Supplementary Material

1. Human annotation user interfaces

Supplementary Figure 1 shows a screenshot of the frame-level classification tool. The segment-level tool was very similar.

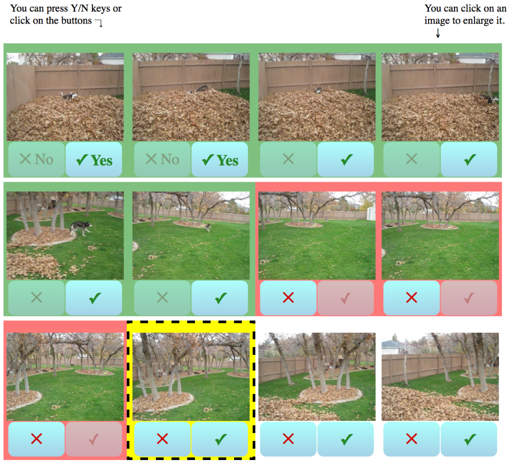
Supplementary Figure 2 shows a screenshot of the bounding box drawing tool (stage 3). The bounding box verification tool (stage 4) was similar. In stages 3 and 4, annotators paid careful consideration to object identity: for example, two different dogs in a segment must result in boxes drawn around only one of the dogs. Moreover, all boxes around that one dog must be annotated. For stages 3 and 4, a drawing-verification approach was chosen over a repeated-drawing strategy for two reasons. First, verification is faster this way. Second, having a single drawn box anchors the attention of the verifiers, avoiding problems when multiple instances of the object class are present.

2. Attention span of human annotators

In order to gather reliable data, it was necessary to define the classes precisely, so as to avoid too many corner cases. For example, just asking whether an "airplane" is present may bring up questions like: "what if it's a toy airplane?". Ideally, one would have liked to present the human annotators with the dictionary definition for the class. In practice, however, the attention span of the average untrained, unvetted annotator made this infeasible. In fact, we found that in order to get consistent answers it helped to simplify the questions as much as possible. Some annotators tended to not read the questions completely, even when they consisted of only a handful of lines. This presented a dilemma: on the one hand we needed well-defined classes, on the other hand the questions had to be short. To resolve this dilemma, we opted to split a question into a series of binary choices. Each choice was made by a different rater. Only frames which got a positive result for a given choice made it to the next choice. For example, for the segment-level annotations for the "airplane" class, we used the following three choices:

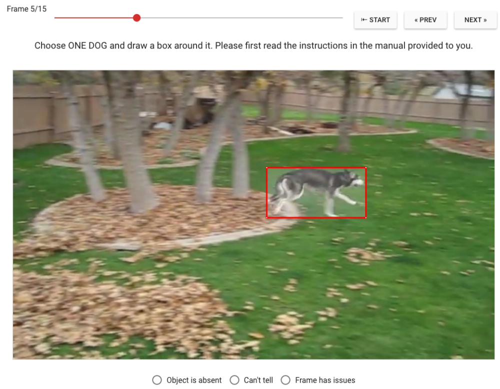
1. Can you see the OUTSIDE of a **real airplane** in any frame? Please answer YES even if you cannot see the whole airplane, provided you are confident it is an airplane. Include seaplanes, stealth bombers, *etc*.

2. If the airplane in these frames is:
   • filmed from the perspective of **someone outside the plane** like a ground observer or someone on another plane → answer **YES**;
   • filmed from the perspective of **someone inside the plane** like its pilot or a passenger → answer **NO**.
   If **uncertain**, please answer **NO**.


You can press Y/N keys or click on the buttons ⌐

You can click on an image to enlarge it. ↓

Supplementary Figure 1. Screenshot of the human annotation tool used to gather frame-level classification labels (stage 2). For each frame, the annotators has to answer whether the class was present or absent. A similar tool was used for segment-level labels (stage 1), displaying fewer frames and allowing only one answer per segment.


Frame 5/15

Choose ONE DOG and draw a box around it. Please first read the instructions in the manual provided to you.

○ Object is absent ○ Can't tell ○ Frame has issues

Supplementary Figure 2. Screenshot of the human annotation tool used to gather bounding boxes (stage 3). At the top, a scroll bar allows navigating through the frames of the segment. The box (red rectagle) can be drawn by clicking and dragging on the image. Three categorical options at the bottom allow the annotator to indicate (i) the absence of the object, (ii) uncertainty, or (iii) problems with the interface. A choice of (i) created an *absent-tag*, which was included in the data set. A choice of (ii) or (iii) resulted in the annotation being discarded. A similar tool was used for the verification stage (stage 4), which instead presented an unchangeable box and an option button to enter the correctness of the box.

3. If the airplane in these frames is:
   • **REAL** → answer **YES**;
   • **NOT REAL** like a **TOY**, cartoon, or **VIDEO GAME** → answer **NO**.
   If **uncertain or no airplane**, please answer **NO**.

Notice how some options were structured so that most of the information is at the beginning of the question ("Can you see the outside of a real airplane [...]"). Also, acting on the assumption that annotators read the question only up to the point when they feel they know what it is about, we employed another design principle: structuring the first phrase so that it conveyed zero information until it conveys most of the information. In the example, the phrase "If the airplane in these frames is filmed from the [...] answer yes" tells you very little about what the task unless it is read up to the last word. Finally, using caps, bold, and bullets may have helped keep the annotators attention on the text for a bit longer.

3. Bounding box drawing guidelines

The following rules were observed by annotators during stages 3 and 4.

- Objects should be boxed even if only a small part is visible, as long as it is recognizable (*airplane example* in figure 1).
- It does not need to be recognizable within the frame in question. The context provided by other frames can be used to deduce the object's identity (*train example* in figure 1).
- Only the visible part of the object should be boxed. No inference can take place as to hidden or out-of-frame parts (*bear example* in figure 1).
- If an object extends on either side of an occlusion (for example, an elephant behind a narrow tree), one box should be used to include all the visible parts of the object (*airplane example* in figure 1).
- The first box is drawn on a random frame within the segment that has a positive classification according to stage 2. (After that, the annotator works forward and backward from that frame.)

4. Human annotation detailed statistics

Supplementary Tables 1 and 2 show the complete counts for all classes for classifications and detections, respectively. Supplementary Table 3 shows quantitative measures of size and motion for the bounding boxes (next pages).

5. Relevant GitHub locations

The following are locations for related GitHub models:
Inception-v3:
https://github.com/tensorflow/models/tree/master/slim
Inception-ResNet-v2:
https://github.com/tensorflow/models/tree/master/slim
Faster-RCNN:
https://github.com/rbgirshick/py-faster-rcnn

6. Per-class object detection baseline

Supplementary Table 4 shows the difficulty of object detection for each class (next pages).

	Positives		Negatives	
	Frames	Videos	Frames	Videos
airplane	384,448	9,314	38,847	4,446
bear	354,730	7,792	57,595	5,493
bicycle	266,317	6,352	11,121	2,348
bird	476,734	11,239	49,680	5,569
boat	370,723	10,920	24,630	3,667
bus	410,742	13,800	60,724	7,473
car	306,850	10,733	26,539	2,372
cat	694,265	33,019	55,281	8,582
cow	465,637	17,201	69,988	9,028
dog	555,055	15,748	37,606	5,993
elephant	319,778	7,469	47,802	4,900
giraffe	49,031	1,660	8,400	1,094
horse	532,403	12,494	36,681	5,292
knife	506,180	9,563	34,256	3,518
motorcycle	338,870	12,900	24,096	4,523
person	1,810,968	79,319	132,449	21,700
potted plant	236,509	6,940	21,326	2,766
skateboard	440,274	13,499	63,138	10,274
toilet	153,312	9,895	83,915	7,994
train	339,639	11,628	76,197	5,361
truck	343,773	10,672	30,232	3,891
umbrella	189,727	7,784	25,805	4,325
zebra	26,169	1,070	7,493	823
NONE	26,457	1,589	–	–
ALL	9,527,784	316,235	1,021,508	128,712

Supplementary Table 1. Human annotation classification counts. We count the number of unique frames and unique videos that have been annotated as having (“positives”) or not having (“negatives”) the class. Due to the fact that we are listing *unique* videos, and the fact that occasionally more than one class is annotated per video, the “ALL” row is not necessarily the sum of the class rows.

	Bounding Boxes		Absent Tags	
	Frames	Videos	Frames	Videos
airplane	223,712	6,932	45,319	3,621
bear	231,264	6,271	31,611	3,610
bicycle	189,955	6,122	70,911	4,168
bird	228,363	8,434	42,927	4,367
boat	225,819	8,419	41,001	4,073
bus	210,565	9,132	59,121	5,670
car	246,807	9,506	25,354	2,748
cat	251,472	13,828	21,867	3,882
cow	197,630	10,732	73,058	7,259
dog	240,308	10,229	31,717	4,780
elephant	220,213	6,297	50,059	4,324
giraffe	42,378	1,601	10,587	1,149
horse	232,774	8,466	42,356	4,318
knife	264,296	6,837	11,785	2,127
motorcycle	223,333	10,516	48,266	4,828
person	1,285,776	68,427	283,112	36,075
potted plant	169,260	6,036	70,349	4,889
skateboard	192,731	9,352	75,308	7,752
toilet	139,783	9,342	79,622	7,558
train	239,737	8,861	45,897	3,783
truck	228,212	8,484	38,366	3,882
umbrella	114,040	5,123	90,111	6,101
zebra	20,113	1,019	7,989	782
ALL	5,597,399	236,102	1,291,979	129,465

Supplementary Table 2. Human annotation detection counts. We count the number of unique frames and unique videos that have been annotated with bounding boxes (if the object is present) or absent tags. Due to the fact that we are listing *unique* videos, and the fact that occasionally more than one object is annotated per video, the “ALL” row is not necessarily the sum of the class rows.

	PF	CF	MA	C-RMS	A-RMS
airplane	0.86	0.80	0.43	0.094	0.103
bear	0.88	0.80	0.24	0.106	0.083
bicycle	0.72	0.65	0.24	0.138	0.092
bird	0.78	0.69	0.19	0.155	0.085
boat	0.87	0.79	0.26	0.114	0.087
bus	0.80	0.73	0.42	0.086	0.123
car	0.91	0.85	0.58	0.075	0.095
cat	0.92	0.84	0.44	0.115	0.121
cow	0.72	0.65	0.30	0.120	0.102
dog	0.86	0.76	0.27	0.165	0.125
elephant	0.80	0.73	0.32	0.100	0.102
giraffe	0.78	0.71	0.35	0.115	0.121
horse	0.84	0.75	0.22	0.129	0.107
knife	0.96	0.89	0.33	0.122	0.126
motorcycle	0.83	0.75	0.46	0.126	0.127
person	0.80	0.70	0.25	0.122	0.096
potted plant	0.77	0.71	0.41	0.094	0.091
skateboard	0.71	0.58	0.05	0.190	0.047
toilet	0.65	0.56	0.41	0.148	0.123
train	0.87	0.81	0.50	0.072	0.111
truck	0.87	0.81	0.51	0.083	0.113
umbrella	0.78	0.70	0.37	0.122	0.123
zebra	0.67	0.60	0.33	0.119	0.122

Supplementary Table 3. Measures of object motion. Each value is an average over all the segments for the corresponding class. *Present Fraction (PF)*: fraction of the segment frames in which the object is present. *Continuous Fraction (CF)*: fraction of the frames in the longest sequence in which the object was continuously present. This is an indication of how often the object enters and leaves the field of view. *Mean Area (MA)*: mean area of the box. *Center RMS (C-RMS)*: root-mean-square of the distances the center of the box travels from each frame to the next. This is a measure of sideways object/camera motion. *Area RMS (A-RMS)*: root-mean-square of the change in area from each frame to the next. This is an indication of the amount of depth-wise object/camera motion. *For all*: areas and distances are measured in the relative coordinate system in which both axes run from 0 to 1, regardless of the aspect ratio of the video. Distances and area changes were only measured over contiguous frames. Everything is based on data at 1 frame per second. Note that there may be significant motion not captured by these quantities, such as: (i) relative movement “in place” like in the case of a spinning wheel, (ii) movement of the background, as would be seen when a racecar is kept well centered in the field of view but the background “passes by”, or (iii) movement of an object that spans the field of view such as a train passing by while a steady camera only captures one wagon at a time.

	COCO model		YT-BB model	
eval on:	COCO	YT-BB	COCO	YT-BB
airplane	0.56	0.65	0.41	0.72
bear	0.80	0.45	0.66	0.68
bicycle	0.27	0.15	0.14	0.40
bird	0.17	0.29	0.15	0.45
boat	0.17	0.23	0.09	0.47
bus	0.56	0.61	0.47	0.77
car	0.29	0.43	0.06	0.81
cat	0.72	0.49	0.61	0.62
cow	0.40	0.38	0.25	0.59
dog	0.58	0.29	0.48	0.52
elephant	0.58	0.55	0.50	0.67
giraffe	0.80	0.77	0.54	0.67
horse	0.63	0.40	0.41	0.56
knife	0.10	0.16	0.02	0.60
motorcycle	0.35	0.48	0.26	0.59
person	0.41	0.12	0.23	0.41
potted plant	0.21	0.15	0.08	0.39
skateboard	0.42	0.41	0.22	0.43
toilet	0.68	0.60	0.29	0.71
train	0.49	0.58	0.55	0.73
truck	0.29	0.38	0.14	0.71
umbrella	0.26	0.29	0.11	0.55
zebra	0.56	0.87	0.49	0.59

Supplementary Table 4. Measured difficulty of object detection for each class trained on the COCO and YT-BB data sets. All values are calculations of the mean average precision (mAP) across precision-recall curves. Each column indicates evaluation on COCO and YT-BB, respectively.