EXAMPLE-BASED VISUAL OBJECT COUNTING WITH A SPARSITY CONSTRAINT

Yi Wang[1], *Y. X. Zou*[1]*, *Jin Chen*[1], *Xiaolin Huang*[1], *Cheng Cai*[2]

[1]ADSPLAB/ELIP, School of ECE, Peking University, Shenzhen, 518055, China
[2]Department of Computer Science, College of Information Engineering, Northwest A&F University, Yangling, 712100, China
*Corresponding author: zouyx@pkusz.edu.cn

ABSTRACT

For existing mainstream visual object counting (VOC) methods, training data insufficiency will lead to significant performance degradation. To address this challenge, we propose a novel sparsity-constrained example-based VOC method. Given a test image, its counts are estimated by integrating over its density map, and our method will predict such density map based on patch using training examples. Specifically, image patches and their counterpart density maps generated from annotated training images share similar local geometry on manifolds. Such local geometry can be captured by locally linear embedding (LLE) only when data are well-sampled. However, training data are poorly sampled due to their insufficiency. To handle this problem, we impose sparsity on the local optimization based on LLE, where the chosen examples favor the similar structure of input patches. Extensive experiments on public datasets demonstrate the effectiveness and competitiveness of our method by using simple features and a few training images.

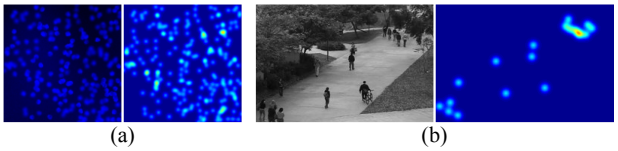
Index Terms—visual object counting, example-based, locally linear embedding, sparsity constraint, nearest neighbors

1. INTRODUCTION

Visual object counting (VOC) is to estimate the quantity of objects in the image or video. It is a highly demanded real-world application and can be widely applied to various fields like wildlife census, crowd analysis, traffic surveillance, etc.

VOC is full of challenges as the objects of interest often overlap with each other. It is difficult for conventional object detection based VOC methods to make a fairly good prediction under such circumstances. These methods cast the original counting problem as the individual object detection, which is also a difficult task to solve when objects get crowded.

To handle crowded scenes in VOC problem, so far, most mainstream VOC methods are based on global regression and object density estimation in a supervised way. For global regression-based VOC (GR-VOC) methods [1, 2, 6-8], they devoted to learning an effective mapping between the global features extracted from images and their corresponding counts. Usually such features fuse many well-designed low-level characteristics like segment features, edge features, etc [6]. And the mapping is learnt by miscellaneous regression algorithms such as ridge regression (RR) [1], Gaussian process regression (GPR) [6], etc. For object density estimation based VOC (DE-VOC) methods [3, 7, 9], they do the counting by estimating an image density whose integral over any image region yields the object counts within that region [3]. Lempitsky firstly proposed such DE-VOC through pixel-wise object density map regression. Zhou extended Lempitsky's work by making it practical for arbitrary objects and scenes [9], and Fiaschi applied random regression trees to improve training efficiency [7]. Given sufficient training images, GR-VOC and DE-VOC are both effective.



(a) (b)

Fig. 1. Images with objects and their corresponding density maps (displayed in jet colormap). (a) cell image; (b) pedestrian image in dataset UCSD

However, manually annotated training images are often insufficient in real application. This happens when VOC is applied to a new unseen scene because the annotation of training data is time and labor consuming. Under such conditions, both GR-VOC and DE-VOC face the significant degradation on performance. To address this challenge, we propose a sparsity-constrained example-based VOC (SE-VOC) method by estimating the object density over the generalization of a few training images. Specifically, our method is motivated by manifold learning, especially locally linear embedding (LLE) [10, 11]. From the observation (Figure 1) that images in the datasets we use share high similarity with their counterpart density maps on object shapes and distribution in spatial space, which suggests the reasonable assumption that the manifold of patches extracted from images share the similar local geometry with the manifold of their corresponding patches extracted from counterpart generated density maps in two feature spaces. For LLE, such local geometry can be well characterized by the relationship between feature vectors in the same neighborhood with well-sampled data, and the object density map can be reconstructed by preserving such local geometry. In order to compute local geometry with under-sampled data, a quadratic program constrained by locality and sparsity simultaneously is formulated. Considering its efficient solution, we develop an algorithm called approximated sparsity-constrained example-based VOC (ASE-VOC) method. The pipeline of ASE-VOC is shown in Figure 2.

This work was partially supported by the Shenzhen Science & Technology Fundamental Research Program (No: JCYJ20150430162332418).

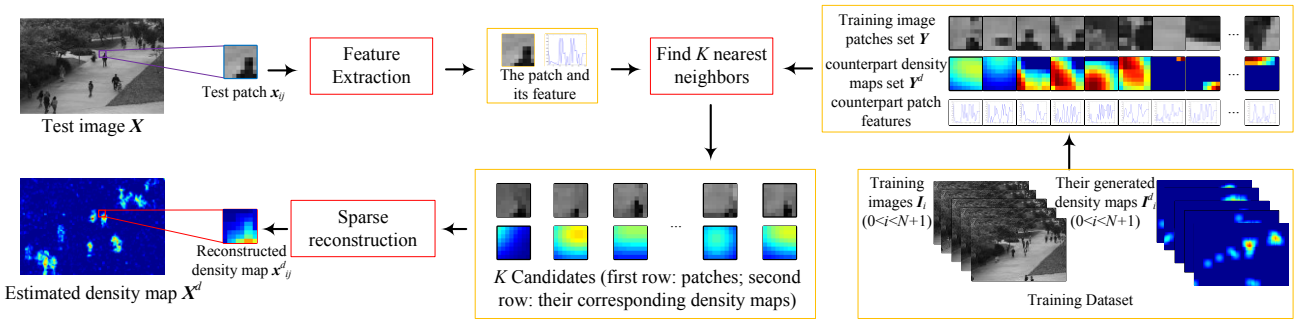


Fig. 2. The pipeline of our proposed method. Red boxes stand for operations and orange boxes stand for data.

The rest of paper is organized as follows: in Section 2, we elaborate the problem formulation of SE-VOC and propose an effective and efficient method to solve it; Section 3 presents solid experiments and experimental analysis both on cell and pedestrian data; Section 4 concludes our work and future plan.

2. METHOD

2.1. Ground truth density map for training

Following the work in [3, 9], our method demands a set of N training images $\boldsymbol{I}_1, \boldsymbol{I}_2, \dots, \boldsymbol{I}_N$. For each image $\boldsymbol{I}_i (1 \le i \le N)$, all objects presented in it are assumed to be annotated with a set of 2D points $\boldsymbol{P}_i$. So the ground truth density function for every pixel $p \in \boldsymbol{I}_i$ is defined as a sum of 2D Gaussian kernels based on the annotated points:

$$F_i^o(p) = \sum_{P \in \boldsymbol{P}_i} \mathcal{N}(p; P, \delta^2) \tag{1}$$

where P is a user-annotated dot and δ is the smoothness parameter. δ is set to 3 for all experiments in Section 3. With the definition in Eqn. (1), the ground truth density map $\boldsymbol{I}_i^d$ of training image $\boldsymbol{I}_i$ is defined as

$$\forall p \in \boldsymbol{I}_i^d, \boldsymbol{I}_i^d(p) = F_i^o(p) \tag{2}$$

Some instances of $\boldsymbol{I}_i^d$ are displayed in Figure 1.

With the density map $\boldsymbol{I}_i^d$, the object count $c(\boldsymbol{I}_i)$ can be computed by integrating over the density map

$$c(\boldsymbol{I}_i) = \sum_{p \in \boldsymbol{I}_i^d} \boldsymbol{I}_i^d(p) \tag{3}$$

For our method, training data are desired in patch form. Consequently, a set of image patches $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_M\}$ $(\boldsymbol{y}_i \in \mathbb{R}^{n \times 1})$ are extracted from the training images $\boldsymbol{I}_i$, $i \in \{1,2,\dots,N\}$, and the density maps set $\boldsymbol{Y}^d = \{\boldsymbol{y}_1^d, \boldsymbol{y}_2^d, \dots, \boldsymbol{y}_M^d\}$ of corresponding patches are extracted from $\boldsymbol{I}_i^d$, $i \in \{1,2,\dots,N\}$. All patches from $\boldsymbol{Y}$ and $\boldsymbol{Y}^d$ can be seen as feature vectors in two feature spaces, respectively. By using Eqn. (3) with $\boldsymbol{y}_i^d$, the object count $c(\boldsymbol{y}_i)$ of every image patch $\boldsymbol{y}_i$ can be computed.

2.2. Formulation of example-based VOC

The main objective for GR-VOC and DE-VOC is to learn the mappings $\mathcal{F}$: $g(\boldsymbol{I}_i) \to c(\boldsymbol{I}_i)$ and $\mathcal{G}$: $h(\boldsymbol{I}_i) \to \boldsymbol{I}_i^d$, respectively, where g is usually a fused features extractor and h is a dense features extractor. Different from GR-VOC or DE-VOC methods, we learn to estimate the density of input image patch over the generalization of a few images. As assumed that two manifolds, which are formed by image patches and their counterpart density maps respectively, share the similar local geometry. Inspired by LLE, such local geometry of a feature vector can be characterized by how the feature vector can be linearly reconstructed by its neighbors. For example, given a test image patch $\boldsymbol{x}$ with unknown density, we compute its reconstruction weights of its neighbors chosen from $\boldsymbol{Y}$ by minimizing the reconstruction error. Then the density map $\boldsymbol{x}^d$ can be estimated by applying the reconstruction weights to the density maps of neighboring patches from $\boldsymbol{Y}^d$. In VOC, we name this method as example-based VOC (E-VOC). Similar to the formulation in [10, 11], E-VOC can be modeled as:

$$\forall i \in \{1,2,\dots,K\}, w_i^* = \arg\min_{w_i} ||\boldsymbol{x} - \sum_{\tilde{\boldsymbol{y}}_i \in \tilde{\boldsymbol{Y}}} w_i \tilde{\boldsymbol{y}}_i||_2^2 \tag{4}$$

$$\text{s.t.} \quad \forall \tilde{\boldsymbol{y}}_i \in \tilde{\boldsymbol{Y}}, D\big(f(\boldsymbol{x}), f(\tilde{\boldsymbol{y}}_i)\big) \le \varepsilon;$$
$$\forall \boldsymbol{y} \in \boldsymbol{Y} - \tilde{\boldsymbol{Y}}, D\big(f(\boldsymbol{x}), f(\boldsymbol{y})\big) \ge \varepsilon; \sum_{i=1}^{K} w_i = 1 \tag{5}$$

$$\boldsymbol{x}^d \cong \sum_{\tilde{\boldsymbol{y}}_i^d \in \tilde{\boldsymbol{Y}}^d} w_i^* \tilde{\boldsymbol{y}}_i^d \tag{6}$$

where $\tilde{\boldsymbol{Y}}$ is a training patch subset formed by the K nearest neighbors of $\boldsymbol{x}$ from $\boldsymbol{Y}$, and $\tilde{\boldsymbol{y}}_i$ is a training patch belonging to $\tilde{\boldsymbol{Y}}$. $f(.)$ is a geometrical features extractor. $\varepsilon > 0$ and its value ensures that $\tilde{\boldsymbol{Y}}$ only contains K elements, which suggests constraints in Eqn. (5) are intended to choose K nearest neighbors of $\boldsymbol{x}$ from $\boldsymbol{Y}$ to form $\tilde{\boldsymbol{Y}}$. $D(.)$ is a similarity measurement function. $\tilde{\boldsymbol{y}}_i^d$ is the density map of $\tilde{\boldsymbol{y}}_i$.

In E-VOC, Eqn. (5) defines the neighborhood of $\boldsymbol{x}$. With the neighbors in that neighborhood, Eqn. (4) obtains the local geometry of $\boldsymbol{x}$, and Eqn. (6) reconstructs the target density map $\boldsymbol{x}^d$ by preserving such local geometry.

As its constrained least squares form, Eqn. (4) has an analytic solution ($K < n$) and w_i can be solved efficiently. The implementation of constraints in Eqn. (5) is usually realized by K-nearest-neighbors (KNN) algorithm.

2.3. Sparsity-constrained example-based VOC (SE-VOC)

E-VOC can work well with a few training images, but its results are unstable and not accurate enough compared with GR-VOC and DE-VOC. It is caused by the following two factors: 1) when the training data are sampled sufficiently, E-VOC works well but its performance is affected by the neighborhood size. Such neighborhood size is preset by adjusting ε or K. As the Figure 3 illustrated, if K is too small, neighbors selected are not enough to characterize the local

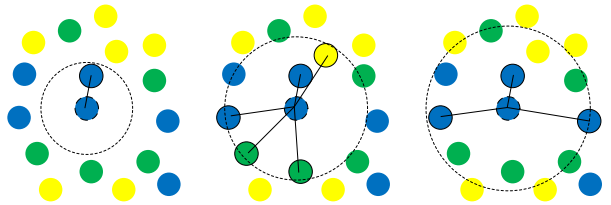


Fig. 3. The illustration of the impact about the number of neighbors K on reconstruction process. Solid Circles filled with same colors are of same geometry, different colors are of different geometries. (a) K is too small; (b) K is large; (c) automatic selection based on sparsity constraint.

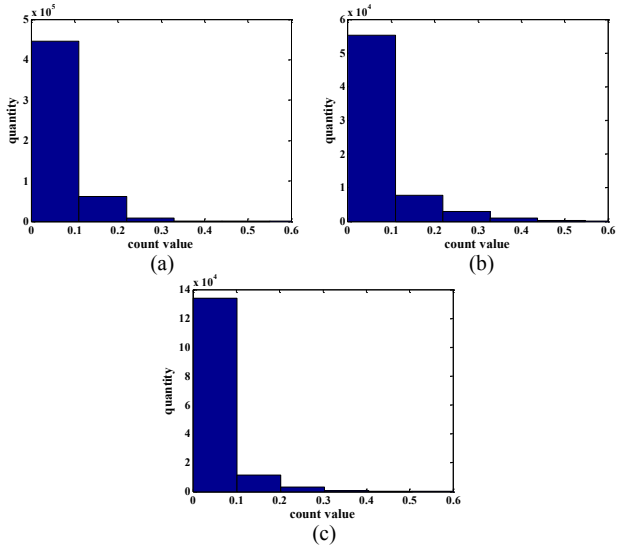


Fig. 4. The distribution about the quantity of image patches (patch size: 4×4, step size: 2) according to their counts (computed by their density maps) extracted from 32 randomly chosen images of (a) cell [3], (b) UCSD [6], (c) Mall [1] datasets.

geometry (Figure 3(a)); on the contrary, E-VOC tends to select neighbors with different geometries, thus it cannot capture the desired local geometry for reconstructing the density map (Figure 3(b)). 2) Our statistics on datasets used shows that the distribution about the quantity of image patches is highly sparse and imbalanced based on their counts. Thus nearest neighbors would fail to depict the local geometry. Just as illustrated in Figure 4, the higher the count value (also can be viewed as object density) is, the fewer the quantity of patches will be, so the amount of object-crowded regions in datasets are limited, which means object-crowded regions are under-sampled. According to [10, 12], LLE only works when data are well-sampled. Therefore, it is hard to ensure that E-VOC deriving from LLE can capture local geometry of the input patch under such circumstances (which is similar to Figure 3(b)).

To address the problems caused by these two factors, inspired by the properties of sparsity and its applications in manifold learning [12-14], we improve the model in Eqn. (4)-(5) by imposing the locality and sparsity simultaneously on searching neighbors. This will encourage opting nearby feature vectors of input one as fewer as possible with the same or similar geometry in feature space (Figure 3(c)). Through the improved model, we can learn proper local geometry with under-sampled data and avoid specifying the neighborhood size at the same time.

Specifically, we reformulate the optimization problem in Eqn. (4)-(5) by constraining sparsity on neighbors selection as:

$$\min ||\boldsymbol{x} - \boldsymbol{Y}\boldsymbol{w}||_2^2 \ \text{s.t.}\ ||\boldsymbol{d}\lfloor \boldsymbol{w} \rfloor_+||_2^2 \le \varepsilon, ||\boldsymbol{w}||_0 \le t, \mathbf{1}^{\mathrm{T}}\boldsymbol{w} = 1 \tag{7}$$

where $\boldsymbol{w} = [w_1, w_2, \dots, w_M]^{\mathrm{T}}$ and $\boldsymbol{d} = [\, D(f(\boldsymbol{x}), f(\boldsymbol{y}_1)), D(f(\boldsymbol{x}), f(\boldsymbol{y}_2)), \dots, D(f(\boldsymbol{x}), f(\boldsymbol{y}_M))]$. $\lfloor \boldsymbol{w} \rfloor_+$ means converting all non-zeros in $\boldsymbol{w}$ to 1, and t is the maximum number of non-zero elements in $\boldsymbol{w}$. It is noted that the optimization problem in Eqn. (7) is NP-hard according to the properties of L_0-norm. Researches in [15] suggests the desired sparse $\boldsymbol{w}^*$ can be solved efficiently by using L_1-norm as follows:

$$\min ||\boldsymbol{x} - \boldsymbol{Y}\boldsymbol{w}||_2^2 \ \text{s.t.}\ ||\boldsymbol{d}\lfloor \boldsymbol{w} \rfloor_+||_2^2 \le \varepsilon, ||\boldsymbol{w}||_1 \le t, \mathbf{1}^{\mathrm{T}}\boldsymbol{w} = 1 \tag{8}$$

Its equivalent formulation in Lagrange multipliers form is:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} ||\boldsymbol{x} - \boldsymbol{Y}\boldsymbol{w}||_2^2 + \lambda_1 ||\boldsymbol{d}\lfloor \boldsymbol{w} \rfloor_+||_2^2 + \lambda_2 ||\boldsymbol{w}||_1 \ \text{s.t.}\ \mathbf{1}^{\mathrm{T}}\boldsymbol{w} = 1 \tag{9}$$

where λ_1 and λ_2 are the regularization coefficients for trading off the locality and sparsity. Second term enforces choosing nearby vectors while third term enforce the sparsity in selecting potential candidates. The sparsity constraint evade the decision on the neighborhood size by using neighbors as fewer as possible, and it favors the neighbors with similar structure. With the joint constraint of locality and sparsity exerted by second and third term, respectively, the selected neighboring candidates prefer to share the same or similar geometry with the input patch $\boldsymbol{x}$.

2.4. Approximated SE-VOC for computational efficiency

The SE-VOC tries to select few local training patches to reconstruct input patch, but Eqn. (9) does this work in the whole example space (spanned by elements in $\boldsymbol{Y}$), which is quite time-consuming. Inspired by [10, 11, 16], an efficient implementation of SE-VOC (ASE-VOC) is developed through solving locality and sparsity respectively to estimate the candidates and reconstruction weights. In order to avoid solving Eqn. (9) directly, we first choose the K nearest neighbors (Here $K \gg t$ for sparse representation of $\boldsymbol{w}$) of input patch from training examples as the local dictionary $\boldsymbol{D_Y}$ and then solve the sparse weights $\boldsymbol{w}$ as:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} ||\boldsymbol{x} - \boldsymbol{D_Y}\boldsymbol{w}||_2^2 + \lambda ||\boldsymbol{w}||_1 \ \text{s.t.}\ \mathbf{1}^{\mathrm{T}}\boldsymbol{w} = 1 \tag{10}$$

where $\boldsymbol{w} = [w_1, w_2, \dots, w_K]^{\mathrm{T}}$.

Although in this way locality and sparsity do not work simultaneously, the dictionary $\boldsymbol{D_Y}$ constructed by a relatively larger size of the neighborhood has exerted local constraint on the final sparse solution. Therefore, the result calculated by ASE-VOC is similar to that of SE-VOC but with higher computational efficiency.

The classical setting of K in KNN is $K = 5$ or more or less. Here a larger $K = 128$ or more is set to ensure the sparseness of $\boldsymbol{w}$. During the count estimation of an image, the most computational cost process in our proposed method is the searching part. For the acceleration of this part, KD-Tree structure has been applied for the lower complexity $O(K \cdot \log M)$. As ASE-VOC can individually tackle multiple image patches simultaneously, a parallel version is constructed for further acceleration.

The complete ASE-VOC algorithm is summarized as following Algorithm 1. It also has been illustrated in Figure 2.

Algorithm 1 (ASE-VOC)

Input: test image $\boldsymbol{X}$, training examples sets $\boldsymbol{Y}$ and $\boldsymbol{Y}^d$
Output: the density map $\boldsymbol{X}^d$, the estimated count $c(\boldsymbol{X})$
1: For each input patch $\boldsymbol{x}_{ij}$ extracted by $\boldsymbol{P}_{ij}$ in test image $\boldsymbol{X}$, where $\boldsymbol{P}_{ij}$ is a projection matrix that extracts the (i,j)th patch from $\boldsymbol{X}$ ● Find the candidates set $\boldsymbol{D}_{\boldsymbol{Y}} = \{\boldsymbol{y}_{t_1}, \boldsymbol{y}_{t_2}, \dots, \boldsymbol{y}_{t_K}\}, \boldsymbol{D}_{\boldsymbol{Y}} \subseteq \boldsymbol{Y}$, whose elements are the most similar K patches compared with $\boldsymbol{x}_{ij}$ based on the $D(.)$ and $f(.)$. The counterpart density maps set $\boldsymbol{D}_{\boldsymbol{Y}}^d = \{\boldsymbol{y}_{t_1}^d, \boldsymbol{y}_{t_2}^d, \dots, \boldsymbol{y}_{t_K}^d\}$ is formed from $\boldsymbol{Y}^d$ according to $\boldsymbol{D}_{\boldsymbol{Y}}$. ● Find the final selected examples and their corresponding weights $\boldsymbol{w}$ by using the orthogonal matching pursuit [17] to solve Eqn. (10). ● Compute the density map patch: $\boldsymbol{x}_{ij}^d = \boldsymbol{D}_{\boldsymbol{Y}}^d \boldsymbol{w}$. Put $\boldsymbol{x}_{ij}^d$ into the $\boldsymbol{X}^d$ based on $\boldsymbol{P}_{ij}$. 2: Get the estimated density map of $\boldsymbol{X}$: $\boldsymbol{X}^d$, and the estimated count of $\boldsymbol{X}$: $c(\boldsymbol{X}) = \sum_{p \in \boldsymbol{X}^d} \boldsymbol{X}^d(p)$.

2.5. The definition of neighborhood

In our proposed ASE-VOC method, it is noted that local geometry of a feature vector is computed in its neighborhood. Hence, neighborhood has a significant impact on the estimation result. Such neighborhood is defined by the distance metric $D(.)$ and geometric feature extractor $f(.)$ as Algorithm 1 describes. Because $f(.)$ simply uses raw data or foreground features in our experiments, we focus more on $D(.)$.

The distance metric $D(.)$ measures the similarity of two different patches and then defines the neighborhood of the input patch. As shown in Figure 4 that the data we use are sparse and imbalanced, it is difficult for metric learning [18] to learn an appropriate distance metric that generalizes well to unseen test data, so we consider three commonly used distance metrics:

Euclidean distance metric. It (also known as L_2-norm) yields a generalized similarity involving the influence of all dimensions of feature vectors.

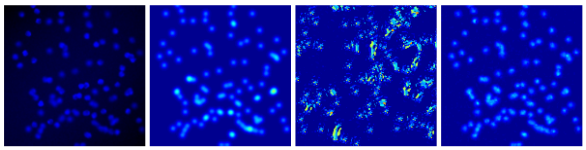
Manhattan distance metric. It (also known as L_1-norm) is more robust and less sensitive to outliers compared with Euclidean distance metric, which has been demonstrated in [19, 20].

Chebyshev distance metric. It (also known as L_∞-norm) is preferred when dealing with strict match between features because it use the maximum difference of each dimension as the dissimilarity. Thus it ensures that similar feature vectors have quite close values in each dimension.

In our paper, the proposed ASE-VOC with these three distance metrics has their own advantages under different conditions. The experiments in Section 3.4 will illustrate it.

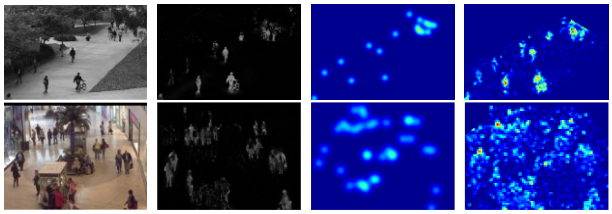
3. EXPERIMENTAL RESULTS

In order to validate the effectiveness of our proposed ASE-VOC method under the circumstances of a small amount of training images and simple features, we conduct experiments on three public benchmark datasets, including microscopy images about bacterial cells [3], UCSD [6] and Mall [1] pedestrian datasets. For ASE-VOC method, unless otherwise specified, only 16 images are chosen randomly from training set for providing examples, and related results are computed on average from 5 different draws of training set.



(a) (b) (c) (d)

Fig. 5. The cell density maps. (a) original cell image; (b) density map(ground truth); (c) Lempitsky's; (d) ours.



(a) (b) (c) (d)

Fig. 6. The pedestrian density maps. The first row is about UCSD and the second is about Mall dataset. (a) original surveillance image; (b) difference image(foreground features); (c) density map(ground truth); (d) ours.

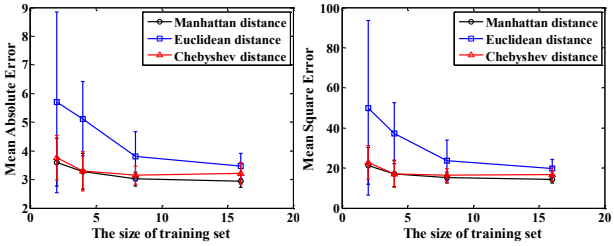


Fig. 7. Mean absolute error (left) and mean square error (right) computed by 5 different draws of training using different distance metrics and various size of training set.

3.1. Bacterial cell dataset

This dataset comprises 200 vivid synthetic bacterial cell images. Every image contains 171 ± 64 cells. The image size is 256×256 with RGB three color channels, but only blue channel has useful information. For evaluation of state-of-art methods and our proposed one on this dataset, the first 100 images are used for training and the remaining ones are for validation. For 5 different random subsets consisting of N ($N = 1,2,4,\dots,32$) images from training set, mean absolute errors and their standard deviations would be calculated. As we adhered to the experimental protocols in [3], the performance of presented methods [3] can be directly comparable.

For our proposed ASE-VOC method, the patch size is set to 4×4; either for training or validation, patch step is set to 2. The number of nearest neighbors K is set to 128 and the maximum sparsity t is set to 5. The used features are just raw data extracted from blue channel of images.

The results are shown in Table 1. Compared with the classical detection based or GR-VOC methods, our method gives better results with no matter what the size of training samples is, and converges fast with the increasing number of training images; Compared with the Lempitsky's DE-VOC method, our method offers more accurate estimation even with 1 or 2 training samples, and provides competitive result when training set grows. From Figure 5, it

Table 1. Mean absolute errors (MAE) for cell counting

Method	Feature	validation	N=1	N=2	N=4	N=8	N=16	N=32
RR [1]	(1)	counting	67.3±25.2	37.7±14.0	16.7±3.1	8.8±1.5	6.4±0.7	5.9±0.5
KRR [4]	(1)	counting	60.4±16.5	38.7±17.0	18.6±5.0	10.4±2.5	6.0±0.8	5.2±0.3
detection [3]	(2)	counting	28.0±20.6	20.8±5.8	13.6±1.5	10.2±1.9	10.4±1.2	8.5±0.5
detection [3]	(2)	detection	20.8±3.8	20.1±5.3	15.7±2.0	15.0±4.1	11.8±3.1	12.0±0.8
density learning [3]	(1)	MESA	9.5±6.1	6.3±1.2	4.9±0.6	4.9±0.7	**3.8±0.2**	**3.5±0.2**
E-VOC (L_∞-norm)	(3)	Counting	20.5±11.8	5.5±1.1	**4.4±0.6**	5.2±0.6	5.0±0.2	4.8±0.5
ASE-VOC (L_1-norm)	(3)	Counting	7.6±2.1	6.6±2.6	5.6±1.5	**4.4±0.2**	4.5±0.6	3.6±0.1
ASE-VOC (L_2-norm)	(3)	Counting	**7.0±3.3**	**4.9±1.3**	4.8±1.5	4.5±0.6	4.3±0.3	4.2±0.4
ASE-VOC (L_∞-norm)	(3)	Counting	8.1±3.6	5.9±0.9	4.9±1.1	4.8±0.7	3.9±0.3	3.6±0.1

(1) Dense SIFT+Bag of words; (2) Dense SIFT; (3) Raw data (extracted from blue channel)

is obvious that the density map generated by our method is more similar to the ground truth compared with Lempitsky's. It is noted our method just used simple raw data as the features. The mean absolute errors (MAE) produced by E-VOC (K is set to 5) method suddenly drops with $N = 4$ and then rises with $N = 8$ in Table 1, which proves the unstability caused by under-sampled data and a fixed number of nearest neighbors. In comparison, our ASE-VOC (with L_∞-norm) offers a more stable and accurate estimation.

3.2. UCSD dataset

This dataset consists of 2000 sequential frames extracted from UCSD video captured by hand-held camera. The video frame is of size 238×158, and it contains 29 ± 9 pedestrians on average.

The experimental setting follows that in [6] and frames from 601-1400 are used as training data and the rest is for testing. The features used by comparative algorithms are referred in [8].

The parameter setting of our ASE-VOC method is the same as that used in Section 3.1. Considering the computational cost, we scale down the image size to 116×76. The results in Table 3 prove the competitiveness of our method (ASE-VOC with L_∞-norm) compared with mainstream GR-VOC methods even with fewer training samples and simple foreground features.

We also compare ASE-VOC with GR-VOC and DE-VOC methods following the experimental setting in [3]. It splits into 4 different training and testing sets: 1) 'maximal': training on frames 600:5:1400; 2) 'downscale': training on frames 1205:5:1600; 3) 'upscale': training on frames 805:5:1100; 4) 'minimal': training on frames 640:80:1360. The frames do not show in training procedure would be tested. The related results are displayed in Table 2. It is clear from Table 2 that our method is comparable to state-of-arts with L_∞-norm. Especially in 'minimal' training set, our method outperforms other algorithms with only 10 training images and simple foreground features.

The density maps produced by our method in Figure 6 also demonstrate the intrinsic similarity between the original image and its density map in spatial space (Figure 6(d)).

3.3. Mall dataset

Mall dataset is recorded by surveillance camera in a shopping mall, where background is quite complicated with varied illumination conditions [1]. The mall video has 2000 frames with size of 640×480 and has 33 ± 20 pedestrians per frame on average.

We adhered to the protocols used in [1]: the first 800 frames are for training and the rest ones for testing. The features used by comparative algorithms are referred in [8]. The parameter settings are also the same as those used in Section 3.1 except the image size is down scaled to 160×120. The comparative results are reported in Table 3. It is noted that with fewer samples and simple foreground features, our method outperforms mainstream GR-VOC methods both on mean absolute error (MAE) and mean square error (MSE) with L_1-norm.

Table 2. Mean absolute errors (MAE) on UCSD dataset (Fea: Features)

Method	Fea	max	down	up	min
Regression [2]	(1)	2.07	2.66	2.78	N/A
Regression [5]	(1)	1.8	2.34	2.52	4.46
Density+MESA [3]	(1)	**1.7**	**1.28**	**1.59**	2.02
Density+RF [7]	(1)	**1.7**	2.16	1.61	2.2
ASE-VOC (L_∞-norm)	(2)	2.01	2.32	2.48	**1.82**

(1) Fused features (difference image + raw data + gradient image) + feature selection;
(2) Foreground features.

Table 3. Crowd counting performance comparison (TrN: the number of training images; Fea: Features)

Method	TrN	Fea	UCSD		Mall	
			MAE	MSE	MAE	MSE
RR [1]	800	(1)	2.25	7.82	3.59	19.0
KRR [4]	800	(1)	2.16	7.45	3.51	18.1
GPR [6]	800	(1)	2.24	7.97	3.72	20.1
CA-RR [8]	800	(2)	**2.07**	**6.86**	3.43	17.7
ASE (L_1-norm)	16	(3)	3.46	16.70	**2.94**	**14.3**
ASE (L_∞-norm)	16	(3)	2.35	8.4	3.22	16.8

(1) Fused features (segment features + internal edge features + texture features);
(2) Cumulative attributes;
(3) Foreground features.

3.4. The impact of training size and distance metric

In this part, the impact of the size of training set and similarity measurement $D(.)$ is evaluated. Part of results have be shown in Table 1 on the cell dataset. It is clear that with the increase of training samples, the MAE descends steadily with different $D(.)$ for ASE-VOC.

Among the tested distance metrics, using Euclidean distance gets lowest MAE when N=1 or N=2, but its potential prediction ability is limited compared with Manhattan and Chebyshev distance when the amount of training images is increasing. In our analysis, the image structure in cell dataset is simple, and Euclidean distance offers more generalized similar candidates, hence it performs better than others when N is extremely small. In general, Chebyshev distance (L_∞-norm) performs best for cell data considering both accur-

acy and stability.

For pedestrian data in Mall, related results are displayed in Figure 7. Experimental setting is the same as that in Section 3.3. It is noted that Manhattan distance (L_1-norm) achieves best result overall. The estimation error gradually decreases with the incline of the training set size. The result produced by Chebyshev distance is close to Manhattan distance. But it may be unstable when the training set size is larger than 4, the estimation error grows as the training set size does. In our analysis, it could be caused by the fact that the environment in Mall dataset is more complicated than cell or UCSD, strict matching offered by Chebyshev distance may cause overfitting, and its probability would boost when training samples grow rapidly. To alleviate it, a larger neighborhood size may be helpful.

4. CONCLUSION AND FUTURE WORK

We have developed an efficient sparsity-constrained example-based method to estimate the object counts in an image. It uses similar local geometry shared between two manifolds of image patches and their counterpart density maps to reconstruct the density map. As ASE-VOC uses the generalization over training samples, fewer training data are demanded compared with mainstream methods. Intensive experiments prove the effectiveness of our method even with simple features on a few training images, and its robustness against the resolution of test images.

In future work, we will study on the effects of employing different image features and similarity measurements on the performance of ASE-VOC method. To further reduce the computational complexity, more efficient search algorithms such as approximated-nearest-neighbors will be investigated.

5. REFERENCES

[1] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature Mining for Localised Crowd Counting," in *BMVC*, 2012, p. 3.

[2] D. Kong, D. Gray, and H. Tao, "A viewpoint invariant approach for crowd counting," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, 2006, pp. 1187-1190.

[3] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in Neural Information Processing Systems*, 2010, pp. 1324-1332.

[4] S. An, W. Liu, and S. Venkatesh, "Face recognition using kernel ridge regression," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 2007, pp. 1-7.

[5] D. Ryan, S. Denman, C. Fookes, and S. Sridharan, "Crowd counting using multiple local features," in *Digital Image Computing: Techniques and Applications, 2009. DICTA'09.*, 2009, pp. 81-88.

[6] 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*, 2008, pp. 1-7.

[7] L. Fiaschi, R. Nair, U. Koethe, and F. Hamprecht, "Learning to count with regression forest and structured labels," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, 2012, pp. 2685-2688.

[8] K. Chen, S. Gong, T. Xiang, and C. C. Loy, "Cumulative attribute space for age and crowd density estimation," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013, pp. 2467-2474.

[9] Y. Zhou and J. Luo, "A practical method for counting arbitrary target objects in arbitrary scenes," in *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, 2013, pp. 1-6.

[10] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science,* vol. 290, pp. 2323-2326, 2000.

[11] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004, pp. I-I.

[12] S. Huang, C. Cai, and Y. Zhang, "Dimensionality reduction by using sparse reconstruction embedding," in *Advances in Multimedia Information Processing-PCM 2010*, ed: Springer, 2010, pp. 167-178.

[13] E. Elhamifar and R. Vidal, "Sparse manifold clustering and embedding," in *Advances in neural information processing systems*, 2011, pp. 55-63.

[14] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Sparse embedding: A framework for sparsity promoting dimensionality reduction," in *Computer Vision–ECCV 2012*, ed: Springer, 2012, pp. 414-427.

[15] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal " *Communications on pure and applied mathematics,* vol. 59, pp. 797-829, 2006.

[16] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2010, pp. 3360-3367.

[17] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit," *CS Technion,* vol. 40, pp. 1-15, 2008.

[18] E. P. Xing, M. I. Jordan, S. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with side-information," in *Advances in neural information processing systems*, 2002, pp. 505-512.

[19] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 31, pp. 210-227, 2009.

[20] Y. Li, C. Cai, G. Qiu, and K.-M. Lam, "Face hallucination based on sparse local-pixel structure," *Pattern Recognition,* vol. 47, pp. 1261-1270, 2014.