Training-Free Dataset Pruning for Instance Segmentation

Anonymous authors
Paper under double-blind review

Abstract

Existing dataset pruning techniques primarily focus on classification tasks, limiting their applicability to more complex and practical tasks like instance segmentation. Instance segmentation presents three key challenges: pixel-level annotations, instance area variations, and class imbalances, which significantly complicate dataset pruning efforts. Directly adapting existing classification-based pruning methods proves ineffective due to their reliance on time-consuming model training process. To address this, we propose a novel **T**raining-**F**ree **D**ataset **P**runing (**TFDP**) method for instance segmentation. Specifically, we leverage shape and class information from image annotations to design a Shape Complexity Score (SCS), refining it into a Scale-Invariant (SI-SCS) and Class-Balanced (CB-SCS) versions to address instance area variations and class imbalances, all without requiring model training. We achieve state-of-the-art results on VOC 2012, Cityscapes, and MS COCO datasets, generalizing well across CNN and Transformer architectures. Remarkably, our approach accelerates the pruning process by an average of **1349**$\times$ on COCO compared to the adapted baselines.

1 Introduction

Current dataset pruning methods (Coleman et al., 2019; Toneva et al., 2019; Tan et al., 2023) focus on image classification tasks, while neglecting more complex and practical tasks such as instance segmentation. Classification is relatively simple, typically dealing with images containing one primary object. In contrast, instance segmentation faces greater challenges, handling real-world images with multiple objects of varying classes, areas, and positions within a single image.

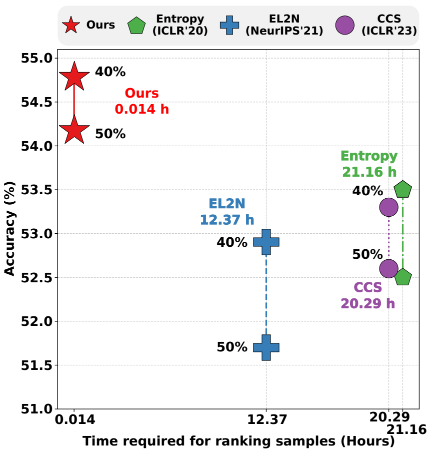


Figure 1: Comparison of AP$_{50}$ and runtime for sample ranking on COCO dataset: Our method vs. Entropy, EL2N, and CCS at 40% and 50% pruning rates. Our approach demonstrates superior efficiency and accuracy.

In this paper, we address dataset pruning for instance segmentation and identify three unique challenges. 1) ***Pixel-level annotations.*** Unlike classification tasks where each image has a single one-hot category label, instance segmentation tasks require labeling each pixel of an image, often with multiple different category labels present in a single image (Lin et al., 2014). 2) ***Variable instance areas.*** While images in classification tasks generally deal with images of consistent resolution, instance segmentation involves objects of varying areas within the same image. Fig. 2a and Appendix B demonstrates this diversity in multiple datasets, aligning with the observations in (Lin et al., 2014). 3) ***Class imbalance.*** In contrast to classification tasks where the image count of each category is typically uniform, instance segmentation (as shown in Fig. 2b) inherently contains imbalanced object counts across classes. This imbalance stems from the natural image collection process, which often captures multiple objects of varying frequencies in real-world scenes (Oksuz et al., 2020). Addressing these challenges is crucial for advancing instance segmentation dataset pruning.

Additionally, existing dataset pruning methods face a fundamental contradiction: they often require a time-consuming model training process to identify important samples despite aiming to reduce overall training time. This paradox stems from several fundamental issues inherent in existing methods: 1) These methods often necessitate training on the entire dataset to calculate relevance scores (Paul et al., 2021; Coleman et al., 2019; Pleiss et al., 2020; Toneva et al., 2019), negating the intended time-saving benefits. 2) Samples selected based on a single model's output often show limited generalization to models with different architectures (Yang et al., 2023), further compromising efficiency gains. 3) In real-world scenarios, model training may be infeasible due to insufficient resources (Khouas et al., 2024), rendering existing methods inapplicable to these scenarios.

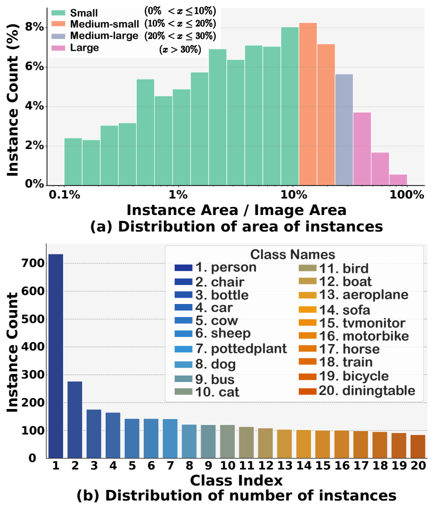


Figure 2: Visualization of VOC 2012 dataset to show variable instance area (a) and class imbalance (b).

To address the problems mentioned above, we propose a novel **T**raining-**F**ree **D**ataset **P**runing (**TFDP**) pipeline for instance segmentation that does not require training on any data in advance. Instance segmentation is sensitive to boundary regions (Tang et al., 2021; Zhang et al., 2021): early in training, predicted masks primarily cover the object's central area, while later stages refine the boundary pixels. Many studies have focused on making models pay more attention to boundaries to enhance final performance (Cheng et al., 2020; Wang et al., 2022a; Borse et al., 2021; Zhang et al., 2021). Therefore, leveraging the rich shape information provided by *pixel-level annotations* of masks, we designed the **Shape Complexity Score** (**SCS**), which characterizes the importance of an instance by calculating the complexity of each mask's boundary. Furthermore, to address the inherent *scale variability* arising from varying object sizes, we implemented **S**cale-**I**nvariant for ***SCS (SI-SCS)***, allowing this score to fairly represent the complexity of mask boundaries regardless of size. Additionally, to solve the problem of significant *class imbalances* in instance segmentation tasks (Oksuz et al., 2020), which can lead to noticeable class variability when simply summing instance-level scores to calculate image-level scores. We also design the **C**lass-**B**alanced for ***SCS (CB-SCS)***. Specifically, we normalize instance importance scores within each class across images, ensuring each class contributes equally, regardless of its instance count. This design allows us to obtain a unified metric that enables fair comparisons across images, regardless of the number of objects. These effective designs not only address the challenges faced in instance segmentation but also enable the superiority of our method in both time efficiency and performance (see Fig. 1).

As the first work on dataset pruning in instance segmentation and to fully validate the effectiveness of our method as illustrated in Fig. 3, we also adapt existing classification-oriented dataset pruning methods to this task, implementing some baselines with strong performance. Specifically, instead of calculating the importance score for each image in the classification task, we assign an importance score for each pixel based on existing criteria. We then aggregate the scores within each object and subsequently across objects in an image to derive an image-wise score. In our experiments, we conduct extensive comparisons between our implemented baselines and our model-independent TFDP, including performance comparisons, generalizability, and time consumption, to foster a foundational contribution to future work.

In summary, our contributions are as follows:

1) To the best of our knowledge, we are the **first** to introduce a training-free dataset pruning framework for instance segmentation.
2) We adapt existing classification-oriented methods to instance segmentation and establish strong baselines for comparison.
3) We propose a novel model-independent importance criterion ***SCS*** based on the shape information of masks to prune samples. Additionally, we implement Scale-Invariant and Class-Balanced versions to address the issues of scale variability and class imbalance.
4) Our method achieves the best results on mainstream instance segmentation datasets such as VOC 2012, Cityscapes, and MS COCO, without utilizing any model outputs. It also demonstrates

enhanced generalizability across various architectures (including CNN-based and Transformer-based networks) while offering a significantly faster and more practical pruning process.

2 Related Work

2.1 Dataset Compression

Dataset Pruning, or Coreset Selection, reduces dataset size by selecting key samples based on specific criteria. Herding (Welling, 2009) and Moderate (Xia et al., 2022) measure feature space distances, while Entropy (Coleman et al., 2019) and Cal (Margatina et al., 2021) focus on uncertainty. EL2N (Paul et al., 2021) uses gradient magnitudes to quantify importance. Some methods (Tang et al., 2023; Okanovic et al., 2023; Xu et al., 2023; Dolatabadi et al., 2022) improve training efficiency via online selection. GradMatch (Killamsetty et al., 2021) and Craig (Mirzasoleiman et al., 2020) minimize gradient differences between full and pruned datasets, while ACS (Huang et al., 2023) accelerates quantization-aware training by selecting high-gradient samples. However, these methods primarily focus on classification tasks and have not explored other more complex computer vision tasks further. Additionally, they rely on models to calculate importance scores, which is not only time-consuming but also results in limited generalizability.

Dataset Distillation is an another direction to compress datasets, aiming to learn a synthetic dataset that can recover the performance of the full dataset (Nguyen et al., 2021a;b; Zhou et al., 2022; Loo et al., 2022; Zhao et al., 2021; Jiang et al., 2022; Lee et al., 2022; Loo et al., 2023; Zhao & Bilen, 2023; Wang et al., 2022b; Zhao et al., 2023; Cazenavette et al., 2022; Du et al., 2023; Cui et al., 2023; Liu et al., 2023; Tukan et al., 2023; Shin et al., 2023; Yin et al., 2023; Yin & Shen, 2023; Shao et al., 2024; Zhou et al., 2024). However, since every pixel requires a gradient update during the optimization process, the training cost is significantly higher than dataset pruning.

2.2 Instance Segmentation

Instance segmentation is an important and challenging task in computer vision, as it requires instance-level and pixel-level predictions simultaneously. The existing methods can be roughly summarized into the following categories. 1) *Top-down methods* (Li et al., 2017; He et al., 2017; Liu et al., 2018b; Huang et al., 2019; Chen et al., 2019b; Bolya et al., 2019; Chen et al., 2020; Zhang et al., 2020) solve the problem from the perspective of object detection, *i.e.*, detecting first and then segmenting the object in the box. In particular, recent methods of (Chen et al., 2020; Zhang et al., 2020; Xie et al., 2020) build their methods on the anchor-free object detectors (Tian et al., 2019), showing promising performance. 2) *Bottom-up methods* (Newell et al., 2017; De Brabandere et al., 2017; Liu et al., 2017; Gao et al., 2019) view the task as a label-then-cluster problem, *e.g.*, learning the per-pixel embeddings and then clustering them into groups. 3) *Direct methods* (Wang et al., 2020a;b) perform instance segmentation directly without box detection or embedding learning. 4) *Transformer-based methods.* More recently, QueryInst (Fang et al., 2021) and Mask2Former (Cheng et al., 2022) proposed to decode random queries to objects for end-to-end instance segmentation frameworks by extending DETR (Carion et al., 2020). Instance segmentation typically requires large training datasets, but training efficiency in this domain remains understudied.

3 Method

3.1 Problem Definition

We first define the task of dataset pruning for instance segmentation. We have a complete training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{D}$, where D is the total number of images in the dataset. Here, $x_i \in \mathcal{X}$ is an input image, and $y_i \in \mathcal{Y}$ represents the ground truth for instance segmentation. Unlike classification tasks where the ground truth for an image corresponds to a single label (category), the ground truth for instance segmentation assigns a label (category) to each pixel of the image through masks. Specifically, each y_i contains a set of labeled instances:

$$\{(c_{i,j}, m_{i,j})\}_{j=1}^{G_i} \tag{1}$$

where G_i is the number of instances in image i, $c_{i,j}$ is the class label of the j-th instance, and $m_{i,j}$ is a binary mask that defines the spatial pixel boundaries of the instance.

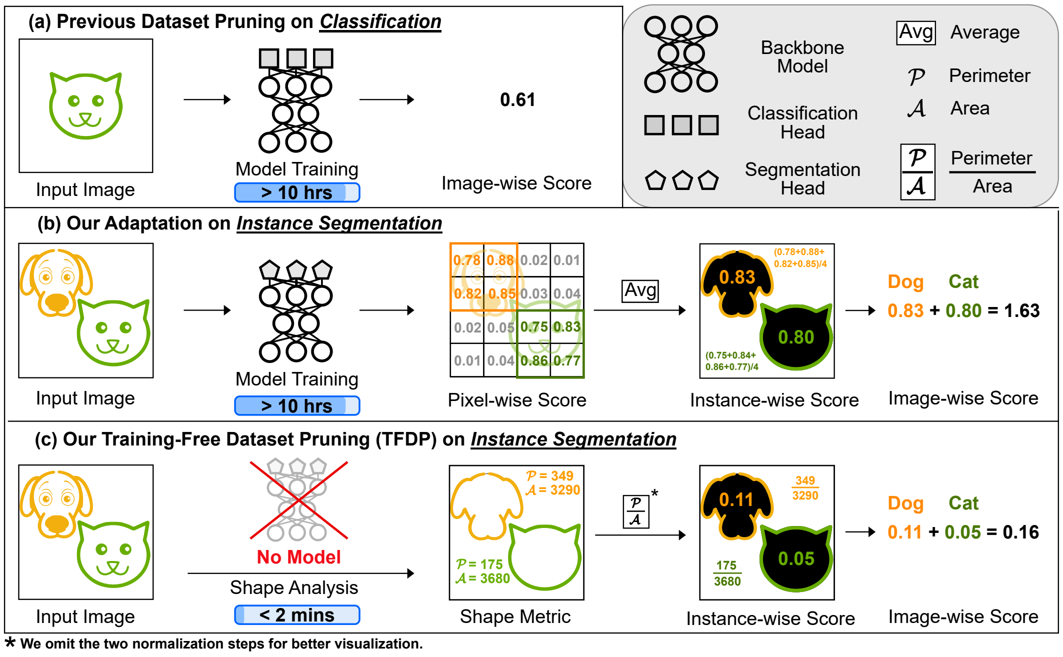


Figure 3: Comparison of different dataset pruning pipelines. (a) Pruning classification dataset requires model training. (b) Our adaptation of previous methods on instance segmentation by training a segemntation head and computing the importance score for each pixel. (c) The proposed method that is training-free and model-independent.

The objective of dataset pruning is to reduce the size of $\mathcal{D}$ by selecting a subset $\mathbb{S} \subseteq \mathcal{D}$ that maintains or maximizes the performance of a model trained on this reduced set compared to training on the entire dataset $\mathcal{D}$, thereby improving training efficiency and reducing training time. In our approach, we reduce the number of images (image-level) rather than instance annotations (instances-level) since the large volume of image data primarily impacts training time and storage requirements. While annotations for instance segmentation tasks include category labels, bounding boxes, and segmentation masks, their storage footprint remains significantly smaller compared to the images themselves. For instance, in the COCO dataset, images consume about 17 GB, whereas annotations occupy merely 0.7 GB.

3.2 Adapted Baselines for Instance Segmentation

Existing mainstream dataset pruning methods for classification primarily rely on model-derived logits for each image to calculate scores. However, for instance segmentation, each image requires the model to compute a logit for every pixel rather than for an entire image, making these classification-oriented methods unsuitable without modifications. To ensure a fair comparison, as shown in Fig. 4, we implement some strong baselines: we adapt these above-mentioned methods to suit instance segmentation tasks without altering their criteria for sample selection.

In the context of instance segmentation, we employ the widely-used Mask R-CNN framework as an example, and it can be directly applied to other segmentation models in the same manner. The mask loss in Mask R-CNN is computed using a pixel-wise binary cross-entropy loss between the predicted masks and the ground truth masks for each class. The mask loss L_m for each instance is defined as:

$$L_m = -\frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \left[y_{a,b} \log(\hat{y}_{a,b}) + (1 - y_{a,b}) \log(1 - \hat{y}_{a,b}) \right], \tag{2}$$

where H and W are the height and width of the RoI mask respectively, $y_{a,b}$ is the ground truth label at pixel (a, b) (1 for object, 0 for background), and $\hat{y}_{a,b}$ is the predicted logit of the pixel belonging to

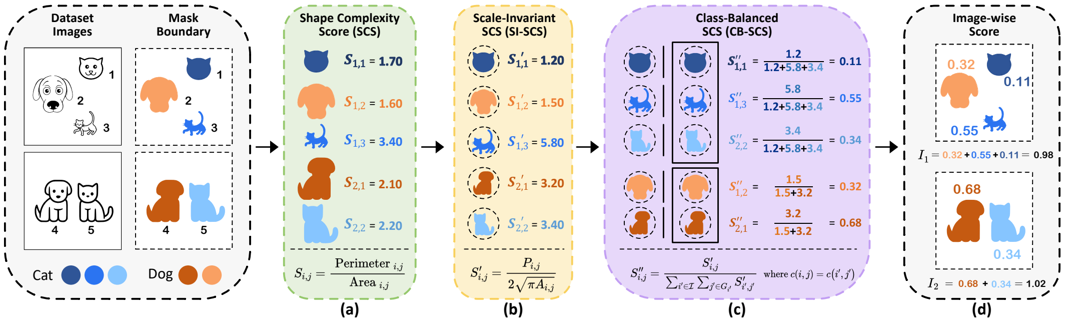

Dataset Images
Mask Boundary
1
2
3
4
5
Cat
Dog
Shape Complexity Score (SCS)
$S_{1,1}$ = 1.70
$S_{1,2}$ = 1.60
$S_{1,3}$ = 3.40
$S_{2,1}$ = 2.10
$S_{2,2}$ = 2.20
$S_{i,j} = \frac{\text{Perimeter}_{i,j}}{\text{Area}_{i,j}}$
(a)
Scale-Invariant SCS (SI-SCS)
$S'_{1,1}$ = 1.20
$S'_{1,2}$ = 1.50
$S'_{1,3}$ = 5.80
$S'_{2,1}$ = 3.20
$S'_{2,2}$ = 3.40
$S'_{i,j} = \frac{P_{i,j}}{2\sqrt{\pi A_{i,j}}}$
(b)
Class-Balanced SCS (CB-SCS)
$S''_{1,1} = \frac{1.2}{1.2+5.8+3.4} = 0.11$
$S''_{1,3} = \frac{5.8}{1.2+5.8+3.4} = 0.55$
$S''_{2,2} = \frac{3.4}{1.2+5.8+3.4} = 0.34$
$S''_{1,2} = \frac{1.5}{1.5+3.2} = 0.32$
$S''_{2,1} = \frac{3.2}{1.5+3.2} = 0.68$
$S''_{i,j} = \frac{S'_{i,j}}{\sum_{i' \in \mathcal{I}} \sum_{j' \in G_{i'}} S'_{i',j'}}$ where $c(i,j) = c(i',j')$
(c)
Image-wise Score
0.32
0.11
0.55
$I_1 = 0.32 + 0.55 + 0.11 = 0.98$
0.68
0.34
$I_2 = 0.68 + 0.34 = 1.02$
(d)

Figure 4: Overview of the proposed method. We introduce the Shape Complexity Score (***SCS***), in which we leverage the Perimeter-to-Area ratio to represent the boundary complexity. Following this, we apply scale normalization and intra-class normalization to address the inherent scale variability and class imbalance in instance segmentation tasks.

the object. This formulation allows for pixel-level precision in predicting masks, ensuring detailed and accurate segmentation outputs.

To adapt previous dataset pruning methods to instance segmentation tasks, we extend pixel-level importance scoring to instance-level and image-level scoring. Our model computes per-pixel logits for each predicted instance. We then apply established classification importance metrics (e.g., EL2N) to these logits. The instance-level score is obtained by averaging the pixel scores within each instance. The image's importance score is derived from the sum of its instance scores. Formally, the importance score I for the image i is calculated as follows:

$$I_i = \sum_{j=1}^{G_i} \left(\frac{1}{H_{i,j}} \sum_{a,b} s_{a,b,j} \right), \tag{3}$$

where G_i is the number of instances in the image i, $H_{i,j}$ is the number of pixels in the j-th instance mask in image i, and $s_{a,b,j}$ is the importance score at pixel (a, b) in the j-th instance mask. For *pixel-to-instance* score aggregation, we choose *average* to avoid area bias introduced by sum, which favors larger instances with more pixels. For *instance-to-image* aggregation, we use *sum* since images with more instance masks contain more objects and thus more information. Related experiments can also be found in the Appendix D.1.

3.3 Training-Free Dataset Pruning for Instance Segmentation

While effective, our adapted baselines are time-consuming and lack cross-architecture generalizability. To overcome these issues, we propose a novel Training-Free Dataset Pruning (TFDP) method for instance segmentation, shown in Fig. 4.

3.3.1 Shape Complexity Score (SCS)

We propose the **S**hape **C**omplexity **S**core (***SCS***, **Fig. 4a**), a novel model-independent metric that leverages the shape information of instance masks to select challenging samples. For each image x_i in our dataset, we have a set of G_i instance masks $\{M_{i,1}, M_{i,2}, ..., M_{i,G_i}\}$. The ***SCS*** is calculated for each instance mask as follows:

1. Mask Preparation: For each $M_{i,j}$, we obtain a binary mask $B_{i,j}$:

$$B_{i,j}(a, b) = \begin{cases} 1 & \text{if } (a, b) \in M_{i,j} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

2. Contour Extraction: We extract the set of contours $\mathcal{T}_{i,j} = \{T_{i,j,1}, T_{i,j,2}, ..., T_{i,j,Z}\}$ from $B_{i,j}$ and select the primary contour:

$$T^*_{i,j} = \arg \max_{T \in \mathcal{T}_{i,j}} \text{Area}(T) \tag{5}$$

3. Perimeter Calculation: The perimeter $P_{i,j}$ is calculated as:

$$P_{i,j} = \text{Perimeter}(T^*_{i,j}) = \sum_{k=1}^{H_{i,j}} \sqrt{(a_k - a_{k+1})^2 + (b_k - b_{k+1})^2}, \tag{6}$$

where $(a_{H_{i,j}+1}, b_{H_{i,j}+1}) = (a_1, b_1)$ to close the contour.

4. Area Calculation: The area $A_{i,j}$ of the instance is computed as:

$$A_{i,j} = \text{Area}(T^*_{i,j}) = \frac{1}{2}\left|\sum_{k=1}^{H_{i,j}} (a_k b_{k+1} - a_{k+1} b_k)\right|, \tag{7}$$

where (a_k, b_k) are the coordinates of the k-th point in the contour $T^*_{i,j}$ with $H_{i,j}$ points.

5. Shape Complexity Score Computation: The ***SCS*** $S_{i,j}$ for the j-th instance in the i-th image is defined as:

$$S_{i,j} = \frac{P_{i,j}}{A_{i,j}}. \tag{8}$$

This perimeter-to-area ratio increases with the boundary intricacy of the instance's shape, providing a measure of its complexity. To the best of our knowledge, the ***SCS*** is the first model-independent metric in the dataset pruning area that leverages the shape information of instances' masks to select challenging samples. This approach effectively alleviates the issue of limited generalizability caused by biases in selection based on specific model predictions and significantly reduces computational overhead.

3.3.2 Scale-Invariant SCS (SI-SCS)

The **S**hape **C**omplexity **S**core (***SCS***) exhibits a clear bias towards scale, which substantially impacts instance segmentation tasks, as shown in Fig. 2a. Specifically, smaller-scale instance masks receive higher scores, even when their boundaries are the same. To address this, we propose the **S**cale-**I**nvariant ***SCS*** (***SI-SCS***, **Fig. 4b**) .

For a polygon with perimeter $P_{i,j}$ and area $A_{i,j}$, scaling by factor f results in:

$$\frac{P'_{i,j}}{A'_{i,j}} = \frac{f P_{i,j}}{f^2 A_{i,j}} = \frac{P_{i,j}}{f A_{i,j}} \tag{9}$$

This decreases with increasing scale, biasing towards smaller instances.

To solve this, we normalize ***SCS*** using a circle (the shape with the minimum perimeter for a given area) as a reference. Let $S^\circ_{i,j}$ denote the ***SCS*** of a circle:

$$S^\circ_{i,j} = \frac{P_{i,j}}{A_{i,j}} = \frac{2\pi r}{\pi r^2} = \frac{2}{r} = 2\sqrt{\frac{\pi}{A_{i,j}}} \tag{10}$$

The ***SI-SCS*** is defined as:

$$S'_{i,j} = \frac{S_{i,j}}{S^\circ_{i,j}} = \frac{P_{i,j}}{2\sqrt{\pi A_{i,j}}} \tag{11}$$

For a scaled polygon (factor f), we have:

$$S'_{i,j}(f) = \frac{P_{i,j} \times f}{2\sqrt{\pi \cdot (A_{i,j} \times f^2)}} = \frac{P_{i,j} \times f}{2\sqrt{\pi A_{i,j}} \times f} = \frac{P_{i,j}}{2\sqrt{\pi A_{i,j}}} = S'_{i,j} \tag{12}$$

This demonstrates that ***SI-SCS*** is scale-invariant, depending only on the boundary complexity and not the instance's scale.

3.3.3 Class-Balanced SCS (CB-SCS)

While Scale-Invariance addresses intra-instance shape variability, inter-instance imbalance due to class imbalance remains a significant challenge. Instance segmentation datasets frequently exhibit highly skewed distributions of class instances (Oksuz et al., 2020), as illustrated in Fig. 2a and Appendix B. Naive aggregation of instance-level importance scores within an image can result in rankings disproportionately influenced by high-frequency classes, thereby diminishing the contributions of less frequent classes.

To mitigate this class imbalance issue, we propose the **C**lass-**B**alanced ***SCS*** (***CB-SCS***, **Fig. 4c)** score to ensures fair representation of all classes, regardless of the number of instances in each class. For each class, we normalize individual instance scores by the total score of all instances in that class across different images. Our normalized scoring method balances class influence, preventing overrepresented classes from dominating while simultaneously preserving the impact of classes with limited examples. Formally, the score for the i-th image is,

$$S''_{i,j} = \frac{S'_{i,j}}{\sum_{i' \in \mathcal{I}} \sum_{j' \in G_{i'}} S'_{i',j'}} \quad \text{where } c(i,j) = c(i',j'), \tag{13}$$

where $S'_{i,j}$ is the *SI-SCS* of the j-th instance in image i, $c(i,j)$ denotes the class of the j-th object in image i.

Image-level Score. Consequently, we can directly sum these normalized scores across all instances in i-th image to obtain a balanced image-level score formally defined as I_i (**Fig. 4d**):

$$I_i = \sum_{j}^{G_i} S''_{i,j}, \tag{14}$$

where G_i denotes the number of instances in the image i.

4 Experiments

4.1 Experiment settings

Datasets. To evaluate the proposed method TFDP, we conduct instance segmentation experiments on three mainstream datasets VOC 2012 (Everingham et al., 2010), Cityscapes (Cordts et al., 2016), and MS COCO (Lin et al., 2014).

Evaluation Metrics. For all datasets, we evaluate instance segmentation results using the standard COCO protocol and report the average precision metrics for both mask and box AP (averaged over IoU thresholds): mAP, AP_{50}, AP_{75}, AP_{S}, AP_{M}, AP_{L}. Due to the page limits more details about datasets and evaluation metrics are provided in Appendix C.1 and C.2, respectively.

Baseline Comparisons. Six baselines are used for comparison: 1) **Random**; 2) **Entropy** (Coleman et al., 2019); 3) **Forgetting** (Toneva et al., 2019); 4) **EL2N** (Paul et al., 2021); 5) **AUM** (Pleiss et al., 2020); 6) **CCS** (Zheng et al., 2023). For all methods (except Random), we make specific adaptations for the instance segmentation task, as described in Sec. 3.2. As is common practice (He et al., 2017; Wang et al., 2020b), all network backbones are pre-trained on the ImageNet-1k classification set (Deng et al., 2009) and then fine-tuned on the instance segmentation dataset. For hyperparameters, we follow the settings described in the original paper with details provided in the Appendix C.3.

4.2 Primary Results

MS COCO (Tab. 1). Tab. 1a shows the results on the COCO dataset. Our TFDP method outperforms all baselines across all pruning rate settings. For example, when pruning 50% of samples, TFDP still achieves 53.4% AP_{50}, which is 1.7% and 2.2% higher than adapted Entropy and EL2N, respectively. Moreover, TFDP's performance with only 80% of the data already matches the performance with 100% of the data. Additionally, with just 50% of the data selected by our TFDP method, it consistently outperforms random pruning by 30%. Impressively, compared to other baselines that require model training, the selection time for the model-independent TFDP is almost negligible. For more results on the COCO dataset, please refer to the Appendix D.2. **Bounding-Box (bb) AP Results.** Following previous work (He et al., 2017), we also report the bounding-box object detection results on three datasets in Tab. 1b. Our method TFDP significantly outperforms random pruning

		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.	$\uparrow$ **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.	$\uparrow$ **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.

Table 1: Results on COCO with backbone network Mask R-CNN. 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 $p = 0\%$. `Time` indicates the time consumption for sample ranking, with details provided in Sec. 4.3. `Diff` for the improvement in time represents the average improvement, excluding Random since it does not consume time. `Diff` for the performance denotes the difference between our TFDP method and random pruning, as all baselines are also our adapted methods.

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.	$\uparrow$ **164×**	-	**+0.9**	**+6.6**	**+7.2**	**+9.7**	$\uparrow$ **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 $p = 0\%$.

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.

at all pruning rate settings. Notably, by selecting only 50% of the data, our TFDP consistently surpasses all baseline methods that select 70% of the data (30% pruning rate) in both mAP^{bb} and ${AP_{50}}^{bb}$ performance. Compared to the Mask AP results, the performance improvement in bounding-box AP is more pronounced.

VOC and Cityscapes (Tab. 2). Performance results on VOC and Cityscapes are reported in Tab. 2. Our TFDP method demonstrates superior performance on VOC. For instance, with the pruning rate of 40% and 50%, Mask R-CNN trained on the pruned VOC achieves mask AP_{50} accuracies of 36.2% and 33.4%, surpassing random pruning by 7.2% and 9.7%, respectively. On Cityscapes, TFDP also consistently shows improvements. For example, at a pruning rate of 40%, TFDP exceeds the performance of random pruning at 40% by 4.4%, and at 30% by 1.3%. Additionally, our TFDP method consistently outperforms the adapted SOTA methods (e.g., EL2N and CCS) on VOC and Cityscapes at all pruning rate settings. For more results on the VOC and Cityscapes dataset, please refer to the Appendix D.3.

SI	CB	VOC			Cityscapes			COCO		
p		30%	40%	50%	30%	40%	50%	30%	40%	50%
random		32.0	29.0	23.7	22.1	19.0	16.9	53.8	52.1	52.0
-	-	36.6	34.0	28.8	22.4	21.5	19.1	54.2	52.9	52.5
✓	-	37.8	35.4	32.4	22.2	22.8	17.9	54.2	53.3	52.1
-	✓	35.9	33.5	30.7	24.9	22.4	17.1	54.4	54.0	52.7
✓	✓	**38.6**	**36.2**	**33.4**	**25.4**	**23.4**	**19.4**	**54.8**	**54.2**	**53.4**

Table 4: Ablation study of two components ***SI-SCS*** (SI) and ***CB-SCS*** (CB). When neither normalization is used (both marked as '-'), it indicates that only ***SCS*** is applied.

	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**).

Generalization to Other CNN-based and Transformer-based Models (Tab. 3). In Tab. 3, we show an experiment assessing the effectiveness of images selected by Mask R-CNN when tested on instance segmentation networks with new architectures. We included other CNN-based network SOLO-v2 (Wang et al., 2020b), as well as Transformer-based networks QueryInst (Fang et al., 2021) to test for generalizability. Appendix D.4 shows more results of the generalization evaluation. Additionally, we conducted experiments on different backbones (e.g., ResNet-101 and ResNeXt-101) as shown in Appendix D.5, further validating the scalability of our method.

Ablation Study (Tab. 4). Tab. 4 demonstrates the effectiveness of our components on three datasets. We notice using them alone does not give satisfactory results, and applying both ***SI-SCS*** and ***CB-SCS*** delivers the best performance. Further analysis on the Scale-Invariance design is provided in Appendix D.6.

4.3 More Analysis

Robust Performance Across Training Paradigms (Tab. 5). Our TFDP is a method that prepares an pruned dataset independent of any specific training process. To comprehensively evaluate its effectiveness, we use TFDP-pruned datasets in two scenarios: 1) Models pre-trained on ImageNet-1k (Results shown in Tab. 1, Tab. 2 and Tab. 3), 2) Models pre-trained on COCO (He et al., 2017; Liu et al., 2018a) as shown in Tab. 5. In both cases, models are fine-tuned on either the full or TFDP-pruned dataset. Impressively, TFDP outperforms all baselines across different pruning rates in both cases. Even when pruning 50% of the data, it (AP_{50} 62.8%) surpasses the performance achieved with the full dataset (AP_{50} 61.8%). This demonstrates TFDP's effectiveness as a universal preprocessing step regardless of the subsequent training strategy. More experimental settings and results for training scenarios 2) are provided in Appendix D.7.

Performance Gain at High Pruning Rates (Fig. 5). To further validate our method, we also conduct experiments under high pruning rates, as shown in Fig. 5. Compared to random pruning, our method consistently improves performance across pruning rates from 60% to 90%.

Time Consumption Results (Tab. 6). We detail the time required to calculate importance scores for all samples in the dataset, including model training, score computation via model inference, and stratified selection. All time tests were conducted in the same environment, with details available in Appendix D.8, and the times reported are averages of three trials. As shown in Tab. 6, our method significantly reduces sample selection time, with greater time-saving effects on larger datasets, highlighting its effectiveness in the big data era.

Visualization (Fig. 6). Fig. 6 shows that our method effectively distinguishes "hard" (top-ranked) and "easy" images (least-ranked). Top-ranked images contain either many intricate-shaped objects (subfig-a, b, c) or instances of rare classes (subfig-d, e), while least-ranked images typically contain a single object with a simple contour (subfig-f, g, h, i, j, k). We use the top-1 image (subfig-d, e) from MS COCO to explain how scale variability and class imbalance issues are addressed. First, despite having various scales, ***SI-SCS*** assigns all toasters similar scores, illustrating the effectiveness of our **S**cale-**I**nvariant ***SCS (SI-SCS)***. Second, despite having relatively simple shapes, ***CB-SCS*** assigns a high score to this image due to the rarity of the class "toaster", showcasing our **C**lass-**B**alanced ***SCS (CB-SCS)***. The distribution Fig. 8 in the Appendix reveals that the number of toasters is the

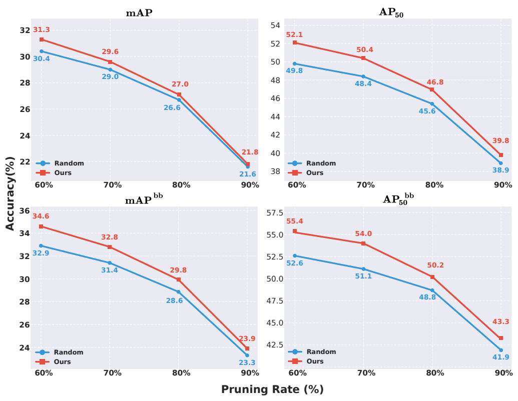


Figure 5: Experiments of high pruning rates (from 60% to 90%) on MS COCO dataset. Segmentation metrics include mAP, AP_{50} and the corresponding bounding-box version, mAP^{bb} and ${AP_{50}}^{bb}$.

Dataset		Ours	EL2N	Entropy	AUM/ Forgetting	CCS
VOC	$\mathcal{T}$	-	722.49 s	1265.40 s	1272.31 s	1272.31 s
	$\mathcal{I}$	7.19 s	39.39 s	39.71 s	-	-
	$\mathcal{S}$	-	-	-	-	0.009 s
	Total	**7.19 s**	761.88 s	1305.11 s	1272.31 s	1272.32 s
Cityscapes	$\mathcal{T}$	-	2.89 h	5.49 h	5.54 h	5.54 h
	$\mathcal{I}$	182.4 s	429.29 s	432.33 s	-	-
	$\mathcal{S}$	-	-	-	-	0.029 s
	Total	**182.4 s**	3.01 h	5.61 h	5.54 h	5.54 h
COCO	$\mathcal{T}$	-	11.35 h	20.13 h	20.29 h	20.29 h
	$\mathcal{I}$	50.4 s	3676.48 s	3696.69 s	-	-
	$\mathcal{S}$	-	-	-	-	0.99 s
	Total	**50.4 s**	12.37 h	21.16 h	20.29 h	20.29 h

Table 6: Detailed time calculation. $\mathcal{T}$ denotes the model training time. $\mathcal{I}$ is the score computation via model inference. $\mathcal{S}$ is the stratified selection time, for CCS (Zheng et al., 2023).

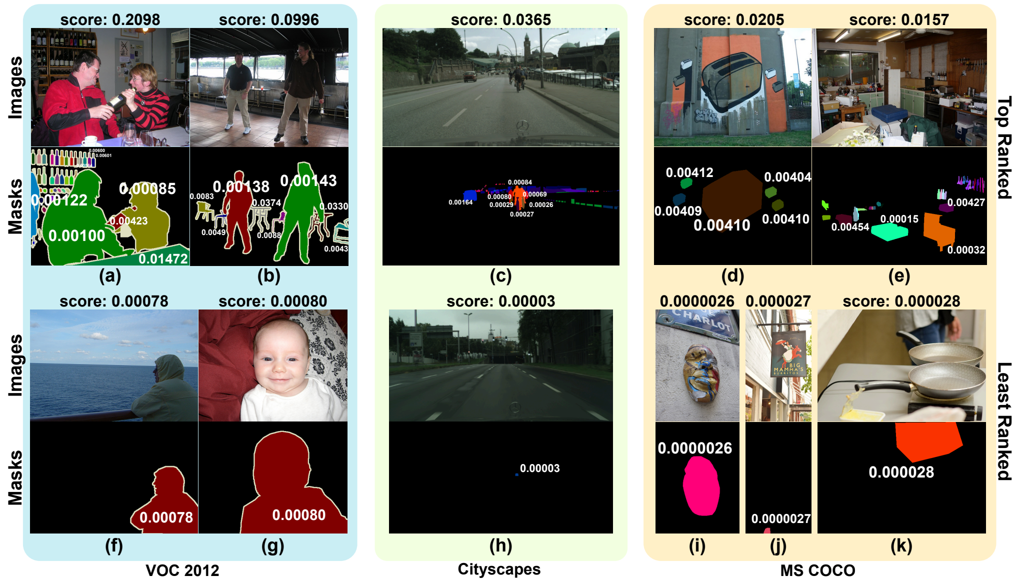


Figure 6: Visualization of TFDP-selected images on different datasets. The original aspect ratios of the images are preserved. Scores of too small objects are omitted for better visualization.

second-to-last least counted object. More visualizations, examples and analysis are provided in Appendix D.9.

5 Conclusion and Future Work

We introduce Training-Free Dataset Pruning (TFDP) for instance segmentation, leveraging mask annotations and novel scoring methods to address scale variability and class imbalance. Our approach significantly reduces pruning time while improving performance, outperforming adapted state-of-the-art baselines and demonstrating strong cross-architecture generalization. In future work, we plan to extend the training-free concept to diverse datasets. This expansion will include video dataset pruning by incorporating temporal information and motion patterns, language dataset pruning through analysis of text structure, syntax, and semantic richness. Exploring the theoretical foundations of training-free pruning methods is another interesting direction.

References

Y Ayad. Assessment of landscape ecological metrics: Shape complexity and fragmentation of the abandoned strip mine patches in toby creek watershed. *Clarion University of Pennsylvania*, 2005.

Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *ICCV*, 2019.

Shubhankar Borse, Ying Wang, Yizhe Zhang, and Fatih Porikli. Inverseform: A loss function for structured boundary-aware segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5901–5911, 2021.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan. Blend-Mask: Top-down meets bottom-up for instance segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019a.

Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollar. TensorMask: A foundation for dense object segmentation. In *Proc. Int. Conf. Comput. Vis.*, 2019b.

Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022.

Tianheng Cheng, Xinggang Wang, Lichao Huang, and Wenyu Liu. Boundary-preserving mask r-cnn. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pp. 660–676. Springer, 2020.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *Int. Conf. Learn. Represent.*, 2019.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Scaling up dataset distillation to imagenet-1k with constant memory. In *Proc. Int. Conf. Mach. Learn.*, pp. 6565–6590. PMLR, 2023.

Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv:1708.02551*, 2017.

Michael John De Smith, Michael F Goodchild, and Paul Longley. *Geospatial analysis: a comprehensive guide to principles, techniques and software tools*. Troubador publishing ltd, 2007.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 248–255, 2009.

Hadi M. Dolatabadi, Sarah M. Erfani, and Christopher Leckie. L-robustness and beyond: Unleashing efficient adversarial training. In *Eur. Conf. Comput. Vis.*, 2022.

Jiawei Du, Yidi Jiang, Vincent TF Tan, Joey Tianyi Zhou, and Haizhou Li. Minimizing the accumulated trajectory error to improve dataset distillation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88: 303–338, 2010.

Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Instances as queries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6910–6919, October 2021.

Naiyu Gao, Yanhu Shan, Yupei Wang, Xin Zhao, Yinan Yu, Ming Yang, and Kaiqi Huang. SSAP: Single-shot instance segmentation with affinity pyramid. In *Proc. Int. Conf. Comput. Vis.*, 2019.

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

Xijie Huang, Zechun Liu, Shih-Yang Liu, and Kwang-Ting Cheng. Efficient quantization-aware training with adaptive coreset selection. *arXiv preprint arXiv:2306.07215*, 2023.

Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring R-CNN. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

Zixuan Jiang, Jiaqi Gu, Mingjie Liu, and David Z Pan. Delving into effective gradient matching for dataset condensation. *arXiv preprint arXiv:2208.00311*, 2022.

Aymen Rayane Khouas, Mohamed Reda Bouadjenek, Hakim Hacid, and Sunil Aryal. Training machine learning models at the edge: A survey. *arXiv preprint arXiv:2403.02619*, 2024.

Krishnateja Killamsetty, Sivasubramanian Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *Proc. Int. Conf. Mach. Learn.*, pp. 5464–5474, 2021.

Saehyung Lee, Sanghyuk Chun, Sangwon Jung, Sangdoo Yun, and Sungroh Yoon. Dataset condensation with contrastive signals. In *Proc. Int. Conf. Mach. Learn.*, pp. 12352–12364, 2022.

Wenwen Li, Michael F Goodchild, and Richard Church. An efficient measure of compactness for two-dimensional shapes and its application in regionalization problems. *International Journal of Geographical Information Science*, 27(6):1227–1250, 2013.

Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Shu Liu, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. Sequential grouping networks for instance segmentation. In *Proc. Int. Conf. Comput. Vis.*, 2017.

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*, pp. 8759–8768, 2018a.

Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018b.

Yanqing Liu, Jianyang Gu, Kai Wang, Zheng Zhu, Wei Jiang, and Yang You. DREAM: Efficient dataset distillation by representative matching. *arXiv preprint arXiv:2302.14416*, 2023.

Noel Loo, Ramin Hasani, Alexander Amini, and Daniela Rus. Efficient dataset distillation using random feature approximation. In *Proc. Adv. Neural Inform. Process. Syst.*, 2022.

Noel Loo, Ramin Hasani, Mathias Lechner, and Daniela Rus. Dataset distillation with convexified implicit gradients. In *Proc. Int. Conf. Mach. Learn.*, 2023.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. *arXiv preprint arXiv:2109.03764*, 2021.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *Proc. Int. Conf. Mach. Learn.*, pp. 6950–6960, 2020.

Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Proc. Adv. Neural Inform. Process. Syst.*, 2017.

Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. In *Proc. Int. Conf. Learn. Represent.*, 2021a.

Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset distillation with infinitely wide convolutional networks. In *Proc. Adv. Neural Inform. Process. Syst.*, pp. 5186–5198, 2021b.

Patrik Okanovic, Roger Waleffe, Vasilis Mageirakos, Konstantinos E Nikolakakis, Amin Karbasi, Dionysis Kalogerias, Nezihe Merve Gürel, and Theodoros Rekatsinas. Repeated random sampling for minimizing the time-to-accuracy of learning. *arXiv preprint arXiv:2305.18424*, 2023.

Kemal Oksuz, Baris Can Cam, Sinan Kalkan, and Emre Akbas. Imbalance problems in object detection: A review. *IEEE transactions on pattern analysis and machine intelligence*, 43(10): 3388–3415, 2020.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *Adv. Neural Inform. Process. Syst.*, volume 34, pp. 20596–20607, 2021.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. In *Adv. Neural Inform. Process. Syst.*, volume 33, pp. 17044–17056, 2020.

Shitong Shao, Zeyuan Yin, Muxin Zhou, Xindong Zhang, and Zhiqiang Shen. Generalized large-scale data condensation via various backbone and statistical matching. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

Seungjae Shin, Heesun Bae, Donghyeok Shin, Weonyoung Joo, and Il-Chul Moon. Loss-curvature matching for dataset selection and condensation. In *International Conference on Artificial Intelligence and Statistics*, pp. 8606–8628, 2023.

Haoru Tan, Sitong Wu, Fei Du, Yukang Chen, Zhibin Wang, Fan Wang, and Xiaojuan Qi. Data pruning via moving-one-sample-out. In *Adv. Neural Inform. Process. Syst.*, 2023.

Chufeng Tang, Hang Chen, Xiao Li, Jianmin Li, Zhaoxiang Zhang, and Xiaolin Hu. Look closer to segment better: Boundary patch refinement for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13926–13935, 2021.

Zhenyu Tang, Shaoting Zhang, and Xiaosong Wang. Exploring data redundancy in real-world image classification through data selection. *arXiv preprint arXiv:2306.14113*, 2023.

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *Proc. Int. Conf. Comput. Vis.*, 2019.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *Int. Conf. Learn. Represent.*, 2019.

Murad Tukan, Alaa Maalouf, and Margarita Osadchy. Dataset distillation meets provable subset selection. *arXiv preprint arXiv:2307.08086*, 2023.

Chi Wang, Yunke Zhang, Miaomiao Cui, Peiran Ren, Yin Yang, Xuansong Xie, Xian-Sheng Hua, Hujun Bao, and Weiwei Xu. Active boundary loss for semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 2397–2405, 2022a.

Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. Cafe: Learning to condense dataset by aligning features. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 12196–12205, 2022b.

Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. SOLO: Segmenting objects by locations. In *Proc. Eur. Conf. Comput. Vis.*, 2020a.

Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2020b.

Max Welling. Herding dynamical weights to learn. In *Proc. Int. Conf. Mach. Learn.*, pp. 1121–1128, 2009.

Haoyu Wu, Clare Flynn, Carole Hall, Christian Che-Castaldo, Dimitris Samaras, Mathew Schwaller, and Heather J Lynch. Penguin colony georegistration using camera pose estimation and phototourism. *Plos one*, 19(10):e0311038, 2024.

Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *Int. Conf. Learn. Represent.*, 2022.

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. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

Xilie Xu, Jingfeng Zhang, Feng Liu, Masashi Sugiyama, and Mohan Kankanhalli. Efficient adversarial contrastive learning via robustness-aware coreset selection. In *Adv. Neural Inform. Process. Syst.*, 2023.

Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. Dataset pruning: Reducing training data by examining generalization influence. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=4wZiAXD29TQ`.

Zeyuan Yin and Zhiqiang Shen. Dataset distillation in large data era. *arXiv preprint arXiv:2311.18838*, 2023.

Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. In *Proc. Adv. Neural Inform. Process. Syst.*, 2023.

Gang Zhang, Xin Lu, Jingru Tan, Jianmin Li, Zhaoxiang Zhang, Quanquan Li, and Xiaolin Hu. Refinemask: Towards high-quality instance segmentation with fine-grained features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6861–6869, 2021.

Rufeng Zhang, Zhi Tian, Chunhua Shen, Mingyu You, and Youliang Yan. Mask encoding for single shot instance segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, pp. 6514–6523, 2023.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *Proc. Int. Conf. Learn. Represent.*, 2021.

Ganlong Zhao, Guanbin Li, Yipeng Qin, and Yizhou Yu. Improved distribution matching for dataset condensation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 7856–7865, 2023.

Haizhong Zheng, Rui Liu, Fan Lai, and Atul Prakash. Coverage-centric coreset selection for high pruning rates. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=QwKvL6wC8Yi`.

Muxin Zhou, Zeyuan Yin, Shitong Shao, and Zhiqiang Shen. Self-supervised dataset distillation: A good compression is all you need. *arXiv preprint arXiv:2404.07976*, 2024.

Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. In *Proc. Adv. Neural Inform. Process. Syst.*, 2022.

A TRAINING-FREE DATASET PRUNING (TFDP) ALGORITHM

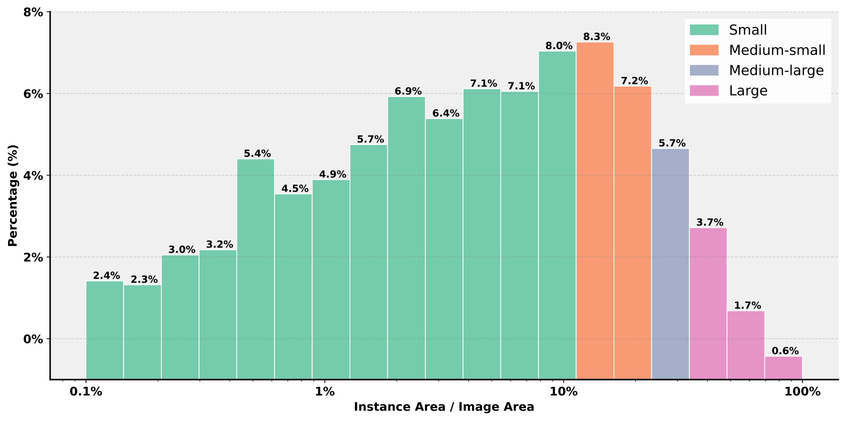
Algorithm 1 Training-Free Dataset Pruning (TFDP)

```
Input: Complete training dataset $\mathcal{D}$. Number of images to select $K$.
 1: Initialize: $\mathbb{S} = \emptyset$, set of selected samples
 2: for $i = 1$ to $D$ do
 3:     for $j = 1$ to $G_i$ do
 4:         Calculate $P_{i,j}$ and $A_{i,j}$ for each mask $M_{i,k}$ in image $i$
 5:         Compute SCS $S_{i,j}$                                   ▷ Defined in Equation 8
 6:         Scale-Invariant SCS by circle ratio $S'_{i,j}$          ▷ Defined in Equation 12
 7:     end for
 8: end for
 9: for $i = 1$ to $D$ do
10:     for $k = 1$ to $G_i$ do
11:         Compute Class-Balanced SCS $S''_{i,j}$ for each instance mask $m_{i,j}$ in image $i$
12:     end for
13:     Calculate image score $I_i$                                 ▷ Defined in Equation 14
14: end for
15: Sort all images in $\mathcal{D}$ based on $I_i$ in descending order
16: Select top $K$ images based on sorted scores to form subset $S$
Output: Selected subset $\mathbb{S} = \{(x_i, y_i)\}_{i=1}^{K}$; Importance scores based on $I_i$
```

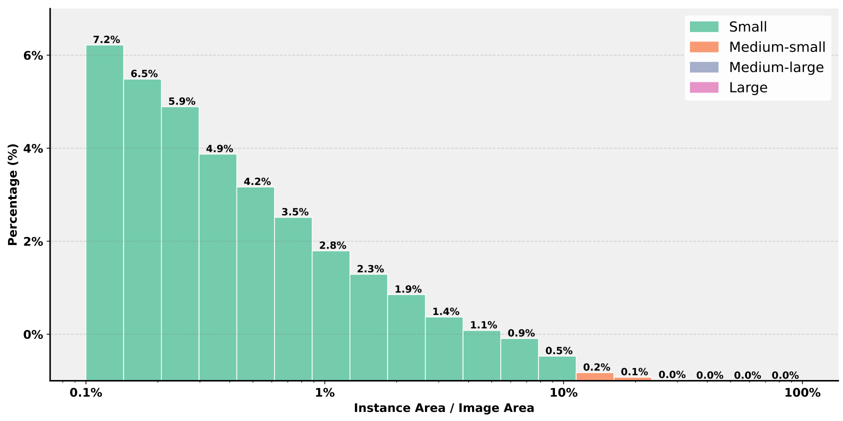
Algorithm 1 illustrates the code implementation process of our TFDP.

B Datasets Analysis

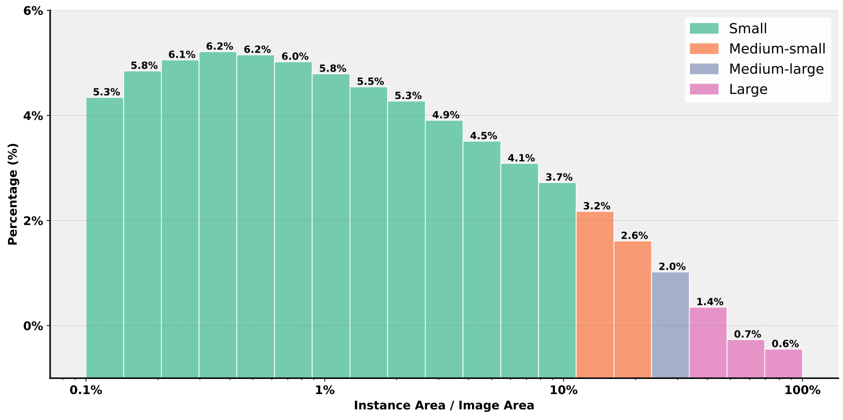
We show statistical details of more datasets, including VOC, Cityscapes and COCO. As shown in Fig. 8, there are significant differences in object areas across the datasets, along with a clear class imbalance.



(a) Distribution of VOC 2012 dataset.

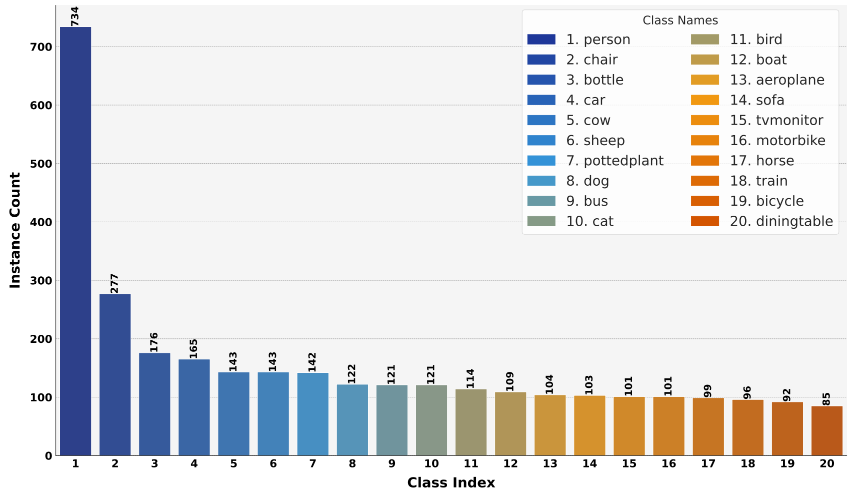


(b) Distribution of Cityscapes dataset.

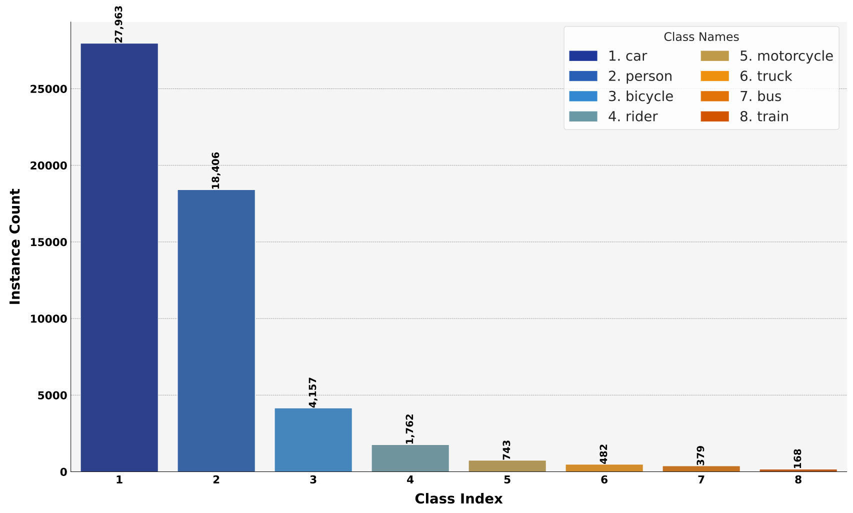


(c) Distribution of MS COCO dataset.

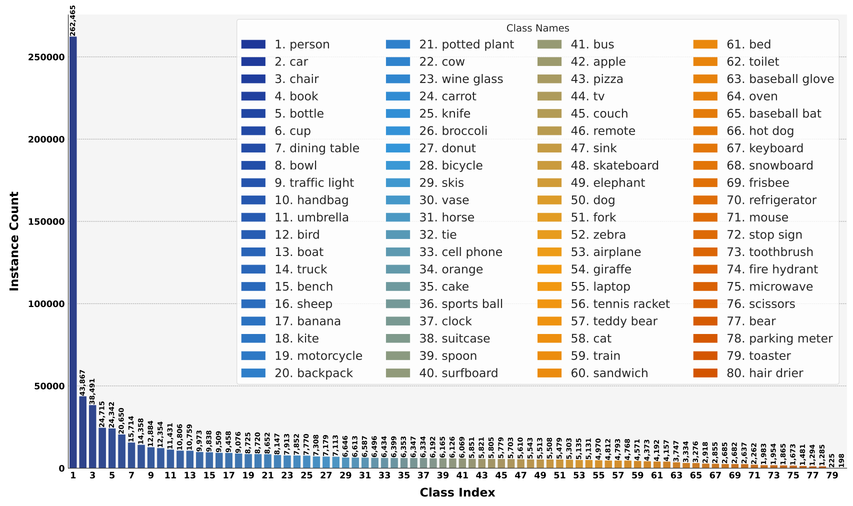
Figure 7: Distribution of area of instances.



(a) Distribution of VOC 2012 dataset.



(b) Distribution of Cityscapes dataset.



(c) Distribution of MS COCO dataset.

Figure 8: Distribution of number of instances.

C Experiment Details

C.1 Dataset Details

Our experiments are conducted on three different datasets:

- **Pascal VOC 2012** features 2,913 images, split into 1,464 for training and 1,449 for validation, with 6,929 segmentations available for instance segmentation task.
- **Cityscapes** consists of 9 object categories for instance-level semantic labeling. This dataset is more challenging since each image can contain a much larger number of instances of each class than in VOC, most of which are very small. It comprises 2975 training images from 18 cities, 500 validation images from 3 cities.
- **MS COCO** is a popular instance segmentation dataset, which contains an 80-category label set with instance-level annotations. Following previous works (He et al., 2017; Wang et al., 2020b), we use the COCO `train 2017` (118K training images) for training, and the ablation study is carried out on the `val 2017` (5K validation images).

C.2 Evaluation Metrics

For all datasets, we evaluate instance segmentation results using the standard COCO protocol (Lin et al., 2014). This protocol provides a comprehensive set of metrics to assess the performance of object detection and instance segmentation models. We report the average precision (AP) metrics for both mask and bounding box predictions, which are averaged over multiple Intersection over Union (IoU) thresholds. The key metrics we report are:

- **mAP (mean Average Precision):** This is the primary metric, calculated by averaging AP across all object categories and IoU thresholds (typically from 0.5 to 0.95 in steps of 0.05).
- **AP_{50}:** Average Precision at IoU threshold of 0.5. This metric is more lenient, considering detections as correct if they have at least 50% overlap with the ground truth.
- **AP_{75}:** Average Precision at IoU threshold of 0.75. This is a stricter metric, requiring more accurate localization of objects.
- **AP_S, AP_M, AP_L:** These metrics evaluate performance on small, medium, and large objects respectively.
  - AP_S: AP for small objects (area $< 32^2$ pixels)
  - AP_M: AP for medium objects ($32^2 <$ area $< 96^2$ pixels)
  - AP_L: AP for large objects (area $> 96^2$ pixels)

These size-specific metrics help assess the model's performance across different scales of objects, which is crucial for understanding its effectiveness in various real-world scenarios.

C.3 Experiments Settings

Except for the generalization and scalability experiments, we use ResNet-50 as the backbone for Mask R-CNN and employ FPN to extract multi-scale features. All models and training hyperparameters were trained using the default hyperparameters as specified in MMDetection (Chen et al., 2019a).

Adapted Baseline Settings. For EL2N, in alignment with the description of the model in the early training stage from the original paper, we use the model at halfway through the total training duration to compute EL2N scores. For CCS, we strictly follow the hyperparameters in the original paper, including the hard pruning rate β and the number of strata k. It is worth noting that our TFDP does not contain any hyperparameters, further demonstrating the practicality of our TFDP approach.

Primary Experiment Settings. For the VOC experiment, we used a single NVIDIA 3090 GPU. For the Cityscapes experiment, we used two NVIDIA 3090 GPUs. For the COCO experiment, we used two NVIDIA A100 80G GPUs. **Time Consumption Experiment Settings.** To ensure a fair comparison, all time consumption experiments were conducted on the same machine: PyTorch on

Ubuntu 20.04, with NVIDIA RTX 3090 GPUs and CUDA 11.3. We used two NVIDIA 3090 GPUs for training and one NVIDIA 3090 GPU for inference.

D More Experiments

D.1 Comparison between summation and average

We compared two methods for aggregating object scores through pixels: summation and average. As shown in Tab. 7, the average-based method performs significantly better than the summation-based method. This is because the summation-based method tends to favor larger masks, i.e., instances with more pixels, causing smaller instances to be overlooked.

	mAP			AP_{50}			AP_{75}			AP_S			AP_M			AP_L		
	Avg.	Sum	Diff.	Avg.	Sum	Diff.	Avg.	Sum	Diff.	Avg.	Sum	Diff.	Avg.	Sum	Diff.	Avg.	Sum	Diff.
Forgetting	28.7	31.0	**+2.3**	47.7	52.1	**+4.4**	31.1	35.1	**+4.0**	12.3	16.7	**+4.4**	31.6	33.4	**+1.8**	40.1	42.8	**+2.7**
Entropy	28.1	31.1	**+3.0**	47.0	51.7	**+4.7**	30.6	34.7	**+4.1**	12.7	16.2	**+3.5**	31.1	33.2	**+2.1**	39.9	42.7	**+2.8**
EL2N	25.4	30.3	**+4.9**	45.3	50.9	**+5.6**	28.9	33.2	**+4.3**	11.0	14.3	**+3.3**	30.1	32.1	**+2.0**	38.7	41.3	**+2.6**
AUM	26.7	29.9	**+3.2**	46.6	51.2	**+4.6**	29.7	33.9	**+4.2**	11.4	15.6	**+4.2**	30.3	32.9	**+2.6**	39.0	41.2	**+2.2**
CCS	28.4	30.9	**+2.5**	47.2	51.8	**+4.6**	31.0	34.8	**+3.8**	12.1	16.7	**+4.6**	31.1	33.2	**+2.1**	40.3	42.9	**+2.6**

Table 7: Comparison between averaging (`Avg.`) and summation (`Sum`).

D.2 MS COCO result on AP_S, AP_M, and AP_L

Tab. 8 shows more detailed AP results on the COCO dataset for different object areas (small, medium, large). Our TFDP consistently achieves stable improvements.

	AP_S					AP_M					AP_L				
	0%	20%	30%	40%	50%	0%	20%	30%	40%	50%	0%	20%	30%	40%	50%
Random	16.1	15.9	14.1	13.1	13.2	36.7	36.0	34.6	33.2	33.2	49.9	48.8	48.1	46.2	45.6
Forgetting	-	15.6	15.0	14.6	14.5	-	35.7	35.1	34.2	33.7	-	49.1	46.5	44.9	43.6
Entropy	-	15.5	14.9	14.3	14.4	-	35.9	35.0	34.1	33.9	-	48.9	46.7	45.1	43.8
EL2N	-	15.6	14.5	14.4	14.4	-	36.3	35.2	34.1	33.6	-	47.9	46.3	44.6	43.2
AUM	-	15.8	14.7	14.6	14.4	-	36.3	35.4	34.3	34.1	-	48.5	46.6	45.3	44.0
CCS	-	15.8	15.2	14.8	14.7	-	36.0	34.9	34.6	34.1	-	48.4	46.7	45.0	45.2
Ours	-	**16.2**	**15.5**	**15.5**	**15.1**	-	**37.1**	**36.3**	**36.0**	**35.3**	-	**49.3**	**48.5**	**46.9**	**46.2**
Diff.	-	**+0.3**	**+1.4**	**+2.4**	**+1.9**	-	**+1.1**	**+1.7**	**+2.8**	**+2.1**	-	**+0.5**	**+0.4**	**+0.7**	**+0.6**

(a) The mask AP (%) results for different object areas (small, medium, large) compare different dataset pruning baselines on COCO.

	${AP_S}^{bb}$					${AP_M}^{bb}$					${AP_L}^{bb}$				
	0%	20%	30%	40%	50%	0%	20%	30%	40%	50%	0%	20%	30%	40%	50%
Random	21.9	21.8	19.6	18.4	18.4	40.9	40.2	38.4	37.1	36.8	48.9	47.8	46.4	44.1	44.3
Forgetting	-	20.8	20.7	20.1	20.1	-	40.1	39.9	38.4	37.8	-	47.5	45.3	43.7	42.4
Entropy	-	21.5	20.7	20.3	**20.7**	-	40.3	39.5	38.3	37.9	-	47.6	46.0	43.8	43.3
EL2N	-	21.1	20.8	20.2	19.8	-	40.7	39.7	38.4	37.9	-	47.1	45.1	43.5	41.8
AUM	-	22.0	20.7	20.4	20.2	-	40.8	40.0	38.7	38.4	-	47.7	45.7	44.0	43.0
CCS	-	21.6	20.8	20.6	20.0	-	40.4	39.5	38.9	38.2	-	47.4	45.6	44.3	43.9
Ours	-	**22.4**	**21.1**	**21.4**	20.6	-	**41.1**	**40.4**	**40.5**	**39.5**	-	**48.3**	**47.7**	**46.0**	**45.3**
Diff.	-	**+0.6**	**+1.5**	**+3.0**	**+2.2**	-	**+0.9**	**+2.0**	**+3.4**	**+2.7**	-	**+0.5**	**+1.3**	**+1.9**	**+1.0**

(b) The bbox AP (%) results for different object areas (small, medium, large) compare different dataset pruning baselines on COCO.

Table 8: More results on COCO. 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%. `Diff.` denotes the difference between our method and random pruning, and the improvement in time is the average improvement.

D.3 Cityscapes result on mAP, AP50, and AP75

Tab. 9 shows more detailed AP results on the Cityscapes dataset for different IoU thresholds. Our TFDP consistently achieves stable improvements.

	mAP					AP_{50}					AP_{75}				
	0%	20%	30%	40%	50%	0%	20%	30%	40%	50%	0%	20%	30%	40%	50%
Random	12.5	11.2	9.1	8.0	7.2	27.6	26.1	21.8	19.0	16.9	10.0	8.7	6.7	6.1	5.3
Forgetting	-	11.3	10.3	8.3	7.3	-	25.8	23.2	19.3	17.1	-	8.2	7.6	5.7	4.9
Entropy	-	11.5	10.2	8.4	7.0	-	26.4	23.8	20.7	17.2	-	8.6	7.4	5.8	5.1
EL2N	-	11.6	10.1	9.2	**8.0**	-	26.2	23.1	20.8	17.9	-	8.9	7.7	7.4	**6.6**
AUM	-	11.1	10.7	9.2	7.8	-	25.3	24.5	21.2	18.4	-	8.1	8.1	7.0	6.2
CCS	-	10.9	10.4	8.5	7.0	-	25.4	24.0	20.0	17.0	-	8.2	8.0	6.8	5.0
Ours	-	**12.4**	**10.9**	**9.6**	7.5	-	**27.5**	**25.4**	**23.3**	**19.4**	-	**9.5**	**8.2**	**7.2**	5.4
Diff.	-	**+1.2**	**+1.8**	**+1.6**	**+0.3**	-	**+1.4**	**+3.6**	**+4.3**	**+2.5**	-	**+0.8**	**+1.5**	**+1.1**	**+0.1**

(a) The mask AP (%) results for different IoU thresholds (0.5 to 0.95, 50, 75) compare different dataset pruning baselines on Cityscapes.

	mAP^{bb}					${AP_{50}}^{bb}$					${AP_{75}}^{bb}$				
	0%	20%	30%	40%	50%	0%	20%	30%	40%	50%	0%	20%	30%	40%	50%
Random	15.7	14.5	12.1	10.6	9.3	33.0	31.7	27.2	24.2	21.7	12.3	11.2	8.7	7.7	6.7
Forgetting	-	13.9	13.1	11.0	9.7	-	31.0	29.3	26.4	22.0	-	11.4	9.9	7.4	6.6
Entropy	-	14.8	13.4	11.5	9.4	-	31.6	29.6	26.5	21.6	-	11.8	10.1	8.0	6.8
EL2N	-	14.9	13.3	11.7	**10.4**	-	32.0	29.2	26.0	23.0	-	11.8	9.8	9.1	7.2
AUM	-	14.3	13.2	12.0	10.3	-	31.0	29.9	26.5	**23.8**	-	11.1	10.0	9.0	**7.3**
CCS	-	13.5	13.3	10.9	9.4	-	30.1	29.7	25.3	21.6	-	10.0	9.6	7.8	6.7
Ours	-	**15.3**	**14.1**	**12.6**	10.0	-	**32.4**	**31.2**	**28.5**	23.5	-	**11.3**	**10.1**	**9.1**	6.6
Diff.	-	**+0.8**	**+2.0**	**+2.0**	**+0.7**	-	**+0.7**	**+4.0**	**+4.3**	**+1.8**	-	**+0.1**	**+1.4**	**+1.4**	**-0.1**

(b) The bbox AP (%) results for different IoU thresholds (0.5 to 0.95, 50, 75) compare different dataset pruning baselines on Cityscapes.

Table 9: More results on 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 $p = 0\%$. `Diff.` denotes the difference between our method and random pruning, and the improvement in time is the average improvement.

D.4 Cross-Architecture Generalization Experiments

Tab. 10 shows more detailed results on models with different architectures (such as SOLO-v2 and QueryInst) for different IoU thresholds, and our TFDP consistently achieves stable improvements. Since the adapted baselines rely on model-specific pruning, the data selected using Mask R-CNN shows varying degrees of degradation when applied to other architectures. This is especially evident in the Transformer-based model QueryInst, where the data selected by CNN-based Mask R-CNN performs worse than random selection under many pruning rate settings.

	SOLO-v2						QueryInst					
$p = 50\%$	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	mAP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Random	31.3	51.1	32.6	11.6	34.2	48.4	33.3	52.8	35.6	15.0	35.4	51.8
Entropy	31.4	51.6	33.0	12.5	35.0	46.6	33.5	53.9	35.6	16.1	36.3	49.4
EL2N	30.7	50.3	32.2	11.5	34.6	45.3	32.8	52.7	35.0	15.7	36.0	48.4
AUM	31.0	51.0	32.3	11.5	34.7	45.6	33.9	54.0	36.4	15.8	36.8	49.9
CCS	31.8	52.1	33.2	12.1	35.5	47.7	33.3	53.5	35.6	15.6	35.9	49.8
Ours	**32.2**	**52.3**	**34.1**	**12.3**	**35.5**	**48.6**	**34.5**	**55.0**	**36.9**	**15.8**	**37.5**	**51.9**
Diff.	**+0.9**	**+1.2**	**+1.5**	**+0.7**	**+1.3**	**+0.2**	**+1.2**	**+2.2**	**+1.3**	**+0.8**	**+2.1**	**+0.1**

Table 10: More detailed results in the generalization ability to different architectures on COCO dataset.

D.5 Network Scaling Experiments

Tab. 11 shows results to verify TFDP's scalability on the COCO dataset for different backbones (such as ResNet-101 and ResNeXt-101). On backbones with more parameters and stronger performance, our TFDP can still further improve the model's performance, demonstrating the good scalability of our method.

		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.

D.6 Analysis on Scale-Invariance Design

To further validate the effect of SI design, which is applied to mitigate the small-scale bias of the SCS, we compare the Average Precision scores at different scales (AP_S, AP_M, AP_L) following the COCO standard metrics before and after applying SI design in Tab. 12. Our results indicate that, prior to applying SI, the SCS scores were significantly biased towards smaller-scale images, resulting in poor performance in AP_M and AP_L. Conversely, after applying the Scale-Invariant design, there was a noticeable improvement in AP_L levels, without any decline in AP_S results.

	30%			40%			50%		
p	AP_S	AP_M	AP_L	AP_S	AP_M	AP_L	AP_S	AP_M	AP_L
w/o SI	20.6	12.9	17.8	20.5	12.2	16.3	19.5	11.3	13.8
w/ SI	22.6	13.4	20.7	21.7	14.1	18.3	20.2	12.5	15.9
$\uparrow$	+2.0	+0.5	**+2.9**	+1.2	+1.9	**+2.0**	+0.7	+1.2	**+2.1**

Table 12: Ablation study of the SI-SCS. Comparison of mask AP for objects of different scales followed by COCO official metrics (Lin et al., 2014).

D.7 More Results on the Second Training Paradigm

As mentioned in He et al. (2017): A major difficulty with the Cityscapes dataset is the limited number of training samples available for certain categories, such as *truck, bus* and *train*, which only have around 200-500 examples each, making it challenging to train models effectively. To partially remedy this issue, we follow He et al. (2017) and report the results using COCO pre-training to verify our method further. Specifically, we strictly follow He et al. (2017) and initialize the corresponding 7 categories in Cityscapes using a pre-trained COCO Mask R-CNN model, with the *rider* category being randomly initialized.

In this fine-tuning setting, as shown in Tab. 13, our TFDP achieves more significant improvements. Specifically, our method consistently surpasses random pruning as well as other baselines. Notably, our pruning is better than the full dataset, even at a 50% pruning ratio. This may be due to TFDP pruning some low-quality or noisy data that could negatively impact model performance. Furthermore, in the fine-tuning setting, using the full dataset might lead to overfitting, thereby affecting the final performance. These results demonstrate that our method is not only applicable but also more promising under the fine-tuning setting.

	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**

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

	mAP^{bb}					$AP_{50}{}^{bb}$				
	0%	20%	30%	40%	50%	0%	20%	30%	40%	50%
Random	40.9	39.6	39.6	40.2	39.5	66.3	64.9	64.9	65.2	64.0
Entropy	-	39.5	41.0	39.2	39.4	-	63.9	67.3	66.1	65.4
EL2N	-	38.1	39.3	35.2	37.2	-	62.7	65.1	61.8	62.5
AUM	-	36.3	40.8	39.8	39.0	-	62.6	65.8	64.2	64.0
CCS	-	41.0	41.0	40.1	38.7	-	66.0	66.0	64.8	64.3
Ours	**-**	**42.1**	**41.1**	**41.4**	**42.0**	**-**	**67.4**	**68.7**	**67.5**	**67.9**
Diff.	**-**	**+2.5**	**+1.5**	**+1.2**	**+2.5**	**-**	**+2.5**	**+3.8**	**+2.3**	**+3.9**

(b) The bbox AP (%) results compare different dataset pruning baselines on Cityscapes (pre-trained on COCO).

Table 13: Fine-tuned results with COCO pre-trained Mask R-CNN on Cityscapes.

D.8 Time Consumption Details

To ensure a fair comparison, all time consumption experiments were conducted on a same machine: PyTorch on Ubuntu 20.04, with NVIDIA RTX 3090 GPUs and CUDA 11.3. We used two NVIDIA 3090 GPUs for training and one NVIDIA 3090 GPU for inference.

The following section provides explanations of some time consumption details. EL2N (Paul et al., 2021) and Entropy (Coleman et al., 2019) measure the importance of samples by calculating the distance between output logits and the ground-truth one-hot labels, which includes the time for both model training and inference (Scoring). According to the original settings of EL2N, it only requires model weights in the early training stage, hence the training time is shorter than Entropy. AUM (Pleiss et al., 2020) and Forgetting (Toneva et al., 2019) assess the difficulty of samples by tracking changes in the logits across different epochs during training, thus the process includes model training and scoring through training logs. The time difference between the two is minimal, and we report them as a single method. For CCS, in addition to the time taken to obtain AUM scores, the time for Stratified selection should also be factored in. However, the proposed TFDP does not require a model and only needs the time for scoring through pixel-level annotations, which is significantly less than the time required by existing model-based sample selection methods.

D.9 More Visualizations

To further explore the effectiveness of our method, we provide more visualization on VOC, Cityscapes, and COCO in Fig. 9. Notably, ***SCS*** itself can successfully distinguish "hard" and "easy" images as shown in Fig 9 (a). We can visually tell the top-ranked images are more complex than the least-ranked images. By applying **S**cale-**I**nvariant ***SCS*** , the effect of scale has been eliminated. We can observe from the least-ranked images in Fig. 9 (b) that the criterion changes from simply picking the largest objects to easier shapes (e.g., circles). Lastly, by considering class balance, the criterion selects not only the complex shapes but also the images that can best cover the distribution of the dataset.

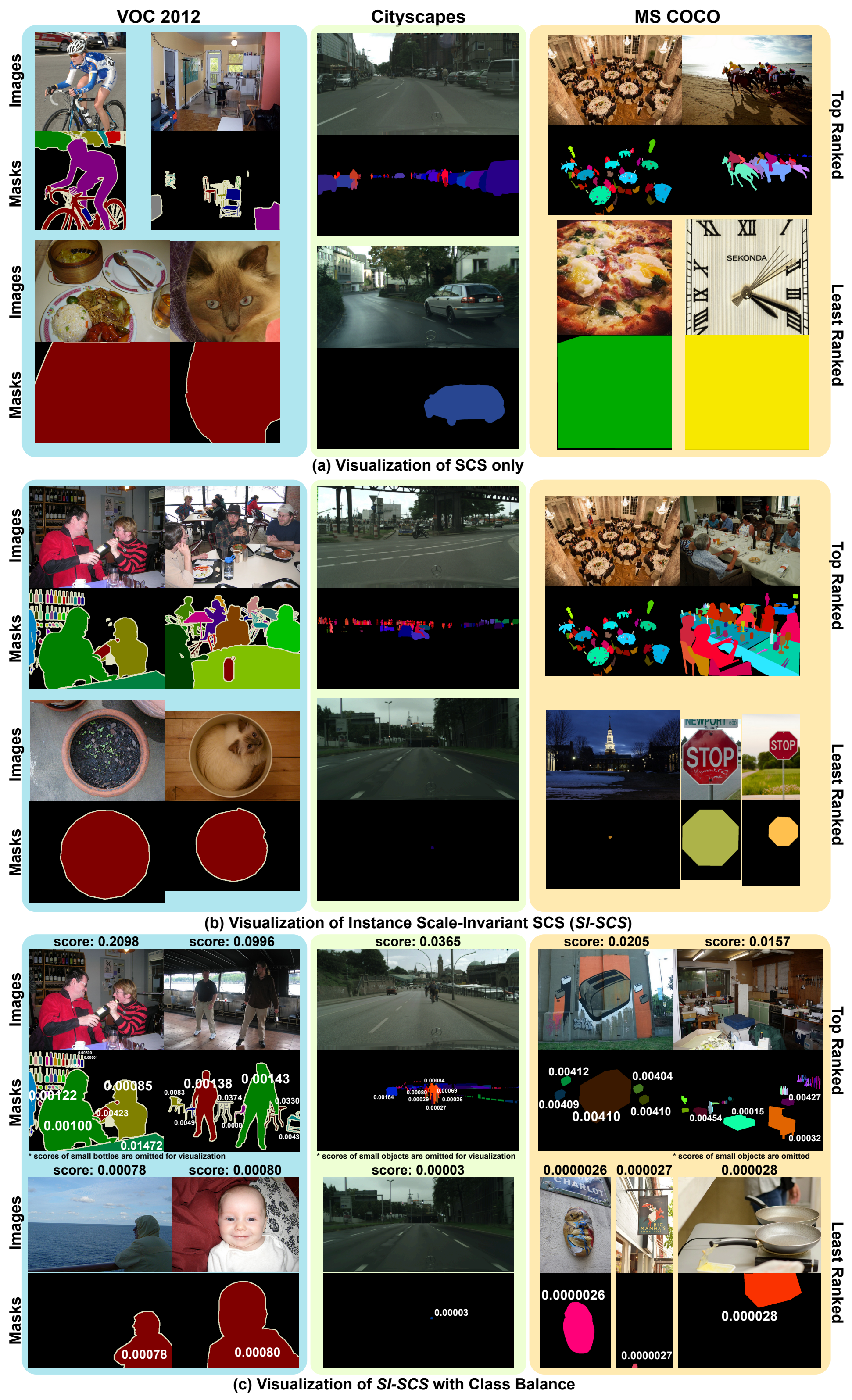

VOC 2012
Cityscapes
MS COCO
Images
Masks
Top Ranked
Least Ranked
(a) Visualization of SCS only
(b) Visualization of Instance Scale-Invariant SCS (SI-SCS)
score: 0.2098
score: 0.0996
score: 0.0365
score: 0.0205
score: 0.0157
* scores of small bottles are omitted for visualization
* scores of small objects are omitted for visualization
* scores of small objects are omitted
score: 0.00078
score: 0.00080
score: 0.00003
0.0000026
0.000027
0.000028
(c) Visualization of SI-SCS with Class Balance

Figure 9: More visualization of the proposed method.

E More explanation and analysis

E.1 More explanation about the formula

In this section, we have provided a detailed explanation of formulae (5) to (12).

Formula (5): Shape Complexity Score. Computation in formula (5), T represents a contour within the binary mask $B_{i,j}$. Specifically, $T^*_{i,j}$ is defined as the primary contour, chosen from a set of contours $\mathcal{T}_{i,j}$ based on its area. This primary contour is selected using the formula:

$$T^*_{i,j} = \arg\max_{T \in \mathcal{T}_{i,j}} \text{Area}(T). \tag{5}$$

Here, $T_{i,j}$ denotes the set of all extracted contours from $B_{i,j}$, and Area(T) computes the area of contour T. The selection of $T^*_{i,j}$ is critical as it captures the most complex part of the instance's shape, influencing subsequent calculations.

Formula (6): Perimeter Calculation. After determining $T^*_{i,j}$, the perimeter $P_{i,j}$ is calculated as:

$$P_{i,j} = \text{Perimeter}(T^*_{i,j}) = \sum_{k=1}^{H_{i,j}} \sqrt{(a_k - a_{k+1})^2 + (b_k - b_{k+1})^2}. \tag{6}$$

Here, $H_{i,j}$ represents the number of points in the contour $T^*_{i,j}$, and (a_k, b_k) are the coordinates of the k-th point. The contour is closed by connecting the last point back to the first, ensuring a complete loop for accurate perimeter measurement.

Formula (7): Area Calculation. The area $A_{i,j}$ of the primary contour is computed using the shoelace formula:

$$A_{i,j} = \text{Area}(T^*_{i,j}) = \frac{1}{2}\left|\sum_{k=1}^{H_{i,j}} (a_k b_{k+1} - a_{k+1} b_k)\right|. \tag{7}$$

This method is accurate for polygons formed by the points of $T^*_{i,j}$, providing a direct computation of the area enclosed by the contour.

Formula (8): Shape Complexity Score. The Shape Complexity Score SCS for the j-th instance in the i-th image is defined by the perimeter-to-area ratio:

$$S_{i,j} = \frac{P_{i,j}}{A_{i,j}}. \tag{8}$$

This ratio is a direct measure of the boundary intricacy, where a higher score indicates a more complex shape.

Formulae (9) to (12): Scale-Invariant SCS (SI-SCS). To mitigate the scale bias inherent in SCS, we introduce the Scale-Invariant Shape Complexity Score (SI-SCS), which normalizes SCS using the ratio for a circle (the geometric shape with the minimum perimeter-to-area ratio) as a benchmark:

$$\frac{P'_{i,j}}{A'_{i,j}} = \frac{f P_{i,j}}{f^2 A_{i,j}} = \frac{P_{i,j}}{f A_{i,j}} \tag{9}$$

$$S^\circ_{i,j} = \frac{2\pi r}{\pi r^2} = 2\sqrt{\frac{\pi}{A_{i,j}}} \tag{10}$$

$$S'_{i,j} = \frac{S_{i,j}}{S^\circ_{i,j}} = \frac{P_{i,j}}{2\sqrt{\pi A_{i,j}}} \tag{11}$$

$$S'_{i,j}(f) = \frac{P_{i,j} \times f}{2\sqrt{\pi \cdot (A_{i,j} \times f^2)}} = \frac{P_{i,j}}{2\sqrt{\pi A_{i_j}}} \tag{12}$$

These formulae confirm that SI-SCS remains constant regardless of the scaling factor f, ensuring that the complexity score is solely dependent on the shape complexity, not the size of the instance.

Overall, our approach significantly reduces computational overhead by leveraging shape information to identify and prune less complex instances, enhancing training efficiency without sacrificing model performance.

E.2 More discussion and derivation of SI-SCS.

In this section, we provide a more detailed explanation and justification for the Scale-Invariant Shape Complexity Score (SI-SCS) in our method.

Problem Identification. The standard Shape Complexity Score (SCS) is sensitive to the scale of instance masks, which can result in a bias: Smaller instance masks tend to have higher complexity scores than larger ones when their perimeter-to-area ratios are computed, given that the score increases disproportionately for smaller areas.

Scale-Invariant Transformation. To address this scale dependency, we introduced the Scale-Invariant Shape Complexity Score (SI-SCS), which normalizes the complexity score by the simplest geometric figure, a circle, known for having the optimal perimeter-to-area ratio. The formulation proceeds as follows:

Standard Shape Complexity Score for a Circle $S^{\circ}_{i,j}$: To establish a baseline for the simplest shape, we compute the SCS for a circle, where the perimeter P and area A relationship is used to derive:

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

Here, r represents the radius of the circle, which relates the area to the perimeter in the most balanced way possible.

Definition of SI-SCS. Using the circle's SCS as a reference, the SI-SCS is defined to normalize the influence of scale on the complexity score:

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

This equation ensures that the complexity score is corrected for any geometric scaling, aligning scores across different sizes by referencing the optimal shape.

Scale Invariance Verification. For a polygon scaled by a factor f, the normalized score remains consistent, confirming the scale invariance:

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

This demonstrates that regardless of how much the instance is scaled, the complexity score remains constant, focusing only on the shape's inherent complexity and not its size.

E.3 More analysis of the experimental results

Table 1: Results on COCO. 1) Our method achieves peak performance improvement at a pruning rate of 40%. 2) As the pruning rate increases, the performance improvement of existing methods gradually diminishes. At a pruning rate of 20%, most baseline methods show minimal improvement, whereas our method further enhances network performance, even surpassing that of the full-dataset-trained model. 3) Random demonstrates strong performance, surpassing some of the baseline methods we implemented at many pruning rate settings. This aligns with findings from dataset pruning in image classification tasks.

Table 2: Results on VOC and Cityscapes. 1) The smaller the dataset size, the greater the performance differences between different pruning rates. 2) The smaller the dataset size, the more significant the performance improvement achieved by our method (up to a 9.7% increase on VOC).

Table 3: Results on Different Architectures. 1) Baselines show limited performance on new architectures, indicating that data selected based on specific models lacks generalization. 2) Our method consistently improves performance across various architectures (including direct methods and transformer-based models), demonstrating that our model-independent approach selects data with strong generalizability.

Table 4: Ablation Study. To understand the effects of different components (SI and CB) presented in Tab. 4, we can first look at the dataset distribution. As provided in Fig. 8 (a) VOC, (b) Cityscapes and (c) COCO, we notice the distribution of the two datasets are both long-tailed; however, the differences between the two distributions are also huge.

The class distributions of the two datasets illustrate a significant difference in their characteristics. In VOC 2012, class frequencies range from 734 (class 1) to 85 (class 20), showing a relatively balanced distribution. While there is some variation, the moderate gap between the most and least frequent classes makes VOC 2012 suitable for evaluating performance metrics like Scale-Invariance (SI) without major concerns about class representation. On the other hand, Cityscapes demonstrates a highly imbalanced distribution, with the most frequent class (class 1) having 27,963 samples compared to only 168 samples for the least frequent class (class 8). Similarly, the class distribution of COCO datasets (Fig. 8 (c)) is also highly imbalanced, and our class balance is more important. This severe imbalance amplifies the need for class balancing (CB) to ensure underrepresented classes are not overlooked, particularly in scenarios involving pruning or model compression.

The VOC dataset's relatively balanced distribution explains why the SI metric plays a pivotal role in enhancing its performance. Under a 30% pruning rate, SI improves the results from 36.6 to 37.8. However, placing excessive emphasis on an already balanced dataset can result in less effective sample selection, leading to a minor performance drop. In contrast, the pronounced imbalance in Cityscapes makes class balancing more impactful. For instance, under a 30% pruning rate, the least represented class in Cityscapes might be entirely ignored if only the SI score is considered. With the inclusion of our CB technique, performance improves significantly, from 22.4 to 24.9.

In summary, while SI is essential for improving overall performance, CB addresses specific limitations by mitigating class imbalances, especially in datasets like Cityscapes. Acting as complementary approaches, the combination of SI and CB delivers the best performance across diverse datasets.

Table 5: Results on Cityscapes (Pre-trained on COCO). 1) Unlike previous dataset pruning methods that train from scratch (Toneva et al., 2019; Paul et al., 2021; Pleiss et al., 2020; Zheng et al., 2023), we are the first to test performance under a pre-training setting, recognizing that COCO pre-trained models are widely used in segmentation tasks (He et al., 2017). 2) Our method performs exceptionally well in the pre-trained setting. Even when pruning 50% of the data, it achieves an AP50 of 62.8%, surpassing baseline performance. This further demonstrates the practicality of our method in real-world scenarios.

Table 6: Detailed Time Calculation. To enhance clarity, we provide a detailed breakdown of the time calculation settings and performance analysis in this table.

1) Settings. To ensure a fair comparison, all time consumption experiments were conducted on the same machine: PyTorch on Ubuntu 20.04, with NVIDIA RTX 3090 GPUs and CUDA 11.3. We used two NVIDIA 3090 GPUs for training and one NVIDIA 3090 GPU for inference.

2) Explanations and analysis. The following section provides explanations of some time consumption details. EL2N and Entropy measure the importance of samples by calculating the distance between output logits and the ground-truth one-hot labels, which includes the time for both model training and inference (Scoring). According to the original settings of EL2N, it only requires model weights in the early training stage, hence the training time is shorter than Entropy. AUM and Forgetting assess the difficulty of samples by tracking changes in the logits across different epochs during training, thus the process includes model training and scoring through training logs. The time difference between the two is small, and we report them as a single method.

3) For CCS, in addition to the time taken to obtain AUM scores, the time for Stratified selection should also be factored in. However, our proposed TFDP does not require a model and only needs the time for scoring through pixel-level annotations, which is significantly less than the time required by existing model-based sample selection methods.

E.4 More details about Shape Complexity Score (SCS).

In the proposed TFDP, we use the perimeter-to-area (P-to-A) ratio as a measure of shape complexity. This approach is not only theoretically sound but also empirically effective in consistently and objectively distinguishing between simple and complex shapes. In this section, we provide a detailed analysis of the P-to-A ratio.

Previous work. The P-to-A ratio is a well-established metric used historically and contemporarily across various scientific fields to measure shape complexity and compactness. It has been utilized since the early 19th century and continues to be a fundamental measure in both theoretical and practical applications (Li et al., 2013).

Moreover, the P-to-A ratio is not only used for geometric analysis (De Smith et al., 2007) but is also widely applied in shape analysis across various fields, including habitat conditions in landscape studies (Wu et al., 2024) and land usage in urban planning (Ayad, 2005).

P-to-A ratio in our paper. The perimeter-to-area ratio is a well-established geometric metric used to quantify the complexity of an object's shape. The key idea behind using the P-to-A ratio as a Shape Complexity Score (SCS) is grounded in the principle that more intricate shapes generally have longer perimeters relative to their areas compared to simpler shapes. This ratio thus provides a straightforward and effective measure of the intricacy and irregularity of the boundaries of instance masks, which are critical features in tasks such as instance segmentation.

1) Theoretical Basis. The P-to-A ratio is inversely proportional to compactness. A circle, which is the most compact shape, has the lowest possible P-to-A ratio. In contrast, shapes with extensions, indentations, or protrusions—features that increase boundary complexity—exhibit higher P-to-A ratios. Thus, the P-to-A ratio directly correlates with the amount of detail and irregularity in a shape's boundary, making it an ideal metric for assessing shape complexity.

2) Model Independence: Unlike metrics derived from model-specific predictions or features, the P-to-A ratio relies solely on the geometric properties of the shape itself. This independence from model biases ensures that the SCS is universally applicable and comparably fair across different datasets and segmentation tasks.

Intuitive example. To effectively demonstrate the utility of the perimeter-to-area (P-to-A) ratio in measuring shape complexity, consider two shapes with identical areas but different boundary configurations—a perfect circle and a gear-shaped circle.

1) Circle: Let's define a standard circle with a radius r . The area A of this circle is given by the formula $A = \pi r^2$. Since a circle has the shortest possible perimeter for a given area, its perimeter P is $P = 2\pi r$. This geometry results in a P-to-A ratio of $\frac{2\pi r}{\pi r^2} = \frac{2}{r}$, which is minimized due to the circle's symmetrical and smooth boundary.

2) Gear-shaped Circle: Now, consider a gear-shaped circle, which is essentially a standard circle of radius r with gear teeth extending outward and inward alternately around its circumference. Suppose each tooth extends an additional length t outward from the circle, and the indentations go t inward. Let's denote the extended radius as $(r+t)$ for the teeth and $(r-t)$ for the indentations. The effective perimeter of this shape will be significantly greater due to the additional length added by each tooth and indentation, calculated as $P' = 2\pi(r+t) + 2\pi(r-t) = 4\pi r$ for one tooth and one indentation respectively, over the entire circumference. Assuming there are n such pairs evenly distributed, the perimeter significantly increases, while the area remains πr^2.

The perimeter-to-area ratio is not only theoretically sound but also empirically effective in distinguishing between simple and complex shapes in a consistent and unbiased manner. This metric is particularly advantageous in our context of dataset pruning, where identifying and focusing on more complex instances can lead to more robust and generalized model training.