A Perceptual Front-End for Probability Learning: Object Detection with YOLO

Zhe Fan[1,2], **Zilong Wang**[3], **Thomas R. Shultz**[2,4]
zhe.fan@mail.mcgill.ca, zilong.wang@mail.mcgill.ca, thomas.shultz@mcgill.ca
[1]Department of Neuroscience, McGill University
[2]Department of Computer Science, McGill University
[3]Integrated Program in Neuroscience, McGill University
[4]Department of Psychology, McGill University
Montreal, QC, H3A 1G1 CANADA

Abstract

Neural Probabilistic Learner and Sampler (NPLS) is an algorithm that has simulated children's non-symbolic probability learning from visual stimuli such as collections of different colors of marbles. Although NPLS closely simulates the cognitive process of probability learning, the training of such learning algorithms often uses binary encoding of inputs that represent the perceived visual stimuli, avoiding simulation of the visual perception of the stimuli. Here, the computer vision technique You Only Look Once (YOLO) (Jocher et al., 2021; Redmon et al., 2016), is integrated into the workflow of an NPLS simulation probability learning experiments with children. YOLO is a convolutional neural network (CNN) designed to detect objects. The model's performance on marble datasets is tested through an analysis of precision and recall. Results indicate that the YOLO model, when trained sufficiently, outputs predictions on marble image datasets with high accuracy and precision. We also analyze YOLO's suitability as a biologically plausible model of visual processing, interfering with YOLO's training process by shortening the training time to examine the effects of perceptual errors on simulated probabilistic reasoning.

Keywords: Applied Machine Learning; Computer Vision; Probabilistic Learning; Child Development; Cognitive Science; Neural Network; Simulation;

Introduction

Probability learning is an integral aspect of life as humans and animals rely on probabilistic estimations to make decisions for survival and other needs. Recent work by Shultz and Nobandegani (2022) on the Neural Probability Learner and Sampler algorithm (NPLS) showed that probability learning can occur even in infants without the explicit use of counting and dividing. Shultz and Nobandegani's (2022) simulations replicated the empirical results of Denison and Xu's (2014) infant probability reasoning task. During the task, infants were presented with two jars containing different proportions of fake lollipops of two colors, one of which is preferred over the other by the infants. The NPLS network learned the underlying probability distributions by adjusting its connection weights to reduce the sum of squared error (Shultz & Nobandegani, 2022).

However, one of the limitations of the NPLS network is the requirement of a coding scheme to represent the image stimuli (Shultz & Nobandegani, 2022). This coding scheme (Table 1) accurately extracts the probability distributions of the favored objects shown to the infants. Thus, the NPLS network simulates the cognitive aspect of probability learning, but the perceptual component is still missing – specifically, what drives the conversion of viewed images into these numerically coded representations?

Table 1 depicts the coding scheme of a binary probability distribution above, serving as the input for NPLS, representing two jars of lollipops where the "Input" value denotes jar 1 vs. jar 2, and the target outputs are arranged such that jar 1 contains a 3:1 ratio of preferred to unpreferred types, and jar 2 contains a 1:3 ratio of preferred to unpreferred types.

Table 1: The Coding Scheme of a Binary Probability Distribution as Input for NPLS

3:1		1:3	
Input	Output	Input	Output
1	1	2	1
1	1	2	0
1	1	2	0
1	0	2	0

A computational model of visual processing may provide a more comprehensive, biologically plausible simulation of perception and cognition for probability reasoning. Furthermore, using a computational model to convert raw image stimuli into learnable codes for NPLS would aid the simulation of a wide array of empirical experiments involving probability reasoning, particularly those that use collections of physical objects, such as marbles and dot patterns.

A convolutional neural network (CNN) could simulate the perceptual process of converting pictures or objects into the frequency ratios required by NPLS. Inspired by visual cortex neurons such as simple and complex cells (Hubel & Wiesel, 1962), CNNs take advantage of the concept of receptive fields (Hubel & Wiesel, 1962; Lindsay, 2021; Luo et al., 2016) and feature-extraction using convolutional kernels (Bogdan et al., 2019) to make inferences from visual stimuli. Among various types of CNNs, the YOLO (You Only Look Once) model is intuitively appealing compared to some of its competitors, such as Regional CNN (R-CNN) or sliding window approaches (Felzenszwalb et al., 2009; Girshick, 2015), due to YOLO's ability to see the entire image at once without dividing a single image into multiple sub-problems when detecting objects (Redmon et al., 2016). Given the efficiency of human vision and the limited amount of time for visual saccades dur-

In M. Goldwater, F. K. Anggoro, B. K. Hayes, & D. C. Ong (Eds.), *Proceedings of the 45th Annual Conference of the Cognitive Science Society.*

ing psychology experiments, we suspect the YOLO model's ability to parse through an entire scene may capture the visual perception of subjects in psychological experiments. We here determine whether YOLO could provide an accurate perceptual front end to the numerical inputs of NPLS.

O'Grady and Xu's (2020) probabilistic reasoning experiments involve children choosing from two collections of marbles the one that contains a higher proportion of the target color. YOLO's performance is evaluated in a machine-learning context of recall and precision, to ensure its reliability. Also, an interesting hypothesis about the concept of perceptual error can be investigated: would errors made during the perceptual process influence probabilistic reasoning?

Methods

We here determine whether YOLO could provide an accurate perceptual front end to the numerical inputs of NPLS.

NPLS

An NPLS network does both learning and sampling. The learning portion uses sibling-descent cascade-correlation (SDCC), which has simulated many cognitive phenomena. During the training process, the SDCC performs constructive learning: starting as a two-layer network and gradually recruiting new hidden units one at a time as needed to reduce sum of squared error (SSE) (Baluja & Fahlman, 1994), computed as

$$E = \sum_o \sum_p (A_{op} - T_{op})^2 \tag{1}$$

where A_{op} represents the prediction made by the SDCC network for output o and pattern p, and T_{op} represents the ground-truth target for output o and pattern p.

As SDCC learns, it recruits new neurons into the network as needed to reduce error, so the network topology is not designed by the programmer but is constructed by the learning algorithm (Baluja & Fahlman, 1994). The learning is split into input and output phases. During the output phase, the network aims to reduce E. During the input phase, the weights to candidate hidden units are trained to increase the covariance C:

$$C = \frac{\sum_o |\sum_p (h_p - \langle h \rangle)(e_{op} - \langle e_o \rangle)|}{\sum_o \sum_p (e_{op} - \langle e_o \rangle)^2} \tag{2}$$

where C is the covariance between candidate-hidden-unit activation and network error, h_p represents the candidate hidden unit activation from pattern p, $\langle h \rangle$ represents the mean candidate hidden unit activation, e_{op} represents the residual error at output o for pattern p, and e_o represents the mean residual error for all input training patterns. The highest correlating unit is then installed on the highest layer of hidden units, or on its own layer, depending on which results in a better correlation (Baluja & Fahlman, 1994).

Finally, output unit activations are computed with an asymmetric sigmoid function to create values between 0 and 1 (Baluja & Fahlman, 1994; Shultz & Nobandegani, 2022)

The SDCC model was modified for the purpose of learning probability distributions. SDCC's determinism would cause it to recruit new hidden units non-stop as it is never satisfied with the higher error of probabilistic outcomes (Shultz & Nobandegani, 2022). To circumvent this, SDCC is allowed to keep track of its own error reduction during training, and parameters for threshold and patience can be applied to control the learning process. During the output phase of SDCC, error reduction based on the error function is continued until the error stagnates. Then, SDCC switches to the input phase by recruiting a new hidden unit, and weights are again adjusted according to (2). Stagnation of error is characterized by the absence of progress greater than the threshold for the number of training epochs defined by the patience (Shultz & Nobandegani, 2022). An additional loop with its own set of thresholds and patience is introduced to monitor the learning cycles of SDCC, where each cycle contains an input phase and the output phase that follows (Shultz & Doty, 2014). These modifications allow the NPLS to learn unnormalized, multivariate probability distributions from examples that indicate the occurrence of any outputs in the presence of a particular input (Kharratzadeh & Shultz, 2016). Lastly, the sampling process of NPLS is characterized by a Markov Chain Monte Carlo (MCMC) sampling algorithm, which can simulate experiments related to probabilistic reasoning and decision-making in humans (Dasgupta et al., 2017; Shultz & Nobandegani, 2022).

YOLO

YOLO, designed originally by Redmon et al. (2016), is a supervised object detection algorithm that is able to predict bounding boxes and class probabilities directly from images in one evaluation. In object detection, the bounding box is a box responsible for detecting an object in an image, characterized by its center coordinate, width, and height. The class probability is a variable associated with the bounding box, responsible for describing the probability that this object is an instance of a particular class (Redmon et al., 2016). In YOLO, images are divided into an S by S grid, such that each grid is able to predict B bounding boxes. Each bounding box is associated with a confidence score for whether it contains an object or not. Each bounding box is also associated with a class probability vector C, which contains the probability scores for each candidate label. The predictions are encoded by an S by S by (B*5 + C) tensor. Furthermore, the YOLO network is trained using a sum-squared-error function to minimize the distance error between the predicted bounding box's location and class and the ground truth's bounding box's location and class (Redmon et al., 2016). See Figure 1 for an example.

The training process of YOLO is similar to how a basic CNN is trained. As with any supervised learning algorithm, the YOLO network is trained through the backpropagation of sum-squared error from the target bounding box variables vs. the predicted variables (Redmon et al., 2016). Neurons in a convolutional layer are mapped from one neuron to the next

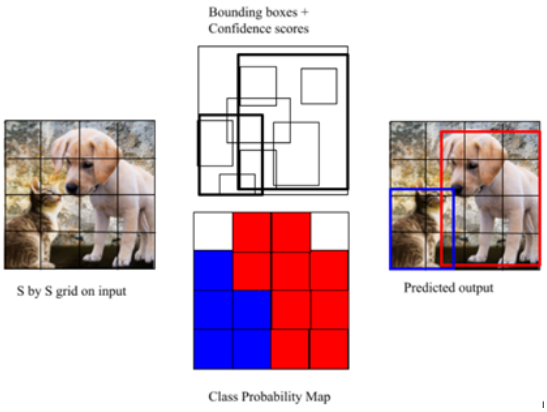

Figure 1: YOLO prediction. Left: an input image is contained in an S by S grid (left). Top: for each grid, the bounding box described that grid, and the object confidence score is recorded. The higher the confidence, the thicker the illustrated boundary. (bottom): the class probability score of each grid, each color represents the most probable class contained in that grid. (right): by combining the top and the bottom figures, an overall detection result can be generated by choosing the bounding boxes with the highest confidence score from grids that have the highest-class probability. By dividing an image into small grids, we can encode additional information within each grid. One could imagine each grid to contain an array of information: the number of bounding boxes, the position and size of each bounding box, the confidence score associated with each bounding box, and class probability scores for the list of possible classes (cat and dog).

neuron in the subsequent layer through convolutions using a weight-containing kernel. In YOLO, the neuronal activities of non-output layers are activated with a leaky Rectified Linear Unit (ReLU) function to introduce non-linearity. The output neuronal activities use a linear activation function. During the training process, the objective function that YOLO is trying to optimize is a sum-squared error (SSE) function to minimize the discrepancy between the predicted bounding box's locations/classifications vs. the target bounding box's locations/classifications. (Redmon et al., 2016).

Experimental Design to Simulate O'Grady and Xu's (2019) Dataset

O'Grady and Xu's (2020) experiments examined children's behavior when presented with an image containing two collections of marbles on two trays. One of the trays contained a higher proportion of favorable or target marbles, the color of which the children have learned to prioritize beforehand (the red marbles in this case). For example, if presented with two collections of marbles, the winning collection contains 80% red and 20% white, and the losing collection contains 60% red and 40% white, then the correct choice for the children to pick would be the former collection. A metric that describes the proportion of marbles present in an image is the RoR (ratio of ratios), which is given by the proportion of target marbles in the "winning" collection divided by the proportion of target marbles in the "losing" collection. While in the original paper O'Grady and Xu (2020) had two major sections of experiments, here we only considerred the second section where there are three different types of experiments conducted on children of ages 8, 10, and 12.

In the total equal trials (experiment 1), both the winning and losing collections had the same number of marbles. In the number vs. proportion trials (experiment 2), the winning collection had a higher proportion of targets, but the losing collection had more targets than the winning group. In the size anti-correlated trials (experiment 3), the winning collection had a higher proportion of targets, but the losing collection contained targets with larger sizes than the non-targets. For each experimental setup, the goal is to record the number of correct choices of the favorable collection vs. the total number of choosing attempts by subjects of a specific age category when presented with two marble collections that are characterized by a particular RoR value.

The empirical results showed that, when the RoR of an image was greater, children tended to select the more favorable collection (the one with more red marbles than white marbles) with a higher probability in all three experimental trials. However, on average, children performed better during experiment 1 than experiments 2 and 3, regardless of their age. This may be due to the lack of confounding factors such as frequency or size in experiment 1 that could result in heuristic shortcuts that are not based on proportional reasoning (O'Grady & Xu, 2020). On average, children of older ages achieved higher winning group selection probability than children of younger ages. Another important characteristic of the results is that the probability of selecting the winning group increased linearly with respect to the log of RoR. This suggests that Weber's law is involved during the decision-making process (O'Grady & Xu, 2020).

To train a YOLO network, the Yolov5 repository by Jocher et al. (2021) is used. The marble image dataset is divided into a training set, a validation set, and a testing set. The training set contains 54 randomly selected images from a total set of 263 images used for the experiments of O'Grady and Xu (2020). Image labeling is performed on the red marbles and white marbles in the images, ensuring the labeled bounding boxes properly enclose each marble.

The following image augmentation steps are performed to improve the generalizability of the training set: (a) split images into 2 equivalent halves (because each original image contained two collections placed side by side, this multiplies the total number of images by two), (b) alter the hue of the image by 26 degrees in both directions (this multiplies the total number of images by three), (c) blur the images by up to 10 pixels, and (d) add random noise pixels up to 4% of the image's pixels. The purpose of such preprocessing steps is to prevent overfitting on a small sample of datasets. The augmented training set now contains up to 324 images (see

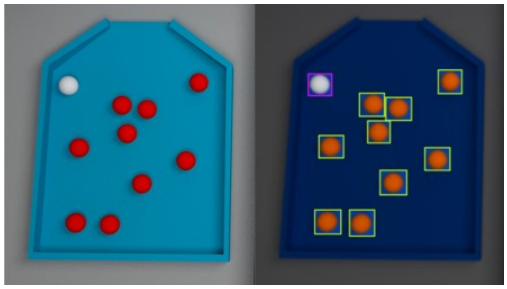

Figure 2: Example of a labeled training image, applied with image modifications for data augmentation. Images were obtained from O'Grady and Xu (2020).

Figure 2 for an example of image labeling).

The YOLO model selected, YOLOv5s by Jocher et al. (2021), is a relatively lightweight and efficient YOLO model suitable for the training of small datasets using the K80 GPU provided by Google Collaboratory. Training on the 324 images is performed using the following parameters: grid size = 540 pixels, number of epochs = 150, batch size = 16, and the initialized yolov5s.pt lower-layer weights trained on the COCO dataset (Common Objects in Context) which contains a list of everyday objects (Lin et al., 2014).

Training the YOLO model with the custom dataset images serves two purposes: to introduce the concept of red and white marbles to the model, such that it knows to label objects with these classes, and to ensure a higher-than-chance performance on the classification of marbles. Furthermore, analyses on training performance are evaluated through a precision/recall analysis as well as the calculation of the mAP (mean average precision) for object detection models.

The trained YOLO model is used to perform detections on all 324 images used in the experiment, and the frequencies of appearance of each type of marble in the collection of marbles are recorded. A Python script is written to convert these frequencies into inputs for the NPLS workflow and the training outputs of NPLS are used for MCMC sampling. The results of the MCMC sampling, which is quantified by the mean-selection probability for the target group vs. the ratio of ratio (RoR), are compared against the empirical data via correlational studies. The score threshold parameter (ST) of the NPLS model is set to correspond with the cognitive maturity of children of different ages. The lower the ST parameter is, the lower the training error must be before the NPLS terminates its training process. A score of 0.51 corresponds with children of age 8, a score of 0.50 corresponds with children of age 10, and a score of 0.49 corresponds with children of age 12 (Wang et al., 2023).

The performance of YOLO + NPLS on the marble datasets is evaluated for experiment 1 and experiment 3 performed by O'Grady and Xu (2020). The omission of experiment 2 is based on the argument that if the results of experiment 1 show a good correlation with empirical data, then the YOLO model should be effective at performing accurate predictions on same-sized marbles and there would be no need to examine the effects of perception experiment 2, which also use same sized marbles. However, because experiment 3 contained different-sized marbles, then it would be interesting to see if YOLO would perceive, for example, larger marbles with higher confidence scores, which may inadvertently imply the existence of a heuristic shortcut in decision-making. Additionally, an "early stopping" YOLO model trained for only five epochs is also tested to see the effects of perceptual error on probability reasoning.

Results

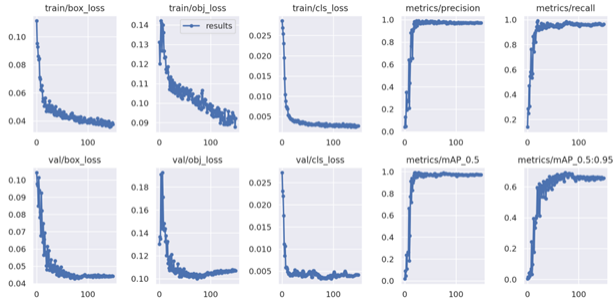

Figure 3: Results of full YOLO training over 150 epochs. From the top to the bottom row, left to right: 1. The error of the predicted bounding box coordinates/size over epochs. 2. The error of the predicted presence of an object vs. the ground truth over epochs. 3. The error of the predicted class probability over epochs. 4. The precision of the model over epochs. 5. The recall of the model over epochs. (6,7,8). The same metrics as (1,2,3) but on a small validation set of 38 images. 9. The mean Average Precision (mAP) over epochs at a confidence threshold of 0.5. 10. The average mAP over epochs for all confidence score thresholds between 0.5 to 0.95, incremented by 0.05 between every level.

The training results (Fig. 3) demonstrate a successful reduction of sum-squared-loss after 150 epochs, as the error reduction stagnates over time. The precision and recall scores converge to 1.0 at the end of the training, which indicates that by the end of the training process, the model confidently detects all present marbles correctly and reliably. The mAP scores at the confidence threshold = 0.5 converge to 1, which suggests that if positive detections are only made for objects predicted with a higher than 50% confidence, then the model predicts objects with perfect precision and recall.

The detection process is run with the experimental images of O'Grady and Xu (2020). For every image, the YOLO model outputs a file containing the bounding boxes and classifications for each prediction made on the image. The predictions are counted and transformed into frequency patterns required for NPLS training (e.g., 55:45, 90:10).

To examine the effects of errors in the perceptual stage, an "early-stopping" model of YOLO is introduced. This model

uses the exact same parameters as before but stops training prematurely after only five epochs, causing some perceptual errors. An examination of the model's training metrics (Fig 4) shows that the model is only about 73% accurate (shown by a mAP@0.5 score of 0.73) However, the probability reasoning simulation of the NPLS using outputs from this YOLO model still shows moderate correlation (see Table 3) with the empirical data. This suggests that perceptual errors made by YOLO do not actually affect the proportion of observed marble colors, but rather just the number of marbles seen.

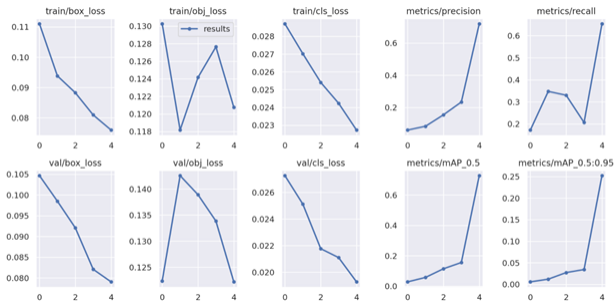

Figure 4: Training results metrics of YOLO after only five epochs. See equations 3, 4, 5 for an explanation of these metrics.

To examine the growth of "winning" group selection probability as RoR increases, linear regression of selection probability vs. log(RoR) is conducted on all of the simulated trials, shown in Table 2.

Table 2: Linear regression of log(RoR) for all of the simulation trials. Column 1: type of trial; column 2: R^2 adjusted score; column 3: p-value for R^2 adjusted score.

Simulation Type *	Adjusted R^2 Score	p <
1 (Age 8)	0.962	0.0001
1 (Age 10)	0.731	0.0001
1 (Age 12)	0.639	0.0001
2 (Age 8)	0.789	0.0001
2 (Age 10)	0.537	0.0001
2 (Age 12)	0.463	0.001
3 (Age 8)	0.985	0.0001
3 (Age 10)	0.970	0.001
3 (Age 12)	0.946	0.0001

To compare the simulated results with the empirical data, the correlation between the "winning" group selection rate of the empirical data and that of the simulation is calculated for each simulated trial (Table 3).

Discussion

Overall, the YOLO model performs highly accurate detections after sufficient rounds of training on the marble dataset by O'Grady and Xu (2020). After 150 epochs of training on about a fifth of the data, the model produces frequencies of target and non-target marbles in a matter of seconds. From a utilitarian perspective, the use of perceptual models such as YOLO greatly increases the efficiency of data processing for simulation experiments and extends the models to cover realistic perception of visual stimuli.

Table 3: Pearson correlation between the empirical data (O'Grady & Xu, 2020) and the simulated winning group selection rates from YOLO and NPLS.

Simulation Type *	Pearson r	p <(N = 22)
1 (Age 8)	0.721	0.001
1 (Age 10)	0.717	0.001
1 (Age 12)	0.842	0.00001
2 (Age 8)	0.555	0.01
2 (Age 10)	0.477	0.05
2 (Age 12)	0.513	0.05
3 (Age 8)	0.688	0.001
3 (Age 10)	0.864	0.00001
3 (Age 12)	0.872	0.00001

*Notes for Table 2 and Table 3: the simulation type numbers correspond to the empirical experiment the YOLO + NPLS model is trying to simulate as well as the specific parameter set-ups for the model:
1 - Total Equal (Experiment 1) + Fully Trained YOLO (150 epochs) predictions with confidence score ≥ 0.5 ;
2 - Total Equal (Experiment 1) + "Early Stop" Trained on YOLO (5 Epochs) predictions with confidence score ≥ 0.5 ;
3 - Size Anti-Correlated (Experiment 3) + Fully Trained YOLO predictions with confidence score ≥ 0.5;
The age label (in brackets) by the simulation type numbers signifies the age of children that generated the empirical data during the psychological experiment. The score threshold (ST) of the NPLS model is selected according to the age of the children it is trying to simulate.

One of the more interesting aspects of YOLO is its application in experiment 3 of O'Grady and Xu's (2020) study. The original coding technique used to generate inputs to the NPLS cannot capture the effect of the size of certain marbles. Fortunately, a detection algorithm like YOLO generates probability scores for each predicted marble, and this may be used as a metric to imply the amount of focus a subject may place on targets of varying sizes. It is particularly interesting to observe that larger marbles are typically assigned a higher confidence score than smaller marbles. The explanation for this phenomenon in a machine learning context may be that smaller objects are more prone to noises because they take up a bigger portion of pixels of smaller objects than they do for larger objects. Smaller bounding boxes are associated with distance losses that are less prioritized in YOLO's loss func-

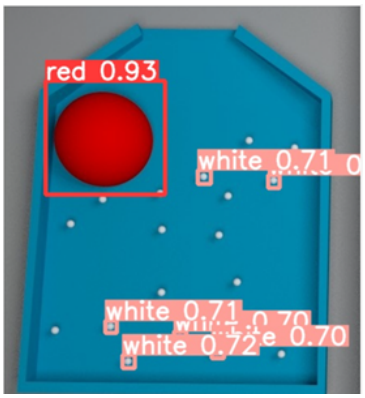

Figure 5: Confidence scores associated with larger objects are typically greater than those associated with smaller objects.

tion optimization, and their features are more difficult to extract by the CNN model due to their small size. To propagate the effect of marble size using YOLO, the confidence score threshold must be carefully chosen such that it can effectively separate predictions of lower confidence from those that are of higher confidence.

Psychologically, large objects tend to capture human attention in visual search (Proulx, 2010). The higher confidence score assigned to larger objects in YOLO suggests that it may simulate this aspect of human vision, as shown by the example in Figure 5. The simulation of experiment 3 shows that a perception algorithm such as YOLO has the potential to simulate attentional aspects of human perception, which can be explored further. Future studies on YOLO could be employed to see if it could simulate realistic human performance and reproduce the same level and type of error that humans could make.

Lastly, linear regression of the winning group selection probability plotted against log(RoR) shows that with a fully trained YOLO model, the selection probability of NPLS increases linearly with respect to log(ROR). This is consistent with the empirical data, which suggests that children would typically follow Weber's law when determining the winning group (O'Grady & Xu, 2020). However, when using the early-stopped YOLO model, this effect was not as apparent, as shown by the relatively lower adjusted R^2 score compared to the fully trained model (Table 2). This suggests that further work must be done to determine an optimal level of training error permitted by YOLO that still adheres to Weber's Law. Interestingly, the simulations of experiment 3 using a fully trained YOLO model produced good adjusted R^2 scores, suggesting that the model is indeed adhering to Weber's law when presented with area anti-correlated collections of marbles and using a confidence threshold that differentiates the sizes of the marbles. Furthermore, the correlation between the simulated and empirical results (Table 3) shows that the simulated probability of winning group selection using YOLO and NPLS correlated strongly with the empirical data for both experiment 1 and experiment 3 when the model is fully trained. When the model's training is stopped early, the simulated results only moderately correlated with the empirical data, suggesting that the perceptual errors due to the early-stopping training of YOLO alone may not be sufficient to characterize the kinds of perceptual errors a human subject would make.

There have been debates about whether machine learning, specifically supervised learning algorithms such as CNN, serves as a good model of biological visual processing (Whittington & Bogacz, 2019). The CNN is inspired by the visual system, borrowing from the concept of the isolated, binary simple cells and the integrative, distributed complex cells (Hubel & Wiesel, 1962; Lindsay, 2021). However, the biological plausibility of backpropagation has been long disputed as it requires the propagation of a global error signal based on the target to the earlier layers of a neural network (Whittington & Bogacz, 2019). Such engineering seemed implausible to the model of Hebbian plasticity, which describes learning as a localized process between neighboring co-active neurons (Hebb, 1949; Whittington & Bogacz, 2019). Other visual processing models, such as the Boltzmann Machine (Hinton, 2007) or the neocognitron (Fukushima & Miyake, 1982), can process and recognize visual stimuli via representation learning, which is unsupervised and does not rely on backpropagation. However, the models focus on the extraction of representations or patterns that are useful for recognition but can be difficult to implement for detection. More extensive analysis of these models should be done before applying them to simulations of certain cognitive phenomena. Furthermore, a potential limitation of all object detection models could be the fact that it enumerates all individual objects during visual processing, which may not be the case for human subjects with limited time to observe the images.

Our work shows that YOLO shows promise as a perceptual front-end for cognitive simulations of probabilistic learning in developing children using NPLS. There are some interesting directions of research that could be studied with YOLO and NPLS. For example, more research could be done to determine the ideal confidence-score threshold for YOLO to study the effect of the physical features of an object on a human observer's ability to detect it.

References

Baluja, S., & Fahlman, S. E. (1994). Reducing network depth in the cascade-correlation learning architecture.

Bogdan, V., Bonchiş, C., & Orhei, C. (2019). Custom extended sobel filters. *arXiv preprint arXiv:1910.00138*.

Dasgupta, I., Schulz, E., & Gershman, S. J. (2017). Where do hypotheses come from? *Cognitive psychology*, *96*, 1–25.

Denison, S., & Xu, F. (2014). The origins of probabilistic inference in human infants. *Cognition*, *130*(3), 335–347.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2009). Object detection with discriminatively

trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, *32*(9), 1627–1645.

Fukushima, K., & Miyake, S. (1982). Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets: Proceedings of the us-japan joint seminar held at kyoto, japan february 15–19, 1982* (pp. 267–285).

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the ieee international conference on computer vision* (pp. 1440–1448).

Hebb, D. O. (1949). The first stage of perception: growth of the assembly. *The Organization of Behavior*, *4*(60), 78–60.

Hinton, G. E. (2007). Boltzmann machine. *Scholarpedia*, *2*(5), 1668.

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, *160*(1), 106.

Jocher, G., Stoken, A., Chaurasia, A., Borovec, J., Xie, T., Kwon, Y., … others (2021). ultralytics/yolov5: v6. 0-yolov5n'nano'models. *Roboflow integration, TensorFlow export, OpenCV DNN support*, *10*.

Kharratzadeh, M., & Shultz, T. (2016). Neural implementation of probabilistic models of cognition. *Cognitive Systems Research*, *40*, 99–113.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., … Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer vision–eccv 2014: 13th european conference, zurich, switzerland, september 6-12, 2014, proceedings, part v 13* (pp. 740–755).

Lindsay, G. W. (2021). Convolutional neural networks as a model of the visual system: Past, present, and future. *Journal of cognitive neuroscience*, *33*(10), 2017–2031.

Luo, W., Li, Y., Urtasun, R., & Zemel, R. (2016). Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, *29*.

O'Grady, S., & Xu, F. (2020). The development of non-symbolic probability judgments in children. *Child Development*, *91*(3), 784–798.

Proulx, M. J. (2010). Size matters: large objects capture attention in visual search. *PloS one*, *5*(12), e15293.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 779–788).

Shultz, T., & Doty, E. (2014). Knowing when to quit on unlearnable problems: another step towards autonomous learning. In *Computational models of cognitive processes* (pp. 211–221).

Shultz, T., & Nobandegani, A. (2022). A computational model of infant learning and reasoning with probabilities. *Psychological Review, 129(6)*, 1281-1295.

Wang, Z., Shultz, T., & Nobandegani, A. (2023). *A computational model of children's learning and use of probabilities across different ages.* arXiv q-bio.NC. Retrieved from https://arxiv.org/abs/2305.04128

Whittington, J. C., & Bogacz, R. (2019). Theories of error back-propagation in the brain. *Trends in cognitive sciences*, *23*(3), 235–250.