BugSR— Improving Tiny Instance Segmentation on the MassID45 Dataset

John Quinto[1,2] **Scott C. Lowe**[2] **Akshita Gupta**[3] **Johanna Orsholm**[4]
Prajakta Darade[5] **Iuliia Zarubiieva**[1,2] **Brendan Furneaux**[6] **Tommi Mononen**[7]
Tomas Roslin[4,7] **Graham W. Taylor**[1,2]

[1]University of Guelph [2]Vector Institute [3]Technische Universität Darmstadt
[4]Swedish University of Agricultural Sciences [5]Indian Institute of Technology Indore
[6]University of Jyväskylä [7]University of Helsinki

Abstract

Automated biodiversity monitoring is crucial for addressing the global decline in insect populations. While most vision-based monitoring efforts analyze images of individual specimens, large-scale monitoring efforts yield “bulk images” where thousands of small insects are imaged in a single, high-resolution image. General object detection and instance segmentation models struggle to localize these insects due to the lack of discriminative visual features for tiny insects and the lack of relevant pretraining data. In this work, we present a case study that explores the effectiveness of super-resolution (SR) as a preprocessing step for tiny insect detection. We use the Mixed Arthropod Sample Segmentation and Identification (MassID45) dataset as a testbed for this task. MassID45 is the first dataset of its kind to feature high-resolution bulk images with instance segmentation and taxonomic classification labels for thousands of small, densely-packed insects. Our experiments show that the bilinear interpolation used in previous MassID45 baselines is suboptimal, and that applying more sophisticated upsampling methods boosts performance across multiple instance segmentation architectures. Leveraging several upsampling methods, ranging from bicubic interpolation to more sophisticated GAN and transformer-based SR models, we achieve an average precision score of 52.8% on the MassID45 test set, representing an increase of 9.3 points from the previous baseline. These performance gains are most pronounced for small objects, proving that SR reconstructs visual details that aid in tiny object detection. Ultimately, our work establishes SR as an important step for bulk image analyses and automated biodiversity monitoring efforts.

1 Introduction and Background

Insects represent about half of all biodiversity on Earth [28]. Unfortunately, anthropogenic climate change threatens insect populations worldwide [33, 3], and by extension, the ecosystems and food webs that insects support [24]. A major bottleneck in understanding this crisis is the shortage of taxonomic experts [36, 22]. Computer vision can address this limitation by automatically analyzing images of bulk samples, where multiple specimens are imaged all at once. These images derived from bulk samples are referred to as *bulk images*.

From a computer vision perspective, detecting and classifying insects from bulk images presents several challenges. First, it is difficult to distinguish and localize small, densely-packed insects from visually similar background debris, which can include loose insect parts. Moreover, a single bulk image can contain several thousand specimens, each of which is only represented by a few

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: The 3rd Workshop on Imageomics: Discovering Biological Knowledge from Images Using AI.

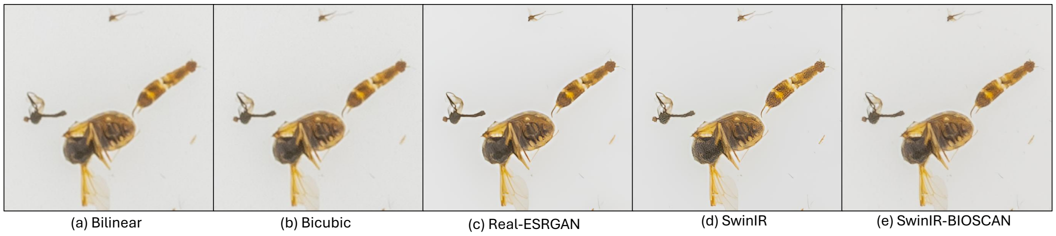

Figure 1: Visual comparison of an example patch upscaled using bilinear interpolation (a) versus the 4 SR methods explored in this work (b-e). Bicubic interpolation (b) produces slightly clearer textures but appears almost identical to bilinear interpolation (a). The deep SR methods — Real-ESRGAN (c), SwinIR (d), and SwinIR-BIOSCAN (e) — generate visually sharp textures and contours, with SwinIR producing more exaggerated textures than Real-ESRGAN and SwinIR-BIOSCAN. Best viewed on a color display with zoom.

pixels. Insects in the resulting bulk image may appear blurry or lack clear morphological details, complicating downstream tasks like detection and classification. To address these challenges, we propose the use of super-resolution (SR), an image processing technique that reconstructs plausible high-resolution details from low-resolution images [13]. While SR has shown promise for small detection tasks in other fields like satellite imagery [25] and plant identification [13], its effectiveness for bulk insect imagery remains unexplored.

Many large-scale insect monitoring efforts rely on bulk samples, as sorting and imaging the specimens individually is labor-intensive. While many computer vision datasets have been developed for images of single insect specimens, there is a significant domain shift between single-specimen images and the bulk images from ecological studies (see Appendix A). To address this challenge, we perform a case study using the Mixed Arthropod Sample Segmentation and Identification (MassID45) dataset, which we introduced earlier in [21]. MassID45 is the first dataset of its kind to feature high-resolution bulk images with instance segmentation and taxonomic annotations for thousands of small, densely-packed insects. Through this work, we demonstrate that super-resolution preprocessing addresses the core challenge of tiny arthropod detection in bulk imagery, establishing SR-enhanced pipelines as an effective paradigm for microscopic biological specimen detection and significantly improving existing instance segmentation benchmarks on MassID45.

2 Dataset and Methods

2.1 MassID45 dataset

The MassID45 dataset addresses the challenge of analyzing dense, unsorted bulk insect samples from real-world ecological surveys [21]. It provides 45 bulk arthropod samples collected with flight interception traps from Sweden and Finland, resulting in 49 high-resolution bulk images with instance-level segmentation masks and taxonomic labels for 17,937 arthropod specimens (see Appendix B for details). In this work, we focus on the first stage of bulk image analyses: detecting insects from background and debris. We frame this task as a single-class instance segmentation problem, using the mask annotations to encode morphological information that can aid downstream tasks like classification and biomass estimation [24, 6]. Given the abundance of small, nearly transparent insects and their high density, MassID45 serves as a challenging benchmark for tiny instance segmentation.

2.2 Experimental setup

Due to the high resolution of the bulk images, processing entire images within GPU memory would be infeasible. Thus, we adopt our same sliding window approach from [21] to divide the bulk images into 512 × 512 pixel tiles. The data preprocessing, partitioning, and augmentation pipelines follow our original study [21] and are described in Appendices C.1 – C.3.

Motivation for SR. The instance segmentation models we evaluate expect a fixed input size of 1024 × 1024 pixels. Thus, the 512 × 512 pixel tiles are upsampled by a factor of two during training and inference, which we found to be the optimal upscaling factor for MassID45 [21]. We initially

Table 1: Comparison of instance segmentation performance for different detectors and upsampling methods. For each metric, the performance change relative to its bilinear interpolation baseline is shown in subscript (blue for improvement ↑, orange for decline ↓). For each detector, the upsampling method with the **best** performance is bolded, and the second-best is underlined for each AP metric.

Detector	Upsampling Method	$AP_{50:5:95}$	AP_{50}	AP_{75}	$AP^{S}_{50:5:95}$	$AP^{M}_{50:5:95}$	$AP^{L}_{50:5:95}$
Mask R-CNN	Bilinear interpolation	42.5	83.1	36.6	20.0	41.6	70.4
	Bicubic interpolation	$\mathbf{52.8}_{\uparrow 10.3}$	$\mathbf{84.3}_{\uparrow 1.20}$	$\underline{57.8}_{\uparrow 21.2}$	$\mathbf{34.3}_{\uparrow 14.3}$	$\mathbf{53.5}_{\uparrow 11.9}$	$\underline{77.2}_{\uparrow 6.80}$
	Real-ESRGAN	$49.8_{\uparrow 7.30}$	$82.1_{\downarrow 1.00}$	$52.2_{\uparrow 15.6}$	$29.1_{\uparrow 9.10}$	$50.6_{\uparrow 9.00}$	$76.4_{\uparrow 6.00}$
	SwinIR	$\underline{52.4}_{\uparrow 9.90}$	$83.6_{\uparrow 0.50}$	$\mathbf{58.0}_{\uparrow 21.4}$	$31.5_{\uparrow 11.5}$	$53.1_{\uparrow 11.5}$	$\mathbf{77.7}_{\uparrow 7.30}$
	SwinIR-BIOSCAN	$52.1_{\uparrow 9.60}$	$\underline{84.1}_{\uparrow 1.00}$	$56.6_{\uparrow 20.0}$	$\underline{32.0}_{\uparrow 12.0}$	$\underline{53.4}_{\uparrow 11.8}$	$76.0_{\uparrow 5.60}$
Mask2Former	Bilinear interpolation	41.4	78.7	37.4	20.5	40.0	71.1
	Bicubic interpolation	$48.7_{\uparrow 7.30}$	$79.3_{\uparrow 0.60}$	$\underline{51.1}_{\uparrow 13.7}$	$30.8_{\uparrow 10.3}$	$48.4_{\uparrow 8.40}$	$\underline{78.1}_{\uparrow 7.00}$
	Real-ESRGAN	$46.6_{\uparrow 5.20}$	$79.9_{\uparrow 1.20}$	$46.0_{\uparrow 8.60}$	$27.1_{\uparrow 6.60}$	$45.1_{\uparrow 5.10}$	$76.8_{\uparrow 5.70}$
	SwinIR	$\mathbf{50.5}_{\uparrow 9.10}$	$\mathbf{81.6}_{\uparrow 2.90}$	$\mathbf{53.3}_{\uparrow 15.9}$	$\mathbf{31.8}_{\uparrow 11.3}$	$\mathbf{50.7}_{\uparrow 10.7}$	$\mathbf{78.3}_{\uparrow 7.20}$
	SwinIR-BIOSCAN	$\underline{49.4}_{\uparrow 8.00}$	$\underline{80.9}_{\uparrow 2.20}$	$51.0_{\uparrow 13.6}$	$\underline{31.5}_{\uparrow 11.0}$	$\underline{48.8}_{\uparrow 8.80}$	$\mathbf{78.3}_{\uparrow 7.20}$
Mask DINO	Bilinear interpolation	43.5	80.9	40.1	21.1	43.5	73.1
	Bicubic interpolation	$\underline{51.3}_{\uparrow 7.80}$	$81.5_{\uparrow 0.60}$	$\underline{54.7}_{\uparrow 14.6}$	$\mathbf{32.9}_{\uparrow 11.8}$	$51.2_{\uparrow 7.70}$	$\mathbf{80.3}_{\uparrow 7.20}$
	Real-ESRGAN	$49.3_{\uparrow 5.80}$	$81.6_{\uparrow 0.70}$	$50.0_{\uparrow 9.90}$	$28.9_{\uparrow 7.80}$	$49.5_{\uparrow 6.00}$	$\underline{77.6}_{\uparrow 4.50}$
	SwinIR	$\mathbf{51.5}_{\uparrow 8.00}$	$\mathbf{82.5}_{\uparrow 1.60}$	$\mathbf{55.4}_{\uparrow 15.3}$	$\underline{31.9}_{\uparrow 10.8}$	$\mathbf{52.3}_{\uparrow 8.80}$	$77.2_{\uparrow 4.10}$
	SwinIR-BIOSCAN	$50.6_{\uparrow 7.10}$	$\underline{82.0}_{\uparrow 1.10}$	$54.5_{\uparrow 14.4}$	$31.2_{\uparrow 10.1}$	$\underline{52.2}_{\uparrow 8.70}$	$77.5_{\uparrow 4.40}$

performed upscaling with bilinear interpolation, the default method in Detectron2 [38]. However, bilinear interpolation is limited; it is unable to reconstruct details lost during imaging as it computes the linear average of neighbouring pixels, leading to blur in the boundaries of existing insect contours. This is problematic for detecting small insects, which are already represented by a small number of pixels. Moreover, image quality is limited by blur from the use of aperture 22, JPEG compression, and other unknown degradations [21]. Thus, we investigate whether more advanced SR techniques can recover these details to improve detection performance.

SR methods. We compare the baseline bilinear upsampling against bicubic interpolation, as well as three SR methods designed to enhance real-world images with unknown degradations: Real-ESRGAN [34], SwinIR [16], and SwinIR-BIOSCAN. SwinIR-BIOSCAN is a SwinIR model fine-tuned on high-quality arthropod images from the BIOSCAN-5M dataset [8] (see Appendix C.4 for details). Implementation details for these SR methods can be found in Appendix C.5, while their effects on a sample image patch are visualized in Figure 1.

Detector architectures. We leverage the same supervised instance segmentation architectures from [21]: Mask R-CNN [11], Mask2Former [4], and Mask DINO [15], improving upon these baselines by incorporating image SR into the augmentation and preprocessing pipelines. Following our previous work [21], we employ transfer learning to train our detectors, then use the SAHI algorithm [2] to map the tiled predictions back to the bulk images. Training and inference details can be found in Appendix C.6.

Evaluation metrics. We report COCO-style average precision (AP) metrics, consistent with our experiments from [21]. This includes $AP_{50:5:95}$ (averaged over IoU thresholds from 50% to 95%), AP_{50}, AP_{75}, and $AP_{50:5:95}$ broken down by object size: $AP^{S}_{50:5:95}$ (small), $AP^{M}_{50:5:95}$ (medium), and $AP^{L}_{50:5:95}$ (large), which are defined in Figure 5 in Appendix B. All evaluation metrics are reported on the MassID45 test partition.

3 Results

3.1 Performance evaluation of SR models

As seen in Table 1, all of the tested SR methods achieve a higher $AP_{50:5:95}$ than bilinear interpolation, irrespective of which detector is used. The top-performing combination is Mask R-CNN with bicubic interpolation, which represents a 9.3% improvement in $AP_{50:5:95}$ from the previous leader Mask DINO.

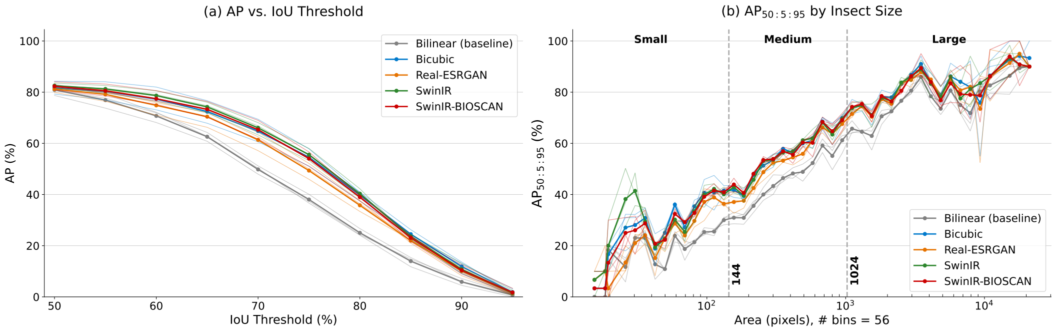

Figure 2: Average precision (AP) for each super-resolution method versus the bilinear interpolation baseline. The AP metrics are averaged across the three detectors, where: a) shows AP at fixed IoU thresholds from 50% to 95% in increments of 5%, and b) shows the AP$_{50:5:95}$ for 56 logarithmically-spaced area bins. Each bold line indicates the detector-averaged AP$_{50:5:95}$, while the thin lines indicate the performance of individual detectors.

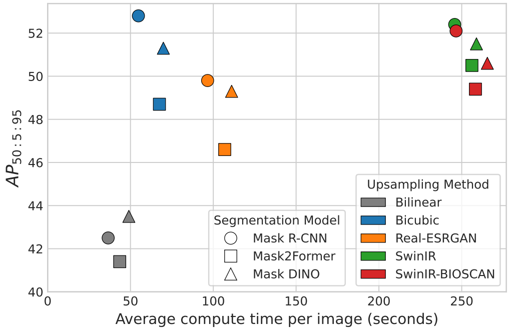

Figure 3: AP$_{50:5:95}$ and average compute times for each pair of upsampling method and detector model. The bilinear interpolation baselines from [21] are shown in gray. Average compute times were computed on the MassID45 test partition (6 bulk images), including the time needed to upsample images and perform inference using each detector. Differences in compute time between SwinIR and SwinIR-BIOSCAN should be interpreted as noise.

We see the most dramatic improvements for AP$_{75}$, which only rewards highly precise predictions (i.e., an IoU of $\geq 75\%$ agreement with the ground truth). For instance, Mask R-CNN and bicubic interpolation raised AP$_{75}$ by 21.4 points, from 36.6% to 58.0%. Thus, the SR methods used in this work reconstruct insect contours more accurately than bilinear interpolation, allowing the instance segmentation models to learn to predict tighter, more precise masks.

Precise instance mask predictions are particularly important for insects that are very small or have fine morphological details like legs, wings, and antennae. We see comparatively smaller gains for AP$_{50}$ (i.e., an IoU of $\geq 50\%$ agreement with the ground truth), which rewards coarse detections and is less sensitive to the vague boundaries produced by bilinear interpolation. The AP performance of each upsampling method across varying IoU thresholds is shown in Figure 2a.

Another key insight lies in the improvements seen when stratifying our performance evaluation by object area. A breakdown of AP$_{50:5:95}$ by insect size is shown in Figure 2b, with the bilinear baseline shown in gray. The general trend is that as the object size increases, AP$_{50:5:95}$ increases because the ratio of border pixels (which are harder to classify) to total pixels decreases. Consequently, SR is most beneficial for small insects, and least beneficial for larger insects. Interestingly, Real-ESRGAN performs worse than the other SR methods for small and medium insects, while SwinIR excels on the smallest insects (<32 pixels). These results confirm that SR enhances tiny instance segmentation through two mechanisms: traditional interpolation methods sharpen existing object boundaries, while GAN-based approaches generate additional discriminative features that aid detection of objects lacking clear visual characteristics.

Despite performing comparably to its pretrained counterpart, SwinIR-BIOSCAN does not surpass the detection performance of SwinIR, which was trained on high-resolution images from various contexts, like outdoor scenes [16, 1, 30]. This result may be due to the abundance of insects with transparent parts in BIOSCAN-5M, particularly specimens of the order *Diptera*, which comprise 63% of the training data for SwinIR-BIOSCAN. Fine-tuning on *Diptera* may encourage SwinIR to generate more transparent textures, which, while more biologically accurate, are harder to detect due to their similarity to the background. Moreover, we did not filter out images where insects are blurry or out of focus, which have been shown to produce misclassifications on BIOSCAN-1M [7]. As a result, SwinIR-BIOSCAN produces contours and textures that are less sharp than SwinIR, but also reduces the prevalence of hallucinated or exaggerated textures (see Figure 1d-e). Future work should explore how well SwinIR-BIOSCAN performs on downstream taxonomic classification tasks compared to bicubic interpolation and the pretrained SR methods.

3.2 Performance–compute trade-off

We also evaluated the trade-off between total compute time and performance for each SR method and detector (see Figure 3). We measured total compute times by recording the average upscaling time and inference time per test partition image. We conducted this analysis for each combination of SR method and detector, measuring the time needed for SR upscaling and model inference times using 1 NVIDIA A40 GPU. We used $\text{AP}_{50:5:95}$ to measure the performance of each detector and SR method.

This analysis reveals that bicubic interpolation and Mask-RCNN achieve the best $\text{AP}_{50:5:95}$ score, while consuming significantly less compute than their deep SR counterparts, and only marginally more compute than the bilinear interpolation baseline. For this dataset, bicubic upscaling is sufficient as the insect localization task is essentially distinguishing foreground from background, which does not require the fine texture details generated by the GAN-based SR methods. While SwinIR-based methods achieve comparable gains as bicubic interpolation — and provide the best performance for Mask DINO and Mask2Former — they require much higher computational resources. For large-scale biodiversity monitoring efforts, the compute-efficient combination of Mask R-CNN and bicubic interpolation is significantly more feasible, given the large number of bulk samples that would need to be processed, and the potential lack of computational resources available, especially if the analysis is to be performed at field research stations.

4 Conclusions

We addressed the challenge of tiny instance segmentation on the densely-packed bulk images of the MassID45 dataset. We demonstrated that all tested SR methods, especially basic bicubic interpolation, substantially improve upon the baseline performance we established in [21]. This improvement stems from addressing the limitations of simple bilinear interpolation, which can blur edges and smooth over visual details that are crucial for detection of tiny insects comprising only tens of pixels.

Our experiments showed that more sophisticated upsampling methods, ranging from bicubic interpolation to transformer-GAN models like SwinIR, can significantly improve detection performance across several instance segmentation architectures. Replacing bilinear upsampling with the SR methods explored in this work resulted in improved baselines for the MassID45 dataset, with the most significant gains being observed for small insects and the AP_{75} metric. Additionally, we investigated whether fine-tuning SR models on high-resolution insect images can improve detection performance. Our fine-tuned SwinIR-BIOSCAN model achieved slightly worse performance than its pretrained counterpart, but visually produced images with fewer hallucinated textures.

Most importantly, our analysis showed that simply replacing bilinear interpolation with bicubic interpolation was sufficient to achieve performance gains, and that deep SR methods did not provide a sufficient benefit to justify the additional compute cost. In fact, the performance gains from bicubic interpolation were most significant for Mask R-CNN, the most compute-efficient model. This result points to the viability of low-compute upscaling and detection methods for vision-based ecological monitoring efforts, where access to intensive computational resources may not be possible.

Our work establishes SR preprocessing as an effective approach for tiny instance segmentation tasks where object detail preservation is critical. The consistent improvements across multiple architectures and SR methods suggest this approach could benefit other domains involving small-scale object detection, such as medical imaging, satellite imagery analysis, and automated quality control in manufacturing. For biodiversity monitoring specifically, the enhanced detection accuracy demonstrated here could enable more reliable automated species counting and distribution mapping in field-collected samples, reducing the manual annotation burden that currently limits large-scale ecological studies.

5 Acknowledgments

The generation of samples were funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 856506; ERC-synergy project LIFEPLAN). Barcoding, metabarcoding and imaging of the samples were funded by the Swedish Environmental Protection Agency (agreement 225-20-002 with TR). This work was supported by the AI and Biodiversity Change (ABC) Global Center, which is funded by the US National Science Foundation under Award No. 2330423 and Natural Sciences and Engineering Research Council of Canada under Award No. 585136. We also acknowledge the support of the Government of Canada's New Frontiers in Research Fund Award No. NFRFT-2020-00073. PD was supported by Mitacs through the Mitacs Globalink Research Internship program. GWT was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Award No. RGPIN-2019-04737, the Canada Research Chairs program Award No. CRC-2021-00561, and the Canada CIFAR AI Chairs program. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute.

References

[1] E. Agustsson and R. Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[2] F. C. Akyon, S. O. Altinuc, and A. Temizel. Slicing aided hyper inference and fine-tuning for small object detection. *2022 IEEE International Conference on Image Processing (ICIP)*, pages 966–970, 2022.

[3] P. Cardoso, P. S. Barton, K. Birkhofer, F. Chichorro, C. Deacon, T. Fartmann, C. S. Fukushima, R. Gaigher, J. C. Habel, C. A. Hallmann, et al. Scientists' warning to humanity on insect extinctions. *Biological conservation*, 242:108426, 2020.

[4] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar. Masked-attention mask transformer for universal image segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1280–1289, 2022.

[5] J. Ding, N. Xue, G.-S. Xia, X. Bai, W. Yang, M. Y. Yang, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang. Object detection in aerial images: A large-scale benchmark and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7778–7796, 2022.

[6] A. Galloway, D. Brunet, R. Valipour, M. McCusker, J. Biberhofer, M. K. Sobol, M. Moussa, and G. W. Taylor. Predicting dreissenid mussel abundance in nearshore waters using underwater imagery and deep learning. *Limnology and Oceanography: Methods*, 20(4):233–248, 2022.

[7] Z. Gharaee, Z. Gong, N. Pellegrino, I. Zarubiieva, J. B. Haurum, S. C. Lowe, J. T. McKeown, C. C. Ho, J. McLeod, Y.-Y. C. Wei, J. Agda, S. Ratnasingham, D. Steinke, A. X. Chang, G. W. Taylor, and P. Fieguth. A step towards worldwide biodiversity assessment: the BIOSCAN-1M insect dataset. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Proceedings of the 37th International Conference on Neural Information Processing Systems*, volume 36 of *NIPS '23*, pages 43593–43619, Red Hook, NY, USA, 2023. Curran Associates Inc.

[8] Z. Gharaee, S. C. Lowe, Z. Gong, P. M. Arias, N. Pellegrino, A. T. Wang, J. B. Haurum, I. Zarubiieva, L. Kari, D. Steinke, G. W. Taylor, P. Fieguth, and A. X. Chang. BIOSCAN-5M:

A multimodal dataset for insect biodiversity. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 36285–36313. Curran Associates, Inc., 2024.

[9] S. Gillies, C. van der Wel, J. Van den Bossche, M. W. Taves, J. Arnott, B. C. Ward, and others. Shapely, Jan. 2025.

[10] Z. Gong, A. T. Wang, X. Huo, J. B. Haurum, S. C. Lowe, G. W. Taylor, and A. X. Chang. CLIBD: Bridging vision and genomics for biodiversity monitoring at scale. In *International Conference on Learning Representations*, 24 Apr. 2025.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[12] A. Jain, F. Cunha, M. J. Bunsen, J. S. Cañas, L. Pasi, N. Pinoy, F. Helsing, J. Russo, M. Botham, M. Sabourin, J. Fréchette, A. Anctil, Y. Lopez, E. Navarro, F. P. Pimentel, A. C. Zamora, J. A. R. Silva, J. Gagnon, T. August, K. Bjerge, A. G. Segura, M. Bélisle, Y. Basset, K. P. McFarland, D. Roy, T. T. Høye, M. Larrivée, and D. Rolnick. Insect identification in the wild: The ami dataset. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XXXVII*, page 55–73, Berlin, Heidelberg, 2024. Springer-Verlag.

[13] T. Jiang, Q. Yu, Y. Zhong, and M. Shao. Plantsr: Super-resolution improves object detection in plant images. *Journal of Imaging*, 10(6), 2024.

[14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2017.

[15] F. Li, H. Zhang, H. Xu, S. Liu, L. Zhang, L. M. Ni, and H.-Y. Shum. Mask DINO: Towards a unified transformer-based framework for object detection and segmentation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3041–3050, 2023.

[16] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte. SwinIR: Image restoration using swin transformer. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 1833–1844, 2021.

[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[18] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR*, 2017.

[19] F. Mazen. Arthropod taxonomy orders object detection in ArTaxOr dataset using YOLOX. *Journal of Engineering and Applied Science*, 70, 09 2023.

[20] H.-Q. Nguyen, T.-D. Truong, X. B. Nguyen, A. Dowling, X. Li, and K. Luu. Insect-Foundation: A foundation model and large-scale 1M dataset for visual insect understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21945–21955, Los Alamitos, CA, USA, June 2024. IEEE Computer Society.

[21] J. Orsholm, J. Quinto, H. Autto, G. Banelyte, N. Chazot, J. deWaard, S. deWaard, A. Farrell, B. Furneaux, B. Hardwick, N. Ito, A. Kar, O. Kalttopää, D. Kerdraon, E. Kristensen, J. McKeown, T. Mononen, E. Nein, H. Rogers, T. Roslin, P. Schmitz, J. Sones, M. Sujala, A. Thompson, E. V. Zakharov, I. Zarubiieva, A. Gupta, S. C. Lowe, and G. W. Taylor. A multi-modal dataset for insect biodiversity with imagery and dna at the trap and individual level. *arXiv preprint arXiv:2507.06972*, 2025.

[22] D. L. Pearson, A. L. Hamilton, and T. L. Erwin. Recovery plan for the endangered taxonomy profession. *BioScience*, 61(1):58–63, 01 2011.

[23] S. Schneider. Summary of feature engineered and deep learning approaches for Animal Re-ID, 2018.

[24] S. Schneider, G. W. Taylor, S. C. Kremer, P. Burgess, J. McGroarty, K. Mitsui, A. Zhuang, J. R. deWaard, and J. M. Fryxell. Bulk arthropod abundance, biomass and diversity estimation using deep learning for computer vision. *Methods in Ecology and Evolution*, 13(2):346–357, 2022.

[25] J. Shermeyer and A. Van Etten. The effects of super-resolution on object detection performance in satellite imagery. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1432–1441, 2019.

[26] L. N. Smith and N. Topin. Super-convergence: Very fast training of residual networks using large learning rates. *arXiv preprint arXiv:1708.07120*, 2017.

[27] D. Steinke, S. Ratnasingham, J. Agda, H. Ait Boutou, I. Box, M. Boyle, D. Chan, C. Feng, S. Lowe, J. McKeown, J. McLeod, A. Sanchez, I. Smith, S. Walker, C.-Y. Wei, and P. Hebert. Towards a taxonomy machine – a training set of 5.6 million arthropod images. *bioRxiv*, 2024.

[28] N. E. Stork, J. McBroom, C. Gely, and A. J. Hamilton. New approaches narrow global species estimates for beetles, insects, and terrestrial arthropods. *Proceedings of the National Academy of Sciences - PNAS*, 112(24):7519–7523, 2015.

[29] A. Svenning, G. Mougeot, J. Alison, D. Chevalier, N. C. Molina, S.-Q. Ong, K. Bjerge, J. Carrillo, T. T. Høye, and Q. Geissmann. A general method for detection and segmentation of terrestrial arthropods in images. *bioRxiv*, 2025.

[30] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, L. Zhang, B. Lim, et al. Ntire 2017 challenge on single image super-resolution: Methods and results. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[31] P. Tresson, D. Carval, P. Tixier, and W. Puech. Hierarchical classification of very small objects: Application to the detection of arthropod species. *IEEE Access*, 9:63925–63932, 2021.

[32] F. O. Ünel, B. O. Özkalayci, and C. Çiğla. The power of tiling for small object detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 582–591, 2019.

[33] D. L. Wagner. Insect declines in the anthropocene. *Annual Review of Entomology*, 65(Volume 65, 2020):457–480, 2020.

[34] X. Wang, L. Xie, C. Dong, and Y. Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 1905–1914. IEEE, 2021.

[35] S. Waqas Zamir, A. Arora, A. Gupta, S. Khan, G. Sun, F. Shahbaz Khan, F. Zhu, L. Shao, G.-S. Xia, and X. Bai. iSAID: A large-scale dataset for instance segmentation in aerial images. In *IEEE/CVF Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.

[36] Q. D. Wheeler, P. H. Raven, and E. O. Wilson. Taxonomy: Impediment or expedient? *Science*, 303(5656):285–285, 2004.

[37] X. Wu, C. Zhan, Y.-K. Lai, M.-M. Cheng, and J. Yang. IP102: A large-scale benchmark dataset for insect pest recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8779–8788, 2019.

[38] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[39] K. Zhang, J. Liang, L. Van Gool, and R. Timofte. Designing a practical degradation model for deep blind image super-resolution. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4771–4780. IEEE, 2021.

Appendices

In these appendices, we review related insect datasets, provide dataset details for MassID45, and further outline our experimental procedures. The appendices are summarized below.

- Appendix A. Literature review of related insect datasets.
- Appendix B. Additional details about the MassID45 dataset, including summary statistics.
- Appendix C. Additional details about our experimental procedures, including data preprocessing, data partitioning, data augmentations, training details for SwinIR-BIOSCAN, implementation details for the different SR methods, and our training and inference pipelines for the instance segmentation models.

A Review of related insect datasets

The threat to insect populations globally has prompted the development of image datasets to support insect monitoring, primarily through taxonomic classification. We review several of these datasets, focusing on their suitability for bulk insect detection and the gaps addressed by MassID45 [21]. Early efforts included the IP102 dataset [37], which featured 75,000 images for identifying agricultural pests. To facilitate the training of foundation models for insect understanding, significantly larger datasets were introduced, including Insect-1M, [20], BIOSCAN-1M [7], and BIOSCAN-5M [8], the latter derived from a raw dataset of 5.6 million Keyence microscope images paired with DNA barcodes [27]. The BIOSCAN datasets are unique in their use of DNA barcoding, a molecular technique that uses short genetic sequences to identify species with high accuracy. In the absence of distinguishing morphological features from images, DNA barcoding enables the development of more accurate multi-modal models for taxonomic classification [10, 8]. These large-scale datasets are crucial for advancing taxonomic studies. However, they only contain images of individual specimens and do not address the challenges associated with bulk images, where insects must first be localized.

As mentioned previously, there is a domain shift between single-specimen images and the bulk images produced by large-scale ecological monitoring programs. To address this gap, Schneider et al. [24, 23] compiled ALUS, the first dataset containing bulk imagery with individual annotations for taxonomic classification and biomass estimation. However, they did not provide box or instance mask annotations, instead using the watershed algorithm to localize insects in their bulk images. [31] provided a dataset with both bounding box and taxonomic annotations, addressing the localization problem by training an object detection model. Their dataset consisted of arthropod images taken in the wild, with some images containing very tiny, densely clustered insects. Similarly, the ArTaxOr dataset [19] provided in-the-wild images with annotations for object detection and taxonomic classification. The taxonomic labels were limited to eight relatively balanced orders, avoiding the long-tailed nature of most taxonomic classification problems [7], but also limiting its value for ecological studies. To facilitate automatic insect monitoring, AMI-Traps [12] introduced a dataset of expert-annotated images taken using insect camera traps. While a custom object detection model was developed to crop moths and other insects from the trap images using synthetic data, no object detection benchmarks were reported, and the predicted boxes were ultimately reviewed and corrected by annotators. Additionally, fine-grained taxonomic labels were only assigned to insects that annotators identified as moths [12].

A step towards automated insect detection was made by [29], which proposed `flatbug`, a dataset and corresponding YOLOv8 model for counting arthropods in images taken from diverse lab or field-based imaging setups. Focusing on class-agnostic instance segmentation and object detection, `flatbug` [29] was trained on arthropod images from 23 arthropod image datasets, some of which were re-annotated to include instance masks. The sub-datasets comprised images taken from the previously mentioned ALUS [24], BIOSCAN [7, 8], AMI [12], and ArTaxOr [19] datasets. Unlike MassID45, which contains an average of 366 insects per bulk image [21], the majority of these datasets do not have a comparable density. Only two of the 23 sub-datasets (totaling 81 images) used in `flatbug` [29] are considered “high” or “very high” density.

Among the datasets above, only [31], `flatbug` [29], and, to a limited extent, AMI-Traps [12] contain ground-truth annotations suitable for training and evaluating detection models on bulk images. Of these, only `flatbug` [29] provides instance mask annotations. However, `flatbug` [29] incorporates

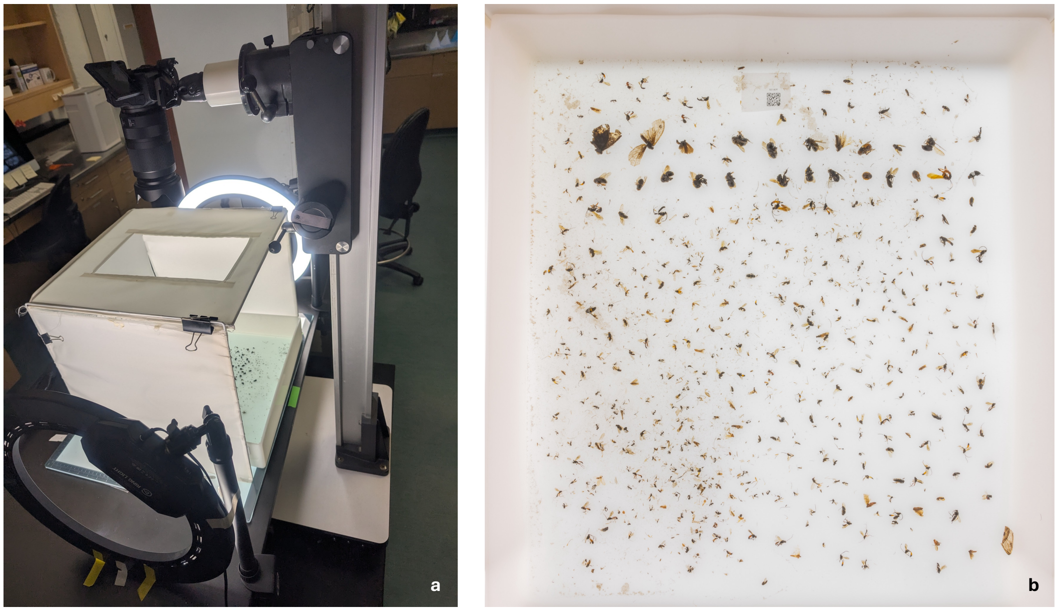

Figure 4: (**a**) Imaging setup used to capture bulk images of the MassID45 dataset, including the positioning of the camera, light cube and ring light sources. (**b**) A representative image captured using the described imaging setup, with the sides trimmed, showcasing the high density of the insect specimens. Images are reproduced from [21].

insect images from multiple domains and/or imaging setups, including some individual specimen images. We avoid these domain shifts, restricting our analyses to bulk images obtained from a standardized imaging and annotation protocol (flight interception traps).

In [21], we introduced the MassID45 dataset to address these gaps, providing much-needed training data for localizing small insects from bulk images. MassID45 also provides DNA barcoding information similar to [7] and [8], although we do not leverage genetic information in this work. We opted to perform insect localization using the instance mask annotations provided by MassID45, as they have several benefits compared to bounding boxes. These include encoding morphological information that may be useful for downstream classification tasks, and enabling biomass estimation [24, 6].

B MassID45 dataset details

The imaging equipment and one example bulk image are shown in Figure 4. Detailed data collection and annotation procedures can be found in our previous work [21].

The 49 annotated bulk images contain masks for 17,937 arthropods, with mask areas ranging between 15.1 and 83,182.4 pixels. Each bulk image contains an average of 366 insect instances, with the most densely-packed bulk image having 3,228 insects. The mean mask area is 1,152.2 pixels, while the median is 343.4 pixels. Figure 5 shows a detailed breakdown of the distribution of insect mask areas. The insect masks are split into 3 categories based on their area. Under the MS-COCO definition of small objects ($<32 \times 32$ pixels in area) [17], 76.5% of the MassID45 insects would be considered as “small”. To increase the granularity in our performance evaluations for different object sizes, we used area thresholds from iSAID [35]. We define “small” as <144 pixels, “medium” as ≥ 144 but <1024 pixels, and “large” as ≥ 1024 pixels.

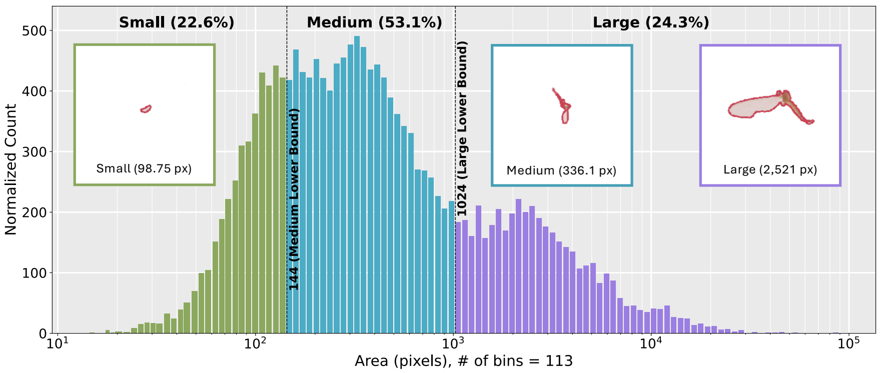

Figure 5: Distribution of insect mask areas in MassID45 [21]. The insect masks are split into 3 groups based on area: "small" (< 144 pixels), "medium" (≥ 144 but < 1024 pixels), and "large" (≥ 1024 pixels) insects. Counts are adjusted such that the area of a bar is proportional to the count in that bin. The three images show the median masks for small, medium, and large insects, all at the same magnification. Figure is reproduced from [21].

C Experiments — additional details

C.1 Data preprocessing

Instance segmentation models require systematic preprocessing to handle the large, complex bulk images in MassID45. Due to the large number of specimens, each bulk image was split into 4 × 4 equally-sized sub-images during the annotation process, with overlapping borders to ensure complete arthropod coverage. Following our methodology from [21], we first mapped sub-image annotations back to the original bulk image coordinates, then used the Shapely library [9] to correct geometric artifacts. Self-intersecting polygons (where annotation boundaries cross themselves) were repaired, and internal holes within insect masks—such as gaps between legs—were filled to create single concave hulls. This hole-filling prevents models from incorrectly learning that arthropods should contain background regions and makes the learning task easier. Each bulk image was manually cropped to remove areas without specimens, reducing computational overhead while preserving all annotated insects.

The resulting images remained too large for direct processing on available GPU hardware, requiring a tiling strategy, similar to [32] and [2]. We divided bulk images into 512 × 512 pixel tiles using a sliding window with the same 60% overlap as [21], ensuring that insects truncated at tile boundaries appear complete in adjacent tiles. Following [5], we retained partially visible annotations only when more than 10% of the original specimen area remained within the tile boundary.

C.2 Dataset partitions

We adopted the same train/validation/test splits we defined in [21] for the 49 bulk images (see Table 2). To prevent data leakage, all image tiles derived from a single bulk image are assigned to the same split. Insect counts include instances that were partially split across tiles.

Table 2: Dataset partitioning and composition. The "# Insects" column denotes the number of insects in each partition after tiling.

Partition	# Bulk Images	% Bulk Images	# Tiled Images	# Insects
Training	40	81.6%	17,062	110,520
Validation	3	6.1%	1,244	5,867
Testing	6	12.2%	1,586	6,241
Total	**49**	**100%**	**19,892**	**122,628**

Table 3: Geometric and color-based data augmentations used for the training data, where p denotes the probability of applying each transformation. Adapted from [21].

Category	Augmentation	Parameters
Geometric	Random horizontal flip	$p = 0.5$
	Random rotation	{0°, 90°, 180°, 270°}, $p = 0.25$ each
Color	Random brightness	Uniform in range $[-15\%, +15\%]$
	Random contrast	Uniform in range $[-10\%, +10\%]$
	Random saturation	Uniform in range $[-15\%, +15\%]$

C.3 Data augmentations

All experiments are conducted on the 512 × 512 pixel tiles extracted from the bulk images. To improve generalization, we employ our data augmentation scheme from [21], implementing geometric and colour transforms aimed at improving robustness to insects in different orientations and lighting conditions. These data augmentations are detailed in Table 3.

C.4 Fine-tuning SwinIR on BIOSCAN-5M

We investigated whether an SR model fine-tuned on arthropod images could reconstruct visual details from our insects more accurately than pretrained SR models. Leveraging our best deep learning-based SR method, SwinIR (see Table 1), we performed fine-tuning on the BIOSCAN-5M dataset, which contains high-quality Keyence images of single arthropod specimens [8].

To ensure the training images from BIOSCAN-5M were taxonomically relevant to the insects in MassID45, we selected a subset of images that matched the range of taxonomic labels in the MassID45 train partition. We performed this matching using the `train` partition of BIOSCAN-5M, which only contains specimens with complete taxonomic labels. Some taxonomic labels in MassID45 corresponded to intermediate ranks not present in BIOSCAN-5M, such as superfamily. For such cases, we resolved the label to the next-highest parent rank (e.g., resolving superfamily *Ichneumonoidea* to its parent order, *Hymenoptera*). Moreover, each labelled insect in MassID45 could be represented by several taxonomic labels, some being high-confidence and some being low-confidence [21]. We aggregated all taxonomic labels from the MassID45 insects in the training partition into a flat list of target labels that could be used to find matching specimen images from BIOSCAN-5M. For each target label, we randomly selected 5 specimen images from BIOSCAN-5M matching that taxonomic target label without replacement. This matching process yielded a training dataset (40,540 images; 90%) and validation dataset (4504 images; 10%) that could be used for fine-tuning SwinIR.

Following previous work from [34], we then generated high-resolution (HR) and low-resolution (LR) image pairs for training. We obtained these training pairs "on-the-fly" using the degradation models from [34, 39], which applies various image degradations at random to simulate real-world image artifacts. Such degradations included 2× down-sampling, as well as random amounts of Gaussian noise, blur, and/or JPEG compression. We trained the SwinIR-BIOSCAN model to perform 2× upscaling, initializing the network with the pretrained weights from the 2× SwinIR model. During training, unsharp masking (USM) was applied to the ground truth images to improve the visual sharpness of the reconstructed images, similar to [34]. The SwinIR model was fine-tuned for 5000 iterations, using a batch size of 32, a learning rate of 1×10^{-4}, and the Adam optimizer [14]. The training was performed using 4 NVIDIA RTX6000 GPUs.

We then performed inference on the degraded validation images. We report validation results for the SwinIR-BIOSCAN model, including common SR metrics like Peak-Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM), in Table 4. The fine-tuning process yields considerable improvements in PSNR and SSIM, although these are not necessarily indicative of superior instance segmentation performance on the MassID45 data, which we assess in Section 3.1 above.

Table 4: Peak-Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) metrics for the finetuned SwinIR model on a validation set of 4504 images from the BIOSCAN-5M dataset.

Model	PSNR (dB)	SSIM
SwinIR	29.52	0.8759
SwinIR-BIOSCAN	**31.43**	**0.8942**

C.5 SR implementation details

For Real-ESRGAN, we used the pretrained `RealESRGAN_x2plus.pth` checkpoint [34], and for SwinIR, we used the `003_realSR_BSRGAN_DFO_s64w8_SwinIR-M_x2_GAN.pth` checkpoint [16].

Bicubic and bilinear interpolation could be directly applied to the full bulk images, yielding images 2× larger in height and width. For these two interpolation methods, we used a CPU with 8 cores. We used 1 NVIDIA A40 GPU when performing inference with the Real-ESRGAN, SwinIR, and SwinIR-BIOSCAN models. Due to GPU computing limitations, however, the Real-ESRGAN and SwinIR models could not ingest full bulk images. When upscaling with Real-ESRGAN or SwinIR, the bulk images were divided into 512×512 shards, then 2× SR was performed on each shard.

Real-ESRGAN and SwinIR used different strategies for stitching the shards to form 2× upscaled bulk images. Real-ESRGAN processed shards that were larger than 512×512, allowing the SR model to see context beyond the shard. When stitching the patches back together, only the output corresponding to the original shard boundaries was retained, with the processed padded regions being discarded. Conversely, SwinIR only processed the regions within the shards, but added overlap between the shards. When stitching the upscaled shards back together, SwinIR averaged the outputs in the overlapping regions between the shards.

For Real-ESRGAN, the shard context was increased to 308 pixels to avoid stitching artifacts. For SwinIR and SwinIR-BIOSCAN, an overlap of 32 pixels between shards was sufficient to avoid stitching artifacts. Once the 2× upscaled bulk images were obtained, they were then split into 1024×1024 pixel tiles with 60% overlap for training and inference. These 1024×1024 pixel tiles contained the same image content as the original 512×512 tiles, but now upscaled by a factor of 2. Inference with the deep SR models was performed using 1 NVIDIA A40 GPU.

C.6 Detector training and inference details

Training details. All supervised models were implemented using the Detectron2 library [38]. To harness transfer learning, each model was initialized with weights from a ResNet-50 backbone pre-trained on the MS-COCO dataset [17]. We fine-tuned each model for 15,000 iterations with a batch size of 8, using the AdamW optimizer [18] and a one-cycle cosine-annealed learning rate scheduler [26] with a peak learning rate of 5×10^{-5} and a warm-up period of 4,500 iterations. Training was performed using 4 NVIDIA RTX6000 GPUs.

Inference details. Following our methodology from [21], we used the SAHI algorithm [2] to aggregate the tiled predictions and map them back to the original bulk image during inference. We only merged predictions that had an intersection-over-union (IoU) of at least 50% and confidence greater than 25%. After upscaling, the predictions were made on 2× larger tiles; thus, the 2D polygon coordinates of the predicted masks were divided by 2 to match the original scale of the ground truth annotations.