People Counting in High Density Crowds from Still Images

Ankan Bansal*, K. S. Venkatesh
Department of Electrical Engineering, Indian Institute of Technology, Kanpur, India.

* Corresponding author. Email: ankan@iitk.ac.in
Manuscript submitted July 25, 2015; accepted October 16, 2015.
doi: 10.17706/ijcee.2015.7.5.316-324

Abstract: We present a method of estimating the number of people in high density crowds (hundreds to thousands of individuals) from still images. Unlike most existing works our method uses only still images to estimate the count. At this scale, we cannot rely on just one set of features for count estimation. We, therefore, use a fusion of multiple sources, *viz*. interest points (SIFT), Fourier analysis, wavelet decomposition, GLCM features and head detections, to estimate the counts. Each of these sources gives a separate estimate of the count along with confidences and other statistical measures which are then combined to obtain the final estimate. We tested our method on an existing dataset of fifty images containing over 64000 individuals. Further, we added another fifty annotated images of crowds and tested on the complete dataset of hundred images containing over 87000 individuals.

Key words: People counting, extremely dense crowd, texture analysis.

1. Introduction

Crowd counting has several real-world applications: crowd management, safety control and urban planning, monitoring crowds for surveillance, modeling crowds for animation and crowd simulation. Crowd size may also be an indicator of comfort level in public spaces or of an imminent stampede.

Many of the proposed automated systems for density and count estimation suffer from some limitations: inability to handle large crowds (thousands of people); reliance on temporal constraints; reliance on detecting, tracking and analysing individual persons in crowds. Another important limitation that some of these methods suffer from is the requirement of installed infrastructure on the site.

Most existing people counting methods can be divided into three categories: (1) pixel-based analysis; (2) texture-based analysis; and (3) object-level analysis.

Pixel based methods employ very local features such as edge information or individual pixel analysis to obtain counts [1]-[8]. These methods focus on estimating the density rather than count. Texture based methods rely on texture modelling through the analysis of image patches [1], [5], [7], [9]-[15]. Some texture analysis methods suggested in literature include grey-level co-occurrence matrix, Fourier analysis and fractal dimension. Object level methods try to locate individual persons in a scene [6], [8], [15]-[22]. But these methods work best only for very low density crowds. For high density crowds the evidence for the presence of a single person is scarce. Even for low density crowds, partial occlusions, perspective effects, cluttered background etc. negatively impact the performance of object-level methods.

Brostow and Cipolla [16] and Rabaud and Belongie [19] count moving people in videos by estimating contiguous regions of coherent motion. Addressing concerns about preservation of privacy in tracking people for counting, Chen *et al.* propose a texture based method for counting [1].

Some works estimate the relationship between low-level features and the density or count by training regression models. Some of these methods are global, which learn a single regression function for the entire image/video [1], [2], [4], [23]. But these methods make an implicit assumption that the density is the same over the image, which is not valid for most images. Some regression methods can be local which divide the image into cells and perform regression for each cell [11], [3], [15]. These methods deal with the problems mentioned above efficiently. An alternate multi-output regression model was proposed by Chen *et al.* [5].

The aim of this paper is to develop an effective texture-based method to solve the problem of counting the number of people in extremely dense crowds. The method should work well for dense crowds but should be robust to variations in density. For very dense crowds, a single feature or detection method alone cannot provide an accurate count due to low resolutions, occlusions, foreshortening and perspective. We build upon the work of Idrees *et al.* [15] and propose a model that combines sources of complementary information extracted from the images.

Appearance based features like SIFT descriptors are also useful to estimate the texture elements. SIFT features have been shown to be successful for crowd detection [9]. Idrees *et al.* also used SIFT as one of the features for estimating crowd counts.

The main contribution of this work is the use of multiple texture analysis sources to estimate the counts for dense crowds. We employ Fourier analysis, GLCM features and wavelet transform to analyse the texture information. Wavelet features have not been used for crowd counting or density estimation before. Along with these, we use head-detections and SIFT descriptors for our framework.

Most existing methods have been tested on low to medium density crowds, e.g., UCSD dataset [1] (11-46 people per frame), Mall dataset [5] (13-53 people per frame) and PETS dataset [24] (3-40 people per frame). In contrast, we show the performance of our method on the UCF crowd counting dataset [15] of 50 images containing between 96 and 4633 people per image. Further, we complement the dataset with 50 more images to expand it to 100 images and demonstrate the robustness and accuracy of our method on this new combined dataset.

2. Methodology

Fig. 1 is an overview of our framework. We divide the image into small cells and obtain the estimate from each cell to counter the variations in density of the crowd. The final output is the sum of all cell counts.

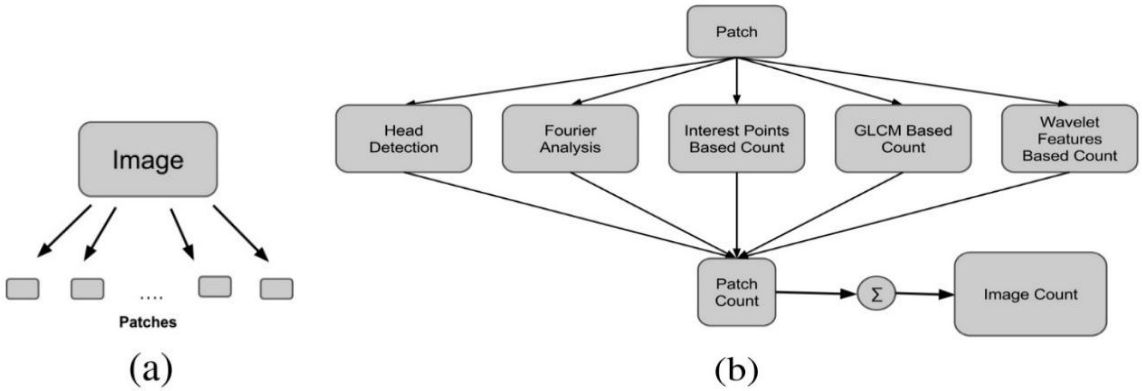

Fig. 1. A flow chart illustrating the methodology adopted in this paper. (a) The image is first divided into small cells in a grid. (b) Count is estimated for each cell by fusing estimates from five different methods.

2.1. Counting in a Cell

For a given cell *P*, we estimate the counts and confidences from five different sources. These are

combined to obtain a final estimate of the person count for that cell.

2.1.1. Interest-points based count

Arandjelovic [9] used a statistical model based on quantized SIFT features to segment an image into crowd and non-crowd regions. Subsequently, Idrees *et al.* [15] used interest points to estimate counts and to get a confidence score of whether the cell represents a crowd. We follow this idea to calculate the counts. Given a training set, we obtain SIFT features and cluster then into a codebook of size *K*. We use sparse SIFT features to train a Support Vector Regression (SVR) model using the counts at each patch from ground truth and use the trained model to obtain counts for new images patches. We calculate the SIFT features using the VL-FEAT library [25].

Due to the sparse nature of SIFT features, the probability of observing k_i instances of the i-th SIFT word can be modeled as a Poisson distribution. Suppose, for a cell containing crowd, the expected number of detections of the i-th SIFT word is λ_i^+. Then, the probability of observing k_i instances of the i-th SIFT word in a cell containing a crowd can be written as $p(k_i|crowd) = e^{-\lambda_i^+}[\lambda_i^+]^{k_i}/k_i!$. We can write a similar expression for a non-crowd cell (λ_i^-). Assuming independence between counts of any two SIFT words in a cell, the log of the likelihood ratio of crowd and non-crowd patches is: $\mu = \log p(k_1, k_2, \dots, k_K|\text{crowd}) - \log p(k_1, k_2, \dots, k_K|\neg\text{crowd}) = \sum_{i=1}^{K}[\lambda_i^- - \lambda_i^+ + k_i(\log\lambda_i^+ - \log\lambda_i^-)]$, where μ can be interpreted as the confidence about the presence of crowd in an image cell.

2.1.2. Counts from texture analysis methods

We employ three different texture analysis methods which separately give an estimate of the count which will be used later to give a final estimate of the count in the cell.

Fourier Analysis: Fourier analysis can be used to capture the repetitions in crowds. Since we are dealing with small cells and not the complete image, we can safely assume that the crowd density in a cell is uniform. In this case, the Fourier transform, $f(\omega)$, will show the repeated occurrence of people as peaks.

For a given cell, P, in an image, we calculate the gradient, $\nabla(P)$, and apply a low pass filter to remove high frequency components. Then we apply inverse Fourier transform to obtain the reconstructed image patch, P_r. The local maxima in the reconstructed image give an estimate of the total person count in that cell. We also calculate the several other statistical measures, such as entropy, mean, variance, skewness and kurtosis for both P_r and the difference $|\nabla(P) - P_r|$. We use the count and these measures as input for the next step (Section 2.2).

GLCM Features: Many people have used GLCM features for density/count estimation [11-[13], [26]. We adapt similar features to estimate the number of people. We quantize the image and calculate the joint-conditional probability density function, $f(i, j|d, \theta)$, with distance, d = 1, and angles, $\theta \in \{0°, 45°, 90°, 135°\}$. We calculate the following texture features:

Dissimilarity: $D(d,\theta) = \sum_{i,j} f(i,j|d,\theta)|i-j|$; Homogeneity: $H(d,\theta) = \sum_{i,j} f(i,j|d,\theta)/\left[1+(i-j)^2\right]$;

Energy: $E(d, \theta) = \sum_{i,j} f(i,j|d,\theta)^2$; Entropy: $P(d, \theta) = -\sum_{i,j} f(i,j|d,\theta)\log f(i,j|d,\theta)$.

So we obtain 16 (four for each θ) features for each image cell. We then train a support vector regression model using these features and ground truths from cells from the training images. We pass the count estimate and statistical measures such as variance, skewness and kurtosis of the GLCM matrices as features to the next stage.

Wavelet Decomposition: For each cell, P, we calculate the three-step pyramid-structured wavelet transform to obtain the 10 lower resolution sub-images, then calculate the energies in each of them as: $e = \frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N}|I(i,j)|$, where I is a sub-image with resolution $M\times N$. Thus, we obtain a ten-dimensional feature vector for each image cell. Texture energies are distributed differently for different texture patterns. Thus the energy features calculated above can be used for discriminating crowds and estimating crowd

counts. We train an SVR with these feature vectors using ground truth counts from the training data as outputs. We also calculate statistical measures such as variance, skewness and kurtosis of the 10 lower resolution sub-images and pass these measures along with the count to the next step.

2.1.3. Count from head detection

Detecting humans is not possible in dense crowds due to severe occlusions. Only the heads may be visible at this scale. So we estimate the count by detecting heads in the image. We used a Deformable Part Model [21] trained on the INRIA Person dataset and applied only the filter corresponding to heads with a low threshold. This is because heads are often very small and partially occluded in such images.

We find that there are many false positives and negatives in the detection results. However, we perform a lot better for nearby/larger heads. Since the texture analysis methods are crowd-blind and work well mostly for very dense crowds, we need SIFT-based analysis and head detections for adding robustness to the system such that we can perform accurately in relatively low-density environments too.

Each detection is accompanied by the scale and confidence associated with it. For each cell we return the number of detection, η_H, and means and variances of the scales and confidences.

2.2. Total Count in a Cell

We densely sample cells from the training images and obtain counts and other features from all the above methods. We then use the annotations to train a ε-SVR with the counts and other features from above as inputs and the final estimate of the count as output. This SVR combines the information obtained from the five different sources to give an estimate of the patch count.

The total person count of the image is finally obtained by summing the counts obtained from all cells in the grid. Here we are assuming that the cell counts are independent. This is a reasonable assumption because we are dealing with widely varying viewpoints, perspective effects and crowd densities. We believe that putting neighborhood constraints, as done in [15], limits the efficacy of multi-source count estimation to crowds with mostly uniform densities.

3. Experiments

3.1. Dataset

We first use the publicly available UCF crowd counting dataset to compare our results to past work. This dataset contains 50 images having 96 to 4633 people with an average of 1280 people per image. The authors of [15] provide the ground truth dot-annotations with the images, i.e., each person is marked with a dot. There are 63974 annotations in the 50 images.

We, then, collected 50 more images from various sources and added annotations to extend the above dataset to 100 images. We included a wide variety of viewpoints and perspectives in these images. This was done to ensure that we have an estimate of the robustness of this system. We finally had 100 images with 87135 annotations containing, on average, 871 persons per image; the number varying from 81 to 4633.

3.2. Evaluation Metrics

We use absolute error (AE), $\mu_{AE} = \frac{1}{N}\sum_{i=1}^{N}|\eta_i - \hat{\eta}_i|$, and normalized absolute error (NAE), $\mu_{NAE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\eta_i - \hat{\eta}_i|}{\eta_i}$, for evaluating the performance. Here $\hat{\eta}_i$ is the estimated count, η_i is the actual ground truth count, and N is the number of cells/images. We report the mean and deviations of both AE and NAE for both the UCF dataset and our extended dataset. We report the per-cell performances too.

3.3. Evaluation

We randomly divided the UCF dataset into groups of 10 and ran 5-fold cross validation. We compare the

performance of our model with the models presented in [15], [27], and [3] in Table 1. These methods are among the very few suited for this problem because most other methods rely either on videos or on human detection, and cannot be used with the UCF dataset.

Table 1. Comparison of the Proposed Method with [3] ,[15], and [27] Using the Means and Standard Deviations of Absolute Error (AE) and Normalized Absolute Error (NAE) for UCF Crowd Counting Dataset

Method	AE	NAE
Rodriguez *et al.*	655.7±697.8	0.706±1.02
Lempitsky *et al.*	493.4±487.1	0.612±0.916
Proposed	514.1±526.4	0.542±0.484
Idrees *et al.*	419.5±541.6	0.313±0.271

The method presented by in [27] used head detections for counting while Lempitsky *et al.* [3] used SIFT features to learn a regression function for counting. The authors of [15] found that [27] performs best around counts of 1000, but as we move away on either side, its error increases. This is because the estimated counts are fairly steady across the dataset and do not respond well to change in crowd density. It overestimates the counts in the low count images and underestimates in the high count images leading to high absolute error for both these cases. However, [3] performs well at higher counts but poorly in terms of NAE for lower counts. The MESA-distance in [3] is designed to minimize the maximum AE across image during training. Images with higher counts tend to have higher AE, and thus, the algorithm focusses mainly on these images. The model gets biased towards high density images. From Fig. 2, we see that the proposed method too performs poorly for some low-count images. However, the method performs quite well in the middle and high count range (>1000 individuals per image).

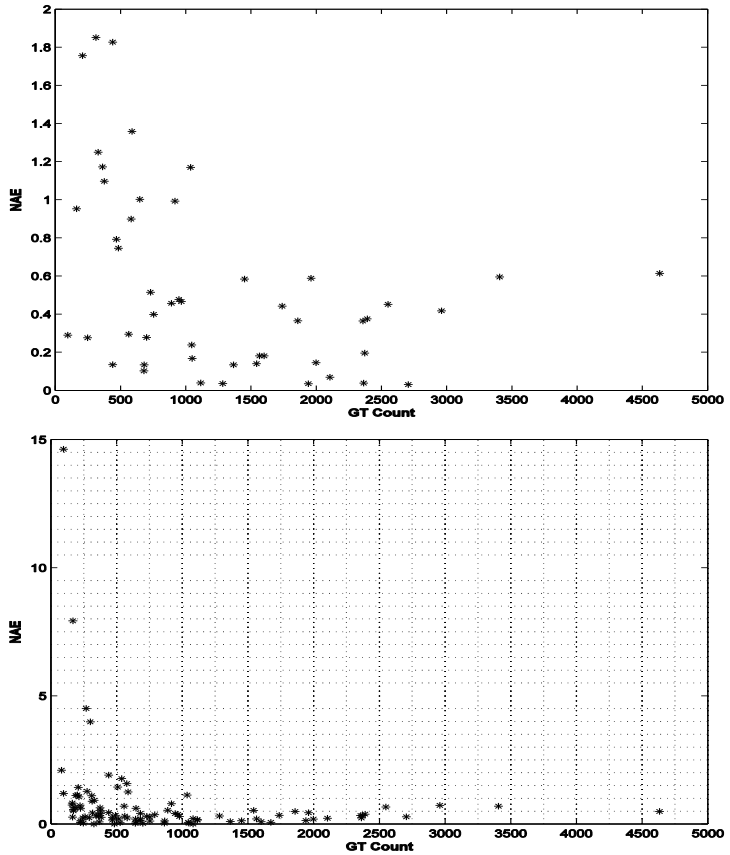

Fig. 2. (Best viewed digitally) NAE vs. ground truth counts for (left) UCF 50; and (right) complete dataset.

We observe that, unlike our method, all the other three methods tend to underestimate the counts in the high density range. Note here that most images in this category have a very high-resolution. They have a low chance of having individuals missed during annotation. Also, the per cell density increases super-linearly for this group, which is linear for other categories. Since, there are very few of such images, they could have been taken as outliers during training. Our method mostly gives average or better performances on such images. It relies heavily on the texture information from the images to estimate the counts. So it performs well on the higher density crowds, since the texture approximation is best applicable to such crowds.

We also evaluated the performance of our proposed method on the extended dataset of 100 images. This dataset has much more diversity than the original dataset because it has crowds present in varying densities and visible from various viewpoints. This is useful for testing the robustness of the algorithm. For this case, we divided the dataset into sets of 25 and ran 4-fold cross validation.

Table 2. Per-patch and Per-image Results for the Complete Dataset of 100 Images

	AE	NAE
Per-patch	9.5±14.682	-
Per-image	377.7±480.8	0.666±1.123

Table 2 gives the performance of the algorithm on the final dataset at both the patch level and the image level. We obtained a mean absolute error of 377.7 with standard deviation 480.8 and a mean normalized error of 0.666 with standard deviation 1.123. The reason for a seemingly higher NAE is evident from Fig. 2. There are a very small number of images in the low crowd-density category which drive the mean NAE up. Our method does not work quite well for some low density images. Fig. 3 shows some images with the lowest and highest absolute errors.

Est: 641.5, GT: 644 Est: 1668.1, GT: 1674 Est: 1012.3, GT: 3406 Est: 1557.9, GT: 4633

Fig. 3. Estimated counts for images with lowest (first two) and highest (last two) absolute errors (AE).

We observe that most of the images for which we get high absolute errors are very high density crowds. These images mostly contain extreme perspective variations. Also, some of the images have lens distortions which may be a reason for poor estimates. We also note that the NAE is very high for some of the images in the low density region. We believe that texture methods do not perform very well for such images. Further research in this area could focus on finding ways to pre-determine the density of crowds in different image regions so that methods like head detections and interest-point analysis could be given more importance for regions with low crowd density. Also, only a few images are driving the average absolute error and NAE up. Removing just the worst 10% performing images from the final dataset and considering the rest 90 images reduces the absolute error to 256.3 ± 217.7 and the NAE to a very low 0.407 ± 0.328.

Fig. 4 shows the per patch performance of the algorithm. We observe that, for higher density crowds, the mean absolute error per patch increase with increase in actual count. The absolute error per patch is almost constant, and very small, till around image 90, i.e., for images with counts less than about 2000. This is a demonstration of the efficacy of algorithm presented.

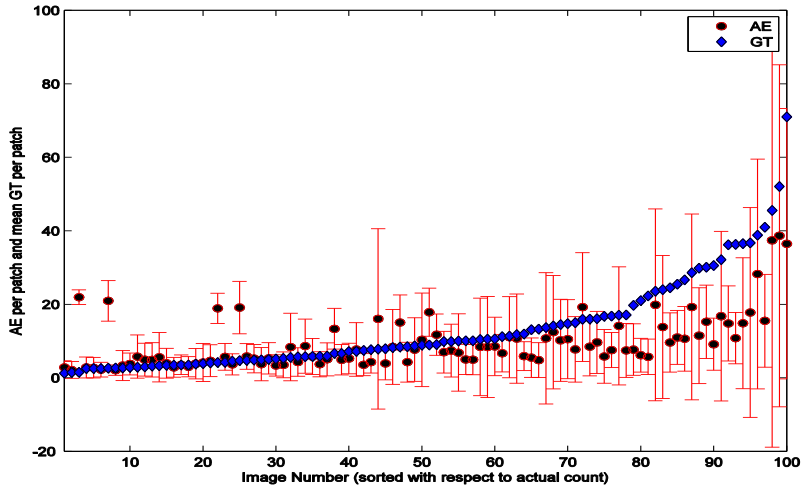

Fig. 4. (Best viewed digitally) Analysis of patch estimates in terms of absolute error per patch. The image numbers have been sorted with respect to the actual counts. Black dots are the mean absolute errors, red bars represent the standard deviations and blue diamonds are the ground truths.

4. Conclusion

We considered a method for estimating the number of people in extremely dense crowds from still images. The counting problem at this scale has barely been tackled before. We presented a method that uses information from multiple sources of information (head detections, interest points based counting and texture analysis methods) to estimate the count in an image. Each of these constituent parts gives an independent estimate of the count, along with confidences and other features, which are then fused to give a final estimate. We presented results of extensive tests and experiments we performed. We also introduced a new dataset of still images along with annotations which can complement the existing UCF dataset. The results are very promising and, since the model is extremely simple, it can be applied for real-time counting in critical areas like pilgrimage sites.

References

[1] Chan, A. B., Liang, Z.-S. J., & Nuno, V. (2008). Privacy preserving crowd monitoring: Counting people without people models or tracking. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.* IEEE.

[2] Kong, D., Douglas, G., & Hai, T. (2005). Counting pedestrians in crowds using viewpoint invariant training. *BMVC*.

[3] Lempitsky, V., & Andrew, Z. (2010). Learning to count objects in images. *Advances in Neural Information Processing Systems*.

[4] Ryan, D., Simon, D., Clinton, F., & Sridha, S. (2009). Crowd counting using multiple local features. *Digital Image Computing: Techniques and Applications,* 81-88.

[5] Chen, K., Chen, C. L., Gong, S. G., & Xiang, T. (2012). Feature mining for localized crowd counting. *BMVC*, *1(2)*.

[6] Li, M., Zhang, Z. X., Huang, K. Q., & Tan, T. (2008). Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. *Proceedings of ICPR 19th International Conference on Pattern Recognition*. IEEE.

[7] Chan, A. B., & Nuno, V. (2012). Counting people with low-level features and Bayesian regression. *IEEE Transactions on Image Processing, 21(4)*, 2160-2177.

[8] Zhao, T., Ram, N., & Bo, W. (2008). Segmentation and tracking of multiple humans in crowded environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(7)*, 1198-1211.

[9] Arandjelovic, O. (2008). Crowd detection from still images. *Proceedings of the British Machine Vision Association Conference*.

[10] Marana, A. N., Velastin, S. A., Costa, L. F., & Lotufo, R. A. (1998). Automatic estimation of crowd density using texture. *Safety Science*, *28(3)*, 165-175.

[11] Ma, W., Lei, H., & Liu, C. P. (2010). Crowd density analysis using co-occurrence texture features. *Proceedings of 2010 5th International Conference on Computer Sciences and Convergence Information Technology*. IEEE.

[12] Chan, A. B., Mulloy, M., & Nuno, V. (2009). Analysis of crowded scenes using holistic properties. *Proceedings of Performance Evaluation of Tracking and Surveillance Workshop at CVPR*.

[13] Marana, A. N., *et al*. (1998). On the efficacy of texture analysis for crowd monitoring. *Proceedings of International Symposium on Computer Graphics, Image Processing, and Vision*.

[14] Marana, A. N., Costa, L. F., Fotufo, R. A., & Velastin, S. A. (1999). Estimating crowd density with Minkowski fractal dimension. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing: Vol. 6.* IEEE.

[15] Idrees, H., Imran, S., Cody, S., & Mubarak, S. (2013). Multi-source multi-scale counting in extremely dense crowd images. *Proceedings of 2013 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

[16] Brostow, G. J., & Roberto, C. (2006). Unsupervised Bayesian detection of independent motion in crowds. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition: Vol. 1*.

[17] Ge, W. J., & Robert, T. C. (2009). Marked point processes for crowd counting. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

[18] Felzenszwalb, P. F., & Daniel, P. H. (2006). Efficient belief propagation for early vision. *International Journal of Computer Vision*, *70(1)*, 41-54.

[19] Rabaud, V., & Serge, B. (2006). Counting crowded moving objects. *Proceedings of 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition: Vol. 1*.

[20] Rodriguez, M., *et al*. (2011). Data-driven crowd analysis in videos. *Proceedings of 2011 IEEE International Conference on Computer Vision* (pp. 1235-1242). IEEE.

[21] Felzenszwalb, P., David, M., & Deva, R. (2008). A discriminatively trained, multiscale, deformable part model. *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-8).

[22] Felzenszwalb, P. F., Ross, B. G., David, M., & Deva, R. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9)*, 1627-1645.

[23] Cho, S.-Y., Chow, T. W. S., & Leung, C.-T. (1999). A neural-based crowd estimation by hybrid global learning algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 29(4)*, 535-541.

[24] Ferryman, J., & Ellis, A. (2010). PETS2010: Dataset and challenge. *Proceedings of 2010 Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance*. IEEE.

[25] Vedaldi, A., & Brian, F. (2010). VLFeat: An open and portable library of computer vision algorithms. *Proceedings of the International Conference on Multimedia*. ACM.

[26] Chan, A. B., & Nuno, V. (2008). Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(5)*, 909-926.

[27] Rodriguez, M., Ivan, L., Josef, S., & Audibert, J.-Y., (2011). Density-aware person detection and tracking in crowds. *Proceedings of 2011 IEEE International Conference on Computer Vision* (pp. 2423-2430).

Ankan Bansal was born in India in 1992. He received the B.Tech. and M.Tech. degrees in electrical engineering from Indian Institute of Technology, Kanpur, India. His main areas of interest are object recognition, scene understanding and biometric recognition.

He started his PhD program at the University of Maryland, College Park, USA in August 2015.

K. S. Venkatesh received the B.E. degree in electronics from Bangalore University, India, in 1987, the M.Tech. degree in communication from Indian Institute of Technology, Kanpur in 1989, and the PhD degree in signal processing from IIT Kanpur in 1995.

He is currently with the Faculty of the Electrical Engineering Department at IIT Kanpur. His interests include image and video processing and their application in computer vision.