Using Robust Networks to Inform Lightweight Models in Semi-Supervised Learning for Object Detection

Jonathan Worobey, Shawn Recker, and Christiaan Gribble

{jonathan.worobey, shawn.recker, christiaan.gribble}@survice.com

Overview

We propose a semi-supervised model training approach that temporarily utilizes the capacity of robust networks to efficiently train low latency models with limited hand-labeled data and a larger pool of unlabeled data (Figures 1 and 2). This approach results in more accurate lightweight models with minimal cost from hand-labeled data while also providing an efficient way of curating ground-truth datasets.

We test our proposed method on the publicly available Okutama-Action dataset [1]. In our experiments, we test one robust deep object detection network (Faster R-CNN [2] with NASNet [3]) and two lightweight networks based on the SSD [4] meta-architecture (MobileNetV2 [5] and Inception-v2 [6]). All models are fine-tuned from COCO [7] pretrained models.

We consider any image in the inferenced dataset containing an object with less than a 0.5 intersection over union (IoU) with its corresponding ground truth label to be erroneous; results are shown in Figure 3.

We simulate three methods of handling errors to create the final training datasets: (1) Ignore all errors, (2) Discard all erroneous images, and (3) Replace all erroneous image labels with ground-truth labels. Figure 4 shows dataset curation speed-up for each method of handling errors.

The three resulting datasets are used to train our lightweight models; the results are shown in Figures 5 and 6.

Methodology

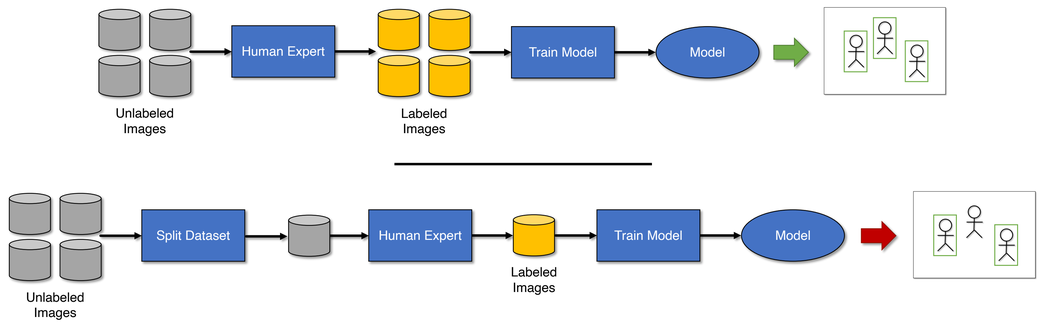


Figure 1. ***Standard method of training a lightweight object detection model.*** A human expert labels every image used to train the model. Generally speaking, using more labeled images (top) leads to improved performance—particularly for lightweight models—while using fewer labeled images (bottom) leads to poor performance. Unfortunately, hand-labeling data is both expensive and time-consuming, thus limiting a researcher's ability to experiment iteratively.

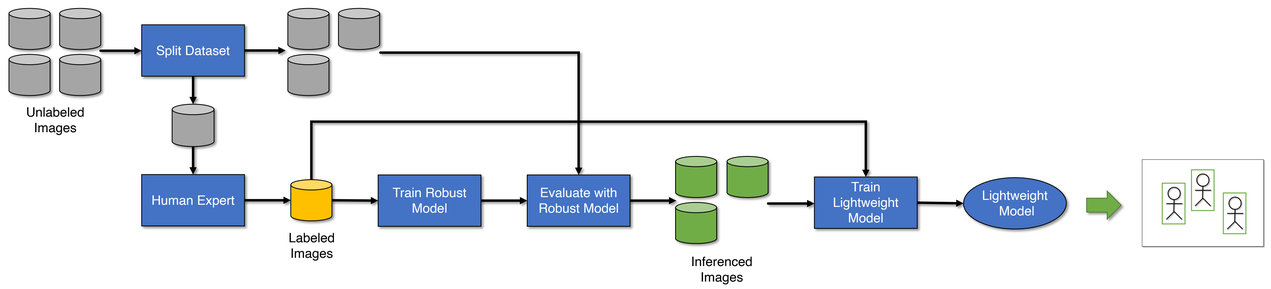
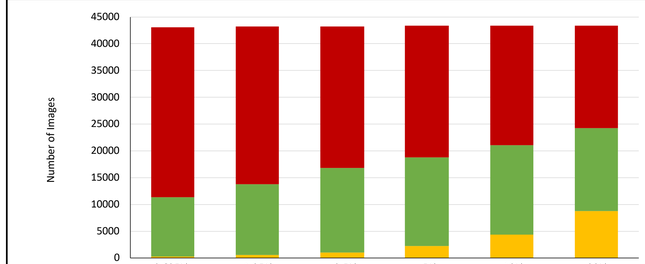


Figure 2. ***Proposed method of training a lightweight object detection model.*** Here, the unlabeled dataset is split into two subsets, one smaller and one larger. The smaller subset is labeled by a human expert, minimizing the hand-labeling cost. A robust object detection model is trained on this small, ground-truth subset. We then evaluate the larger, unlabeled portion using the robust model to create the inferenced dataset. The combined ground-truth subset and the inferenced dataset are then used to train the lightweight model. As in the standard approach, using more labeled images leads to improved performance; however, with our approach, costs associated with hand-labeling images are significantly reduced.

Inferenced Results



	0.625%	1.25%	2.5%	5%	10%	20%
Erroneous	31826.0 ± 622.3	29464.2 ± 1046.9	26499.0 ± 557.4	24582.2 ± 711.4	22324.0 ± 841.2	19060.8 ± 221.7
Correctly Inferenced	11022.8 ± 633.0	13221.0 ± 1015.0	15689.0 ± 546.6	16583.2 ± 697.5	16721.4 ± 829.8	15630.0 ± 219.9
Human Expert Labeled	271	543	1087	2174	4348	8697

Figure 3. ***Distribution of inferenced datasets.*** Image examples are classified after the inferenced dataset is created. Erroneous images are handled in various ways before training lightweight models. (Image count ± standard deviation.)

Dataset Curation Speed-Up

Split Ratio	0.625%	1.25%	2.5%	5%	10%	20%
Ignored	72.63x	50.21x	30.55x	17.54x	9.32x	4.86x
Discarded	1.27x	1.63x	2.05x	2.09x	1.93x	1.65x
Replaced	1.24x	1.27x	1.35x	1.35x	1.33x	1.31x

(a) *Curation wall-clock time speed-up factors compared to the standard method of labeling the full Okutama-Action training set. (Higher is better.)*

	0.625%	1.25%	2.5%	5%	10%	20%
Ignored	2.23x	1.61x	1.30x	1.15x	1.07x	1.04x
Discarded	23.60x	12.02x	6.00x	3.45x	2.15x	1.50x
Replaced	130.39x	64.02x	29.47x	14.92x	7.53x	3.85x

(b) *Curation wall-clock time factors compared to the standard method of labeling only the training set of the split ratio. (Lower is better.)*

Figure 4. ***Dataset curation comparisons.*** Calculations assume a human-expert label time of 30 seconds per bounding box, a review time of 0.0167 seconds (60 fps) per image, and a discard time of 5 seconds per image, a computer training time of 10 hours, and a computer inference time of 0.5 seconds per image.

Definitions

Split Ratio: Ratio between the size of the training dataset and the size of the combined training and inferenced (unlabeled) datasets.

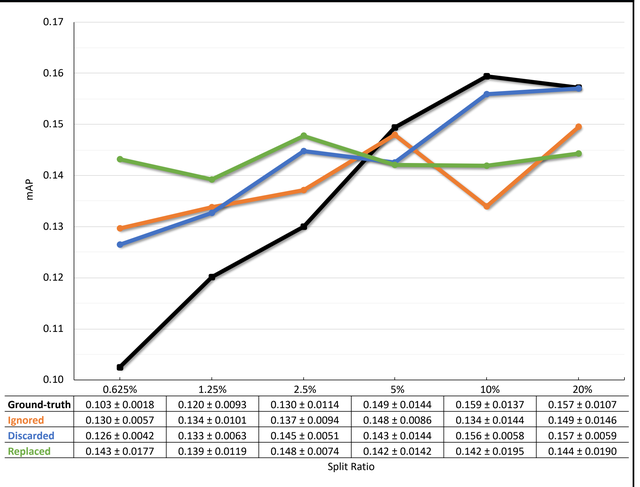
Inferenced Dataset: Dataset automatically generated by evaluating the unlabeled dataset using the robust model.

Ignored Dataset: Raw inferenced dataset created by the robust model—erroneous images are left unhandled. Easy to create, but achieves the weakest performance.

Discarded Dataset: Inferenced dataset minus any erroneous images. Ideally requires minimal human-expert oversight to remove poor examples.

Replaced Dataset: Inferenced dataset with all erroneous image labels replaced with ground-truth labels.

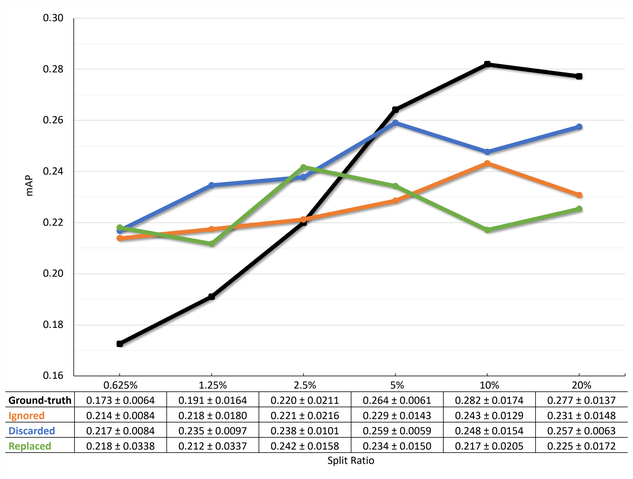
SSD with MobileNetV2



	0.625%	1.25%	2.5%	5%	10%	20%
Ground-truth	0.103 ± 0.0018	0.120 ± 0.0093	0.130 ± 0.0114	0.149 ± 0.0144	0.159 ± 0.0137	0.157 ± 0.0107
Ignored	0.130 ± 0.0057	0.134 ± 0.0101	0.137 ± 0.0094	0.148 ± 0.0086	0.134 ± 0.0144	0.149 ± 0.0146
Discarded	0.126 ± 0.0042	0.133 ± 0.0063	0.145 ± 0.0051	0.143 ± 0.0144	0.156 ± 0.0058	0.157 ± 0.0059
Replaced	0.143 ± 0.0177	0.139 ± 0.0119	0.148 ± 0.0074	0.142 ± 0.0142	0.142 ± 0.0195	0.144 ± 0.0190

Figure 5. ***Lightweight model results, SSD with MobileNetV2.*** Trained with five-fold cross validation. (mAP ± standard deviation.)

SSD with Inception-v2



	0.625%	1.25%	2.5%	5%	10%	20%
Ground-truth	0.173 ± 0.0064	0.191 ± 0.0164	0.220 ± 0.0211	0.264 ± 0.0061	0.282 ± 0.0174	0.277 ± 0.0137
Ignored	0.214 ± 0.0084	0.218 ± 0.0180	0.221 ± 0.0216	0.229 ± 0.0143	0.243 ± 0.0129	0.231 ± 0.0148
Discarded	0.217 ± 0.0084	0.235 ± 0.0097	0.238 ± 0.0101	0.259 ± 0.0059	0.248 ± 0.0154	0.257 ± 0.0063
Replaced	0.218 ± 0.0338	0.212 ± 0.0337	0.242 ± 0.0158	0.234 ± 0.0150	0.217 ± 0.0205	0.225 ± 0.0172

Figure 6. ***Lightweight model results, SSD with Inception-v2.*** Trained with five-fold cross validation. (mAP ± standard deviation.)

Conclusion

Both lightweight models show significant improvement at the smaller split ratios, 0.625% to 2.5% for MobileNetV2 and 0.625% to 1.25% for Inception-v2, demonstrating that the inferenced dataset is effective in improving lightweight model performance.

Further, dataset curation speed-up is significant across all split ratios and methods of handling erroneously inferenced images as compared to the standard method of labeling the full dataset.

References

[1] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector" in *European Conference on Computer Vision*. Springer, 2016, pp. 21–37.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks" in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[3] M. Barekatain, M. Martí, H.-F. Shih, S. Murray, K. Nakayama, Y. Matsuo, and H. Prendinger, "Okutama-Action: An Aerial View Video Dataset for Concurrent Human Action Detection" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 28–35.

[4] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning Transferable Architectures for Scalable Image Recognition" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8697– 8710.

[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[6] C.Szegedy, V.Vanhoucke, S.Ioffe, J.Shlens, and Z. Wojna,"Rethinking the Inception Architecture for Computer Vision" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[7] 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 *European Conference on Computer Vision*. Springer, 2014, pp. 740–755.