Joint COCO and Mapillary Workshop at ICCV 2019: COCO panoptic segmentation Challenge Track

Technical Report: Pixel Consensus Voting for Panoptic Segmentation

Haochen Wang
CMU (Work done at TTIC)
whc@ttic.edu

Ruotian Luo
TTI-Chicago
rluo@ttic.edu

Greg Shakhnarovich
TTI-Chicago
greg@ttic.edu

Abstract

The core of our approach, Pixel Consensus Voting, is a framework for bottom-up instance segmentation based on discretized pixel voting. Unlike a proposal-based sliding window detector that reasons about densely enumerated bounding boxes or masks, our method detects instances as a result of the consensus among pixel-wise votes. We implement vote aggregation and back-projection efficiently via convolutional operations. This detection pipeline is complementary to FCN-style semantic segmentation, producing a unified architecture for panoptic segmentation, which is efficient and competitively accurate.

1. Introduction

We propose a three-stage, bottom-up approach to instance segmentation. In the *voting* stage, for every pixel we predict the location (center) of an object it belongs to, or record an "abstention" vote if it's not part of an object. In the second *inference* stage, the accumulated votes are used to obtain object masks via back-projection. Finally, in the *semantic* stage, pixel-wise semantic segmentation predictions are used to assign categories to the inferred masks and to predict labels for non-object pixels, completing the panoptic segmentation map.

Notably, and in contrast to the currently dominant lineage of detection work derived from the RCNN family [2, 11, 3], our approach does not involve reasoning about bounding boxes or masks [1]. Instead, the masks emerge from the consensus among pixel-wise votes. In this our approach is a descendant of early Hough-based methods such as the Implicit Shape Models (ISM) [6].

Relationship to prior work In ISM, voting for object centers relies on memorized mapping from vector-quantized patches to offsets. In contrast, in our work patches are replaced with feature representation extracted from every pixel (over a large receptive field), and we use a learned machinery (convnet) to cast grid-discretized votes.

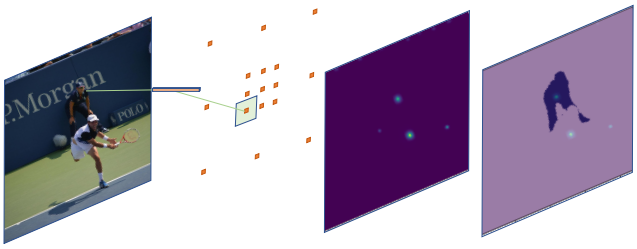

Figure 1: Pixels cast probabilistic votes with dilated deconvolution, and at each local maxima of the voting heatmap we convolve a query filter to get the instance mask. The deconv kernel is simplified for visualization.

Relationship to recent/concurrent work Some recent work follows broadly similar philosophy, but differs from ours in many ways. Neven et al. [9] cast learning to vote for object centers as a regression task, followed by clustering. This is different from our deconvolutional voting mechanism, and their results on COCO are to our knowledge significantly worse than ours. In ExtremeNet [13] and CornerNet [5] a convnet is trained to predict object keypoints. These models treat keypoint prediction as an end to end task, and are unable to account for pixel ownership that our model permits. As a result they are restricted to bounding box detection and cannot be used directly for instance/panoptic segmentation.

We offer two contributions to the field of image understanding:

- A voting-based, bottom-up detection philosophy. It's novel relative to the currently prevalent ideas, although it is very similar in spirit to older work.
- A highly efficient mechanism for vote accumulation and inference, based on convolutions.

Our approach is a potential alternative to region/bounding box/mask-based methods, all fundamentally derived from sliding window classification (top-down methods). While at the moment the accuracy of our approach lags behind state of the art, we believe that continuing improvements by us and others may bring it to match and exceed state of the art.

2. Pixel Consensus Voting

Given an input image, PCV starts with a convolutional neural network extracting a shared representation (feature tensor) that is fed to two independent sub-branches. The semantic branch predicts the category label for every pixel. The voting branch predicts for every pixel whether the pixel is part of an instance mask, and if so, where it is relative to the mask centroid. This prediction is framed as classification over a set of regions (grid cells) around the voting pixel, which allows capturing uncertainty about the offset. Both branches are trained with standard cross-entropy loss. The prediction from the voting branch are aggregated into a voting heatmap (*accumulator array* in the Hough transform terminology). A key technical innovation in PCV is a dilated convolution mechanism implementing this efficiently. Local maxima (peaks) of the hetmap are detection candidates. At each peak, we convolve a query filter to back-project the pixels that favor this particular peak above all others. These pixels together form a category-agnositc instance segmentation mask. Finally, we merge the instance and semantic segmentation masks using a simple greedy strategy, yielding the complete panoptic segmentation output.

2.1. Backbone and Feature Extraction

The focus of our work is to develop a general strategy to model and segment instances. To this end PCV reduces the training of instance recognition to pixel labelling, which can be tackled by various descendants of Fully Convolutional Network [8] . For the COCO Panoptic Challenge, we adopt the designs of recents works [12, 4] that repurpose a Feature Pyramid Network (FPN) [7] for semantic segmentation. We upsample the features from each stage of FPN, respectively at $1/32, 1/16, 1/8$ and $1/4$ of input resolution, to a uniform size of $1/4$ of the input scale, and reduces the channel dimensions from 256 to 128 with 1×1 conv, before channel-wise concatenation. On top of this merged representation we apply 2 deformable conv, a 1×1 conv, softmax and $4\times$ nearest neighbor upsampling to generate per-pixel labels. Note that we apply softmax first before upsampling since it is faster to produce instance masks at a lower resolution. The semantic branch predicts the labels for all 153 COCO Panoptic categories, and is a departure from PFPN [4] which lumps all 'thing' classes into a single category.

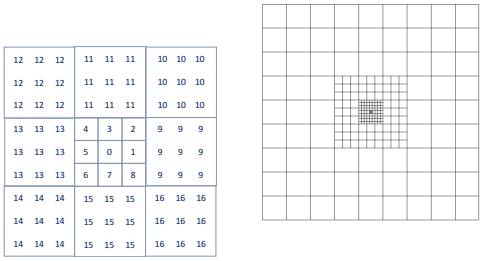

Figure 2: Left: A simple discretization of the 9×9 area around an instance centroid. Each number represents what the voting ground truth label for the pixel should be with the centroid at position (4, 4). We refer to this mask as the query filter. It is used for both ground truth assignment and back-projection. Right: the complete discretization schema of the 243×243 area around an instance centroid. It consists of 233 bins.

2.2. Region Discretization and the Query Filter

For a particular pixel, we discretize its surrounding regions into square bins whose diameters expand radially outward. It stands in contrast to predicting a direct (x, y) centroid offset vector that has been used extensively in prior works. This design is motivated by a few considerations.

First of all, perfect centroid prediction is not necessary for accurate instance masks. What matters is the consensus among pixels. Large instances can naturally tolerate more coarse predictions than small instances. We discretize the regions in such a way that the further the distance from the instance centroid, the more coarse the bin gets. Pixels of a small instance need to distinguish between tiny bins in order to find out its centroid, while the pixels of a large instance only need to point out a rough area its centroid might fall into. We verify through an oracle experiment that this radially coarsening discretization is sufficient for high quality instance segmentation.

Training a direct (x, y) regression with $l1/l2$ loss conflates prediction with uncertainty. An offset vector is a limited representation. If a pixel is unsure about where its center might be, then its best bet is to point in the middle of a few candidate regions, creating spurious peaks and false positives. This insight is echoed by the evolution of bounding box object detectors. Popular detectors such as Faster-RCNN [11] and YOLOv2 [10] have moved beyond direct bounding box regressions and discretize the space of bounding boxes into a few major modalities represented by anchor boxes. They classify a proposal into one of these modalities and use regression for correction. Taking this insight to the extreme, we use only classification, and cast the votes probabilistically to reflect the uncertainty within each pixel.

We implement our discretization schema with a query fil-

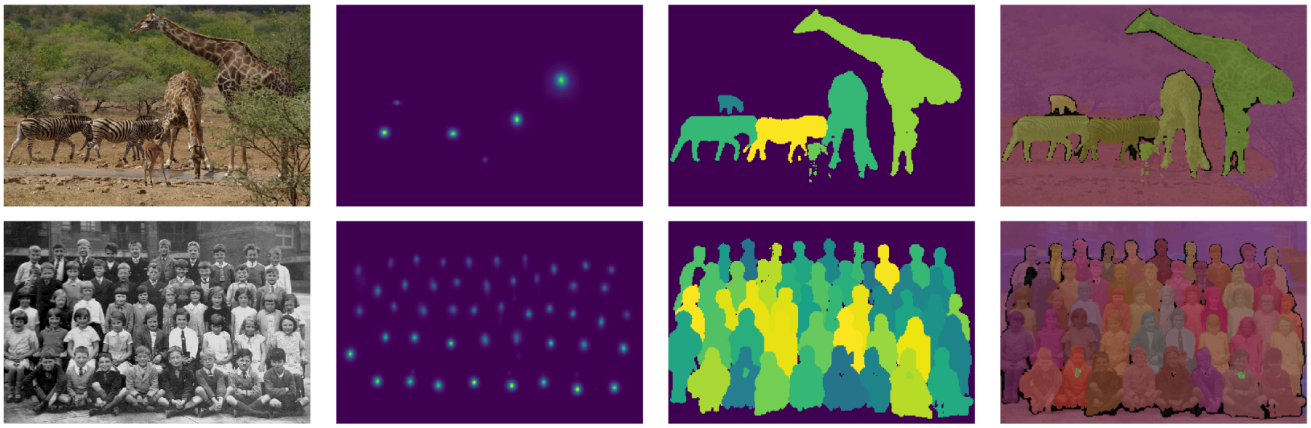

Figure 3: PCV results on COCO. From left to right: input image, voting heatmap, instance mask, full panoptic mask

ter. As shown in the left of figure 2, a query filter is a square mask that records the ground truth voting label each surrounding pixel should have if its instance centroid is placed right at the center of this filter. Pixels far from the centroid tend to have similar labels, reflecting the increased tolerance for uncertainty. During voting ground truth assignment, we overlay the query filter on top of an instance mask and align the center of the filter with the instance centroid. The voting label for each pixel is then directly read off from the aligned filter. For stuff pixels that do not belong to any instances, we create an 'abstention' label as an extra class. For the parts of an instance that falls outside this filter, we ignore them during training. Note that since inference is performed at $1/4$ of input resolution, our full query filter which has a side length of 243 can cover an instance as large as 972^2 pixel squares.

2.3. Voting as Dilated Deconvolution

Given a voting tensor of size $[H, W, C]$, where C is the number of discrete bins, consider a particular pixel: voting involves distributing the predicted likelihood over the C regions to their respective locations around this pixel. This process can be implemented efficiently via dilated deconvolution.

Deconvolution is the backward pass of convolution. A conv kernel aggregates spatial information to a single point, whereas a deconv kernel spreads a point signal to spatial locations. Deconv kernel is most often used for feature upsampling and the parameters of the kernel are learnable. For the purpose of vote aggregation, however, we initialize the deconv kernel to 1-hot across each channel and fix the parameters. Dilation in this case enables a pixel to cast its vote to faraway locations.

Consider the toy example of the query filter in Figure 2 Left. This schema discretizes a 9×9 region into an inner 3×3 grid of dilation 1, encircled by an outer 3×3 grid of dilation 3, hence $C = 9 + 8 = 17$ distinct voting classes. To aggregate the votes, we split the $[H, W, 17]$ vote tensor along the channel into 2 parts of size $[H, W, 9]$ and $[H, W, 8]$, and apply two deconv kernels of size $[9, 1, H, W]$ of dilation 1 and $[8, 1, H, W]$ of dilation 3 to produce two outputs of size both $[H, W, 1]$, sending all the votes to the center of each bin. To spread the votes out evenly within each bin, we apply constant-value smoothing kernels of the same spatial size as the respective dilations. Finally we sum the two outputs into a single voting heatmap. The deconv kernel for the full query filter is implemented analogously.

2.4. Back-Projection as Convolutional Filtering

Given the local maximas in the voting heatmap, back-projection aims to determine for every peak the pixels that favor this particular maxima above all others. We first record the argmax vote index at each pixel. Then, within the basin of a local maxima, we convolve the query filter and perform equality comparison against the argmax voting indices. Recall the semantics of the query filter: it marks what a neighboring pixel should vote if that pixel believes the filter center coincides with the instance centroid. Hence by convolving the query filter within a local maxima basin this way, we pick up all the pixels whose strongest vote falls within this peak region. This operation can be easily implemented on a GPU and in our model we extend the comparison to top 5 votes rather than just the argmax vote.

2.5. Implementation Details

Training PCV is trained on COCO Panoptic Segmentation. We resize the image such that the length of the shorter side is 800 and the length of the longer side does not exceed 1333. The input resizing is consistent for both training and testing. Left-right flipping is the only data augmentation used. We use SGD with momentum 0.9 and set the learning rate at 0.005, weight decay at 0.0001. The model

	All			Things			Stuff		
	PQ	SQ	RQ	PQ	SQ	RQ	PQ	SQ	RQ
ResNet 50	37.9	77.99	47.46	40.60	78.63	50.56	33.83	77.03	42.77
Frozen ResNext 152	40.99	78.78	50.88	46.05	79.97	56.64	33.35	77.00	42.18

is trained on 8 GPUs with batch size of 16 for a total of 90k iterations (around 13 epochs). The learning rate decays by a factor of 10 at 60k and 80k iterations. Batch norms in ResNet and FPN are frozen in our current setup. Both branches are trained with cross-entropy loss and we weight the two losses equally.

Instance Mask Centroid We define the centroid of an instance mask as its center of mass. Alternatively one could use the center of bounding boxes. Center of mass produces better results and is the more natural choice.

Peak Finding To locate the local maxima from the voting heatmap, we do a simple thresholding at 2.0 and then select all the connected components as local maxima basins. The results are stable across various thresholds and local maxima finders.

3. Experiments

We report results on COCO Panoptic Segmentation Val set that contains 5000 images. Using a ResNet 50 backbone, we reach a PQ of 37.9 with PQ things at 40.60. We also train our model on a deeper ResNeXt 152 backbone that freezes all layers up to stage 5 of the backbone so as to maintain batch size. This heavier model improves PQ things to 46.05. Our results so far are significantly below the state of the art.

4. Conclusion

Currently dominant object detection methods rely on the notion of bounding boxes as a scaffolding for generating and evaluating object masks. This is a top-down approach: first, a region likely to contain an object is selected, and then the prediction is verified and refined (in the case of Mask-RCNN to produce the accurate bounding box, category, and instance mask). Historically, this was in part motivated by the initial focus on bounding-box detection tasks (e.g., PASCAL VOC challenge). But when one removes the bounding box concept from the task, as in panoptic segmentation, the need to reason about bounding boxes is no longer obvious. Instead, we can try to directly go from image pixels to object masks, in a bottom-up fashion. PCV, an approch for doing just that, is an alternative to regions-based methods. This is still an early stage of our work on PCV and we expect that much can be improved, and we continue to experiment with modifications in voting and inference mechanisms.

Acknowledgements

We thank Michael Maire for many helpful discussions and feedbacks.

References

[1] Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollár. Tensormask: A foundation for dense object segmentation. *arXiv preprint arXiv:1903.12174*, 2019. 1

[2] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 1

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[4] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

[5] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *The European Conference on Computer Vision (ECCV)*, 2018. 1

[6] Bastian Leibe, Alevs Leonardis, and Bernt Schiele. Robust object detection with interleaved categorization and segmentation. *International journal of computer vision*, 77(1-3):259–289, 2008. 1

[7] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2

[8] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2

[9] Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[10] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 2

[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 2

[12] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8818–8826, 2019. 2

[13] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl. Bottom-up object detection by grouping extreme and center points. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1