FOURIERNET: COMPACT MASK REPRESENTATION FOR INSTANCE SEGMENTATION USING DIFFERENTIABLE SHAPE DECODERS

*Nuri Benbarka, *Hamd ul Moqeet Riaz, Andreas Zell

University of Tübingen, Wilhelm-Schickard-Institute for Computer Science, 72076, Tübingen, Germany

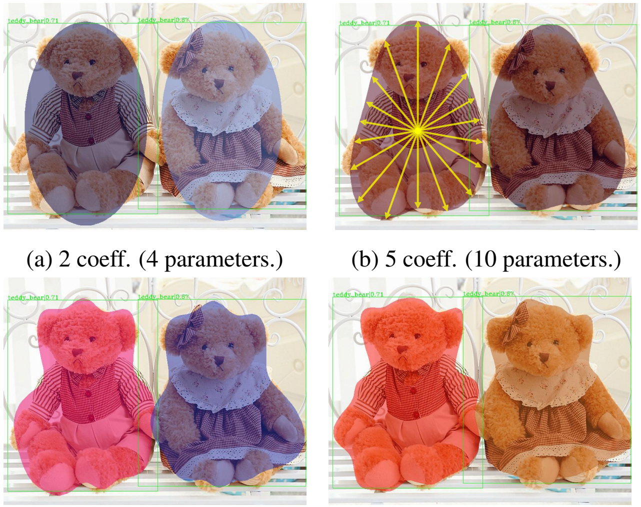
ABSTRACT

We present FourierNet a single shot, anchor-free, fully convolutional instance segmentation method, which predicts a shape vector that is converted into contour points using a numerical transformation. Compared to previous methods, we introduce a new training technique, where we utilize a differentiable shape decoder, which achieves automatic weight balancing of the shape vector's coefficients. Fourier series was utilized as a shape encoder because of its coefficient interpretability and fast implementation. By using its lower frequencies we were able to retrieve smooth and compact masks. FourierNet shows promising results compared to polygon representation methods, achieving 30.6 mAP on the MS COCO 2017 benchmark. At lower image resolutions, it runs at 26.6 FPS with 24.3 mAP. It achieves 23.3 mAP using just 8 parameters to represent the mask, which is double the amount of parameters to predict a bounding box. Code will be available at: github.com/cogsys-tuebingen/FourierNet.

Index Terms— Instance segmentation, Fourier series, anchor-free, shape encoding, differentiable algorithms.

1. INTRODUCTION

With the recent emergence of deep learning, combined with readily available data and higher computational power, the use of autonomous machines has become a realistic option in many decision making processes. In applications such as autonomous driving and robot manipulation, the first and foremost task is to perceive and understand the scene before a decision is made. Instance segmentation is one of the techniques used for scene understanding. It categorizes each pixel/region of an image by a specific class and at the same time distinguishes different instance occurrences. Among the instance segmentation methods are *two-stage* methods that produce a bounding box and then classify the pixels within that box as foreground or background [1, 2, 3, 4, 5, 6]. Although these are still dominant in terms of prediction accuracy, they are computationally intensive. There is a growing trend to use simpler, faster *single-stage* instance segmentation methods, that do not require initial bounding box proposals [7, 8, 9, 10, 11, 12]. In one of the latest approaches, ESE-Seg [9] have encoded the objects' contours using function approximations such as Chebychev polynomials and Fourier series. They trained a network to predict a shape vector (a vector of coefficients) that is transformed into contour points in the polar representation. The main advantage of this method is that it requires less parameters to represent the mask, compared to binary grid or polygon representations [13]. However, they regress the shape vector directly. We argue that direct regression of the shape vector does not weigh each coefficient according to its impact on the mask and prevents the model from learning the true data distribution.



(a) 2 coeff. (4 parameters.)

(b) 5 coeff. (10 parameters.)

(c) 10 coeff. (20 parameters.)

(d) 20 coeff. (40 parameters.)

Fig. 1: Comparison between the predicted mask of using varying number of Fourier coefficients. 1b shows how rays construct a contour from a center point.

Therefore, we propose an alternative training method in which the network outputs are passed through a *differentiable shape decoder* to obtain contour points that are used to calculate the loss. In this case, the losses of other polygon representation methods, e.g. PolarIOUloss [8] and Chamfer loss [12], can be used and the network is trained for its main task. The gradients of these losses are back-propagated through the decoder and the weight balancing of the different shape vector's coefficients is done automatically.

*These authors contributed equally to this work.

2. RELATED WORK

2.1. Two stage instance segmentation

Two-stage instance segmentation splits the task into two subtasks, object detection and then segmentation. The most popular instance segmentation method is Mask R-CNN [1], which is constructed on top of Faster R-CNN [14] by adding a mask branch in parallel to the bounding box and the classification branches. Moreover, RoI-Align was used instead of RoI-Pooling. Following on from Mask R-CNN, PANet [2] improved the information flow from the backbone to the heads by using bottom-up paths in the feature pyramid and adaptive feature pooling. In Mask Scoring R-CNN [3], the IoU of the predicted mask was estimated and it was used to improve the prediction scores. HTC [4] introduced the cascade of masks by merging detection and segmentation features and achieved enhanced detections. ShapeMask [6] introduced the concept of class-dependent shape priors and used them as preliminary estimates to obtain the final detection. CenterMask [5] built its work on FCOS [15], an efficient one-stage anchor-free object detector, instead of Faster R-CNN and followed it with a spatial attention-guided mask branch. The above methods use binary-grid representation of masks. In contrast, PolyTransform [13] uses a polygon representation and requires a mask for the first stage. The initial mask is refined by a deforming network to obtain the final prediction. These methods accomplish state-of-the-art accuracy, but they are generally slower than one stage methods.

2.2. One stage instance segmentation

YOLACT [7] generated prototype masks and simultaneously produced bounding boxes and combination coefficients. The bounding boxes crop an area of the prototype masks and the combination coefficients were used to merge them to construct the final mask. Likewise, Embedmask [10] generated pixel embeddings that differentiate each instance in the image and simultaneously produced bounding boxes and proposal embeddings. Here, the mask is constructed by comparing the proposal embedding with all pixel embeddings in the produced bounding box area. In addition to the previous binary-grid representation approaches, there are a few methods that employ polygon representation. ExtremeNet [11] used keypoint detection to obtain the extreme points of an object. Then a rough mask was created by forming an octagon from the extreme points. Polarmask [8] performed a dense regression of the distances from the mask center to points on the outer contour in polar coordinates. Additionally, since it was shown that the detections near object boundaries were generally poor [15], they used the concept of centerness, which gives greater importance to the detections near the center and thus enhanced the prediction quality. Dense RepPoints [12] introduced a general framework for using points to represent bounding boxes, polygon masks or boundary-binary maps. ESE-Seg [9] trained a network to predict a shape vector that is transformed into contour points in the polar representation. Although it requires fewer parameters, their training method is not optimal as mentioned before. Therefore, we propose *differentiable shape decoders* for training which is explained in the next section.

3. OUR METHOD

3.1. Mask representation

FourierNet uses an organized polygon representation [12] to represent masks. For each feature point i near the center of mass of the contour, N rays are extended to the point of intersection with the object boundary. The angle between the rays $\Delta\theta$ is constant and defined by $360°/N$. The length of these rays from the center point are described by $P_i = \{p_{0,i}, p_{1,i}, ..., p_{N-1,i}\}$. If there is more than one intersection point, the point with the longest distance is selected. Furthermore, a constant $\epsilon = 10^{-6}$ is assigned to rays which do not have intersection points, which occurs when the feature point i is outside or on the boundary of the contour. Note that the emerged contour would only be an approximation of the ground truth contour even with a high number of rays, however the IOU values can be up to 0.95 [9].

To determine P_i, an *Inverse Fast Fourier Transform* (IFFT) is applied to the coefficients predicted by the network (figure 2). The inverse discrete Fourier transform is defined by

$$p_{n,i} = \frac{1}{N} \sum_{k=0}^{N-1} x_{k,i} e^{\frac{j2\pi kn}{N}}, \tag{1}$$

where $p_{n,i}$ is the n_{th} ray in P_i and $x_{k,i}$ is the k_{th} coefficient of X_i, which is the Fourier transform of P_i. In cases where we predict more rays than Fourier coefficients, the network predicts a subset of the coefficients $S_i \subset X_i$ and then we pad the rest of the output tensor with zeros on higher frequency components. This is done to equalize the dimensions before and after the IFFT. Note that the IFFT algorithm is differentiable and therefore the training is done directly on P_i and thus the name *differentiable shape decoder*.

3.2. FourierNet

FourierNet is an anchor-free, fully convolutional, single shot network (figure 2). It has a backbone followed by a top down feature pyramid network (FPN) [16] with lateral connections. It has 5 spatial resolutions (heads) at the reduction levels 8, 16, 32, 64, and 128 with respect to the original image. Each head predicts a set of classification scores, centerness, and Fourier coefficients at each spatial location in the feature map. The classification branch predicts scores for each class and is trained using *focal loss* [17].

Centerness is a term that measures the closeness of a feature point to the center of a mask. Polar Centerness (PC) [8]

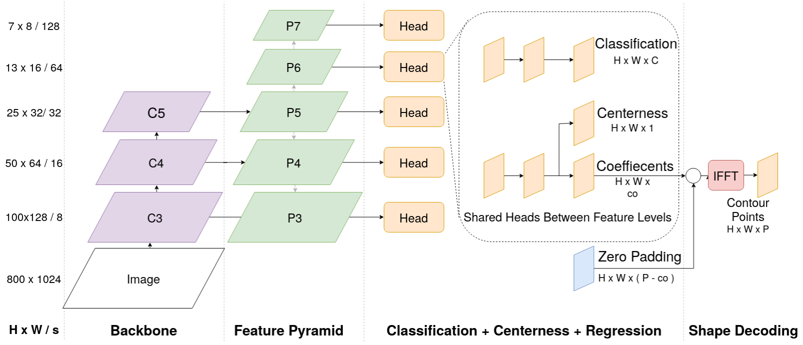


Fig. 2: The FourierNet architecture

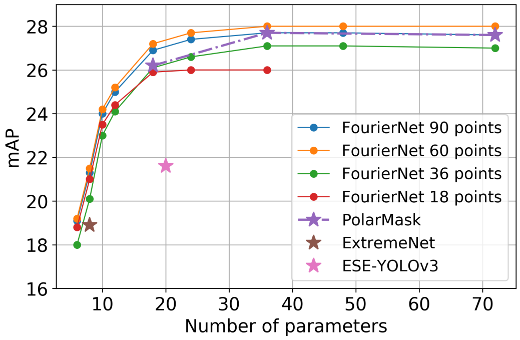


Fig. 3: mAP vs. number of parameters

was adopted, it is defined for the i_{th} feature point as

$$PC_i = \sqrt{\frac{\min(p_{0,i}, p_{1,i}, ..., p_{N-1,i})}{\max(p_{0,i}, p_{1,i}, ..., p_{N-1,i})}}. \qquad (2)$$

It is trained using *binary cross entropy loss*. During inference, this value is multiplied with the classification score to keep the locations, which could possibly produce the best mask. We argue this metric would be low if the object's mask shape is not a circle, and since it is multiplied by the classification score, it will lower the probability of such objects being predicted. To overcome this problem, PolarMask introduced a *Centerness Factor* (CF), which is a hyperprameter added to the centerness to increase its value. To the best of our knowledge, this offset defeats the purpose of centerness, since it artificially raises the confidence and sometimes even exceeds 1. Moreover, it does not explicitly solve the problem of low centerness of non circular objects. Therefore, we introduce *Normalized Centerness* (NC) which is defined for a feature point i by

$$NC_i = \frac{PC_i}{PC_{\max}}, \qquad (3)$$

where PC_{max} is the polar centerness of the *center of mass* of an instance. The maximum value of the NC_i is clamped to 1, when center of mass does not have the highest polar centerness value.

As for the Fourier coefficients, they are passed through the IFFT to obtain the distances. Afterwards the distances are trained using the *Polar IOU loss* [8]. There is an optional branch to produce bounding boxes, and when used, they are trained using the *IOU loss* as in UnitBox [18]. The total loss equals the sum of all losses mentioned earlier.

4. EXPERIMENTS

The experiments were done on the COCO 2017 benchmark [19] which is divided into 118K training and 5K validation images. Unless otherwise stated, all the experiments were done using a pre-trained ResNet-50 [20] on ImageNet [21]. The networks were trained for 12 epochs with an initial learning rate of 0.01 and a mini-batch of 4 images. The learning rate was reduced by a factor of 10 at epochs 8 and 11. Stochastic gradient descent (SGD) with momentum (0.9) and weight decay (0.0001) was used for optimization. The input images were resized to 1280×768 pixels. Our work is based on PolarMask implementation [8], which uses the mmdetection framework [22].

4.1. Ablation study

4.1.1. Number of coefficients and contour points

Multiple FourierNets having 18, 36, 60, and 90 contour points and 36 complex coefficients (72 parameters) were trained. Figure 3 illustrates the progression of these networks' accuracy. Each curve in this plot was generated by testing a network multiple times. Each point in the curve refers to a test where a subset of the network output with the lowest frequency coefficients was used. The rest of the output tensor was padded with zeros as explained in section 3.1. It can be seen that for fewer parameters the effect of the number of contour points is negligible. As the number of parameters increase, the mAP sharply increases until around 18 parameters. After that range the mAP saturates, but the saturation value depends on the number of contour points. Generally, when using more contour points, the results are better which is validated by the result of 60 and 90 points networks and the slight advantage of the former falls in the range of a statistical margin. Figure 1 shows the effect of suppressing the higher frequencies. As we use less coefficients, we obtain smoother contours. When two coefficients are utilized, all the predictions become ellipses.

4.1.2. Coefficients regression (CR) vs. Differentiable shape decoding (DSD)

The lower frequency coefficients of a Fourier series have higher impact on the contour which can be inferred from the experiments in section 4.1.1. However, unweighted direct coefficient regression focuses equally on all coefficients during

Method	B.Bone	Rep.	Param.	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	FPS	GPU
two stage											
Mask RCNN [1]	RX-101	binary grid	784	37.1	60.0	39.4	16.9	39.9	53.5	5.6	1080Ti
PANet [2]	RX-101	binary grid	784	**42.0**	**65.1**	**45.7**	22.4	**44.7**	**58.1**	-	-
HTC [4]	RX-101	binary grid	784	41.2	63.9	44.7	**22.8**	43.9	54.6	2.1	TitanXp
one stage											
ESE-Seg-416 [9]	DN-53	shp. encoding	20	21.6	**48.7**	22.4	-	-	-	**38.5**	1080Ti
FourierNet-640	R-50	shp. encoding	20	**24.3**	42.9	**24.4**	6.2	25.9	42.0	26.6	2080Ti
ExtremeNet [11]	HG-104	polygon	**8**	18.9	44.5	13.7	**10.4**	20.4	28.3	3.1	-
FourierNet	RX-101	shp. encoding	**8**	**23.3**	**46.7**	**21.1**	10.3	**25.2**	**34.4**	**6.9**	2080Ti
EmbedMask [10]	R-101	binary grid	†	**37.7**	**59.1**	**40.3**	**17.9**	**40.4**	**53.0**	13.7	V100
YOLACT-700 [7]	R-101	binary grid	†	31.2	50.6	32.8	12.1	33.3	47.1	**23.4**	TitanXp
PolarMask [8]	RX-101	polygon	36	32.9	55.4	33.8	15.5	35.1	46.3	7.1*	2080Ti
FourierNet	RX-101	shp. encoding	36	30.6	50.8	31.8	12.7	33.7	45.2	6.9	2080Ti

Table 1: Comparision with state-of-the-art for instance segmentation on COCO test-dev. † The number of parameters are dependent on the size of the bounding box cropping the pixel embedding or mask prototype. * speed tested on our machines.

training, which is not optimal for shape decoders. On the contrary, when trained on contour points, the optimizer can inherently learn to prioritize the lower frequency coefficients of Fourier series and achieve *automatic weight balancing*. To verify this hypothesis, we trained a network with 18 coefficients and regressed the coefficients directly using a smooth L1 loss. It attained a mAP of 5.3, which is poor compared to a similar network trained on contour points (26 mAP from figure 3) and it validates our initial intuition. Moreover, the qualitative results of CR showed out of size masks which is a sign of errors in low frequency coefficients. ESE-Seg [9] also used CR and compared various function approximators. They reported the best performance on Chebyshev polynomials and argued that they have the best numerical distribution. However, we argue that if they had used optimized weights for Fourier coefficients during training, they would reach better performance. This was verified with our results using DSD because it does automatic weight balancing.

4.1.3. Polar centerness (PC) vs. Normalized centerness (NC)

Two networks with 90 contour points and 36 coefficients were trained on NC and PC. From the results in table 2, it can be seen that NC is better than PC, when the CF is set to zero, which means that it is generally a better centerness metric. However, to obtain the best performance, we still need to use the CF hyperparameter.

4.2. Comparison to state-of-the-art

A FourierNet-640 was trained with an image resolution of 640 x 360 to compare with a ESE-Seg-416 [9]. With a comparable backbone and same number of parameters, our result is 2.7 mAP higher and it runs in real-time. To compare to state-of-the-art methods, a FourierNet with a ResNeXt101 backbone [23], 90 contour points, and 36 coefficients was trained.

Method	CF	mAP	AP_{50}	AP_{75}
Polar	0	26.3	42.8	**27.7**
Normalized	0	**27.0**	**47.8**	26.9
Polar	0.5	**27.7**	46.4	**28.6**
Normalized	0.5	27.0	**47.9**	26.9

Table 2: Polar centerness vs. Normalized polar centerness

The quantitative results are shown in table 1 and an example of a prediction shown in figure 1. Compared to ExtremeNet [11], using 8 parameters, our results are better, especially with AP_L and AP_{75}, with an increase of 6.1 and 7.4, respectively. This means that our mask quality is superior when using few parameters. It can be seen that FourierNet is comparable to PolarMask when using the same number of parameters, with a small loss in speed due to the IFFT. However, the qualitative results are visually better with smoother contours. In general, our method is comparable to polygon methods, but falls short in performance compared to binary grid methods.

5. CONCLUSION

FourierNet is a single stage anchor-free method for instance segmentation. It uses a novel training technique with IFFT as a differentiable shape decoder. Moreover, since lower frequencies impact the mask the most, we obtained a compact representation of masks using only those low frequencies. Therefore, FourierNet outperformed all methods which use less than 20 parameters quantitatively and qualitatively. Even compared to object detectors, FourierNet can yield better approximations of objects using slightly more parameters. Our FourierNet-640 achieves a real-time speed of 26.6 FPS. We hope this method can inspire the use of differentiable decoders in other applications.

6. REFERENCES

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

[2] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768.

[3] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang, "Mask scoring r-cnn," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6409–6418.

[4] Kai Chen et al., "Hybrid task cascade for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 4974–4983.

[5] Youngwan Lee and Jongyoul Park, "Centermask: Real-time anchor-free instance segmentation," *arXiv preprint arXiv:1911.06667*, 2019.

[6] Weicheng Kuo, Anelia Angelova, Jitendra Malik, and Tsung-Yi Lin, "Shapemask: Learning to segment novel objects by refining shape priors," *arXiv preprint arXiv:1904.03239*, 2019.

[7] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee, "Yolact: Real-time instance segmentation," in *ICCV*, 2019.

[8] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo, "Polarmask: Single shot instance segmentation with polar representation," *arXiv preprint arXiv:1909.13226*, 2019.

[9] Wenqiang Xu, Haiyang Wang, Fubo Qi, and Cewu Lu, "Explicit shape encoding for real-time instance segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5168–5177.

[10] Hui Ying, Zhaojin Huang, Shu Liu, Tianjia Shao, and Kun Zhou, "Embedmask: Embedding coupling for one-stage instance segmentation," *arXiv preprint arXiv:1912.01954*, 2019.

[11] Xingyi Zhou, Jiacheng Zhuo, and Philipp Krahenbuhl, "Bottom-up object detection by grouping extreme and center points," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 850–859.

[12] Ze Yang, Yinghao Xu, Han Xue, Zheng Zhang, Raquel Urtasun, Liwei Wang, Stephen Lin, and Han Hu, "Dense reppoints: Representing visual objects with dense point sets," *arXiv preprint arXiv:1912.11473*, 2019.

[13] Justin Liang, Namdar Homayounfar, Wei-Chiu Ma, Yuwen Xiong, Rui Hu, and Raquel Urtasun, "Polytransform: Deep polygon transformer for instance segmentation," *arXiv preprint arXiv:1912.02801*, 2019.

[14] 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*, 2015, pp. 91–99.

[15] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He, "Fcos: Fully convolutional one-stage object detection," *arXiv preprint arXiv:1904.01355*, 2019.

[16] 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*, 2017, pp. 2117–2125.

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[18] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang, "Unitbox: An advanced object detection network," in *Proceedings of the 24th ACM international conference on Multimedia*. ACM, 2016, pp. 516–520.

[19] Tsung-Yi Lin et al., "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[22] Kai Chen et al., "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[23] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.