An Evaluation of Crowd Counting Methods, Features and Regression Models

David Ryan, Simon Denman, Sridha Sridharan, Clinton Fookes

Image and Video Laboratory, S1101, Queensland University of Technology,
2 George St, Brisbane, Australia, 4000.
Phone: +61 7 3138 9329

Abstract

Existing crowd counting algorithms rely on holistic, local or histogram based features to capture crowd properties. Regression is then employed to estimate the crowd size. Insufficient testing across multiple datasets has made it difficult to compare and contrast different methodologies. This paper presents an evaluation across multiple datasets to compare holistic, local and histogram based methods, and to compare various image features and regression models. A K-fold cross validation protocol is followed to evaluate the performance across five public datasets: UCSD, PETS 2009, Fudan, Mall and Grand Central datasets. Image features are categorised into five types: size, shape, edges, keypoints and textures. The regression models evaluated are: Gaussian process regression (GPR), linear regression, K nearest neighbours (KNN) and neural networks (NN). The results demonstrate that local features outperform equivalent holistic and histogram based features; optimal performance is observed using all image features except for textures; and that GPR outperforms linear, KNN and NN regression.

Keywords: crowd counting, holistic features, local features, histogram features, regression

1. Introduction

Crowd size estimation is an important task for both operational and security purposes. The distribution of people throughout a public space can be used to gather business intelligence, such as consumer shopping patterns, or to ensure that normal operating conditions are maintained. Overcrowding may be an indicator of congestion, delay or security-related abnormalities such as fighting and rioting.

As closed-circuit television (CCTV) becomes ubiquitous, it grows increasingly difficult for human operators to monitor all of the available data due to the sheer number of cameras installed. For example, there are estimated to be between 1.85 million [40] and 4.2 million [64] CCTV cameras installed in the United Kingdom alone. In most cases, security footage is used to investigate events after they occur, rather than to generate real-time alerts during an evolving situation.

In recent years, researchers have turned to computer vision based surveillance technologies to monitor crowds automatically from CCTV. Existing crowd counting algorithms are predominantly holistic in nature, employing machine learning techniques to perform regression between image features and crowd size [71, 24, 59, 65, 45, 53, 48, 83, 43, 8]. In recent years a number of local systems have also been proposed, although many of these algorithms are detection based and rely on assumptions about camera placement or visibility of human features such as head, face or body parts [51, 85, 15, 90]. Other local approaches divide an image into a number of subregions and perform counting locally [47, 5, 50, 13, 22, 75]. Histogram based approaches have also been proposed in which local information is accumulated into histogram bins and represented on a holistic level [48, 49].

Insufficient testing across multiple datasets has made it difficult to compare and contrast different methodologies. A comprehensive analysis across multiple datasets is required to compare local and holistic methods, and to compare various image features and regression models.

This paper uses a cross validation protocol to evaluate the performance of various methods, features and regression models across five public datasets. Image features are categorised into five types: size, shape, edges, keypoints and textures. The regression models evaluated are: Gaussian process regression (GPR), linear regression, K nearest neighbours (KNN) and neural networks (NN). The following methods are evaluated: holistic (in which features are extracted across an image and regression is performed globally); local (in which foreground segmentation is used to localise groups and to perform feature extraction and regression locally); and a histogram based approach [48].

Our experiments demonstrate that local features outperform equivalent holistic features and histogram based features; best performance is observed using all image features except for textures; and that Gaussian process regression outperforms linear, K-nearest neighbours and neural network regression.

The remainder of this paper is structured as follows: Section 2 presents the literature review; Section 3 introduces the benchmark datasets used in this evaluation; Section 4 describes the

☆This research was supported by an Australian Research Council (ARC) Linkage grant No: LP0990135.

Email addresses: david.ryan@qut.edu.au (David Ryan), s.denman@qut.edu.au (Simon Denman), s.sridharan@qut.edu.au (Sridha Sridharan), c.fookes@qut.edu.au (Clinton Fookes)

System Component	Parameters Evaluated
Counting method	Holistic
	Histograms (intermediate)
	Local
Image features	Size
	Shape
	Edges
	Keypoints
	Texture
Regression model	Gaussian process regression (GPR)
	Linear
	K-nearest neighbours (KNN)
	Neural network (NN)

Table 1: A taxonomy of crowd counting methods, image features and regression models used in this evaluation.

system design; Section 5 presents the experimental results of the evaluation; and Section 6 discusses the conclusions of this research.

2. Literature Review

Crowd counting algorithms are generally categorised into two groups: holistic and local. Holistic approaches use global image features to describe each frame in a video sequence, and a classifier or regression model is used to map between the feature space and the crowd size estimate. Local approaches, by contrast, utilise local image features to detect, track or count pedestrians within local regions of an image. In this case the crowd size is the sum of its parts. An intermediate approach has also been proposed [49, 48] which utilises blob size histograms based on local segments and expresses this information on a holistic level.

Section 2.1 describes the holistic approaches; Section 2.2 discusses the intermediate approach; and Section 2.3 describes local approaches. Table 1 presents a taxonomy of system components used in this evaluation and Table 2 summarises the regression based algorithms discussed in the following literature review.

2.1. Holistic Approaches

Holistic crowd counting algorithms use global image features to estimate the size of a crowd. They may also be described as "mapping-based" approaches because they map directly between the feature space and the crowd size estimate. Features used by these systems include textures [59], foreground pixels [24] and edge features [48], amongst others, while the classification and regression strategies have included linear regression [24], neural networks [59, 48] and Gaussian process regression [8].

Textural approaches are based on the notion that low density crowds exhibit course textures and high density crowds exhibit fine textures. Rather than estimate the number of people directly, these approaches classify the crowd density using a four or five point scale.

Marana [59, 57] proposed the use of grey level coocurrence matrix (GLCM) based statistics [41] for crowd density estimation. Marana also proposed the Minkowski fractal dimension [60]. Xiaohua [83] proposed the use of the 2D discrete wavelet transform (DWT) as a basis for extracting textural features, while Rahmalan [69] proposed Translation Invariant Orthonormal Chebyshev Moments (TIOCM). Rahmalan's evaluation observed superior performance of textural features on an afternoon dataset, "because the afternoon data has smaller variation of illumination when compared with morning data". When morning and afternoon datasets were combined to form a larger mixed set, performance decreased compared to the afternoon dataset alone due to these illumination changes over time. This highlights the principle limitation of textural features: they are sensitive to the scene background, and are thus impractical for real world use as they would need to be re-trained after any significant background change.

Other holistic crowd counting algorithms have utilised features such as foreground pixels and edges. While these features are located at points of interest they are aggregated on a holistic level. Regazzoni [71, 72] proposed a number of edge features, such as vertical edges, "for detecting the bodies (i.e. legs and arms)". More recently, a number of algorithms have attempted to segment the foreground using background modelling techniques. The rationale for this approach is described by Cho [16]:

> It is clear that a human observer has absolutely no problem in distinguishing a very dense crowd from the background. It is believed that human brain is well trained and would be likely to use the ratio of "crowd area" to "background area" as an estimate for the crowd density. This idea could be applied quantitatively to computer-based density estimation if the image-pixels corresponding to the crowd could be separated from those of the background.

Davies [24] found that the relationship between the number of foreground pixels and the number of people in the scene was approximately linear, as was the case for edge pixels. Cho [17, 16, 18] also used edge and foreground pixel counts and proposed a fast training algorithm for feedforward neural networks. Huang [45] calculated the percentage of foreground pixels in each sub-region of the image, and these values were used to populate a feature vector which served as inputs to a neural network for regression.

For the purposes of indoor crowd estimation over a short period of time, these approaches were shown to be successful. However, these approaches relied on a static background model, making the system sensitive to lighting changes over longer periods of time, whether sudden or gradual. Adaptive background models such as [79, 88, 89, 27, 26, 25] are robust against such changes, and have been adopted in more recent crowd counting applications.

		Image Features					
Method	**Reference**	**Size**	**Shape**	**Edges**	**Keypoints**	**Texture**	**Model**
Holistic	Regazzoni [71]			✔			EKF/BBN
	Davies [24]	✔		✔			Linear
	Marana [59, 58, 61, 57]					✔	NN
	Marana [60]					✔	NN
	Cho [17, 16, 18]	✔		✔			NN
	Paragios [65]	✔					Linear
	Huang [45]	✔					NN
	Ma [53]	✔					Linear
	Rahmalan [69]					✔	NN
	Xiaohua [83]					✔	SVM tree
	Hou [43, 44]	✔					NN
	Chan [8, 6, 10, 7]	✔	✔	✔	✔	✔	GPR
	Zhang [84]	✔	✔	✔	✔	✔	GPR / Ensemble (KNN+NN)
	Tan [80]	✔	✔	✔	✔	✔	Linear
Intermediate	Kong [49, 48]	✔	✔				NN/Linear
Local							
Motion regions	Conte [19, 20, 22, 21]				✔		ϵ-SVR
	Ryan [75, 76]	✔	✔	✔	✔		GPR/Linear
	Celik [5]	✔					Linear
	Kilambi [46, 47]	✔	✔				Linear (Cylinder model)
	Fehr [32]	✔	✔				Linear (Cylinder model)
Grid	Chen [13]	✔	✔	✔		✔	Linear
Pixelwise	Lempitsky [50]	✔	✔				Linear

Table 2: High level summary of regression based crowd counting systems. See the main text (Section 2) for a full description.

The analyses of Davies [24], Cho [17, 16, 18] and Huang [45] were based on scenes with a relatively high camera angle, in which the effects of perspective were not apparent. When perspective distortion is significant, the total number of foreground pixels is less likely to be a reliable indicator of crowding, because objects in the distance appear smaller and therefore contribute fewer pixels to the foreground mask.

Paragios [65] and Ma [53] introduced the use of quasi-calibration, obtained from the "relative size variation of the projection height and widths of a rigid object as the object translates in depth," [65] which is modelled as a linear function of the row and column coordinates. A 'density map' is calculated using the quasi-calibration, whereby a weight is assigned to each pixel to compensate for the effects of perspective. The weighted sum of pixels in the foreground mask was used to detect excessive crowding above a threshold in [53]. Hou [43, 44] utilised a similar approach, using a density map to accumulate a weighted foreground pixel count, and performing regression with a neural network to estimate the crowd size.

Chan [8, 6, 10, 7] proposed a holistic algorithm which extracted a very large number of features from each image in order to account for occlusion and other non-linearities such as segmentation errors. The segmentation is based on dynamic textures [9], yielding two boolean foreground masks: one for motion in each direction. Holistic image features included foreground area, perimeter pixel count, edge orientation histogram and textural features. In total, 30 features are extracted and Gaussian process regression (GPR) and Bayesian Poisson regression (BPR) was used to predict the number of pedestrians walking in each direction.

Dimensionality reduction techniques have been used by some authors. Zhang [84] proposes high dimensionality holistic features followed by dimensionality reduction using principal component analysis and kernel dimension reduction. Tan [80] automatically selects a subset of 129 holistic features and uses semi-supervised elastic net regression (sparse linear model) on this reduced feature set.

In summary, holistic approaches are based on the intuition that a global metric (crowd size) is best estimated from global image properties (holistic features). However, crowd size is difficult to monitor due to the high variation in crowd behaviours, distribution and density. Local and intermediate approaches seek to address this.

2.2. Intermediate Approaches

An intermediate approach was proposed by Kong [49, 48] in which blob size *histograms* to describe image features on a holistic level. The blob size histogram and edge orientation histogram were used to capture the range of object sizes and their appearance in a scene. With each pixel weighted by its value in a density map, the size of each blob is calculated and used to categorise the blob into a histogram bin (see Section 4.7). The blob size histogram serves to separate the blobs present in an image; it would be expected that noise contributes to the smallest histogram bin, while individual pedestrians and small groups contribute to successively larger bins, for example. The exact nature of the relationship is learned by the regression model, but the use of blob size histogram bins as image features should enable the system to distinguish between groups of people and individuals.

Kong also used the Canny edge detector [4] to extract edge pixels and their angle of orientation. These pixels are masked by the foreground so that edges in the background are ignored. An edge angle histogram is constructed with eight bins between 0° and 180°. The edge orientation histogram "can distinguish edges caused by pedestrians, which are usually vertical, with other scene structures such as noise, shadows and cars" [48]. There is support for this statement in other visual surveillance research. For example, Dalal [23] described the histogram of oriented gradients (HOG) for the explicit purpose of human detection (although their approach is block based and employs local normalisation).

The feature vector used by Kong to represent an image is the concatenation of the blob size histogram and the edge angle histogram (Section 4.7). Both linear regression and neural network regression were used to model the crowd size in their approach. We consider this method in our evaluation as an intermediate approach between holistic and local features.

2.3. Local Approaches

Local approaches to crowd counting utilise detectors or features which are specific to individuals or groups of people within an image. These groups are independently analysed, so that the total crowd estimate is the sum of its parts. Generally these methods can be categorised as follows:

1. **Detection based** approaches utilise head, face or human detectors, and/or segmentation algorithms to obtain the approximate location of each individual within the scene. Crowd counting is then performed directly as a subsequent step.
2. **Localisation based** methods divide an image into a number of subregions and then apply regression-based counting techniques locally.

In sparse crowds it is appropriate to use individual pedestrian detection [23, 34, 33]. These approaches are best suited for sparse environments in which the detected object is fully visible. As this paper is concerned with crowded and occluded environments, these methods are not discussed in detail here. A survey of existing pedestrian detection methods can be found in [31, 28].

An alternative to pedestrian detection is crowd segmentation. This approach attempts to explain the observed image features by estimating the approximate spatial arrangement of pedestrians in the scene. Zhao [86] suggested that this information can be inferred from the foreground mask, and proposed a method for human segmentation within a model-based Bayesian framework. The human 3D model consisted of four ellipsoids with adjustable parameters. The optimal solution is estimated using the RJMCMC algorithm to traverse the solution space non-exhaustively within regions of high probability. A weakness of this technique is the inability to perform real-time segmentation in crowded situations containing more than 10-15 occupants,

due to the high dimensionality of the solution space. The utility of various pose models is also questionable in larger crowds where such information is likely to be occluded from view.

RJMCMC has also been used by other authors to perform crowd segmentation. For example, Ge [35, 38, 39, 37] proposed an example-based approach, by constructing a mixture model of Bernoulli shapes to represent foreground humans from a training dataset; and extended this to a multi-camera framework in [38, 36].

Instead of using explicit shape models, Dong [29] utilised an example-based approach in which shape descriptors were used to represent blobs in compact form. A blob's approximate shape is encoded using Fourier descriptors, discarding high frequency coefficients as these contribute little to the overall blob shape. The training dataset contained groups of pedestrians arranged in various configurations, so that new data can be assessed by interpolation using K Nearest Neighbours (KNN) regression. Dong's approach was demonstrated on groups of size 1-6. This approach is limited by the amount of training data available and cannot scale to arbitrarily large crowds. As group size increases, it becomes increasingly difficult to obtain sufficient training data for all of the various pedestrian configurations, and the example-based approach becomes insufficient.

A number of other crowd segmentation approaches have been proposed, using for example 2D models and expectation maximisation [73, 52], however they are not designed to operate in arbitrarily large crowds. Other approaches have sought to use symmetry [14], or tracking [62, 68]. However as with the above techniques, these are best suited to sparsely populated environments.

Overhead cameras have also been proposed to simplify the counting problem [77, 78, 81], however as general purpose CCTV cameras are rarely installed in such a configuration these approach are unlikely to be applicable to the majority of existing installations.

Head detection has been proposed by a number of authors [85, 51, 66, 11, 12, 90]. These approaches are useful in crowds where the face of each individual is always visible to the camera, although they do not provide a general solution to the crowd counting problem.

The aforementioned approaches are **detection based** algorithms and are generally based on the assumption of low crowd density or specific camera placement. By contrast, **localisation based** strategies divide the image into a number of subregions and attempt to count groups within the crowd locally.

Conte [19, 20, 22, 21] proposed moving keypoint clustering to perform group localisation. In this approach, SURF [3] was used to detect keypoints within an image. These points are then masked by optical flow so that stationary points are ignored. The remaining moving points are clustered into groups using the K-means algorithm, from which localised group size estimation is performed. These approaches are limited to moving pedestrians because the keypoints are masked by the optical flow field.

Foreground detection has been used by a number of authors. Celik [5] proposed a blob based algorithm which does not require training. It assumes a direct linear relationship between the number of pixels within a blob segment and the number of people represented by that segment, in order to obtain an estimate for each group. Similarly, Kilambi [46, 47] and Fehr [32] modelled a group of pedestrians as an elliptical cylinder, assuming a constant spacing between people within the group. Tracking a large blob over several frames increases the robustness of the group size estimate. However, the application to complex crowds in which blobs regularly split and merge may be challenging. Ryan [75, 76] applied regression to each blob in an image to obtain a group count for each segment, so that the total crowd size is the sum of the group estimates. This approach extracts training data from each blob in the training dataset and uses these local annotations to train the regression model rather than the holistic count.

A number of authors have used a grid of subregions, whereby an image is divided into a number of smaller cells and analysed locally, or even on a pixelwise basis. These approaches have been used to detect local abnormalities with binary classifiers [82], to classify discrete density levels [30, 55, 56, 54] or to explicitly count crowds within in each cell [13, 50].

Chen [13] used local feature mining to count crowds directly. Features were extracted from equally sized cells in a rectangular grid. Multiple output ridge regression was used to capture both global and local trends in the image. Lempitsky [50] estimated the fractional crowd density at each *pixel*, so that integrating the density over any region would yield the number of people in that region. Each pixel was represented by a feature vector containing local foreground and gradient information. A linear model was used to obtain the fractional density at each pixel. The linear coefficients were selected based on the MESA distance to minimise the maximum error in any subarray of the training images.

In summary, local approaches subdivide the counting problem and perform detection or regression locally.

3. Benchmark Datasets

This section describes the benchmark datasets used in this evaluation. The majority of crowd counting evaluations have focused on single datasets such as UCSD and PETS 2009 [8, 19, 76] or private datasets [24, 59, 48]. The use of limited datasets can result in overfitting due to the lack of varying crowding conditions.

Five benchmark datasets were used to evaluate the performance of crowd counting algorithms and parameters in this study. These are summarised in Table 3 and Table 4.

The **PETS 2009** database was released prior to the Eleventh IEEE International Workshop on Performance Evaluation of Tracking and Surveillance [1] in order to test a multitude of visual surveillance tasks: object tracking, crowd counting and event recognition. Two sequences were designated for counting the number of people in the image, labelled 13-57 and 13-59, and a region of interest is specified for View 1. Additionally, sparse crowd sequences (12-34 and 12-43) and a very densely crowded sequence (14-06) were selected for this analysis. These sequences capture a good variation in crowd prop-

Dataset	Sequence	Test Set	Training Subset	Crowd Size
PETS 2009	12-34	0:794	20:40:780	2 to 8
	12-43	0:106	5:10:105	1 to 7
	13-57	0:220	5:10:215	5 to 34
	13-59	0:240	5:10:235	3 to 25
	14-06	0:200	5:10:195	0 to 42
Fudan	1	1:300	10:20:290	3 to 15
	2	1:300	10:20:290	2 to 15
	3	1:300	10:20:290	1 to 14
	4	1:300	10:20:290	2 to 11
	5	1:300	10:20:290	0 to 12
Grand Central	1	1000, 6000	1000, 6000	132 to 152
	2	11000, 16000	11000, 16000	151 to 160
	3	21000, 26000	21000, 26000	125 to 138
	4	31000, 36000	31000, 36000	141 to 176
	5	41000, 46000	41000, 46000	200 to 245
UCSD	1	1:400	10:20:390	12 to 27
	2	401:800	410:20:790	11 to 25
	3	801:1200	810:20:1190	11 to 40
	4	1201:1600	1210:20:1590	29 to 45
	5	1601:2000	1610:20:1990	17 to 31
Mall	1	1:400	20:40:380	13 to 50
	2	401:800	420:40:780	20 to 50
	3	801:1200	820:40:1180	20 to 53
	4	1201:1600	1220:40:1580	17 to 48
	5	1601:2000	1620:40:1980	20 to 48

Table 3: The benchmark datasets used to evaluate the proposed crowd counting algorithm. The total number of frames is listed, and a subset of these frames have been annotated at regular intervals with ground truth. (The frames of the UCSD and Mall datasets are 1-indexed, while the remaining datasets are 0-indexed. We retain the indexing used by the original authors.)

	PETS 2009	Fudan	Grand Central	UCSD	Mall
Length (frames)	1565	1500	46009	2000	2000
Frame Rate (fps)	~7	10	23	10	<2
Resolution	768×576	320×240	720×480	236×158	640×480
Colour	RGB	Grey	Grey	Grey	RGB
Location	Outdoor	Outdoor	Indoor	Outdoor	Indoor
Shadows	Yes	Yes	No	No	Yes
Reflections	No	No	Yes	No	Yes
Loitering	No	Yes	Yes	No	Yes
Crowd Size	0 to 42	0 to 15	125 to 245	13 to 53	11 to 45

Table 4: Summary of the various conditions in the benchmark datasets.

erties at different times. Annotations for these datasets were obtained from Milan [63].

The **Fudan** dataset was introduced by Tan [80] and contains five sequences each of length 300 frames. Holistic ground truth for each frame is provided with the dataset, and additional local annotations were added manually to train the system. These manual annotations were performed on frames 10:20:290 from each sequence, as indicated in Table 3.

The **Grand Central** dataset was introduced by Zhou [87] to model the collective behaviour of crowds. The footage is provided in greyscale captured from New York's Grand Central station. Due to the extremely large size of this crowd, annotation of every frame is not feasible, therefore a sparse subset of frames has been selected over a long period of time (33 minutes) and annotated individually[1].

The **UCSD** pedestrian database was introduced by Chan [8] and contains 2000 annotated frames of pedestrian traffic moving in two directions along a walkway.

The **Mall** pedestrian database was introduced by Chen [13] and contains 2000 annotated frames of pedestrian traffic moving and stopping inside a cluttered indoor shopping centre.

Collectively, these datasets feature a wide variety of environmental conditions and crowd configurations. Details on the resolution and frame rate of the datasets is provided in Table 4, and example images from each are shown in Figure 1.

A 5-fold cross validation procedure is used to evaluate crowd counting methodologies. The Fudan dataset lends itself to 5-fold cross validation as it is already divided into five sequences of length 300 frames. The UCSD and Mall datasets are each 2000 frames, which are divided into 5×400 frame sequences. The PETS 2009 dataset contains numerous sequences designed for various challenges (tracking, crowd counting and event detection). Five of these were selected, as described in Table 3. Finally, the Grand Central dataset contains extremely large crowds of up to 245 people. In order to capture different crowd properties over time, the frames are annotated at extremely sparse intervals and then divided into five subsets.

At each fold of the cross validation, one sequence is withheld for testing, while the remaining four sequences are used to train the system. From these four training sequences, a subset of frames is selected to train the system. The training subsets used for each sequence are shown in Table 3.

The predictive performance of a crowd counting system is evaluated using three criteria: mean absolute error (MAE), the mean square error (MSE) and mean relative error (MRE). These metrics are commonly used within the field for evaluating system performance [24, 48, 8].

According to Regazzoni [71]: "End users accept a mean error of 20% with respect to the real number of people present in a controlled area." This means that a system should achieve MRE $< 20\%$ to meet the minimum accuracy requirements of system operators.

[1]These annotations will be made publicly available to other researchers. Please contact the authors for a copy of this data.

4. System Design

In this paper we evaluate crowd counting algorithms under the following categories:

1. **Holistic.** In this approach, features are extracted across an entire region of interest (ROI) and regression is used to estimate the size of the crowd directly.
2. **Local.** In this approach, features are extracted from local segments in the image and regression is performed locally to estimate the number of people in each segment. The crowd size is a direct summation of these local estimates.
3. **Histograms.** In this approach, features are extracted from local segments and accumulated into a blob size histogram as proposed by Kong [48], and this is represented at a holistic level. This approach is considered as an intermediate approach between local and holistic methods.

Detection based approaches are omitted from this evaluation because our datasets include heavily occluded crowds in low resolution images.

Holistic crowd counting algorithms employ regression between global image features and crowd size. By contrast, local approaches divide the image into a set of smaller segments to which regression is applied locally. In this evaluation we use foreground segments as the basis for localisation [75, 76, 5, 47]. This approach is adopted because foreground segmentation localises relevant objects in the scene (groups of people) and does not generate an exceedingly large dataset as might be the case for a grid of cells or pixels [13, 82, 30, 50]. The same foreground detection algorithm is used for the holistic, local and histogram features so that all approaches use the same motion segmentation.

In general, features can be categorised under the following headings:

1. **Size** refers to the magnitude of any interesting segments extracted from an image which are deemed to be relevant, such as the foreground pixel count. (Section 4.1).
2. **Shape** pertains to the orientation and shape descriptors of these areas, segments or objects detected in an image. (Section 4.2).
3. **Edge** refers to the relative change in pixel intensities across an image, and this is typically measured by means of a binary edge detector. (Section 4.3).
4. **Keypoints** include any other points of interest, such as corners, that are detected in an image. (Section 4.4).
5. **Texture** refers to general descriptors of an image such as contrast and homogeneity. (Section 4.5).

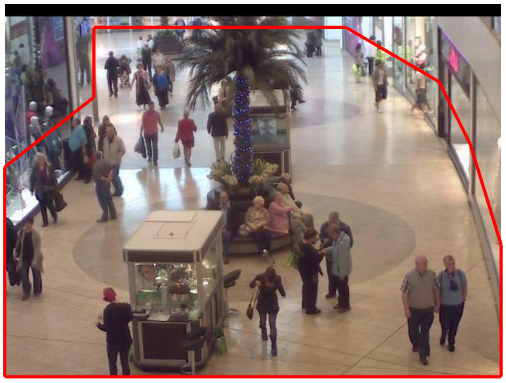
These features are discussed in subsequent sections (4.1-4.5) and summarised in Table 5. Section 4.6 discusses the regression models used in this evaluation. Section 4.7 describes the histogram features as proposed by Kong [48].

(a) PETS 2009 [1]

(b) Fudan [80]

(c) Grand Central [87]

(d) UCSD [8]



(e) Mall [13]

Figure 1: Images from each of the five benchmark datasets used in this crowd counting evaluation.

Category	Local Features	Holistic Features	Description
Size	A_n, L_n	A, L	Area and perimeter length
Shape	$V_n(0) \cdots V_n(3)$	$V(0) \cdots V(3)$	Perimeter orientation histogram
Edges	$H_n(0) \cdots H_n(5)$	$H(0) \cdots H(5)$	Edge orientation histogram
Keypoints	K_n^{SURF}, K_n^{FAST}	K^{SURF}, K^{FAST}	SURF and FAST keypoint features
Textures	$T_n^c, T_n^h, T_n^e, T_n^s$	T^c, T^h, T^e, T^s	Contrast, homogeneity, energy, entropy

Table 5: Feature categories used for crowd counting in this evaluation. The subscript n indicates the index of the blob under consideration, whereas the equivalent holistic feature is represented by omitting the subscript, as shown in Equation 4, for example.

4.1. Size

Size refers to the magnitude of any detected regions, such as motion segments, in an image. Davies [24] proposed the use of the foreground pixel count as a measure of the holistic crowd size, while Ma [53] introduced the density map S to weight each foreground pixel to compensate for perspective. We use the method described by Chan [8] which applies a weight $S(i, j)$ to each pixel (i, j) based on the relative sizes of reference objects in a scene.

The set of foreground pixels within the region of interest is denoted B, and the *weighted area* of the foreground is denoted A. This is calculated using the density map, S, as follows:

$$A = \sum_{(i,j) \in B} S(i, j) \tag{1}$$

This area directly captures the size of the foreground normalised for perspective. In practice, the presence of occlusions will lead to non-linearities in the relationship between crowd size and weighted foreground.

The equivalent *local* feature is extracted by segmenting the foreground into a set of connected components, which are individually labelled, and enumerated by n. The notation B_n is used to represent the set of pixels which belong to the nth blob. In set terminology, the collection of blobs $\{B_n\}$ is a *partition* of the set B. The weighted area of each blob, A_n, is:

$$A_n = \sum_{(i,j) \in B_n} S(i, j) \tag{2}$$

Note that $A = \sum_n A_n$ because the holistic foreground area is the sum of its segmented parts.

Another size feature is perimeter length. The set of perimeter pixels P_n is obtained by tracing along the boundary of the nth blob, and the set of all perimeter pixels in an image is denoted $P = \cup_n P_n$. Perimeter pixels are a one-dimensional feature, and are thus weighted using the square root of the density map S as in [48, 8]. The weighted perimeter of the nth blob segment is therefore:

$$L_n = \sum_{(i,j) \in P_n} \sqrt{S(i, j)} \tag{3}$$

And the equivalent holistic perimeter length is:

$$L = \sum_n L_n \tag{4}$$

The perimeter length supplements the area feature to provide a more complete description of crowd size.

4.2. Shape

Perimeter pixels provide valuable shape information about an object. Aside from the perimeter length, which measures the object *size*, the orientation of the perimeter pixels also contain important shape information. For example, Dong [29] encoded a blob's approximate shape using Fourier descriptors and Chan [8, 7] used a perimeter orientation histogram.

It is therefore intuitive and computationally efficient to use an orientation histogram with 4 bins, each corresponding to the direction of an adjacent pixel (0°, 45°, 90°, 135°). When tracing the perimeter from one boundary pixel to the next, the direction of movement determines which histogram bin receives the pixel's vote. The vote weight is the square root of the density map, $\sqrt{S(i, j)}$, as perimeter pixels are a one dimensional feature. Vertical edges in the absence of horizontal features are more likely to indicate individuals in a scene, whereas a combination of many perimeter pixels at all orientations may indicate larger crowds.

The value stored in each histogram bin h constitutes a feature, and the four shape features are denoted $V_n(h)$, for $h \in [0, 3]$. The equivalent holistic features are:

$$V(h) = \sum_n V_n(h) \tag{5}$$

This is simply the sum of the local perimeter orientation features, taken at a holistic level.

4.3. Edges

Edges have been commonly used in crowd counting systems. For example, Kong [48] introduced the use of an edge angle histogram on a holistic scale, while Davies [24], Chan [9] and many others have used the total number of edge pixels on a holistic level, regardless of orientation. The boolean edge detection at each pixel is denoted $D(i, j) \in \{0, 1\}$ with 1 denoting an edge.

In this evaluation, an edge orientation histogram is constructed for each foreground segment in an image using the following procedure. For the nth blob segment, a histogram of edge orientations H_n is constructed by allocating each edge pixel to a histogram channel, based on the pixel's unsigned orientation $\angle G(i, j)$. The orientation bins are evenly divided over the range $[0, 180°]$, and a total of 6 bins are used. Each edge

pixel within the blob contributes a weighted vote to a histogram bin, equal to $\sqrt{S(i,j)}$ to normalise for perspective. The value of the hth histogram bin is denoted $E_n(h)$, and the orientation angle for that bin is lower-bounded by θ_h:

$$E_n(h) = \sum_{(i,j)\in B_n} \begin{cases} \sqrt{S(i,j)} & \text{if } \theta_h \le \angle G(i,j) < \theta_{h+1} \\ & \text{and } D(i,j) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

The edge orientation histogram is used to help distinguish between humans and other structures in the scene [48]. Edges also help to identify occlusions when multiple pedestrians partially block one another from view. Although the blob's *size* features are reduced by occlusions, the edge features become stronger due to the overlapping body parts, differing skin tones and conflicting clothing.

At the holistic level, the edge orientation histogram is calculated as follows:

$$E(h) = \sum_n E_n(h) \quad (7)$$

Canny edge detection [4] is used due to its use of non-maximum suppression and hysteresis thresholding which results in a cleaner output.

4.4. Keypoints

Keypoints refer to specific pixels of interest, such as corners, which are detected in an image. Keypoints are useful for detecting salient points of interest in a scene, and these are often indicative of human crowding. For example, Conte [22] used speeded-up robust features (SURF) [3], to detect keypoints within an image. The number of moving keypoints was used to predict crowding. Similarly, Albiol [2] utilised Harris corners [42] to estimate crowd size on a holistic level.

Two types of feature detectors are considered for this evaluation. Firstly, corners are detected using the 'FAST' algorithm recently proposed by Rosten [74], and the set of keypoints detected within the foreground blob segment n is denoted κ_n^{FAST}. Secondly, SURF keypoints [3] are extracted and this set of keypoints is denoted κ_n^{SURF}.

The two keypoint features are then calculated as follows:

$$K_n^{FAST} = \sum_{(i,j)\in\kappa_n^{FAST}} \sqrt{S(i,j)} \quad (8)$$

$$K_n^{SURF} = \sum_{(i,j)\in\kappa_n^{SURF}} \sqrt{S(i,j)} \quad (9)$$

Note that the notation κ_n^{FAST} is used to refer to a *set* of keypoints, while K_n^{FAST} represents the scalar keypoint feature that is calculated from this set.

The keypoints are masked by the foreground detection result, so that keypoints belonging to background objects and surrounding structures are not included in the feature vector.

The equivalent holistic keypoint features are:

$$K^{FAST} = \sum_n K_n^{FAST} \quad (10)$$

$$K^{SURF} = \sum_n K_n^{SURF} \quad (11)$$

This is the sum of the local keypoint features.

4.5. Texture

Textural features have been used in the literature, such as GLCM based features [59] and others (Section 2.1). These features are primarily used for density *classification*, where crowd densities are measured using a four or five point scale. Less commonly, these features have also been used for direct counting via regression, e.g. Chan [8].

The GLCM is calculated for a given offset (δ_i, δ_j). Using the notation $(i', j') = (i + \delta_i, j + \delta_j)$, the GLCM is calculated across a region of interest R as follows:

$$G(r,c) = \sum_{(i,j)\in R\cap(i',j')\in R} \begin{cases} 1 & \text{if } I_q(i,j) = r \cap I_q(i',j') = c \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

where I_q represents the quantisation of image I to 8 grey levels. The symmetric GLCM is denoted $G_s = G + G^T$. The normalised GLCM represents a probability distribution of pixel cooccurrences:

$$f(r,c) = \frac{G_s(r,c)}{\sum_{r,c} G_s(r,c)} \quad (13)$$

Contrast, homogeneity, energy and entropy features are then calculated as follows:

$$T^c = \sum_{r,c} (r-c)^2 f(r,c) \quad (14)$$

$$T^h = \sum_{r,c} \frac{f(r,c)}{1+(r-c)^2} \quad (15)$$

$$T^e = \sum_{r,c} f(r,c)^2 \quad (16)$$

$$T^s = \sum_{r,c} -f(r,c)\log f(r,c) \quad (17)$$

In this evaluation $(\delta_i, \delta_j) = (1, 0)$ is used. Additional offsets did not confer an improvement in our experiments. The equivalent local features are calculated by first computing the GLCM for each blob n. This is done by substituting B_n for R in Equation 12 and then computing features locally $(T_n^c, T_n^h, T_n^e, T_n^s)$.

Image texture provides information about crowd density [59, 57], and this is predictive of crowd size on a holistic level because the region of interest R is fixed. However, for local segments, B_n, of variable size, texture is poorly correlated with crowd size (Figure 2). Nonetheless local textures are included in this evaluation for completeness.

4.6. Regression Models

Four types of regression models are evaluated in this research. For comparison we use Gaussian process regression (GPR) [76, 8], linear regression [75, 48], K-Nearest Neighbours (KNN) with $K = 1, 2, 4, 8, 16, 32$ [29], and a neural network (NN) [48, 44] with a Sigmoid activation function and one hidden input layer (containing 4, 8, 16 or 32 neurons). In total there are 12 regression models with these various parameters. The system is trained locally and holistically by annotating pedestrians as in [76].

4.7. Histogram Features

As an intermediate between local and holistic features, the histogram features proposed by Kong [49, 48] are also evaluated in this analysis. The blob size histogram and edge orientation histogram were used to capture the range of object sizes and their appearance in a scene. The size of the nth blob is denoted A_n as in Equation 2. The blob size histogram is then constructed as follows: the value in the kth histogram bin is denoted $H(k)$, and the blob size for that bin is lower-bounded by a_k, then:

$$H(k) = \sum_n \begin{cases} A_n & \text{if } a_k \leq A_n < a_{k+1} \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

That is, each histogram bin accumulates the weighted sum of pixels belonging to those blobs whose size falls within the predefined range established for that bin. Kong uses six histogram bins ($k \in [0, 5]$) of width $W = 500$, such that:

$$a_k = \begin{cases} Wk & \text{if } k < 6 \\ \infty & \text{if } k = 6 \end{cases} \tag{19}$$

The blob size histogram serves to separate the blobs present in an image and to place them into predefined categories. It would be expected that noise contributes to the smallest histogram bin, while individual pedestrians and small groups contribute to the second or third bins, for example. The relationship is learned by the regression model, but the use of blob size histogram bins as image features will enable it to distinguish between groups of people and individuals.

Kong's algorithm was implemented as faithfully as possible to [49, 48], however some assumptions were necessary. Although Kong used a bin width of $W = 500$ for the blob size histogram, this value is not be suitable for all datasets due to differences in image resolution and camera positioning. Instead, the bin width is set to roughly $\frac{2}{3}$ of the size of a person in the scene, so that smaller blobs (noise) are assigned to the first histogram bin and larger groups occupy the other bins. This provides good separation between different blob sizes, as is the intent of the algorithm.

Kong also used the edge orientation histogram (Section 4.3, Equation 7) with eight bins to "distinguish edges caused by pedestrians, which are usually vertical, with other scene structures such as noise, shadows and cars" [48]. These pixels are masked by the foreground so that those edges in the background are ignored.

The feature vector used by Kong to represent an image is the concatenation of the blob size histogram and the edge angle histogram. Both linear regression and neural network regression were used to model the crowd size in [48]. For completeness we also evaluate GPR and KNN regression (Section 4.6) on Kong's feature set.

5. Evaluation

This section presents the results of the evaluation: Section 5.1 compares various feature vectors for crowd counting and Section 5.2 compares a number of regression models. Section 5.3 then compares the performance of local, holistic and histogram based methodologies to one another.

5.1. Comparison of Features

This section compares the performance of various image features, and combinations thereof, for crowd counting. The features were discussed in Section 4 and summarised in Table 5. Gaussian process regression is selected as the regression model in this section because this provides the best predictive performance (Section 5.2).

The features are aggregated on either a holistic or local level. (Histogram features, as described in Section 4.7, are evaluated in section 5.3). Features are categorised into the following categories: size (S), shape (P), edges (E), keypoints (K) and texture (T). Multiple features are represented by letter combinations, such as 'EK', which denotes the concatenation of edge and keypoint features.

Various combinations of these features are assessed, as shown in Tables 6-8. Using the 5-fold cross validation procedure described in Section 3, error rates are calculated across all frames for each dataset, and reported in terms of MAE and MRE.

Table 6 summarises the results for local features. Average error rates are reported under their respective columns, as well as a *ranking* from 1 to 31 indicating the relative performance of each feature set. For example, when assessing local features on the Fudan dataset, the lowest MRE is observed when *size, shape, edges and keypoints* are used (SPEK), whereas the highest error rate is observed from texture features alone. These feature sets are ranked 1 and 31 respectively. In each column, the top three results (ranked 1 to 3) are indicated in bold.

The best performing feature sets for the UCSD dataset contain a combination of either three or four types of features. In general, it can be seen that performance improves on this dataset as more features are included; poor performance is particularly seen when only one feature type is used. Similarly for the PETS 2009 dataset, individual features (particularly shape and texture features taken alone) exhibit relatively poor performance, with the MRE falling above the 20% threshold of acceptability suggested by Regazzoni [71]. However, more features tend to perform better in general, and a mean relative error of 16% is observed for the best local feature sets, as indicated in bold.

The best performance is observed on the Fudan dataset when all features except textures are used, and the worst performance

	UCSD		PETS 2009		Fudan		Mall		Grand Central	
Local Features	**MAE**	**MRE**	**MAE**	**MRE**	**MAE**	**MRE**	**MAE**	**MRE**	**MAE**	**MRE**
S	2.07 (25)	8.65% (26)	2.18 (21)	21.97% (27)	1.11 (28)	18.48% (28)	2.96 (26)	9.85% (26)	8.49 (11)	5.60% (16)
P	3.13 (29)	12.76% (29)	3.43 (30)	26.89% (30)	1.24 (30)	20.88% (30)	3.91 (29)	12.64% (29)	15.31 (29)	10.18% (29)
E	1.58 (9)	7.02% (11)	1.95 (12)	17.27% (8)	0.97 (16)	16.17% (14)	2.52 (5)	8.26% (5)	13.99 (28)	9.18% (28)
K	2.10 (27)	8.76% (27)	2.01 (13)	17.89% (13)	1.04 (26)	17.54% (26)	**2.45 (2)**	**8.12% (3)**	11.22 (26)	6.68% (24)
T	7.21 (31)	29.66% (31)	8.66 (31)	90.14% (31)	2.05 (31)	33.80% (31)	7.68 (31)	26.24% (31)	68.04 (31)	43.26% (31)
SP	2.02 (24)	8.30% (21)	1.87 (6)	18.92% (16)	1.00 (24)	17.40% (25)	3.03 (27)	9.97% (27)	10.18 (19)	6.08% (20)
SE	1.55 (7)	6.81% (8)	2.05 (17)	18.97% (17)	0.97 (17)	15.92% (10)	2.61 (10)	8.43% (8)	9.57 (17)	5.89% (18)
SK	1.85 (17)	7.77% (18)	2.09 (19)	21.53% (24)	1.06 (27)	17.86% (27)	2.50 (4)	8.20% (4)	**5.77 (1)**	**3.77% (1)**
ST	2.02 (22)	8.42% (23)	2.36 (26)	21.95% (25)	0.93 (5)	**15.53% (2)**	2.91 (24)	9.63% (24)	9.25 (14)	5.36% (11)
PE	1.52 (4)	6.59% (5)	**1.77 (2)**	**15.58% (1)**	0.95 (12)	16.11% (11)	2.66 (13)	8.60% (13)	11.15 (24)	7.05% (26)
PK	1.89 (19)	7.84% (20)	1.88 (9)	17.52% (12)	0.93 (4)	15.76% (5)	2.58 (8)	8.48% (10)	**7.43 (3)**	4.92% (7)
PT	3.26 (30)	13.38% (30)	3.19 (29)	24.11% (29)	1.21 (29)	19.45% (29)	4.04 (30)	12.73% (30)	20.83 (30)	13.84% (30)
EK	1.53 (5)	6.72% (7)	2.09 (18)	17.45% (11)	0.95 (9)	15.79% (7)	**2.44 (1)**	**8.08% (1)**	12.10 (27)	7.45% (27)
ET	1.70 (16)	7.75% (16)	2.31 (25)	21.28% (23)	0.95 (10)	16.25% (15)	2.86 (23)	9.35% (23)	10.56 (22)	6.98% (25)
KT	2.25 (28)	9.11% (28)	2.67 (28)	22.02% (28)	**0.92 (2)**	15.79% (8)	2.53 (6)	8.45% (9)	**7.28 (2)**	**4.58% (3)**
SPE	**1.45 (1)**	**6.21% (1)**	**1.71 (1)**	**16.24% (3)**	0.96 (14)	16.17% (13)	2.69 (14)	8.74% (14)	9.15 (13)	5.48% (14)
SPK	1.89 (18)	7.76% (17)	1.88 (8)	19.15% (18)	0.93 (7)	15.88% (9)	2.61 (11)	8.53% (11)	7.79 (5)	4.85% (6)
SPT	2.08 (26)	8.44% (25)	1.84 (5)	16.91% (4)	0.99 (21)	17.09% (24)	3.17 (28)	10.36% (28)	8.36 (9)	4.97% (9)
SEK	1.58 (10)	6.87% (10)	2.02 (14)	19.27% (19)	0.95 (13)	15.76% (6)	**2.48 (3)**	**8.08% (2)**	7.80 (6)	4.82% (5)
SET	1.63 (13)	7.17% (13)	2.20 (22)	19.75% (20)	1.01 (25)	17.06% (22)	2.80 (20)	9.13% (20)	10.66 (23)	6.09% (21)
SKT	2.02 (23)	8.42% (24)	2.36 (27)	21.95% (26)	0.93 (6)	**15.53% (3)**	2.91 (25)	9.63% (25)	9.25 (15)	5.36% (12)
PEK	**1.51 (3)**	**6.48% (3)**	**1.78 (3)**	**15.70% (2)**	**0.93 (3)**	15.62% (4)	2.56 (7)	8.27% (6)	9.03 (12)	5.51% (15)
PET	1.55 (6)	6.68% (6)	2.09 (20)	18.72% (15)	0.98 (20)	16.84% (19)	2.83 (21)	9.22% (21)	10.53 (21)	6.66% (22)
PKT	2.02 (21)	8.39% (22)	2.02 (15)	17.27% (9)	0.96 (15)	16.45% (17)	2.84 (22)	9.25% (22)	10.47 (20)	5.91% (19)
EKT	1.67 (15)	7.52% (15)	2.25 (23)	20.47% (22)	0.94 (8)	16.27% (16)	2.63 (12)	8.57% (12)	11.17 (25)	6.68% (23)
SPEK	**1.46 (2)**	**6.23% (2)**	1.78 (4)	16.97% (5)	**0.92 (1)**	**15.51% (1)**	2.58 (9)	8.34% (7)	8.12 (8)	4.93% (8)
SPET	1.55 (8)	6.59% (4)	1.88 (7)	17.23% (6)	1.00 (22)	17.08% (23)	2.75 (17)	8.93% (17)	8.49 (10)	5.07% (10)
SPKT	1.92 (20)	7.84% (19)	1.92 (11)	17.45% (10)	0.98 (19)	16.95% (20)	2.78 (18)	9.06% (18)	9.30 (16)	5.38% (13)
SEKT	1.65 (14)	7.24% (14)	2.26 (24)	20.24% (21)	1.00 (23)	16.97% (21)	2.80 (19)	9.09% (19)	9.88 (18)	5.65% (17)
PEKT	1.62 (12)	7.03% (12)	2.05 (16)	18.15% (14)	0.95 (11)	16.15% (12)	2.71 (15)	8.81% (15)	7.91 (7)	4.72% (4)
SPEKT	1.60 (11)	6.82% (9)	1.90 (10)	17.25% (7)	0.97 (18)	16.74% (18)	2.72 (16)	8.89% (16)	7.49 (4)	**4.44% (2)**

Table 6: Comparison of **local features** on each dataset. Mean absolute error (MAE) and mean relative error (MRE) are reported, and the rank (1 to 31) is shown in parentheses. (S = Size, P = Shape, E = Edges, K = Keypoints, T = Texture.)

	UCSD		PETS 2009		Fudan		Mall	
Holistic Features	**MAE**	**MRE**	**MAE**	**MRE**	**MAE**	**MRE**	**MAE**	**MRE**
S	6.64 (25)	31.97% (27)	2.06 (4)	21.21% (9)	1.12 (29)	18.86% (29)	3.05 (12)	10.09% (12)
P	3.31 (24)	13.71% (24)	3.34 (24)	27.11% (20)	1.45 (31)	24.88% (31)	4.32 (15)	13.31% (15)
E	14.68 (31)	41.67% (31)	2.60 (15)	22.85% (14)	0.93 (15)	15.98% (17)	15.85 (26)	50.25% (27)
K	1.97 (17)	8.57% (15)	2.11 (5)	20.54% (8)	0.99 (21)	16.44% (20)	**2.80 (1)**	**8.80% (1)**
T	2.76 (22)	12.01% (22)	3.29 (23)	34.71% (24)	1.21 (30)	19.93% (30)	7.98 (16)	27.66% (16)
SP	7.86 (30)	34.43% (29)	**1.84 (2)**	18.76% (6)	1.08 (27)	18.05% (27)	9.64 (19)	30.34% (19)
SE	**1.60 (2)**	**7.18% (3)**	2.69 (19)	21.50% (10)	0.96 (19)	16.02% (18)	15.92 (27)	49.68% (26)
SK	1.73 (7)	7.75% (8)	2.21 (8)	21.92% (11)	1.01 (24)	16.81% (23)	**2.87 (3)**	9.33% (5)
ST	7.06 (26)	30.28% (25)	3.73 (27)	47.93% (29)	0.88 (4)	14.97% (6)	2.92 (7)	9.63% (9)
PE	1.65 (5)	7.28% (4)	2.68 (18)	18.95% (7)	1.01 (25)	17.00% (25)	2.89 (4)	**9.18% (3)**
PK	7.40 (28)	36.77% (30)	**1.88 (3)**	**17.72% (3)**	0.93 (16)	15.82% (15)	**2.81 (2)**	**8.85% (2)**
PT	2.89 (23)	12.86% (23)	3.98 (30)	40.50% (26)	1.10 (28)	18.68% (28)	3.40 (14)	10.69% (14)
EK	7.51 (29)	32.72% (28)	2.64 (16)	23.84% (16)	0.95 (18)	16.30% (19)	2.94 (10)	9.39% (8)
ET	1.90 (15)	8.61% (16)	2.41 (12)	27.02% (19)	0.90 (10)	15.55% (14)	21.07 (29)	71.08% (29)
KT	2.45 (21)	11.14% (21)	2.41 (11)	24.56% (18)	**0.82 (2)**	**13.90% (3)**	9.41 (18)	29.14% (17)
SPE	**1.58 (1)**	**6.95% (1)**	2.15 (7)	**17.59% (2)**	1.02 (26)	17.09% (26)	3.11 (13)	10.21% (13)
SPK	1.76 (8)	7.66% (7)	**1.83 (1)**	18.40% (5)	0.94 (17)	15.93% (16)	2.93 (9)	9.34% (6)
SPT	2.02 (18)	8.98% (18)	2.58 (14)	24.15% (17)	0.88 (6)	14.73% (4)	14.05 (23)	47.01% (22)
SEK	1.64 (4)	7.29% (5)	2.66 (17)	22.33% (13)	0.98 (20)	16.49% (21)	2.97 (11)	9.71% (11)
SET	1.86 (12)	8.44% (13)	3.78 (29)	44.89% (28)	0.88 (7)	14.84% (5)	20.52 (28)	68.65% (28)
SKT	7.06 (27)	30.28% (26)	3.73 (28)	47.93% (30)	0.88 (5)	14.97% (7)	2.92 (8)	9.63% (10)
PEK	1.71 (6)	7.55% (6)	2.12 (6)	**17.24% (1)**	1.00 (23)	16.83% (24)	2.92 (6)	9.30% (4)
PET	1.87 (13)	8.46% (14)	3.06 (21)	29.82% (22)	0.89 (8)	15.01% (8)	14.04 (22)	48.15% (23)
PKT	2.33 (20)	10.69% (20)	2.40 (10)	22.19% (12)	**0.81 (1)**	**13.71% (1)**	13.86 (21)	46.91% (21)
EKT	1.85 (10)	8.38% (11)	2.87 (20)	34.23% (23)	0.89 (9)	15.17% (9)	8.24 (17)	30.30% (18)
SPEK	**1.62 (3)**	**7.13% (2)**	2.28 (9)	18.15% (4)	1.00 (22)	16.80% (22)	2.89 (5)	9.38% (7)
SPET	1.86 (11)	8.27% (10)	4.40 (31)	48.44% (31)	0.92 (14)	15.37% (13)	27.26 (30)	89.34% (31)
SPKT	2.11 (19)	9.45% (19)	2.46 (13)	23.14% (15)	**0.82 (3)**	**13.75% (2)**	13.79 (20)	46.74% (20)
SEKT	1.77 (9)	8.00% (9)	3.66 (26)	44.73% (27)	0.91 (13)	15.27% (12)	14.41 (24)	49.04% (24)
PEKT	1.91 (16)	8.68% (17)	3.07 (22)	29.24% (21)	0.90 (12)	15.25% (11)	14.67 (25)	49.55% (25)
SPEKT	1.88 (14)	8.42% (12)	3.65 (25)	34.99% (25)	0.90 (11)	15.22% (10)	27.36 (31)	88.53% (30)

Table 7: Comparison of **holistic features** on each dataset. Mean absolute error (MAE) and mean relative error (MRE) are reported, and the rank (1 to 15) is shown in parentheses. (S = Size, P = Shape, E = Edges, K = Keypoints, T = Texture.)

is seen when individual feature types are used. This is consistent with the results for the UCSD and PETS 2009 datasets. Shape and texture features alone also perform poorly on the Mall dataset. Due to the reflective surfaces and complicated structure of this scene, foreground segmentation is relatively poor, resulting in substantial noise and unusually-shaped blobs. It is not surprising, therefore, than shape features perform relatively poorly under these conditions. Nonetheless, performance is still quite good in terms of MRE (less than 10%).

Finally, the Grand Central dataset confirms that single features perform poorly, with more stable performance obtained using combined feature sets. Although the MAE is quite high for this dataset, this is explained by the large crowd size which ranges from 125 to 245 people. The mean *relative* error is less than 5%, well within the threshold of acceptability.

In summary, the MRE is less than 20% for the most accurate local feature sets on each dataset. The UCSD, Mall and Grand Central Datasets achieve MRE $< 10\%$ while the PETS 2009 and Fudan datasets achieve MRE $< 20\%$.

Table 7 summarises the results for holistic features. Due to the small number of annotated frames in the Grand Central dataset, it was not possible to obtain a trained model for the holistic system, but the results for the other datasets are shown. Relatively poor performance is observed for features taken *individually*. Improved performance is seen when a combination of multiple feature types are used. For example, a combination of three features exhibits optimal results on the UCSD, PETS 2009 and Fudan datasets. The Mall dataset is an exception, where keypoints outperform other features such as size and shape due to the relatively noisy motion segmentation.

In order to identify dominant trends the data is pooled across all datasets as follows. Firstly, feature sets are ranked from 1 to 31 as shown in parentheses in Tables 6 and 7, and the *average rank* across all datasets is reported in Table 8 for each feature set. For example, shape alone (P) obtains an average ranking of 22.5 out of 31 (with the holistic approach), indicating a consistently poor performance for this feature across all datasets. By contrast, when *size, shape and keypoints* (SPK) are used on the holistic level, an average rank of 8.5 is observed in terms of MRE.

Average ranking across multiple datasets provides a clearer picture than any individual dataset taken alone. It becomes clear from Table 8 that when more local features are used (aside from texture), the average ranking across all datasets is improved. The best local feature vector is: *size, shape, edges, keypoints* (SPEK) with an average rank of 4.8 out of 31 in terms of MAE. This feature vector suffers a reduction in performance if any feature is omitted, although the omission of size does not make a very large difference (PEK ranks 5.6). This suggests that *size* may not be as crucial in achieving optimal performance as expected.

The size feature is a direct measure of foreground pixels within each blob segment, and foreground pixel counting has formed the basis of most traditional algorithms in the literature. These results suggest that foreground detection provides a suitable means for *segmenting* an image (for the purpose of localisation), but the actual *size* of these segments is not a critical

	Average Rank					
	Local Features			**Holistic Features**		
Features	**MAE**	**MSE**	**MRE**	**MAE**	**MSE**	**MRE**
S	22.2	20.8	24.6	17.5	17.0	19.3
P	29.4	29.4	29.4	23.5	25.0	22.5
E	14.0	14.2	13.2	21.8	21.3	22.3
K	18.8	19.8	18.6	11.0	11.0	11.0
T	31.0	31.0	31.0	22.8	23.3	23.0
SP	20.0	18.8	21.8	19.5	19.8	20.3
SE	13.6	13.2	12.2	16.8	16.5	14.3
SK	13.6	11.0	14.8	10.5	10.3	11.8
ST	18.2	18.0	17.0	16.0	13.8	17.3
PE	11.0	11.8	11.2	13.0	15.8	9.8
PK	8.6	12.0	10.8	12.3	13.3	12.5
PT	29.6	29.6	29.6	23.8	23.0	22.8
EK	12.0	14.2	10.6	18.3	17.8	17.8
ET	19.2	19.6	20.4	16.5	14.3	19.5
KT	13.2	13.6	15.2	13.0	12.5	14.8
SPE	8.6	8.0	9.0	11.8	12.0	10.5
SPK	9.8	9.6	12.2	**8.8**	**9.5**	**8.5**
SPT	17.8	18.4	18.0	15.3	15.5	15.3
SEK	9.2	10.2	8.4	13.0	12.5	12.5
SET	20.6	19.4	19.2	19.0	17.8	18.5
SKT	19.2	19.0	18.0	17.0	14.8	18.3
PEK	5.6	7.0	6.0	10.3	12.5	8.8
PET	17.6	16.0	16.6	16.0	16.8	16.8
PKT	18.6	20.4	17.8	13.0	13.8	13.5
EKT	16.6	16.0	17.6	14.0	12.5	15.3
SPEK	**4.8**	**4.2**	**4.6**	9.8	11.8	8.8
SPET	12.8	11.6	12.0	21.5	22.0	21.3
SPKT	16.8	17.2	16.0	13.8	13.3	14.0
SEKT	19.6	20.4	18.4	18.0	17.5	18.0
PEKT	12.2	11.0	11.4	18.8	19.0	18.5
SPEKT	11.8	10.6	10.4	20.3	20.8	19.3

Table 8: **Average rank** of feature vectors (when ranked from 1 to 31) across all datasets. Values shown are not actual error rates, but rather an average ranking. Refer to Tables 6-7 for detailed rankings.

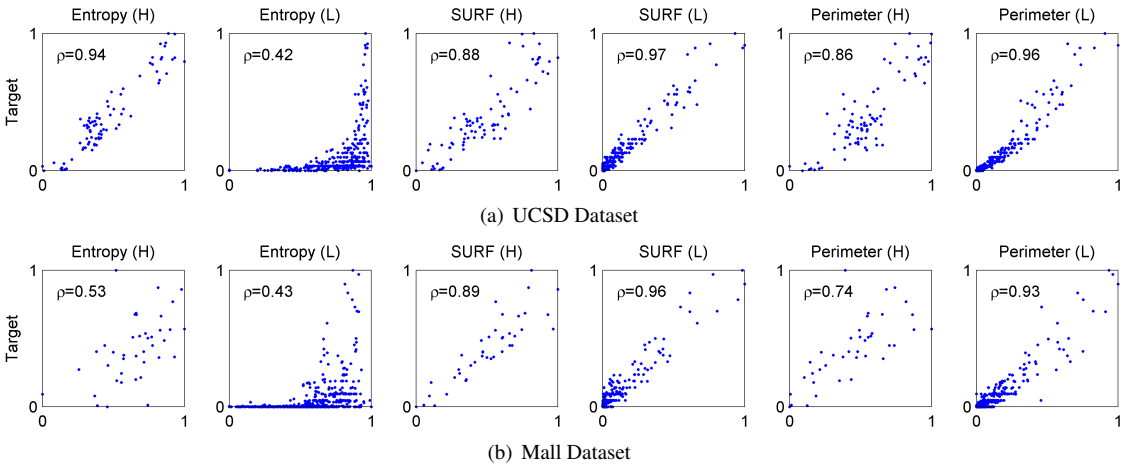


Figure 2: Relationship between selected holistic (H) and local (L) features and targets on the UCSD and Mall datasets (values are scaled). Pearson correlation coefficient ρ is shown. Correlation is higher for local features than for holistic features except in the case of texture (e.g. entropy). Texture is more informative for the UCSD dataset than the Mall dataset due to its relatively untextured background (Figure 1).

feature for crowd counting; in occluded and complicated crowd scenes, it is the presence of edges, keypoints and shape cues within those segments appear to be the most important features. Nevertheless, segment size does provide an intuitive measurement of the physical space occupied by a group, and it does provide a modest improvement in these experiments.

When holistic features are used, the best performance is observed for the feature set comprised of *size, shape, keypoints* (SPK). This feature vector does not include edges or textures, which have been used widely in the literature on a holistic level (Section 2).

Figure 2 plots the relationship between selected features and crowd size on the UCSD and Mall datasets. Due to the relatively untextured background of the UCSD dataset (Figure 1(d)), pedestrians introduce texture to the image, resulting in a relatively linear relationship between entropy and crowd size ($\rho = 0.94$, Figure 2(a)). By contrast, the Mall dataset features a highly textured background (Figure 1(e)): in this environment, pedestrians may either contribute or occlude texture, resulting in a poorer correlation ($\rho = 0.53$, Figure 2(b)).

These results lead to some interesting conclusions:

1. When local features are used, optimal performance is observed with size, shape, edge and keypoint features. However, *size* based features are not as critical for achieving optimal crowd counting results as expected. Optimal performance is observed with other local features such as *shape*, *keypoints* and *edges*. Note, however, that these features are still masked by the foreground detection result, and the segmentation of 'blobs' in the foreground is used as a basis for localisation. Therefore background modelling and foreground detection continues to play an important role in these experiments.
2. When holistic features are used, optimal performance is observed with size, shape and keypoint features. Texture and edge based features do not achieve the best results, despite their widespread usage. This is due to the inclusion of datasets with highly textured backgrounds such as PETS 2009 (Figure 1(a)) and the Mall dataset (Figure 1(e)).
3. Individual features perform relatively poorly compared to multiple feature combinations.

These conclusions can be used to inform the design and implementation of crowd counting systems in practice.

5.2. Comparison of Regression Models

This section compares the performance of various regression models for crowd counting. Holistic, local and histogram based algorithms are evaluated. Local features are evaluated using the feature vector: size, shape, edges, keypoints (SPEK). Holistic features are evaluated using size, shape and keypoints (SPK). These feature vectors were selected due to their optimal performance in Section 5.1. Histogram features are implemented as described by Kong [48] (see Section 4.7).

As in Section 5.1, 5-fold cross validation was used. For comparison we use Gaussian process regression (GPR), linear regression, K-Nearest Neighbours (KNN) with $K = 1, 2, 4, 8, 16, 32$, and a neural network (NN) with a Sigmoid activation function and one hidden input layer (containing 4, 8, 16 or 32 neurons). In total there are 12 regression models with various parameters. Note that in some cases training fails with the neural network model and large error values are reported in these instances.

Table 9 summarises the results of various regression models using local features. Average error rates are reported in terms of MAE and MRE, and regression models are ranked from 1 to 12 on each dataset, with 1 corresponding to the most accurate regression model and 12 the least accurate. The GPR and linear models provide most accurate performance on the UCSD dataset, with a MAE of 1.46 and 1.56 respectively. There is a significant reduction in performance for the third most accurate regression model (KNN with $K = 4$), for which a MAE of 2.72 was observed. Similarly on the PETS 2009 dataset, these regression models exhibited a MAE of 1.78, 1.77 and 3.00 respectively. The GPR and linear models rank in the top two

positions on the Fudan and Mall datasets, consistent with their performance on the UCSD and PETS 2009 datasets. The GPR model also ranks highest on the Grand Central dataset. These results provide strong support for the use of GPR and linear regression in conjunction with local features.

Table 10 summarises the results of various regression models using holistic features. As was observed with local features, the GPR and linear models ranked highest on the UCSD, PETS 2009 and Fudan datasets, with other regression models exhibiting a substantial reduction in performance by comparison. GPR also ranked highly on the Mall dataset.

Table 11 presents the results of various regression models using histogram features. Although suitable performance is observed with most regression models, the optimal model differs between datasets. In order to identify dominant trends, the data is pooled using the *average rank* across all datasets. Table 12 presents these results.

For histogram features, the best performance was seen with the linear model, which had an average rank of 3.0 out of 12 in terms of MAE. The neural networks also ranked very highly when 8 or 16 neurons were used in the hidden layer. These results confirm that linear regression and neural networks are the most appropriate regression models to be used in conjunction with Kong's histogram based feature set, as proposed by the author [48].

For local features, GPR outperforms the other models with an average ranking of 1.2 out of 12 in terms of MRE. Similarly for holistic features, GPR achieves an average rank of 1.25. These results provide very strong support for the use of Gaussian process regression in both local and holistic crowd counting systems, compared to linear, KNN or NN regression. Linear regression provides optimal performance for the histogram features proposed by Kong.

5.3. Comparison of Holistic, Local and Histogram Features

This section compares the performance of local, holistic and histogram features to one another. Local and holistic features are evaluated using GPR, while histogram features are evaluated using linear regression. These regression models were selected due to their optimal performance in Section 5.2.

Table 13 presents the performance of the holistic, local and histogram based approaches side-by-side. The following feature vectors were selected due to their optimal performance in Section 5.1:

- Size, Shape, Keypoints (SPK)
- Size, Shape, Edges, Keypoints (SPEK)

The first feature set is optimal for holistic systems, while the second is optimal for local systems (see Table 8). Histogram features are also presented in Table 13, and because these features are based on blob sizes and edge orientations, we also include the 'Size, Edges' feature vector for a similar comparison.

Each row in Table 13 lists the results for a given feature set, and the best result (holistic, local or histogram) is indicated in bold. For each dataset the best result across all system configurations is underlined.

For the Mall dataset, local features outperform holistic and histogram features in all experiments, regardless of the feature vector used. On the Fudan dataset, local features outperform holistic features except for when 'Size, Edges' are used; in that case, optimal performance is observed for holistic features (in terms of MAE). Regardless, the best performance on the Fudan dataset (across all system configurations) is observed for a combination of local features (size, shape, edges, keypoints).

The PETS 2009 dataset produces mixed results: when 'Size, Shape, Keypoints' are used, holistic features outperform local features. However, local features perform best with the other configurations. Similarly, results are mixed on the UCSD dataset, although optimal performance is observed with local features (size, shape, edges, keypoints).

In each dataset, the best performance is underlined, and the lowest error rates are observed with a local approach in each case. These results provide strong support for the use of local features rather than holistic or histogram features on these datasets.

Figure 3 shows two screenshots of the crowd counting algorithm operating using local features: the group estimate for each segment is rounded to the nearest integer and the total crowd estimate is shown at the top of the image. Figure 4 plots the estimate of the local, holistic and histogram based approaches against the ground truth for a number of sequences. These figures provide qualitative evidence for the algorithms evaluated in this paper across a wide range of conditions. The difference between these algorithms is most evident on sequence 13-57 of the PETS 2009 dataset, for which the local approach is most accurate.

5.4. Processing Speed

In this section we report the processing speed of the algorithms. For comparison, the datasets with the smallest and largest images were selected: the UCSD dataset has a resolution of 236×158 pixels, whereas the PETS 2009 dataset has a resolution of 768×576.

The baseline algorithm for this section is based on local features with Gaussian process regression (GPR) and a feature vector of 'Size, Shape, Edges, Keypoints' (SPEK). This configuration was selected due to its optimal performance in Sections 5.1–5.3. This algorithm operated at 20.2 fps and 2.7 fps on the UCSD and PETS 2009 datasets respectively.

Table 14 compares the processing speed with various feature vectors. When 'size' or 'shape' features are omitted from the feature vector (i.e. PEK, SEK), little change is observed in processing speed: the algorithm operates at 20.2 and 2.7–2.8 fps on the UCSD and PETS 2009 datasets. As local features require blob localisation, the calculation of size and shape features require little additional overhead. A small improvement in processing speed is observed by omitting 'edge' features: 21.7 and 3.0 fps respectively. The greatest improvement in processing speed occurs when 'keypoints' are omitted, as the SURF and FAST algorithms are skipped. In this case the speed increases to 27.8 and 4.3 fps.

	UCSD		PETS 2009		Fudan		Mall		Grand Central	
Regression Model	**MAE**	**MRE**	**MAE**	**MRE**	**MAE**	**MRE**	**MAE**	**MRE**	**MAE**	**MRE**
GPR	**1.46 (1)**	**6.23% (1)**	**1.78 (2)**	**16.97% (1)**	**0.92 (1)**	**15.51% (1)**	**2.58 (2)**	**8.34% (1)**	**8.12 (1)**	**4.93% (3)**
Linear	**1.56 (2)**	**6.48% (2)**	**1.77 (1)**	**17.75% (2)**	**0.94 (2)**	**16.02% (2)**	**2.58 (1)**	**8.52% (2)**	27.40 (8)	17.41% (8)
KNN (K=1)	2.89 (5)	10.75% (7)	3.02 (5)	19.79% (6)	1.16 (7)	19.27% (7)	3.45 (7)	11.22% (7)	9.85 (4)	6.31% (4)
KNN (K=2)	2.77 (4)	10.02% (4)	3.00 (4)	19.00% (4)	1.07 (5)	17.58% (6)	3.05 (5)	9.94% (5)	**8.15 (2)**	**4.53% (1)**
KNN (K=4)	**2.72 (3)**	**9.63% (3)**	**3.00 (3)**	**18.69% (3)**	**1.01 (3)**	16.47% (4)	**2.89 (3)**	9.28% (4)	**9.23 (3)**	**4.81% (2)**
KNN (K=8)	2.90 (6)	10.02% (5)	3.21 (6)	19.36% (5)	1.01 (4)	**16.22% (3)**	2.92 (4)	**9.20% (3)**	12.26 (5)	6.55% (6)
KNN (K=16)	3.12 (7)	10.56% (6)	3.53 (7)	20.69% (7)	1.13 (6)	16.92% (5)	3.25 (6)	9.96% (6)	12.61 (6)	6.35% (5)
KNN (K=32)	3.76 (8)	12.37% (8)	3.81 (8)	22.13% (8)	1.42 (10)	19.42% (8)	4.19 (8)	12.51% (8)	21.74 (7)	11.71% (7)
NN (4)	8.13 (10)	33.08% (10)	4.11 (9)	30.42% (9)	1.36 (9)	22.10% (10)	26.06 (12)	87.83% (12)	163.41 (10)	105.19% (10)
NN (8)	9.15 (11)	43.08% (11)	4.79 (10)	39.42% (10)	1.27 (8)	20.48% (9)	13.02 (10)	43.40% (10)	305.06 (12)	182.16% (12)
NN (16)	4.36 (9)	19.26% (9)	8.60 (11)	119.57% (11)	2.79 (11)	45.89% (11)	16.70 (11)	57.61% (11)	185.97 (11)	128.90% (11)
NN (32)	11.70 (12)	54.86% (12)	33.85 (12)	545.98% (12)	3.22 (12)	59.70% (12)	12.18 (9)	40.32% (9)	143.54 (9)	85.59% (9)

Table 9: Comparison of **regression models for local features** on each dataset. Mean absolute error (MAE) and mean relative error (MRE) are reported, and the rank (1 to 12) is shown in parentheses.

	UCSD		PETS 2009		Fudan		Mall	
Regression Model	**MAE**	**MRE**	**MAE**	**MRE**	**MAE**	**MRE**	**MAE**	**MRE**
GPR	**1.76 (1)**	**7.66% (1)**	**1.83 (1)**	**18.40% (1)**	**0.94 (1)**	**15.93% (1)**	**2.93 (2)**	**9.34% (2)**
Linear	**1.89 (3)**	**8.18% (3)**	**1.87 (2)**	**18.81% (2)**	**0.95 (2)**	**15.93% (2)**	**3.02 (3)**	10.08% (4)
KNN (K=1)	2.53 (6)	10.68% (7)	2.92 (6)	24.98% (4)	1.30 (9)	21.62% (8)	3.43 (7)	10.90% (7)
KNN (K=2)	2.35 (5)	9.77% (5)	2.82 (5)	**23.74% (3)**	1.16 (6)	19.98% (6)	3.17 (4)	**10.00% (3)**
KNN (K=4)	2.53 (7)	9.89% (6)	2.92 (7)	25.84% (6)	**1.06 (3)**	**18.28% (3)**	3.21 (5)	10.21% (5)
KNN (K=8)	3.02 (10)	11.06% (8)	3.16 (9)	28.98% (8)	1.12 (5)	19.97% (5)	3.44 (8)	10.99% (8)
KNN (K=16)	3.61 (11)	13.48% (11)	3.65 (10)	41.45% (11)	1.33 (10)	24.91% (11)	3.96 (11)	12.88% (11)
KNN (K=32)	4.58 (12)	18.07% (12)	4.72 (12)	60.14% (12)	1.75 (12)	33.78% (12)	4.93 (12)	16.36% (12)
NN (4)	2.09 (4)	9.38% (4)	2.98 (8)	35.87% (10)	1.10 (4)	19.38% (4)	3.31 (6)	10.64% (6)
NN (8)	**1.88 (2)**	**8.02% (2)**	**2.66 (3)**	27.21% (7)	1.29 (8)	21.81% (9)	**2.81 (1)**	**9.33% (1)**
NN (16)	2.70 (8)	11.67% (10)	3.98 (11)	32.43% (9)	1.27 (7)	20.58% (7)	3.60 (10)	11.92% (10)
NN (32)	2.74 (9)	11.58% (9)	2.77 (4)	25.03% (5)	1.47 (11)	22.56% (10)	3.56 (9)	11.64% (9)

Table 10: Comparison of **regression models for holistic features** on each dataset. Mean absolute error (MAE) and mean relative error (MRE) are reported, and the rank (1 to 12) is shown in parentheses.

	UCSD		PETS 2009		Fudan		Mall	
Regression Model	**MAE**	**MRE**	**MAE**	**MRE**	**MAE**	**MRE**	**MAE**	**MRE**
GPR	**1.56 (2)**	**6.94% (2)**	**2.51 (3)**	23.70% (5)	**1.03 (3)**	**17.21% (2)**	14.92 (12)	51.46% (12)
Linear	**1.53 (1)**	**6.82% (1)**	**2.38 (2)**	23.95% (6)	**1.02 (2)**	**17.44% (3)**	3.42 (7)	11.49% (7)
KNN (K=1)	2.64 (9)	11.35% (9)	2.97 (10)	23.68% (4)	1.42 (11)	23.28% (10)	3.74 (9)	11.82% (9)
KNN (K=2)	2.44 (6)	10.43% (7)	2.84 (9)	**23.64% (3)**	1.32 (9)	22.38% (9)	3.31 (5)	10.58% (4)
KNN (K=4)	2.44 (7)	10.27% (6)	2.60 (4)	**23.21% (2)**	1.23 (7)	20.50% (6)	3.22 (4)	**10.33% (3)**
KNN (K=8)	2.89 (10)	11.43% (10)	2.61 (5)	25.04% (7)	1.25 (8)	21.56% (8)	3.41 (6)	10.93% (6)
KNN (K=16)	3.54 (11)	13.62% (11)	2.67 (7)	28.16% (8)	1.35 (10)	24.38% (11)	3.87 (10)	12.59% (10)
KNN (K=32)	4.29 (12)	16.87% (12)	3.76 (12)	47.73% (12)	1.82 (12)	34.39% (12)	4.85 (11)	15.95% (11)
NN (4)	**1.79 (3)**	7.88% (4)	2.79 (8)	31.45% (9)	**1.00 (1)**	**16.92% (1)**	**3.19 (3)**	10.60% (5)
NN (8)	1.81 (4)	**7.73% (3)**	2.62 (6)	32.26% (10)	1.05 (4)	18.16% (4)	**3.17 (1)**	**10.08% (1)**
NN (16)	2.55 (8)	10.55% (8)	**2.35 (1)**	**20.92% (1)**	1.05 (5)	19.04% (5)	**3.19 (2)**	**10.25% (2)**
NN (32)	2.04 (5)	8.58% (5)	3.24 (11)	40.73% (11)	1.10 (6)	21.29% (7)	3.59 (8)	11.50% (8)

Table 11: Comparison of **regression models for histogram features** on each dataset. Mean absolute error (MAE) and mean relative error (MRE) are reported, and the rank (1 to 12) is shown in parentheses.

	Average Rank								
	Local Features			Holistic Features			Histogram Features		
Regression Model	**MAE**	**MSE**	**MRE**	**MAE**	**MSE**	**MRE**	**MAE**	**MSE**	**MRE**
GPR	**1.4**	**1.2**	**1.4**	**1.3**	**1.3**	**1.3**	5.0	5.3	5.3
Linear	2.8	2.8	3.2	2.5	2.3	2.8	**3.0**	**2.5**	4.3
KNN (K=1)	5.6	4.4	6.2	7.0	7.0	6.5	9.8	10.0	8.0
KNN (K=2)	4.0	4.0	4.0	5.0	5.3	4.3	7.3	7.5	5.8
KNN (K=4)	3.0	4.0	3.2	5.5	5.8	5.0	5.5	6.3	4.3
KNN (K=8)	5.0	5.2	4.4	8.0	8.0	7.3	7.3	8.3	7.8
KNN (K=16)	6.4	7.0	5.8	10.5	10.3	11.0	9.5	10.3	10.0
KNN (K=32)	8.2	9.0	7.8	12.0	12.0	12.0	11.8	11.8	11.8
NN (4)	10.0	10.0	10.2	5.5	4.3	6.0	3.8	2.8	4.8
NN (8)	10.2	9.4	10.4	3.5	4.8	4.8	3.8	**2.5**	4.5
NN (16)	10.6	10.2	10.6	9.0	9.5	9.0	4.0	4.8	**4.0**
NN (32)	10.8	10.8	10.8	8.3	7.8	8.3	7.5	6.3	7.8

Table 12: **Average rank** of regression models across all datasets. Values shown are not actual error rates, but rather an average ranking. (Note that the average rank for *holistic* and *histogram* features do not include the Grand Central dataset.)

		Local		Holistic		Histogram	
Dataset	**Features**	**MAE**	**MRE**	**MAE**	**MRE**	**MAE**	**MRE**
UCSD	Size, Edges	1.55	**6.81%**	1.60	7.18%	**1.53**	6.82%
	Size, Shape, Keypoints	1.89	7.76%	**1.76**	**7.66%**		
	Size, Shape, Edges, Keypoints	**1.46**	**6.23%**	1.62	7.13%		
PETS 2009	Size, Edges	**2.05**	**18.97%**	2.69	21.50%	2.38	23.95%
	Size, Shape, Keypoints	1.88	19.15%	**1.83**	**18.40%**		
	Size, Shape, Edges, Keypoints	**1.78**	**16.97%**	2.28	18.15%		
Fudan	Size, Edges	0.97	**15.92%**	**0.96**	16.02%	1.02	17.44%
	Size, Shape, Keypoints	**0.93**	**15.88%**	0.94	15.93%		
	Size, Shape, Edges, Keypoints	**0.92**	**15.51%**	1.00	16.80%		
Mall	Size, Edges	**2.61**	**8.43%**	15.92	49.68%	3.42	11.49%
	Size, Shape, Keypoints	**2.61**	**8.53%**	2.93	9.34%		
	Size, Shape, Edges, Keypoints	**2.58**	**8.34%**	2.89	9.38%		

Table 13: Comparison of local, holistic and histogram features. GPR was used for the local and holistic approaches, while linear regression was used for the histogram based approach. Each row represents a different feature set, and the best result (local, holistic or histogram) is indicated in bold, in terms of MAE and MRE. The best overall result for each dataset is underlined.

(a) Grand Central. (b) PETS 2009.

Figure 3: Visualisation of a local features based crowd counting system. Local counts are displayed on each blob, and the total count is shown at the top of the image.

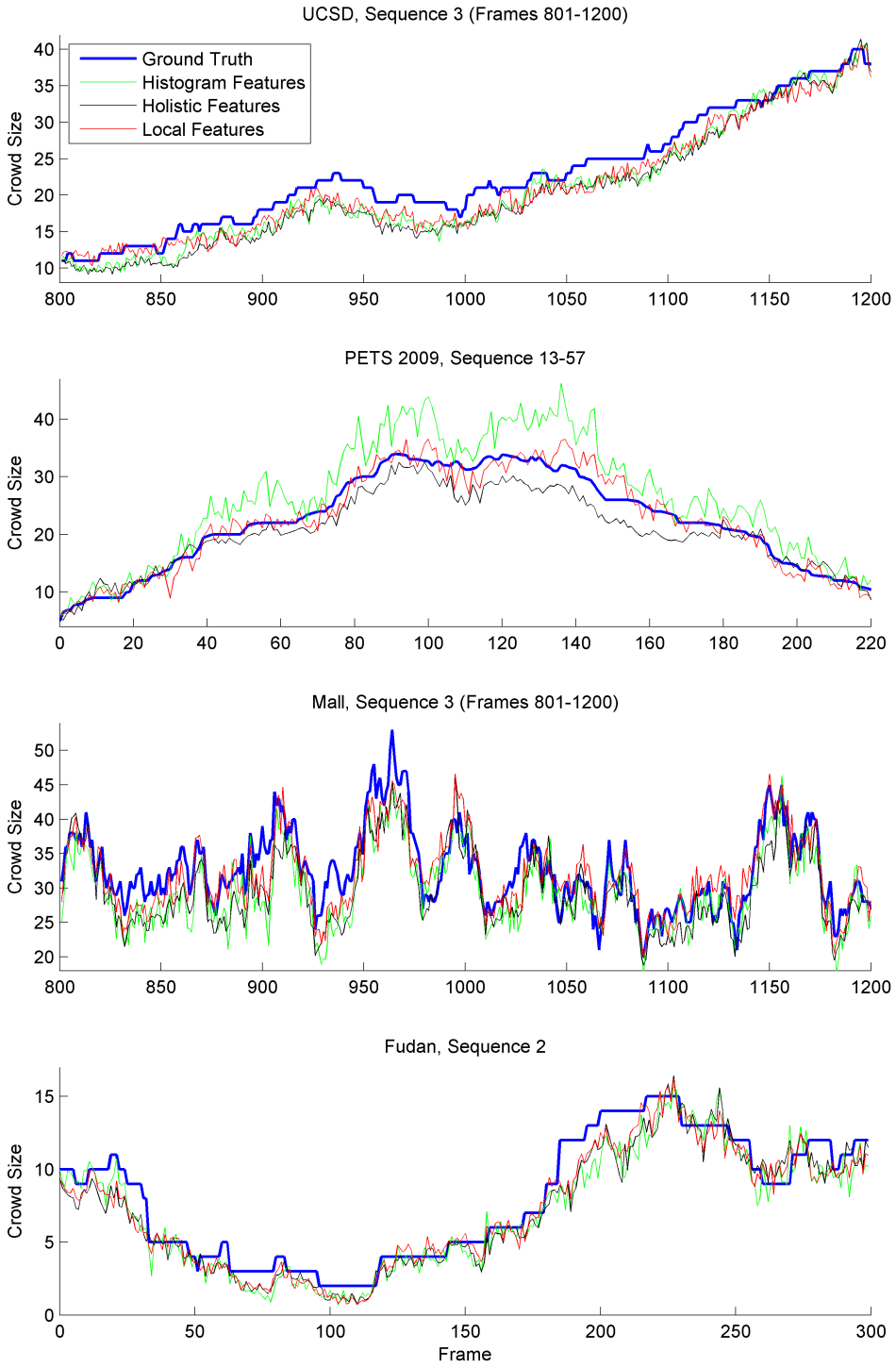


Figure 4: Ground truth compared to crowd size estimate for holistic, local and histogram based approaches on four sequences.

	Processing Speed (fps)	
Features	**UCSD**	**PETS 2009**
Size, Shape, Edges, Keypoints (SPEK)	20.2	2.7
Size, Edges, Keypoints (SEK)	20.2	2.7
Shape, Edges, Keypoints (PEK)	20.2	2.8
Size, Shape, Keypoints (SPK)	21.7	3.0
Size, Shape, Edges (SPE)	27.8	4.3

Table 14: Comparison of processing speed using different feature vectors. This evaluation was performed using local features with Gaussian process regression.

	Processing Speed (fps)	
Regression Model	**UCSD**	**PETS 2009**
GPR	20.2	2.7
Linear	22.5	2.8
KNN (K=4)	22.5	2.8
NN (8)	22.5	2.8

Table 15: Comparison of processing speed using various regression models. This evaluation was performed using local features, with the following feature vector: Size, Shape, Edges, Keypoints (SPEK).

Regression models are compared in Table 15. Similar processing speed is observed for the linear, K nearest neighbours and neural network regression models: 22.5 and 2.8 fps for the UCSD and PETS 2009 datasets respectively. Gaussian process regression is slightly slower (20.2 and 2.7 fps) due to the large matrix calculations required to calculate the mean of the predictive distribution [70].

Finally, Table 16 compares the local, holistic and histogram based methods. For this comparison, linear regression and a feature vector comprised of 'Size, Edges' was used to enable a direct comparison with the method of Kong [48]. No difference was observed between the local, holistic and histogram based methods in our implementation. However, it should be noted that our implementation has not been optimised for each of the methods individually, and additional improvements are likely to be obtained by doing so. For example, the framework calculates holistic features as the sum of local features (Equation 4); this method is useful for the current evaluation, in which local and holistic features are compared, however a holistic-only system would likely omit the blob localisation step entirely. For this reason an optimised holistic system would be expected to operate slightly faster than its local counterpart.

These algorithms were implemented in C++ and processed on a single CPU core. The main bottleneck is motion segmentation, which can be improved significantly using GPU acceleration as in [67] for example. Keypoint detection and Gaussian process regression also incur additional overhead, and improvements to these components may be observed with multithreading or GPU acceleration.

	Processing Speed (fps)	
Method	**UCSD**	**PETS 2009**
Local	32.3	4.5
Holistic	32.3	4.5
Histogram	32.3	4.5

Table 16: Comparison of processing speed using various counting methods. This evaluation was performed using linear regression and the 'Size, Edges' (SE) feature vector. This configuration was selected to enable a direct comparison with the intermediate method of [48] which uses blob size and edge histograms in conjunction with a linear regression model.

6. Conclusions

This paper evaluated feature types and regression models for crowd counting using local, holistic and histogram based approaches. Local features are specific to foreground segments in an image, and are used to estimate the size of each group. The local approach is annotated, trained and tested at a local level, whereas the holistic approach takes place across the entire image. The histogram approach accumulates information about local objects into histogram bins, and this information is represented at a holistic level. The following conclusions were reached as a result of this analysis:

- The use of local features consistently outperformed holistic features and histogram features (Section 5.3).
- For the local approach, a greater quantity of features generally improved performance compared to fewer features, with the exception of textures (Section 5.1). The best performance was observed with the feature vector: 'size, shape, edges, keypoints'. The omission of 'size' did not significantly reduce the overall performance (Table 8).
- For the holistic approach, edge and texture features did not provide optimal performance despite their widespread usage in the literature. Instead, best performance was seen with the feature vector: 'size, shape, keypoints' (Section 5.1).
- The use of Gaussian process regression consistently outperformed linear regression, K-nearest neighbours and neural networks, for both the local and holistic approach (Section 5.2). For the histogram based approach, the optimal regression models were linear regression and neural networks, consistent with the algorithm proposed by Kong [48].

Future research is warranted across a wider range of datasets to confirm these findings and to establish if additional feature sets or regression models can improve performance further. A comparison of existing motion segmentation algorithms [89, 25, 79] and localisation strategies [13, 75, 50] may provide additional insight into current crowd counting technology. The present data suggests that optimal performance is observed when GPR is employed with multiple local features.

References

[1] *PETS 2009, Eleventh IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*. 2009. http://www.cvg.rdg.ac.uk/PETS2009/.

[2] A. Albiol and J. Silla. Statistical video analysis for crowds counting. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 2569 –2572, November 2009.

[3] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, June 2008.

[4] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, November 1986.

[5] H. Celik, A. Hanjalic, and E.A. Hendriks. Towards a robust solution to people counting. In *Image Processing, 2006 IEEE International Conference on*, pages 2401 –2404, October 2006.

[6] A. B. Chan, M. Morrow, and N. Vasconcelos. Analysis of crowded scenes using holistic properties. In *Performance Evaluation of Tracking and Surveillance workshop at CVPR 2009*, page 101–108, Miami, Florida, 2009.

[7] A. B. Chan and N. Vasconcelos. Counting people with low-level features and bayesian regression. *Image Processing, IEEE Transactions on*, 21(4):2160 –2177, April 2012.

[8] A.B. Chan, Z.-S.J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –7, June 2008. Dataset: http://www.svcl.ucsd.edu/projects/peoplecnt/.

[9] A.B. Chan and N. Vasconcelos. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(5):909–926, May 2008.

[10] A.B. Chan and N. Vasconcelos. Bayesian poisson regression for crowd counting. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 545 –551, October 2009.

[11] Duan-Yu Chen and Kuan-Yi Lin. A novel viewer counter for digital billboards. In *Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP '09. Fifth International Conference on*, pages 653 –656, September 2009.

[12] Duan-Yu Chen, Chih-Wen Su, Yi-Chong Zeng, Shih-Wei Sun, Wei-Ru Lai, and Hong-Yuan Mark Liao. An online people counting system for electronic advertising machines. In *Proceedings of the 2009 IEEE international conference on Multimedia and Expo*, ICME'09, page 1262–1265, Piscataway, NJ, USA, 2009. IEEE Press.

[13] Ke Chen, Chen Change Loy, Shaogang Gong, and Tony Xiang. Feature mining for localised crowd counting. pages 21.1–21.11. British Machine Vision Association, 2012. Dataset: http://www.eecs.qmul.ac.uk/˜ccloy/downloads_mall_dataset.html.

[14] Li Chen, Ji Tao, Yap-Peng Tan, and Kap-Luk Chan. People counting using iterative mean-shift fitting with symmetry measure. In *Information, Communications Signal Processing, 2007 6th International Conference on*, pages 1 –4, December 2007.

[15] Tsong-Yi Chen, Chao-Ho Chen, Da-Jinn Wang, and Yi-Li Kuo. A people counting system based on face-detection. In *Genetic and Evolutionary Computing (ICGEC), 2010 Fourth International Conference on*, pages 699 –702, December 2010.

[16] Siu-Yeung Cho, T. W.S Chow, and Chi-Tat Leung. A neural-based crowd estimation by hybrid global learning algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 29(4):535–541, August 1999.

[17] Siu-Yeung Cho and Tommy W. S. Chow. A fast neural learning vision system for crowd estimation at underground stations platform. *Neural Process. Lett.*, 10(2):111–120, October 1999.

[18] T.W.S. Chow, J.Y.-F. Yam, and S.-Y Cho. Fast training algorithm for feedforward neural networks: application to crowd estimation at underground stations. *Artificial Intelligence in Engineering*, 13(3):301–307, July 1999.

[19] D. Conte, P. Foggia, G. Percannella, F. Tufano, and M. Vento. Counting moving people in videos by salient points detection. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1743 –1746, August 2010.

[20] D. Conte, P. Foggia, G. Percannella, F. Tufano, and M. Vento. A method for counting people in crowded scenes. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 225 –232, September 2010.

[21] D. Conte, P. Foggia, G. Percannella, and M. Vento. A method based on the indirect approach for counting people in crowded scenes. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 111 –118, September 2010.

[22] Donatello Conte, Pasquale Foggia, Gennaro Percannella, Francesco Tufano, and Mario Vento. A method for counting moving people in video surveillance videos. *EURASIP Journal on Advances in Signal Processing*, 2010(1):231240, June 2010.

[23] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886 –893 vol. 1, June 2005.

[24] A. C Davies, Jia Hong Yin, and S. A Velastin. Crowd monitoring using image processing. *Electronics & Communication Engineering Journal*, 7(1):37–47, February 1995.

[25] S. Denman, C. Fookes, and S. Sridharan. Improved simultaneous computation of motion detection and optical flow for object tracking. In *Digital Image Computing: Techniques and Applications, 2009. DICTA '09.*, pages 175 –182, December 2009.

[26] Simon Denman, Vinod Chandran, and Sridha Sridharan. An adaptive optical flow technique for person tracking systems. *Pattern Recognition Letters*, 28(10):1232–1239, July 2007.

[27] Simon Paul Denman. *Improved Detection and Tracking of Objects in Surveillance Video*. PhD thesis, 2009.

[28] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):743 –761, April 2012.

[29] Lan Dong, V. Parameswaran, V. Ramesh, and I. Zoghlami. Fast crowd segmentation using shape indexing. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1 –8, October 2007.

[30] Nan Dong, Fuqiang Liu, and Zhipeng Li. Crowd density estimation using sparse texture features. *Journal of Convergence Information Technology*, 5(6):125–137, August 2010.

[31] M. Enzweiler and D.M. Gavrila. Monocular pedestrian detection: Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2179–2195, 2009.

[32] D. Fehr, R. Sivalingam, V. Morellas, N. Papanikolopoulos, O. Lotfallah, and Youngchoon Park. Counting people in groups. In *Advanced Video and Signal Based Surveillance, 2009. AVSS '09. Sixth IEEE International Conference on*, pages 152 –157, September 2009.

[33] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, June 2008.

[34] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627 – 1645, September 2010. http://www.cs.brown.edu/˜pff/latent-release4/.

[35] W. Ge and R.T. Collins. Marked point processes for crowd counting. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2913 –2920, June 2009.

[36] Weina Ge and Robert T. Collins. Crowd detection with a multiview sampler. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, volume 6315, pages 324–337. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[37] Weina Ge, Robert T. Collins, and R. Barry Ruback. Vision-based analysis of small groups in pedestrian crowds. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(5):1003 –1016, May 2012.

[38] Weina Ge and R.T. Collins. Evaluation of sampling-based pedestrian detection for crowd counting. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, pages 1 –7, December 2009.

[39] Weina Ge and R.T. Collins. Crowd density analysis with marked point processes [applications corner]. *Signal Processing Magazine, IEEE*, 27(5):107 –123, September 2010.

[40] Graeme Gerrard and Richard Thompson. Two million cameras in the UK. *CCTVImage*, (42):10–12, 2011.

[41] R.M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786 – 804, May 1979.

[42] C. Harris. A combined corner and edge detector. *Proc. Alvey Vision Conf., 1988*, 1988.

[43] Ya-li Hou and G.K.H. Pang. Automated people counting at a mass site. In *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on*, pages 464 –469, September 2008.

[44] Ya-Li Hou and G.K.H. Pang. People counting and human detection in a challenging situation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 41(1):24 –33, January 2011.

[45] D. Huang, T.W.S. Chow, and W.N. Chau. Neural network based system for counting people. In *IECON 02 [Industrial Electronics Society, IEEE 2002 28th Annual Conference of the]*, volume 3, pages 2197 – 2201 vol.3, November 2002.

[46] P. Kilambi, O. Masoud, and N. Papanikolopoulos. Crowd analysis at mass transit sites. In *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pages 753 –758, September 2006.

[47] Prahlad Kilambi, Evan Ribnick, Ajay J. Joshi, Osama Masoud, and Nikolaos Papanikolopoulos. Estimating pedestrian counts in groups. *Computer Vision and Image Understanding*, 110(1):43–59, April 2008.

[48] D. Kong, D. Gray, and Hai Tao. A viewpoint invariant approach for crowd counting. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 1187 –1190, 2006.

[49] Dan Kong, Doug Gray, and Hai Tao. Counting pedestrians in crowds using viewpoint invariant training. In *Proc. British Machine Vision Conference (BMVC)*, 2005.

[50] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *Advances in Neural Information Processing Systems (NIPS)*, December 2010.

[51] Sheng-Fuu Lin, Jaw-Yeh Chen, and Hung-Xin Chao. Estimation of number of people in crowded scenes using perspective transformation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 31(6):645 –654, November 2001.

[52] X. Liu, P.H. Tu, J. Rittscher, A. Perera, and N. Krahnstoever. Detecting and counting people in surveillance applications. In *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*, pages 306 – 311, September 2005.

[53] R. Ma, L. Li, W. Huang, and Q. Tian. On pixel count based crowd density estimation for visual surveillance. In *Cybernetics and Intelligent Systems, 2004 IEEE Conference on*, volume 1, pages 170 – 173 vol.1, December 2004.

[54] Wenhua Ma, Lei Huang, and Changping Liu. Advanced local binary pattern descriptors for crowd estimation. In *Computational Intelligence and Industrial Application, 2008. PACIIA '08. Pacific-Asia Workshop on*, volume 2, pages 958 –962, December 2008.

[55] Wenhua Ma, Lei Huang, and Changping Liu. Crowd estimation using multi-scale local texture analysis and confidence-based soft classification. In *Intelligent Information Technology Application, 2008. IITA '08. Second International Symposium on*, volume 1, pages 142 –146, December 2008.

[56] Wenhua Ma, Lei Huang, and Changping Liu. Crowd density analysis using co-occurrence texture features. In *Computer Sciences and Convergence Information Technology (ICCIT), 2010 5th International Conference on*, pages 170 –175, December 2010.

[57] A. Marana, M. Cavenaghi, R. Ulson, F. Drumond, George Bebis, Richard Boyle, Darko Koracin, and Bahram Parvin. Real-time crowd density estimation using images. In *Advances in Visual Computing*, volume 3804 of *Lecture Notes in Computer Science*, pages 355–362. Springer Berlin / Heidelberg, 2005.

[58] A. N Marana, L. F Costa, R. A Lotufo, and S. A Velastin. On the efficacy of texture analysis for crowd monitoring. In *International Symposium on Computer Graphics, Image Processing, and Vision, 1998. Proceedings. SIBGRAPI '98*, pages 354–361. IEEE, October 1998.

[59] A. N Marana, S. A Velastin, L. F Costa, and R. A Lotufo. Estimation of crowd density using image processing. In *: 1997/074), IEE Colloquium on Image Processing for Security Applications (Digest No*, pages 11/1–11/8. IET, March 1997.

[60] A.N. Marana, L. Da Fontoura Costa, R.A. Lotufo, and S.A. Velastin. Estimating crowd density with minkowski fractal dimension. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 6, pages 3521 –3524 vol.6, March 1999.

[61] A.N. Marana, S.A. Velastin, L.F. Costa, and R.A. Lotufo. Automatic estimation of crowd density using texture. *Safety Science*, 28(3):165–175, April 1998.

[62] O. Masoud and N. P Papanikolopoulos. A novel method for tracking and counting pedestrians in real-time using a single camera. *IEEE Transactions on Vehicular Technology*, 50(5):1267–1278, September 2001.

[63] Anton Milan. *Data*. 2012. http://www.gris.informatik.tu-darmstadt.de/~aandriye/data.html.

[64] Clive Norris, Mike McCahill, and David Wood. The growth of CCTV: a global perspective on the international diffusion of video surveillance in publicly accessible space. *Surveillance & Society*, 2(2/3), September 2002.

[65] N. Paragios and V. Ramesh. A MRF-based approach for real-time subway monitoring. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001*, volume 1, pages I–1034– I–1040 vol.1. IEEE, 2001.

[66] M. Patzold, R.H. Evangelio, and T. Sikora. Counting people in crowded environments by fusion of shape and motion information. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 157 –164, September 2010.

[67] Vu Pham, Phong Vo, Vu Thanh Hung, and Le Hoai Bac. GPU implementation of extended gaussian mixture model for background subtraction. In *2010 IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*, pages 1 –4, November 2010.

[68] V. Rabaud and S. Belongie. Counting crowded moving objects. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 705 – 711, June 2006.

[69] H. Rahmalan, M.S. Nixon, and J.N. Carter. On crowd density estimation for surveillance. In *Crime and Security, 2006. The Institution of Engineering and Technology Conference on*, pages 540 –545, June 2006.

[70] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

[71] C. S Regazzoni, A. Tesei, and V. Murino. A real-time vision system for crowding monitoring. In , *International Conference on Industrial Electronics, Control, and Instrumentation, 1993. Proceedings of the IECON '93*, pages 1860–1864 vol.3. IEEE, November 1993.

[72] C.S. Regazzoni, A. Tesei, and G. Vernazza. A bayesian network for automatic visual crowding estimation in underground stations. In *Image Technology: Advances in Image Processing, Multimedia and Machine Vision*, pages 203–230. Springer, 1996.

[73] J. Rittscher, P.H. Tu, and N. Krahnstoever. Simultaneous estimation of segmentation and shape. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 486 – 493 vol. 2, June 2005.

[74] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):105 –119, January 2010.

[75] 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.*, pages 81 –88, December 2009.

[76] David Ryan, Simon Denman, Sridha Sridharan, and Clinton Fookes. Scene invariant crowd counting and crowd occupancy analysis. In *Video Analytics for Business Intelligence*, pages 161–198. Springer-Verlag, 2012.

[77] A.J. Schofield, P.A. Mehta, and T.J. Stonham. A system for counting people in video images using neural networks to identify the background scene. *Pattern Recognition*, 29(8):1421–1428, August 1996.

[78] A.J. Schofield, T.J. Stonham, and P.A. Mehta. Automated people counting to aid lift control. *Automation in Construction*, 6(5–6):437–445, September 1997.

[79] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages 2 vol. (xxiii+637+663), 1999.

[80] Ben Tan, Junping Zhang, and Liang Wang. Semi-supervised elastic net for pedestrian counting. *Pattern Recognition*, 44(10–11):2297–2304, October 2011. Dataset: http://www.iipl.fudan.edu.cn/~zhangjp/Dataset/fd_pede_dataset_intro.htm.

[81] T. Teixeira and A. Savvides. Lightweight people counting and localizing in indoor spaces using camera sensor nodes. In *Distributed Smart Cameras, 2007. ICDSC '07. First ACM/IEEE International Conference on*, pages 36 –43, September 2007.

[82] Xinyu Wu, Guoyuan Liang, Ka Keung Lee, and Yangsheng Xu. Crowd density estimation using texture analysis and learning. In *Robotics and Biomimetics, 2006. ROBIO '06. IEEE International Conference on*, pages 214 –219, December 2006.

[83] Li Xiaohua, Shen Lansun, and Li Huanqin. Estimation of crowd density based on wavelet and support vector machine. *Transactions of the Institute of Measurement and Control*, 28(3):299–308, 2006.

[84] Junping Zhang, Ben Tan, Fei Sha, and Li He. Predicting pedestrian counts in crowded scenes with rich and high-dimensional features. *Intelligent Transportation Systems, IEEE Transactions on*, 12(4):1037 –1046, December 2011.

[85] Xiaowei Zhang and G. Sexton. Automatic human head location for pedestrian counting. In *Image Processing and Its Applications, 1997., Sixth International Conference on*, volume 2, pages 535 –540 vol.2, July 1997.

[86] Tao Zhao and R. Nevatia. Bayesian human segmentation in crowded situations. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II – 459–66 vol.2, June 2003.

[87] Bolei Zhou, Xiaogang Wang, and Xiaoou Tang. Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2871 –2878, June 2012. Dataset: http://www.ee.cuhk.edu.hk/˜xgwang/grandcentral.html.

[88] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2*, page 28–31, Washington, DC, USA, 2004. IEEE Computer Society.

[89] Zoran Zivkovic and Ferdinand van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773 – 780, 2006.

[90] Keju Zu, Fuqiang Liu, and Zhipeng Li. Counting pedestrian in crowded subway scene. In *Image and Signal Processing, 2009. CISP '09. 2nd International Congress on*, pages 1 –4, October 2009.