Automatic Annotation of Object Instances by Region-Based Recurrent Neural Networks

Ionuţ Fîciu, Radu Stîlpeanu, Cosmin Ţoca, Anca Petre, Carmen Pătraşcu and Mihai Ciuc
University Politehnica of Bucharest
Faculty of Electronics, Telecommunications and Information Technology
Applied Electronics and Information Engineering Department
1-3, Iuliu Maniu Ave., Bucharest, Romania 061071

***Abstract*—In recent years, a wide variety of automatic, semi-automatic and manual approaches to image annotation have been proposed. These prerequisites have been driven by continuous advances of deep learning algorithms that often encounter the problem of insufficient or inappropriate training data, as well as sub-par markings' accuracy which can have a direct impact on the model's performance regardless. The main contribution of this paper is the development of a complex annotation framework able to automatically generate high-quality markings. The annotation work-flow aims to be an iterative process allowing automatic labeling of object bounding boxes, while simultaneously predicting the polygon outlining the object instance inside the box. The markings' format is fully compatible with COCO Detection & Panoptic APIs that provide open-source interfaces for loading, parsing, and visualizing annotations. Following the completion of the research project funding this research, the code will be publicly available.**

I. Introduction

The main purpose of developing not only large-scale datasets [1], [2], [3], [4], [5], but also containing very precise information about both the location of objects and relationship between them is to understand the semantics of scenes. Most of the scene understanding requirements come from the progress of the applications and devices we use on a daily basis. Industry interest is gradually passing from the precise location of objects in images [6], [7], [8], [9] and pixel-wise or even instance-level segmentation [9], [10], [11], [12], [13], [14], [15], [16] to understanding the actions that take place in video sequences [17], [18], [19], [20], [21]. However, compelling needs of both mobile and automotive industries relate to locating and delimiting objects with high precision on their contours. This shift to more difficult and complete tasks is also reflected in the challenges[1][2] announced by some of the most important conferences of the year. The motivation is to unify the typically distinct detection and segmentation tasks in order provide a highly comprehensive evaluation suite for modern visual recognition and segmentation algorithms. This is an important breakthrough toward real-world vision systems, needed for different application fields, such as augmented reality or autonomous driving.

The reasoning for using more sizable and varied datasets, as well as the need for rapid annotation of extensive datasets comes from the fact that the identity of many objects can only be inferred using contextual information. But, due to the high performance that deep learning technologies have demonstrated on current detection and segmentation benchmarks, research communities often use the same approaches deployed for inference in order to annotate extensive datasets of images depicting scenes.

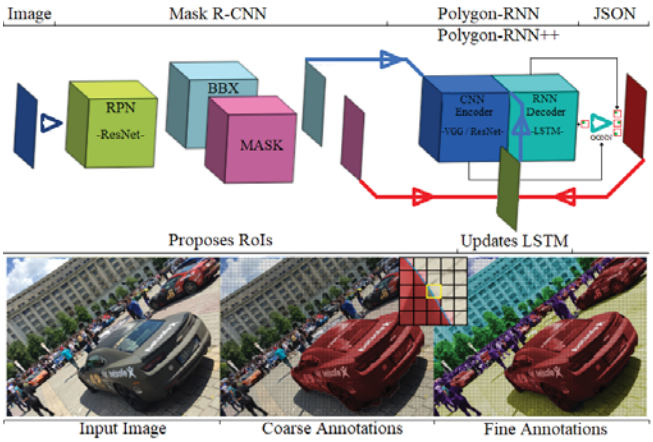


Fig. 1. **The Automatic Annotation Mechanism**: Mask R-CNN[9] detects objects while simultaneously generates a segmentation mask for each instance. Each bounding box is used by Polygon-RNN[22] to produce vertices of the polygon outlining the object. A new polygon is computed for each object, based on the segmentation mask previously derived. Starting from the biggest error between the two polygons, the RNN (two-layer convolutional LSTM[23] network) is updated recursively in order to obtain a precise object boundary.

The main focus of the work presented in this paper is to make the annotation process as fast as possible by eliminating the dependency of human interaction, while yielding the quality of markings as precise as the ones provided by manual labeling, currently available in a limited number of datasets [2], [4].

Consequently, the paper proposes a combination of object detection [6], pixel-wise instance-level segmentation [9] and a recurrent approach for predicting the polygon outlining the object [22]. More specifically, in the proposed automatic annotation framework, Mask R-CNN [9] provides bounding boxes that are used by Polygon-RNN [22] to predict the polygon outlining for each object instance inside the box.

[1]COCO & Mapillary Joint Recognition Challenge (ECCV2018)
[2]Berkeley Deep Drive Challenge (CVPR 2018)

978-1-5386-8445-0/18/$31.00 ©2018 European Union

Moreover, Mask R-CNN also provides an instance level segmentation mask that is then compared to the prediction given by the Polygon-RNN in order to obtain a global error measure between the segmentation map and initial object boundaries. In the end, starting from the biggest error, we iteratively update the LSTM to re-predict the spatial location of the erroneous vertex given better priors. Based on its recurrent nature, all the other vertices of the polygon will be automatically adjusted.

The remainder of this paper is organized as follows. Section II provides a brief overview of related mechanisms for image annotation. In Section III we propose an approach to automatically annotate object instances in images for their use in developing, training and evaluating detection or segmentation algorithms. A detailed series of experimental results is presented in Section IV, followed by the main conclusions depicted in Section V.

II. Related work

Advances in the field of computer vision have usually been driven by continuous development of large-scale datasets, their generation being an expensive and time consuming process. However, the most important problems still remaining in the case of automatic annotation are the time and resources needed for manual annotation, and quality of markings.

Manual annotation has long been managed and widely used through crowd-sourcing services like Amazon's Mechanical Turk [24]. In this regard, there is a long list of annotation tools [25], [26], [27], [28], [29], [30], which fall short of bringing the annotations together in a unified manner because of the lack of a common procedure. Using a unified approach could allow the easy integration of additional objects or images in the dataset.

Semi-supervised and interactive annotation [31], [32] rely on frequent user intervention to provide guidance or correct errors. These methods typically focus more on learning the object's appearance based on manual annotations.

However, there are a number of image datasets [4] that have been obtained using automatic or unsupervised methods that assume no human input during the annotation process.

Research communities often use the same approaches deployed for inference in order to annotate extensive datasets, but they generally lead to less reliable results when used for training, given their lack of accuracy and increased variability characterizing the resulted annotations.

Because of this, current large-scale benchmarks [3], [4], [1], [33], [34] are typically the result of an interactive annotation process based on accurate polygonal delineation of objects boundaries.

A relevant example of both interactive and unsupervised annotation algorithm is Polygon-RNN++ [35]. Relying on Polygon-RNN [22] the authors employ Faster R-CNN [6] to automate the annotation process. The predicted boxes are then fed to the model to produce polygonal instance segmentation of the object inside the box. In its interactive use mode, the annotation process requires a human in the loop to correct an erroneously predicted vertex.

Fig. 2. **Related Vertices:** List of pairs $\mathcal{F}(\nu_i^{\mathcal{P}}, \nu_i^{\mathcal{M}})$ of vertices of the two polygons $\mathcal{M}$ (green) and $\mathcal{P}$ (red). To illustrate, we have chosen a picture on which $\mathcal{P}$ goes wrong. Vertices of $\nu^{\mathcal{M}}$ are used to update the long short-term memory network.

There are several other differences between the two approaches. The last one changes the encoder network from VGG-16 [36] to a ResNet-50 [37] that has been modified to increase the resolution of the output feature maps without reducing the receptive field of the neurons.

Also, the recurrent decoder has undergone some fundamental changes. The authors added a branch to the same network that predicts the first vertex but also an attention mechanism that gates certain locations in the feature maps and focuses only on the relevant information in the next time step. The decoding mechanism also involves the addition of an evaluator network aiming to effectively choose among multiple candidate polygons. At the end, the authors added a Gated Graph Neural Network (GGNN) [38] to generate polygons at a much higher resolution by computing relative displacements of each of the initial vertices with respect to their corresponding ground-truths.

In order to overcome previously discussed limitations, but also to remove the need for human interaction to correct errors we propose an automatic approach for annotating object instances which combines Mask R-CNN and Polygon-RNN in an iterative mechanism for annotating large-scale image datasets. The errors are analyzed and corrected by a recursive and automated approach that sequentially instantiates a series of vertices of the polygon which are then fed back to the model, helping the model to correct its prediction in the next time steps in order to minimize a global measure of the variation between iterations.

III. The automatic annotation mechanism

Undergoing to the first part of the proposed mechanism, respectively getting bounding boxes for each candidate object and its class, we use proposals given by Mask R-CNN [9]. Unlike Acuna's work[35], the advantage of using Mask R-CNN is given by a better framing of the bounding box due to

Fig. 3. **Qualitative Results on Cityscapes [4]:** Going from left to right are shown the: segmentation maps obtained directly with **Mask R-CNN [9]**, polygons got from **Polygon-RNN++ w/ GGNN [35]** based on bounding boxes given by Mask R-CNN and outputs obtained with the **Iterative Approach (Ours)**. There could be noticed several improvements on the object boundaries when either one or the other fails.

Fig. 4. **Polygon-RNN++ w/ GGNN:** Qualitative Results on Cityscapes [4] and DAVIS [39]. Because the initial point may be erroneous, these errors can propagate across the polygon.

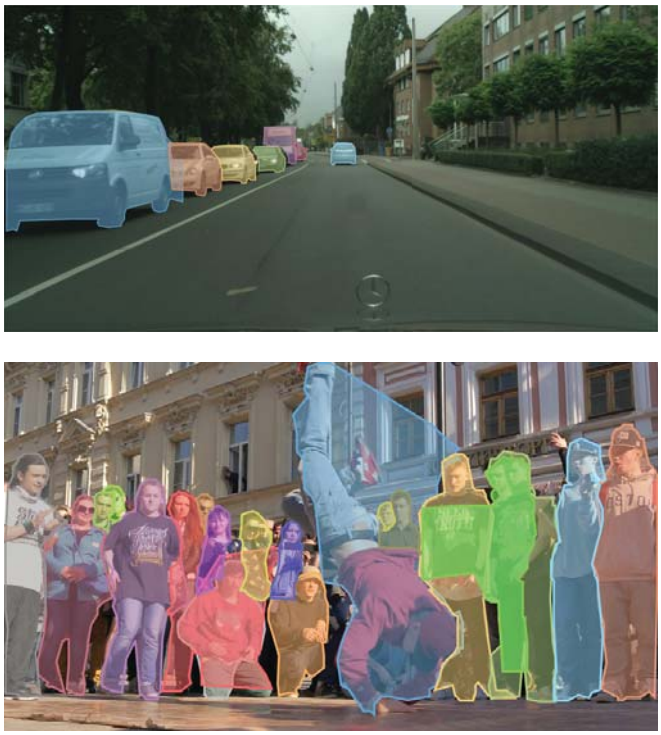
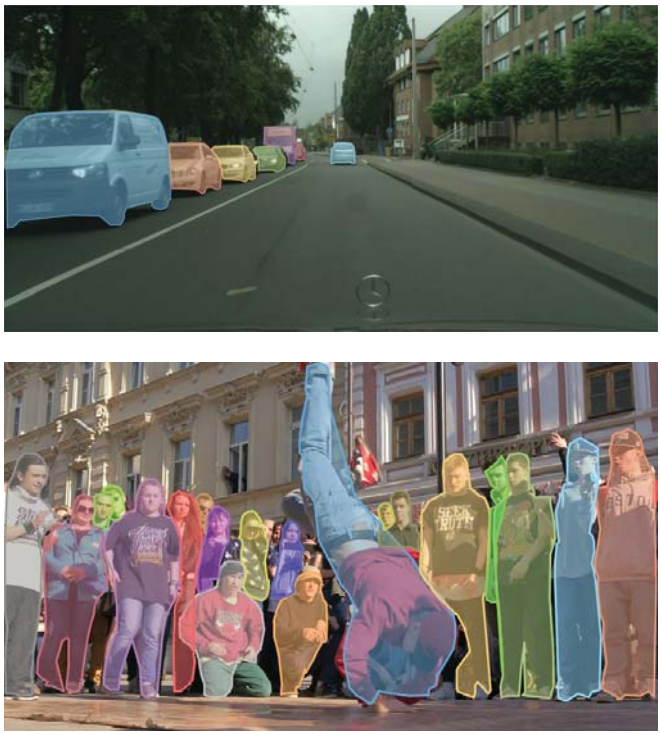


Fig. 5. **Iterative Approach (Ours):** Qualitative Results on Cityscapes [4] and DAVIS [39]. By automatically selecting better priors for the recurrent neural network the polygons are correctly adjusted.



the computation of the value of each sampling point by bilinear interpolation from the nearby grid points on the feature map.

Getting down to sequentially producing vertices of the polygon outlining the object, we fed all the bounding boxes together with a larger context around the object to Polygon-RNN [22]. In this way we get the initial prediction of vertices $\nu_i^{\mathcal{P}}$ of the polygon $\mathcal{P}$, where $i \in \{1, \ldots, n\}, n \in \mathbb{N}, \nu^{\mathcal{P}} \in \mathbb{R}^2$.

The main advantage of Polygon-RNN is that it allows the annotator to correct a misplaced vertex that is fed to the model at the following iteration replacing the initial prediction and getting the model back on the right track.

Instead of relying on a human annotator to make the corrections, we propose to use the segmentation mask given by Mask R-CNN.

Updating the erroneous vertices assumes determining relevant points on the mask outline. These points are used to build a polygon $\mathcal{M}$ related to $\mathcal{P}$.

Let's denote its vertices by $\nu_j^{\mathcal{M}}$, with $j \in \{1, \ldots, n\}, n \in \mathbb{N}$ and $\nu^{\mathcal{M}} \in \mathbb{R}^2$.

The relationship between $\mathcal{P}$ and $\mathcal{M}$ is given by:

$$\nu_i^{\mathcal{M}} = \arg\min_{\forall j} \left\{ \mathrm{d}^2(\nu_i^{\mathcal{P}}, \nu_j^{\mathcal{M}}) \right\}, \quad i \in \{1, \ldots, n\} \qquad (1)$$

where $\mathrm{d}^2(\nu^{\mathcal{P}}, \nu^{\mathcal{M}})$ is the squared Euclidean distance between $\nu^{\mathcal{P}}$ and $\nu^{\mathcal{M}}$, and $\nu^{\mathcal{P}}, \nu^{\mathcal{M}} \in \mathbb{R}^2$.

In other words, we build a polygon $\mathcal{M}$ from the mask outline by determining the minimum distance from each vertex of the polygon $\mathcal{P}$ to the boundary.

Let's consider the list of pairs $\mathcal{F}(\nu_i^{\mathcal{P}}, \nu_i^{\mathcal{M}})$ of vertices, as shown in Fig. 2, such that $\mathcal{F}$ is sorted in descending order by $\mathrm{d}^2(\nu_i^{\mathcal{P}}, \nu_i^{\mathcal{M}}), i \in \{1, \ldots, n\}, n \in \mathbb{N}$.

TABLE I
QUANTITATIVE RESULTS ON CITYSCAPES VALIDATION SET (500 IMAGES) [4]

Average Precision (AP [%])	Person	Car	Truck	Bus	Train	Motocycle	Bicycle
Polygon-RNN++ w/ Mask R-CNN	26.3%	38.1%	40.7%	**50.2%**	**48.7%**	12.5%	12.0%
Iterative Approach (Ours)	**29.2%**	**40.0%**	**41.9%**	48.4%	27.5%	12.5%	**12.3%**

The vertices $\nu_i^{\mathcal{M}}, i \in \{1, \ldots, n\}, n \in \mathbb{N}$ are then iteratively fed to the RNN, helping the network to correct its prediction in the following iterations.

The iterative process ends when the global loss function $\mathcal{L} = \sum_{i=1}^{n} \mathrm{d}^2(\nu_i^{\mathcal{P}}, \nu_i^{\mathcal{M}})$ has minor changes from one iteration to the next. Both Polygon-RNN and Polygon-RNN++ are supported within the proposed annotation infrastructure. The GGNN is optional, but when used it upscales, adds points and builds a polygon resembling human annotation.

The results of the automatic annotation approach proposed by the paper are stored in a series of JSON files, one for each image in the dataset. These files contain a polygon describing the contour for each labeled object. The resulting bounding boxes are computed using the minimal and maximal coordinates for the polygons of each object, in both horizontal and vertical directions.

The format of the annotations complies with a widely used marking standard, and namely the COCO Detection & Panoptic APIs [40] that allows using the open-source interfaces for loading, parsing, and visualizing annotations.

IV. RESULTS

This section focuses on both qualitative and quantitative evaluation of the proposed mechanism of automatically annotating objects by measuring the improvements brought by the suggested approach to the original model.

Qualitative results are shown in Fig. 4 and Fig. 5 which highlight the improvements on the edge of the objects made by the iterative approach proposed by the paper both on Cityscapes [4] and DAVIS [39].

Fig. 3 shows the cases in which the two algorithms Mask R-CNN and Polygon-RNN are processed separately and the improvements brought by the iterative method proposed by the paper over both.

To quantify the effect of the proposed approach but also to ensure our results cover a large range of real case scenarios we have employed the Cityscapes Dataset [4] that is an established benchmark used for instance level segmentation.

Since we have no exact method to measure the improvements we use the validation set of Cityscapes which is manually annotated and assess the performance gain brought by the proposed iterative mechanism besides Polygon-RNN++[35] on top of which we added Mask R-CNN[9] to get bounding-box proposals of object regions. The original paper [35] uses Faster R-CNN [6] as bounding box proposal algorithm, but for an accurate assessment we replace it with Mask R-CNN.

The validation set provides a wide number of manually annotated objects, which helps achieve a better generalization of the algorithms behaviour in complex situation.

Quantitative results are reported in Table I, where the quality gain is evaluated on instance-level. More precisely, we measure the average precision (AP) on the region level for each class and average it across a range of overlap thresholds (ranging from 0.5 to 0.95 in steps of 0.05) to avoid a bias towards a specific value. The overlap is computed at the region level, making it equivalent to the intersection over union (IoU) of a single instance. We penalize multiple predictions of the same ground truth instance as false positives, see the Cityscapes Evaluation Metrics[3].

The proposed mechanism outperforms the base model by up to 3% on the most frequent categories (persons, cars, trucks and bicycles). Results on the other categories of objects in the validation set do not seem to be relevant for this test case, either do not appear in images at all or appear very rarely.

V. CONCLUSIONS

The paper presents an unsupervised approach to instance-level boundary annotations, designed to automatically improve the accuracy of object boundaries, as well as create a practical flow that allows for easy integration of extensive object classes and images by effectively adapting to novel, out-of-domain datasets. Considering the obtained results, the proposed mechanism shows improvements on pixel-wise instance-level object segmentation up to 3%. Following a visual inspection of the results, it can be noticed that the quality improvements are achieved on the outline of the objects. The proposed solution complies with a widely used marking standard, and namely the COCO Detection & Panoptic APIs [40] that allows using the open-source interfaces for loading, parsing, and visualizing annotations.

ACKNOWLEDGEMENT

The work has been funded by the Romanian National Authority for Scientific Research and Innovation, through the Executive Unit for Higher Education, Research, Development and Innovation Funding CNCS/CCCDI-UEFISCDI, project number PN-III-P2-2.1-PED-2016-0292, within PNCDI III.

REFERENCES

[1] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," 2014.

[2] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," 2010.

[3] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ADE20K dataset," 2016.

[4] 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," 2016.

[3]Cityscapes Evaluation Scripts: github.com/mcordts/cityscapesscripts

[5] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving video database with scalable annotation tooling," 2018.
[6] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," 2015.
[7] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2015.
[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," 2015.
[9] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," 2017.
[10] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," 2016.
[11] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," 2015.
[12] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015.
[13] G. Lin, A. Milan, C. Shen, and I. D. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," 2016.
[14] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," 2016.
[15] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017.
[16] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters - improve semantic segmentation by global convolutional network," 2017.
[17] C. Feichtenhofer, A. Pinz, R. P. Wildes, and A. Zisserman, "What have we learned from deep representations for action recognition?" 2018.
[18] J. Y. Ng, J. Choi, J. Neumann, and L. S. Davis, "Actionflownet: Learning motion representation for action recognition," 2016.
[19] J. Liu, G. Wang, L. Duan, P. Hu, and A. C. Kot, "Skeleton based human action recognition with global context-aware attention LSTM networks," 2017.
[20] J. Zhu, W. Zou, Z. Zhu, and L. Li, "End-to-end video-level representation learning for action recognition," 2017.
[21] C. Wu, M. Zaheer, H. Hu, R. Manmatha, A. J. Smola, and P. Krähenbühl, "Compressed video action recognition," 2017.
[22] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler, "Annotating object instances with a polygon-rnn," 2017.
[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Nov. 1997.
[24] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data?" 2011.
[25] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," May 2008.
[26] X. Qin, S. He, Z. V. Zhang, M. Dehghan, and M. Jägersand, "Bylabel: A boundary based semi-automatic image annotation tool," 2018.
[27] R. Kawamura, "Rectlabel," 2017.
[28] D. K. Iakovidis, T. Goudas, C. Smailis, and I. Maglogiannis, "Ratsnake: A versatile image annotation tool with application to computer-aided diagnosis," 2014.
[29] A. Klaser, "Lear," 2017.
[30] A. Bréhéret, "Pixel annotation tool," 2017.
[31] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "Scribblesup: Scribble-supervised convolutional networks for semantic segmentation," 2016.
[32] Y. Chen, J. Pont-Tuset, A. Montes, and L. V. Gool, "Blazingly fast video object segmentation with pixel-wise metric learning," 2018.
[33] G. Neuhold, T. Ollmann, S. R. Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," Oct. 2018.
[34] R. Mottaghi, X. Chen, X. Liu, N. G. Cho, S. W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," June 2014.
[35] D. Acuna, H. Ling, A. Kar, and S. Fidler, "Efficient interactive annotation of segmentation datasets with polygon-rnn++," 2018.
[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.
[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.
[38] Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel, "Gated graph sequence neural networks," 2015.
[39] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbelaez, A. Sorkine-Hornung, and L. V. Gool, "The 2017 DAVIS challenge on video object segmentation," 2017.
[40] A. Kirillov, K. He, R. B. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," 2018.