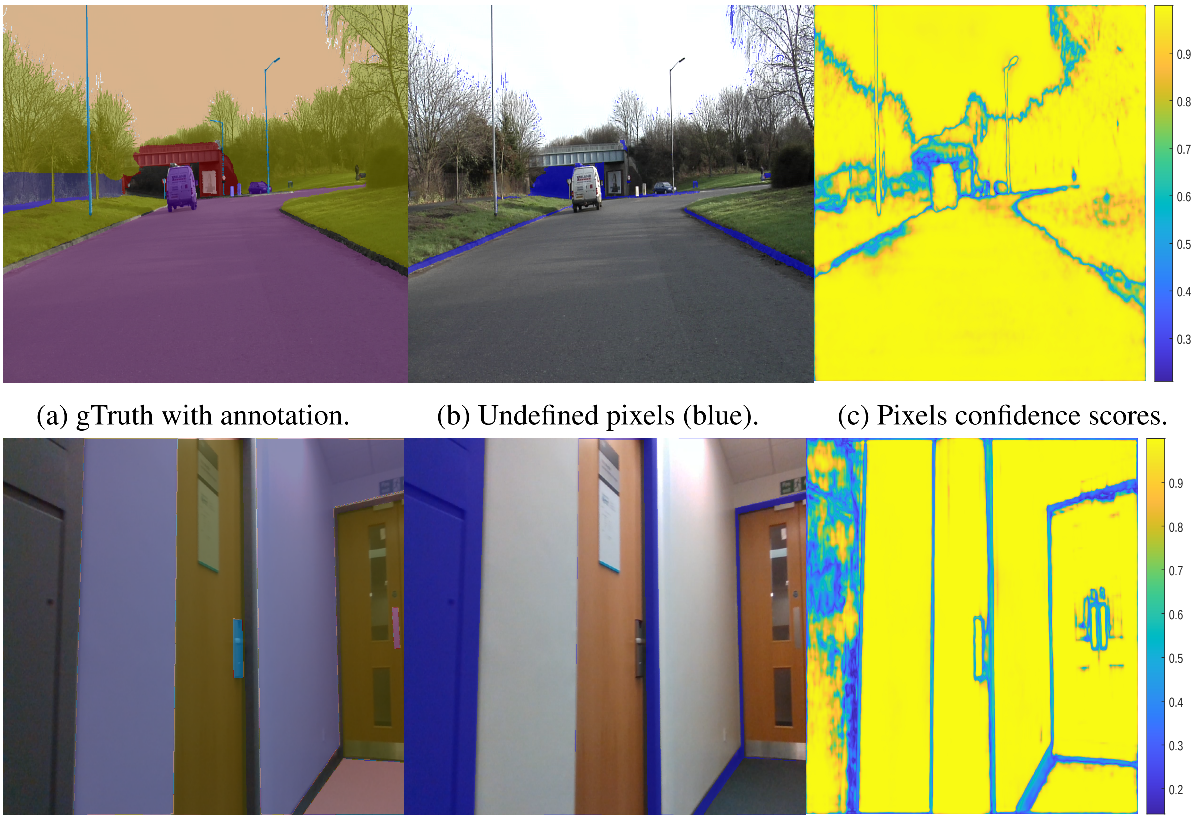
This chapter is organised as follows: research methodology is presented in section 7.2. Experimental setup is discussed in section 7.3. Section 7.4 presents the results and findings. Lastly, the chapter is concluded in section 7.5.

7.2 Methodology

Semantic Segmentation evaluation metrics such as accuracy, IoU and BF score do not incorporate pixels' scores into their calculations. Pixels' scores reflect the degree of confidence a pixel belongs to a specific class from a set of predefined classes. For semantic segmentation tasks, all pixels of an image have to be assigned to one of the predefined classes, even though these pixels might not belong to any of the classes. For example, if the predefined classes for a particular semantic segmentation system do not include a car class, but an image that needs to be classified by the system contains a car, the system will assign the car's pixels to any predefined class. Usually, these pixels have very low scores. Nevertheless, these pixels contribute to the system's performance as they are used by the traditional evaluation metrics during the evaluation process.

On the other hand, Pixels of objects of non-interest are usually kept unlabelled in the ground truth (gTruth) data (undefined or void pixels). While these pixels should not be used for the system evaluation, traditional evaluation metrics do not exclude them.

We propose a novel evaluation technique that incorporates pixels threshold in the evaluation process. The method is similar to posterior probability in statistics [167, 168] at which all the 'undefined' pixels in the gTruth data are assigned to an extra class called 'Reject' class. Usually, these pixels belong to objects of non-interest, object borders or unannotated pixels. In the case of CamVid dataset, these pixels might belong to far objects or pavement borders (Fig 7.1b). Whereas these pixels belong to door frames and cupboards for the indoor dataset (Fig 7.1e). Fig 7.1c and Fig 7.1f show that these pixels have the lowest confidence scores.



(a) gTruth with annotation. (b) Undefined pixels (blue). (c) Pixels confidence scores.

(d) gTruth with annotations. (e) Undefined pixels (blue). (f) Pixels confidence scores.

Fig. 7.1 **Undefined pixels result in low pixels' confidence scores.** The first row is for CamVid dataset while the second one is for the indoor dataset.

To compare the predicted pixels with the gTruth ones, the predicted pixels below a predefined threshold are assigned to the 'Reject' class. If the trained system is robust enough, these low score pixels should belong to objects of non-interest. In addition, these objects are not in the predefined set of classes, and the system has not seen them before. Then the predicted output of the system is evaluated against the gTruth data at different threshold values to investigate the system's behaviour and the threshold impact on the overall system performance. Fig 7.2 illustrate the methodology.

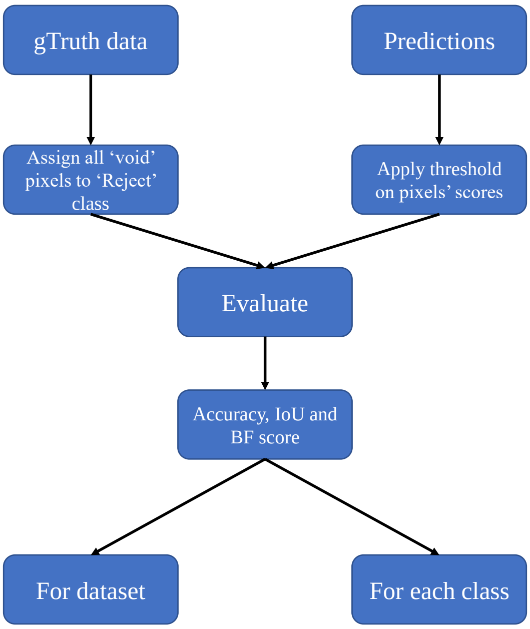

gTruth data
Predictions
Assign all 'void' pixels to 'Reject' class
Apply threshold on pixels' scores
Evaluate
Accuracy, IoU and BF score
For dataset
For each class

Fig. 7.2 **Methodology.**

7.2.1 Semantic segmentation evaluation metrics

The performance of semantic segmentation systems can be evaluated using the following metrics: Accuracy, IoU and Mean BF score. Each metric reflects a specific quality of the system, such as the ability of the system to classify pixels correctly or the alignment of the predicted pixels with the gTruth ones. However, the aforementioned metrics do not consider pixels' confidence scores in their calculations. The softmax layer of a typical semantic segmentation system based on convolutional layers outputs several scores for each pixel in the image corresponding to the number of classes. The highest value represents the class of that pixel. In case of uncertainty, and sometimes border pixels, even the highest score pixel across all classes has a low value. Nevertheless, they still contribute to the system performance.

Two types of accuracy can be calculated for a dataset: Global Accuracy (GA) and Mean Accuracy (MA). GA is calculated regardless of the class as the ratio between correctly classified pixels to the total number of pixels (7.1). Whereas MA is the average accuracy of all classes in all images. Accuracy of each class can be calculated as the ratio of correctly classified pixels to the total number of pixels in that class using (7.2). A major limitation of GA and MA measures is the bias in the presence of imbalanced classes.

$$GA = \frac{TP+TN}{TP+FP+FN+TN} \tag{7.1}$$

Where TP, TN, FP, FN are True Positive, True Negative, False Positive, and False Negative, respectively.

$$Acc_{class} = \frac{TP}{TP+FN} \tag{7.2}$$

Similarly, Mean IoU for a dataset can be calculated as the average IoU of all classes in all images. IoU (also known as Jaccard index) for each class is the ratio between correctly classified pixels to the total number of predicted and gTruth pixels in that class (7.3). For disproportionately distributed classes, weighted IoU can be reported to better reflect the system behaviour. It is the standard IoU but weighted by the number of pixels of each class in the dataset. IoU metrics evaluate the number of correctly classified pixels but do not reflect the quality of the object's boundaries, which can be considered a disadvantage. Trimap [169] is introduced to overcome this drawback by evaluating the segmentation accuracy around the segment boundaries using a predefined narrow band around the contours. At the same time, pixels in this predefined band contribute to the accuracy calculations. The technique proposes to measure pixels accuracy within a defined region around the object boundaries rather than considering all image pixels to better assess the system's ability to capture objects' boundaries.

Nevertheless, choosing the optimal band size is challenging and might vary depending on the application.

$$IoU_{class} = \frac{TP}{TP+FP+FN} \tag{7.3}$$

Information retrieval [170] approaches have used Precision-Recall curves as a standard evaluation metric. The metric was first used to evaluate edge detectors by Abdou and Pratt [171]. Precision measures the ratio of detections that are TP rather than FP (7.6). Whereas Recall measure the ratio of the detected TP rather than missed (7.7). The parametric Precision-Recall curve captures the trade-off between accuracy and noise while the detector threshold changes [172]. A permissible trade-off for a particular application between noise and accuracy can be defined by the relative cost α in the F1 score equation (7.4).

$$F1_{score} = \frac{P \cdot R}{(1-\alpha) \cdot P + \alpha \cdot R} \tag{7.4}$$

F1 score calculates the weighted harmonic mean of Precision and Recall. The maximum F1 score, which is the point on the curve where the optimal detector threshold occurs, can be reported as an indication of the detector's performance [172]. In our experiments, we set α to 0.5. Thus, (7.4) can be simplified to (7.5). Also using (7.6) and (7.7), (7.5) can be simplified to (7.8).

$$F1_{score} = \frac{2 \cdot P \cdot R}{P+R} \tag{7.5}$$

Where P and R are the Precision and Recall, respectively.

$$P = \frac{TP}{TP+FP} \tag{7.6}$$

$$R = \frac{TP}{TP + FN} \tag{7.7}$$

So,

$$F1_{score} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{7.8}$$

Trimap [169] does not fully capture the quality of the contours. Thus, a semantic segmentation contour-based accuracy metric called Boundary F1 score (BF score) is proposed [173]. BF score metric is inspired by boundary-based evaluation measure [174] [175] and F1 score [172]. The boundary-based evaluation measure and F1 score define a distance tolerance to decide if a match has happened between pixel boundary points in the prediction and gTruth images.

Boundary-based measure [174] calculates the minimum euclidean distance between two sets of points where the sets represent the boundaries of two segments (gTruth and prediction). Hence, the mean and the standard deviation is calculated from the distance distribution between the two sets. A small mean and standard deviation indicate high matching. Whereas the weighted harmonic mean of Precision and Recall is used in the case of F1 score [172] to estimate the point for the optimal detector threshold.

BF score [173] extended F1 score to semantic segmentation tasks. The proposed metric (BF score) has been used to calculate one value per class to evaluate classes independently. BF score sets the distance tolerance to 0.75% of the length of the image diagonal.The same ratio is used in our experiments.

Mean BF score or contour matching score, which measure the alignment of the predicted and gTruth boundaries, is the average BF score of all classes in all images for a dataset (7.10). Whereas Mean BF score of a class is the average F1 score (7.8) of that class over all images (7.9).

$$MeanBF_{\text{score}}^{\text{class}} = \frac{\sum F1_{\text{score}}^{\text{class}}}{\text{no\# of images}} \tag{7.9}$$

$$MeanBF_{\text{score}}^{\text{dataset}} = \frac{\sum_{classes} BF_{\text{score}}}{\text{no\# of images}} \tag{7.10}$$

Zhang et al. [176] introduce two semantic segmentation measures to reflect over and under segmentation. The introduced metrics are region-based, unlike standard metrics such as IoU, which are pixel-based. Over segmentation happens when the gTruth region overlaps with many predicted regions of the same object. In contrast, under segmentation happens when the predicted region overlaps with at least two different gTruth regions. The measures take a value in the range of zero to one, where zero indicates no over or under segmentation. A major limitation of these metrics is that they cannot be used to measure classification accuracy [176].

IoU is the standard evaluation metric for PASCAL VOC challenge [132]. However, solely depending on a specific metric to assess a SS system is insufficient. Csurka et al. [173] argue that systems' parameters for a segmentation algorithm should be optimised on the target metric for fair comparisons as different segmentation algorithms can be optimal for different evaluation metrics. Besides, per-image metrics can provide more details of the system's performance and allow more detailed comparisons. Thus, in our experiments, we have reported accuracy, IoU, and BF score for the dataset and for each class. We believe that these metrics are complementary to each other. Furthermore, incorporating pixels' confidence scores with these metrics can reveal another level of information regarding the system's performance.

7.2.2 Proposed technique relation to the existing metrics

The calculations of standard SS evaluation metrics ignore pixels' confidence scores. We propose to incorporate pixels' confidence scores in the calculation process of these metrics because of the important information that can be reflected by these scores. First, we predefine a pixel score threshold. If the pixel's value after the softmax layer (the highest pixel value across all pixel classes) is below this threshold, its category is assigned to the 'Reject' class. Class 'Reject' cannot contribute to any of the evaluation metrics calculations. For a robust system, a low confidence pixel score usually represents a high uncertainty pixel. This pixel can be for a class of non-interest (i.e., has not been predefined for the task) or a pixel between the borders of predefined classes of interest. Lastly, the system is evaluated using the standard metrics.

After thresholding, predicted pixels corresponding to gTruth ones count towards TP, predicted pixels different from gTruth ones count towards FP, and unpredicted gTruth pixels count towards FN (Fig. 7.3).

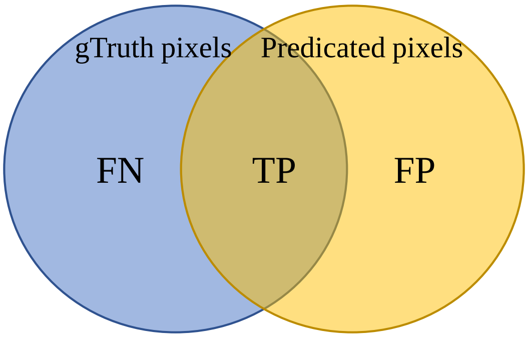


Fig. 7.3 **Illustration of TP, FP and FN for pixels.**

The novelty of the proposed technique is in assessing the system under different conditions (pixels' threshold values). SS systems under various conditions can behave differently. Consequently, the system behaviour should be well-investigated using several impacting factors. The most important impacting factor is the pixel itself. Thus, we believe the pixels' scores should contribute to the evaluation process of any SS system.

The segment matching for the IoU denoted by $\text{IoU}(p,g)$ in (7.11) for the Panoptic Quality metric is different from the Semantic Segmentation IoU as the former is calculated between two segments (p and g for the predicted and gTruth segments, respectively). Whereas the latter is calculated based on the output pixel labels and completely ignore object-level labels. Also, the segment matching threshold of 0.5 used by PQ is distinct from the proposed pixel confidence thresholding. The former is computed on the segment matching IoU level (object level). In contrast, the latter is computed on the pixel confidence score level.

$$\mathrm{PQ} = \frac{\sum_{(p,g)\in TP} \mathrm{IoU}(p,g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \tag{7.11}$$

Similar to PQ, the over and under region-based semantic segmentation measures [176] use the predicted region confidence score in the evaluation process. Predicted region confidence score is calculated as the numerical mean of the confidence scores of all the pixels enclosed in that region. As the over and under segmentation measures compare regions, they are different from the proposed technique which operates on the pixel level.

The proposed process is simple and straight forward (Fig. 7.2). However, the added dimension of the pixels' confidence scores helps to exclude the contribution of a specific area in an image with respect to the overall performance of the system. This excluded area might be undefined by the annotator, yet it still contributes to the metrics calculations, which is undesirable in many cases.

7.3 Experimental setup

In our experiments, we have reported GA, MA, Mean IoU, Weighted IoU and Mean BF score using four different pixels' threshold values (0.2, 0.4, 0.6, 0.8) that are monotonically increasing. The choice of these threshold values helps to capture the system's behaviour under a wide range of conditions. The evaluation metrics are calculated for the dataset and the individual classes.

7.3.1 Dataset

Two datasets are used to test the proposed evaluation technique: CamVid [154] and the indoor dataset [126].

CamVid dataset has 701 images annotated on the pixel level for 32 classes. Images are captured outdoors from the perspective of a driving car. We group the 32 classes of the dataset into 11 classes for simplicity as some of the 32 original classes have very limited objects. These

11 classes are Building, Pole, Road, Pavement, Tree, Sign/Symbol, Car, Pedestrian, Bicyclist, Sky, and Fence.

The dataset contains some undefined pixels which belong to non-of-interest objects or overlooked pixels. We also define an extra ‘Reject’ class for the purpose of the experiments for which all the undefined pixels are assigned. Pixels distribution of the gTruth data is shown in Fig. 7.4.

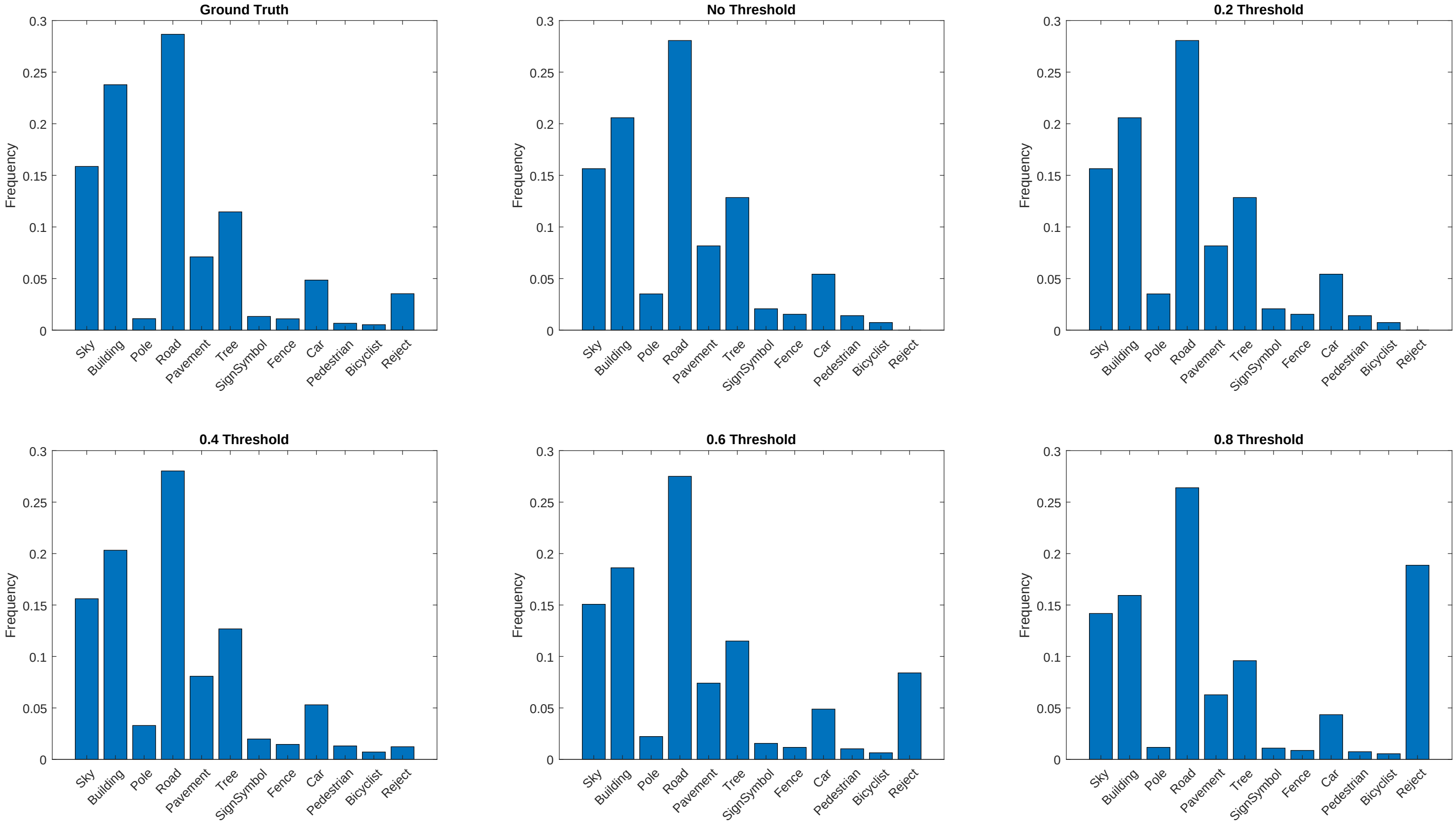


Fig. 7.4 **Pixels distribution at different threshold values for the CamVid test set.**

The dataset is split randomly into 70% for training (491), 15% for validation (105 images) and 15% for testing(105 images).

Indoor dataset has 1,549 images annotated on the pixel level for 9 classes: Door, Carpet floor, Background wall, Fire extinguisher, Key slot, Push button, and different kinds of door handles such as Moveable, Pull and Push door handles.

Similarly the dataset contains some undefined pixels which are assigned to the ‘Reject’ class (Fig. 7.5) and follows the same splitting ratio as CamVid with 1,084 images for training, 232 images for validation and 233 images for testing.

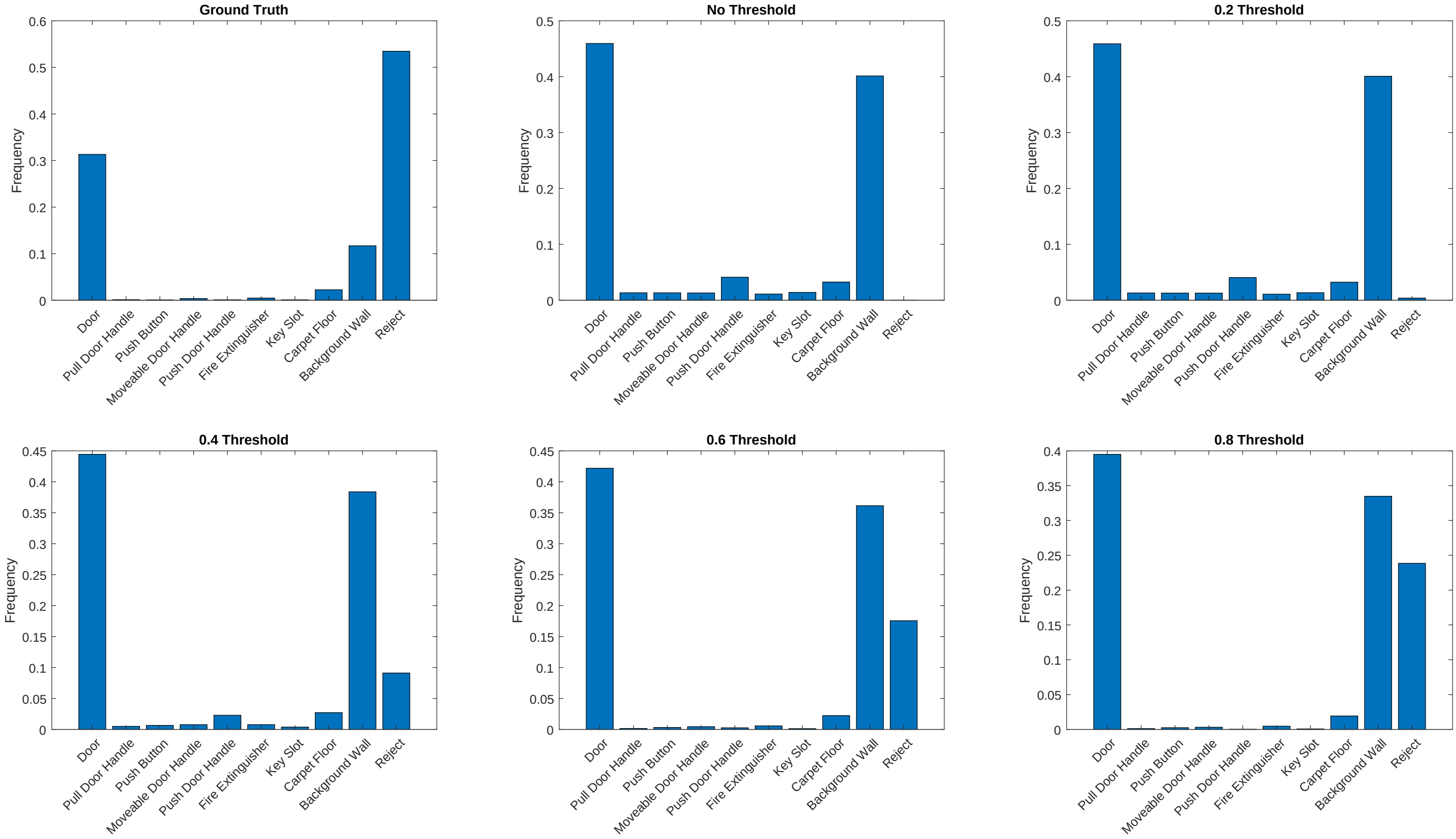


Fig. 7.5 **Pixels distribution at different threshold values for the Indoor test set.**

7.3.2 System architecture

The convolutional neural network architecture that has been used for training is based on the encoder-decoder DeepLab Version 3 plus (DLV3+) [90] for semantic segmentation. The architecture’s base network uses residual blocks [51] that help the systems to process high-resolution images (960×540×3 and 960×720×3 pixels for the CamVid and the indoor datasets, respectively) without losing information because of vanishing gradients. In addition, the system’s decoder has a simple design but with high efficiency.

Very deep networks suffer from vanishing and exploding gradients [147, 148]. Residual blocks help to mitigate this problem by reusing the activations from previous layers until the adjacent layer learns its weights. This allows the network to better learn low-level features without performance degradation as the network goes deeper. The elegance of this architecture is that these short-cut connections do not add either extra parameters or computational complexity [51].

7.3.3 Training

The systems are trained end-to-end on the CamVid and the indoor datasets using Stochastic Gradient Descent with 0.9 Momentum (SGDM) as the training optimiser. A starting learning rate of 0.001 which is then dropped by a factor of 0.3 every ten epochs. To avoid sequence memorisation, training images are shuffled every epoch. Also, L2 regularisation is used to limit overfitting. To enhance the overall system accuracy, data augmentation is employed with X and Y translations. Additionally, different hyper-parameters and optimisation algorithms are tried to achieve the highest performance. Moreover, for reproducibility, systems are trained several times under the same configurations.

To avoid bias in favour of dominant classes, inverse frequency weighting is used to balance classes weightings. Image normalisation is employed to rescale all the pixels' values in the range of zero to one. System are trained on relatively high-resolution images of 960×720×3 and 960×540×3 for the CamVid and the indoor datasets respectively. Training on high-resolution images is believed to enhance the system's ability to semantically segment small size objects alongside medium and large size ones. Also, high-resolution images boost the effectiveness of large rate atrous convolution kernels used by DLV3+ because the kernels' weights are applied to actual pixels and not to zero paddings.

7.4 Results and discussion

The trained systems on the CamVid and the indoor datasets have achieved validation losses of 0.368 and 0.564, respectively. The 0.75% tolerance distances for the BF score metric calculations are 9 and 8 pixels for the CamVid and Indoor datasets, respectively.

7.4.1 CamVid dataset results

Results show interesting behaviours of the evaluation metrics regarding different size objects using various threshold values. Thresholding has proved that the pixels' confidence scores significantly impact the system's performance with respect to the datasets and the individual classes. Consequently, the proposed technique can be used to optimise the system on a specific application, task, or group of objects of interest.

Table 7.1 shows that the system's performance on the CamVid dataset varies under different thresholds. While applying no threshold, it has achieved the highest MA and Weighted IoU. Whereas higher threshold values have achieved better GA, Mean IoU and Mean BF score. Consequently, it is feasible to optimise the system on a specific evaluation metric for a specific application or challenge.

Similar observations can be extracted from Table 7.2. Although applying no threshold has achieved the highest accuracy for all classes regardless of the object's size or pixels distribution, higher threshold values have achieved better IoU and mean BF score.

Table 7.1 **Evaluation metrics for the CamVid test set at different thresholds values.**

Metrics / Threshold	*Global Accuracy*	*Mean Accuracy*	*Mean IoU*	*Weighted IoU*	*Mean BFScore*
No threshold	0.895	**0.864**	0.667	**0.831**	0.690
0.2	0.895	0.864	0.667	0.831	0.690
0.4	0.900	0.858	0.672	0.831	0.697
0.6	0.929	0.815	**0.682**	0.817	**0.699**
0.8	**0.960**	0.731	0.663	0.772	0.660

Table 7.2 **Class metrics for the CamVid test set at different threshold values.**

Threshold Values	No Threshold			0.2			0.4			0.6			0.8		
Metrics / Class	*Acc*	*IoU*	*M.BF*	*Acc*	*IoU*	*M.BF*	*Acc*	*IoU*	*M.BF*	*Acc*	*IoU*	*M.BF*	*Acc*	*IoU*	*M.BF*
Sky	**0.940**	**0.908**	**0.905**	0.940	0.908	0.905	0.939	0.908	0.905	0.920	0.901	0.892	0.880	0.872	0.831
Building	**0.816**	**0.796**	**0.633**	0.816	0.796	0.633	0.809	0.791	0.615	0.753	0.742	0.518	0.654	0.650	0.385
Pole	**0.731**	0.240	0.578	0.731	0.240	0.578	0.721	0.249	0.588	0.635	0.294	0.630	0.476	**0.322**	**0.645**
Road	**0.941**	**0.928**	**0.817**	0.941	0.928	0.817	0.940	0.927	0.816	0.928	0.919	0.784	0.896	0.892	0.706
Pavement	**0.903**	0.741	0.750	0.903	0.741	0.750	0.899	0.742	**0.751**	0.865	**0.746**	0.749	0.782	0.718	0.696
Tree	**0.904**	0.780	0.722	0.904	0.780	0.722	0.901	**0.783**	**0.726**	0.859	0.778	0.707	0.760	0.723	0.598
SignSymbol	**0.766**	0.456	0.543	0.766	0.456	0.543	0.757	0.463	0.555	0.698	**0.496**	0.597	0.592	0.495	**0.633**
Fence	**0.806**	0.571	0.564	0.806	0.571	0.564	0.798	0.584	0.584	0.737	**0.605**	0.600	0.639	0.583	**0.608**
Car	**0.925**	0.804	0.760	0.925	0.804	0.760	0.919	0.808	0.767	0.888	**0.811**	**0.768**	0.829	0.788	0.725
Pedestrian	**0.859**	0.457	0.625	0.859	0.457	0.625	0.849	0.474	0.649	0.794	0.518	0.716	0.693	**0.533**	**0.719**
Bicyclist	**0.915**	0.656	0.555	0.915	0.656	0.555	0.909	0.665	0.598	0.885	0.695	0.669	0.834	**0.715**	**0.787**

Large size objects, and therefore high pixels distribution (Fig. 7.4) such as Sky, Building, and Road, have achieved the best performance under no pixels' scores threshold. Large-medium and medium size objects, such as Tree and Pavement, have achieved better IoU and Mean BF score using moderate pixels' threshold values of 0.4 and 0.6. For medium-small and small size objects, which have the lowest pixels distributions but vital to many applications such as Pole, SignSymbol, Fence, Car, Pedestrian, and Bicyclist, applying high threshold values have achieved the best performance in terms of higher IoU and Mean BF scores (Table 7.2).

Objects' sizes and the frequency of pixels have a great impact on the system's behaviour. Consequently, they have a direct impact on the evaluation process. As an example, when the number of pixels that are assigned to the 'Reject' class increase due to the increase in the threshold values, IoU and mean BF score of things classes, that are mainly of medium and small sizes, increase (Fig. 7.4 and Table 7.2).

On the other hand, large size and high pixels frequency classes (stuff classes) have performed best using low or no pixels' threshold values. Thus, optimising the network on a specific class or group of classes for a particular task is straight forward thanks to the thresholding technique.

Remarkable results are shown in Fig. 7.6 which depicts the per-image IoU at different threshold values. The number of images that have achieved an overall IoU of more than 0.5 increases as we increase the threshold values. Consequently, pixels' threshold values can directly impact the classifier performance, which in this case can indicate the enhancement of the system performance on the IoU metric. The performance boost can be attributed to the high uncertainty of the undefined pixels and pixels at the object's borders that can be elevated using the appropriate pixels' threshold values to reduce the impact of fuzzy pixels quantitatively and qualitatively.

Fig. 7.7 shows that as the pixels' threshold values increase, the number of rejected pixels increase. Mainly these rejected pixels represent the ones with the lowest confidence scores at the borders of the objects. High threshold values can result in well-defined object borders such

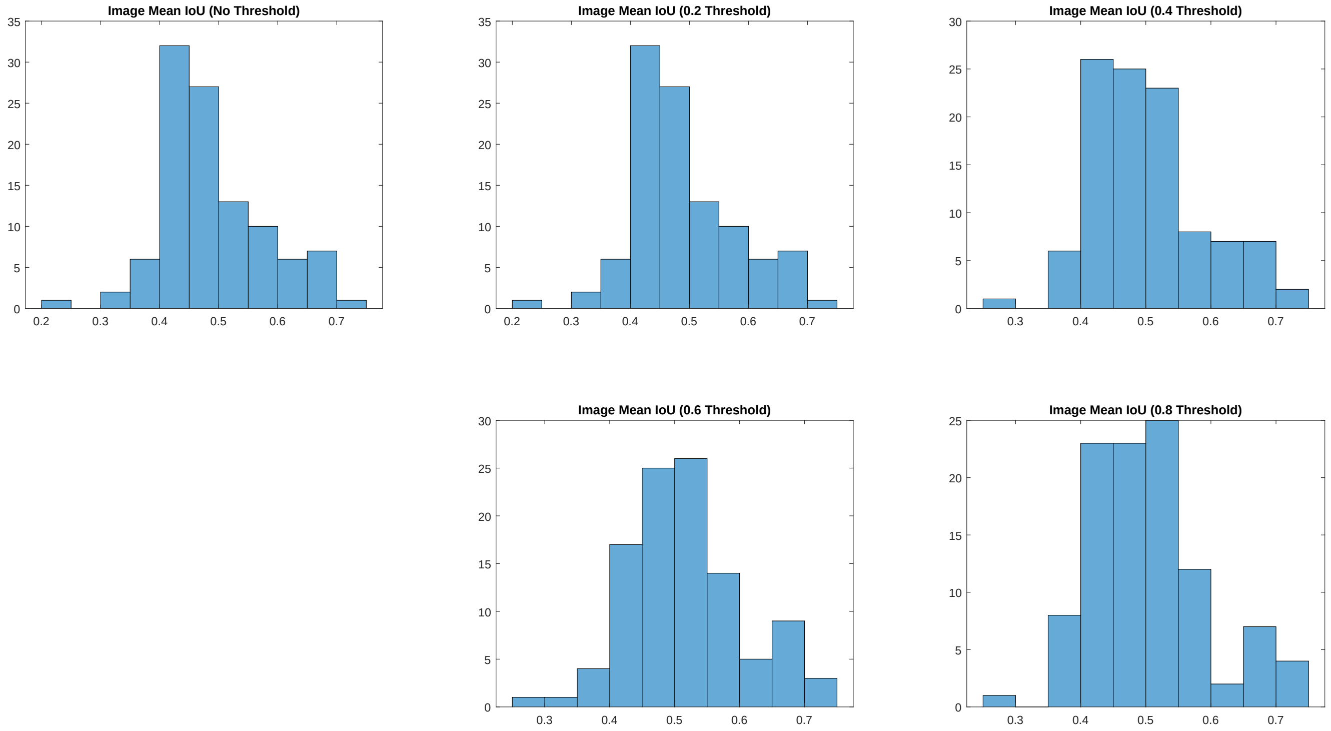


Fig. 7.6 **Histogram of per-image IoU for CamVid test set at different threshold values.** The x-axis represents the IoU value and the y-axis represents the number of images.

as the pedestrian at 0.6 and 0.8 threshold values. This can be attributed to the high uncertainty of border pixels. However, very high threshold values can wrongly reject pixels of objects of interest.

7.4.2 Indoor dataset results

Many objects in the Indoor dataset can be described, with regard to their sizes, as very small or tiny which makes the SS system task challenging. Also, the Indoor dataset have relatively higher distribution of undefined pixels compared to the CamVid dataset. Similar to the CamVid dataset, the trained SS system on the indoor dataset has achieved the highest MA and Weighted IoU when no threshold is applied (Table 7.3). Whereas the highest GA, Mean IoU, and Mean BF score are achieved using the highest threshold value of 0.8.

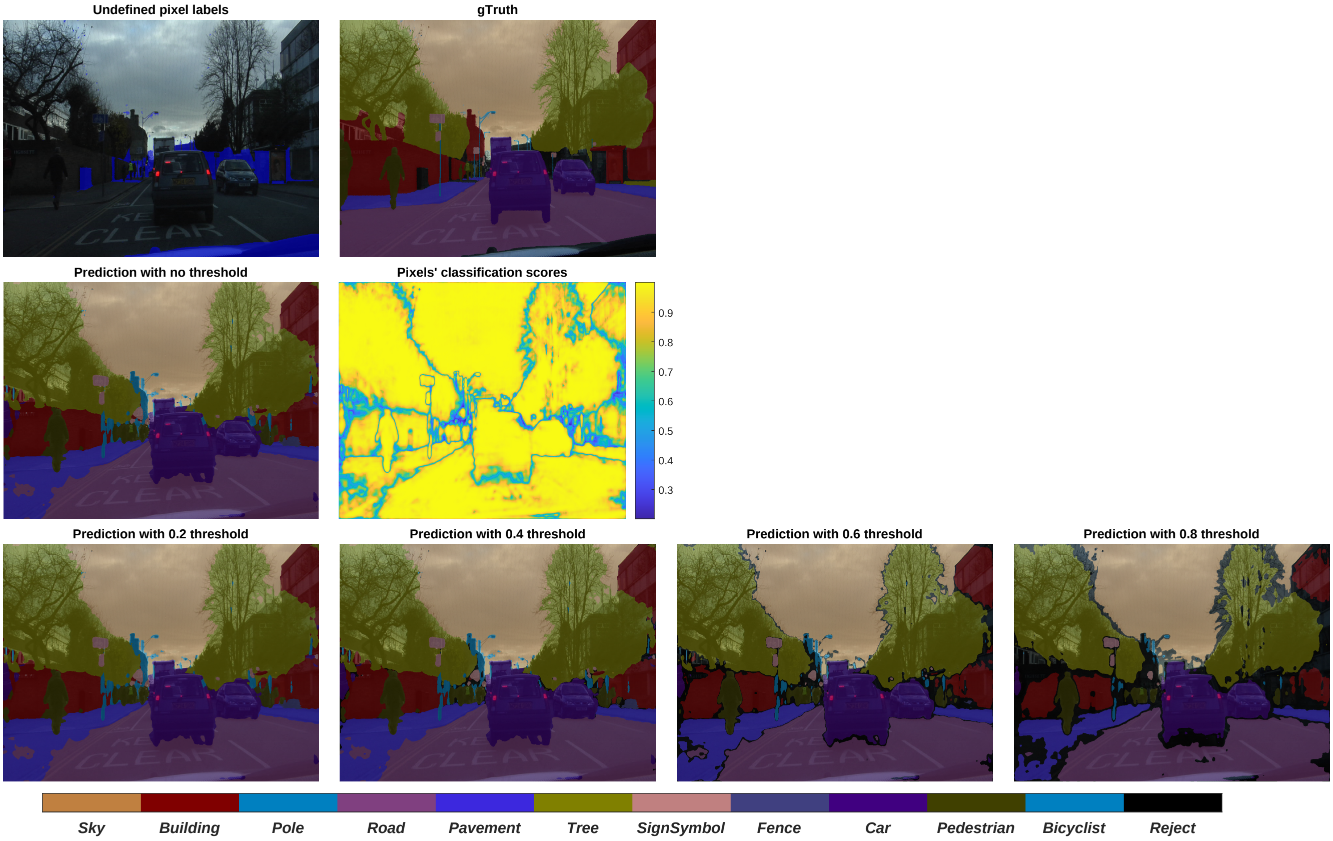

Undefined pixel labels
gTruth
Prediction with no threshold
Pixels' classification scores
0.9
0.8
0.7
0.6
0.5
0.4
0.3
Prediction with 0.2 threshold
Prediction with 0.4 threshold
Prediction with 0.6 threshold
Prediction with 0.8 threshold
Sky
Building
Pole
Road
Pavement
Tree
SignSymbol
Fence
Car
Pedestrian
Bicyclist
Reject

Fig. 7.7 **Qualitative results of the proposed thresholding technique on the CamVid dataset.**

Individual classes have achieved the highest accuracy when no pixel's threshold value is applied (Table 7.4). Large size and high pixels' distributions objects (Fig. 7.5) such as Door, Carpet floor, Background wall have also achieved the highest IoU at no threshold value. Very small size, thus, the lowest pixels' distributions objects such as different kinds of door handles and Push button have achieved high performance in terms of IoU and Mean BF score at high threshold values. Tiny object such as Key slot has a achieved its highest IoU and Mean BF score at the highest threshold value of 0.8.

Table 7.3 **Evaluation metrics for the indoor test set at different thresholds values.**

Metrics / Threshold	*Global Accuracy*	*Mean Accuracy*	*Mean IoU*	*Weighted IoU*	*Mean BFScore*
No threshold	0.970	**0.791**	0.572	**0.963**	0.673
0.2	0.970	0.791	0.572	0.963	0.673
0.4	0.980	0.755	0.598	0.961	0.673
0.6	0.994	0.696	0.635	0.957	0.666
0.8	**0.997**	0.669	**0.639**	0.948	**0.771**

Table 7.4 **Class metrics for the indoor test set at different threshold values.**

Threshold Values	**No Threshold**			**0.2**			**0.4**			**0.6**			**0.8**		
Metrics / Class	*Acc*	*IoU*	*M.BF*	*Acc*	*IoU*	*M.BF*	*Acc*	*IoU*	*M.BF*	*Acc*	*IoU*	*M.BF*	*Acc*	*IoU*	*M.BF*
Door	**0.983**	**0.983**	0.870	0.983	0.983	0.870	0.982	0.982	**0.871**	0.979	0.979	0.862	0.974	0.974	0.843
Door Handle	**0.593**	0.150	0.593	0.593	0.150	0.593	0.531	0.191	0.541	0.454	0.267	0.412	0.448	**0.305**	**0.601**
Push Button	**0.790**	0.338	0.571	0.790	0.338	0.571	0.748	0.437	**0.656**	0.642	0.597	0.351	0.634	**0.610**	0.556
Door Handle	**0.786**	0.665	0.543	0.786	0.665	0.543	0.781	0.680	0.559	0.752	**0.698**	0.587	0.703	0.679	**0.621**
Push Door Handle	**0.533**	0.090	0.341	0.532	0.090	0.339	0.497	0.103	**0.353**	0.289	0.182	0.250	0.220	**0.216**	0.323
Fire Extinguisher	**0.909**	**0.889**	0.650	0.909	0.889	0.658	0.900	0.885	0.729	0.870	0.863	0.775	0.824	0.822	**0.799**
Key Slot	**0.654**	0.186	0.488	0.654	0.186	0.488	0.510	0.268	0.215	0.471	0.320	0.494	0.457	**0.385**	**0.874**
Carpet Floor	**0.900**	**0.889**	**0.751**	0.900	0.889	0.751	0.881	0.874	0.742	0.853	0.851	0.714	0.818	0.817	0.665
Background Wall	**0.967**	**0.962**	0.778	0.967	0.962	0.778	0.964	0.960	0.773	0.955	0.954	0.792	0.941	0.940	**0.816**

Fig. 7.8 shows the enhancement in per-image IoU as the threshold values that are applied to the pixels' confidence scores increase. This is clearer in the indoor dataset than the CamVid dataset as the indoor dataset has more undefined pixels, consequently, more 'Reject' pixels. Applying the appropriate threshold value with respect to the application helps to reduce the impact of fuzzy pixels, resulting in better system performance.

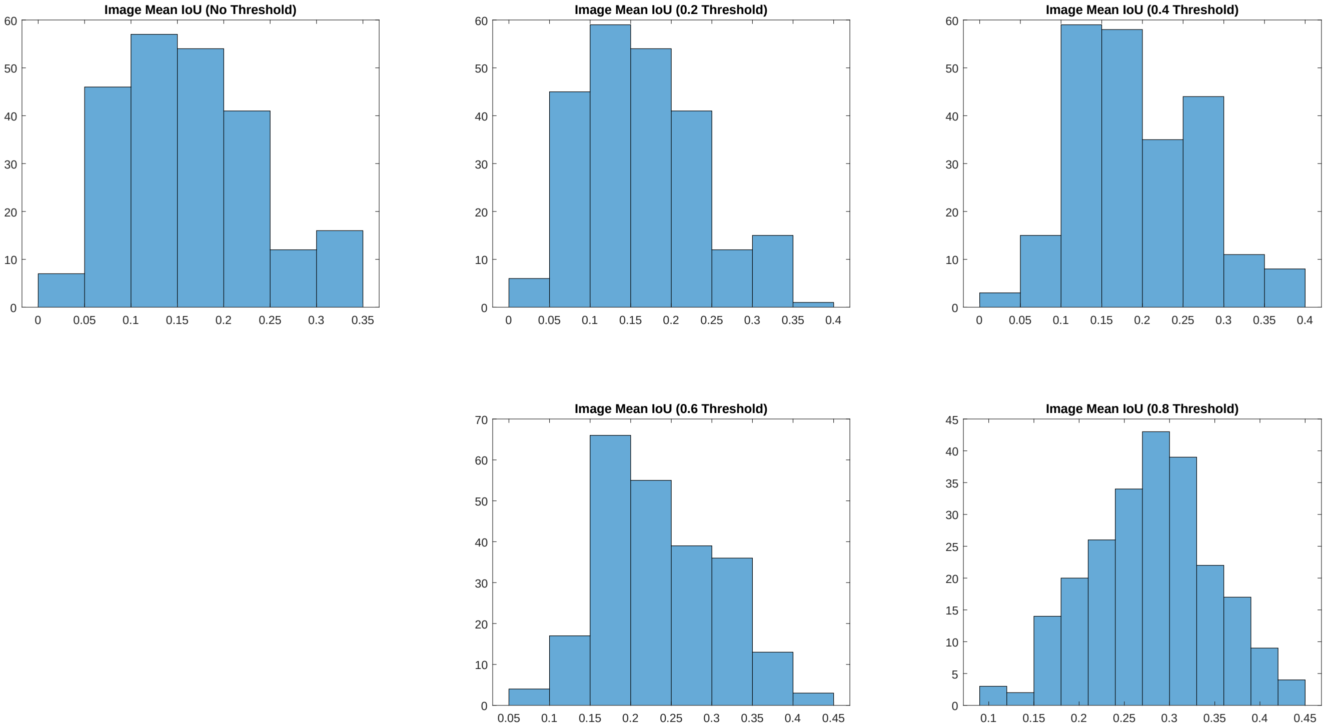


Fig. 7.8 **Histogram of per-image IoU for the indoor test set at different threshold values.** The x-axis represents the IoU value and the y-axis represents the number of images.

The indoor SS system's behaviour, in terms of performance, on different size and pixels distribution objects is very similar to the CamVid one. It can be concluded from the experiments on both datasets that low or no pixels' threshold values result in high performance on large size objects, especially stuff classes. In contrast, moderate and high pixels' threshold values enhance the system's ability to capture medium, small, and tiny objects with accurate boundaries and better alignment of segments.

Similar to Fig. 7.7, Fig. 7.9 shows how the undefined pixels, that usually have the lowest confidence scores and belong to objects of non-interest such as the cupboard, can be excluded

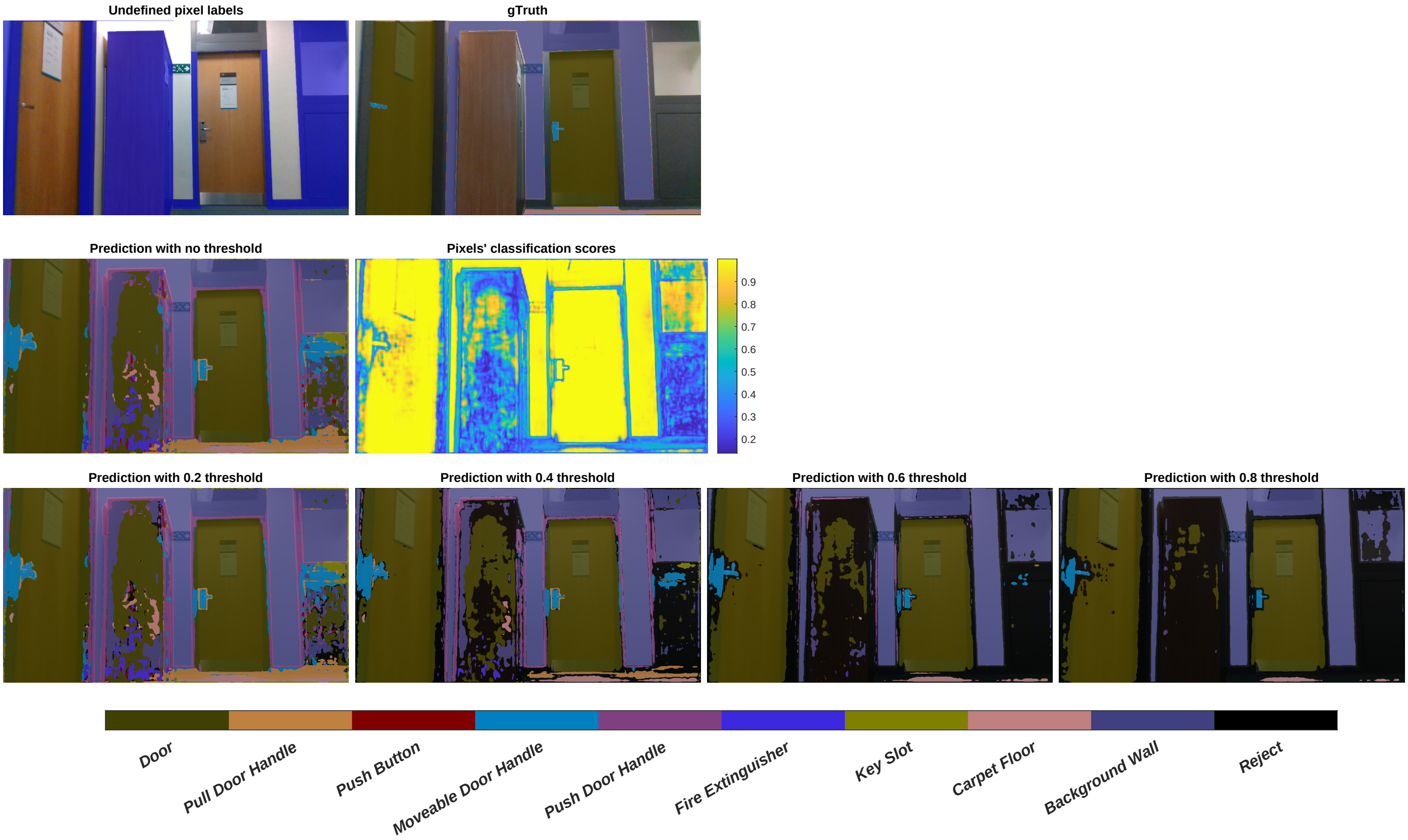


Fig. 7.9 **Qualitative results of the proposed thresholding technique on the indoor dataset.**

from the scene and the evaluation metrics calculations using high threshold values. This can help the SS system to concentrate on objects of interest that are important for the application. Consequently, better human-system interaction can be achieved. It can also be seen that the top right corner of the image has a part of the wall, which was not annotated (undefined), yet the robust system was able to classify it correctly and has not assigned its pixels to the reject class even at high threshold values.

7.5 Conclusion

Pixels are the main building blocks of any image. Thus, we believe their confidence scores should contribute to the evaluation metrics. Nevertheless, standard evaluation metrics have overlooked pixels' scores in the evaluation of SS tasks. A novel technique that incorporates pixels' confidence scores in the evaluation process of semantic segmentation systems is presented. The proposed thresholding technique has been applied to many statistical problems, which signifies its efficiency. However, its incorporation with SS evaluation metrics adds a further dimension to the standard metrics.

Results have shown the high potential of the thresholding technique as it helps to suppress fuzzy pixels that do not belong to any classes of interest and emerge pixels that belong to classes of importance to the application. Furthermore, it can be concluded from the results that optimising the SS systems on large size objects (stuff classes) can be achieved using no or low pixels' threshold values. At the same time, the SS systems' performance on medium, small and tiny objects can be boosted using high pixels' threshold values. The future work will investigate the trade-off between correctly rejected and incorrectly rejected pixels.

Chapter 8

Conclusion

8.1 Concluding comments

This chapter concludes the research and discusses the findings. The adaptation of deep learning technologies in computer vision applications has introduced many challenges and raised many questions. The inability of deep learning solutions to generalise on never seen before data is an active research area. A deep learning system trained to classify images from a certain data distribution will fail to classify images from a different data distribution. Another challenging topic is the transparency of these systems. The robustness of deep learning systems in terms of their ability to explain the main motive behind the model's predictions is an important research area. Failing to explain the model's behaviour may delay the adaption of these technologies in real-life applications due to regulator's requirements, especially in sensitive research areas, such as medical applications, where there should be no tolerance for errors.

This thesis proposes deep learning-based computer vision solutions to process images from different data distributions using shared models. The proposed systems do not require retaining and can simultaneously achieve adequate performance on different distribution data.

In addition, the proposed techniques to visualise model predictions can help to understand the model's behaviour and give insights into the operation of the black box systems. This can accelerate the approval of such systems in real-life applications.

Understanding the impact of vibrations on the smart computer vision systems is a key step to designing robust systems. Moreover, it helps to mitigate the negative impact of those vibrations. Consequently, powered wheelchair systems can be enhanced and adapted to reduce the impact of undesirable vibrations.

The main findings of this thesis are presented in section 8.2. Also, the future work is highlighted in section 8.3.

8.2 Main contributions and research findings

The answers to the research question raised in the introduction chapter are:

- *How do the objects of a dataset, in terms of size and distribution, impact the trained deep learning models? Given the wide variety of deep learning-based detection systems, how can we choose the best detector for a given application?*

We found that objects of interest to the application are missing. Standard datasets contain general objects, such as a door handle class. However, in the proposed dataset, more specialised small size objects are introduced. For example, the proposed dataset contains three different types of door handles: moveable, pull, and push door handles. Distinguishing between different kinds of door handles is vital for many applications, as each door handle requires a different kind of manipulation. Consequently, collecting and annotating application-specific datasets is essential to the proposed application

Chapter 5 investigates deep learning-based detection systems, where a roadmap to choose the optimal detector for a given application is proposed. We studied the impact of the base network, feature extraction layers, and the number of anchor boxes on the detector performance.

The study highlighted important findings, such as increasing the number of anchor boxes could negatively impact the performance of the detection system. In contrast, using data augmentation techniques can produce robust and accurate systems.

The choice of the feature extraction layer is another important observation in the study. In conclusion, we found that early feature extraction layers (up the network) help to detect small size objects but produce weak encoded features for large size objects. In contrast, late layers (down the network) can better encode the features of large size objects, but small size object features are lost due to the successive down-sampling of feature maps through the network's layers. Multi-head detectors, where the detector makes predictions at different feature extraction layers, produce the best results on datasets with multi-size objects.

- *In terms of the applicability of these systems, can we avoid long training times spent by computer vision systems based on DL? Can a DL system process images from two different distributions simultaneously (e.g. large and small size objects, or indoor and outdoor environment objects)? What are the necessary modifications that can achieve this purpose?*

Chapter 6 proposes architecture modifications to semantic segmentation systems to enable them to process images from two different distributions. The main advantage of the proposed shared systems is manifested in their needless for retraining. This allows the usage of existing and user-customised systems simultaneously, which can save long training and validation times. Several architectural designs have been proposed, where each design produces a shared system with advantages and disadvantages. The system choice is based on the application, as some applications require high processing speeds while others require high accuracy.

In conclusion, systems that can process images from one distribution have achieved the highest performance in terms of accuracy and speed. However, shared systems that can process images from different distributions have achieved comparable accuracy and adequate processing speed. Nevertheless, shared systems do not need retraining and can process images from different distributions.

- *What are the challenges in adopting such technologies in real-life scenarios? How reliable are the proposed systems? What level of system transparency can be attained? For example, could explainable AI (XAI) offer levels of transparency and decision justification required in realistic implementations?*

In chapter 4, different visualisation methods for CNNs are explored to understand the model's predictions. Unlike conventional classification methods, where the output can be mapped to a particular input, the outputs of deep learning models can depend on several unrelated hidden units. The inability to understand the black box behaviour can negatively impact the robustness and question the reliability of deep learning systems.

Thus, two visualisation methods have been proposed (WS-Grad and Concat-Grad). The heatmaps of the proposed techniques can achieve two important characteristics: 1) they can visualise the fine-grained details of the target object. 2) they can localise the discriminative regions of the target object. Only one of these characteristics can be achieved using state-of-the-art visualisation methods.

In conclusion, the proposed techniques greatly contribute to this research area. Visualising and understanding the model's predictions and what features contribute to the model's decisions can increase the trust in the black box systems and boost the approval process of such systems in real-life applications.

- *What level of environmental understanding can AI-based computer vision systems achieve, i.e. could pixel classification be achieved, and could semantic information be extracted from arbitrary video sequences? Are there available semantic segmentation evaluation metrics accurately assessing system performance? Could pixels' confidence scores enhance the validity of the evaluation process and provide better insights into the operating characteristics of the assessed systems?*

Pixels' confidence scores carry useful information regarding the concerned object. However, standard metrics do not incorporate them in the evaluation process. We can better understand and analyse the system performance by incorporating pixels' confidence scores in the evaluation process. Also, calculating standard metrics at different pixels threshold gives a better understanding of the system behaviour under different conditions.

In conclusion, experiments show that rejecting low score pixels can result in accurate prediction with sharp object borders. Moreover, it better reflects the semantic segmentation system's performance. Furthermore, different threshold values applied to the pixels' confidence scores to reject uncertainty pixels allow the optimisation of the SS system on different size objects.

- *Can a smart computer vision system based on deep learning help visually impaired users to navigate safely? What level of assistance can it offer?*

In chapter 6, semantic segmentation systems based on DL that can classify every pixel in the captured images are introduced. These systems have been deployed on a powered wheelchair for practical use and evaluation. The proposed systems illustrate efficiency in providing environmental cues and distance to target objects. With the system, users could accurately approach target objects and understand their surroundings. An important step for human-system interaction that we believe the proposed system can satisfy.

Another advantage of the proposed system is that it does not take control from the powered wheelchair user. It is a non-intrusive system that guides the user without overriding the user's commands. This is a very important and recommended approach by disabled users and clinics, as users like to be in full control of their wheelchairs. Thus, the proposed system acts as a guide.

We can conclude that the proposed systems can offer levels of assistance to visually impaired disabled users. They can also help to understand the surroundings and increase the user's independence. The proposed system helped the users to estimate more accurately the distance to the target objects with a relative error of 19.8% and 18.4% for the conditions of a) semi-neglect and b) short-sightedness, respectively, compared to errors of 47.8% and 5.6% without the SS system.

- *What performance degradation should we expect in a deep learning-based computer vision system installed on a mobile platform (e.g. a powered wheelchair) due to mechanical vibrations caused by different types of terrains traversed in everyday use?*

Chapter 6 concludes that mechanical vibrations can impact SS systems installed on powered wheelchairs. The experiments show that the detection accuracy of the semantic segmentation system has been impacted negatively due to mechanical vibrations. Our results indicated a deterioration of 4% in the performance due to these vibrations. This observation should be considered when a computer vision system is designed to be used in an environment that can introduce undesirable vibrations to the computer vision capturing system.

8.3 Recommendations for future work

Adding more objects and expanding the categories of the proposed dataset is an important step that can enhance the proposed systems' accuracy and comprehensiveness. Also, other shared architectures that can process multi-distribution data with high accuracy and speed are going to be explored. On the other hand, the behaviour of the proposed semantic segmentation evaluation technique needs adaptation at high threshold values to avoid rejecting pixels incorrectly.

Another important topic that needs further investigation is the robustness of the proposed explanation systems. Applying sanity checks on the proposed visualisation techniques can help to assess their reliability under various conditions.

Lastly, the used powered wheelchair to investigate the impact of mechanical vibrations on the smart computer vision systems does not have a suspension system. Powered wheelchairs suspension systems aim to reduce the impact of vibrations induced while driving the powered wheelchair on uneven terrains. However, it may also amplify the vibrations. An interesting topic that needs further investigation and reflection on the performance of systems installed on powered wheelchairs without suspension systems.

References

[1] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019.

[2] Kang Tong, Yiquan Wu, and Fei Zhou. Recent advances in small object detection based on deep learning: A review. *Image and Vision Computing*, 97:103910, 2020.

[3] Yanming Guo, Yu Liu, Theodoros Georgiou, and Michael S Lew. A review of semantic segmentation using deep neural networks. *International journal of multimedia information retrieval*, 7(2):87–93, 2018.

[4] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.

[5] Sampo Kuutti, Richard Bowden, Yaochu Jin, Phil Barber, and Saber Fallah. A survey of deep learning applications to autonomous vehicle control. *IEEE Transactions on Intelligent Transportation Systems*, 22(2):712–733, 2020.

[6] Massimo Bertolini, Davide Mezzogori, Mattia Neroni, and Francesco Zammori. Machine learning for industrial applications: A comprehensive literature review. *Expert Systems with Applications*, 175:114820, 2021.

[7] Ruhul Amin Khalil, Nasir Saeed, Mudassir Masood, Yasaman Moradi Fard, Mohamed-Slim Alouini, and Tareq Y Al-Naffouri. Deep learning in the industrial internet of things: Potentials, challenges, and emerging applications. *IEEE Internet of Things Journal*, 8(14):11016–11040, 2021.

[8] Jinjiang Wang, Yulin Ma, Laibin Zhang, Robert X Gao, and Dazhong Wu. Deep learning for smart manufacturing: Methods and applications. *Journal of manufacturing systems*, 48:144–156, 2018.

[9] Lei Cai, Jingyang Gao, and Di Zhao. A review of the application of deep learning in medical image classification and segmentation. *Annals of Translational Medicine*, 8(11), 2020.

[10] Anabia Sohail, Asifullah Khan, Noorul Wahab, Aneela Zameer, and Saranjam Khan. A multi-phase deep cnn based mitosis detection framework for breast cancer histopathological images. *Scientific Reports*, 11(1):1–18, 2021.

[11] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature medicine*, 25(1):24–29, 2019.

[12] Luis Bote-Curiel, Sergio Munoz-Romero, Alicia Gerrero-Curieses, and José Luis Rojo-Álvarez. Deep learning and big data in healthcare: A double review for critical beginners. *Applied Sciences*, 9(11):2331, 2019.

[13] Kailun Yang, Luis M Bergasa, Eduardo Romera, Ruiqi Cheng, Tianxue Chen, and Kaiwei Wang. Unifying terrain awareness through real-time semantic segmentation. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1033–1038. IEEE, 2018.

[14] Sachin Mehta, Hannaneh Hajishirzi, and Linda Shapiro. Identifying most walkable direction for navigation in an outdoor environment. *arXiv preprint arXiv:1711.08040*, 2017.

[15] Linda Fehr, W Edwin Langbein, and Steven B Skaar. Adequacy of power wheelchair control interfaces for persons with severe disabilities: A clinical survey. *Journal of rehabilitation research and development*, 37(3):353–360, 2000.

[16] Lilly Jensen. User perspectives on assistive technology: a qualitative analysis of 55 letters from citizens applying for assistive technology. *World Federation of Occupational Therapists Bulletin*, 69(1):42–45, 2014.

[17] Yoshio Matsumotot, Tomoyuki Ino, and Tsukasa Ogsawara. Development of intelligent wheelchair system with face and gaze based interface. In *Proceedings 10th IEEE International Workshop on Robot and Human Interactive Communication. ROMAN 2001 (Cat. No. 01TH8591)*, pages 262–267. IEEE, 2001.

[18] George Constantin Rascanu and Razvan Solea. Electric wheelchair control for people with locomotor disabilities using eye movements. In *15th International Conference on System Theory, Control and Computing*, pages 1–5. IEEE, 2011.

[19] Prateek Arora, Anshul Sharma, Anmoal Singh Soni, and Aman Garg. Control of wheelchair dummy for differently abled patients via iris movement using image processing in matlab. In *2015 Annual IEEE India Conference (INDICON)*, pages 1–4. IEEE, 2015.

[20] Martin Henderson, Stephen Kelly, Robert Horne, Michael Gillham, Matthew Pepper, and Jean-Marc Capron. Powered wheelchair platform for assistive technology development. In *2014 Fifth International Conference on Emerging Security Technologies*, pages 52–56. IEEE, 2014.

[21] P Viswanathan, J Little, AK Mackworth, and A Mihailidis. Adaptive navigation assistance for visually-impaired wheelchair users. In *Proceedings of the IROS 2011 Workshop on New and Emerging Technologies in Assistive Robotics*. Citeseer, 2011.

[22] Jesse Leaman and Hung Manh La. A comprehensive review of smart wheelchairs: past, present, and future. *IEEE Transactions on Human-Machine Systems*, 47(4):486–499, 2017.

[23] Phuoc Pham, Duy Nguyen, Tien Do, Thanh Duc Ngo, and Duy-Dinh Le. Evaluation of deep models for real-time small object detection. In *International conference on neural information processing*, pages 516–526. Springer, 2017.

[24] J-D Yoder, Eric T Baumgartner, and Steven B Skaar. Initial results in the development of a guidance system for a powered wheelchair. *IEEE Transactions on Rehabilitation Engineering*, 4(3):143–151, 1996.

[25] Pooja Viswanathan, Jennifer Boger, Jesse Hoey, and Alex Mihailidis. A comparison of stereovision and infrared as sensors for an anti-collision powered wheelchair for older adults with cognitive impairments. In *2nd International Conference on Technology and Aging, Toronto*. Citeseer, 2007.

[26] Sotirios Chatzidimitriadis, Paul Oprea, Michael Gillham, and Konstantinos Sirlantzis. Evaluation of 3d obstacle avoidance algorithm for smart powered wheelchairs. In *2017 seventh international conference on emerging security technologies (EST)*, pages 157–162. IEEE, 2017.

[27] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[28] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer, 2014.

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[30] Navteen Dalal, Bill Triggs, et al. Object detection using histograms of oriented gradients. In *Pascal VOC Workshop, ECCV*, 2006.

[31] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.

[32] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. Ieee, 2001.

[33] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998.

[34] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

[35] 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.

[36] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[37] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[38] Lorien Y Pratt et al. Discriminability-based transfer between neural networks. *Advances in neural information processing systems*, pages 204–204, 1993.

[39] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014.

[40] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[41] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[42] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[43] Karel Lenc and Andrea Vedaldi. R-cnn minus r. *arXiv preprint arXiv:1506.06981*, 2015.

[44] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[45] 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.

[46] Junjie Yan, Zhen Lei, Longyin Wen, and Stan Z Li. The fastest deformable part model for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2497–2504, 2014.

[47] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[48] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[49] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[50] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[51] 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.

[52] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[53] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018.

[54] Upesh Nepal and Hossein Eslamiat. Comparing yolov3, yolov4 and yolov5 for autonomous landing spot detection in faulty uavs. *Sensors*, 22(2):464, 2022.

[55] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

[56] Xiang Long, Kaipeng Deng, Guanzhong Wang, Yang Zhang, Qingqing Dang, Yuan Gao, Hui Shen, Jianguo Ren, Shumin Han, Errui Ding, et al. Pp-yolo: An effective and efficient implementation of object detector. *arXiv preprint arXiv:2007.12099*, 2020.

[57] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[58] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.

[59] Zhiqiang Shen, Zhuang Liu, Jianguo Li, Yu-Gang Jiang, Yurong Chen, and Xiangyang Xue. Dsod: Learning deeply supervised object detectors from scratch. In *Proceedings of the IEEE international conference on computer vision*, pages 1919–1927, 2017.

[60] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[61] Chenyi Chen, Ming-Yu Liu, Oncel Tuzel, and Jianxiong Xiao. R-cnn for small object detection. In *Asian conference on computer vision*, pages 214–230. Springer, 2016.

[62] Zhe Zhu, Dun Liang, Songhai Zhang, Xiaolei Huang, Baoli Li, and Shimin Hu. Traffic-sign detection and classification in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2110–2118, 2016.

[63] Nhat-Duy Nguyen, Tien Do, Thanh Duc Ngo, and Duy-Dinh Le. An evaluation of deep learning methods for small object detection. *Journal of Electrical and Computer Engineering*, 2020, 2020.

[64] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

[65] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[66] 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.

[67] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

[68] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[69] Bharat Singh and Larry S Davis. An analysis of scale invariance in object detection - snip. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3578–3587, 2018.

[70] Sean Bell, C Lawrence Zitnick, Kavita Bala, and Ross Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2874–2883, 2016.

[71] Mate Kisantal, Zbigniew Wojna, Jakub Murawski, Jacek Naruniec, and Kyunghyun Cho. Augmentation for small object detection. *arXiv preprint arXiv:1902.07296*, 2019.

[72] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *European conference on computer vision*, pages 354–370. Springer, 2016.

[73] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2129–2137, 2016.

[74] Tao Kong, Anbang Yao, Yurong Chen, and Fuchun Sun. Hypernet: Towards accurate region proposal generation and joint object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 845–853, 2016.

[75] Xingyu Zeng, Wanli Ouyang, Binh Yang, Junjie Yan, and Xiaogang Wang. Gated bi-directional cnn for object detection. In *ECCV*, 2016.

[76] Yukun Zhu, Raquel Urtasun, Ruslan Salakhutdinov, and Sanja Fidler. segdeepm: Exploiting segmentation and context in deep neural networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4703–4711, 2015.

[77] Spyros Gidaris and Nikos Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *Proceedings of the IEEE international conference on computer vision*, pages 1134–1142, 2015.

[78] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[79] Sergey Zagoruyko, Adam Lerer, Tsung-Yi Lin, Pedro O Pinheiro, Sam Gross, Soumith Chintala, and Piotr Dollár. A multipath network for object detection. *arXiv preprint arXiv:1604.02135*, 2016.

[80] Jianan Li, Xiaodan Liang, Yunchao Wei, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. Perceptual generative adversarial networks for small object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1222–1230, 2017.

[81] Yancheng Bai, Yongqiang Zhang, Mingli Ding, and Bernard Ghanem. Sod-mtgan: Small object detection via multi-task generative adversarial network. In *ECCV*, 2018.

[82] Changqing Cao, Bo Wang, Wenrui Zhang, Xiaodong Zeng, Xu Yan, Zhejun Feng, Yutao Liu, and Zengyan Wu. An improved faster r-cnn for small object detection. *IEEE Access*, 7:106838–106846, 2019.

[83] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):640–651, 2017.

[84] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[85] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2012.

[86] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014.

[87] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[88] Liang Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018.

[89] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[90] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[91] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[92] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[93] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[94] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[95] Dan Ciresan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*, pages 2843–2851, 2012.

[96] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015.

[97] Urs Gasser and Virgilio AF Almeida. A layered model for ai governance. *IEEE Internet Computing*, 21(6):58–62, 2017.

[98] Bernd W Wirtz, Jan C Weyerer, and Ines Kehl. Governance of artificial intelligence: A risk and guideline-based integrative framework. *Government Information Quarterly*, page 101685, 2022.

[99] Weiyu Wang and Keng Siau. Artificial intelligence: A study on governance, policies, and regulations. *MWAIS 2018 proceedings*, 40, 2018.

[100] Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu. Interpreting cnns via decision trees. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6261–6270, 2019.

[101] Xiaolin Zhang, Yunchao Wei, Guoliang Kang, Yi Yang, and Thomas Huang. Self-produced guidance for weakly-supervised object localization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 597–613, 2018.

[102] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[103] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[104] Matthew D Zeiler, Graham W Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *2011 International Conference on Computer Vision*, pages 2018–2025. IEEE, 2011.

[105] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016.

[106] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*. Citeseer, 2014.

[107] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[108] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*, 2017.

[109] Pieter-Jan Kindermans, Kristof T Schütt, Maximilian Alber, Klaus-Robert Müller, Dumitru Erhan, Been Kim, and Sven Dähne. Learning how to explain neural networks: Patternnet and patternattribution. *arXiv preprint arXiv:1705.05598*, 2017.

[110] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[111] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 685–694, 2015.

[112] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

[113] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

[114] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[115] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*, 2018.

[116] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[117] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[118] Chunshui Cao, Xianming Liu, Yi Yang, Yinan Yu, Jiang Wang, Zilei Wang, Yongzhen Huang, Liang Wang, Chang Huang, Wei Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2956–2964, 2015.

[119] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018.

[120] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pages 3429–3437, 2017.

[121] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. *arXiv preprint arXiv:1705.07857*, 2017.

[122] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311, 2017.

[123] Nigel W John, Serban R Pop, Thomas W Day, Panagiotis D Ritsos, and Christopher J Headleand. The implementation and validation of a virtual environment for training powered wheelchair manoeuvres. *IEEE transactions on visualization and computer graphics*, 24(5):1867–1878, 2017.

[124] Philippe S Archambault, Jodie Ng Fuk Chong, Gianluca Sorrento, François Routhier, and Patrick Boissy. Comparison of powered wheelchair driving performance in a real and in a simulated environment. In *2011 International Conference on Virtual Rehabilitation*, pages 1–7. IEEE, 2011.

[125] Matt Bailey, Andrew Chanler, Bruce Maxwell, Mark Micire, Katherine Tsui, and Holly Yanco. Development of vision-based navigation for a robotic wheelchair. In *2007 IEEE 10th International Conference on Rehabilitation Robotics*, pages 951–957. IEEE, 2007.

[126] Elhassan Mohamed, Konstantinos Sirlantzis, and Gareth Howells. A pixel-wise annotated dataset of small overlooked indoor objects for semantic segmentation applications. *Data in Brief*, 40:107791, 2022.

[127] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[128] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *arXiv preprint arXiv:1608.05442*, 2016.

[129] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 601–608, 2011.

[130] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[131] B. Hua, Q. Pham, D. T. Nguyen, M. Tran, L. Yu, and S. Yeung. Scenenn: A scene meshes dataset with annotations. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 92–101, 2016.

[132] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[133] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

[134] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4829–4837, 2016.

[135] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[136] Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. " what is relevant in a text document?": An interpretable machine learning approach. *PloS one*, 12(8):e0181142, 2017.

[137] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[138] Sebastian Lapuschkin, Alexander Binder, Grégoire Montavon, Klaus-Robert Muller, and Wojciech Samek. Analyzing classifiers: Fisher vectors and deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2912–2920, 2016.

[139] Farhad Arbabzadah, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Identifying individual facial expressions by deconstructing a neural network. In *German Conference on Pattern Recognition*, pages 344–354. Springer, 2016.

[140] Irene Sturm, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Interpretable deep neural networks for single-trial eeg classification. *Journal of neuroscience methods*, 274:141–145, 2016.

[141] Qingjie Meng, Christian Baumgartner, Matthew Sinclair, James Housden, Martin Rajchl, Alberto Gomez, Benjamin Hou, Nicolas Toussaint, Veronika Zimmer, Jeremy Tan, et al. Automatic shadow detection in 2d ultrasound images. In *Data Driven Treatment*

Response Assessment and Preterm, Perinatal, and Paediatric Image Analysis, pages 66–75. Springer, 2018.

[142] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 24–25, 2020.

[143] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[144] Martin Pelikan, David E Goldberg, Erick Cantú-Paz, et al. Boa: The bayesian optimization algorithm. In *Proceedings of the genetic and evolutionary computation conference GECCO-99*, volume 1, pages 525–532. Citeseer, 1999.

[145] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks. 2015.

[146] Jianbo Shi et al. Good features to track. In *1994 Proceedings of IEEE conference on computer vision and pattern recognition*, pages 593–600. IEEE, 1994.

[147] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[148] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[149] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

[150] Anna Carlsson and Jörgen Lundälv. Acute injuries resulting from accidents involving powered mobility devices (PMDs)—Development and outcomes of PMD-related accidents in Sweden. *Traffic Injury Prevention*, 20(5):484–491, jul 2019.

[151] Ronald P. Gaal, Nancy Rebholtz, Ralf D. Hotchkiss, and Peter F. Pfaelzer. Wheelchair rider injuries: Causes and consequences for wheelchair design and selection. *Journal of Rehabilitation Research and Development*, 1997.

[152] Wan Yin Chen, Yuh Jang, Jung Der Wang, Wen Ni Huang, Chan Chia Chang, Hui Fen Mao, and Yen Ho Wang. Wheelchair-related accidents: relationship with wheelchair-using behavior in active community wheelchair users. *Archives of Physical Medicine and Rehabilitation*, 2011.

[153] PJ Nelson, G Verburg, D Gibney, and L Korba. The smart wheelchair. a discussion of the promises and pitfalls. In *RESNA 13th Annual Conference*, pages 307–308, 1990.

[154] G. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognit. Lett.*, 30:88–97, 2009.

[155] Elhassan Mohamed, Konstantinos Sirlantzis, and Gareth Howells. Indoor/outdoor semantic segmentation using deep learning for visually impaired wheelchair users. *IEEE Access*, 9:147914–147932, 2021.

[156] Elhassan Mohamed, Konstantinos Sirlantzis, and Gareth Howells. Application of transfer learning for object detection on manually collected data. In *Proceedings of SAI Intelligent Systems Conference*, pages 919–931. Springer, 2019.

[157] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[158] Elhassan Mohamed, Konstantinos Sirlantzis, and Gareth Howells. Incorporation of rejection criterion - a novel technique for evaluating semantic segmentation systems. In *2021 14th International Conference on Human System Interaction (HSI)*, pages 1–7, 2021.

[159] Joy Goodman-Deane, Sam Waller, Alice-Catherine Collins, and John Clarkson. Simulating vision loss: what levels of impairment are actually represented? In *Contemporary Ergonomics and Human Factors 2013: Proceedings of the international conference on Ergonomics & Human Factors 2013, Cambridge, UK, 15-18 April 2013*, page 347. CRC Press, 2013.

[160] Matteo Zallio, Sam Waller, Camelia Chivaran, and John Clarkson. Visual accessibility and inclusion. an exploratory study to understand visual accessibility in the built environment. In *SMART ACCESSIBILITY 2021 The Sixth International Conference on Universal Accessibility in the Internet of Things and Smart Environments*. ThinkMind, 2021.

[161] G Nicolás Marichal, María Tomás-Rodríguez, Ángela Hernández, Salvador Castillo-Rivera, and P Campoy. Vibration reduction for vision systems on board uav using a neuro-fuzzy controller.

[162] CF Periu, A Mohsenimanesh, C Laguë, and NB McLaughlin. Isolation of vibrations transmitted to a lidar sensor mounted on an agricultural vehicle to improve obstacle detection. *Canadian Biosystems Engineering*, 55, 2013.

[163] Mechanical vibration and shock — evaluation of human exposure to whole-body vibration — part 1: General requirements. Standard, International Organization for Standardization, Geneva, CH, May 1997.

[164] Ryusuke Miyamoto, Yuta Nakamura, Miho Adachi, Takeshi Nakajima, Hiroki Ishida, Kazuya Kojima, Risako Aoki, Takuro Oki, and Shingo Kobayashi. Vision-based road-following using results of semantic segmentation for autonomous navigation. In *2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin)*, pages 174–179. IEEE, 2019.

[165] Li-Jia Li, Richard Socher, and Li Fei-Fei. Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2036–2043. IEEE, 2009.

[166] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019.

[167] Qing Tao, Gao-Wei Wu, Fei-Yue Wang, and Jue Wang. Posterior probability support vector machines for unbalanced data. *IEEE Transactions on Neural Networks*, 16(6):1561–1573, 2005.

[168] Nathaniel T Stevens and Luke Hagar. Comparative probability metrics: Using posterior probabilities to account for practical equivalence in a/b tests. *The American Statistician*, pages 1–15, 2022.

[169] Pushmeet Kohli, Philip HS Torr, et al. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 82(3):302–324, 2009.

[170] C. J. V. Rijsbergen. Information retrieval. In *Encyclopedia of GIS*, 1979.

[171] Ikram E Abdou and William K Pratt. Quantitative design and evaluation of enhancement/thresholding edge detectors. *Proceedings of the IEEE*, 67(5):753–763, 1979.

[172] David R Martin, Charless C Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence*, 26(5):530–549, 2004.

[173] Gabriela Csurka, Diane Larlus, Florent Perronnin, and France Meylan. What is a good evaluation measure for semantic segmentation?. In *BMVC*, volume 27, pages 10–5244, 2013.

[174] Qian Huang and Byron Dom. Quantitative methods of evaluating image segmentation. In *Proceedings., international conference on image processing*, volume 3, pages 53–56. IEEE, 1995.

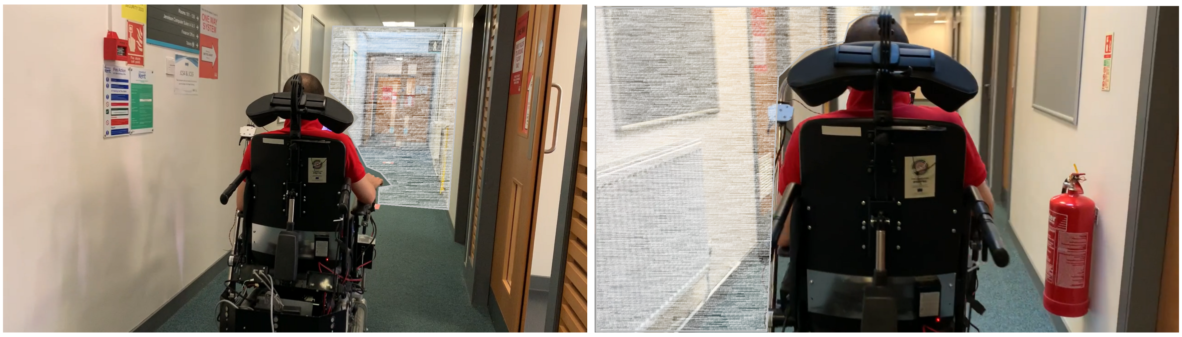
[175] J. Freixenet, X. Muñoz, D. Raba, J. Martı, and X. Cufí. Yet another survey on image segmentation: Region and boundary information integration. In *ECCV*, 2002.

[176] Yuxiang Zhang, Sachin Mehta, and Anat Caspi. Rethinking semantic segmentation evaluation for explainability and model selection. *arXiv preprint arXiv:2101.08418*, 2021.

[177] Mechanical vibration — laboratory method for evaluating vehicle seat vibration — part 1: Basic requirements. Standard, International Organization for Standardization, Geneva, CH, 2016.

Appendix A

Semantic Segmentation Supplementary Material



(a) Short-sightedness

(b) Semi-neglect

Fig. A.1 **Visually impaired users.** Illustrated by the clouded areas, short-sightedness users cannot see far object (a), while semi-neglect users cannot see half of the scene (b).

Table A.1 **Per-class metrics of the indoor system using FCN-8s on the test set.**

Metrics Classes	Accuracy	IoU	Mean BF Score
Door	0.981	0.979	0.852
PullDoorHandle	0.582	0.159	0.623
PushButton	0.764	0.238	0.631
MoveableDoorHandle	0.780	0.616	0.492
PushDoorHandle	0.622	0.090	0.350
FireExtinguisher	0.897	0.853	0.481
KeySlot	0.722	0.205	0.677
CarpetFloor	0.917	0.883	0.741
BackgroundWall	0.945	0.941	0.647

Table A.2 **Per-class metrics of the indoor system using FCN-16s on the test set.**

Metrics Classes	Accuracy	IoU	Mean BF Score
Door	0.980	0.979	0.843
PullDoorHandle	0.587	0.128	0.579
PushButton	0.757	0.269	0.540
MoveableDoorHandle	0.748	0.624	0.483
PushDoorHandle	0.598	0.078	0.306
FireExtinguisher	0.896	0.862	0.612
KeySlot	0.638	0.184	0.675
CarpetFloor	0.917	0.881	0.760
BackgroundWall	0.940	0.938	0.824

Table A.3 **Per-class metrics of the indoor system using FCN-32s on the test set.**

Metrics Classes	Accuracy	IoU	Mean BF Score
Door	0.977	0.977	0.827
PullDoorHandle	0.491	0.123	0.391
PushButton	0.713	0.238	0.425
MoveableDoorHandle	0.795	0.505	0.469
PushDoorHandle	0.596	0.049	0.348
FireExtinguisher	0.903	0.851	0.804
KeySlot	0.506	0.279	0.497
CarpetFloor	0.918	0.893	0.753
BackgroundWall	0.916	0.910	0.800

Table A.4 **Per-class metrics of the indoor system using SegNet with VGG-16 on the test set.**

Metrics / Classes	Accuracy	IoU	Mean BF Score
Door	0.975	0.975	0.830
PullDoorHandle	0.559	0.153	0.642
PushButton	0.751	0.277	0.690
MoveableDoorHandle	0.774	0.614	0.510
PushDoorHandle	0.705	0.075	0.493
FireExtinguisher	0.907	0.839	0.397
KeySlot	0.635	0.190	0.662
CarpetFloor	0.906	0.888	0.659
BackgroundWall	0.947	0.938	0.650

Table A.5 **Per-class metrics of the indoor system using SegNet with VGG-19 on the test set.**

Metrics / Classes	Accuracy	IoU	Mean BF Score
Door	0.977	0.976	0.827
PullDoorHandle	0.542	0.149	0.639
PushButton	0.774	0.201	0.713
MoveableDoorHandle	0.787	0.570	0.576
PushDoorHandle	0.676	0.075	0.322
FireExtinguisher	0.907	0.802	0.368
KeySlot	0.602	0.172	0.696
CarpetFloor	0.894	0.873	0.659
BackgroundWall	0.932	0.927	0.668

Table A.6 **Per-class metrics of the indoor system using U-Net on the test set.**

Metrics / Classes	Accuracy	IoU	Mean BF Score
Door	0.787	0.758	0.603
PullDoorHandle	0.256	0.062	0.267
PushButton	0.092	0.017	0.244
MoveableDoorHandle	0.281	0.140	0.219
PushDoorHandle	0.328	0.043	0.380
FireExtinguisher	0.691	0.591	0.304
KeySlot	0.279	0.039	0.307
CarpetFloor	0.945	0.467	0.579
BackgroundWall	0.877	0.701	0.432

Table A.7 **Per-class metrics of the indoor system using DLV3+ with ResNet-50 on the test set.**

Metrics Classes	Accuracy	IoU	Mean BF Score
Door	0.974	0.974	0.840
PullDoorHandle	0.555	0.102	0.437
PushButton	0.821	0.234	0.609
MoveableDoorHandle	0.783	0.661	0.460
PushDoorHandle	0.653	0.102	0.276
FireExtinguisher	0.855	0.846	0.582
KeySlot	0.562	0.278	0.557
CarpetFloor	0.915	0.892	0.765
BackgroundWall	0.971	0.965	0.768

Table A.8 **Per-class metrics of the indoor system using DLV3+ with Xception on the test set .**

Metrics Classes	Accuracy	IoU	Mean BF Score
Door	0.979	0.979	0.853
PullDoorHandle	0.497	0.200	0.467
PushButton	0.759	0.261	0.630
MoveableDoorHandle	0.788	0.617	0.447
PushDoorHandle	0.621	0.0818	0.258
FireExtinguisher	0.932	0.883	0.612
KeySlot	0.843	0.172	0.534
CarpetFloor	0.890	0.886	0.749
BackgroundWall	0.961	0.958	0.752

Appendix B

Vibration Impact on User's Health and Comfort

B.1 Introduction

Seated individuals who are exposed to whole-body vibrations for a long period of time are at the risk of injury [163]. Electrical powered wheelchair (EPW) users can fit into this category as they drive for a prolonged period of time in dynamic environments, exposing themselves to whole-body vibrations. There is an increase in health risk to the lumbar spine and the connected nervous system because of the long-term high-intensity whole-body vibrations [163]. This risk can be attributed to the biodynamics behaviour of the spine: horizontal displacement and torsion of the segments of the vertebral column. According to the ISO-2631 standard [163], the digestive system, the genital/urinary system, and the female reproductive organs are also impacted but with lower probability. Moreover, the health risks are likely to increase when the duration and the vibration intensity increase, while rest periods can reduce the risk.

B.2 Vibration impact on user’s health and comfort

B.2.1 Methodology

System installation

Acceleration should be measured at the points from which the vibration is considered to enter the human body (Fig. B.1). The considered frequencies for quantifying the impact of vibrations on health, comfort, and vibration perception are in the range of 0.5 to 80 Hz. However, the considered frequencies for motion sickness are in the range of 0.1 to 0.5 Hz [163].

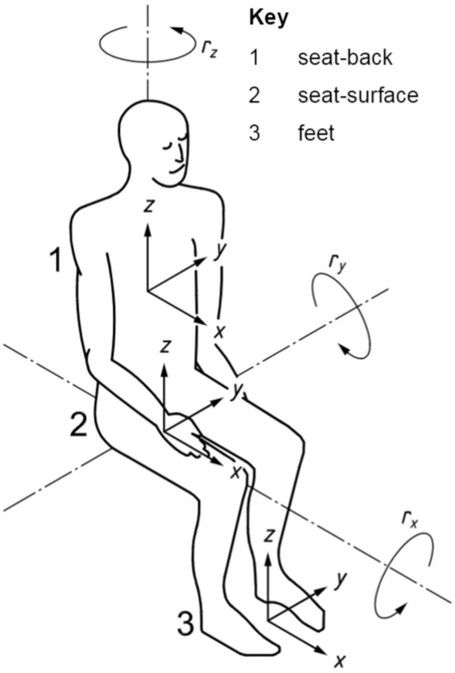
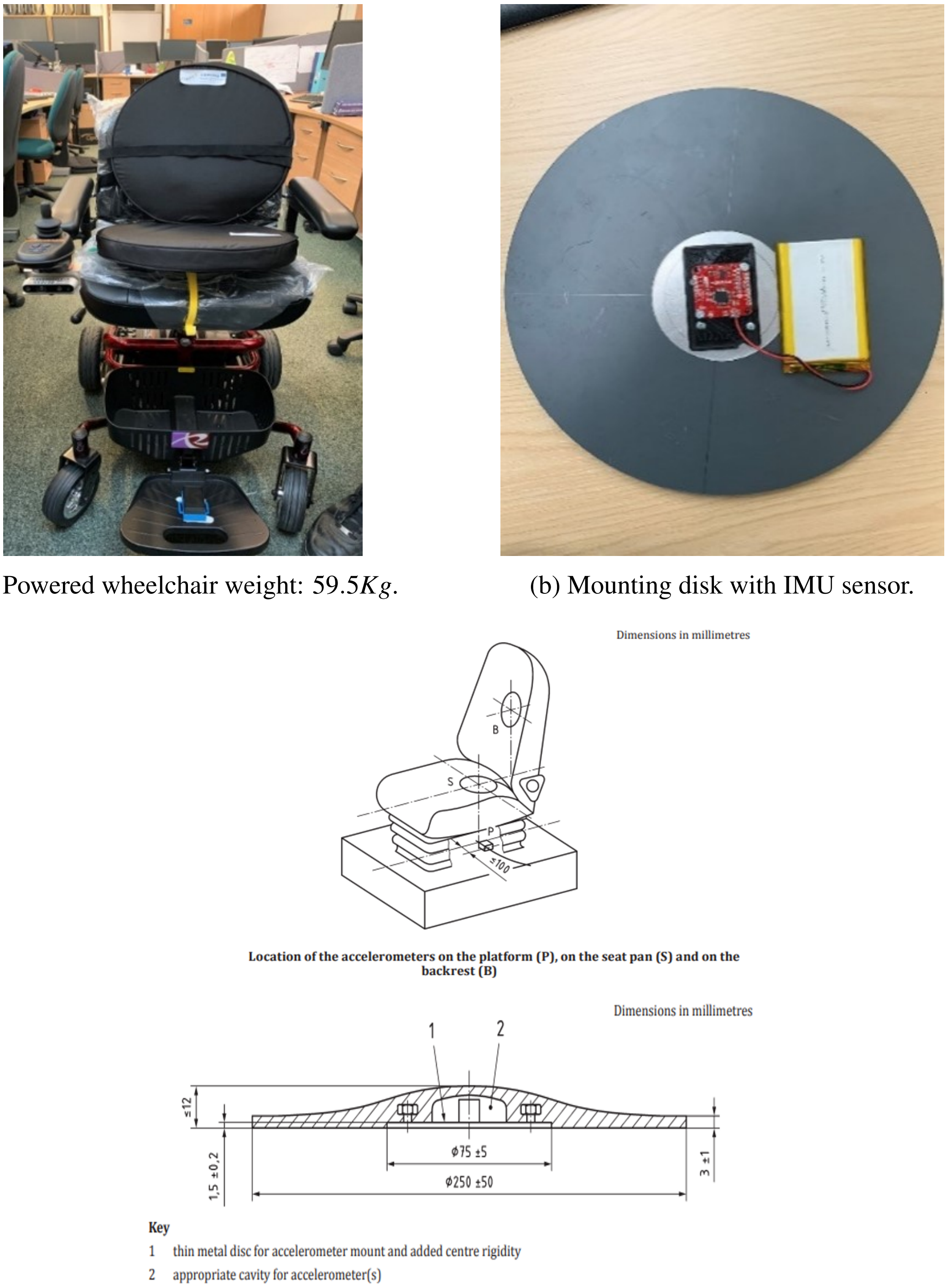


Fig. B.1 **Points of measuring the vibrations according to the ISO-2631 standard [163].**

Sensors should be located at the interface between the human and the source of vibration. This study considered three areas of contact for a seated person to quantify the vibration as they represent the points from which the vibrations enter the human body: seat surface ‘pan’ z-axis, seatback x-axis and feet z-axis (Fig. B.1 and Fig. B.2). Following the ISO standard 10326

[177], an extra sensor is installed on the powered wheelchair chassis/battery for referencing (Fig. B.2c).


Dimensions in millimetres
B
S
P
≤100
Location of the accelerometers on the platform (P), on the seat pan (S) and on the backrest (B)
Dimensions in millimetres
1
2
≤12
1,5 ±0,2
ϕ75 ±5
ϕ250 ±50
3 ±1
Key
1 thin metal disc for accelerometer mount and added centre rigidity
2 appropriate cavity for accelerometer(s)

(a) Powered wheelchair weight: 59.5Kg.

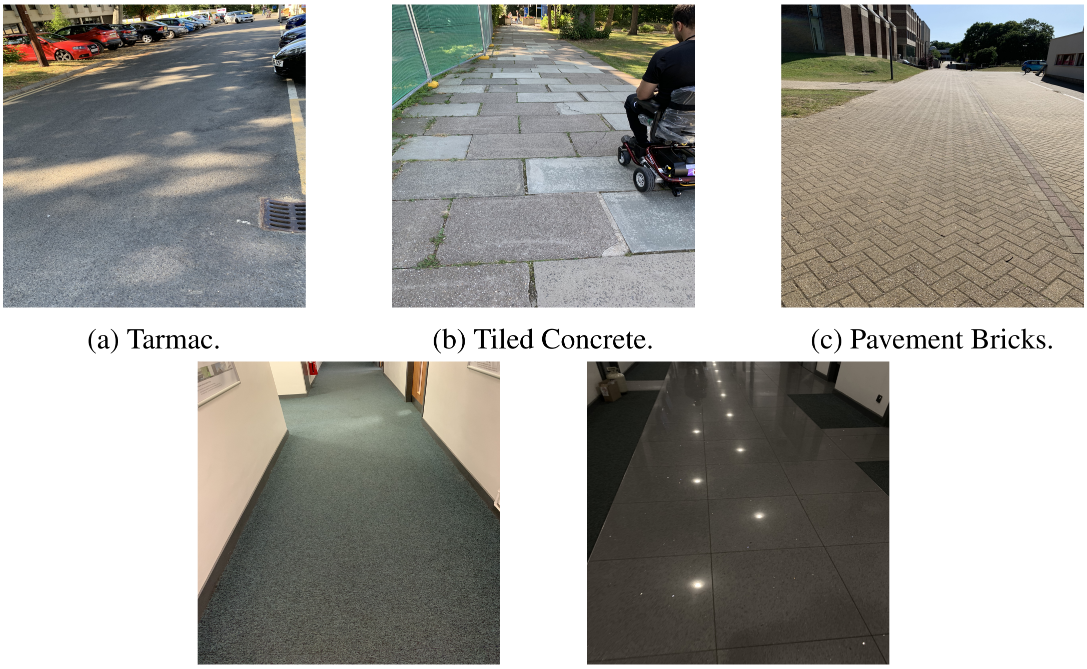
(b) Mounting disk with IMU sensor.

(c) ISO 10326-1 for mounting disk with sensor and location of installation.

Fig. B.2 **System installation for data collection.**

For data collection, the sensors are installed on the Roma Reno II Power Chair (Fig. B.2a). Fig. B.2b show the mounting disk with the IMU (Inertial Measurement Unit) sensor.

Five different types of terrains are trialled during the data collection (Fig. B.3). Outdoor types are tarmac, tiled concert and pavement bricks, whereas indoor types are carpet floor and tiled floor.



(a) Tarmac.

(b) Tiled Concrete.

(c) Pavement Bricks.

(d) Carpet Flooring.

(e) Tiled flooring.

Fig. B.3 **Terrain types.**

Table B.1 shows the users' weights, heights, average speeds, and completion times.

Table B.1 **User's information.**

User **Information**	**User 1**	**User 2**	**User 3**	**User 4**	**Mean/Std**
Weight (*kg*)	48	78	94	117	84.2 / 28.9
Height (*cm*)	160	179	184	183	176.5 / 11.2
Track time (*min*)	3.4	3.5	3.7	3.6	3.5 / 0.12
Average speed (Km/h)	22.4	22.4	21.1	21.2	21.7 / 0.72

Evaluation

Three evaluation metrics are presented: weighted root mean square acceleration $a_w(m/s^2)$ (B.1), Maximum Transient Vibration Value ($MTVV$) (B.2), and estimated Vibration Dose Value ($eVDV$) (B.3).

$$a_w = \left(\frac{1}{T} \int_0^T a_w^2(t)dt \right) \Bigg)^{\frac{1}{2}} \tag{B.1}$$

$$a_w(t) : weighted\ acceleration\ (m/s^2)$$

$$T : duration\ of\ the\ measurement\ (s)$$

$MTVV$ represents the highest magnitude of $a_w(t_0)$ during the measurement period (T).

$$MTVV = \max\left[a_w\left(t_0\right)\right] \tag{B.2}$$

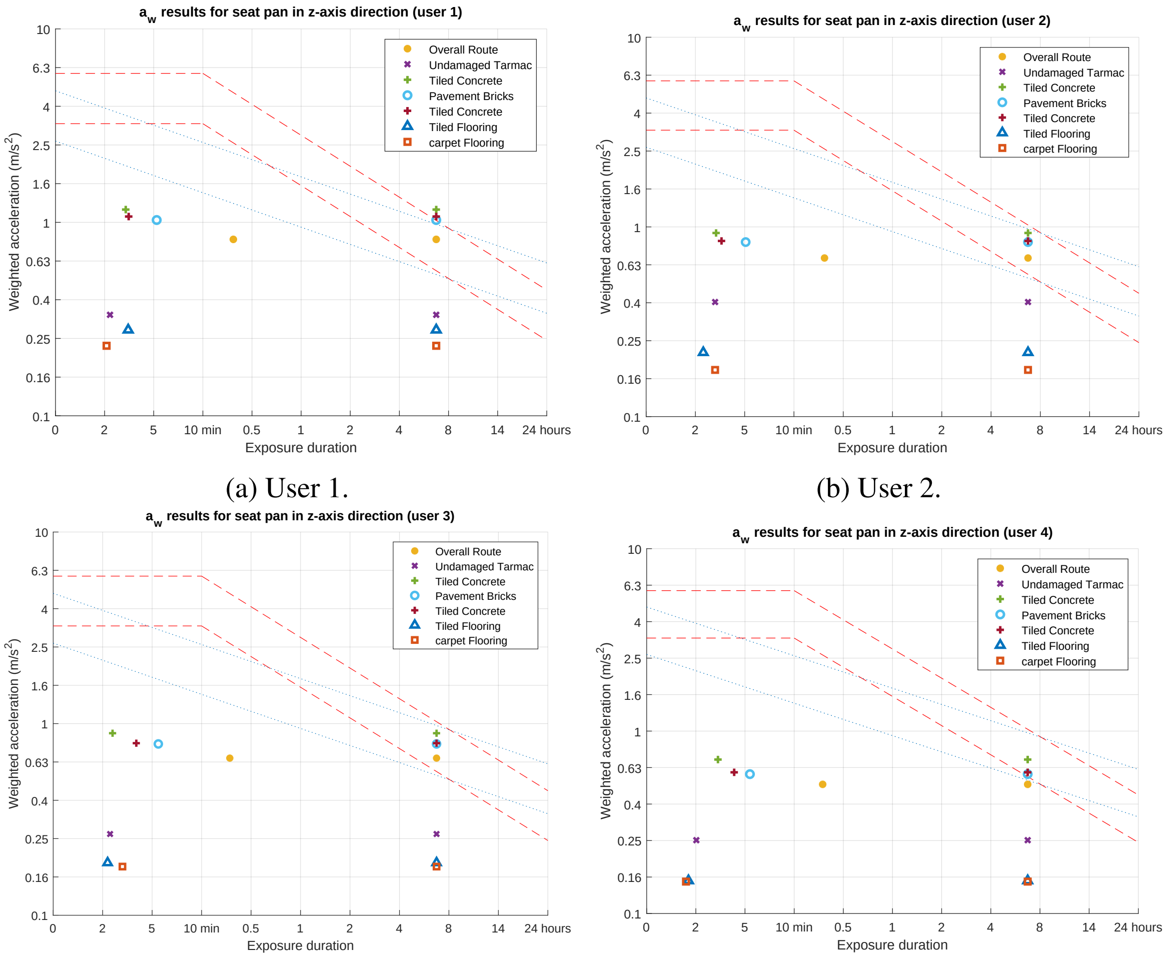
Estimated Vibration Dose Value is more sensitive to peaks than basic evaluation methods.

$$eVDV = 1.4a_wT^{\frac{1}{4}} \tag{B.3}$$

B.2.2 Results

Vibration effect on health

Following the ISO-2631 standard [163], a health guidance caution zone is indicated by the dashed red and the dotted blue lines that are shown in Fig. B.4. Below the zone, the health effects due to vibrations have not been documented. Inside the zone, caution with respect to health risks due to vibration is indicated. The health risk is likely above the zone. These recommendations are based on exposures to vibrations in the range of 4 to 8 *hours* [163]



(a) User 1.

(b) User 2.

(c) User 3.

(d) User 4.

Fig. B.4 **RMS frequency weighted acceleration for seat pan in z-axis direction.**

Table B.2 shows the results for the five different terrains for the four users. Terrain surfaces such as tiled concrete and pavement bricks achieved the highest a_w. These types of terrains can have a negative impact on the user's health when used for commuting for a long period of time (4 to 8 *hours*). Also, lightweight users are significantly impacted by the whole-body vibrations compared to heavyweight users.

Table B.2 **RMS weighted frequency acceleration (a_w), $MTVV$, and $eVDV$ for different terrain types on the seat pan (W_k).**

	User 1				User 2				User 3				User 4			
	$a_w(m/s^2)$	$MTVV(m/s^2)$	$MTVV/a_w$	$eVDV(7.5h)$	$a_w(m/s^2)$	$MTVV(m/s^2)$	$MTVV/a_w$	$eVDV(7.5h)$	$a_w(m/s^2)$	$MTVV(m/s^2)$	$MTVV/a_w$	$eVDV(7.5h)$	$a_w(m/s^2)$	$MTVV(m/s^2)$	$MTVV/a_w$	$eVDV(7.5h)$
Tarmac	0.342	0.342	1.00	-	0.404	0.444	1.10	-	0.282	0.288	1.02	-	0.258	0.258	1.00	-
Tiled Concrete	1.093	1.093	1.00	-	0.947	0.980	1.03	-	0.910	0.910	1.00	-	0.706	0.773	1.09	-
Pavement Bricks	1.035	1.035	1.00	-	0.808	0.820	1.01	-	0.788	0.792	1.01	-	0.584	0.625	1.07	-
Tiled Concrete	1.055	1.087	1.03	-	0.885	0.923	1.04	-	0.807	0.843	1.04	-	0.601	0.672	1.11	-
Tiled Flooring	0.284	0.368	1.29	-	0.220	0.249	1.13	-	0.198	0.198	1.00	-	0.157	0.162	1.02	-
Carpet Flooring	0.236	0.236	1.00	-	0.180	0.199	1.74	-	0.184	0.193	1.04	-	0.152	0.158	1.03	-
Overall	0.850	1.267	1.49	15.24	0.685	0.984	1.43	12.30	0.647	0.955	1.47	11.57	0.522	0.775	1.48	9.35

On the other hand, there are no health risks of driving the EPW on indoor terrain surfaces such as tiled floor and carpet floor for a long period of time for all users. Also, the whole-body vibrations produced by outdoor terrains such as undamaged tarmac are not harmful.

Vibration effect on comfort

Whole-body vibrations impact on user's comfort can be measured using the vibration total value (a_v) (B.4). Table B.3 presents the guidance bases of assessing the impact of vibrations on user's comfort according to the ISO-2631 standard [163].

$$a_v = \left(k_x^2 a_{wx}^2 + k_y^2 a_{wy}^2 + k_z^2 a_{wz}^2\right)^{\frac{1}{2}} \tag{B.4}$$

Table B.3 **ISO-2631 guidance table of vibrations effect on comfort.**

Less than 0.315 (m/s^2)	Not uncomfortable
0.315 (m/s^2) to 0.63 (m/s^2)	A little uncomfortable
0.5 (m/s^2) to 1 (m/s^2)	Fairly uncomfortable
0.8 (m/s^2) to 1.6 (m/s^2)	Uncomfortable
1.25 (m/s^2) to 2.5 (m/s^2)	Very uncomfortable
Greater than 2 (m/s^2)	Extremely uncomfortable

Table B.4 shows the impact of whole-body vibrations on the user's comfort. For all users, the experience is "Fairly uncomfortable" according to the ISO guidance table (Table B.3). Whereas the lightest weight user (user 1) has experienced the highest a_v "Uncomfortable" and the heaviest user (user 4) has experienced the lowest a_v "A little uncomfortable".

Table B.4 **Vibrations effect on user's comfort using a_v in all axes of the seat pan.**

Metric / **User**	$a_v(m/s^2)$	**State**
User 1	0.920	Fairly uncomfortable / Uncomfortable
User 2	0.763	Fairly uncomfortable
User 3	0.715	Fairly uncomfortable
User 4	0.583	A little uncomfortable / Fairly uncomfortable

Probability of vibration perception

Table B.5 shows the weighted and unweighted RMS acceleration (recommended metric by the ISO standard) for the overall route for all users. Results show high perceptibility of vibration for all powered wheelchair users.

Table B.5 **Weighted and unweighted RMS acceleration.**

	Unweighted RMS acceleration (m/s^2)	**Weighted RMS acceleration** (m/s^2)
User 1	1.214	0.850
User 2	0.949	0.685
User 3	0.891	0.647
User 4	0.735	0.522

B.3 Conclusion

Results show the potential risks on user's health as a result of prolonged drive of powered wheelchairs (which is normally the case) on terrains such as tiled concrete and pavement bricks. The tendency of the health risk increases for lightweight users compared to heavyweight ones as heavyweights can dampen low vibrations. Therefore, we recommend against the usage of such types of terrains that disabled users may use for commuting.