Training-Free Dataset Pruning For Instance Segmentation

Yalun Dai[1,2,3], **Lingao Xiao**[1,2,4], **Ivor W. Tsang**[1,2,3], **Yang He**[1,2,4*]
[1]CFAR, Agency for Science, Technology and Research, Singapore
[2]IHPC, Agency for Science, Technology and Research, Singapore
[3]Nanyang Technological University, [4]National University of Singapore
daiy0018@e.ntu.edu.sg, xiao_lingao@u.nus.edu
{Ivor_Tsang, He_Yang}@cfar.a-star.edu.sg

Yalun Dai

CREATING GROWTH, ENHANCING LIVES

Dataset Pruning

Pruning

Full CIFAR-100 Dataset
(size = 50,000)

Pruned Dataset
(size = 10,000)

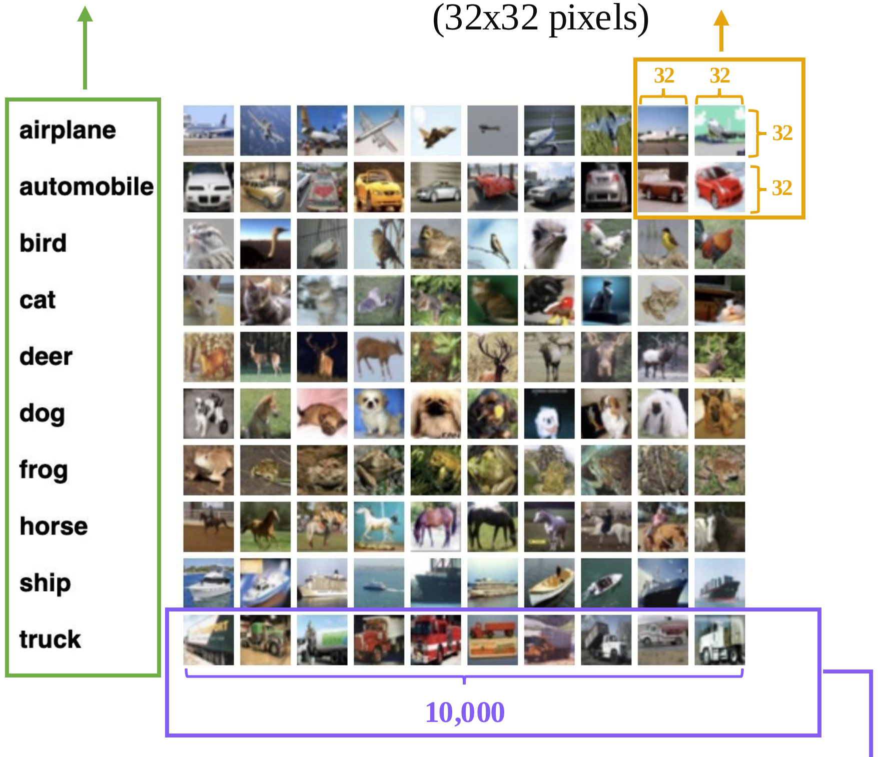
Similar training accuracy

Dataset Pruning has **NOT** been explored in **instance segmentation** task.

Classification VS **Instance Segmentation**

1. **Image-level** Annotations
(one-hot label)

2. **Uniform** Image Resolution
(32x32 pixels)



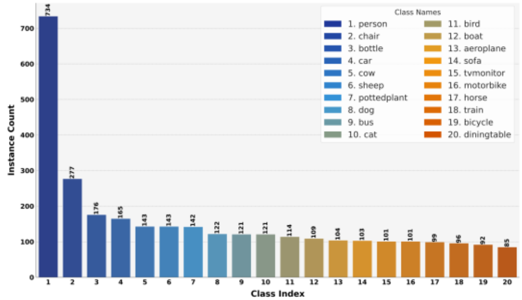
3. **Balanced** class distribution
(each class with 10,000 images)

1. **Pixel-level Annotations**

Lable:

2. **Variable Instance Areas**

3. **Class Imbalance**

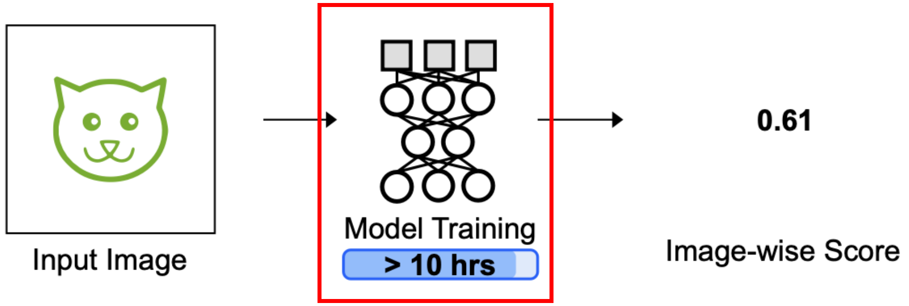


(Distribution of number of class instances on VOC.)

Fundamental issues inherent in **existing methods**.

1) Existing Dataset Pruning on Classification.

a. **Infeasible** for instance segmentation.

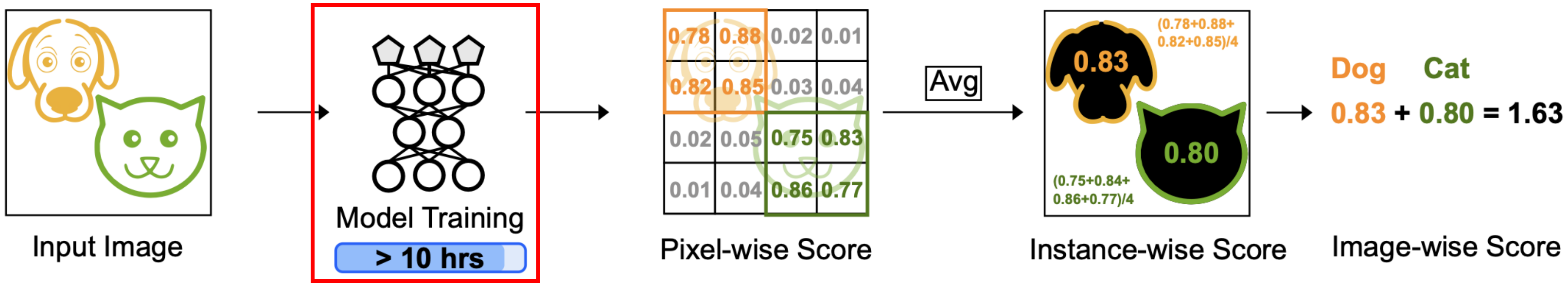


❌ **Infeasible!**

---

2) Our Adaptation on Instance Segmentation.

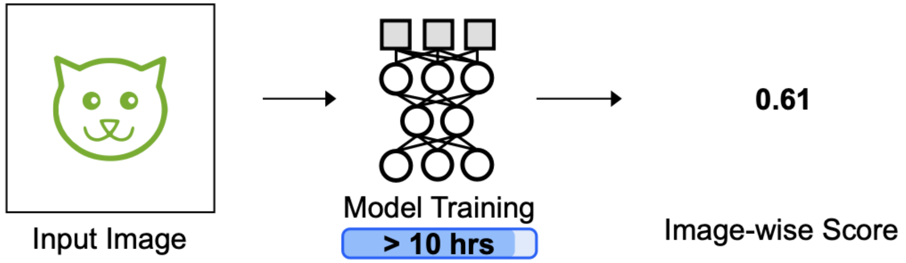
a. **Slow** speed.
b. **Limited** generalization.
c. **Impractical** for segmentation.



❌ Feasible but still **Inefficient!**

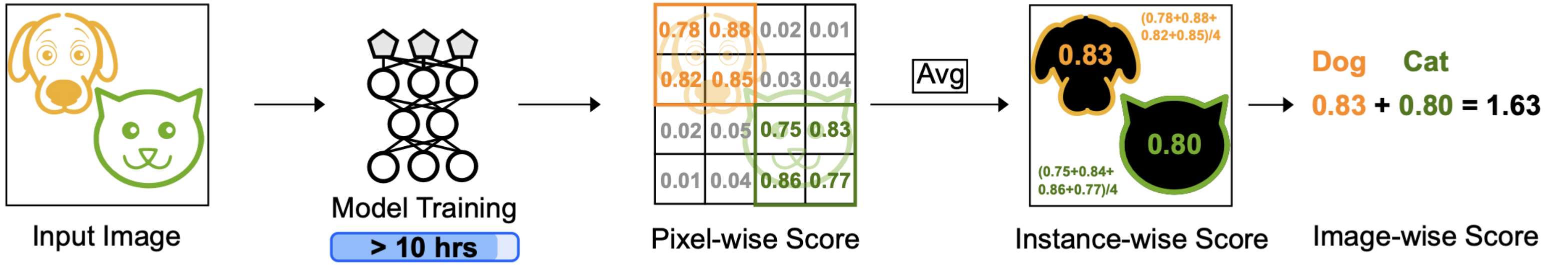
1) Existing Dataset Pruning on Classification.

a. **Infeasible**.

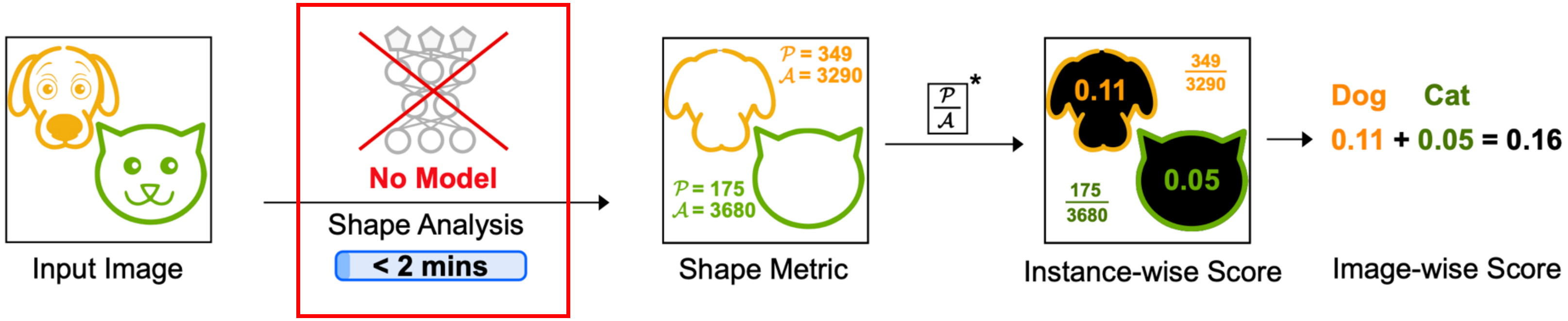


2) Our Adaptation on Instance Segmentation.

a. Feasible but still **Inefficient**.



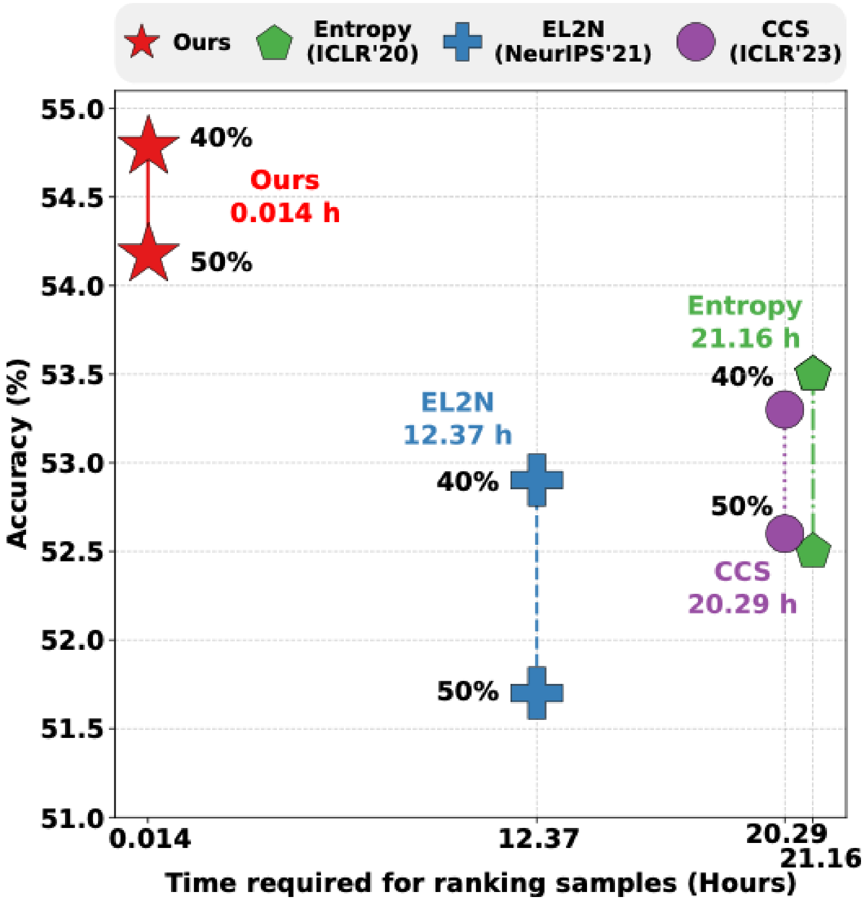
3) Our Training-Free Dataset Pruning (TFDP) on Instance Segmentation.



a. **Faster** speed.
b. **Better** generalization.

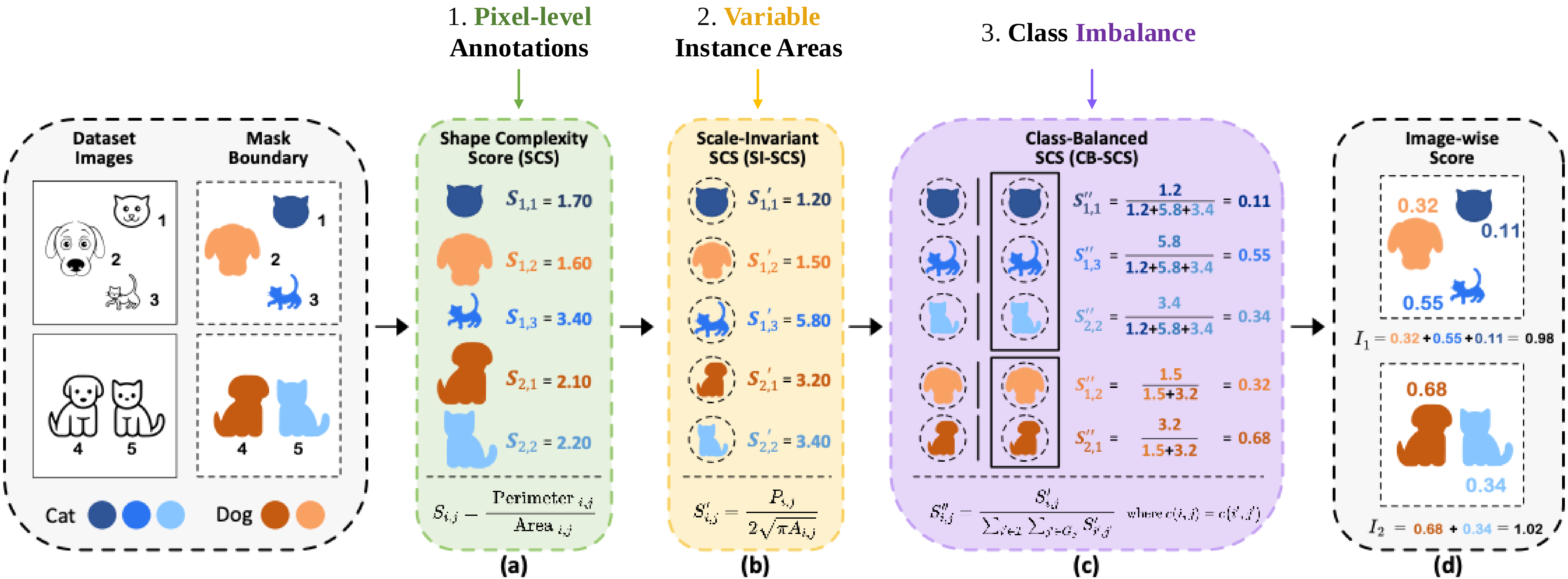
Efficient!

To the best of our knowledge, we are the **first** to introduce a **training-free** dataset pruning framework for instance segmentation.

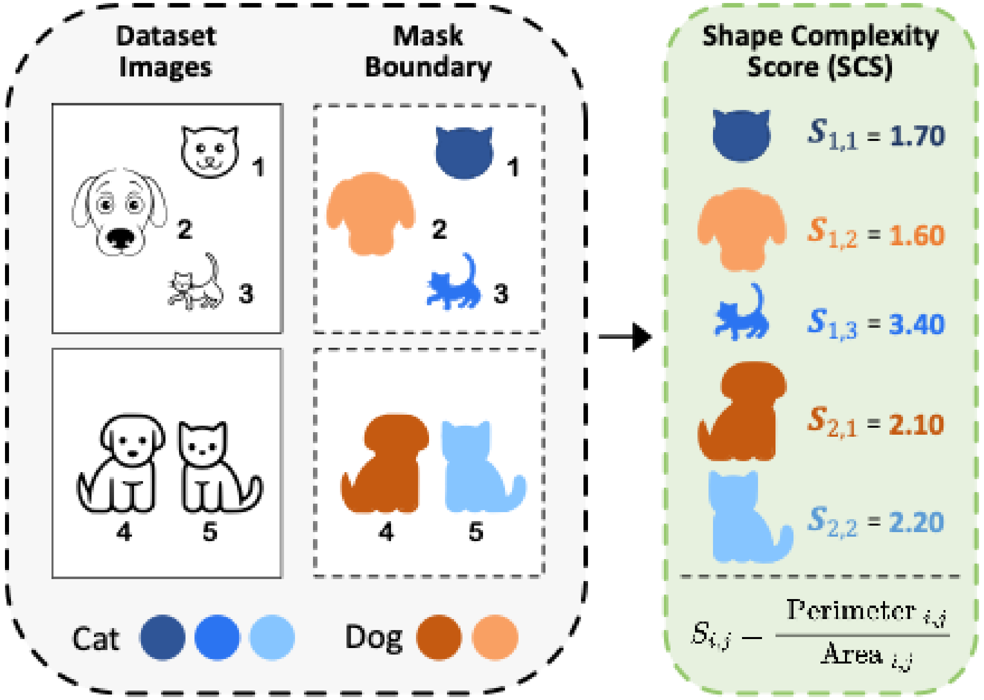


1. **Better** performance.
2. **Better** generalizability.
3. **Faster** speed (average of **1349×** on COCO).

Training-Free Dataset Pruning (TFDP) Framework



Shape Complexity Score (SCS)



(a)

1. Pixel-level Annotations

Lable:
cat
dog

Mask Preparation

↓

Contour Extraction

↓

Perimeter Calculation $P_{i,j}$

↓

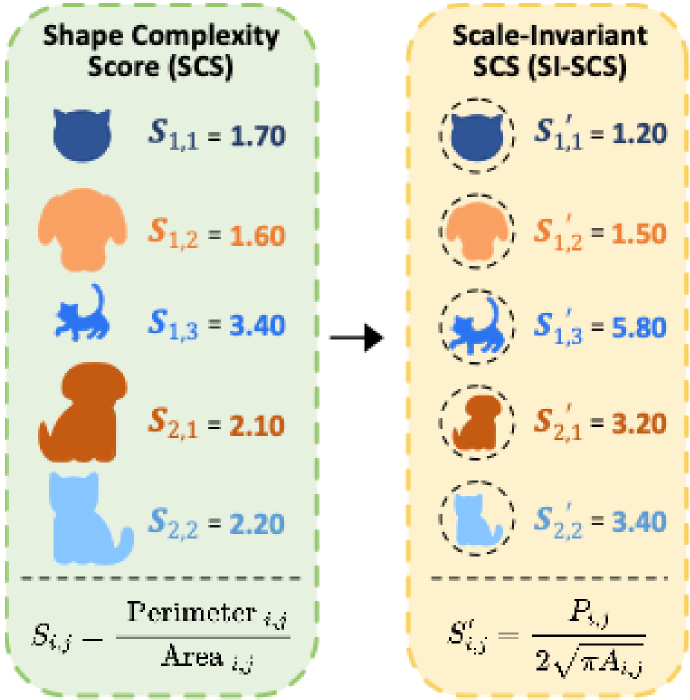
Area Calculation $A_{i,j}$

↓

Shape Complexity Score Computation

$$S_{i,j} = \frac{P_{i,j}}{A_{i,j}}$$

Scale- Invariant SCS (**SI-SCS**)



(a) (b)

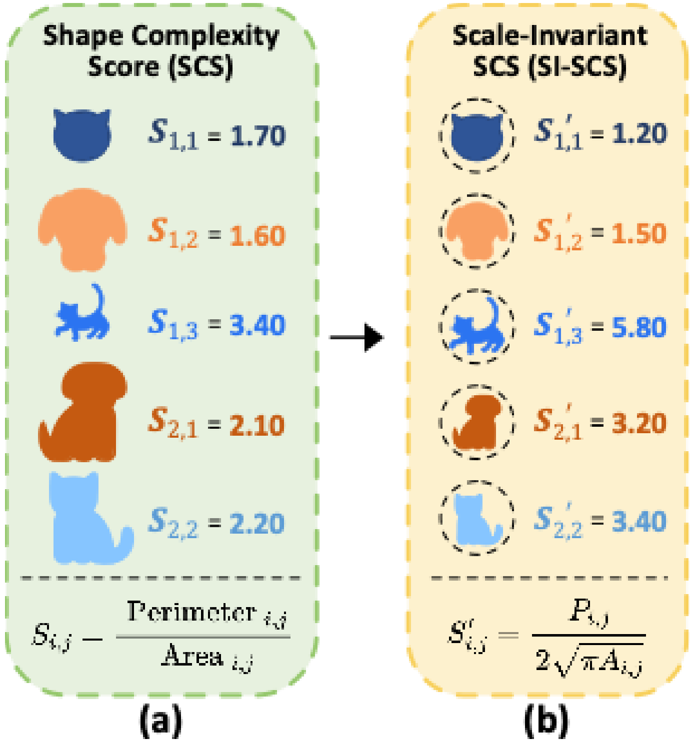
2. Variable Instance Areas

Scale bias Problem

For polygons of the same shape, scaling by factor **f** results in:

$$\frac{P'_{i,j}}{A'_{i,j}} = \frac{f\,P_{i,j}}{f^2\,A_{i,j}} = \frac{P_{i,j}}{f\,A_{i,j}}$$

Scale- Invariant SCS (**SI-SCS**)



Scale bias Solution

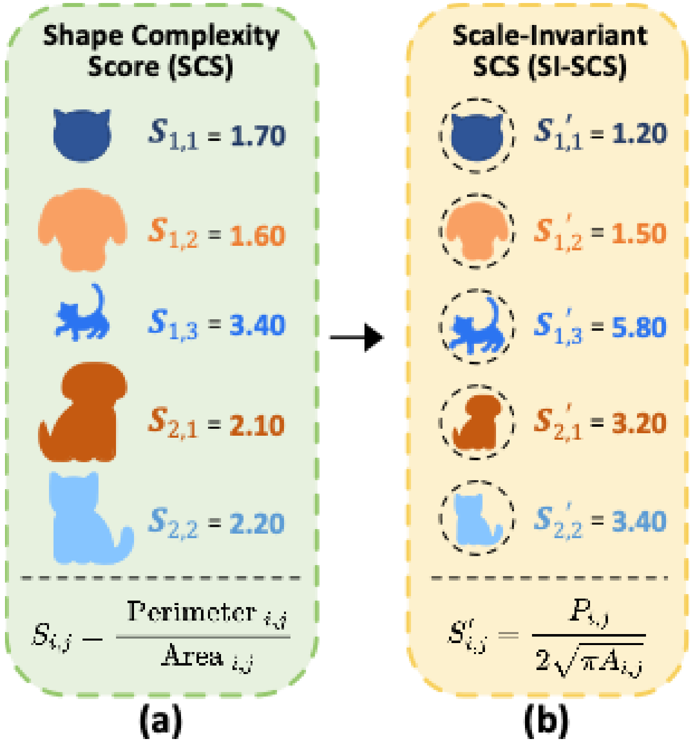
Normalize SCS using a circle,
the **SI-SCS** is defined as:

$$S'_{i,j} = \frac{S_{i,j}}{\boxed{S^{\circ}_{i,j}}} = \frac{P_{i,j}}{2\sqrt{\pi A_{i,j}}}$$

$$\boxed{S^{\circ}_{i,j} = \frac{P_{i,j}}{A_{i,j}} = \frac{2\pi r}{\pi r^2} = \frac{2}{r} = 2\sqrt{\frac{\pi}{A_{i,j}}}}$$

SCS of a circle

Scale- Invariant SCS (**SI-SCS**)



Scale bias Solution

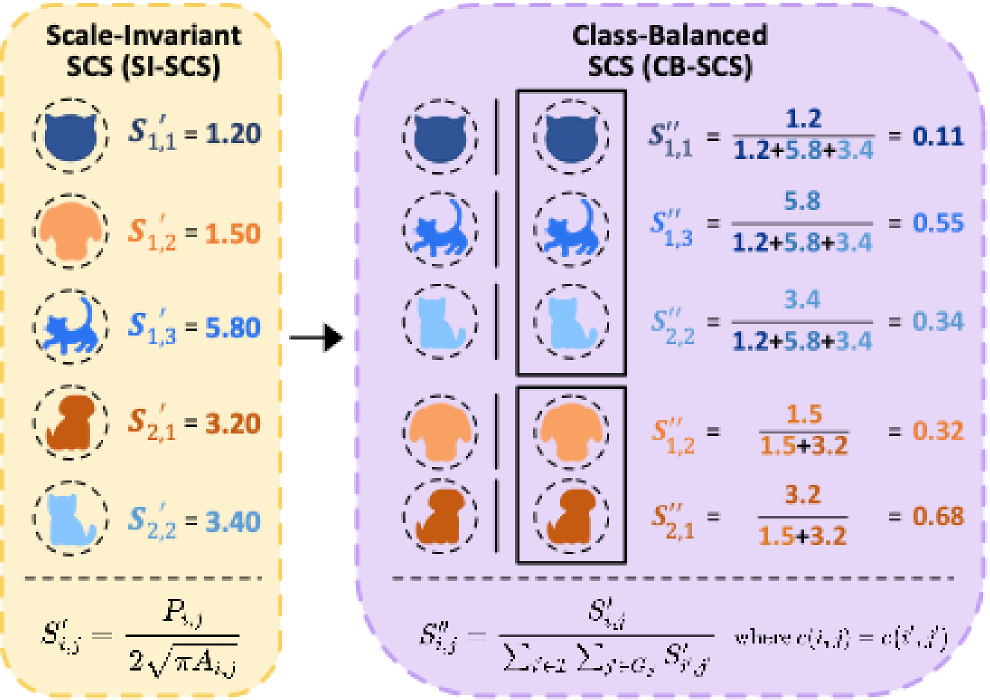
Normalize SCS using a circle,
the **SI-SCS** is defined as:

$$S'_{i,j} = \frac{S_{i,j}}{S^{\circ}_{i,j}} = \boxed{\frac{P_{i,j}}{2\sqrt{\pi A_{i,j}}}}$$

$$\boxed{S'_{i,j}(f) = \frac{P_{i,j} \times f}{2\sqrt{\pi \cdot (A_{i,j} \times f^2)}} = \frac{P_{i,j} \times f}{2\sqrt{\pi A_{i,j}} \times f} = \frac{P_{i,j}}{2\sqrt{\pi A_{i,j}}} = S'_{i,j}}$$

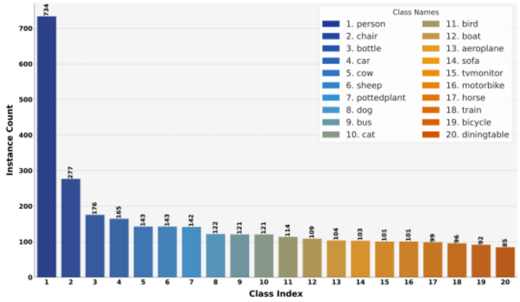
Scale invariant!

Class-**B**alanced SCS (**CB-SCS**)



(b) (c)

3. **Class** **Imbalance**



(Distribution of number of class instances on VOC.)

Class bias Solution

Normalize individual instance SI-SCS by the total score of all instances in that class across different images.

$$S''_{i,j} = \frac{S'_{i,j}}{\sum_{i'\in\mathcal{I}}\sum_{j'\in G_{i'}} S'_{i',j'}} \quad \text{where } c(i,j) = c(i',j')$$

Results on **COCO** dataset. (Mask R-CNN)

		mAP					AP_{50}					AP_{75}				
p	Time	0%	20%	30%	40%	50%	0%	20%	30%	40%	50%	0%	20%	30%	40%	50%
Random	-	34.2	33.6	32.1	31.1	30.8	55.2	54.5	52.8	51.1	51.0	36.5	35.6	34.1	33.2	32.7
Forgetting	20.29 h	-	33.1	32.3	31.4	30.4	-	54.2	53.4	52.2	51.2	-	35.2	34.3	33.4	32.1
Entropy	21.16 h	-	33.2	32.3	31.4	30.9	-	54.4	53.5	52.5	51.7	-	35.5	34.5	33.2	32.6
EL2N	12.37 h	-	33.4	32.1	31.2	30.5	-	54.5	52.9	51.7	51.2	-	35.6	34.2	33.2	32.0
AUM	20.29 h	-	33.5	32.4	31.5	31.0	-	54.6	53.3	52.4	51.7	-	35.5	34.7	33.4	32.8
CCS	20.29 h	-	33.4	32.4	31.7	31.5	-	54.1	53.3	52.6	52.3	-	35.6	34.4	33.6	33.2
Ours	**0.014 h**	-	**34.4**	**33.6**	**33.1**	**32.5**	-	**55.5**	**54.8**	**54.2**	**53.4**	-	**36.7**	**35.4**	**35.1**	**34.3**
Diff.	↑ **1349**×	-	**+0.8**	**+1.5**	**+2.0**	**+1.7**	-	**+1.0**	**+2.0**	**+3.1**	**+2.4**	-	**+1.1**	**+1.3**	**+1.9**	**+1.6**

(a) The mask AP (%) results compare different dataset pruning baselines on COCO.

		mAP^{bb}					${AP_{50}}^{bb}$					${AP_{75}}^{bb}$				
p	Time	0%	20%	30%	40%	50%	0%	20%	30%	40%	50%	0%	20%	30%	40%	50%
Random	-	37.7	37.0	35.3	34.0	33.8	58.3	57.6	56.1	53.8	54.3	41.1	40.1	38.0	37.3	36.3
Forgetting	20.29 h	-	36.8	35.6	34.5	34.0	-	57.7	56.2	55.2	54.4	-	40.4	38.7	37.5	36.8
Entropy	21.16 h	-	36.7	35.8	34.7	34.3	-	57.6	56.7	55.8	55.2	-	40.0	39.2	37.8	37.4
EL2N	12.37 h	-	36.9	35.7	34.7	34.0	-	57.7	56.4	55.1	54.5	-	40.1	38.9	37.6	36.5
AUM	20.29 h	-	37.0	35.8	34.8	34.3	-	57.9	56.6	55.6	55.1	-	40.6	39.0	38.1	37.1
CCS	20.29 h	-	36.8	35.7	35.2	34.7	-	57.6	56.5	56.1	55.7	-	40.3	39.1	38.2	37.5
Ours	**0.014 h**	-	**37.8**	**37.2**	**36.7**	**35.9**	-	**58.8**	**58.1**	**57.6**	**56.9**	-	**41.1**	**40.2**	**39.9**	**38.8**
Diff.	↑ **1349**×	-	**+0.8**	**+1.9**	**+2.7**	**+2.1**	-	**+1.2**	**+2.0**	**+3.8**	**+2.6**	-	**+1.0**	**+2.2**	**+2.6**	**+2.5**

(b) The bounding-box (bb) AP (%) results compare different dataset pruning baselines on COCO.

Generalization Experiments. (SOLO-v2 and QueryInst)

Model	**SOLO-v2**		**QueryInst**	
p	40%	50%	40%	50%
Random	51.4	51.1	53.3	52.8
Entropy	52.8	51.6	55.6	53.9
EL2N	52.1	50.3	55.0	52.7
AUM	52.5	51.0	55.6	54.0
CCS	53.0	52.1	55.0	53.5
Ours	**53.1**	**52.3**	**55.9**	**55.0**
Diff.	**+1.7**	**+1.2**	**+2.6**	**+2.2**

Table 3: The AP_{50} (%) results in the generalization ability to different architectures on COCO dataset.

Scalability Experiments. (Mask R-CNN)

		mAP				AP_{50}				AP_{50}			
Validation Network		20%	30%	40%	50%	20%	30%	40%	50%	20%	30%	40%	50%
ResNet-101	Random	35.5	34.8	34.0	33.1	56.6	55.7	54.7	53.8	37.9	36.9	36.3	35.0
	Ours	**35.8**	**35.3**	**34.9**	**34.3**	**57.1**	**56.6**	**56.4**	**55.6**	**38.2**	**37.6**	**37.3**	**36.6**
	Diff.	**+0.3**	**+0.5**	**+0.9**	**+1.2**	**+0.5**	**+0.9**	**+1.7**	**+1.8**	**+0.3**	**+0.7**	**+1.0**	**+1.6**
ResNeXt-101	Random	36.7	36.2	35.4	34.3	58.4	58.0	56.8	55.4	39.2	38.4	37.8	36.3
	Ours	**37.2**	**36.6**	**36.1**	**35.5**	**59.2**	**58.8**	**58.2**	**57.6**	**39.9**	**39.0**	**38.4**	**37.9**
	Diff.	**+0.5**	**+0.4**	**+0.7**	**+1.2**	**+0.8**	**+0.8**	**+1.4**	**+2.2**	**+0.7**	**+0.6**	**+0.6**	**+1.6**

(a) The mask AP (%) results for different IoU thresholds (0.5 to 0.95, 50, 75) of different backbones on COCO.

		mAP				AP_{50}^{bb}				AP_{50}^{bb}			
Validation Network		20%	30%	40%	50%	20%	30%	40%	50%	20%	30%	40%	50%
ResNet-101	Random	39.2	38.4	37.6	36.4	59.7	58.8	57.9	56.8	42.8	41.9	40.8	39.6
	Ours	**39.8**	**39.4**	**38.9**	**38.1**	**60.4**	**60.2**	**59.7**	**58.9**	**43.5**	**42.6**	**42.4**	**41.2**
	Diff.	**+0.6**	**+1.0**	**+1.3**	**+1.7**	**+0.7**	**+1.4**	**+1.8**	**+2.1**	**+0.7**	**+0.7**	**+1.6**	**+1.6**
ResNeXt-101	Random	40.8	40.3	39.1	38.0	61.6	61.4	59.9	58.7	44.7	44.2	42.8	41.5
	Ours	**41.4**	**41.0**	**40.3**	**39.6**	**62.6**	**62.1**	**61.4**	**60.9**	**45.4**	**45.0**	**44.0**	**43.4**
	Diff.	**+0.6**	**+0.7**	**+1.2**	**+1.6**	**+1.0**	**+0.7**	**+1.5**	**+2.2**	**+0.7**	**+0.8**	**+1.2**	**+1.9**

(b) The bbox AP (%) results for different IoU thresholds (0.5 to 0.95, 50, 75) of different backbones on COCO.

Table 11: The AP_{50} (%) results in the scalability ability to the different backbones of Mask R-CNN on the COCO dataset.

Results on **Cityscapes** and **VOC** datasets. (Mask R-CNN)

Dataset	VOC						Cityscapes					
p	Time	0%	20%	30%	40%	50%	Time	0%	20%	30%	40%	50%
Random	-	40.9	39.4	32.0	29.0	23.7	-	27.6	26.1	21.8	19.0	16.9
Forgetting	21.21 min	-	33.6	30.8	28.1	21.6	5.54 h	-	25.8	23.2	19.3	17.1
Entropy	21.75 min	-	38.4	34.2	31.7	29.3	5.61 h	-	26.4	22.2	20.1	17.2
El2N	12.70 min	-	39.1	35.3	32.1	29.8	3.01 h	-	26.2	22.6	20.3	17.4
AUM	21.21 min	-	35.2	31.0	26.3	19.2	5.54 h	-	25.3	24.5	21.2	18.4
CCS	21.21 min	-	38.8	35.4	34.3	30.8	5.54 h	-	25.4	24.1	19.9	17.0
Ours	**0.12 min**	-	**40.3**	**38.6**	**36.2**	**33.4**	**0.0051 h**	-	**27.5**	**25.4**	**23.4**	**19.4**
Diff.	**↑ 164×**	-	**+0.9**	**+6.6**	**+7.2**	**+9.7**	**↑ 100×**	-	**+1.4**	**+3.6**	**+4.4**	**+2.5**

Table 2: The mask AP (%) results compare different dataset pruning baselines on VOC and Cityscapes. The pruning rate p represents the percentage of data removed from the full training dataset during pruning. The performance on the full dataset is indicated by $p = 0\%$.

Results on **Cityscapes** datasets (Mask R-CNN **Pre-trained** on COCO).

	mAP					AP_{50}				
	0%	20%	30%	40%	50%	0%	20%	30%	40%	50%
Random	36.4	35.4	35.0	35.6	33.8	61.8	60.5	60.8	60.6	58.9
Entropy	-	34.7	35.6	34.5	34.3	-	61.3	62.6	61.4	60.3
EL2N	-	34.2	34.1	35.2	32.1	-	59.2	59.7	61.8	57.3
AUM	-	36.3	34.9	34.4	33.9	-	62.6	60.9	59.4	59.4
CCS	-	36.1	36.1	35.0	34.0	-	61.7	61.7	60.7	59.7
Ours	-	**36.9**	**36.6**	**36.6**	**36.6**	-	**62.8**	**63.9**	**63.4**	**62.8**
Diff.	-	**+1.5**	**+1.6**	**+1.0**	**+2.8**	-	**+2.3**	**+3.1**	**+2.8**	**+3.9**

Table 5: The mask AP (%) results on Cityscapes (**pre-trained on COCO**).

CREATING GROWTH, ENHANCING LIVES

Institute of High Performance Computing

IHPC

https://github.com/he-y/dataset-pruning-for-instance-segmentation

code

THANK YOU

www.a-star.edu.sg