Vehicle Counting in Surveillance Videos

Yiqi Yan[1], Irene Cheng[2]

[1]Department of Communication Engineering, Northwestern Polytechnical University, [2]Department of Computing Science, University of Alberta

UNIVERSITY OF ALBERTA

Baseline

Vehicle Counting Using Double Virtual Lines[1]

Motivation

- ❑ Traditionally, virtual loops are assigned for counting vehicles. However, repeat counting may occur when vehicles are roadway departure due to overtaking or crossing.
  Solution: Double Virtual Lines (DVL)
- ❑ After the background subtraction operation, we get a binary image of foreground objects (the vehicles). However, inevitable noise may cause false counting.
  Solution: template convolution.

Overview of the method

❑ **DVL assignment**

The DVL is assigned by estimating the vehicle's 2-D projection on the image plane. The projective transformation matrix of the camera is needed.

❑ **Background subtraction**

Mixture of Gaussians (MOG) is used to model the background. Foreground mask is computed by subtracting background from the original image.

❑ **Vehicle detection**

Morphological filtering is used to remove the holes and enhance the targets. Concretely, dilation operation with a disk-shaped structuring element is used.

❑ **Vehicle location & Vehicle counting**

Template convolution is used to locate and count the vehicles. The template is a matrix filled with 1's, whose height is the same as the distance between DVLs. The convolutional operation is performed only in the detection zone, i.e. between the DVLs. After performing template convolution, I got the convolutional curve just like the figure on the left, where each peak indicates a vehicle. Counting rules are then designed based on convolutional curves.

Results

I use a video downloaded from the Internet to test my code. The image here shows one of the frames. There are three numbers shown on the top-left corner (791, +1, 39). The first number is the current frame number. The second number is the number of the newly counted vehicles. The third number is the total number of vehicles that have been counted.
The algorithm runs well and gives exactly the right result.

How to improve?

Current problems

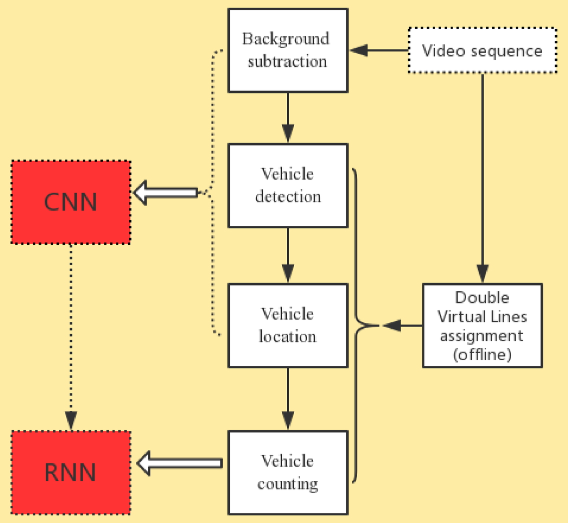
❑ Hard to generalize

The values of parameters are greatly influenced by viewing perspective, environment, etc. Therefore, it will be disturbing to port the system to a new environment.

❑ This is a pipeline method

In a pipeline framework, each module has its own parameters. This makes it difficult to debug the system.

Possible Methods of improvement



Deep learning model learns parameters themselves, and they allow end-to-end training

- ❑ Use CNN to locate the foreground targets and extract features.
- ❑ Use RNN to model temporal relation for accurate counting.

[1] Xu H, Zhou W, Zhu J, Huang X, and Wang W. Vehicle counting based on double virtual lines. Signal, Image and Video Processing. 2017 Jul 1;11(5):905-12

Improvement

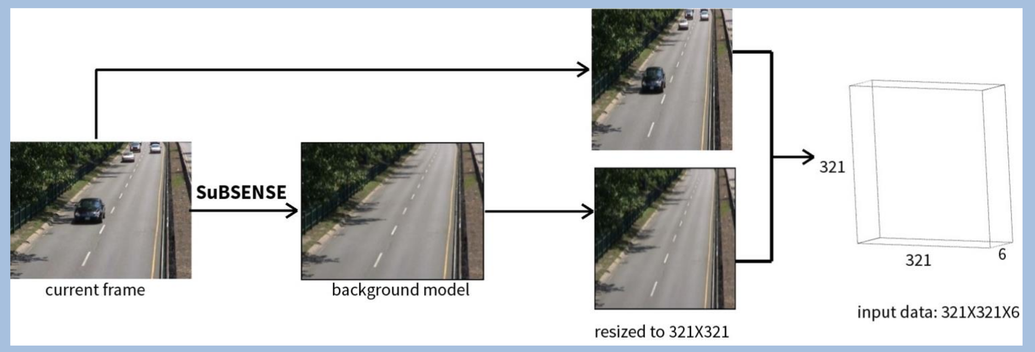
DL Background Subtraction

In traditional background subtraction methods, foreground targets are detected by subtracting the background image from the original image. Therefore, these methods relies much on the quality of the background model. When the background image is not perfect, the algorithm tends to fail.

Let's consider another way. What if we COMPARE the original image with its background, rather than just SUBTRACT. This kind of comparison is a complicated non-linear mapping. Deep CNN can help with this.

Generate background image

Given one frame from the video. I get the background image using one traditional method called SuBSENSE. Then the current frame and the background image are resized to 321×321. Finally, the two resized images are stacked to form a 321×321×6 data cube, which is used as the input of the neural network.



CNN for background subtraction

I use 50-layer Resnet for feature extraction. Decovonlutional layers are used to upsample the feature map to the original spatial size (321×321). Additional max-pooling layers are used to eliminate extra zeros in decovonlutional feature maps. This is important to reduce checkboard artifacts.

Before Resnet, convolutional operation with 1×1 filter is used to map the input data cube into a 3-channel feature map. This is because Resnet only takes input that has 3 channels.

As for loss function, I choose pixel-wise binary cross entropy.

	Filter size / Pooling window size	Stride (H,W,D)	Input size	Output size
pre-conv	1x1x6x3	1, 1, --	321x321x6	321x321x3
resnet_50	----	----	321x321x3	21x21x2048
3D avg_pool	1x1x48	1,1,40	21x21x2048	21x21x51
deconv_1	3x3x32x51	2, 2, --	21x21x51	43x43x32
3D max_pool	3x3x2	1, 1, 2	43x43x32	41x41x16
deconv_2	3x3x8x16	2, 2, --	41x41x16	83x83x8
2D max_pool	3x3	1, 1, --	83x83x8	81x81x8
deconv_3	3x3x4x8	2, 2, --	81x81x8	163x163x4
2D max_pool	3x3	1, 1, --	163x163x4	161x161x4
deconv_4	3x3x1x4	2, 2, --	161x161x4	323x323x1
2D max_pool	3x3	1, 1, --	323x323x1	321x321x1
conv	1x1x1x1	1, 1, --	321x321x1	321x321x1

Results

Dataset

I use part of CDnet 2014 dataset to train the model. Some videos are removed from the dataset, because the foreground objects are intermittent in those videos, which is not the case I will consider in traffic videos. I get over 40k training samples and over 1k test samples.

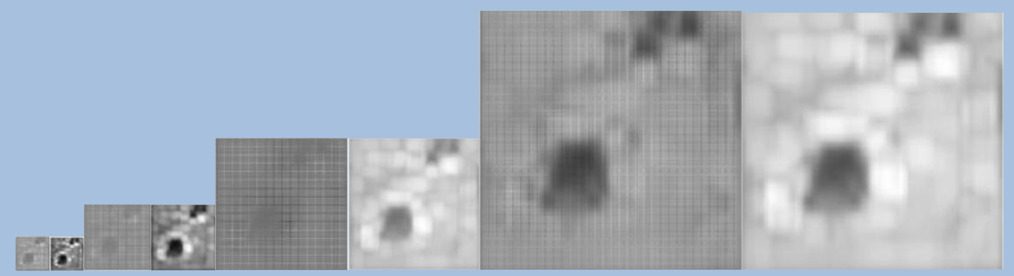
Training

I train the model with Adam optimizer for 10000 steps. The average loss reduces to as low as 0.01 on training set, and 0.02 on test set. Shown below is the result of one frame in the test set.

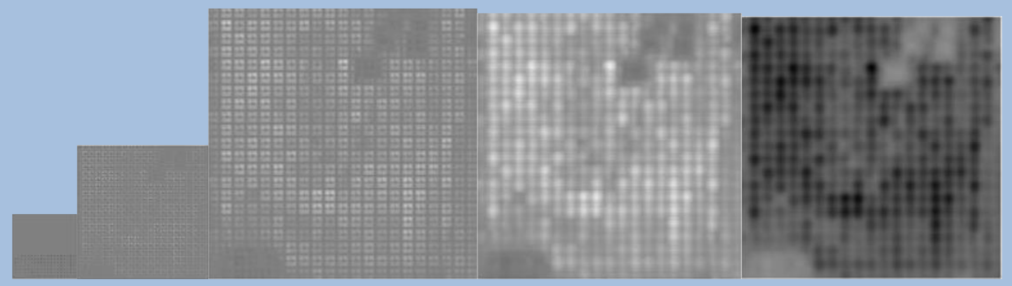
original frame groundtruth CNN output

Checkboard artifacts

The max-pooling layers in the model are used to eliminate extra zeros in decovonlutional feature maps. Without those layers, checkboard artifacts will impair the performance of the model. Please refer to the visualization of deconvolutional feature maps shown below.



with additional max-pooling



without additional max-pooling

Future Work

- ❑ Go on with the CNN model: I'm now considering using recurrent convolution network to segment foreground objects.
- ❑ Build the RNN model, which receives the feature extracted by CNN, and outputs the vehicle counting result.