# FuseMODNet: Real-Time Camera and LiDAR based Moving Object Detection for robust low-light Autonomous Driving

Hazem Rashed[1], Mohamed Ramzy[2], Victor Vaquero[3], Ahmad El Sallab[1],
Ganesh Sistu[4] and Senthil Yogamani[4]
[1]Valeo R&D, Egypt [2] Cairo University [3]IRI BarcelonaTech, Spain [4]Valeo Vision Systems, Ireland
firstname.lastname@valeo.com, mohamed.ibrahim98@eng-st.cu.edu.eg, vvaquero@iri.upc.edu

## Abstract

*Moving object detection is a critical task for autonomous vehicles. As dynamic objects represent higher collision risk than static ones, our own ego-trajectories have to be planned attending to the future states of the moving elements of the scene. Motion can be perceived using temporal information such as optical flow. Conventional optical flow computation is based on camera sensors only, which makes it prone to failure in conditions with low illumination. On the other hand, LiDAR sensors are independent of illumination, as they measure the time-of-flight of their own emitted lasers. In this work we propose a robust and real-time CNN architecture for Moving Object Detection (MOD) under low-light conditions by capturing motion information from both camera and LiDAR sensors. We demonstrate the impact of our algorithm on KITTI dataset where we simulate a low-light environment creating a novel dataset "Dark-KITTI". We obtain a 10.1% relative improvement on Dark-KITTI, and a 4.25% improvement on standard KITTI relative to our baselines. The proposed algorithm runs at 18 fps on a standard desktop GPU using* $256 \times 1224$ *resolution images.*

## 1. Introduction

Autonomous Driving (AD) scenarios are considered very complex environments as they are highly dynamic containing multiple object classes that move at different speeds in diverse directions [12, 11]. For an autonomous vehicle, it is critical to fully understand the motion model of each of the surrounding elements as well as to be able to plan the ego-trajectories based on the future states of these objects, therefore avoiding collision risks. There are two types of motion in a typical autonomous driving scene, i.e. the one of the surrounding obstacles and the motion of the ego-vehicle. Due to the movement of the camera reference itself, it is challenging to successfully classify the surrounding objects as moving or static, because even static objects will be perceived as moving. Motion segmentation implies two tasks that are performed jointly. The first one focuses on object selection, in which objects of specific interesting classes are highlighted such as pedestrians or vehicles. The second one focuses on motion classification, in which a binary classifier predicts whether the observed object is dynamic or static.

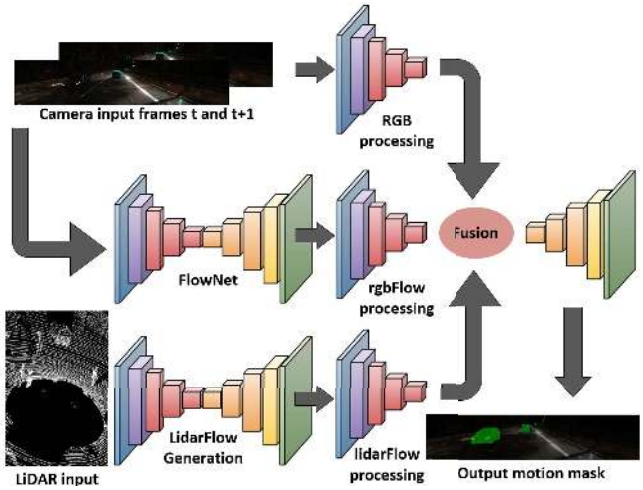


Figure 1: Proposed Network Architecture

Modern vehicles are equipped with various sensors to be able to fully perceive the surrounding environment, each one having its advantages and disadvantages. For instance, ultrasonic sensors provide good performance of depth measurement for close obstacles but they lack semantic information and perform poorly for far objects. Camera sensors instead, provide rich color information from which scene semantics can be extracted however, they lack of depth information and rely on scene illumination, being the performance of any camera-based perception tasks highly degraded in bad illumination conditions such as at night scenes. On the other hand, LiDAR sensors provide accurate depth and geometric information of the environment,

although they generate big and sparse point clouds that may suppose a computational bottleneck. Nevertheless, unlike camera sensors, LiDARs rely on the Time of Flight (ToF) concept and therefore they can perform much better under low illumination or light changing conditions.

Data fusion has been proven to provide improved performance in various tasks such as [27, 32, 14]. In this work, we focus on fusing Camera and LiDAR information for the purpose of moving objects detection. Our proposed architecture attempts to capture rich motion information from both camera and LiDAR sensors which is combined with scene semantics from the camera images. To summarize, the contributions of this work include:

- We extend the publicly available KittiMoSeg [32] dataset almost x10 times, expanding from 1300 frames only to a new amount of 12919 images. The dataset is available at https://sites.google.com/view/fusemodnet
- We create the new Dark-KITTI dataset to simulate low illumination autonomous driving environments.
- We propose a novel CNN architecture for MOD fusing both RGB and LiDAR information. Our implementation performs on real-time, and therefore is suitable for time-critical applications such as autonomous driving.
- We analyze different fusion methodologies for maximum performance as well as study motion representations for both RGB frames and LiDAR points clouds.

The rest of the paper is organized as follows: a review of the related work is presented in Section 2. Our methodology including the dataset preparation and the used network architectures is detailed in Section 3. Experimental setup and final results are illustrated in Section 4. Finally, Section 5 concludes the paper.

## 2. Related Work

**Motion Segmentation using Camera sensor:** Classical approaches have been proposed for moving objects detection based on geometrical understanding of the scene such as [25] which was used to estimate objects motion masks. Wehrwein et al. [38] introduced assumptions about the camera motion model to model the background motion in terms of homography. This approach cannot be used in autonomous driving application due to the errors arising from the limited assumptions such as camera translations. Classical methods provide poor performance compared to deep learning methods in addition to high complexity due to complicated pipelines used. For instance, Menze et al. [25] running time is 50 minutes per frame which makes it impossible for usage in a real-time application such as the autonomous driving. Deep learning algorithms are becoming successful beyond object detection [30] for applications like visual SLAM [26], depth estimation [17], soiling detection [33] but it is still relatively less explored for MOD task.

Jain et. al.[14] proposed a method to exploit optical flow for generic foreground segmentation. This work is designed for generic object segmentation and does not focus on classifications of objects as Moving or Static. Drayer et. al.[6] proposed a video segmentation algorithm that is based on R-CNN detection. The approach is not practical as well for autonomous driving application due to its complexity where it runs on image in 8 seconds. Siam et al. [32, 29] explored motion segmentation using deep network architectures, however these networks rely only on camera RGB images which is prone to failure in low illumination conditions. FisheyeMODNet [39] extends MODNet for fisheye camera images using WoodScape dataset [40].

**Motion Segmentation using LiDAR sensor:** Most of LiDAR-based methods that have been used for motion segmentation problem were based on clustering methods such as [4] which predicts the points motion by methods such as RANSAC, and then clustering takes place for object-level perception. Vaquero et al. [34] initially clustered vehicles points and then performed motion segmentation on the objects after matching the objects through sequential frames. Deep Learning has been utilized in various methods for object detection on point clouds. In [18] 3D convolution is used over the point cloud to obtain the vehicles bounding boxes. Other methods project the 3D points on 2D images to make use of 2D convolutions on the image 2D space [19]. None of these methods are able to segment moving objects from static ones. Recent work [5] learns movable and non-movable objects from two input lidar scans. This method uses implicit learning for motion information through two sequential lidar scans and does not utilize the color information from camera sensor, which motivates our work towards fusion of both camera and LiDAR sensors.

**Fusion:** Fusion has been explored through classical and deep learning methods, and it has proven to be very important for many tasks. The most common way for multimodal fusion using classical approaches is Kalman filter [15] and its variants. CNNs have been exploited as well for multimodal fusion where they generally provide improved performance over Kalman filters at the cost of complexity. Deep fusion has been explored for the task of semantic segmentation [27, 28, 9] using fusion between RGB images and optical flow and depth. Several methods have been visited to fuse camera and LiDAR sensors for various tasks such as [24] which implemented an algorithm for 3D semantic segmentation. Pedestrian detection has been improved significantly using fusion between RGB images and infrared maps [16, 20, 37]. Modern vehicles are usually equipped with various sensors to perceive the environment which we propose to leverage using a deep fusion network.

## 3. Methodology

In this section we discuss dataset preparation, and the proposed architecture for our experiments.

### 3.1. Dataset Preparation

Our proposed method fuses color images with motion signals obtained from different sensors to generate motion masks as output. In this section we describe the inputs preparation and outputs of our architecture.

**Annotations Generation:** In order to train our deep model for maximum generalization on the motion segmentation task, we need motion masks annotations from a large driving dataset. There is huge limitation in publicly available datasets regarding moving objects detection. Siam et al. [32] provides 1300 images only with weak annotation for MOD task. Valada et al. [36] provides 255 annotated frames only on KITTI dataset, and 3475 annotated frames on Cityscapes [3] dataset. Cityscapes does not have LiDAR point clouds, and therefore will not be helpful for our low-light purposes. Behley et al. [1] provides MOD annotations for 3D point clouds only, but not for dense pixels. We therefore build our own Motion Object Detection dataset. For that, we adopt the method in [32] to generate motion masks from KITTI in order to extend the KittiMoSeg dataset. Initially, we project the existing 3D bounding boxes from 3D LiDAR frame to 2D pixel coordinate system, as use the given tracking information to compute velocity vectors for each of the surrounding objects in 3D space. In addition, we use GPS readings to compute the ego-vehicle velocity vector for the camera sensor where the difference between both velocities is calculated and compared to a threshold for classifying the objects as moving or static. Finally, MaskRCNN [10] segmentation masks are used for refining the obtained output masks. We applied this approach on KITTI-raw frames which have corresponding LiDAR points clouds and tracklets information, obtaining a dataset with a total number of 12919 frames which we split into 80% for training and 20% for testing.

**Color Signal:** Our objective is to develop a complete system for moving object detection to work robustly under any illumination condition. For that purpose we require to evaluate our algorithm, in addition to conventional AD scenes, into other more challenging low illumination environments where camera-only based systems would fail due to the lack of textured information. As far as we know, there exists no dataset providing low-illumination or night scenes in addition to the information needed to generate our MOD annotation. For that reason, we make use of the Image-to-Image translation technique of [22] to generate dark images from the KITTI dataset that mimic night AD scenes. To be able to generate dark realistic frames, we trained UNIT [22] network using 2000 KITTI[8] images and 2000 night images from [41]. Figure 2 shows a sample of our newly generated dataset which we call Dark-KITTI. It comprises of 12919 night images corresponding to KITTI-raw frames. Other approaches such as [44, 21] have been attempted to simulate Dark-KITTI images, however we found [22] to be more realistic as illustrated in the final row of Figure 2.

Figure 2: An example of our different night generation methods, **Top to Bottom:** Input KITTI Image, Neural Style Transfer [21], CycleGAN[43], UNIT[22]

**Motion Signal:** The key input for moving object detection that we give to our system is the motion information obtained from the scene. In order to build an illumination-independent system, we intend to perceive motion from both camera and LiDAR sensors. Motion can be either implicitly learned from temporally sequential frames, or provided explicitly to the system through an input motion map, as for example optical flow maps. For obtaining motion from LiDAR information, we leverage a recent approach [35] that learns to model optical flow maps from LiDAR point clouds. Using this approach, we have the advantage of understanding motion of the surrounding scene even in darkness because LiDAR is illumination independent. In addition to these optical flow maps from LiDAR which we term as "lidarFlow", we generate image-based optical flow using the FlowNet [13] algorithm over RGB images which we term as "rgbFlow". There is a significant degradation on rgbFlow when it is generated from the Dark-KITTI dataset compared to standard KITTI frames, which is expected given that rgbFlow is illumination-dependent.

In our experiments, we prove that both lidarFlow and rgbFlow are complementary to each other and that the inclu-

sion of LiDAR-based motion signals significantly improve MOD results. In order to align our images with the output from [35], we crop the upper part of the dataset frames to be 256x1224 which has no impact on MOD because the moving objects are in the lower part of the image. Figure 3 shows a sample of our generated Dark-KITTI dataset along with the corresponding optical flow maps generated from [13, 35]. It can be observed that RgbFlow using high-illumination images during day provide high intensity motion vectors. However, there exists some distortions such as the ones due to shadow on the ground as illustrated on the 3rd row in Figure 3, where shadow pixels are perceived as moving pixels and also combined with the moving pixels from the cars. For low-illumination rgbFlow in the fourth row, it can be appreciated that it is hard for image-based optical flow algorithms to compute motion vectors in bad lighting conditions, obtaining more distortions in the output flow map. On the other hand, lidarFlow in the final row provides improved optical flow in such challenging conditions where there are less distortions than rgbFlow at night, and no shadow-based distortion because LiDAR does not capture color textures. Yet, due to the sparsity of the LiDAR point clouds which increases with further objects, motion of far objects is modelled with difficulty compared to flow maps from dense RGB images.

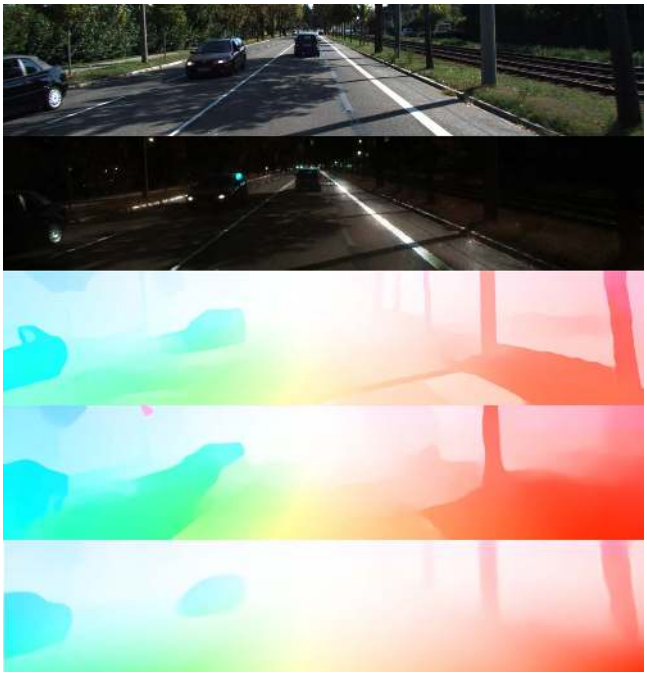

Figure 3: Sample from our Dark-KITTI dataset and the corresponding optical flow images. **Top to Bottom:** KITTI image; Dark-KITTI image; rgbFlow from KITTI image using FlowNet[13]; degraded rgbFlow on Dark-KITTI image; lidarFlow[35] obtained just using LiDAR information.

### 3.2. Network Architecture

In this section, we detail our baseline architectures, and the different implemented fusion approaches.

**Baseline Architecture:** We set our baseline based on [7], which presents an encoder-decoder schema. Our encoder is responsible of extracting features before the upsampling phase done by the decoder and is based on [42], which uses point-wise group convolutions and channel shuffling. This in turn reduces computation cost at a high accuracy level which is perfect for a real-time application such as needed on autonomous driving systems. Our decoder is based on [23] which is composed of three deconvolution layers that provide the final output image size. This approach has the advantage of low complexity as well as provides a lightweight network architecture able to fit on autonomous driving embedded platforms. Detailed analysis of efficient design techniques for segmentation is discussed in [31, 2]. Two classes are used to train the network, i.e, Moving and Non-Moving. In addition to the static objects, background pixels are considered as Non-Moving, therefore the number of static pixels exceeds the number of moving pixels. Weighted cross-entropy is used to overcome this class imbalance problem. We make use of this architecture to evaluate a baseline performance using RGB images only.

**Early Fusion:** Early-Fusion is referred to as data-fusion where fusion is done on the data level before any feature extraction. The same baseline network architecture is utilized in this case, however the input data is concatenated at the very beginning. This architecture has the advantage of low-complexity compared to Mid-Fusion approach, as the number of weights is kept similar to the baseline architecture being the main difference on the input layer only.

**Mid Fusion:** Mid-Fusion refers to feature-level-fusion where features are extracted from each input separately using an encoder that is exclusive to each input. Fusion is done by concatenating feature maps that are generated from each stream before upsampling in the decoder. This architecture provides the best fusion performance, however it has higher cost than early-fusion as the number of weights in the encoder part is doubled.

**Hybrid Fusion:** This architecture makes use of both early and mid-Fusion. We use it in various experiments as illustrated in Table 1, in an attempt to maximize the benefit of the input modalities while avoiding too much complexity for the model at hand. For instance, we fuse 4 inputs, i.e, RGB, rgbFlow, lidarFlow, LiDAR depth through early-fusion in one branch between RGB and rgbFlow, and early-fusion in another branch for LiDAR depth and lidarFlow. The output of both branches is fused through Mid-Fusion.

**Proposed Architecture:** We aim at finding the best schema to combine RGB images, rgbFlow and lidarFlow. For that purpose, we construct a three-stream mid-Fusion

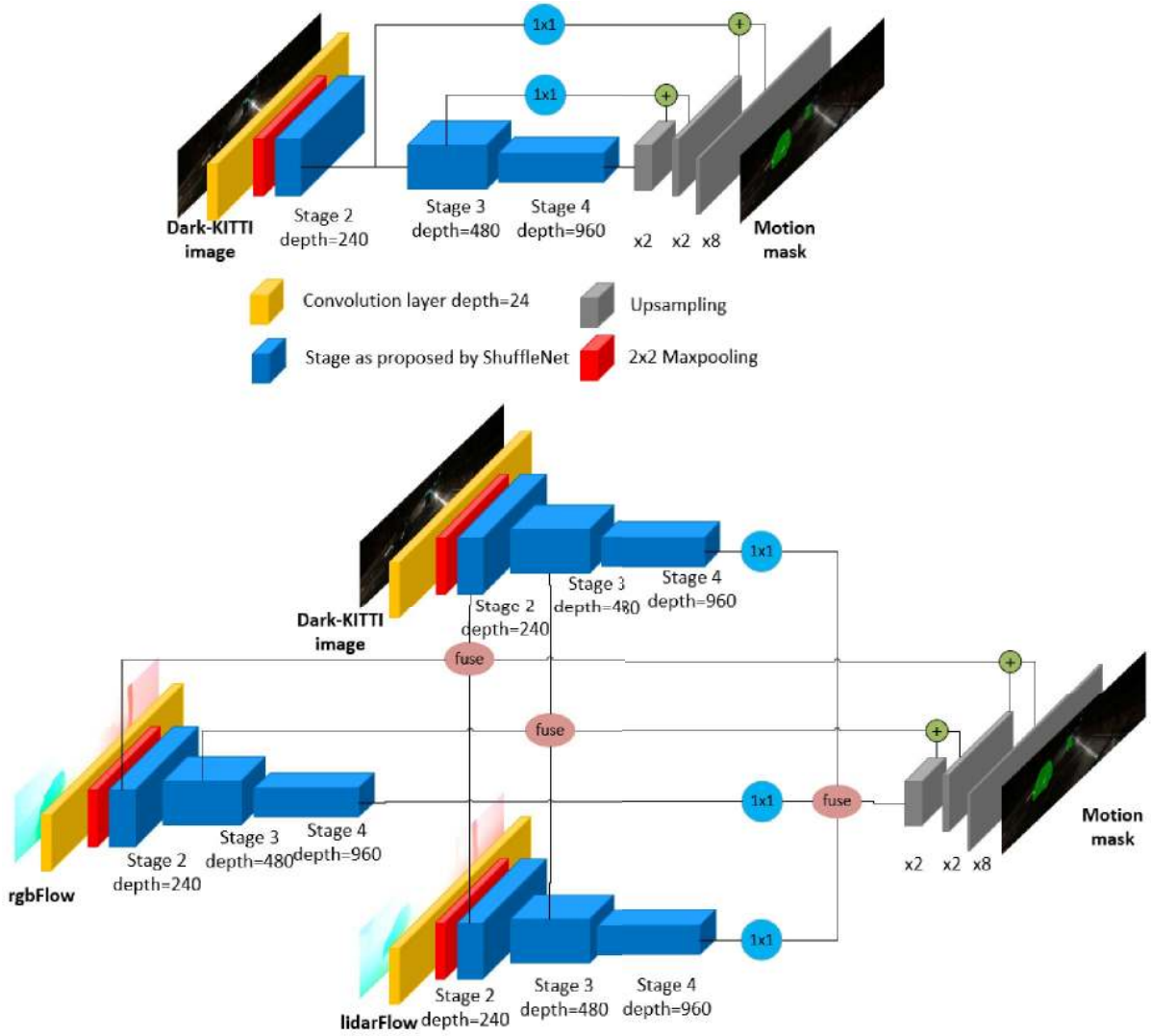


Figure 4: **Top:** Baseline architecture based on [7]. **Bottom:** Proposed fusion architecture.

network which has three encoders for RGB, rgbFlow and lidarFlow separately. We evaluate this approach on KITTI and Dark-KITTI datasets, where results demonstrate the improved performance on both datasets as detailed in section 4.2.

# 4. Experiments

## 4.1. Experimental Setup

In all our experiments, ShuffleSeg [7] model was used with pretrained ShuffleNet encoder on Cityscapes dataset for Semantic Segmentation. For the decoder part, FCN8s decoder has been utilized with randomly initialized weights. L2 regularization with weight decay rate of $5e^{-4}$ and Batch Normalization are incorporated. We trained all our models End-To-End with weighted binary cross-entropy loss for 200 epochs and batch size 6. Adam optimizer is used with learning rate of $1e^{-4}$. For inputs with number $n$ of channels lower than 3, we discarded the difference of depth from the filters of the first convolutional layer. For the rest of inputs, we increased the depth of the filters by the first $n$ channels of the single filter to match the first layer with the new input shape, initializing the corresponding weights randomly.

## 4.2. Experimental Results

We provide a table of quantitative results for both day and night images evaluated on KITTI and Dark-KITTI datasets. Qualitative evaluation on both datasets is illustrated in Figure 5.

Table 1 demonstrates our results using mean IoU metric for both moving and background class and IoU for the moving objects, in addition to class-wise IoU for "Moving" class. We refer to early-fusion by "x" while "+" denotes mid-fusion where both of them together imply hybrid fu-

sion. RGB-only experiments serve as a baseline for comparative purpose where we evaluate our network architecture to segment moving objects using color information only without either explicit or implicit motion signal for the network. Significant improvement for 13% in moving class IoU has been observed after fusion with optical flow, which is consistent with previous conclusions in [32, 29]. We attempt to minimize complexity through early-fusion architecture as we focus on real-time architecture for autonomous driving. However it is found that early-fusion architecture only (RGB x rgbFlow) is not capable of extracting the required features compared to Mid-Fusion which is consistent with other literature such as [27, 4]. Thus we continue our experiments using Mid or Hybrid fusion. Mid-Fusion experiment with rgbFlow (RGB + rgbFlow) serves as a comparison baseline as well because our motivation is to evaluate the augmentation of motion information from LiDAR sensor. (RGB + lidarFlow) shows improved performance over RGB-only, however overall accuracy is still below (RGB + rgbFlow).

Nevertheless, we argue that both lidarFlow and rgbFlow are complementary to each other where rgbFlow benefits from dense color information which is helpful to understand motion for far objects, however illumination plays a great role in the quality of optical flow from RGB images. On the other hand, lidarFlow might not provide the best motion estimate of far objects due to increased sparsity when the objects are far away, however, it is illumination independent due to relying on TOF concept which is perfect for low illumination scenes motion estimation. Our approach is proven experimentally through the (RGB + rgbFlow + lidarFlow) experiment where we obtain absolute improvement of 4% and relative improvement of 10% in IoU over (RGB + rgbFlow). We attempt to fuse optical flow information before feature extraction through hybrid-fusion (RGB + (rgbFlow x lidarFlow)), in addition to experimentation of leveraging depth points through a two stream approach (RGB x rgbFlow) + (LiDAR x lidarFlow). LidarFlow augmentation shows improvement in results over the baseline (RGB + rgbFlow) which proves our approach. However, our three-stream approach gives the network more flexibility to combine features from each input for maximum accuracy.

Implicit motion learning has been explored in (RGB time $t$ x RGB time $t+1$) + (LiDAR depth $t$ x LiDAR depth $t+1$) where the network is expected to learn motion implicitly without optical flow computation. An improvement is observed compared to RGB-only baseline however we obtain degradation in performance compared to explicit motion learning, and this is expected because the network learns to model motion vectors implicitly in addition to its original task which is MOD. We evaluate our approach on KITTI dataset, and we show that lidarFlow augmentation improves accuracy of moving objects even in high-illumination images where 2% improvement in IoU is observed compared to camera-only solution. These results demonstrate that our approach is beneficial for motion segmentation task regardless of illumination parameter which was a drawback in the previous literature.

Table 1: Quantitative results on KITTI and Dark-KITTI. "+" refers to Mid-Fusion. "x" refers to Early-Fusion. Both together refer to Hybrid-Fusion.

| Type | mIoU | Moving IoU |
|---|---|---|
| Dark-KITTI | | |
| RGB-only | 62.6 | 26.5 |
| RGB + rgbFlow | 69.2 | 39.5 |
| RGB x rgbFlow | 61.68 | 24.86 |
| RGB + lidarFlow | 68.7 | 38.5 |
| (RGB time $t$ x RGB time $t+1$) + (LiDAR depth $t$ x LiDAR depth $t+1$) | 66.26 | 33.83 |
| (RGB x rgbFlow) + (LiDAR depth x lidarFlow) | 69.92 | 40.93 |
| RGB + (rgbFlow x lidarFlow) | 69.8 | 40.75 |
| **RGB + rgbFlow + lidarFlow** | **71.2** | **43.5** |
| KITTI | | |
| RGB-only | 65.6 | 32.7 |
| RGB + rgbFlow | 74.24 | 49.36 |
| RGB + lidarFlow | 70.27 | 41.64 |
| (RGB time $t$ x RGB time $t+1$) + (LiDAR depth $t$ x LiDAR depth $t+1$) | 66.68 | 34.67 |
| RGB + (rgbFlow x lidarFlow) | 72.21 | 45.45 |
| **RGB + rgbFlow + lidarFlow** | **75.3** | **51.46** |

Table 2: Comparison between the tested architectures for MOD task. Frame per second (fps) is used as a metric to evaluate real-time performance. Evaluation is performed on 256x1224 resolution images on Titan X Pascal GPU.

| Type | fps |
|---|---|
| Baseline architecture | 40 |
| Two-Stream Mid-Fusion architecture | 25 |
| Three-Stream proposed Mid-Fusion architecture | 18 |

Figure 5 demonstrates our results obtained in Table 1. The first column shows results of our algorithm on KITTI dataset and the second one reports Dark-KITTI results. The input RGB images are shown in the first row. The second row shows the input optical flow maps of KITTI and Dark-KITTI. The third row shows lidarFlow map and ground truth. Fourth row reports results of MOD using only color information as an input. It is shown that the network only learned to segment the cars and not the moving cars as shown in both KITTI and Dark-KITTI results. Some of the parked cars are not segmented because it might be implicitly learned that cars in that position are not interesting. However, this is not based on motion information, and this

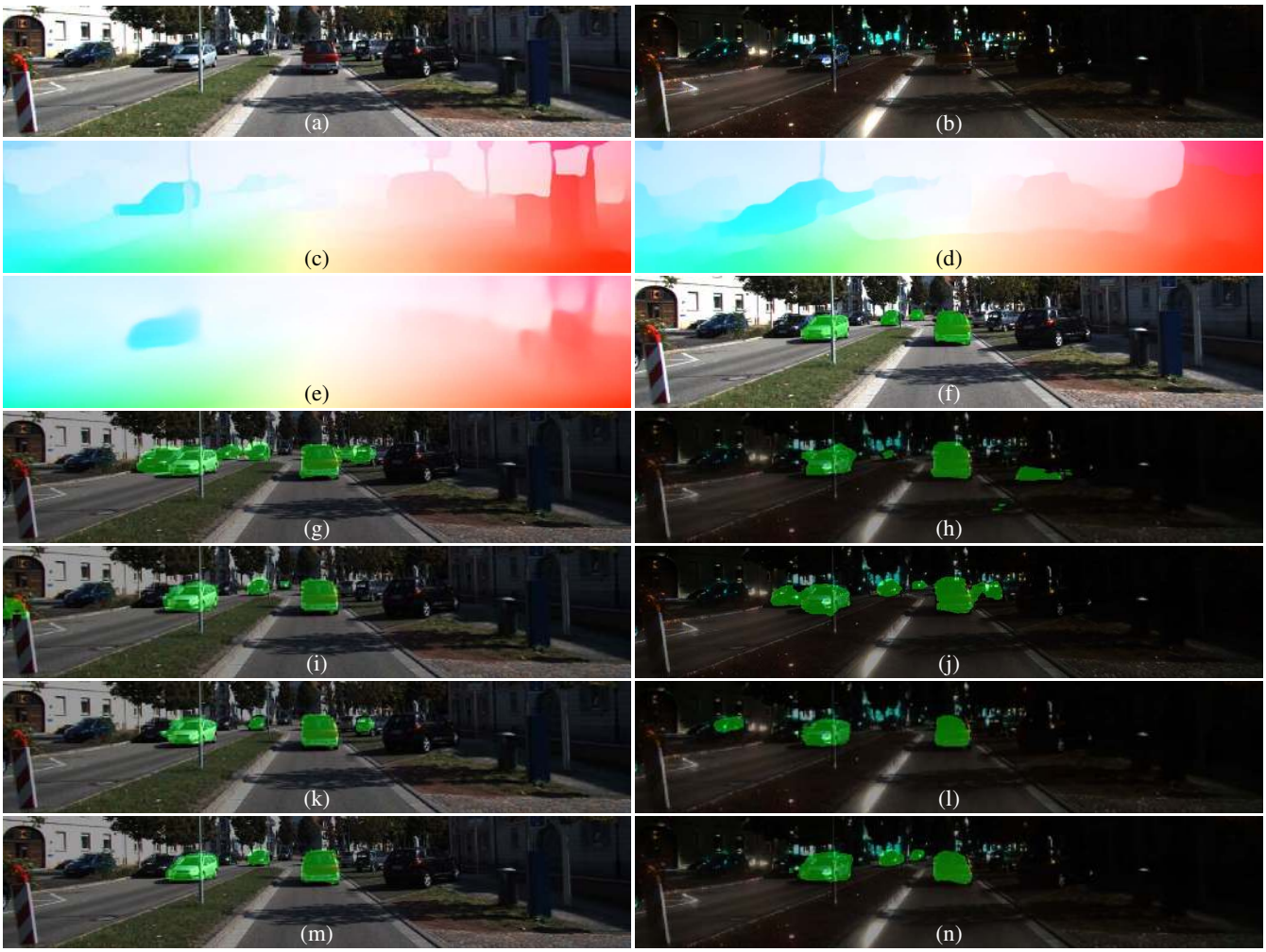


Figure 5: Qualitative comparison of our algorithm on KITTI and Dark-KITTI datasets. First column shows inputs and results on KITTI while second shows results on Dark-KITTI. **(a),(b)** show the input RGB images. **(c),(d)** show rgbFlow. **(e)** shows lidarFlow. **(f)** shows Ground Truth. **(g),(h)** show output using RGB-only. **(i),(j)** show output of (RGB + rgbFlow). **(k),(l)** show output of (RGB + lidarFlow). **(m),(n)** show output of (RGB + rgbFlow + lidarFlow).

is expected because there is no motion information given to the algorithm either explicitly or implicitly.

In Dark-KITTI, only two vehicles are segmented because of low illumination where it is even hard to segment them using human eyes. Fusion with optical flow in the fifth row has improved results significantly on both datasets however, there are too many false positives in Dark-KITTI dataset as in (h) due to inaccurate optical flow because of low illumination of the scene. The sixth row shows results of fusion of color information from camera and motion information from LiDAR. Results show improved performance over (RGB + rgbFlow) especially on Dark-KITTI dataset. This is due to illumination independent optical flow from lidarFlow [35]. However, far objects are still not captured correctly due to increased sparsity with far objects. The seventh row demonstrate the results of our proposed architecture which combines color information, motion information from both camera and LiDAR sensors. Results show the benefit of fusion where the network was able to maximize accuracy from both sensors and segment the scene moving objects.

Figure 6 shows an example of failure of our algorithm where it is shown that output without augmentation of lidarFlow in a high-illumination image is slightly better than using lidarFlow. In this sample, the ego-vehicle is static, and there is only one car that is moving in the scene as illustrated in ground truth. The rgbFlow obtained during day which is shown in (c) provides maximum accuracy when it is fused with RGB as illustrated in (i). Due to inaccurate motion map obtained from LiDAR which is shown in (e), some distortions took place when this input was fused with rgbFlow. This is illustrated in (m) compared to (i). However, the distortion is minimal where the network is still able to learn motion mask correctly even with noisy li-

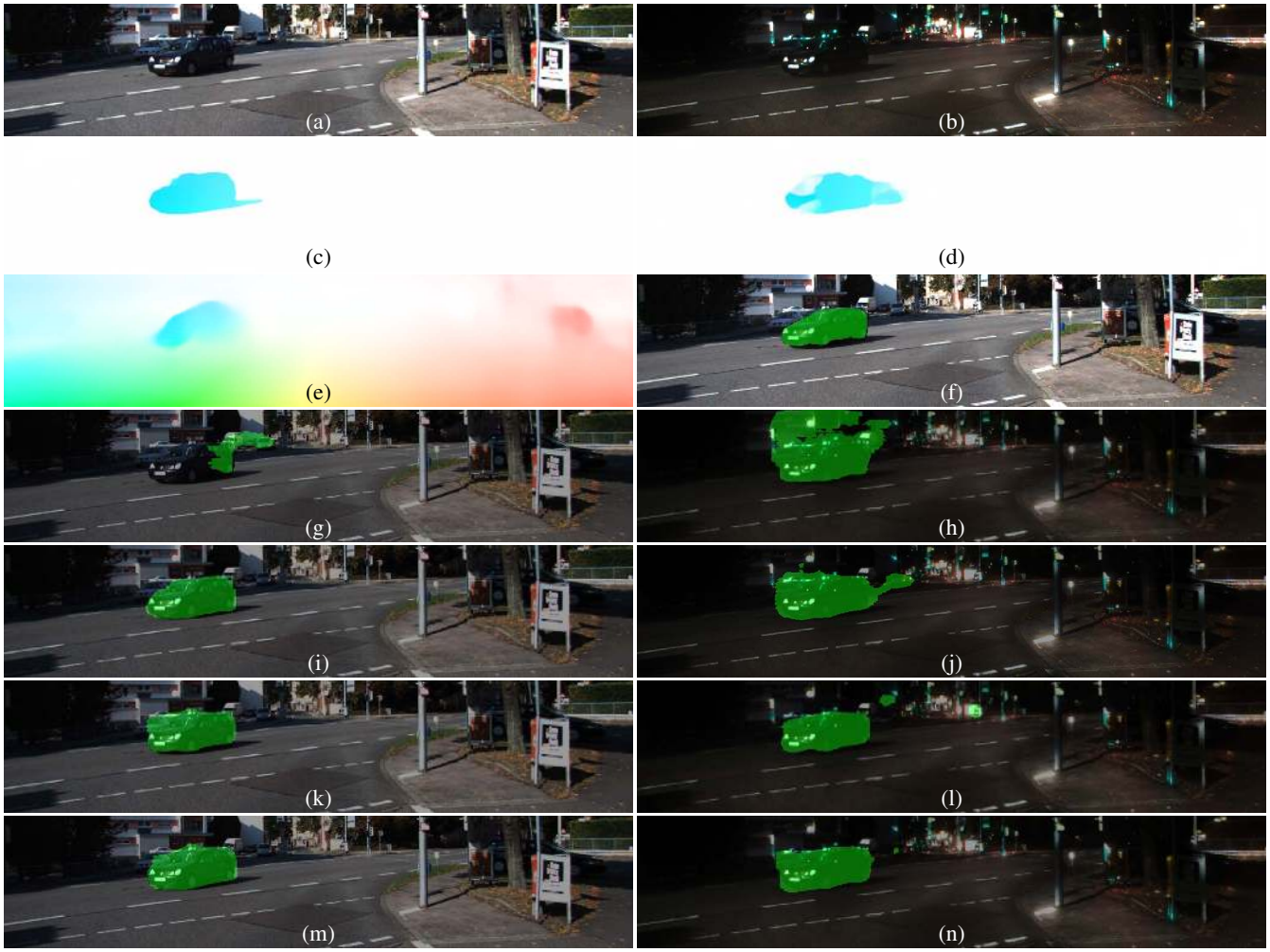


Figure 6: Qualitative comparison of our algorithm on KITTI and Dark-KITTI datasets. First column shows inputs and results on KITTI while second shows results on Dark-KITTI. **(a),(b)** show the input RGB images. **(c),(d)** show rgbFlow. **(e)** shows lidarFlow. **(f)** shows Ground Truth. **(g),(h)** show output using RGB-only. **(i),(j)** show output of (RGB + rgbFlow). **(k),(l)** show output of (RGB + lidarFlow). **(m),(n)** show output of (RGB + rgbFlow + lidarFlow).

darFlow. Moreover, overall moving IoU has improved with 2% after augmentation of lidarFlow with rgbFlow for high-illumination images as illustrated in Table 1. On the other hand, for Dark-KITTI dataset, the fusion with the noisy lidarFlow improves performance of low-illumination images as illustrated in (n) compared to (j) which provides that our algorithm is illumination independent and works perfectly in all lighting conditions. Table 2 shows real-time evaluation performance of our algorithm. Our proposed model runs 18 fps which is suitable for real-time application such as the autonomous driving. The results are reported using images of resolution 256x1224 on Titan X Pascal GPU.

## 5. Conclusions

We explored the impact of leveraging LiDAR sensor for understanding scene motion for MOD especially for low-illumination autonomous driving conditions. We created our own dataset Dark-KITTI to evaluate our algorithm in low-light conditions by extending the public MOD dataset [32]. We constructed different fusion algorithms to empirically study best fusion methodology. We proposed a novel architecture that fuses color signal with motion information that is captured from both camera and LiDAR sensors. Our model is evaluated on both night and day images and we obtain improved performance in both of them. The proposed architecture is designed for real-time performance for autonomous driving application where our most complex algorithm runs at 18 fps. We hope that this study encourages further research in construction of better fusion networks.

## References


[1] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019. 3

[2] A. Briot, P. Viswanath, and S. Yogamani. Analysis of efficient cnn design techniques for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 663–672, 2018. 4

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. 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. 3

[4] A. Dewan, T. Caselitz, G. D. Tipaldi, and W. Burgard. Motion-based detection and tracking in 3d lidar scans. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4508–4513. IEEE, 2016. 2, 6

[5] A. Dewan, G. L. Oliveira, and W. Burgard. Deep semantic classification for 3d lidar data. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3544–3549. IEEE, 2017. 2

[6] B. Drayer and T. Brox. Object detection, tracking, and motion segmentation for object-level video segmentation. *arXiv preprint arXiv:1608.03066*, 2016. 2

[7] M. Gamal, M. Siam, and M. Abdel-Razek. ShuffleSeg: Real-time semantic segmentation network. *arXiv preprint arXiv:1803.03816*, 2018. 4, 5

[8] A. Geiger, P. Lenz, and R. 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. 3

[9] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Asian conference on computer vision*, pages 213–228. Springer, 2016. 2

[10] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 3

[11] M. Heimberger, J. Horgan, C. Hughes, J. McDonald, and S. Yogamani. Computer vision in automated parking systems: Design, implementation and challenges. *Image and Vision Computing*, 68:88–101, 2017. 1

[12] J. Horgan, C. Hughes, J. McDonald, and S. 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

[13] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1647–1655, 2016. 3, 4

[14] S. D. Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 2117–2126. IEEE, 2017. 2

[15] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960. 2

[16] D. Konig, M. Adam, C. Jarvers, G. Layher, H. Neumann, and M. Teutsch. Fully convolutional region proposal networks for multispectral person detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 49–56, 2017. 2

[17] V. R. Kumar, S. Milz, C. Witt, M. Simon, K. Amende, J. Petzold, S. Yogamani, and T. Pech. Monocular fisheye camera depth estimation using sparse lidar supervision. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2853–2858. IEEE, 2018. 2

[18] B. Li. 3d fully convolutional network for vehicle detection in point cloud. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1513–1518. IEEE, 2017. 2

[19] B. Li, T. Zhang, and T. Xia. Vehicle detection from 3d lidar using fully convolutional network. *arXiv preprint arXiv:1608.07916*, 2016. 2

[20] C. Li, D. Song, R. Tong, and M. Tang. Illumination-aware faster r-cnn for robust multispectral pedestrian detection. *Pattern Recognition*, 85:161–171, 2019. 2

[21] Y. Li, M.-Y. Liu, X. Li, M.-H. Yang, and J. Kautz. A closed-form solution to photorealistic image stylization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 453–468, 2018. 3

[22] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pages 700–708, 2017. 3

[23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 4

[24] K. E. Madawy, H. Rashed, A. E. Sallab, O. Nasr, H. Kamel, and S. Yogamani. Rgb and lidar fusion based 3d semantic segmentation for autonomous driving, 2019. 2

[25] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070, 2015. 2

[26] S. Milz, G. Arbeiter, C. Witt, B. Abdallah, and S. 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. 2

[27] H. Rashed, A. El Sallab, S. Yogamani, and M. ElHelw. Motion and depth augmented semantic segmentation for autonomous navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2, 6

[28] H. Rashed, S. Yogamani, A. E. Sallab, P. Krízek, and M. ElHelw. Optical flow augmented semantic segmentation networks for automated driving. In *VISIGRAPP*, 2019. 2

[29] M. Siam, S. Eikerdawy, M. Gamal, M. Abdel-Razek, M. Jagersand, and H. Zhang. Real-time segmentation with appearance, motion and geometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5793–5800. IEEE, 2018. 2, 6

[30] M. Siam, S. Elkerdawy, M. Jagersand, and S. 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. 2

[31] M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, and M. Jagersand. Rtseg: Real-time semantic segmentation comparative study. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1603–1607. IEEE, 2018. 4

[32] M. Siam, H. Mahgoub, M. Zahran, S. Yogamani, M. Jagersand, and A. El-Sallab. ModNET: Moving object detection network with motion and appearance for autonomous driving. *arXiv preprint arXiv:1709.04821*, 2017. 2, 3, 6, 8

[33] M. Uricar, P. Krizek, G. Sistu, and S. Yogamani. Soilingnet: Soiling detection on automotive surround-view cameras. *arXiv preprint arXiv:1905.01492*, 2019. 2

[34] V. Vaquero, I. del Pino, F. Moreno-Noguer, J. Sola, A. Sanfeliu, and J. Andrade-Cetto. Deconvolutional networks for point-cloud vehicle detection and tracking in driving scenarios. In *2017 European Conference on Mobile Robots (ECMR)*, pages 1–7. IEEE, 2017. 2

[35] V. Vaquero, A. Sanfeliu, and F. Moreno-Noguer. Hallucinating dense optical flow from sparse lidar for autonomous vehicles. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 1959–1964. IEEE, 2018. 3, 4, 7

[36] J. Vertens, A. Valada, and W. 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. 3

[37] J. Wagner, V. Fischer, M. Herman, and S. Behnke. Multispectral pedestrian detection using deep fusion convolutional neural networks. In *ESANN*, 2016. 2

[38] S. Wehrwein and R. Szeliski. Video segmentation with background motion models. In *BMVC*, volume 245, page 246, 2017. 2

[39] M. Yahiaoui, H. Rashed, L. Mariotti, G. Sistu, I. Clancy, L. Yahiaoui, V. R. Kumar, and S. Yogamani. Fisheyemodnet: Moving object detection on surround-view cameras for autonomous driving. *arXiv preprint arXiv:1908.11789*, 2019. 2

[40] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uřičář, S. Milz, M. Simon, K. Amende, C. Witt, H. Rashed, S. Chennupati, S. Nayak, S. Mansoor, X. Perroton, and P. Perez. : A multi-task, multi-camera fisheye dataset for autonomous driving. *CoRR*, abs/1905.01489, 2019. To appear in ICCV 2019. 2

[41] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018. 3

[42] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018. 4

[43] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2242–2251, 2017. 3

[44] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017. 3