Two Stage Semantic Segmentation by SEEDS and Fork Net

Conference Paper · February 2020

DOI: 10.1109/CALCON49167.2020.9106468

4 authors:

Aritra Mukherjee
Birla Institute of Technology and Science - Hyderabad Campus
17 PUBLICATIONS **59** CITATIONS
SEE PROFILE

Prithwish Jana
Georgia Institute of Technology
20 PUBLICATIONS **105** CITATIONS
SEE PROFILE

Sayak Chakraborty
Queen Mary University of London
2 PUBLICATIONS **10** CITATIONS
SEE PROFILE

Sanjoy Kumar Saha
Jadavpur University
121 PUBLICATIONS **1,170** CITATIONS
SEE PROFILE

Two Stage Semantic Segmentation by SEEDS and Fork Net

Aritra Mukherjee
Department of Computer Science & Engg.
Jadavpur University
Kolkata, India
kalpurush1601@gmail.com

Prithwish Jana
Department of Computer Science & Engg.
Jadavpur University
Kolkata, India
jprithwish@gmail.com

Sayak Chakraborty
Department of Computer Science & Engg.
Calcutta Institute of Engineering and Management
Kolkata, India
sayak.cs19@gmail.com

Sanjoy Kumar Saha
Department of Computer Science & Engg.
Jadavpur University
Kolkata, India
sks_ju@yahoo.co.in

***Abstract*—Semantic segmentation of image is one of the most challenging and researched topic in the field of computer vision. Statistical methods can be employed for the task with low computational resources, but in a diverse natural environment, it fails to label many complicated objects. Deep learning methods are quite popular now for high accuracy but dense semantic segmentation at pixel level accuracy is very resource-intensive and not suitable for robot vision. Proposed methodology merges the best of both worlds to semantically label superpixels computed by a statistical method, with a deep net. The deep convolution network is novel in its use of superpixels in different fields of vision. The methodology is tested on the Pascal VOC dataset and compared with recent popular approaches. The results show that the proposed methodology is on par with the best results.**

***Index Terms*—Deep learning, Semantic segmentation, Superpixel**

I. Introduction

The problem of semantically segmenting an image is an important problem and many researchers tried to solve the problem through various approaches. Semantic segmentation is different from normal segmentation in the sense that, segments are formed and labelled semantically rather than just being formed. The goal is not just to segment the image in a statistically meaningful way, but to also label each pixel into a human recognizable class. Statistical measures like coherence of texture, colour, hue and other features solve the problem satisfactorily in a simple environment. But it may not be suitable in a complex environment where a region or object can have high variance in its own features. Deep learning method comes to rescue in such cases but for pixel-level categorization, these approaches are computationally very expensive. The proposed methodology tries to blend the strength of the two approaches by a dual-stage method. First, a fast and lightweight superpixel segmentation method, namely SEEDS (Superpixels Extracted via Energy-Driven Sampling), segments the image into small superpixels. The number of superpixels, thus formed, is much less compared to the total number of pixels in the image. Next, a deep convolutional network, named as Fork Net by us, is fed with a dual input viz. the superpixel and its neighbourhood for classification. Thus, the total number of times the deep network runs is reduced significantly without compromising much on pixel-level accuracy. This results in a very fast system, that is capable of being applied in real-time applications like robot vision and semantic SLAM (Simultaneous Localization And Mapping). The novelty of the proposed methodology lies in the combination of two approaches viz., using statistical method for superpixel generation and deep classifier for semantic labelling, alongside the design of Fork Net. It is trained and tested on the Pascal VOC dataset and compared against recent semantic segmentation methods, that employ deep network.

The paper is organized as follows. Brief introduction is followed by a review of past work in Section II. Section III and IV describe the proposed methodology and experimental results respectively. It is concluded in Section V.

II. Previous Works

Pixel-level semantic segmentation of images and videos by deep networks, is a relatively recent area of research [1], [2]. Convolutional Neural Network (CNN) is a popular approach for pixel-level labelling. Farabet et al. [3] proposed to use CNN on different scaled patches centered on a pixel, to classify it. Chen et al. [5] used DCNN (Deep CNN) followed by a fully connected CRF (Conditional Random Field) for pixel-level labelling, which was further enhanced later [12]. FCN (Fully Convolutional Network) by Long et al. [7] is yet another popular method for semantic segmentation. It adapts any traditional image classification network for labelling the pixels of variable-sized images. It uses a skip architecture that combines results from deep and shallow layers. These methods work on each pixel and thus take a longer execution time.

Some segmentation approaches result into a region-based output. Girshick et al. [4] used a selective search method [22]

ISBN: 978-1-7281-4283-8 PART NO.: CFP20O01-ART

to generate object proposals as bounding boxes, and then classified them by a CNN. Hariharan et al. [6] employed MCG [23] to get region proposals and used CNN for classifying these regions. But, annotating image for pixel-level semantic labelling is a tiresome work. Dai et al. [8] proposed a method to avoid this and used bounding box annotations and auto-generated candidate masks to train a network. Noh et al. [9] proposed a Deconvolutional Neural Network approach to generate a semantically segmented output at one shot. An instance-aware system was proposed by Dai et al. [10] that contains separate networks for estimating instances, masks and its subsequent category in a cascaded way. A similar work was later proposed by Li et al. [14]. These methods use a fixed FOV (field of view) which restricts the context for classification.

Attention network along with multiscaled input for pixel label refinement was proposed by Chen et al. [11]. Lin et al. [13] proposed a multiscaled input into parallel networks with multi-path refinement. The fusion at the end mitigated the loss due to pooling in traditional approaches. Atrous spatial pyramid pooling alongwith a deconvolution network, like that of Noh et al. [9], was proposed by Chen et al. [15] to handle the fuzzyness of categorization at object boundaries. Zhang et al. [16] followed a similar model and further enhanced the combination by roping in global context encoding. Though the accuracy was high, the speed of operations was not real-time.

Zhao et al. [17] proposed a selective multi-scaled input approach to refine only those portions of the image that has a low variance in class score, rather than all pixels. This served as an improvement over the famous PSPNet [30]. Additionally, the selective approach enhanced speed by using a heuristic approach that eliminated exhaustive evaluation of pixels. With enhanced speed, semantic segmentation became possible on robotic applications. Moreover, to further elevate the efficiency, Tsai et al. [18] proposed the use of adversarial network by adapting it in the context of semantic segmentation.

In recent time, Xian et al. [20] proposed a similar idea for pixel-level classification of novel classes with very few training instances. They relied on knowledge transfer using semantic projection network. Wu et al. [19] took this idea of adapting to unseen environments a step further by generating annotated synthetic datasets based on trained data, and further adapting the network to it. Recently, the idea of *neural architecture search* is also gaining steam. Here, the network changes its cell-level architecture to adapt to a given domain more efficiently, than manual tuning [21].

From the above study, it can be concluded that the field of dense semantic segmentation has become much active in the recent years. Researchers are trying to enhance accuracy, eliminate model bias and yet make the system faster for real-time operations. Most of the methodologies have a common theme of using CNN, FCN, atrous pooling and multi-scaled input in some combination. The proposed methodology thus employs CNN along with expanding reception field, not at pixel-level but at superpixel level somewhat similar to [6]. The proposed methodology also takes philosophical design cues from the works of [7], [11], [16].

III. Proposed Methodology

As mentioned earlier, the proposed methodology tries to combine the best of both statistical and deep learning methods to shape an efficient system for pixel-level semantic image segmentation. The advantage of statistical superpixelation lies in its speed, without a burden of computational overload. But the accuracy of such approach is low and cannot label the segments semantically. Deep CNN results into more accurate semantic classification but it requires huge computational overload even with GPUs. As pixel-level classification requires its neighbourhood patch to be formed as the input, in its most unsophisticated form, the number of times the deep net needs running equals the number of pixels. Even with multi-scaled input, the segmented output is in the form of a fuzzy heatmap and some post-processing like CRF is needed to obtain a crisp output. The proposed methodology uses a lightweight statistical method for superpixel formation and then the deep net runs only the number of times equal to the count of superpixels. As a superpixel covers all pixels in a neighbourhood that possess similar statistical features like hue, local texture etc., the label of a superpixel can be spread to all its member pixels with high confidence. This significantly increases the speed without compromising much on accuracy. A block diagram of the entire process is given in Fig. 1.

The proposed methodology can be broken down into two broad steps *i.e.* superpixel formation and its classification. We used SEEDS superpixel method proposed by Bergh et al. [24]. Prior to superpixelation, the images were pre-processed for a finer edge correspondence. First an edge-preserving image smoothing filter, known as fast non-local means denoising [25], was applied. As edges play a major role in superpixel-based segmentation, measures were taken to further enhance the edges. A spatial derivative of first-order and second-order are computed on the filtered image. Then the pre-processed image is formed as a weighted average of intensities of the denoised image and the two derivatives (first and second), with the weight ratio as 4:2:1. The output thus obtained is used for superpixelation.

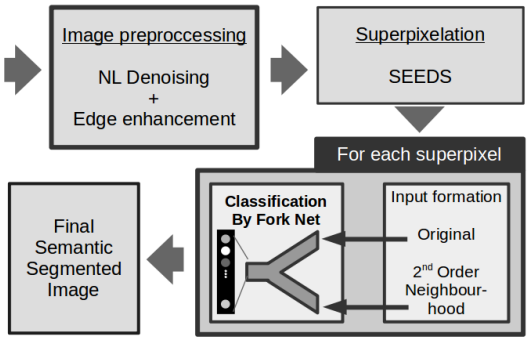

Fig. 1. Overall block diagram

SEEDS superpixelation algorithm [24] segments an image into superpixels by optimizing an energy function consisting of two terms. One term focuses on the colour distribution in a

ISBN: 978-1-7281-4283-8 PART NO.: CFP20O01-ART

superpixel, whose maximization results in colour homogeneity. The other one is the boundary term that refines the shape of the superpixel. The energy term is modeled by the ratio of inter-class and intra-class pixel hue variation, and thereby maximizing it results in redistribution of boundary pixels to correspond to actual segment edges. The process initiates by dividing an image into regular non-overlapping blocks as initial superpixels. It then iterates in a Hill-climbing optimization fashion which tries to refine segment boundaries, alongside maintaining color consistency. In comparison to traditional segmentation, SEEDS is extremely fast and usable for near real-time applications. The input parameters for SEEDS are the maximum number of superpixels allowed, number of iterations and edge refinement factor. SEEDS over-segments the image, but this serves as an advantage for our purpose whereby we avoid accidental coverage of multiple objects with a single superpixel. After extensive experiments, the number of maximum superpixels is kept fixed at 200 and the refinement level is set to the highest for a crisp segment edge. The output of SEEDS is a labelled image with labels varying from 0 to $N_{sp}-1$, where N_{sp} is the number of optimal superpixels under the maximum limit computed by SEEDS. A table of bounding box information for all the superpixels and a neighbourhood graph is computed on the output thus obtained. This is deemed necessary for the next phase.

To process the input to the Fork Net, we use the pre-computed neighbourhood graph. The inputs for semantic classification are the original superpixels obtained by SEEDS and the superpixels formed by their selective second-order neighbours, searched through BFS. Apart from the distance from the root, we have used another parameter to identify potential neighbours. We define this by the ratio of coefficient correlation [26] and Bhattacharyya distance [27] of the neighbour superpixel and the root. If the ratio is greater than 1, the corresponding superpixel is considered for neighbourhood formation. An example is given in Fig. 2. The driving philosophy behind this is to give the superpixel and its context as the input to the deep net classifier for better accuracy. As the network works in a parallel manner, no extra time is required than a single-stream CNN, just a little extra GPU memory.

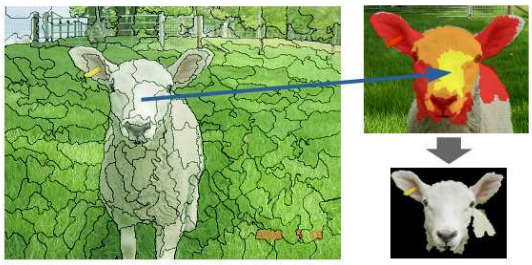

Fig. 2. Neighbourhood formation for second order superpixels. The yellow is the root superpixel, the orange are selected first-order and the red are selected second-order superpixels.

The proposed model is a two-stream framework shaped like a fork, and hence the name. It extracts features from the parallel streams and ultimately combines them, to generate predictions for each super-pixel. For all the superpixels, the two streams (*Stream1* and *Stream2*) takes the root superpixel and the neighbourhood superpixel respectively. Since the root superpixel covers a very small field of view, the size of *Stream1* images are kept on the smaller side, i.e. 32×32. But such small images can often lead to misinterpretations even to the human eye, and may not always be independently meaningful. In contrast, to keep up with the intricacies in *Stream2* images, its size is kept higher, i.e. 128×128. The expanded neighbourhood has the context necessary for proper classification of the root superpixel, but is susceptible to be bewildered by heterogeneous object-classes. Thus the two streams are combined. An early-fusion technique is followed to combine scores at the kernel-level, as it proffers the best results when the convolutional-pooling streams deal with analogous data [34]. Semi-global features on a larger FOV of superpixel neighbourhood, combined with detailed local features of the superpixel alone gives a better semantic context for classifying each superpixel. The complete architecture is elaborated in Fig. 3.

Stream2 follows similar layer stacking to that of VGG-16 [28], with modifications of decreased kernel sizes in the bottom-most convolutional block. Unlike *Stream2*, *Stream1* deals with smaller images which exhibit lesser chances of class heterogeneity. Thereby, features can be exploited from them via a relatively shallow network and thus the top two convolutional blocks of *Stream2* are excluded in *Stream1*. After the max-pooling layers of each stream, the spatial data is pooled using Global average pooling. This puts on a mean pooling on the spatial dimensions until each spatial dimension is reduced to one, and leaves other dimensions unaltered. The output from the global pooling layers from each of the streams, are concatenated into 768 nodes. This is followed by two dense fully-connected layers, the latter of which outputs class prediction values. The network is trained on superpixels obtained from Pascal VOC dataset [29]. There was a total of 2063 training images resulting in 288190 superpixels, 65% of which were chosen randomly.

After the classification of semantic classes for each superpixel, all the pixels belonging to that are labelled with the same class. There is no post processing needed as the segment edges are already crisp due to superpixel optimization, thus saving computation time.

IV. Experimental Results

Proposed methodology has been implemented using C++ along with opencv [31] library for pre-processing, SEEDS and the neighbourhood superpixel generation part. Python has been taken up for the deep net part along with Keras [32] over Tensorflow [33]. For testing, the Pascal VOC [29] dataset has been used with 850 test images generating 118039 superpixels in total. The methodology was compared with the mean of intersection over union (mIoU) metric which is standard for Pascal leader-board. The performance is compared with

ISBN: 978-1-7281-4283-8

PART NO.: CFP20O01-ART

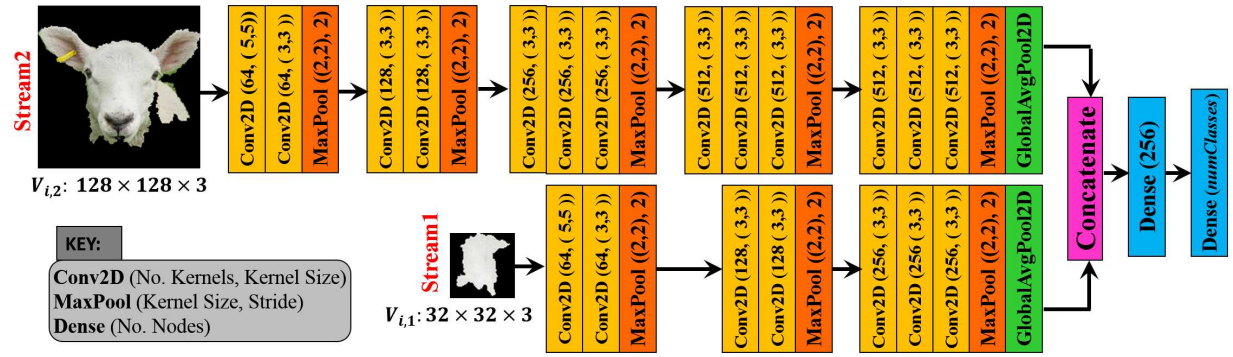

Fig. 3. Overall architecture of Fork Net. Stream 1 takes the root superpixel and stream 2 takes the contextual neighbourhood for semantic classification of the root superpixel.

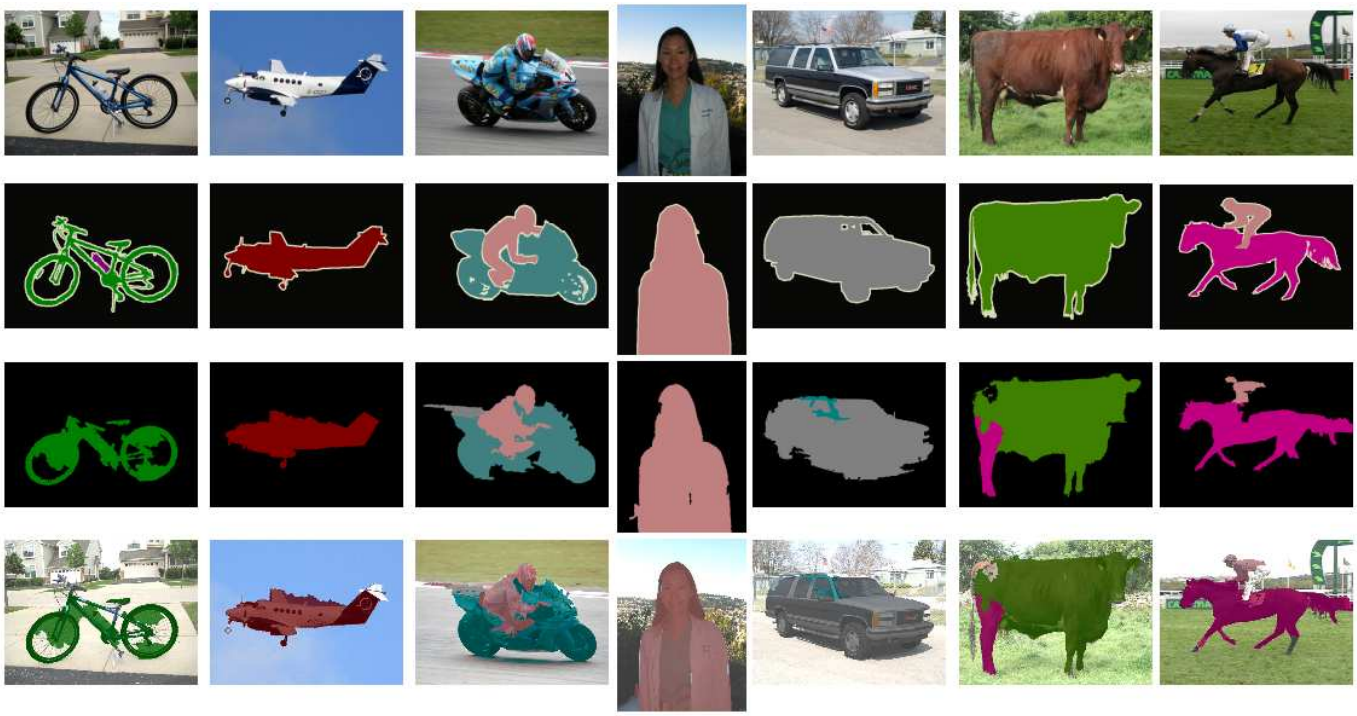

Fig. 4. Semantic segmentation output of our system, first row is input, second row is ground truth, third row is our result and last row is the result overlayed on input

the works like, FCN [7], DeepLabv2 [5], DeconvNet [9], DeconvNet [9] and EncNet [16]. Table I shows our average mIoU scores in comparison to others over all the classes in Pascal VOC. Performance data of these systems are taken from EncNet [16]. It is observed that the proposed methodology has best results in some classes, although the overall mIoU score is a little less. But with respect to speed, proposed methodology is quite fast. The total process takes an average time of 2 seconds for an image running on a 2.8GHz processor with single core active and a GPU of 4GB VRAM with only 92MB memory in use. Although this is not yet ready for real-time use, but with further tuning of statistical parameters of SEEDS and hyperparameters of the Fork Net, better speed can be achieved.

Fig. 4 shows some of the output and corresponding ground truth. It can be observed that the class 'person' is giving excellent results when present alone but with other classes, some of the superpixels corresponding to person are not classified properly. Due to this and the fact that a large number of test images contains person, there is a significant drop in the accuracy of the class. In future, this aspect may be looked into. Furthermore, many state-of-the-art approaches are trained on multiple dataset to improve the accuracy. It can be also tried to train the Fork Net with multiple dataset.

V. Conclusion

We have proposed a dense semantic segmentation methodology using statistical superpixel generation and deep network superpixel classification approach. It provides the accuracy

ISBN: 978-1-7281-4283-8

PART NO.: CFP20O01-ART

TABLE I
COMPARISON OF DIFFERENT APPROACHES OVER CLASSES OF PASCAL VOC DATASET ON MEAN INTERSECTION OVER UNION PARAMETER

Methodology	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	overall
FCN [7]	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	65.9	62.2
DeepLabv2 [5]	84.4	64.5	81.5	63.6	65.9	85.1	79.1	83.4	30.7	74.1	69.8	79.0	76.1	83.2	80.8	69.7	82.2	60.4	73.1	63.7	71.6
DeconvNet [9]	89.9	39.3	79.7	63.9	68.2	87.4	81.2	86.1	28.5	77.0	62.0	79.0	80.3	83.6	80.2	58.8	83.4	54.3	80.7	65.0	72.5
PSPNet [30]	91.8	71.9	94.7	71.2	75.8	95.2	89.9	95.9	39.3	90.7	71.7	90.5	94.5	88.8	89.6	72.8	89.6	64.0	85.1	76.3	82.6
EncNet [16]	94.1	69.2	96.3	76.7	86.2	96.3	90.7	84.2	38.8	90.7	72.3	90.0	92.5	88.8	87.9	68.7	92.6	59.0	86.4	73.4	82.9
Ours	87.8	90.1	77.6	86.8	89.6	82.4	79.2	78.1	81.0	85.5	84.6	72.9	80.7	85.8	57.5	86.7	85.0	82.1	76.7	90.8	81.9

comparable to the state-of-the-art methods, but is also reasonably faster. However, the proposed methodology is in its nascent stage and active research is on to enhance both, accuracy and speed. As the system is mainly intended for the use in robot vision in a near real-time pace, a superpixel segmentation based on disparity map of stereo image is also under consideration. The proposed methodology has achieved satisfactory results in its primary stage and holds promise to deliver better results with further modifications.

REFERENCES

[1] B. Zhao, J. Feng, X. Wu, and S. Yan, "A survey on deep learning-based fine-grained object classification and semantic segmentation," *International Journal of Automation and Computing*, vol. 14, no. 2, pp. 119-135, 2017.

[2] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," *Applied Soft Computing*, vol. 70, pp. 41-65, 2018.

[3] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915-1929, 2012.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580-587, 2014.

[5] L.C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A.L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014, in press.

[6] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *Proc. European Conference on Computer Vision, Springer, Cham*, pp. 297-312, 2014.

[7] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431-3440, 2015.

[8] J. Dai, K. He, and J. Sun, "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *Proc. IEEE International Conference on Computer Vision*, pp. 1635-1643, 2015.

[9] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE International Conference on Computer Vision*, pp. 1520-1528, 2015.

[10] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3150-3158, 2016.

[11] L.C. Chen, Y. Yang, J. Wang, W. Xu, and A.L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3640-3649, 2016.

[12] 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," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834-848, 2017.

[13] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1925-1934, 2017.

[14] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2359-2367, 2017.

[15] L.C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. European Conference on Computer Vision*, pp. 801-818, 2018.

[16] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7151-7160, 2018.

[17] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," in *Proc. European Conference on Computer Vision*, pp. 405-420, 2018.

[18] Y.H. Tsai, W.C. Hung, S. Schulter, K. Sohn, M.H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7472-7481, 2018.

[19] Z. Wu, X. Wang, J.E. Gonzalez, T. Goldstein, and L.S. Davis, "ACE: Adapting to changing environments for semantic segmentation," in *Proc. IEEE International Conference on Computer Vision*, pp. 2121-2130, 2019.

[20] Y. Xian, S. Choudhury, Y. He, B. Schiele, and Z. Akata, "Semantic projection network for zero-and few-label semantic segmentation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8256-8265, 2019.

[21] C. Liu, L.C. Chen, F. Schroff, H. Adam, W. Hua, A.L. Yuille, and L. Fei-Fei, "Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 82-92, 2019.

[22] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[23] P. Arbeláez, J. Pont-Tuset, J.T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 328-335, 2014.

[24] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool, "Seeds: Superpixels extracted via energy-driven sampling," in *Proc. European Conference on Computer Vision*, pp. 13-26, 2012.

[25] A. Buades, B. Coll, and J.M. Morel, "Non-local means denoising," *Image Processing On Line*, vol. 1, pp. 208-212, 2011.

[26] R. Taylor, "Interpretation of the correlation coefficient: a basic review," *Journal of Diagnostic Medical Sonography*, vol. 6, pp. 35-39, 1990.

[27] A. Bhattacharyya, "On a measure of divergence between two multinomial populations," *Sankhyā: the Indian Journal of Statistics*, pp. 401-406, 1946.

[28] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. International Conference on Learning Representations*, 2015.

[29] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303-338, 2010.

[30] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2881-2890, 2017.

[31] G. Bradski, "The opencv library," *Dr Dobb's J. Software Tools*, vol. 25, pp. 120-125, 2000.

[32] F. Chollet, and others, "Keras," 2015. Available: https://keras.io

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

[34] P. Jana, S. Bhaumik, and P.P. Mohanta, "A multi-tier fusion strategy for event classification in unconstrained videos," in *Proc. International Conference on Pattern Recognition and Machine Intelligence, Springer, Cham*, pp. 515-524, 2019.

ISBN: 978-1-7281-4283-8 PART NO.: CFP20O01-ART