Udacity Machine Learning Engineer Nanodegree Capstone Proposal

Peter Bauer

October 15th 2020

Pedestrian Detection

Project Overview

As final project of the Udacity Machine Learning Engineer Nanodegree, a pedestrian detection system will be created. Therefore, the Penn-Fudan Database for Pedestrian Detection and Segmentation[4] is used.

The completion of this Capstone project is split into four parts:

1. Data Exploration and Generation
   Explore a suitable dataset to become familiar with its properties
   If necessary, preprocess the date for later usage
2. Model Training
   Crate a supervised learning model that takes in image patches and classifies them as pedestrian or non-pedestrian
3. Inference
   Test the created system on unseen data and measure its performance
4. Report
   Create a project report including all necessary details

Domain Background

Computer Vision plays a crucial role in state-of-the art autonomous vehicles. Especially in urban areas, an autonomous system must be able to perceive its environment and interact with it to prevent accidents. In general, Computer Vision is an interdisciplinary field of research, that is aiming to teach computers a high-level understanding from digital images or videos. Therefore Computer Vision is concerned with the extraction, analysis and understanding of useful information from images [1].

Problem Statement

Pedestrian detection is an application area of Computer Vision and a non trivial task to solve. There are multiple difficulties that the final algorithm has to cope with. Some of them are changing light, weather and environmental conditions, others are the pedestrians size, its orientation and clothing as well as occlusion in the given image . Due to this varying parameters no hard coded algorithm for object detection can be utilized, either a system has to learn from data to obtain knowledge.

Although many modern systems use a sensor fusion including stereo vision and lidar technology, the object detection from a given input image is still the main component of most vision-based safety systems.

Datasets and Inputs

As mentioned in the project overview, the data from the Penn-Fudan Database for Pedestrian Detection and Segmentation is used. The complete dataset consists of 170 images with 423 marked pedestrians.

Folders in the zip-file of the dataset:

- Annotation Folder
  Text-files that contain useful information for each image
- PedMasks folder
  PNG images of masks for each picture (will not be used in this project)
- PNGImages folder
  Contains all pictures of the dataset in .PNG format
- Added-object-list.txt
  Text file that counts the number of detected persons in each picture
- Readme.txt
  General information

Solution Statement

Regions of interest

To find a pedestrian in a given image, so called regions of interest have to be defined. They shift over the image at different scale and search for the desired object. This concept is also called detection via classification, since you classify the region of interest and receive a position for it afterwards. A huge disadvantage of the sliding window approach is its computationally complexity. A vast amount of image regions at different scale (using a image pyramid) have to be classified. This is especially for real-time systems often not feasible. To reduce the amount of regions some heuristics can be used. For example it is not possible to detect a pedestrian walking in the sky or on the wall of a building. Furthermore, the shape of a pedestrian is roughly known in advance. Modern object detection algorithms like Faster-RCNN use trained neural networks to create regions of interest faster and more efficient. Since this project does not aim to work in real-time scenarios, the simple and traditional Computer Vision approach, using a sliding window and a image pyramid, will be used in the project to create regions of interest.

Detection

There are many different ways to tackle pedestrian detection (or object detection in general). Some of them are for example a Haar wavelet cascade , a neural network using LRF features or histograms of oriented gradients as features which are classified by using a linear SVM [2].

Furthermore, good detection systems are often trained on large datasets. Since this is not directly feasible for this project, a concept called feature extraction will be used to create the classifier[3]. In feature extraction a already existing neural network is used. To get the desired results, the output layer is removed and replaced with a new layer that corresponds to the given problem. Afterwards, the last layer of the network is trained again to adapt to the new task. With this concept, a relatively good performing model can be build without a huge dataset. Combining the region of interest generation with the neural network (trained on pedestrian classification) yields our desired detector.

Benchmark Models

As a benchmark for the detection, the inbuild person detector of OpenCV can be used. This utilizes the above mentioned histogram of oriented gradients in combination with a linear SVM. The model can be easily be applied on the same dataset and therefore provides a good ground truth.

Evaluation Metrics

Both, the person classifier and the detection system as a complete pipeline need to be evaluated. For the classifier an simple accuracy score can be used. This should be sufficient, since the non-pedestrian pictures can be created out of the given data. By creating as many negative samples as positives, the accuracy score of the classifier is not affected by data imbalance and therefore meaningful.
For the detection itself, a simple intersection over union (iou) approach could be used. This means, that a detection is valid, if it overlaps with the ground-truth bounding box by a specified iou-value.

$$iou = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

Using this definition and a certain threshold value all proposed regions can be grouped into True-Positive, False-Positive, True-Negative and False-Negative.

References

[1] https://en.wikipedia.org/wiki/Computer_vision#:~:text=Computer%20vision%20is%20an%20interdisciplinary,human%20visual%20system%20can%20do.

[2] http://www.gavrila.net/pami09.pdf

[3] https://pytorch.org/tutorials/beginner/finetuning_torchvision_models_tutorial.html

[4] https://www.cis.upenn.edu/~jshi/ped_html/