2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

CNN-based Fisheye Image Real-Time Semantic Segmentation

Álvaro Sáez[1], Luis M. Bergasa[1], Eduardo Romera[1], Elena López[1], Rafael Barea[1] and Rafael Sanz[2]

Abstract— **Semantic segmentation based on Convolutional Neural Networks (CNNs) has been proven as an efficient way of facing scene understanding for autonomous driving applications. Traditionally, environment information is acquired using narrow-angle pin-hole cameras, but autonomous vehicles need wider field of view to perceive the complex surrounding, especially in urban traffic scenes. Fisheye cameras have begun to play an increasingly role to cover this need. This paper presents a real-time CNN-based semantic segmentation solution for urban traffic images using fisheye cameras. We adapt our Efficient Residual Factorized CNN (ERFNet) architecture to handle distorted fish-eye images. A new fisheye image dataset for semantic segmentation from the existing CityScapes dataset is generated to train and evaluate our CNN. We also test a data augmentation suggestion for fisheye image proposed in [1]. Experiments show outstanding results of our proposal regarding other methods of the state of the art.**

I. INTRODUCTION

Understanding the surrounding environment is an essential task for autonomous vehicles. Semantic segmentation aims to solve this problem by parsing images into different regions with different semantic categories such as pedestrians, road, buildings, traffic signals, etc. at a pixel level. This provides relevant information that covers most of the needs of autonomous vehicles in an unified way [2].

Semantic segmentation solution based on Convolutional Neural Networks (CNNs) standout over other state-of-the-art solutions, as they can be trained end-to-end to accurately classify multiple object categories in an image at the pixel level. The success of this technique is due to the existence of large-scale training datasets [3], [4], high performance Graphics Processing Units (GPUs) and excellence open source deep learning frameworks [5], [6], [7].

Urban traffic scenes can be very complex, with unpredictable behaviors of dynamic traffic participants and challenging situations, especially at roundabouts and intersections. Therefore, a complete and real-time surrounding perception is mandatory. Multiple sensors can be used to face the problem, including ultrasound, radar, LIDAR, cameras, etc. Among then, cameras are a good solution because they offer deep information, are cheap and easy to handle. The problem is that standard cameras have a limited field of view, demanding many of them to cover the whole surrounding area. An alternative is to perceive wide-angle views for semantic segmentation by using fisheye cameras, which are able to provide the entire frontal hemispheric view of 180º. Exploiting this option, only two cameras would be theoretically needed to cover the 360º. However, fisheye cameras introduce strong distortion on the images, so elements on the scene appear warped on them and standard algorithms cannot be applied for image segmentation tasks.

Several solutions have been presented to deal with this challenge that also influences other vision tasks such as classification or detection. Initial approaches tried to handle the problem by un-warping fisheye original images and then applying standard algorithms to the undistorted image like Local Binary Pattern (LBP) [8] or DPM [9]. Other approaches reprojected the fish-eye image using pinhole camera models to correct the distortion [10] [11].

The previously proposed methods showed a good performance in different tasks, but they also present a strong dependency on the intrinsic camera calibration parameters. Even with a good knowledge of the used set of parameters, the un-warping process usually hurts the image quality, and the obtained image may present some differences with the original that negatively impacts subsequent processes. They also force the designing of complex preprocessing stages that increase the response time of the developed systems and make them non-viable for real-time applications. Summarizing, learning based methods trained in conventional images are difficult to be applied over undistorted images.

Due to this difficulties, latest works have been focused on adapting existing image processing techniques to handle the uncorrected images directly. These proposals must tackle three important problems: 1) how to manage the strong distortion in the fisheye images, 2) the lack of large-scale dataset with pixel-level annotated images and 3) computational resources needed to implement perception systems for real-time applications. In this sense, latest state-of-the-art methods have tried to generate new artificial datasets using existing ones adding fisheye distortion on them forwards to adapt neuronal networks to information provided by wide-angle field of view devices. In [12] an spherical perspective imaging model is used to generate a fisheye dataset based on the ETH pedestrian benchmark. In [1] authors used the perspective projection equation of fisheye cameras to remap the pixels from the original dataset images to the new distorted ones.

*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 EU Structural Fund.

[1]Álvaro Sáez, Luis M. Bergasa, Eduardo Romera, Elena López and Rafael Barea are with the Department of Electronics, University of Alcalá (UAH), Alcalá de Henares, Madrid, Spain alvaro.saezc@edu.uah.com, luism.bergasa@uah.es, eduardo.romera@edu.uah.es, elena.lopez@uah.es, rafael.barea@uah.es

[2]Rafael Sanz is with the Department of Systems Engineering and Automation, University of Vigo, Pontevedra, Spain rsanz@uvigo.es

This paper presents an efficient CNN-based semantic segmentation proposal for urban traffic images using fisheye cameras on board a real vehicle. A new fisheye image dataset for semantic segmentation based on CityScapes dataset [3] is proposed in Section II. In Section III a new data augmentation strategy for our fisheye images is tested to evaluate its generalization performance. An Efficient Residual Factorized CNN for real-time semantic segmentation (ERFNet) is proposed in Section IV. Finally, experimental results that validate our proposal are presented in Section V and conclusions in Section VI.

II. Fisheye Image Dataset

In conventional pinhole model cameras light directly maps into the image as in equation (1):

$$\rho_{pinhole} = f tan(\theta) \tag{1}$$

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

However, fisheye camera imaging model is different: fisheye lenses can be designed following various mathematic models. Among them, the most generic one is the equidistance projection, as in (2):

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

Despite this, there are other less used models such as stereographic projection (3), orthogonal projection (4) or equisolid angle projection (5):

$$\rho_{stereographic} = 2f tan(\theta/2) \tag{3}$$

$$\rho_{orthogonal} = f sin(\theta) \tag{4}$$

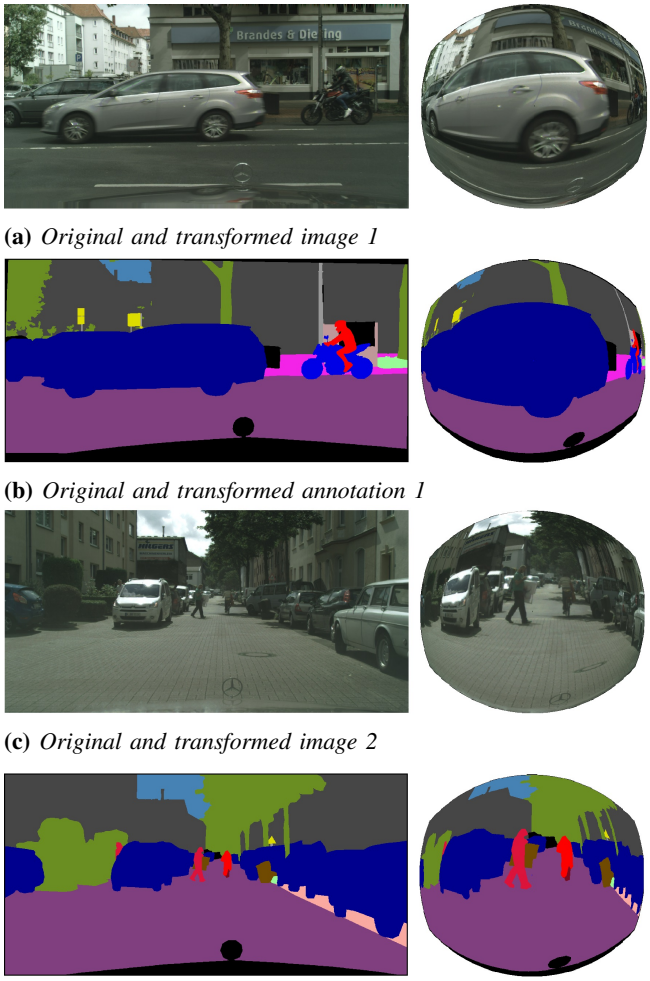
$$\rho_{equisolid} = 2f sin(\theta/2) \tag{5}$$

Any of the previous equations together with (1) can be used to define a remapping between an original conventional image and a new synthetic fisheye image that will only depend on a focal length, as proposed in [1]. The final remapping relationship will link the distance between a single pixel ($P_c = (x_c, y_c)$) and the principal point ($U_c = (u_{cx}, u_{cy})$) on the conventional image (d_c) with its equivalent distance between the single pixel ($P_f = (x_f, y_f)$) and the principal point ($U_f = (u_{fx}, u_{fy})$) on the new fisheye image (d_f). This relation for the equidistance projection is described by:

$$d_c = f tan(d_f/f) \tag{6}$$

With $d_c = \sqrt{(x_c - u_{cx})^2 + (y_c - u_{cy})^2}$ for the conventional image, and $d_f = \sqrt{(x_f - u_{fx})^2 + (y_f - u_{fy})^2}$ for the fisheye image.

Using the previous equation, we generated a new set of images from CityScapes dataset, as showed on Fig.1. CityScapes is a large-scale dataset for semantic urban scene understanding that contains 5,000 dense pixel-level annotated images selected from 27 cities with 19 classes for evaluation. The images are split into three subsets: 2,975 for training, 500 for validation and 1,525 for test. The training and validation sets (RGB images and annotated images) are transformed to fisheye images following equation (6). The corresponding annotated images are not available for the test set. To generate the complete dataset, we used the validation set for testing.



(a) *Original and transformed image 1*

(b) *Original and transformed annotation 1*

(c) *Original and transformed image 2*

(d) *Original and transformed annotation 2*

Fig. 1: Examples of transformation from original CityScapes images and annotations to fisheye ones

The remapping process implies a big scale loss on the images. In order to deal with this and also to adapt the final images to our ConvNet we scale images after the remapping to 576x640 resolution. We transform the entire training and validation sets, using bilinear interpolation for original images and nearest-neighbor for label images. For the remapping process we initially use an arbitrary focal length of $f_0 = 159$ as in [1], but additionally a new data augmentation strategy is tested.

III. Data Augmentation Proposal

Data augmentation is used to enlarge the training data using label-preserving transformations. There are many techniques typically applied in semantic segmentation such as flipping, rotation, scaling, cropping and color jittering. Au-

(a) Original images **(b)** f_1 = 96 **(c)** f_0 = 159 **(d)** f_2 = 242

Fig. 2: Original images and remapping for different focal length values

thors in [1] proposed a new data augmentation method specially designed for fisheye image and called zoom augmentation. They augmented training dataset with additional images by changing the focal length of the fisheye camera with two empirically calculated values regarding the baseline, a smaller one ($f_1 = 96$) and a bigger one ($f_2 = 242$).

For comparison reasons we reproduce the same values for the focal length and results are shown in Fig.2. As a general conclusion, higher scales introduce lower distortion and smaller scales higher distortion. In the experimental results section semantic segmentation performance will be analyzed according to this parameter.

We implement a variation regarding the explained zoom augmentation technique consisting in randomly changing, following a Gaussian distribution, the focal length between the two defined values ($f_1 = 96$, $f_2 = 242$). Theoretically, this is the best strategy to obtain an optimal zoom augmentation in a range if the number of samples is representative.

IV. CNN Architecture

Last trends in top-performance network designing have led to the development of large deep architectures for networks pushed by the appearance of residual layers that avoid the degradation problem allowing the gradient to be propagated through a big number of layers, making possible very deep networks with hundreds of layers. However, very large networks are not adequate for real-time applications and they have even been proven inefficient in certain works focused in the image classification [13] [14] and semantic segmentation tasks [15], [16].

Different trends have tried to achieve more efficient networks by aggressively reducing the number of network parameters. This works led to real-time working architectures but with a poor accuracy performance.

Our ERFNet [17] proposal presents a ”wider” (as opposed to “deeper”) architecture while still took advantage on residual convolutional layers but with a different novel approach, leading to an extremely efficient model with real-time performance in computationally heavy tasks like semantic segmentation.

Residual layers were originally proposed in [18] with two possible designs: bottleneck and non-bottleneck. Bottleneck layers are more extended due to efficiency reasons, but non-bottleneck has demonstrated performance benefits in certain shallow architectures like ResNet. The proposed architecture is built by stacking layers based on a novel redesign of the non-bottleneck residual layer. To keep this efficiency-performance trade-off, the ERFNet repurposes the non-bottleneck design to be entirely built with convolutions with factorized (1D) kernels in order to reduce computation.

The model follows an encoder-decoder architecture like SegNet [19] and ENet [20], avoiding the need of using skip layers to refine the output, as shown on Fig.3. The encoder block is formed by a total number of 16 layers, including both downsampling and the redesigned non-bottleneck convolutional layers. The decoder block consists of 7 layers that perform the deconvolution (upsampling) of the feature maps to the original input image size and a final log-softmax loss layer that provides class probabilities. The model also includes Batch-Normalization to accelerate convergence, and dropout with a probability of 0.3 trying to avoid overfitting and as a measure of regularization. This sequential architecture has the ability of handling complex fisheye images by using a simple downsampled-upsampled feature maps process.

V. Experimental Results

To validate our proposal we plan two kind of experiments. The first one is focused on the data augmentation strategy and the second one in comparing results with other similar proposals of the state of the art.

We use the Cityscapes dataset according to the explained in Section II. We train our models on the training set uniquely, without using the validation set. All accuracy results are reported using the commonly adopted Intersection-over-Union (IoU) metric. Our model is trained using the deep

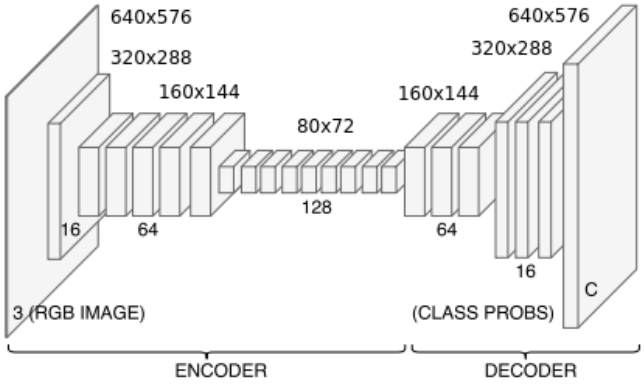


Fig. 3: Diagram that depicts the proposed segmentation net (ERFNet). Volumes correspond to the feature maps produced by each layer. All spatial resolution values are with regard to an example input (640x576).

learning framework Pytorch which stands out in efficiency and speed terms. All of our code is open source and it is available at *https:github.com/Eromeraerfnet_pytorch*.

Training is performed using a batch size of 6, 180 epochs and the Adam optimization of Stochastic Gradient Descent with a starting learning rate of 5e-4 and a weights decay parameter of 1e-4, adjusting the learning rate on each epoch. The class weighing technique proposed in [20] is used during the training $w_{class} = \frac{1}{ln(c+p_{class})}$ setting c = 10, which empirically gave good results in our case.

The training process takes place in two phases: on a first stage the encoder block is trained during 90 epochs with downsampled annotations and then, on a second phase, during another 90 epochs the decoder is included as well to train the network end-to-end and generate segmented images with the same resolution that the input images.

A. Data Augmentation

To analyze the benefits of our data augmentation proposal, we train our ERFNet using distorted images and performing simple data augmentation by doing random horizontal flips and translations of 0-2 pixels in both axes. Rotation and scale augmentation are not used because, according to [1], do not give additional improvements for these images. In addition, zoom augmentation for the three different focal lengths ($f_1 = 96$, $f_0 = 159$, $f_2 = 242$) are carried out. As baseline, we include the original ERFNet pre-trained on the Imagenet dataset and trained on CityScapes by using non-distorted images, to demonstrate that standard training is not suitable for this new challenge (fish eye images) due to their strong distortion. Finally, zoom augmentation with random focal length is evaluated.

Once we get the fifth trained models, we test them on the validation set corresponding to the same focal length used for the training. For the original ERFNet we use the test set with the a priori most favorable focal length and for the random zoom augmentation we use the validation set for the central focal length ($f_0 = 159$).

As shown on Table I, the original ERFNet gets the worse results when fisheye distortion is added. It obtains a 38,2 % IoU for the validation set with the most favorable distortion level ($f_2 = 242$). Experiments also demonstrate that stronger distortions over the fisheye images degrade the segmentation performance. Obtained IoU average varies from 46,6 % to 60,2 % respectively from the weakest distortion level (f_2) to the strongest one ($f_1 = 96$), as we depict on Table I. Stronger distortions imply higher variations on the appearance of the objects in the scene, which forces the CNN to learn better and more generalizable features. Finally, results for our random zoom augmentation proposal are close the most favorable case (f_2), but using the validation set for (f_0), which shows the generalization capability of this technique.

B. Comparison to the State of the Art

We compare our trained model with others used on similar experiments. Our network shows better performance than one of the best state-of-art method consisting on the Overlapping Pyramid Pooling Net (OPP-Net) [1] trained with and without the additional data augmentation technique named zoom augmentation. We use the same process during the training and the test for the two networks, consisting in fusing the data obtained for the three focal length in the training and validate with the (f_0) set. Results are shown in Table II. The OPP-Net without zoom augmentation achieves an IoU of 52,6%, improving until the 55,6% for the ERFNet. Adding zoom augmentation improves previous results, achieving 54,5% for the OPPNet+AUG and 57,0% for the ERFNet+AUG, with an increase of 1,4%. Using some empirically calculated distortions in the zoom augmentation strategy (AUG) helps to obtain more generalizable features. Choosing random values for the focal length (ERFNet + rnd AUG) reaches similar results and does not improve AUG performance. Finally, the ERFNet obtains its best score with an IoU of 59,3% starting from a pre-trained Imagenet model and using additional data augmentation (AUG2) that includes adding random cropping, color jittering and randomly modifying the aspect ratio of the images as used in [21]

As an additional advantage, our network is able to run at more than 45 frames per second (fps) on an unique Titan X (approximately 0,022 s per image), achieving a clearly real-time data processing capability even at embedded systems like Jetson TX2 ($>$ 15 fps).

In a final set of experiments, we focus on enhancing our results by modifying the initial architecture of our network. We substitute the original decoder of our ConvNet with the pyramidal module of the PSPNet [22] that, presumably, keeps better the contextual information of the scene by combining data from different sub-regions of the image. The sub-region pyramid pooling module uses a global pooling layer and three finer non-overlapping pooling layer with four different bin sizes. We performed training with the basic dataset (ERFNet PSP) and adding the previously proposed data augmentation (ERFNet PSP + AUG).

As it can be seen in Table II, the modified network obtains slightly worse results for both with and without

TABLE I: Per-class IoU (%) on the fisheye CityScapes validation set for different focal lengths

Network	Roa	Sid	Bui	Wal	Fen	Pol	TLi	TSi	Veg	Ter	Sky	Ped	Rid	Car	Tru	Bus	Tra	Mot	Bic	IoU
Original ERFNet	84.1	51.2	60.6	19.1	8.55	20.0	19.4	32.1	77.8	36.3	85.6	43.2	29.7	63.6	33.5	4.2	0.9	21.9	35.0	38.2
ERFNet $f_1 = 96$	95.6	56.9	73.2	20.7	16.8	27.5	23.0	40.8	78.8	36.1	82.6	60.4	37.3	83.5	24.6	49.9	6.9	20.9	49.7	46.6
ERFNet	96.8	65.7	79.3	28.8	21.3	35.1	32.8	48.3	84.6	45.6	87.9	67.9	44.3	87.4	42.8	66.1	27.6	38.4	38.4	55.6
ERFNet $f_2 = 242$	97.4	70.4	83.8	28.3	38.0	38.9	39.6	56.2	86.8	48.7	89.4	71.0	49.2	89.0	51.2	63.4	39.1	42.8	60.0	60.2
ERFNet + random f	96.9	67.6	80.7	31.1	21.9	36.2	37.4	49.1	84.7	46.4	88.5	68.5	48.8	87.7	42.7	6775	24.9	41.5	58.4	56.8

TABLE II: 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
OPPNet	96.5	61.4	78.4	23.7	22.8	24.6	28.4	41.1	82.5	39.1	87.2	63.3	34.2	85.8	40.2	56.7	39.2	41.2	53.9	52.6
ERFNet	96.8	65.7	79.3	28.8	21.3	35.1	32.8	48.3	84.6	45.6	87.9	67.9	44.3	87.4	42.8	66.1	27.6	38.4	38.4	55.6
OPPNet + AUG	96.7	63.5	79.6	26.9	25.4	25.6	30.6	44.0	83.2	43.0	88.8	65.7	39.4	86.7	48.6	55.3	37.2	40.1	55.4	54.5
ERFNet + AUG	96.9	66.8	80.3	34.4	23.8	36.3	36.2	50.1	85.0	47.9	87.3	69.0	47.6	87.7	47.6	64.6	22.9	41.8	57.0	57.0
ERFNet + rnd AUG	96.9	67.6	80.7	31.1	21.9	36.2	37.4	49.1	84.7	46.4	88.5	68.5	48.8	87.7	42.7	6775	24.9	41.5	58.4	56.8
ERFNet PSP	96.6	64.5	77.4	33.1	22.9	28.5	30.2	44.4	82.2	45.9	85.7	62.2	41.9	86.0	42.6	58.8	33.2	21.9	54.4	53.3
ERFNet PSP + AUG	96.8	64.8	79.4	32.7	25.5	31.2	34.1	46.4	83.8	46.1	87.8	67.8	45.5	87.5	50.0	65.1	23.5	38.1	56.2	55.9
Pretrained ERFNet + AUG2	**97.1**	**67.6**	**81.5**	**35.0**	**26.3**	**37.3**	**38.8**	**52.4**	**85.0**	**48.1**	**88.9**	**69.8**	**50.2**	**89.0**	**56.6**	**71.5**	**24.9**	**45.2**	**60.5**	**59.3**

data augmentation cases. Nevertheless, this network achieves better scores on classes with few training data. Also, data augmentation works better for this case achieving a 2,6% IoU improvement.

Qualitative results shown on Fig.4 prove that the specific fisheye training designed for the network noticeably improves the segmentation of the objects placed on the borders of the image, which have the higher distortion level. Some pedestrians, vehicles, traffic signals and riders are ignored by the original network due to strong changes on their shape, but are precisely detected by the modified network. Segmentation on the central region slightly gets better as well, but demonstrates a minor improvement. These facts evidence that the ERFNet correctly learns different features for the same class with different distortion levels depending on the area of the image where the class appears. Besides, zoom augmentation markedly improves segmentation results, adding a higher generalization capability, and random zoom augmentation works in a similar way. White region on the ground-truth and distorted images is an empty area consequence of the fisheye remapping process. All the pixels associated with that region are ignored during the classifying process. In the evaluated images, the region appears as wrongly classified, as shown on Fig.4. Real fisheye cameras also present this region due to their projective model. Characterizing this area just by adding a simple region of interest will be needed on a real application in order to avoid wrong conclusions.

VI. Conclusions

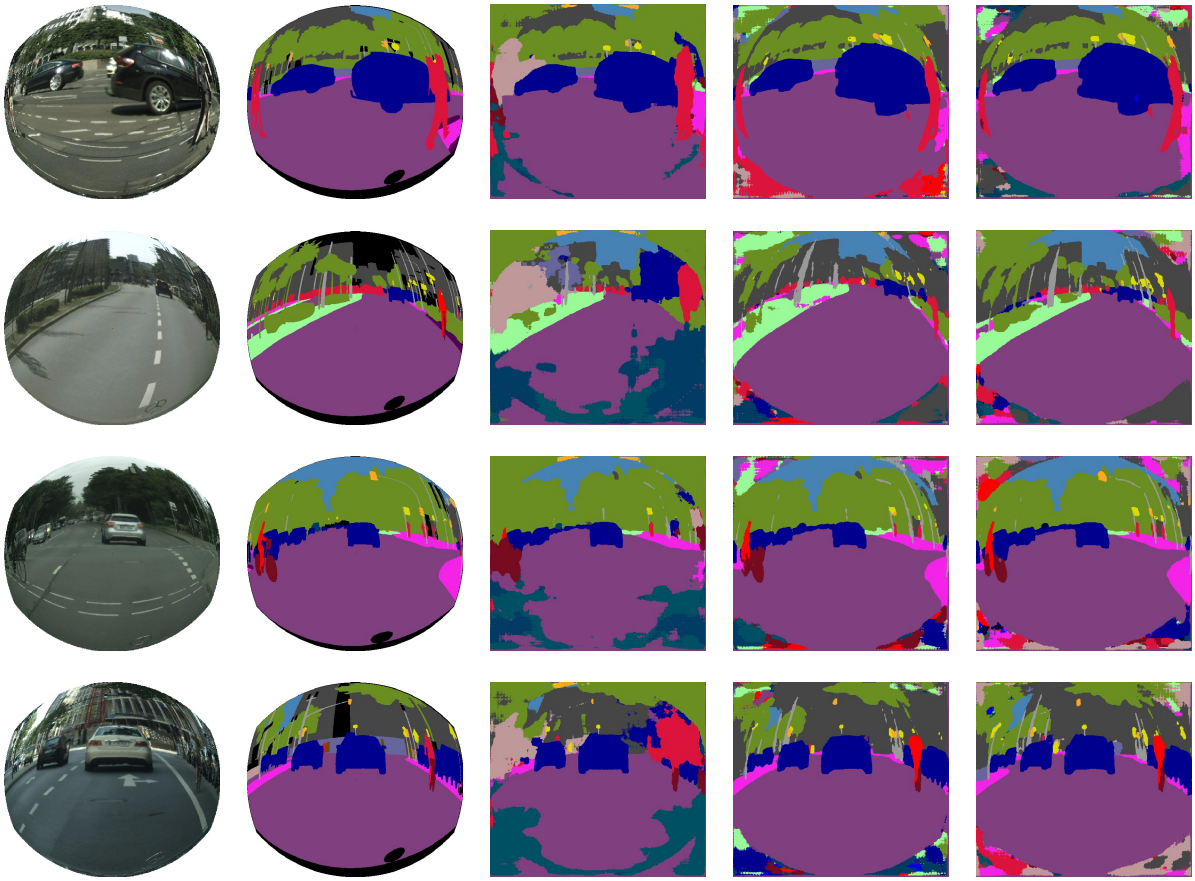
This paper proposes a real time CNN-based image semantic segmentation solution adapted to fisheye cameras for urban traffic images. The ERFNet has shown better performance than the OPP-Net, which uses a pyramidal pooling module, in the exploration of context information in the images. In addition, the original decoder of our CNN gives better results than using the pyramidal pooling module of the PSPNet, especially designed to have a good contextual information of the image by combining data from different regions. As conclusion, the sequential architecture based on an encoder block producing downsampled feature maps and a subsequent decoder that upsamples feature maps to match the input resolution, gives better results that using ad-hoc pyramidal pooling strategies for fisheye images for the ERFNet. Besides, our proposal is the only one able to run in real-time. To solve the lack of large-scale training dataset, a new fisheye image dataset for semantic segmentation is generated from CityScapes. Finally, a proposal for the data augmentation strategy presented in [1] and based on random changes of focal length for zoom augmentation is tested showing that does not improve the zoom augmentation strategy.

Our final goal is to install fisheye cameras in our autonomous vehicle in order to take advantage of their wider field of view. The perception system should provide real-time semantic segmentation of the car surrounding to complement other sensors as a LIDAR, a stereo camera and a GPS. To do that, we plan to apply our model with the images taken from our own perception system, using additional training and data augmentation processes to fine tune the model to our environment.

References

[1] L. Deng, M. Yang, Y. Qian, C. Wang, and B. Wang, "Cnn based semantic segmentation for urban traffic scenes using fisheye camera," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*, pp. 231–236, IEEE, 2017.

[2] E. Romera, L. M. Bergasa, and R. Arroyo, "Can we unify monocular detectors for autonomous driving by using the pixel-wise semantic segmentation of cnns?," *arXiv preprint arXiv:1607.00971*, 2016.



(a) Ground Truth **(b)** Distorted Image **(c)** Original ERFNet **(d)** ERFNet + AUG **(e)** ERFNet+AUG2

Fig. 4: Qualitative results for different tested network models

[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*, pp. 3213–3223, 2016.

[4] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *European conference on computer vision*, pp. 44–57, Springer, 2008.

[5] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 675–678, ACM, 2014.

[6] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[7] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, no. EPFL-CONF-192376, 2011.

[8] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[9] E. Hsiao, P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," 2009.

[10] A. Broggi, E. Cardarelli, S. Cattani, P. Medici, and M. Sabbatelli, "Vehicle detection for autonomous parking using a soft-cascade adaboost classifier," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pp. 912–917, IEEE, 2014.

[11] S. Silberstein, D. Levi, V. Kogan, and R. Gazit, "Vision-based pedestrian detection for rear-view cameras," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pp. 853–860, IEEE, 2014.

[12] Y. Qian, M. Yang, C. Wang, and B. Wang, "Self-adapting part-based pedestrian detection using a fish-eye camera," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*, pp. 33–38, IEEE, 2017.

[13] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[14] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European Conference on Computer Vision*, pp. 525–542, Springer, 2016.

[15] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *arXiv preprint arXiv:1606.00915*, 2016.

[16] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe, "Full-resolution residual networks for semantic segmentation in street scenes," *arXiv preprint arXiv:1611.08323*, 2016.

[17] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Efficient convnet for real-time semantic segmentation," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*, pp. 1789–1794, IEEE, 2017.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[19] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *arXiv preprint arXiv:1505.07293*, 2015.

[20] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147*, 2016.

[21] J. M. A. Eduardo Romera, Luis M. Bergasa and M. Trivedi, "Train here, deploy there: Robust segmentation in unseen domains," in *Proceedings of the IEEE conference on Intelligent Vehicles Symposium*, p. to appear, IEEE ITS, 2018.

[22] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *arXiv preprint arXiv:1612.01105*, 2016.