August 22, 2013

RT0951
Computer Science 11 pages

Research Report

Bayesian Unsupervised Vehicle Counting

Takayuki Katasuki, Tetsuro Morimura, Tsuyoshi Idé

IBM Research - Tokyo
IBM Japan, Ltd.
NBF Toyosu Canal Front Building
6-52, Toyosu 5-chome, Koto-ku
Tokyo 135-8511, Japan

Bayesian Unsupervised Vehicle Counting

Takayuki Katasuki, Tetsuro Morimura, and Tsuyoshi Idé
IBM Research - Tokyo
{kats, tetsuro, goodidea}@jp.ibm.com

August 22, 2013

Abstract

This paper defines a novel task, unsupervised vehicle-counting from images, and presents the first practical solution that gives an accuracy comparable to supervised alternatives.

The major application of the task is Web-camera-based city traffic monitoring systems, where vehicle-counting must be done from very low-quality images. In this case, existing object-detection classifiers are impractical because of low resolution, poor viewing angle, and frequent occlusions. Also, the cost of preparing many training images is often prohibitive, since the quality of the images can be too low even for manual vehicle-counting. This calls for an unsupervised and robust approach to vehicle counting.

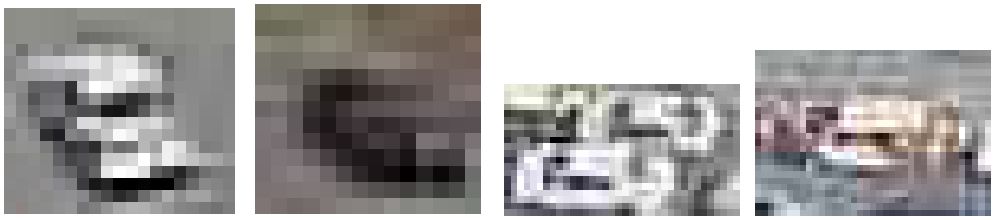
We formalize the problem as a task for Bayesian density estimation, where the number of vehicles is related to the total area of pixels that may correspond to vehicles in an image. We use the infinite Gaussian mixture model with a specific definition on the mean value with the framework of nonparametric Bayes and variational Bayes (VB) for model selection and computational efficiency. Using real-world Web-camera images, we show that the accuracy of the proposed approach is good enough for our application and robust for image quality. To the best of our knowledge, this is the first practical method for counting objects from images without training data.

1 Introduction

Object recognition and extracting information from the results, such as object-counting, are popular tasks in the machine learning and computer vision literature [1, 2, 3]. Video surveillance for Intelligent Transportation Systems is included in this category. A number of cities have started using video cameras as non-intrusive traffic monitoring tools with low costs and high flexibility compared to hardware sensors such as inductive loops embedded in roads. Automatic license plate recognition is a recent successful application [4]. Special-purpose cameras producing high-resolution images are used in most scenarios, but growing attention is being paid to the use of low-cost Internet-linked cameras to address cost and scalability concerns. However, unlike the mature technology of inductive loop sensors, the use of such Web-cameras is still challenging. Figure 1 shows typical Web-camera images of vehicles. We see that the resolution of the images is quite limited, the viewing angle is quite poor, and there are many occlusions.

For the video-based vehicle-counting task, two types of approaches have been proposed in the literature: (1) Individual vehicle recognition, and (2) Qualitative analysis. The first approach attempts to recognize individual vehicles in the images. Examples of existing methods include vehicle and non-vehicle classification [5, 6], and template matching [7, 8, 9, 10, 11]. However, this approach is not very useful in most practical situations, because it is quite sensitive to the image quality and training data set.

To address these shortcomings, the second approach (qualitative analysis) attempts to directly extract relevant metrics from images by skipping the expensive step of vehicle recognition. However, most of the existing methods give only qualitative metrics such as a relative level of congestion, and are not capable of estimating an absolute value of the number of vehicles [12, 13].



(a) example 1 (b) example 2 (c) example 3 (d) example 4

Figure 1: Examples of Web-camera images, which are magnified to capture individual vehicles. See later sections for details.

This paper proposes a new paradigm for the task of *vehicle-counting from images without training data*. We formulate this task based on the framework of density estimation in machine learning. Concretely, we first extract a scalar feature x from an image, and then we compute a probability distribution for the number of vehicles as a function of x. For the definition of x, we use the total area of pixels that may correspond to vehicles in an image. The key idea is to introduce a particular type of Infinite Gaussian Mixture Models (GMM) for the probability distribution of x so that the distribution represents the likelihood of the image features given discrete values for the number of vehicles. The proposed infinite GMM is clearly different from a traditional Infinite GMM [14] because the mean value of the GMM is a linear function of the index number of GMM's component that can directly represent the number of vehicles, in contrast to a traditional GMM whose mean value does not have any meaning. This is a novel alternative for an Infinite GMM.

Due to the unsupervised nature of the task, our method requires no training data for calibration. In addition, thanks to the fully Bayesian treatment, our method is quite robust against low-quality observations. To the best of our knowledge, this is the first work that addresses the task of unsupervised object counting. Our method opens a new door for a practical framework for counting objects without training data.

2 Related work

As mentioned in the Introduction, video-based traffic monitoring systems proposed to date are categorized into two approaches, depending on whether or not they use individual vehicle recognition.

In the first approach, which is based on explicit vehicle recognition, once all of the vehicles are identified in an image, vehicle-counting is a trivial task. For vehicle recognition, existing studies use either image patch classification or template matching. For examples of image patch classification, Choi, Sung, and Yang [5] and Kembhavi, Harwood, and Davis [6] attempted vehicle/non-vehicle classification based on many training images. Examples of template matching include feature tracking for edges and lines characteristic of vehicles [7, 8, 9], as well as targeted recognition of windshields [10] and headlights [10, 11]. As mentioned, these approaches are quite sensitive to the quality of the images [12, 13, 15] and the training data set. Also, preparing the required training images is quite costly. These are clear contrasts to our unsupervised approach.

The second approach, which does not do vehicle recognition as its first step, leads to a simple and robust framework for traffic volume analysis. Unfortunately, existing approaches are still incapable of estimating the numbers of vehicles. They produce only relative metrics such as a congestion level [12, 13, 15].

Our framework for unsupervised vehicle-counting makes two assumptions: (1) The number of vehicles in an image, d, is discrete, and (2) The total Vehicle Pixel Area x is a linear function of the value of d. We translate these assumptions into the infinite GMM having a specific sequence of the mean for x, with the aid of a stick-breaking process (SBP) [16] for d in a fully Bayesian framework.

Infinite GMM and SBPs have been used in the method of nonparametric Bayes [14, 16, 17, 18]. A popular application of nonparametric Bayes can be categorized in the class of clustering tasks, where the number of clusters can be automatically determined from the Bayesian formulation. While our method shares some features with nonparametric Bayesian clustering or density estimation, the task is clearly different in that we give a special meaning to the individual cluster centers. That is, in our task, the density model is designed so that each cluster center corresponds to a particular value of the number of vehicles d. Finally, in our formulation, a new variational

Bayes (VB) algorithm is proposed for efficient inference.

3 Infinite Gaussian Mixture Model for Vehicle-Counting

3.1 Problem Setting

We are given N low-quality images, and each of the images is represented by a feature $\boldsymbol{x}$. Our task is to estimate the numbers of vehicles $\boldsymbol{d} \in \mathbb{N}^N$ for the N images, given a set of image features $\boldsymbol{x}$ without any other training images. Here the n-th dimension of $\boldsymbol{d}$ corresponds to the number of vehicles in the n-th image. In other words, we want to learn the relationship between $\boldsymbol{d}$ and the joint image feature $\boldsymbol{x}$.

3.1.1 Image feature

We assume that each image is represented by a scalar feature called Vehicle Pixel Area (VPA). In this case, $\boldsymbol{x} \in \mathbb{R}^N$. The VPA feature of an image is computed using these steps: First, we manually choose a focus area to monitor for vehicles, as suggested in Figure 2. Then we binarize the focus area using a discriminant analysis technique [19] and count the number of white pixels. This is a raw score for the image feature. Finally, the raw score is normalized to be in $[-1, 1]$ by dividing by half of the maximum raw score in the N images and subtracting one. Note that this feature extraction algorithm works for any frame-rate, even for still images, and is quite robust to the image quality.

3.1.2 Probabilistic vehicle-counting framework

We employ a probabilistic formulation to estimate $\boldsymbol{d}$. For this purpose, we define an indicator vector $\boldsymbol{\eta}$ instead of d, based on the 1-of-K notation. For example, if $\boldsymbol{\eta}_n = [0, 0, 1, 0, ..., 0]^\top$, then the number of vehicles in the n-th image is two, where superscript $\top$ denotes the transpose. Since we do not know the maximum number of vehicles in advance, we assume that the dimension of $\boldsymbol{\eta}_n$ is infinity. Now our goal is to estimate $\boldsymbol{H} \equiv [\boldsymbol{\eta}_1, \boldsymbol{\eta}_2, ..., \boldsymbol{\eta}_N] \in \mathbb{R}^{\infty \times N}$ from the N images.

In the probabilistic formulation, an optimal $\boldsymbol{H}$ is found through the posterior distribution $p(\boldsymbol{H}|\boldsymbol{x})$. Using the model parameters $\boldsymbol{\phi}_{\boldsymbol{x}}$ and $\boldsymbol{\phi}_{\boldsymbol{H}}$ (explicitly defined later), we solve the optimization problem as

$$\begin{aligned} \boldsymbol{H}^* &= \operatorname*{argmax}_{\boldsymbol{H}} p(\boldsymbol{H}|\boldsymbol{x}) \\ &= \operatorname*{argmax}_{\boldsymbol{H}} \int p(\boldsymbol{x}|\boldsymbol{H}, \boldsymbol{\phi}_{\boldsymbol{x}}) p(\boldsymbol{H}|\boldsymbol{\phi}_{\boldsymbol{H}}) p(\boldsymbol{\phi}_{\boldsymbol{x}}, \boldsymbol{\phi}_{\boldsymbol{H}}) \mathrm{d}\boldsymbol{\phi}_{\boldsymbol{x}} \mathrm{d}\boldsymbol{\phi}_{\boldsymbol{H}}, \end{aligned} \tag{1}$$

where $\boldsymbol{H}^*$ denotes an optimal solution, which is the optimal solution from the Bayesian perspective. Notice that this formulation requires no training data. This is extremely useful in practice, since we can avoid the exceedingly time-consuming and costly step of manual vehicle-counting and labeling.

In the following section, we explain in detail the Bayesian density estimation model used to compute $p(\boldsymbol{H}|\boldsymbol{x})$.

3.2 Observation Process for the Image Feature

Figure 2 illustrates the observation process for the VPA. As shown in the figure, we assume that the VPA feature x is a linear function of the number of vehicles d, apart from the additive Gaussian noise represented by ϵ_x. Here the two parameters, $\theta_0 \in \mathbb{R}$ and $\theta_1 \in \mathbb{R}$, are the unknown model parameters to be learned. In the proposed model, we have a different value of the mean of the Gaussian, depending on the value of d, as $f(\theta_0, \theta_1, d) \equiv \theta_1 d + \theta_0$. Since d can take an arbitrary natural number value in $[0, \infty)$, the joint observation process over the entire N images can be written as

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{H}, \theta_0, \theta_1, \beta) &\equiv \prod_{n=1}^{N} \prod_{d=0}^{\infty} \mathcal{N}(x_n; f(\theta_0, \theta_1, d), \beta^{-1})^{\eta_{n,d}} \\ &= \frac{\exp(-\frac{\beta}{2} \sum_{n=1}^{N} \sum_{d=0}^{\infty} \eta_{n,d} (x_n - f(\theta_0, \theta_1, d))^2)}{(2\pi\beta^{-1})^{\frac{N}{2}}}, \end{aligned} \tag{2}$$

where $\mathcal{N}$ denotes a Gaussian distribution whose explicit expression is given in the Appendix. The parameter β (> 0) is the precision (inverse variance) for the observation process, and is treated as an unknown parameter to be learned.

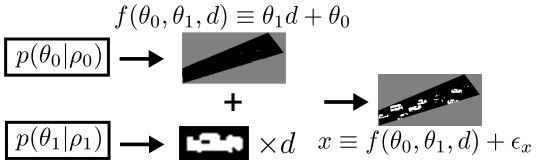


Figure 2: The observation process

This is an infinite GMM with the specific restriction on its mean value by the linear function $f(\theta_0, \theta_1, d)$. The nonparametric Bayesian framework allows us to readily handle the infinity, as described in the next subsection.

3.3 Prior Distributions for the Parameters

In the observation process defined above, we have four parameters, $\boldsymbol{H}$, θ_0, θ_1, and β. We define prior distributions for these parameters for a fully Bayesian treatment.

3.3.1 Stick-breaking process prior

First, we introduce an SBP prior [16] for $\boldsymbol{H}$ using an additional parameter $\boldsymbol{v}$ ($0 \leq v_d \leq 1$) as

$$p(\boldsymbol{H}|\boldsymbol{v}) \equiv \prod_{n=1}^{N} \prod_{d=0}^{\infty} \left(v_d \prod_{k=0}^{d-1} (1 - v_k) \right)^{\eta_{n,d}}. \tag{3}$$

In general, SBPs have the property of automatic determination of model complexity. In our context, the SBP is useful to remove the redundant clusters (see Fig. 3), so that we can obtain the simplest model that fits the data best.

From Eq. (3), we see that, for each component with $\eta_{n,d} = 1$, the probability is given by successively breaking a unit length stick into an infinite number of pieces. The size of each piece is the product of the rest of the stick and an independent generating value v_d.

Regarding the SBP parameter $\boldsymbol{v}$, we use the following hyperprior distribution [18, 20]

$$p(\boldsymbol{v}|\alpha) \equiv \prod_{d=0}^{\infty} \mathrm{Beta}(v_d|1, \alpha), \tag{4}$$

where Beta is the beta distribution (see the Appendix for its explicit definition), and α (> 0) is a hyperparameter controlling the degree of sparseness of SBP and also to be learned. Note that in the SBP formulation with VB the infinite dimension of the model is replaced with a finite (large) value when implementing the algorithm (see the section on the experimental results).

3.3.2 Conjugate priors for other parameters

Regarding prior distributions for θ_0, θ_1, and β, we simply use the conjugate priors:

$$p(\theta_0|\rho_0) \equiv \mathcal{N}(\theta_0|\mu_{\theta_0}^{(0)}, \rho_0), \quad p(\theta_1|\rho_1) \equiv \mathcal{N}(\theta_1|\mu_{\theta_1}^{(0)}, \rho_1), \quad p(\beta) \equiv \mathrm{Gamma}(\beta|a_\beta^{(0)}, b_\beta^{(0)}), \tag{5}$$

where Gamma represents the Gamma distribution (see the Appendix for explicit definitions), and the parameters $\mu_{\theta_0}^{(0)}$, $\mu_{\theta_1}^{(0)}$, $a_\beta^{(0)}$, and $b_\beta^{(0)}$ are treated as input parameters given as a part of the model. Here the superscript (0) indicates that these parameters are used for the initial values of the VB procedure.

Finally, we define the hyperprior distributions for α, ρ_0, and ρ_1 using the conjugate priors:

$$p(\rho_0, \rho_1, \alpha) \equiv \mathrm{Gamma}(\rho_0|a_{\rho_0}^{(0)}, b_{\rho_0}^{(0)})\, \mathrm{Gamma}(\rho_1|a_{\rho_1}^{(0)}, b_{\rho_1}^{(0)})\, \mathrm{Gamma}(\alpha|a_\alpha^{(0)}, b_\alpha^{(0)}), \tag{6}$$

where $a_{\rho_0}^{(0)}$, $b_{\rho_0}^{(0)}$,$a_{\rho_1}^{(0)}$, $b_{\rho_1}^{(0)}$,$a_\alpha^{(0)}$, and $b_\alpha^{(0)}$ are input parameters. For the values we used, see the section on the experimental results.

From the above, the distribution of $\boldsymbol{x}$ marginalized over all of the parameters $p(\boldsymbol{x})$ will resemble Figure 3. It has infinite number of equally-spaced clusters having variances depending on the values of d.

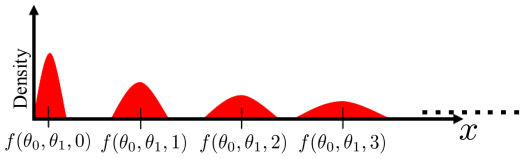


Figure 3: The proposed infinite Gaussian mixture model for Vehicle-Counting

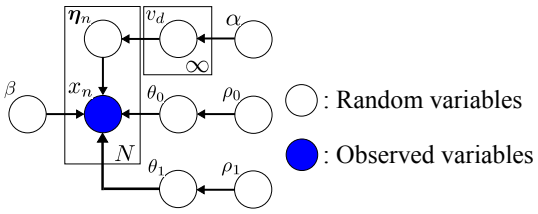


Figure 4: The graphical model

We can summarize the proposed generative model including all of the parameters as shown in Fig. 4. First, α, ρ_0, ρ_1, and β are generated, after that $\boldsymbol{v}$, θ_0, and θ_1 are generated using α, ρ_0, and ρ_1, and then the number of vehicles $\boldsymbol{\eta}_n$ is generated using $\boldsymbol{v}$. Finally, observation variable x_n is generated according to the observation process using $\boldsymbol{\eta}_n$, β, θ_0, and θ_1. The dimension of the parameters of the proposed infinite Gaussian mixture is *not* infinity, which is in contrast to previous nonparametric Bayesian formulations [14, 16, 17, 18]. This nature makes it possible to give a special meaning to the value of the individual cluster centers.

4 Posterior Inference for Vehicle-Counting

4.1 Joint Distribution

The joint distribution for all of the random variables $\boldsymbol{z} \equiv \{\boldsymbol{H}, \theta_0, \theta_1, \beta, \rho_0, \rho_1, \boldsymbol{v}, \alpha\}$ as well as $\boldsymbol{x}$ can now be explicitly given as

$$p(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{x}|\boldsymbol{H}, \theta_0, \theta_1, \beta)p(\boldsymbol{H}|\boldsymbol{v})p(\theta_0|\rho_0)p(\theta_1|\rho_1)p(\boldsymbol{v}|\alpha)p(\beta)p(\rho_0, \rho_1, \alpha). \tag{7}$$

In this section, we derive the relevant marginal and conditional distributions such as the posterior distribution $p(\boldsymbol{z}|\boldsymbol{x})$.

4.2 Variational Bayes Solution

As mentioned earlier, our goal is to obtain $p(\boldsymbol{H}|\boldsymbol{x})$. While it is not possible to obtain an exact analytical solution, an approximated analytics solution can be found through a VB algorithm [21].

The starting point of the VB approach is to assume a trial distribution $q(\boldsymbol{z})$ that approximates the true posterior in a factorized form:

$$q(\boldsymbol{z}) \equiv q(\boldsymbol{H})q(\theta_0, \theta_1)q(\beta, \rho_0, \rho_1, \boldsymbol{v})q(\alpha). \tag{8}$$

Then we identify the optimal trial distribution that minimizes the Kullback-Leibler (KL) divergence between the trial and the true distributions. Finally, in a popular approach of VB [22], we solve the updating equations as

$$q^{(0)}(z_i) \equiv p(z_i), \tag{9}$$

$$q^{(t+1)}(z_i) \propto \exp\langle \ln p(\boldsymbol{z}|\boldsymbol{x})\rangle_{\prod_{j\neq i} q^{(t)}(z_j)}, \tag{10}$$

where the angle brackets $\langle\bullet\rangle_\circ$ denote the expectation of $\bullet$ with respect to a distribution $\circ$.

(a) Nationkimathi (b) Westistg (c) Ukulimamombasard (d) Haileselasie (e) Harambeetaifa

Figure 5: Traffic monitoring Web-camera images [23].

From Eqs. (9) and (10), the trial distribution of $\boldsymbol{H}$ is given as

$$q^{(t)}(\boldsymbol{H}) = \prod_{n=1}^{N} \mathrm{Discrete}(\boldsymbol{\eta}_n; \boldsymbol{\mu}_{\boldsymbol{\eta}_n}^{(t)}), \quad \text{where} \quad \boldsymbol{\mu}_{\boldsymbol{\eta}_n}^{(t)} \equiv [\mu_{\eta_{n,1}}^{(t)}, \mu_{\eta_{n,2}}^{(t)}, ...\mu_{\eta_{n,\infty}}^{(t)}], \tag{11}$$

and Discrete is the discrete distribution (see the Appendix for the explicit definition). The trial distributions of θ_0, θ_1, β, ρ_0. ρ_1, $\boldsymbol{v}$, and α are given as:

$$q^{(t)}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{\mu}_{\boldsymbol{\theta}}^{(t)}, \boldsymbol{\Sigma}_{\boldsymbol{\theta}}^{(t)}), \quad \text{where} \quad \boldsymbol{\theta} \equiv [\theta_0, \theta_1], \tag{12}$$

$$\begin{aligned} q^{(t)}(\beta, \rho_0, \rho_1, \boldsymbol{v}) = \mathrm{Gamma}(\beta; a_\beta^{(t)}, b_\beta^{(t)})\, \mathrm{Gamma}(\rho_0; a_{\rho_0}^{(t)}, b_{\rho_0}^{(t)}) \\ \times \mathrm{Gamma}(\rho_1; a_{\rho_1}^{(t)}, b_{\rho_1}^{(t)}) \left[\prod_{d=0}^{\infty} \mathrm{Beta}(v_d; a_{v_d}^{(t)}, b_{v_d}^{(t)}) \right], \quad \text{and} \end{aligned} \tag{13}$$

$$q^{(t)}(\alpha) = \mathrm{Gamma}(\alpha; a_\alpha^{(t)}, b_\alpha^{(t)}). \tag{14}$$

To solve the updating equations, Eqs. (9) and (10), we first compute $q^{(t+1)}(\boldsymbol{H}, \theta_0, \theta_1, \beta)$ using $q^{(t)}(\rho_0, \rho_1, \boldsymbol{v}, \alpha)$. Then we compute $q^{(t+1)}(\rho_0, \rho_1, \boldsymbol{v})$ using $q^{(t+1)}(\boldsymbol{H}, \theta_0, \theta_1, \beta) q^{(t)}(\alpha)$. Finally we compute $q^{(t+1)}(\alpha)$ using $q^{(t+1)}(\boldsymbol{H}, \theta_0, \theta_1, \beta, \rho_0, \rho_1, \boldsymbol{v})$. Here, we simply compute only the parameters of these distributions because we can compute the expectations in Eq. (9) and (10) analytically, thanks to conjugate modeling. The specific update equations of the parameters are omitted here due to space limitations.

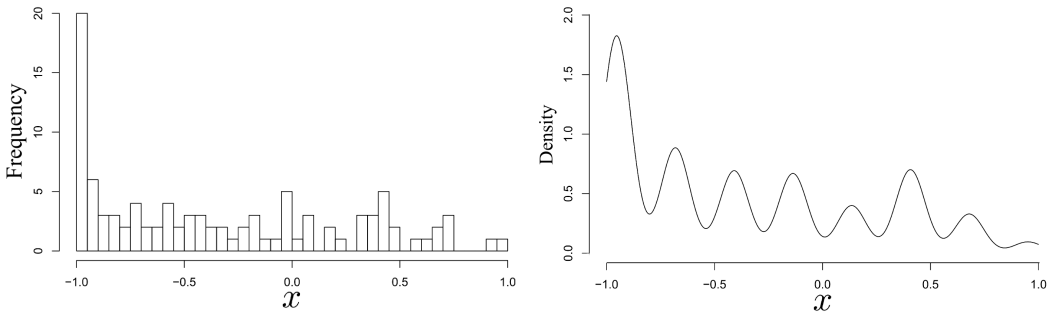
For the initial parameters for θ_0, θ_1, β, ρ_0, ρ_1, $\boldsymbol{v}$, and α, we use the same values as those of the priors. Based on the VB updates, we obtain the final outcome as $q^{(\infty)}(\boldsymbol{z})$, which corresponds to an approximation of the posterior $p(\boldsymbol{z}|\boldsymbol{x})$. In practice, we stop the VB iterations when this condition is satisfied:

$$\frac{(D_{\mathrm{KL}}(p(\boldsymbol{z}) \| q^{(t+1)}(\boldsymbol{z})) - D_{\mathrm{KL}}(p(\boldsymbol{z}) \| q^{(t)}(\boldsymbol{z})))^2}{D_{\mathrm{KL}}(p(\boldsymbol{z}) \| q^{(t)}(\boldsymbol{z}))^2} < 10^{-10}. \tag{15}$$

5 Experimental Results

5.1 Data set

We tested our vehicle-counting method using real-world Web-camera images captured in Nairobi, Kenya [23], as shown in Figure 5. The images were captured at five different road locations with the same size of 640×480 pixels. The number of images was $N = 100$ for each location. As indicated by the rectangles in the figure, we manually defined a focus area in each location. The average number of pixels in the focus areas is about 10 000, and only several hundred of the pixels are occupied on average by individual vehicles. This means that each vehicle is represented by roughly 16×12 pixels, as shown in Fig. 1. It is clear that the resolution is too low for existing object recognition approaches [1, 24, 25].



(a) observed histogram of x (b) estimated density model of x

Figure 6: Observed histogram and estimated density of $\boldsymbol{x}$

5.2 Hyperparameters

For the VB inference, we set the SBP dimensions D to be 100, which is equivalent to N. Also, we used hyperparameter values of $a_\beta^{(0)} = a_{\rho 0}^{(0)} = a_{\rho 1}^{(0)} = a_\alpha^{(0)} = 1$ and $b_\beta^{(0)} = b_{\rho 0}^{(0)} = b_{\rho 1}^{(0)} = b_\alpha^{(0)} = 10^{-10}$. In addition, we used $\mu_{\theta_0}^{(0)} = -1$ and $\mu_{\theta_1}^{(0)} = 0.3$. We chose them to be as "non-informative" as possible in a fully Bayesian framework and to have a quite flat distribution. Also, preliminary experiments show the accuracy of the algorithm is insensitive to changes in the values of the hyperparameters.

5.3 Comparison of estimation errors

Table 1 compares our unsupervised approach with some supervised methods, simple linear regression alternatives, which are based on the estimators of least squares (LS), least median of squares (LMS), least absolute values (LAV), and MM-estimator (MM), and the object recognition approach by Viola and Jones [24] (VJ). We omit the details of these alternatives due to space limitations. Notice that the comparison is disadvantageous to our method, because these supervised alternatives *require additional labeled training images*. We used additional 100 images for each location for LS, LMS, LAV, and MM, on top of the $N = 100$ data sets. Also, we used additional $1,000$ images for each positive and negative examples in the VJ training, where the training data-set used for the VJ is popular benchmarking images and additional manual labeled several hundred images. For the training of the supervised alternatives, manual vehicle-counting and labeling were used to create the labeled images, which took almost several days to prepare. In our method, the computational time for VB inference was only about several seconds on a moderate laptop computer.

As an accuracy metric, we used the relative mean absolute error (RMAE) defined as

$$\mathrm{RMAE} = \frac{1}{N} \sum_{n=1}^{N} \frac{|d_{\mathrm{true}}^{(n)} - d_{\mathrm{estimate}}^{(n)}|}{d_{\mathrm{true}}^{(n)} + 1}, \tag{16}$$

where $d_{\mathrm{true}}^{(n)}$ is the true number of vehicles in the n-th image, and $d_{\mathrm{estimate}}^{(n)}$ is the estimated number of vehicles for the n-th image.

From Table 1, we see that the overall performance of our method is comparable to or even better than the supervised alternatives. This is rather surprising, because our method does not use any training data. In addition, our method gives quite stable RMAE scores for the various camera locations in contrast to the supervised alternatives, which have significantly worse scores at the Nationkimathi due to outliers and occlusions, and also achieves the lowest standard deviation value of RMAE. This result clearly demonstrates the robustness of our approach.

Finally, for a reality check of the VB inference, Fig. 6 shows a comparison between the estimated $p(\boldsymbol{x})$ and the true distribution created from the data. To get $p(\boldsymbol{x})$, we marginalized all of the parameters and variables except for $\boldsymbol{x}$. The result confirms that the estimated density is consistent with the true observed histogram.

Table 1: Comparison between the proposed unsupervised method and supervised alternatives in terms of RMAE (smaller is better).

		proposed	LS	LMS	LAV	MM	VJ
	training data size	0	100	100	100	100	1000
RMAE for each road	nationkimathi	0.226	1.29	0.146	0.924	0.957	0.447
	westistg	0.245	0.391	0.214	0.198	0.209	0.245
	ukulimamombasard	0.216	0.171	0.206	0.201	0.188	0.289
	haileselasie	0.182	0.111	0.188	0.118	0.163	0.166
	harambeetaifa	0.179	0.153	0.078	0.105	0.138	0.406
Average		0.208	0.424	0.166	0.309	0.331	0.311
Standard deviation		0.026	0.446	0.050	0.310	0.314	0.103

6 Concluding Remarks

In this paper, we defined a novel task, *unsupervised vehicle-counting from images*, and proposed an effective framework for this task based on Bayesian density estimation using the infinite Gaussian mixture model with a specific definition on the mean value. Surprisingly, our fully unsupervised approach without any training data was comparable to or even better than some supervised alternatives. Using real-world data, we demonstrated that our approach is quite robust to the quality of images.

For future work, improving the feature extraction step would be an interesting research area. Although we used a simple feature by thresholding method in this paper, we can include many other features into the VB formulation. Also, applying the proposed approach to other applications, such as crowd counting and cell counting would be another interesting topic.

Appendix

Definitions

We give the definitions of the gamma, beta, and Gaussian distributions:

$$\mathrm{Gamma}(x; a, b) \equiv \frac{b^a}{\Gamma(a)} x^{a-1} \mathrm{e}^{-bx} \quad (x > 0),$$

$$\mathrm{Beta}(x; a, b) \equiv \frac{1}{B(a, b)} x^{a-1} (1-x)^{b-1} \quad (0 < x < 1),$$

$$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \equiv |2\pi\boldsymbol{\Sigma}|^{-\frac{1}{2}} \mathrm{e}^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \quad (\boldsymbol{x} \in \mathbb{R}^N),$$

where Γ and B denote the gamma and beta functions, respectively. Also, $|\bullet|$ denotes the determinant of a given matrix. We also define the discrete distribution with the parameters $\{\mu_d | d = 1, ..., D\}$:

$$\mathrm{Discrete}(\boldsymbol{x}; \boldsymbol{\mu}) \equiv \prod_{d=1}^{D} \mu_d^{x_d},$$

where $\sum_{d=1}^{D} x_d = 1$ for $x_d \in \{0, 1\}$, and $\sum_{d=1}^{D} \mu_d = 1$ for $0 \leq \mu_d \leq 1$.

In these definitions, the variables are not related to the variables that appear in the main text.

References

[1] Stan Z Li, ZhenQiu Zhang, Harry Shum, and HongJiang Zhang. Floatboost learning for classification. In *Advances in neural information processing systems*, volume 15, 2002.

[2] Pierre Moreels and Pietro Perona. Common-frame model for object recognition. In *Advances in neural information processing systems*, pages 953–960, 2004.

[3] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *Advances in neural information processing systems*, 2010.

[4] N. Buch, S.A. Velastin, and J. Orwell. A review of computer vision techniques for the analysis of urban traffic. *IEEE Trans. on Intelligent Transportation Systems*, 12(3):920 –939, 2011.

[5] Jae-Young Choi, Kyung-Sang Sung, and Young-Kyu Yang. Multiple vehicles detection and tracking based on scale-invariant feature transform. In *Proc. IEEE Intl Conf. Intelligent Transportation Systems*, pages 528 –533, 2007.

[6] A. Kembhavi, D. Harwood, and L.S. Davis. Vehicle detection using partial least squares. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(6):1250 –1265, 2011.

[7] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 97)*, pages 495 –501, 1997.

[8] Z. Kim and J. Malik. Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking. In *Proc. IEEE Intl. Conf. on Computer Vision*, pages 524 –531 vol.1, 2003.

[9] Xinyu Liu, Danya Yao, Li Cao, Lihui Peng, and Zuo Zhang. A feature-based real-time traffic tracking system using spatial filtering. In *Proc. IEEE Intl. Conf. on Intelligent Transportation Systems*, pages 514 –518, 2001.

[10] K. Robert. Video-based traffic monitoring at day and night vehicle features detection tracking. In *Proc. IEEE Intl. Conf. on Intelligent Transportation Systems*, pages 1 –6, 2009.

[11] Yen-Lin Chen, Bing fei Wu, Hao-Yu Huang, and Chung-Jui Fan. A real-time vision system for nighttime vehicle detection and traffic surveillance. *IEEE Trans. on Industrial Electronics*, 58(5):2030 –2044, 2011.

[12] S. Hu, J. Wu, and L. Xu. Real-time traffic congestion detection based on video analysis. *Journal of Information and Computational Science*, 9(10):2907 – 2914, 2012.

[13] Xiao-Dong Yu, Ling-Yu Duan, and Qi Tian. Highway traffic information extraction from skycam mpeg video. In *Proc. IEEE Intl. Conf. on Intelligent Transportation Systems*, pages 37 – 42, 2002.

[14] Carl Edward Rasmussen. The infinite gaussian mixture model. *Advances in neural information processing systems*, 12(5.2):2, 2000.

[15] S. Santini. Analysis of traffic flow in urban areas using web cameras. In *Fifth IEEE Workshop on Applications of Computer Vision*, pages 140 –145, 2000.

[16] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[17] Thomas S. Ferguson. A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics*, 1(2):209–230, 1973.

[18] David M. Blei and Michael I. Jordan. Variational inference for dirichlet process mixtures. *Bayesian Analysis*, 1:121–144, 2005.

[19] N. Otsu. A threshold selection method from gray-level histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, 9(1):62 –66, jan. 1979.

[20] M. West. Hyperparameter estimation in Dirichlet process mixture models. *Duke University Technical Report*, 92-A03, 1992.

[21] Hagai Attias and London Wcn Ar. Inferring parameters and structure of latent variable models by variational bayes. In *Proc. of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 21–30. Morgan Kaufmann Publishers, 1999.

[22] C. Bishop, D. Spiegelhalter, and J. Winn. VIBES: A variational inference engine for Bayesian networks. In *Advances in Neural Information Processing Systems 15*, pages 777–784. MIT Press, 2003.

[23] AccessKenya.com. http://traffic.accesskenya.com/.

[24] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[25] Pablo Negri, Xavier Clady, Shehzad Muhammad Hanif, and Lionel Prevost. A cascade of boosted generative and discriminative classifiers for vehicle detection. *EURASIP Journal on Advances in Signal Processing*, 2008:136, 2008.