Crowd Counting and Profiling: Methodology and Evaluation

Chen Change Loy, Ke Chen, Shaogang Gong, and Tao Xiang

Abstract Video imagery based crowd analysis for population profiling and density estimation in public spaces can be a highly effective tool for establishing global situational awareness. Different strategies such as counting by detection and counting by clustering have been proposed, and more recently counting by regression has also gained considerable interest due to its feasibility in handling relatively more crowded environments. However, the scenarios studied by existing regression-based techniques are rather diverse in terms of both evaluation data and experimental settings. It can be difficult to compare them in order to draw general conclusions on their effectiveness. In addition, contributions of individual components in the processing pipeline such as feature extraction and perspective normalisation remain unclear and less well studied. This study describes and compares the state-of-the-art methods for video imagery based crowd counting, and provides a systematic evaluation of different methods using the same protocol. Moreover, we evaluate critically each processing component to identify potential bottlenecks encountered by existing techniques. Extensive evaluation is conducted on three public scene datasets, including a new shopping centre environment with labelled ground truth for validation. Our study reveals new insights into solving the problem of crowd analysis for population profiling and density estimation, and considers open questions for future studies.

Chen Change Loy
Vision Semantics Limited, London E1 4NS, UK
e-mail: ccloy@visionsemantics.com

Ke Chen
Queen Mary University of London, London E1 4NS, UK
e-mail: cory@eecs.qmul.ac.uk

Shaogang Gong
Queen Mary University of London, London E1 4NS, UK
e-mail: sgg@eecs.qmul.ac.uk

Tao Xiang
Queen Mary University of London, London E1 4NS, UK
e-mail: txiang@eecs.qmul.ac.uk

1 Introduction

The analysis of crowd dynamics and behaviours is a topic of great interest in sociology, psychology, safety, and computer vision. In the context of computer vision, many interesting analyses can be achieved [91], e.g. to learn the crowd flow evolvement and floor fields [3], to track an individual in a crowd [65], to segment a crowd into semantic regions [51, 93], to detect salient regions in a crowd [53], or to recognise anomalous crowd patterns [41, 60]. A fundamental task in crowd analysis that enjoys wide spectrum of applications is to automatically count the number of people in crowd and profile their behaviours over time in a given region.

One of the key application areas of crowd counting is public safety and security. Tragedies involving large crowds often occur, especially during religious, political, and musical events [35]. For instance, a crowd crush at the 2010 Love Parade music festival in Germany, caused a death of 21 people and many more injured (see Fig. 1). And more recently a stampede happened near the Sabarimala Temple, India with death toll crosses hundred. These tragedies could be avoided, if a safer site design took place and a more effective crowd control was enforced. Video imagery based crowd counting can be a highly beneficial tool for early detection of over-crowded situations to facilitate more effective crowd control. It also helps in profiling the population movement over time and across spaces for establishing global situational awareness, developing long-term crowd management strategies, and designing evacuation routes of public spaces.

In retail sectors, crowd counting can be an intelligence gathering tool [76] to provide valuable indication about the interest of customers through quantifying the number of individuals browsing a product, the queue lengths, or the percentage of store's visitors at different times of the day. The information gathered can then be used to optimise the staffing need, floor plan, and product display.

Video imagery based crowd counting for population profiling remains a non-trivial problem in crowded scenes. Specifically, frequent occlusion between pedestrians and background clutter render a direct implementation of standard object segmentation and tracking infeasible. The problem is further compounded by visual ambiguities caused by varying individual appearances and body articulations, and group dynamics. External factors such as camera viewing angle, illumination changes, and distance from the region of interest also pose great challenges to the counting problem.

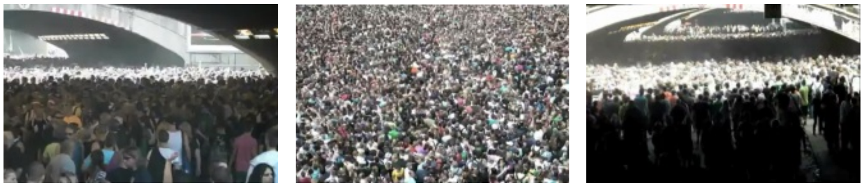


Fig. 1 Example of surveillance footage frames captured during the Love Parade music festival in Germany, 2010, before the fatalities occurred. Images from www.dokumentation-loveparade.com/.

Various approaches for crowd counting have been proposed. A popular method is *counting by detection* [24], which detects instances of pedestrian through scanning the image space using a detector trained with local image features. An alternative approach is *counting by clustering* [7, 63], which assumes a crowd to be composed of individual entities, each of which has unique yet coherent motion patterns that can be clustered to approximate the number of people. Another method is inspired by the capability of human beings, in determining density at a glance without numerating the number of pedestrians in it. This approach is known as *counting by regression* [12, 22], which counts people in crowd by learning a direct mapping from low-level imagery features to crowd density.

In this study, we provide a comprehensive review, comparative evaluation, and critical analysis on computer vision techniques for crowd counting, also known as crowd density estimation, and discuss crowding counting as a tool for population profiling. We first present a structured critical overview of different approaches to crowd counting reported in the literature, including pedestrian detection, coherent motion clustering, and regression-based learning. In particular, we focus on the regression-based techniques that have gain considerable interest lately due to their effectiveness in handling more crowded scenes. We then provide analysis of different regression-based approaches to crowd counting by systematic comparative evaluation, which gives new insights into contributions of key constituent components and potential bottlenecks in algorithm design. To facilitate our experiments, we also introduce a new shopping mall dataset of over 60,000 pedestrians labelled in 2000 video frames, i.e. the largest dataset to date in terms of the number of pedestrian instances captured in realistic crowded public space scenario for crowd counting and profiling research.

2 Survey of the State of the Art

The taxonomy of crowd counting algorithms can be generally grouped into three paradigms, namely counting by detection, clustering, and regression. In this section, we provide an overview on each of the paradigms, with a particular focus on the counting by regression strategy that has shown to be effective on more crowded environments.

2.1 Counting by Detection

The following is a concise account of pedestrian detection with emphasise on counting application. A more detailed treatment on this topic can be found in [24].

Monolithic detection: The most intuitive and direct approach to numerate the number of people in a scene is through detection. A typical pedestrian detection ap-

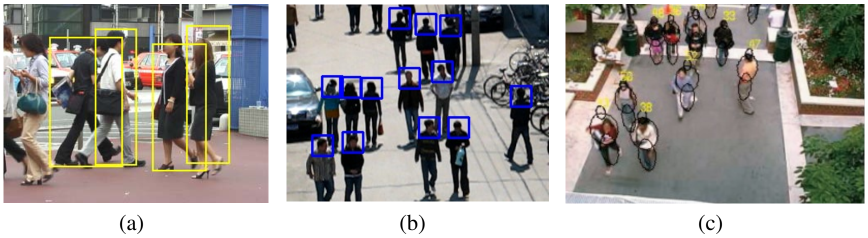


Fig. 2 Pedestrian detection results obtained using (a) monolithic detection, (b) part-based detection, and (c) shape matching. Images from [43, 47, 92].

proach is based on monolithic detection [21, 43, 78], which trains a classifier using the full-body appearance of a set of pedestrian training images (see Fig. 2(a)). Common features to represent the full-body appearance include Haar wavelets [80], gradient-based features such as histogram of oriented gradient (HOG) feature [21], edgelet [85], and shapelets [68]. The choice of classifier imposes significant impact on the speed and quality of detection, often requiring a trade-off between these two. Non-linear classifiers such as RBF Support Vector Machines (SVMs) offer good quality but suffer from low detection speed. Consequently, linear classifiers such as boosting [81], linear SVMs, or Random/Hough Forests [28] are more commonly used. A trained classifier is then applied in a sliding window fashion across the whole image space to detect pedestrian candidates. Less confident candidates are normally discarded using non-maximum suppression, which leads to final detections that suggest the total number of people in a given scene. Whole body monolithic detector can generates reasonable detections in sparse scenes. However, it suffers in crowded scenes where occlusion and scene clutter are inevitable [24].

Part-based detection: A plausible way to get around the partial occlusion problem to some extent is by adopting a part-based detection method [26, 48, 86]. For instance, one can construct boosted classifiers for specific body parts such as the head and shoulder to estimate the people counts in a monitored area [47] (see Fig. 2(b)). It is found that head region alone is not sufficient for reliable detection due to its shape and appearance variations. Including the shoulder region to form an omega-like shape pattern tends to give better performance in real-world scenarios [47]. The detection performance can be further improved by tracking validation, i.e. associating detections over time and rejecting spurious detections that exhibit coherent motion with the head candidates [62]. In comparison to monolithic detection, part-based detection relaxes the stringent assumption about the visibility of the whole body, it is thus more robust in crowded scenes.

Shape matching: Zhao et al. [92] define a set of parameterised body shapes composed of ellipses, and employ a stochastic process to estimate the number and shape configuration that best explains a given foreground mask in a scene. Ge and Collins [29] extend the idea by allowing more flexible and realistic shape prototypes than just simple geometric shapes proposed in [92]. In particular, they learn a mixture model of Bernoulli shapes from a set of training images, which is then employed to search for maximum a posteriori shape configuration of foreground objects, revealing not only the count and location, but also the pose of each person in a scene.

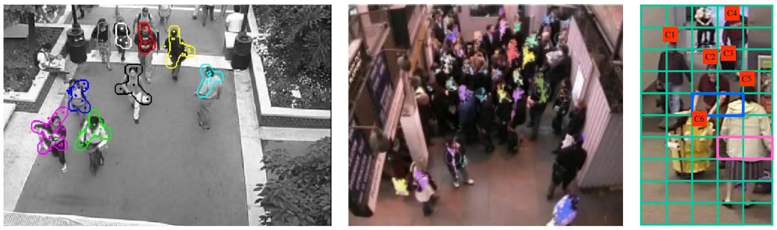
Multi-sensor detection: If multiple cameras are available, one can further incorporate multi-view information to resolve visual ambiguities caused by inter-object occlusion. For example, Yang et al. [88] extracted the foreground human silhouettes from a network of cameras to establish bounds on the number and possible locations of people. In the same vein, Ge and Collins [30] estimate the number of people and their spatial locations by leveraging multi-view geometric constraints. The aforementioned methods [30, 88] are restricted since a multi-camera setup with overlapping views is not always available in many cases. Apart from detection accuracy improvement, the speed of detection can benefit from the use of multi-sensors, e.g. the exploitation of geometric context extracted from stereo images [5].

Transfer learning: Applying a generic pedestrian detector to a new scene cannot guarantee satisfactory cross-dataset generalisation [24], whilst training a scene-specific detector for counting is often laborious. Recent studies have been exploring the transfer of generic pedestrian detectors to a new scene without human supervision. The key challenges include the variations of viewpoints, resolutions, illuminations, and backgrounds in the new environment. A solution to the problem is proposed in [82, 83] to exploit multiple cues such as scene structures, spatio-temporal occurrences, and object sizes to select confident positive and negative examples from the target scene to adapt a generic detector iteratively.

2.2 Counting by Clustering

The counting by clustering approach relies on the assumption that individual motion field or visual features are relatively uniform, hence coherent feature trajectories can be grouped together to represent independently moving entities. Studies that follow this paradigm include [63], which uses a Kanade-Lucas-Tomasi (KLT) tracker to obtain a rich set of low-level tracked features, and clusters the trajectory to infer the number of people in the scene (see Fig. 3(a)); and [7], which tracks local features and groups them into clusters using Bayesian clustering (see Fig. 3(b)). Another closely related method is [77], which incorporates the idea of feature constancy into a counting by detection framework. The method first generates a set of person hypotheses of a crowd based on head detections. The hypotheses are then refined iteratively by assigning small patches of the crowd to the hypotheses based on the constancy of motion fields and intra-garment colour (see Fig. 3(c)).

The aforementioned methods [7, 63] avoid supervised learning or explicit modelling of appearance features as in the counting by detection paradigm. Nevertheless, the paradigm assumes motion coherency, hence false estimation may arise when people remaining static in a scene, exhibiting sustained articulations, or two objects



(a) (b) (c)

Fig. 3 (a) and (b) show the results of clustering coherent motions using methods proposed in [63] and [7] respectively. (c) shows the pairwise affinity of patches (strong affinity = magenta, weak affinity = blue) in terms of motion and colour constancy; the affinity is used to determine the assignment of patches to person hypotheses [77]. Images from [7, 63, 77].

sharing common feature trajectories over time. Note that counting by clustering only works with continuous image frames, not static images whilst the counting by detection and regression do not have this restriction.

2.3 *Counting by Regression*

Despite the substantial progress being made in object detection [24] and tracking [90] in recent years, performing either in isolation or both reliably in a crowded environment remains a non-trivial problem. Counting by regression deliberately avoids actual segregation of individual or tracking of features but estimate the crowd density based on holistic and collective description of crowd patterns. Since neither explicit segmentation nor tracking of individual are involved, counting by regression becomes a feasible method for crowded environments where detection and tracking are severely limited intrinsically.

One of the earliest attempts in exploring the use of regression method for crowd density estimation is by Davies et al. [22]. They first extract low-level features such as foreground pixels and edge features from each video frame. Holistic properties such as foreground area and total edge count are then derived from the raw features. Consequently, a linear regression model is used to establish a direct mapping between the holistic patterns and the actual people counts. Specifically, a function is used to model how the input variable (i.e. the crowd density) changes when the target variables (i.e. holistic patterns) are varied. Given an unseen video frame, conditional expectation of the crowd density can then be predicted given the extracted features from that particular frame. Since the work of Davies et al. [22], various methods have been proposed following the same idea with improved feature sets or more sophisticated regression models, but still sharing a similar processing pipeline as in [22] (see Fig. 4). A summary of some of the notable methods is given in Table 1. In the following subsections, we are going to have detailed discussion on the main components that constitute the counting by regression pipeline, namely feature representation, geometric correction, and regression modelling.

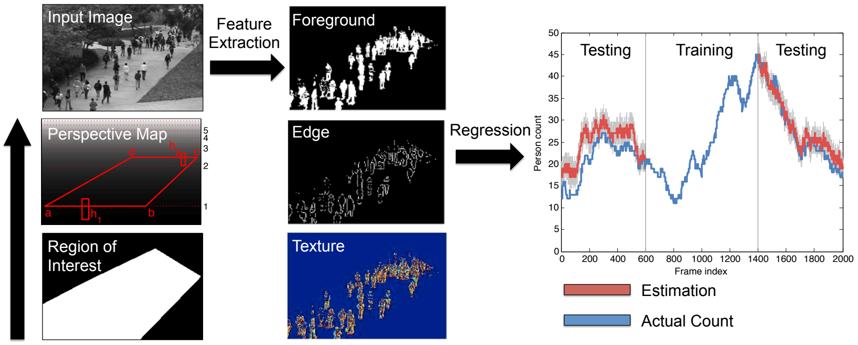


Fig. 4 A typical pipeline of counting by regression: first defining the region of interest and finding the perspective normalisation map of a scene, then extracting holistic features and training a regressor using the perspective normalised features.

2.3.1 Feature Representation

The question of crowd representation or abstraction must be addressed before a regression function can be established. Feature representation concerns the extraction, selection, and transformation of low-level visual properties in an image or video to construct intermediate input to a regression model. A popular approach is to combine several features with complementary nature to form a large bank of features [13].

Foreground segment features: The most common or arguably the most descriptive representation for crowd density estimation is foreground segment, which can be obtained through background subtraction, such as mixture of Gaussians-based technique [73] or mixture of dynamic textures-based method [10]. Various holistic features can be derived from the extracted foreground segment, for example:

- Area – total number of pixels in the segment.
- Perimeter – total number of pixels on the segment perimeter.
- Perimeter-area ratio – ratio between the segment perimeter and area, which measures the complexity of the segment shape.
- Perimeter edge orientation – orientation histogram of the segment perimeter.
- Blob count – the number of connected components with area larger than a predefined threshold, e.g. 20 pixels in size.

Various studies [13, 22, 54] have demonstrated encouraging results using the segment-based features despite its simplicity. Several considerations, however, has to be taken into account during the implementation. Firstly, to reduce spurious foreground segments from other regions, one can confine the analysis within a region of interest (ROI), which can be determined manually or following a foreground accumulation approach [54]. Secondly, different scenarios may demand different background extraction strategies. Specifically, dynamic background subtraction [73] can cope with gradual illumination change but have difficulty in isolating people that

	Year	Features								Learning		Datasets
		segment	edge	texture	shape	intensity	gradients	motion	others	regression method	level	
Davies et al. [22]	1995	✓	✓	–	–	–	–	–	–	Linear regression	global	–
Marana et al. [59]	1997	–	–	✓	–	–	–	–	–	Self-organising map neural network	global	–
Cho et al. [16]	1997	✓	✓	–	–	–	–	–	–	Feedforward neural network	global	–
Kong et al. [38, 39]	2005 2006	✓	✓	–	–	–	–	–	–	Feedforward neural network	global	–
Dong et al. [25]	2007	–	–	–	✓	–	–	–	–	Shape matching + locally-weighted regression	segment	USC Campus Plaza
Chan et al. [12–14]	2008 2009	✓	✓	✓	–	–	–	–	–	Gaussian processes	global	UCSD Pedestrian, PETS 2009
Chan et al. [11]	2009	✓	✓	✓	–	–	–	–	–	Bayesian Poisson regression	global	UCSD Pedestrian
Ryan et al. [67]	2009	✓	✓	–	–	–	–	–	–	Feedforward neural network	segment	UCSD Pedestrian
Cong et al. [18]	2009	✓	✓	–	–	–	–	–	–	Polynomial regression	segment	–
Lempitsky et al. [44]	2010	✓	–	–	–	✓	✓	–	–	Density function minimisation based on Maximum Excess over Subarrays distance	pixel	UCSD Pedestrian
Conte et al. [19]	2010	–	–	–	–	–	–	–	number of SURF points	Support vector regression	segment	PETS 2009
Benabbas et al. [4]	2010	✓	–	–	–	–	–	✓	–	Linear regression	segment	PETS 2009
Li et al. [46]	2011	✓	✓	–	–	–	–	–	–	Pedestrian detector + Linear regression	segment	CASIA Pedestrian [45]
Lin et al. [49]	2011	✓	✓	–	–	–	✓	–	–	Gaussian processes	segment	UCSD Pedestrian, PETS 2009
Ke et al. [15]	2012	✓	✓	✓	–	–	–	–	–	Kernel ridge regression	segment	UCSD Pedestrian, PETS 2009, Mall

Table 1 A table summarising existing counting by regression methods. Note that only publicly available datasets are listed in the datasets column.

are stagnant for a long period of time; static background subtraction [51, 66] is able to segment static objects from the background but is susceptible to lighting change. Finally, poor estimation is expected if one employs only foreground area due to inter-object occlusion, as it is possible to insert another person into the mixture and end up with the same foreground area. Enriching the representation with other descriptors may solve this problem to certain extent.

Edge features: While foreground features capture the global properties of the segment, edge features inside the segment carries complementary information about the local and internal patterns [13, 22, 38]. Intuitively, low-density crowds tend to present coarse edges, while segments with dense crowds tend to present complex edges. Edges can be detected using an edge detector such as the Canny edge detector [8]. Note that an edge image is often masked using the foreground segment to discard irrelevant edges. Some common edge-based features are listed as follows

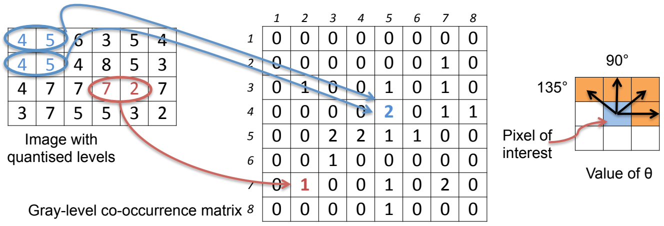
- Total edge pixels – total number of edge pixels.

- Edge orientation – histogram of the edge orientations in the segment.
- Minkowski dimension – the Minkowski fractal dimension or box-counting dimension of the edges [58], which counts how many pre-defined structuring elements are required to fill the edges.

Texture and gradient features: Crowd texture and gradient patterns carry strong cues about the number of people in a scene. In particular, high-density crowd region tends to exhibit stronger texture response [54] with distinctive local structure in comparison to low-density region; whilst local intensity gradient map could reveal local object appearance and shape such as human shoulder and head, which are informative for density estimation. Example of texture and gradient features employed for crowd counting include gray-level co-occurrence matrix (GLCM) [34], local binary pattern (LBP) [61], HOG feature [56], and gradient orientation co-occurrence matrix (GOCM) [56]. A comparative studies among the aforementioned texture and gradient features can be found in [56]. Here we provide a brief description on GLCM and LBP, which are used in our evaluation.

Gray-level co-occurrence matrix (GLCM) [34] is widely used in various crowd counting studies [13, 56, 57, 87]. For instance, Marana et al. [57] uses GLCM to distinguish five different density levels (very low, low, moderate, high, and very high), and Chan and Vasconcelos [12] employ it as holistic property for Bayesian density regression. To obtain GLCM, a typical process is to first quantise the image into 8 gray-levels and masked by the foreground segment. The joint probability or co-occurrence of neighbouring pixel values, $p(i, j \mid \theta)$ is then estimated for four orientations, $\theta \in \{0°, 45°, 90°, 135°\}$. After extracting the co-occurrence matrix, a set of features such as homogeneity, energy, and entropy can be derived for each θ

- Homogeneity – texture smoothness, $g_\theta = \sum_{i,j} \frac{p(i,j \mid \theta)}{1+|i-j|}$
- Energy – total sum-squared energy, $e_\theta = \sum_{i,j} p(i, j \mid \theta)^2$
- Entropy – texture randomness, $h_\theta = \sum_{i,j} p(i, j \mid \theta) \log p(i, j \mid \theta)$



4	5	6	3	5	4
4	5	4	8	5	3
4	7	7	7	2	7
3	7	5	5	3	2

	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	1	0
3	0	1	0	0	1	0	1	0
4	0	0	0	0	2	0	1	1
5	0	0	2	2	1	1	0	0
6	0	0	1	0	0	0	0	0
7	0	1	0	0	1	0	2	0
8	0	0	0	0	1	0	0	0

Fig. 5 Gray-level co-occurrence matrix, with $\theta = 0°$ of a 4-by-6 image. Element (7,2) in the GLCM contains the value 1 because there is only one instance in the image where two, horizontally adjacent pixels have the values 7 and 2. Element (4,5) in the GLCM contains the value 2 because there are two instances in the image where two, horizontally adjacent pixels have the values 4 and 5. The value of θ specifies the angle between the pixel of interest and its neighbour.

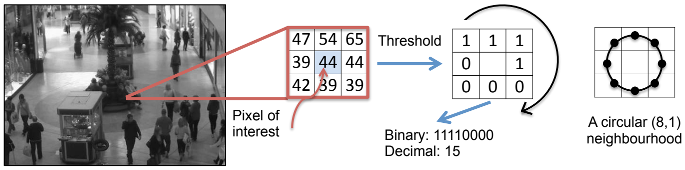


Fig. 6 A basic local binary pattern operator [61] and a circular (8,1) neighbourhood.

An alternative texture descriptor for crowd density estimation [55] is the local binary pattern (LBP) [61]. Local binary pattern has been widely adopted in various applications such as face recognition [2] and expression analysis [70], due to its high discriminative power, invariance to monotonic gray-level changes, and its computational efficiency.

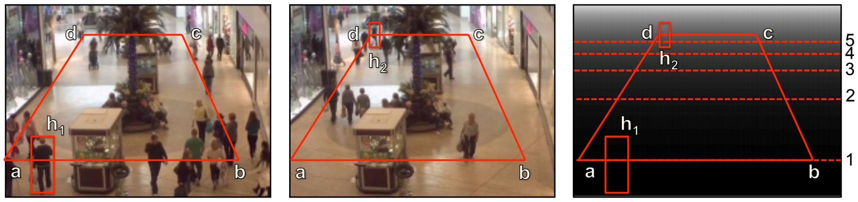
An illustration of a basic LBP operator is depicted in Fig. 6. The LBP operation is governed by a definition of local neighbourhood, i.e.the number of sampling point and radius centering the pixel of interest. An example of a circular (8,1) neighbourhood is shown in Fig. 6. Following the definition of neighbourhood, we sample 8 points at a distance of radius 1 from the pixel of interest and threshold them using the value of the centering pixel. The results are concatenated to form a binary code as the label of the pixel of interest. These steps are repeated over the whole image space and a histogram of labels is constructed as a texture descriptor.

In this study, we employed an extension of the original LBP operator known as *uniform patterns* [61], which frequently correspond to primitive micro-features such as edges and corners. A uniform LBP pattern is binary code with at most two bitwise transitions, e.g. 11110000 (1 transition) and 11100111 (2 transitions) are uniform, whilst 11001001 (4 transitions) is not. In the construction of LBP histogram, we assign a separate bin for every uniform pattern and keep all nonuniform patterns in a single bin, so we have a 58+1-dimension texture descriptor.

2.3.2 Geometric Correction

A problem commonly encountered in counting by regression framework is perspective distortion, in which far objects appear smaller than those closer to the camera view. As a consequence, features (e.g.segment area) extracted from the same object at different depths of the scene would have huge difference in values. The influence is less critical if one divides the image space into different cells, each of which modelled by a regression function; erroneous results are expected if one only uses a single regression function for the whole image space.

To address this problem geometric correction or perspective normalisation is performed to bring perceived size of objects at different depths to the same scale. Ma et al. [54] investigate the influence of perspective distortion to people counting and propose a principled way to integrate geometric correction in pixel counting, i.e. to scale each pixel by a weight, with larger weights given to further objects.



(a) (b) (c)

Fig. 7 (a) and (b) show a reference person at two extremes of a predefined quadrilateral; (c) a perspective map to scale pixels by their relative size in the three-dimensional scene.

A simple and widely adopted perspective normalisation method [44, 49, 67] is described in [13]. The method first determines four points in a scene to form a quadrilateral that corresponds to a rectangle (see Fig. 7). The lengths of the two horizontal lines of the quadrilateral, $\overline{ab}$ and $\overline{cd}$, are measured as w_1 and w_2 respectively. When a reference pedestrian passes the two extremes, i.e. its bounding box's centre touches the $\overline{ab}$ and $\overline{cd}$, its heights are recorded as h_1 and h_2. The weights at $\overline{ab}$ and $\overline{cd}$ are then assigned as 1 and $\frac{h_1 w_1}{h_2 w_2}$ respectively. To determine the remaining weights of the scene, linear interpolation is first performed on the width of the rectangle, and the height of the reference person. A weight at arbitrary image coordinate can then be calculated as $\frac{h_1 w_1}{h' w'}$, where h' and w' representing the interpolants. Here we make an assumption that the horizontal vanishing line to be parallel to the image horizontal scan lines.

When applying the weights to features, it is assumed that the size of foreground segment changes quadratically, whilst the total edge pixels changes linearly with respect to the perspective. Consequently, each foreground segment pixel is weighted using the original weight and the edge features are weighted by square-roots of the weights. Features based on the GLCM are normalised by weighting the occurrence of each pixel pair when accumulating the co-occurrence matrix shown in Fig. 5. To obtain perspective-normalised LBP-based features, we multiply the weights to the occurrence of individual LBP labels in the image space prior to the construction of the LBP label histogram.

The aforementioned method [13] requires manual measurement which could be error-prone. There exist approaches to compute camera calibration parameters based on accumulative visual evidence in a scene. For example, a method is proposed in [40] to find the camera parameters by exploiting foot and head location measurements of people trajectories over time. Another more recent method [50] relaxes the requirement of accurate detection and tracking. This method takes noisy foreground segments as input to obtain the calibration data by leveraging the prior knowledge of the height distribution. With a calibrated 3D model, one can also obtain the perspective map as in [14], which moves a virtual person within the 3D world and measures the number of pixels projected onto the 2D image space.

2.3.3 Regression Models

After feature extraction and perspective normalisation, a regression model is trained to predict the count given the normalised features. A regression model may have a broad class of functional forms. In this section we discuss a few popular regression models for crowd density estimation.

Linear regression: Given a training data comprising N observations $\{\mathbf{x}_n\}$, where $n = 1, \ldots, N$ together with corresponding continuous target values $\{y_n\}$, the goal of regression is to predict the value of y given a new value of $\mathbf{x}$ [6]. The simplest approach is to form of linear regression function $f(\mathbf{x}, \mathbf{w})$ that involves a linear combination of the input variables, i.e.

$$f(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \cdots + w_D x_D, \tag{1}$$

where D is the dimension of features, $\mathbf{x} = (x_1, \ldots, x_D)^{\mathsf{T}}$, and $\mathbf{w} = (w_0, \ldots, w_D)^{\mathsf{T}}$ are the parameters of the model. This model is often known as *linear regression* (LR), which is a linear function of the parameters $\mathbf{w}$. In addition it is also linear with respect to the input variables $\mathbf{x}$.

In a sparse scene where smaller crowd size and fewer inter-object occlusions are observed, the aforementioned linear regressor [4, 22, 46] may suffice since the mapping between the observations and people count typically presents a linear relationship. Nevertheless, given a more crowded environment with severe inter-object occlusion, one may have to employ a nonlinear regressor to adequately capture the nonlinear trend in the feature space [9].

To relax the linearity assumption, one can take a linear combination of a fixed set of nonlinear functions of the input variables, also known as basis functions $\phi(\mathbf{x})$, to obtain a more expressive class of function. It has the form of

$$f(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{x}), \tag{2}$$

where M is the total number of parameters in this model, $\mathbf{w} = (w_0, \ldots, w_{M-1})^{\mathsf{T}}$, and $\boldsymbol{\phi} = (\phi_0, \ldots, \phi_{M-1})^{\mathsf{T}}$. The functional form in (2) is still known as linear model since it is linear in $\mathbf{w}$, despite the function $f(\mathbf{x}, \mathbf{w})$ is nonlinear with respect to input vector $\mathbf{x}$. A polynomial regression function considered in [18] (see Table 1) is a specific example of this model, with the basis functions taking a form of powers of $\mathbf{x}$, that is $\phi_j(\mathbf{x}) = \mathbf{x}^j$. Gaussian basis function and sigmoidal basis function are other possible choices of basis functions.

Parameters in the aforementioned linear model is typically obtained by minimising the sum of squared errors

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \left\{ y_n - \mathbf{w}^{\mathsf{T}} \boldsymbol{\phi}(\mathbf{x}_n) \right\}^2. \tag{3}$$

One of the key limitation of linear model is that the model can get unnecessarily complex give high-dimensional observed data $\mathbf{x}$. Particularly in counting by regression, it is a common practice to exploit high-dimensional features [13]. Some of the elements are not useful for predicting the count. In addition, some of them may be highly co-linear, unstable estimate of parameters may occurs [6], leading to very large magnitude in the parameters and therefore a clear danger of severe over-fitting.

Partial least squares regression: A way of addressing the multicollinearity problem is by *partial least squares regression* (PLSR) [31], which projects both input $\mathbf{X} = \{\mathbf{x}_n\}$ and target variables $\mathbf{Y} = \{y_n\}$ to a latent space, with a constraint such that the lower-dimensional latent variables explain as much as possible the covariance between $\mathbf{X}$ and $\mathbf{Y}$. Formally, the PLSR decomposes the input and target variables as

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathsf{T}} + \varepsilon_x \tag{4}$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^{\mathsf{T}} + \varepsilon_y, \tag{5}$$

where $\mathbf{T}$ and $\mathbf{U}$ are known as score matrices, with the column of $\mathbf{T}$ being the latent variables; $\mathbf{P}$ and $\mathbf{Q}$ are known as loading matrices [1]; and ε are the error terms. The decomposition are made so to maximise the covariance of $\mathbf{T}$ and $\mathbf{U}$. There are two typical ways in estimating the score matrices and loading matrices, namely NIPALS and SIMPLS algorithms [1, 89].

Kernel ridge regression: Another method of mitigating the multicollinearity problem is through adding a regularisation term to the error function in Equation (3). A simple regularisation term is given by the sum-of-squares of the parameter vector elements, $\frac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w}$. The error function becomes

$$E_R(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left\{y_n - \mathbf{w}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right\}^2 + \frac{\lambda}{2}\mathbf{w}^{\mathsf{T}}\mathbf{w}, \tag{6}$$

with λ to control the trade-off between the penalty and the fit. A common way of determining λ is via cross-validation. Using this particular choice of regularisation term with $\phi(\mathbf{x}_n) = \mathbf{x}_n$, we will have error function of *ridge regression* [36].

A non-linear version of the ridge regression, known as *kernel ridge regression* (KRR) [69], can be achieved via kernel trick [71], whereby a linear ridge regression model is constructed in higher dimensional feature space induced by a kernel function defining the inner product

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^{\mathsf{T}}\phi(\mathbf{x}'). \tag{7}$$

For the kernel function, one has typical choices of linear, polynomial, and radial basis function (RBF) kernels. The regression function of KRR is given by

$$f(\mathbf{x}, \boldsymbol{\alpha}) = \sum_{n=1}^{N}\alpha_n k(\mathbf{x}, \mathbf{x}_n), \tag{8}$$

where $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_n\}^\mathsf{T}$ are Lagrange multipliers. This solution is not sparse in the variables α, that is $\alpha_n \neq 0$, $\forall n \in \{1, \ldots N\}$.

Support vector regression: *Support vector regression* (SVR) [42,72] has been used for crowd counting in [87]. In contrast to KRR, the SVR achieves sparseness in α (see Equation (8)) by using the concept of support vectors to determine the solution, which can result in faster testing speed than KRR that sums over the entire training-set [84]. Specifically, the regression function of SVR can be written as

$$f(\mathbf{x}, \boldsymbol{\alpha}) = \sum_{\text{SVs}} (\alpha_n - \alpha_n^*) k(\mathbf{x}, \mathbf{x}_n) + b, \tag{9}$$

where α_n and α_n^* represents the Lagrange multipliers, $k(\mathbf{x}, \mathbf{x}_n)$ denotes the kernel, and $b \in \mathbb{R}$. A popular error function for SVR training is ε-insensitive error function [79], which assigns zero error if the absolute difference between the prediction $f(\mathbf{x}, \boldsymbol{\alpha})$ and the target y is less than $\varepsilon > 0$. *Least-squares support vector regression* (LSSVR) [74] is least squares version of SVR. In LSSVR one finds the solution by solving a set of linear equations instead of a convex quadratic error function as in conventional SVR.

Gaussian processes regression: One of the most popular nonlinear methods for crowd counting is *Gaussian processes regression* (GPR) [64]. It has a number of pivotal properties – it allows possibly infinite number of basis functions driven by the data complexity, and it models uncertainty in regression problems elegantly[1]. Formally, we write the regression function as

$$f(\mathbf{x}) \sim \text{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \tag{10}$$

where Gaussian processes, $\text{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ is specified by its mean function $m(\mathbf{x})$ and covariance function or kernel $k(\mathbf{x}, \mathbf{x}')$

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})], \tag{11}$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))], \tag{12}$$

where $\mathbb{E}$ denotes the expectation value.

Apart from the conventional GPR, various extensions of it have been proposed. For instance, Chan et al. [9] propose a generalised Gaussian process model, which allows different parameterisation of the likelihood function, including a Poisson distribution for predicting discrete counting numbers [11]. Lin et al. [49] employ two GPR in their framework, one for learning the observation-to-count mapping, and another one for reasoning the mismatch between predicted count and actual count due to occlusion.

The key weakness of GPR is its poor tractability to large training sets. Various approximation paradigms have been developed to improve its scalability [64].

[1] One can also estimate the predictive interval in other kernel methods such as KRR [23].

It is worth pointing out that one of the attractive properties of kernel methods such as KRR, SVR, and GPR is the flexibility of encoding different assumptions about the function we wish to learn. For instance, by combining different covariance functions $k(\mathbf{x},\mathbf{x}')$, such as linear, Matérn, rational quadratic, and neural network, one has the flexibility to encode different assumptions on the continuity and smoothness of the GP function $f(\mathbf{x})$. This property is exploited in [13], in which linear and a squared-exponential (RBF) covariance functions are combined to capture both the linear trend and local non-linearities in the crowd feature space.

Random forest regression: Scalable nonlinear regression modelling can be achieved using *random forest regression* (RFR). A random forest comprises of a collection of randomly trained regression trees, which can achieve better generalisation than a single over-trained tree [20]. Each tree in a forest splits a complex nonlinear regression problem into a set of subproblems, which can be more easily handled by weak learners such as a linear model[2]. To train a forest, one optimises an energy over a given training set and associated values of target variable. Specifically, parameters $\boldsymbol{\theta}_j$ of the weak learner at each split node j are optimised via

$$\boldsymbol{\theta}_j^* = \underset{\boldsymbol{\theta}_j \in \mathscr{T}_j}{\operatorname{argmax}} I_j, \tag{13}$$

where $\mathscr{T}_j \subset \mathscr{T}$ is a subset of parameters made available to the j-th node, and I is an objective function that often takes the form of information gain. Given a new observation $\mathbf{x}$, the predictive function is computed by averaging individual posterior distributions of all the trees, i.e.

$$f(\mathbf{x}) = \frac{1}{T} \sum p_t(y|\mathbf{x}), \tag{14}$$

where T is the total number of trees in the forest, $p_t(y|\mathbf{x})$ is the posterior of t-th tree.

The hallmark of random forest is its good performance comparable to state-of-the-art kernel methods (e.g. GPR) but with the advantage of being scalable to large dataset and less sensitive to parameters. In addition, it has the ability of generating variable importance and information about outliers automatically. It is also reported in [20] that forest can yield a more realistic uncertainty in the ambiguous feature region, in comparison to GPR that tends to return largely over-confident prediction.

The weakness of RFR is that it is poor in extrapolating points beyond the value range of target variable within the training data, as we shall explain in more detail in Section 4.1.

[2] There are other weak learners that define the split functions, such as general oriented hyperplane or quadratic function. A more complex splitting function would lead to higher computational complexity.

2.3.4 Additional Considerations

We have discussed various linear and nonlinear functions for performing crowd density regression. Note that the functional form becomes more critical when one does not have sufficient training set that encompasses all the anticipated densities in a scene. If that is the case, extrapolation outside the training range has to be performed, with increasing room of failure when the extrapolation goes further beyond the existing data range, due to the mismatch between the regression assumption and the actual feature to count mapping.

A closely related consideration is at what level the learning should be performed. Most existing methods (see the 'level' column in Table 1) take a global approach by applying a single regression function over the whole image space with input variables being the holistic features of a frame (e.g. total area of foreground segment), and target variable being the total people count in that frame. An obvious limitation of this global approach is that it applies a global regression function over the whole image space, ignoring specific crowd structure in different regions. This can be resolved by dividing the image space up into regions and fitting separate function in each region [56,87]. The regions can be cells having regular size, or having different resolutions driven by the scene perspective to compensate the distortion [56].

One can also approximate the people count at blob-level [46], i.e. estimates the number of people in each foreground blob and obtains the total people count by summing the blob-level counts. Lempitsky et al. [44] go one step further to model the density at each pixel, casting the problem as that of estimating an image density whose integral over any image region gives the count of objects within that region. The aforementioned segment-and-model strategies facilitate counting at arbitrary locations, which is impossible using a holistic approach. In addition, a potential gain in estimation accuracy may be obtained [44]. This however comes at a price of increased annotation effort. e.g. requiring a large amount of dotted annotations on head or pedestrian positions in all training images [44].

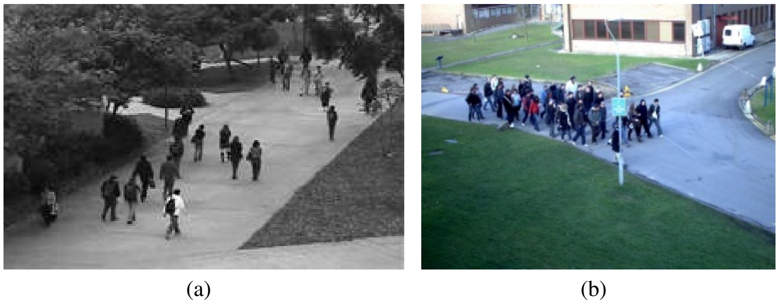
3 Evaluation Settings

Previous work [12, 44, 54, 56] have independently performed analyses on different components in the crowd counting pipeline such as feature extraction, perspective normalisation, and regression modelling. The scenarios studied, however, are rather diverse in terms of both evaluation data and experimental settings. It can be hard to compare them in order to draw general conclusions on their effectiveness. In this study we aim to provide a more exhaustive comparative evaluation to factor out the contributions of different components and identify potential bottlenecks in algorithm design for crowd counting and profile analysis.

3.1 Datasets

Two benchmark datasets were used for comparative algorithm evaluation, namely UCSD pedestrian dataset (*ucsd*) and PETS 2009 dataset (*pets*). Example frames are shown in Fig. 8. Apart from the two established benchmark datasets, a new and more realistic shopping *mall* dataset is also introduced in this study. This *mall* dataset was collected from a publicly accessible webcam in the course of two months from Feb 2011 to Apr 2011. A portion of 2000 frames recorded during peak hours were selected for the comparative algorithm evaluation. As can be seen from the sample images in Fig. 9, this new dataset is challenging in that it covers crowd densities from sparse to crowded, as well as diverse activity patterns (static and moving crowds), under large range of illumination conditions at different time of the day. Also note that the perspective distortion is more severe than the *ucsd* and *pets* datasets, thus individual objects may experience larger change in size and appearance at different depths of the scene. The details of the three datasets are given in Table 2.

For evaluation purpose, we resized the images from the *pets* dataset to 384 × 288, and the images from the *mall* dataset to 320 × 240. All colour images were converted to grayscale images prior to feature extraction. We annotated the data exhaustively by labelling the head position of every pedestrian in all frames. An example of annotated frame is shown in Fig. 9. The ground truth, together with the raw video sequence, extracted features, and the train/test partitions can be downloaded at http://www.eecs.qmul.ac.uk/∼ccloy/.



(a) (b)

Fig. 8 (a) UCSD Pedestrian Dataset (*ucsd*), (b) PETS 2009 Benchmark Dataset (*pets*).

Data	N_f	R	FPS	D	Tp
ucsd [13]	2000	238 × 158	10	11–46	49885
pets [27]	1076	384 × 288	7	0–43	18289
mall	2000	320 × 240	<2	13–53	62325

Table 2 Dataset properties: N_f = number of frames, R = Resolution, FPS = frame per second, D = Density (minimum and maximum number of people in the ROI), and Tp = total number of pedestrian instances.

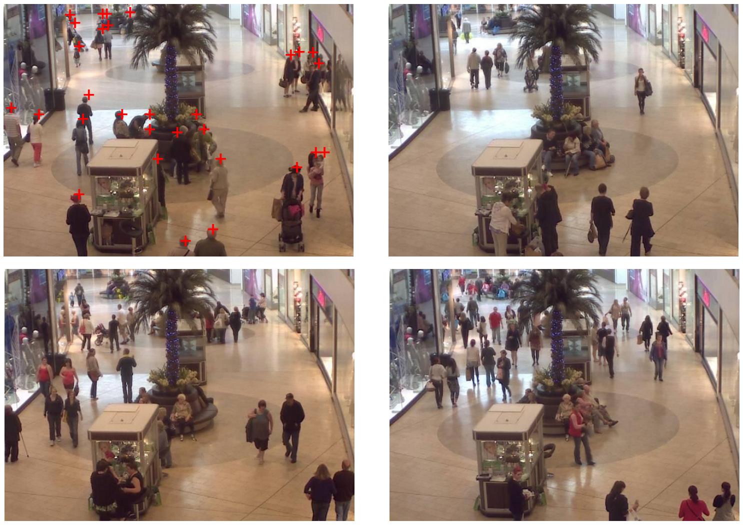


Fig. 9 The new shopping mall dataset. The top-left figure shows an example of annotated frame.

3.2 Features and Regression Models

We selected features and regression methods that are both representative and promising in terms of originally reported performance. While we could not evaluate all the available features or methods exhaustively due to unavailability of original codes and practical time and space constraints, we consider that these evaluations giving an accurate portrait of the state-of-the-art.

We extracted segment, edge, GLCM, and LBP features following the methods described in Section 2.3.1. For both *ucsd* and *pets* datasets, scene lighting were stable so we employed a static background subtraction method based on minimum cuts [17][3] to extract the foreground segments. For the *mall* dataset, gradual illumination change was observed, we therefore adopted a dynamic background modelling method [95].

All features were perspective normalised (see Section 2.3.2) and a feature vector was formed by concatenating the features, into $\mathbf{x} \in \mathbb{R}^D$, which was used as the input for the regression models. Prior to feeding the features into the regression models, all features were scaled to the [0 1] interval. A list of the regression models and their associated settings is given below

- Linear regression (LR)
- Partial least-squares regression (PLSR) – 10 latent components
- Kernel ridge regression (KRR) – linear kernel with four-fold cross-validation for parameter optimisation

[3] Codes available at http://www.eecs.qmul.ac.uk/∼ccloy/.

- Least-squares support vector regression (LSSVR) – linear kernel with four-fold cross-validation for parameter optimisation
- Gaussian processes regression (GPR) – linear kernel + RBF kernel as in [13][4]. The parameters are first initialised to random values and optimised using conjugate gradient optimiser.
- Random forest regression (RFR) – 500 trees, the number of parameters made available for node splitting was fixed to square-root of the feature dimension, and the minimum size of terminal nodes was set to 5.

3.3 Evaluation Metrics

We employed three metrics in performance evaluation. Two of the metrics are widely used as performance indicators for crowd counting, namely *mean absolute error* and *mean squared error*. Mean absolute error is defined as

$$\varepsilon_{\text{abs}} = \frac{1}{N} \sum_{n=1}^{N} |y_n - \hat{y}_n|. \tag{15}$$

Mean squared error is given as

$$\varepsilon_{\text{sqr}} = \frac{1}{N} \sum_{n=1}^{N} (y_n - \hat{y}_n)^2, \tag{16}$$

where N is the total number of test frames, y_n is the actual count, and $\hat{y}_n$ is the estimated count of nth frame. Note that as a result of the squaring of each difference, ε_{sqr} effectively penalises large errors more heavily than small ones. The above two metrics are indicative in quantifying the error of estimation of the crowd count. However, as pointed out by [19], these metrics contain no information about the crowdedness of the region of interest. To that end, [19] proposed another performance metric to take the crowdedness into account – we name it as *mean deviation error*, which is essentially a normalised ε_{abs}

$$\varepsilon_{\text{dev}} = \frac{1}{N} \sum_{n=1}^{N} \frac{|y_n - \hat{y}_n|}{y_n}. \tag{17}$$

4 Performance Comparison

In the following we report comparative evaluation results on three aspects, i.e.model choices, feature robustness, and model sensitivity to perspective.

[4] An interesting aspect not examined in our study is the effect of different kernels and their relations with different kernel methods for crowd regression.

4.1 Model Choices

The goals of this experiment are to (1) compare the performance of different regression models under different crowdedness levels, and (2) evaluate their generalisation capability to unseen density. These two aspects are somewhat less explicitly studied in existing work. However, they are essential since a regressor may behave differently under different crowdedness levels, and often, it needs to extrapolate outside the anticipated density range in real-world scenarios.

We employed the same segment+edge+LBP features across all regression models. To simulate different crowdedness levels, we divided a dataset into two partitions: one for sparse scenario and another one for crowded scenario, of which the details are provided in Table 3.

Data	**Sparse scenario (no. frames)**	**Crowded scenario (no. frames)**
ucsd	1058 ($\leq$23 people, train=400, test=658)	942 ($>$23 people, train=400, test=542)
pets	800 ($\leq$10 people, train=400, test=400)	276 ($>$10 train=100, test=176)
mall	972 ($\leq$30 people, train=400, test=572)	1028 ($>$30 people, train=400, test=628)

Table 3 Number of frames allocated for the sparse and crowded seenarios. Information inside the brackets contain the definition of crowdedness, together with the training and test set proportions.

Model performance under different crowdedness levels: To evaluate a regressor under the sparse scenario, we trained and tested the model using the sparse partition of a dataset. Similar procedures were applied using the crowded partition of a dataset to test a model under crowded scenario. Figure 10 shows the performance of the six regression models under the sparse and crowded scenarios. Note that we only presented the mean deviation error since other metrics exhibited similar trends in this experiment.

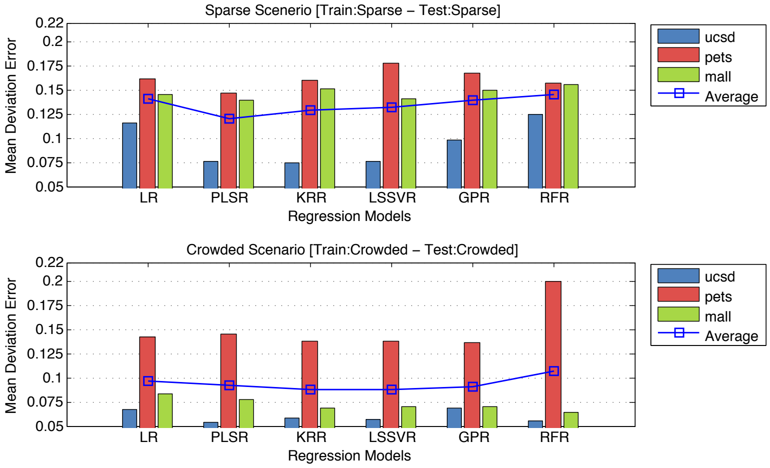


Fig. 10 Comparison of mean deviation error (lower is better) between regression models in sparse and crowd scenarios.

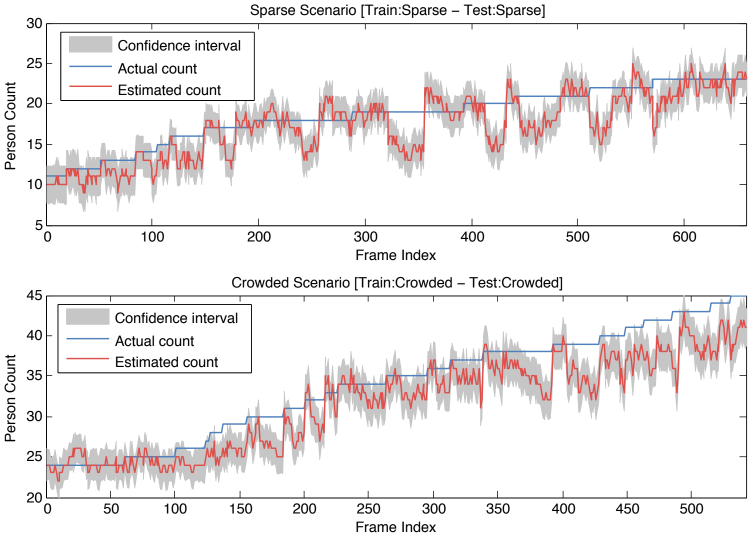


Fig. 11 Labelled ground truth vs. estimated count by Gaussian processes regression on sparse and crowded scenarios of *ucsd* dataset. The estimated count is accompanied by ± two standard deviations corresponding to a 95% confidence interval.

It is evident that models which can effectively deal with multicollinearity issue, such as LSSVR, PLSR, and KRR, consistently performed better than other models in both the sparse and crowded partitions, as shown in Fig. 10. Specifically, over-fitting were less an issue to the aforementioned models, which either add a regularisation term[5] into the error function or by projecting the input variables onto a lower-dimensional space.

In contrast, LR was ill-conditioned due to highly-correlated features, thus yielding poorer performance as compared to LSSVR, PLSR, and KRR. The performance of GPR was mixed. The error rate of RFR was extremely high in the *pets* crowded partition as the forest structure was too complex given the limited amount of training data. As a result, its generalisation capability was compromised due to the overfitting. In other datasets, RFR showed comparable results to other regression methods.

We found that existing performance metrics including the mean deviation error [19], which is normalised by the actual count (see Section 3.3), are not appropriate for comparing scenarios with enormous difference in densities. Specifically, our findings were rather counter intuitive in that all regressors performed better in the crowded scenario than the sparse scenario. We note that the lower mean deviation errors in a crowded scene are largely biased by the much larger actual count serving as the denominator in Equation (17). To vindicate our observation, we plotted the performance of GPR on the *ucsd* dataset in Fig. 11 and found that the regressor performance did not differ much across sparse and crowded scenarios.

[5] [64] provides detailed discussion on the regularisation approach with the Gaussian process viewpoint.

Generalisation to unseen density: To evaluate the generalisation capability of a regression model to unseen density, we tested it against two scenarios: (1) generalising from crowded to sparse environment, and (2) generalising from sparse to crowded environment. In the first scenario, we trained a regressor with the crowded partition and tested it on the sparse partition. We switched the crowded and sparse partitions in the second scenario. The same data partitions in Table 3 were used.

	Train:Crowded - Test:Sparse			Train:Sparse - Test:Crowded		
	Mean Abs. Error	**Mean Sq. Error**	**Mean Dev. Error**	**Mean Abs. Error**	**Mean Sq. Error**	**Mean Dev. Error**
LR	**1.7448**	**4.8034**	**0.1013**	**2.8811**	**13.0382**	**0.0860**
PLSR	2.0208	6.2892	0.1170	4.0934	25.4034	0.1184
KRR	2.0284	6.3176	0.1172	4.1805	26.4459	0.1210
LSSVR	2.0123	6.2202	0.1163	4.2304	27.2070	0.1225
GPR	2.3081	7.6730	0.1330	3.8089	20.6921	0.1119
RFR	6.0851	50.5539	0.3882	9.4671	134.2994	0.2681

(a) *ucsd*

	Train:Crowded - Test:Sparse			Train:Sparse - Test:Crowded		
	Mean Abs. Error	**Mean Sq. Error**	**Mean Dev. Error**	**Mean Abs. Error**	**Mean Sq. Error**	**Mean Dev. Error**
LR	1.3137	3.1612	0.2765	2.5833	11.0978	0.1263
PLSR	1.4087	3.6263	0.2835	2.7428	12.3732	0.1337
KRR	**1.2612**	**2.8237**	**0.2643**	**2.5507**	**10.7971**	**0.1248**
LSSVR	1.4737	3.8763	0.3083	2.6051	11.2500	0.1272
GPR	1.4238	3.5463	0.2849	3.3986	20.1159	0.1631
RFR	6.7138	56.4937	1.7037	9.3877	156.5036	0.4279

(b) *pets*

	Train:Crowded - Test:Sparse			Train:Sparse - Test:Crowded		
	Mean Abs. Error	**Mean Sq. Error**	**Mean Dev. Error**	**Mean Abs. Error**	**Mean Sq. Error**	**Mean Dev. Error**
LR	5.4959	45.9012	0.2414	**4.5360**	**29.5379**	**0.1225**
PLSR	**4.9877**	**35.0432**	**0.2171**	5.6625	42.8628	0.1499
KRR	5.1070	36.1893	0.2225	5.8006	44.0924	0.1534
LSSVR	5.0216	35.2623	0.2189	5.7704	43.6109	0.1526
GPR	5.4969	39.4660	0.2389	6.9426	59.8687	0.1835
RFR	7.1080	64.0175	0.3127	8.6994	95.4601	0.2276

(c) *mall*

Table 4 Comparison of generalisation capability of different regression models to unseen density. Best performance is highlighted in bold.

Regression models that worked well within known crowd density may not perform as good given unseen density. In particular, as shown in Table 4, simple linear regression models such as LR and PLSR returned surprisingly good performance in both the *ucsd* and *mall* datasets, outperforming their non-linear counterparts. The results suggest that the regression assumption of linear regression models, though simple, could be less susceptible to unseen density and matched closer with the feature-to-density trend in the considered scenarios. The performance of RFR was poorest among the regression models. The results agree with our expectation about its weakness in generalisation as discussed in Section 2.3.3.

It was observed that the generalisation performance reported in Table 4, were much poorer than those obtained when we trained and tested a regressor using the same density range. In particular, the regressors tend to overestimate or underestimate depending on the extrapolation direction, as shown in Fig. 12. In addition, the further the extrapolation goes outside the training range, the larger the error in the estimation due to difference between the learned model and the actual feature-to-density trend. Note that there was no concrete evidence to show that generalising from crowded to sparse environment was easier than generalising from sparse to crowded scene.

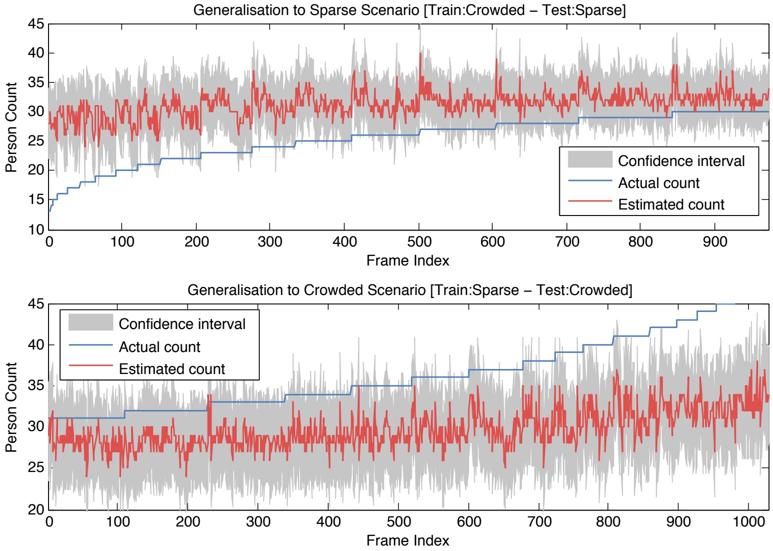


Fig. 12 Generalisation to unseen density: Labelled ground truth vs. estimated count by Gaussian processes regression on *mall* dataset. (a) Training on crowded partition and testing on sparse partition results in over-estimation, and (b) doing the other way round results in under-estimation. The estimated count is accompanied by ± two standard deviations corresponding to a 95% confidence interval.

4.2 Feature Robustness

The objective of this experiment is to compare the performance on using different types of features, e.g. segment-based features, edge-based features, texture-based features (in particular GLCM and LBP), as well as their combination, given different crowdedness levels in a scene. As in Section 4.1, we conducted the evaluation using sparse and crowded partitions. The results are depicted in Fig. 13 and Fig. 14.

Robustness of individual features: It is observed that different features can be more important given different crowdedness levels. In general, the averaged performance suggests that the segment-based features were superior to other features. This is not surprising since the foreground segment carries useful information about the area occupied by objects of interest and it thus intrinsically correlate to the number of pedestrians in a scene. However in the *ucsd* and *mall* datasets, a decrease in performance gap was observed between the edge or texture-based features and the segment-based features when we switched from sparse partition to crowded partition. This observation is intuitive since given a more crowded environment with frequent inter-object occlusion, segment-based features would suffer, whilst edge and texture that inherently encoded the inter-object boundary and internal patterns would carry more discriminative visual cues for density mapping.

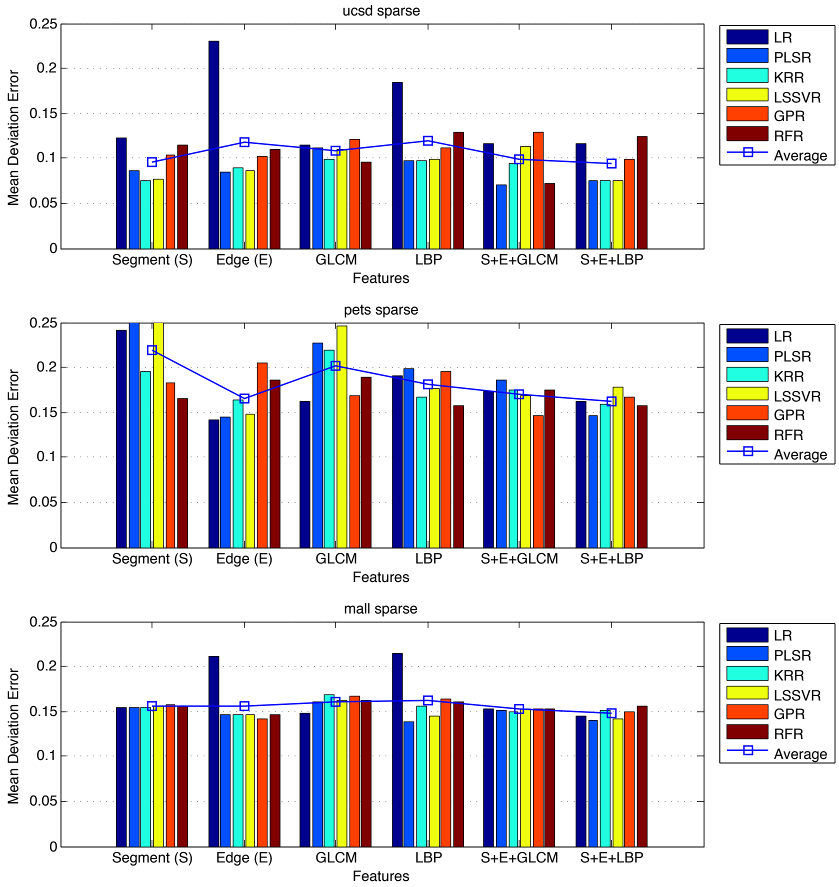


Fig. 13 Sparse partition: the mean deviation error (lower is better) vs. different features.

Does combining features help?: From the averaged performance, it is observed that combining different features together could lead to a better performance in general. For instance, when the LBP-based features were used in combination with the segment and edge-based features, the mean deviation error was reduced by 2%-14%. This finding supports the practice of employing a combination of features (see Table 1).

Nevertheless, when we examined the performance of individual regression models, it was found that combining all the features did not necessarily produce better performance. For example, using the segment-based features alone in the crowded *mall* partition one would get higher performance; or using the edge features alone with RFR gained more accurate counts in the sparse *ucsd* partition. The results suggests the need for feature selection to discover the suitable set of features given different crowd densities and different regression models.

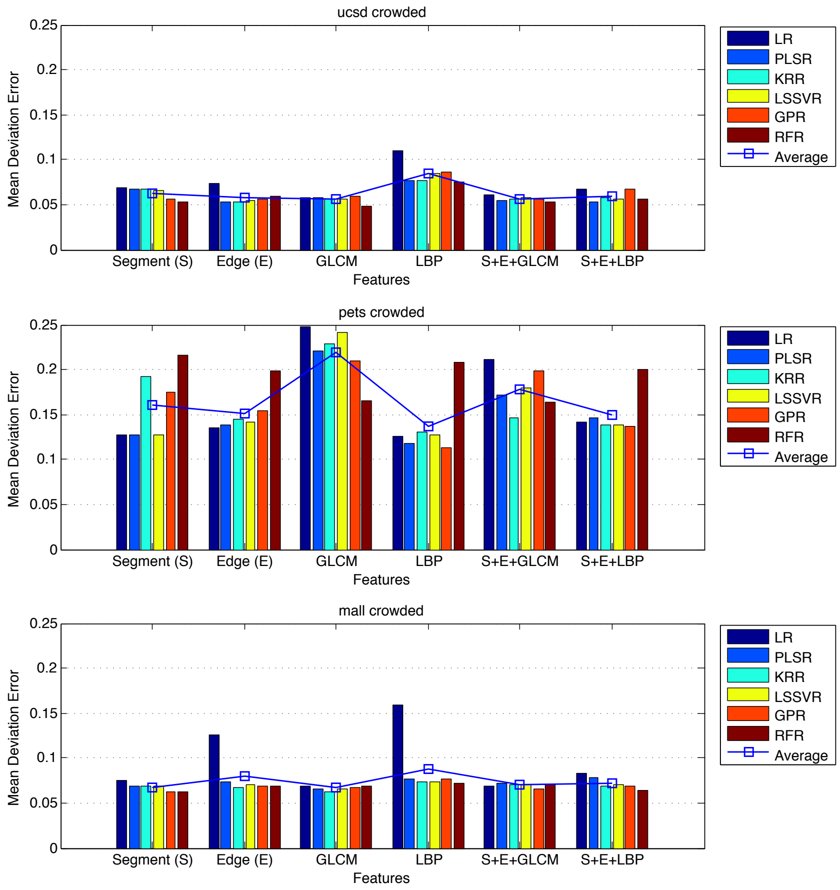


Fig. 14 Crowded partition: the mean deviation error (lower is better) vs. different features.

4.3 Geometric Correction

Geometry correction is critical in crowd counting since objects at different depths of the scene would lead to huge variation in the extracted features. To minimise the influence of perspective distortion, correction is often conducted in existing studies but often without explicit analysis on how its sensitivity would affect the final counting performance. In this experiment, we investigated the sensitivity of crowd counting performance to a widely adopted perspective normalisation method described in [13] (see Section 2.3.2). Evaluation was carried out on the *ucsd* dataset, with 800 frames for training and the remaining 1200 frames held out for testing following the partitioning scheme suggested in [13].

Effectiveness of geometric correction: It is evident from Table 5 that perspective correction is essential in achieving accurate crowd density estimation. Specifically, depending on different regression models, an improvement of around 20% was gained in the mean absolute error by applying perspective correction.

	With Perspective Normalisation			Without Perspective Normalisation		
	Mean Abs. Error	Mean Sq. Error	Mean Dev. Error	Mean Abs. Error	Mean Sq. Error	Mean Dev. Error
LR	2.1608	7.1608	0.1020	2.6308	10.2558	0.1288
PLSR	2.0267	6.6717	0.1007	2.5792	10.0025	0.1271
KRR	2.3433	8.4800	0.1166	2.9167	11.6133	0.1392
LSSVR	2.1100	6.6383	0.1014	2.5825	9.6925	0.1262
GPR	2.1425	7.1358	0.1055	2.7833	10.5200	0.1328
RFR	2.3392	7.9708	0.1129	2.8492	10.8492	0.1332
Average	2.1871	7.3429	0.1065	2.7236	10.4889	0.1312

Table 5 Comparison of mean absolute error (lower is better) on *ucsd* dataset when crowd density was estimated with and without perspective correction.

Sensitivity to errors in geometric correction: It is interesting to examine how a minor error introduced by manual measurement will propagate through the counting by regression pipeline. We manually measure the heights, denoted as h_1 and h_2, of a reference pedestrian at two extremes of the ground plane rectangle of the *ucsd* dataset (see Fig. 15). We varied h_2, the height at the further extreme at $+/-$ 5 pixels with a step size of 1 pixel. Given a frame with resolution of 238×158, this is a reasonable error range that is likely to occur during the manual measurement. Perspective maps within this pixel deviation range were generated, and the crowd counting performances of different models were subsequently recorded.

A minor measurement error in h_2 could result in a great change in perspective map, as shown in Fig. 4.3. Specifically, when h_2 had a smaller value, e.g.$h_2 - 5$ pixels, a steeper slope in the perspective normalisation weight vector was observed. On the contrary, given $h_2 + 5$ pixels, the object size at $\overline{cd}$ was larger so the perspective normalisation weight vector had a lower slope. Using these different perspective maps we evaluated performances of different regression models.

It is clear from the results depicted in Fig. 16 that different perspective maps will lead to drastic difference in estimation performance, e.g. as much as 10% of difference from that obtained using initial measurement. The results suggest that the initial measurement h_2 may not be accurate, since more accurate counts were obtained at $h_2 - 5$ pixels. A subsequent validation through averaging multiple measurements confirmed that the initial measurement indeed deviated from the accurate value. Hence one should not rely on a single round of measurement, but to seek for more reliable perspective statistics by averaging measurements obtained across multiple attempts. Note that deviation from the ‘exact’ perspective map may not necessarily lead to a bad consequence sometimes as the steeper weight slope will counteract the problems of poor segmentation and inter-object occlusion at the back of the scene.

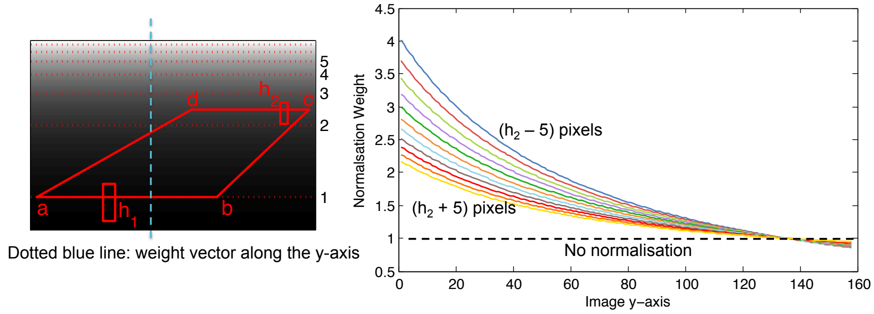


Fig. 15 (Top) Perspective normalisation map of *ucsd* dataset. (Bottom) Each line in the chart corresponds to a weight vector along the y-axis (e.g. the dotted blue line) of each perspective map produced as a result of varying measurement errors in h_2, ranging from -5 pixels to +5 pixels with a step size of 1 pixel.

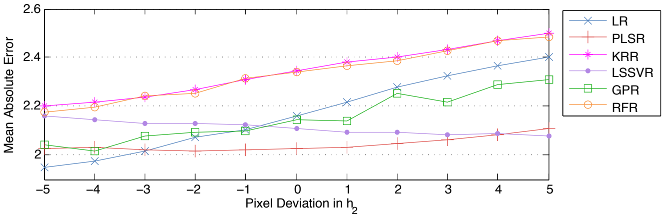


Fig. 16 Mean absolute error on *ucsd* dataset as a result of varying measurement errors in h_2: most regression methods experienced drastic performance change as much as over 10% given just a minor deviation in the manual measurement.

5 Crowd Profiling

One of the ultimate goals of crowd counting is to profile the crowd behaviours and density patterns spatially and temporally, e.g. how many people in a region of interest at what time and predicting the trend. The profiling statistic can serve as useful hints for controlling crowd movements, designing evacuation routes, and improving product display strategy to attract more crowds to a shop. An example of such a crowd profiling application is depicted in Fig. 17, of which the local density map was generated through learning cell-level counts using separate regressors. A more scalable way based on a single regression model with multiple outputs can also be employed [15].

The top row of Fig. 17 shows the footage frames of a shopping mall view overlaid with heat maps, of which the colour codes representing the crowd density, with larger crowd represented by red squares and smaller crowd with blue squares. An interesting usage of the crowd density map is to study the crowd movement profile in front of a shop, e.g. the two selected regions (blue and red) in Fig. 17. The number of people appear in these areas over time can be profiled as shown in the two plots at the bottom of Fig. 17. In addition, activity correlation between these two regions

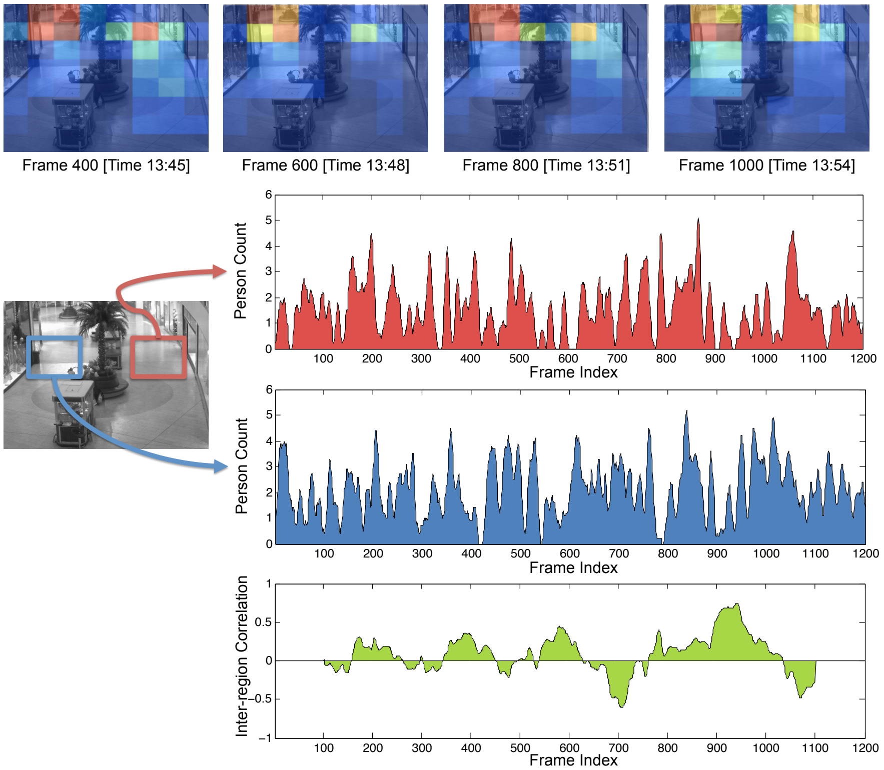


Fig. 17 One of the goals of crowd counting is to profile the crowd behaviours and density patterns spatially and temporally, e.g. how many people in a region of interest at what time (see text for details).

can be computed to examine their crowd flow dependency, as shown in the last plot. Analysing these local crowd patterns over time and their correlations globally can reveal useful information about the shop visitors, such as their interests towards the product display, walking pace, and intention of buying, without the need for registering individual's identities therefore minimising privacy violation.

The crowd counting application can benefit from extensions such as functional learning of regions [75] (e.g. sitting area, entrance of shops) to better reflect the activity modes at different regions; or combination with cooperative multi-camera network surveillance [32, 52] to model the density and activity correlation in the camera network [94].

6 Findings and Analysis

We shall summarise our main findings as follows:

Regression model choices: Our evaluation reveals that regression models that are capable of dealing with multicollinearity among features, e.g. KRR, PLSR, LSSVR generally give better performance than other regression models such as LR and RFR. The aforementioned models, i.e. KRR, PLSR, and LSSVR have not been significantly explored in existing counting by regression literature.

In general, linear model is expected to give poorer performance as its linear property imposes a limitation on the model in capturing only the linear relationship between the people count and low-level features [4,22,46]. In most cases especially in crowded environments, the visual observations and people count will not be linearly related. Nonlinear methods in principle allow one to model arbitrary nonlinearities between the mapping from input variables to target people count. In addition, employing a nonlinear method would help in remedying the dimensionality problem since observations typically exhibit strong correlation in a nonlinear manifold, whose intrinsic dimensionality is smaller than the input space [6].

However, our study suggests that the actual performance of a regression model can be quite different from what one may anticipate, subject to the nature of data, especially when it is applied to unseen density. Despite all the evaluated regression techniques suffer poor extrapolation beyond the training data range, simple linear regression models such as LR, is found to be more resistant towards the introduction of unseen density. Its performance can be better than other nonlinear models such as GPR and LSSVR.

We have emphasised that it is impractical to assume the access to all full density range during the training stage, thus the capability of generalising to unseen density is critical. An unexplored approach of resolving the problem is to transfer the knowledge from other well-annotated datasets that cover wider range of crowd density. This is an open and challenging problem in crowd counting task given different environmental factors of source and target scenes, e.g. variations in lighting conditions and camera orientations.

Features selection: Our results suggest that different features can be more useful given different crowd configurations and densities. In sparse scenes, foreground segment-based features alone can provide sufficient information required for crowd density estimation. However, when a scene becomes crowded with frequent inter-object occlusions, the role of edge-based features and texture-based features becomes increasingly critical. We also found that combining all features do not always help, depending on the dataset and regression model of choice. These findings suggest the importance of feature selection, i.e. selecting optimal feature combinations given different crowd structures and densities, through discarding redundant and irrelevant features. The feature selection problem has been largely ignored in existing crowd counting research.

Perspective correction: The performance of counting by regression can be severely influenced by the accuracy of perspective weight estimation. Perspective map generation based on manual measurement is simple but could be error-prone. We suggest that multiple measurements are necessary to ensure conciseness of the estimation normalisation weights. Robust auto-calibration methods such as [40, 50] are also recommended as an alternative to the manual approach.

7 Further Reading

Interested readers are referred to the following further readings:

- [32] for a general discussion on applications and advances in automated analysis of human activities for security and surveillance
- [33] for a comprehensive treatment of visual analysis of behaviour from algorithm-design perspectives
- [37] for a survey on crowd analysis
- [12] for a detailed discussion on using Bayesian techniques for regression-based counting

References

1. Abdi, H.: Partial least square regression (pls regression). Encyclopedia of Measurement and Statistics pp. 740–744 (2007)
2. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(12), 2037–2041 (2006)
3. Ali, S., Shah, M.: Floor fields for tracking in high density crowd scenes. In: European Conference on Computer Vision, pp. 1–24 (2008)
4. Benabbas, Y., Ihaddadene, N., Yahiaoui, T., Urruty, T., Djeraba, C.: Spatio-temporal optical flow analysis for people counting. In: IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. 212–217 (2010)
5. Benenson, R., Mathias, M., Timofte, R., Gool, L.V.: Pedestrian detection at 100 frames per second. In: IEEE Conference Computer Vision and Pattern Recognition (2012)
6. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer-Verlag (2007)
7. Brostow, G.J., Cipolla, R.: Unsupervised Bayesian detection of independent motion in crowds. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 594–601 (2006)
8. Canny, J.: A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on (6), 679–698 (1986)
9. Chan, A., Dong, D.: Generalized gaussian process models. In: IEEE Conference Computer Vision and Pattern Recognition, pp. 2681–2688. IEEE (2011)
10. Chan, A., Vasconcelos, N.: Modeling, clustering, and segmenting video with mixtures of dynamic textures. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(5), 909–926 (2008)
11. Chan, A., Vasconcelos, N.: Bayesian poisson regression for crowd counting. In: IEEE International Conference on Computer Vision, pp. 545–551. IEEE (2009)

12. Chan, A., Vasconcelos, N.: Counting people with low-level features and bayesian regression. IEEE Transactions on Image Processing **21**(4), 2160–2177 (2012)
13. Chan, A.B., Liang, Z.S.J., Vasconcelos, N.: Privacy preserving crowd monitoring: Counting people without people models or tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–7 (2008)
14. Chan, A.B., Morrow, M., Vasconcelos, N.: Analysis of crowded scenes using holistic properties. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (2009)
15. Chen, K., Loy, C.C., Gong, S., Xiang, T.: Feature mining for localised crowd counting. In: British Machine Vision Conference (2012)
16. Cho, S., Chow, T., Leung, C.: A neural-based crowd estimation by hybrid global learning algorithm. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics **29**(4), 535–541 (1999)
17. Cohen, S.: Background estimation as a labeling problem. In: IEEE International Conference on Computer Vision, vol. 2, pp. 1034–1041 (2005)
18. Cong, Y., Gong, H., Zhu, S., Tang, Y.: Flow mosaicking: Real-time pedestrian counting without scene-specific learning. In: IEEE Conference Computer Vision and Pattern Recognition, pp. 1093–1100 (2009)
19. Conte, D., Foggia, P., Percannella, G., Vento, M.: A method based on the indirect approach for counting people in crowded scenes. In: IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. 111–118. IEEE (2010)
20. Criminisi, A., Shotton, J., Konukoglu, E.: Decision forest for classification, regression, density estimation, manifold learning and semi-supervised learning. Tech. Rep. MSR-TR-2011-114, Microsoft Research (2011)
21. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 886–893 (2005)
22. Davies, A., Yin, J., Velastin, S.: Crowd monitoring using image processing. Electronics & Communication Engineering Journal **7**(1), 37–47 (1995)
23. De Brabanter, K., De Brabanter, J., Suykens, J., De Moor, B.: Approximate confidence and prediction intervals for least squares support vector regression. IEEE Transactions on Neural Networks (99), 1–11 (2011)
24. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: An evaluation of the state of the art. IEEE Transactions on Pattern Analysis and Machine Intelligence (99), 1–1 (2011)
25. Dong, L., Parameswaran, V., Ramesh, V., Zoghlami, I.: Fast crowd segmentation using shape indexing. In: IEEE International Conference on Computer Vision (2007)
26. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(9), 1627–1645 (2010)
27. Ferryman, J., Crowley, J., Shahrokni, A.: Pets 2009 benchmark data. http://www.cvg.rdg.ac.uk/WINTERPETS09/a.html
28. Gall, J., Yao, A., Razavi, N., Van Gool, L., Lempitsky, V.: Hough forests for object detection, tracking, and action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(11), 2188–2202 (2011)
29. Ge, W., Collins, R.: Marked point processes for crowd counting. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2913–2920 (2009)
30. Ge, W., Collins, R.: Crowd detection with a multiview sampler. European Conference on Computer Vision pp. 324–337 (2010)
31. Geladi, P., Kowalski, B.: Partial least-squares regression: a tutorial. Analytica chimica acta **185**, 1–17 (1986)
32. Gong, S., Loy, C.C., Xiang, T.: Security and surveillance. Visual Analysis of Humans pp. 455–472 (2011)
33. Gong, S., Xiang, T.: Visual analysis of behaviour: from pixels to semantics. Springer-Verlag New York Inc (2011)
34. Haralick, R., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE Transactions on Systems, Man and Cybernetics **3**(6), 610–621 (1973)

35. Helbing, D., Farkas, I., Molnar, P., Vicsek, T.: Simulation of pedestrian crowds in normal and evacuation situations. Pedestrian and evacuation dynamics **21** (2002)
36. Hoerl, A., Kennard, R.: Ridge regression: Biased estimation for nonorthogonal problems. Technometrics pp. 55–67 (1970)
37. Jacques Junior, J., Musse, S., Jung, C.: Crowd analysis using computer vision techniques. Signal Processing Magazine, IEEE **27**(5), 66–77 (2010)
38. Kong, D., Gray, D., Tao, H.: Counting pedestrians in crowds using viewpoint invariant training. In: British Machine Vision Conference. Citeseer (2005)
39. Kong, D., Gray, D., Tao, H.: A viewpoint invariant approach for crowd counting. In: International Conference on Pattern Recognition, vol. 3, pp. 1187–1190 (2006)
40. Krahnstoever, N., Mendonca, P.: Bayesian autocalibration for surveillance. In: IEEE International Conference on Computer Vision, vol. 2, pp. 1858–1865. IEEE (2005)
41. Kratz, L., Nishino, K.: Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1446–1453 (2009)
42. Lampert, C.: Kernel methods in computer vision, vol. 4. Now Publishers Inc (2009)
43. Leibe, B., Seemann, E., Schiele, B.: Pedestrian detection in crowded scenes. In: IEEE Conference Computer Vision and Pattern Recognition, vol. 1, pp. 878–885 (2005)
44. Lempitsky, V., Zisserman, A.: Learning to count objects in images. In: Advances in Neural Information Processing Systems (2010)
45. Li, J., Huang, L., Liu, C.: CASIA pedestrian counting dataset: http://cpcd.vdb.csdb.cn/page/showItem.vpage? id=automation.dataFile/1
46. Li, J., Huang, L., Liu, C.: Robust people counting in video surveillance: Dataset and system. In: IEEE International Conference on Advanced Video and Signal-Based Surveillance, pp. 54–59. IEEE (2011)
47. Li, M., Zhang, Z., Huang, K., Tan, T.: Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In: International Conference on Pattern Recognition, pp. 1–4 (2008)
48. Lin, S., Chen, J., Chao, H.: Estimation of number of people in crowded scenes using perspective transformation. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans **31**(6), 645–654 (2001)
49. Lin, T., Lin, Y., Weng, M., Wang, Y., Hsu, Y., Liao, H.: Cross camera people counting with perspective estimation and occlusion handling. In: IEEE International Workshop on Information Forensics and Security (2011)
50. Liu, J., Collins, R.T., Liu, Y.: Surveillance camera autocalibration based on pedestrian height distributions. In: British Machine Vision Conference (2011)
51. Loy, C.C., Xiang, T., Gong, S.: Time-delayed correlation analysis for multi-camera activity understanding. International Journal of Computer Vision **90**(1), 106–129 (2010)
52. Loy, C.C., Xiang, T., Gong, S.: Incremental activity modelling in multiple disjoint cameras. IEEE Transactions on Pattern Analysis and Machine Intelligence (2011)
53. Loy, C.C., Xiang, T., Gong, S.: Salient motion detection in crowded scenes. In: Special Session on ‘Beyond Video Surveillance: Emerging Applications and Open Problems’, International Symposium on Communications, Control and Signal Processing, Invited Paper (2012)
54. Ma, R., Li, L., Huang, W., Tian, Q.: On pixel count based crowd density estimation for visual surveillance. In: IEEE Conference on Cybernetics and Intelligent Systems, vol. 1, pp. 170–173. IEEE (2004)
55. Ma, W., Huang, L., Liu, C.: Advanced local binary pattern descriptors for crowd estimation. In: Pacific-Asia Workshop on Computational Intelligence and Industrial Application, vol. 2, pp. 958–962. IEEE (2008)
56. Ma, W., Huang, L., Liu, C.: Crowd density analysis using co-occurrence texture features. In: International Conference on Computer Sciences and Convergence Information Technology, pp. 170–175 (2010)
57. Marana, A., Costa, L., Lotufo, R., Velastin, S.: On the efficacy of texture analysis for crowd monitoring. In: International Symposium on Computer Graphics, Image Processing, and Vision, pp. 354–361 (1998)

58. Marana, A., da Fontoura Costa, L., Lotufo, R., Velastin, S.: Estimating crowd density with minkowski fractal dimension. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 6, pp. 3521–3524. IEEE (1999)
59. Marana, A., Velastin, S., Costa, L., Lotufo, R.: Estimation of crowd density using image processing. In: Image Processing for Security Applications, pp. 11–1 (1997)
60. Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behaviour detection using social force model. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 935–942 (2009)
61. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(7), 971–987 (2002)
62. Pätzold, M., Evangelio, R., Sikora, T.: Counting people in crowded environments by fusion of shape and motion information. In: IEEE International Conference on Advanced Video and Signal based Surveillance, pp. 157–164. IEEE (2010)
63. Rabaud, V., Belongie, S.: Counting crowded moving objects. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 705–711 (2006)
64. Rasmussen, C.E., Williams, C.K.I.: Gaussian Process for Machine Learning. MIT Press (2006)
65. Rodriguez, M., Laptev, I., Sivic, J., Audibert, J.: Density-aware person detection and tracking in crowds. In: IEEE International Conference on Computer Vision (2011)
66. Russell, D., Gong, S.: Minimum cuts of a time-varying background. In: British Machine Vision Conference, pp. 809–818 (2006)
67. Ryan, D., Denman, S., Fookes, C., Sridharan, S.: Crowd counting using multiple local features. In: Digital Image Computing: Techniques and Applications (2009)
68. Sabzmeydani, P., Mori, G.: Detecting pedestrians by learning shapelet features. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
69. Saunders, C., Gammerman, A., Vovk, V.: Ridge regression learning algorithm in dual variables. In: International Conference on Machine Learning, pp. 515–521 (1998)
70. Shan, C., Gong, S., McOwan, P.W.: Facial expression recognition based on local binary patterns: A comprehensive study. Image and Vision Computing **27**(6), 803–816 (2009)
71. Shawe-Taylor, J., Cristianini, N.: Kernel methods for pattern analysis. Cambridge Univ Pr (2004)
72. Smola, A., Schölkopf, B.: A tutorial on support vector regression. Statistics and computing **14**(3), 199–222 (2004)
73. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(8), 747–757 (2000)
74. Suykens, J., Vandewalle, J.: Least squares support vector machine classifiers. Neural Processing Letters **9**(3), 293–300 (1999)
75. Swears, E., Turek, M., Collins, R., Perera, A., Hoogs, A.: Automatic Activity Profile Generation from Detected Functional Regions for Video Scene Analysis, pp. 241–269. Springer (2012)
76. Tian, Y., Brown, L., Hampapur, A., Lu, M., Senior, A., Shu, C.: Ibm smart surveillance system (s3): event based video surveillance system with an open and extensible framework. Machine Vision and Applications **19**(5), 315–327 (2008)
77. Tu, P., Sebastian, T., Doretto, G., Krahnstoever, N., Rittscher, J., Yu, T.: Unified crowd segmentation. In: European Conference on Computer Vision (2008)
78. Tuzel, O., Porikli, F., Meer, P.: Pedestrian detection via classification on riemannian manifolds. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(10), 1713–1727 (2008)
79. Vapnik, V.: The nature of statistical learning theory. Springer-Verlag New York Inc (2000)
80. Viola, P., Jones, M.: Robust real-time face detection. International journal of computer vision **57**(2), 137–154 (2004)
81. Viola, P., Jones, M., Snow, D.: Detecting pedestrians using patterns of motion and appearance. International Journal of Computer Vision **63**(2), 153–161 (2005)
82. Wang, M., Li, W., Wang, X.: Transferring a generic pedestrian detector towards specific scenes. In: IEEE Conference Computer Vision and Pattern Recognition (2012)

83. Wang, M., Wang, X.: Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3401–3408. IEEE (2011)
84. Welling, M.: Support vector regression. Tech. rep., Department of Computer Science, University of Toronto (2004)
85. Wu, B., Nevatia, R.: Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, vol. 1, pp. 90–97. IEEE (2005)
86. Wu, B., Nevatia, R.: Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. International Journal of Computer Vision **75**(2), 247–266 (2007)
87. Wu, X., Liang, G., Lee, K., Xu, Y.: Crowd density estimation using texture analysis and learning. In: IEEE International Conference on Robotics and Biomimetics, pp. 214–219. IEEE (2006)
88. Yang, D., González-Baños, H., Guibas, L.: Counting people in crowds with a real-time network of simple image sensors. In: IEEE International Conference on Computer Vision, pp. 122–129 (2003)
89. Yeniay, O., Goktas, A.: A comparison of partial least squares regression with other prediction methods. Hacettepe Journal of Mathematics and Statistics **31**(99), 111 (2002)
90. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Journal of Computing Surveys **38**(4), 1–45 (2006)
91. Zhan, B., Monekosso, D.N., Remagnino, P., Velastin, S.A., Xu, L.Q.: Crowd analysis: A survey. Machine Vision and Applications **19**, 345–357 (2008)
92. Zhao, T., Nevatia, R., Wu, B.: Segmentation and tracking of multiple humans in crowded environments. Pattern Analysis and Machine Intelligence, IEEE Transactions on **30**(7), 1198–1211 (2008)
93. Zhou, B., Wang, X., Tang, X.: Random field topic model for semantic region analysis in crowded scenes from tracklets. In: IEEE Conference Computer Vision and Pattern Recognition (2011)
94. Zhu, X., Gong, S., Loy, C.C.: Comparing visual feature coding for learning disjoint camera dependencies. In: British Machine Vision Conference (2012)
95. Zivkovic, Z., van der Heijden, F.: Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognition Letters **27**(7), 773–780 (2006)