Project:detection of traffic participants in complex scenarios

yh3214 Yunxiao Hu, gy2278 Guanhua Yu

Mobile vision and mobile behavior analysis

Feb.3 2020

Abstract:

Nowadays, video files have appeared in almost every aspects in our daily life and in different kinds of industrial projects. A core issue when working with video files is identifying different objects that appear in the video. Our project aims to solve this problem. We will use Convolutional Neural Network(CNN) to do segmentation on complex traffic scenarios and finally load a well-trained network into mobile devices to do recognition and detection in some test instances.

1.Background (Review of Related Literature):

1.1 The research state of deep learning

Deep learning has dramatically improved the state-of-the-art in many different artificial intelligent tasks like object detection, speech recognition, machine translation (LeCun et al.,2015)[1]. Its deep architecture nature grants deep learning the possibility of solving many more complicated AI tasks (Bengio, 2009)[2]. As a result, researchers are extending deep learning to a variety of different modern domains and tasks in additional to traditional tasks like object detection, face recognition, or language models, for example, Osako et al.[3] (2015) uses the recurrent neural network to denoise speech signals, Gupta et al.[4] (2015) uses stacked auto-encoders to discover clustering patterns of gene expressions. Gatys et al.[5] (2015) uses a neural model to generate images with different styles. Wang et al.[6] (2016) uses deep learning to allow sentiment analysis from multiple modalities simultaneously, etc.

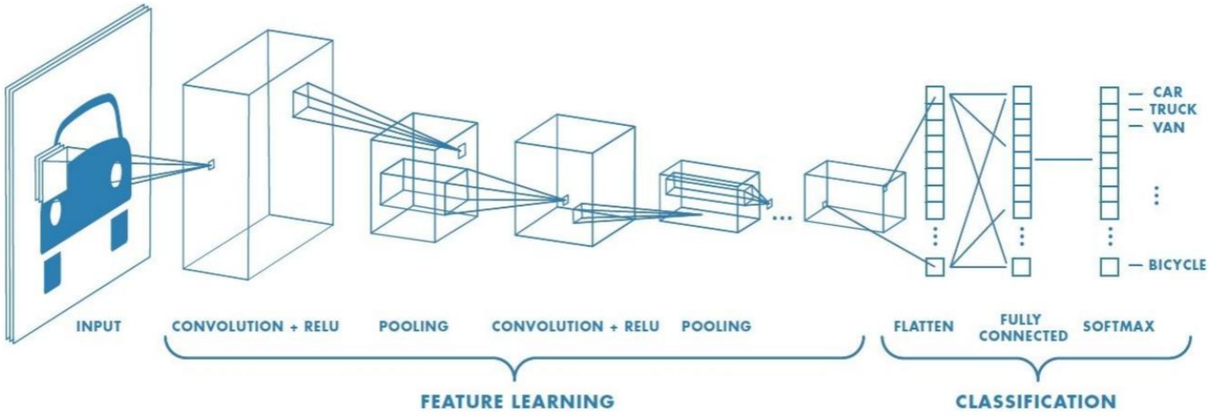
1.2 The introduction of Convolutional Neural Network[8]

Convolutional Neural Networks (CNNs) are analogous to traditional Artificial Neural Networks(ANNs) since that they are comprised of neurons that self-optimize through learning. Each neuron will still receive an input and perform an operation (such as a scalar product followed by a non-linear function) - the basis of countless ANNs. From the input raw matrix data to the final output of the class score, the entire of the network will still express a single perceptive score function (the weight). The last layer will contain loss functions associated with the classes, and all of the regular tips and tricks developed for traditional ANNs still apply.

There are the short summary of CNN architect:

1) As found in other forms of ANN, the input layer will hold the pixel values of the data.
2) The convolutional layer will determine the output of neurons of which are connected to local regions of the input through the calculation of the scalar product between their weights and the region connected to the input volume. The rectified linear unit (commonly shortened to ReLu) aims to apply n element with activation function such as sigmoid to the output of the activation produced by the previous layer.
3) The pooling layer will then simply perform down sampling along the special dimensionality of the given input, further reducing the number of parameters within that activation.

4) The fully-connected layers will then perform the same duties found in standard ANNs and attempt to produce class scores from the activations, to be used for classification. It is also suggested that ReLu may be used between these layers, as to improve performance.



2.Introduction to the Project:

2.1 The goal of the project

We will design the CNN which can help us detect people and cars in the traffic scenario. We also need to implement the code on the phone. Therefore, the final goal should be like using a short video taken in the street through mobile phone camera as a test data, our network will detect the cars and people automatically.

2.2 The method we use

In order to detect moving cars and people in a complex traffic scenario. First we plan to use a pre-trained CNN (ResNet) as a base. Then use our own dataset to reinforce the network. The next step is to save the weight of the neuron network and try to load it in an iphone. Finally we can do the test.

ResNet is an important part of our project. It will help us to do segmentation. There is a special part which is called the residual block. This block was reformulated by degradation problem. If the added layers can be constructed as identity mappings, a deeper model should have training error no greater than its shallower counterpart. The degradation problem suggests that the solvers might have difficulties in approximating identity mappings by multiple nonlinear layers. With the residual learning reformulation, if identity mappings are optimal, the solvers may simply drive the weights of the multiple nonlinear layers toward zero to approach identity mappings.

This method gives the network a chance to be deeper but not cost overfitting. So we use this method to do segmentation for higher accuracy.

After getting our own trained weights of the network, we can upload it to our iphone and use frameworks like caffe, theano, mxnet or Torch to run the codes.

2.3 Potential challenges

Since the training dataset are well-prepared video transferred files, when applying the trained data to our own data, there might be some certain constraints on our video shots, such as the weather(lights and other noise in the background) .

Another important thing is how to deal with big data on a mobile device. How to run code on ios is a big challenge.

3.Introduction to the Dataset:

We choose the dataset from KITTI Vision[9].

3.1 The background of the dataset

The dataset recorded 6 hours of traffic scenarios at 10-100 Hz using a variety of sensor modalities such as high resolution color and grayscale stereo cameras, a Velodyne 3D laser scanner and a high-precision GPS/IMU inertial navigation system. The scenarios are diverse, capturing real-world traffic situations and range from freeways over rural areas to inner city scenes with many static and dynamic objects. The data is calibrated, synchronized and timestamped, and we provide the rectified and raw image sequences. The dataset also contains object labels in the form of 3D tracklets and provide online benchmarks for stereo, optical flow, object detection and other tasks.

3.2 The size of dataset

The raw data set is divided into the categories Road, City , Residential , Campus and Person. For each sequence, we provide the raw data, object annotations in form of 3D bounding box tracklets and a calibration file, as illustrated in Fig. 4. The recordings have taken place on the 26th, 28th, 29th, 30th of September and on the 3rd of October 2011 during daytime. The total size of the provided data is 180 GB

3.3 the short description of row data

All sensor readings of a sequence are zipped into a single file named date_drive.zip, where date and drive are placeholders for the recording date and the sequence number. Besides the raw recordings ('raw data'), we also provide post-processed data ('synced data'), i.e., rectified and synchronized video streams on the dataset website.

There are some photos recorded in the dataset.

4.Plan:

Stage 1 : from now on-03/06/2020. Finding related resources such as research papers about Computer Vision, video files and ResNet. Doing some data preprocessing and setting up the raw model.

Stage 2: 03/07/2020-04/10/2020.Model training and improvements, including fixing the parameters, reducing the loss, making cross validation and doing some analysis and visualization. Making some simple test of the model. Saving the parameters of the model.

Stage 3: 04/10/2020- end of the semester. Trying to upload the weight parameters to the mobile devices. Using frameworks to fit the model with our own test data set. Because the weights of the CNNs come from several training results of different training parameters, it might be of a very large scale. We must solve problems related to those big data on this stage.

Reference:

[1] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. nature, 2015, 521(7553): 436-444.

[2] Yoshua Bengio. Learning deep architectures for ai. Foundations and trends R in Machine Learning, 2(1):1–127, 2009.

[3] Keiichi Osako, Rita Singh, and Bhiksha Raj. Complex recurrent neural networks for denoising speech signals. In Applications of Signal Processing to Audio and Acoustics (WASPAA), 2015 IEEE Workshop on, pages 1–5. IEEE, 2015.

[4] Aman Gupta, Haohan Wang, and Madhavi Ganapathiraju. Learning structure in gene expression data using deep architectures, with an application to gene clustering. In Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on, pages 1328–1335. IEEE, 2015.

[5] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576, 2015.

[6] Haohan Wang, Aaksha Meghawat, Louis-Philippe Morency, and Eric P Xing. Select-additive learning: Improving cross-individual generalization in multimodal sentiment analysis. arXiv preprint arXiv:1609.05244, 2016.

[7] O'Shea K, Nash R. An introduction to convolutional neural networks[J]. arXiv preprint arXiv:1511.08458, 2015.

[8] http://www.cvlibs.net/datasets/kitti/