Analyzing Co-Occurring Bias in YOLOv5

Guanyi Cao, Tara Shukla, Molly Taylor

Introduction

Object detection models play a critical role in a wide range of applications. For many models, visual context—the elements surrounding an object of interest in an image—helps in object recognition. However, a reliance on visual context induces the issue of contextual bias. As models become dependent on familiar co-occurrences or environments, their ability to generalize decreases. For instance, a model might consistently detect a primary object with its co-occurring class and struggle to identify the primary object in the absence of this co-occurring object. This correlation leads to skewed predictions, and can also cause the "hallucination" of objects in contexts where they are typically found but are actually absent [1]. Such biases undermine model reliability and limit their applicability in real-world, diverse environments. In particular, contextual bias can lead a model to make harmful associations concerning race, class, and gender.

This introduces the specific issue of **co-occurring bias,** referring to biases that arise from the frequent pairing of specific objects of interest. A model trained on commonly co-occurring object pairs may exhibit significant performance degradation when these pairings are altered or absent. Addressing co-occurring bias is essential for developing robust object detection systems capable of handling varied and unexpected situations.

The focus of this project is to build upon the work by Singla et al. (2020) in "Don't Judge an Object by Its Context: Learning to Overcome Contextual Bias," by analyzing co-occurring bias in the YOLOv5 object detection model, leveraging images from the COCO and Open Images datasets. Through investigating model performance on commonly co-occurring class pairs identified in the paper and calculating bias, our ultimate goal is to expand upon the paper by identifying new co-occurring pairs.

Related Work

Primary Extension

Our work primarily extends the 2020 paper by Singh et al., "Don't Judge an Object by Its Context: Learning to Overcome Contextual Bias", on contextual bias in object classification models [1]. The authors defined bias as the extent to which a (biasing) object's co-occurrence with another (biased) object, impacts the probability of correctly classifying that biased object. They use the definition:

$$bias(b,z) = \frac{\frac{1}{|I_b \cap I_z|} \sum_{I \in I_b \cap I_z} \hat{p}(i,b)}{\frac{1}{|I_b \setminus I_z|} \sum_{I \in I_b \setminus I_z} \hat{p}(i,b)} \quad (1)$$

Where *b* is the biased class and *z* is its co-occurring class. The paper then identifies biasing objects for each class via arg-max over the bias calculation for each co-occurring object. As an example: If ‘skateboard’ is paired with ‘person’ as its biasing class, then the bias of ‘person’ on ‘skateboard’ would be the ratio between the average prediction probabilities of ‘skateboard’ when it occurs with and without ‘person.’ A higher bias indicates a greater impact of ‘person’ on the model’s ability to predict ‘skateboard.’ Thus, with their biasing object pairs defined, Singla et al. go on to implement two complementary approaches to reduce contextual bias in their model. Their first method used Class Activation Maps (CAMs) as weak localization signals to minimize the spatial overlap between co-occurring objects; the second, split the feature space into two subspaces - one dedicated to representing objects in isolation and another for capturing joint object-context relationships. Both approaches proved to be effective debiasing efforts. Overall, the work addresses the importance of addressing contextual bias in computer vision tasks, and points towards feature decorrelation as one such strategy. As will be discussed in later sections, our main extension of the work is to extend their object bias pair identification, for the purpose of identifying contextual biases in object detection. In addition, while Singh et al. evaluate bias in the context of object classification, we analyze bias for the task of object detection. As object detection focuses on local regions of an image, we consider whether a model that leverages locality and bounding boxes may be better suited to distinguish an object from its context.

Other Related Work

There is an existing field of research exploring the presence and effects of contextual bias in object recognition tasks. First, many scholars have studied biases in object classification and

detection datasets. In a 2015 research study, Model and Shamir explore dataset bias by testing model accuracy on image subsets dominated by context, with no identifying object information: they find that this is a viable method of bias identification in classification [2]. A study by Agarwal, et. al, note the impact of contextual bias caused by dataset co-occurrence, and study strategies for debiasing models via training data curation [3]. This research underscores how object detection models can develop systemic biases by learning to rely on contextual cues rather than intrinsic object features due to training set biases.

Additionally, scholars have studied contextual bias in model performance. Zhang, et.al note that many state of the art recognition models rely on contextual reasoning about context [4]. Thus, these models often fail to detect out of context models. The authors investigate how context impacts object recognition in humans vs in computers; they conducted human psychophysics experiments to measure the role of context in human visual recognition. Then, they proposed a context-aware attention network which leverages local and global context. Zhang et. al's work finds that quantity, quality, and dynamics of context help boost models towards human-level in context recognition; their work contributes to the field's understanding of the dependencies of object recognition tasks, on context.

A 2022 study by Liu, et. al expands on this understanding by exploring how this dependency can foster contextual biases. They present a structural causal model to define causal relationships between object representations, context, and predictions [5]. The authors then design a debiasing module to mitigate the effects of this causal relationship. Other works have explored contextual bias in object detection tasks. In particular, Son and Kusari investigate how background features influence detection performance during domain transfer; they propose new metrics to measure

context bias and demonstrate that aligning foreground features is insufficient for effective domain adaptation [6].

Furthermore, Dreyer et.al create a visualization of neural networks' decision making in object detection, and demonstrate how their method can reveal biases where models rely on background context features [7]. Overall, previous work evaluating contextual bias has studied co-occurrence bias in object recognition tasks, and background bias in object detection tasks; our work seeks to extend the existing field of research by identifying co-occurrence bias in object detection.

System Design and Implementation

Using the biased class pairs identified by Singh et al., we create subsets of the COCO validation and Open Images test set to evaluate the YOLOv5 model on these class pairs. With our validation results, we calculate the model's bias toward detecting the primary object in the presence and absence of its co-occurring class. Then, we identified co-occurring pairs within the COCO training set and evaluated the model's bias on these pairs to assess the importance of co-occurrence in learning bias.

Datasets

We utilized two datasets for this project: COCO and Open Images.

COCO (Common Objects in Context) dataset is a large-scale dataset utilized for object detection, segmentation, and captioning [8]. Containing 80 classes, it is commonly used in computer vision for benchmarking and is designed to suit a variety of tasks and object categories. It is split into three subsets: Train2017, Val2017, and Test2017. For our project, we utilized the Val2017 subset

for evaluation, as the test set is not annotated. Since the Val2017 subset contained only 5k images, we also utilized the Open Images dataset.

Open Images is an extensive dataset from Google that is also frequently used in computer vision research. Its robust collection of images are annotated with image-level labels, object bounding boxes, object segmentation masks, visual relationships, and localized narratives [9]. In contrast to COCO, Open Images contains 600 object classes with hierarchical relationships. Despite the difference in labeling schemes, we found Open Images to be largely compatible with the COCO annotations and, with nearly 120,000 images, it was useful in achieving a thorough evaluation of the YOLOv5 model.

1. Data Processing

For the Open Images dataset to be compatible with a YOLOv5 model trained on COCO, we needed to convert the format of its annotations. We used the FiftyOne library to download the entire Open Images test set in YOLO format. Then, we had to convert Open Image class labels to COCO class labels by 1.) removing labels for classes outside of COCO and 2.) replacing the default class IDs to COCO class IDs. For both the Open Image and COCO evaluation sets, we created folders of images and labels depicting only the primary object of each co-occurring class pair, and then the primary object with its co-occurring class. For example, we created a folder of COCO validation images featuring a skateboard with a person and a separate folder of images featuring a skateboard without a person. We wrote a script that generates a yaml file allowing us to run validation on each folder.

2. Validating Model and Retrieving Performance Metrics

To obtain the model performance, we ran validation on the model across both datasets: the COCO validation set and Open Images test set.

As part of the YOLOv5 library, this outputted some performance metrics, such as precision and recall maps and a confusion matrix, both showcasing the model's performance across all classes. It also outputs a JSON file that contains details about each image, such as its image id, category (class), bounding box coordinates and score.

Through these performance metrics, we were able to analyze areas of higher bias or consistent bias. The data we obtained enabled us to confirm that the YOLOv5 model is indeed impacted by co-occurring bias.

3. Calculating Bias

$$precision\ bias(b,z) = \frac{\frac{1}{|I_b \cap I_z|} P(i,b)}{\frac{1}{|I_b \setminus I_z|} P(i,b)} \quad (2)$$

Based on our observations of the metrics we obtained in validation, we derived a new formula to calculate bias across precision, recall, mean average precision (mAP) of the 0.5 intersection of union (IoU) threshold, and mAP of 0.95 IoU.

Our formula is defined in Equation 2, as follows: given a biased class *b* and its co-occurring class *z,* the bias of that class pair is determined by the ratio of: *b*'s precision in images where both *b* and *z* are present, and *b*'s precision in images where it appears on its own.

This formula for bias calculation is adapted from the Singh et.al paper's definition in Equation 1: since we are working with detection and not classification, comparing the precision, recall, and

accuracy ratios was a reasonable adaptation from Equation 1's comparison of prediction probabilities.

4. Identifying new co-occurring pairs

In the study conducted by Singh et al., co-occurring pairs were defined like so:

$$\mathrm{c} = \arg\max_{\mathrm{z}} \mathrm{bias}(\mathrm{b}, \mathrm{z})$$

Based on their bias formula (see Related Work), this definition takes the bias of every possible class pair, and selects the class pair with the highest bias as the co-occurring pair.

Our co-occurring pairs are defined as:

$$c = \arg\min \frac{\mathrm{count}(a)}{\mathrm{count}(a+b)}$$

where *b* is the most frequently co-occurring class with *a*

Essentially, for each class in the dataset, we find the most common co-occurring class via frequency. The numerator represents the independent frequency of *a,* and the denominator is the joint frequency of *a* and its most frequently co-occurring class *b.* The ratio captures the proportion of times *a* appears independently relative to how often it appears alongside its most frequent co-occurring class *b.*

A lower value of this ratio indicates that *a* and *b* co-occur more frequently relative to how often *a* appears alone – which suggests a strong association between *a* and *b.* Thus, the formula effectively selects the co-occurring pair where the relationship between *a* and *b* is strongest.

This additional definition of biasing pairs supplemented our understanding of co-occurrence bias because it relied on frequency counts instead of calculations of accuracy ratios over all pairs.

Future work could delve into the most optimal formula to identify co-occurrence bias, since our work tested only two.

Results

A. For the first component of our project, we identified co-occurring objects in the COCO train set, which contains 118,00 images. In the figure below, Label B refers to the object that most frequently appears with Label A. The rightmost column represents how often Label A appears alongside Label B, for example, 99 percent of tennis rackets are pictured alongside a person in the dataset. Highlighted rows correspond to class pairs found by Singh et al. to have a high bias.

Label A	Label A Total	Label B	Label B Total	Co-occurrence Count	Co-occurrence Rate
tennis racket	3394	person	64115	3361	0.9903
baseball glove	2629	person	64115	2602	0.9897
baseball bat	2506	person	64115	2477	0.9884
skis	3082	person	64115	3044	0.9877
skateboard	3476	person	64115	3417	0.983
snowboard	1654	person	64115	1622	0.9807
surfboard	3486	person	64115	3365	0.9653
sports ball	4262	person	64115	4097	0.9613
tie	3810	person	64115	3632	0.9533
kite	2261	person	64115	2099	0.9284
backpack	5528	person	64115	5034	0.9106
handbag	6841	person	64115	6222	0.9095
umbrella	3968	person	64115	3447	0.8687
frisbee	2184	person	64115	1821	0.8338
bicycle	3252	person	64115	2643	0.8127
cell phone	4803	person	64115	3830	0.7974

motorcycle	3502	person	64115	2786	0.7955
bus	3952	person	64115	3011	0.7619
fork	3555	dining table	11837	2688	0.7561

B. Then, we calculate the COCO-trained YOLOv5 bias on the pairs identified by Singh et al. and a subset of the co-occurring pairs we identify. We present bias results for the COCO validation set (5000 images) and the Open Images test set (120,000 images).

Open Image Test Set—Biased Classes Identified by Singh et al.

Biased Class	Co-occur Class	Bias - Precision	Bias - Recall	Bias - mAP 50	Bias - mAP95
Wine glass	Person	0.597	1.267	0.926	0.625
Handbag	Person	0.722	0.917	0.791	0.808
Potted plant	Vase	1.265	1.622	1.852	1.778
Spoon	Bowl	4.54	0.921	2.472	2.644
Microwave	Oven	0.81	0.64	0.622	0.545
Keyboard	Mouse	1.523	1.371	1.421	1.518
Tennis racket	Person	1.005	1.133	0.871	0.87
Skateboard	Person	1.88	1.091	0.819	0.801
Baseball glove	Person	1.828	0.8876	1.6621	1.258

** remote, snowboard, and cup all had a very small number of images in the test set without the co-occurring class

COCO Validation Set—Biased Classes Identified by Singh et al.

Biased Class	Co-occur Class	Bias - Precision	Bias - Recall	Bias - mAP 50	Bias - mAP95
Cup	Dining table	0.956	1.474	1.21	1.22
Handbag	Person	1.124	0.806	0.984	0.87
Potted plant	Vase	0.954	1.203	1.041	1.012
Spoon	Bowl	1.0325	1.631	1.303	1.512
Microwave	Oven	1.052	1.132	1.026	1.024
Keyboard	Mouse	1.243	1.392	1.206	1.44

Remote	Person	1.019	0.703	0.753	0.737
Tennis racket	Person	0.805	inf	1.5	1.789
Baseball glove	Person	1.288	0.537	1.163	3.327

COCO Validation Set—Co-occurring COCO Train

Biased Class	Co-occur Class	Bias - Precision	Bias - Recall	Bias - mAP 50	Bias - mAP95
Tie	Person	0.599	1.539	1.1	1.14
Baseball Bat	Person	1.182	1.083	1.032	1.072
Sports Ball	Person	0.794	1.194	0.951	0.685
Surfboard	Person	0.903	0.828	1.067	0.922
Kite	Person	2.17	0.836	1.186	1.027
Frisbee	Person	1.155	0.936	1.1	1.142
Umbrella	Person	1.094	1.331	1.24	1.219
Backpack	Person	0.739	0.53	0.562	0.448

Analysis

We find several classes that co-occur in the COCO train set, presenting an avenue for bias in the model. Indeed, a number of these classes overlap with the ones Singh et al. found to be biased. The two class pairs with the highest co-occurrence rates, tennis racket/person and baseball glove/person, were among the three class pairs with the highest bias in Singh et al. Four other classes, skateboard/person, snowboard/person, sports ball/person, and handbag/person, identified by Singh et al. also appear in our list, providing supporting the idea that co-occurrence is an important factor in model bias. Still, several of Singh et al.'s biased class pairs are not among the most frequently co-occurring in COCO. For example, 98 percent of baseball bat instances occur alongside a person, yet Singh et al. did not find baseball bat/person to be among

the most biased classes according to their model, illustrating that co-occurrence does not necessarily result in bias.

We found that the YOLOv5 model was biased for many of the classes that Singh et al. found to be biased. From the Open Images test set, potted plant/vase, spoon/bowl, and baseball glove/person had the highest bias with respect to mean average precision (50), with values of 2.47, 1.85, and 1.66. From the COCO validation set, spoon/bowl, cup/dining table, and keyboard/mouse had the highest bias with respect to mean average precision (50), with values of 1.303, 1.21, and 1.206. Notably, the mAP50 for biased classes is higher for the Open Images evaluation set than the COCO validation set. One reason for this disparity might be a difference in labeling schemes between the two datasets. For example, the Open Images test set only has two images where spoon and bowl co-occur (compared to 153 images with spoon exclusive), though there appear to be a few unlabeled bowls in the spoon exclusive set.

Still, for some classes identified by Singh et al., bias is not highly present in our results. For example, with the COCO validation set, YOLOv5 achieves higher performance (mAP 50) when identifying a remote in the absence of a person than in the presence of a person, its supposed biasing class. Similarly, YOLOv5 does not exhibit a contextual bias for the handbag/person class with either the COCO or Open Images test set, with the class receiving a mAP50 bias below 1 for both datasets. This result—that some biased classes in Singh et al. are not biased in YOLOv5—suggests that there might be meaningful differences in the datasets used for evaluation, or that the YOLOv5 model—by using object detection, which focuses on local regions of an image—was able to avoid learning bias.

The role of co-occurrence in establishing contextual bias is illuminated by our results in Table 3, where we record the bias of the most commonly co-occurring objects in the COCO train dataset. Of the top eight co-occurring classes (outside of those identified by Singh et al.), YOLOv5 exhibited bias (mAP50) on six of them, showing that co-occurrence is an important factor in a model's contextual bias. It is interesting, though, that only half of these classes exhibited bias with respect to recall. Despite learning objects from many context-similar images, the model had fewer false negatives when detecting surfboard, kite, frisbee, and backpack in the absence of their co-occurring class. This finding illustrates that, while co-occurrence is important to contextual bias, other factors may more strongly impact how the model learns to detect an object inside or outside of its context. In addition, four classes, baseball bat, kite, frisbee, and umbrella are biased with respect to precision, meaning the model is less likely to identify a false positive when the object is depicted alongside its co-occurring object. Kite, in particular, has the greatest precision bias, with a value of 2.17. It is evident that context helps the YOLOv5 model to minimize incorrect predictions but hinders its ability to discern objects in the absence of their context.

So, what might explain the difference in bias between common co-occurring classes in the COCO train dataset? There are multiple ways in which co-occurrence can shape what a model learns. For one, if two objects frequently appear close to each other, the model may recognize the second object as a part of the first object. Alternatively, the co-occurring object may interact with the primary object in a way that affects its position in the image, and consequently, how the model learns the object. For example, the YOLOv5 is much more likely to detect a cup when it appears in an image with a table (recall bias = 1.474). We observe that COCO images with both cup and dining table often depict a cup on a table in visually consistent positions:



In contrast, images with a cup but no dining table seem to present the cup in more varied ways, hindering the model's ability to detect it.

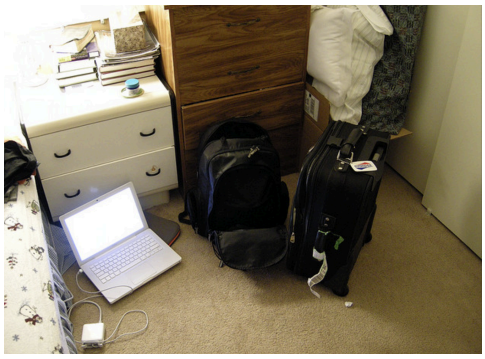
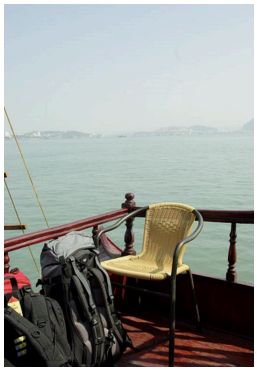
(The parent in the back is holding the cup.)

(There is a cup on the coffee table.)

In the opposite case, the YOLOv5 model performs better when detecting a backpack in the absence of a person. Intuitively, it seems like a person might make it more difficult to detect a backpack, as a person wearing a backpack may lead to occlusion. We do see this occur in the COCO validation set, and we also make another observation: images with a backpack and person tend to be more cluttered than images with just a backpack. For example, in the images below, the backpack is only a small, often occluded, detail in comparison to the human activity.




On the other hand, when a backpack appears without a person, the object tends to be more prominent in the image, perhaps allowing the model to make detections with greater accuracy.


Our results illustrate that co-occurrence is important in establishing contextual bias, yet it is not everything. The particulars of the co-occurring object pairs play a significant role in how the model learns the object. For example, the presence of a person may suggest that the object is altered or positioned differently, or that the object is appearing in a different, more difficult context, such as a crowded plaza. To understand contextual bias, it is important to evaluate both the quantitative calculation of bias and the practical implications of two objects occurring together.

Conclusion and Future Work

This study builds upon the work of Singh et al. by evaluating the role of co-occurrence in contextual bias in object detection. We identify co-occurring class pairs in the COCO train set and calculate bias by evaluating YOLOv5's performance. We find that the YOLOv5 model experiences co-occurring bias, similar to the classification model used by Singh et al, and that co-occurrence is an important—but not definitive—factor in the model's bias. From qualitative observation, we can see that co-occurrence can influence the model's view of an object in a variety of ways, and calculating bias alone does not explain why a model is or isn't able to detect an object out of context.

Future work could explore fine-tuning approaches to enhance the model's ability to detect objects even in the absence of their co-occurring counterparts. Additionally, there is a need to construct datasets that better represent objects of interest out of their typical contexts. For instance, if a skateboard is mostly represented in a dataset in images where a person is standing on it, the model might only be able to detect a single positioning of the skateboard (i.e., flat and side view). Even with 120,000 images through Open Images, we found that some of our subsets

where the object was independent of its co-occurring object were extremely small, undermining the comprehensiveness of our evaluation.

The identification of new commonly co-occurring class pairs informs the curation of more robust and representative datasets, which could lead to the development of more context-agnostic models in not just object detection, but also in other areas of computer vision – mitigating the effects of co-occurring bias.

Discussion—Pivot from Initial Project

Our original project aimed to use the YOLOv5 model on the COCO dataset, identifying a weak-performing class and improving the model on that class via fine-tuning and data augmentation. We built a dataset to fine-tune YOLOv5, prepared to experiment with different data augmentation approaches, and ran train our class-specific dataset. However, we encountered challenges with this approach, as the model became prone to overfitting, limiting its generalizability and undermining the effectiveness of data augmentation. Thus, we switched our paper topic to build on Singh et al., which has been a valuable experience in analyzing a model quantitatively and qualitatively.

References

[1] K. K. Singh, D. Majahan, K. Grauman, Y. J. Lee, M. Feiszli and D. Ghadiyaram. (2020). Don't Judge an Object by Its Context: Learning to Overcome Contextual Bias. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, https://arxiv.org/abs/2001.03152

[2] I. Model and L. Shamir, "Comparison of Data Set Bias in Object Recognition Benchmarks," in IEEE Access, vol. 3, pp. 1953-1962, 2015, doi: 10.1109/ACCESS.2015.2491921.

[3] S. Agarwal, S. Muku, S. Anand, and C. Arora. (2022). Does Data Repair Lead to Fair Models? Curating Contextually Fair Data To Reduce Model Bias. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9706860

[4] M. Zhang, C. Tseng, and G. Kreiman. (2020). Putting visual object recognition in context. *2020 CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, https://arxiv.org/abs/1911.07349

[5] R. Liu, H. Liu, G. Li, H. Hou, T. Yu and T. Yang, "Contextual Debiasing for Visual Recognition with Causal Mechanisms," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, 2022, pp. 12745-12755, doi: 10.1109/CVPR52688.2022.01242.

[6] H. Sun and A. Kusari. (2024). Quantifying Context Bias in Domain Adaptation for Object Detection. *2024 CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, https://arxiv.org/abs/2409.14679

[7] M. Dreyer, R. Achtibat, T. Wiegand, W. Samek and S. Lapuschkin, "Revealing Hidden Context Bias in Segmentation and Object Detection through Concept-specific Explanations," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Vancouver, BC, Canada, 2023, pp. 3829-3839, doi: 10.1109/CVPRW59228.2023.00397

[8] Ultralytics, "COCO Dataset." https://docs.ultralytics.com/datasets/detect/coco/ [Accessed Dec. 13, 2024].

[9] Ultralytics, "Open Images V7 Dataset." https://docs.ultralytics.com/datasets/detect/open-images-v7/ [Accessed Dec. 13, 2024].

[10] ScienceDirect, “YOLOv5.” https://www.sciencedirect.com/topics/computer-science/yolov5

[Accessed Dec. 13, 2024].