Automatic Activity Profiling in Cows to Ensure a Healthy Lifestyle

Author: Alan Clark **– In Association with:** Smartbell & Cambridge Spark **– Date:** 1st April 2020

Abstract

Object detection models were trained and applied to video footage of calves within an enclosed, undercover rearing area. It was demonstrated that by performing transfer learning using a relatively small selection of manually tagged frames, models able to detect calves, their heads and shelter entrances could be developed. The trained models detected around 90 % of objects in a frame on average, and were used to generate metrics detailing: the number of calves in view over a given period of time; the number of these that were within a shelter; and the number with their heads low to the ground – a potential indicator of illness. Further work is needed to assess the ability of the models to generalise to a diverse range of calve rearing environments.

Introduction

The objective of this project was to train and apply object detection models to video footage of calves within an enclosed, undercover rearing area. An example frame is shown in Figure 1, and key metrics are provided in Table 1.

Figure 1 - Example frame from the provided video footage

Video	Pen	Camera	Time	Duration	Quality
0	6	8	07:20	03:08	HD
1	6	7	07:20	01:47	HD
2	3	4	07:20	01:29	SD
3	3	3	07:20	01:08	SD
4	6	2	07:20	01:14	SD
5	6	8	09:35	00:53	HD
6	6	7	09:35	00:51	HD
7	3	4	09:35	00:54	SD
8	3	3	09:35	02:15	SD
9	3	2	09:35	25:06	SD

Table 1 – Summary information for the provided video footage

Through applying object detection models to the footage, answers to the following types of question were sought:

1. Where in the pen do the animals spend their time?
2. With which other animals does each spend their time?
3. Which animals tend to exhibit "head-droop"?
4. Do any of the animals exhibit unusual or odd-one-out behaviour?

To address these questions, it was hoped that a model could be developed to identify calves individually. The majority of the calves in the footage are Friesians, which have unique patterns of black and white markings that an object detection model could potentially be trained to distinguish. Labelled data would be required to train such a model (photos/footage of each calve by itself, or labels for each calve in the provided videos), however this was not readily available. The aims of this project were therefore refined to:

1. Applying pre-trained object detection models to the footage and evaluating their performance
2. Demonstrating that transfer-learning could be employed to produce superior models for identifying calves within the type of environment exhibited in the footage
3. Generating models that can also detect the calves' heads and the entrances to their "igloo" shelters
4. Generating metrics from applying the models to the footage that help start to answer the above questions

Part 1: Pre-Trained Object Detection Models

The TensorFlow Object Detection API[1] framework was used to run pre-trained models from the TensorFlow Detection Model Zoo[2] against the provided footage. Each of the models had been pre-trained on the Microsoft COCO dataset[3], which contains more than 200,000 tagged images across 80 object categories, including cow. The green shaded rows in Table 6 show the results of applying these models to a selection of frames.

For classification problems, metrics such as accuracy, precision, recall and F1-score are used to measure model performance. However, object detection is both a classification (applying the correct label to an object) and localisation (predicting the correct location of an object) problem. The primary metric used to evaluate these models is mAP – mean average precision (see the appendix for a description of how mAP is calculated). In this project only the cow class was important, and so the average precision (AP) for this class was evaluated for each model. The results are shown in the green shaded rows of Table 3.

In addition to AP, the fraction of ground truth objects detected (regardless of predicted class) with an intersection over union (IoU – see the appendix for a description) ≥ 0.5 was measured. The pre-trained models sometimes mis-classified the calves as dogs, sheep or even elephants – other classes of quadruped in the COCO dataset. As is the case for the provided footage, in any scenario where cameras will be covering an area solely populated by cows, predictions of all quadrupeds could be mapped to the cow class. Therefore, having correctly predicted classes is not necessarily the most important factor, and so mAP need not be the only metric used to compare models here. The mean IoU for the detected ground truth boxes (those counted in the aforementioned metric) was also determined. Both values are provided for each model, alongside the AP, in Table 3.

To calculate these metrics it was necessary to produce ground truth boxes for several frames in the provided footage. An example of these (generated using Microsoft

VoTT[4]) is shown in Figure 2, and summary statistics of the tagging are provided in Table 2.

The performance metrics in Table 3 were calculated using only the test dataset. The AP and fraction of ground truth objects detected by the models from the Model Zoo averaged approximately 50%, with the Faster R-CNN ResNet-50 model achieving the highest AP.

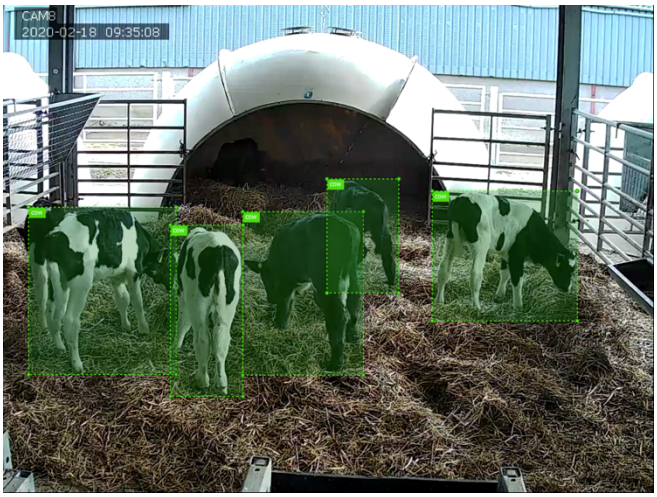


Figure 2 – Example video frame with calves manually tagged using VoTT

Video	Frames			Cows		
	Train	Test	Total	Train	Test	Total
0	15	6	21	78	32	110
1	35	8	43	76	17	93
2	16	5	21	94	30	124
3	5	0	5	33	0	33
4	7	2	9	33	10	43
5	39	14	53	192	67	259
6	45	7	52	275	41	316
7	40	15	55	256	96	352
8	0	1	1	0	6	6
9	3	1	4	21	6	27
Totals	**205**	**59**	**264**	**1058**	**305**	**1363**

Table 2 – Training and test dataset metrics, including both the number of frames and number of individual cows manually tagged

Report Section	Model	AP	Fraction of Calves Detected	Mean IoU
1	Faster R-CNN ResNet-50	0.545	0.523	0.805
1	SSD ResNet-50 FPN	0.482	0.547	0.843
1	Mask R-CNN ResNet-101 Atrous	0.458	0.528	0.862
1	SSD MobileNet v1 FPN	0.333	0.457	0.807
2	Faster R-CNN ResNet-50 with Transfer Learning	0.994	0.981	0.884
3	Faster R-CNN ResNet-50 with Transfer Learning	0.873	0.992	0.927

Table 3 – Average precision (AP), fraction of calves detected, and mean intersection over union (IoU) for the object detection models used in Part 1 (green rows), Part 2 (blue row) and Part 3 (orange row) of this report

Part 2: Transfer Learning to Improve Calf Detection

In the next stage of the project, transfer learning was employed in an attempt to improve on the AP and ground truth object detection scores of the models used in Part 1.

Training a CNN from scratch requires a large amount of data (the COCO dataset has more than 200,000 labelled images) and takes a significant amount of time; even when using GPU-enabled devices. In transfer learning we utilise the architecture of an existing model, along with most of the pre-trained weights that were learned on another dataset. Weights in only the final few layers are discarded and re-learned against a tagged dataset of custom object classes, which need only contain a few hundred labelled images.

The COCO trained Faster R-CNN ResNet-50 model, which yielded the highest AP score in Part 1, was used as the base for transfer learning. Table 2 summarises the data used for training and testing.

Initial attempts were made to train the model with a CPU-device; however, the time required would have limited the number iterations possible to complete for this project. Using Google's Colaboratory[5] platform the same training could be performed on a GPU-enabled device for free (up to 12 hours). Compared to a CPU-device, a 22-fold increase in training speed was observed, as illustrated in Figure 3 and 4.

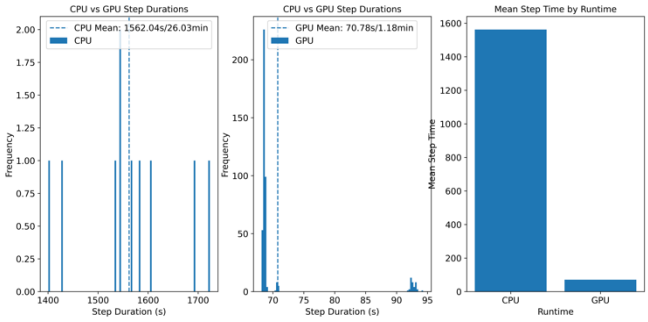


Figure 3 - Time per model training step on CPU and GPU environments

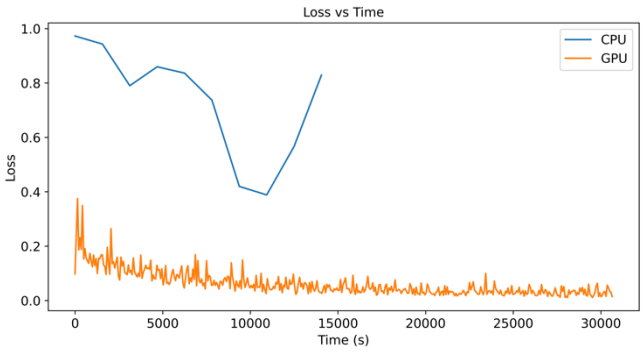


Figure 4 - Loss vs time for training on CPU and GPU environments

When applied to the same set of frames used for model evaluation in Part 1, the results achieved by the transfer-learned model are shown in the blue shaded row of Table 3, and a selection of tagged frames is shown in Figure 5 and the blue shaded row of Table 6.

The achieved AP and fraction of objects detected approaches 1, however these metrics are likely somewhat misleading. The diversity of camera angles in the dataset is limited; the environment in each case is the same (the same pens, but with some differences in lighting conditions); the number of unique calves is small; and the test dataset was pulled from the same selection of videos as the training set. The model tended to predict objects as cows with certainty near or equal to 1, however the only class of object it has been trained to be aware of is cow.

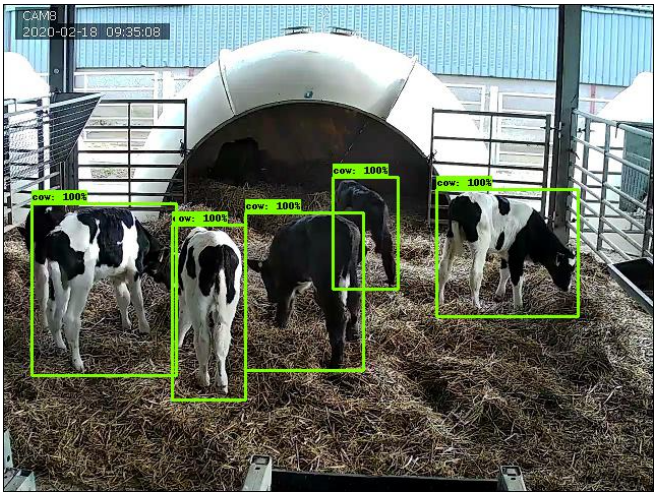


Figure 5 – Calves detected in a sample frame by the Faster R-CNN ResNet-50 model, with weights learned via transfer-learned using tagged footage

Yet, a farmer present in one frame was predicted to be a cow with 45 % confidence – far lower than any actual cow.

The degree of overfitting to the training data and the model's ability to generalise to other environments cannot be stated without a more diverse set of tagged validation data. However, this exercise has demonstrated that with a relatively small set of tagged data and fewer than 12 hours of training it is possible to generate a model that can detect almost all calves in a particular environment. If the current model does not generalise well to other calve rearing environments (say an outdoor space), it would be relatively simple to re-run transfer learning with a more diverse set of tagged frames. If the performance of the generalised model is not sufficiently high, a small selection of models, specialised for different environments, could be trained. Farmers could select the environment most closely matching their own, and guidance can be provided on the positioning of cameras for optimal model performance.

Part 3: Transfer Learning to Detect Additional Classes

In Part 2 it was demonstrated that transfer learning could be used to generate an object detection model that can detect a very high fraction of calves. Building on this success, the next aim was to investigate the ability to use transfer learning to detect more nuanced objects that could help answer two of the initially posed questions: where do the animals spend their time, and do any exhibit "head-droop"?

The training and test datasets used in Part 2 were expanded to include the calves' heads and the entrances to the "igloo" shelters at the rear of their enclosures. Given the strong performance of the model produced in Part 2, some additional calves were tagged: those in very dark conditions or behind gratings, and those largely occluded by other calves. An example tagged frame is shown in Figure 6, and summary statistics of the tagging are provided in Table 4.

Examples of the tagged images produced by the new trained model are shown in Figure 7 and the orange row of Table 6. The results of applying the model to the same test images as used in Parts 1 and 2 (those summarised in Table 4) are provided in the orange row of Table 3. The performance metrics from applying the model to the test dataset that included all three tagged classes are shown in Table 5.

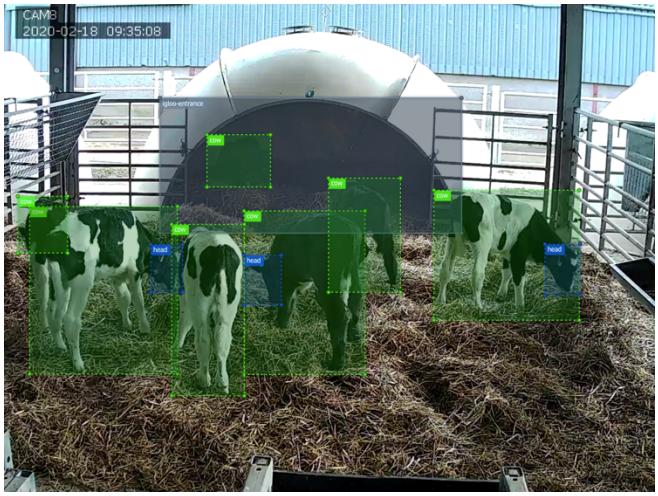


Figure 6 – Example video frame with calves, their heads, and the entrance to their "igloo" shelter manually tagged using VoTT

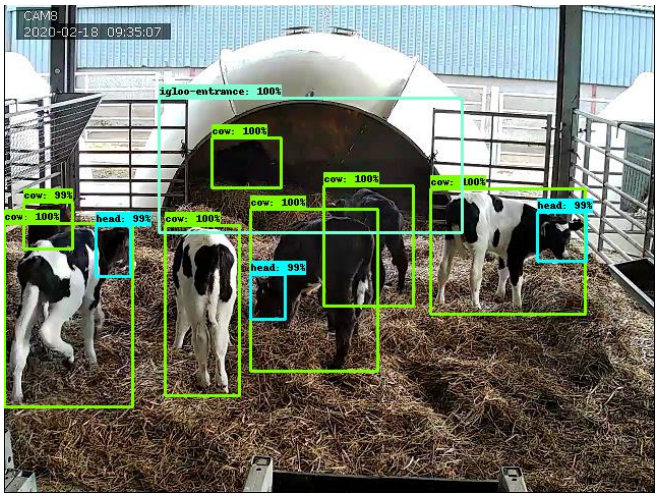


Figure 7 – Calves, their heads and "igloo" entrance detected in a sample video frame by the Faster R-CNN ResNet-50 model, with weights learned via transfer-learned using tagged footage

Video	Frames		Calves		Heads		Igloos	
	Train	Test	Train	Test	Train	Test	Train	Test
0	15	6	81	30	67	24	15	6
1	36	7	79	14	79	14	108	21
2	17	4	169	40	103	27	51	12
3	3	2	38	24	23	12	9	6
4	7	2	42	12	32	8	21	6
5	39	15	282	112	104	41	39	15
6	39	13	276	87	103	35	116	39
7	43	12	321	87	146	40	128	35
8	1	0	8	0	4	0	3	0
9	15	3	118	25	66	20	45	9
Totals	**215**	**64**	**1414**	**431**	**727**	**221**	**535**	**149**

Table 4 - Training and test dataset metrics, including the number of frames and individual calves, calf heads, and "igloo" entrances manually tagged

AP (cow)	0.857
AP (head)	0.172
AP (igloo-entrance)	0.677
mAP	0.569
Fraction of Ground Truth Objects Detected	0.899
Mean IoU for Detected Ground Truth Objects	0.867

Table 5 – Average precision (AP) and mAP scores, fraction of ground truth objects detected, and mean IoU for the model produced in Part 3

The lower performance of this model relative to the one produced in Part 2, as indicated in Table 3, was investigated and found to be largely due to the newer model detecting calves that had not been tagged in the first dataset. This could be an indicator of overfitting to the new training dataset, or a genuine improvement in the ability to detect

calves in less clear conditions. Again, a more diverse dataset would be required to determine which case is true.

Whilst the AP figures for the head and "igloo"-entrance classes shown in Table 5 appear low, there are two possible mitigating causes. Firstly, the model is detecting objects within other objects (heads on cows, and cows inside their "igloos"). Secondly, as observed with the model in Part 2, objects are being predicted with very high scores – values of 1 in many cases. Because of the way AP is calculated for each object class (see the appendix) these two factors make the metric unreliable. The fraction of ground truth objects detected and the associated mean IoU for these are therefore arguably fairer evaluation metrics, and ones which the model performed well against.

Part 4: Extracting Metrics from Object Detections

The model trained in Part 3 was applied to video 9, which has a duration of 25 minutes. A frame was sampled each second, and the following metrics recorded: the number of calves detected; the number that were inside the "igloo" shelter; the number of heads detected; and the number of calves with their head down. A 60 second rolling average of the results obtained are shown in Figure 8.

Figure 8 – Rolling 60s average of the object detection metrics produced from applying the model trained in Part 3 to video 9

To determine whether a cow was inside an "igloo" and identify which calve a detected head likely belonged to, the degree of overlap of every detected object with every other detected object was calculated. This was defined as the area of the overlapping region between the bounding boxes divided by the area of whichever bounding box was smaller. A smaller object completely enveloped by a larger one would therefore have an overlap score of 1.

A detected head was associated with the calf that had the highest overlap; although when calves were very close together this did not always yield the correct answer. Once a head had been associated with a calf, the relative vertical position of the head relative to the calf's bounding box was determined. The model was unable to determine the orientation of detected calves (whether they faced away, towards or sideways from the camera) which made determining whether their head was truly down challenging.

For the results shown in Figure 8, a calf was said to be inside the "igloo" shelter if the overlap with the entrance was greater than or equal to 0.97. A calf was said to have its head down if the relative vertical position of its head relative to its body was less than or equal to 0.7. These parameters would require further investigation to yield more accurate results.

Conclusions & Future Work

This project has shown that with relative ease (0.5-1 days to tag circa. 200 frames, followed by 0.5-1 days of training on a GPU-enabled device) an object detection model that will perform strongly under similar conditions (camera angle, environment, etc.) to the training/test data set can be produced. The ability for such a model to generalise to other environments has not been determined, and the possibility that the models trained in this report overfit their training data cannot be ignored. However, assuming the possible environments for rearing calves is limited, a small selection of models could be trained, which farmers could select from.

The results of this project indicate it is possible to detect the location of calves in a frame and their head position, with a certain degree of confidence, which can then be used to produce summary metrics. A lack of time and available tagged data prevented investigation of whether models could be trained to identify calves individually, however the framework and code produced in this project could be used to perform such training and assess the resulting models.

Other possible areas for future investigation include:

- **Applying augmentation to the tagged data:** flipping, rotating and adjusting the brightness to artificially increase the training and test dataset. The functionality to flip and rotate images is present in the project code, but time was not available to employ it on the datasets.
- **Clusters:** Implementing a metric to detect and report on the clustering of calves. This could be used to identify calves that are not socialising; a potential illness indicator.
- **Object tracking:** an attempt could be made to track individual calves and determine their degree of activity. If it is not possible to train an object detection model to uniquely identify each calf, tracking individual calves may be a suitable alternative way to monitor their behaviour.
- **Re-introducing the ability to detect humans:** a farmer delivering food or new bedding should be an exciting event for calves; not exhibiting increased activity in such circumstances could be a sign of illness.
- **Transfer learning with light/mobile models:** the time needed to detect objects in a single frame tended to be around the 1 second mark, with far superior results again seen on GPU-enabled devices. Whilst potentially suitable for offline analysis on specialised servers, to enable real-time detection on remote devices it may be necessary to investigate training models designed for mobile devices.

References/Resources

1. TensorFlow Object Detection API: https://github.com/tensorflow/models/tree/master/research/object_detection
2. TensorFlow Detection Model Zoo: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md
3. COCO Dataset: http://cocodataset.org
4. Microsoft VoTT: https://github.com/microsoft/VoTT
5. Google Colabs: https://colab.research.google.com

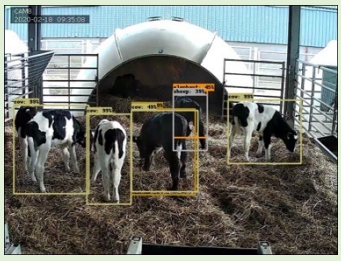
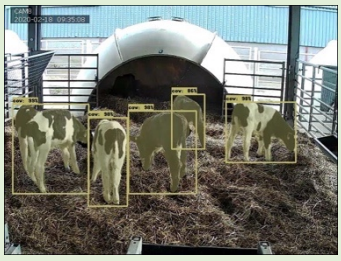
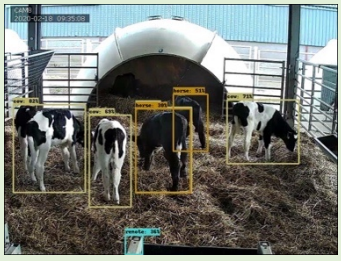
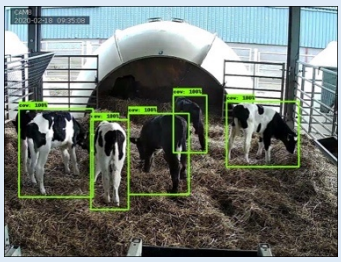
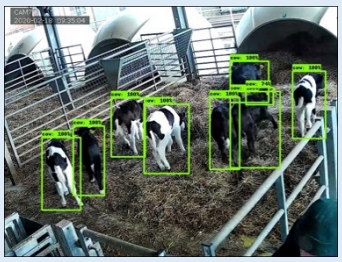
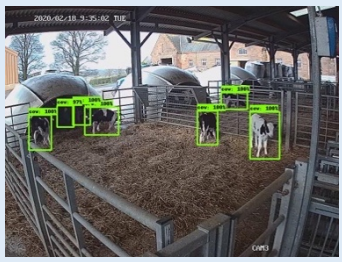
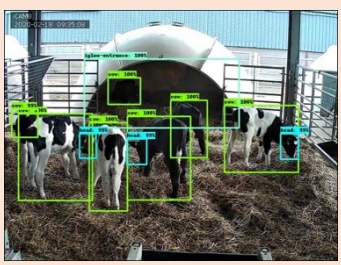
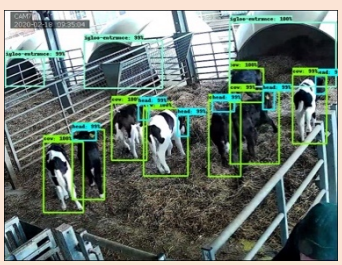
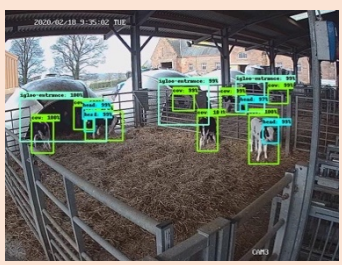
	Video			
	1	5	6	8
Original Frames				
Faster R-CNN ResNet-50				
SSD ResNet-50 FPN				
Mask R-CNN ResNet-101 Atrous				
SSD MobileNet v1 FPN				
Faster R-CNN ResNet-50 with Transfer Learning - Cows Only				
Faster R-CNN ResNet-50 with Transfer Learning – Cows, Heads and "Igloo" Entrance				

Table 6 – Example video frames showing the objects detected by the models used in Part 1 (green rows), Part 2 (blue row) and Part 3 (orange row) of this report. Only objects with a prediction score greater than or equal to 0.25 are shown.

Appendix: Calculating the Mean Average Precision (mAP)

To determine the mAP for an object detection model it is first necessary to calculate the intersection over union (IoU) of all detections. As shown by Equation A2, IoU is the ratio between the area of intersection and the area of union of the ground truth (i.e. manually tagged) bounding boxes and those predicted by the model (see Figure A1).

For a given class of object, we use the IoU to determine the number of true positive (TP), false positive (FP) and false negative (FN) predictions, which are defined as follows:

- **TP**: bounding boxes where the IoU with the ground truth is above a defined threshold (typically 0.5), and the correct object class has been identified
- **FP**: bounding boxes where the IoU with the ground truth is below a defined threshold
- **FN**: instances where the model failed to produce a bounding box for a ground truth

True negatives (TN) are not evaluated as all images are expected to contain at least one ground truth.

Using the definitions above, we can calculate the precision (Equation A3) and recall (Equation A4) of the detections for a given object class across a test set of images.

Each prediction made by the model has a probability score, which we use to rank the predictions from highest to lowest and generate a table of precision vs. recall values – as show in Table A1. We initially consider only the first prediction, determine if it is a TP, calculate the precision and recall values, and then update these values to include the next prediction, and so on. Given the actual number of objects in an image is fixed (i.e. the number of ground truths tagged), the recall increases with each correctly identified object, whereas precision will increase and decrease as it encounters more true and false positives.

To calculate the average precision (AP) for a given class of object, we segment the recall values into 11 parts; from 0 to 1 in intervals of 0.1. We generate interpolated precision values (p_{interp}) for each recall value (r) by taking the value of maximum precision occurring at $\tilde{r}$, where $\tilde{r} \geq r$ – as shown by Equation A1. The interpolated results are plotted alongside the true precision and recall values to produce a Precision/Recall (PR) curve – as shown in Figure A2. The average precision is then calculated as the area under the interpolated PR curve.

$$p_{interp}(r) = \max_{\tilde{r} \geq r} p(\tilde{r})$$

Equation A1

This AP calculation method was used in the Pascal VOC2008 competition and has since been improved – primarily to enhance the ability to measure differences for object classes with low AP value. However, the 2008 method generates a suitable metric for performing comparisons in this project.

The mean of the AP values calculated for each object class in the ground truth label set values is the mean average precision (mAP) of the object detection model.

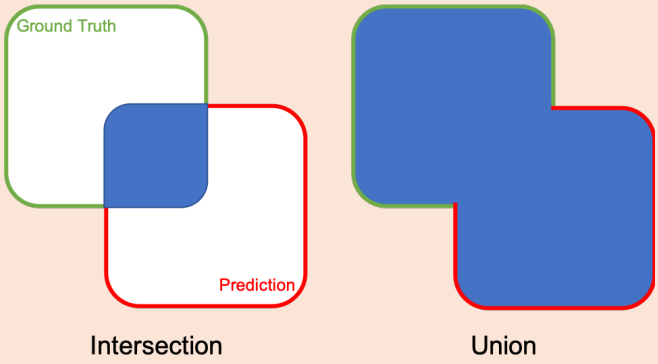


Figure A1 – Illustration of the intersection and union between ground truth and predicted bounding boxes

$$IoU = \frac{a_{intersection}}{a_{groun\ truth} + a_{prediction} - a_{intesection}}$$

Equation A2 - Equation for determining IoU

$$Precision = \frac{TP}{TP + FP}$$

Equation A3 - Precision Equation

$$Recall = \frac{TP}{TP + FN}$$

Equation A4 - Recall Equation

Rank	Score	TP	TP Cumulative Total	Precision	Recall
1	0.933	1	1	1.00	0.17
2	0.855	1	2	1.00	0.33
3	0.829	0	2	0.67	0.33
4	0.811	0	2	0.50	0.33
5	0.751	0	2	0.40	0.33
6	0.615	1	3	0.50	0.50
7	0.587	1	4	0.57	0.67
8	0.384	0	4	0.50	0.67
9	0.367	0	4	0.44	0.67
10	0.325	0	4	0.40	0.67
11	0.311	0	4	0.36	0.67
12	0.241	0	4	0.33	0.67
13	0.220	0	4	0.31	0.67
14	0.171	0	4	0.29	0.67
15	0.166	1	5	0.33	0.83
16	0.086	1	6	0.38	1.00

Table A1 - Precision and recall calculations – table is in descending order based on the score

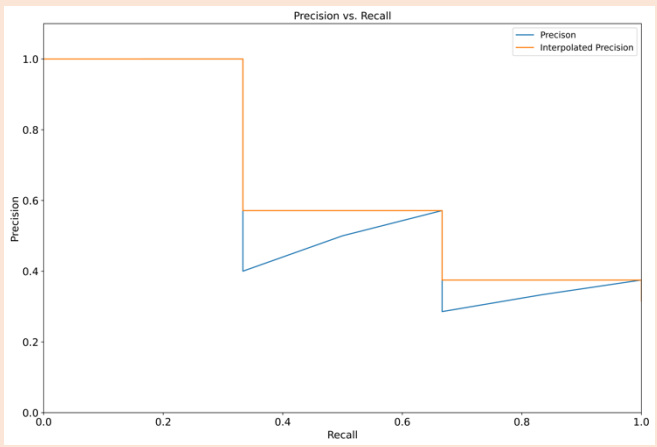


Figure A2 - An example PR curve