Deep Learning vs. Traditional Computer Vision in Autonomous Driving

Richard Gilchrist, Charalampos Kouraklis, Chavinpat Naimee and Aris Tranganidas
Department of Electronic and Electrical Engineering
University of Strathclyde, Glasgow, G1 1XQ, U.K.

***Abstract*—In recent years, deep learning and neural networks have been the main approaches used to tackle complex autonomous driving (AV) challenges. However, in earlier years, traditional approaches have been successfully implemented in many AV sub-fields. This paper serves as a literature review on how and why deep learning has thrived over traditional methods in the main perception challenges of AVs, making a critical comparison and evaluation of the two approaches.**

This paper focuses on methods that have been proposed and implemented to solve perception problems related to AVs, including road lane detection, object detection and classification and depth estimation. Considering road lane detection, it was found that while traditional computer vision (CV) methods have achieved worthwhile results in the field, they do not cope well in non-ideal conditions. Deep learning methods were shown to perform adequately in challenging circumstances, eliminating the need for hand-crafted features, but requiring a large annotated dataset for training.

Considering object detection, both technologies showed to have high accuracy, however deep learning architectures achieve faster, real-time grade results. This comes at the cost of greater computational complexity and increased need of monetary resources. LiDAR technology has historically thrived in the area of 3D object detection and depth estimation, appearing to be the only valid approach for this case. However, recent advancements in deep learning and computer vision have proposed alternatives to LiDAR technology that can significantly decrease the overall cost of the AV without sacrificing accuracy in depth and 3D object detection.

Keywords: Computer Vision; Autonomous Vehicles; Perception; Deep Learning; Convolutional Neural Networks

I. Introduction

Fully autonomous vehicles are becoming incrementally capable of executing their task of total autonomy, requiring a decreasing amount of human intervention. With such technological development, four main areas of operation and research occur: a) perception, b) localization, c) path planning, d) control. The focus of this paper is the perception of AVs, i.e. how the vehicle perceives and understands its immediate environment. This could imply the identification of potential free space necessary for a manoeuvre, detection of road lanes, and of course entities of the road, like other vehicles and pedestrians.

Perception is made possible through computer vision (CV), and sensor fusion (SF). However, this paper concentrates on CV, comparing traditional developments with recent, state-of-the-art implementations. The combination of CV and a SF technology, under the name of LiDAR, will also be examined, along with more cost-effective alternatives of the latter.

CV is a research field related to the development of methods to allow computers to 'see' and perceive, as mentioned above. That is, to be able to understand the content of images and videos and derive useful, meaningful information. In recent years, CV research has moved from using statistical methods to using machine learning (ML) and deep learning (DL) techniques [1].

This paper will examine the evolution in the workflow of traditional CV, from manual feature representation, such as scale invariant feature transform (SIFT) [2], to an end-to-end architecture, where the feature representation and classification occurs without the need for manual intervention. Said architecture is the convolutional neural network (CNN) [3], which provides the basis for sophisticated, multi-classifiers like 'you only look once' (YOLO) v3 [4] and Mask R-CNN [5] that will be examined in the following sections.

After examining the broad background of the aforementioned technologies, this paper will look into their direct application in the major challenges of perception in AVs. These will include lane detection, obstacle recognition, depth estimation and 3D representation. Furthermore, notes will be made on the improvement, or lack thereof, between traditional implementations and recent more sophisticated systems.

Hence to inform the reader, this paper is organised as follows. Section II describes traditional methods and Section III presents the state-of-the-art techniques. Then, Section IV discusses the perception challenges in autonomous driving using computer vision by comparing the conventional and novel approaches.

II. Traditional Methods

The main difference between traditional CV and the deep learning CV methods is the workflow of feature extraction, preparation and engineering [1]. Traditional CV uses feature-based approaches to extract and appropriately mould the data into a form that will be recognisable to simple

classifying algorithms, such as naive Bayes or support vector machines, as shown in this section. The aforementioned difference is showcased in Fig. 1. Features are normally extracted using pre-learned shape features or descriptors such as Haar methods [6] or histogram of oriented gradients (HOG), which have been shown to perform well in human or pedestrian detection [7]. Further feature descriptors include, features from accelerated segment test (FAST) [8] and scale-invariant feature transform (SIFT) [2] which was shown to be outperformed by FAST in [8].

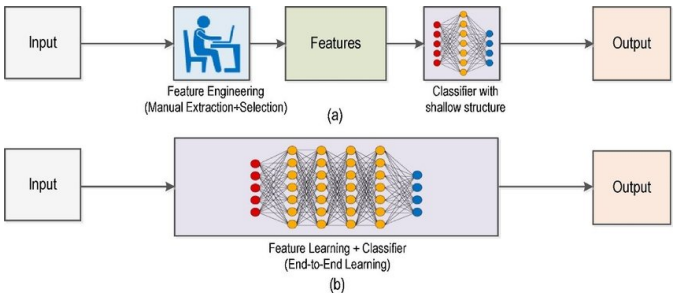

Fig. 1. Traditional CV workflow (a) vs DL workflow (b), taken from [9].

The aforementioned feature description techniques and their relevance to traditional CV in autonomous driving applications are investigated below.

A. Histogram of Oriented Gradients

HOG is described as an extracting feature descriptor that is used in CV for detecting an object. It is an effective method to extract features from the pixel colours for creating an object recognition classifier that uses the basic of image gradient vectors. This approach is related to edge orientation histograms [10] and SIFT [11].

It varies in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast standardisation to improve accuracy. The purpose of this approach is to define the image in a collection of local histograms. These histograms count gradient orientation occurrences in the local part of the picture. The methodology of HOG is explained below [12]:

a) Pre-process images, including normalisation and resizing,
b) Calculate the gradient vector in each pixel, its magnitude and direction,
c) Create histogram of orientation for every cell,
d) Normalise histogram within each block of cells to get a unit weight.

In many cases, the unsigned gradient is set from 0 to 180 degrees, which is a realistic alternative based on empirical studies [12]. The next step, orientation binning, is performed to measure the orientation histogram. For every cell, one histogram is computed by the number of bins. The particularity of this approach is to break the picture into cells. A cell can be defined as a spatial area, like a square with pre-defined pixel size. Then measure the histogram of gradients for each cell by accumulating votes in bins for each orientation. Voting may be weighted by the magnitude of the gradient so that the histogram considers the value of the gradient at a given level [7].

After all histograms have been calculated for each cell, the descriptor vector of an image can be built all histograms into a single vector. In the final step, cell histograms are normalised among a group of cells, called a block [7]. After this normalisation step has been completed, all histograms can be concatenated in a single vector function. Finally, it can be passed into a traditional machine learning classifier such as support vector machine (SVM) for classification tasks.

B. Support Vector Machine

After the detection step, the classification technique will be applied using the recognition system based on a machine learning method. SVM is one of the most potent and classical ways that can be used to identify objects in an image.

An SVM is a machine learning technique that is a powerful and robust approach for the binary classifier with both linear, non-linear problems, regression and outlier detection. The ‘kernel trick’ method was used to learn from data, and it looks for an optimal hyperplane or boundary between the possible results based on the transformations [13]. Besides, it could make entirely complicated relationships of data points and then determine a way to isolate the data based on their labels, especially for non-linear SVM, where the decision boundary is not a straight line [13].

Linear kernel SVM is a fundamental method of SVM in which two classes can be classified using a straight line. The decision boundary of the SVM does not only separate the two groups but also stays as far away from the other training data as possible [14]. Fig. 2 demonstrates that D1, D2 and D3 can divide the data nicely, but D2 would be considered better due to the fact that it stays as far away from the other training data point as possible, to achieve maximum distances from both groups. In Fig. 2, the circles and triangles define the position of data points and x-axis and y-axis represent the unit of data that depend on each dataset.

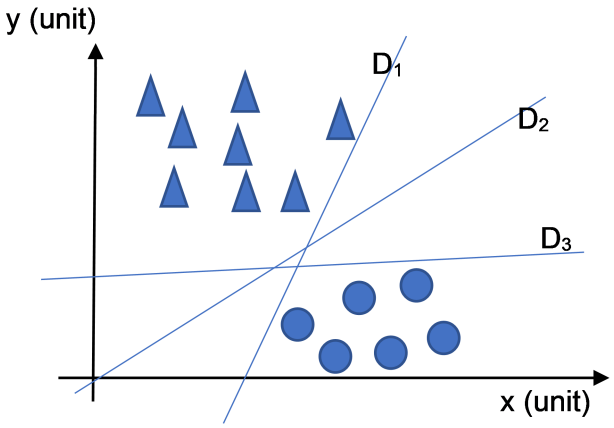

Fig. 2. SVM classifier using a linear kernel

III. Deep Learning Methods

As mentioned above, the difference between traditional and state-of-the-art DL methods is in the workflow of feature extraction and representation. The image acquisition occurs through cameras, and the classification result through a less or more sophisticated classifying architecture respectively. DL methods introduce an end-to-end approach, through deeper networks, which embody the feature extraction and representation, as explained below.

A. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) constitute a category of general artificial neural networks, mainly implemented for image-related operations. The input image goes through a series of filters (feature detectors-kernels), responsible for identifying key feature characteristics in the image and creating the corresponding feature maps. A kernel is a matrix of values called weights, which are trained and responsible for identifying features [1]. The operation of a CNN is indicated by its name and involves the convolution of the input image with a kernel to determine if that feature is present, shown in Fig. 3.

The value indicating how likely a feature is present in a certain area of the image is calculated by the dot product of the kernel and the pixel values in the said area, and occurs until the entire image is covered [3].

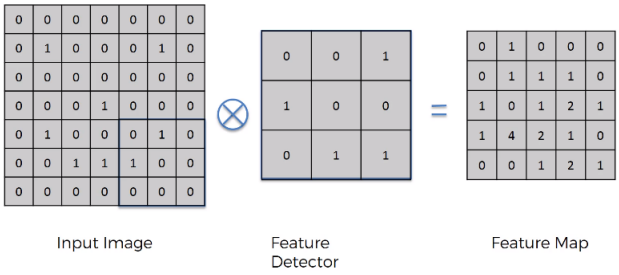

Fig. 3. Convolution operation

The above process comprises the convolutional layer of the network. On top of the convolutional layer which is summed with a bias term, a rectified linear unit (ReLU) is applied in order to increase non-linearity in the image [1]. Dense layers utilising ReLUs (Fig.4), have neurons that either activate or not, resulting in favourable results for image related tasks, whilst being less prone to the vanishing gradient problem [15].

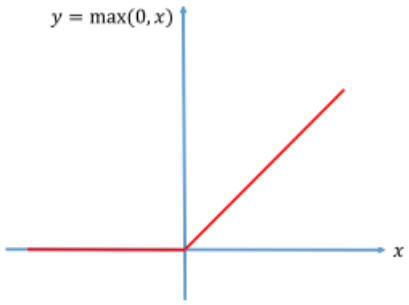

Fig. 4. The ReLU activation function, obtained from [3]

Then a pooling layer is consequently implemented, so that from every feature detector, a corresponding feature map is created. Pooling is mainly used for creating spatial invariance, i.e. allowing the system to identify features, even if they are not ideally represented on the image, in addition to significantly reducing the size of the said image, and avoiding overfitting [3]. Pooling also reduces the amount of memory consumed by the network [1]. Many different types of pooling are implemented in object recognition, highlighted by glabal max pooling and global average pooling as explained thoroughly by Dominik Scherer et al. [16].

The final stage is a flattening operation that reshapes the array of image pixel values into a vector in order to be fed into the consequent fully-connected network. The flattening layer outputs the probabilities of the existence of certain features in the image into the network.

Another important part of CNNs is dropout regularisation, where a certain user-defined percentage of neuron weights and biases are set to zero, so that the network can proceed to adjust its weights without taking into account those

neurons [3]. Dropout is found in the fully-connected stage of the CNN and its purpose is to increase generalisation and avoid over-fitting. The fully connected CNN with all the aforementioned stages is shown in Fig. 5.

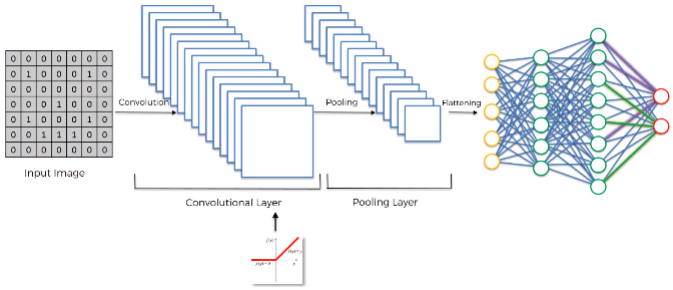

Fig. 5. The fully connected CNN

1) YOLO v3: YOLO v3 is a deep CNN, responsible for identifying and classifying real world objects from the Common Objects in Context (COCO) dataset. Input images are divided into an $S \times S$ cell grid, where each grid is responsible for predicting the object whose center falls within it. Each grid cell predicts B bounding boxes, along with C class probabilities, which are the probabilities of the class that the object is thought to belong in. The total output consists of the coordinates of the centre of the image (x and y), the height and width of the box (h and w), and the confidence of the class that it has been categorised into. Thus the total output of YOLO is S x S x B x 5 [4]. The bounding box prediction is shown in Fig. 6.

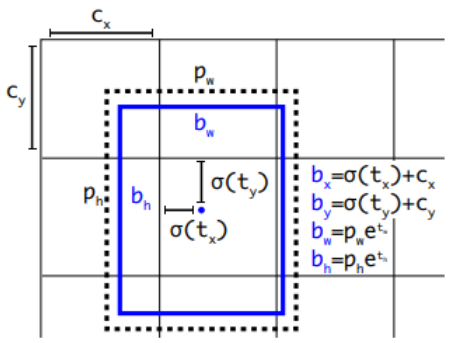

Fig. 6. YOLO v3 bounding box predictions [4]

YOLO v3 consists of 53 convolutional layers, including 3x3 and 1x1 kernels, responsible for feature extraction. The set of said layers is called Darknet-53. The network outperformed all similar networks, such as RetinaNet [17], while maintaining a state-of-the-art mean average precision (mAP).

2) Mask R-CNN: Mask R-CNN [5] may be considered as the evolution of many CNN-based architectures. It is able to handle computer vision challenges that in the past had to be processed separately, using a combination of different algorithms. More specifically, Mask R-CNN provides object detection (bounding box), object classification (class labels) and image segmentation (via a mask) in parallel processes (branches). In this section, we will analyse the main layers of the Mask R-CNN architecture, which can be seen in Fig. 7

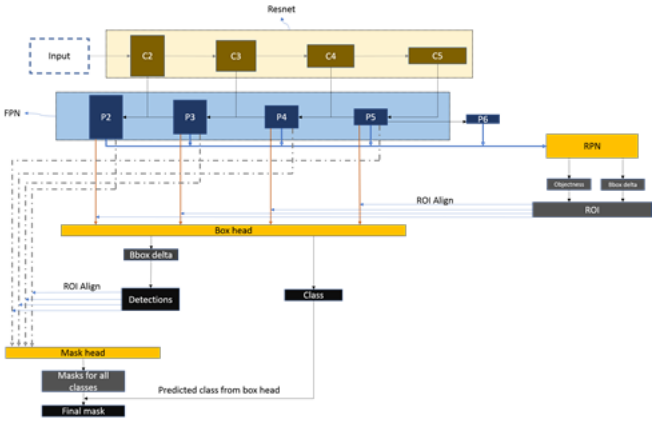

Fig. 7. Mask R-CNN architecture, a simplified view. Obtained from [18]

Mask R-CNN is an extension of faster R-CNN [18] and it consists of two main stages: a region proposal network (RPN) which extracts regions of interest (ROIs), that are received as the input to the bounding box regression network; secondly, a CNN which runs in parallel with the RPN in order to generate the masks around the objects proposed by the RPN.

In order to perform detection, classification and segmentation for the proposed objects in the same architecture, Mask R-CNN consists of several layers. In the following sections, a description of the basic processes of each layer is presented.

- Feature pyramid layer: This layer's structure is based on a ResNet-101 network whose architecture is slightly differentiated in order to extract features at different scales. Its purpose is to detect and extract scale invariant features which are shared trough the different layers of the pyramid. A visual representation of this layer is shown in Fig. 8.
- Region proposal network: In order to deal with the problem of the unknown number of objects that the network has to detect, in this layer, Mask R-CNN uses a sliding $n \times n$ window across the features that were extracted by the FPN. These windows are also called anchor boxes as they are centered around a center point, called an anchor. The purpose of this layer is to perform as a binary classifier, extracting the anchor box and its binary class, either foreground or background.
- Region of interest classifier and regressor: Inputs to this layer are the anchor boxes that are created by the RPN.

Then, the ROI classifier predicts the actual class for each proposed box. Moreover, the ROI regressor is responsible for the refinement of the bounded boxes coordinates, once the class is predicted.

- Mask - Image Segmentation: Regarding all the previous layers mentioned, similar architecture had already been implemented by Faster R-CNN. However, Mask R-CNN's addition is the masking, that is generated by the image segmentation branch. This layer's input is the bounded boxes coordinates and classes generated by the ROI layer. After passing the the proposed boxes through a CNN by the use of bilinear interpolation, the mask is is created around the detected object.

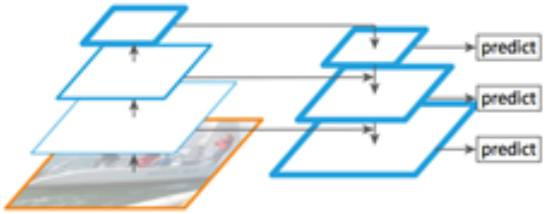

Fig. 8. Feature pyramid network [19]

B. Perception with LiDAR

Although LiDAR is not a computer vision algorithm, since it is referenced in most of the upcoming parts of the paper, we make a brief introduction of this technology in this section. In 2005, David Hall, founder of Velodyne Acoustics, was inspired to build a spinning LiDAR to mount on top of a vehicle. His idea was showcased in DARPA 2007 [20], raising the interest of many companies which actively contributed to the evolution of autonomous driving. Many advancements have been achieved in the improvement of LiDAR technology for AV since then, leading it to becoming one of the main components of state-of-the-art projects in autonomous driving.

LiDAR is a remote sensing technology which illuminates objects using pulses of laser light and calculates the distance by measuring the reflected light. Its purpose is to locate objects in its surroundings by depicting information in a three dimensional (3D) cloud of points which is organised in layers as shown in Fig. 9. LiDAR has been widely used in industrial applications and academic research due to its long range and satisfactory accuracy, especially in distance estimation.

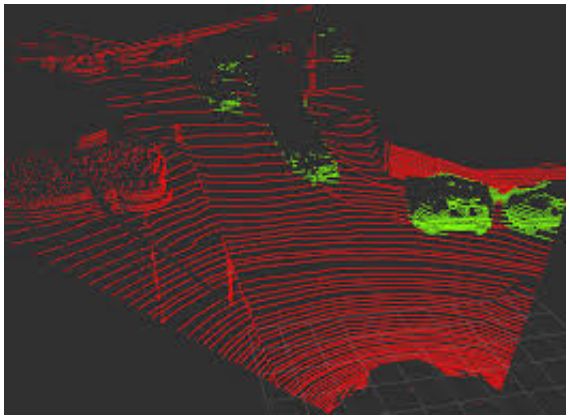

Fig. 9. LiDAR Cloud Point for Autonomous driving [21]

Many breakthrough research papers suggest that LiDAR should be used hand-in-hand with the state-of-the-art algorithms of computer vision in order to enhance accuracy in certain feature extraction. More specifically, LiDAR can be used to refine the initial bounding boxes of the objects detected. This approach makes sense, as the 3D point cloud is capable of extracting more accurate information about the depth and distance of an object than can be obtained from 2D images coming from cameras.

However, it should be taken into account that LiDAR is an expensive high-end technology system that can considerably increase the overall cost of the autonomous vehicle. Moreover, due to its moving parts and the fact that the equipment is based on the roof of the vehicle, danger of potential damage of the equipment raises further concerns.

IV. Perception Challenges in Autonomous Driving using Computer Vision

A. Road Lane Detection

Lane detection is an important task in the development of systems to increase driving safety. As a component in an advanced driver assistance system (ADAS), a lane detection system may find application in a lane departure warning system. Such a system may passively alert the driver when the vehicle is veering off course, or actively take over some vehicle functions in order to maintain the position of the vehicle relative to the boundaries of the road lane it occupies. A 'lane keeping' system such as this, aims to prevent the possibility of colliding with vehicles in neighbouring lanes. Having knowledge of the shape of the road ahead allows an autonomous vehicle to execute path planning and adjust the trajectory accordingly.

Although there are several sensing modalities available for use in automotive applications, it seems logical that the perception problem be approached using vision, because this is how human drivers solve the problem. Additionally, in contrast to objects such as road signs or pavements, lane

markings have very little structure – simply consisting of paint on the road. As a result, modalities such as radar or LiDAR may not prove to be suitable for the task of identifying lane markers or boundaries.

Broadly, roads may be categorised as structured or unstructured. An example of a structured road is a motorway, where there are clearly marked lane boundaries. In contrast, unstructured roads such as country roads, may have few or no lane markers. The most obvious difficulty occurs when the lane markings are absent altogether. A similar but less severe scenario may be when they are discontinuous or faded. The weather and environmental conditions, as well as the surroundings and situational factors may introduce difficulties. Examples of such factors include:

- general poor visibility due to rain or fog
- poor road surface visibility due to snow or water
- shadows and reflections
- occlusion of road lane markings by other objects
- strong sunshine or darkness

There are a number of features which may be used to try to identify lane markings, including colour, edges and shapes. Edge detection helps to identify and retain the useful structural information in an image, while allowing useless or redundant information to be discarded. By retaining only some fraction of the original data, the processing time may be reduced. Edge detection in an image may be approached by computing the intensity gradient at a given point in the image. In practice the intensity gradient may be calculated by performing convolution operations. Examples of commonly used edge detection methods include Sobel, Canny and Prewitt [22].

The Sobel edge detection approach uses two 3x3 separable kernels – one to detect vertical edges and the other, horizontal edges as described in [23]. Convolving the image with these two kernels gives approximations of the derivatives, G_x and G_y, where A is the original image and $*$ represents a convolution operation. Equations 1 and 2 are used to calculate the magnitude, G, and orientation, θ, of the gradient at each point, respectively.

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * A$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

$$G = \sqrt{G_x^2 + G_y^2} \tag{1}$$

$$\theta = arctan\left(\frac{G_x}{G_y}\right) \tag{2}$$

The Canny edge detection method is motivated by having a low error rate and good localisation of the detected edges. Thresholding may be combined with edge detection in order to refine the detected edges. However, achieving an appropriate threshold value that maintains useful features while discarding irrelevant information is non-trivial. In [24], adaptive thresholding for extracting lane markers was implemented with the threshold value being dependent on the characteristics of neighbouring pixels, rather than using a global threshold value. The proposed method was robust to cases which would affect a global threshold, such as street lights, or lights of other vehicles.

Lane markings generally contrast strongly with the road surface, typically being yellow or white in colour. As a result, colour has been used as a feature for identifying lane markings. While images from cameras may typically use the RGB colour space, some work has been done involving the use of colour space transformations in order to identify white and yellow lane markers using different colour spaces such as YCbCr or HSI [25]. In [26], the saturation (S) and intensity (I) channels of the HSI colour space were found to be sufficient for detecting lane markers of different colours. Compared to RGB, the authors report a reduction in mis-detection and an increase in efficiency due to the requirement to process only two channels instead of three.

Changes in illumination can affect the intensity gradient between the road surface and the lane markings, adversely affecting lane detection algorithms. In [27], a gradient-enhancing method for illumination-robust lane detection is proposed, with the detection rate averaging 96%. A method for removing shadows from RGB images is proposed in [28]. The method proposed in [29] utilises the YCbCr colour space to identify yellow and white lane markers, with an average detection rate of 93% achieved under various illumination conditions.

It is desirable to select some sub-section of the full image as a region of interest (ROI). This is in order to reduce the size of the image that must be processed, as well as to minimise useless information such as background and sky which may lead to erroneous detection of features. In [24], the image is cropped using fixed points that correspond to the part of the image above the car bonnet and below the vanishing point. One simple approach to determining the ROI is to select the lower half of the image [30]. While simple, this method may result in features being unclear or otherwise adversely affected by shadows or occlusion [29]. In [29], lines in the image were detected using Canny edge detection and the Hough transform, with the vanishing point being determined using a voting system considering the points where the detected lines intersect. The ROI is then considered to be the sub-section of the image below the vanishing point.

Visual perspective means that parallel lane markers appear

to get closer together as the distance from the camera increases. The point in the image at which the lane markers converge is referred to as the vanishing point. Fitting lines to the identified lane points in an image from the camera's point of view may require high-order polynomials if curved lanes are encountered, while lower-order polynomials are preferred [31]. One approach that compensates for this effect is inverse perspective mapping (IPM). Given an image or a sub-section identified as an ROI, the inverse perspective mapping provides a bird's-eye view of the road, where the lane markers are vertical, parallel and evenly spaced, as illustrated in Fig. 10. Using this approach, 2nd or 3rd order polynomials may be used to fit curved lanes [32].

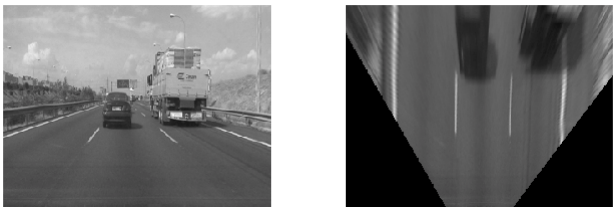

Fig. 10. Inverse perspective mapping [33]

The transform requires a set of parameters and if a fixed set of parameters is used, the method is not robust to changes in roll or pitch angle, corresponding to changing road camber or slope, respectively. An approach proposed in [32] removes the fixed nature of the transform parameters and instead uses a neural network to predict parameters that allow pixels in the transformed image, belonging to lane markers, to be optimally fitted using a 2nd or 3rd order polynomial. Learning the transform parameters in this way ensures the transform is robust to road plane changes.

The Hough transform is a commonly-used feature extraction method applied to the problem of lane detection. Lines in an image may be represented in Hough parameter space by two parameters, r and θ, as shown in Fig. 11, where r is the distance from the origin to the closest point on the line, and θ is the angle between the horizontal axis and the line between the origin and this closest point. A voting procedure is used to determine the properties of the most prominent lines, as described in [34].

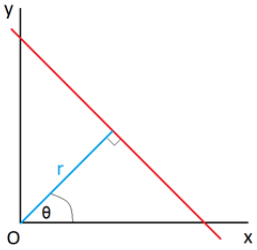

Fig. 11. A line in image space parameterised using r and θ

The methods discussed thus far are limited by the fact that they rely on hand-crafted feature-based methods and the performance of these methods is often adversely affected by challenging conditions such as illumination or weather conditions [35]. The remaining discussion on lane detection focuses on deep learning methods, based partly on CNNs, as previously described.

The presence of rain on the road presents a challenge due to the introduction of reflections, as well as distortion to the lane markers. Lee et al [35] propose an end-to-end multi-task network, with their work specifically focusing on adverse weather and illumination conditions. The authors state that existing datasets did not adequately represent the situations of interest, so they developed their own dataset consisting of 20,000 images each with manually annotated lane and road marking classes, along with vanishing point. The dataset consisted of images representing four scenarios – daytime conditions with no rain, with rain and with heavy rain, as well as night conditions.

During training, data augmentation was utilised, doubling the size of the dataset by flipping each image horizontally. In addition to increasing the size of the dataset, this operation simulates driving situations on the opposite side of the road. Lane and road markings e.g. arrows, were manually annotated in order to produce pixel-level mask annotation for each object, with each pixel containing one of 17 available class labels. To prevent thin annotated features from vanishing through the convolutional and pooling layers, class labels are assigned to grid sections of size 8x8 within the image. In addition, the vanishing point in the image was manually annotated.

The proposed network performed four tasks: grid regression, object detection, multi-label classification and vanishing point prediction. The vanishing point prediction task was included in order to add global contextual information which is useful when predicting lanes that are invisible either due to occlusion or illumination conditions. The network structure consisted of 5 shared convolutional layers, followed by 4 separate branches each containing 3 convolutional layers. Inverse perspective mapping was utilised, then points which were identified as potential candidates for lane segments were clustered prior to a line-fitting regression stage.

The model, VPGNet, was found to be capable of effectively identifying and classifying road lanes and markings under varying conditions, in real-time. When compared to the fully convolutional network, FCN-8s, proposed in [36], VPGNet performed better than FCN-8s in all adverse weather scenarios. Additionally, the forward pass time for VPGNet was significantly shorter than for FCN-8s.

Neven et al [32] frame the lane detection problem as an instance segmentation task. In contrast to semantic

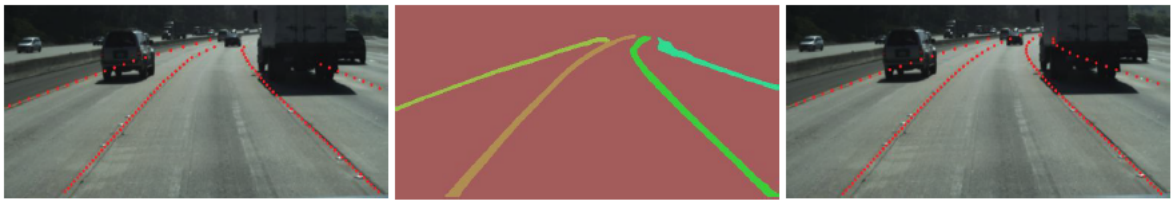

Fig. 12. Results from [32]. *Left:* Ground truth points. *Centre:* Output from LaneNet. *Right:* Prediction after lane fitting.

segmentation, which assigns a class label to each pixel in an image, instance segmentation identifies multiple objects from the same class as individual *instances* of that class. In this context, each lane in the image forms its own instance. A branched, multi-task network was designed, consisting of a lane segmentation branch and a lane embedding branch. The segmentation branch is trained to generate a binary pixel-level output corresponding to either background or lane marker, while the lane embedding branch clusters the segmented lane pixels into different instances.

The dataset used was the tuSimple lane detection dataset [37], where images are annotated with the x-position of a lane at discretised y-positions. In order to construct the ground truth for training, these annotations were manually connected to form a complete line per lane. The ground truth annotations were drawn even when the lane markers were not visible due to occlusion, environmental conditions or simply due to the absence of markers. This was done in an attempt to make the method robust to adverse conditions.

To obtain a lane parameterisation, inverse perspective mapping was used. Instead of using a fixed transformation matrix, a separate network, H-Net, was trained to produce the optimal transform coefficients such that the lane could be fitted using a low-order polynomial. This approach is robust to ground plane changes, unlike other approaches which use a fixed transformation matrix. The mean squared error (MSE) was calculated for predictions made on the validation set using 2nd and 3rd order polynomials, with three transform options – no transform, fixed transform and transform using H-Net. The lowest MSE was achieved using a 3rd order polynomial to fit the points after transformation generated by H-Net. An example of the network's output is illustrated in Fig. 12, where it can be seen that lane markings were predicted despite being occluded by a truck.

B. Object Detection and Classification

Object detection is one of the most popular topics in state-of-the-art computer vision for autonomous driving. The autonomous vehicle must enable to both locate itself in an environment and distinguish objects (moving and stationary) and keep track of them. The vehicle uses exteroceptive sensors such as LiDAR, cameras, inertial sensors and GPS to get information about the environment. These sensors provide the information that can be used and fused to locate the vehicle and track objects in its environment, allowing it to successfully navigate from one point to another.

There are three steps of the methodology of object detection and tracking for autonomous driving, which are localisation, mapping, and tracking objects. Localisation is the method of determining the location of the autonomous vehicle. Mapping requires being able to get the environmental context. Tracking the moving object requires being able to recognise and track the moving objects while travelling.

To perform object detection, the information of the environment generated through the LiDAR and camera is required. Objects are identified, marked, and the distance and direction of the object relative to the autonomous vehicle are determined using the information from these sensors. The images from the camera are used to detect and classify objects, but the LiDAR is used to determine the position or distance of the object in relation to the vehicle. The laser scan, combined with the vehicle pose is used to create a 3D point cloud of the environment. This is projected onto the image.

This section will mainly review the state-of-the-art methods of object detection and classification for an autonomous vehicle. However, some traditional approaches will also be discussed briefly.

1) Process Overview of the Detection and Classification System: Fig. 13 shows the block diagram of the object detection system from the RobotCar [38]. Images collected by the camera are used to detect and identify the object in an image. CNNs are used to detect the objects in the picture. When objects are identified, they are stored in a database. Then detected object is matched with objects in the database to find a correlation or added to the database as a new entity.

When object detection for the image has been performed, the data from the laser scanner is projected on the image. This allows the car to determine the distance and position of the detected objects. This knowledge is combined with the condition of RobotCar and an extended Kalman filter (EKF). Both the object state and the RobotCar are modified using EKF, which allows for a combined position and tracking of objects in the area [38].

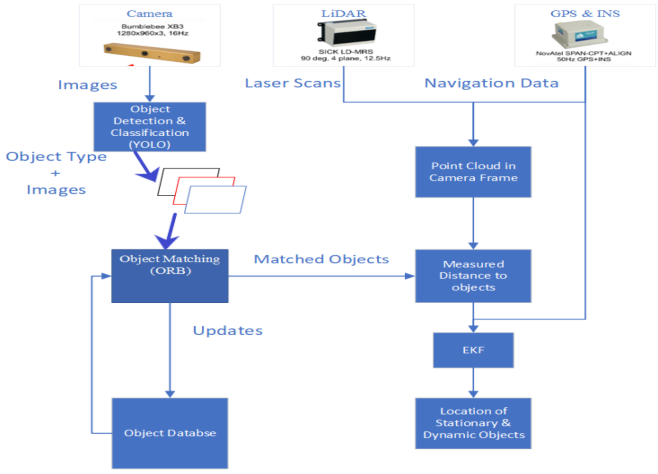

Fig. 13. Process diagram of object detection system, obtained from [38]

2) Traditional Method using HOG-based SVM: HOG-based SVM technique is a standard method that has been widely used until some years ago. There are two main steps, which are to detect and extract the features from an image using HOG, and then use the SVM to classify objects, as shown in Fig. 14. To make this method effective in real-time classification, a sliding window search is applied over an input image and classifying the object in the window [39]. Han et al [39] said to enable to detect objects at all position within one image, the detectors will be re-applied to any possible direction of the rectangular window.

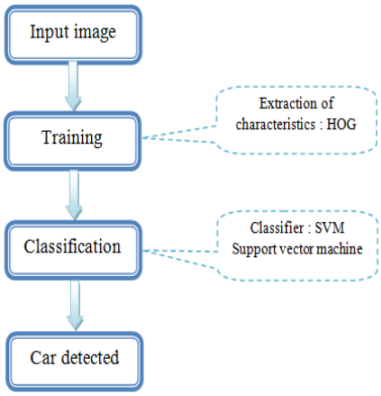

Fig. 14. Car detection diagram, taken from [14].

There are many features in an image that could be extracted and used in classification. The HOG feature descriptor will extract only essential information from an image. The HOG provides a strong generalisation by grouping only visually identical images together. With an SVM, the results in a decision feature that recognises object and non-object patterns efficiently in images under various conditions and results in consistent output on some different datasets [39].

After training using the SVM classification, the objects, cars, are detected in bounding boxes as shown in Fig. 15. [14] performs and analyses the model on many pictures with different environmental factors, including severe weather and lighting conditions. Fig. 15 illustrates the excellent outcomes from detector and classifier. It can be seen that some bounding boxes enclose only some part of a car, but it is considered an acceptable result. [14] obtained around 80% accuracy, which still generates some incorrect results. Fig. 16 shows samples of objects detected incorrectly. It recognised other objects as a vehicle, which is a false positive prediction.

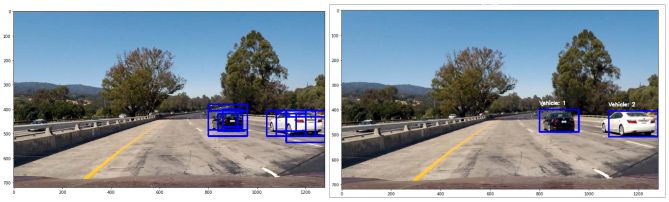

Fig. 15. Detected car using HOG based SVM, taken from [14]

Fig. 16. Incorrect classification by HOG based SVM, taken from [14]

3) Deep Learning Method Using YOLO Algorithm: Automatic object detection and classification is an important key for an autonomous vehicles. The novel techniques could be obtainable with high precision in real-time. Thus, deep learning methods can deal with the real-time situation.

There are many CNN architectures that have outperformed the traditional approaches that use feature extraction. CNN discovers the features of the object during the training step and can generalise the features of the object and identify it. The R-CNN, Mask R-CNN and YOLO algorithms are based on CNN architectures and perform well in real-time with multiple object detection. [38] suggested that YOLO performs faster than other algorithms and still maintains high accuracy.

Fig. 17 shows the result of object detection using YOLO (left) and laser scan projected in a detected image (right), which this information was obtained from the RobotCar [38]. Every object detected has a bounding box that locates the object in the image. This knowledge helps to find the distance between objects and the autonomous car. The projected laser scans can be separated so that only the laser bouncing

back from the identified objects remains after the objects are identified, labelled and located in the image.

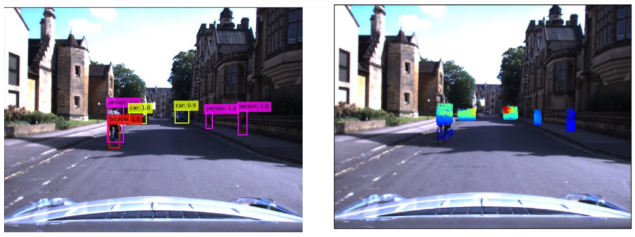

Fig. 17. The result of object detection using YOLO (left) and laser scan projected in a detected image (right), taken from [38]

Fig. 18 displays the image of the objects observed as a point object and vehicle view. The middle of the laser scan projected onto the target is taken as the distance of the target from the RobotCar [38].

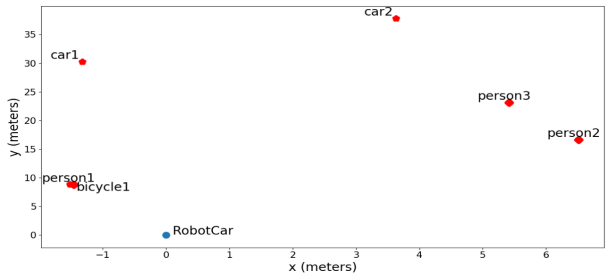

Fig. 18. 2D Map of detected objects in Vehicle view, obtained from [38]

4) Comparison Between HOG with SVM and YOLO for Object Detection and Classification: The above algorithms, HOG-based SVM and YOLO, are well-known for object detection. Even though both methods perform well for detection and classification, YOLO works faster for real-time detection and prediction. It means YOLO-based system is more than 20 times faster than HOG with SVM system, because it performs without a sliding window technique and passes though the model directly. Nevertheless, The drawback of YOLO is more computational complexity and expensive than the traditional HOG method.

From the research [40], they experimented with these algorithms using the same data, which consists of 400 images with 266 vehicles. The results are shown in 3 ways; object classified correctly, an object is not detected, and non-vehicle is identified as a vehicle. [40] said that, HOG-based SVM provided more wrong detection, non-vehicle is recognised as a vehicle, than YOLO. The HOG-based SVM technique sometimes predicted pedestrians, road signs and trees as cars. They also suggested that if a pre-road lane detection is implemented as YOLO can do, this issue might be solved and obtained a better result. In [40] study, YOLO achieved 81.9% accuracy, obviously outperforming HOG-based SVM which achieved 57.8%. However, both methods provided poor accuracy for objects that are far away and small.

Even though both techniques failed to detected objects, HOG-based SVM might enable to recognise using high-resolution images and videos. While it is challenging to use high-resolution data with YOLO because YOLO downsizes images before learn data.

5) Discussion of Noisy Conditions for Object Detection and Classification Using YOLO: In [38], the model of detection, classification, and tracking for the self-driving car was developed. It proposed the results of object detection in a road with three different conditions, which are normal, night and snowy conditions. The objects were detected and identified by using the YOLO algorithm. YOLO is doing well in all three conditions. It can be noticed that YOLO missed some objects in two more dangerous conditions; however, it did not miss objects that were close to the car and not affect the safety. As a result, images that consist of the poor lighting conditions are able to influence the accuracy of YOLO algorithm.

Fig. 19 shows the object detection by the YOLO algorithm in snowy condition. The lighting circumstance during the snowy is worse than the usual condition. This issue can make the object on the image unclear or dim. From Fig. 19, it can be seen that the YOLO does not detect some pedestrians that are far from the car, but it can detect anything close to the car. [38] said that YOLO seems to work in a snowy situation.

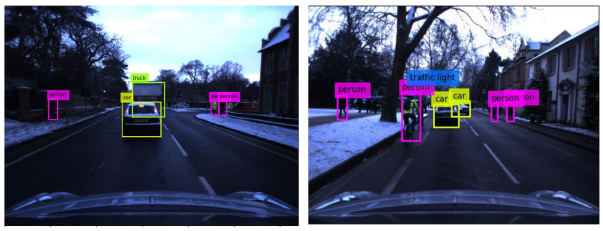

Fig. 19. Object detection in snowy condition using YOLO, obtained from [38]

Low light at night can make a camera not good enough to record the image. Fig. 20 demonstrates the performance of YOLO in the night period. The left image shows that it can detect a person who rode a bicycle but missed the bicycle. While the image on the right cannot identify the bus that is the same one in the left image, but it is far away from the car. Moreover, the blurred bus in the right image also can be detected. Even though in low light conditions, the efficiency of YOLO for object detection and classification performs good, as it is capable of detecting objects in the vicinity of the car that needs to be monitored.

Fig. 20. Object detection in night-time using YOLO, obtained from [38]

C. Depth Estimation

Depth estimation has been one of the key challenges in autonomous driving. From an industry perspective companies have followed different approaches to estimate depth and distance of objects. Leading companies in the area of autonomous vehicles such as Uber, have been using LiDAR technology, a technology which is the profound leader in depth and distance estimation. However, this is considered a controversial topic by experts in autonomous driving, such as Tesla's engineers and innovators, who have stated that implementing LiDAR technology in AV will only lead to a dead end due to its cost and impact on the vehicle design.

In the past DARPA challenges [41], LiDAR had a key role in the field of intelligent sensing of autonomous vehicles. During the 2007 DARPA Urban Challenge [20] two fundamentally distinct approaches to the challenge of depth and distance estimation were showcased: one using high-end LiDAR units, while the other achieving 3D reconstruction of images by the use of CV. Since then, these two alternative approaches have continuously competed with each other and evolved in order to provide advanced solutions in regards to depth and distance estimation.

Reliable and robust distance estimation is crucial in AV as it could affect decision making such as when to take a turn, the distance of a pedestrian or a cyclist etc. Without any doubt, LiDAR technology can provide highly accurate 3D point clouds of the surrounding environment, however there are many reasons alternatives to LiDAR are desirable. The high cost of LiDAR equipment (more than $60,000 per unit), moving parts of the equipment and the fact that it is attached on the roof of the vehicle are some of the factors that made researchers seek alternatives.

Hence, recent efforts have focused on trying to replace LiDAR systems by cheap on-board cameras. In this section we will analyse major advancements of CV algorithms towards 3D reconstruction of a 2D image and examine if such an approach can yet be considered reliable for autonomous driving tasks.

1) Monocular-based 3D Object Detection : This approach is based on depth estimation by the use of a single 2D image coming from an RGB camera. Monocular 3D object detection is a challenging problem. Yet, we can identify geometric features including orientation and keypoints in the 2D image related to 2D bounding boxes to provide reason in 3D space. In this section, we present the latest and most crucial research papers related to monocular-based 3D object detection.

In 2016, Mono3DOD [42] focused on generating a set of 3D object proposals using ground-plane assumption, followed by evaluating each proposed box using semantic segmentation, contextual information, as well as size and location priors and object shape. These object candidates are eventually scored by a CNN, leading to the final detections. At the time this paper was published, experimental evaluation showed that it significantly outperformed all monocular approaches and achieved the best detection performance on the KITTI challenge [43]. Furthermore, it set the base for many other papers that followed up, using the ground-plane approach.

A year later, the Deep MANTA paper was published [44]. Its approach comprised two main steps. First, input images are passed through the Deep MANTA network (a cascaded Faster R-CNN architecture) in order for the 2D bounding boxes to be extracted. The second step is the correlation of the bounding box created and the 3D vehicle dataset that Deep MANTA has been trained on in order to extract the 3D orientations. A major advantage of this method is the underlying idea that 3D information of vehicles can be retrieved since vehicles have specific and well-known geometry. Thus, this approach is able to extract the 3D points even when vehicles are partially occluded, by the use of regression.

2) Stereo depth (disparity) Vision for 3D Object Detection: Stereo vision involves the simultaneous use of two RGB cameras, detecting the same object in order to estimate its depth and distance. Although there are issues such as in some cases an object might be seen only from one of the cameras, or that the greater the distance from an object, the wider the two cameras should be placed (something that is not possible in AV), yet, stereo depth approaches systematically achieve better accuracy compared to monocular methods.

One approach would be to generate 3D proposals by encoding object size priors, placement of objects on the ground plane as well as depth informed features (free space, point cloud density) into an energy function. This process was suggested by [45] in 2019. It is very efficient, since all of the features can be computed in constant time with 3D integral images. 3D proposals are then used to regress the object pose and 2D bounding boxes using Fast-R CNN [46].

The architecture of this network for joint 2D object detection and orientation estimation is built upon Fast R-CNN, which shares the convolutional features across all proposals and uses an ROI pooling layer to compute proposal features. The Fast-R CNN model is further parameterised

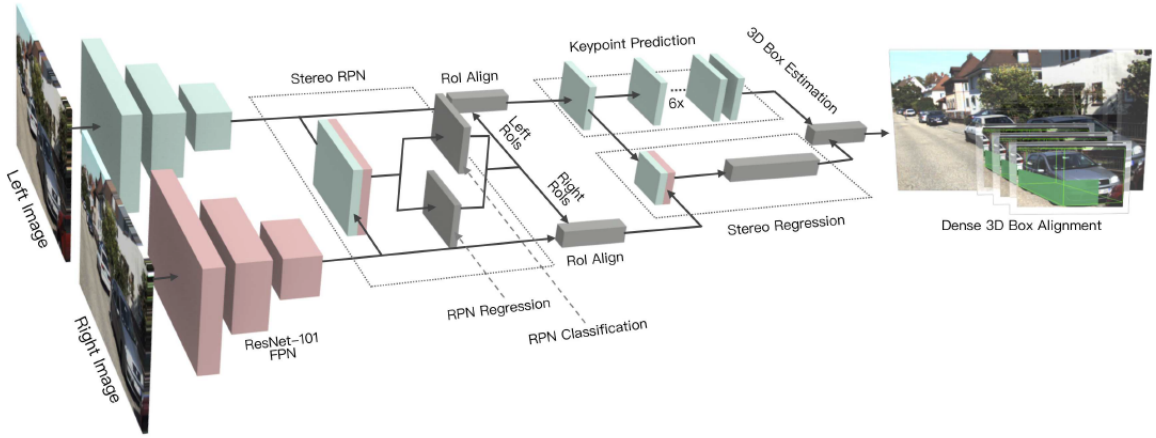

Fig. 21. Stereo R-CNN architecture [47]

by adding a branch after the last convolutional layer, and an orientation regression loss so that it jointly learns object location and orientation. However, the main drawback of this approach is that it does not take advantage of the dense object constraints in raw stereo images.

Recently, [47] presented a novel idea on stereo vision. Based on Faster R-CNN, Stereo R-CNN exploits the sparse and dense, semantic and geometry information in stereo images. It simultaneously creates 2D bounding boxes for both left and right cameras. Extra branches are added to the RPN layer in order to extract information about keypoints and viewpoints and combine it with left and right 2D bounding boxes in order to create the final 3D bounding box.

Finally, the 3D bounding box is accurately recovered by a region-based photometric alignment using left and right ROIs as shown in Fig. 21. This approach achieved to almost double average precision at most of the KITTI tasks and closed the gap with the LiDAR results.

Despite the many advancements in both monocular and stereo image-based 3D object detection, there is still a big gap between these methods and LiDAR-based approaches. Factors such as the physical differences between LiDAR and camera technology certainly contribute to the differences in performance. For example, the error of stereo-based 3D depth estimation increases quadratically with the depth of an object, whereas for LiDAR approaches this relationship is almost linear. However pseudo-LiDAR authors [48] suggest that this gap is not only due to the physical differences of camera and LiDAR systems.

Pseudo-LiDAR is the most well-known work in the field of 3D image reconstruction from monocular (and stereo vision) images, presented in 2018. It is mainly based on the idea that proper representation is crucial for depth estimation. Authors state that using 2D convolution on depth maps is not the right approach, as neighboring pixels in a 2D image could have a considerable distance in 3D space, in contrast to LiDAR, where 3D point cloud representation of object sizes and shapes is invariant to distance.

More specifically, they state that the use of 2D convolutions is based on the misleading assumptions that local neighbours have meaning and also that all neighbourhoods of the 2D image can be operated in an identical way. Although these assumptions would be valid in a 2D space, they are not valid when there is depth in an image. Hence, authors suggest an alternative and more efficient approach for 3D detection, by converting the image-based depth maps (generated by the stereo vision algorithm) into pseudo-LiDAR representations and applying LiDAR algorithms for depth estimation (Fig. 22).

Authors practised their approach on KITTI challenge combining many, and diverse algorithms in their pipeline both for monocular and stereo vision. Experimenting with multiple combinations of depth estimation and 3D object detection algorithms, authors achieved state-of-the-art results in every case. This proves that high accuracy is based on their approach of transforming images into pseudo-LiDAR form, rather than the specific algorithms which were used to process these inputs.

The pseudo-LiDAR approach appears promising to close the gap between LiDAR and CV, while showcasing the potential of CV as a standalone approach. Furthermore, recently, authors of pseudo-LiDAR followed up with pseudo-LiDAR++. The main improvement of which is that the generated pseudo-LiDAR

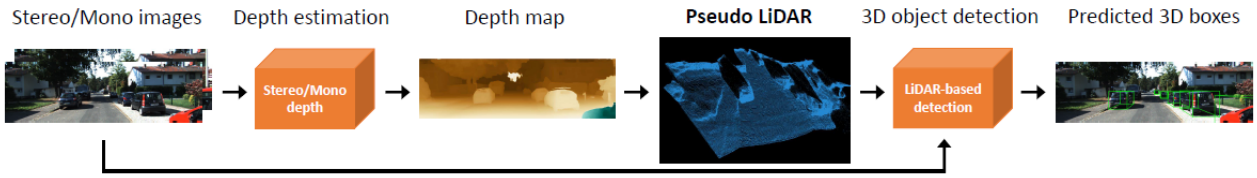

Fig. 22. The proposed pseudo-LiDAR pipeline for image-based 3D object detection. Taken from [48]

point cloud can now be combined with sparse measurements from low-cost LiDARs, which as stand-alone equipment would not provide sufficient information, in order to de-bias the depth estimation. According to the authors, performance of a 64-beam LiDAR unit of a total cost of $75,000 is now reached by pseudo-LiDAR++ with total equipment costing less than $1,000.

V. Conclusion

This paper presented a literature review of why state-of-the-art methods have outperformed traditional methods in the critical challenges of AV perception and making a comparison and discussion on both techniques. In the beginning, algorithms of both conventional and deep learning approaches were presented. The primary distinction between the two methods was found in the workflow of feature engineering. Traditional techniques use a manual feature extraction process followed by a shallow machine learning classifier. Whereas in state-of-the-art methods, deep learning was applied which incorporates an end-to-end architecture.

In the comparison of these conventional and novel methods for lane line detection, it can be seen that while many traditional CV approaches have been applied to the problem, with many achieving satisfactory results, these methods may not cope well with challenging illumination or weather conditions. Deep learning approaches can perform well under challenging conditions while removing the requirement for hand-crafted features. However, they require large annotated datasets for training, which are challenging to obtain.

For object detection and classification, automatic and real-time implementations are absolute necessities for an AV. Even though both traditional and deep learning methods provided acceptable accuracy, deep learning methods can work faster. End-to-end deep learning architectures have outperformed the traditional approaches that use feature extraction. However, deep learning techniques seem to be more computationally complex and expensive than the traditional methods.

For depth estimation, LiDAR technology has thrived in the area of 3D object detection and depth estimation, appearing to be the only valid approach for this case. Still, deep learning and computer vision recent advancements have proposed alternatives to LiDAR technology that can significantly decrease the overall cost of the AV without sacrificing accuracy in-depth and 3D object detection.

Both conventional and deep learning approaches have been discussed, compared and evaluated for AV systems. However, the results that used to compare both techniques sometimes were researched from different papers and different datasets; thus, it is challenging to summarise which method is certainly better. Future research will implement these two technologies mentioned above in each research area using the same dataset. Another topic of study may focus on another field in CV, such as free space detection. Besides, the authors would like to research more on the advanced driver assistance systems in order to reach level 5 that is entirely autonomous systems where no human intervention is required at all.

ACKNOWLEDGMENT

The authors would like to thank Dr Hong Yue and Dr David Harle for their guidance and suggestions. This work was supported by the Department of Electronic and Electrical Engineering, under the University of Strathclyde.

REFERENCES

[1] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, and J. Walsh, "Deep learning vs. traditional computer vision," in *Advances in Computer Vision*, K. Arai and S. Kapoor, Eds. Springer International Publishing, 2020, pp. 128–144.

[2] E. Karami, M. Shehata, and A. Smith, "Image identification using sift algorithm: Performance analysis against different image deformations," *arXiv preprint arXiv:1710.02728*, 2015.

[3] J. Wu, "Introduction to convolutional neural networks," *National Key Lab for Novel Software Technology. Nanjing University. China*, vol. 5, p. 23, 2017.

[4] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[6] C. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. IEEE, 1998, pp. 555–562.

[7] 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)*, vol. 1. IEEE, 2005, pp. 886–893 vol. 1.

[8] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 430–443.

[9] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *Journal of Manufacturing Systems*, vol. 48, pp. 144 – 156, 2018, special Issue on Smart Manufacturing. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0278612518300037

[10] B. Alefs, G. Eschemann, H. Ramoser, and C. Beleznai, "Road sign detection from edge orientation histograms," in *2007 IEEE Intelligent Vehicles Symposium*, 2007, pp. 993–998.

[11] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157 vol.2.

[12] F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi, "Pedestrian detection using infrared images and histograms of oriented gradients," in *2006 IEEE Intelligent Vehicles Symposium*, 2006, pp. 206–212.

[13] A. Ǵéron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow : concepts, tools, and techniques to build intelligent systems*, 2nd ed. O'Reilly Media, Inc., 2019.

[14] S. Bougharriou, F. Hamdaoui, and A. Mtibaa, "Linear svm classifier based hog car detection," in *2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, 2017, pp. 241–245.

[15] S. Hayou, A. Doucet, and J. Rousseau, "On the selection of initialization and activation function for deep neural networks," 2019. [Online]. Available: https://openreview.net/forum?id=H1lJws05K7

[16] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Artificial Neural Networks – ICANN 2010*, K. Diamantaras, W. Duch, and L. S. Iliadis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 92–101.

[17] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *CoRR*, vol. abs/1708.02002, 2017. [Online]. Available: http://arxiv.org/abs/1708.02002

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[19] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.

[20] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, and D. Haehnel, "Junior: The stanford entry in the urban challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.

[21] K. El Madawi, H. Rashed, A. El Sallab, O. Nasr, H. Kamel, and S. Yogamani, "Rgb and lidar fusion based 3d semantic segmentation for autonomous driving," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 7–12.

[22] G. N. Chaple, R. D. Daruwala, and M. S. Gofane, "Comparisions of robert, prewitt, sobel operator based edge detection methods for real time uses on fpga," *2015 International Conference on Technologies for Sustainable Development (ICTSD)*, pp. 1–4, 2015.

[23] O. R. Vincent and O. Folorunso, "A descriptive algorithm for sobel image edge detection," in *in proceedings of Informing Science & IT Education Conference (InSITE*, 2009.

[24] A. Borkar, M. Hayes, M. T. Smith, and S. Pankanti, "A layered approach to robust lane detection at night," in *2009 IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems*. IEEE, 2009, pp. 51–57.

[25] C. Yuan, H. Chen, J. Liu, D. Zhu, and Y. Xu, "Robust lane detection for complicated road environment based on normal map," *IEEE Access*, vol. 6, pp. 49 679–49 689, 2018.

[26] T.-Y. Sun, S.-J. Tsai, and V. Chan, "Hsi color model based lane-marking detection," in *2006 IEEE Intelligent Transportation Systems Conference*. IEEE, 2006, pp. 1168–1172.

[27] H. Yoo, U. Yang, and K. Sohn, "Gradient-enhancing conversion for illumination-robust lane detection," *Intelligent transportation systems, IEEE transactions on.*, vol. 14, no. 3, pp. 1083–1094, 2013.

[28] G. D. Finlayson, S. D. Hordley, and M. S. Drew, "Removing shadows from images," in *European conference on computer vision*. Springer, 2002, pp. 823–836.

[29] J. Son, H. Yoo, S. Kim, and K. Sohn, "Real-time illumination invariant lane detection for lane departure warning system," *Expert Systems with Applications*, vol. 42, no. 4, pp. 1816 – 1824, 2015.

[30] G. Zhang, N. Zheng, C. Cui, Y. Yan, and Z. Yuan, "An efficient road detection method in noisy urban environment," in *2009 IEEE Intelligent Vehicles Symposium*. IEEE, 2009, pp. 556–561.

[31] M. H. Sharif] and C. Djeraba, "An entropy approach for abnormal activities detection in video streams," *Pattern Recognition*, vol. 45, no. 7, pp. 2543 – 2561, 2012.

[32] D. Neven, B. D. Brabandere, S. Georgoulis, M. Proesmans, and L. V. Gool, "Towards end-to-end lane detection: an instance segmentation approach," *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 286–291, 2018.

[33] M. Nieto, J. Arróspide, L. Salgado, and N. Santos, "Video-based driver assistance systems," 01 2008.

[34] V. Voisin, M. Avila, B. Emile, S. Begot, and J.-C. Bardet, "Road markings detection and tracking using hough transform and kalman filter," in *Advanced Concepts for Intelligent Vision Systems*, J. Blanc-Talon, W. Philips, D. Popescu, and P. Scheunders, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 76–83.

[35] S. Lee, J. Kim, J. S. Yoon, S. Shin, O. Bailo, N. Kim, T. Lee, H. S. Hong, S. Han, and I. S. Kweon, "Vpgnet: Vanishing point guided network for lane and road marking detection and recognition," *CoRR*, vol. abs/1710.06288, 2017. [Online]. Available: http://arxiv.org/abs/1710.06288

[36] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[37] "Tusimple lane detection challenge dataset," https://github.com/TuSimple/tusimple-benchmark/issues/3, accessed: 23/04/2020.

[38] M. Aryal, "Object detection, classification, and tracking for autonomous vehicle," Masters thesis, Grand Valley State University, 2018.

[39] F. Han, Y. Shan, R. Cekander, H. S. Sawhney, and R. Kumar, "A two-stage approach to people and vehicle detection with hog-based svm," in *Performance Metrics for Intelligent Systems 2006 Workshop*, 2006, pp. 133–140.

[40] Ö. Kaplan and E. Saykol, "Comparison of support vector machines and deep learning for vehicle detection," in *RTA-CSIT*, 2018.

[41] "Darpa urban challenge," https://www.darpa.mil/about-us/timeline/darpa-urban-challenge, accessed: 23/04/2020.
[42] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2147–2156.
[43] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.
[44] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière, and T. Chateau, "Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1827–1836.
[45] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals using stereo imagery for accurate object class detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1259–1272, 2018.
[46] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.
[47] P. Li, X. Chen, and S. Shen, "Stereo r-cnn based 3d object detection for autonomous driving," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7636–7644.
[48] Y. Wang, W. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8437–8445.