Pedestrian Detection with RCNN

Matthew Chen
Department of Computer Science
Stanford University
mcc17@stanford.edu

Abstract

In this paper we evaluate the effectiveness of using a Region-based Convolutional Neural Network approach to the problem of pedestrian detection. Our dataset is composed of manually annotated video sequences from the ETH vision lab. Using selective search as our proposal method, we evaluate the performance of several neural network architectures as well as a baseline logistic regression unit. We find that the best result was split between using the AlexNet architecture with weights pre-trained on ImageNet as well as a variant of this network trained from scratch.

1 Introduction

Pedestrian tracking has numerous applications from autonomous vehicles to surveillance. Traditionally many detection systems were based off of hand tuned features before being fed into a learning algorithm. Here we take advantage of recent work in Convolutional Neural Networks to pose the problem as a classification and localization task.

In particular we will explore the use of Region based Convolutional Neural Networks. The process starts with separating the video into frames which will be processed individually. For each frame we generate class independent proposal boxes, in our case with a method called selective search. Then we train a deep neural network classifier to classify the proposals as either pedestrian or background.

2 Related Work

This paper many follows an approach known as Regions Convolutional Neural Networks introduced in [6]. This method tackles the problem classifying and localizing objects in an image by running detection on a series of proposal boxes. These proposal boxes are generally precomputed offline using low level class independent segmentation methods such as selective search [12], though recent work has incorporated this process into the neural network pipeline [11]. Given the proposals, a deep convolutional neural network is trained to generate features which are fed into class specific SVM classifiers. This approach as proved successful in localizing a large class of items for the PASCAL VOC challenge.

For the architecture of our CNN we test a baseline logistic method and compare it to results from implementations of well known CNNs. These CNNs include Cifarnet which was developed in [9] for the cifar-10 dataset and Alexnet which won the 2012 Imagenet challenge [10]. Additonally we look at the effect of using pretrained

Figure 1: Original image on top left. Positive selective search bounding boxes on top right. Warped background and pedestrian image on bottom left and right respectively

weights for Alexnet and fine tuning the last layer.

The eth pedestrian tracking dataset was established through through a sequence of papers [4] [3] [5]. These papers use additional information collected including stereo vision and odometry data as additional sources of information for their models. We are only using monocular camera data for each of the collected video sequences.

3 Dataset and Features

The dataset is composed of several sequences of videos produced by a camera on a moving platform. Each frame has hand labelled annotations denoting bounding boxes of pedestrians. Overall there are seven sequences of videos with a combined 4,534 frames. The data was split into a training and test set where we have the frames from five sequences (3,105 frames) in the training set and two sequences (1,429 frames) in the test set as shown in Table 1. The annotations are not complete in that they do not strictly label all pedestrians in a given image. It is usually the case that only pedestrians which take up a certain subjective threshold of the screen are labelled. This leads to what could be some false negatives in the training set.

Statistic	Test	Train
Num Images	774	4952
Avg Pedestrians	7	7
Avg Proposals	2278	2953
Pos Proposals	230	111
Neg Proposals	2048	2842

Table 1: Data statistics split up by training and test sets

Only a subset of the data was used. For the two sequences in the test set, the annotations were sparse in that they were recorded only on every fourth frame. Thus we included only the frames which annotations existed. Additionally, for training, we set the number of proposals used proportional to the number of positive proposals. Specifically we set the ratio at 1:2 positive pedestrian images to negative background images.

4 Methods

The complete pipeline from video frame to bounding box output is shown in Figure 2. We start with a given video sequence and split it up by frames. Then we run a algorithm to generate proposal bounding boxes, in this case we use selective search [12], which we cache for use across the process. We pass these bounding boxes along with the original image to the detector which is our convolutional neural network. The CNN

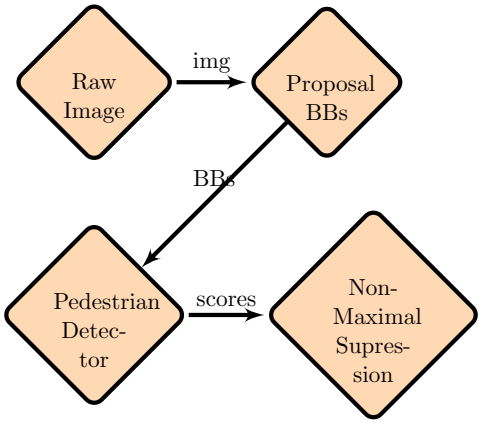

Figure 2: Pedestrian detection pipeline

produces softmax scores for each bounding box which are used in the final non-maximal suppression step.

4.1 Proposal Boxes

We tested two different proposal box method which were popular in the literature [8]. The edge box method [14] actually performed better in terms of the average intersection over union (IOU) across all images in the set as shown in Figure 3. However we opted to use selective search as it proposed fewer boxes and hence increased the runtime of our algorithm. We started by precomputing proposal boxes for all frames in the dataset. Then we created an image processor to preprocess and warp these proposals, which varied in size and aspect ratio, into a fixed size to input into our neural network. The preprocessing involved mean subtraction and whitening for each frame. Using these sub-sampled images we trained a convolutional neural network on top of this data to classify a proposal as either background or pedestrian.

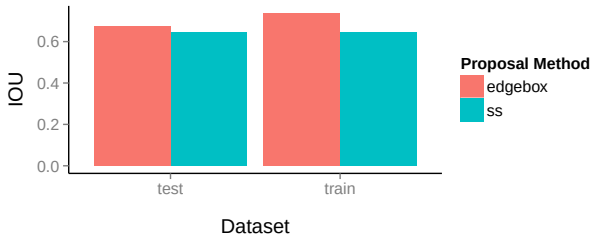

Figure 3: Comparison of Selective Search and EdgeBox proposal algorithms

Given the proposals and ground truth bounding boxes we could then generate a training set where the proposals were labeled based on their overlap with the ground truth images. We used a threshold of 0.3 so that images which had at least this overlap with a given ground truth bounding box were considered positive examples and the rest negative.

4.2 Detector

We start by baselining our results relative to a logistic regression network that we train on our images. Additonally we experiment with various neural network architectures and measure their performance on our task. We start with using the Cifarnet architecture which takes image at a 32x32x3 scale [9].

For the alexnet_pretrained architecture we maintained the exact architecture specified by the initial paper with the exception of the last layer which was replaced by a softmax function with two outputs initialized with random weights [10].The softmax function generates what can be interpreted as the probability for each class and is defined as follows.

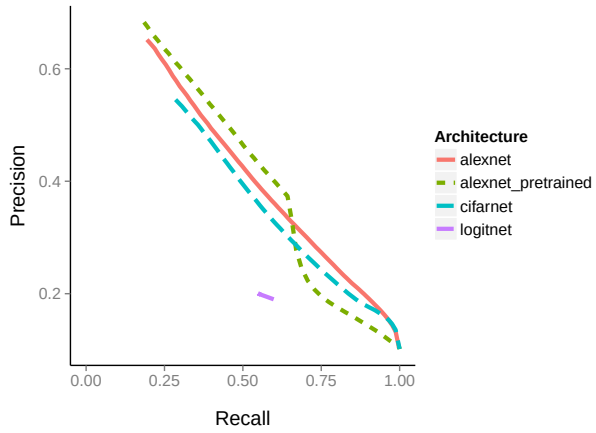

Figure 4: Precision Recall curves for various methods on test set

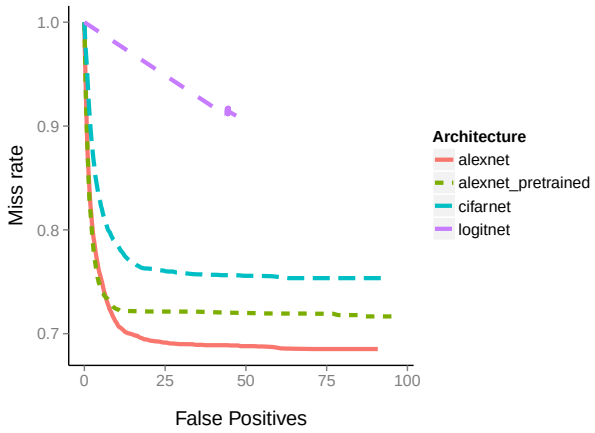

Figure 5: Miss rate to false positives by method

$$p(y = i \mid x; \theta) = \frac{e^{\theta_i^T x}}{\sum_{j=1}^{k} e^{\theta_j^T x}}$$

Next implement a simplified variant of the alexnet architecture in which we remove group convolution and local response normalization layers. We modify the middle convolutional layers maintain full spacial depth form the previous layers . Doing so simplifies the implementation as grouping was used primarily due to memory constraints from training on two separate GPUs in the original implementation. These networks were built using the tensorflow library [1] and trained running on multiple CPUs in parallel (the exact number of CPUs and specifications varied as this was run across multiple servers).

5 Results

After training our net for 4000 iterations, in which each iteration was composed a random proportional sub-sample of 50 images across the entire dataset, we tested each net on the complete test set. Figure 4 shows a comparison of the precision and recall curves of each net when adjusting threshold for classifying the image as pedestrian as opposed to background. The precision and recall is calculated on the test set using the same pedestrian overlap threshold to denote positive examples for proposal boxes.

For AlexNet with pre-trained weights, all the weights besides the fully connected layers and the last convolutional layer were frozen at initialization for fine tuning. All other networks were trained from scratch with weights initialized from a truncated normal distribution with standard deviation proportional to input size.

We find that Alexnet with pretrained weights from Imagenet performs the best in moderate prediction score thresholds while a slightly larger variant of Alexnet, trained from scratch, performs better at higher acceptance thresholds. The Cifarnet performance is not that far behind, which is interesting as it as order of magnitude

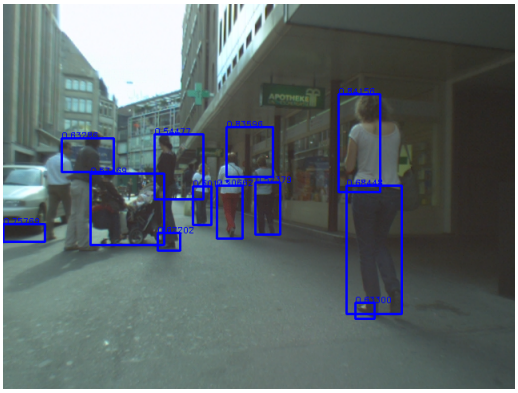

Figure 6: Example final output of our algorithm

fewer parameters and takes input images of size 32x32x3 compared to 227x227x3 for alexnet.

For the non-maximal supression step we aimed for a minimal threshold for our bounding boxes and used 0.05. We then did a final evaluation of our algorithm by looking at the miss rate to false positive curve shown in Figure 5. We can see that our best performance still has a 70 percent miss rate. An example of the final output for a given frame is shown in Figure 6.

6 Discussion

We find that AlexNet with pre-trained weights from Imagenet and fine tuning of the last layers performs the best on a majority of the threshold levels. Overall our approach still has a miss rate that is too high for many real world applications. This rate can be due to several factors. First is the proposal box method. As we have shown the best bounding boxes only had a mean IOU ratio of around 0.6 which would serve as an upper bound on the accuracy of our overall system. Next it is likely the case that our neural networks could have benefited from additional training time to reach convergence as the number of iterations we used was relatively small compared to the amount of time it took to train on imagenet and other large vision benchmarks. Additional hyperparameter tuning of parameters such as regularization on fully connected layers would also likely improve the results.

The use of Region-based convolutional neural networks for the task of pedestrian detection does show promise. Our example output image shows the technique produces reasonable bounding boxes. However additional work needs to be done to improve the overall performance of the system.

7 Future Work

The main constraint of the work presented in this paper was lack of GPU support for running these models due to resource contraints. Most CNN training is currently implemented using one or multiple GPUs which should have a order of magnitude speed up. Training on a GPU would allow for more iterations through the dataset to increase the change of convergence as well as runtime comparisons to current RCNN results in other domains. Additionally the added computational power would enable us to use a larger pedestrian dataset such as [2]. Training on a large dataset such as this one would allow the nets to generalize better. This is especially true for the larger network architectures which require larger training sets corresponding to the larger number of parameters to tune.

References

[1] Martın Abadi et al. "TensorFlow: Large-scale machine learning on heterogeneous systems, 2015". In: *Software available from tensorflow. org* ().

[2] Piotr Dollar et al. "Pedestrian detection: An evaluation of the state of the art". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34.4 (2012), pp. 743–761.

[3] A. Ess et al. "A Mobile Vision System for Robust Multi-Person Tracking". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*. IEEE Press, 2008.

[4] Andreas Ess, Bastian Leibe, and Luc Van Gool. "Depth and appearance for mobile scene analysis". In: *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE. 2007, pp. 1–8.

[5] Andreas Ess et al. "Moving obstacle detection in highly dynamic scenes". In: *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE. 2009, pp. 56–63.

[6] Ross Girshick. "Fast R-CNN". In: *International Conference on Computer Vision (ICCV)*. 2015.

[7] Ross Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE. 2014, pp. 580–587.

[8] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. "How good are detection proposals, really?" In: *arXiv preprint arXiv:1406.6962* (2014).

[9] Alex Krizhevsky and Geoffrey Hinton. *Learning multiple layers of features from tiny images*. 2009.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[11] Shaoqing Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems (NIPS)*. 2015.

[12] Koen EA Van de Sande et al. "Segmentation as selective search for object recognition". In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE. 2011, pp. 1879–1886.

[13] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. "The NumPy array: a structure for efficient numerical computation". In: *Computing in Science & Engineering* 13.2 (2011), pp. 22–30.

[14] C Lawrence Zitnick and Piotr Dollár. "Edge boxes: Locating object proposals from edges". In: *Computer Vision–ECCV 2014*. Springer, 2014, pp. 391–405.