Pedestrian Detection with Convolutional Neural Networks

Joshua Powell

Abstract

The ability of convolutional neural networks to extract features from a raw image, has made them a popular method for image processing, and especially for classification tasks. CNNs will be used in this paper in the pedestrian detection task, that is, identifying the location and size of pedestrians in images by proposing bounding boxes around pedestrians. The CNN approach to pedestrian detection will be compared to a popular non-CNN approach, which uses histograms of oriented gradients (HOG) instead of neural networks. The expectation is that the CNN approach will be able to detect pedestrians with higher accuracy than the HOG approach due to the ability of CNNs to learn many layers of pedestrian features. Several test data sets of images from the CalTech pedestrian detection benchmark data set were created and used to evaluate the accuracy of both approaches. Due to a greater percentage of pedestrians detected and a greater percentage of correct proposals, it was determined that the CNN approach was superior to the HOG approach. The average of all trials for the CNN approach was 59% of pedestrians detected and 25% of correct bounding boxes. That is significantly better than 29% of pedestrians detected and 20% of correct bounding boxes for the HOG detection.

1 Introduction

Pedestrian detection is a computer vision task with the objective of determining the location and size of pedestrians in an image. It is an important task in many industries, especially the self-driving car industry. It is essential for self-driving cars to be able to locate pedestrians and stop or intelligently navigate around them.

Convolutional neural networks (CNNs) are perhaps the most popular method for doing image processing. The idea behind CNNs is that they use convolution and pooling layers in order to extract features from objects. In this case, the objects are pedestrians. Using supervised learning, a CNN can learn the features of pedestrians from training data. Then the CNN will be able to determine whether these same features are present in input images that have not seen before. By systematically determining the presence or absence of pedestrians in many pieces of input images using a CNN, it will be possible, to find the location and size of pedestrians.

Region-based convolutional neural networks (RCNNs) is a method that greatly improved accuracy in the object detection task. RCNNs consist of three steps. In this first step, region proposals are generated using selective search. Then a CNN is used for feature extraction. These features are classified using support vector machines [6].

This paper gives a method of solving the pedestrian detection task similar to the RCNN method. Instead of selective search, region proposals are generated using an exhaustive search in which a window size will be defined. For a given input image, each possible subimage of the size of this window will be passed through a CNN for feature extraction. The softmax activation function will be used for the classification.

The CNN approach this paper implements will be contrasted with a non-CNN approach that utilizes built-in functions of the OpenCV library. More specifically, a pretrained histogram of oriented gradients (HOG) descriptor is used instead of a CNN in order to determine features of pedestrians. A support vector machine is used as a classifier to propose bounding boxes [4]. This approach is implemented on the pyimagesearch website [2].

The data set that will be used is the CalTech pedestrian detection benchmark data set from [1]. It is a very large data set of video streams taken from a car travelling through an urban area. Pedestrians in the video stream vary greatly in size and are often not fully visible in the image. This makes pedestrians difficult to detect.

A comparison will be done between the CNN approach and HOG approach. Due to the wide variety of images in the CalTech data set, a CNN trained with the data will generalize well to many pedestrians in many different environments and lighting situations. Therefore the expectation is that the CNN approach will perform significantly better than the HOG approach on the CalTech data set because of the ability of CNNs to learn features in many different scenarios.

Additionally, a scheme for comparing the accuracy of the bounding box proposals from both the CNN and HOG methods is required. Given an input image, the output of both pedestrian detection techniques is the input image with bounding box proposals. There will also be ground-truth bounding boxes, or hand labelled bounding boxes from the CalTech data set. Many studies suggest using the intersection over union method [5]. The intersection over union of a given bounding box proposal and the ground-truth bounding box is computed by dividing the area of overlap by the area of union. When the intersection over union value is high, the bounding box proposal is a sufficient bounding box.

This paper will use a simplified intersection over union evaluation metric. If there is any overlap between bounding box proposals and ground-truth bounding boxes, the bounding box proposal will be considered a correct bounding box.

2 Pedestrian Detection

2.1 Detector Implementation

Given an image, a classifier determines whether a pedestrian is present within an image or not and does not give any information about the location of the pedestrian. The classification task can be done using CNNs and supervised learning. By learning the features of pedestrians in a training set consisting of CalTech pedestrian images, a CNN can extract features of input images that have not been seen before and determine if the features match up with its learned pedestrian features.

The goal of pedestrian detection is to recognize where all pedestrians are within an image. This was done by using a CNN to classify whether certain subimages are pedestrians and proposing bounding boxes wherever pedestrians are present in the input image. The overall structure of the detection is shown in Figure 1.

A window size, a horizontal stride value, and a vertical stride value were defined. The window was initialized in the top left corner of the input image and the subimage within the bounding box was saved. The window was vertically translated by the pixel distance defined by the vertical stride and the subimage the window is over is saved. This process is repeated. When the window reaches the bottom of the image, the window is moved to the top of the image and horizontally translated by the pixel distance defined by the horizontal stride. The window is then translated downward while saving subimages as before. This process is repeated until the window reaches the bottom right corner of the input image.

The subimages were then all propagated through the CNN classifier. For each subimage, if the classifier determines the subimage to be a pedestrian, a bounding box is drawn over the area of the subimage in the input image. By the end of the detection process, the input image will have bounding box proposals drawn on it.

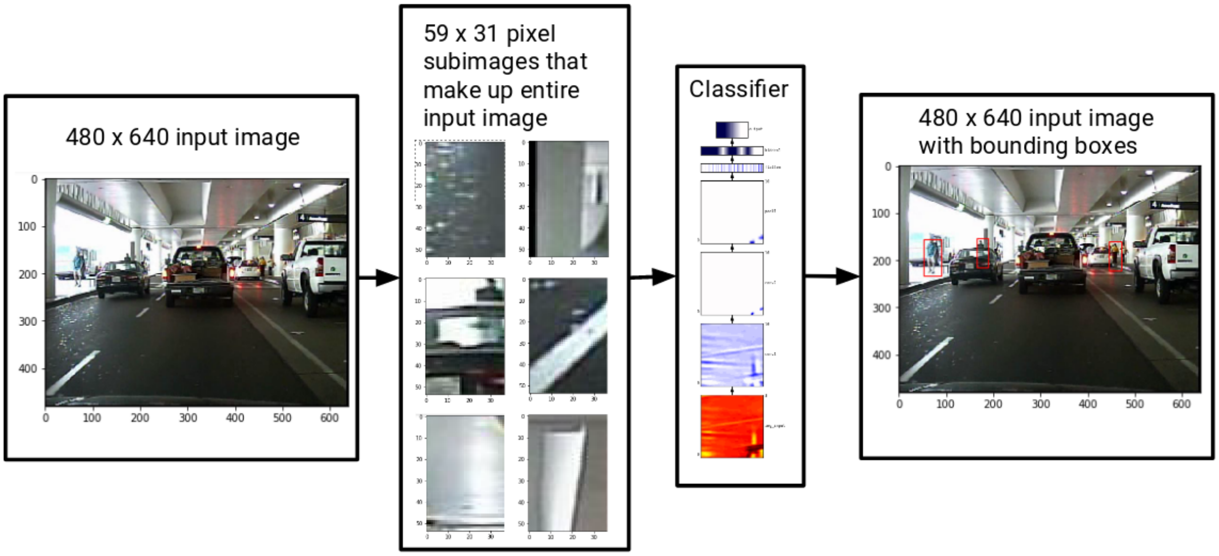
480 x 640 input image
0
100
200
300
400
500
600
59 x 31 pixel subimages that make up entire input image
Classifier
480 x 640 input image with bounding boxes

Figure 1: The structure of the detection is shown. The input is a 640 x 480 image. Subimages of all possible bounding boxes of size 59 x 31 (the average size of ground truth bounding boxes in the CalTech data set) are extracted from the input image and passed into a CNN pedestrian classifier. When the classifier classifies a certain subimage as a pedestrian, a bounding box will be drawn on the input image. The input image with bounding box proposals is outputted.

2.2 Evaluation Method

In theory, if the classifier is precise enough, bounding box proposals will only be drawn over pedestrians. However, the classifier will not be perfect so some method for determining if the bounding box was correctly placed in the image must be created. More specifically, it is important to know if the bounding box proposal is a false positive, that is, a bounding box is proposed over a region with no pedestrian. Conversely, it is also important to know if a bounding box was not proposed over a pedestrian.

After the detection has been run on an image with pedestrians, the image will have some bounding box proposals. There will also be the bounding boxes from the annotation file, which will be assumed to be accurate bounding boxes. The annotation file bounding boxes will be called the ground-truth bounding boxes while the bounding boxes from the detection will be called bounding box proposals.

The bounding box proposals may be larger or smaller than the ground-truth bounding boxes.

If a bounding box proposal correctly detects the position of a pedestrian, there will be some overlap between the bounding box proposal and the ground truth-bounding box. Therefore whenever there is overlap between a bounding box proposal and the ground truth-bounding box, the bounding box proposal will be determined to be a correct detection.

3 Experiments

3.1 CalTech Pedestrian Benchmark Data Set

The Penn-Fudan data set is made up of images of students walking around the University of Pennsylvania campus [3]. For the most part, the images are well lit, the people in the images are large in the frame, and are completely visible as shown by Figure 4 (in the appendix). These aspects make it very easy to use a CNN in order to extract the features of people in the images in the Penn-Fudan data set.

The CalTech pedestrian benchmark data set is a series of video streams collected from a camera on a car as it travelled through urban areas in Los Angeles. There are about ten hours of 640x480 30Hz video taken. The frames of the video streams, which are simply images, were extracted from the video streams. There are about 250,000 frames available. The data set also comes with an annotation file, which gives information about the location of bounding boxes around all pedestrians within each frame of the video streams. There are about 350,000 bounding boxes and 2,300 unique pedestrians [1].

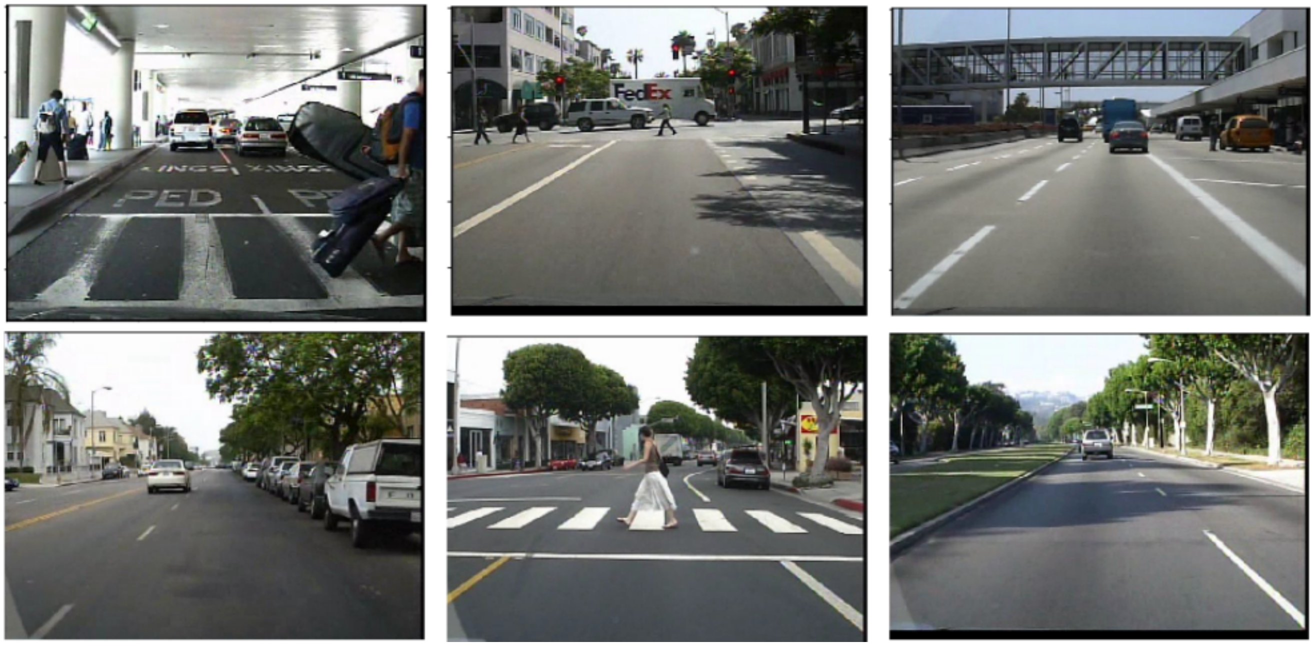

Figure 2: Images from the CalTech pedestrian detection data set. Pedestrians are often difficult to detect as they greatly vary in size and are often not fully visible.

In comparison to the Penn-Fudan data set, the CalTech data set is very challenging. Figure 2 shows some examples of images in the CalTech data set. The size of pedestrians varies greatly. Some are relatively large while others can barely be seen by the human eye. Additionally, many people are not fully visible. The top left image of Figure 2 includes a person that is only half visible. The lighting also varies significantly as the top left image also has very bright lighting in the background while having dimmer lighting in the foreground. These challenges make it difficult for the classifier CNN to learn the features of pedestrians. While a classifier CNN that is trained on the Penn-Fudan data set may be able to learn the hands, head, face, etc. of people, these features may not be visible in the CalTech data set images. Therefore the classifier CNN will have to learn other features in addition to the typical features of people.

The CalTech data set has some pedestrians that are easy to classify, due to the fact that the are large in the image and are completely visible, and many much harder pedestrians to classify. This means that using this data set in a supervised learning task will yield a classifier CNN that will generalize better than a classifier CNN trained with a data set like the Penn Fudan data set.

3.2 Building and Training the Pedestrian Classifier

The classifier is a CNN that takes in an input image and determines whether the image contains a pedestrian or not. There is one input node, the image, and two output nodes. The first output node signifies the absence of a pedestrian while the second signifies the presence of a pedestrian. When the classifier has high confidence that a pedestrian is present in the input image, the first output node will have a value close to 0 and the second output node will have a value close to 1. The CNN was given a simple topology that included two convolution layers, both of size 10, one max pooling layer, and one hidden layer with 10 nodes, in addition to the input layer and output layer. The network topology is summarized in Table 1.

Parameter	**Setting**
Input nodes	1
Num conv layers	2
Num conv layer size	10
Num pool layers	1
Num hidden layers	1
Num hidden nodes	10
Num output nodes	2

Table 1: Parameters of classifier CNN. There were 6119 images in the training set and 1529 images in the testing set.

The classifier CNN weights were obtained using supervised learning. Positive images are images that the classifier should classify as people. The subimage contained within one bounding box from each of 3824 images made up the positive images. In other words, the positive images were the subimages that only contain pedestrians. Since bounding boxes vary in size, the positive images were resized to 64x64 so they could be fed into the CNN. There were also 3824 negative images. Negative images were obtained from collecting one random 64x64 subimage from each of 3824 images from the CalTech data set. It was assumed that, since the pedestrians in an image take up a very small amount of space, by randomly taking 64x64 subimages, these images would not be pedestrians. In total, there were 7648 images, which were combined and shuffled. 80% of the images were used in training and 20% were used in testing (or 6119 images in the training set and 1529 images in the testing set).

3.3 Detection with the CNN Classifier

Four separate data sets each of 100 random images from the Caltech data set were collected. The average ground-truth bounding box dimensions of the Caltech data set was found. This average ground-truth bounding box has dimensions 59 vertical pixels by 31 horizontal pixels. This average bounding box was used as the window that was systematically dragged across each input image. The horizontal and vertical stride values were set to half of the corresponding dimensions of this window. For each input image, a subimage was collected at each step of the window traversal. This totals 731 subimages per input image. The subimages are passed into the classifier and bounding boxes were proposed wherever the classifier determines there to be a pedestrian with high confidence. The bounding box proposals were compared to the ground-truth bounding box, using the evaluation method described before. Any bounding box proposals that had any overlap with ground-truth bounding boxes were considered correct detections. Any bounding box proposals with no overlap with ground-truth bounding boxes were considered false positives.

3.4 Detection with Histogram of Oriented Gradients

The same four data sets of 100 random images each were used as in the above section was used. The HOG method for pedestrian detection was run on each of these images. The bounding box proposals from this method were also evaluated in the same way as the CNN method detector.

4 Results

4.1 Classifier Results

The classifier was able to successfully extract pedestrian features from the CalTech data set. Figure 3 shows two sets of features that were extracted from two images.

Table 2 summarizes the accuracies of the classifier CNN. The training accuracy, validation accuracy, and testing accuracy were all very high. The classifier accuracies were not 100% so there will still be some pedestrians that are not classified correctly. It is important to note that in the CNN detection, there will be 731 images passed into the classifier per images. Therefore there will be a significant amount of false positives and pedestrians that do not get detected.

Accuracy	**Percentage**
Training Accuracy	98%
Validation Accuracy	96%
Testing Accuracy	98%

Table 2: Evaluation of the classifier CNN. 7648 images from the CalTech data set were used to build the classifier with 6119 images in the training set and 1529 images in the testing set.

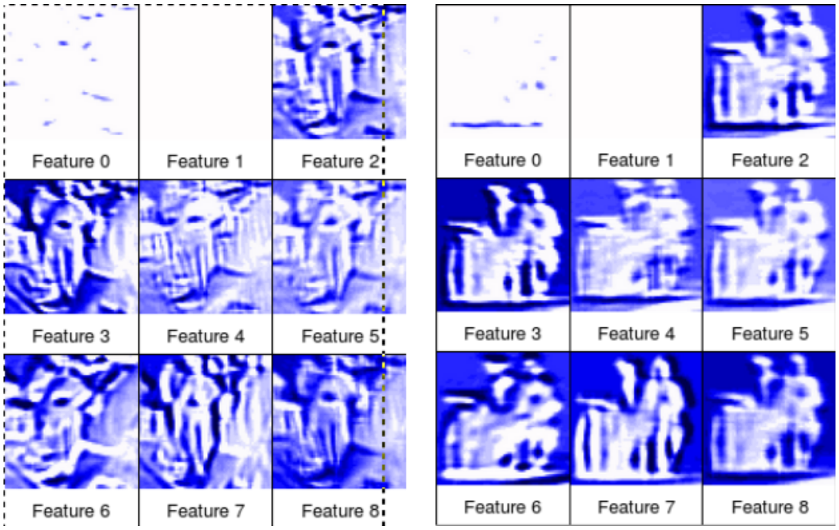

Figure 3: Features extracted from two images from the CalTech data set.

4.2 CNN Pedestrian Detection

One window was used to systematically traverse through the image and collect subimages to be passed into the classifier. This window was 59 pixels by 31 pixels, which was the average size of ground-truth bounding boxes in the CalTech data set. The majority of pedestrians were detected as shown in Table 3. However, there were many false positives and a low percentage of correct proposals.

Performance Characteristic	**Trial 1**	**Trial 2**	**Trial 3**	**Trial 4**	**Average**
% of Pedestrians Detected	65	45	52	73	59
% of Correct Proposals	27	22	23	29	25
Number of False Positives Per Image	3.1	3.6	3.1	3.4	3.2

Table 3: The results of the CNN detection on the CalTech pedestrian data set.

4.3 Histogram of Oriented Gradients Detection

The HOG detection was not able to classify many pedestrians, especially those that were too small, shown by Table 4. This means that a low percentage of pedestrians were detected. While the number of false positives was low, the number of correct proposals was also low.

5 Discussion

The hypothesis of this paper was that the CNN detection method would perform better than the HOG detection method on the diverse CalTech pedestrian detection data set. The CNN method was able to detect a majority of pedestrians. However, there were many false positives, which led

Performance Characteristic	**Trial 1**	**Trial 2**	**Trial 3**	**Trial 4**	**Average**
% of Pedestrians Detected	37	22	24	34	29
% of Correct Proposals	26	18	19	19	20
Number of False Positives Per Image	0.28	0.28	0.25	0.46	0.32

Table 4: The results of the HOG detection on the CalTech pedestrian data set. It is important to note that HOG detection can detect pedestrians that are large in the image. However, it works very poorly on pedestrians that are small on the image and usually does not recognize them.

to a low percentage of correct proposals. For a given input image, horizontal and vertical stride values, and window size, the CNN method checks all possible subimages, making it an exhaustive search. In fact, 731 subimages were classified for every input image. While the classifier accuracies were high, including a 98% testing accuracy, the classifier still had many false positives and was not able to detect some pedestrians, especially since so many subimages are propagated through the classifier.

The HOG method detected less than half of the pedestrians in all trials. The CalTech data set consists of pedestrians of a wide variety of sizes and many of the pedestrians are small in the image. The HOG method works very well on pedestrians that are large. However, it does not generalize well to small pedestrians and will often not detect them. The number of false positives per image was low for this method. However, since so many pedestrians are missed, especially those that are small, the percentage of correct proposals is low.

The percentage of pedestrians detected and the percentage of correct proposals both indicate that the CNN method is superior to the HOG method. For the CNN method, the average percentage of pedestrians detected was 59% and the average percentage of correct proposals was 25%. For the HOG method, the corresponding values are 29% and 20%.

The nature of the exhaustive search of the CNN method causes the average number of false positives to be high per image, 3.2. While the HOG method is not able to generalize to detect small pedestrians so misses many pedestrians, it also does not propose many false positives so the number of false positives per image is low, 0.29, for the HOG method.

There are some further improvements that could be implemented in order to make the CNN method even better than the HOG method. The classifier could be improved. Although the validation and testing accuracies were high, they still allowed for a significant amount of false positives and missed pedestrians because so many subimages were propagated through the classifier for each input image. The training set was of size 6119. A larger training set could yield better results. Also, the topology of classifier was very simple. A more complex topology, obtained by varying the hyperparameters of the CNN, could give higher validation and testing accuracies.

Additionally, only one window size was used to traverse the input images. This means that the bounding box proposals could only be one size. The window was sufficient to accurately classify many pedestrians because the dimensions were the average ground-truth bounding box dimensions of the CalTech data set. Yet, when pedestrians were much larger or much smaller than this bounding box, detections did not occur with high accuracy. For example, if a pedestrian is large in an image and only an arm is visible in the window, the CNN may not be able to classify the arm as a human.

It is important to note that the exhaustive CNN method used in this paper was very slow. It took about twenty seconds to detect all pedestrians in an image. Real time pedestrian detection is required in practical applications, such as self driving cars. Future work will explore methods of increasing speed while also increasing detection accuracy.

6 Acknowledgements

I would like to thank Lisa Meeden for her advice in completing this paper.

References

[1] Caltech pedestrian detection benchmark. www.vision.caltech.edu/Image$_Datasets/CaltechPedestrians/$.

[2] Pedestrian detection opencv. www.pyimagesearch.com/2015/11/09/pedestrian-detection-opencv/.

[3] Penn-fudan databace for pedestrian detection and segmentation. //www.cis.upenn.edu/ jshi/ped$_html/$.

[4] Navneet Dalal and Bill Triggs. Histogram of oriented gradients for human detection. *IEEE Computer Society*, 1(886-893), 2005.

[5] Jordan M. Witte Melanie Mitchell Max H. Quinn, Erik Conser. Active grounding of visual situations. *Unpublished draft.*

[6] Yangxin Zhong Peng Yuan, Yang Yuan. Faster r-cnn with region proposal refinement.

7 Appendix

Penn-Fudan Data Set Examples The Penn-Fudan data set contains images of pedestrian from scenes around the University of Pennsylvania campus [3]. Some images are included below:

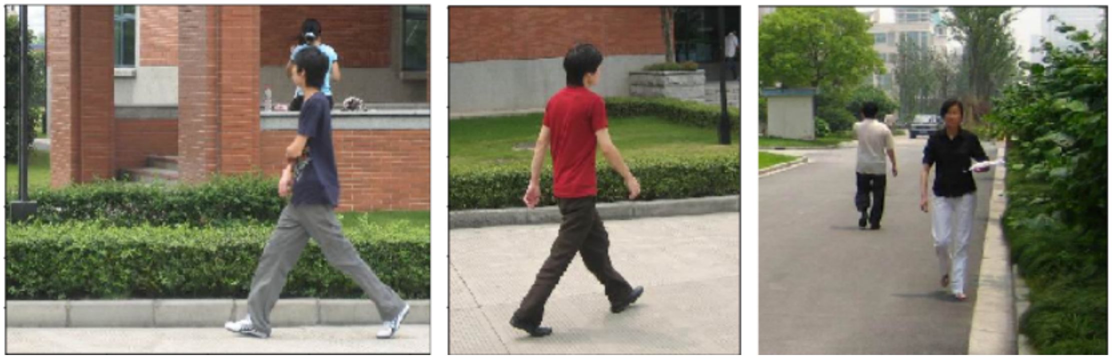

Figure 4: Images from the Penn-Fudan pedestrian detection data set.