Sparse Shape Encoding for Topologically Improved Instance Segmentation

Keyi Liu
Centre for AI & Society
Centre for Vision Research
York University, Toronto, Canada
keyi93@yorku.ca

James H. Elder
Centre for AI & Society
Centre for Vision Research
York University, Toronto, Canada
jelder@yorku.ca

***Abstract*—Neurophysiological studies suggest that neurons in the intermediate visual area V4 of the primate cortex encode a sparse representation of object shape. While there are metabolic arguments for such sparse representations, there are also potential advantages for inference. Here we explore whether sparse shape encoding can yield benefits for object instance segmentation. Specifically, we encode 2D object shape using a Distance Transform Map (DTM) and learn a sparse basis for this representation. To make use of this encoding, we design and train an instance segmentation head to estimate the sparse coefficients representing the shape of each object, and then recover the estimated shape from the zero-crossing level set of the corresponding DTM. Our novel SparseShape encoding approach produces fewer topological errors than the state of the art, yields competitive mask AP on the COCO benchmark and exhibits superior generalization performance on the Cityscapes instance segmentation task. These results suggest that the sparse shape encoding observed in primate cortex has computational advantages that can benefit computer vision instance segmentation systems.**

***Keywords*-Instance Segmentation; Sparse Shape Encoding; Distance Transform; Convolutional Neural Networks;**

I. Introduction

Convolutional neural networks have achieved human-level performance on a number of tasks by building increasingly more global and complex conjunctions of features in higher layers [1]. In primate visual cortex, an analogous increase in receptive field complexity is observed in the intermediate and higher areas, one purpose of which is to compute representations of object shape, known to be crucial for object processing [2], [3]. While some early studies [4], [5] have highlighted similarities in shape processing in human cortex and deep networks, more recent work has suggested that networks trained to classify images rely less on global shape and more on texture and local shape features compared to humans [6], [7], [8]. This suggests that more explicit shape training might be needed to produce networks that are more sensitive to global shape, like the human brain. The regularization induced by more explicit shape modeling might include more stable segmentations and better generalization performance.

Instance segmentation can be treated as a local pixel-level task without any explicit attempt to represent shape. This *mask*-based approach is attractive in its simplicity and aligns well with the pixel-based intersection over union (IoU) method typically used to evaluate segmentation. Note, however, that such binary pixel representations are highly redundant and carry shape information only implicitly, i.e. knowing whether a single pixel is a 1 or a 0 says little about the global shape of the object.

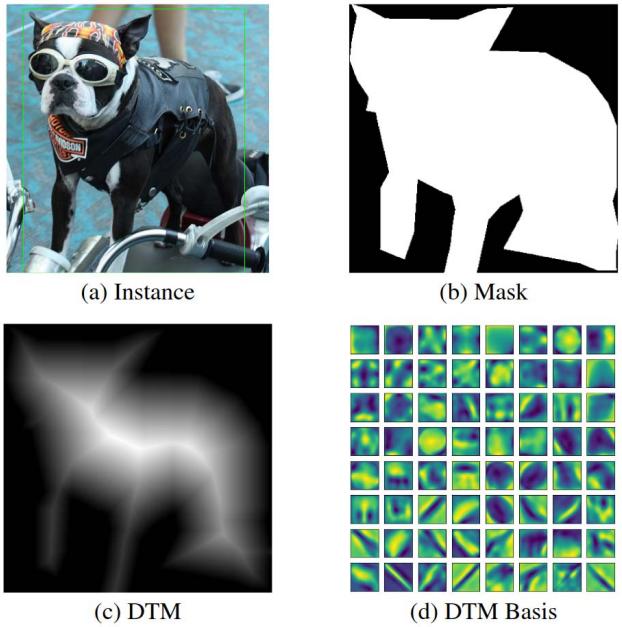


Figure 1: Sparse shape encoding on COCO [9] using distance transform maps (DTMs). (a) Object instance; (b) Ground-truth mask; (c) DTM representation; (d) Dictionary of learned DTM basis functions; A sparse linear combination of these basis functions approximates the DTM in (c).

An alternative is to represent a mask region by its bounding contour. This representation is potentially more efficient as it compresses the 2D region into a 1D contour. Unfortunately, this becomes challenging for higher-order instance topologies, e.g., segmentations with multiple connected components and/or holes. This problem can be addressed either by limiting the family of allowable shapes (e.g., PolarMask [10] assumes shapes are functions of polar angle without holes) or adding

2023 20th Conference on Robots and Vision (CRV) | 979-8-3503-4139-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/CRV60082.2023.00014

979-8-3503-4139-3/23/$31.00 ©2023 IEEE
DOI 10.1109/CRV60082.2023.00014

Authorized licensed use limited to: York University. Downloaded on September 01,2023 at 22:02:59 UTC from IEEE Xplore. Restrictions apply.

additional complexity (e.g., DeepSnake [11] uses an extra part detector to infer disconnected shapes).

We address these challenges by explicitly modeling shape in a novel representational framework that can easily handle complex topologies. Specifically, we replace the standard binary pixel mask with a distance transform map (DTM) representation of shape. Like the binary mask, this DTM encoding allows for the representation of complex topologies (as the union of zero-crossing level sets), but provides richer shape information at the pixel level: since each pixel value encodes the distance to the nearest point on the boundary, it quantifies information about the local degree of "insideness" (Fig. 1(c)).

As for the standard binary representation, the DTM representation is highly redundant. Inspired by neurophysiological evidence that the primate brain uses a sparse coding strategy to encode shape more efficiently [12], we learn a sparse DTM dictionary that allows a shape to be represented as a linear combination of a relatively small number of basis functions, and show that a network head can be trained to infer the coefficients of this representation. Once these coefficients are estimated, the object segmentation can be computed from the zero-crossing level set of the reconstructed DTM. To demonstrate the idea, we adopt the one-stage FCOS detector [13] which has been the detector of choice for most recent instance segmentation systems [14], [15], [16], [17], [18].

In summary, the major contributions of our paper are:

1) A distance transform encoding that improves network ability to estimate shape by encoding richer shape information at the local pixel level.
2) A sparse shape coding method (SparseShape) that encodes each shape as a linear combination of basis shapes, thus reducing redundancy.
3) A novel segmentation and shape reconstruction pipeline that includes a feature fusion strategy to estimate the sparse coefficients and uses these coefficients to estimate the distance transform as a weighted combination of sparse basis functions, identifies the shape as the zero-crossing of this distance transform, and then refines the result using a light-weight residual network (Figure 3) with mask quality scores.
4) A demonstration that this novel sparse shape method leads to substantial improvements in topological accuracy while maintaining competitive mask average precision (AP) over baselines on the MS COCO benchmark [9] and better generalization performance on the Cityscapes [19] traffic dataset.

II. Related Work

A. Region-Based Instance Segmentation

Mask region estimation is most commonly posed as a direct binary pixel-level classification [20], [21], [22], which can be elaborated to better handle variations in object scale and overlap or near-overlap of object instances. CenterMask [23], for example, separately estimates mask shape and scale, and employs a parallel saliency branch to facilitate the separation of neighbouring objects. CondInst [15] produces instance-aware filters to find segmentation masks also with the aid of a global semantic segmentation task.

Beyond direct mask estimation, a number of approaches attempt to decompose mask regions as linear combinations of prototypes. YOLACT [17], for example, employs a "protonet" to estimate region masks and a parallel prediction head to estimate coefficients on these prototypes for each detected object. BlendMask [18] adopts the segmentation bases for the entire image as in YOLACT, and employs attention maps to better capture fine details. In ShapeMask [24], ground-truthed training segmentations are first clustered to form a diverse set of shape priors, and a network is trained to estimate linear coefficients on these priors to estimate novel shapes.

Perhaps most similar to our approach is MEInst [14], which is based on a pre-learned principal component (PCA) subspace encoding of individual object region masks. The segmentation head of the network then estimates the coefficients of this PCA shape encoding. Note, however, that while PCA will minimize the squared error on the training dataset, it will not necessarily minimize error on unseen data, due to overfitting, and if the distribution of shapes is not joint Gaussian, the components will not necessarily be statistically independent, making it harder for the network to estimate coefficients from pixel data.

B. Contour-Based Instance Segmentation

Contour-based methods tackle the instance segmentation task by either directly estimating the coordinates of the vertices on the boundary or estimating a set of coefficients to encode the contour. PolarMask [10] and ESE-Seg [25] both constrain shapes to be functions of polar angle. While PolarMask directly estimates the length of rays from the polar center, ESE-Seg employs a lower-dimensional Chebyshev polynomial basis and trains a network head to estimate the coefficients for each detected object instance.

The polar shape constraint is unfortunately not satisfied for many shapes of interest (e.g., human bodies). DeepSnake [11] relaxes the polar assumption and estimates shapes by recurrent deformation of boundary vertices. Dense RepPoints [26] uses a distance transform to sample a dense set of points on either side of the object boundary to form a regression target. While enjoying some compression by targeting the boundary rather than the region mask, DeepSnake and Dense RepPoints do not apply any subspace encoding on the boundary and are thus still fairly high-dimensional.

C. Coefficient-Based Instance Segmentation

Coefficient-based approaches compress each shape to a lower-dimensional encoding space, and estimate a set of

Authorized licensed use limited to: York University. Downloaded on September 01,2023 at 22:02:59 UTC from IEEE Xplore. Restrictions apply.

coefficients in this compressed space. For example, ESE-Seg [25] encodes shape contours into a lower-dimensional Chebyshev polynomial space. MEInst [14] encodes each mask region into a PCA subspace. YOLACT [17] and BlendMask [18] learn both the subspace basis and the coefficients throughout a deep network.

D. Distance Transform

Distance transforms have been widely adopted for semantic segmentation to reduce blurring of boundaries [27], [28], [29]. Distance transforms can also be estimated in parallel with binary masks as an auxiliary task, and then fused to improve semantic segmentation [30], [31].

For instance segmentation, Bai [29] proposed a modified watershed method related to the distance transform. It uses a pre-trained semantic segmentation system to gate an image and estimate the gradient of the distance transform. The gradient map is then fed to a watershed network that generates a discretized energy map, where instances can be extracted as energy basins. Dense RepPoints [26] uses a distance transform to sample a dense set of points close to the object boundary as a regression target.

E. Relationship to Our SparseShape Approach

MEInst [14] used PCA to address redundancy of binary region-mask representations, while ShapeMask [24] uses a set of shape priors learned by K-means. Our SparseShape approach can be seen as intermediate between the two, in that our sparse components are expected to be more statistically independent and more articulated than the principal components used by MEInst, but are not necessarily complete shapes as in ShapeMask. We hypothesize that this may provide superior results through better regularization and handling of non-Gaussian shape distributions, while allowing more flexible reconstruction of novel shapes. While ShapeMask trains the network using a loss on the estimated mask, both MEInst and SparseShape use a loss directly on the coefficients, which we find to be more effective. Also, we apply this sparse coding approach to DTM representations of shape rather than binary masks to take advantage of the richer pixel-level shape information afforded by the distance transform.

YOLACT [17] and BlendMask [18] encode the joint shape of all objects in an image. We posit that this might limit generalization, since objects may occur in different combinations and different relative positions in novel datasets, and so we encode shapes individually. Unlike the deep watershed method of Bai [29], our SparseShape approach does not require a separate semantic segmentation network, and can be incorporated into one-stage object detection systems. Dense RepPoints [26] leverages a DTM only to sample points near the object boundary, whereas SparseShape directly uses the DTM as a shape representation space.

III. Method

A. Distance Transform Representation

Let G_i denote the set of all foreground pixels in the mask of an object i. The Distance Transform Map (DTM) of this object, $d_i(x)$, is defined as,

$$d_i(x) = \begin{cases} \min_{y \notin G_i} \|x - y\|_2 & \text{if } x \in G_i \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $\|x - y\|_2$ is the Euclidian distance between two pixels x and y. To improve segmentation performance, we modify this slightly by first resizing each mask into a fixed $m \times m$ pixel window, normalizing each DTM to have a maximum value of 1 and setting background pixels to -1 instead of 0 to sharpen the boundary:

$$d_i(x) = \begin{cases} \min_{y \notin G_i} \frac{\|x-y\|_2}{d_i^{max}} & \text{if } x \in G_i \\ -1 & \text{otherwise} \end{cases} \quad (2)$$

where d_i^{max} is the maximum distance in the original DTM.

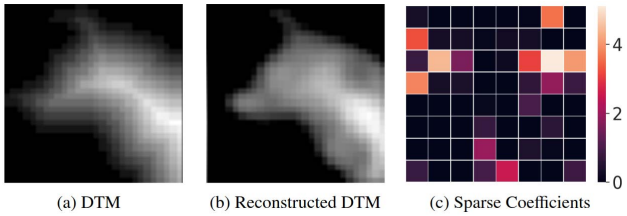
B. Sparse Coding

Sparse coding represents an input vector as a linear combination of a dictionary of basis vectors. To apply it to DTM encoding, we represent each $m \times m$ DTM as a vector $X_i \in \mathbb{R}^M$, where $M = m \times m$. The sparse representation $\hat{X}_i$ of X_i is then given by $\hat{X}_i = \mathbf{D}\alpha_i$ where the dictionary $\mathbf{D} \in \mathbb{R}^{M \times C}$, contains the C basis vectors and $\alpha_i \in \mathbb{R}^C$ is a vector of linear coefficients over these basis vectors.

We learn the dictionary $\mathbf{D}$ from a training dataset of N shapes by optimizing the following regularized objective:

$$\min_{\mathbf{D}, \alpha_i} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{M} \|X_i - \mathbf{D}\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \quad (3)$$

where $\lambda \in \mathbb{R}$ controls the degree of regularization. We solve for the dictionary $\mathbf{D}$ and coefficients α_i iteratively over the COCO training partition using online dictionary learning [32] and the Fast ISTA algorithm [33]. Fig. 2 shows an example sparse DTM reconstruction.



(a) DTM (b) Reconstructed DTM (c) Sparse Coefficients

Figure 2: A sparse shape coding example. (a) DTM representation $\mathbf{X_i}$ of a COCO instance mask; (b) Reconstruction $\mathbf{D}\alpha_i$ of this DTM based on a linear combination of basis DTMs from the dictionary learned using Eqn. 3 with $\lambda = 0.2$. (c) Magnitudes of the sparse coefficients for this instance, corresponding to the DTM basis functions shown in Fig. 1.

Authorized licensed use limited to: York University. Downloaded on September 01,2023 at 22:02:59 UTC from IEEE Xplore. Restrictions apply.

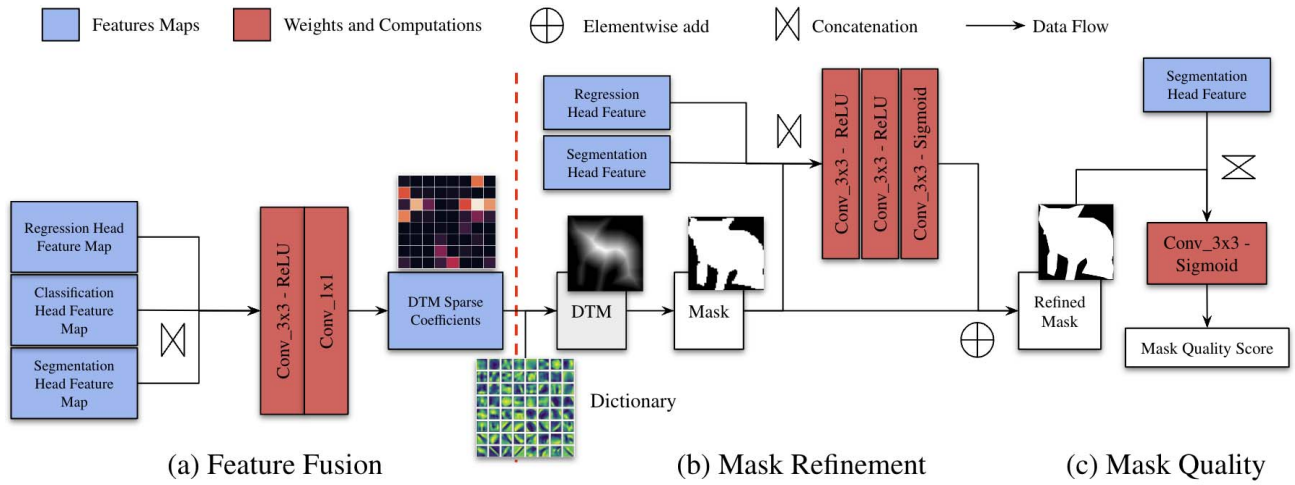


Figure 3: Feature fusion (a) mask refinement module (b) and mask quality score estimation (c). (a) The last feature map from the classification and regression head are concatenated with the segmentation head features to estimate the sparse coefficients of the DTM; (b) The predicted DTM is decoded by linearly combining the estimated sparse coefficients and the pre-learned dictionary. The zero-crossing level set of the DTM forms the initial estimate of the shape. A residual block refines this estimate with guidance from concatenated regression and segmentation features; (c) A lightweight quality scoring layer (3x3 convolution with Sigmoid activation) takes as input the segmentation head features and the refined mask and estimates the IoU between the estimated and ground truth mask.

C. Network Architecture

Our network consists of a one-stage object detector with regression and classification heads, a new segmentation head that incorporates feature fusion across heads and a residual module for mask refinement.

1) One-stage Object Detector: One-stage object detectors have demonstrated superior performance over two-stage methods while being faster and conceptually simpler [34], [13], [35]. We adopt the FCOS detector [13] employed by most recent instance segmentation approaches (e.g., [14], [10], [15], [18], [16]).

2) Instance Segmentation: Instance segmentation consists of two cascaded stages: A coarse segmentation stage with feature fusion (Fig. 3(a)), and a mask refinement stage (Fig. 3(b)) that also estimates a mask quality score (Fig. 3(c)).

Coarse Segmentation Stage: We employ a segmentation head that produces a feature map in parallel with classification and bounding box regression maps. In a standard single-stage instance segmentation framework [14], [10], [15] these three heads do not share features, which ignores dependencies between class, bounding box and segmentation. To address this issue, we fuse features from all three heads with a small network consisting of a 3-by-3 convolutional fusion layer and a 1-by-1 convolutional output layer that produces estimates of the DTM sparse coefficients (Fig. 3(a)).

Mask Refinement Stage: The mask refinement uses these coefficients and the learned dictionary to generate an estimate of the instance DTM and mask as the zero-crossing level set of this DTM. To refine the details of this mask, we employ a lightweight module with a single residual block (Figure 3(b)) that includes two regular and one deformable convolution. This 3-layer residual block accepts as input the current estimated mask and concatenated segmentation and bounding box regression features, and produces a refined mask estimate.

3) Mask Quality Score: Scoring Mask R-CNN [21] incorporates a module that estimates a quality score for each predicted mask that can be used to prioritize more accurate mask predictions during inference.

We adopt this approach here, training a lightweight quality scoring layer (3x3 convolution with sigmoid activation) that takes as input the segmentation head features and the refined mask prediction and estimates the IoU between the estimated and ground truth mask. At inference, confidence scores are first weighted by these estimated IoU quality scores prior to non-maximum suppression, thus prioritizing objects with good estimated segmentations. (Fig. 3(c))

D. Training and Inference

Our training loss is a weighted combination of five components:

$$\mathcal{L} = \lambda_d \mathcal{L}_{det} + \lambda_c \mathcal{L}_c + \lambda_s \mathcal{L}_s + \lambda_m \mathcal{L}_m + \lambda_q \mathcal{L}_q \qquad (4)$$

We adopt the detection loss $\mathcal{L}_{det}$ from FCOS [13]. The sparse coefficient loss $\mathcal{L}_c$ is an L^2 norm on the error of the sparse coefficients. $\mathcal{L}_s$ is an L^1 regularization applied on the estimated coefficients to encourage sparsity. $\mathcal{L}_m$ is the dice loss $\mathcal{L}_{dice}$[36] on the refined mask, reweighted to promote

Authorized licensed use limited to: York University. Downloaded on September 01,2023 at 22:02:59 UTC from IEEE Xplore. Restrictions apply.

hard instances with low IoU:

$$\mathcal{L}_m = \frac{1}{\mathbf{IOU}(\hat{\mathcal{M}}, \mathcal{M}) + \sigma} \mathcal{L}_{dice}(\hat{\mathcal{M}}, \mathcal{M}) \tag{5}$$

where $\hat{\mathcal{M}}$ and $\mathcal{M}$ represent the estimated and ground truth segmentation masks, respectively and $\alpha = 0.25$ is a margin parameter that limits the upweighting to a factor of 4. $\mathcal{L}_q$ is the mask quality estimation loss, given by the L^2 norm of the difference between actual and estimated IoU between ground truth and estimated masks [37].

Note that segmentation losses are applied at two different points: The coefficient loss $\mathcal{L}_c$ and sparsity loss $\mathcal{L}_s$ are applied directly to the estimated sparse coefficients at the output of the feature fusion block, while the mask loss $\mathcal{L}_m$ and the mask quality estimation loss $\mathcal{L}_q$ are applied at the output of the mask refinement module. For all experiments, we set $\lambda_{det} = \lambda_m = \lambda_q = 1$, $\lambda_c = 2.5$, and $\lambda_s = 0.01$.

At inference, we follow standard pipelines [14], [13]. Detected object confidence scores are reweighted using the estimated mask quality scores prior to non-maximum suppression and each surviving object segmentation is rescaled and repositioned according to its estimated bounding box parameters.

IV. Experiments

Following standard practice [14], [15], [10], [17], [18], we train all models on the COCO *train2017* split, perform ablation experiments on the *val2017* split and compare with state-of-the-art (SOTA) methods on the *test-dev* split. Unless specified, we adopt the $1\times$ single-scale training strategy [13], [14] for the ablation experiments, and $3\times$ training strategy with multi-scale training when comparing to SOTA. To evaluate generalization performance, we fine-tune our model on the Cityscapes [19] dataset following the same training and testing method as CondInst [15] for fair comparison.

A. Implementation Details

ResNet-50 [38] with FPN [39], pre-trained on ImageNet, is used as a backbone, with all hyper-parameters adopted from FCOS [13] and MEInst [14]. Our segmentation heads is based on the design employed by MEInst. All the models are trained with stochastic gradient descent for 90K iterations under the $1\times$ training strategy (12 epochs) with an initial learning rate of 0.01 and a batch size of 16. The learning rate is reduced by a factor of 10 at iteration 60K and at 80K. The mask size is $m = 28$ pixels for all experiments, consistent with MEInst. For the $3\times$ training (36 epochs), we match the protocol employed by other SOTA methods [15], [18], [14].

Based on a ResNet-50 backbone, our system runs at 9.6 fps on an Nvidia GTX 1080Ti GPU, faster than Mask RCNN [37] and comparable to other one-stage methods [14], [17], [18]. See supplementary file for more details.

B. Ablation Study

Sparse Coding Parameters: We first study the influence of dictionary size C and sparsity constant λ. Reconstruction error ($1-$ mean IoU, given the ground truth shape as a target) decreases for larger dictionaries, since larger dictionaries are more expressive, and for smaller sparsity constants, since this places more emphasis on the fidelity term of the loss (Fig. 4(a)). The number of active components (defined as coefficient magnitudes greater than 0.001) decreases as the sparsity constant is increased (Fig. 4(b)).

While reconstruction error given the ground truth as target monotonically decreases with dictionary size, this is not the case when coefficients must be estimated by the network, since larger dictionaries might require more network capacity to avoid underfitting, or more training data to avoid overfitting. When fixing the sparsity constant to $\lambda = 0.2$ we find that Mask AP on the validation data peaks at an intermediate dictionary size of $C = 128$ components (Fig. 4(c)). We use these parameters for our remaining experiments.

Shape Representation and Encoding Method: Table I compares AP performance for mask and DTM shape representation methods as well as PCA versus sparse coding in a 2×2 crossed design. We also assess the impact of the dimensionality of the encoding (64 vs 128) for the mask representation case. We do not include feature fusion or mask refinement in these ablation experiments. Overall, we find that the DTM representation improves performance over the mask representation, and sparse coding outperforms PCA.

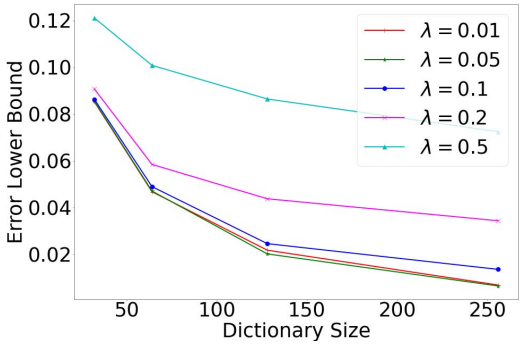
Table I: Evaluation of Mask vs DTM shape representation and PCA vs sparse encoding.

Representation	Encoding	Dict. size C	AP
Mask	PCA [14]	64	0.241
		128	0.245
	Sparse	64	0.243
		128	0.249
DTM	PCA	128	0.250
	Sparse	128	0.257

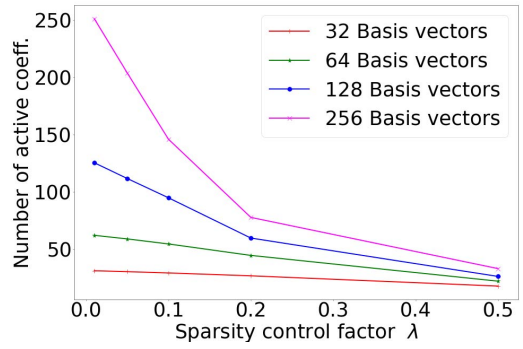
Feature Fusion & Mask Refinement: Table II shows the effectiveness of the proposed feature fusion and mask refinement strategies. Features from both classification and bounding box regression make a contribution to instance segmentation performance, and our lightweight mask refinement module adds an additional boost. We include both of these strategies in our final system.

Mask Quality Modulation: Table III shows the effectiveness of using mask quality during both training and inference. Using the estimated mask quality at inference leads to an improvement of 0.2% on mask AP and incorporating the actual mask IoU during training to upweight hard examples (Eqn. 5) provides an additional 0.4% boost. We incorporate both in our complete SparseShape system.

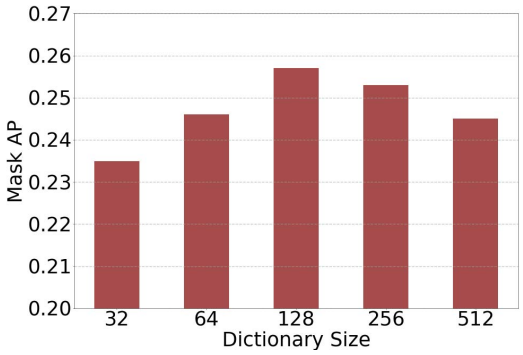
Authorized licensed use limited to: York University. Downloaded on September 01,2023 at 22:02:59 UTC from IEEE Xplore. Restrictions apply.



(a) Lower bound on error



(b) Sparsity



(c) Actual network performance

Figure 4: Effects of dictionary size C (number of basis shapes) and sparse coding regularization constant λ, evaluated on COCO [9] validation data. (a) Lower bound on error, defined as $1-$ mean IoU, given the ground truth segmentation as a target. (b) Sparsity: number of active coefficients (c) Actual network performance (mask AP) for $\lambda = 0.2$.

Table II: Effectiveness of our feature fusion and mask refinement strategies.

Classif.	Bound. box	Refinement	Mask AP
			0.257
✓			0.260
	✓		0.259
✓	✓		**0.263** (+0.6%)
✓	✓	✓	**0.267** (+0.4%)

Table III: Evaluation of the mask quality estimation for training and inference.

System	Mask AP
Baseline	0.267
+ quality (inference [21])	0.269
+ quality (training + inference)	**0.273** (+0.4%)

C. Comparison to State-of-the-Art

1) Mask AP: Table IV compares mask AP performance of our DTM-based SparseShape (SS) instance segmentation system with SOTA one-stage systems on COCO. We organize the table by the class of representation (contour-based or region-based). SparseShape outperforms prior contour-based approaches and the other region-based approaches that use a prior shape dictionary (ShapeMask [24] and MEInst [14]). SparseShape also out-performs YOLACT [17], which estimates segmentations as linear combinations of 'proto-masks'. Comparing to more general region-based approaches, SparseShape out-performs all methods but BlendMask [18] and CondInst [15].

In sum, SparseShape performs at or near the SOTA in terms of mask AP. However, we believe the main advantages of the approach relate to generalization and topological accuracy, as detailed below.

2) Generalization: Explicit incorporation of the statistics of shapes through our shape dictionary has the potential to serve as an inductive prior, which can lead to better generalization. To test this, we compare generalization of SparseShape, CondInst [15] and Mask-RCNN [37] to the Cityscapes instance segmentation dataset [19]. We follow the same procedure as CondInst [15] to transfer our system from COCO to Cityscapes, employing a COCO-trained sparse shape dictionary (see supplementary file for an analysis of how the dictionary employed affects generalization performance). Table V shows that SparseShape outperforms CondInst on both of the two key evaluation metrics: by 0.4% on AP and 2.1% on $AP_{0.5}$. This improvement derives in part from better performance on rarer categories like truck, bus and train, where the inductive prior embodied by our sparse shape dictionary becomes more important. See Supplementary file for details and qualitative results.

Authorized licensed use limited to: York University. Downloaded on September 01,2023 at 22:02:59 UTC from IEEE Xplore. Restrictions apply.

Table IV: Comparison to SOTA on the COCO *test-dev 2017* split. SS *Basic* is our core sparse shape segmentation head; SS *Complete* integrates feature fusion, mask refinement and mask quality score estimation. *R50/101FPN* denotes ResNet-50/101 FPN backbones; *HG-104* refers to the double-stacked hourglass architecture.

Class	System	Backbone	# Train Epochs	Mask mAP
Contour	ESE-Seg[25]	R50FPN	36	0.307
	DeepSnake[11]	HG104	160	0.303
	PolarMask[10]	**R101FPN**	**36**	**0.321**
Region	PANet[40]	R50FPN	36	0.366
	CM-SAM[16]	R50FPN	36	0.347
	CM-PR[23]	R50FPN	36	0.361
	SOLO[41]	R50FPN	36	0.368
	BlendMask[18]	R50FPN	36	0.370
	CondInst[15]	**R50FPN**	**36**	**0.378**
	SS (Basic)	R50FPN	36	0.353
	SS (Complete)	R50FPN	36	0.368
	YOLACT[17]	R101FPN	48	0.312
	ShapeMask[24]	R101FPN	36	0.374
	MEInst[14]	R101FPN	36	0.339
	SS (Complete)	**R101FPN**	**36**	**0.379**

Table V: Generalization performance on the Cityscapes[19] instance segmentation benchmark.

Method	AP	$AP_{0.5}$
Mask-RCNN[37]	0.365	0.622
CondInst[15]	0.369	0.632
Ours	**0.373**	**0.653**

3) Topological Accuracy: Mask AP measures overlap between estimated and ground-truth masks but does not tell us whether the topology of the estimated mask (e.g., number of connected components and holes) matches the topology of the ground truth. Arguably, getting the topology right can be even more important than getting the exact overlap for downstream tasks. Here we compare the topological accuracy of our SparseShape approach with leading SOTA one-stage segmentation approaches (SOLO [41] , CondInst [15] and BlendMask [18]).

Topological MAE.: As a first measure of topological performance we assess how closely the number of connected components in the estimated mask matches the number in the ground truth mask. Specifically, we identify, for each ground truth mask in the COCO validation set, the estimated mask with correct label and highest IoU over a specified threshold. We define the *Topological MAE* (T-MAE) as $\mathbb{E}\left[\left|\hat{k}-k\right|/k\right]$, where $\hat{k}$ and k are the number of connected components in the estimated and ground truth masks, respectively. (If no estimate meets the class and IoU criteria, we ignore the object - missed detections are already assessed by the mask AP measure.)

Table VI shows that SparseShape vastly outperforms (by a factor of 5-12) competing SOTA systems in identifying the correct number of connected components. This staggering difference is due primarily to an over-segmentation of objects by these competing methods, which tend to fragment a single object into multiple pieces (See Supplemenbtary file for a more detailed breakdown.) We note that the method with highest AP (CondInst [15]) also commits the most topological errors, suggesting that there may be a direct trade-off between pixel-level mask AP and topological performance.

Table VI: Topological MAE (error in number of connected components in mask), evaluated on the COCO [9] validation set.

System	Topological MAE	
	IoU@0.5	IoU@0.75
SOLO[41]	0.523	0.396
CondInst[15]	0.686	0.559
BlendMask[18]	0.519	0.431
SparseShape Basic	0.109	0.049
SparseShape Basic + FeatFusion	**0.098**	**0.048**
SparseShape Complete	0.101	0.055

Table VI also shows that the principal basis for the superior topological performance of SparseShape is our core sparse DTM shape coding network (Basic). Feature fusion improves topological performance slightly, while mask refinement slightly lowers topological performance, possibly due to incorrect fragmentation of some objects into multiple components.

Table VII breaks down COCO ground truth objects into four disjoint topological categories: 1C: Objects with a single connected component and no holes; 2C: Objects with two connected components and no holes; $\geq$ 3C: Objects with three or more connected components and no holes; Hole: Objects with holes. We see that the vast majority of COCO objects (89.1%) consist of a single connected component with no holes. Competing SOTA methods greatly underestimate the number of single-component objects and overestimate the number of objects with 2 or more components and/or holes. While biased slightly in the opposite direction, overall SparseShape better matches the distribution of ground truth topologies, as measured by KL-divergence.

D. Qualitative Results

Figure 5 provides a qualitative comparison of SparseShape segmentations with SOTA methods CondInst [15] and BlendMask [18] on 4 example images from the COCO validation set, with a focus on objects with diverse ground truth topologies (one connected component, two connected components, three connected components, and holes). While SparseShape correctly identifies the ground truth topologies, CondInst and Blendmask commit major topological errors. In Fig. 5(a), both CondInst and BlendMask generate multiple

Authorized licensed use limited to: York University. Downloaded on September 01,2023 at 22:02:59 UTC from IEEE Xplore. Restrictions apply.

Table VII: Topological distribution of ground truth and estimated masks on the COCO [9] validation set. KL is the KL-divergence from ground truth.

	1C	2C	$\geq 3C$	Hole	KL
Ground Truth	89.1%	7.8%	2.8%	0.2%	-
SOLO[41]	74.8%	12.3%	11.0%	1.8%	0.113
CondInst[15]	80.6%	8.2%	9.3%	1.8%	0.072
BlendMask[18]	75.0%	10.2%	11.5%	3.2%	0.145
SparseShape	96.4%	2.7%	0.2%	0.7%	**0.049**

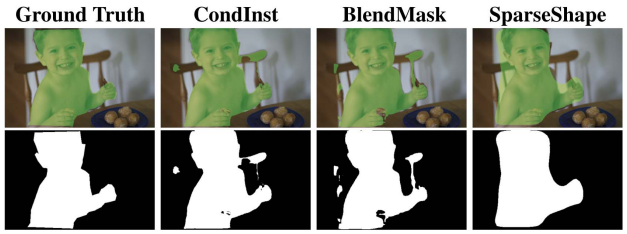
extraneous segments and holes. In Fig. 5(b), both CondInst and Blendmask miss one component and in (c) they both incorrectly join two components. In Fig. 5(d), CondInst fails to estimate the correct hole and BlendMask breaks it into two holes.

V. Conclusions & Future Work

Inspired by evidence for sparse-shape encoding in primate visual cortex, we have developed and studied a novel system for instance segmentation called SparseShape, which reformulates the instance segmentation problem as the estimation of coefficients for a pre-learned dictionary of sparse DTM components, from which the instance masks can be reconstructed. We show that this system provides superior generalization and topological performance while maintaining competition mask mAP compared to SOTA methods. Future work could involve integrating the high-resolution global semantic map employed by methods such as CondInst [15] in order to improve the articulation of segmentations, and investigating whether estimating the sparse shape coefficients indeed forces the network to rely more on global shape information for object classification.

Acknowledgment

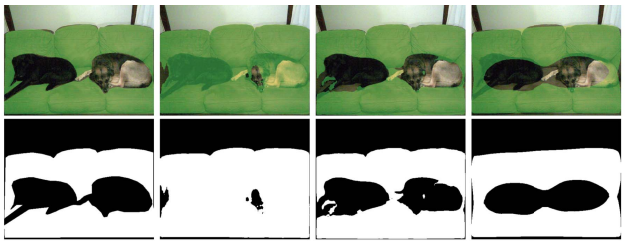
This research was supported by the ORF-RE project, Intelligent Systems for Sustainable Urban Mobility (ISSUM), the Vision: Science to Applications (VISTA) program, and by a sub-contract from the University of Toronto.



(a) Segmentation with 1 connected component

(b) Segmentation with 2 connected components

(c) Segmentation with 3 connected components



(d) Segmentation with holes

Figure 5: Comparison of SparseShape with SOTA (using a ResNet-50 FPN backbone) on diverse topologies from the COCO validation set

References

[1] N. Kriegeskorte, "Deep neural networks: A new framework for modeling biological vision and brain information processing," *Annual Review of Vision Science*, vol. 1, no. 15, 2015.

[2] A. Pasupathy and C. Connor, "Shape representation in area V4: Position-specific boundary conformation," *Journal of Neurophysiology*, vol. 86, pp. 2505–2519, 2001.

[3] C. Connor, S. Brincat, and A. Pasupathy, "Transformation of

Authorized licensed use limited to: York University. Downloaded on September 01,2023 at 22:02:59 UTC from IEEE Xplore. Restrictions apply.

shape information in the ventral pathway," *Current Opinion in Neurobiology*, vol. 17, no. 140-147, 2007.

[4] J. Kubilius, S. Bracci, and H. P. O. de Beeck, "Deep neural networks as a computational model for human shape sensitivity," *PLoS Computational Biology*, vol. 12, no. 4, 2016.

[5] S. Ritter, D. G. Barrett, A. Santoro, and M. M. Botvinick, "Cognitive psychology for deep neural networks: A shape bias case study," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2017, pp. 2940–2949.

[6] N. Baker, H. Lu, G. Erlikhman, and P. J. Kellman, "Deep convolutional networks do not classify based on global object shape," *PLOS Computational Biology*, vol. 14, no. 12, p. e1006613, 2018.

[7] J. Elder, "Shape from contour: Computation and representation," *Annual Review of Vision Science*, vol. 4, pp. 423–450, 2018.

[8] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[9] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft coco: Common objects in context," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 740–755, 2014.

[10] E. Xie, P. Sun, X. Song, W. Wang, X. Liu, D. Liang, C. Shen, and P. Luo, "PolarMask: Single shot instance segmentation with polar representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 12 190–12 199.

[11] S. Peng, W. Jiang, H. Pi, X. Li, H. Bao, and X. Zhou, "Deep snake for real-time instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8530–8539.

[12] E. Carlson, R. Rasquinha, K. Zhang, and C. Connor, "A sparse object coding scheme in Area V4," *Current Biology*, vol. 21, pp. 288–293, 2011.

[13] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," *Proceedings of the International Conference on Computer Vision (ICCV)*, pp. 9626–9635, 2019.

[14] R. Zhang, Z. Tian, C. Shen, M. You, and Y. Yan, "Mask encoding for single shot instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 223–10 232.

[15] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 282–298.

[16] Y. Lee and J. Park, "Centermask : Real-time anchor-free instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 13 903–13 912.

[17] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: real-time instance segmentation," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019, pp. 9156–9165.

[18] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan, "BlendMask: Top-down meets bottom-up for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8570–8578.

[19] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3213–3223.

[20] L.-C. Chen, A. Hermans, G. Papandreou, F. Schroff, P. Wang, and H. Adam, "Masklab: Instance segmentation by refining object detection with semantic and direction features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4013–4022.

[21] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring R-CNN," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 6402–6411.

[22] J. Dai, K. He, Y. Li, S. Ren, and J. Sun, "Instance-sensitive fully convolutional networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, pp. 534–549.

[23] Y. Wang, Z. Xu, H. Shen, B. Cheng, and L. Yang, "CenterMask: Single shot instance segmentation with point representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9313–9321.

[24] W. Kuo, A. Angelova, J. Malik, and T.-Y. Lin, "ShapeMask: Learning to segment novel objects by refining shape priors," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019, pp. 9206–9215.

[25] W. Xu, H. Wang, F. Qi, and C. Lu, "Explicit shape encoding for real-time instance segmentation," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019, pp. 5167–5176.

[26] Z. Yang, Y. Xu, H. Xue, Z. Zhang, R. Urtasun, L. Wang, S. Lin, and H. Hu, "Dense RepPoints: Representing visual objects with dense point sets," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[27] H. Kervadec, J. Bouchtiba, C. Desrosiers, E. Granger, J. Dolz, and I. B. Ayed, "Boundary loss for highly unbalanced segmentation," *International Conference on Medical Imaging with Deep Learning*, vol. 102, pp. 285–296, 2019.

[28] D. Karimi and S. E. Salcudean, "Reducing the hausdorff distance in medical image segmentation with convolutional neural networks," *IEEE Transactions on Medical Imaging*, vol. 39, no. 2, pp. 499–513, 2020.

[29] M. Bai and R. Urtasun, "Deep watershed transform for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2858–2866.

Authorized licensed use limited to: York University. Downloaded on September 01,2023 at 22:02:59 UTC from IEEE Xplore. Restrictions apply.

[30] F. Navarro, S. Shit, I. Ezhov, J. Paetzold, A. Gafita, J. C. Peeken, S. E. Combs, and B. H. Menze, "Shape-aware complementary-task learning for multi-organ segmentation," *Machine Learning in Medical Imaging*, pp. 620–627, 2019.

[31] S. Dangi, C. A. Linte, and Z. Yaniv, "A distance map regularized CNN for cardiac cine MR image segmentation," *Medical Physics*, vol. 46, no. 12, 2019.

[32] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2009, pp. 689–696.

[33] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. on Imaging Sciences*, vol. 2, pp. 183–202, 2009.

[34] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9756–9765.

[35] X. Zhou, D. Wang, and P. Krahenbuhl, "Objects as points," *arXiv 1904.07850*, 2019.

[36] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proceedings of the International Conference on 3D Vision (3DV)*, 2016, pp. 565–571.

[37] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017, pp. 2961–2969.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[39] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.

[40] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8759–8768.

[41] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li, "SOLO: segmenting objects by locations," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 649–665.

Authorized licensed use limited to: York University. Downloaded on September 01,2023 at 22:02:59 UTC from IEEE Xplore. Restrictions apply.