Understanding Traffic Density from Large-Scale Web Camera Data

Shanghang Zhang[†,‡], Guanhang Wu[†], João P. Costeira[‡], José M. F. Moura[†]
[†]Carnegie Mellon University, Pittsburgh, PA, USA
[‡]ISR - IST, Universidade de Lisboa, Lisboa, Portugal

{shanghaz, guanhanw}@andrew.cmu.edu, jpc@isr.ist.utl.pt, moura@andrew.cmu.edu

Abstract

Understanding traffic density from large-scale web camera (webcam) videos is a challenging problem because such videos have low spatial and temporal resolution, high occlusion and large perspective. To deeply understand traffic density, we explore both optimization based and deep learning based methods. To avoid individual vehicle detection or tracking, both methods map the dense image feature into vehicle density, one based on rank constrained regression and the other based on fully convolutional networks (FCN). The regression based method learns different weights for different blocks of the image to embed road geometry and significantly reduce the error induced by camera perspective. The FCN based method jointly estimates vehicle density and vehicle count with a residual learning framework to perform end-to-end dense prediction, allowing arbitrary image resolution, and adapting to different vehicle scales and perspectives. We analyze and compare both methods, and get insights from optimization based method to improve deep model. Since existing datasets do not cover all the challenges in our work, we collected and labelled a large-scale traffic video dataset, containing 60 million frames from 212 webcams. Both methods are extensively evaluated and compared on different counting tasks and datasets. FCN based method significantly reduces the mean absolute error (MAE) from 10.99 to 5.31 on the public dataset TRANCOS compared with the state-of-the-art baseline.

1. Introduction

Traffic congestion leads to the need for a deep understanding of traffic density, which together with average vehicle speed, form the major building blocks of traffic flow analysis [27]. Traffic density is the number of vehicles per unit length of a road (e.g., vehicles per km) [19]. This paper focuses on traffic density estimation from webcam videos, which are of low resolution, low frame rate, high occlusion and large perspective. As illustrated in Figure 1, we select a region of interest (yellow dotted rectangle) in a video stream, and count the number of vehicles in the region for
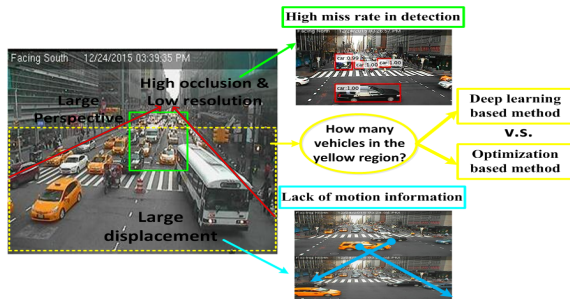

Figure 1. Problem Statement

each frame. Then the traffic density is calculated by dividing that number by the region length.

Nowadays, many cities are being instrumented with surveillance cameras. However, due to network bandwidth limitations, lack of persistent storage, and privacy concerns [11], these videos present several challenges for analysis (illustrated in Figure 1): (i) *Low frame rate*. The time interval between two successive frames of a webcam video typically ranges from $1s$ to $3s$, resulting in large vehicle displacement. (ii) *Low resolution*. The resolution of webcam videos is 352×240. Vehicle at the top of a frame can be as small as 5×5 pixels. Image compression also induces artifacts. (iii) *High occlusion*. Cameras installed at urban intersections often capture videos with high traffic congestion, especially during rush hours. (iv) *Large perspective*. Cameras are installed at high points to capture more video content, resulting in videos with large perspective. Vehicle scales vary dramatically based on their distance to the camera. These challenges make the existing work for traffic density estimation has many limitations.

1.1. Limitation of Related Works

Several traffic density estimation techniques have been developed in the literature, but they perform less accurately on the webcam data due to the above challenges:
Detection based methods. These methods [38, 29] try to identify and localize vehicles in each frame. They perform poorly in low resolution and high occlusion videos. Figure 1

1

shows detection results by Faster RCNN [26]. Even though trained on our collected and annotated webcam dataset, it still exhibits very high missing rate.

Motion based methods. Several methods [8, 9, 23] utilize vehicle tracking to estimate traffic flow. These methods tend to fail due to low frame rate and lack of motion information. Figure 1 shows a large displacement of a vehicle (black car) in successive frames due to the low frame rate. Some vehicles only appear once in the video and their trajectory cannot be well estimated.

Holistic approaches. These techniques [33] perform analysis on the whole image, thereby avoiding segmentation of each object. [15] uses a dynamic texture model based on Spatiotemporal Gabor Filters for classifying traffic videos into different congestion types, but it does not provide accurate quantitative vehicle densities. [30] formulates the object density as a linear transformation of each pixel feature, with a uniform weight over the whole image. It suffers from low accuracy when the camera has large perspective.

Deep learning based methods. Recently, several deep learning based methods have been developed for object counting[35, 36, 25, 37, 2]. The network in [35] outputs a 1D feature vector and fit a ridge regressor to perform the final density estimation, which cannot perform pixel-wise prediction and lose the spatial information. The estimated density map cannot have the same size as the input image. [25] is based on fully convolutional networks but the output density map is still much smaller than the input image, because it does not have deconvolutional or upsampling layers. [2] jointly learns density map and foreground mask for object counting, while it does not solve the large perspective and object scale variation problems.

To summarize, detection and motion based methods tend to fail in high congestion, low resolution, and low frame rate videos, because they are sensitive to video quality and environment conditions. The holistic approaches perform poorly in videos with large perspective and variable vehicle scales. Besides, most of existing methods are incapable of estimating the exact number of vehicles. [13, 18, 10, 34].

1.2. Contributions

To deeply understand traffic density and overcome the challenges from real-world webcam data, we explore both deep learning based and optimization based methods. The optimization based model (OPT-RC) embeds scene geometry through the rank constraint of multiple block-regressors, and motivates the deep-learning model FCN-MT. FCN-MT shares the idea of mapping local feature into vehicle density with OPT-RC, while it replaces the BG subtraction, feature extractor, and block-regressors with fully convolutional networks. With extensive experiments, we analyze and compare both methods, and get insights from optimization based method to improve deep model.
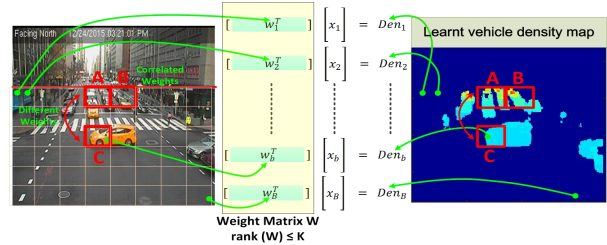

Figure 2. Intuition for OPT-RC.

Optimization Based Vehicle Density Estimation with Rank Constraint (OPT-RC). Inspired by [30], which maps each pixel feature into vehicle density with a uniform weight, we propose a regression model to learn different weights for different blocks to increase the degrees of freedom on the weights, and embed geometry information. It outperforms work [30] and obtains high accuracy in low quality webcam videos, especially overcoming large perspective challenge. We first divide the target region into blocks, extract features for each block, and subtract background. As illustrated in Figure 2, we linearly map each block feature $\mathbf{x}_b$ into vehicle density $\text{Den}_b = \mathbf{w}_b^\top \mathbf{x}_b$. To avoid large errors induced by large perspective, we build one regressor per block with different weights $\mathbf{w}_b$, and learn the optimal weights. All the weight vectors are stacked into a weight matrix $\mathbf{W} = \begin{bmatrix} \mathbf{w}_1^\top; & \mathbf{w}_2^\top; & ... & \mathbf{w}_B^\top \end{bmatrix}$. To handle high dimensionality and capture the correlations among weight vectors of different blocks, rank constraint is imposed on the weight matrix $\mathbf{W}$. The motivation behind such treatment is illustrated in Figure 2. Due to large perspective, vehicles have different scales in block A and C, and their corresponding vehicle densities are also different. Yet for block A and B, the vehicle densities are similar. Thus we build the weight matrix $\mathbf{W}$ to reflect both the diversity and the correlation among weight vectors. After the vehicle density map is estimated, the vehicle count can be obtained by integrating the vehicle density map. Finally, the traffic density is obtained by dividing the vehicle counts by the length of the target region.
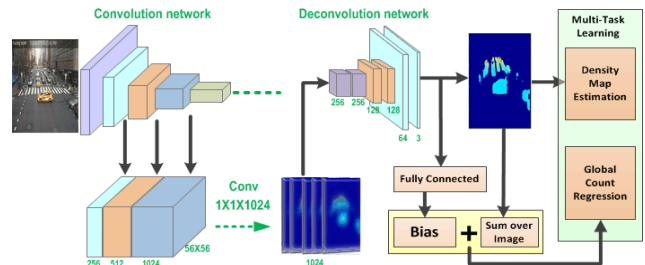

Figure 3. Framework of FCN-MT

FCN Based Multi-Task Learning for Vehicle Counting (FCN-MT). To avoid individual vehicle detection or tracking, besides the proposed optimization based model, we further propose an FCN based model to jointly learn

vehicle density and vehicle count. The framework is illustrated in Figure 3. To produce the density map that has the same size as input image, we design a fully convolutional network [21] to perform pixel-wise prediction whole-image-at-a-time by dense feedforward computation and backpropagation. Instead of applying simple bilinear interpolation for upsampling, we add deconvolution layers on top of convolutional layers, whose parameters can be learnt during training.

There are two challenges of FCN-based object counting: (1) object scale variation, and (2) reduced feature resolution[7]. To avoid large errors induced by scale variations, we jointly perform global count regression and density estimation. Single task (density estimation) method only encourages networks to approximate ground truth density and directly sums the densities to get the count, which suffers from large error when there is extreme occlusion or oversized vehicles. Yet the multi-task framework is fundamental to account for such deviations, enabling related objectives achieve better local optima, improving robustness, and providing more supervised information. Furthermore, instead of directly regressing the global vehicle count from the last feature map, we develop a residual learning framework to reformulate global count regression as learning residual functions with reference to the sum of densities in each frame. Such design avoids learning unreferenced functions and eases the training of network. The second challenge is caused by the repeated combination of max-pooling and striding. To solve this problem, we produce denser feature maps by combining appearance features from shallow layers with semantic features from deep layers. We then add a convolution layer after the combined feature volume with 1x1 kernels to perform feature re-weighting. The re-weighted features better distinguish foreground and background. Thus the whole network is able to accurately estimate vehicle density without foreground segmentation.

Webcam Traffic Video Dataset (WebCamT) We collected and labelled a large-scale webcam traffic dataset, which contains 60 million frames from 212 webcams installed in key intersections of the city. This dataset is annotated with vehicle bounding box, orientation, re-identification, speed, category, traffic flow direction; weather and time. Unlike existing car dataset KITTI[14] and Detrac[32], which focus on vehicle models, our dataset emphasizes real world traffic network analysis in a large metropolis. This dataset has three benefits: (i) It motivates research on vision based traffic flow analysis, posing new challenges for state-of-the-art algorithms. (ii) With various street scenes, it can serve as benchmark for transfer learning and domain adaptation. (iii) With large amount of labeled data, it provides training set for various learning based models, especially for deep learning based techniques.

Contributions of this paper are summarized as follows:

1. We propose an optimization based density estimation method (OPT-RC) that embeds road geometry in the weight matrix and significantly reduces error induced by perspective. It avoids detecting or tracking individual vehicles.

2. We propose FCN based multi-task learning to jointly estimate vehicle density and count with end-to-end dense prediction. It allows arbitrary input image resolution, and adapts to different vehicle scale and perspective.

3. We collect and annotate a large-scale webcam traffic dataset, which poses new challenges to state-of-the-art traffic density estimation algorithms. To the best of our knowledge, it is the first and largest webcam traffic dataset with elaborate annotations.

4. With extensive experiments on different counting tasks, we verify and compare the proposed FCN-MT with OPT-RC, and obtain insight for future study.

The rest of paper is outlined as follows. Section 2 introduces the proposed OPT-RC. Section 3 introduces the proposed FCN-MT. Section 4 presents experimental results, and Section 5 compares OPT-RC with FCN-MT.

2. Optimization Based Vehicle Density Estimation with Rank Constraint

To overcome limitations of existing work, we propose a block-level regression model with rank constraint. The overall framework is described in Section 1.2. We first perform foreground segmentation based on GrabCut [3]. To automate the segmentation process, we initialize the background and foreground based on the difference between the input frame and the pure background image, which is generated by using a reference image with no vehicles taken in light traffic periods and transferring it to other time periods by brightness adjustment. We assume that a stream of N images $I_1, \cdots, I_N$ are given, for which we select a region of interest and divide it into J blocks. The block size can vary from 16×16 to 1×1, depending on the width of the lane and the length of the smallest vehicle. A block $B_j^{(i)}$ in each image I_i is represented by a feature vector $\mathbf{x}_j^{(i)} \in \mathbb{R}^K$. Examples of particular choices of features are given in the experimental section. It is assumed that each training image I_i is annotated with a set of 2D bounding boxes, centered at pixels $P_i = \{p_1, \cdots, p_{c(i)}\}$, where $c(i)$ is the total number of annotated vehicles in the i-th image. The density functions in our approach are real-valued functions over pixel grids, whose integrals over image regions equal to the vehicle counts. For a training image I_i, we calculate the ground truth density based on the labeled bounding boxes (shown in Figure 4). The pixel p covered by a set of bounding boxes $O(p)$ has density $D(p)$ defined as

$$D(p) = \sum_{o \in O(p)} \frac{1}{A(o)}, \qquad (1)$$

where $A(o)$ denotes the area of bounding box o. Then, we define the density $D(B_j)$ of a block as

$$D(B_j) = \sum_{p \in B_j} D(p). \tag{2}$$

Given a set of training images together with their ground truth densities, for each block B_j, we learn a block-specific linear regression model that predicts block-level density $\widehat{D}(B_j)$ given its feature representation $\mathbf{x}_j$ by

$$\widehat{D}(B_j) = \mathbf{w}_j^\top \mathbf{x}_j, \tag{3}$$

where $\mathbf{w}_j \in \mathbb{R}^K$ is the coefficient vector of the linear regression model to be learned for block j. We assign different weights to different blocks. To capture the correlations and commonalities among the regression weight vectors at different blocks, we encourage these vectors to share a low-rank structure. To avoid overfitting, we add ℓ_2-regularization to these weight vectors. To encourage sparsity of weights, ℓ_1-regularization is imposed. Let $\mathbf{W} \in \mathbb{R}^{K \times J}$, where the j-th column vector $\mathbf{w}_j$ denotes the regression coefficient vector of block j. To this end, we define the following regularized linear regression model with low-rank constraint

$$\min_{\mathbf{W}} \quad \frac{1}{2N} \sum_{i=1}^{N} \sum_{j=1}^{J} (\mathbf{w}_j^\top \mathbf{x}_j^{(i)} - D(B_j^{(i)}))^2 + \alpha \|\mathbf{W}\|_F^2 + \beta |W|_1$$
$$\text{s.t.} \quad \text{rank}(\mathbf{W}) \leq r$$
$$\tag{4}$$

To solve this rank-constrained problem that has a non-smooth objective function, we develop an accelerated projected subgradient descent (APSD) [20] algorithm outlined in Algorithm 1, which iteratively performs subgradient descent, rank projection, and acceleration. Two sequences of variables $\{\mathbf{A}_k\}$ and $\{\mathbf{W}_k\}$ are maintained for the purpose of performing acceleration.

Subgradient Descent Subgradient descent is performed over variable $\mathbf{A}_k$. We first compute the subgradient $\triangle \mathbf{A}_k$ of the non-smooth objective function $\frac{1}{2N} \sum_{i=1}^{N} \sum_{j=1}^{J} (\mathbf{a}_j^\top \mathbf{x}_j^{(i)} - D(B_j^{(i)}))^2 + \alpha \|\mathbf{A}\|_F^2 + \beta |\mathbf{A}|_1$, where the first and second terms are smooth, hence their subgradients are simply gradients. For the third term $|\mathbf{A}|_1$ which is non-smooth, its subgradient $\partial \mathbf{A}$ can be computed as

$$\partial A_{ij} = \begin{cases} +1 & \text{if } A_{ij} \geq 0 \\ -1 & \text{if } A_{ij} < 0 \end{cases} \tag{5}$$

Adding the subgradients of the three terms, we obtain the subgradient $\triangle \mathbf{A}_k$ of the overall objective function. Then $\mathbf{A}_k$ is updated by

$$\mathbf{A}_k \leftarrow \mathbf{A}_k - t_k \triangle \mathbf{A}_k. \tag{6}$$

Algorithm 1 Accelerated Projected Subgradient Descent

Input: Data $\mathcal{D}$, rank r, regularization parameters α, β
1: **while** not converged **do**
2:     Compute gradient $\triangle \mathbf{A}_k$
3:     $\mathbf{A}_k \leftarrow \mathbf{A}_k - t_k \triangle \mathbf{A}_k$
4:     Compute top r singular values and vectors of $\mathbf{A}_k$: $\mathbf{U}_r, \mathbf{\Sigma}_r, \mathbf{V}_r$
5:     $\mathbf{W}_{k+1} \leftarrow \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^\top$
6:     $t_{k+1} \leftarrow \frac{1}{2}(1 + \sqrt{1 + 4t_k^2})$
7:     $\mathbf{A}_{k+1} \leftarrow \mathbf{A}_k \frac{t_{k-1}-1}{t_k}(\mathbf{W}_k - \mathbf{W}_{k-1})$
8: **end while**
Output: $\mathbf{W} \leftarrow \mathbf{W}_{k+1}$

Rank Projection In lines 4-5, we project the newly obtained $\mathbf{A}_k$ to the feasible set $\{\mathbf{W}|\text{rank}(\mathbf{W}) \leq r\}$, which amounts to solving the following problem

$$\begin{aligned} \min_{\mathbf{W}} \quad & \|\mathbf{A}_k - \mathbf{W}\|_F^2 \\ \text{s.t.} \quad & \text{rank}(\mathbf{W}) \leq r \end{aligned} \tag{7}$$

According to [12], the optimal solution $\mathbf{W}^*$ can be obtained by first computing the largest r singular values and singular vectors of $\mathbf{A}_k$: $\mathbf{U}_r, \mathbf{\Sigma}_r, \mathbf{V}_r$, then setting $\mathbf{W}^*$ to $\mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^\top$.

Acceleration In lines 6-7, acceleration is performed by updating the step size according to the following rule: $t_{k+1} \leftarrow \frac{1}{2}(1 + \sqrt{1 + 4t_k^2})$, and adding a scaled difference between consecutive $\mathbf{W}$s to $\mathbf{A}$.

It is worth noting that this optimization problem is not convex and the APSD algorithm may lead to local optimal. It helps to run the algorithm multiple times with different random initializations.

3. FCN Based Multi-Task Learning

We also propose an FCN based model to jointly learn vehicle density and global count. The vehicle density estimation can be formulated as $D(i) = F(X_i; \Theta)$, where X_i is the input image, Θ is the set of parameters of the FCN-MT model, and $D(i)$ is the estimated vehicle density map for image i. The ground truth density map can be generated in the same way as Section 2.

3.1. Network Architecture

Inspired by the FCN used in semantic segmentation[21], we develop FCN to estimate the vehicle density. After the vehicle density map is estimated, the vehicle count can be obtained by integrating the vehicle density map. However, we observed that variation of vehicle scales induces error during the direct integration. Especially, the large buses/trucks (oversized vehicles) in close view induce sporadically large errors in the counting results. To solve this problem, we propose a deep multi-task learning framework based on FCN to jointly learn vehicle density map and vehicle count. Instead of directly regressing the count from the last feature map or the learnt density map, we develop

a residual learning framework to reformulate global count regression as learning residual functions with reference to the sum of densities in each frame. The overall structure of our proposed FCN-MT is illustrated in Figure 3, which contains convolution network, deconvolution network, feature combination and selection, and multi-task residual learning.

The convolution network is based on pre-trained ResNets[17]. The pixel-wise density estimation requires high feature resolution, yet the pooling and striding reduce feature resolution significantly. To solve this problem, we rescale and combine the features from $2a$, $3a$, $4a$ layers of ResNets. We then add a convolution layer after the combined feature volume with 1x1 kernels to perform feature re-weighting. By learning parameters of this layer, the re-weighted features can better distinguish foreground and background pixels. We input the combined feature volume into the deconvolution network, which contains five deconvolution layers. Inspired by the deep VGG-net[28], we apply small kernels of 3x3 in the deconvolution layers. The feature is mapped back to the image size by deconvolution layers, whose parameters can be learnt from the training process[24]. A drawback of deconvolution layer is that it may have uneven overlap when the kernel size is not divisible by the stride. We add one convolution layer with 3x3 kernels to smooth the checkerboard artifacts and alleviate this problem. Then one more convolution layer with 1x1 kernels is added to map the feature maps into density map.

3.2. Multi-Task Learning

At the last stage of the network, we jointly learn vehicle density and count. The vehicle density is predicted by the last convolution 1x1 layer from the feature map. Euclidean distance is adopted to measure the difference between the estimated density and the ground truth. The loss function for density map estimation is defined as follows:

$$L_D(\Theta) = \frac{1}{2N} \sum_{i=1}^{N} \sum_{p=1}^{P} \|F(X_i(p); \Theta) - F_i(p)\|_2^2, \quad (8)$$

where N is the number of training images and $F_i(p)$ is the ground truth density for pixel p in image i.

For the second task, global count regression, we reformulate it as learning residual functions with reference to the sum of densities, which consists of two parts: (i) base count: the integration of the density map over the whole image; (ii) offset count: predicted by two fully connected layers from the feature map after the convolution 3x3 layer of the deconvolution network. We sum the two parts together to get the estimated vehicle count, as shown in the following equation:

$$C(i) = G(D(i); \gamma) + \sum_{p=1}^{P} D(i, p), \quad (9)$$

where γ is the learnable parameters of the two fully connected layers, $D(i, p)$ indicates the density of each pixel p in image i. We hypothesize that it is easier to optimize the residual mapping than to optimize the original, unreferenced mapping. Considering that the vehicle count for some frames may have very large value, we adopt Huber loss to measure the difference between the estimated count and the ground truth count. The count loss for one frame is defined as follows:

$$L_\delta(i) = \begin{cases} \frac{1}{2}(C(i) - C_t(i))^2 & \text{for} |C(i) - C_t(i)| \le \delta, \\ \delta |C(i) - C_t(i)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases} \quad (10)$$

where $C_t(i)$ is the ground truth vehicle count of frame i, $C(i)$ is the estimated loss of frame i. δ is the threshold to control the outlier in the training sets. Then overall loss function for the network is defined as:

$$L = L_D + \lambda \frac{1}{N} \sum_{i=1}^{N} L_\delta(i), \quad (11)$$

where λ is the weight of count loss, and is tuned to achieve best accuracy. By simultaneously learning the two related tasks, each task can be better trained with much fewer parameters. The loss function is optimized via batch-based Adam and backpropagation. As FCN-MT adapts to different input image resolutions and variation of vehicle scales and perspectives, it is robust to different scenes.

4. Experiments

We extensively evaluate the proposed methods on different datasets and counting tasks: (i) We first introduce our collected and annotated webcam traffic dataset (WebCamT). (ii) Then we evaluate and compare the proposed methods with state-of-the-art methods on WebCamT dataset and present an interesting application to detect the change of traffic density patterns in NYC on Independence Day. (iii) We evaluate our proposed methods on the public dataset TRANCOS[25]. (iv) We evaluate our methods on the public pedestrian counting dataset UCSD [4] to verify the robustness and generalization of our model.

4.1. Webcam Traffic Data Collection

As there is no existing labeled real-world webcam traffic dataset, in order to evaluate our proposed method, we utilize existing traffic web cameras to collect continuous stream of street images and annotate rich information. Different from existing traffic datasets, webcam data are challenging for analysis due to the low frame rate, low resolution, high occlusion, and large perspective. We select 212 representative web cameras, covering different locations, camera perspective, and traffic states. For each camera, we downloaded videos for four time intervals each day (7am-8am, 12pm-1pm; 3pm-4pm; 6pm-7pm). These cameras have frame rate around 1 frame/second and resolution 352×240. Collecting these data for 4 weeks generates 1.4 Terabytes of video data
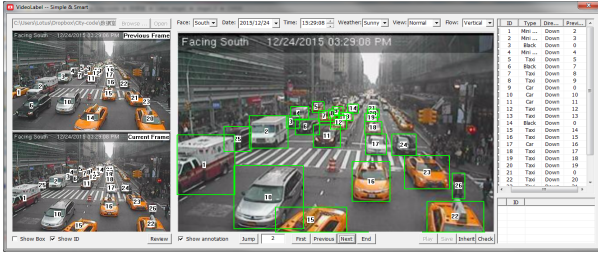
Figure 4. Annotation Instance.

consisting of 60 million frames. To the best of our knowledge, WebCamT is the first and largest annotated webcam traffic dataset to date.

We annotate 60,000 frames with the following information: (i) *Bounding box*: rectangle around each vehicle. (ii) *Vehicle type*: ten types including taxi, black sedan, other cars, little truck, middle truck, big truck, van, middle bus, big bus, other vehicles. (iii) *Orientation*: each vehicle orientation is annotated into four categories: 0, 90, 180, and 270 degrees. (iv) *Vehicle density*: number of vehicles in ROI region of each frame. (v) *Re-identification*: we match the same car in sequential frames. (vi) *Weather*: five types of weather, including sunny, cloudy, rainy, snowy, and intensive sunshine. The annotation for two successive frames is shown in Figure 4. The dataset is divided into training and testing sets, with 42,200 and 17,800 frames, respectively. We select testing videos taken at different time from training videos. WebCamT serves as an appropriate dataset to evaluate our proposed method. It also motivates research on vision based traffic flow analysis, posing new challenges for the state-of-the-art algorithms [1].

4.2. Quantitative Evaluations on WebCamT

We evaluate the proposed methods on the testing set of WebCamT, containing 61 video sequences from 8 cameras, and covering different scenes, congestion states, camera perspectives, weather and time of the day. Each video has 352×240 resolution, and frame rate around 1 frame/sec. The training set has the same resolution, but is from different videos. Three metrics are employed for evaluation: (i) Mean absolute error (MAE); (ii) Mean square error (MSE); (iii) Average relative error (ARE), which is the average of all the test frame relative errors.

For OPT-RC, to compare with the baseline methods, we extract SIFT feature and learn visual words for each block. The block size is determined by the lane width and the smallest vehicle length. Parameters in Eq.(4) are selected by cross-validation. For FCN-MT, we divide the training data into two groups: downtown and parkway. In each group, we balance the training frames with less than 15 vehicles and the frames with more than 15 vehicles. The network architecture is explained in Section 3.1 and some

---

[1]Please email the authors if you are interested in the dataset.

parameters are shown in Figure 3. The weight λ of vehicle count loss in Eq.(11) is 0.1. More details can be found in the released code link: `https://github.com/polltooh/traffic_video_analysis`.

Baseline approaches. We compare our method with two methods: *Baseline 1: Learning to count* [30]. It maps the feature of each pixel into object density with uniform weight for the whole image. For comparison, we extract dense SIFT features [22] using VLFeat [31]. The ground truth density is computed as a normalized 2D Gaussian kernel based on the center of each labeled bounding box. *Baseline 2: Hydra*[25]. It learns multi-scale regression networks using a pyramid of image patches extracted at multiple scales to perform final density prediction. We train Hydra 3s model on the same training set as FCN-MT.

Table 1. Accuracy comparison on WebCamT

Method	Downtown		Parkway	
	MAE	ARE	MAE	ARE
Baseline 1	5.91	0.5104	5.19	0.5248
Baseline 2	3.55	0.6613	3.64	0.6741
OPT-RC	4.56	0.6102	4.24	0.6281
FCN-MT	2.74	0.7175	2.52	0.784

Experimental Results. Below, we compare the errors of the proposed and baseline approaches in Table 1. From these results, we conclude that FCN-MT outperforms the baseline approaches and OPT-RC for all the measurements. As the testing data cover different congestion states, camera perspectives, weather conditions and time of the day, these results verify the generalization and robustness of FCN-MT. OPT-RC outperforms the non-deep learning based Baseline 1 and shows comparable results with Baseline 2, but require much less training data. Compared with FCN-MT, OPT-RC is less generalizable, but it can learn smooth density map with geometry information. Figure 5 shows the original image (a), learned density map from Baseline1 (b), and learned density map from OPT-RC (c). We see that the density map from Baseline 1 can not reflect the perspective present in the video, while density map from our method captures well the camera perspective. Figure 6 shows the density map learned from FCN-MT. Without foreground segmentation, the learned density map can still estimate the region of vehicles, and distinguish background from foreground in both sunny and cloudy, dense and sparse scenes. Yet due to the uneven overlaps of the deconvolution layers, checkerboard artifacts are created on the learnt density map.

Figure 7 and Figure 8 shows the estimated traffic density of OPT-RC and FCN-MT from long time sequences and multiple cameras, respectively. MAE of each camera's estimated traffic density is shown at the bottom right of each plot. From the results, we see that FCN-MT has more accurate estimation than OPT-RC, yet both methods can cap-

ture the trends of the traffic density. For the same time of day, traffic density from downtown cameras are on average higher than that from parkway cameras. For the same camera location, traffic density during nightfall (18:00-19:00) is generally higher than at other times of the day. Especially for parkway cameras, nightfall traffic density increases significantly when compared to traffic density in the morning and at noon. As the test videos cover different locations, weather, camera perspectives, and traffic states, those results verify the robustness of the proposed methods.
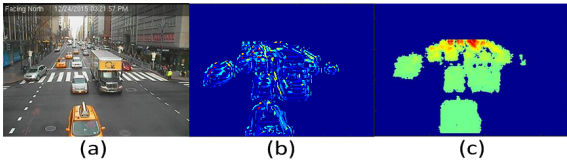
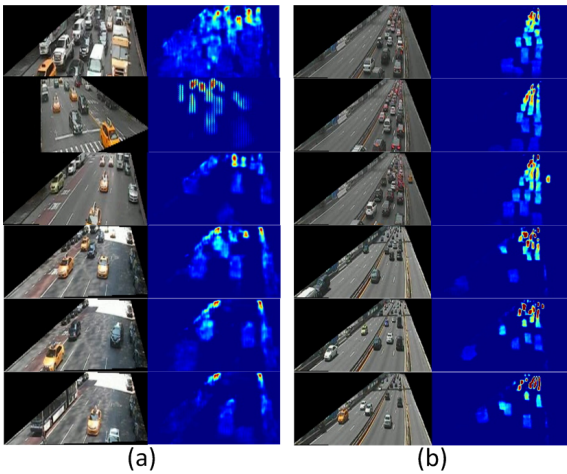Figure 5. Comparison of OPT-RC and Baseline 1.

Figure 6. Density map from FCN-MT: (a) Downtown; (b) Parkway. Top three rows are cloudy; Bottom three rows are sunny.
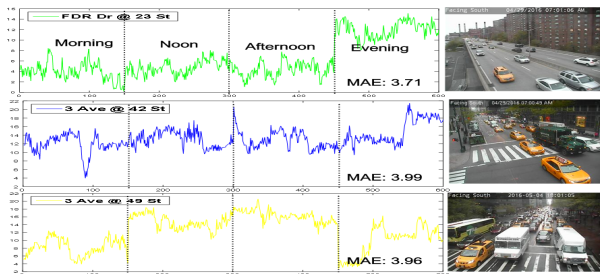

Figure 7. Estimated traffic density from OPT-RC for three cameras. (Left) estimated traffic density curve for each camera, where the X-axis represents frame index and the Y-axis represents traffic density. MAE for each curve is shown at the bottom right of each plot. To show the time series for one day, we select 150 frames for each time interval (morning, noon, afternoon, and evening). (Right) one sample image for each camera.

Special Event Detection. An interesting application of traffic density estimation is to detect the change of traffic
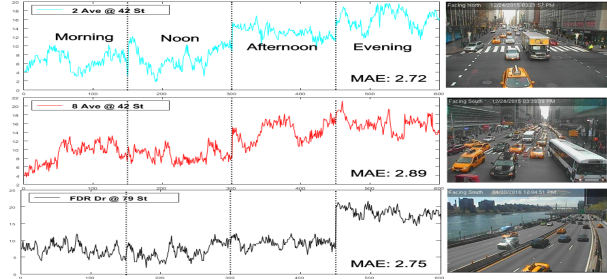

Figure 8. Estimated traffic density from FCN-MT for three cameras. Same setting as Figure 7.
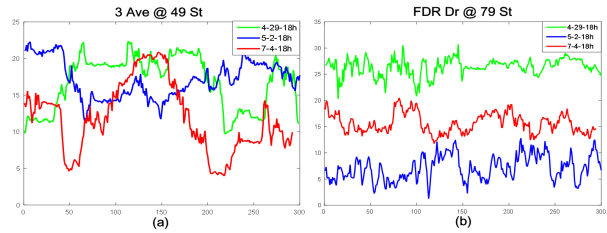

Figure 9. Independence Day traffic density detection. X-axis: frame index. Y-axis: vehicle count.

density when special event happens in the city. To verify the capability to detect such changes of our method, we test FCN-MT on two cameras and multiple days. From the results we find that traffic density on July 4th 18h is different from that on other normal days, as illustrated in Figure 9. For the camera in downtown (3Ave@49st), the traffic density on July 4th is averagely lower than that of other days, and the traffic can be very sparse periodically. This corresponds to the fact that several roads around 3Ave@49st are closed after 3 pm due to fireworks show on Independence Day, resulting in less traffic around. For the camera on parkway (FDR Dr @ 79st), the average traffic is less then Friday (4-29), but more than a normal Monday (5-2). As July 4th is also Monday, it is supposed to have similar traffic as May 2nd. The detected increase of the traffic density on July 4th corresponds to the fact that FDR under 68 St is closed, resulting in more traffic congested to FDR above 68 St. All those observations verify that our method can detect traffic density change when special events happen.

4.3. Quantitative Evaluations on TRANCOS

To verify the efficacy of our methods, we also evaluate and compare the proposed methods with baselines on a public dataset, TRANCOS[25]. TRANCOS provides a collection of 1244 images of different traffic scenes, obtained from real video surveillance cameras, with a total of 46796 annotated vehicles. The objects have been manually annotated using dots. It also provides a region of interest (ROI) per image, defining the region considered for the evaluation. This database provides images from very different scenarios and no perspective maps are provided. The ground truth object density maps are generated by placing a Gaussian

Table 2. Results comparison on TRANCOS dataset

Method	MAE	ARE
Baseline 1	13.76	0.6412
Baseline 2-CCNN	12.49	0.6743
Baseline 2-Hydra	10.99	0.7129
OPT-RC	12.41	0.6674
FCN-ST	5.47	0.827
FCN-MT	5.31	0.856

Table 3. Results comparison on UCSD dataset

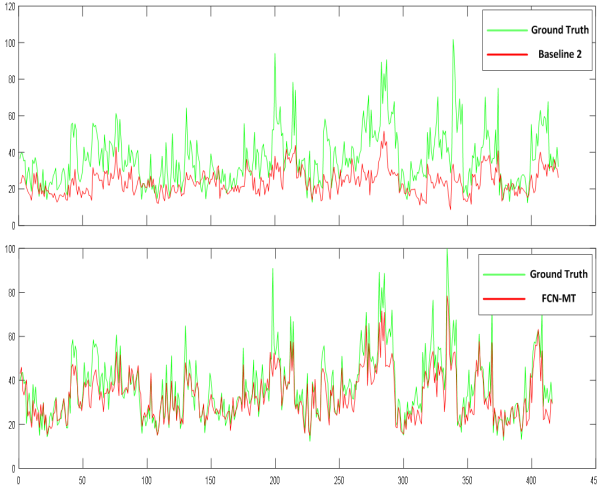| Method	MAE	MSE
Kernel Ridge Regression [1]	2.16	7.45
Ridge Regression [6]	2.25	7.82
Gaussian Process Regression [4]	2.24	7.97
Cumulative Attribute Regression [5]	2.07	6.86
Cross-scene DNN[35]	1.6	3.31
OPT-RC	2.03	5.97
FCN-MT	1.67	3.41

Figure 10. Comparing FCN-MT and Baseline 2-Hydra. X-axis: frame index. Y-axis: vehicle count.

Kernel in the center of each annotated object[16].

We compare our methods with baselines in Table 2. Baseline 1 and OPT-RC have the same settings as evaluated in WebCamT. Baseline 2-CCNN is a basic version of the network in [25], and Baseline 2-Hydra augments the performance by learning a multi-scale non-linear regression model. FCN-ST is the single task implementation (only vehicle density estimation) of FCN-MT for ablation analysis. Baseline 2-CCNN, Baseline 2-Hydra, FCN-ST, and FCN-MT are trained on 823 images and tested on 421 frames following the separation in [25]. From the results, we see that FCN-MT significantly reduces the MAE from 10.99 to 5.31 compared with Baseline 2-Hydra. FCN-MT also outperforms the single task method FCN-ST and verifies the efficacy of multi-task learning. From Figure 10 we can also see that the estimated counts from FCN-MT fit the ground truth better than the estimated counts from Baseline 2. OPT-RC outperforms Baseline 1 and obtains results comparative to Baseline 2.

4.4. Quantitative Evaluations on UCSD Dataset

To verify the generalization of the proposed methods on different counting tasks, we evaluate the proposed methods on crowd counting dataset UCSD[4]. This dataset contains 2000 frames chosen from one surveillance camera, with frame size 158×238 and frame rate $10 fps$. Average number of people in each frame is around 25.

By following the same settings with [4], we use frames from 601 to 1400 as training data, and the remaining 1200 frames as test data. OPT-RC has the same setting as evaluated in WebCamT. ROI mask is used on the input images and combined feature maps in FCN-MT and OPT-RC. Table 3 shows the results of our methods and existing methods. From the results we can see OPT-RC outperforms the non-deep learning based methods, but is less accurate than the deep learning-based methods in[35]. FCN-MT outperforms all the non-deep learning based methods and gets comparative accuracy with the deep learning-based methods in[35]. These results show that though our methods are developed to solve the challenges in webcam video data, they are also robust to other type of counting tasks.

5. Comparison of OPT-RC and FCN-MT

From the extensive experiments, we highlight some difference between OPT-RC and FCN-MT: (i) OPT-RC can learn the geometry information by learning different weight for different blocks in the image. (ii) As the handcrafted SIFT feature is not discriminative enough to distinguish background and foreground, OPT-RC heavily relies on background subtraction. Nevertheless FCN-MT extracts hierarchical feature maps. The combined and re-weighted features are quite discriminative for foreground and background. Thus FCT-MT does not require background subtraction. (iii) The learned density map of FCN-MT suffers from checkerboard artifacts in some cases.

Despite of these differences, FCN-MT and OPT-RC still have strong connection: both methods map the image into vehicle density map and overcome the challenges of webcam video data. FCN-MT replaces the BG subtraction, feature extractor, and block-regressors of OPT-RC with fully convolutional networks. Both methods avoid detecting or tracking individual vehicle, and adapt to different vehicle scales. For the future research, domain transfer learning will be explored to enable the model more robust to multiple cameras.

References

[1] S. An, W. Liu, and S. Venkatesh. Face recognition using kernel ridge regression. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. IEEE, 2007. 8

[2] C. Arteta, V. Lempitsky, and A. Zisserman. Counting in the wild. In *European Conference on Computer Vision*, pages 483–498. Springer, 2016. 2

[3] A. B. C. Rother, V. Kolmogorov. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH*, 2004. 3

[4] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7. IEEE, 2008. 5, 8

[5] K. Chen, S. Gong, T. Xiang, and C. Change Loy. Cumulative attribute space for age and crowd density estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2467–2474, 2013. 8

[6] K. Chen, C. C. Loy, S. Gong, and T. Xiang. Feature mining for localised crowd counting. In *BMVC*, volume 1, page 3, 2012. 8

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016. 3

[8] Y.-L. Chen, B.-F. Wu, H.-Y. Huang, and C.-J. Fan. A real-time vision system for nighttime vehicle detection and traffic surveillance. *IEEE Transactions on Industrial Electronics*, 2011. 2

[9] Z. Chen, T. Ellis, and S. A. Velastin. Vehicle detection, tracking and classification in urban traffic. In *15th International IEEE Conference on Intelligent Transportation Systems*, 2012. 2

[10] T. E. Choe, M. W. Lee, and N. Haering. Traffic analysis with low frame rate camera networks. In *IEEE Conference on Computer Vision and Pattern Recognition-Workshops*, 2010. 2

[11] J. Du and Y.-C. Wu. Network-wide distributed carrier frequency offsets estimation and compensation via belief propagation. *IEEE Transactions on Signal Processing*, 61(23):5868–5877, 2013. 1

[12] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1936. 4

[13] K. Garg, S.-K. Lam, T. Srikanthan, and V. Agarwal. Real-time road traffic density estimation using block variance. In *IEEE Winter Conference on Applications of Computer Vision*, 2016. 2

[14] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 3

[15] W. N. Gonçalves, B. B. Machado, and O. M. Bruno. Spatiotemporal gabor filters: a new method for dynamic texture recognition. *arXiv preprint arXiv:1201.3612*, 2012. 2

[16] R. Guerrero-Gómez-Olmedo, B. Torre-Jiménez, R. López-Sastre, S. Maldonado-Bascón, and D. Onoro-Rubio. Extremely overlapping vehicle counting. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 423–431. Springer, 2015. 8

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 5

[18] S. Hua, J. Wua, and L. Xub. Real-time traffic congestion detection based on video analysis. *J. Inf. Comput. Sci*, 2012. 2

[19] B. S. Kerner. *Introduction to modern traffic flow theory and control: the long road to three-phase traffic theory*. 2009. 1

[20] H. Li and Z. Lin. Accelerated proximal gradient methods for nonconvex programming. In *Advances in neural information processing systems*, pages 379–387, 2015. 4

[21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 3, 4

[22] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 6

[23] G. Mo and S. Zhang. Vehicles detection in traffic flow. In *Sixth International Conference on Natural Computation*, 2010. 2

[24] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015. 5

[25] D. Onoro-Rubio and R. J. López-Sastre. Towards perspective-free object counting with deep learning. In *European Conference on Computer Vision*, pages 615–629. Springer, 2016. 2, 5, 6, 7, 8

[26] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015. 2

[27] R. Shirani, F. Hendessi, and T. A. Gulliver. Store-carry-forward message dissemination in vehicular ad-hoc networks with local density estimation. In *IEEE 70th Vehicular Technology Conference*, 2009. 1

[28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[29] E. Toropov, L. Gui, S. Zhang, S. Kottur, and J. M. F. Moura. Traffic flow from a low frame rate city camera. In *IEEE International Conference on Image Processing*, 2015. 1

[30] A. Z. V. Lempitsky. Learning to count objects in images. In *Advances in Neural Information Processing Systems*, 2010. 2, 6

[31] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008. 6

[32] L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang, and S. Lyu. DETRAC: A new benchmark and protocol for multi-object detection and tracking. *arXiv CoRR*, abs/1511.04136, 2015. 3

[33] F. Xia and S. Zhang. Block-coordinate frank-wolfe optimization for counting objects in images. In *Advances in Neural Information Processing Systems Workshops*, 2016. 2

[34] X.-D. Yu, L.-Y. Duan, and Q. Tian. Highway traffic information extraction from skycam mpeg video. In *IEEE International Conference on Intelligent Transportation Systems*, 2002. 2

[35] C. Zhang, H. Li, X. Wang, and X. Yang. Cross-scene crowd counting via deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 833–841, 2015. 2, 8

[36] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 589–597, 2016. 2

[37] Z. Zhao, H. Li, R. Zhao, and X. Wang. Crossing-line crowd counting with two-phase deep neural networks. In *European Conference on Computer Vision*, pages 712–726. Springer, 2016. 2

[38] Y. Zheng and S. Peng. Model based vehicle localization for urban traffic surveillance using image gradient based matching. In *15th International IEEE Conference on Intelligent Transportation Systems*, 2012. 1