WoodScape: A multi-task, multi-camera fisheye dataset for autonomous driving

Senthil Yogamani, Ciarán Hughes, Jonathan Horgan, Ganesh Sistu, Padraig Varley, Derek O'Dea,
Michal Uřičář, Stefan Milz, Martin Simon, Karl Amende, Christian Witt, Hazem Rashed,
Sumanth Chennupati, Sanjaya Nayak, Saquib Mansoor, Xavier Perrotton, Patrick Pérez

https://github.com/valeoai/WoodScape firstname.lastname@valeo.com

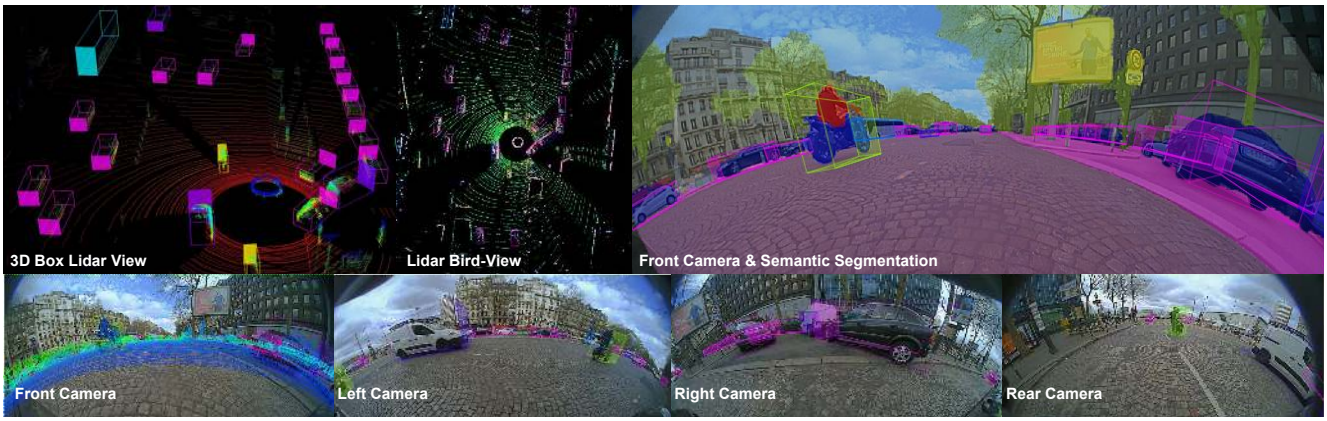

Figure 1: We introduce WoodScape, the first fisheye image dataset dedicated to autonomous driving. It contains four cameras covering 360° accompanied by a HD laser scanner, IMU and GNSS. Annotations are made available for nine tasks, notably 3D object detection, depth estimation (overlaid on front camera) and semantic segmentation as illustrated here.

Abstract

Fisheye cameras are commonly employed for obtaining a large field of view in surveillance, augmented reality and in particular automotive applications. In spite of their prevalence, there are few public datasets for detailed evaluation of computer vision algorithms on fisheye images. We release the first extensive fisheye automotive dataset, WoodScape, named after Robert Wood who invented the fisheye camera in 1906. WoodScape comprises of four surround view cameras and nine tasks including segmentation, depth estimation, 3D bounding box detection and soiling detection. Semantic annotation of 40 classes at the instance level is provided for over 10,000 images and annotation for other tasks are provided for over 100,000 images. With WoodScape, we would like to encourage the community to adapt computer vision models for fisheye camera instead of using naïve rectification.

1. Introduction

Fisheye lenses provide a large field of view (FOV) using a highly non-linear mapping instead of the standard perspective projection. However, it comes at the cost of strong radial distortion. Fisheye cameras are so-named because they relate to the 180° view of the world that a fish has observing the water surface from below, a phenomenon known as Snell's window. Robert Wood originally coined the term in 1906 [58], and constructed a basic fisheye camera by taking a pin-hole camera and filling it with water. It was later replaced with a hemispherical lens [3]. To pay homage to the original inventor and coiner of the term "fisheye", we have named our dataset WoodScape.

Large FOV cameras are necessary for various computer vision application domains, including video surveillance [28] and augmented reality [46], and have been of particular interest in autonomous driving [23]. In automotive, rear-view fisheye cameras are commonly deployed in existing vehicles for dashboard viewing and reverse parking. While commercial autonomous driving systems typically make use of narrow FOV forward facing cameras at present, full 360° perception is now investigated for handling more complex use cases. In spite of this growing interest, there is relatively little literature and datasets available. Some examples of the few datasets that have fisheye are: Visual SLAM ground truth for indoor scenes with omni-directional cameras in [7], SphereNet [9] containing 1200 labelled images of parked cars using 360° cameras (not strictly fisheye) and, in automotive, the Oxford Robotcar

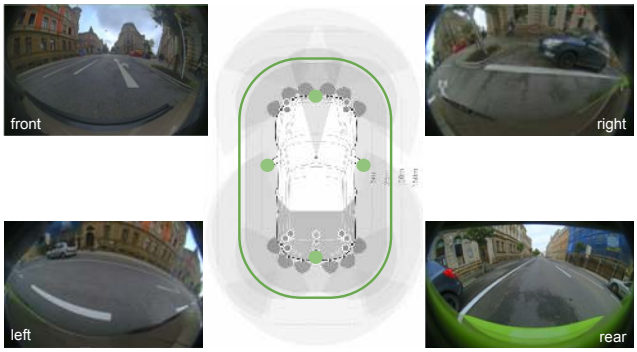

Figure 2: Sample images from the surround-view camera network showing wide field of view and 360° coverage.

dataset [37] containing a large scale relocalization dataset.

WoodScape is a comprehensive dataset for 360° sensing around a vehicle using the four fisheye cameras shown in Figure 2. It aims at complementing the range of already existing automotive datasets where only narrow FOV image data is present: among those, KITTI [17] was the first pioneering dataset with a variety of tasks, which drove a lot of research for autonomous driving; Cityscapes [10] provided the first comprehensive semantic segmentation dataset and Mapillary [39] provided a significantly larger dataset; Apolloscape [24] and BDD100k [59] are more recent datasets that push the annotation scale further. WoodScape is unique in that it provides fisheye image data, along with a comprehensive range of annotation types. A comparative summary of these different datasets is provided in Table 1. The main contributions of WoodScape are as follows:

1. First fisheye dataset comprising of over 10,000 images containing instance level semantic annotation.
2. Four-camera nine-task dataset designed to encourage unified multi-task and multi-camera models.
3. Introduction of a novel soiling detection task and release of first dataset of its kind.
4. Proposal of an efficient metric for the 3D box detection task which improves training time by 95x.

The paper is organized as follows. Section 2 provides an overview of fisheye camera model, undistortion methods and fisheye adaption of vision algorithms. Section 3 discusses the details of the dataset including goals, capture infrastructure and dataset design. Section 4 presents the list of supported tasks and baseline experiments. Finally, Section 5 summarizes and concludes the paper.

2. Overview of Fisheye Camera Projections

Fisheye cameras offer a distinct advantage for automotive applications. Given their extremely wide field of view, they can observe the full surrounding of a vehicle with a minimal number of sensors, with just four cameras typically being required for full 360° coverage (Figure 2). This advantage comes with some drawbacks in the significantly more complex projection geometry that fisheye cameras exhibit. That is, images from fisheye cameras display severe distortion.

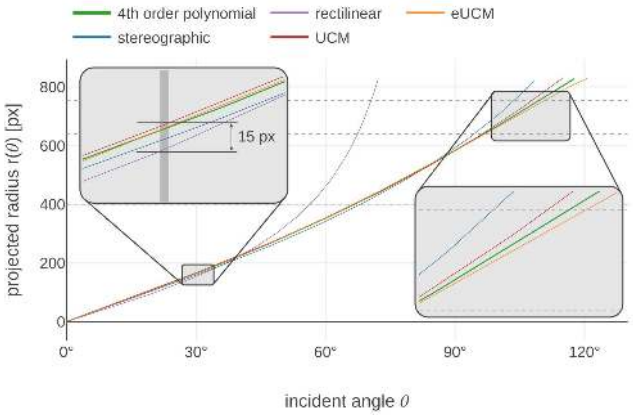

Figure 3: Comparison of fisheye models.

Typical camera datasets consist of narrow FOV camera data where a simple pinhole projection model is commonly employed. In case of fisheye camera images, it is imperative that the appropriate camera model is well understood either to handle distortion in the algorithm or to warp the image prior to processing. This section is intended to highlight to the reader that the fisheye camera model requires specific attention. We provide a brief overview and references for further details, and discuss the merits of operating on the raw fisheye versus undistortion of the image.

2.1. Fisheye Camera Models

Fisheye distortion is modelled by a radial mapping function $r(\theta)$, where $r(\theta)$ is the distance on the image from the centre of distortion, and is a function of the angle θ of the incident ray against the optical axis of the camera system. The centre of distortion is the intersection of the optical axis with the image plane, and is the origin of the radial mapping function $r(\theta)$. Stereographic projection [22] is the simplest model which uses a mapping from a sphere to a plane. More recent projection models are Unified Camera Model (UCM) [1, 7] and eUCM (Enhanced UCM) [27]. More detailed analysis of accuracy of various projection models is discussed in [25]. These models are not a perfect fit for fisheye cameras as they encode a specific geometry (e.g. spherical projection), and errors arising in the model are compensated by using an added distortion correction component.

In WoodScape, we provide model parameters for a more generic fisheye intrinsic calibration that is independent of any specific projection model, and does not require the added step of distortion correction. Our model is based on a fourth order polynomial mapping incident angle to image radius in pixels ($r(\theta) = a_1\theta + a_2\theta^2 + a_3\theta^3 + a_4\theta^4$). In

Figure 4: Undistorting the fisheye image: (a) Rectilinear correction; (b) Piecewise linear correction; (c) Cylindrical correction. Left: raw image; Right: undistorted image.

our experience, higher orders provide no additional accuracy. Each video sequence in the dataset is provided with parameters for the fourth order polynomial model of fisheye intrinsics.

As a comparison, to give the reader an understanding of how different models behave, Figure 3 shows the mapping function $r(\theta)$ for five different projection models, which are Polynomial, Rectilinear, Stereographic, UCM and eUCM. The parameters of the fourth order polynomial are taken from a calibration of our fisheye lens. We optimized the parameters for the other models to match this model in a range of 0° to 120° (i.e. up to FOV of 240°). The plot indicates that the difference to the original fourth order polynomial is about four pixels for UCM and one pixel for eUCM for low incident angles. For larger incident angles, these models are less precise.

2.2. Image Undistortion vs. Model Adaptation

Standard computer vision models do not generalize easily to fisheye cameras because of large non-linear distortion. For example, translation invariance is lost for a standard convolutional neural net (CNN). The naïve way to develop algorithms for fisheye cameras is to perform rectilinear correction so that standard models can be applied. The simplest undistortion is to re-warp pixels to a rectilinear image as shown in Figure 4 (a). But there are two major issues. Firstly, the FOV is greater than 180°, hence there are rays incident from behind the camera and it is not possible to establish a complete mapping to a rectilinear viewport. This leads to a loss of FOV, this is seen via the missing yellow pillars in the corrected image. Secondly, there is an issue of resampling distortion, which is more pronounced near the periphery of the image where a smaller region gets mapped to a larger region.

The missing FOV can be resolved by multiple linear viewports as shown in Figure 4 (b). However there are issues in the transition region from one plane to another. This can be viewed as a piecewise linear approximation of the fisheye lens manifold. Figure 4 (c) demonstrates a quasi-linear correction using a cylindrical viewport, where it is linear in vertical direction and straight vertical objects like pedestrians are preserved. However, there is a quadratic distortion along the horizontal axis. In many scenarios, it provides a reasonable trade-off but it still has limitations. In case of learning algorithms, a parametric transform can be optimized for optimal performance of the target application accuracy.

Because of fundamental limitations of undistortion, an alternate approach of adapting the algorithm incorporating fisheye model discussed in previous section could be an optimal solution. In case of classical geometric algorithms, an analytical version of non-linear projection can be incorporated. For example, Kukelova et al. [32] extend homography estimation by incorporating radial distortion model. In case of deep learning algorithms, a possible solution could be to train the CNN model to learn the distortion. However, the translation invariance assumption of CNN fundamentally breaks down due to spatially variant distortion and thus it is not efficient to let the network learn it implicitly. This had led to several adaptations of CNN to handle spherical images such as [52] and [9]. However, spherical models do not provide an accurate fit for fisheye lenses and it is an open problem.

3. Overview of WoodScape Dataset

3.1. High-Level Goals

Fisheye: One of the main goals of this dataset is to encourage the research community to develop vision algorithms natively on fisheye images without undistortion. There are very few public fisheye datasets and none of them provide semantic segmentation annotation. Fisheye is particularly beneficial to automotive low speed manoeuvring scenarios such as parking [21] where accurate full coverage near field sensing can be achieved with just four cameras.

Multi-camera: Surround view systems have at least four cameras rigidly connected to the body of the car. Pless [42] did pioneering work in deriving a framework for modeling a network of cameras as one, this approach is useful for geometric vision algorithms like visual odometry. However, for semantic segmentation algorithms, there is no literature on joint modeling of rigidly connected cameras.

Multi-task: Autonomous driving has various vision tasks and most of the work has been focused on solving individual tasks independently. However, there is a recent trend [30, 53, 51, 8] to solve tasks using a single multi-task model to enable efficient reuse of encoder features and also

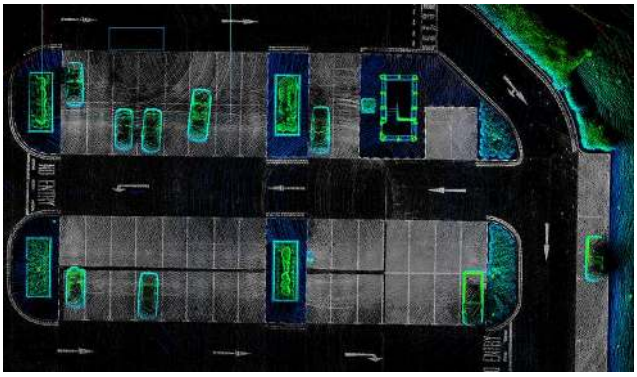

Figure 5: SLAM point cloud top-view of a parking lot. Height of the objects is color coded (green for high value, blue for medium value and grayscape for low value).

provide regularization while learning multiple tasks. However, in these cases, only the encoder is shared and there is no synergy among decoders. Existing datasets are primarily designed to facilitate task-specific learning and they don't provide simultaneous annotation for all the tasks. We have designed our dataset so that simultaneous annotation is provided for various tasks with some exceptions due to practical limitations of optimal dataset design for each task.

3.2. Dataset Acquisition

Our diverse dataset originates from three distinct geographical locations: USA, Europe, and China. While the majority of data was obtained from saloon vehicles there is a significant subset from a sports utility vehicle ensuring a strong mix in sensor mechanical configurations. Driving scenarios are divided across the highway, urban driving and parking use cases. Intrinsic and extrinsic calibrations are provided for all sensors as well as timestamp files to allow synchronization of the data. Relevant vehicle's mechanical data (e.g. wheel circumference, wheel base) are included. High-quality data is ensured via quality checks at all stages of the data collection process. Annotation data undergoes a rigorous quality assurance by highly skilled reviewers. The sensors recorded for this dataset are listed below:

- 4x 1MPx RGB fisheye cameras (190° horizontal FOV)
- 1x LiDAR rotating at 20Hz (Velodyne HDL-64E)
- 1x GNSS/IMU (NovAtel Propak6 & SPAN-IGM-A1)
- 1x GNSS Positioning with SPS (Garmin 18x)
- Odometry signals from the vehicle bus.

Our WoodScape dataset provides labels for several autonomous driving tasks including semantic segmentation, monocular depth estimation, object detection (2D & 3D bounding boxes), visual odometry, visual SLAM, motion segmentation, soiling detection and end-to-end driving (driving controls). In Table 1, we compare several properties of popular datasets against WoodScape. In addition to providing fisheye data, we provide data for many more tasks

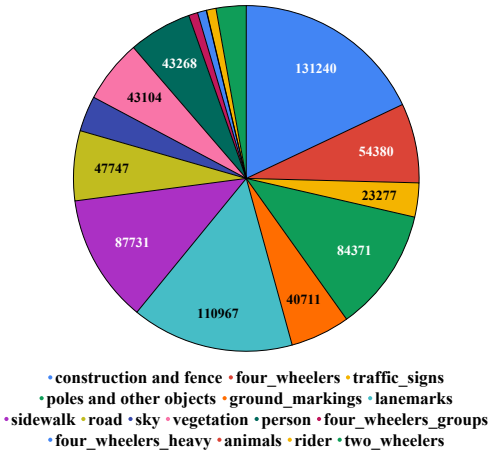

Figure 6: Distribution of instances of semantic segmentation classes in WoodScape. Minimum size is 300 pixels.

than is typical (nine in total), providing completely novel tasks such as soiled lens detection. Images are provided at 1MPx 24-bit resolution and videos are uncompressed at 30fps ranging in duration from 30s to 120s. The dataset also provides a set of synthetic data using accurate models of the real cameras, enabling investigations of additional tasks. The camera has a HDR sensor with a rolling shutter and a dynamic range of 120 dB. It has features including black level correction, auto-exposure control, auto-gain control, lens shading (optical vignetting) compensation, gamma correction and automatic white balance for color correction.

The laser scanner point cloud provided in our data set is accurately preprocessed using a commercial SLAM algorithm to provide a denser point cloud ground truth for tasks such as depth estimation and visual SLAM, as shown in Figure 5. In terms of recognition tasks, we provide labels for forty classes, the distribution of the main classes is shown in Figure 6. Note, that for the purposes of display in this paper, we have merged some of the classes in Figure 6 (e.g. 'two_wheelers' is a merge of 'bicycle' and 'motorcycle').

3.3. Dataset Design

The design of a dataset for machine learning is a very complex task. Unfortunately, due to the overwhelming success of deep learning, recently it does not get as much attention as it still deserves in our opinion. However, at the same time, it was shown that careful inspection of the training sets for outliers improves the robustness of deep neural networks [36], especially with regards to the adversarial examples. Therefore, we believe that whenever a new dataset is released, there should be a significant effort spent not only on the data acquisition but also on the careful consistency check and on the database splitting for the needs of training, model selection and testing.

Sampling strategy: Let us define some notation and nam-

Table 1: Summary of various autonomous driving datasets containing semantic annotation

Task/Info	Quantity	KITTI [17]	Cityscapes [10]	Mapillary [39]	nuScenes [6]	ApolloScape [24]	BDD100k [59]	WoodScape Ours
Capture Information	Year	2012/14/15	2016	2017	2018	2018	2018	2018/19
	State/cities	1/1	2/50	50+/100+	2/2	1/4	1/4	5+/10+
	Other sensors	1 LiDAR GPS	-	-	1 LiDAR GPS, IMU 5 RADAR	2 LiDAR GNSS IMU	1 GPS IMU	1 LiDAR GNSS IMU
Camera Information	Cameras	4	2	-	6	6	1	4
	Tasks	6	1	1	1	4	2	9
Segmentation	Classes	8	30	66	-	25	40	40
	Frames	400	5k	25k	-	140k	5.7k	10k
2D Bounding Box[1]	Classes	3	-	-	-	-	10	7
	Frames	15k	-	-	-	-	5.7k	10k
3D Bounding Box	Classes	3	-	-	25	1	-	3
	Frames	15k	-	-	40k	5k+	-	10k
Depth Estimation	Frames	93k	-	-	-	-	-	400k
Motion Segmentation	Frames	1.6k	-	-	-	-	-	10k
Soiling Detection	Frames	-	-	-	-	-	-	5k
Visual SLAM/Odometry	Videos	33	-	-	-	-	-	50
End-to-end Driving	Videos	-	-	-	-	-	-	500
Synthetic Data	Frames	-	-	-	-	-	-	10k

[1]2D box annotation can be obtained for other datasets from instance segmentation.

ing conventions, which we will refer to first (we follow the definitions provided in [4]). A *population* is a set of all existing feature vectors. A subset of the population collected during some process is called a *sample set* $\mathcal{S}$. A *representative set* $\mathcal{S}^*$ is significantly smaller than $\mathcal{S}$, while capturing most of the information from $\mathcal{S}$ (compared to any different subset of the same size), and has low redundancy among the representatives it contains.

In an ideal world, we would like our training set to be equal to $\mathcal{S}^*$. This is extremely difficult to achieve in practice. One approach to approximate this is the concept of the *minimal consistent subset of a training set*, where, given a training set $\mathcal{T}$, we are interested in a subset $\mathcal{T}^*$, being the smallest set such that $\mathrm{Acc}(\mathcal{T}^*) = \mathrm{Acc}(\mathcal{T})$, where $\mathrm{Acc}(\cdot)$ denotes the selected accuracy measure (e.g. the Jaccard index). Note, that computation of accuracy implies having the ground truth labels. The purpose is to reduce the size of the training set by removing non-informative samples, which do not contribute to improving the learned model, and therefore put some ease on the annotation efforts.

There are several ways of obtaining $\mathcal{T}^*$. One frequently used approach is instance selection [40, 35, 26]. There are two main groups of instance selection: wrappers and filters. The wrapper based methods use a selection criterion based on the constructed classifier's accuracy. Filter based methods, on the other hand, use a selection criterion which is based on an unrelated selection function. The concept of a minimal consistent subset is crucial for our setup, where we record image data from video cameras. Collecting frames at a frame rate of 30fps, particularly at low speeds, ultimately leads to significant image overlap, therefore, having an effective sampling strategy to distill the dataset is critical. We used a combination of a wrapper method using selection criterion based on the classifier's accuracy [40] and a simple filter based on the image similarity measurement.

Data splitting and class balancing: The dataset is split into three chunks in ratio of $6:1:3$, namely training, validation, and testing. For classical algorithms, all the data can be used for testing. As the names suggest, the training part will serve for training purposes only, the validation part can be either joined with the training set (e.g. when the sought model does not require hyper-parameter selection) or be used for model selection, and finally, the testing set is used for model evaluation purposes only. The dataset supports correct hypothesis evaluation [55], therefore multiple splits are provided (5 in total). Depending on the particular task (see Section 4, for the full list), the class imbalance may be an issue [19], therefore, task-specific splits are also provided. Full control of the splitting mechanism is provided allowing for each class to be represented equally within each split (i.e. stratified sampling).

GDPR challenges: The recent General Data Protection Regulation (GDPR) regulation in Europe has given rise to challenges in making our data publicly available. More than one third of our dataset is recorded in Europe and is therefore GDPR sensitive due to visible faces of pedestrians and license plates. There are three primary ways to handle privacy namely (1) Manual blurring, (2) GAN based retargeting and (3) Stringent data-handling license agreement. Blurring is the commonly used approach wherein privacy sensitive regions in the image are manually blurred. There

is also the possibility of using GAN based re-targeting wherein faces are exchanged by automatically generated ones [31]. In the recent EuroCity persons dataset [5], the authors argued that any anonymization measure will introduce a bias. Thus they released their dataset with original data and a license agreement which enforces the user to strictly adhere to GDPR. We will follow a similar approach.

4. Tasks, Metrics and Baseline experiments

Due to limited space, we briefly describe the metrics and baseline experiments for each task and they are summarized in Table 2. Test dataset for each task consists of 30% of the respective number of annotated samples listed in Table 1. Code is available on WoodScape GitHub and sample video results are shared in supplementary material.

4.1. Semantic Segmentation

Semantic segmentation networks for autonomous driving [47] have been successfully trained directly on fisheye images in [12, 45]. Due to absence of fisheye datasets, they make use of artificially warped images of Cityscapes for training and testing was performed on fisheye images. However, the artificial images cannot increase the originally captured FOV. Our semantic segmentation dataset provides pixel-wise labels for 40 object categories, comparatively Cityscapes dataset [10] provides 30 for example. Figure 6 illustrates the distribution of main classes. We use ENet [41] to generate our baseline results. We fine-tune their model for our dataset by training with categorical cross entropy loss and Adam [29] optimizer. We chose Intersection over Union (IoU) metric [16] to report the baseline results shown in Table 2. We acheive a mean IoU of 51.4 on this test set. Figure 7 shows sample results of segmentation on fisheye images from our test set. The four camera images are treated the same, however it would be interesting to explore customization of the model for each camera. The dataset also provides instance segmentation labels to explore panoptic segmentation models [34].

4.2. 2D Bounding Box Detection

Our 2D object detection dataset is obtained by extracting bounding boxes from instance segmentation labels for 7 different object categories including pedestrians, vehicles, cyclist and motorcyclist. We use Faster R-CNN [43] with ResNet101 [20] as encoder. We initialize the network with ImageNet [11] pre-trained weights. We fine-tune our detection network by training on both KITTI [18] and our object detection datasets. Performance of 2D object detection is reported in terms of mean average precision (mAP) when IoU$\geq$ 0.5 between predicted and ground truth bounding boxes. We achieve a mAP score of 31 which is significantly less than the accuracy achieved in other datasets. This was expected as bounding box detection is a difficult task on fisheye (the orientation of objects in the periphery of images being very different from central region). To quantify this better, we tested a pre-trained network for person class, and a poor mAP score of 12 was achieved compared to our dataset trained value of 45. Sample results of the fisheye trained model are illustrated in Figure 7. We observe that it is necessary to incorporate the fisheye geometry explicitly, which is an open research problem.

Table 2: Summary of results of baseline experiments.

Task	Model	Metric	Value
Segmentation	ENet [41]	IoU	51.4
2D Bounding Box	Faster R-CNN [43]	mAP (IoU>0.5)	31
Soiling Detection	ResNet10 [20]	Category (%)	84.5
Depth Estimation	Eigen [14]	RMSE	7.7
Motion Segmentation	MODNet [49]	IoU	45
Visual Odometry	ResNet50 [20]	Translation (<5mm)	51
		Rotation (<0.1°)	71
Visual SLAM	LSD SLAM [15]	Relocalization (%)	61
3D Bounding Box Detection - Complex YOLO [50]			
Metric for Training	**AP (%)**	**AOS (%)**	**Runtime (ms)**
3D-IoU	64.38	85.60	95
S_{srt}	62.46	88.43	1

4.3. Camera Soiling Detection

The task of soiling detection was to our best knowledge first defined in [56]. Unlike the front camera which is behind the windshield, the surround view cameras are usually directly exposed to the adverse environmental conditions, and thus prone to becoming soiled or water drops forming on the lens. As the functionality of visual perception degrades significantly, detection of soiled cameras is necessary for achieving higher levels of automated driving. As it is a novel task, we discuss it in more detail below.

We treat the camera soiling detection task as a mixed multilabel-categorical classification problem, i.e. we are interested in a classifier, which jointly classifies a single image with a binary indicator array, where each 0 or 1 corresponds to missing or present class, respectively and simultaneously assigns a categorical label. The classes to detect are $\{\text{opaque}, \text{transparent}\}$. Typically, opaque soiling arises from mud and dust (Figure 8 right image), and transparent soiling arises from water and ice (Figure 8 left image). However, in practice it is common to see water producing "opaque" regions in the camera image.

Annotation for 5k images is performed by drawing polygons to separate soiled from unsoiled regions, so that it can be modeled as a segmentation task if necessary. We evaluate the soiling classifier's performance via an example-based accuracy measure for each task separately, i.e. the average Jaccard index of the testing set: $\frac{1}{n}\sum_{i=1}^{n}\frac{|\mathbf{Y}_i \cap \mathbf{Z}_i|}{|\mathbf{Y}_i \cup \mathbf{Z}_i|}$, where $\mathbf{Y}_i \in \mathcal{Y} = \{0,1\}^k$ denotes the label for the i-th testing sample, $\mathbf{Z}_i$ denotes the classifier's prediction and n denotes the cardinality of the testing set and k the length of the label vector. We use a small baseline network (ResNet10 encoder + 3-layer decoder) and achieved a precision of 84.5% for the multilabel classification.

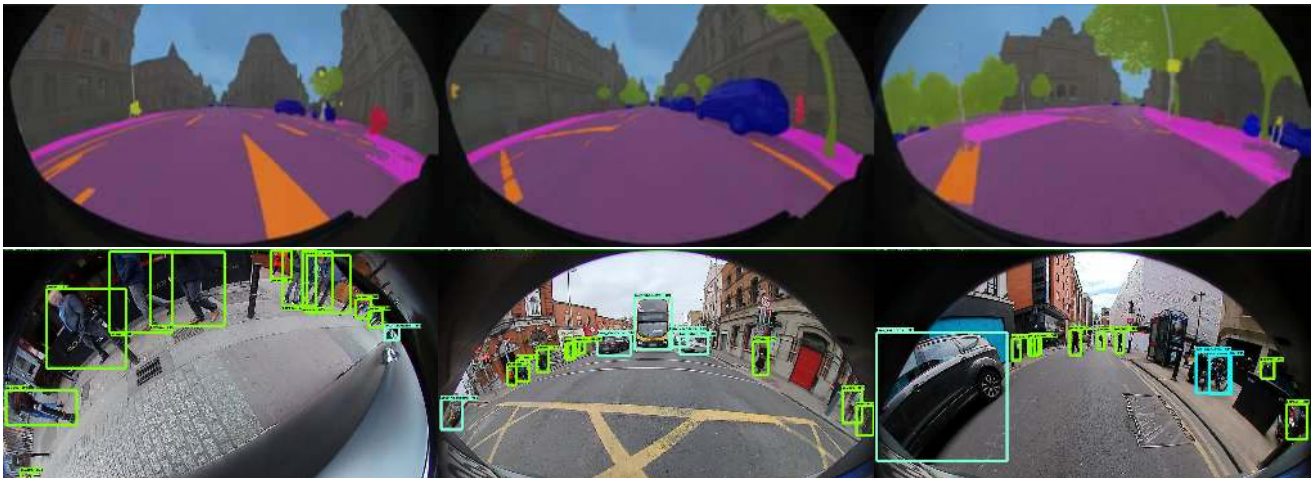

Figure 7: Qualitative results of Segmentation using ENet [41] (top) and Object detection using Faster RCNN [43] (bottom)

4.4. 3D Bounding Box Detection

3D box annotation is provided for 10k frames with 3 classes namely 'pedestrian', 'vehicle' and 'cyclist'. In general, 3D IoU [18] is used to evaluate 3D bounding box predictions, but there are drawbacks, especially for rotated objects. Two boxes can reach a good 3D IoU score, while overlapping in total with an opposite heading. Additionally, an exact calculation in 3D space is a time consuming task. To avoid those problems, we introduce a new evaluation metric called Scaling-Rotation-Translation score (*SRTs*). *SRT* is based on the idea that two non-overlapping 3D boxes can easily be transformed with respect to each other by using independent rigid transformations: translation S_t, rotation S_r and scaling S_s. Hence, S_{srt} is composed by:

$$S_s = 1 - \min\left(\frac{|1-s_x| + |1-s_y| + |1-s_z|}{w_s}, 1\right)$$

$$S_r = \max\left(0, 1 - \frac{\theta}{w_r \pi}\right) \quad S_t = \max\left(0, \frac{r_1 + r_2 - t}{r_1 + r_2}\right)$$

$$r_{1/2} = \frac{d_{1/2} \cdot w_t}{2} \qquad w_t, w_r, w_s \in (0, 1]$$

where $s_{x,y,z}$ denotes size ratios in x, y, z directions, θ determines the difference of the yaw angles and t defines the Euclidean distance between the two box centers. S_t is calculated with respect to the size of the two objects based on the length of the diagonals $d_{1/2}$ of both objects that are used to calculate two radii $r_{1/2}$. Based on the penalty term p_t we define the full metric by:

$$S_{srt} = p_t \cdot (\alpha\, S_s + \beta\, S_t + \gamma\, S_r) \quad \alpha + \beta + \gamma = 1$$

$$p_t = \begin{cases} 0, & \text{if } r_1 + r_2 < t \\ 1, & \text{otherwise} \end{cases}$$

w_s, w_t and w_r can be used to prioritize individual properties (e.g. $w_s \to$ size, $w_t \to$ angle). For our baseline experiments we used $w_s = 0.3$, $w_t = 1$, $w_r = 0.5$, $\gamma = 0.4$ and $\alpha = \beta = 0.3$ to add more weight to the angle, because our experiments have shown that translation or scaling is easier to learn. For baseline, we trained Complex-YOLO [50] for a single class ('car'). We repeated training two times, first optimized on 3D-IoU [18] and second optimized on S_{srt} using a fixed 50:50 split for training and validation. For comparison, we present 3D-IoU, orientation and runtime following [18] on *moderate* difficulty, see Table 2. Runtime is the average runtime of all box comparisons for each input during training. Even though this comparison uses 3D-IoU, we achieve similar performance for average precision (3D-IoU), with better angle orientation similarity (AOS) and much faster computation time.

4.5. Monocular Depth Estimation

Monocular Depth estimation is an important task for detecting generic obstacles. We provide more than 100k images of all four cameras (totaling 400k) using ground truth provided by LiDAR. Figure 1 shows a colored example where blue to red indicates the distance for the front camera. As the depth obtained is sparse, we also provide denser point cloud based on SLAM'd static scenes as shown in Figure 5. The ground truth 3D points are projected onto the camera images using our proposed model discussed in Section 2.1. We also apply occlusion correction to handle difference in perspective of LiDAR and camera similar to the method proposed in [33]. We run the semi-supervised approach from [33] using the model proposed by Eigen [14] as baseline on our much larger dataset and obtained an RMSE (Root Mean Square Error) value of 7.7.

4.6. Motion Segmentation

In automotive, motion is a strong cue due to ego-motion of the cameras on the moving vehicle and dynamic objects around the vehicle are the critical interacting agents. Additionally, it is helpful to detect generic objects based on

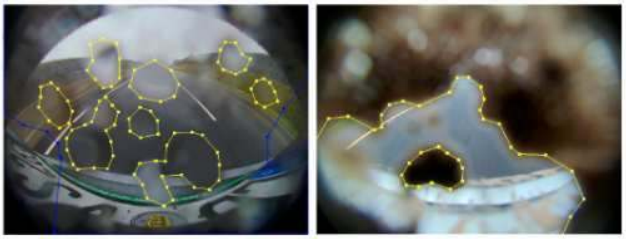

Figure 8: Soiling annotation

Figure 9: Visual SLAM baseline results (left) based on raw fisheye images (right)

Figure 10: Synthetic images modelling fisheye optics

motion cues rather than appearance cues as there will always be rare objects like kangaroos or construction trucks. This has been explored in [49, 57, 48] for narrow angle cameras. In our dataset, we provide motion masks annotation for moving classes such as vehicles, pedestrians and cyclists for over 10k images. We also provide previous and next images for exploring multi-stream models like MODNet [49]. Motion segmentation is treated as a binary segmentation problem and IoU is used as the metric. Using MODNet as baseline network, we achieve an IoU of 45.

4.7. Visual Odometry/SLAM

Visual Odometry (VO) is necessary for creating a map from the objects detected [38]. We make use of our GNSS and IMU to provide annotation in centimetre level accuracy. The ground truth contains all the six degrees of freedom upto scale and the metric used is percentage of frames within a tolerance level of translation and rotation error. Robustness could be added to the visual odometry by performing a joint estimation from all four cameras. We provide 50 video sequences comprising of over 100k frames with ground truth. The video sequences can also be used for Visual SLAM where we focus on relocalization of a mapped trajectory and the metric is same as VO. We use a fisheye adapted LSD-SLAM [15] as our baseline model as illustrated in Figure 9 and accuracies are provided in Table 2.

4.8. Synthetic Data Domain Transfer

Synthetic data is crucial for autonomous driving for many reasons. Firstly, it provides a mechanism to do rigorous corner case testing for diverse scenarios. Secondly, there are legal restrictions like recording videos of a child. Finally, synthetic data is the only way to obtain dense depth and optical flow annotation. There are several popular synthetic datasets like SYNTHIA [44] and CARLA [13]. We will provide a synthetic version of our fisheye surround view dataset, as shown in Figure 10. The main goal is to explore domain transfer from synthetic to real domain for semantic segmentation and depth estimation tasks.

4.9. End-to-End Steering/Braking

Bojarski et al. demonstrated end-to-end learning [2] for steering and recently it was applied to fisheye cameras [54]. Although this approach is currently not mature for deployment, it can be either used as a parallel model for redundancy or as an auxiliary task to improve accuracy of other tasks. In the traditional approach, perception is independently designed and it is probably a more complex intermediate problem to solve than what is needed for a small action space driving task. Thus we have added end-to-end steering and braking tasks to encourage modular end-to-end architectures and to explore optimized perception for the control task. The latter is analogous to hand-eye co-ordination of human drivers where perception is optimized for driving.

5. Conclusions

In this paper, we provide an extensive multi-camera fisheye dataset for autonomous driving with annotation for nine tasks. We hope that the release of the dataset encourages development of native fisheye models instead of undistorting fisheye images and applying standard models. In case of deep learning algorithms, it can help understand whether spatial distortion can be learned or it has to be explicitly modeled. In future work, we plan to explore and compare various methods of undistortion and explicit incorporation of fisheye geometry in CNN models. We also plan to design a unified multi-task model for all the listed tasks.

Acknowledgement

We would like to thank our colleagues Nivedita Tripathi, Mihai Ilie, Philippe Lafon, Marie Yahiaoui, Sugirtha Thayalan, Jose Luis Fernandez and Pantelis Ermilios for supporting the creation of the dataset, and to thank our partners MightyAI for providing high-quality semantic segmentation annotation services and Next Limit for providing synthetic data using ANYVERSE platform.

References

[1] Joao P Barreto. Unifying image plane liftings for central catadioptric and dioptric cameras. *Imaging Beyond the Pinhole Camera*, pages 21–38, 2006. 2

[2] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016. 8

[3] WN Bond. A wide angle lens for cloud recording. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 44(263):999–1001, 1922. 1

[4] Tomas Borovicka, Marcel Jirina Jr., Pavel Kordik, and Marcel Jirina. Selecting representative data sets. In Adem Karahoca, editor, *Advances in Data Mining Knowledge Discovery and Applications*, chapter 2. IntechOpen, Rijeka, 2012. 5

[5] Markus Braun, Sebastian Krebs, Fabian Flohr, and Dariu Gavrila. EuroCity persons: A novel benchmark for person detection in traffic scenes. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 6

[6] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019. 5

[7] David Caruso, Jakob Engel, and Daniel Cremers. Large-scale direct SLAM for omnidirectional cameras. In *Proceedinsg of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 141–148. IEEE, 2015. 1, 2

[8] Sumanth Chennupati, Ganesh Sistu, Senthil Yogamani, and Samir Rawashdeh. Auxnet: Auxiliary tasks enhanced semantic segmentation for automated driving. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, pages 645–652, 2019. 3

[9] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. SphereNet: Learning spherical representations for detection and classification in omnidirectional images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–533, 2018. 1, 3

[10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016. 2, 5, 6

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. Ieee, 2009. 6

[12] Liuyuan Deng, Ming Yang, Yeqiang Qian, Chunxiang Wang, and Bing Wang. CNN based semantic segmentation for urban traffic scenes using fisheye camera. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 231–236. IEEE, 2017. 6

[13] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. 8

[14] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015. 6, 7

[15] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *European conference on computer vision*, pages 834–849. Springer, 2014. 6, 8

[16] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 6

[17] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2, 5

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 6, 7

[19] Xinjian Guo, Yilong Yin, Cailing Dong, Gongping Yang, and Guang-Tong Zhou. On the class imbalance problem. In *Proceedings of the Fourth International Conference on Natural Computation (ICNC)*, pages 192–201, 2008. 5

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[21] Markus Heimberger, Jonathan Horgan, Ciarán Hughes, John McDonald, and Senthil Yogamani. Computer vision in automated parking systems: Design, implementation and challenges. *Image and Vision Computing*, 68:88–101, 2017. 3

[22] Thomas J Herbert. Area projections of fisheye photographic lenses. *Agricultural and Forest Meteorology*, 39(2-3):215–223, 1987. 2

[23] Jonathan Horgan, Ciarán Hughes, John McDonald, and Senthil Yogamani. Vision-based driver assistance systems: Survey, taxonomy and advances. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 2032–2039. IEEE, 2015. 1

[24] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The ApolloScape dataset for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 954–960, 2018. 2, 5

[25] Ciarán Hughes, Patrick Denny, Edward Jones, and Martin Glavin. Accuracy of fish-eye lens models. *Applied Optics*, 49(17):3338–3347, 2010. 2

[26] Norbert Jankowski and Marek Grochowski. Comparison of instances selection algorithms I. Algorithms survey. In *Proceedings of the 7th International Conference on Artificial*

Intelligence and Soft Computing ICAISC, pages 598–603, 2004. 5

[27] Bogdan Khomutenko, Gaetan Garcia, and Philippe Martinet. An enhanced unified camera model. *IEEE Robotics and Automation Letters*, 1(1):137–144, 2016. 2

[28] Hyungtae Kim, Jaehoon Jung, and Joonki Paik. Fisheye lens camera based surveillance system for wide field of view monitoring. *Optik*, 127(14):5636–5646, 2016. 1

[29] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. 6

[30] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6129–6138, 2017. 3

[31] Pavel Korshunov and Sébastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685*, 2018. 6

[32] Zuzana Kukelova, Jan Heller, Martin Bujnak, and Tomas Pajdla. Radial distortion homography. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 639–647, 2015. 3

[33] Varun Ravi Kumar, Stefan Milz, Christian Witt, Martin Simon, Karl Amende, Johannes Petzold, Senthil Yogamani, and Timo Pech. Near-field depth estimation using monocular fisheye camera: A semi-supervised learning approach using sparse LiDAR data. In *CVPR Workshop*, 2018. 7

[34] Qizhu Li, Anurag Arnab, and Philip HS Torr. Weakly- and semi-supervised panoptic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 102–118, 2018. 6

[35] Huan Liu and Hiroshi Motoda. On issues of instance selection. *Data Mining and Knowledge Discovery*, 6(2):115–130, 2002. 5

[36] Yongshuai Liu, Jiyu Chen, and Hao Chen. Less is more: Culling the training set to improve robustness of deep neural networks. In *In Proceedings of the 9th International Conference on Decision and Game Theory for Security (GameSec)*, pages 102–114, 2018. 4

[37] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017. 2

[38] Stefan Milz, Georg Arbeiter, Christian Witt, Bassam Abdallah, and Senthil Yogamani. Visual slam for automated driving: Exploring the applications of deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 247–257, 2018. 8

[39] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The Mapillary Vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4990–4999, 2017. 2, 5

[40] José Arturo Olvera-López, Jesús Ariel Carrasco-Ochoa, Jos'e Francisco Martínez Trinidad, and Josef Kittler. A review of instance selection methods. *Artificial Intelligence Review*, 34(2):133–143, 2010. 5

[41] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. ENet: A deep neural network architecture for real-time semantic segmentation, 2016. 6, 7

[42] Robert Pless. Using many cameras as one. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003. 3

[43] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015. 6, 7

[44] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, 2016. 8

[45] Alvaro Sáez, Luis M Bergasa, Eduardo Romeral, Elena López, Rafael Barea, and Rafael Sanz. CNN-based fisheye image real-time semantic segmentation. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1039–1044. IEEE, 2018. 6

[46] Dieter Schmalstieg and Tobias Hollerer. *Augmented reality: principles and practice*. Addison-Wesley Professional, 2016. 1

[47] Mennatullah Siam, Sara Elkerdawy, Martin Jagersand, and Senthil Yogamani. Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2017. 6

[48] Mennatullah Siam, Mostafa Gamal, Moemen Abdel-Razek, Senthil Yogamani, and Martin Jagersand. RTSeg: Real-time semantic segmentation comparative study. In *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018. 8

[49] Mennatullah Siam, Heba Mahgoub, Mohamed Zahran, Senthil Yogamani, Martin Jagersand, and Ahmad El-Sallab. MODNet: Motion and appearance based moving object detection network for autonomous driving. In *Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2859–2864, 2018. 6, 8

[50] Martin Simon, Stefan Milz, Karl Amende, and Horst-Michael Gross. Complex-YOLO: An Euler-region-proposal for real-time 3D object detection on point clouds. In *The European Conference on Computer Vision (ECCV) Workshops*, September 2018. 6, 7

[51] Ganesh Sistu, Isabelle Leang, Sumanth Chennupati, Stefan Milz, Senthil Yogamani, and Samir Rawashdeh. NeurAll: Towards a unified model for visual perception in automated driving. In *International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2019. 3

[52] Yu-Chuan Su and Kristen Grauman. Kernel transformer networks for compact spherical convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9442–9451, 2019. 3

[53] M. Teichmann, M. Weber, M. Zöllner, R. Cipolla, and R. Urtasun. MultiNet: Real-time joint semantic reasoning for autonomous driving. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, pages 1013–1020, June 2018. 3

[54] Marin Toromanoff, Emilie Wirbel, Frédéric Wilhelm, Camilo Vejarano, Xavier Perrotton, and Fabien Moutarde. End to end vehicle lateral control using a single fisheye camera. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3613–3619, 2018. 8

[55] Michal Uřičář, David Hurych, Pavel Křížek, and Senthil Yogamani. Challenges in designing datasets and validation for autonomous driving. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, pages 653–659. INSTICC, SciTePress, 2019. 5

[56] Michal Uřičář, Pavel Křížek, David Hurych, Ibrahim Sobh, Senthil Yogamani, and Patrick Denny. Yes, we GAN: applying adversarial techniques for autonomous driving. *Electronic Imaging*, 2019(15):48–1–48–17, 2019. 6

[57] Johan Vertens, Abhinav Valada, and Wolfram Burgard. Smsnet: Semantic motion segmentation using deep convolutional neural networks. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, Canada, 2017. 8

[58] Robert W Wood. Fish-eye views, and vision under water. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 12(68):159–162, 1906. 1

[59] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018. 2, 5