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

Jonathan Worobey, Shawn Recker, and Christiaan Gribble
Applied Technology Operation
SURVICE Engineering
4695 Millennium Drive
Belcamp, MD 21017 USA
{jonathan.worobey,shawn.recker,christiaan.gribble}@survice.com

***Abstract*—A common trade-off among object detection algorithms is accuracy-for-speed (or vice versa). To meet our application's real-time requirement, we use a Single Shot Multi-Box Detector (SSD) model. This architecture meets our latency requirements; however, a large amount of training data is required to achieve an acceptable accuracy level. While unusable for our end application, more robust network architectures, such as Regions with CNN features (R-CNN), provide an important advantage over SSD models—they can be more reliably trained on small datasets. By fine-tuning R-CNN models on a small number of hand-labeled examples, we create new, larger training datasets by running inference on the remaining unlabeled data. We show that these new, inferenced labels are beneficial to the training of lightweight models. These inferenced datasets are imperfect, and we explore various methods of dealing with the errors, including hand-labeling mislabeled data, discarding poor examples, and simply ignoring errors. Further, we explore the total cost, measured in human and computer time, required to execute this workflow compared to a hand-labeling baseline.**

***Index Terms*—Semi-Supervised Learning, Object Detection**

I. Introduction

We are actively exploring gesture recognition techniques for the control of unmanned aerial vehicles (UAVs), such as the Tactical Resupply Vehicle (TRV) platform depicted in Figure 1. A first step towards this goal is the curation of high-quality datasets featuring examples across a broad spectrum of gestures, locations, camera angles, times of day, human operators, and so forth. Due to the nature of our target use case, we capture high resolution imagery of human subjects at various distances; as a result, subjects sometimes comprise only a small area of the image, as illustrated in Figure 2. Consequently, the localization of human subjects before (or in conjunction with) the classification of their gestures is necessary, regardless of the method we ultimately employ.

At real-time frame rates, a distinct gesture will span dozens of frames. The cost of a human expert to label every image of such datasets is prohibitively expensive. Therefore, we seek new ways of minimizing this human-expert labeling cost.

Fig. 1. *SURVICE Engineering's Tactical Resupply Vehicle (TRV) platform.* We are developing autonomous capabilities for the TRV family of multi-rotor systems. The TRV-400, pictured here, is capable of lifting up to 400 lbs. over multiple kilometers. The platform is currently under development as a logistics resupply vehicle.

Fig. 2. *Single frame of a human subject performing the "Move Right" gesture.* This high-resolution video frame shows a human subject at a distance in a view from a UAV. We are actively exploring gesture recognition techniques for UAV control, and any such algorithm will likely need to localize the human subject before (or in conjunction with) classifying the gesture.

978-1-7281-4732-1/19/$31.00 ©2019 IEEE

State-of-the-art object detection algorithms are based on deep convolutional neural networks (CNN). These networks only reach their full potential when trained on very large datasets (e.g., COCO [1]). While transfer learning front-loads much of the work via model pretraining to minimize both the training time and the amount of data required, many thousands or tens of thousands of examples may still be needed to ensure generalizability of a model. These data are often easy to collect but difficult and expensive to accurately hand-label.

At the same time, a common trade-off among object detection algorithms is accuracy-for-speed (or vice versa). While accuracy is always desired, many use cases require low latency inference for real-time detection. For instance, various object detection tasks on UAVs require rapid and consistent detections using low size, weight, and power (SWaP) devices. These requirements ultimately restrict the number of parameters that can be used and thus limit the capacity of the network.

We demonstrate a semi-supervised 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. In particular, we train a robust object detection model using this small, ground-truth subset and evaluate the larger, unlabeled portion using the robust model to create an *inferenced dataset*. The combined ground-truth subset and the inferenced dataset are then used to train a lightweight model. 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 for gesture recognition.

II. Background

We use two common types of object detection meta-architectures to evaluate our proposed approach: Regions with CNN features (R-CNN) [2] and Single Shot MultiBox Detector (SSD) [3]. Both the R-CNN and SSD families of object detection meta-architectures use pretrained image classification networks as a base feature extraction network. These networks include Inception [4], NASNet [5], MobileNet [6], and others.

R-CNN is a robust, two-stage object detection meta-architecture in which the first stage handles region proposal via selective search and the second handles object detection given these proposed regions. These multiple stages lead to accurate bounding box inferences at the expense of speed, however. The Faster R-CNN [7] meta-architecture improves the speed of R-CNN by targeting the selective search region proposal algorithm, replacing it with the Region Proposal Network (RPN). However, for mobile applications, even Faster R-CNN is typically still too slow for real-time object detection [8].

The single-stage object detection meta-architecture, SSD, targets the deficiencies of two-stage algorithms by detecting objects in a given image via a single pass through the model (i.e., without relying on a region proposal algorithm). SSD is similar to Faster R-CNN's RPN, but instead of passing the region proposals to a box predictor, SSD immediately predicts the region's class [3]. This design is highly flexible, and without distinct region proposal and object detection stages, the model size can be dramatically reduced, ultimately leading to faster inference times.

The R-CNN family of algorithms has one important advantage over single-stage object detection networks—namely, an independent bounding box regression algorithm, downstream of the feature extraction network. In practice, we notice that this additional step typically leads to improved object localization, suggesting that this architecture is ideal for inferring objects for downstream training.

A variety of work has focused on the so-called *teacher-student* training method involving robust and lightweight networks. For example, both Shen et al. [9] and Chen et al. [10] utilize the teacher's (robust) model loss to inform the student (lightweight) model via hint learning and knowledge distillation. Similarly, Li et al. [11] train a student model via feature map mimicking, while Hong et al. [12] use a generative adversarial network (GAN) to train an SSD model. Whereas, Tang et al. [13] train an image classifier on more readily available, and less expensive to label, image classification data. Knowledge transfer via visual similarity is then used to train an object detector.

Other approaches rely on classical techniques to generate training data. For example, Wang et al. [14] use a classical object detection algorithm to suggest objects from unlabeled data and then use soft label boosting to train an object detector.

While the common intent here is to maximize the performance of lightweight models, only our proposed method results in a human-readable dataset. In particular, our inferenced dataset can be viewed by a human expert to both verify the integrity of the labels and, if necessary, fix any erroneous images. Beyond the assurance that the lightweight model is being trained on accurate data, our approach also provides for network architecture independence by decoupling the robust and lightweight networks.

In contrast, Vondrick et al. [15] and Misra et al. [16] also propose methods that result in human-readable datasets. In particular, Vondrick et al. [15] develop an interactive video annotation application that uses classical methods to interpolate bounding box locations between hand-labeled keyframes. Our approach differs from this method in that we use deep learning rather than classical methods for automatic labeling. Misra et al. [16] train an object detection network without a teacher network by training on a small number of examples and iteratively generating new labels through its own inference. Our work differs from this approach in that the robust networks we use ultimately learn features of the dataset with less data than is required in this iterative method.

Our proposed method thus produces a human-readable dataset that eases verification of label correctness, and it decouples training of robust and lightweight networks for reusability. At the same time, the methods proposed by Vondrick et al. [15] and Misra et al. [16] can both be used in conjunction with our proposed approach, either by using classical techniques to interpolate bounding boxes between keyframes or by iteratively training a robust network.

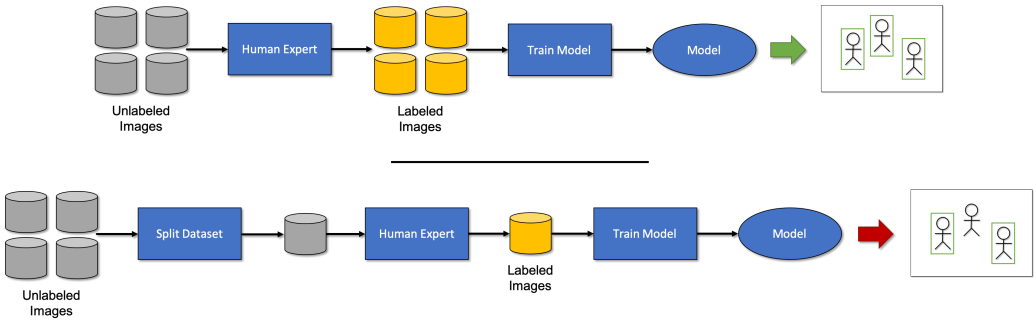


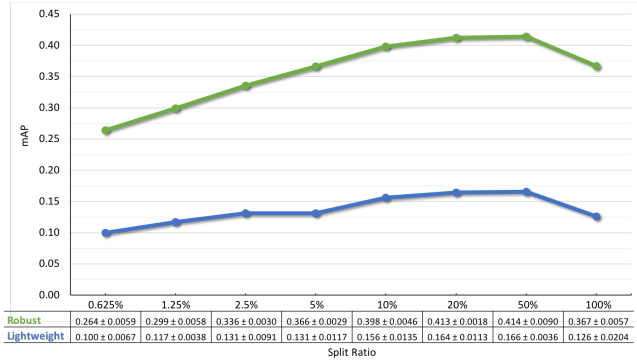
Fig. 3. *Standard method of training a lightweight object detection model.* Here, a human expert labels every image used to train the model. Generally speaking, more labeled images (top) leads to improved performance—particularly for lightweight models—while fewer labeled images (bottom) leads to poor performance. Unfortunately, hand-labeling data is both time-consuming and expensive, and often limits a researcher's ability to experiment iteratively.

III. Method

The standard method of training lightweight networks, in which the burden of labeling large datasets is placed on a human expert, is illustrated in Figure 3. This approach is undesirable due to its high cost, both in wall-clock time and in human-expert time. In fact, the cost of labeling by human experts is often high enough to prohibit a researcher's ability to experiment iteratively, so careful and sometimes tedious planning of data labeling activities is required under the standard method.

Robust deep object detection networks, which favor accuracy over speed, tend to train to an acceptable performance with smaller amounts of data, as demonstrated in Figure 4. Here, we randomly select subsets of various sizes from the Okutama-Action [17] dataset and train both a robust network, Faster R-CNN with NASNet, and a lightweight network, SSD with MobileNetV2, on each subset. We call the ratio between the size of the labeled subset and the size of the labeled plus unlabeled subsets the *split ratio*. Figure 4 shows that our robust network achieves a higher mean average precision (mAP) than our lightweight network at every split ratio—that is, robust networks do, in fact, achieve an acceptable level of performance with training datasets comprising few images than do lightweight networks. This observation leads to our hypothesis that, for certain datasets of sufficient size, there exists a split ratio at which a robust model can achieve near ground-truth level performance while a lightweight model will still benefit from additional training data.

We seek to improve the standard method of training a lightweight model by exploiting the observed behavior of robust and lightweight models in this context. Our proposed approach is illustrated in Figure 5. Specifically, we train a robust model on a small amount of data labeled from a larger pool of unlabeled data. The remaining unlabeled data is automatically labeled by running robust inference on each image and labeling inferenced objects as ground-truth objects. The quality of this inferenced dataset depends on the robust model and, in practical applications, the resulting data is often flawed. We explore various methods of accounting for these flaws, including simply ignoring errors, discarding images



	0.625%	1.25%	2.5%	5%	10%	20%	50%	100%
Robust	0.264 ± 0.0059	0.299 ± 0.0058	0.336 ± 0.0030	0.366 ± 0.0029	0.398 ± 0.0046	0.413 ± 0.0018	0.414 ± 0.0090	0.367 ± 0.0057
Lightweight	0.100 ± 0.0067	0.117 ± 0.0038	0.131 ± 0.0091	0.131 ± 0.0117	0.156 ± 0.0135	0.164 ± 0.0113	0.166 ± 0.0036	0.126 ± 0.0204

Fig. 4. *Robust v. lightweight model inference results.* A robust model (Faster R-CNN with NASNet) and lightweight model (SSD with MobileNet) are trained on the Okutama-Action dataset at various split sizes. As can be seen, performance of Faster R-CNN with NASNet plateaus faster than that of SSD with MobileNetV2.

containing poor examples, and hand-labeling the mislabeled objects.

The desired lightweight model is then trained on the combined inferenced and hand-labeled datasets and typically achieves higher accuracy than if trained on only the hand-labeled data.

IV. Experiments

We test our proposed method on the publicly available Okutama-Action [17] pedestrian dataset, containing 54,356 training images. Several exemplar images from the dataset are shown in Figure 6. Okutama-Action consists of 3840×2160 (4K) video frames tracking human actions via a UAV. For the purposes of this paper, we ignore the specific action labels and object tracking information and consider only the object localization of each visible person.

We select the Okutama-Action dataset for two reasons: First, it is highly representative of our use case, as massive amounts of object detection data can be captured with video cameras, providing an easy and common way of generating training data. Second, datasets generated via video sequences are easily verifiable. One can view a video of raw frames combined with a visualization of the corresponding labels at real-time

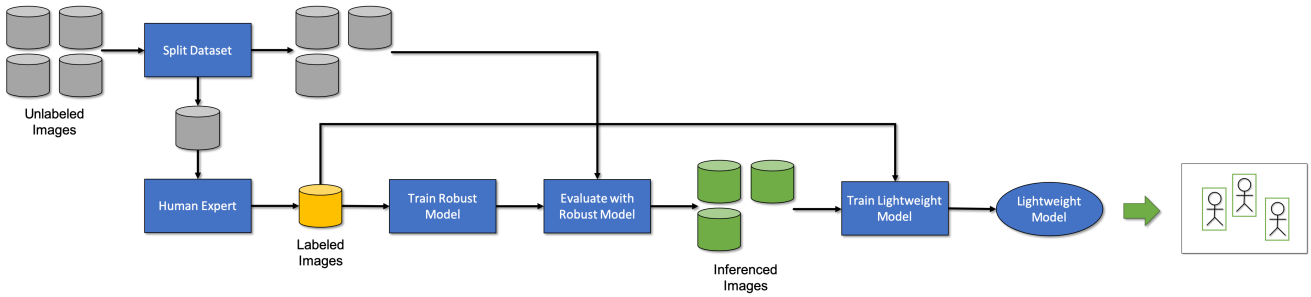


Fig. 5. *Proposed method of training a lightweight object detection model.* The unlabeled dataset is split into two subsets, one smaller and one larger. The smaller subset is labeled by a human-expert, minimizing the hand-labeling cost. A robust object detection model is trained on this small, ground-truth subset. We then evaluate the larger, unlabeled portion using the robust model, creating 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, more labeled images leads to improved performance; with this approach, however, costs associated with hand-labeling images are significantly reduced.

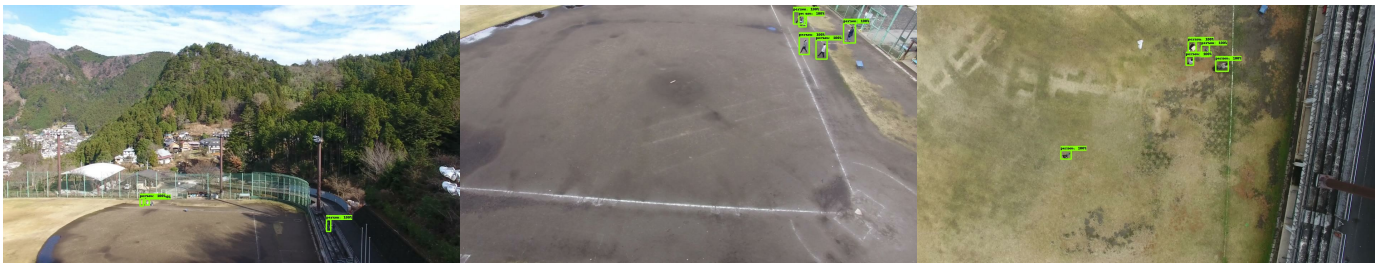


Fig. 6. *Dataset exemplars.* Okutama-Action is a publicly available video dataset that tracks pedestrian locations and actions from an aerial view. The training set consists of 54,356 images with a variable number of persons visible in each frame. In our context, we discard the action labels provided by this dataset and utilize only the objects' localizations instead.

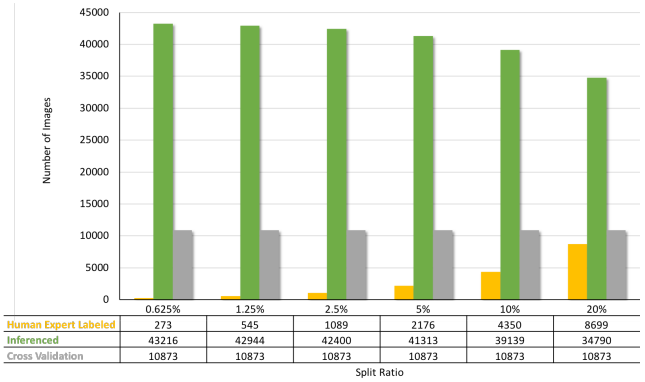
frame rates. This characteristic is in contrast to datasets of arbitrary order and subject matter, which require significantly more human processing time to verify each frame, ultimately reducing the benefit of automatically labeling training data.

A. Network Architectures

In our experiments, we test one robust and two lightweight deep object detection networks. For a robust network, we choose Faster R-CNN with NASNet [5], as it is the most robust object detection network available within the TensorFlow Object Detection API [8]. The two lightweight networks are both based on the SSD meta-architecture. The first uses MobileNetV2 [6] as the feature detector, a smaller network designed for use on limited performance machines such as mobile phones. The second uses Inception-v2 [4], a heavier network still capable of real-time performance on low SWaP devices, such as the NVIDIA Jetson Xavier.

B. Methodology

We perform five-fold cross-validation on each dataset and split ratio. Of the typical 80% remaining training data in a five-fold cross-validation training subset, we further split this data according to the split ratio. For example, given a split ratio of 10%, the ground truth training set is divided into the following subsets: 20% validation, 8% training, and 72% unlabeled. The size of each subset for each split ratio is shown in Figure 7. For each cross-validation iteration, we train a robust model on this small training dataset. Following our proposed method, we then create a new inferenced dataset from the unlabeled dataset using the robust model.

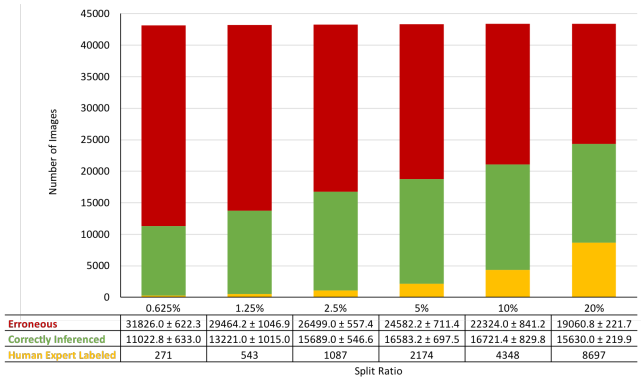


	0.625%	1.25%	2.5%	5%	10%	20%
Human Expert Labeled	273	545	1089	2176	4350	8699
Inferenced	43216	42944	42400	41313	39139	34790
Cross Validation	10873	10873	10873	10873	10873	10873

Fig. 7. *Split ratios used in our experiments.* We perform five-fold cross-validation on each dataset and split ratio. The relative sizes of ground-truth training dataset and inferenced dataset range from 0.625% to 20%, with absolute sizes as indicated.

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. Furthermore, if either the ground truth labels or the inferenced labels contain an object not found in the other, we also consider the inferenced image to be erroneous.

We simulate three different methods of handling these errors, resulting in three new training datasets:



	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

Fig. 8. *Inferenced dataset image distributions.* We train five robust models per split ratio. Each image of the inferenced dataset is considered correctly inferenced or erroneous. Erroneous images are handled in various ways, including ignoring, discarding, and replacing.

1) Ignore all erroneous images, maintaining the inferenced dataset as is.
2) Discard all erroneous images, which simulates the relatively easy task of a human expert scanning through the inferenced dataset and discarding poor examples.
3) Replace all erroneous images, specifically using ground-truth labels, which simulates the costly process of hand-labeling a potentially large number of images.

These three datasets are then used to train both of our lightweight models. We compare the accuracy of these models with models trained on both the full ground-truth dataset and the small hand-labeled dataset in the next section.

V. Results

We fine-tune five Faster R-CNN with NASNet models per split ratio of the Okutama-Action dataset. These split ratios range from 0.625% to 20%. Each model is trained for 100,000 iterations on 1280×720 resolution images at a batch size of one. Further, we use a dropout layer with a 0.5 keep probability in the second stage box predictor to help prevent overfitting. The results for each split ratio are shown in the first six columns of Figure 4.

As described in Section IV-B, we train two types of lightweight models per inferenced dataset. The models of the first type, SSD with MobileNetV2, are each trained for 100,000 iterations on 300×300 resolution images at a batch size of 24, while the models of the second type, SSD with Inception-v2, are each trained for 50,000 iterations on 1280×720 resolution images at a batch size of eight. These training parameters are chosen empirically to maximize training performance and minimize overfitting across split ratios.

The distributions of our inferenced results generated by our robust models are shown in Figure 8. The number of correctly inferenced images increases steadily from the 0.625% split ratio to the 5% split ratio and plateaus at 10%. This number then decreases at the 20% split ratio, due in part to the diminishing size of the unlabeled image pool. However, the total number of high-quality training examples—that is, the sum of hand-labeled images plus correctly inferenced images—grows across the entire range of split ratios.

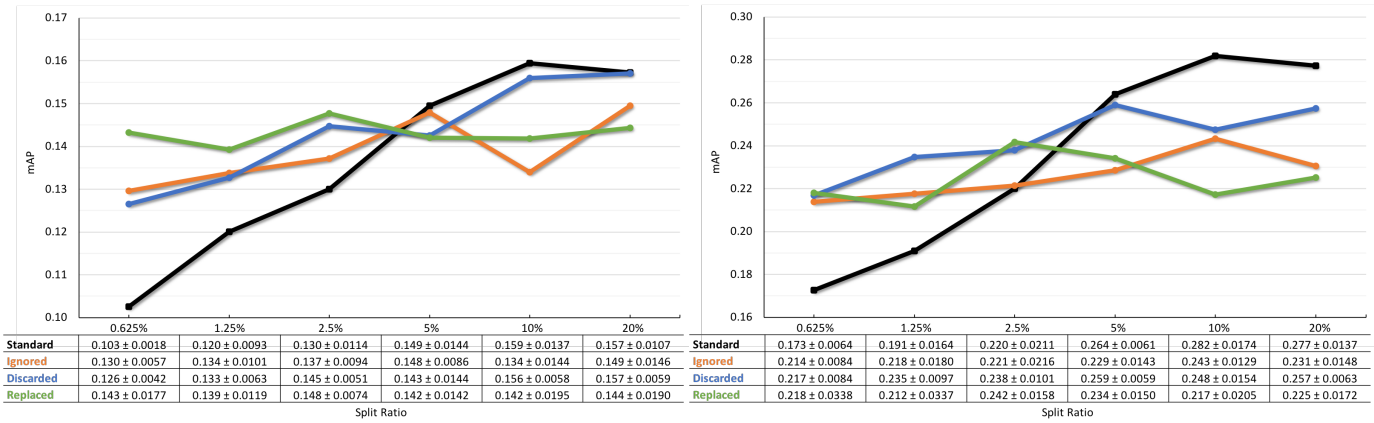
As outlined in Section IV-B, we handle erroneously labeled images in three different ways—ignoring, discarding, or replacing—leading to three inferenced datasets. We also train lightweight models on the full ground-truth dataset and the holdback dataset used to train the robust model. The resulting mean mAP values and standard deviations of our MobileNetV2 and Inception-v2 models are shown in Figure 9. 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 all three forms of the inferenced dataset are effective in improving lightweight model performance.

A. Cost-Benefit Analysis of Human Labeling

Bounding box labeling by human experts is a time-consuming process. Su et al. [18] measure the median time required to label high quality object detection examples in ImageNet [19]. They find the median labeling time to be 34.5 seconds per box, with another 7.8 seconds dedicated to coverage verification, the process of verifying that no more objects exist in the image after the final object is labeled. Additionally, they find the mean times to be substantially larger, at 72.7 seconds per box for labeling and 15.3 seconds for coverage verification. Based on these results, we use a conservative estimate of 30 seconds per bounding box in the following cost estimates.

Our work focuses specifically on video object detection datasets, so we reduce review time during curation of discarded and replaced datasets to the time required by a human expert to view the dataset sequentially. In the ideal case, where every object of the inferenced dataset is labeled correctly, the dataset can be viewed at approximately real-time frame rates, resulting in a review time that is a fraction of the time required to hand-label a dataset of the same size. Unfortunately, little research exists regarding the time required by a human expert to discard erroneous images. However, as a result of spatial locality exhibited by objects between frames, it seems reasonable to assume that the cognitive burden imposed by coverage verification in sequential video frames is less than for the same task in a sequence of random images. As such, we assume that the time required for a human expert to discard an erroneous image is significantly less than the time required for coverage verification on a random image. Under this assumption and using the coverage verification estimates by Su et al. [18], we use a conservative estimate of five seconds for discarding an erroneous image.

We thus compare dataset curation time imposed by each method of handling errors in inferenced datasets—ignoring, discarding, and replacing—to the time required to label the full training dataset using the standard method. The results of this comparison are shown in Table I. Here, the baseline rate, $1\times$, quantifies the time required to label the entire dataset under the standard method; values greater than $1\times$ thus indicate



(a) *SSD with MobileNetV2* (b) *SSD with Inception-v2*

Fig. 9. *Lightweight inference results.* Two different models, SSD with MobileNetV2 (a) and SSD with Inception-v2 (b), are evaluated against the three error-handling strategies, with the standard method shown for reference. Five-fold cross validation is used for each split level and each strategy. Both lightweight models show improvement across all three strategies at the smaller split ratios.

(a) Human Expert Time Only

	0.625%	1.25%	2.5%	5%	10%	20%
Ignore	161.8×	81.0×	39.7×	20.2×	10.0×	5.0×
Discard	36.7×	31.4×	23.4×	15.2×	8.7×	4.7×
Replace	1.3×	1.3×	1.4×	1.4×	1.3×	1.3×

(b) Faster R-CNN with NASNet and SSD with MobileNetV2

	0.625%	1.25%	2.5%	5%	10%	20%
Ignore	29.2×	24.8×	18.9×	13.0×	7.9×	4.4×
Discard	18.1×	16.7×	14.2×	10.7×	7.0×	4.2×
Replace	1.2×	1.2×	1.3×	1.3×	1.3×	1.3×

(c) Faster R-CNN with NASNet and SSD with Inception-v2

	0.625%	1.25%	2.5%	5%	10%	20%
Ignore	29.7×	25.1×	19.0×	13.0×	7.9×	4.5×
Discard	18.3×	16.9×	14.3×	10.8×	7.0×	4.2×
Replace	1.2×	1.2×	1.3×	1.3×	1.3×	1.3×

TABLE I
Dataset Curation Speedup Factors

better performance for our proposed method. As can be seen in Table I(a), the dataset curation time at every split ratio and for every method is significantly decreased (indicated by speedup factors greater than $1\times$). The *Ignore* method comparison is based simply on the amount of time needed to label the split ratio subset and thus exhibits the maximum possible speedup. However, the *Discard* and *Replace* methods include further manual processing by human experts downstream of this initial labeling; even so, these methods prove to require significantly less time than manually labeling the entire dataset.

The comparisons in Table I(a) ignore the machine time necessary for model training and creating inferenced datasets. Human-expert time typically costs significantly more than computer time, so it is, perhaps, the most interesting comparison to make. Nevertheless, the results shown in Table I(b) and I(c) include the training times observed in our experiments, demonstrating that our methods provide significant improvement across all split ratios, even when machine time is include and weighted equally.

In particular, training imposes about 42 hours per Faster R-CNN model, just less than 10 hours per SSD with MobileNetV2 model, and nearly 9 hours per SSD with Inception-v2 model; inference time averages about 0.64 seconds per image. When this added machine time is given equal weight with our human-expert time to simulate the wall-clock time necessary to execute the full workflow, our method still provides significant improvement across all split ratios.

In both comparisons and for all three error-handling methods, we see the largest speedups at the smaller split ratios. This result correlates with our lightweight training results, which exhibit the greatest improvements in these same split ratios. Thus, for the Okutama-Action dataset, we see that at small split ratios our proposed method is effective at both increasing the performance of lightweight models and decreasing the cost, in human-expert time and wall-clock time, of dataset curation.

VI. Conclusions and Future Work

We propose a semi-supervised dataset curation method to reduce the burdens imposed by hand-labeling large video datasets for object detection. Our approach utilizes a small number of human-expert labeled frames to train a robust object detection network that then automatically labels the remaining unlabeled images, creating an inferenced dataset. The combined ground-truth subset and the inferenced dataset are then used to train a lightweight model. This dataset is imperfect, so we explore three methods of handling erroneously labeled images. We demonstrate that inferenced datasets are effective in training lightweight models across all three error-handling strategies. Further, we show that dataset curation time with our proposed method is significantly faster than with the standard

method, and thus minimizes the costs associated with hand-labeled data.

Future work includes experimentation with additional robust and lightweight networks and with additional video datasets for object detection, as well as eventual integration and deployment in our deep learning workflows for UAV control via gesture recognition.

ACKNOWLEDGMENTS

This work was supported in part by research grants from the U.S. Marine Corps, the U.S. Navy, and the U.S. Air Force.

REFERENCES

[1] 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.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[16] I. Misra, A. Shrivastava, and M. Hebert, "Watch and learn: Semi-supervised learning for object detectors from video," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[7] 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.

[8] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7310–7311.

[9] J. Shen, N. Vesdapunt, V. N. Boddeti, and K. M. Kitani, "In teacher we trust: Learning compressed models for pedestrian detection," *arXiv preprint arXiv:1612.00478*, 2016.

[10] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Advances in Neural Information Processing Systems*, 2017, pp. 742–751.

[11] Q. Li, S. Jin, and J. Yan, "Mimicking very efficient network for object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[12] W. Hong and J. Yu, "GAN-knowledge distillation for one-stage object detection," *arXiv preprint arXiv:1906.08467*, 2019.

[13] Y. Tang, J. Wang, B. Gao, E. Dellandréa, R. Gaizauskas, and L. Chen, "Large scale semi-supervised object detection using visual and semantic knowledge transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2119–2128.

[14] W. Wang, Y. Wang, F. Chen, and A. Sowmya, "A weakly supervised approach for object detection based on soft-label boosting," in *2013 IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE, 2013, pp. 331–338.

[15] C. Vondrick, D. Patterson, and D. Ramanan, "Efficiently scaling up crowdsourced video annotation," *International Journal of Computer Vision*, vol. 101, no. 1, pp. 184–204, 2013.

[17] 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.

[18] H. Su, J. Deng, and L. Fei-Fei, "Crowdsourcing annotations for visual object detection," in *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.