

Article

A Robust Vehicle Detection Model for LiDAR Sensor Using Simulation Data and Transfer Learning Methods

Kayal Lakshmanan ^{1,*}, Matt Roach ¹, Cinzia Giannetti ¹, Shubham Bhoite ¹, David George ²,
Tim Mortensen ¹, Manduhu Manduhu ³, Behzad Heravi ², Sharadha Kariyawasam ² and Xianghua Xie ¹

¹ Faculty of Science and Engineering, Swansea University, Swansea SA1 8EN, UK

² Vortex IoT, Neath SA11 1NJ, UK

³ School of Computing, Engineering and Physical Sciences, University of the West of Scotland, Paisley PA1 2BE, UK

* Correspondence: k.lakshmanan@swansea.ac.uk

Abstract: Vehicle detection in parking areas provides the spatial and temporal utilisation of parking spaces. Parking observations are typically performed manually, limiting the temporal resolution due to the high labour cost. This paper uses simulated data and transfer learning to build a robust real-world model for vehicle detection and classification from single-beam LiDAR of a roadside parking scenario. The paper presents a synthetically augmented transfer learning approach for LiDAR-based vehicle detection and the implementation of synthetic LiDAR data. A synthetic augmented transfer learning method was used to supplement the small real-world data set and allow the development of data-handling techniques. In addition, adding the synthetically augmented transfer learning method increases the robustness and overall accuracy of the model. Experiments show that the method can be used for fast deployment of the model for vehicle detection using a LiDAR sensor.

Keywords: transfer learning; vehicle detection; LiDAR sensor; faster-RCNN; synthetic LiDAR data generation



Citation: Lakshmanan, K.; Roach, M.; Giannetti, C.; Bhoite, S.; George, D.; Mortensen T.; Manduhu, M.; Heravi, B.; Kariyawasam, S.; Xie, X. A Robust Vehicle Detection Model for LiDAR Sensor Using Simulation Data and Transfer Learning Methods. *AI* **2023**, *4*, 461–481. <https://doi.org/10.3390/ai4020025>

Academic Editor: Giovanni Diraco

Received: 20 February 2023

Revised: 12 May 2023

Accepted: 19 May 2023

Published: 1 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In our real-world context, one of the biggest challenges facing city planners and governments is the environmental impact of traffic congestion. In the UK alone, the DEFRA (UK Government Department for Environment Food & Rural Affairs, London, UK) clean air strategy includes a Three billion plan to improve air quality and reduce harmful emissions. Moreover, $\approx 30\%$ of urban traffic comes from cars searching for parking spaces, with drivers in the UK wasting nearly two whole days (44 h) on average annually (close to three full days in London—67 h) circling the city streets to find vacant spaces. Furthermore, $\approx 33\%$ of parking spaces are underutilised daily [1], making parking an essential component of sustainable transportation management, especially for high-density large cities.

The topic of parking has received comparatively little study upon which to ground the future development of smart city policies [2]. A primary contributor is that many cities lack the basic information about parking resources [3], particularly on-street parking, due to the effort required to obtain the necessary data through traditional, labour-intensive parking surveys [2]. Detailed data describing the usage of parking spaces over temporal and spatial regions of interest would provide valuable insight, revealing the parking needs, habits, and trends of motorists [4,5]. However, due to the vast and sparse spatial and temporal regions of interest, on-street parking does not lend itself to easy assessment. Conventional methodologies of performing surveys of on-street parking are to walk or drive through the area of interest, manually tally the number of parked vehicles, and typically only provide coarse measures, such as percentage occupancy [2]. The data from these methods are thus used to gain insight into general parking trends in an area rather than real-time space occupancy, which could inform drivers contemplating a city centre visit [2]. A method

which automatically assesses the availability of parking spaces in urban areas would ease congestion and pollution in city centres while increasing driver convenience and have an impact on the productivity of a city region. In practice, various types of sensors for automated traffic monitoring are employed in driving applications, such as loop detectors, road sensors, radar sensors, and Bluetooth sensors [6]. Loop detectors are reliable and cost-effective for detecting vehicles, triggering traffic signals, and managing traffic flow, but they require regular maintenance and can be affected by environmental factors [7]. Road sensors are durable and collect accurate data on traffic flow, occupancy, and speed, but they can be expensive to install and maintain [8]. Radar sensors are accurate and detect a wide range of vehicle types and sizes but can be affected by electromagnetic interference [9]. Bluetooth sensors are inexpensive and easy to install but are limited by the presence of Bluetooth-enabled devices [10]. These traditional sensors typically provide traffic frequency counts in a given location and do not provide high-resolution micro-traffic data, including speed, location, direction, and timestamp [11].

Recently, higher fidelity sensing technology, vision-based approaches and LiDAR sensors have been employed in a wide range of applications to obtain high-resolution traffic data for detecting vehicles and pedestrians, road boundaries, road facilities, and traffic lanes. Vision-based and LiDAR sensors provide improvements in detection range and accuracy compared to traditional traffic monitoring sensing [12–14]. However, vision-based systems do not perform well in complex scenes with variable lighting conditions [11], and privacy concerns might limit the installation locations. The recent development of low-cost manufacturing of LiDAR sensors allows cost-efficient deployment along the roadside to provide high-resolution micro-traffic data. These sensor technologies rely on a post-processing step applying various types of analysis to extract actionable insights from the data collected. These range from basic rule-based feature extraction to state-of-the-art data-driven approaches and are covered in detail in Section 2. However, there are still several limitations in deploying low-cost LiDAR for the real-time detection and monitoring of vehicles. One of the major challenges for the data-driven approaches is the lack of existing high-quality labelled data, which is an especially difficult challenge when the development is based on new technology since, without prior data, it is not possible to create a data-driven model. The research community is combating this challenge in varied domains by making use of synthetic data and transfer learning [15–17]. Transfer learning is a learning paradigm by which it is possible to adapt a machine learning model from one domain and reuse it to build a new model on another domain or perform a different but related task. With transfer learning, a common two-step approach to overcome the scarcity of data is to train a model on large volumes of synthetically generated data and, subsequently, adopt this model by using a small sample of real data [18]. Although this process has been widely applied in computer vision [19], there is limited research on its application to improve the robustness and accuracy of vehicle detection models for low-cost manufacturing LiDAR sensors. In the literature, there are several types of transfer learning, including homogenous, heterogenous, inductive, and transductive transfer learning [20,21], and there are different methods used for transfer learning, namely feature-based and model-based transfer learning [20]. In our study, an inductive type of transfer learning is used, and a feature-based method is used for transfer learning.

This paper addresses this research gap by proposing a novel approach which uses synthetic augmentation to create a robust data-driven model for vehicle detection given a new signal modality capture method, namely low-cost single beam LiDAR sensors, for deployment in a real-world context. Following extensive experimentation, it is shown that the proposed method improves the accuracy of a vehicle prediction task compared to transfer learning without synthetic data augmentation.

This paper is organised as follows: following the introduction, Section 2 describes the related work, Section 3 discusses the research methodology, Section 4 discusses the implementation of synthetic data and their results, Section 5 provides the real-world data

capturing, processing and model training, Section 6 provides the comparative results of real-world data and, finally, Section 7 concludes with some remarks.

2. Related Work

This section covers the research landscape for a multi-model vehicle or object classification for traffic applications. LiDAR and video are the two most popular choices of sensors to detect vehicle presence; each technique has distinct advantages and disadvantages. When thinking about the task of vehicle detection, humans find the use of camera data intuitive, and it can be straightforward to label data; as such, it has been the basis for many approaches [11,22]. These approaches use image data from a camera, each pixel represented by grey-scale or colour information, and any objects have to be recognised and segmented (boundaries identified) before their position in space can be determined. Moreover, this can become particularly challenging in low light, e.g., nighttime, conditions where object colours and boundaries can become increasingly hard to establish. Utilising equipment based on LiDAR technology is one approach to overcome this limitation. LiDAR data can be more computationally efficient to process and provide effective coverage of both short and long distances compared to camera images [2,23]. LiDAR collects high-fidelity point clouds, i.e., a set of data that provides a distance from the sensor to the surfaces in the scene; as such, the location and scale of any objects are captured in the raw data. Additionally, since LiDAR measures the return signal of light emitted from the device, the ambient conditions have very little impact on the returned data. Additionally, new advancements in low-cost LiDAR sensor manufacture also enable the capturing of high-resolution micro-traffic data.

LiDAR-Based Vehicle Detection: There are two different approaches commonly used for vehicle detection and classification with roadside LiDAR data: feature-based approaches and data-driven-based approaches.

Feature-based approaches use hand-crafted feature extraction, e.g., height, width, length, middle drop, etc., from the LiDAR data to classify the vehicles. Using this approach, there are several ways the LiDAR data can be utilised for vehicle detection; for example, the measured LiDAR distance decreases when a vehicle enters the beam, and the corresponding vehicle height is calculated using simple geometry [24]; Ref. [25] identified robust features for supervised vehicle classification with LiDAR profile data as an input; Ref. [26] developed a procedure to extract high-resolution vehicle trajectories with roadside LiDAR sensor data, and these trajectories are applied for traffic performance evaluation; Ref. [27] developed a laser-based vehicle classification system based on different criteria, geometrical configuration, occlusion reasoning, sensor specifications, and tracking information. In many cases, these extracted features from feature-based approaches are fed into different classification models such as decision trees, support vector machines and principal component analysis [28]. Whilst these approaches are simple and effective, they are not robust to noise and complex scenes; for example, they cannot completely deal well with occlusion, as inferred from [27,28].

Data-driven approaches utilise different neural networks for the task of classification of 3D point clouds generated by the LiDAR sensor [29]. Recently, Convolutional Neural Networks (CNNs) have achieved great success in object detection tasks in both camera and LiDAR data. Several works [30–32] take images captured with cameras and apply end-to-end unified fully convolution network frameworks that predict object confidence and object location (bounding boxes) simultaneously. Moreover, detection and localisation have been expanded to 3D LiDAR data for autonomous driving systems [33]. Chen et al. [34] fused both the LiDAR point cloud features and local image features based on the region-based fusion network to regress the 3D localisation task and 3D object detection, and the method outperformed all other LiDAR-based methods for 2D detection when validating on open-source KITTI data set [35].

Two main challenges exist in our real-world application of retrospectively installed LiDAR scanners on street furniture. First, the scanners are installed and connected to IOT resource-constrained devices, which possess some computational power and latency

limitations due to the wireless connection. Secondly, the location of the installation on the lamppost means that the data contain many occlusions of vehicles, as illustrated in Figure 1. Object detection algorithms are commonly used for detecting vehicles in images and videos. There are several popular object detection algorithms, such as YOLO (You Only Look Once) [36], SSD (Single Shot MultiBox Detector), and Faster R-CNN (Region-based Convolutional Neural Network) [37]. These algorithms are based on deep learning techniques, specifically Convolutional Neural Networks (CNNs), which have proven to be effective for image recognition tasks and are known for their adaptability and open-source capabilities. Each algorithm has its own strengths and weaknesses. YOLO is known for its speed and real-time performance, making it ideal for applications such as autonomous driving. SSD strikes a balance between speed and accuracy and is also a popular choice for vehicle detection [38]. On the other hand, Faster R-CNN is also known for its accuracy and is commonly used for tasks such as object tracking because of its ability to detect occluded objects [39]. It uses a region proposal network to generate potential object locations, which allows it to detect objects even when they are partially occluded or obscured by other objects in the scene [40], which is extremely useful in our vehicle parking application. In [38], it has been found that the Faster RCNN model is well balanced for recall and precision ratio; however, YOLOv3 has a higher recall ratio than its precision, which means YOLOv3 has more misclassifications. Hence, Faster RCNN was solely chosen to perform all the tasks for the proposed methodology.

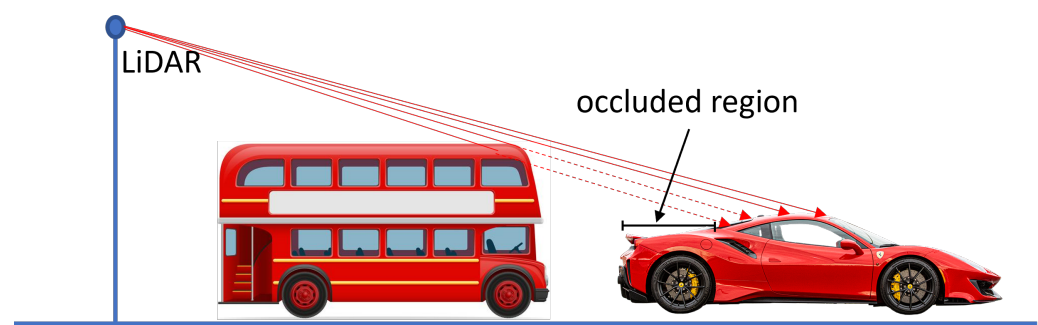


Figure 1. Occluded region of the vehicle. The car is partially occluded by the bus parked behind it.

One of the most perennial challenges with data-driven approaches, such as CNN, RCNN and faster RCNN, in real-world settings, is the lack of large amounts of annotated data [41]. In the absence of real-world data, accurate synthetic data have been used in many applications [42–44]. For example, Wang et al. [41] generated a synthetic image for photorealistic and non-photorealistic images and then applied the transfer learning method for vehicle detection using a Faster RCNN. Transfer learning improves learning in a new task (target domain) through the transfer of knowledge from a related task (source domain) that has already been learned [45]. Specifically, transfer learning improves model performance by starting with the learnt weights from a base model [45] and then refining through learning based on limited data of the target task. It follows that the base models need to be well-built and validated to achieve greater performance. Moreover, transfer learning breaks the constraint that the training and test data sets need to follow the same distribution [46]. This has benefited several fields when there is insufficient data to train the model, such as denoising, plant sciences, seismic fault detection, structural damage recognition and risk prediction [47–49]. However, the two data sets employed should be in similar fields; transfer learning cannot be used if there is no relationship between them.

In transfer learning, there are usually two common strategies: feature extraction and fine-tuning [49]. In feature extraction, all parameters in the neural network model of the source domain, apart from the final fully connected layer (often called the softmax), are frozen. The tensor from the final output of the frozen layers is extracted and flattened as features, which is used as input to train a classifier such as a multilayer perceptron or

Support Vector Machine (SVM) to achieve the target task [50]. For fine-tuning, a natural approach is to optimise all the parameters of the deep network using the target training data. However, fine-tuning the entire network may lead to overfitting if the target data set is limited. Alternately, the parameters of the remaining initial layers can be frozen at their previously trained values while the final few layers of the deep network are fine-tuned. Based on the data size, problem complexity and detection expectation, the above two strategies can be applied in different situations [49].

This paper presents a method for occlusion robust, localisation, detection and classification of vehicles using a low-cost single-beam LiDAR sensor. A faster RCNN model is trained using synthetic data via transfer learning. There have been a few studies that use synthetic-to-real transfer learning in various domains and applications [47–49]. However, this is the first study to adopt synthetic-to-real transfer learning of roadside LiDAR sensor data in vehicle detection applications. Moreover, such a method is tested on real-world LiDAR data for vehicle detection. The main contributions of the work are:

- A synthetic LiDAR data generation tool.
- Comparison of transfer learning with and without synthetic data:
 - From camera-based real-world data to our LiDAR capture data.
 - From camera-based real-world data via a large synthetic data set, which synthesises our real-world data set accurately to our LiDAR capture data.
- This comparison demonstrates in our application that Synthetically Augmented Transfer Learning contributes to an increase in the performance of the classification model.

3. Methodology

This work compares and improves an object detection model trained with two different transfer learning bases: first, using a large image data set that is captured differently (via camera) from our application data set (LiDAR) and second, adopting a step using a sizeable synthetic data set that accurately reflects our real-world LiDAR data set. The research methodology of this work is described in Figures 2 and 3. Figure 2 describes the Classic Transfer Learning method where a pre-trained object detection Faster RCNN model trained on the Common Objects in COntext (COCO) data set [51] is utilised as a base model, and then, our real LiDAR data are transfer-learned from this base model. Figure 3 describes the Synthetically Augmented Transfer Learning method where a large synthetically generated data set is used in transfer learning before finalising the learning with the real LiDAR data.

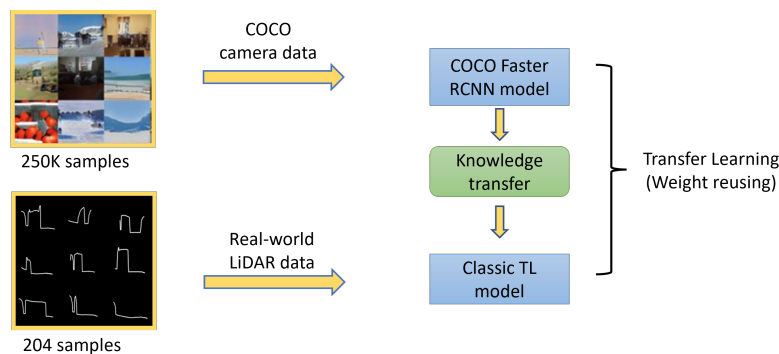


Figure 2. Classic Transfer Learning method: Classic TL model is transfer learned using real-world LiDAR data from the COCO Faster RCNN model.

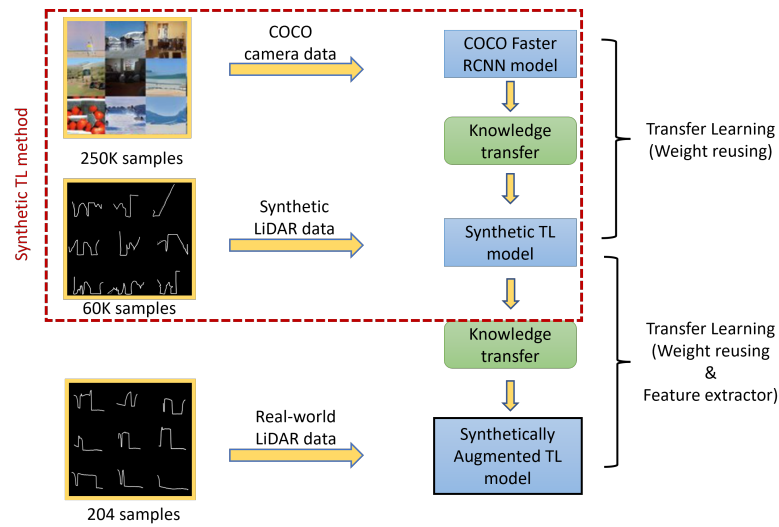


Figure 3. Synthetically Augmented Transfer Learning method: Synthetically augmented TL model is transfer learned using real-world LiDAR data from the Synthetic TL method, whereas the synthetic TL model is transfer learned using synthetic LiDAR data from the COCO Faster RCNN model.

4. Synthetic Data

4.1. Data Generation

This section describes the simulator that allows us to generate the synthetic data on which a model is trained to detect, locate and classify vehicles within the LiDAR signal. The environment is comprised of two fundamental components in the scene: the LiDAR set up on the side of the road and the 3D vehicles. The distance, p_h , as seen in Figure 4 between the object profile point and the ground surface, is computed and used to create the vehicle profile image.

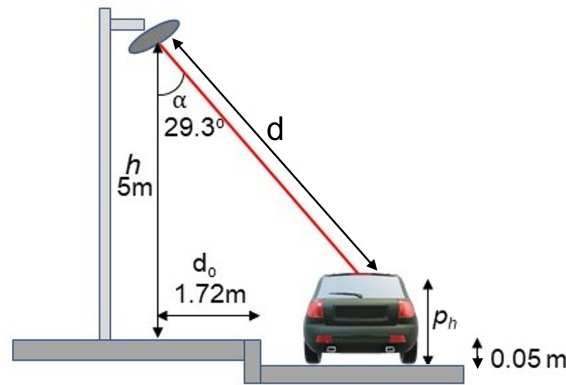


Figure 4. Illustration of our LiDAR system.

In the LiDAR system, the ground is selected as a relative object for computing distance values; note, the LiDAR itself is selected as the relative object in the real system. As illustrated, the distance d , between LiDAR and the object profile point, the distance between LiDAR and the ground, h , and the angle between the laser beam and the vertical axis α , the distance between the object profile point and the ground, p_h , are computed. This is completed using the principle of ray–triangle intersection since each vehicle 3D model provided is a set of triangle faces provided by ShapeNet [52].

4.2. Implementation and Fundamentals

Our simulation generates many different configurations of scenes to give a varied data set of LiDAR profiles intended to mimic real-world data. In this section, the implementation choices made along with the system’s basic functionality are outlined.

The 3D models provided by ShapeNet are in a standard file format (.obj) which defines a shape as a triangular mesh, which means the surface of the 3D object consists of many triangles connected along each edge. These shapes can be fed into our program and take advantage of the triangular representation to obtain precise locations of LiDAR intersections. ShapeNet 3D Vehicles are split into nine classes: Car, 4 × 4, Van, Motorcycle, Bicycle, Bus, Truck, Double-Decker and Minibus, which are shown in Figure 5. This work presents results based on seven and, later, with the real-world data, five classes, over 5000 models, each with a unique LiDAR profile, was used in our experiments. Using varied combinations of these shapes and other parameters, see Table 1, allows us to make a vast data set of diverse LiDAR profiles.

Ray Casting and Vehicle Intersections: In order to simulate a LiDAR sensor, intersections between lines from our virtual LiDAR sensor to the rendered vehicle need to be found. A technique called ray casting is used for this purpose. Ray casting works by providing a point in 3D space and a vector describing a direction to cast a ray and monitoring the 3D scene to detect any intersections with objects. Each angle of the LiDAR's rotation casts one ray and detects the intersection as one of three things: intersection with the vehicle, intersection with the road, and no intersection. The different rays are shown in Figure 6. Ray casting not only allows us to determine if the ray intersected with an object but also precisely where it intersected and the length of the ray. A rendering library called VTK (Visualisation Toolkit) was used to implement ray casting and intersection calculations [53] to construct our synthetic data.

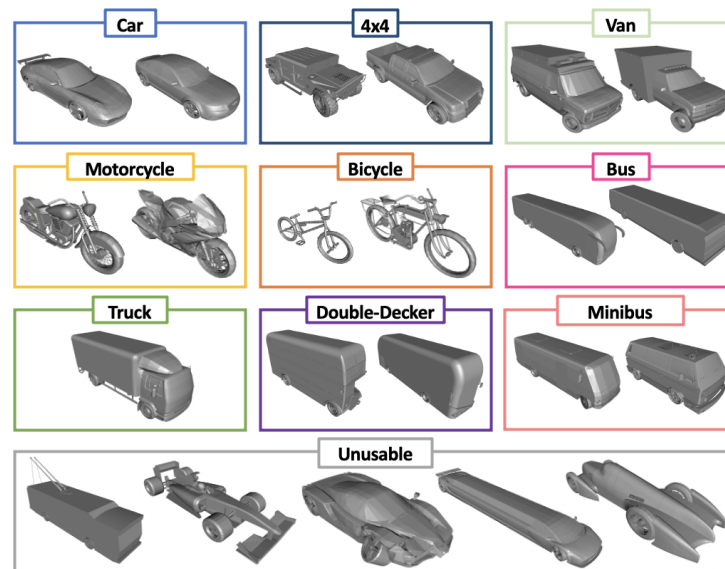


Figure 5. ShapeNet 3D Vehicles, split into nine classes: Car, 4 × 4, Van, Motorcycle, Bicycle, Bus, Truck, Double-Decker and Minibus.

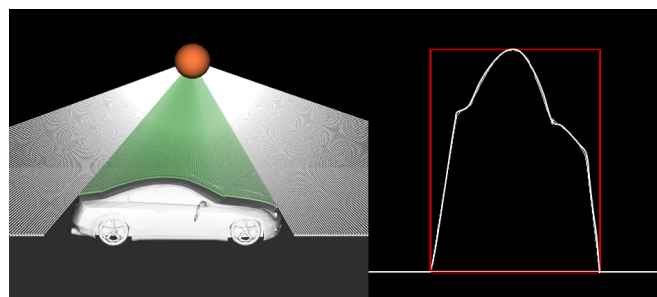


Figure 6. Rendering of the intersections between LiDAR rays and a vehicle's surface, the left image shows how the LiDAR ray intersects the 3D vehicle model, and the right image shows the generated 2D profile image (scaled).

Table 1. Parameters and assumptions.

Parameter	Description	Type	Value
LiDAR Height	h	Fixed	5 m
LiDAR Road Offset	d_o	Fixed	1.72 m
LiDAR Angle	α	Fixed	29.3°
LiDAR Increment Angle	L_i	Fixed	1°
Arc of Interest	L_a	Fixed	120°
Sweep Frequency	L_{sf}	Fixed	5.5 Hz
Sweep Time	L_{st}	Fixed	0 s
Scan Range	L_r	Fixed	12 M
Vehicle Type	V_t	Variable	$\in SN$
Vehicle Scale	V_s	Fixed	
Vehicle Driving Direction	V_d	Variable	$\in \{F, N\}$
Vehicle Angle	θ	Variable	$\theta \sim \mathbb{U}(-9, 9)$
Samples per Revolution	S_r	Derived	720
Samples per Second	S_t	Derived	3960
Samples in Sweep per Second	S_{st}	Derived	1320
Multiple parked vehicles	mv	Fixed	5

Assumption	Description
LiDAR Increment Angle	L_i is constant, meaning each time the sensor rotates, it is by a constant value
Samples per Revolution	L_{sr} is a whole number, this means that for each revolution of the LiDAR there is no drift
$L_{sr} \times L_i = 360^\circ$	resulting in a single full revolution each sweep based on the above assumptions
$\alpha = 0$	LiDAR is directly above the moving vehicle, and the sweep is perpendicular to the road
$L_{st} = 0$	The LiDAR sweep is instantaneous for computational efficiency

4.3. Data Simulations

Many external conditions impact the data collected from real LiDAR sensors. Our experiments aim to simulate many of these conditions to make our synthetic data more realistic. This section outlines several implemented methods.

4.3.1. Vehicle Rotations

Not every vehicle will park under the LiDAR straight and parallel; therefore, our simulations rotate the vehicle. To simulate this augmentation, the vehicle is rotated in 3D space around a specific axis, Y . Here, 3D space is orientated such that the Y -axis points upwards, the X -axis is perpendicular to the road, and the Z -axis runs parallel to the road. This means that applying the rotation to the Y -axis is enough to turn the vehicle realistically, assuming the vehicle is initially placed in the scene parallel to the Z -axis. The Y -axis rotation matrix shown in Equation (1) is applied to every point in the 3D shape to apply a rotation to the vehicle at θ° , which varies for each vehicle in each scene to increase the variability of the data set. An example rendering of vehicle rotation is shown in Figure 7.

$$Y_{axis} \begin{bmatrix} P'_x \\ P'_y \\ P'_z \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (1)$$

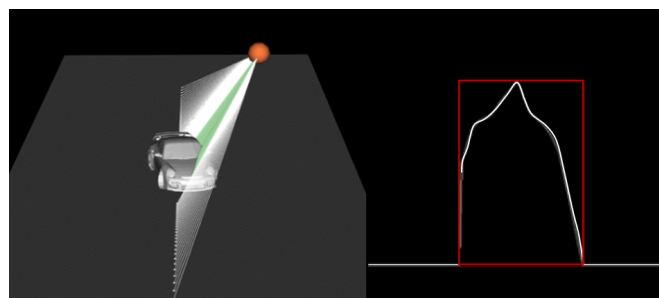


Figure 7. Rendering of a vehicle passing the LiDAR sensor at an angle, θ , from our simulations.

4.3.2. Creating Varied Scenes

Each vehicle is placed in the scene (always, at least partially under the LiDAR) in a range of various positions with respect to the X -axis and Z -axis, including changing the vehicle rotation as explained in Section 4.3.1. A random combination of the vehicle profiles is used to create Multi-vehicle Scenes. As our mounted real-world LiDAR sensors are typically monitoring parking bays, the majority of scans will likely include multiple parked vehicles mv . In order for the trained model to deal well with occlusion, as depicted in Figure 1, it is also necessary to transfer learn on multiple vehicles parked at varying distances apart underneath the LiDAR sensor (an example of this is shown in Figure 8). Unlike in the real-world data, as the type of vehicles used in each scene is known, the labelled can be easily generated.

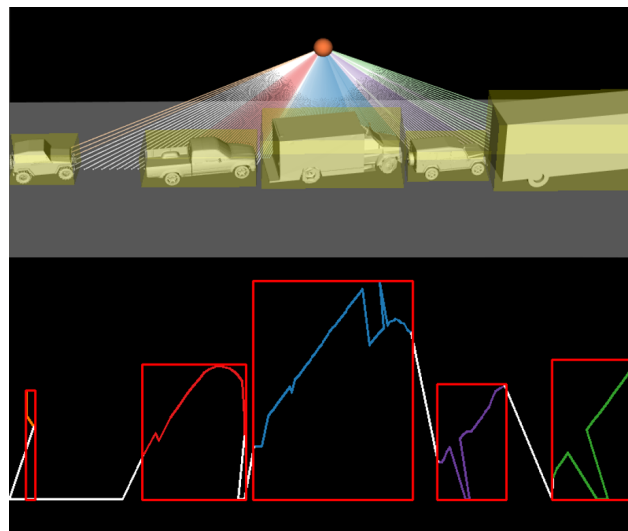


Figure 8. Rendering of five vehicles parked along with the resulting LiDAR image.

4.3.3. Parameters and Assumptions

In previous sections, the most significant variable in the vehicle profile scene generation process has been highlighted. This section outlines a full list of the parameters taken into consideration while computing the synthetic LiDAR data, which are provided in Table 1. As can be seen in column three, there are two primary types of parameters, fixed and variable parameters; there are also derived, i.e., those that are calculated from other parameters. These parameters are chosen to best represent our real-world context; an illustration of LiDAR context with parameters is depicted in Figure 4.

4.4. Synthetic Training and Results

This section describes model training using the synthetic TL method, illustrated in the dashed red box in Figure 3, using the base model of Faster RCNN. The synthetic data set includes a training data set of 60,000 samples, a validation data set of 14,425 samples, and a test data set of 43,272 samples, each containing samples from seven distinct classes. First, a pre-trained Faster RCNN [51] model based on ResNet using the COCO data set was adopted as the base model. Faster RCNN comprises three different parts: convolutional layers, where filters are trained to extract the appropriate features from the image; a Region Proposal Network, which operates on the last CNN feature map to predict object proposals; a fully connected layer, which predicts the objects and bounding boxes. This model is optimised through stochastic gradient descent.

Second, the model is tuned via the weight-reusing transfer learning method; i.e., take the Faster RCNN model and remove its final layer, then add the last layer (classification layer) of our own and train the new model. All the weights in the model are updated whilst training on the synthetic data set: 25,000 steps and 2000 warm-up steps. Some simple

preliminary experiments in varying steps and warm-up steps were carried out; however, they did not materially affect the learning and therefore used the parameters suggested in [54]. Finally, the transfer learned object detection model is tested using the synthetic test data set. This procedure is reused for all the tasks carried out in this work.

Figure 9 shows the confusion matrix of the synthetic test data set for 43,272 total samples. The diagonal elements show the correctly predicted class percentage, and the other elements show the error dispersion percentage for each class. The total classification accuracy is 88.36%; as can be seen, the classes Bus and Double-Decker are the most difficult to classify, with an accuracy of 68% and 62%, respectively. The algorithm is also good at detecting the location of the vehicles in the scene given by the mean and variance of the Intersection over Union (IoU) metric, defined in Equation (2); these values are 0.88 and 0.23, respectively.

$$IoU = \frac{\text{Area of Intersection}}{\text{Area of Union}} \tag{2}$$

Figure 10 shows the Receiver Operating Characteristic (ROC) curve of the test data set for all classes. The X-axis shows the false positive rate, and the Y-axis shows the true positive rate. It can be seen that all classes are predicted well, with an Area Under the Curve (AUC) between 0.97 and 1.

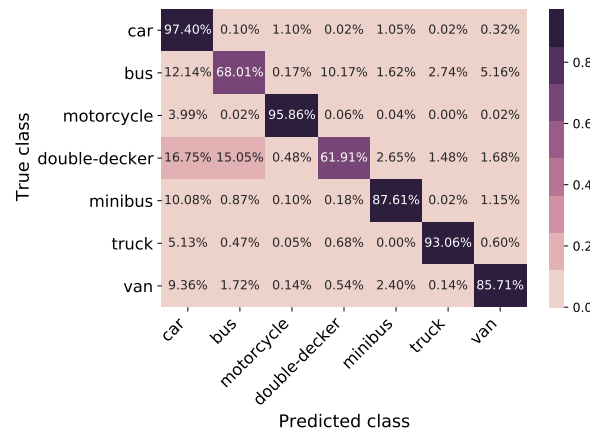


Figure 9. Confusion matrix of test data set for 43,272 total samples; classification accuracy—88.36%.

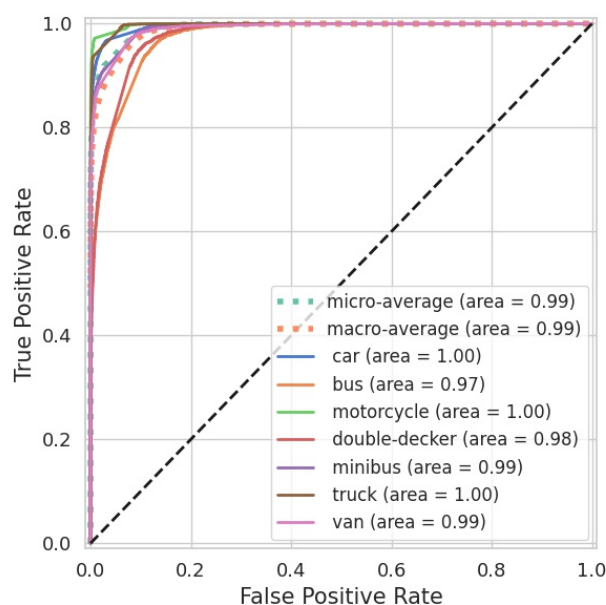


Figure 10. ROC curve of the test data set for 43,272 samples.

5. Real-World Data

Having established a working model using synthetic data, this section outlines the LiDAR sensor installation, capturing of the LiDAR data from the LiDAR, and the pre-processing of the data in order to test the performance of the synthetically trained Faster RCNN model on real-world data.

5.1. Hardware Installation of LiDAR Sensor

The LiDAR used is Slamtec RPLIDAR S1, which adopts time-of-flight ranging technology that guarantees a ranging accuracy of ± 5 cm over its full measurement range. It still has a stable and accurate ranging resolution even across a long distance. It also effectively avoids substantial daylight interference and has a stable ranging and high-resolution mapping performance in an outdoor environment. The device is small, easy to use, portable, and compact to fit several required applications. RPLIDAR S1 runs clockwise to perform a 360-degree omnidirectional laser range scanning for its surrounding environment (see Figure 11). In addition, this technology has integrated wireless power, optical communication, non-contact energy, and signal transmission technology in place of a slip ring or other physical connection subject to wear and a more limited lifespan [55]. This particular sensor was selected due to its combination of low cost, compact size, long range, and high scan rate. A 3D LiDAR sensor would be beneficial; the cost of such devices is orders of magnitude higher, and they are more computationally intensive to process due to the larger amount of data produced. Due to COVID restrictions in place for the majority of this funded work, LiDAR installation for capturing real-world data was limited to one location.



Figure 11. LiDAR hardware installation.

5.2. LiDAR Data Retrieval and Processing

The LiDAR sensor scans the scene and stores data on the cloud in real time. For each second, the LiDAR sensor monitors ten frames per second (ten times 360 samples, at 1° per sample) and saves the data in polar coordinates stored in a MySQL database. The data are retrieved using the SQL query for a given time interval and consist of: the sweep angle β in degrees and distance (d) in millimetres. Here, β is a given angle in the Arc of Interest L_a at this instance in time. Even though the LiDAR scans 360 degrees, the interest lies in where the LiDAR scans the road, that is, 120 degrees, L_a . An example of a polar plot and polar to the Cartesian plot is shown in Figure 12, where LiDAR scans the road. In the polar image, the polar coordinates (d, β) show the captured vehicle profiles and the road. The polar coordinates are converted to Cartesian coordinates ($x = d \cos(\beta)$, $y = d \sin(\beta)$) to analyse the vehicle profiles. The data shown in Figure 12 are representative of raw data from the device and contain noise and artefacts, which must be filtered from the data to allow object detection.

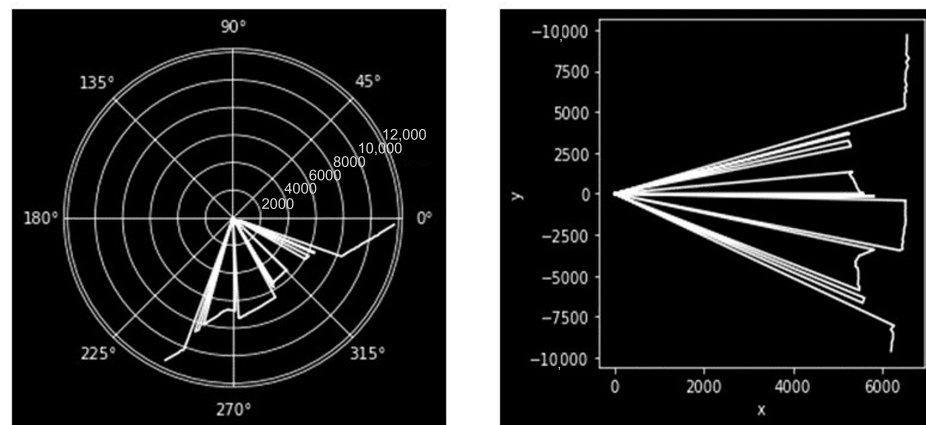


Figure 12. Polar plot (left) and polar to Cartesian Plot (right) of the LiDAR data.

Not all real-world LiDAR scans provide accurate vehicle profiles. Several noise reduction or filtering steps needed to be considered for the Cartesian plot to achieve a usable profile, which are detailed below:

- Eliminate LiDAR points appearing too close to the detector. The LiDAR can return very small or zero distance values in place of true values in some conditions.
- Restrict the data between 3000 and 7000 mm to obtain only vehicle profiles in the Cartesian plot in Figure 12.
- Remove the image when there are no vehicles parked on the road by limiting x between 6000 and 7000 in the Cartesian plot in Figure 12.
- Remove any LiDAR images that occasionally produce an insufficient profile, that is, when the vehicle profile is not adequately caught; for instance when LiDAR takes a profile of a fast-moving vehicle, it simply captures a vertical line that represents nothing.
- Re-scale the image by restricting the x and y -axis to match the profile of synthetic data.

The vehicle profiles shown in Figure 13 (left), along with the real-time image of parked vehicles (right), have been subject to the described processing.

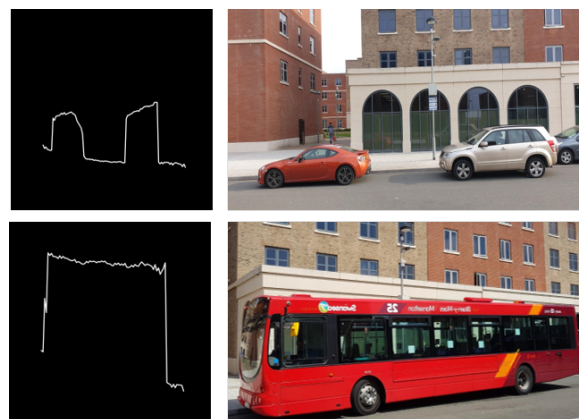


Figure 13. Samples of processed LiDAR image (left) and vehicle parked in real time (right).

5.3. Creating Ground Truth Labelled Data

In order to validate the performance of the system when deployed in the real world, we need to establish some ground truth labelled data. This section focuses on the building of a semi-automatic labelling pipeline for our LiDAR data using video captured from cameras mounted on the lamppost located at the deployment site.

The camera set-up consisted of two IP cameras (RLC—410W) that were mounted on the same lamppost as that of the LiDAR sensor. Each camera has a viewing angle of 80 degrees, thereby covering the LiDAR range of 120 degrees. The cameras have a built-in

motion detection sensor that triggers video capture, therefore filtering out unwanted data. The cameras operated around the clock, and the data accumulated were captured and saved via file transfer protocol.

A camera-based object detection algorithm was utilised to label the data since this area of work is very mature and has a number of pre-trained models that work well at this task. Specifically, the YOLO (You Only Look Once [56]) object detection algorithms and their successive series (YOLO V1, V2, V3, V4, and V5) have been proven to outperform other state-of-the-art object detection methods. The YOLO_V5 model is adopted in this work since it has 53.7% mAP (mean Average Precision) on COCO data [57], which is better than the YOLO_V4 with 43.5% mAP on COCO data [58] and has an ease of implementation and speed. Despite its merits in our test, it appeared that the model was mispredicting some vehicles. It is suspected that a major contributor to misprediction was the camera angle in combination with the number of objects in the camera view. Figure 14 illustrates two errors; in the left image, on the far left of the image, a car is misclassified as a different object (green bounding box); in the centre, a car is predicted correctly (orange bounding box), while on the right, a car is not classified at all. In the right image, the objects are classified correctly as cars; on the far left of this image, there is part of a car that is not classified.

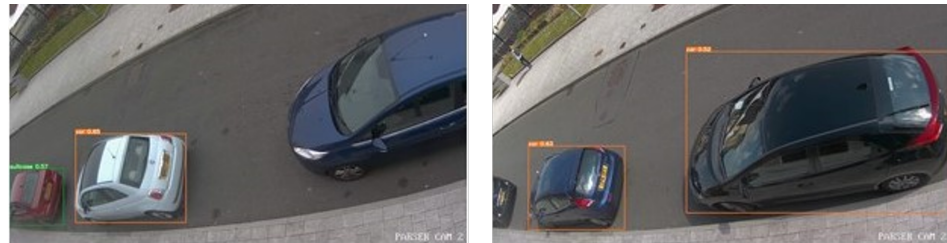


Figure 14. Object detected by YOLO_V5 model for images captured in different angles.

Following a confidence threshold and filtering for classes of interest (Car, Bus, Motorcycle, Minibus and Van), an initial ground truth table was generated, as shown in Table 2, which contains the class, bounding box coordinates, prediction score, timestamp, and a Boolean daytime flag. This ground truth label is then linked to the LiDAR data by merging identical timestamps before finally being cross-checked manually to ensure that the labels generated are correct. We noticed a reduction in accuracy for the camera-captured data during the nighttime (defined as post-sunset and pre-sunrise) with a 43% lower accuracy than in daylight. Following the manual annotation, a robust data set comprising 303 labelled LiDAR images was created, out of which 413 LiDAR profiles have been created. The number of LiDAR images used for testing was affected by COVID restrictions on the movement of people introduced at deployment times during the research, resulting in more extremely low volumes of vehicles in the real-world data than originally planned. COVID restrictions also limited the study to a single fixed location of the sensor. This challenge also presented an opportunity for the work, since these restrictions exacerbated the challenge for the transfer learning approach, stressing the limits of the abilities of the learning method to perform under low levels of real-world training data.

Table 2. Ground Truth Table.

Class	Bounding Box	Timestamp	Prediction Score	Daytime Flag
Car	[0.9189, 0.2878, 0.1621, 0.3576]	2021-08-13 05:44:24	0.48485	Night
Truck	[0.0471, 0.7732, 0.0884, 0.3076]	2021-08-13 10:55:05	0.46900	Day
Car	[0.0470, 0.7746, 0.0902, 0.3006]	2021-08-13 10:55:05	0.49065	Day
Car	[0.8817, 0.8777, 0.2341, 0.2444]	2021-08-13 18:06:41	0.74662	Day
Car	[0.1920, 0.2434, 0.1447, 0.4270]	2021-08-13 18:06:41	0.84887	Day
Car	[0.8153, 0.2673, 0.1962, 0.3569]	2021-08-13 18:06:41	0.90770	Day
Car	[0.8817, 0.8718, 0.2341, 0.2548]	2021-08-13 18:07:13	0.59120	Day

6. Comparative TL Model Results Tested on Real-World LiDAR Data

This section provides a comparison of three models, all tested on the real-world LiDAR data set: the classic TL model from the base model generated on the camera-based COCO data set; the synthetic TL model, starting with the same COCO data set base model and transfer learning (weight reusing) using our synthetically generated data; and finally, the synthetically augmented TL model, which uses the synthetic TL model as a base model and further transfer learning using the real-world data. Results for both weight reusing and feature extraction are presented. These three methods of transfer learning are presented in Figures 2 and 3 in Section 3. The real-world LiDAR test data set contains 101 images with 158 LiDAR samples with ground truth annotations for class labels and bounding boxes and is used for testing all three methods. Where the real-world data is used for transfer learning (weight reusing, feature extraction), the split is approximately 67% for train and validation and 33% for testing. Note that the split is not a random 70–30 or 80–20; some conditions have been manually added to make sure to balance all classes in all training, test and validation since the real-world data set is imbalanced across classes due to the prevalence of different vehicle types. In addition, the number of cars is an order of magnitude larger than all other classes, with the number of buses being the smallest class. The following sections present results and some reflections on the three models.

6.1. Classic TL Model

The weight-reusing method for transfer learning is adopted to train the classic TL Model using the real-world LiDAR training data set. Note that the feature extraction method of transfer learning is not effective in this case, since there is limited descriptive information to adapt from the COCO base model due to the different signal modalities. Whilst training the model, it was established that the model's accuracy did not change by varying the step and warm-up step hyperparameters; therefore, these values are fixed to 1000 and 100, respectively.

The results for the COCO Faster RCNN model trained using the weights reusing method are shown in a confusion matrix; see Figure 15 and the ROC curve in Figure 16. The X-label and Y-label in the confusion matrix displayed the predicted and true classes, and the diagonal values represented the correctly predicted classes. The model obtained an overall classification accuracy of 83.75% and an F1 score of 0.837 on the real-world LiDAR test set. It can be seen in the ROC curve that the majority of classes have an AUC of $\approx 80\%$ or over except for Van and Minibus. This results in a micro-average AUC of 90%; however, in our case, the macro-average is more representative due to imbalances in the data set, with an AUC of 79%. This performance is achieved through transfer learning based on a relatively small number (256) of real-world data sets.

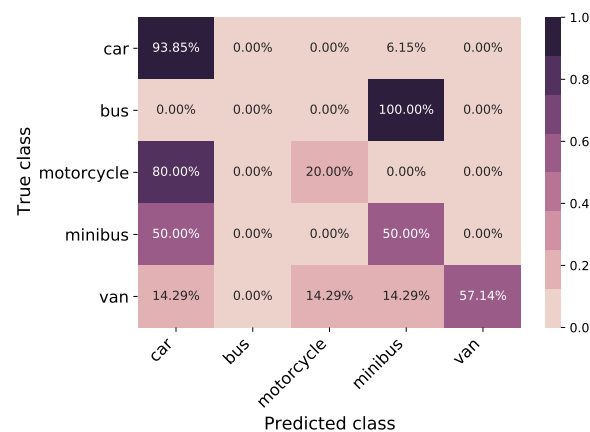


Figure 15. Confusion matrix of classic TL model trained and tested on real LiDAR data; the overall classification accuracy of 83.75%.

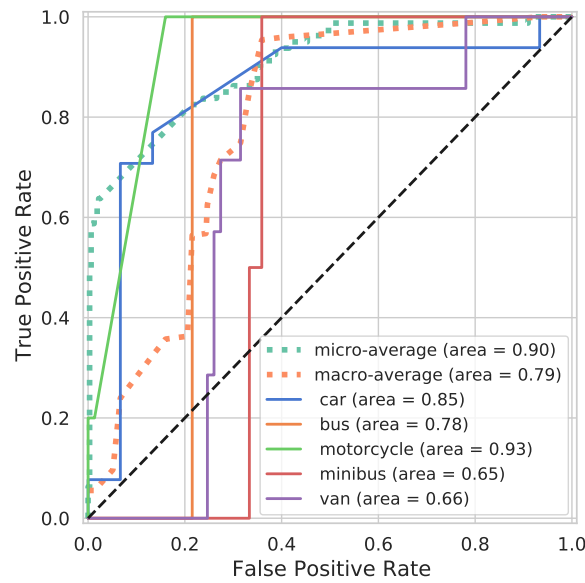


Figure 16. ROC curve of Classic TL model trained and tested on real LiDAR data.

6.2. Synthetic TL Model

In contrast to the other two methods (classic transfer learning and synthetically augmented transfer learning), the synthetic TL model, illustrated in the red box in Figure 3, uses synthetic data only to train, and it is tested on the real-world LiDAR test data set.

The results for the COCO Faster RCNN model trained using synthetic data are shown in a confusion matrix; see Figure 17 and the ROC curve in Figure 18. The model obtained an overall classification accuracy of 61.25% and an F1 score of 0.605 on the real-world LiDAR test set. It can be observed that the majority of misclassified cars are labelled as buses; this typically happens when the only edge of the car is visible in the LiDAR data. The model does not deal well with predicting motorcycles, which is likely due to the different profiles for this vehicle type across the synthetic and real-world data. In contrast to other vehicles, the model can be seen to predict buses perfectly with an AUC of 1.0, which is depicted by the orange line in the ROC curve in Figure 18. This is likely an anomaly down to the number of buses in the test data set (note the limited tests using real-world data led to the stepped appearance of the performance profile). Otherwise, across the board, the performance is considerably worse than the classic TL model with the micro and macro-average AUCs of 0.74% and 0.68%, respectively. Moreover, there are many classes that struggle to discriminate at all: car, motorcycle, and minibus; nevertheless, there is some discriminatory information captured by the synthetic TL model.

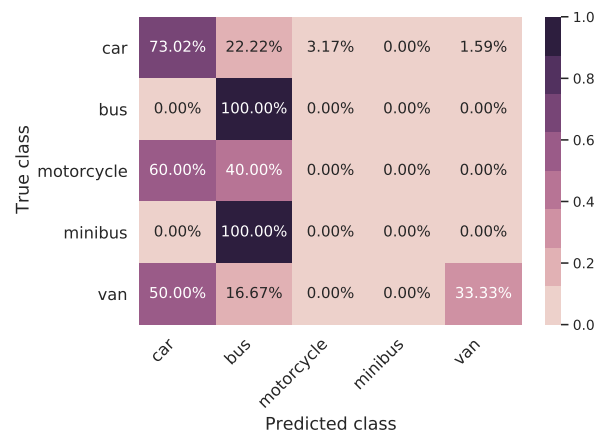


Figure 17. Confusion matrix of synthetic TL model tested on real LiDAR data; the overall classification accuracy is 61.25%.

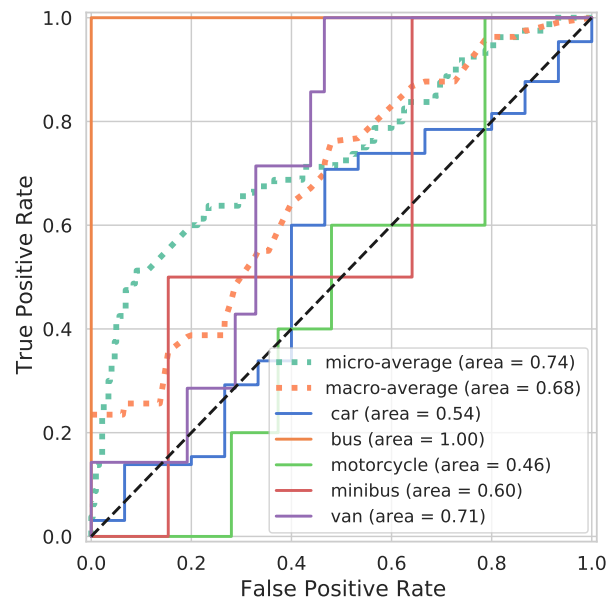


Figure 18. ROC curve of synthetic TL model tested on real LiDAR data.

6.3. Synthetically Augmented TL Model

In this section, synthetic data and real-world data are combined in a two-step transfer learning process to see if the synthetic data can add any discriminatory value to the classifier. The synthetically augmented TL model is depicted in Figure 3. Two sets of transfer learning results are discussed here: feature extraction and the weight-reusing transfer learning methods.

First, the results of the feature extraction transfer learning method are briefly presented. This method uses the representation learned by the synthetic TL model to extract meaningful features from real-world data sets, where the synthetic TL model layers are frozen except for the top layer (softmax classification layer). The model obtained an overall classification accuracy of 86.25% and an F1 score of 0.872 on the real-world LiDAR test set. Encouragingly, these results are better than the classic TL model. However, the weight-reusing transfer learning performance is better with a similar error profile of the feature extraction transfer learning method; therefore, the detailed results of the weight-reusing transfer learning method are detailed below.

The transfer learning weight reusing method unfreezes the entire model and re-trains the real-world LiDAR data with a low learning rate, making the model more accurate. The results for this synthetically augmented transfer learning method are shown in a confusion matrix; see Figure 19 and the ROC curve in Figure 20. The model obtained an overall classification accuracy of 91.25% and an F1 score of 0.906 on the real-world LiDAR test set. These figures illustrate that the weight-reusing method can classify the classes more accurately than the classic transfer learning method.

In the ROC curve, all the classes perform well except for Van, since this model is transfer learned on a small number of real-world data sets. The micro and macro averages are improved along with the AUC of the Car, Motorcycle, Minibus, and Van classes compared to the synthetic TL and classic transfer learning methods. Even though the synthetically augmented transfer learning method's AUC is similar to that of the classic transfer learning, there is a difference in profile and a significant improvement in error rate, as defined in [59]. Compared to the classic TL model, this model reduces the classification accuracy error by 46.15% and the F1 score by 40.81%, demonstrating a significant improvement. A summary of all the considered models tested on the real-world LiDAR data set is displayed in Table 3, which shows for each model the classification accuracy, weighted F1 score, Intersection over Union (IOU) mean, IOU variance, and computational time. All models have similar values when comparing IOU mean and variance except for the synthetic TL model. All

models have similar computational time except for the synthetic TL model, which is trained on a larger data set and for an order-of-magnitude more time step. This model brings an increase in F1 score performance of 7.5% over the classic TL model. A visualisation of detected vehicles in the LiDAR scan using our synthetic augmented transfer learning (weight reusing) method is shown in Figure 21.

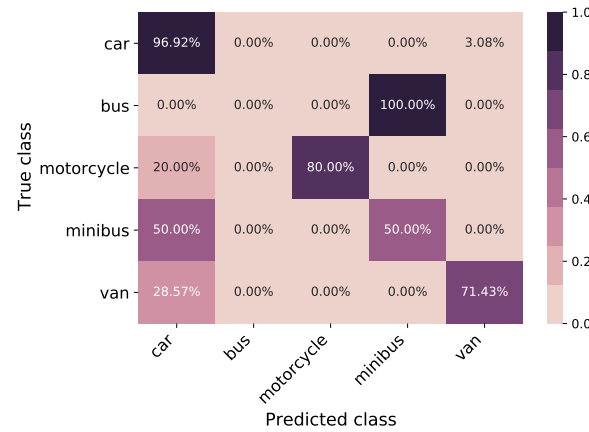


Figure 19. Confusion matrix synthetically augmented TL model (weight reusing) trained on real LiDAR test data; the overall classification accuracy is 91.25%.

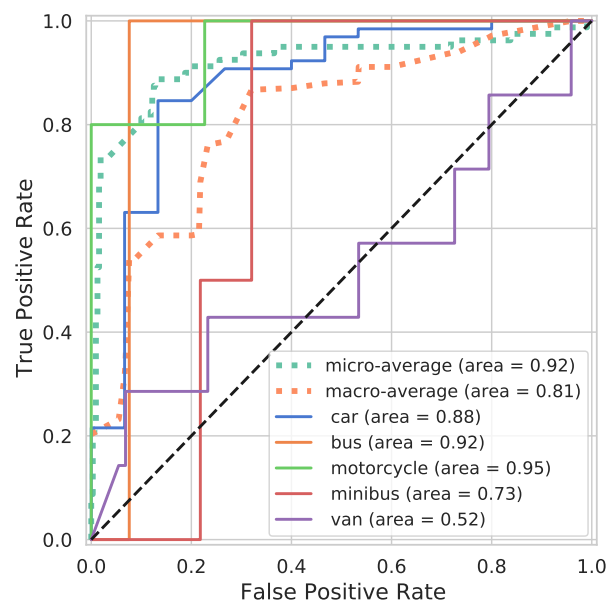


Figure 20. ROC curve of synthetically augmented TL model (weight reusing) trained on real LiDAR test data.

Table 3. Summary of models’ performance tested on real-world LiDAR data (error reduction is in comparison to CTL; note the synthetic TL model increases error). The bold font depicts the best model’s scores.

Model	Accuracy	Error Reduction	F1 Score	Error Reduction	IOU Mean	IOU Variance	Computation Time
Classic transfer learning (CTL)	83.75%	N/A	0.837	N/A	0.418	0.272	688.006 s
Synthetic TL model	61.25%	-58.73%	0.605	-58.06%	0.252	0.211	17250 s
Synthetic augmented TL (weight reusing)	91.25%	46.15%	0.906	40.81%	0.520	0.273	693.86 s
Synthetic augmented TL (feature extraction)	86.25%	36.3%	0.872	26.5%	0.494	0.263	688.01 s

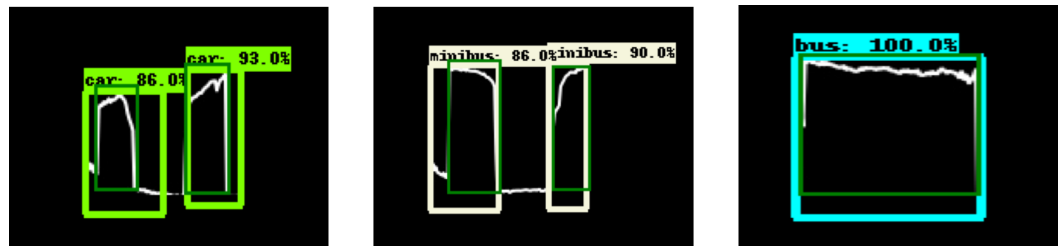


Figure 21. IOU visualisation of real-world data.

To evaluate the relative performance of the synthetically augmented TL model compared to the classic TL model, we tested the following hypothesis with a chosen significance level of 0.05:

- Null hypothesis H_0 : There is no difference between the synthetically augmented TL model and the classic TL model.
- Alternative hypothesis H_1 : The synthetically augmented TL model exhibits higher performance than the classic TL model.

By performing the Chi-square test across the two groups using the F1 scores of the models, we obtained a Chi-square value of 3.89 and a corresponding p -value of 0.048. $p < 0.05$ Hence, we reject the null hypothesis and accept the evidence to conclude that the synthetically augmented TL model exhibits statistically significantly higher performance than the classic TL model.

7. Conclusions

This paper has presented a novel approach for roadside vehicle detection using a synthetically augmented transfer learning method and LiDAR data. An efficient synthetic LiDAR data generation tool has been implemented, which takes advantage of the large number of vehicle shapes offered by the ShapeNet data set along with ray casting to mimic the LiDAR. The resulting data set is rich in both variabilities of vehicle profile and external factors, making it a good substitute for real-world LiDAR data until that is available. Furthermore, extensive experiments show that our synthetically augmented transfer learning method improves the performance of the object detection model, reducing the classification accuracy error rate by 46.1% and the F1 score error rate by 40.81% compared to the classic transfer learning method (Section 6.1). Although many researchers combined transfer learning and synthetic data for various applications, there are not any practical, cost-effective LiDAR sensor applications or synthetically enhanced transfer learning for vehicle identification/detection in our understanding. The methodology has been tested on real-world LiDAR data and validated using camera data. The proposed method enables the fast deployment of object detection models when high-fidelity labelled data are scarce and can be therefore used to scale up the development of cost-effective LiDAR-based parking solutions.

Further work intends to carry out the sensitivity analysis showing how variability in the synthetic data influences the model performance and further increases the computational efficiency of the training process, and it reduces time to effective deployment in a range of varied locations.

Author Contributions: Conceptualisation, B.H., M.R. and C.G.; methodology, M.M., D.G., K.L., M.R. and C.G.; software, D.G.; validation, K.L. and M.R.; formal analysis, K.L., M.R. and C.G.; investigation, K.L. and D.G.; resources, T.M.; data curation, S.B. and D.G.; writing—original draft preparation, K.L.; writing—review and editing, K.L., M.R. and C.G.; visualization, K.L.; supervision, M.R., C.G. and S.K.; project administration, S.K.; funding acquisition, B.H., M.R. and X.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Innovate UK grant number 23634.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data collected for this work are under a commercial confidentiality agreement.

Acknowledgments: The authors would like to thank: Innovate UK for funding the work on the PARSER project; Vortex IoT, the lead on the PARSER project, for the opportunity to collaborate on the work; collaborators on the project—British Telecommunications Plc and Swansea and City Council—for their contributions; and all the people in the wider team for creating a fun and enjoyable working environment. The authors would like to acknowledge the UK Engineering and Physical Sciences Research Council (EPSRC) (EP/S001387/1; EP/V061798/1). The authors also would like to acknowledge the Ser Cymru Infrastructure Grant, AccelerateAI, part-funded by the ERDF via the Welsh Government.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the result.

References

1. British Parking Association. Available online: <https://www.britishparking.co.uk/Library-old/Blueprint-for-Parking-2017-2021/136174> (accessed on 21 January 2023).
2. Thornton, D.A.; Redmill, K.; Coifman, B. Automated parking surveys from a LIDAR equipped vehicle. *Transp. Res. Part C Emerg. Technol.* **2014**, *39*, 23–35. [CrossRef]
3. De Cerreño, A.L. Dynamics of on-street parking in large central cities. *Transp. Res. Rec.* **2004**, *1898*, 130–137. [CrossRef]
4. Zhao, P.; Guan, H.; Wang, P. Data-driven robust optimal allocation of shared parking spaces strategy considering uncertainty of public users' and owners' arrival and departure: An agent-based approach. *IEEE Access* **2020**, *8*, 24182–24195. [CrossRef]
5. Chai, H.; Ma, R.; Zhang, H.M. Search for parking: A dynamic parking and route guidance system for efficient parking and traffic management. *J. Intell. Transp. Syst.* **2019**, *23*, 541–556. [CrossRef]
6. Chen, S.; Chen, Y.; Zhang, S.; Zheng, N. A novel integrated simulation and testing platform for self-driving cars with hardware in the loop. *IEEE Trans. Intell. Veh.* **2019**, *4*, 425–436. [CrossRef]
7. Yuan, Y.; Van Lint, H.; Van Wageningen-Kessels, F.; Hoogendoorn, S. Network-wide traffic state estimation using loop detector and floating car data. *J. Intell. Transp. Syst.* **2014**, *18*, 41–50. [CrossRef]
8. Barceló, J.; Kuwahara, M.; Miska, M. *Traffic Data Collection and Its Standardization*; Springer: Berlin/Heidelberg, Germany, 2010.
9. Yang, Z.; Pun-Cheng, L.S. Vehicle detection in intelligent transportation systems and its applications under varying environments: A review. *Image Vis. Comput.* **2018**, *69*, 143–154. [CrossRef]
10. Díaz, J.J.V.; González, A.B.R.; Wilby, M.R. Bluetooth traffic monitoring systems for travel time estimation on freeways. *IEEE Trans. Intell. Transp. Syst.* **2015**, *17*, 123–132. [CrossRef]
11. Lv, B.; Xu, H.; Wu, J.; Tian, Y.; Zhang, Y.; Zheng, Y.; Yuan, C.; Tian, S. LiDAR-enhanced connected infrastructures sensing and broadcasting high-resolution traffic information serving smart cities. *IEEE Access* **2019**, *7*, 79895–79907. [CrossRef]
12. Du, X.; Ang, M.H.; Rus, D. Car detection for autonomous vehicle: LIDAR and vision fusion approach through deep learning framework. In Proceedings of the 2017 IEEE/RSSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 749–754.
13. Chavez-Garcia, R.O.; Aycard, O. Multiple sensor fusion and classification for moving object detection and tracking. *IEEE Trans. Intell. Transp. Syst.* **2015**, *17*, 525–534. [CrossRef]
14. Liu, W.; Xia, X.; Xiong, L.; Lu, Y.; Gao, L.; Yu, Z. Automated vehicle sideslip angle estimation considering signal measurement characteristic. *IEEE Sens. J.* **2021**, *21*, 21675–21687. [CrossRef]
15. Pan, S.J.; Kwok, J.T.; Yang, Q. Transfer learning via dimensionality reduction. In Proceedings of the AAAI, Chicago, IL, USA, 13–17 July 2008; Volume 8, pp. 677–682.
16. Douarre, C.; Schielein, R.; Frindel, C.; Gerth, S.; Rousseau, D. Transfer learning from synthetic data applied to soil-root segmentation in X-ray tomography images. *J. Imaging* **2018**, *4*, 65. [CrossRef]
17. Jung, S.; Park, J.; Lee, S. Polyphonic sound event detection using convolutional bidirectional lstm and synthetic data-based transfer learning. In Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019), Brighton, UK, 12–17 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 885–889.
18. Liu, W.; Quijano, K.; Crawford, M.M. YOLOv5-Tassel: Detecting tassels in RGB UAV imagery with improved YOLOv5 based on transfer learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 8085–8094. [CrossRef]
19. Xiao, A.; Huang, J.; Guan, D.; Zhan, F.; Lu, S. Transfer learning from synthetic to real LiDAR point cloud for semantic segmentation. *arXiv* **2021**, arXiv:2107.05399.
20. Weiss, K.; Khoshgoftaar, T.M.; Wang, D. A survey of transfer learning. *J. Big Data* **2016**, *3*, 1–40. [CrossRef]
21. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]

22. Hu, H.N.; Cai, Q.Z.; Wang, D.; Lin, J.; Sun, M.; Krahenbuhl, P.; Darrell, T.; Yu, F. Joint monocular 3D vehicle detection and tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 5390–5399.
23. Broome, M.; Gadd, M.; De Martini, D.; Newman, P. On the road: Route proposal from radar self-supervised by fuzzy LiDAR traversability. *AI* **2020**, *1*, 558–585. [[CrossRef](#)]
24. Lee, H.; Coifman, B. Side-Fire Lidar-Based Vehicle Classification. *Transp. Res. Rec.* **2012**, *2308*, 173–183. [[CrossRef](#)]
25. Sandhawalia, H.; Rodriguez-Serrano, J.A.; Poirier, H.; Csurka, G. Vehicle type classification from laser scanner profiles: A benchmark of feature descriptors. In Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, 6–9 October 2013; pp. 517–522.
26. Sun, Y.; Xu, H.; Wu, J.; Zheng, J.; Dietrich, K.M. 3-D Data Processing to Extract Vehicle Trajectories from Roadside LiDAR Data. *Transp. Res. Rec.* **2018**, *2672*, 14–22. [[CrossRef](#)]
27. Nashashibi, F.; Bargeton, A. Laser-based vehicles tracking and classification using occlusion reasoning and confidence estimation. In Proceedings of the 2008 IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands, 4–6 June 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 847–852.
28. Wu, J.; Xu, H.; Zheng, Y.; Zhang, Y.; Lv, B.; Tian, Z. Automatic Vehicle Classification using Roadside LiDAR Data. *Transp. Res. Rec.* **2019**, *2673*, 153–164. [[CrossRef](#)]
29. Habermann, D.; Hata, A.; Wolf, D.; Osório, F.S. Artificial neural nets object recognition for 3D point clouds. In Proceedings of the 2013 Brazilian Conference on Intelligent Systems, Fortaleza, Brazil, 19–24 October 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 101–106.
30. Pang, G.; Neumann, U. 3D point cloud object detection with multi-view convolutional neural network. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 585–590.
31. Huang, L.; Yang, Y.; Deng, Y.; Yu, Y. DenseBox: Unifying Landmark Localization with End to End Object Detection. *arXiv* **2015**, arXiv:1509.04874.
32. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. *arXiv* **2013**, arXiv:1312.6229.
33. Li, B.; Zhang, T.; Xia, T. Vehicle Detection from 3D Lidar Using Fully Convolutional Network. *arXiv* **2016**, arXiv:1608.07916.
34. Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-view 3D object detection network for autonomous driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1907–1915.
35. The KITTI Vision Benchmark Suite. 2015. Available online: <http://www.cvlibs.net/datasets/kitti> (accessed on 21 November 2022).
36. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2015**, arXiv:1506.02640.
37. Girshick, R. Fast R-CNN. *arXiv* **2015**, arXiv:1504.08083.
38. Wang, H.; Yu, Y.; Cai, Y.; Chen, X.; Chen, L.; Liu, Q. A comparative study of state-of-the-art deep learning algorithms for vehicle detection. *IEEE Intell. Transp. Syst. Mag.* **2019**, *11*, 82–95. [[CrossRef](#)]
39. Tourani, A.; Soroori, S.; Shahbahrami, A.; Khazaei, S.; Akoushideh, A. A robust vehicle detection approach based on faster R-CNN algorithm. In Proceedings of the 2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA), Tehran, Iran, 6–7 March 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 119–123.
40. Xu, Q.; Zhang, X.; Cheng, R.; Song, Y.; Wang, N. Occlusion problem-oriented adversarial faster-RCNN scheme. *IEEE Access* **2019**, *7*, 170362–170373. [[CrossRef](#)]
41. Wang, Y.; Deng, W.; Liu, Z.; Wang, J. Deep learning-based vehicle detection with synthetic image data. *IET Intell. Transp. Syst.* **2019**, *13*, 1097–1105. [[CrossRef](#)]
42. Tremblay, J.; Prakash, A.; Acuna, D.; Brophy, M.; Jampani, V.; Anil, C.; To, T.; Cameracci, E.; Boochoon, S.; Birchfield, S. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 969–977.
43. de Melo, C.M.; Torralba, A.; Guibas, L.; DiCarlo, J.; Chellappa, R.; Hodgins, J. Next-generation deep learning based on simulators and synthetic data. *Trends Cogn. Sci.* **2021**, *26*, 174–187. [[CrossRef](#)]
44. Lakshmanan, K.; Gil, A.J.; Auricchio, F.; Tessicini, F. A fault diagnosis methodology for an external gear pump with the use of Machine Learning classification algorithms: Support Vector Machine and Multilayer Perceptron. Loughborough University Research Repository. 2020. Available online: https://repository.lboro.ac.uk/articles/conference_contribution/A_fault_diagnosis_methodology_for_an_external_gear_pump_with_the_use_of_Machine_Learning_classification_algorithms_Support_Vector_Machine_and_Multilayer_Perceptron/12097668/1 (accessed on 21 January 2023).
45. Torrey, L.; Shavlik, J. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*; IGI Global: Hershey, PA, USA, 2010; pp. 242–264.
46. Pan, W.; Xiang, E.; Liu, N.; Yang, Q. Transfer learning in collaborative filtering for sparsity reduction. In Proceedings of the AAAI Conference on Artificial Intelligence, Atlanta, GA, USA, 11–15 July 2010; Volume 24.
47. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A survey on deep transfer learning. In Proceedings of the International Conference on Artificial Neural Networks, Rhodes, Greece, 4–7 October 2018; Springer International Publishing: Cham, Switzerland, 2018; pp. 270–279.

48. Cunha, A.; Pochet, A.; Lopes, H.; Gattass, M. Seismic fault detection in real data using transfer learning from a convolutional neural network pre-trained with synthetic seismic data. *Comput. Geosci.* **2020**, *135*, 104344. [CrossRef]
49. Gao, Y.; Mosalam, K.M. Deep transfer learning for image-based structural damage recognition. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 748–768. [CrossRef]
50. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
51. Huang, J.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z.; Song, Y.; Guadarrama, S.; et al. Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7310–7311.
52. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. ShapeNet: An Information-Rich 3D Model Repository. *arXiv* **2015**, arXiv:1512.03012.
53. Visualisation Toolkit. Available online: <https://vtk.org/?msckid=c10ed6e5d13111ecbd9df41841d1b5f3> (accessed on 21 January 2023).
54. Yu, H.; Chen, C.; Du, X.; Li, Y.; Rashwan, A.; Hou, L.; Jin, P.; Yang, F.; Liu, F.; Kim, J.; et al. TensorFlow Model Garden. 2020. Available online: <https://github.com/tensorflow/models> (accessed on 21 January 2023).
55. RPLIDAR S1 Portable TOF Laser Range Scanner. Available online: <https://www.slamtec.com/en/Lidar/S1> (accessed on 21 January 2023).
56. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
57. Object Detection Yolo5 Implementation. 2022. Available online: <https://github.com/maheshlaksh05/Object-Detection-Yolo5-Implementation> (accessed on 21 January 2023).
58. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
59. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.-F. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 248–255.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.