Object detection for Verification Based Annotation

Oliver Batchelor* and Richard Green†
Department of Computer Science, University of Canterbury
Christchurch, New Zealand
Email: *oliver.batchelor@canterbury.ac.nz, †richard.green@canterbury.ac.nz

***Abstract*—In this paper, we examine the properties of an object detector for Verification Based Annotation (VBA), where annotation is performed by having machine annotations checked and verified (and corrected, in this work) by a human annotator. We show that with a few modifications from standard practice, a convolutional neural network based object detector can be a robust aid to annotation, without a large degree of parameter tweaking. We previously annotated a variety of small-scale datasets which we attempt to validate some of our ideas and assumptions for an object detector used for VBA.**

Our approach is to use high-resolution images, training on image crops (as opposed to the usual practice of resizing input images to a fixed resolution), and find this method successful, being more accurate and robust than down-scaling. A particular interest is the effect of localisation noise and systematic bias in annotations. We characterise the impact on object detection performance and compare to human levels, we find noise has a large impact, especially with fewer training examples.

***Index Terms*—verification, annotation, object detection, neural network, human-in-the-loop**

I. Introduction

Verification Based Annotation (VBA) is a form of *Human-in-the-loop* machine learning, a collaboration between machine and human. Human-in-the-loop approaches can offer improved engagement in activities that would otherwise be a laborious task. In Verification Based Annotation (VBA) algorithms machine annotations are verified by a human annotator, examples include question answering [1], semi-automatic segmentation [2], or as in this work modifying machine predictions [3]–[5].

In this paper, we examine the properties of an object detector used as part of a VBA process. We are most interested in the robustness to varying parameters, such as image size, number and size of objects, and to initialisation. The ability to learn incrementally is key (and in particular, learn from a minimal number of examples).

A. Verification Based Annotation

VBA is a form of *Human-in-the-loop* machine learning, a collaboration between a machine learning algorithm and a human user. The goal, where it relates to image annotation, is to make the most effective use of annotator time and reduce cognitive load.

Verification also plays a large part in ensuring consistency between human annotators in crowdsourcing efforts [6]. The annotations of any one user cannot be fully trusted, and there can be significant variation between annotators. Often large organisations gamify the annotation process by having users annotate and verify labels as part of a proof-of-human process [7].

Weaker algorithms (machine learning or otherwise) can be used to generate proposals which can then be validated by an annotator. An example of this is in [5] where computer vision algorithms generate proposed counts of a penguin colony, and a human operator marks false negatives and false positives.

Human verification is fast; in [1], a yes/no verification is reported as taking 1.6 seconds on average. For a full annotation of a ImageNet Large Scale Visual Recognition Challenge (ILSVRC) image, in [6] the time to draw a bounding box is reported at 26 seconds (42 seconds after quality control), but [8] reports only 7 seconds per box using a more effective input method involving clicking extremities of objects rather than selecting corners.

II. Setup and Method

A. Object detection

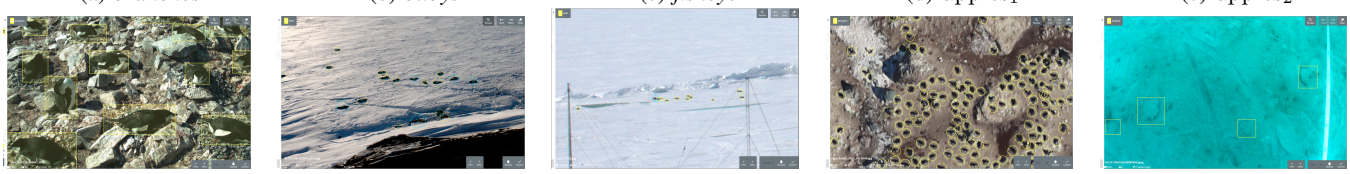
The object detector is based on a single-shot Convolutional Neural Network (CNN) detector called RetinaNet [9], an anchor-box based single stage object detector. This detector was selected for its simplicity and efficiency while having close to state-of-the-art accuracy. Some modifications from the method in [9] and brief motivations for each are:

- **Sharing**: no shared weights between classification sub-networks at different pyramid levels, to make training much faster at the beginning.
- **Non normalised loss**: We do not normalise of the loss function (by the number of anchor matches), to accommodate images with no positive annotations.
- **High resolution**: We train on crops of high-resolution images, but for inference operate on complete images. The motivation is to make the object detection task as easy as possible; larger objects are easier to detect [10].
- **Cyclical learning rates**: We use of cyclical learning rates in training, to accommodate incremental additions to the training set.
- **Extra decoder layers**: We used an extra residual block at each resolution to combine shortcut connections after concatenation with features from lower resolutions.

We test some of the hypotheses which led to these decisions. Additionally, we present a study on how localisation noise and systematic error impacts training. The tolerance to localisation

978-1-7281-4187-9/19/$31.0©2019 IEEE

(a) *branches* (b) *buoys* (c) *fisheye* (d) apples$_1$ (e) apples$_2$



(f) *penguins* (g) *seals* (h) *scott base* (i) *penguin survey* (j) *scallops*

Fig. 1: Representative images of datasets (and annotations) annotated in this work

TABLE I: Overview of datasets, showing the variety in image and object size and number.

dataset	annotations	images	box length	image size	train crop	AP_{COCO}	automated
penguins	7473	306	255 ± 118	2048×1536	800	75.9	82.6%
branches	2249	451	41.5 ± 13.9	400×400	320	62.6	76.8%
seals	4351	240	68.7 ± 20.8	3920×1600	1024	80.7	93.4%
$seals_b$	1256	82	63.4 ± 17	3920×1600	1024	72.9	87.3%
scott base	7759	301	15 ± 3.21	3927×500 – 5200×700	400	81.4	84.8%
$apples^1$	21637	300	78.4 ± 14.9	2592×1728	1024	51.8	75.1%
$apples^2$	13418	168	92.8 ± 11.9	3008×2008	1024	74.5	76.1%
$scallops_e$	3669	6741	114 ± 40.2	1280×1024	800	65.0	62.3%
fisheye	2598	367	96.6 ± 32.7	2048×1944	1024	78.9	91.8%
$buoys_d$	7221	207	38.9 ± 42.8	1920×1080	600	70.9	89.9%
penguin survey	13210	352	22.6 ± 2.11	406×405 – 672×448	400	61.6	89.5%

Subscripts denote different annotators. Datasets specified without subscript are annotated by the authors. AP_{COCO} is measured on the split validation set. Box length is the length of the longest side (or diameter for circle annotations). Automated is the total proportion of annotations created from object detections without edit.

error (as well as the ability of the object detector to accurately localise objects) is crucial to how efficient a VBA system can be.

B. Metrics

We use Average Precision (AP) at different levels of Intersection over Union (IoU) threshold for evaluation. This is the AP across all examples and not a mean Average Precision (mAP) (largely academic because most datasets here are single class). This is notated AP_t for IoU threshold t. In particular two thresholds we use are AP_{50} for a relaxed matching threshold and AP_{75} for a strict threshold and then AP_{COCO} for an average over a range of thresholds. AP_{COCO} is defined as the mean of AP_t for $t \in [0.5, 0.55..0.95]$.

C. Training parameters

In all experiments in this work I use a Stochastic Gradient Descent (SGD) optimiser with the base learning rate set to 0.001, momentum 0.9 and weight decay 0.0001 in all cases. Total loss is averaged across each mini-batch, and batch sizes of 8 are used. The balance factor between localisation and classification loss is $\lambda = 2.5$ in all cases. A cyclical learning rate is used during over each batch, with a log interpolated (geometric) multiplier reducing the learning rate by a factor of 0.1 at the end of the batch.

Image augmentation is used, crops taken at varied scale ($3/4$–$4/3$) and aspect ratio ($\pm 10\%$), flips (0.5 probability). A range of photo-metric augmentations: brightness ($\pm 10\%$), contrast ($\pm 10\%$), gamma adjustment ($\pm 0.1\%$), hue (± 5) and saturation (± 5) adjustment.

D. Datasets

A selection of images of the datasets and typical annotations used in this paper can be seen in Figure 1 and a summary of the different variations can be seen in Table I, most of the datasets were annotated by the author, a few annotated by third parties and one was annotated by both. The datasets span a wide variety of image resolutions, and object sizes; the images in some cases are reduced resolution from the captured images, in other cases reducing the resolution would make object detection impossible.

III. High resolution inference

We looked at two different possibilities for performing inference on a full-resolution image using an object detector network trained only on crops of images: (a) pass in the full image using the property that the network is *fully convolutional* and (b) tile images the same size as training crops and collapse the predictions using a combined Non-Maxima Suppression (NMS). To facilitate this idea, the box annotations on the edge

of images are estimates of the full bounds of the object, and the anchor boxes are not cropped to the edge of the image.

The first method is to pass in the full image to the neural network, even though it has only been trained on much smaller crops. Object detection networks are flexible and work across a range of input image resolutions. All layers are either convolutions or do not reshape the feature maps (aside from up/down-sampling). Passing in a larger image results in a larger feature map and set of box outputs at each layer of the pyramid.

The second inference method is to tile multiple inferences at the size the model was trained at, across the full image size a overlap buffer region. The result is sets of overlapping predictions at the edges which be decimated using NMS.

Image size has its limits for both training and inference, as the complexity of the model and the size of the memory on the Graphics Processing Unit (GPU) determines the maximum size image which can be processed. We can process the large images used in this work because of the relatively simple backbone model used (ResNet-18 [11]). Evaluation using tiling makes it feasible to use larger image sizes and more complicated models, even if multiple inferences take more time.

A. Effect of image resolution and crop size

For this experiment we vary the scale and crop size to test the idea that a high-resolution object detector can be trained on small parts of images, then the whole image used in inference.

TABLE II: Effect of scale and crop size on validation accuracy (percent of best AP_{COCO}). Average across datasets ($apples$[1], *penguins*, *scallops*, *seals*)

	scale	12.5%	25%	50%	100%
crop	512	0.0	2.4	59.3	82.8
	768	17.4	68.2	90.0	96.4
	1024	28.5	81.9	95.0	100.0

Using a larger crop size proved more accurate in all of the four datasets, and using a larger crop was beneficial in all cases, being more accurate and more stable to train. The larger size, however, trains much more slowly. In this case a factor of 3.3 slower for half scale or half-size crop, larger factors having less speedup due to constants such as loading the images. Despite the relatively low requirement for fast inference (to keep up with a single annotator); this could be used for example for image selection, to find images of most uncertainty for active learning.

B. Inference method for large images

We compared the two methods (tiling vs inference on the full image) by using both methods for testing against the validation set of several different training runs. It can be seen in figure 3 that both perform very similarly. Sometimes one is marginally better, and sometimes the other is marginally better, even within the same training run.

IV. Incremental training

When using the annotation tool, image annotations become available incrementally. Here we investigate the difference between training with all examples annotated upfront compared to training with images added to the training set incrementally.

During annotation, the validation set is incrementally built. Here we test against the final validation sets. In future, it would be better to use cross-validation, to make better use of training data, and provide more accurate testing (at the expense of extra training time).

Figure 4 shows the training plot for each dataset, where the validation accuracy (AP_{COCO}) is plotted against training time for both *incremental* and *full* cases. Different datasets improve at different rates with more data. However, all are restricted by the dataset size and improve with more data.

V. Effect of localisation noise

In this experiment, we examine how tolerant the object detector is to two factors: (a) random noisy annotations, (b) systematic bias and combinations of the two. All human annotation contains a certain amount of noise as well as a bias which varies from person to person and activity to activity, so it is expected an object detector can tolerate a certain amount of labelling noise in its training data.

A VBA based system by nature of using an object detector trained on noisy inputs, will therefore not produce entirely accurate localised predictions. The hope is that such a system will eventually produce localisations that are *good enough* so that predictions from the object detector can be accepted without taking up valuable annotator time. In this experiment, we aim to quantify how noise degrades performance, and therefore establish some idea of what guideline for the level of precision is required during annotation.

A. Human threshold

Figure 5 shows in more detail the distribution of IoU overlap for detections which have been modified. The density of transformed detections peaks at around 0.80–0.85 depending on the dataset. It can be reasonably assumed that the real accuracy of the detections, which are used unmodified, would lie between that peak and 1.0. Human annotation also has some degree of variability (reference [8] gave a mean IoU overlap of 88 for a box input method with Pascal VOC ground truth).

One minus the peak density seems like a reasonable surrogate measure for the human verification threshold. From the datasets annotated in this thesis the *penguins* dataset annotation has the lowest threshold, at approximately 0.15, and as a result, it seems likely the most precise localisation. For the *penguin survey* the threshold is highest at around 0.35. This makes sense because counting was emphasised as opposed to precise annotation, so, the localisation threshold is higher.

B. Method

We added noise and systematic bias to five different datasets, covering a range of parameters (resolutions, object

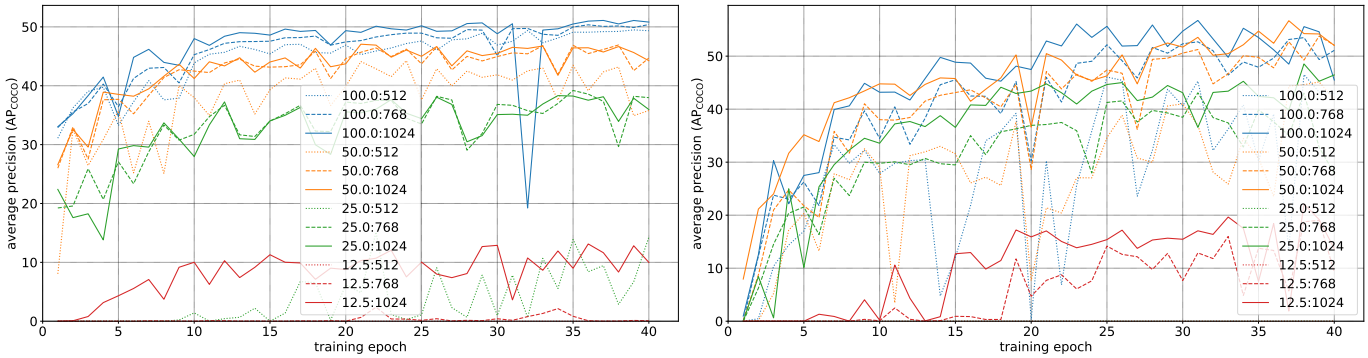


Fig. 2: Comparison training at different crop sizes and scales for (a) *apples*[1] and (b) *scallops* dataset. Crop size is presented as the crop size in the original image, the crop used for training is scaled by the given factor.

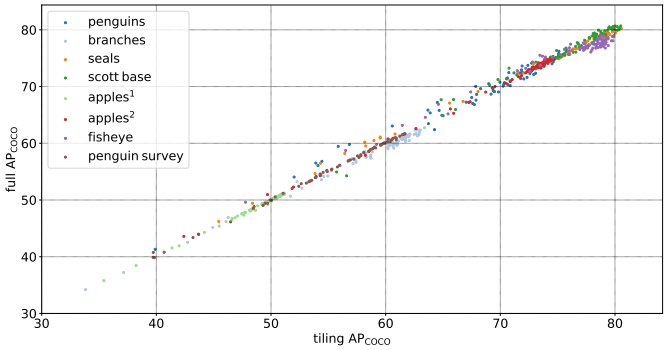


Fig. 3: Comparison of different inference methods across one training run inference using tiling vs. inference on full images. Training occurs on crops and evaluating on full images.

sizes, instance counts), where validation accuracy was highest among the datasets created in this work. These five are: *branches*, *apples*[2] (50% scale), *scott base*, *seals* (50% scale) and *penguins* (50% scale). Three datasets are down-sampled to save time. Two datasets are not down-sampled: one of them because the resolution is already low, and the other because the objects are already very small for the object detector.

Noise is added by randomly moving the box centre and randomly scaling the box size. The box centre (c_x, c_y) is moved as a proportion of the size of the box to become ($\hat{c}_x$, $\hat{c}_y$), and the width and height (w and h) of the box are multiplied by translation (t_x, t_y) and scaling (s_x, s_y) factors sampled from a normal distribution. The standard deviation of the distribution σ controls the magnitude of the translation and scaling.

Systematic bias is added as a proportion of box width and height also, controlled by a parameter factor Δ; in this experiment, the box is always moved up and to the right. The final transformed box is then specified by $\hat{c}_x$, $\hat{c}_y$, $\hat{w}$, $\hat{h}$ according to equation 1.

$$
\begin{aligned}
s_w, s_h &\sim \mathcal{N}(1,\, \sigma^2) \\
t_x, t_y &\sim \mathcal{N}(0,\, \sigma^2) \\
\hat{c}_x &= c_x + (\Delta + t_x)w \\
\hat{c}_y &= c_y + (\Delta + t_y)h \\
\hat{w} &= s_x w \\
\hat{h} &= s_h h
\end{aligned}
\tag{1}
$$

The noise is consistent throughout training as opposed to being added as a form of data label augmentation. Systematic offset factor Δ is also added to the validation set, assuming such a form of error would be uniform across the data. With an offset, the challenge for the object detector is if the detector can adapt its estimations when the real object in question is translated with respect to the centres in the feature map of the receptive field. Noise, on the other hand, is added only to the training data and not to the validation data as adding noise changes the mean of box centre or size (on average).

The object detector is trained for 40 epochs in five different configurations of noise and systematic offset: $\sigma \in [0\%, 4\%, 8\%, 16\%, 32\%]$ and $\Delta \in [0\%, 4\%, 8\%, 16\%, 32\%]$. Object detectors are trained with standard parameters for each dataset (except the three which are trained at 50% scale), and the impact of the noise and systematic offset is measured by looking at the peak validation AP at different thresholds.

After noticing overfitting occurring in noisy cases, the set of experiments were repeated with reduced training data to examine how noise affected generalisation. We used two low training data scenarios, firstly at 25% of training examples, then at 6.25%.

The amount of noise and the resulting mean IoUs with the original box can be seen visually in Figure 6. It can be seen that the levels of noise and systematic offset, both have approximately the same mean IoUs with respect to the original boxes for the particular level of offset or bias.

The impact in terms of degradation of object detection performance can be seen in Table III, shown as the reduction in performance from the baseline zero noise and zero offset

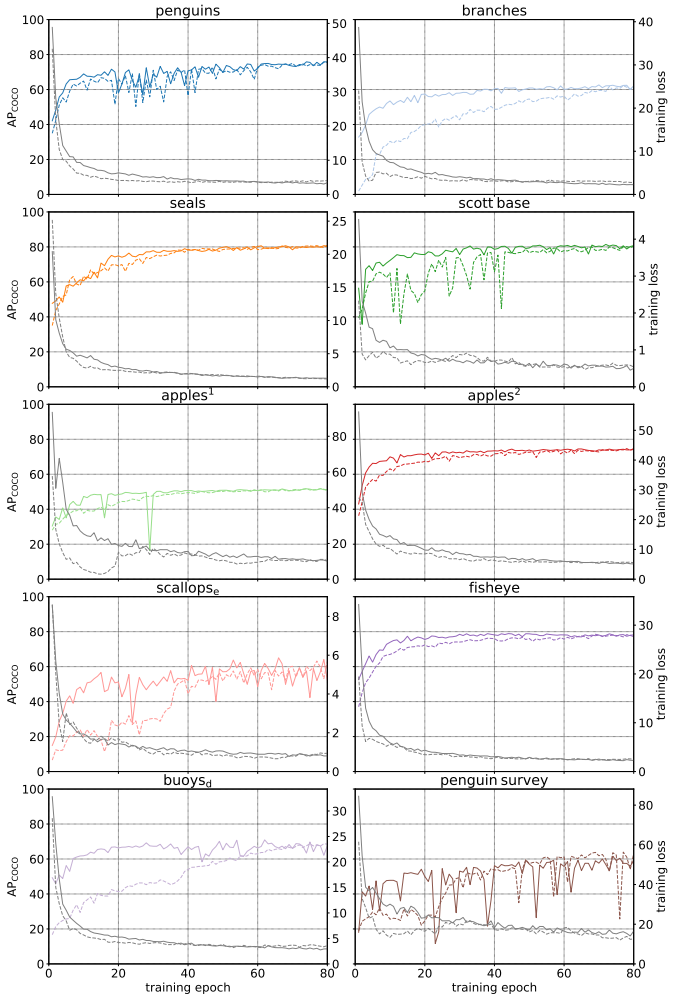


Fig. 4: Incrementally adding examples vs. training with all examples from the beginning. Dotted lines are incremental training, solid lines are training with all examples up front. Grey lines at the bottom of each chart show the training loss.

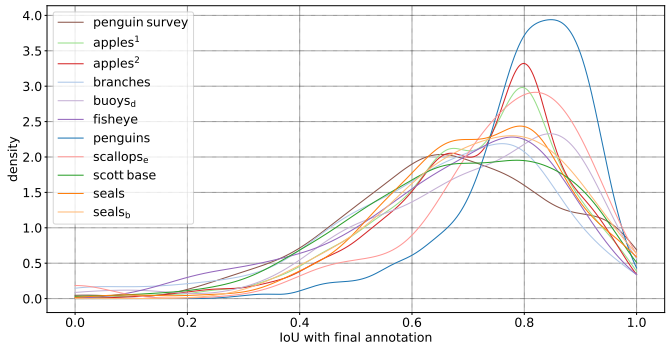
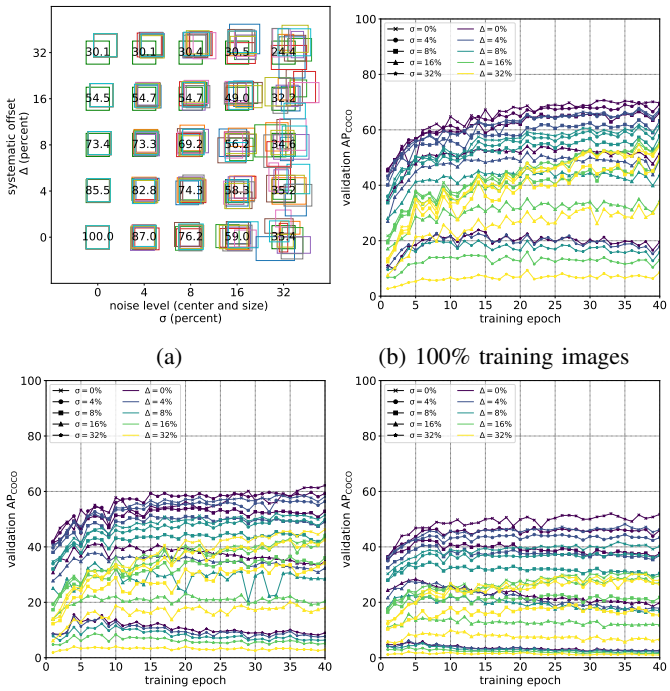


Fig. 5: Density plot of IoU overlap for detection with respect to annotation, for transformed detections.

cases (shown as the absolute value, in bold). Two matching thresholds are shown, a relaxed matching threshold AP_{50} and at strict matching threshold AP_{75}.



(a) (b) 100% training images

(c) 25% training images (d) 6.25% training images

Fig. 6: (a) Samples of the different noise and systematic bias added, showing reference box with 10 samples; overlaid with the mean IoU for that condition. (b), (c), (d) AP_{COCO} validation after training with with varying noise and systematic bias added, and varying training set size. Average of training runs of 5 different datasets, at half-resolution: *seals*, *apples*^{2} and *penguins*; at full-resolution: *scott base*, *branches*

The object detector at a relaxed matching threshold AP_{50} is robust to a small amount of noise and systematic offset; less than 8% offset or noise shows only a small impact on AP_{50} degradation in each case, even in the low data scenarios. For the full amount of data, AP_{50} is only degraded 10% at $\sigma = 16\%, \Delta = 16\%$, yet in the same case AP_{75} is reduced by 71%.

For the full amount of training data, AP_{75} is also robust to a small amount of noise or offset, roughly 2% degradation of performance for 4% noise or offset. Human variation of box annotation in [8] was reported as a mean IoU of 88, which corresponds to the noise case of $\sigma = 4\%$. At that noise level, performance degradation is minimal, even for a strict matching criterion, degrading AP_{75} by around 2%. Keep in mind that these datasets were human-annotated to begin with, so the noise added is extra noise.

Sensitivity to noise and systematic offset is seen to increase significantly with less training data. At $\sigma = 4\%$, a level of noise similar to a human annotator, AP_{75} is degraded by 2.1% at 100% images, increasing to 6.1% at 25% images and 10.1% at 6.25% of the training images. This increased sensitivity can be seen across the range of noise and offset, and at 6.25% of the images, AP_{50} is degraded approximately double the

TABLE III: Best validation AP_{50}, AP_{75}, with different levels of added noise (σ) and systematic bounding box offset (Δ) and at different sizes of training data. Baseline ($\sigma = 0$, $\Delta = 0$) in each case shown as absolute value in bold, other cases shown as percent change. Mean and standard deviation of 5 different datasets, at half-resolution: *seals*, $apples^2$ and *penguins*; at full-resolution: *scott base*, *branches*.

(a) 100% training data

		$\Delta = 0\%$	$\Delta = 4\%$	$\Delta = 8\%$	$\Delta = 16\%$	$\Delta = 32\%$
AP_{50}	$\sigma = 0\%$	$\mathbf{95.1 \pm 2.7}$	$-0.6 \pm 0.6\%$	$-0.7 \pm 0.7\%$	$-2.8 \pm 2.1\%$	$-10.1 \pm 10.6\%$
	$\sigma = 4\%$	$-0.3 \pm 0.6\%$	$-0.4 \pm 0.6\%$	$-1.3 \pm 1.4\%$	$-2.8 \pm 1.7\%$	$-10.4 \pm 10.3\%$
	$\sigma = 8\%$	$-1.1 \pm 0.8\%$	$-1.3 \pm 1.7\%$	$-2.4 \pm 1.5\%$	$-4.0 \pm 2.3\%$	$-12.6 \pm 9.8\%$
	$\sigma = 16\%$	$-6.9 \pm 4.7\%$	$-8.4 \pm 5.3\%$	$-8.0 \pm 4.4\%$	$-10.2 \pm 3.7\%$	$-23.8 \pm 11.7\%$
	$\sigma = 32\%$	$-32.6 \pm 6.8\%$	$-32.1 \pm 7.1\%$	$-36.1 \pm 4.4\%$	$-42.2 \pm 4.8\%$	$-61.5 \pm 6.6\%$
AP_{75}	$\sigma = 0\%$	$\mathbf{84.0 \pm 8.5}$	$-2.7 \pm 1.5\%$	$-9.1 \pm 3.4\%$	$-27.3 \pm 21.5\%$	$-24.1 \pm 17.0\%$
	$\sigma = 4\%$	$-2.1 \pm 1.1\%$	$-2.9 \pm 1.1\%$	$-10.5 \pm 4.0\%$	$-26.8 \pm 21.0\%$	$-24.3 \pm 15.8\%$
	$\sigma = 8\%$	$-7.5 \pm 7.7\%$	$-8.6 \pm 5.2\%$	$-17.0 \pm 7.7\%$	$-42.5 \pm 14.9\%$	$-32.7 \pm 12.7\%$
	$\sigma = 16\%$	$-23.8 \pm 16.4\%$	$-30.1 \pm 16.7\%$	$-42.4 \pm 14.3\%$	$-71.0 \pm 8.9\%$	$-63.4 \pm 11.7\%$
	$\sigma = 32\%$	$-81.3 \pm 7.2\%$	$-83.4 \pm 6.0\%$	$-86.6 \pm 3.2\%$	$-94.9 \pm 2.5\%$	$-97.0 \pm 1.3\%$

(b) 25% training data

		$\Delta = 0\%$	$\Delta = 4\%$	$\Delta = 8\%$	$\Delta = 16\%$	$\Delta = 32\%$
AP_{50}	$\sigma = 0\%$	$\mathbf{86.9 \pm 7.7}$	$-0.4 \pm 1.4\%$	$-2.0 \pm 1.7\%$	$-7.2 \pm 6.2\%$	$-18.3 \pm 16.5\%$
	$\sigma = 4\%$	$0.4 \pm 2.6\%$	$0.5 \pm 1.5\%$	$-1.6 \pm 2.5\%$	$-8.1 \pm 7.4\%$	$-19.8 \pm 14.8\%$
	$\sigma = 8\%$	$-1.2 \pm 1.3\%$	$-1.9 \pm 1.7\%$	$-3.1 \pm 2.7\%$	$-9.1 \pm 7.6\%$	$-23.1 \pm 16.7\%$
	$\sigma = 16\%$	$-9.0 \pm 7.7\%$	$-9.2 \pm 6.6\%$	$-11.9 \pm 9.1\%$	$-20.4 \pm 10.9\%$	$-38.3 \pm 12.6\%$
	$\sigma = 32\%$	$-41.3 \pm 8.6\%$	$-43.5 \pm 7.8\%$	$-46.3 \pm 8.1\%$	$-55.8 \pm 7.1\%$	$-75.6 \pm 3.8\%$
AP_{75}	$\sigma = 0\%$	$\mathbf{72.8 \pm 16.4}$	$-6.3 \pm 6.4\%$	$-18.6 \pm 9.0\%$	$-39.5 \pm 17.5\%$	$-29.2 \pm 19.2\%$
	$\sigma = 4\%$	$-6.1 \pm 5.0\%$	$-8.7 \pm 7.4\%$	$-24.7 \pm 11.6\%$	$-38.2 \pm 20.8\%$	$-34.8 \pm 17.6\%$
	$\sigma = 8\%$	$-19.1 \pm 17.8\%$	$-21.7 \pm 14.8\%$	$-33.2 \pm 14.7\%$	$-56.3 \pm 14.8\%$	$-51.1 \pm 18.9\%$
	$\sigma = 16\%$	$-39.5 \pm 24.5\%$	$-48.3 \pm 24.8\%$	$-59.3 \pm 19.5\%$	$-82.8 \pm 12.6\%$	$-76.1 \pm 15.3\%$
	$\sigma = 32\%$	$-86.9 \pm 7.9\%$	$-87.2 \pm 9.1\%$	$-92.3 \pm 5.5\%$	$-97.6 \pm 1.4\%$	$-98.7 \pm 0.5\%$

(c) 6.25% training data

		$\Delta = 0\%$	$\Delta = 4\%$	$\Delta = 8\%$	$\Delta = 16\%$	$\Delta = 32\%$
AP_{50}	$\sigma = 0\%$	$\mathbf{75.1 \pm 23.2}$	$-3.9 \pm 7.7\%$	$-4.2 \pm 6.5\%$	$-15.4 \pm 11.8\%$	$-31.7 \pm 19.6\%$
	$\sigma = 4\%$	$-6.7 \pm 9.9\%$	$-2.1 \pm 3.2\%$	$-4.0 \pm 2.0\%$	$-15.7 \pm 14.1\%$	$-31.6 \pm 19.8\%$
	$\sigma = 8\%$	$-2.2 \pm 2.7\%$	$-4.6 \pm 6.1\%$	$-8.8 \pm 9.1\%$	$-18.9 \pm 10.1\%$	$-42.8 \pm 17.6\%$
	$\sigma = 16\%$	$-18.7 \pm 9.8\%$	$-21.0 \pm 15.5\%$	$-21.9 \pm 8.3\%$	$-30.2 \pm 11.6\%$	$-58.8 \pm 12.7\%$
	$\sigma = 32\%$	$-68.6 \pm 12.2\%$	$-70.8 \pm 10.6\%$	$-71.6 \pm 8.4\%$	$-76.6 \pm 9.0\%$	$-87.1 \pm 5.3\%$
AP_{75}	$\sigma = 0\%$	$\mathbf{61.5 \pm 26.7}$	$-8.8 \pm 6.4\%$	$-29.3 \pm 9.8\%$	$-55.1 \pm 13.6\%$	$-46.9 \pm 14.1\%$
	$\sigma = 4\%$	$-10.1 \pm 5.0\%$	$-16.7 \pm 8.9\%$	$-40.6 \pm 14.1\%$	$-58.7 \pm 19.1\%$	$-51.8 \pm 14.4\%$
	$\sigma = 8\%$	$-25.7 \pm 15.2\%$	$-32.9 \pm 18.1\%$	$-51.4 \pm 12.9\%$	$-76.8 \pm 10.9\%$	$-76.5 \pm 6.7\%$
	$\sigma = 16\%$	$-63.7 \pm 20.6\%$	$-67.9 \pm 23.2\%$	$-77.0 \pm 15.2\%$	$-92.4 \pm 7.7\%$	$-93.4 \pm 5.1\%$
	$\sigma = 32\%$	$-97.4 \pm 1.8\%$	$-97.2 \pm 1.8\%$	$-98.0 \pm 1.2\%$	$-98.7 \pm 0.7\%$	$-99.6 \pm 0.2\%$

amount compared to at 25% of the images.

VI. Conclusions

We presented our approach to object detection for Verification Based Annotation, where (with a few minor changes) an object detector [9] can be robust, without a large degree of parameter tweaking, for object detection in a wide variety of domains with a variety of image resolution and object size.

By training with image crops, higher accuracy is possible using the full image resolution (at the expense of training time, however during annotation, time is plentiful). In each case, using larger crop sizes proved better, more accurate and more stable. Two high-resolution inference methods were approximately equal, with the tiling approach allowing larger images at the expense of inference time. Incremental training was shown to be practical, with different datasets scaling in accuracy (and in time required for training) differently with the addition of extra images.

An investigation of the impact of annotation localisation noise and systematic error on box annotation shows that training with many images is robust to a small amount of localisation error. However, with fewer images the sensitivity is much higher, leading to degradation of performance with even a small amount of noise; this suggests a focus on accurate object localisation is important, especially near the beginning of annotation, for both annotator and tool author.

Acknowledgment

Thanks to Dr Regina Eisert for use of seal images (seals, scott base), Antarctica New Zealand for use of the aerial penguin survey images (aerial survey) and Lincoln Agritech for apple dataset (apples_2).

References

[1] Dim P. Papadopoulos, Jasper R. R. Uijlings, Frank Keller, and Vittorio Ferrari. We don't need no bounding-boxes: Training object class detectors using only human verification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, number 1, pages 854–863. IEEE, 6 2016.

[2] Llus Castrejón, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating object instances with a polygon-RNN. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-Janua, pages 4485–4493, 4 2017.

[3] Angela Yao, Juergen Gall, Christian Leistner, and Luc Van Gool. Interactive object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3242–3249, 2012.

[4] Olga Russakovsky, Li-Jia Li, and Li Fei-Fei. Best of both worlds: Human-machine collaboration for object annotation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 07-12-June, pages 2121–2131. IEEE, 6 2015.

[5] Stephen McNeill, Kerry Barton, Phil Lyver, and David Pairman. Semi-automated penguin counting from digital aerial photographs. In *2011 IEEE International Geoscience and Remote Sensing Symposium*, pages 4312–4315. IEEE, 7 2011.

[6] Hao Su, Jia Deng, and Li Fei-fei. Crowdsourcing Annotations for Visual Object Detection. *Proc. AAAI Human Computation'12*, pages 40–46, 2012.

[7] Luis Von Ahn, Benjamin Maurer, Colin Mcmillen, David Abraham, and Manuel Blum. reCAPTCHA : Human-Based Recognition via Web Character. *Science*, 321(5895):1465–1468, 2015.

[8] Dim P. Papadopoulos, Jasper R.R. Uijlings, Frank Keller, and Vittorio Ferrari. Extreme Clicking for Efficient Object Annotation. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-Octob, pages 4940–4949, 2017.

[9] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal Loss for Dense Object Detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007. IEEE, 10 2017.

[10] Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8693 LNCS(PART 5):740–755, 2014.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition.