DATA-EFFICIENT STREAM-BASED ACTIVE DISTILLATION FOR SCALABLE EDGE MODEL DEPLOYMENT

Dani Manjah, Tim Bary*, Benoît Gérin, Benoît Macq and Christophe de Vleeschouwer*

ICTEAM, UCLouvain, 1348 Louvain-la-Neuve, Belgium

ABSTRACT

Edge camera-based systems are continuously expanding, facing ever-evolving environments that require regular model updates. In practice, complex teacher models are run on a central server to annotate data, which is then used to train smaller models tailored to the edge devices with limited computational power. This work explores how to select the most useful images for training to maximize model quality while keeping transmission costs low. Our work shows that, for a similar training load (*i.e.*, iterations), a high-confidence stream-based strategy coupled with a diversity-based approach produces a high-quality model with minimal dataset queries.

***Index Terms*—** Knowledge distillation, active distillation, dataset pruning, edge computing.

1. INTRODUCTION

Edge camera systems are a cost-effective and privacy-preserving solution for visual analytics tasks such as object detection and classification. As these networks scale to improve coverage and resilience [1], each added sensor introduces a new visual domain, making it infeasible to rely on a single, universal Deep Neural Network (DNN). Additionally, existing sensors are subject to distribution shifts over time [2], requiring frequent model updates.

However, the limited compute and memory of edge devices constrain DNN size and performance [3, 4]. Scalable deployment pipelines are essential for reducing retraining costs and maintaining system reliability.

Stream-Based Active Distillation (SBAD) addresses these challenges by training lightweight *Student* models locally, using data pseudo-labeled by a powerful *Teacher* model [5]. Its on-the-fly confidence-based sampling selects only images the Student is most confident about, reducing noisy labels and enabling continuous adaptation.

While effective, confidence alone can result in redundant datasets and unnecessary data transfer. To improve efficiency, we introduce D-SBAD, which adds an on-device filtering stage to SBAD. As shown in Fig. 1, candidate frames are buffered and filtered using a lightweight embedding model that selects a diverse subset of samples for training.

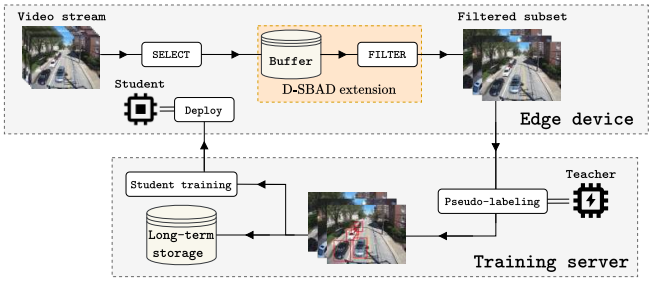

Fig. 1. Candidate frames extracted from the device stream are stored in a buffer, where a pool-based filtering module selects high-quality samples to reduce server-bound training data while maintaining dataset quality. The selected frames are then pseudo-annotated by a teacher model to train the next iteration of the edge model.

Our experiments on 15 WALT cameras [6] show that this two-stage selection significantly reduces bandwidth and training costs while preserving model performance. By maximizing embedding diversity with minimal overhead, D-SBAD enhances the scalability of edge learning systems.

2. RELATED WORK

2.1. Online Knowledge Distillation

Knowledge Distillation (KD) encompasses techniques where a *Student* model learns from another *Teacher* model [7]. This approach can be used to create compact models that are based on the capabilities of a more complex model, and is especially useful when deploying on edge devices, where storage and computational capacity become critical factors [8]. Through regular updates, the *Teacher*'s knowledge ensures that the *Student* model continues to perform effectively even when operating on changing data distributions thanks to its greater robustness [9].

Recently, KD has supported innovative applications in video analytics, particularly through *Online Distillation* [2, 10], where a compact model's weights are updated in

*D. Manjah and T. Bary have equal contributions.

D. Manjah, T. Bary and B. Gérin are funded by the Walloon region under grant No. 8881 (PIT ATMP) and No. 2010235 (ARIAC).

real-time to mimic the output of a larger, pre-existing model.

2.2. Active Distillation

Typical **Active Learning** assumes the presence of a perfect annotator, an assumption that does not hold in a distillation framework where pseudo-labeling can lead to the accumulation of errors due to incorrect predictions (*i.e.*, *confirmation bias*) in semi-supervised or unsupervised learning [11]. To mitigate noisy labels in KD, the **Active Distillation (AD)** framework has been proposed by [12, 5, 13] as a query-efficient AL method that is robust to imperfections in the *Teacher* model. Specifically, in **SBAD** [5], the authors demonstrated that high-confidence automatic labeling of training samples plays a crucial role in the AD selection process. By selecting samples where the Student model exhibits high confidence, the framework reduces labeling errors and enhances the reliability of the distillation process. In this work, we retain the **SBAD** and its top-confidence selection rule due to its effectiveness in handling large volumes of unlabeled images and its ability to continuously adapt to incoming cameras without requiring labeled data at test time [14]. However, we propose a two-stage approach that balances exploration and exploitation, similar to **PPAL** [15]. In that sense, we extend SBAD **with a filtering module designed to identify a subset that is representative of the camera's domain after exploration.**

2.3. Pool-Based Active Learning

Pool-based methods aim to select an informative subset from the pool of data available in the buffer. **Uncertainty-based** methods select samples based on predictive confidence scores or error rates. The **Minmax** strategy [16] selects the least confident samples from the unlabeled pool, assuming that low-confidence predictions are more informative for training. **Error-based** strategies refine this process by selecting samples where predicted and actual errors diverge. Wu et al. [17] proposed **Entropy-Diversity Sampling**, combining entropy-based NMS (Non-Maximum Suppression) with a diversity-based prototype selection phase to maximize intra- and inter-class variation. **Density-based** methods select samples that best represent the underlying data distribution. Yang et al. [15] proposed **PPAL**, a two-stage strategy using uncertainty sampling followed by category-conditioned similarity matching to ensure diversity in the selected set.

2.4. Dataset Pruning

Dataset pruning belongs to coreset techniques with the focus of selecting a representative subset from a larger dataset to maintain or enhance model performance while reducing training data size [18]. **Entropy** [19] is a classic approach but primarily limited to classification tasks. Many state-of-the-art methods, such as **CSOD** [20] or **AUM** [21], rely on ground truth labels. **CSOD** trains a model to identify informative samples, while **AUM** removes mislabeled or ambiguous data using ground truth information. In edge computing environments, utilizing the edge model in a self-supervised manner can introduce confirmation bias, particularly given the model's limited capacity. Alternatively, querying a larger, general-purpose *Teacher* model, or an *Oracle*, conflicts with scalability objectives due to increased bandwidth usage and annotation costs associated with transmitting images.

Self-supervised methods that compute per-sample importance scores, such as distance to other samples [22], or that utilize clustering heuristics to identify representative samples [15], offer alternatives. Finally, **MODERATE** [23] employs redundancy and representativeness strategies for dynamic and moderate pruning, while **TFDP** introduces a training-free dataset pruning approach for instance segmentation tasks [24]. In this work, we benchmark **MODERATE** and **TFDP** as they focus on reducing redundancy and selecting representative samples without any labels.

3. D-SBAD

Our framework follows a client-server dynamic, where a lightweight, pre-trained *Student* model θ, deployed on an edge device (*e.g.*, a camera), processes a video stream $\mathbf{X}$. To ensure the *Student* remains up-to-date, it is periodically fine-tuned on a server using a curated set $\mathcal{T}$ of training samples selected from $\mathbf{X}$. The server hosts a larger, general-purpose *Teacher* model Θ, which generates pseudo-labels for the selected images. These pseudo-labels act as ground truth during fine-tuning of θ. Since transmitting and storing all frames on the server is infeasible in practice, the selected set $\mathcal{F}$ must adhere to a frame budget B. Our objective is therefore to select the B frames from $\mathbf{X}$ that maximize the *Student*'s performance after training.

3.1. Sample Selection

Due to the client's limited storage, it cannot retain the entire video stream for retrospective sampling. Instead, the client must decide in real time whether an image I_t is informative enough to be transmitted to the server. The baseline approach from [5] selects and immediately transmits B images to the remote server using a given `SELECT` strategy. We propose an enhanced selection pipeline incorporating a filtering step. Instead of selecting exactly B images upfront, the client first applies `SELECT` to construct for an acquisition period T, an initial "candidate set" $\mathcal{S}$ which is temporarily stored in an image buffer. This set is then refined to obtain the final subset $\mathcal{F}$ of size B. This additional step improves the likelihood of selecting high-quality images while maintaining the same bandwidth and storage constraints on the server side. The modified pipeline is illustrated in Figure 1 and detailed in Algorithm 1.

Algorithm 1 D-SBAD

Require: *Student* model θ, *Teacher* model Θ, training frame budget $B \geq 1$, video stream $\mathbf{X}$, window of collection T, `SELECT` strategy, `FILTER` strategy.
Ensure: θ, a fine-tuned *Student* model.

$\mathcal{S} \leftarrow \emptyset$
$t \leftarrow 0$ ▷ Timestamp
for $t = 1, \ldots, T$ **do**
  Observe current frame I_t from $\mathbf{X}$
  if `SELECT`$(I_t) =$ TRUE **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup (I_t,\ \text{Labels} = \emptyset)$
  end if
end for
$\mathcal{F} \leftarrow$ `FILTER`$(\mathcal{S}, B)$
for $I_k \in \mathcal{F}$ **do**
  $\tilde{P}_k \leftarrow \Theta(I_k)$ ▷ Infer pseudo-labels
  $\text{Labels}(I_k) \leftarrow (\tilde{P}_k)$ ▷ Update Pseudo-Labels
end for
$\theta \leftarrow train(\theta, \mathcal{F})$

4. MATERIALS AND METHODS

4.1. Dataset

We use the Watch and Learn Time-lapse (WALT) dataset [6], which consists of footage from 15 cameras capturing vehicle circulation in public spaces over one to five weeks. The dataset includes various lighting, weather conditions, and viewpoints, with sample counts ranging from 5,000 to 40,000 frames per week due to varying recording rates. Each camera has a dedicated test set. Together, these test sets total 2450 images and 16,877 human-annotated vehicle instances, as per the instructions in [5]. Images are used without preprocessing, except for grouping COCO labels [25] "bikes, cars, motorcycles, buses, and trucks" into a "vehicle" category.

4.2. Models

We employ the YOLO11 architecture [4] for both the student and teacher models: the *Student* model θ is a YOLO11n (3.2M parameters), and the *Teacher* Θ is a YOLO11x (68.2M parameters). Both are initialized with pre-trained weights from the COCO dataset [25]. The *Student* model is fine-tuned with a constant iteration criterion. For a fixed batch size of 16, the number of epochs is adjusted in inverse proportion to the training set size B to maintain the same computational cost across different set sizes. Other training parameters follow the defaults from [4].

4.3. Evaluation

The performance of the *Student* models is evaluated using the mAP_{50-95} metric, which calculates the mean Average Precision (mAP) at intersection-over-union (IoU) thresholds ranging from 0.50 to 0.95 in increments of 0.05.

4.4. Initial Sampling

We sample using `TOP-CONFIDENCE` [5], where an image is transmitted if the maximum *Student's* confidence exceeds a threshold, computed as the $1 - \alpha$ quantile ($\alpha = 0.1$) during a warm-up phase of size $w = 720$ images.

4.5. Filtering Strategies

We benchmark filtering methods, compatible with edge constraints and that do not require ground-truth labels.

- **Farthest First (FF)** [26]: We provide a deterministic version of the FF algorithm. Namely, FF considers a function ϕ to map any image i to its representation in a latent space. Given a selected set $\mathcal{F}$ and a candidate set $\mathcal{S}$, where $\mathcal{F} \cap \mathcal{S} = \emptyset$, the selection proceeds as follows:

  1. The first image $I_* \in \mathcal{S}$ is selected as the one lying in the densest region of the latent space:
  $$I_* = \arg\max_{I \in \mathcal{S}} \sum_{I' \in \mathcal{S}} \langle \phi(I), \phi(I') \rangle. \quad (1)$$
  2. The remaining $B-1$ images are iteratively added such that they minimize the maximum cosine similarity to the selected set:
  $$I_b = \arg\min_{I' \in \mathcal{S}} \max_{I \in \mathcal{F}} \text{cosim}(\phi(I), \phi(I')). \quad (2)$$

  Our implementation follows Algorithm 2.

- **Training-Free Dataset Pruning (TFDP)**: Initially designed for image segmentation, TFDP [24] computes a *Shape Complexity Score (SCS)* for each instance detected in an image. This score is given by:
$$S_{i,j} = \frac{P_{i,j}}{2\sqrt{\pi A_{i,j}}}, \quad (3)$$
where i is the image, j the instance, $P_{i,j}$ the instance mask perimeter, and $A_{i,j}$ the instance mask area. In our case, we employ the detection boxes as the segmentation mask. The B images with the highest SCS sum constitute the final training set.

- **Moderate Coreset** [23] first embeds all images i into a latent space using a mapping function ϕ. It then selects the B images whose distances from the dataset center are closest to the median distance in the latent space.

- **Least Confidence** [27] selects the B images from $\mathcal{S}$ for which the *Student* model has the lowest confidence in its predicted labels.

- **Random** selects images with a uniform probability. We repeated the experiments with six seeds.

Algorithm 2 Deterministic Farthest First

Require: Set of images $\mathcal{S}$, budget B, mapping function ϕ.

$I_* \leftarrow \arg\max_{I \in \mathcal{S}} \sum_{I' \in \mathcal{S}} \langle \phi(I), \phi(I') \rangle$
$\mathcal{F} \leftarrow \{I_*\}$
$\mathcal{S} \leftarrow \mathcal{S} \setminus \{I_*\}$
for $b \in [1, B-1]$ **do**
    $I_b \leftarrow \arg\min_{I' \in \mathcal{S}} \max_{I \in \mathcal{F}} \text{cosim}\,(\phi(I), \phi(I'))$
    $\mathcal{F} \leftarrow \mathcal{F} \cup \{I_b\}$
    $\mathcal{S} \leftarrow \mathcal{S} \setminus \{I_b\}$
end for
return $\mathcal{F}$

5. RESULTS AND DISCUSSION

We evaluate filtering strategies, and the ratio between collected and transmitted image sets. The temporal collection period T is experimentally simulated, such that the collected set size is a γ-multiple of the image budget B, *i.e.*, $|\mathcal{S}| = \gamma B$.

5.1. Impact of Filtering Strategy

Fig. 2 compares the average mAP_{50-95} for filtering strategies described in Section 4.5, with parameters set as $\gamma = 8$ and training for 100 epochs. Two baselines are included: SBAD selects B frames without filtering ($|\mathcal{F}| = |\mathcal{S}| = B$), while SBAD-$\gamma$ uses the full collected set ($|\mathcal{F}| = |\mathcal{S}| = \gamma B$). Furthermore, we report the performance of pre-trained YOLO11 *Student* and *Teacher*. To ensure fair comparison, epochs for SBAD-γ are reduced to match iteration counts.

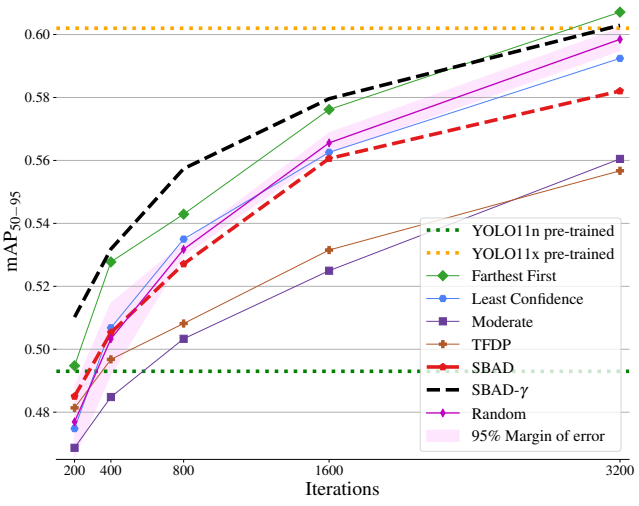

Fig. 2. Performance comparison of *Student* models trained on datasets from various `FILTER` strategies, with $\gamma = 8$. All methods except SBAD-γ use fixed 100 epochs. SBAD-γ training epochs are adjusted by factor $1/\gamma$.

Among evaluated methods, *Farthest First* (FF) yields the highest performance, comparable to SBAD-γ despite using eight times fewer images. Conversely, both TFDP and Moderate Coreset perform below the unfiltered baseline, suggesting suboptimal selection strategies in this use-case. Furthermore, FF provides *Students* superior to pretrained YOLO11n from 200 iterations and can outperform YOLO11x *Teacher* at 3200 iterations. Future work could investigate alternative initial sampling schemes, *e.g.,* the methods of [10, 28] to broaden the range of starting points.

5.2. Impact of the Candidate Set Size

The choice of the initial collection size ($\mathcal{S}$) impacts performance and client-side costs (buffer size and stream duration). Fig. 3 illustrates average mAP_{50-95} for FF as a function of the temporal acquisition T, here defined by the multiplier $\gamma = 1, 2, 4, 8, 12$ and budget $B = 32, 64, 128, 256$.

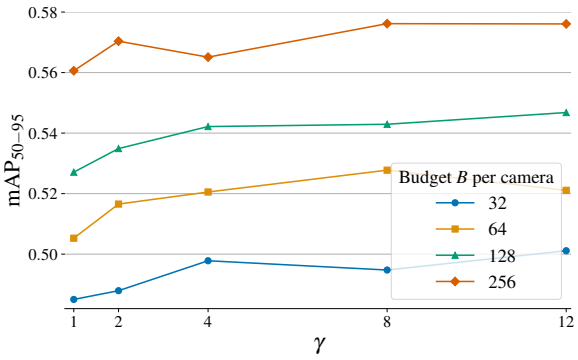

Fig. 3. mAP_{50-95} of the *Student* models trained on datasets generated by the FF `FILTER` strategy, for different values of γ and B. All models are trained for 100 epochs.

Results highlight significant performance gains when increasing from no exploration ($\gamma = 1$) to moderate exploration ($\gamma = 2$). Further increases in γ provide diminishing returns, suggesting an optimal trade-off at moderate collection set sizes. To further enhance bandwidth performance, methods can rely on efficient and secure protocols [29, 30].

6. CONCLUSION

We proposed a filtering step within the SBAD framework to enhance scalability by reducing the amount of transmitted data while maintaining model performance. Experiments demonstrated that the Farthest First algorithm, optimizing latent space coverage, provided the best filtering performance, matching or exceeding baselines using substantially fewer images. Our results confirm the practicality and effectiveness of compact embeddings generated by edge-compatible ViT models such as DINOv2. Future work will investigate lightweight self-supervised pipelines that can be fully executed on edge devices, thereby eliminating the dependency on server-side.

7. REFERENCES

[1] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos, "Sensing as a service model for smart cities supported by Internet of Things," *Transactions on Emerging Telecommunications Technologies*, vol. 25, no. 1, pp. 81–93, 2014.

[2] Dani Manjah, Davide Cacciarelli, Christophe De Vleeschouwer, and Benoît Macq, "Camera clustering for scalable stream-based active distillation," *Expert Systems with Applications*, vol. 290, pp. 128408, 2025.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016.

[4] Glenn Jocher, Ayush Chaurasia, and Jing Qiu, "Ultralytics YOLO," 2023.

[5] Dani Manjah, Davide Cacciarelli, Baptiste Standaert, Mohamed Benkedadra, Gauthier Rotsart de Hertaing, Benoît Macq, Stéphane Galland, and Christophe De Vleeschouwer, "Stream-based active distillation for scalable model deployment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2023, pp. 4998–5006.

[6] N. Dinesh Reddy, Robert Tamburo, and Srinivasa G. Narasimhan, "Walt: Watch and learn 2d amodal representation from time-lapse imagery," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 9356–9366.

[7] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," 2015.

[8] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang, "Model compression and acceleration for deep neural networks: The principles, progress, and challenges," *IEEE Signal Processing Magazine*, 2018.

[9] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada, "Semi-supervised learning for domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33.

[10] Michele Boldo, Mirco De Marchi, Enrico Martini, Stefano Aldegheri, and Nicola Bombieri, "Domain-adaptive online active learning for real-time intelligent video analytics on edge devices," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 43, no. 11, pp. 4105–4116, Nov 2024.

[11] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness, "Pseudo-labeling and confirmation bias in deep semi-supervised learning," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[12] Trong Nghia Hoang, Shenda Hong, Cao Xiao, Bryan Low, and Jimeng Sun, "Aid: Active distillation machine to leverage pre-trained black-box models in private data settings," in *Proceedings of the Web Conference 2021*, 2021, pp. 3569–3581.

[13] Benoît Gérin, Anaïs Halin, Anthony Cioppa, Maxim Henry, Bernard Ghanem, Benoît Macq, Christophe De Vleeschouwer, and Marc Van Droogenbroeck, "Multi-stream cellular test-time adaptation of real-time models evolving in dynamic environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2024, pp. 4472–4482.

[14] Jian Liang, Ran He, and Tieniu Tan, "A comprehensive survey on test-time adaptation under distribution shifts," *International Journal of Computer Vision*, pp. 1–34, 2024.

[15] Chenhongyi Yang, Lichao Huang, and Elliot J. Crowley, "Plug and play active learning for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 17784–17793.

[16] Soumya Roy, Asim Unmesh, and Vinay P Namboodiri, "Deep active learning for object detection," *29th British Machine Vision Conference(BMVC)*, 2018.

[17] Jiaxi Wu, Jiaxin Chen, and Di Huang, "Entropy-based active learning for object detection with progressive diversity constraint," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 9397–9406.

[18] Dan Feldman, Melanie Schmidt, and Christian Sohler, "Turning big data into tiny data: Constant-size coresets for k-means, pca, and projective clustering," *SIAM Journal on Computing*, vol. 49, no. 3, pp. 601–657, 2020.

[19] 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," *CoRR*, vol. abs/1906.11829, 2019.

[20] Hojun Lee, Suyoung Kim, Junhoo Lee, Jaeyoung Yoo, and Nojun Kwak, "Coreset selection for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2024, pp. 7682–7691.

[21] Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger, "Identifying mislabeled data using the area under the margin ranking," in *Advances in Neural Information Processing Systems*, H. Larochelle,

M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, Eds. 2020, vol. 33, pp. 17044–17056, Curran Associates, Inc.

[22] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen, "Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[23] 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 *The Eleventh International Conference on Learning Representations*, 2022.

[24] Yalun Dai, Lingao Xiao, Ivor Tsang, and Yang He, "Training-free dataset pruning for instance segmentation," in *The Thirteenth International Conference on Learning Representations*, 2025.

[25] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014.

[26] Yonatan Geifman and Ran El-Yaniv, "Deep active learning over the long tail," *CoRR*, vol. abs/1711.00941, 2017.

[27] Mingkun Li and Ishwar K Sethi, "Confidence-based active learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 8, pp. 1251–1261, 2006.

[28] C. De Roover, C. De Vleeschouwer, F. Lefebvre, and B. Macq, "Robust image hashing based on radial variance of pixels," in *IEEE International Conference on Image Processing 2005*, Sep. 2005, vol. 3, pp. III–77.

[29] Ayoub Massoudi, Frédéric Lefèbvre, Christophe De Vleeschouwer, and Francois-Olivier Devaux, "Secure and low cost selective encryption for jpeg2000," in *2008 Tenth IEEE International Symposium on Multimedia*, Dec 2008, pp. 31–38.

[30] Cédric Verleysen, Nathalie Merlin, and Christophe De Vleeschouwer, "Adapting jpeg2000 bit allocation to preserve features of interest," in *2011 4th International Congress on Image and Signal Processing*, Oct 2011, vol. 2, pp. 602–606.