– COS429 Final Project Report –

Interpretability of Pedestrian Detection

Indu Panigrahi and Raymond Liu

Abstract

Ensuring the robustness of detection systems employing computer vision models requires that we understand visual cues that these models rely on for pedestrian detection. This project explores the interpretability of pedestrian detection by using a Faster R-CNN and the Caltech Pedestrian Dataset. First, we use masking to determine that the model does depend on some form of visual cues. Then, we examine the the illumination of the pedestrians themselves as a visual cue and, as a complement, the effect of two potential non-pedestrian visual cues, sidewalks and crosswalks, on detection. Finally, based on the observed effects, we manipulate the weights in the training loss to reduce the model's dependence on sidewalks and crosswalks and achieve improved results.

1. Introduction

Within the past decade, convolutional neural networks (CNNs) have become a popular tool for computer vision problems such as pedestrian detection. However, CNNs are often treated as a "black box" in the sense that while they can perform a task well, we gain little insight into how the network internally approaches the task. This trend is especially true for deeper and more complex CNNs which have millions of parameters. Many automated vision systems rely on these complex CNNs to perform object detection. In general, a comprehensive understanding of how CNNs detect an object exposes a model's shortcomings and can point to methods to overcome those shortcomings. For pedestrian detection in particular, understanding how automated vision systems in self-driving cars detect pedestrians could help reduce automobile accidents and potentially save lives.

Our project targets this problem of interpretability, how a model learns from a dataset, in pedestrian detection. More specifically, we seek to understand what visual cues, if any, object detection models use to detect pedestrians. First, we determine whether or not the model relies on

any visual cues. Then, we examine the effect of illumination of pedestrians on detection. Finally, we examine the effect of the presence of two particular non-pedestrian objects: sidewalks and crosswalks. Based on our results from these experiments, we then introduce an improvement on the model by manipulating the training loss weights.

2. Related Work

A related study [4] focuses on the relation between pedestrian skin tones and detection and concludes that object detection systems in autonomous vehicles perform worse on pedestrians with darker skin tones than other pedestrians. Similarly, this project examines the effect of pedestrian illumination. We group pedestrians into categories of illumination just as the skin tone study groups pedestrians into categories of skin tone based on the Fitzpatrick scale [4]. However, rather than manually categorizing the pedestrians based on skin tone, we automatically categorize pedestrians based on the median pixel value within their bounding boxes. As a result, our categories incorporate illumination and clothing with skin tone.

Another study [3] builds and uses a dataset of images captured at night to investigate pedestrian illumination in a nighttime setting. This study concludes that detection of pedestrians during the night is more difficult than detection during the day. However, even pedestrians in daylight can be low-lit (Fig. 1), and our project explores the effect of pedestrian illumination in such settings.

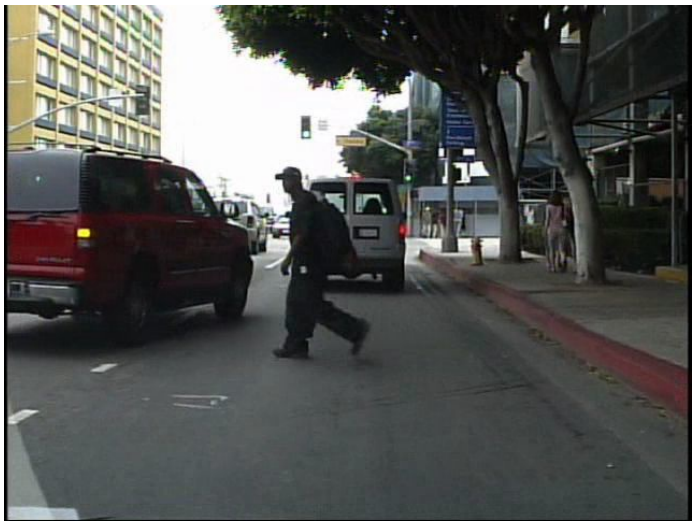

Figure 1: A low-lit pedestrian during the daytime.

3. Normalized Average Precision

In two of our experiments, we split the images in our dataset into categories based on a characteristic chosen for the experiment and run inference on each category. As a result, the total number of pedestrians varies between categories. This imbalance could invalidate a direct comparison of the average precisions across categories. Instead, we use normalized average precision (AP_N) [2].

For a given confidence level c, precision is typically calculated as shown in Eq. 1.

$$P(c) = \frac{R(c) \cdot N_j}{R(c) \cdot N_j + F(c)} \tag{1}$$

where N_j is the number of objects in class j (equivalently the number of pedestrians in a category), $R(c)$ is the recall (i.e., the fraction of all objects detected), and $F(c)$ is the number of false positives. However, for the same detection rate and false positive rate, the precision would be higher for larger N_j than for lower N_j. To mitigate this issue, we use normalized average precision (AP_N), which replaces N_j with a constant N to normalize the precision (Eq. 2) [2].

$$P(c) = \frac{R(c) \cdot N}{R(c) \cdot N + F(c)} \tag{2}$$

where N is the average N_j over all categories being compared (Eq. 3).

$$N = \frac{1}{n} \sum_{i=1}^{n} N_i \tag{3}$$

Categories of images with similar detection rates and false positive rates will have similar AP_N values.

4. Baseline Model

Before conducting our experiments, we fine-tuned a Faster R-CNN model [5] that had been pre-trained on COCO 2017 on the Caltech Pedestrian Dataset [1]. We used a Faster R-CNN because it is a popular object detection model. The full dataset contains about 250,000 frames of videos of

several drives through areas in the Los Angeles metropolitan area. However, due to storage and computational constraints, we reduced the dataset size by randomly sampling 1,100 frames. We then split the data into 700 training images, 200 validation images, and 200 test images.

For evaluation, we combine our validation and test images because we want to increase the number of images for evaluation without disrupting the proportional split and without exceeding our contraints. On the combined validation and test set, the model reaches a mean AP_N of 27.13, with a 95% confidence interval of [24.63, 29.63]. After inspecting the images in the dataset, we found a few ambiguous bounding boxes that only outlined cars that the model would likely, and rationally, not identify as pedestrians; these situations likely caused a lower average precision.

5. Determining the Existence of Visual Cues

5.1. Implementation

First, we determine if the model depends on any visual cues to detect pedestrians. These cues could be other objects in the image, such as sidewalks or crosswalks, or characteristics of the pedestrians themselves, such as illumination of the pedestrian. However, this experiment specifically determines the existence of visual cues; the later experiments (Sections 6 and 7) explore certain types of visual cues. Our implementation is as follows:

1. *Generate images with masked non-pedestrian pixels*

   For each image in our combined validation and test set, we set pixels that are not within the bounding box of any pedestrian to zero (Fig. 2).

2. *Run inference on masked images*

   We run inference on the masked set with our trained model and save the normalized average precision of each image for evaluation.

5.2. Evaluation and Analysis

Since we generate the masked images from the combined validation and test images, the combined validation and test set contains the same number of pedestrians as the masked set. That is, both sets

Figure 2: An image with non-pedestrian pixels masked out.

each contain 859 instances of pedestrians. Thus, to calculate the AP_N values, we use $N = 859$. This achieves the same effect as using non-normalized AP. On the masked images, the model reaches a mean AP_N of 12.93, with a 95% confidence interval of [10.69, 15.12].

We perform a two-sample t-test to determine if the sample mean of the AP_N values on the original images (i.e., baseline) is equal to the sample mean of the AP_N values on the masked images.

At the 5% significance level, there is a statistically significant difference in the mean AP_N values between the original and the masked images; the 95% confidence interval for the difference in means does not contain 0, and the p-value is lower than 0.05 (Table 1). Thus, we conclude that the model does indeed rely on visual cues to detect pedestrians.

T statistic	**DF**	**p-value**	**Difference in means**	**95% confidence interval**
8.314	788.557	4.023e-16	14.201	[10.848, 17.553]

Table 1: AP_N comparison between the combined validation and test images and the corresponding masked images.

6. Exploring the Effect of Pedestrian Illumination

6.1. Implementation

Now that we have determined that external visual cues do exist, we next examine the effect of the pedestrians themselves on the model's ability to detect them. Specifically, we examine the effect of illumination – how brightly lit the pedestrian is. Our implementation is as follows:

1. *Categorize pedestrians into illumination categories*

   We categorize each pedestrian based on a median pixel value that we calculate from a sub-region within the pedestrian bounding box. Specifically, we obtain each sub-region by vertically cropping the middle third of the box and then horizontally cropping the middle half (Fig. 3). Then, for each pedestrian we calculate the median pixel value of the cropped section. If the median pixel value is greater than 100, we place the image containing the pedestrian into the lower illumination category; otherwise, we place the image into the higher illumination category. Fig. 4 and Fig. 5 show example images with lower-lit and higher-lit pedestrians. If an image contains both a highly-illuminated pedestrian and a lowly-illuminated pedestrian, we discard the image because we want to compare the model's performance on distinct categories.

   We use sub-regions rather than the entire bounding boxes to account for a situation where a lower illumination pedestrian is present against a highly illuminated background (e.g., a pedestrian when the sun is behind them); we would want to place this pedestrian in the lower illumination category. In this sense, clothing and skin tone can contribute to illumination as well as sunlight.

2. *Run inference on each category*

   We run inference separately on each category with our trained model and save the normalized average precision of each image for evaluation.

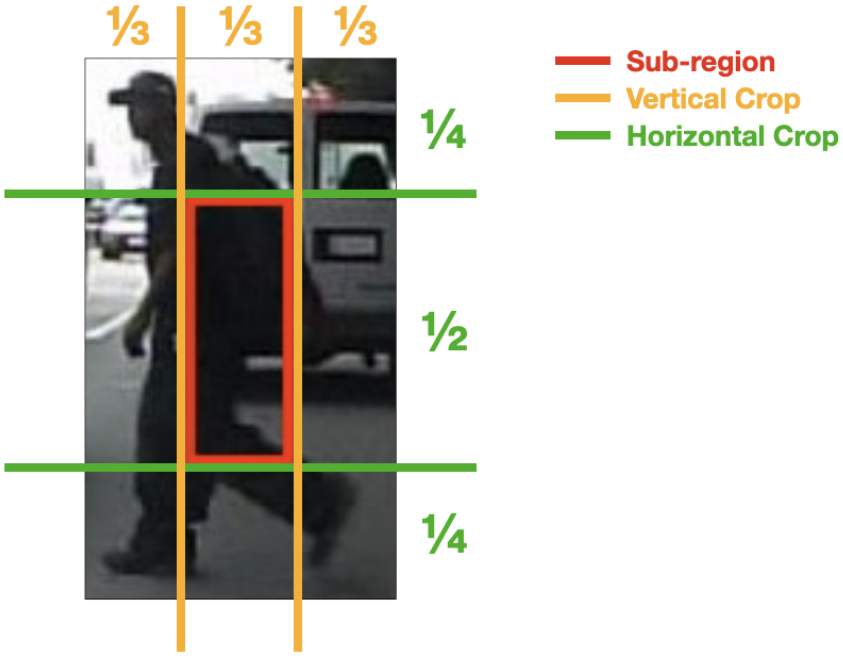

Figure 3: An example sub-region of a pedestrian bounding box.

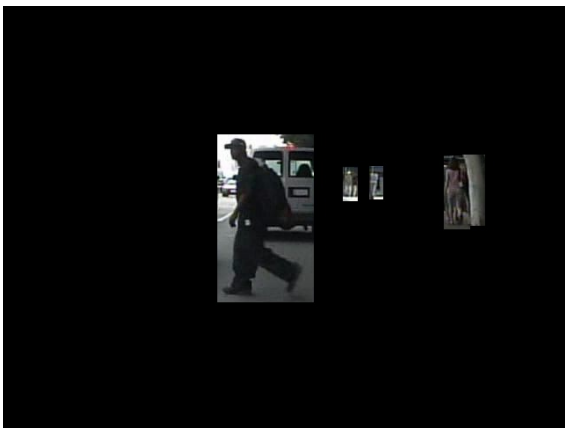

Figure 4: Lower-lit pedestrians.

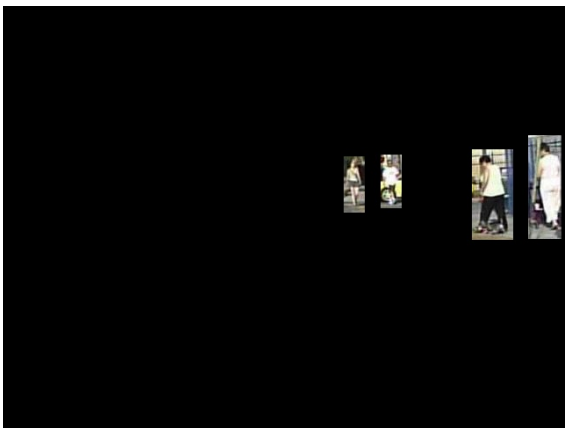

Figure 5: Higher-lit pedestrians.

6.2. Evaluation and Analysis

To calculate the AP_N values, we used $N = 519$ which is the average N_j over the two categories (Eq. 4).

$$N = \frac{1}{2}(N_{lower} + N_{higher}) \tag{4}$$

The mean AP_N for the images in the lower illumination category was 10.62, with a 95% confidence interval of [8.24, 13.00]. The mean AP_N for the images in the higher illumination category

was 12.29, with a 95% confidence interval of [9.19, 15.39].

We perform a two-sample t-test to determine if the sample mean of the AP_N values on the lower-illuminated images is equal to the sample mean of the AP_N values on the higher-illuminated images.

At the 5% significance level, there exists no statistically significant difference between the mean AP_N values of lower- and higher-lit pedestrians (Table 2). We theorize that this trend occurs because the model relies more on the strength of pedestrian outlines (i.e., contrast with background) than illumination.

T statistic	DF	p-value	Difference in means	95% confidence interval
-0.841	358.309	0.401	-1.667	[-5.565, 2.231]

Table 2: AP_N **comparison between images with lower- and higher-lit pedestrians.**

7. Exploring the Effect of Sidewalk and Crosswalk Presence

7.1. Implementation

As a complement to the intra-pedestrian visual cue examined in the previous experiment, we next examine the effect of two potential non-pedestrian visual cues: sidewalks and crosswalks. Our implementation is as follows:

1. *Categorize pedestrians into Sidewalk, Crosswalk, and None*

   We manually categorize the images into three categories: images with pedestrians on sidewalks, images with pedestrians on crosswalks, and images with pedestrians on neither sidewalks nor crosswalks (Fig. 6). We refer to an image from each category as a Sidewalk image, Crosswalk image, and None image respectively. If an image falls into more than one category, such as an image containing both a pedestrian on a sidewalk and a pedestrian on a crosswalk, we discard the image because we want to compare the model's performance on distinct categories.

   Furthermore, we categorize the training set and combined validation and test set separately. That is, we obtain six sets in total: Training Sidewalk images, Training Crosswalk images,

Training None images, Validation/Test Sidewalk images, Validation/Test Crosswalk images, and Validation/Test None images. In this experiment, we use the Validation/Test sets.

Figure 6: Sidewalk image (left), Crosswalk image (middle), None image (right).

2. *Run inference on each category*

   We run inference separately on each category with our trained model and save the normalized average precision of each image for evaluation.

7.2. Evaluation and Analysis

To calculate the AP_N values, we used $N = 248.67$ which is the average N_j over the three categories (Eq. 5).

$$N = \frac{1}{3}(N_{sidewalk} + N_{crosswalk} + N_{none}) \tag{5}$$

On Sidewalk images, Crosswalk images, and None images, the model reaches a mean AP_N of 44.54, 49.12, and 33.44, with 95% confidence intervals of [40.24, 48.82], [41.43, 56.80], [23.45, 43.43], respectively.

We perform a two-sample t-test to compare the sample mean of the AP_N values on the Sidewalk images, the sample mean of the AP_N values on the Crosswalk images, and the sample mean of the AP_N values on the None images.

At the 5% significance level, there exists no statistically significant difference in the mean AP_N values between Sidewalk images and Crosswalk images (Table 3). However, there does exist a statistically significant difference in the mean AP_N values between Sidewalk images and None images (Table 4), as well as between Crosswalk images and None images (Table 5).

Thus, we conclude that the presence of sidewalks and crosswalks has a significant effect on pedestrian detection.

T statistic	DF	p-value	Difference in means	95% confidence interval
-1.052	56.351	0.297	-4.580	[-13.300, 4.140]

Table 3: AP_N **comparison between Sidewalk images and Crosswalk images.**

T statistic	DF	p-value	Difference in means	95% confidence interval
2.561	45.835	0.014	15.675	[3.354, 27.996]

Table 4: AP_N **comparison between Sidewalk images and None images.**

T statistic	DF	p-value	Difference in means	95% confidence interval
2.101	31.522	0.044	11.095	[0.334, 21.855]

Table 5: AP_N **comparison between Crosswalk images and None images.**

8. Improvement on the Model: Manipulating the Training Loss

8.1. Implementation

Since the model performs significantly worse on images of pedestrians positioned on neither a sidewalk nor a crosswalk, we introduce a change in the training loss calculation such that if the model encounters such an image, the training loss weight for that image increases. If the model encounters any other image, the model uses the default weighting.

Specifically, we modify the code to scale the training loss by 1.5 when the model encounters a None image in the training set (previously referred to as Training None). We then re-train the Faster R-CNN model starting from the original COCO 2017 pre-trained state and run inference on the Sidewalk, Crosswalk, and None categories just as in Section 7.

Before training the model, we use the previously categorized Training None images to generate a text file of a list of the file names of Training None images. Then, the steps of our modifications during an iteration of training are as follows:

1. *Write current image name*

   At the beginning of the iteration, the model writes the name of the current image file into a text file.

2. *Check if Training None image*

   During the calculation of training loss for the current image, the model reads the current image name and the list of Training None image names from the corresponding text files. The model then checks if the list of Training None image names contains the current image name.

3. *If Training None image, scale the loss by 1.5*

   If the list of Training None image names contains the current image name, multiply the training loss and default weight by 1.5.

4. *Otherwise, leave the loss unchanged*

   If the list of Training None image names does not contain the current image name, leave the training loss weight unchanged.

In this modification, the weight becomes a hyperparameter that one can tune. We initially tried doubling the loss; however, we found that doing so causes the model to ultimately perform worse so we reduced the scale to 1.5.

8.2. Evaluation and Analysis

We perform a two-sample t-test identical to the one presented in section 7.

To calculate the AP_N values, we used $N = 248.67$ which is the average N_j over the three categories (Eq. 6).

$$N = \frac{1}{3}(N_{sidewalk} + N_{crosswalk} + N_{none}) \tag{6}$$

On Sidewalk images, Crosswalk images, and None images, the re-trained model reaches a mean AP_N of 43.71, 52.15, and 36.90, with 95% confidence intervals of [39.64, 47.78], [43.43, 60.87], [26.51, 47.28], respectively. These metrics demonstrate a decrease of 0.83 for Sidewalk, an increase of 3.03 for Crosswalk, and an increase of 3.46 for None.

We perform a two-sample t-test to compare the sample mean of the AP_N values on the Sidewalk

images, the sample mean of the AP_N values on the Crosswalk images, and the sample mean of the AP_N values on the None images.

At the 5% significance level, there still exists no statistically significant difference in the mean AP_N values between sidewalk images and crosswalk images (Table 6). Additionally, there still does exist a statistically significant difference in the mean AP_N values between Crosswalk images and None images (Table 7). However, there no longer exists a statistically significant difference in the mean AP_N values between Sidewalk images and None images (Table 8).

This reduction in the difference between the model's performance on Sidewalk images and None images suggests that the model became more familiar with pedestrians positioned on neither sidewalks nor crosswalks. Thus, we conclude that up-weighting None images in the training loss decreases the model's reliance on sidewalks as a visual cue for pedestrians.

T statistic	DF	p-value	Difference in means	95% confidence interval
-1.776	49.075	0.082	-8.440	[-17.988, 1.108]

Table 6: AP_N **comparison between Sidewalk images and Crosswalk images for the re-trained model.**

T statistic	DF	p-value	Difference in means	95% confidence interval
1.258	29.837	0.218	6.810	[-4.247, 17.866]

Table 7: AP_N **comparison between Sidewalk images and None images for the re-trained model.**

T statistic	DF	p-value	Difference in means	95% confidence interval
2.314	48.627	0.025	15.250	[2.002, 28.499]

Table 8: AP_N **comparison between Crosswalk images and None images for the re-trained model.**

9. Conclusion

In conclusion, our experiments show that the model does indeed depend on visual cues to detect pedestrians. Our modification to the training loss weights effectively reduced the model's reliance on sidewalks as a visual cue, and we believe that further hyperparameter tuning and increased storage and computational power could reduce the model's dependence on crosswalks as well.

Though we specifically experimented with a Faster R-CNN, our approach applies to any object detection model.

Before performing the experiments, we thought that masking the non-pedestrian portions of the images would worsen the performance of the model. Our analysis of inference on masked images suggests that this hypothesis is indeed true. Furthermore, we thought that the model would detect pedestrians positioned on a crosswalk or sidewalk more often than pedestrians standing on the middle of a road. Our comparison between inference on images of pedestrians on crosswalks, sidewalks, and neither crosswalks nor sidewalks suggests that this trend is true and reinforces one of the initial motivations for this project which is ensuring the robustness of a car's visual detection system despite a lack of crosswalks and sidewalks. Based on the study done on pedestrian skin tones, we thought that the model would detect highly-lit pedestrians better than low-lit pedestrians. However, the model performs similarly on pedestrians of either illumination level.

The main caveats to our implementation came from the quality of the dataset. Some images in the Caltech Pedestrian Dataset contain ambiguous bounding boxes – bounding boxes where the pedestrian is very difficult to detect, or missing entirely, in which case the pedestrian would be unreasonable to detect. For example, the rightmost bounding box in Fig. 7 contains part of a car rather than a distinguishable pedestrian. During inference, the model sometimes does not detect pedestrians within these ambiguous boxes which penalizes the AP during evaluation. One way to mitigate this penalty is by introducing a label named "ambiguous" for these types of boxes and having the evaluation ignore the presence or lack of the detection of bounding boxes labeled as "ambiguous". Furthermore, the images in this dataset came from drives "through neighborhoods in the greater Los Angeles metropolitan area chosen for their relatively high concentration of pedestrians" [1]. Thus, most, if not all, of the images depict paved roads and contain objects found in urban settings. However, even within the U.S., there exist dirt and farm roads that many people use on a daily basis, so models trained on this dataset may be biased against such situations. Therefore, it would be interesting to see if our conclusions translate to these types of settings.

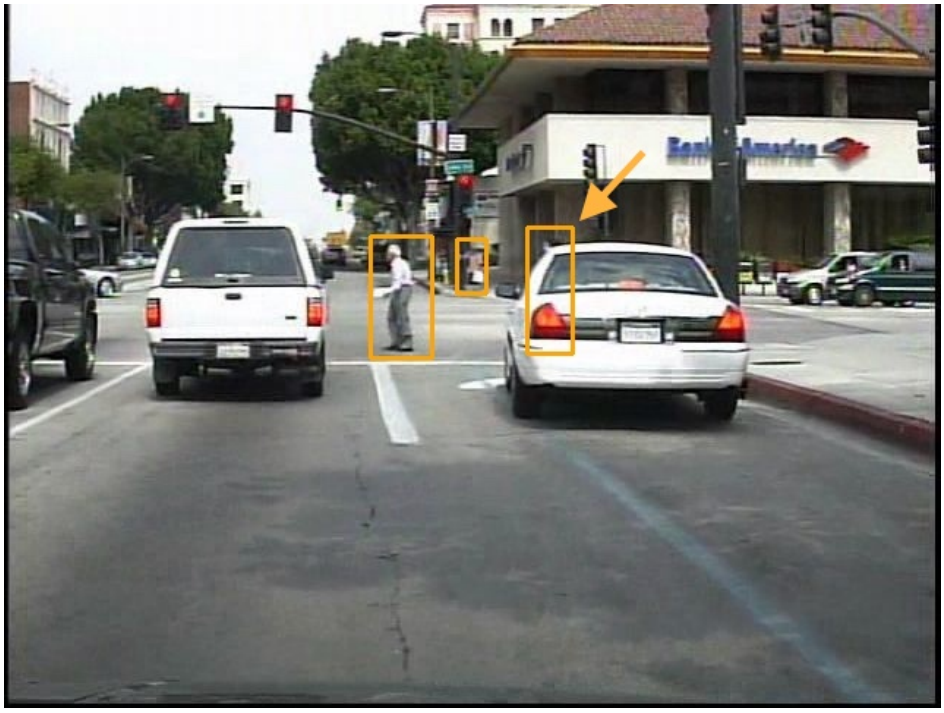

Figure 7: An ambiguous bounding box.

References

[1] P. Dollár *et al.*, "Pedestrian detection: An evaluation of the state of the art," *PAMI*, vol. 34, 2012.

[2] D. Hoiem, Y. Chodpathumwan, and Q. Dai, "Diagnosing error in object detectors," in *Computer Vision – ECCV 2012*, A. Fitzgibbon *et al.*, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 340–353.

[3] L. Neumann *et al.*, "Nightowls: A pedestrians at night dataset," in *Computer Vision – ACCV 2018*, C. V. Jawahar *et al.*, Eds. Cham: Springer International Publishing, 2019, pp. 691–705.

[4] B. Wilson, J. Hoffman, and J. Morgenstern, "Predictive inequity in object detection," 2019.

[5] Y. Wu *et al.*, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

Acknowledgements

We would like to thank Professor Olga Russakovsky and our graduate TA Sunnie Kim for advising this project. Most of the code for this project came from the referenced Detectron2 module, and other code was adapted from Indu's Fall Junior Independent Work.

Code for this project is available at:

https://github.com/ind1010/pedestrian_detection_interpretability