sensors

MDPI

Article

Real-Time Semantic Segmentation for Fisheye Urban Driving Images Based on ERFNet [†]

Álvaro Sáez, Luis M. Bergasa *, Elena López-Guillén, Eduardo Romera, Miguel Tradacete, Carlos Gómez-Huélamo and Javier del Egido

Electronics Department, University of Alcalá, Campus Universitario, 28805 Alcalá de Henares, Spain; alvaro.saezc@edu.uah.es (Á.S.); elena.lopezg@uah.es (E.L.-G.); eduardo.romera@edu.uah.es (E.R.); miguel.tradacete@edu.uah.es (M.T.); carlos.gomezh@edu.uah.es (C.G.-H.); javier.egido@edu.uah.es (J.d.E.)

* Correspondence: luism.bergasa@uah.es; Tel.: +34-91-885-6569

[†] This paper is an extended version of our paper published in Sáez, A.; Bergasa, L.M.; Romeral, E.; López, E.; Barea, R.; Sanz, R. CNN-based Fisheye Image Real-Time Semantic Segmentation. In Proceedings of IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018.

Received: 12 December 2018; Accepted: 21 January 2019; Published: 25 January 2019

Abstract: The interest in fisheye cameras has recently risen in the autonomous vehicles field, as they are able to reduce the complexity of perception systems while improving the management of dangerous driving situations. However, the strong distortion inherent to these cameras makes the usage of conventional computer vision algorithms difficult and has prevented the development of these devices. This paper presents a methodology that provides real-time semantic segmentation on fisheye cameras leveraging only synthetic images. Furthermore, we propose some Convolutional Neural Networks(CNN) architectures based on Efficient Residual Factorized Network(ERFNet) that demonstrate notable skills handling distortion and a new training strategy that improves the segmentation on the image borders. Our proposals are compared to similar state-of-the-art works showing an outstanding performance and tested in an unknown real world scenario using a fisheye camera integrated in an open-source autonomous electric car, showing a high domain adaptation capability.

Keywords: fisheye; intelligent vehicle; CNN; deep learning; distortion

1. Introduction

The most critical task for autonomous vehicles is understanding their surroundings. A good real-time scene-comprehension is vital to a vehicle so it can drive in an unknown environment in a safe way. The semantic segmentation task proposes a solution for this challenge based on image pixel-level classification in multiple semantic categories such as vehicles, pedestrians, traffic signals, etc., satisfying most of the vehicle needs in a unified way [1].

The remarkable success of semantic segmentation solutions during the last few years has been closely related to the breakthrough of deep learning methods, which have proven to widely outperform previous state-of-the-art machine learning techniques [2,3]. Among these techniques, the success of Convolutional Neural Networks (CNNs) has been pushed by the development of excellent open-source deep learning frameworks [4,5], by the progression of specific computational hardware such as Graphics Processing Units (GPUs), and by the appearance of large-scale training datasets [6,7].

The comprehension of a vehicle's surroundings becomes even more challenging in complex environments such as urban traffic scenes, where the behavior of dynamic traffic participants like pedestrians or vehicles is unpredictable, or specific situations such as intersections or roundabouts that require big volumes of information to be adequately handled. Accordingly, a full real-time perception of the scene is a compulsory need for autonomous vehicles. Different sensors can be used in order to cover this need as cameras, LiDAR, radar, ultrasound, etc. Cameras clearly stand out among other solutions as they are able to generate real-time high-level semantic information while remaining easy to manage, cheap and present low power consumption.

However, the limited field of view of traditional cameras complicates the management of complex environments since cameras are expected to cover the 360° surroundings. The number of devices that compose the perception system is a critical parameter to be optimized, given that a high number of cameras involve high processing times and the fulfillment of a set of hard tasks such as sensor calibration, synchronization and data-fusion.

Fisheye cameras have started to play an increasingly important role in autonomous vehicles because of their ultra-wide field of view. These devices allow for acquiring more scene information using only a sensor at the cost of radial distortion in the images. With fields of view higher than 180 degrees, only two of these cameras are theoretically needed to cover the all of the vehicle's surroundings. In addition, current autonomous vehicles are betting on redundant and robust perception systems. Fisheye cameras can clearly help in the achievement of these objectives in order to reach safe and reliable driving.

Despite the discussed advantages, distortion associated with these cameras prevents the use of standard computer vision algorithms on the acquired images, making the integration into autonomous vehicles difficult. Furthermore, the application of deep learning techniques to these kinds of images presents many problems such as the lack of large-scale annotated datasets or the management of the distortion, which has caused that only some few works of the state of the art have focused on adapting current semantic segmentation methods to fisheye cameras.

This paper is an extension of our previous conference publication [8]. This work proposes robust deep learning techniques and some CNN architectures able to handle fisheye distortion correctly and that allows real-time fisheye semantic segmentation without the need for using pixel-level hand-annotated images. Moreover, our proposals have been validated in an open-source dataset such as CityScapes and an additional dataset obtained from our open-source autonomous electric car.

This paper is organized as follows: Section 2 examines previous related works. Section 3 introduces the generation of a specific fisheye dataset and some fisheye data augmentation techniques. Sections 4 and 5 present our CNN architecture proposals based on Efficient Residual Factorized Network (ERFNet), the training strategy and the performed experiments. Finally, Section 6 presents some qualitative results for a real autonomous vehicle.

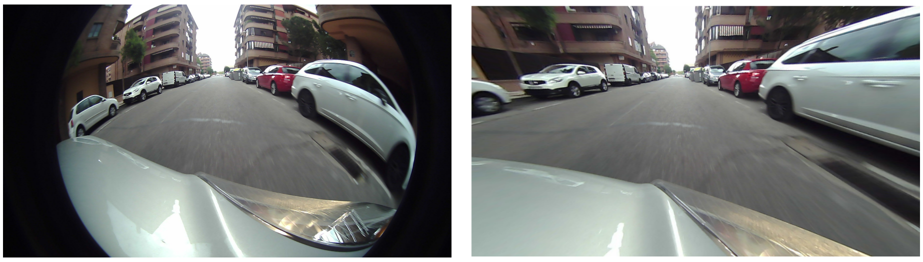
2. Related Work

The main issue with fisheye cameras is how to correctly handle distortion. Distortion is heterogeneous over the different fisheye image areas [9], being a function of both the radial angle and the distance between the principal point of the camera and the image points of the detected objects. This adds complexity to the training of CNNs, as they are forced to learn complicated features that allow the detection of objects with changing appearances depending on their position in the image in order to perform an accurate detection.

An initial approach to deal with the problem is the undistortion of the captured images in order to apply traditional vision techniques [10]. In [11], an end-to-end multi-context collaborative deep network that leveraged semantic information was used to remove distortion from single fisheye images achieving an outstanding performance but with an inadmissible processing time for real-time tasks.

Authors in [12] successfully used a region based CNN (R-CNN) to perform multi-class object detection on panoramic images that were constructed with three fisheye images. The distortion was corrected using a simple and fast approach based on longitude-latitude projection, as correction accuracy was not considered a key issue for object detection.

Nevertheless, none of the previous works achieved a good quality corrected image in a reasonable processing time as the image undistortion process has several difficulties: the strong dependency on intrinsic camera calibration parameters, the high consumption of computational resources that penalizes real-time processes and, finally, a remarkable loss of image quality, leading to information loss all over the image, but especially in the boundaries as shown in Figure 1. These regions are critical, as they gather a big part of the scene information. To deal with this problem, in [13], a CNN-based preprocessing stage and a multi-frame-based view transformation were proposed and applied in an Around View Monitor system (AVM). However, this approach uses separated CNN frameworks for image enhancement and up-scaling and hole filling method can be improved.



(**a**) Original Image (**b**) Undistorted Image

Figure 1. Example of fisheye image undistortion.

These inconveniences have caused the sprouting up of other approaches that try to adapt existing image processing techniques to work with the distorted images directly instead of the opposite. The lack of available large-scale annotated datasets for non-conventional camera images, like fisheye, has forced the generation of synthetic datasets with additional fisheye distortion leveraging existing ones like CityScapes [7]. In [14], the ETH Pedestrian Benchmark [15] and a spherical perspective imaging model were used to generate a fisheye dataset to allow pedestrian detection with ultra-wide Field Of View (FOV) cameras using a Deformable Part Model (DPM) [16]. In [17], the perspective projection equation of equidistant fisheye camera was used to transform CityScapes images in a new distorted dataset using a mathematical remapping relationship. In [18], the same technique was used combined with additional images generated with a SYNTHIA simulator [19].

The most relevant features to be learned in the CNN learning process are the appearance of the detected objects, their shape and their contextual information [20]. Previous works identified that fisheye distortion penalizes the first two points, while the third one becomes especially important as the appearance of the objects becomes closely related to their position in the images.

Multiple ideas have been proposed to incorporate more context information in order to improve the results of the classification task. Most works have focused on obtaining wide receptive fields to capture valuable information. This can be achieved by including down-sampling stages followed by a set of convolutional layers. However, the down-sampling operation implies the reduction of the feature maps' scale and, hence, the loss of information. In [21], dilated convolution or Atrous convolution were proposed to enlarge the receptive field of filters without reducing the resolution nor increasing the number of parameters by adding a fixed separation between kernel elements. Deformable convolutions [22] introduce a similar approach where 2D offsets were added to the kernel sampling locations expanding the receptive field of convolutions and improving the ability of modelling geometric transformations [18] but markedly augmenting the number of the network parameters. In order to avoid the increase of the number of parameters, handcrafted structures, like the pyramidal parsing module [23], have been proposed.

In conclusion, multiple ideas are currently proposed to improve CNN performance on fisheye images and handle fisheye distortion correctly, but most of them are not able to achieve real-time semantic segmentation on real fisheye images without resorting to manually annotated images during training.

3. Fisheye Synthetic Dataset and Data Augmentation

The generation of large-scale datasets involves expensive and time-consuming tasks such as data acquisition and the corresponding data annotation. Semantic segmentation tasks require pixel-wise annotations, which makes labeling extremely difficult for this kind of images. As we advanced in our precious publication [8], our approach consists of taking advantage of public annotated datasets applying distortions models over RGB and labeled images (ground truth) in order to shortcut the hard manual labelling task. Therefore, we have developed a synthetic dataset from CityScapes using a generic fisheye camera model to add artificial distortion to the images.

3.1. Synthetic Fisheye Dataset

Conventional pinhole cameras have a limited field of view defined by their imaging projection, according to the following Equation (1):

$$\rho_{pinhole} = ftan(\theta), \tag{1}$$

where ρ is the distance between the image point and the camera principal point, f is the focal length of the camera and θ is the angle between the incoming light ray and the image principal axis.

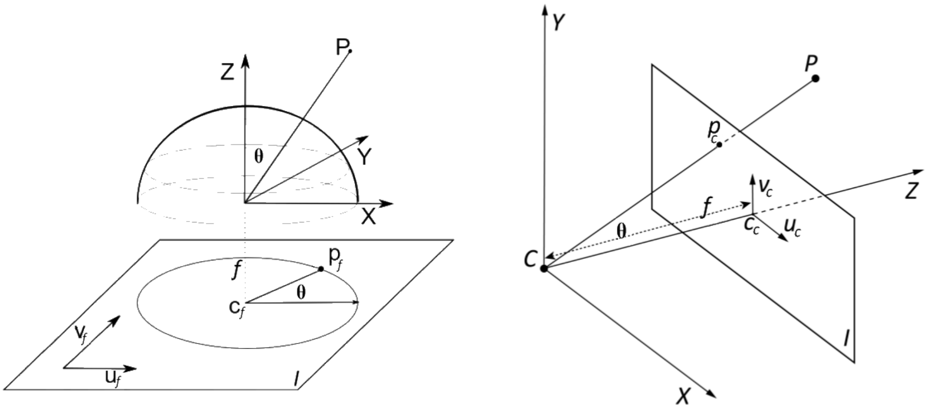
In the case of fisheye camera modelling, there are several mathematical models that can be used to design fisheye lenses. Among them, the most widespread one is the equidistant fisheye, which is described by Equation (2):

$$\rho_{equidistance} = f\theta. \tag{2}$$

Using the previous equations, a remapping can be defined between the pixels on a conventional image ($p_c = (u_c, v_c)$) and its analogous pixels on a synthetic fisheye image ($p_f = (u_f, v_f)$), depending only on the f parameter, which determines the level of added distortion as shown on Equation (3):

$$d_c = ftan(d_f/f), \tag{3}$$

where $d_c = \sqrt{(u_c - u_{cu})^2 + (v_c - v_{cv})^2}$ measures the distance between a single pixel ($p_c = (u_c, v_c)$) and the principal point ($c_c = (u_{cu}, v_{cv})$) for the conventional image, and $d_f = \sqrt{(u_f - u_{fu})^2 + (v_f - u_{fv})^2}$ represents the equivalent distance for the fisheye image. A comparison between these two camera models is represented in Figure 2.

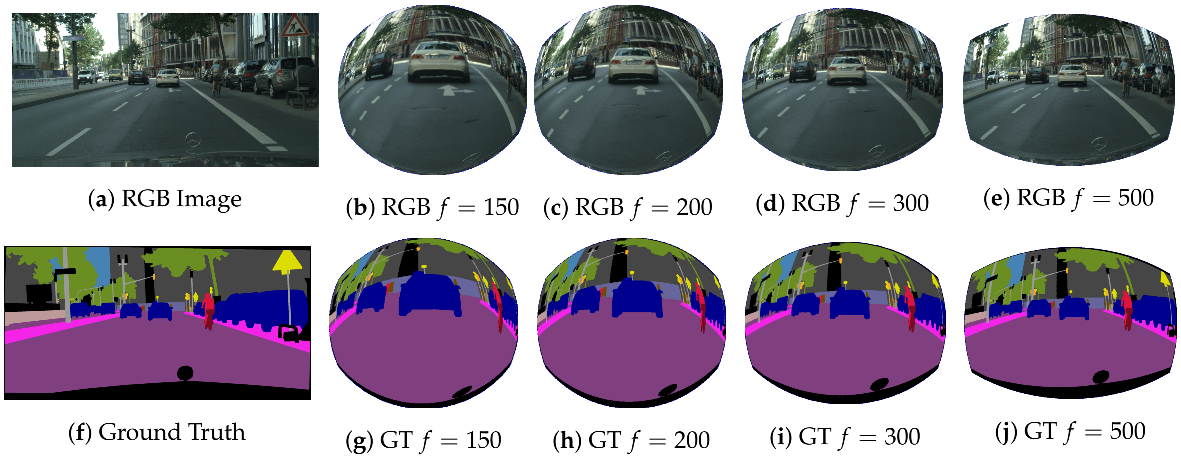


(**a**) Fisheye camera model

(**b**) Pinhole camera model

Figure 2. Comparative between camera models.

Leveraging the previous equation and the CityScapes dataset, a new collection of synthetic distorted images was produced to carry out the training of CNNs with fisheye images representing urban scenes. CityScapes is an optimal dataset for autonomous driving applications, as it is focused on urban scene understanding. It provides 5000 dense pixel-wise annotated images separated into three different subsets (2975 for training, 500 for validation and 1525 for test) and, additionally, another 20,000 with coarse annotations from 27 European cities with 19 classes for evaluation. In our case, only the fine full dataset, including training and validation subsets images, for both RGB and annotated images, were transformed. Original images were resized to 640 × 576, using bi-linear interpolation for RGB images and nearest-neighbor for label images in order to adapt them to our CNN architecture. Some examples of the final synthetic dataset can be seen in Figure 3.


(**a**) RGB Image (**b**) RGB $f = 150$ (**c**) RGB $f = 200$ (**d**) RGB $f = 300$ (**e**) RGB $f = 500$

(**f**) Ground Truth (**g**) GT $f = 150$ (**h**) GT $f = 200$ (**i**) GT $f = 300$ (**j**) GT $f = 500$

Figure 3. Example of synthetic images and ground truth with different distortions.

3.2. Fisheye Data Augmentation

The features learned by a CNN during training rely mostly on the specific images used during this process. Therefore, these features are limited to the domains in which these images were acquired and should have the property to generalize to different domains. However, achieving robustness in other domains is not an easy task, and deep networks are often prone to overfitting even with thousands of training images. This difficulty is even greater when synthetic images are employed during training, given that appearance in synthetic images is less rich and varied than in real images.

In order to obtain general features, data diversity must be high, due to the huge, different patterns CNNs are forced to learn to be able to distinguish between multiple categories in changing detection conditions. Data augmentation aims to enlarge the training datasets, preserving the available labels by applying different transformations.

For the semantic segmentation task, numerous techniques are typically applied such as geometric augmentations (translations, rotations, horizontal flips, etc.), texture augmentations (color jittering, changes in brightness, contrast, etc.) [24] or specific transformations. For instance, authors in [17] proposed an augmentation technique specifically oriented for fisheye images named zoom augmentation. This data augmentation claimed the use of images with various distortions during training by adopting different fixed values for the f parameter from Equation (3) aiming to obtain better generalization abilities. In our previous publication [8], we proposed a modification of this method employing randomly chosen distortions, eliminating the selection process of the fixed distortions and achieving an equal performance.

4. CNN Architecture and Training

The demanding needs of real-time applications have boosted the development of efficient network architectures, leaving behind large deep architectures that achieved outstanding performances at the expense of the consumption of computational resources, using different ideas such as Conditional Random Fields (CRFs) [21], residual layers [25] or dilated convolutions [26]. Initial approaches were able to reach real-time semantic segmentation by strongly reducing the number of network parameters, but obtaining poor performances [27].

Our previous proposal ERFNet [28] achieved a remarkable trade-off between efficiency and accuracy. The network has an encoder–decoder structure, like other efficient CNNs such as Enet [29] or SegNet [30], but demonstrates a notable performance due to the use of non-bottleneck residual layers. The use of these layers is more unusual than the bottleneck layers due to efficiency reasons, but non-bottleneck layers have exposed performance improvements in certain shallow architectures like ResNet. However, ERFNet proposes a redesign of these layers using factorized (1D) kernels to build the residual blocks, in order to reduce computation and achieve an efficient architecture while keeping an equivalent performance to the non-bottleneck layers.

We adopt ERFNet as our baseline CNN architecture. The basic architecture of ERFNet is presented in Figure 4. Encoders and decoders are both built by stacking 1D-non-bottleneck layers in a sequential way. The encoder module consists of a reduced number of layers including three downsampling blocks and convolutional stages. Encoder is meant to take input images and "encode" them into deep features that represent activations to different image classes. Obtaining good features at this point is essential to produce good classification results. We include three downsampler blocks to perform 8× downsampling in total, which was selected to optimize the trade-off between low-res features (which is more efficient and includes more context) and high-res features (which has better feature localization at the pixel level but is more computationally expensive). In addition, we include dilated convolutions in some of the encoder's blocks to effectively increase gathering of context without affecting efficiency or resolution.

The decoder module includes upsampling and convolutional layers and a final classification log–softmax loss layer. The decoder stage is meant to preprocess encoded features up to the input's resolution and provide the final probabilities for each of the trained classes. Thus, the final layer is a volume with a number of slices equal to the number of classes, where each slice contains the per-pixel probabilities of that class. In order to take the predicted (or most probable class), the argmax of this volume is calculated. Many networks use a large decoder, but we chose a relatively small one because the decoder is only meant to upsample features and convert to probabilities, without affecting much to the extraction of good features. Therefore, the encoder does most of the feature extraction job and our light decoder transforms these representations into meaningful outputs in the shape of probabilities.

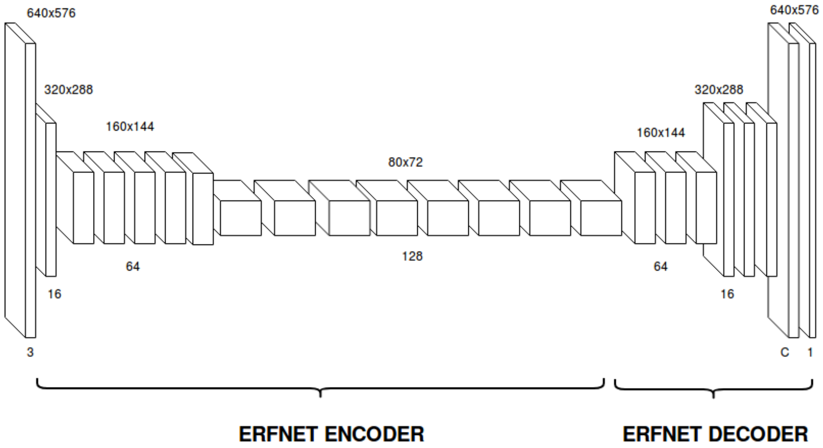


Figure 4. ERFNet baseline architecure.

Considering the significance of context information in fisheye images, we also study the use of an alternative architecture, consisting of replacing the original ERFNet decoder by a handcrafted pyramidal pooling module [31]. Four different pyramid levels are used in the module, including 1/8, 1/4, 1/2 and 1 scale blocks, followed by an upsampling stage and the final log–softmax classification layer. The scheme for this second network is shown in Figure 5 and the layer disposal in Table 1.

The training of both architectures (baseline and modified) is divided into two different stages: on the first one, the encoder module is trained individually using downsampled annotations as ground truth during 90 epochs with a batch size of 6. For the second one, the complete architecture (including the decoder or the pyramidal module) are trained together to produce end-to-end semantic segmentation for another 90 epochs.

The Adam optimization of Stochastic Gradient Descent is used, starting with a learning rate of 5×10^{-4}, which is exponentially decreased on each epoch, and including a weight decay of 1×10^{-4} for regularization. We employ the class weighing technique introduced in [29] $w_{class} = \frac{1}{ln(c+p_{class})}$ fixing $c = 10$ during the entire training.

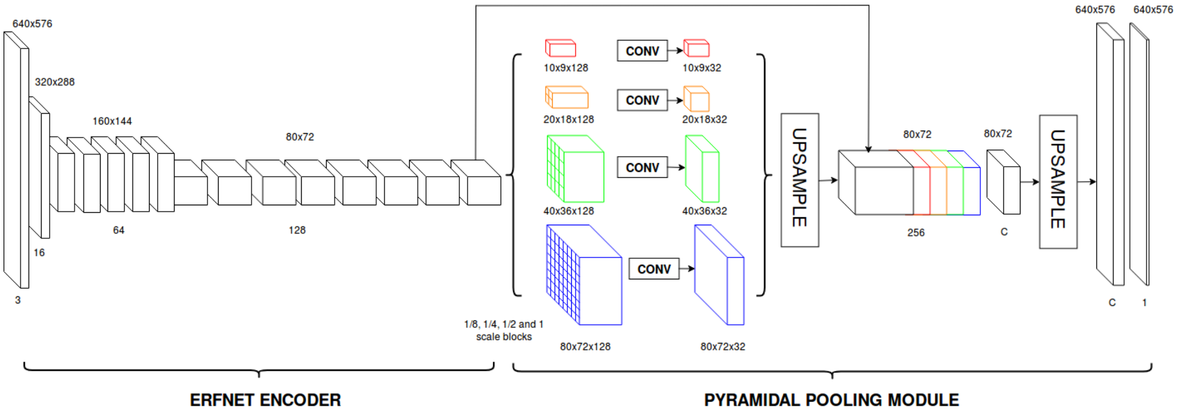


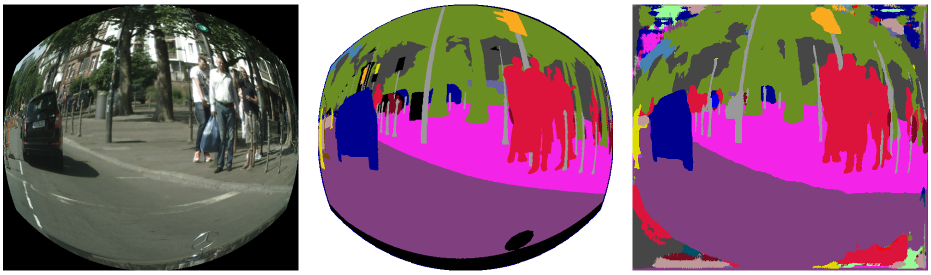
Figure 5. Diagram that depicts the proposed segmentation CNN (ERFNetPSP). Volumes correspond to the feature maps produced by each layer for an example input of 640 × 576.

Table 1. Layer disposal of our proposed architecture.

Layer	Type	Out-F	Out-Re
1	Down-sampler block	16	320 × 288
2	Down-sampler block	64	160 × 144
3–7	5 × Non-bt-1D	64	160 × 144
8	Down-sampler block	128	80 × 72
9	Non-bt-1D (dilated 2)	128	80 × 72
10	Non-bt-1D (dilated 2)	128	80 × 72
11	Non-bt-1D (dilated 4)	128	80 × 72
12	Non-bt-1D (dilated 8)	128	80 × 72
13	Non-bt-1D (dilated 16)	128	80 × 72
14	Non-bt-1D (dilated 2)	128	80 × 72
15	Non-bt-1D (dilated 4)	128	80 × 72
16	Non-bt-1D (dilated 8)	128	80 × 72
17	Non-bt-1D (dilated 2)	128	80 × 72
18a	Layer 17 feature map	128	80 × 72
18b	Pooling and Convolution	32	80 × 72
18c	Pooling and Convolution	32	40 × 36
18d	Pooling and Convolution	32	20 × 18
18e	Pooling and Convolution	32	10 × 9
18	Up-Sampler and Concatenation	256	80 × 72
19	Convolution	C	80 × 72
20	Up-Sampler	C	640 × 576

The black corners in Figure 6a are characteristic of fisheye images. These regions appear on the syntheticly distorted images and on their associated ground truth, as a consequence of the pixel remapping process as shown in Figure 6a. However, the pixels included on those regions are ignored during both training and evaluation. Figure 6c shows an example of segmentation of a synthetic fisheye image.

As it can be seen, these regions present a very heterogeneous segmentation that harms the context information on the borders of the useful parts of the image, which are essential because they contain an important area of the total FOV of the camera. Context information is the most determinant feature for the segmentation of the image borders, due to the strong distortion they present. As a consequence, the performance of the CNN in this area is clearly degraded.



(**a**) Distorted Image (**b**) Distorted Ground Truth (**c**) Segmented Image

Figure 6. Example of heterogeneous segmentation on the borders.

In order to preserve this context information and improve the segmentation on the borders, a new training strategy is proposed. We identify those areas a priori and add an additional 20th class to represent them during training.

5. Experiments

For the validation of the proposed architectures, data augmentation and training strategies, different experiments are presented. The first one evaluates our data augmentation strategy and the second one studies the performance of the different architectures compared to other proposals of the state of the art.

To prove the benefits of our data augmentation proposal, a comparative experiment with other approaches of the state of the art was carried out. For the fixed zoom augmentation, three datasets were generated as in [17] with $f_0 = 159$, $f_1 = 96$ and $f_2 = 242$, respectively. For the random zoom-augmentation, the focal length values were randomly changed following a Gaussian distribution, generating five distorted images for each training image as we did in [8]. Additional data augmentation including color jittering (randomly modifying brightness, saturation and contrast to develop a more robust training to light changes), random-cropping (to prepare the CNN for scale and aspect-ratio and scale changes), mirroring, rotations (between 0 and 90 degrees) and arbitrary 0–2 pixels translations was carried out (full data augmentation).

Table 2 presents the experimental results. The first three lines correspond to other state-of-the-art works. Dilation 10 [26] includes dilated convolutions to improve the aggregation of information. ResNet-26 presents the score for a modified 26 layer ResNet with bottleneck blocks and dilated convolutions [17]. OPPNet [17] is composed of a dilated fully convolutional feature extractor block followed by an overlapping pyramid pooling module which analyzes the images at different scales aiming to obtain more context information.

According to Table 2 results, our ERFNet proposals outperform previous state-of-the-art work even without data augmentation. Using ImageNet pre-training improves performance regarding to basic training. The three data augmentation techniques improve both basic and pretrained performances, showing the importance of data diversity. Random and fixed zoom-augmentation provide similar results for both architectures, the random augmentation being more beneficial for the ERFNetPSP and the fixed one for the basic ERFNet architecture. The application of additional data augmentation techniques improves the final results even more, reaching 58.3% and 59.3%, respectively. Both networks clearly stand out in front of previous works, exceeding by 4.8 and 3.8% the previous best score (OPPNet).

In a second experiment, an alternative synthetic dataset with a lower distortion ($f = 240$) was generated. The two proposed architectures were trained without any data augmentation, tested on the validation subset and compared to other state-of-the-art results in the same conditions for fair comparison. Training with an additional class proposed on Section 4 was also tested (ERFNet20), in order to study its benefits.

Table 2. Architectures and data augmentation performances.

Network	Data Augmentation	Class IoU (%)
Dilation10 [26]	None	51.7
ResNet-26 [17]	None	52.0
OPPNet [17]	None	52.6
	Fixed z-aug	**54.5**
ERFNetPSP	None	53.3
	Pretrained	55.5
	Fixed z-aug	55.9
	Random z-aug	56.2
	Full & pretrained	**58.3**
ERFNet [28]	None	55.6
	Pretrained	56.3
	Random z-aug	56.8
	Fixed z-aug	57.0
	Full & pretrained	**59.3**

The comparison includes a set of network models derived from ERFNet [18]. The original ERFNet was re-implemented in MXNet [5] with additional batch normalization layers after each convolutional layer and with 2 × 2 kernels with a stride of 2 on the deconvolution layers (ERFNetMx). Additionally, two extra models were proposed incorporating restricted deformable convolutions (RDCNet), which use a reduced number of parameters, and factorized restricted deformable convolutions (FRDCNet), which can be implemented using 1D kernels.

Furthermore, some additional CNNs were re-implemented in Pytorch to widen the comparative including: a modified PSPNet [31] built by a ResNet-101 with deformable convolutions as the feature extraction block followed by a pyramid pooling module. A modified DRN-D-54 following the proposal of Dilated Residual Networks [32] and including dilated convolutions [26] and SegNet [30], which is able to provide real-time semantic segmentation at the expense of a loss of performance. Finally, ERFNet20 shows the score for the ERFNet training with an additional class to correctly identify the borders of the images. ERFNet and ERFNetPSP models were trained in two stages, as in the the previous experiment, and the rest of CNNs were trained during 200 epochs, using the proposed parameters for them in their respective publications. Results of this experiment are listed in Tables 3 and 4.

Table 3 shows in the second column the mean class IoU% obtained by the different networks for an image resolution of 640 × 576. In the third column, the forward time in seconds using a single GTX 1080Ti is depicted. A different image resolution of 814 × 512 was adopted for this column in order to compare results with other works of the literature. Results show that the best IoU is achieved by the ERFNetPSP model with 61.6% outperforming the best previous score for this distortion level (RDCNet) by 3.7%. ERFNet achieves a similar score (61.5%) for 19 classes, rising to 62.2% for the 20 classes version (ERFNet20), due to the good segmentation results in the border areas (black zones in the image). In this last case, training improves the detection of the classes with fewer training samples that appear close to the borders as shown in Figure 7. Our ERFNet proposals obtain a higher score than for the rest of CNNs. From the re-implemented group of architectures, only PSPNet (59.2%) improves the RDCNet performance. The DRNet-D-54 achieves a similar score (57.6%) and SegNet clearly has worse performance (50.1%).

The modified MXNet ERFNet presents a poor performance (55.1%), but it is improved by the addition of factorized restricted deformable convolutions (56.1%) and restricted deformable convolutions (57.9%). As shown in Table 3, regarding processing time, ERFNet MX is the fastest architecture, needing only 0.016 s to process a 814 × 512 image and achieving more than 63 fps. The second fastest is RDCNet with 0.018 s and 55 fps, followed by the original ERFNet (50 fps) and ERFNetPSP (>45 fps). From the rest of the tested networks, only SegNet works in real time (>14 fps) and PSPNet and DRN-D-54 have low frame ratings (4 and 6 fps).

Table 3. Performance-efficiency comparison for the presented architectures.

Architecture	**Class IoU (%)** **(640 × 576)**	**Forward Pass Time (s)** **(814 × 512)**
ERFNetPSP	**61.6**	0.022
ERFNet [28]	61.5	0.020
ERFNet20	**62.2**	0.020
PSPNet [31]	59.2	0.22
RDCNet [18]	57.9	0.018
DRN-D-54 [32]	57.6	0.146
FRDCNet [18]	56.1	-
ERFNet MX [18]	55.1	**0.016**
SegNet [30]	50.1	0.07

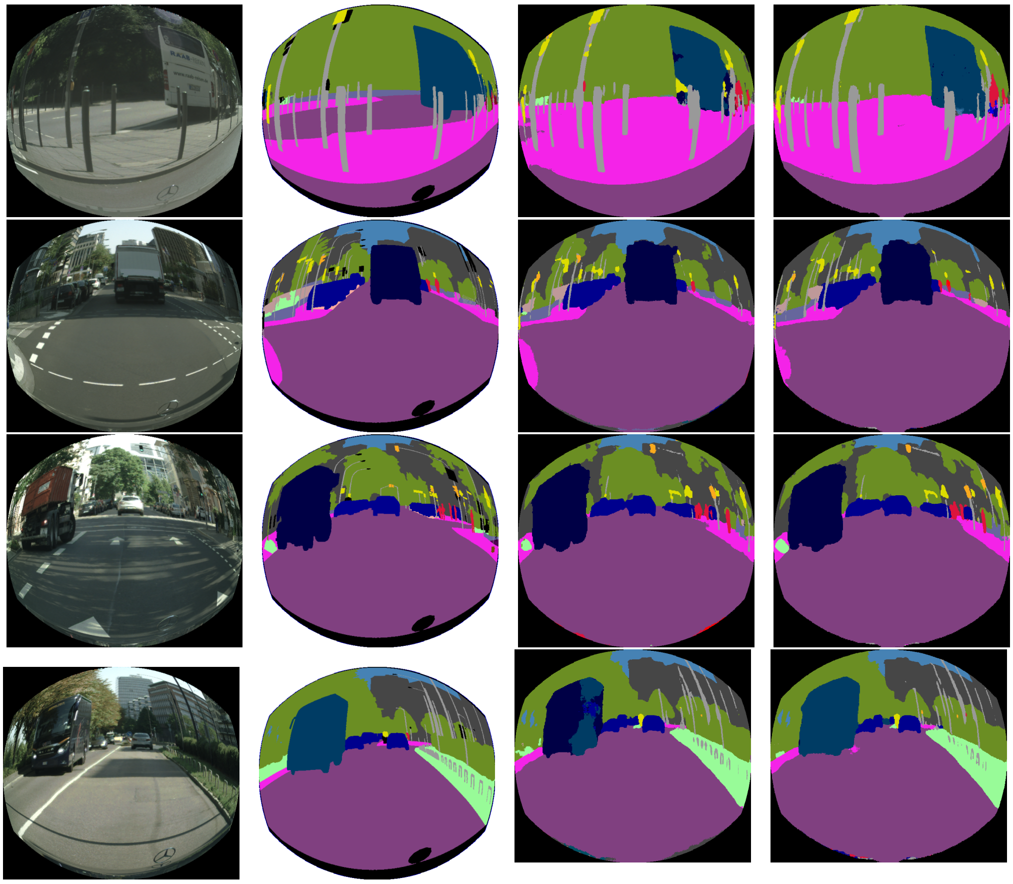

(**a**) RGB image (**b**) Ground-Truth (**c**) Segmentation with 19 classes + ROI (**d**) Segmentation with 20 classes

Figure 7. Comparison between different training strategies.

Table 4 shows detailed per-class results for the tested networks on the 640 × 576 dataset. As it can be seen, most of the best per-class scores are achieved by the ERFNet architectures. However, ERFNetPSP obtains the best results due to its outstanding performance in classes with few samples during training. Qualitative results for this table are presented in Figure 8.

Table 4. Per-class IoU (%) on the fisheye CityScapes validation set compared to similar works.

Network	**Roa**	**Sid**	**Bui**	**Wal**	**Fen**	**Pol**	**TLi**	**TSi**	**Veg**	**Ter**	**Sky**	**Ped**	**Rid**	**Car**	**Tru**	**Bus**	**Tra**	**Mot**	**Bic**	**IoU**
SegNet	96.8	65.1	79.6	25.7	19.9	29.6	29.1	36.6	84.3	42.9	89.3	58.4	29.1	85.3	37.6	48.9	17.6	25.8	49.8	50.1
DRNet	97.0	67.7	82.4	34.6	31.4	30.3	36.5	49.6	85.4	47.3	88.9	66.4	43.3	87.2	51.3	65.5	35.4	36.0	57.4	57.6
PSPNet	97.2	68.7	83.2	34.9	31.3	31.4	37.5	48.7	85.7	47.7	89.5	66.7	44.6	88.3	57.4	67.7	**45.0**	40.1	59.1	59.2
ERFNet 20	97.4	70.2	83.8	34.7	31.6	**40.7**	42.2	55.2	**87.1**	52.4	89.5	69.1	47.9	88.7	52.3	69.4	30.5	40.9	60.1	60.3
ERFNet	**97.4**	**70.8**	**83.9**	**37.2**	29.4	39.2	41.9	**55.3**	86.8	**53.2**	89.8	**69.7**	**49.4**	**89.1**	56.2	**76.1**	42.3	**41.4**	59.9	61.5
ERFNetPSP	97.3	70.1	83.2	35.9	**31.8**	37.2	**42.3**	54.7	86.3	52.3	**90.4**	68.7	48.2	88.8	**64.2**	74.2	41.6	41.1	**60.2**	**61.7**

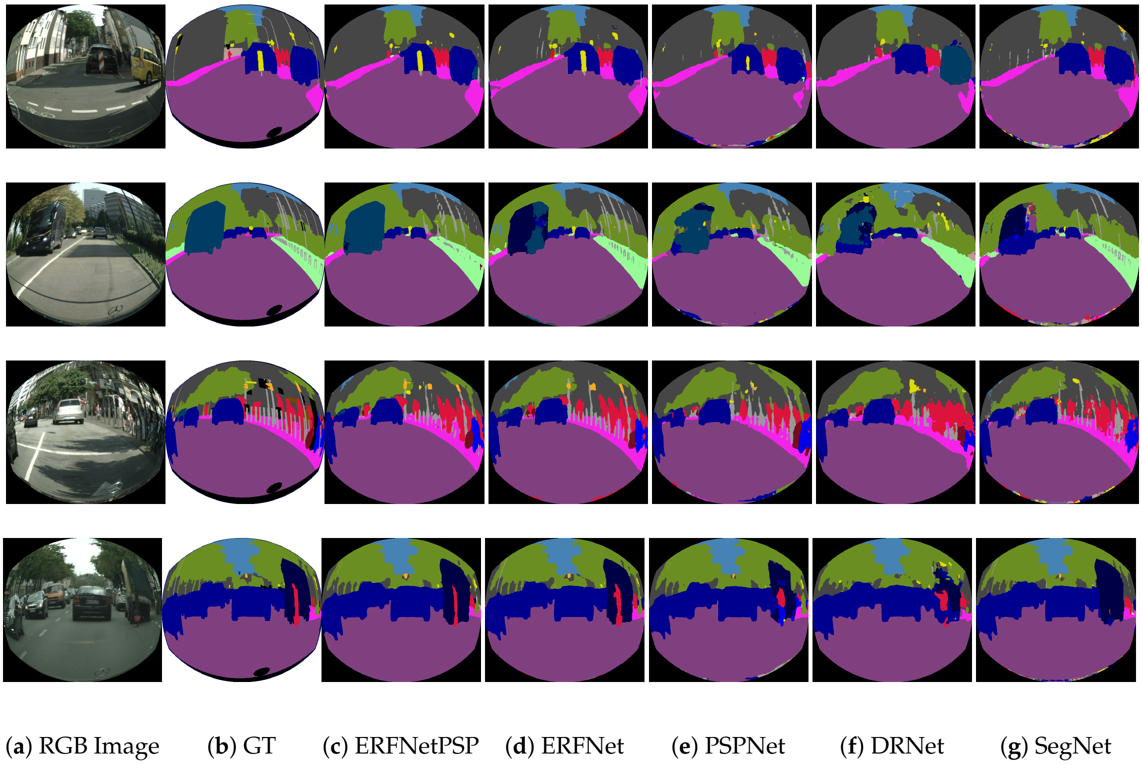


Figure 8. Qualitative results for different tested network models.

6. Application to a Real Fisheye Camera

This experiment aims to demonstrate the generalization abilities provided by the suggested architectures and training techniques based on synthetic images obtained by using distortion models over normal FOV images, applying them to the images captured by a real fisheye camera, over an urban driving scenario similar to the one used during training, but never seen before. With that purpose, a HD fisheye camera with a 180° FOV and a 1920 × 1080 resolution (USBFHD01M-BL180), manufactured by ELP, was used to record a set of sequences in the Campus of the University of Alcala (Spain) using our open-source autonomous electric car.

The previous training was not adequate for the real fisheye camera, due to the difference between resolutions and aspect-ratio of the synthetic images regarding the real ones. A new specific training adapted to the real fisheye camera was carried out, using nine new images with random distortions between $f_{inf} = 200$ and $f_{sup} = 700$, and a new one with $f_c = 500$ with resolutions of 1120 × 792, in order to preserve a similar aspect-ratio to the real fisheye camera (1536 × 1080 without borders).

Both ERFNet and ERFNetPSP architectures were trained using the new range of distortions and resolution. A quantitative validation was performed using the transformed CityScapes validation subset for these new parameters. Once again, the ERFNetPSP reaches the best performance, obtaining a mean IoU of 69.6% while the baseline ERFNet achieves an IoU of 68.3% for this validation subset.

Table 5 shows the frame-rate achieved by the introduced architectures using a single GTX 1080Ti for 1536 × 1080 images.

Table 5. Forwarding time and frame-rate for the presented architectures.

Architecture	**Forward Pass Time (s) (1536 × 1080)**	**Frames per Second**
ERFNetPSP	0.088	11.4
ERFNet [28]	0.081	**12.3**

Figure 9 analyzes the performance of the two architectures on the validation subset as a function of distortion without using any data augmentation. As we can see, ERFNetPSP achieves better performance than ERFNet for medium and high distortions. For the strongest distortions, which damage context information, the behaviour is the opposite. Performance of both architectures improve as the added distortion is reduced and becomes similar to the performance of ERFNet on the original CityScapes dataset.

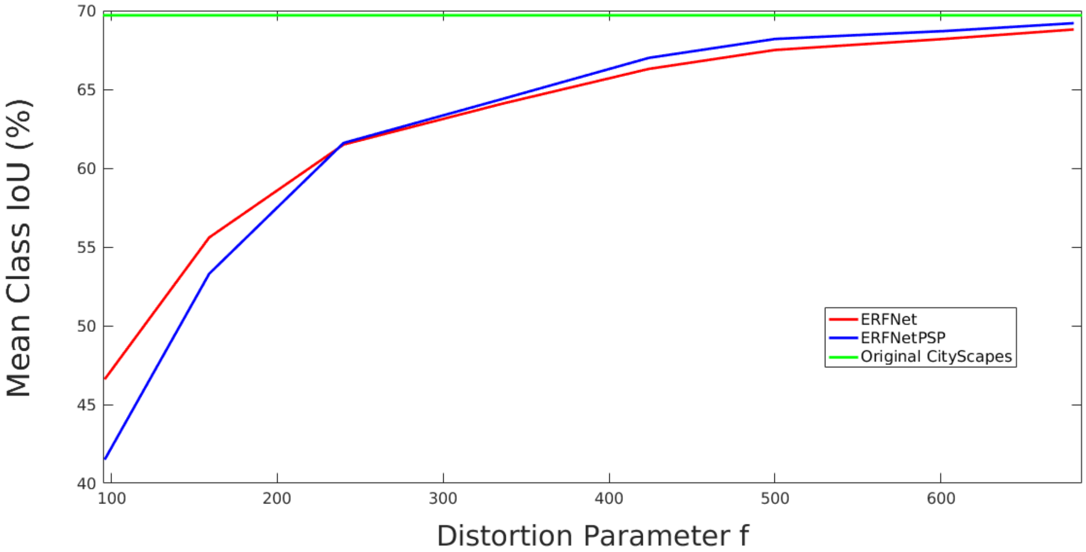


Figure 9. Mean class IoU performance vs. level of added distortion for basic training.

Figure 10 depicts a similar analysis but includes a pre-trained model on ImageNet and full data augmentation (random distortions and geometric and color transformations) in the training. As it can be seen, performance of both CNNs is clearly better, obtaining the best results with the ERFNetPSP architecture for light and medium distortions and with the baseline network for strong distortions. From the analysis of the real camera, an approximate value of 350 is estimated for the parameter f. Therefore, following the graphics, ERFNetPSP with ImageNet pre-training and full data augmentation obtains the best semantic segmentation results.

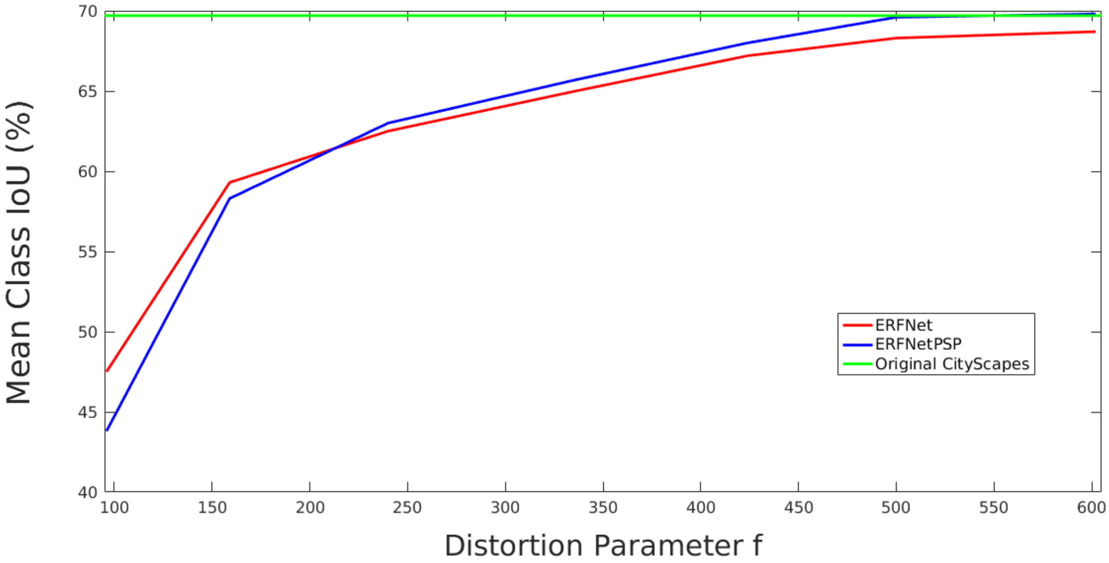
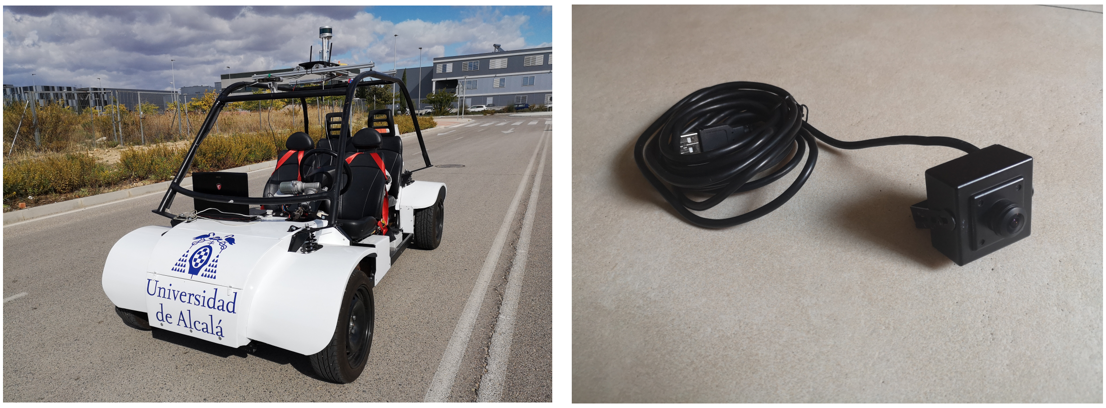


Figure 10. Results for validation subset with training with full data augmentation.

Fisheye camera was integrated in an open-source autonomous car prototype [33,34] as a complement to its main perception system, formed by a ZED camera, manufactured by StereoLabs, a VLP-16 LiDAR, manufactured by Velodyne, a HiPer Pro RTK-GPS receiver by TOPCON and odometry sensors by Kubler. Environment perception of the prototype is based on 3D semantic segmentation obtained from the fusion of LiDAR and segmented images, which is able to detect obstacles in a 3D environment [35]. Semantic segmentation for the fisheye camera runs on an embedded Jetson TX2 GPUs, manufactured by NVIDIA, and reaches 10 fps, which is the acquisition frequency of the rest of the sensors. Figure 11 shows the electric prototype and the camera used during the tests.



(**a**) SmartElderlyCar prototype (**b**) ELP 180° fisheye camera

Figure 11. Autonomous open-source electric car and fisheye camera.

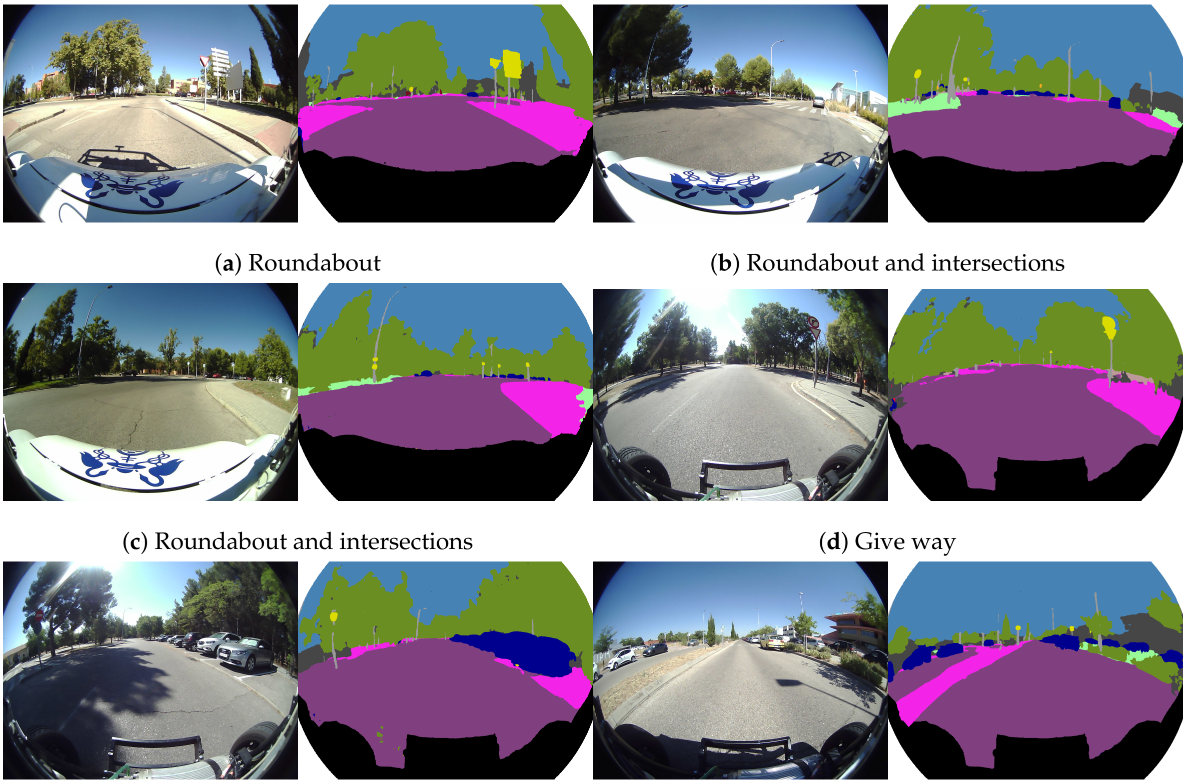
Due to the absence of annotated ground-truth, only qualitative results are exposed in this section. To provide a convincing validation, results are split focusing on the main groups of segmented classes defined in Cityscapes, and using some representative Campus images captured from the autonomous vehicle. To facilitate the understanding of the segmentation, we provide the Cityscapes color legend in Figure 12.

Void	Road	Sidewalk	Building	Wall	Fence	Pole	Traffic Light	Traffic Sign	Vegetation
Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle

Figure 12. Cityscapes color legend.

Figure 13 illustrates various complex situations focused on the "flat group", mainly composed of road and sidewalk classes, where the wider FOV of fisheye cameras clearly improves the scene comprehension about driving areas achieved with traditional cameras. Different images including roundabouts, intersections, pedestrian crosswalks and give-ways are depicted, where the road and the sidewalk classes are correctly segmented even in glare images and with obstacles, which helps to delimit the areas where the vehicle can drive in an autonomous way. The wide FOV of this camera provides more information about the lateral zones of the vehicle, which is vital in order to perform turning maneuvers in a safe way.

Figure 14 shows some representative examples for the "human group" segmentation (person and rider classes), which is very important to correctly detect vulnerable users and avoid accidents. Figure 14a shows how fisheye cameras help to handle dynamic crosswalks, where many pedestrians on the sidewalks and on the road are detected at the same time with just one camera, providing excellent scene-understanding. On the left side of Figure 14b, we can find a bicycle and a rider correctly segmented and, in Figure 14c,d, different pedestrians segmented at short and long distances, respectively.



(**a**) Roundabout

(**b**) Roundabout and intersections

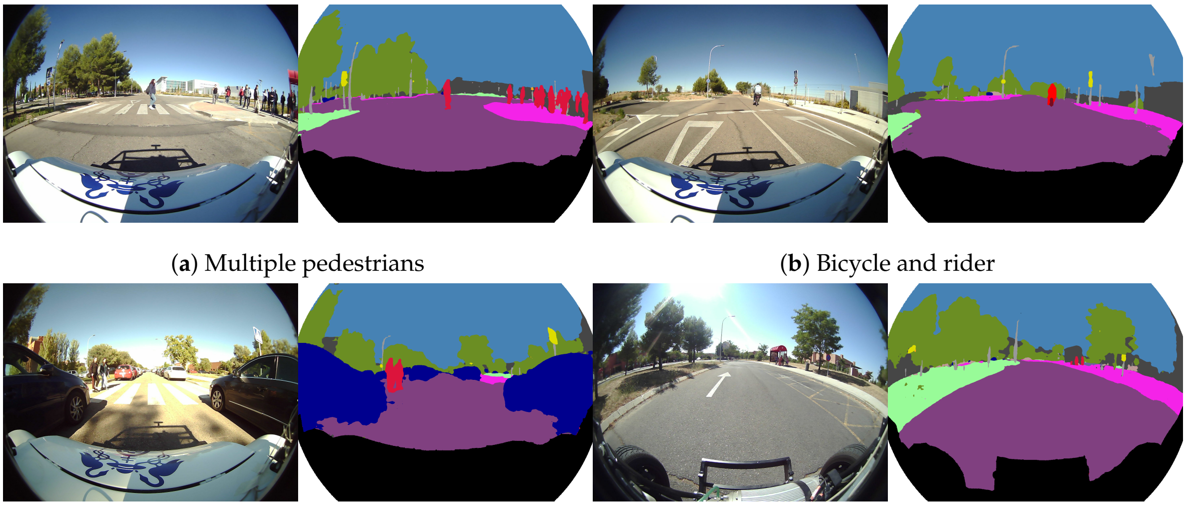
(**c**) Roundabout and intersections

(**d**) Give way

(**e**) Straight road and glares

(**f**) Straight road with median strip

Figure 13. Real fisheye camera semantic segmentation examples for flat group.



(**a**) Multiple pedestrians

(**b**) Bicycle and rider

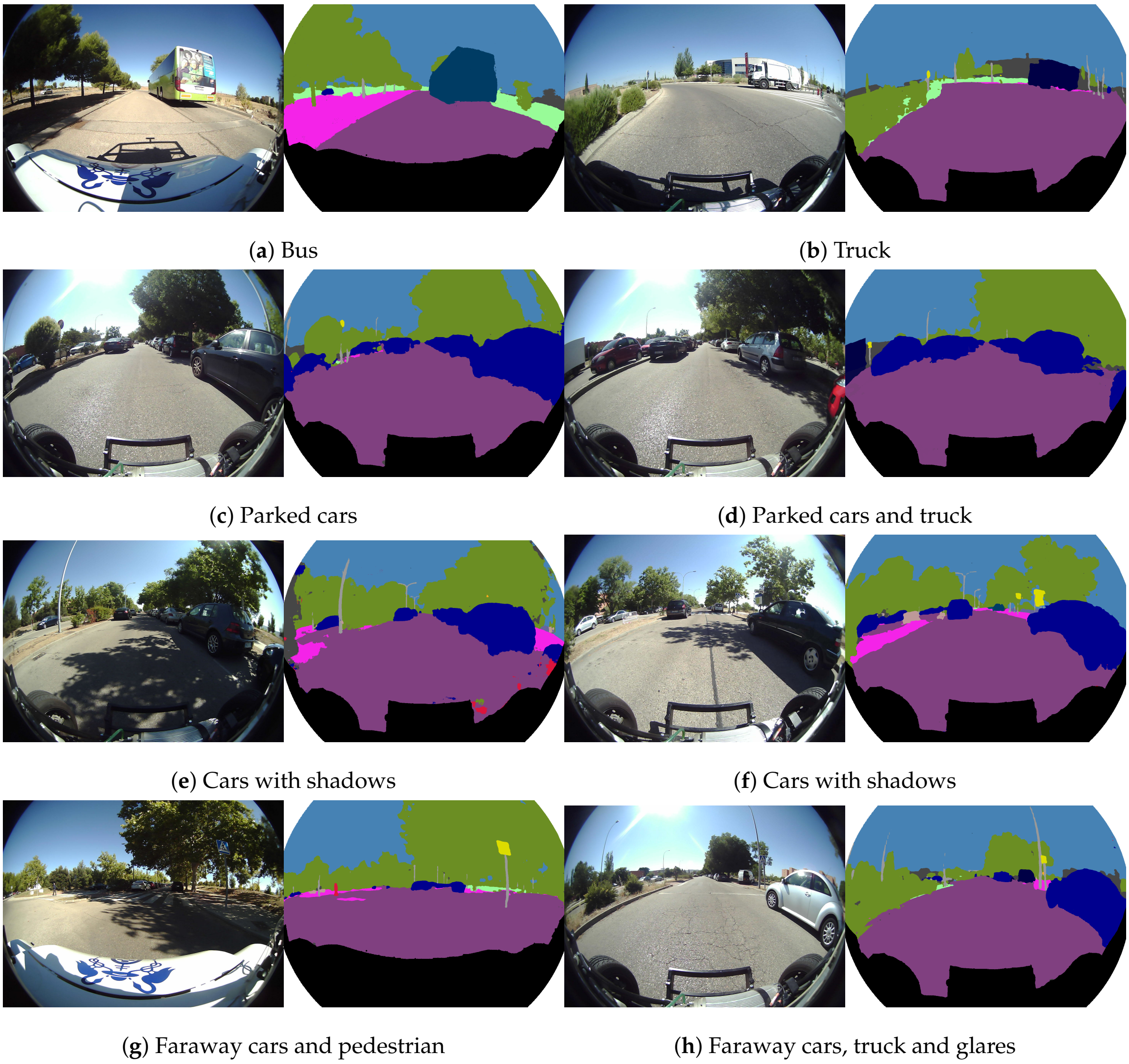
(**c**) Crossing pedestrians

(**d**) Faraway pedestrians

Figure 14. Real fisheye camera semantic segmentation examples for the human group.

Figure 15 depicts some examples for the "vehicle group" segmentation, which includes car, truck, bus, motorcycle and bicycle classes. Figure 15a shows a case of segmented bus and Figure 15b a segmented truck. Figure 15c,d illustrate the segmentation of many cars parked in both sides of the road. Figure 15e,f show cars segmented under hard shades, and Figure 15g,h a couple of cases of long distance segmented cars.

Figure 16 illustrates some segmentation cases focused on the "construction group" (building and fence classes). On the right side of Figure 16a,b, the segmentation of different fences are depicted, and Figure 16c,d show a couple of images with many segmented buildings.



(**a**) Bus

(**b**) Truck

(**c**) Parked cars

(**d**) Parked cars and truck

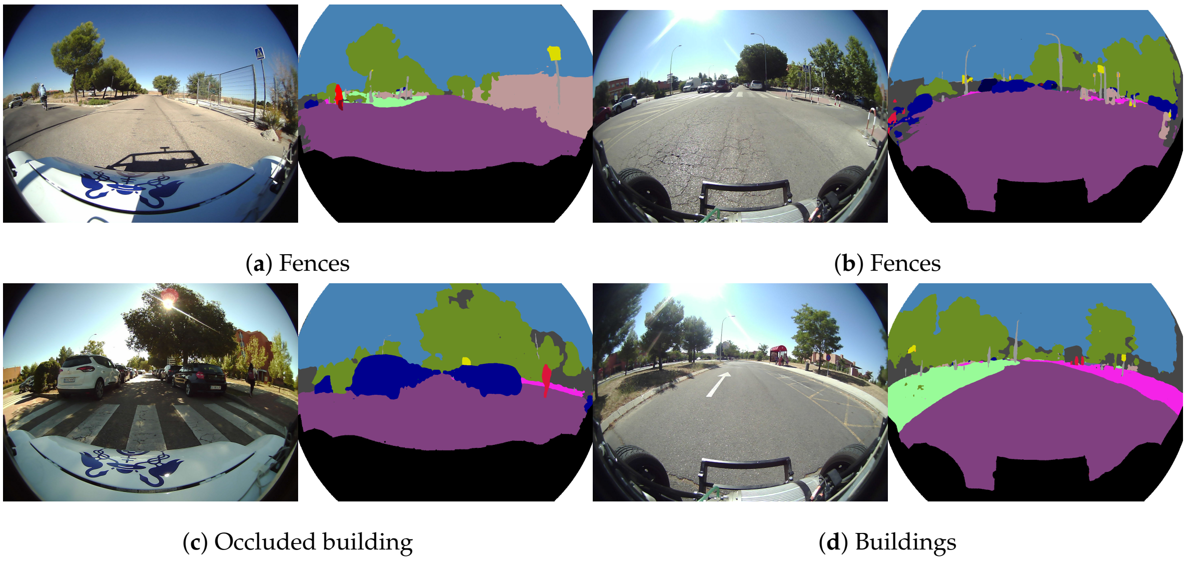
(**e**) Cars with shadows

(**f**) Cars with shadows

(**g**) Faraway cars and pedestrian

(**h**) Faraway cars, truck and glares

Figure 15. Real fisheye camera semantic segmentation examples for the vehicle group.



(**a**) Fences

(**b**) Fences

(**c**) Occluded building

(**d**) Buildings

Figure 16. Real fisheye camera semantic segmentation examples for the construction group.

The "object group" segmentation, composed of the following classes: pole, traffic light and traffic signs, is shown in Figure 17. These images show many cases of correct segmentation for pole and traffic sign classes. These objects are usually very small in the image and less frequent than other classes, which is derived from a few pieces of training data and therefore a more difficult segmentation.

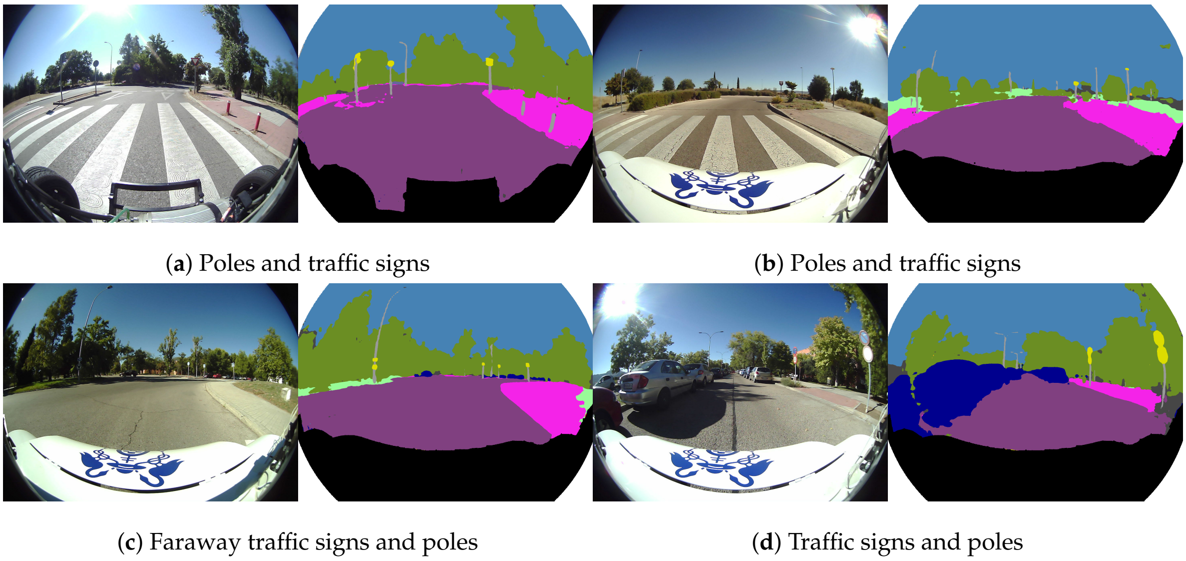

(**a**) Poles and traffic signs
(**b**) Poles and traffic signs
(**c**) Faraway traffic signs and poles
(**d**) Traffic signs and poles

Figure 17. Real fisheye camera semantic segmentation examples for the object group.

Segmentation of the "nature group", which includes vegetation and terrain classes, and the "sky group", which only contains the sky class, are well represented in all of the previous images. Their influence is secondary in autonomous vehicles' applications mainly due to the fact that they are faraway from the driving area. However, there are some cases where nature classes define the limits of the road (Figure 14d or Figure 15a) and should be taken into account.

Results demonstrate that the ERFNetPSP architecture provides real-time good quality semantic segmentation being able to detect even the classes with reduced number of training data and showing a robust behaviour to shadows and lighting changes.

Despite the good results, segmentation has still some problems dealing with glares (which are common in fisheye cameras due to its wide FOV) and with big classes with changing appearances such as the sky, as we can see in Figure 18.

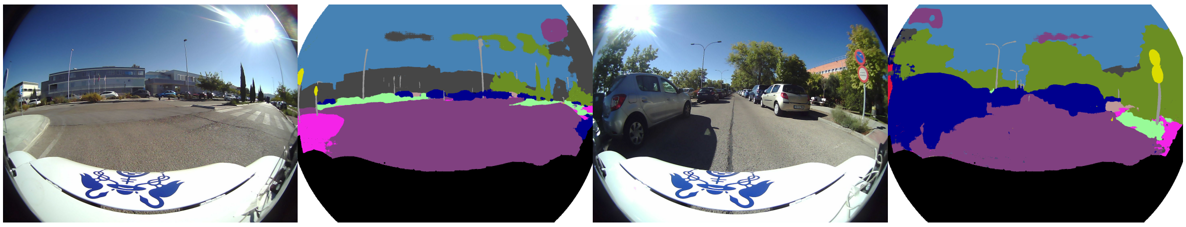


Figure 18. Problems with sky segmentation and glares.

An additional problem is that appearances of the classes present near the edges, corresponding to lateral objects located on the left/right FOV limits, are not included in the training dataset, which is captured from a conventional FOV camera. This fact degrades the obtained segmentation in the edge regions, which tend to associate small classes with more available classes such as building, road or sky.

7. Conclusions

This paper proposed a methodology to achieve real-time semantic segmentation based on ERFNet over real fisheye images, leveraging only synthetic images and, therefore, solving the lack of large-scale fisheye datasets while avoiding the heavy task of data annotation. The two introduced architectures (ERFNet and ERFNetPSP) achieve better results than the best state-of-the-art works in various synthetic datasets. Furthermore, the ability of ERFNetPSP to handle distortion by the prioritization of context information is proven, showing a better performance than ERFNet. The model also demonstrates better leveraging our data augmentation strategy, reaching the ERFNet performance in the original CityScapes dataset. Additionally, alternative training with an extra class to segment the image borders is presented. Our proposals have been validated with images taken from a real fisheye camera in unseen scenarios, showing a high capacity of domain adaptation without using a fine-tuning process with manually annotated data.

Future work involves the development of a full 360° vision system based on three fisheye cameras and the following fusion with the LiDAR sensor data, in order to develop a complete 3D surrounding perception system. In addition, we plan to research methods to improve segmentation of the small classes close to the fisheye image borders.

Author Contributions: L.M.B. and E.L.-G. conceived the methodology and coordinated the research. Á.S. and E.R. implemented the algorithms. M.T. and C.G.-H. analyzed the data. E.R. and J.d.E. contributed materials, methods and tools. Á.S. and L.M.B. wrote the paper. L.M.B., E.R. and E.L.-G. revised the paper.

Funding: This work has been funded in part from the Spanish MINECO/FEDER through the SmartElderlyCar project (TRA2015-70501-C2-1-R), the DGT through the SERMON project (SPIP2017-02305), and from the RoboCity2030-III-CM project (Robótica aplicada a la mejora de la calidad de vida de los ciudadanos. fase III; S2013/MIT-2748), funded by Programas de actividades I+D (CAM) and cofunded by the EU Structural Fund.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Romera, E.; Bergasa, L.M.; Arroyo, R. Can we unify monocular detectors for autonomous driving by using the pixel-wise semantic segmentation of CNNs? *arXiv* **2016**, arXiv:1607.00971.
2. Guo, Y.; Liu, Y.; Oerlemans, A.; Lao, S.; Wu, S.; Lew, M.S. Deep learning for visual understanding: A review. *Neurocomputing* **2016**, *187*, 27–48. [CrossRef]
3. Voulodimos, A.; Doulamis, N.; Doulamis, A.; Protopapadakis, E. Deep learning for computer vision: A brief review. *Comput. Intell. Neurosci.* **2018**, *2018*, 7068349. [CrossRef] [PubMed]
4. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv* **2016**, arXiv:1603.04467.
5. Chen, T.; Li, M.; Li, Y.; Lin, M.; Wang, N.; Wang, M.; Xiao, T.; Xu, B.; Zhang, C.; Zhang, Z. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv* **2015**, arXiv:1512.01274.
6. Neuhold, G.; Ollmann, T.; Bulò, S.R.; Kontschieder, P. The mapillary vistas dataset for semantic understanding of street scenes. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5000–5009.
7. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
8. Sáez, A.; Bergasa, L.M.; Romera, E.; Guillén, E.L.; Barea, R.; Sanz, R. Cnn-based fisheye image real-time semantic segmentation. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1039–1044.
9. Fremont, V.; Bui, M.T.; Boukerroui, D.; Letort, P. Vision-based people detection system for heavy machine applications. *Sensors* **2016**, *16*, 128. [CrossRef] [PubMed]
10. Varga, R.; Costea, A.; Florea, H.; Giosan, I.; Nedevschi, S. Super-sensor for 360-degree environment perception: Point cloud segmentation using image features. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–8.
11. Yin, X.; Wang, X.; Yu, J.; Zhang, M.; Fua, P.; Tao, D. Fisheyerecnet: A multi-context collaborative deep network for fisheye image rectification. *arXiv* **2018**, arXiv:1804.04784.

12. Deng, F.; Zhu, X.; Ren, J. Object detection on panoramic images based on deep learning. In Proceedings of the 2017 3rd International Conference on Control, Automation and Robotics (ICCAR), Nagoya, Japan, 24–26 April 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 375–380.
13. Choi, D.Y.; Choi, J.H.; Choi, J.W.; Song, B.C. CNN-based pre-processing and multi-frame-based view transformation for fisheye camera-based avm system. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 4073–4077.
14. Qian, Y.; Yang, M.; Wang, C.; Wang, B. Self-adapting part-based pedestrian detection using a fish-eye camera. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 33–38.
15. Dollár, P.; Wojek, C.; Schiele, B.; Perona, P. Pedestrian detection: A benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami, FL, USA, 20–25 June 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 304–311.
16. Felzenszwalb, P.; McAllester, D.; Ramanan, D. A discriminatively trained, multiscale, deformable part model. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008), Anchorage, AK, USA, 23–28 June 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 1–8.
17. Deng, L.; Yang, M.; Qian, Y.; Wang, C.; Wang, B. CNN based semantic segmentation for urban traffic scenes using fisheye camera. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 231–236.
18. Deng, L.; Yang, M.; Li, H.; Li, T.; Hu, B.; Wang, C. Restricted deformable convolution based road scene semantic segmentation using surround view cameras. *arXiv* **2018**, arXiv:1801.00708.
19. Ros, G.; Sellart, L.; Materzynska, J.; Vazquez, D.; Lopez, A.M. 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 3234–3243.
20. Shotton, J.; Winn, J.; Rother, C.; Criminisi, A. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In Proceedings of the European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Springer: Berlin, Germany, 2006; pp. 1–15.
21. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef] [PubMed]
22. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 764–773.
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin, Germany, 2014; pp. 346–361.
24. Romera, E.; Bergasa, L.M.; Alvarez, J.M.; Trivedi, M. Train here, deploy there: Robust segmentation in unseen domains. In Proceedings of the IEEE Conference on Intelligent Vehicles Symposium, Changshu, China, 26–30 June 2018.
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
26. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. In Proceedings of the International Conference on Learning Representations (ICLR), San Juan, PR, USA, 2–4 May 2016.
27. Treml, M.; Arjona-Medina, J.; Unterthiner, T.; Durgesh, R.; Friedmann, F.; Schuberth, P.; Mayr, A.; Heusel, M.; Hofmarcher, M.; Widrich, M.; et al. Speeding up semantic segmentation for autonomous driving. In Proceedings of the MLITS, NIPS Workshop, Barcelona, Spain, 9 December 2016.
28. Romera, E.; Alvarez, J.M.; Bergasa, L.M.; Arroyo, R. Efficient convnet for real-time semantic segmentation. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1789–1794.
29. Paszke, A.; Chaurasia, A.; Kim, S.; Culurciello, E. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv* **2016**, arXiv:1606.02147.
30. Badrinarayanan, V.; Handa, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv* **2015**, arXiv:1505.07293.

31. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
32. Yu, F.; Koltun, V.; Funkhouser, T. Dilated residual networks. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
33. Tradacete, M.; Sáez, Á.; Arango, J.F.; Huélamo, C.G.; Revenga, P.; Barea, R.; López-Guillén, E.; Bergasa, L.M. Positioning system for an electric autonomous vehicle based on the fusion of multi-gnss rtk and odometry by using an extented kalman filter. In Proceedings of the Workshop of Physical Agents, Madrid, Spain, 22–23 November 2018; Springer: Berlin, Germany, 2018; pp. 16–30.
34. Murciego, E.; Huélamo, C.G.; Barea, R.; Bergasa, L.M.; Romera, E.; Arango, J.F.; Tradacete, M.; Sáez, Á. Topological road mapping for autonomous driving applications. In Proceedings of the Workshop of Physical Agents, Madrid, Spain, 22–23 November 2018; Springer: Berlin, Germany, 2018.
35. Barea, R.; Pérez, C.; Bergasa, L.M.; López-Guillén, E.; Romera, E.; Molinos, E.; Ocana, M.; López, J. Vehicle detection and localization using 3d lidar point cloud and image semantic segmentation. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 3481–3486.