Object Permanence in Videos: DNN Performance vs. Human Ability

Irene Zhou
Dept. of EECS
Dept. of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA, USA
zhoui@mit.edu

Mona Abdelrahman
Dept. of EECS
Dept. of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA, USA
monaabd@mit.edu

Sarah Abodalo
Dept. of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA, USA
sabodalo@mit.edu

***Abstract*—Object permanence, the idea that an object still exists when not in view, is a fundamental aspect of human cognition that typically develops in infancy. However, object permanence is often left out of machine vision models designed to identify objects in video. As machine vision technology becomes more widespread, it is important to understand the key differences between machine and human visual processing, and how we can bridge these gaps. In order to compare object permanence capabilities in humans and machine models, we created a custom video dataset of objects undergoing occlusion, including frames with partial and full occlusions. Human object detection performance on the dataset was assessed via survey, and compared with the SOTA object detection algorithm YOLOv3, as well as a novel recurrence-based model we call YOLO-LSTM. Although human performance degrades significantly when identifying object size and location under occlusion, human subjects almost always recognize that occluded objects still exist, while YOLOv3 shows linearly decreasing performance as objects become more occluded. YOLO-LSTM has a lower accuracy overall than YOLO-v3, but demonstrates more human-like behavior.**

***Index Terms*—deep neural networks, object permanence, machine vision**

I. Introduction

Object permanence is a fundamental, intuitive skill that humans acquire in infancy. When a human sees a stationary object temporarily obscured, they will continue to recognize roughly where the object exists in space and what it is. Little is known about the development of object permanence in humans; although Piaget thought physical manipulation of objects was required, other studies have shown that young infants are able to reason about hidden objects visually [1]. Furthermore, maturation of the frontal lobe is associated with acquiring object permanence, as young infants are able to remember the object's existence for longer durations [2]. However, because object recognition systems typically work on inputs of single images – processing videos as individual frames without memory – the problem of object permanence through partial or full occlusion becomes more difficult to solve.

978-1-7281-7571-3/20/$31.00 ©2020 IEEE

Tracking objects in video and dealing with temporary occlusions has been a longstanding problem in machine vision, and has thus garnered a wide variety of proposed solutions. Many solutions focus on utilizing partial occlusions, such as identifying when an existing object is partially occluded, and using it as a benchmark for the location of the entire object [3]. Although these solutions are promising for partial occlusions, they ignore full occlusions entirely. The most common solutions involve using linear and nonlinear dynamics models – most popularly, Kalman filters – to estimate the trajectory of objects which are occluded temporarily while in motion. These models, however, have been shown to perform poorly for certain situations, such as poor video quality, stationary objects, and nonlinear object motion [4].

While existing solutions primarily focus on identifying partially occluded objects in individual images [5], or in the case of videos, tracking moving objects with fairly linear trajectories, we focus more on an object permanence perspective, with basic, large, moving occlusions in video frames. To this end, we compare human and computational performance in situations of both partial and full occlusions over stationary objects.

Object permanence in machine vision has many implications in industry applications, most notably with the case of autonomous vehicles. One of the biggest issues with these autonomous driving systems is that they will encounter many objects that may be occluded on the road, and objects that move behind others, making it critical that these models are able to handle occlusions and to recognize the existence of blocked objects, even when they are not directly in view [6].

II. Methods

A. Dataset

Our dataset consists of two parts: a pretraining image dataset for the preliminary parts of our model, and a video dataset. The pretraining dataset consists of 10000 320x320-pixel images constructed using the Pillow Python library [7]. The images contain one unoccluded object - either a circle, a square, or a triangle - over a noisy background of small colored pixel blobs and large black bars. Fig. 1a shows an example of a pretraining image. For the purposes of training, the objects in question are

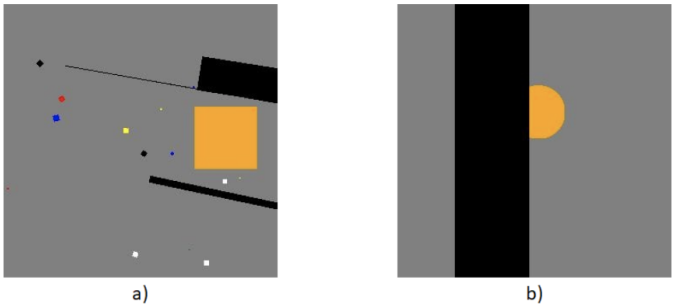

Fig. 1. a) An example of an image used for pretraining, showing the primary shape in orange with a noisy background. b) An example of a frame from the video training dataset, containing a partially occluded object.

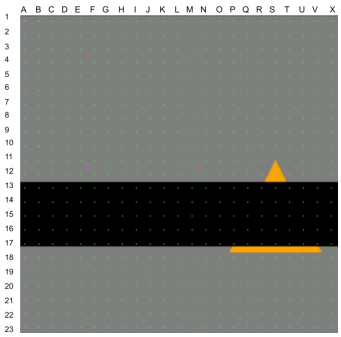

Fig. 2. Example of video frame with a partially occluded object and 23x23 point-grid overlay used in human surveys.

annotated with their label and bounding box. The side length of the object's square bounding box is between 10% and 33% of the side length of the full image. These images are used to pretrain the CNN and YOLOv3 [8].

The video training dataset consists of 1000 15-frame 320x320-pixel videos displaying a black vertical or horizontal bar moving over - and occluding - a single stationary object, as well as videos displaying an object moving under a stationary black bar occlusion. The object's bounding box is always contained entirely within the image. Fig. 1b shows an example of a frame from the video dataset, where a circular object is partially occluded by a vertical occlusion. Although the frame rate (5fps for a 3sec video) is severely limited compared to a typical video (24 fps), the lower frame rate decreases the amount of disk space required to store the datasets, and ensures that both humans and the model process the videos on a frame-by-frame basis. Each object has a bounding box with dimension between 10% and 25% of the dimension of the full image, and each occlusion has a width between 15% and 33% of the image dimension. Approximately 50% of the videos contain a frame where the object is fully occluded. The direction of movement of the bar or image is randomized. The ground truth annotation for these videos is the label and bounding box for the object - including occluded portions - in each frame; these annotations are generated automatically as a part of the image and video creation process in Python. The velocity of the occlusions is about 23 pixels/frame (1/14th of the image dimension) in either the horizontal or vertical direction. Object velocity is on average 23 pixels/frame in all directions.

B. Human Experiment

The stimuli for the human experiment consisted of the aforementioned dataset videos, as well as screenshots from those videos at varying amounts of object occlusion. It was expected that manipulating the percentage of object occlusion would impact the accuracy of bounding box identification.

Our experiment was conducted on Amazon Mechanical Turk (MTurk) as well as a Google Form emailed to the MIT undergraduate community via dormitory mailing lists. Each participant was presented with a comprehension test followed by six trials. Subjects were provided instructions and necessary information to complete the task (regarding object permanence, minimum bounding boxes, etc). Each trial consisted of the participant watching a video from the dataset and identifying the object from a list of labels. The participant then saw a still image of the occluded object overlaid with a 23x23 grid of points (Fig. 2) and was asked to identify the four points in the grid that made up the vertices of a box that most closely contained the full object (simulating a bounding box), including parts which may be occluded. The six trials included a variety of amounts of occlusion, ranging from 10% to 100%. The initial comprehension test consisted of three questions asking the participants to identify an object from an image, to identify a minimum bounding box from several options, and to correctly identify the vertices of a bounding box from a grid of points. The comprehension test was used to ensure participants understood the task, as well as to filter out random responses. Between MTurk and the MIT survey, 251 participants were recruited.

C. Computational Experiment

1) Model Overview: The most important aspects of object permanence are the memory that an object continues to exist even when out of sight, and the ability to extrapolate the object's current location from past locations. Thus, we designed a model with these memory and extrapolation features in mind, using a recurrent structure to retain historical information for predicting the bounding box and label of the object at each frame of the video. The recurrent layers receive two streams of spatial information: a general representation of the major features in each video frame, and an estimation of the bounding box and label for each frame without historical input. The bounding box and label estimate provide a jumping-off point for the final output prediction, and the general frame features provide supportive information about each frame that inform any deviations from the estimated output. To generate the frame features, we chose to use convolutional layers, which are widely used for this purpose in image processing. We also chose YOLOv3 - a state of the art object detection algorithm - for the output estimations, and LSTM - which has been shown

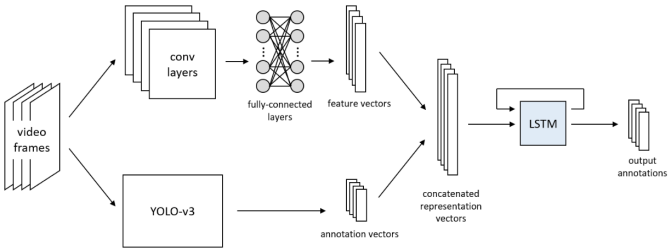

Fig. 3. Structure of the YOLO-LSTM model. Video frames are fed sequentially into pre-trained YOLO-v3 and convolutional layers, the outputs of which are concatenated and passed through an LSTM to produce the final image annotations for each frame.

to be effective for processing sequences and retaining long-term historical information - for the recurrent structure.

Our computational model (Fig. 3) is inspired by existing work [9] on object tracking and occlusion handling in video which supports the effectiveness of recurrent structures in video object tracking. The full model consists of three parts: CNN, YOLOv3, and LSTM. The CNN is trained to classify the objects in the images, and is used to extract an informative feature vector from images similar to those used in our video dataset. YOLOv3 is trained to identify the bounding box and label of the object in images, and outputs the predicted annotation vector containing the label and bounding box of the object detected in the image with the highest confidence. If no object is detected in the image, YOLOv3 outputs an annotation vector of zeros. For the video dataset, the feature vector from the CNN and the annotation vector from YOLOv3 are concatenated to form a vector representation of each frame in a video, and the vectors are passed into the LSTM sequentially in single-video sequences of frames with the combined annotations of the CNN and YOLOv3. The LSTM outputs a corresponding final sequence of vectors containing a label and bounding box for each frame.

In order to make the architecture more efficient, the CNN and YOLOv3 are first trained on the pretraining image dataset, which contains background noise but no object occlusions. The video frames are then passed through both the trained CNN and YOLOv3 individually to generate sequences of representative vectors used to train the LSTM. The model was implemented in PyTorch, and all computations were performed on a Tesla K80 GPU via Google Colab.

2) CNN Details: The CNN takes 3-channel RGB image inputs, and consists of 4 max-pooled 2D convolutional layers, followed by 3 fully-connected layers. The max pooling window is 2x2, and the convolutional layers have kernel size 5 with output channels 16, 32, 64, 64 respectively. The fully-connected layers have output size 512, 64, 3 respectively, with the last layer representing the number of classes in our data. ReLU is used as the activation function for all of the convolutional and fully-connected layers, but the output layer has a linear activation. Because the network is performing strictly single-class classification, we use a cross-entropy loss function. After pretraining, we draw the feature vector from the second-to-last fully-connected layer in the CNN, resulting in a feature vector of size 64.

The CNN was trained for 15 epochs on our 10000-image pretraining dataset, with a batch size of 64. The optimizer was basic SGD, with a 0.001 learning rate and 0.9 momentum. This training resulted in a classification loss of 0.03.

3) YOLOv3 Details: We used an available open-source implementation of YOLOv3, specifically YOLOv3-Tiny [8]. This smaller version of YOLOv3 is much more lightweight, and since there are only 3 classes in our dataset, it provides similar performance as regular YOLOv3. For this part, we started off with weights that were trained on the MSCOCO database. The YOLOv3-Tiny model was then trained on our pretraining data with 16 epochs of 500 iterations each, and these new weights were saved. The video frames were then passed through the trained model to produce the predicted annotations for our objects.

In many of the video frames, YOLOv3 detected zero or multiple objects. Because YOLO-LSTM is designed to exclusively handle single object detection, we represented zero detection as a vector of zeros and used only the object detected by YOLOv3 with the greatest confidence in each frame. This resulted in a processed output consisting of a one-hot vector representing the object label, and a 4-element vector representing the center coordinates and dimensions of the bounding box (x, y, w, h).

4) LSTM Details: The LSTM takes inputs of size 72 (the concatenated outputs of the CNN and YOLOv3) and has 2 hidden layers of size 512. The hidden layers are followed by a fully-connected layer corresponding to each output. The model outputs two vectors: one of size 3 representing the object classification and the other of size 4 representing the bounding box. We implemented a custom loss function to deal with the two different pieces of this output. For the object classification, we used cross entropy loss (CE), which is commonly used in single-class classification. For the bounding box localization, we used root mean squared error (RMSE) on the bounding box vector, a standard regression loss function. Because the bounding box vector is represented as proportions of the full image size within the range [0, 1], the RMSE tends to be minimized in comparison to the classification loss, and the model may be biased towards optimizing the bounding box vector alone. Thus, we add a multiplier β to the RMSE before summing the two losses to get the final value. The value of the multiplier (in this case, $\beta = 5$ was determined via cross-validation.

$$Loss = CE(label) + \beta * RMSE((x, y, w, h)) \quad (1)$$

The LSTM was trained for 1000 epochs on our 1000-video training dataset, with a batch size of 1, representing one video as a sequence of 15 frames per iteration. We used an Adam optimizer with learning rate 0.001.

III. Results

A. Human Experiment

We evaluated human accuracy for the bounding boxes by comparing the vertices provided by participants to the ground truth, which was determined manually using the same point grid. Object localization accuracy was evaluated using the IOU (intersection-over-union) metric. Object identification accuracy was calculated by comparing the number of correct responses to the total number of responses. Of the 251 total responses, 215 remained after filtering out participants who incorrectly answered the comprehension test or incorrectly identified the object for more than half of the trials. Bounding box vertices were sorted into one of four categories based on whether the bounding box was correct, cut off part of the object, was too large, or was incorrect or ineligible. The 'cuts off object' category includes responses in which the occluded part of the object was neglected as well as responses in which the presence of the occlusion was acknowledged, but the bounding box provided was not large enough. Incorrect or ineligible responses were defined as responses which did not provide coordinates that formed a box or coordinates that did not contain the object at all.

Fig. 4 shows the results of this analysis (based on the data from Table 1). Evidently, the percentage of accurate bounding box annotation remains relatively stagnant occlusion but experiences a significant drop in accuracy upwards of 83%.

As mentioned before, bounding box accuracy across the various categories was evaluated using the IOU metric. Fig. 5 shows the average IOU across participants at various amounts of occlusion. The quality of bounding box annotation remains relatively stagnant with a significant drop in accuracy when occlusion is upwards of 83%.

To evaluate the object identification accuracy, percentages were calculated as the number of correct identifications over the total number of eligible responses. Table 2 conveys these results. Object identification accuracy and bounding box accuracy have a positive correlation, according to Pearson's correlation coefficient ($r = .59$).

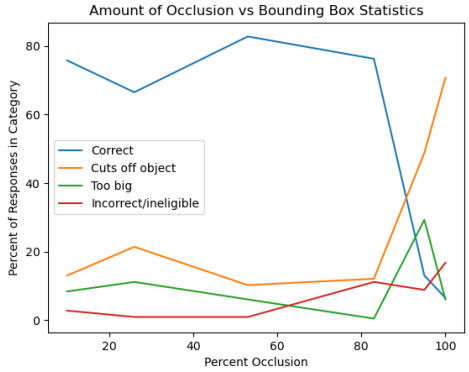

Fig. 4. Proportion of responses falling in each mistake category for varying amounts of object occlusion.

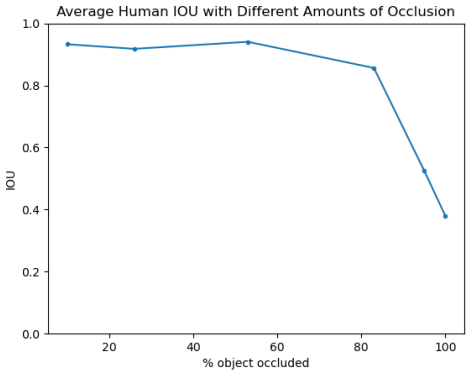

Fig. 5. Average IOU of valid human survey responses for different amounts of occlusion.

TABLE I
PROPORTION OF SURVEY RESPONSES IN EACH MISTAKE CATEGORY

% Occluded	Correct	Cut off object	Too big	Incorrect
10%	75.81%	13.02%	8.37%	2.79%
26%	66.51%	21.40%	11.16%	0.93%
53%	82.79%	10.23%	6.05%	0.93%
83%	76.28%	12.09%	0.47%	11.16%
95%	13.02%	48.84%	29.30%	8.84%
100%	6.51%	70.70%	6.04%	16.74%

TABLE II
PROPORTION OF SURVEY RESPONSES WITH CORRECT OBJECT LABEL

Occlusion Amount	Correct ID
10% occluded	97.67%
26% occluded	97.67%
53% occluded	97.67%
83% occluded	98.60%
95% occluded	98.14%
100% occluded	95.81%

B. Computational Experiment - YOLOv3 Intermediate Data

In order to demonstrate the existing occlusion-handling power of YOLOv3, we collected intermediate data from the first half of our network. Individual frames from the video dataset were run through the trained YOLOv3 model to evaluate its object recognition performance. Similar to the evaluation of the human data, the performance of the mode on object localization was evaluated using the IOU metric, and performance for object classification was evaluated with a simple accuracy.

As we can see in Fig. 6, the IOU of the model has a negative linear relationship with the percent of the object that is occluded, unlike the sharp dropoff in accuracy seen in the human data. From examples of YOLOv3 outputs (Fig. 9), we can see that this relationship is a direct result of the bounding box capturing only the unoccluded portions of each object. This graph was generated by binning ranges of occlusion percentage and averaging all of the IOU values associated with

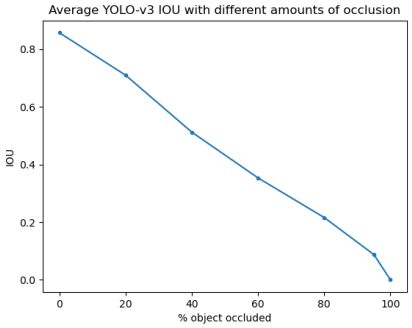

Fig. 6. Average IOU of bounding boxes outputted by YOLO-v3 alone for different amounts of occlusion.

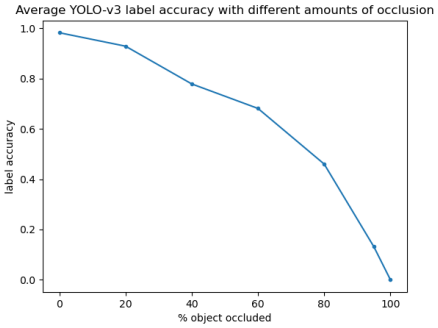

Fig. 7. Proportion of objects labeled correctly by YOLO-v3 alone for varying amounts of occlusion.

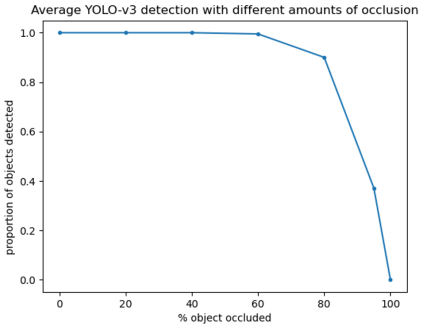

Fig. 8. Proportion of objects which are successfully detected by YOLO-v3 in images with varying amounts of occlusion.

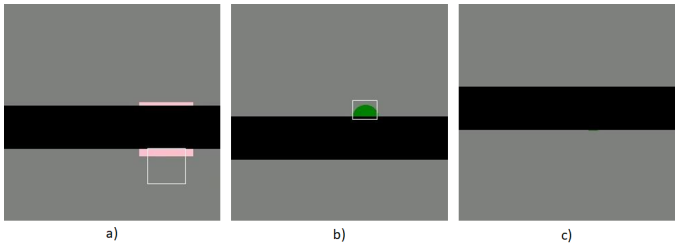

Fig. 9. Examples of common mistakes made by YOLO-v3 in detecting occluded objects. Predicted bounding boxes are shown in white. a) Minimal overlap between bounding box and square object; bounding box does not overlap with the occlusion. b) Bounding box around non-occluded portion of circular object; minimal overlap with occlusion. c) No object detected (full occlusion).

those occlusions (e.g. the IOU at 40% occlusion is the average of occlusions in the range [30%, 50%)).

Fig. 7 uses the same binning method to show that label accuracy (correctly identifying the object in the frame) also drops off to zero as % occlusion rises to 100, but not in a linear fashion. Errors also include if no object is recognized in the frame. This is expected behavior from a model that can only process single images.

Fig. 8 also uses the same binning method, this time showing the proportion of cases where YOLOv3 fails to detect an object in the image entirely. While YOLOv3 can successfully recognize that an object is in the image when the object is up to 60% occluded, there is a sharp dropoff starting at 80% occlusion. Of the total 15000 frames that made up the video dataset, only 14128 had objects that were recognized by YOLOv3. With YOLO-LSTM, we hope to mitigate this effect.

Fig. 9a shows an example of a YOLOv3 error when the object is mostly occluded. As shown here, YOLO does not recognize that the object is under the occlusion, and most of the predicted bounding box lies outside of the object itself. However, in this case the object was recognized correctly.

Fig. 9b shows an example of an incorrectly identified object with an inaccurate bounding box predicted by YOLOv3. This bounding box is approximately cut off where the occlusion meets the object. This is also an interesting example because YOLOv3 predicted the shape as a triangle.

Fig. 9c shows a case for which YOLOv3 makes no prediction. This means that YOLOv3 does not recognize that the object has been passed through under the occlusion. YOLOv3 usually returns nothing for this prediction, but the pretrained model was modified to return all zeroes for these cases.

C. Computational Experiment - YOLO-LSTM Final Data

We evaluated YOLO-LSTM using a 500-video test dataset generated using the same method and parameters as the training data, and compared the performance of YOLOv3 by itself and the combined YOLO-LSTM model. The training and test datasets had no overlapping items. The greatest points of interest are the average IOU and label accuracy for varying amounts of occlusion.

We can see from Fig. 10 that although YOLO- LSTM has, on average, a lower IOU than YOLOv3 alone, YOLO-LSTM performs far better at higher levels of occlusion. However, YOLO-LSTM often outputs bounding boxes that are too large or offset from the ground truth, resulting in an IOU of just 0.57 even with no occlusions. Qualitatively, the shape of the IOU curve for YOLO-LSTM is much more similar to the shape of the human IOU curve, and does not mirror the linear decrease seen with YOLOv3 alone. Although YOLO-LSTM makes some YOLO-like errors (Fig. 12) with partial occlusions, it tends to make more human-like errors with full occlusion (Fig. 13).

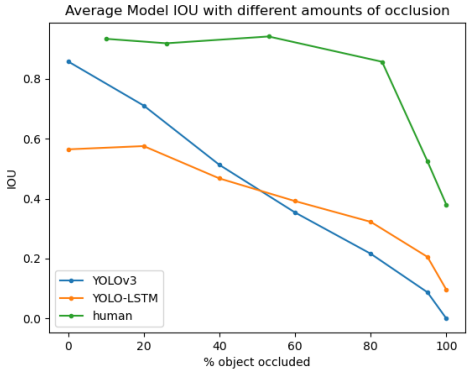

Fig. 10. Average IOU of bounding boxes produced by humans, YOLO-v3, and YOLO-LSTM for different amounts of occlusion.

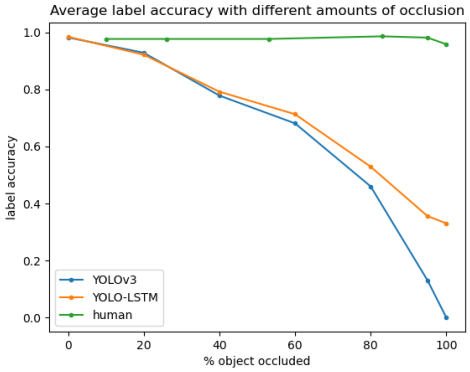

Fig. 11. Proportion of objects labeled correctly by humans, YOLO-v3, and YOLO-LSTM for varying amounts of occlusion.

Fig. 11 demonstrates that YOLO-LSTM has a similar label accuracy as YOLO-v3 from 0-40% occlusion, and surpassesYOLO-v3 at higher levels of object occlusion. However, it is important to note that the model still makes YOLO-like labeling errors for partially occluded frames, and drops to 33% accuracy - the same as random chance - at 100% occlusion.

Fig. 12 and Fig. 13 provide examples of common types of errors that YOLO-LSTM makes when the object is under partial and full occlusion. The first and last frames in both figures represent a low-occlusion scenario, and when compared to Fig. 9a, we can see that both YOLO-LSTM and YOLOv3 make oversized bounding boxes centered farther away from the occlusion than the center of the object. However, in most of the frames, YOLO-LSTM bounding boxes in occlusion conditions generally overlap with the occlusion, implying some understanding that the object is not fully in view. By contrast, YOLOv3 never significantly overlaps the bounding box with the occlusion. YOLO-LSTM also makes very humanlike errors when the object is fully occluded (the third frame in Fig. 12 and Fig. 13). Similarly to human results, YOLO-LSTM does not accurately identify the exact location of a fully-occluded object, but does place the bounding box over the occlusion.

Fig. 12. 5 frames of a vertical occlusion passing over a square object, with bounding boxes defined by YOLO-LSTM in white.

Fig. 13. 5 frames of a circular object passing under a horizontal occlusion, with bounding boxes defined by YOLO-LSTM in white.

The model retains information that the object continues to exist, and recognizes that the object is occluded.

IV. Discussion

We expected to see very high accuracy for the bounding box task in the behavioral experiment given that object permanence is acquired in infancy. Therefore, the bounding box accuracy results were rather surprising; one may expect a decline in accuracy with increasing amounts of occlusion given the nature of the task, but such significant decline was not expected. Object identification accuracy also proved to be quite high, despite a slight decline in accuracy in the full occlusion condition. Because the objects were basic shapes and subjects were provided the three possible label choices, high identification accuracy was expected.

Although the low bounding box accuracy was not expected in the full occlusion condition, one should note the importance of the number of responses. Out of the 215 responses, only 36 participants provided a bounding box that did not include the object or did not provide a bounding box at all. This pattern of responses indicates that participants understand object permanence, but had trouble identifying the exact location of the minimum bounding box (Fig. 14). This pattern is also evident in the 95% occlusion case.

The lower than expected accuracy across all conditions can be attributed to possible confounds in the experiment in comparison to more natural situations. For example, the artificial nature of our occlusions could have led to suspicions of further video editing. The method of identifying vertices of a bounding box out of a grid may also have led to higher error rates than if participants were allowed to draw their own bounding box. There also exists the possibility that participants misunderstood or misread the task, for example counting the occlusion as a shape or neglecting the occluded portions of the objects. For the purposes of this study, the human experiment was designed to work in a simple, text-based format which could easily be shared with a wide audience across a variety of different platforms (in this case, primarily MIT undergraduates located worldwide). In the future, it would be interesting to

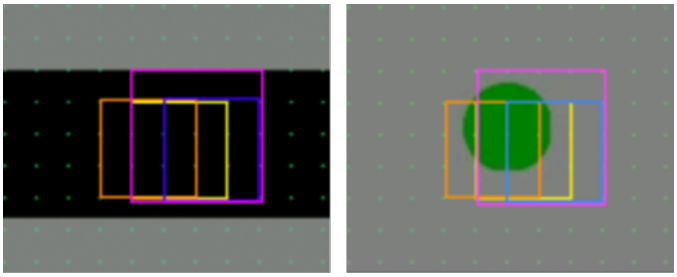

Fig. 14. The 4 most popular bounding box responses (accounting for 33% of all responses) provided by survey participants in the fully occluded condition, shown on a) the occlusion, and b) the true object with occlusion removed.

replicate this experiment with a more sophisticated design which allows participants to actively draw a bounding box.

The results of this model are promising, but lack frame-by-frame accuracy for object localization, particularly for states where the object is unoccluded. There are many potential ways in which this problem could be mitigated. For example, due to storage and processing limitations, the dataset and model used for this study were kept relatively small and simple - with a small feature vector, a training set of 1000 videos, and few hidden layers. YOLO-LSTM could potentially show improvement simply by increasing the number of videos in the training dataset, and increasing the number of hidden layers and the size of the feature vector from the CNN. Another area of improvement is the loss function; the current loss function performs a simple regression on the bounding box vector (x, y, w, h), which is not representative of the actual metrics we use to evaluate bounding box correctness. This could be improved by incorporating the spatial overlap concept of IOU into the loss function itself. The network could also benefit from more informative general frame features; the CNN currently extracts image features with the sole purpose of classification, which may not create features that are particularly applicable to object localization, and would not extend to images containing multiple objects. Future improvements on this model could use an alternative implementation of the CNN that also considers object detection, potentially even sourcing features from the internal layers of YOLOv3.

The results of the human experiment and YOLO-LSTM also indicate a more nuanced interpretation of human object permanence. Simply comparing predicted and actual bounding boxes - particularly when the object is heavily occluded - may not fully represent the human understanding of object permanence. For example, IOU is the same for two different cases: complete failure to detect an object, and non-overlapping bounding boxes, but there is a notable difference between misidentifying the exact location of an object and thinking it has disappeared entirely. Our experiments show that humans do not retain very accurate information about the exact size and location of an object when it is occluded, but they do know that the object continues to exist behind the occlusion, and tend to select a bounding box over the occlusion. Even if a human participant completely misses the object's location, predicting that the object is somewhere behind the occlusion is still a stronger indication of object permanence than predicting that the object has disappeared. This demonstrates a critical flaw in the IOU metric, and indicates that it may be important to develop more humanlike metrics for object detection in the future. The computational results from YOLOv3 demonstrate a great potential for improvement in video object detection when the information given to the neural net is expanded past individual frames. As discussed above, YOLO-LSTM makes an interesting combination of YOLO-like and human-like mistakes. It may be worthwhile in the future to perform a deeper analysis of the types of errors made by humans and YOLO-LSTM to see what conditions lead it to output more human-like versus YOLO-like results.

Future improvements could also include integrating the knowledge we have of the development of human object permanence and the visual system into how we develop training and testing datasets, how we train the models, and how the different components of the models interact with each other. This is a challenge however, because unlike other human visual processes, object permanence is not fully understood. This model, as well as other object detection models with recurrent elements [9][10], demonstrates that recurrent connections may be part of the answer.

ACKNOWLEDGMENT

We thank Dr. Pawan Sinha, Hunter Oren King, and Nhat Minh Le for their valuable support, guidance, and encouragement throughout this project.

REFERENCES

[1] Baillargeon, Renée, and Julie DeVos. "Object Permanence in Young Infants: Further Evidence." Child Development, vol. 62, no. 6, 1991, pp. 1227–1246. JSTOR, www.jstor.org/stable/1130803

[2] Baird, A. Kagan, Jerome Gaudette, Thomas Walz, Kathryn Hershlag, Natalie Boas, David. (2002). "Frontal lobe activation during object permanence: Data from near-infrared spectroscopy." NeuroImage. 16. 1120-5. 10.1006/nimg.2002.1170.

[3] Pan, J., Hu, B. (2007). Robust Occlusion Handling in Object Tracking. 2007 IEEE Conference on Computer Vision and Pattern Recognition.

[4] Shantaiya, S., Verma, K., Mehta, K. (2015). Multiple Object Tracking using Kalman Filter and Optical Flow. European Journal of Advances in Engineering and Technology, 34–39.

[5] Pan, J., Hu, B. (2007). Robust Occlusion Handling in Object Tracking. 2007 IEEE Conference on Computer Vision and Pattern Recognition. doi: 10.1109/cvpr.2007.383453

[6] Y How to make self-driving cars safer on roads. (2019, March 28). Retrieved from https://www.sciencedaily.com/releases/2019/03/190328154209.htm

[7] Image Module — Pillow (PIL Fork) 7.1.2 documentation https://pillow.readthedocs.io/en/stable/reference/Image.html

[8] ultralytics/yolov3: YOLOv3 in PyTorch https://github.com/ultralytics/yolov3

[9] Ning, G., Zhang, Z., Huang, C., Ren, X., Wang, H., Cai, C., He, Z. (2017, May). Spatially supervised recurrent convolutional neural networks for visual object tracking. In 2017 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1-4). IEEE.

[10] Yun, S., Kim, S. (2019, October). Recurrent YOLO and LSTM-based IR single pedestrian tracking. In 2019 19th International Conference on Control, Automation and Systems (ICCAS) (pp. 94-96). IEEE.