Performance of OpenCV HOG Model for Various Image Quality Changes

Soumik Sinha

Introduction

Pedestrian detection is a crucial task for computer vision specifically in the application of self-driving cars and road safety. Accurate and robust pedestrian detection models are essential for ensuring the safety and efficiency of a range of applications such as autonomous vehicles and surveillance systems. However, real-world situations often provide challenging conditions. One example is image quality. Variations in image resolution, lighting, blurring, and contrast can pose a threat to the safety of these systems. This research paper aims to investigate the robustness and adaptability of a computer vision model in detecting pedestrians.

The paper focuses on the Histogram of Oriented Gradients (HOG) algorithm, a widely-used technique for pedestrian detection, implemented in the OpenCV computer vision library [1, 2]. The HOG is a computer vision technique used to understand the shapes and textures in images. It does this by analyzing and finding the directions of changes in image brightness, essentially by looking at how the pixel values change from one region to another. HOG divides an image into small parts, calculates the orientations of these brightness changes in each part, and then creates a unique description of these patterns which can be used to capture information about the shapes and textures within the image.

In this research, I explore how well a pedestrian detection model based on HOG, as implemented in OpenCV, performs under different image quality conditions. By varying image quality factors like brightness, contrast, and blur, this paper aims to understand how robust the model is in scenarios where image conditions may not be ideal.

Methods

The data I use is from Kaggle containing pictures, diagrams, and traffic [3]. Two sets of pictures are there: pictures with pedestrians and pictures without them. I have found 40 images that would work best for the project. The data is not very clean as it had various stock images of cars and pedestrians and also had some images taken from a higher height meaning that the angle of the picture wouldn't be accurate for autonomous cars. Additionally in the "no pedestrian" data set, I have found images containing people on scooters and bikes however those would be considered pedestrians in my project. I find better images throughout the dataset that look as if they came from a self-driving car camera with real traffic and pedestrians on the road. I also make sure that all of the images I choose have similar image qualities in terms of brightness, contrast, and blur so that I could accurately test the influence of those factors on the model.

As mentioned before I am using the OpenCV package for the HOG algorithm. Additionally, I am using it to alter the characteristics of the images in the data. The HOG algorithm detects the pedestrians themselves and outputs the regions. To check if there are any pedestrians, I use an if statement to check if there are regions outputted by the HOG model.

To check the performance of the HOG model when altering image brightness, contrast, etc. I take a baseline accuracy measurement. This is done by looping through the images in the data, having the HOG model run on the image, and checking whether it correctly or incorrectly guessed the presence of a pedestrian.
I find that the baseline accuracy for the HOG model is just found by calculating the percent of images that the model guesses correctly out of the 20 images with pedestrians and the 20 without.

Discussion

The expectation of the model performance when altering the data is that the HOG model will perform worse. With different levels of brightness, contrast, and blur the model would be less able to detect pedestrians as it wasn't specifically trained on them. However, I expect the model wouldn't be too far from the baseline accuracy and may only get a few images wrong.

1. Baseline

The OpenCV HOG model performs fairly well for images that haven't been altered. It correctly classifies 35 of the 40 images, giving it an accuracy of 0.875. The model performs very well for the images without pedestrians as it gets all 20 of them correct. It never mistakes anything else for a pedestrian. However, it doesn't do as well as detecting pedestrians in the images which is much more important. It correctly identifies 15 of the 20 images as having a pedestrian.


Both of the above images are examples of when the model correctly identifies and highlights a pedestrian on the street.


However, these 2 images show the limitations of the model as it does not detect any pedestrians here.

2. Brightness Comparison

The accuracy of the model changes after the brightness is both increased and decreased. When the brightness is increased the model's accuracy drops to 0.65 when classifying the images with pedestrians and 1.0 when classifying images without them. The model is able to correctly detect pedestrians in many of the bright images as it did detect the unaltered ones except for a few. For example in the following images, for the one on the left (unaltered) the model correctly sees a pedestrian; however it doesn't for the one on the right (increased brightness). This may be because the pedestrian blends into the road too much when the rightness is increased.

The accuracy after lowering the image brightness drops to 0.5 for images with pedestrians and 1.0 for images without. For example, in the following images, the pedestrians are undetected by the model when the brightness of the image is lower.

In general, it seems like not altering the brightness is better for the model. The model works slightly better with higher brightness than lower brightness overall; however I notice an image where the opposite is true. On the right is the brightened image which the model failed to detect; however, when the brightness is decreased, the model works better. This may be because the original image is right and lowering the brightness allows the person to blend it less with the background.

3. Contrast Comparison

The accuracy of the model, when the contrast is increased, is 0.7 for images with pedestrians and 0.9 for images without pedestrians. The accuracy of the model, when the contrast is decreased, is also 0.7 for images with pedestrians and 1.0 for images without pedestrians. While the lower contrast generally performs better than the higher contrast, there are cases where higher contrast does better than lower contrast. The following example shows a higher contrast image (left) in which the model works better than the low contrast image (right). This is an image for which the baseline model isn't able to detect that there is a pedestrian.

However, adding more contrast also results in the model detecting pedestrians in images where there aren't any like in the example below. This may be because the higher contrast makes it so that the model mistakes the traces and edges that it finds for people.


4. Blur

As expected the blurry images result in a decreased performance of the model. The model had an accuracy of 0.6 for pedestrians and 0.95 for those without pedestrians for the blurred images. While I do get quite a few of them correct, it misses some that the baseline gets. Below is an example of this. Baseline (left) found the pedestrian, while blurry (left) did not.


However, it would be more important to guess the images with pedestrians correctly. There is much more danger in getting images with pedestrians incorrect than getting the images without them incorrect. More importance should be placed on getting a higher accuracy for just the pedestrian images.

In general altering the images results in a lower performance. In situations with the real world, situations like bad weather, or night images would result in a lower performance. However, on a case-by-case basis, some images do better when altered. For example, increasing contrast helps in some images, perhaps because the HOG model is able to detect the changes in pixel intensity easier. To improve the model, changes could be made to the images. For example, if there is a dark background, the model might benefit by increasing the brightness or vice versa. If the image has a very high contrast and mistakes cars and other objects for people, it would be better to lower the image contrast.

In real-world scenarios, there is a possibility that the change in image quality will improve from the baseline model and can be used in such algorithms to improve the performance of those models.

Conclusion

How does the OpenCV HOG model perform with changes to the image quality, specifically its brightness, contrast, and blur, when detected by pedestrians on the road? In general, the accuracy of altering those image characteristics decreases the performance of the model. The model performs better without any alterations. However, in some cases, it can improve the model.

Works Cited

1. Mallick, Satya. "Histogram of Oriented Gradients Explained Using Opencv." *LearnOpenCV*, 30 Nov. 2021, learnopencv.com/histogram-of-oriented-gradients/.
2. "Home." *OpenCV*, 27 Sept. 2023, opencv.org/.
3. Tejasvagarwal. "Pedestrian No Pedestrian." *Kaggle*, 9 Dec. 2017, www.kaggle.com/datasets/tejasvdante/pedestrian-no-pedestrian.