Pedestrian Detection using Convolutional Neural Networks

Karthik Kumarasubramanian
University of Texas at Arlington
Arlington, Texas
karthik.kumarasubramani@mavs.uta.edu

Abstract

Pedestrian Detection in real time has become an interesting and a challenging problem lately. With the advent of autonomous vehicles and intelligent traffic monitoring systems, more time and money are being invested into detecting and locating pedestrians for their safety and towards achieving complete autonomy in vehicles. For the task of pedestrian detection, Convolutional Neural Networks have been very promising over the past decade. ConvNets have a typical feed-forward structure and they share many properties with the visual system of the human brain. This project hopes to make a detailed explanation on the Overfeat architecture for pedestrian detection. The detection is done on Benchmark dataset - TUD-Crossing Brussels Dataset.

1. Introduction

Object detection is something that scientists have been teaching machines to perfect over the past few decades. Detecting people, especially pedestrians is one critical sub problem in object detection and it is a separate area of research in its own, which has grown popular over the years due its varied applications.

1.1. Pedestrian Detection

Pedestrian Detection is being used in a variety of fields such as Autonomous Driving, Traffic Monitoring Systems, Intelligent Surveillance, etc. There is a lot of research being done on detecting pedestrians through machine vision approaches, and this task is challenging due to a variety of reasons like occlusions in pedestrian data, fast moving pedestrians. Quite a few applications are inseparable from the pedestrian detection technology, such as the intelligent surveillance system and the autopilot system. Despite the great improvements in accuracy, the task of pedestrian detection is still a great challenge with various difficulties that requires more meticulous design and optimization. Over the past few decades, pedestrian detection methods have adopted a variety of different measures. Some of the methods are aimed at increasing the speed of detecting. On the contrary, the other methods have focused on the accuracy. Especially, Convolutional Neural Networks (CNN) have appeared as the state-of-the-art technology in the accuracy of a host of computer vision tasks. And methods based on the deep learning usually precede the previous traditional ones by a wide margin in the comprehensive performance.

When the deep network is employed in the task of pedestrian detection, a host of measures have analogous computation pipelines. For the most of the detection frameworks, they usually proceed in two phases. In the first stage, utilizing the original image in the pixel level, they are designed to extract the high-level spatial properties or the high-level features in order to gain some regions of interest. Then, the features of those regions are fed into a classifier or several classifiers that judge if such a region describes a pedestrian.

1.2. Convolutional Neural Network

Convolutional Neural Networks (CNNs or ConvNets) are a type of neural networks that has been proven very effective in a variety of applications such as object detection and recognition, image classification and several Natural Language Processing (NLP) applications. ConvNets have been successful in recognizing faces, people, traffic signs, and have also been powering self driving cars, robots, etc. There are four main operations that are critical to a typical ConvNet, namely Convolution, Non-linearity, Pooling or Sub sampling and Classification. These operations are achieved using three fundamental layers in the network architecture. They are the Convolutional Layer, the Pooling Layer and the Fully Connected layer.

1.3. Convolutional Layer

Intuitively, the Convolutional Layer does the convolution operation which is nothing but extracting features from the input image, called as feature maps. This operation preserves the spatial information in the image pixels by learning features using filters. A filter or a kernel in ConvNets is like a small window or a matrix that is slided over the input image. During this sliding process, the filters compute the

dot product of the corresponding image pixels and produce a feature map or an activation map. The filters act as feature detectors from the input image and different values of feature matrices will produce different feature maps.

1.4. Pooling Layer

The pooling layer (or the sub-sampling layer) reduces the dimensionality of the generated feature maps. This can be done in several ways. Consider a window on the feature map. One way is to sum all the values of the feature map in the window. Other ways are to take the max (called Max Pooling) or the average (called Average Pooling) of the values of the feature that come in the window. This way it can be ensured that the most important information in the feature maps is retained.

1.5. Fully Connected Layer

The fully connected layer is nothing but a traditional Multi-Layer Perceptron with an activation function like Softmax, SVM, etc. This layer in most of the cases does the classification part. The output from the convolutional and the pooling layers represent very high level features from the input image and this layer uses those features for classifying categories specified in the training data. ConvNets have been most useful in image based applications due to their effective representation of image data and the ability to learn image features with great accuracy and precision

1.6. Goal of the project

The primary goal of the project is to present an architecture of ConvNets to evaluate their performance on this particular task. The task is performed on TUD-Crossing dataset which contains 1019 frames with 508 frames labelled pedestrians.

2. Convolution Neural Network for Pedestrian Detection

The fundamental idea behind ConvNets is to learn complex features from pixel-level contents, capitalizing on a sequence of operations such as filtering, normalization, pooling, etc. the pipeline of ConvNets in pedestrian detection is very much in line with that of object detection i.e., the pipeline starts from raw image from which the proposal of candidate regions are proposed, then higher level representations are extracted to be applied pixel-to pixel. Then these extracted features are fed to a classifier which estimates if the extracted region depicts a pedestrian.

2.1. ConvNets for Object Detection

ConvNets are used for image based recognition are (i) Feature extraction and classification are integrated into a single structure and they are totally adaptive, (ii) ConvNets extract 2D image features at high scales and (iii) It is resistant to geometric and local distortions in an image. ConvNets largely rely on huge training datasets, training procedure based on backpropagation with optimization algorithms like gradient descent, etc. In the context of detecting pedestrians, the last layer typically contains a single neuron that acts as a classifier that determines if the input region contains a pedestrian or not. In spite of these models being powerful, they need large datasets with annotations to yield accurate results.

2.2. Overfeat for Pedestrian Detection

This paper presents three fundamental ideas for handling objects of various sizes, shapes and positions in an image,

1. To apply ConvNets in various portions of the image, at various scales in a sliding window fashion.

2. To train the system to produce a prediction of location and size of bounding boxes corresponding to an object relative to the window.

3. To accumulate the evidence for each category at every location and size.

2.2.1 Classification

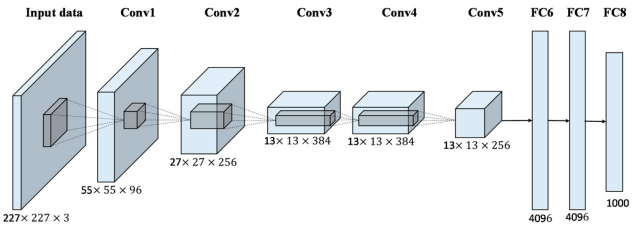

Figure 1. AlexNet Architecture.

In the classification task, each image is assigned a label pertaining to the main object in the image. The architecture is similar to that of AlexNet - it has eight layers. The first five are convolutional layers and the remaining are the fully connected layers. During training, the model uses fixed sized inputs like AlexNet and the output of the last fully connected layer is fed to a thousand way softmax.The weights are randomly initialized and then updated using Stochastic Gradient Descent (SGD). Then a dropout of 0.4 is employed on the fully connected layers. Finally, while testing, the image is entirely explored by running the network densely at different locations and scales.

2.2.2 Localization

After the classification, the bounding box of each classified object is returned along with a confidence value with respect to the ground truth. This confidence value must be larger

than 0.5 for the bounding box to qualify. For localization, the previous model is modified by replacing the classifier with a regression network. Then the regression predictions are combined with the classification results in all the locations. The regression network is trained with an L2 loss after which the individual predictions are merged together according a greedy strategy.

2.2.3 Detection

Detection training is very much similar to the classification training except that this is done in a spatial manner. One main advantage in the detection part is that multiple locations of the image can be trained simultaneously and the weights can be shared between these locations. The primary difference between the localization and the detection tasks is the necessity to predict a background class when no object is present

3. Experiment and Results

3.1. Experiment Setup

The experiment was carried out on Windows 10 machine with an NVIDIA GeForce GTX 1050 GPU. The experiment is expected to be programmed in Python 3.7 and Tensorflow 2.0.

3.2. Dataset

The TUD–Crossing Dataset - has about 1019 frames of urban driving data with a resolution of 640 x 480 pixels with 2000 fully visible and partially occluded pedestrians labelled in the frames. The annotations are in the Interface Definition Language (*.idl) format having bounding boxes around each pedestrian in the form (xmin,ymin,xmax,ymax) denoting the starting and ending (x,y) coordinates of the bounding box. The train, test and evaluation data was separated manually in 70 - 30 fashion.

3.3. Results

The networks were trained with an overall dropout of 0.5 and with Adam optimizer. The ConvNet approach Overfeat gives an AUC measure of 0.35.

4. Conclusion

Pedestrians tend to move in a random, time-sensitive manner and the car that is driving autonomously has to take a lot of decisions in a very short period of time. Sometimes, ConvNets are not able to extract patterns efficiently from this kind of sequential data.

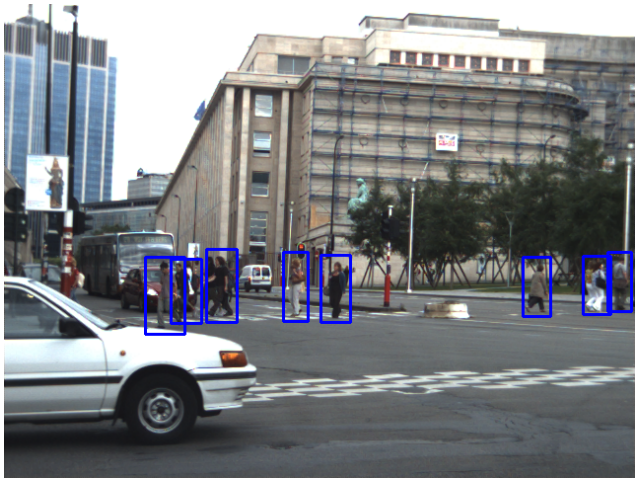

Figure 2. Sample of Detected Pedestrian.

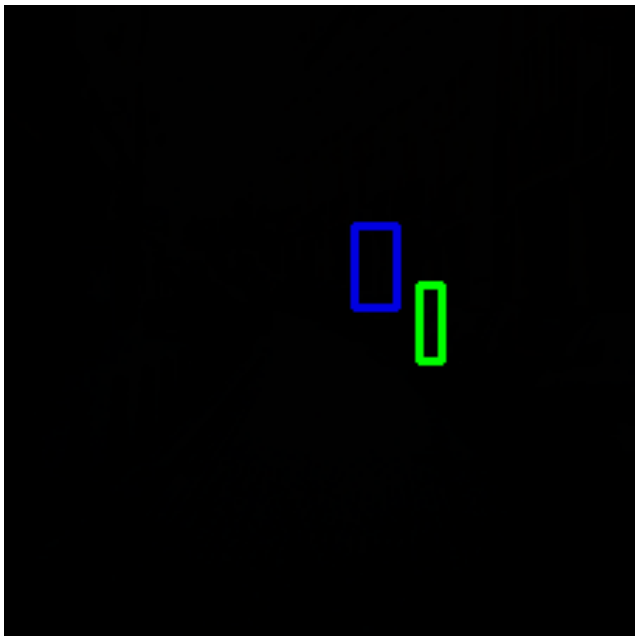

Figure 3. Comparision of test and predicted boxes.

5. References

1. Balaji, Vivek Arvind. Convolutional and Recurrent Neural Networks for Pedestrian Detection. Diss. 2016
2. https://github.com/L0SG/Pedestrian_Detection
3. https://github.com/JTKBowers/CNN-people-detect
4. https://www.hindawi.com/journals/mpe/2018/3518959/
5. https://engmrk.com/alexnet-implementation-using-keras/