Taken Out of Context: Towards Mitigating Contextual Bias in Computer Vision

Nicole Meister

May 10, 2021

advised by

Prof. Olga Russakovsky

Submitted in partial fulfillment
of the requirements for the degree of
Bachelor of Science in Engineering
Department of Electrical Engineering
Princeton University

I hereby declare that this Independent Work report represents my own work in accordance with University regulations.

Nicole Meister

Nicole Meister

Taken Out of Context: Towards Mitigating Contextual Bias in Computer Vision

Nicole Meister

Abstract

Contextual cues are traditionally leveraged in computer vision to improve accuracy. However, over-reliance on co-occurring objects compromises a model's generalizability, especially when an object occurs "exclusively" or without its context. A weighted loss model is a competitive contextual bias mitigation method that improves accuracy on images with the exclusively occurring biased class. However, this model compromises the performance when the biased class co-occurs with its context. Our key idea is to selectively combine the standard and weighted model to use the weighted model only when analyzing exclusive images and the standard model for all other images to leverage the standard model's contextual bias for when images contain co-occurring objects. This improves the accuracy of recognizing a category in the absence of its context, without compromising on performance when it co-occurs with context. In this report we describe a stacked model implementation, a decision rule algorithm, and the extensions explored improve the decision rule. The decision rule algorithm demonstrates strong potential in mitigating contextual bias, but there is room for improvement especially in increasing the accuracy of determining if an image is "exclusive" or "non-exclusive."

ACKNOWLEDGEMENTS

To Professor Olga Russakovsky, thank you so much for your insight, guidance, and infinite wisdom imparted during our meetings. I have improved so much as a researcher because of you and I am beyond grateful.

To Sunnie Kim, thank you for taking a chance on me and letting me join the Reproducibility Challenge as a random COS429 student. Your organization, work ethic, and ways of solving problems (and Slack emoji usage) are so inspiring to me.

To Sharon Zhang, thank you for answering all of my questions, and for your willingness to bounce ideas off each other. I'm really going to miss our Friday lunches and encouraging comments on Strava runs, but I'm so excited for your future! I am really blessed to have such inspiring role models.

Thank you to everyone in the VisualAI lab for the weekly meetings and tutorials as I have learned so much this semester. I look forward to another exciting and productive year.

Thank you also to my second reader, Dr. Jason Lee, for his support and guidance.

Lastly, thank you to my family and friends for their support during this semester and to Catalina at the Whitman dining hall for always bringing a smile to my face.

Contents

1 Introduction

Computer vision relies heavily on the availability of large-scale annotated datasets [1] for training. However, many of these datasets are afflicted by bias [2], specifically contextual bias. For example, "skateboards" are often overwhelming depicted with a "person" riding the skateboard and in images these two classes often co-occur. This results in a largely imbalanced dataset where "skateboard" and "person" appear more often than simply "skateboard" on its own. For example, when looking deeper into the label statistics of training data, the Microsoft Common Objects in COntext (COCO) [1] is a dataset containing Flickr images containing 91 common object categories. When looking at images containing the "skateboard" class in the COCO training dataset, there are 2473 images where "skateboard" occurs with "person" and only 38 images where "skateboard" occurs without "person."

Figure 1: When looking at training images of the class "skateboard" from the COCO-Dataset, they co-occur overwhelming with "person". The co-occurring class "person" induces bias on the biased class "skateboard."

Such co-occurrence patterns and data skew inadvertently induce contextual bias in datasets, which could consequently seep into models trained on them. When models overly rely on context, they may not generalize to settings where typical co-occurrence patterns are absent. During test time we see the effects of this inability to generalize as the average prediction probability of skateboard in an exclusive image is 46% compared to 79% on co-occurring images.

Contextual cues have been traditionally leveraged in computer vision to improve accuracy.

For example, when creating Deformable Part Model, Felzenszwalb et. al [3] specifically use contextual information to boost the performance of their model. Similarly, early work by Oliva & Torralba [4] investigates the role of context in object recognition as it recognizes that natural way of representing the context of an object is in terms of its relationship to other objects. Context becomes especially crucial for our visual system when the visual signal is ambiguous or incomplete (e.g., due to occlusion, viewpoint of the scene capture, etc.). However, over-reliance on co-occurring objects compromises a model's generalizability, especially when an object occurs "exclusively" or without its context.

To build upon previous work targeting contextual bias, we draw inspiration from Singh et al's recent work titled "Don't Judge an Object by Its Context: Learning to Overcome Contextual Bias." [5] The goal of this paper is to improve the accuracy of computer vision models to recognize a category in the absence of its context, without compromising on performance when it co-occurs with its context. Given that [5] provides no publicly available code, we reproduced all the tables and figures from the paper to validate their method and appropriately build off of their contributions.

After reproducing the results in this paper, a natural next step was to develop different techniques of accurately recognizing a category both in the presence and absence of its context. We noticed that certain models perform extremely well on exclusive images but their performance on the co-occurring images was compromised. This led us to investigate techniques of model ensembling and stacking to improve the accuracy of computer vision models to recognize a category in the absence of its context, without compromising on performance when it co-occurs with its context. In this report, our ultimate goal is to develop a method that increases the exclusive mAP above the method proposed by [5] and maintains the co-occur mAP. We explore this goal by implementing a stacking method and developing a decision rule algorithm.

2 Contextual Bias Problem Setup

In this section, we will describe the relevant results from reproducing Singh et. al's [5] method and results, specifically regarding the *standard* model, how to quantify contextual

bias, identify biased category pairs, their proposed method, competitive contextual bias mitigation methods, and the reproduced results.

2.1 Standard Model

To measure the extent of contextual bias, Singh et. al [5] train standard multi-label classifier with the binary cross entropy loss on the COCO-Stuff dataset.

We downloaded COCO-Stuff [6] from `https://github.com/nightrome/cocostuff`. COCO-Stuff includes all 164K images from COCO-2017 (train 118K, val 5K, test-dev 20K, test-challenge 20K), but only the training and validation set annotations are publicly available. It covers 172 classes: 80 thing classes, 91 stuff classes and 1 class designated 'unlabeled.' Excluding the 'unlabeled' category, we have in total 171 categories.

Singh et. al use a ResNet-50 [7] pre-trained on ImageNet [8] as a backbone and optimized with stochastic gradient descent (SGD) with a momentum of 0.9, no weight decay, and a batch size of 200. The *standard* model is optimized with an initial learning rate of 0.1 for 60 epochs, then decreased to 0.01 for 40 epochs for a total of 100 epochs. The input images are randomly resize-cropped to 224×224 and randomly flipped horizontally during training. While we found that our standard model slightly underperforms against the paper's results, we conducted an extensive hyperparameter search where we varied the learning rate, weight decay, and the epoch at which the learning rate to confirm these were the optimal hyperparameters.

2.2 Quantifying Contextual Bias

To identify biased categories pairs (b, c) where b denotes the biased class and c denotes the co-occurring class, Singh et. al describe the following intuition: Contextual bias is prevalent when (1) the prediction probability of b drops significantly in the absence of c and (2) b co-occurs frequently with c. They formulate a definition of bias between two categories b and z as the ratio between average prediction probabilities of b when it occurs with and without z where z appears at least 20% of the time with b. This definition of bias requires a trained model, unlike the more common definition of bias that only requires co-occurrence

counts in a dataset [9].

More formally, for a category z:

- $\mathbb{I}_b \cap \mathbb{I}_z$: the set of images where b occurs with z
- $\mathbb{I}_b \backslash \mathbb{I}_z$: sets of images where b occurs without z
- $\hat{p}(i, b)$: the prediction probability of an image i for a category b obtained from training a standard multi-label classifier

The contextual bias between two categories b and z is defined as follows:

$$\text{bias(b, z)} = \frac{\frac{1}{|\mathbb{I}_b \cap \mathbb{I}_z|} \sum_{i \in \mathbb{I}_b \cap \mathbb{I}_z} \hat{p}(i, b)}{\frac{1}{|\mathbb{I}_b \backslash \mathbb{I}_z|} \sum_{i \in \mathbb{I}_b \backslash \mathbb{I}_z} \hat{p}(i, b)} \tag{1}$$

The co-occurring class is the class that exhibits the most bias as defined by Eq. 1:

$$c = \underset{z}{\text{argmax}}\, \text{bias(b, z)} \tag{2}$$

2.3 Identifying Biased Category Pairs

Singh et. al [5] found that focusing on the top 20 biased category pairs sufficiently captures biased categories in COCO-Stuff. Thus similarly, when reproducing [5], we used a *standard* model (Section 2.1) trained on an 80 split of the COCO-Stuff training data and calculated bias on the 20 split of the training dataset. For each category b, we calculated the bias between b and its frequently co-occurring categories (appears at least 20% of the time with b), and defined category c as the context category that most biases b, i.e. has the highest bias value. After the bias calculation, we identified 20 (b, c) pairs with the highest bias values. This definition of bias is directional; it only captures the bias c incurs on b and not the other way around. [5]

In Table 1, we report the co-occurrence, exclusive, and other counts for the paper's 20 biased category pairs. The co-occurrence count is the number of images where b and c co-occur; the exclusive count is the number of images where b occurs without c. Overall, the bias values of the paper's biased category pairs calculated with our model are similar to the

paper's values. Furthermore, most (18 of the 20) biased categories overlap, although their context categories sometimes differ.

Biased category pairs		Bias		Training (82,783)		Test (40,504)		Biased category pairs (Ours)		
Biased (b)	Context (c)	Paper	Ours	Co-occur	Exclusive	Co-occur	Exclusive	Biased (b)	Context (c)	Bias
cup	dining table	1.76	1.85	3,186	3,140	1,449	1,514	car	road	1.73
wine glass	person	1.80	1.59	1,151	583	548	304	potted plant	furniture-other	1.75
handbag	person	1.81	2.25	4,380	411	2,035	209	spoon	bowl	1.75
apple	fruit	1.91	2.12	477	627	208	244	fork	dining table	1.78
car	road	1.94	1.73	5,794	2,806	2,842	1,331	bus	road	1.79
bus	road	1.94	1.79	2,283	507	1,090	259	cup	dining table	1.85
potted plant	vase	1.99	1.73	930	2,152	482	1,058	mouse	keyboard	1.87
spoon	bowl	2.04	1.75	1,314	954	638	449	remote	person	1.89
microwave	oven	2.08	1.59	632	450	291	217	wine glass	dining table	1.94
keyboard	mouse	2.25	2.11	860	601	467	278	clock	building-other	1.97
skis	person	2.28	2.21	2,180	29	984	9	keyboard	mouse	2.11
clock	building	2.39	1.97	1,410	1,691	835	840	apple	fruit	2.12
sports ball	person	2.45	3.61	2,607	105	1,269	55	skis	snow	2.22
remote	person	2.45	1.89	1,469	666	656	357	handbag	person	2.25
snowboard	person	2.86	2.40	1,146	22	522	11	snowboard	person	2.40
toaster	ceiling	3.70	1.98	60	91	30	44	skateboard	person	3.41
hair drier	towel	4.00	3.49	54	74	28	41	sports ball	person	3.61
tennis racket	person	4.15	1.26	2,336	24	1,180	10	hair drier	sink	6.11
skateboard	person	7.36	3.41	2,473	38	1,068	24	toaster	oven	8.56
baseball glove	person	339.15	31.32	1,834	19	820	9	baseball glove	person	31.32

Table 1: (Left) The paper's 20 most biased category pairs and their bias values, both what's reported in the paper and what we've calculated with our trained model. (Middle) The number of co-occurring and exclusive images for each pair. (Right) The 20 most biased categories we've identified with our trained model.

2.4 Feature-split Method

Singh et. al [5] propose two methods to mitigate contextual bias. The first method (titled *CAM-based*) improves the model's ability to recognize a category in the absence of its context; however, it compromises on performance when it co-occurs with its context. The second method, titled *feature-split* is described in this section.

Figure 2: Our illustration of the *feature-split* method: This method splits the feature space into two subspaces to separately represent category and context. Specifically, they propose using a dedicated feature subspace to learn examples of biased categories appearing without their context.

Given a deep neural network, let x denote the D-dimensional output of the final pooling layer just before the fully-connected (fc) layer. Let the weight matrix associated with fc layer be $W \in R^{D \times M}$, where M denotes the number of categories given a multi-label dataset. The predicted scores inferred by a classifier (ignoring the bias term) are $\hat{y} = W^T \mathrm{x}$. To separate the feature representations of a biased category from its context, the paper does a random row-wise split of W into two disjoint subsets: W_o and W_s (dimension $\frac{D}{2} \times M$). Consequently, x is split into x_o and x_s, and $\hat{y} = W_o^T \mathrm{x}_o + W_s^T \mathrm{x}_s$. When a biased category occurs without its context, the paper disables backpropagation through W_s, forcing the network to learn only through W_o, and set x_s to $\bar{\mathrm{x}}_s$ (the average of x_s over the last 10 mini-batches) (Fig. 2).

For a single training batch, we first forwarded the entire batch through the model to obtain one set of scores $\hat{y}_{\text{non-exclu}} = W_o^T \mathrm{x}_o + W_s^T \mathrm{x}_s$ and the corresponding features from the avgpool layer, which directly precedes the fc layer. We made a separate copy of these features and replaced x_s with $\bar{\mathrm{x}}_s$, then calculated a new set of output scores $\hat{y}_{\text{exclu}} = W_o^T \mathrm{x}_o + W_s^T \bar{\mathrm{x}}_s$. Separate loss tensors for each of these outputs were computed, and elements corresponding to the exclusive and non-exclusive examples in the unmodified and modified loss tensors were zeroed out, respectively. The final loss tensor was obtained by adding these two together, and standard backpropagation was done using this final loss tensor. The gradients were calculated with respect to a weighted binary cross entropy loss:

$$L_{\text{WBCE}} = -\alpha \big[t \log(\sigma(\hat{y})) + (1 - t) \log(1 - \sigma(\hat{y})) \big], \tag{3}$$

where t is the ground-truth label, σ is the sigmoid function, and α is the ratio between the

number of training images in which a biased category occurs in the presence of its context and the number of images in which it occurs in the absence of its context. A higher value of α indicates more data skewness.

2.5 Additional Competitive Contextual Bias Mitigation Methods

In addition to the *standard* model, [5] compare their proposed methods with several competitive strong baselines that improve performance on the exclusively occurring images. It should be noted that these baselines do not aim to maintain performance on the co-occuring images and [5] uses these baselines to demonstrate the benefit of the two proposed methods *CAM-based* and *feature-split*. These strong baselines are trained for 20 epochs on top of the *standard* model using a learning rate of 0.01, a batch size of 200, and SGD with 0.9 momentum. All models were trained on a single RTX 3090 GPU provided by the Princeton Computing Clusters and evaluated on the last epoch. Overall, the total training time for each method took 35-43 hours.

1. **Remove co-occur labels**: For each b, edit the label vector to remove the c label for images in which b and c co-occur.

2. **Remove co-occur images**: Remove training images and its corresponding labels where any b and c co-occur. This process removes 29,332 images and leaves 53,451 images in the training set.

3. **Split-biased**: Split each b into two classes: 1) $b \setminus c$ and 2) $b \cap c$. Unlike other "stage 2" models, this model is trained from scratch for 100 epochs (20 additional epochs) as it has 20 additional classes.

4. **Weighted loss**: For each b, apply 10 times higher weight to the loss for class b when b occurs exclusively.

5. **Negative penalty**: For each (b, c), apply a large negative penalty to the loss for class c when b occurs exclusively. In our email communication, the authors said that the negative penalty means a 10 times higher weight to the loss.

6. **Class-balancing loss** [10]: For each b, put the images in three groups: exclusive, co-occurring, and other. The weight for each group is $(1-\beta)/(1-\beta^n)$ where n is the number of images for each group and β is a hyperparameter ($\beta = 0.99$)

2.6 Evaluation Setup

The mAP (Mean Average Precision) is reported on exclusive and co-occur distributions which are illustrated in Figure 3. For each (b, c) pair, the test set can be divided into three sets: *co-occurring* images that contain both b and c, *exclusive* images that contain b but not c, and *other* images that do not contain b. Then for each (b, c) pair, there are two test distributions: 1) the "exclusive" distribution containing *exclusive* and *other* images and 2) the "co-occur" distribution containing *co-occurring* and *other* images (Fig. 3). The *other* images are likely included in both distributions because otherwise, both distributions would have small sizes and only consist of positive images where b occurs, disabling the mAP calculation.

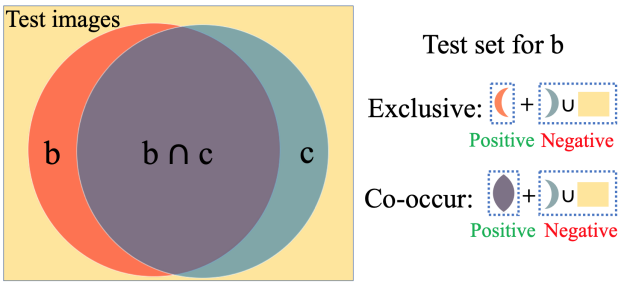


Figure 3: Evaluation Setup: Exclusive mAP refers to the results on both the exclusive (positive) images and the other (negative) images and similarly for the co-occur distribution. Figure adapted from [5].

2.7 Reproduced Results

The results from reproducing the methods of Singh et. al [5] are shown in Table 2 and Figure 4. Overall, the general pattern and performance of models matches. We verify the claim that the proposed methods improve the accuracy of computer vision models to recognize a category in the absence of its context, without compromising on performance when it co-occurs with its context. We use this set up as the basis of our proposed algorithm

and the performance of the *feature-split* method as a state-of-the-art method to compare our own method described in future sections. We are especially pleased to report that this reproducibility work was selected for publication in the Rescience C Journal [11] and some of the text and results in this section are adapted from our submitted report.

Method	Exclusive		Co-occur	
	Paper	Ours	Paper	Ours
standard	24.5	23.9	**66.2**	**65.0**
remove labels	25.2	24.5	65.9	64.6
remove images	28.4	**29.0**	28.7	59.6
split-biased	19.1	25.4	64.3	64.7
weighted	**30.4**	28.5	60.8	60.0
negative penalty	23.8	23.9	66.1	64.7
class-balancing	25.0	24.6	66.1	64.7
feature-split	28.8	28.1	66.0	64.8

Table 2: Performance of different methods on "exclusive" and "co-occur" distributions of COCO-Stuff with best results in bold.

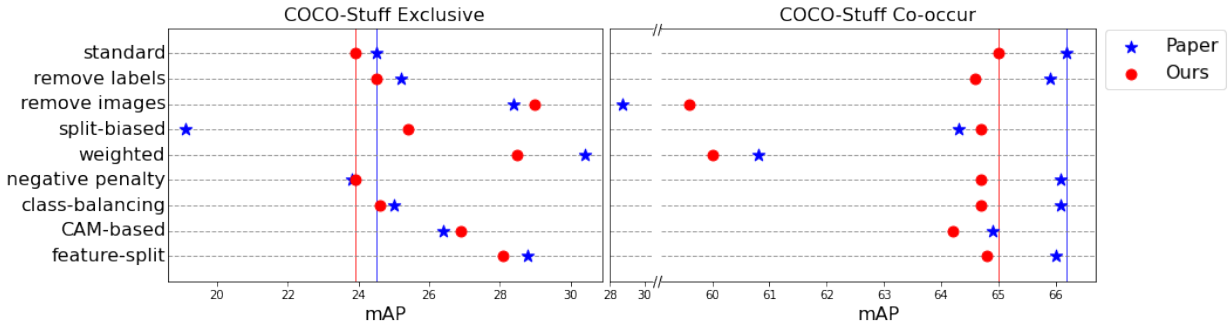


Figure 4: A visual comparison of the results on COCO-Stuff from Table 2. The blue and red lines mark the paper's and our *standard* mAPs.

3 Model Ensembling and Stacking

A natural next step after reproducing the results from Singh et. al [5] was to pursue this goal further and develop different techniques of accurately recognizing a category both in the presence and absence of its context. We noticed some strong baselines, specifically *weighted* model performs extremely well on exclusive images but its performance on the co-occurring images was compromised. This led us to investigate techniques of model ensembling and stacking to see if we can improving the accuracy of computer vision models to recognize a

category in the absence of its context, without compromising on performance when it co-occurs with its context. In the following Section (Sections 3 and 4) our ultimate goal is to develop a method that increases the exclusive mAP above the *feature-split* method and maintains the co-occur mAP.

3.1 Model Averaging

Model averaging is one of the simplest and most common ensemble approaches for neural networks. It involves taking an average of the output probability for all base models (the last layer of the neural network) and returning it as the new predicted probability [12].

Methods: To apply this method to our problem of mitigating contextual bias, we take an unweighted average, weighted average, and maximum of the sigmoid output probabilities of the *standard* and *weighted* model described in Section 2.1 and 2.5. Ju et. al warn that unweighted averaging is reasonable for similar base models of comparable performance [13], yet it is important to note the *standard* and *weighted* model do not have comparable performance. Additionally, since we are tuning hyperparameters in these methods, the *standard* and *weighted* model are trained on the 80 split of the COCO-Stuff training dataset and evaluated on the 20 split of the COCO-Stuff training dataset.

Method	Exclusive	Co-occur
standard	20.7	**65.2**
weighted	**24.4**	60.0
unweighted average	22.4	63.2
max	24.2	60.6
weighted average	23.1	62.2
feature-split	22.4	64.5

Table 3: mAP results from an unweighted average, weighted average, and maximum of the sigmoid output probabilities of the *standard* and *weighted* model. The weighted average involves giving the *standard* model a 30% weighting and the *weighted* model a 70% weighting.

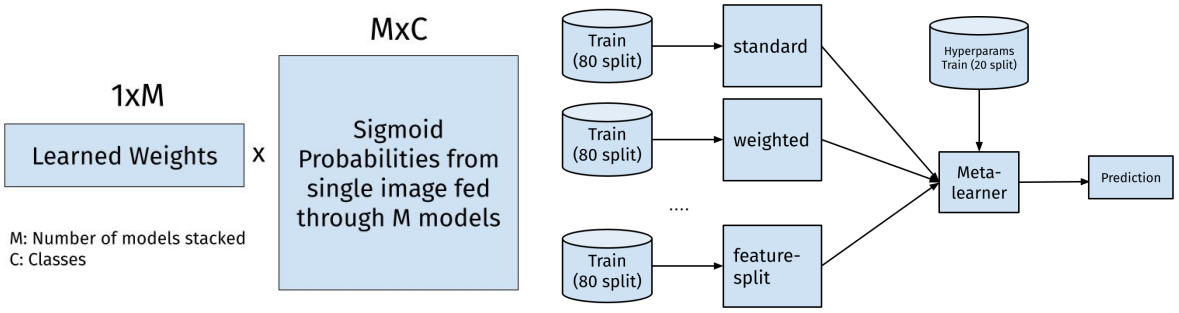
Results: As expected, the results of the model in Table 3 averaging did not outperform *feature-split* or other previous methods as the exclusive mAP from the *weighted* model is the highest among all models and the *standard*'s co-occur mAP is the highest among all models; however, these results made clear the distinct trade-off between exclusive mAP and co-occur

mAP .

3.2 Model Ensembling and Stacking

A good meta-learner should be intelligent enough to combine the strength of models. To take model averaging one step further, we explored ensemble learning, a method of training several models to combine them together to make predictions [13]. Ensemble learning methods have gained popularity especially in Kaggle competitions because of their simplicity and superior prediction performance in practice. Proposed by [14], the basic idea of stacking is to 'stack' the sigmoid output probabilities of m models, $f_1, ..., f_m$, by linearly combine them with weights $a_i, i \in 1, ..., m$ (Fig. 5a):

$$f_{\text{stacking}}(\mathbf{x}) = \sum_{i=1}^{m} a_i f_i(\mathbf{x}) \tag{4}$$



(a) Weights Learned by A Meta-learner (b) Stacking Visual

Figure 5: Visual of Processes Involved in Stacking

Methods: We implemented stacking using $m = 6$ base models: standard, weighted, negative penalty, class-balancing, CAM-based, and featuresplit (Fig. 5b). We want the images from the 80 split of the training data to serve as probability features for the meta-learner, yet we do not want the model to have trained on or seen these images before. Thus, the train 80 split is partitioned into 5 folds and we train 5 version of each base model. Each base model is trained on $\frac{4}{5}$ of the training set and computes features for the remaining $\frac{1}{5}$ (ie. The *standard* model is trained on fold 1, 2, 3, 4 and the probability features are produced from fold 5 to create $\frac{1}{5}$ of the training meta features. Another version of the *standard* model

is trained on fold 1, 2, 3, 5 and the probability features are produced from fold 4 to create $\frac{2}{5}$ of the training meta features).

Results and Analysis: Ultimately, the 6 weights of this simple stacked meta-learner converged to an equal distribution likely due to the small number of exclusive images. Even if the probabilities contained information about a different model behavior to exclusive images, there exist so few exclusive images that the meta-learner learns to trust each model equally. Additionally, this meta-learner was a simple linear combination of weights and could not learn complex relationships within the probability distributions. However, exploring ensembling served as valuable inspiration for building the decision rule algorithm described in Section 4.

4 Decision Rule Algorithm

From Section 3, it evident that programming a meta-learner to learn that the *weighted* model performs better with exclusive images is difficult. However, given that we know the *weighted* model performs better with exclusive images, why not feed the *weighted* model only exclusive images and the *standard* model non-exclusive images? In this section we describe the decision rule algorithm and explore two main extension to improve the ability of determining if an image is "exclusive" or "non-exclusive" through additional post-hoc features. Our goal is once again to develop a method that increases the exclusive mAP above the feature-split method and maintains the co-occur mAP.

4.1 Methods

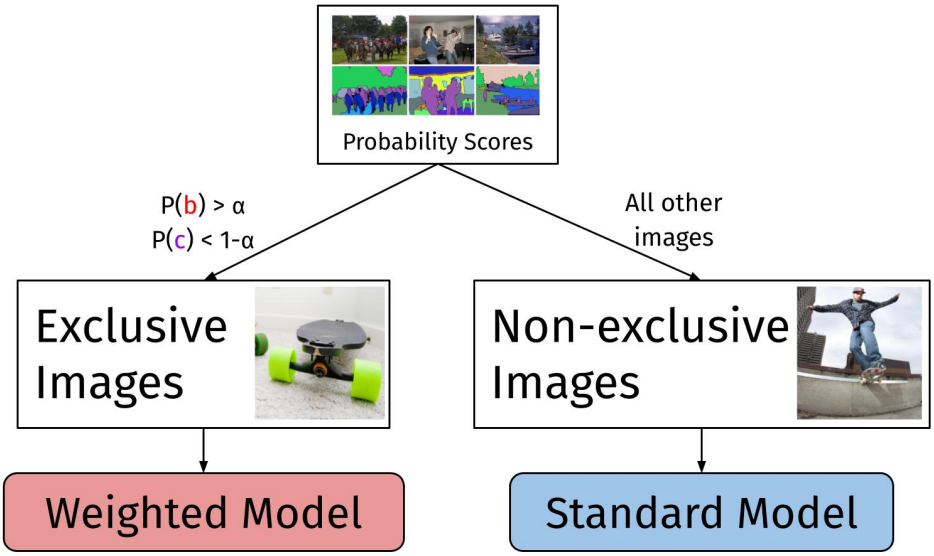


Figure 6: Decision Rule Algorithm

A decision rule consist of a conditional IF-THEN statement to make a prediction [15]. As depicted in Fig. 6, when applied to our goal, if the image is exclusive, it is fed through the *weighted* model as it performs well on exclusive images. If this image is non-exclusive, it is fed through the *standard* model as it can leverage contextual cues to its advantage.

To determine if an image is exclusive or non-exclusive, we look at the probability scores from the *standard* or *weighted* model. We define a probability cut-off α where images with the property $P(b) > \alpha$ and $P(c) > 1-\alpha$ are exclusive and all other images are non-exclusive. Then, for each of the 20 biased category pairs, if categorize an image as exclusive, we replace the probability score of the biased class with that of the weighted model. If the image is not exclusive, the standard probability scores remain. Finally, we evaluate each model on the exclusive and co-occur distributions as determined by the ground truth labels. We pass in the final scores of the model and calculate mAP.

Additionally, to determine the upper bound of this method, we report results when making the decision using the ground truth labels (Fig 7a).

When implementing this method, we train a *standard* and *weighted* model on the 80 split of the training dataset and tune hyperparameters on the validation dataset (20 split of the

training dataset). We would like to acknowledge and emphasize that the numbers reported in Section 4.2 are evaluated on the validation dataset to determine optimal hyperparameters, but ultimately these numbers will be updated to reflect the results from images from the test set fed through the model trained on the 80 split of the training dataset.

4.2 Results

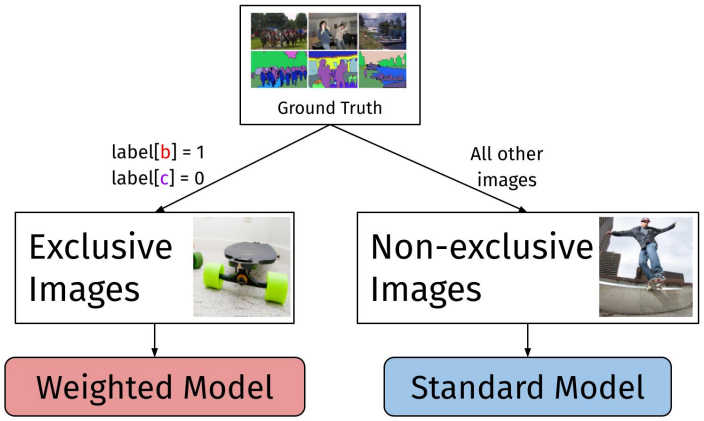
Method	Exclusive	Co-occur
standard	20.7	**65.2**
weighted	24.4	60.0
average	22.4	63.2
decision with ground truth	**38.8**	**65.2**
decision with standard prob ($\alpha = .33$)	23.2	62.5
decision with standard prob ($\alpha = .53$)	22.4	63.9
decision with weighted prob ($\alpha = .67$)	23.4	61.7
feature-split	22.4	64.5

Table 4: mAP results from the decision rule algorithm.

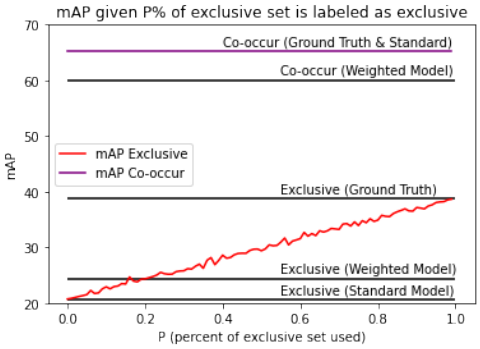
The results from the ground truth experiment (Table 4) suggest high potential in the decision rule algorithm as it achieves a 38.8 exclusive mAP and 65.2 co-occur mAP. This outperforms *weighted* model's exclusive mAP (+14.4) and maintains the *standard* model's performance on co-occurring images because the standard model probability for non-exclusive images are unchanged.

In Fig. 7b, we adjust the amount of truly exclusive images fed through the *weighted* model and see that when 20% of truly exclusive images and 0% of non-exclusive images are fed through the *weighted* model, the exclusive mAP equal that of the *weighted* model.

When the exclusive images are determined by the model's probability scores as it would be in a real-world application, the decision rule algorithm outperforms simple model averaging, but falls short when compared to state-of-the-art contextual bias mitigation methods, *feature-split*. Results of the decision rule model are reported using standard and weighted scores given the alpha that optimizes the F1 score (Fig. 8c and Fig. 8d). Additionally, we plot the mAP as α changes (Fig. 8a and Fig. 8b). Results are reported given the highest F1 score, yet we emphasize that optimal alpha depends on the trade-off one is willing to make between the performance on exclusive and co-occurring images.



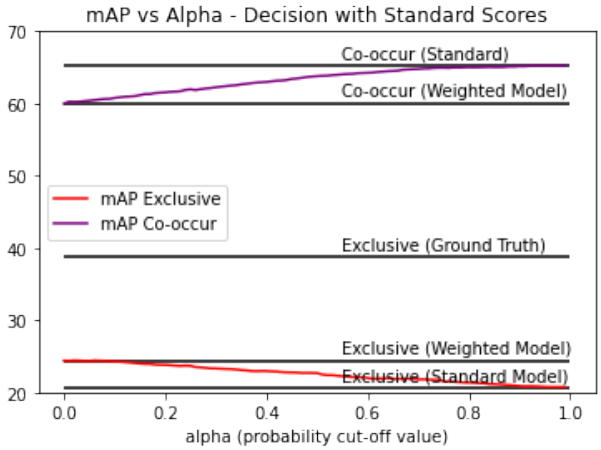
(a) Visual



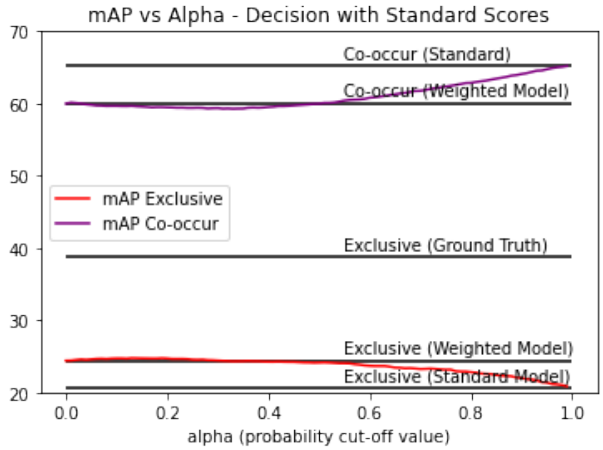
(b) mAP vs decision probability cut-off using ground truth labels.

Figure 7: Setup and results from implementing decision rule algorithm with ground truth labels.

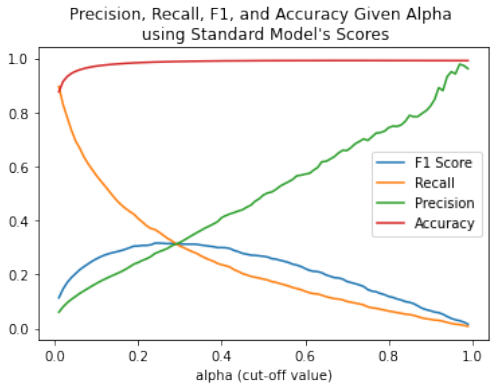
To compare this method's performances to state-of-the-art methods such as the *feature-split* method by [5], we provide the decision rule algorithm with an $\alpha = .53$ to make the decision based on the *standard* model's output. From this model, the exclusive mAP is equivalent to that of *feature-split*, yet the co-occur mAP slightly underperforms the *feature-split* method with a -0.6 difference in mAP.



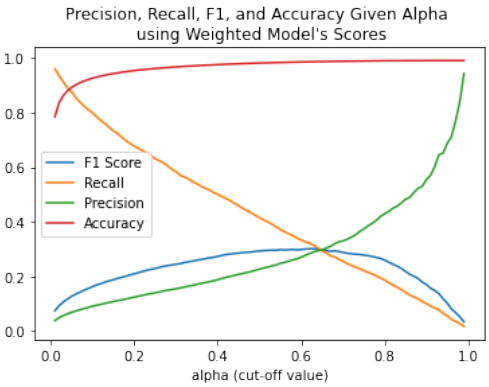
(a) Decision using *standard* model output.



(b) Decision using *weighted* model output.



(c) Precision, Recall, F1, and Accuracy scores of the decision rule when



(d) Decision using *weighted* model output.

Figure 8: **(a)** and **(b)**: mAP vs decision probability cut-off using *standard* or *weighted* model output, respectively. **(c)** and **(d)**: Precision, Recall, F1 Score, and Accuracy scores of the decision rule given the *standard* or *weighted* model output, respectively.

4.3 Analysis

We use this decision rule to balance the strengths and weakness of the *standard* and *weighted* model and the trade-off between the exclusive and co-occur mAP is evident in Fig. 8a and 8b. When more images (both exclusive and non-exclusive) are fed into the *weighted* model, the exclusive mAP increases, but it lowers the performance on the co-occurring images because the *weighted* model performs worse on co-occurring images.

The ground truth exclusive mAP increased significantly because the exclusive distribution contains the "exclusive" images and "other" images (images that do not contain the biased

class b). Compared to the *standard* model, the *weighted* model performs worse on "other" images which decreases the exclusive mAP. With the ground truth method, only the exclusive images in the exclusive distribution are fed through the *weighted* model; the "other" images are fed through the *standard* model. We developed this insight by looking at mean probability scores for label b, $\bar{P}(b)$, where for exclusive images, a larger $\bar{P}(\hat{b})$ indicates a better model and for "other" images, a lower $\bar{P}(\hat{b})$ indicates a better model because it recognizes that the biased class does not exist in the image. On exclusive images, $\bar{P}_{\text{standard}}(\hat{b}) = 0.33$ and $\bar{P}_{\text{weighted}}(\hat{b}) = 0.45$. On the "other" images, $\bar{P}_{\text{standard}}(\hat{b}) = 0.019$ and $\bar{P}_{\text{weighted}}(\hat{b}) = 0.035$. Since there exist many more "other" images in the dataset than exclusive images, when the *standard* model's performance on the "other" images is leveraged, the mAP on the exclusive distribution (both the exclusive and "other" images) increases.

Additionally, to investigate the accuracy of the decision rule algorithm, we look at the top 10% of the following value: $P_{\text{weighted}}(b)+1-P_{\text{standard}}(c)$ and find that 53.9% of the images are correctly identified as exclusive. This provides insight into why the decision rule's exclusive mAP does not increase far beyond the *weighted's* exclusive mAP as many of the images fed through the weighted model are not truly exclusive.

The high increase in exclusive mAP using the ground truth labels to make decisions demonstrate promising results for the decision rule algorithm, given that we can develop better ways to identify images as exclusive. These results motivate the following two extensions and explorations described in Section 4.4 and 4.5.

4.4 Extension 1: Training a Model to Identify Exclusive Images

From the results described in Section 4.2 and specifically the precision recall graphs (Fig. 8c and Fig. 8d), it is evident that simply using the model output probabilities are not enough to make an accurate decision. In this section, we explore training a model to identify exclusive images.

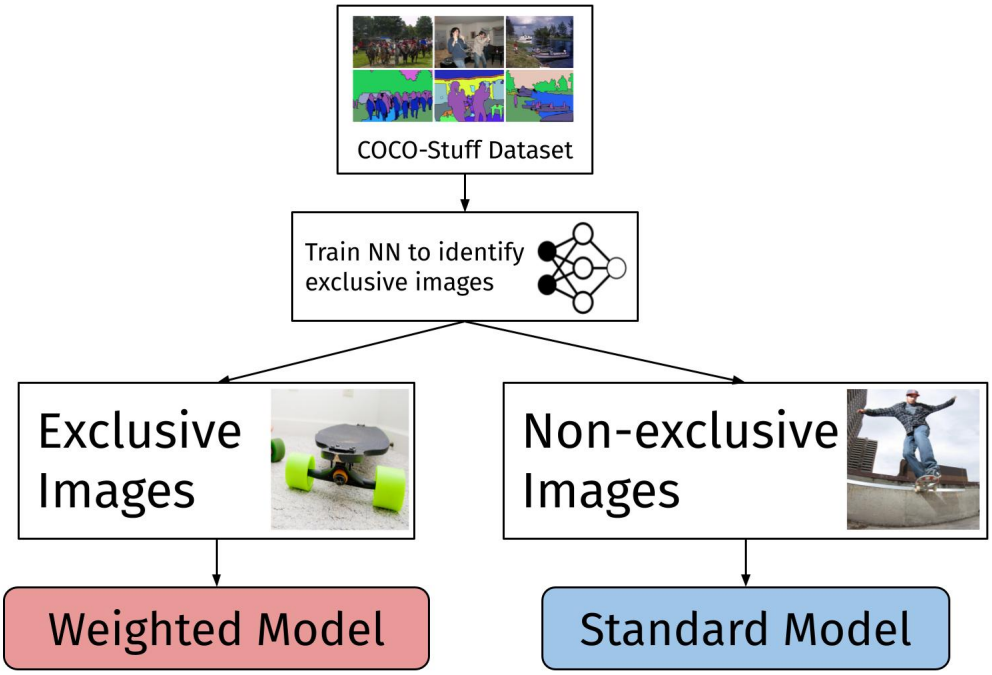


Figure 9: Training a NN to identify exclusive images for the Decision Rule Algorithm

Methods: To train a model to identify exclusive images, we input images from the 80 split training dataset and output two classes for each of the 20 biased category pairs where 0 corresponds to non-exclusive image and 1 corresponds to exclusive image with biased category b. We used ResNet-50 [7] pre-trained on ImageNet [8] as a backbone and optimized with stochastic gradient descent (SGD) with a momentumof 0.9, no weight decay, and a batch size of 200. It is optimized with an initial learning rate of 0.1 for 60 epochs and later dropped to 0.01 for 40 epochs fora total of 100 epochs. To provide labels to train on, we converted the 171-dimensional class label vector to a 40-dimensional label vector.

Given that there exists few instances of exclusive images, there exists a large class imbalance. We experiment with three reweighting techniques to solve the issue of class imbalance, fine-tune a *standard* model to leverage learned features, 3-way classification ((0) b1c0; (1) b1c1; (2) b0c0+b0c1), 4-way classification ((0) b1c0; (1) b1c1; (2) b0c0; (3) b0c1) and adjust the learning rate (1e-1, 1e-2, 1e-3, 1e-4). Given that there are two groups of images, exclusive and non-exclusive, the weight for each group is calculated. Given that N: total number of images, i: exclusive or non-exclusive class, $n[i]$: total number of images in class i, and $w[i]$:is

i's associated weight, we implemented with following three methods:

1. Class balancing: $w[i] = (1-\beta)/(1-\beta^n[i])$ where β is a hyperparameter ($\beta = 0.99999$ due to the high class count of non-exclusive images)
2. $w[i] = 1 - (\frac{n[i]}{N})$
3. $w[i] = \frac{N}{2n[i]}$

Results and Analysis: Without reweighting the classes, the precision of identifying exclusive images was near zero. When applying reweighting methods, the precision increased slightly to around 3% but does not approach the performance we desire. For example, when applying class balancing reweighting method (Table 5, the precision of identifying an exclusive image was 2.8% because even though 1982 truly exclusive images were classified as exclusive, 68481 truly non-exclusive images were also classified as exclusive. This means that the weighted model (which performs worse on non-exclusive images) will be evaluating many of these images, which is not what we desire as it will decrease the co-occur mAP. When adding additional classes (ie. 4 way classification of b0c0, b0c1, b1c0, b1c1), the precision increased, yet recall decreased. Building off of the standard model resulted in similar precision and recall values. Additionally, when identifying a biased category pair with fewer exclusive images examples (ie. ski and person) we saw that the precision and recall was zero.

	b1c0 (exclusive)	Non-exclusive
Exclusive ($n = 2952$)	1982	870
Non-exclusive ($n = 328188$)	68381	259807

(a) Confusion Matrix

	b1c0 (exclusive)	Non-exclusive
Precision	0.028	1
Recall	0.67	0.79
F1 Scores	0.05	0.88

(b) Precision, Recall, F1 Score

Table 5: Results from training a model to identify exclusive images with a class balancing reweighting method

The model likely struggled to identify exclusive images because identifying exclusive images is a more complex task than image classification. It requires the model to both

identify the images in the image and learn that this is an exclusive image. Additionally as seen from the competitive contextual bias mitigation methods in Section 2.5 (class-balancing loss model), contextual-bias can be formulated as a class imbalance problem, but we observe sub-optimal results with this method. Similarly when trying to mitigate this class imbalance issue when training a model to identify exclusive vs non-exclusive images, we demonstrate the difficulty in overcoming this class-imbalance issue as the precision of identifying exclusive images is low.

4.5 Extension 2: Identify Exclusive Images from CAMs

Class Activation Maps (CAMs) were first introduced by Zhou et. al [16] and can be applied to this contextual bias problem by using CAMs as a proxy for object localization information. For an image I and category b, CAM(I, b) indicates the discriminative image regions used by a deep network to identify b.

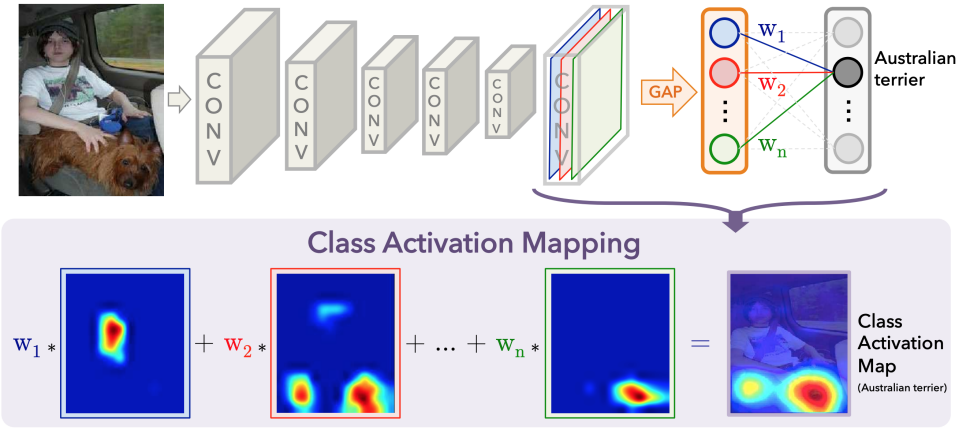


Figure 10: Class Activation Mapping. Figure adapted from [16]

.

As shown in Fig 10, the last convolution layer of the CNN provides a set of activation maps. Typically, these activation maps are averaged into a single scalar value, which are combined into a vector as input into the fully connected layer. For a particular class prediction, each of the weights connected to the scalars correspond to the convolutional units in the fully connected layer. These weights create a weighted sum of activation maps. Thus, the predicted class score can be mapped back to the previous convolutional layer to generate the

CAMs to provide a heatmap that indicates the region of the image that is most important to making the prediction of that particular class.

Our reproduced CAMs in Fig. 11 show that the standard model is "looking" at the wrong object (person) to make the skateboard prediction.

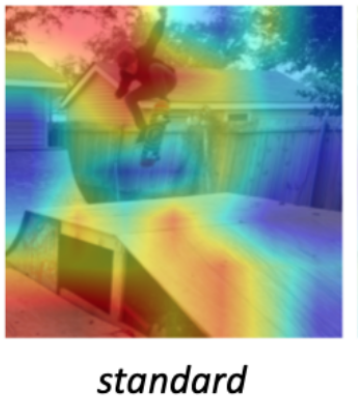


Figure 11: CAM from standard model of image containing "skateboard" and "person" class

.

Methods: CAMs can be applied in this contextual bias problem because we expect the standard model's CAMs looks at the co-occurring class when trying to identify the biased class the weighted model's CAMs looks at the biased-class when trying to identify the biased class. This means there will be a difference in the CAM maps. If there is a distinct difference between these two maps, this could be another indicator of an exclusive image.

We measure the L2 distance between the CAMs of the weighted model and standard model. Additionally, to measure the noise within the CAMs, we measure the L2 distance between the CAMs of two versions of the standard model.

To implement these CAMs, we follow the official CAM implementation: `https://github.com/zhoubolei/CAM`.

Results and Analysis: Overall, a more extensive analysis is necessary to determine if the CAMs are useful in distinguishing between exclusive and non-exclusive images, but we report our current results.

One key issue is that the CAMs are pretty noisy. While we expect that when identifying the biased class, the weighted model's CAMs should look at the biased-class while the

standard model should look at the co-occurring class, this is often not the case as depicted in Figure 12. We experiment with thresholding the bottom 90% of CAM values to isolate the most relevant sections and display preliminary results in Figure 12; however, this work is also still in progress. When analyzing the L2 distance distances between the standard and weighted CAMs, the values do not differ much from L2 distance between two standard models, which suggests that the weighted CAMs do not provide distinct results. However, this method is still currently being explored, tested, and refined.

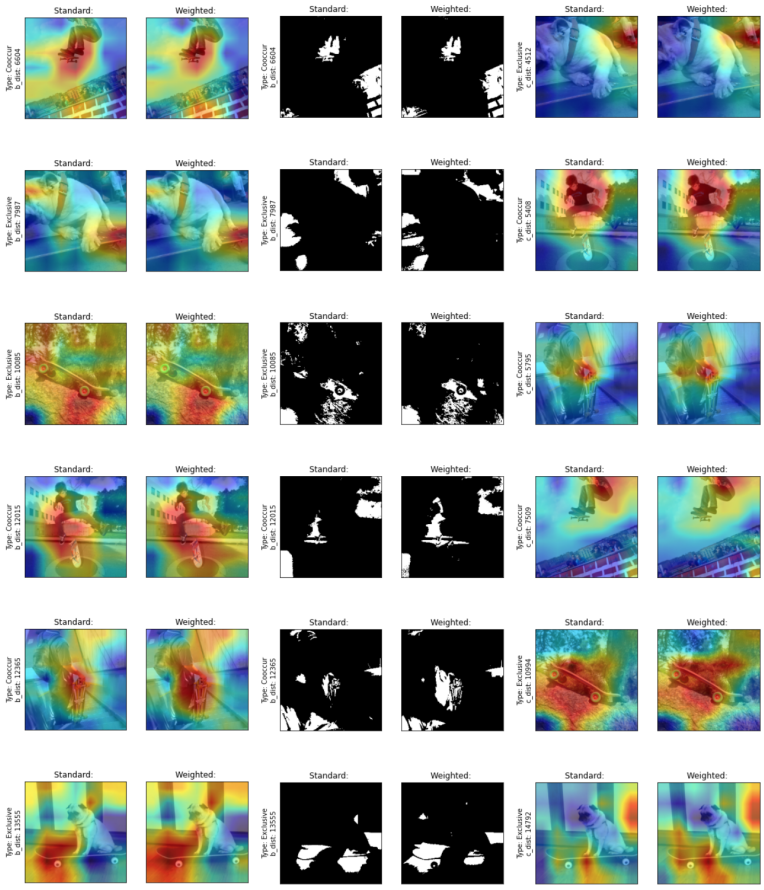


Figure 12: Standard and Weighted CAMs of 3 co-occur and 3 exclusive images on the right ordered based on increasing L2 Distance. The left two columns are standard and weighted CAMS on the biased class, the middle two columns are CAMs where the bottom 90% of pixel have been zeroed out, and the right two columns are the standard and weighted CAMs on the co-occurring class ordered by increasing distance in CAMs.

.

5 Conclusions and Future Work

In this report we describe the relevant results from reproducing Singh et. al's [5] method, specifically regarding the *standard* model, how to quantify contextual bias, identify biased category pairs, their proposed method, competitive contextual bias mitigation methods, and the reproduced results. Reproducing Singh et. al's [5] paper provided the results and models to set up the foundation for additional explorations, namely stacking, the decision rule algorithm, and the extensions upon the decision rule algorithm. Although the stacking method saw limited success, we present results from the decision rule algorithm that demonstrate strong potential in mitigating contextual bias. Specifically, when using ground truth labels to make the decision, the exclusive mAP increases by 14.4 while maintaining the co-occur performance. When the exclusive images are determined by the model's probability scores as it would be in a real-world application, the decision rule algorithm outperforms simple model averaging, but falls short when compared to [5]'s *feature-split* method as it can achieve an equivalent exclusive mAP, but has a -0.6 drop in co-occur mAP.

This decision rule algorithm has many exciting paths forward. One path is to further explore how to appropriately leverage CAMs within this problem by more thoroughly analyzing the difference bteween the CAMs. Additionally, it would be help to conduct a more extensive analysis to distinguish between the noise in the CAMs vs. relevant information. More specifically, we would like to investigate if there is a difference between the distance between two standard models and a standard and weighted model for exclusive vs. non-exclusive images.

Additionally, there are many other features that seem relevant and informative to this problem of identifying exclusive images. For example, a paper by Bahat & Shakhnarovich [17] propose a method to estimate confidence in the classifier's prediction by apply image transformations and analyzing the classifier's reaction to such perturbations. Since the *standard* model struggles to identify exclusive images, or has low confidence on these images, we draw inspiration from [17] by feeding in exclusive images to see how much each model reacts on the exclusive vs co-occur vs other images. Our hope is that on the exclusive images, the *weighted* model will change its probabilities as it is confident in its predictions

on exclusive images, while *standard* would not be as confident on these exclusive images. This would provide another method of improving the decision rule if we can get a confidence metric for our model's classification of objects in the images.

Appendix A Relevant Code

A.1 Reproduced Results & Project Repository

Github link to reproducing Singh et. al [5]: `https://github.com/princetonvisualai/ContextualBias`

Github link for other explorations outlined in this report: `https://github.com/nmeister/JuniorIW`

A.2 Decision Rule Algorithm

Decision Rule Algorithm

```
1
2 def decision_rule(name='decision rule (standard): ', decision_scores=
    standard_scores, alpha_min=0, alpha_max=1):
3   # Decide which score to used based on the ground-truth label
4
5   # Initialize with standard scores
6   alphas = np.arange(alpha_min, alpha_max, 0.01)
7   mAP_all, mAP_unbiased_all, mAP_exclusive_all, mAP_cooccur_all = [], [],
     [], []
8
9   for alpha in alphas:
10    print('alpha: ', alpha)
11    final_scores = copy.deepcopy(standard_scores) #fixed
12    name = name
13
14    # Loop over each of the 20 biased category pairs
15    for k in range(len(biased_classes_list)):
16        b = biased_classes_list[k]
17        c = biased_classes_mapped[b]
18
19        # Categorize the images into co-occur/exclusive/other
20        #cooccur = (decision_scores[:,b]>alpha) & (decision_scores[:,c] >
    alpha)
```

```
        # exclusive = ((weighted_scores[:,b])+ (1-standard_scores[:,c])) > alpha
        exclusive = (decision_scores[:,b]>alpha) & (decision_scores[:,c] < 1-alpha)
        #other = (~exclusive) & (~cooccur)

        # For exclusive images, replace standard scores with weighted scores
        for i in range(nsamples):
            if exclusive[i]: final_scores[i, b] = weighted_scores[i, b]


    all, unbiased, excl, co = calculate_results(name, final_scores)
    mAP_all.append(all)
    mAP_unbiased_all.append(unbiased)
    mAP_exclusive_all.append(excl)
    mAP_cooccur_all.append(co)
  return mAP_all, mAP_unbiased_all, mAP_exclusive_all, mAP_cooccur_all

mAP_all, mAP_unbiased_all, mAP_exclusive_all, mAP_cooccur_all = decision_rule(decision_scores=standard_scores)

mAP_all, mAP_unbiased_all, mAP_exclusive_all, mAP_cooccur_all = decision_rule(decision_scores=weighted_scores)
```

Listing 1: Relevant Decision Rule Algorithm Code

Ground Truth Decision Rule

```
# Decide which score to used based on the ground-truth label
amount_exclu = np.arange(0, 1, 0.01)
all_list, nonbiased_list, exclu_list, cooc_list = [], [], [], []
for amount_e in amount_exclu:
  print('amount_exclusive: ', amount_e)
  # Initialize with standard scores
  final_scores = copy.deepcopy(standard_scores)

  # Loop over each of the 20 biased category pairs
```

```
  count = 0
  for k in range(len(biased_classes_list)):
      b = biased_classes_list[k]
      c = biased_classes_mapped[b]

      # Categorize the images into co-occur/exclusive/other
      cooccur = (labels[:,b]==1) & (labels[:,c]==1)
      exclusive = (labels[:,b]==1) & (labels[:,c]==0)
      other = (~exclusive) & (~cooccur)
      # For exclusive images, replace standard scores with weighted scores
      for i in range(nsamples):
          if exclusive[i]:
              # prob of recording this exclusive example
              s = np.random.binomial(1, amount_e, 1)
              if s[0]==True:
                final_scores[i, b] = weighted_scores[i, b]
  name = 'gt decision'
  all, nonbiased, exclu, cooc = calculate_results(name, final_scores)
  all_list.append(all)
  nonbiased_list.append(nonbiased)
  exclu_list.append(exclu)
  cooc_list.append(cooc)
```

Listing 2: Relevant Decision Rule Algorithm Code

A.3 Evaluation

```
# final scores: probability scores (in decision rule algorithm, some rows
   have been replaces by the weighted model's results)
def calculate_results(name, final_scores):

    # Calculate mAP of all categories or the 60 non-biased categories
    APs = []
    for k in range(171):
        APs.append(average_precision_score(labels[:,k], final_scores[:,k])
   )
```

```
    mAP = np.nanmean(APs)
    mAP_nonbiased = np.nanmean([APs[i] for i in unbiased_classes_mapped])

    # Calculate exclusive/co-occur AP for each of the 20 biased categories
    exclusive_AP_list = []
    cooccur_AP_list = []
    biased_classes_list = list(biased_classes_mapped.keys())

    for k in range(len(biased_classes_list)):
        b = biased_classes_list[k]
        c = biased_classes_mapped[b]

        # Categorize the images into co-occur/exclusive/other
        cooccur = (labels[:,b]==1) & (labels[:,c]==1)
        exclusive = (labels[:,b]==1) & (labels[:,c]==0)
        other = (~exclusive) & (~cooccur)

        cooccur_AP = average_precision_score(labels[cooccur+other, b], final_scores[cooccur+other, b])
        exclusive_AP = average_precision_score(labels[exclusive+other, b], final_scores[exclusive+other, b])
        cooccur_AP_list.append(cooccur_AP)
        exclusive_AP_list.append(exclusive_AP)

    print('{}: all - {:.1f}, nonbiased - {:.1f}, exclusive - {:.1f}, cooccur - {:.1f}'.format(name, mAP*100, mAP_nonbiased*100,
                                                        np.mean(exclusive_AP_list)*100,
                                                        np.mean(cooccur_AP_list)*100))

    return mAP*100, mAP_nonbiased*100, np.mean(exclusive_AP_list)*100, np.mean(cooccur_AP_list)*100


print('All, Non-biased, Exclusive, Co-occur\n')
```

```
38
39 name = 'standard'
40 final_scores = copy.deepcopy(standard_scores)
41 calculate_results(name, final_scores)
42
43 name = 'weighted'
44 final_scores = copy.deepcopy(weighted_scores)
45 calculate_results(name, final_scores)
46
47 if split == 'validation':
48     name = 'featuresplit'
49     final_scores = copy.deepcopy(fs_scores)
50     calculate_results(name, final_scores)
51
52 name = 'average'
53 final_scores = copy.deepcopy((standard_scores+weighted_scores)/2)
54 calculate_results(name, final_scores)
55
56 name = 'max'
57 final_scores = copy.deepcopy(np.maximum(standard_scores, weighted_scores))
58 calculate_results(name, final_scores)
59
60 weight = 0.3
61 name = '{}*standard + {}*weighted'.format(weight, 1-weight)
62 final_scores = copy.deepcopy(weight*standard_scores + (1-weight)*
    weighted_scores)
63 calculate_results(name, final_scores)
```

References

[1] Tsung-Yi Lin et al. "Microsoft COCO: Common Objects in Context". In: *European Conference on Computer Vision (ECCV)*. 2014. URL: https://www.microsoft.com/en-us/research/publication/microsoft-coco-common-objects-in-context/.

[2] Antonio Torralba and Alexei A. Efros. "Unbiased look at dataset bias". In: *CVPR 2011*. 2011, pp. 1521–1528. DOI: 10.1109/CVPR.2011.5995347.

[3] Pedro F. Felzenszwalb et al. "Object Detection with Discriminatively Trained Part-Based Models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.9 (2010), pp. 1627–1645. DOI: 10.1109/TPAMI.2009.167.

[4] Aude Oliva and Antonio Torralba. "Torralba, A.: The role of context in object recognition. Trends Cogn. Sci. 11, 520-527". In: *Trends in cognitive sciences* 11 (Jan. 2008), pp. 520–7. DOI: 10.1016/j.tics.2007.09.009.

[5] Krishna Kumar Singh et al. "Don't Judge an Object by Its Context: Learning to Overcome Contextual Bias". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[6] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. "COCO-Stuff: Thing and Stuff Classes in Context". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[7] K. He et al. "Deep Residual Learning for Image Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016. DOI: 10.1109/CVPR.2016.90.

[8] Jia Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009.

[9] Jieyu Zhao et al. "Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints". In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2017.

[10] Yin Cui et al. "Class-Balanced Loss Based on Effective Number of Samples". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[11] Sunnie S. Y. Kim et al. "[Re] Don't Judge an Object by Its Context: Learning to Overcome Contextual Bias". In: *ML Reproducibility Challenge 2020*. 2021. URL: https://openreview.net/forum?id=PRXM8-O9PKd.

[12] Leo Breiman. "Stacked regressions". In: *Machine learning* 24.1 (1996), pp. 49–64.

[13] Cheng Ju, Aurélien Bibaut, and Mark van der Laan. "The relative performance of ensemble methods with deep convolutional neural networks for image classification". In: *Journal of Applied Statistics* 45.15 (2018). PMID: 31631918, pp. 2800–2818. DOI: 10.1080/02664763.2018.1441383. eprint: https://doi.org/10.1080/02664763.2018.1441383. URL: https://doi.org/10.1080/02664763.2018.1441383.

[14] David H. Wolpert. "Stacked Generalization". In: *Neural Networks* 5 (1992), pp. 241–259.

[15] Robert C Holte. "Very simple classification rules perform well on most commonly used datasets". In: *Machine learning* 11.1 (1993), pp. 63–90.

[16] Bolei Zhou et al. "Learning Deep Features for Discriminative Localization". In: *Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[17] Yuval Bahat and Gregory Shakhnarovich. "Classification Confidence Estimation with Test-Time Data-Augmentation". In: *CoRR* abs/2006.16705 (2020). arXiv: 2006.16705. URL: https://arxiv.org/abs/2006.16705.