# Crowd Density Estimation for Outdoor Environments

Marjan Jalali Moghaddam, Elham Shaabani, and Reza Safabakhsh
Amirkabir University of Technology
Tehran, Iran
64542728, +9821
{jalalymarjan, eshabani, safa}@aut.ac.ir

## ABSTRACT
Crowd density analysis is crucial in management and control of the crowds and ensure safety. In this paper, we proposed two crowd density estimation methods using texture descriptors of the image in outdoor scenes. Two methods based on different techniques, one using Local Binary Pattern and Gabor filters, and the other using 5 statistics of Grey Level Co-occurrence Matrix and firefly classification algorithm, are applied on the dataset. The assessment and comparison with other algorithms have been carried out using different sets of PETS 2009 dataset. The proposed approaches can carry out the estimation more accurately, the rate of true classification is 93.16% and 92.99%, respectively. The results show that our proposed algorithms outperform the other algorithms with a significant margin.

## Categories and Subject Descriptors
I.4.3 **[Image Processing and Computer Vision]**: Enhancement--Filtering; I.4.7 **[Image Processing and Computer Vision]**: Feature Measurement—Texture.

## General Terms
Measurement.

## Keywords
Crowd Density Estimation; Texture Analysis; Local Binary Pattern; Gabor Filter; Grey Level Co-occurrence Matrix; Firefly Classification.

## 1. INTRODUCTION
One of the most important problems in intelligent video surveillance is crowd analysis which includes crowd density estimation. There has been an increasing interest in crowd density estimation due to its importance in management and control of the crowds and ensuring safety. Generally, the computer vision techniques for crowd density estimation can be categorized as follows:

**Count-based approach** in which the number of people in the image are counted [1, 2]. In this approach, different techniques can be utilized including shape-based [1-3], tracking-based [4], map-based [5-2], and feature-based techniques.

**Classification-based approach** in which the crowd is classified based on the density [6, 7, 8, 9]. In the feature-based approach, the density features such as texture, edge, HOG, and Chebyshev Moments are extracted from the image. Using different algorithms, the system learns the relation between the feature vector and the density. This approach is more effective in crowded scenes.

As this paper considers the classification approach, the existing methods in this category are briefly discussed below.

Rahmalan et al. applied the discrete orthogonal Chebyshev Moment to estimate the crowd density. They extracted features using GLDM [8], Minkowski Fractal Dimension (MFD) [9] and Translation Invariant Orthonormal Chebyshev Moments (TIOCM) [10]. Hou et al. estimated the number of people in a complicated scene using the relationship between the foreground pixels and the number of people [11].

Li et al. proposed an improved estimation approach based on texture analysis. They first used a combination of optical flow and background subtraction method to remove background. They then extracted features from the whole foreground using texture analysis. They also proposed a new feature combining both the energy and the entropy [12]. Wen et al. applied a set of 2-D Gabor filters including different directions and frequencies to extract the texture features which provide for the crowded scene. The texture features are the mean and variance of the filtered output images [13]. Wang et al. proposed a Local Binary Pattern Co-occurrence Matrix (LBPCM) based algorithm to extract texture features for crowd density estimation. LBPCM are extracted from both gray and gradient images [14]. Fradi et al. estimated crowd density using multi scaled patch and a discriminant subspace of the high-dimensional Local Binary Pattern (LBP) feature vector. For dimension reduction, they used the combination of Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA) [7].

In this paper, we proposed two robust and accurate methods to classify the outdoor scene images to different crowd density classes. We do not apply any background subtraction algorithms. The proposed methods outperform the other algorithms significantly on different brightness levels and shadows. Images are made up of different patterns. There is a correlation between density and texture patterns. Crowd images ranging from low to high densities consist of coarse to fine patterns. The techniques presented in this paper use this fact to classify images to different crowd density levels in outdoor scenes. The illumination variations pose difficulties in using outdoor scenes as input data. In the first technique, we used Local Binary Pattern (LBP) method and multi-channel Gabor filters to extract features and a Least Square Support Vector Machine (LSSVM) for the task of crowd density classification (LBPG). In the second technique, we used

BICT 2014, December 01-03, Boston, United States

DOI 10.4108/icst.bict.2014.257913

GLCM method to extract features and firefly algorithm (FA) to classify them (GLCM-Firefly).

The advantages of the proposed approaches are as follows:

- No need to subtract the background.
- No need to eliminate shadow.
- Mitigates the issues of perspective distortion and occlusion.

The rest of this paper is organized as follows. Section 2 reviews LBP and Gabor and presents the LBPG scheme. Section 3 reviews GLCM and FA and presents GLCM-Firefly. Section 4 reports experimental results and Section 5 concludes the paper.

# 2. PROPOSED METHODS

## 2.1 The LBPG Method

### A. Gabor Filters

Tan proposed a multi-channel spatial filtering approach for texture feature extraction and classification [15]. A 2-D Gabor filter is a product of a 2-D Gaussian with oriented sinusoids. An appropriate filter will extract useful information such as spots and edges. The Gabor filter responds to the strong edges of filter orientation with large magnitudes. Moreover, it provides some degree of illumination, scale, translation, and rotation invariance. So, filtering can be performed at different scales to find patterns of different sizes. Therefore, the Gabor filter output can give effective texture descriptors for crowd density estimation. In this paper, feature extraction in the space and spatial-frequency domains can be done with even-symmetry ($h_e$) and odd-symmetry ($h_o$) Gabor filters via convolution as:

$$h_e(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \cos\left(2\pi\omega_0(x\cos\theta_0 + y\sin\theta_0)\right) \quad (1)$$

$$h_o(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \sin\left(2\pi\omega_0(x\cos\theta_0 + y\sin\theta_0)\right) \quad (2)$$

$$q_e(x,y) = p(x,y) \otimes h_e(x,y) \quad (3)$$

$$q_o(x,y) = p(x,y) \otimes h_o(x,y) \quad (4)$$

$$q(x,y) = \sqrt{q_e^2(x,y) + q_o^2(x,y)} \quad (5)$$

where $\omega_0, \theta_0, \sigma$ are the central frequency, orientation, and spatial constant, respectively, p is the input image, and $\otimes$ denotes two dimensional linear convolution.

If the original image is multiplied by a constant, the filter response will also be multiplied by that constant. However, in the real world, the illumination and color variations are not monotonous. Therefore, we apply LBP to reduce the sensitivity of this approach to monotonic illumination variations.

### B. Local Binary Pattern

Local Binary Pattern [16] is a simple efficient operator for texture analysis. The main reason that we apply it is its robustness to monotonic gray-scale changes. It is widely used in many applications, including face recognition, face detection, and texture classification. LBP labels the pixels with the value obtained from its neighborhood pixels. It considers a 3*3 neighborhood called a unit. Thresholding the neighborhoods by the center value reduces the neighbors to 1 or 0 (Fig. 1.b). To each neighbor position a weight is assigned (Fig. 1.c). The thresholded values are multiplied by their corresponding weights (Fig 1. d). Finally, the values of the eight neighbors are summed up to obtain the value of this unit.

### C. The First Proposed Method (LBPG)

The proposed method is shown in Fig. 2. First, the RGB input images are converted to gray level images and are resized to 320x240 to speed up the proposed algorithm. Then we enhance the image with a 3×3 mean filter, and histogram equalization. These steps are specified as preprocessing step in Fig. 2. Next we apply LBP algorithm to reduce the monotonic gray-scale changes of the enhanced image. In the next step, the features are extracted using 24 two-dimensional visual cortical Gabor filters. We apply multi-channel Gabor filters with various directions and frequencies to extract features from the LBP image. We assigned six values including 2, 4, 8, 16, 32, 64 for frequencies ($\omega_0$), four values consisting of $0^\circ$, $45^\circ$, $90^\circ$ and $135^\circ$ for orientations, and $\frac{1}{\omega_0}$ for spatial constant σ. Since the histograms of the channel output images are often close to a Gaussian shape [15], only the mean values and the standard deviations of channel output images are computed and used as texture features. Therefore, the means and the standard deviations of the channel output images are used as 48 texture features. These features include 24 channel output means and 24 channel output standard deviations.

Finally, we classify crowd density using the Least Square Support Vector Machine (LSSVM) toolbox [17] which is a reformulation to the standard SVMs which leads to solving linear KKT systems. SVM is a powerful methodology for solving problems in nonlinear classification. We apply multi-class coding, and use 'code_MOC' for loading parameter. In order to make an LS-SVM model (with RBF kernel), we need two tuning parameters: $\delta^2$ (sig2), the squared bandwidth and γ (gamma), the regularization parameter, determining the trade-off between the training error minimization and smoothness. The optimal value of gamma and sig2 are tuned according to the data [17].

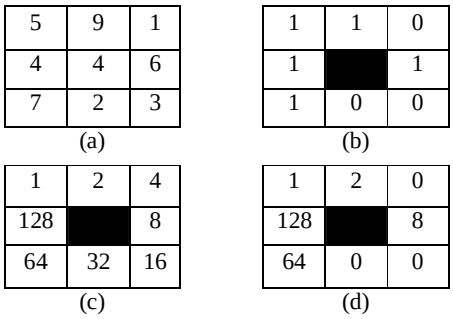


**Figure 1. Two-level LBP of the texture unit.**

## 2.2 The GLCM – Firefly Method

### A. Gray Level Co-occurrence Matrix

The grey level co-occurrence matrix (GLCM) [6, 18] is a two-dimensional histogram which is formed of the estimation of second-order joint conditional probability density functions, $f(i,j|d,\theta)$. Each element $f(i,j|d,\theta)$ is the probability of the pair of grey levels (i,j) in the image, separated by a distance d along the direction θ. The calculation of $f(i,j|d,\theta)$ for all different pairs of (d,θ) has a high computational complexity. Therefore, one distance (1 pixel) and four angles (0, 45, 90, and 135) are used. In this paper, the following five spread indicators proposed by Haralick [18] are used:

**Contrast** measures the local intensity contrast over the whole image.

**Correlation** measures the joint probability occurrence of the specified pixel pairs.

**Energy** provides the sum of squared elements in the GLCM.

**Entropy** calculates the entropy of an intensity image.

**Homogeneity** measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal.

## B. Firefly Algorithm

The firefly algorithm (FA) [19] is inspired by the short and rhythmic flashes of fireflies. The flashes lead to attraction of mating partners and potential prey. Most fireflies' vision is usually limited to several hundred meters at night because of two reasons. First, the light intensity and distance have a negative correlation. That is, it obeys the inverse square law ($\text{light intensity} \propto \frac{1}{distance^2}$). Moreover, the light is absorbed by the air. Fireflies are unisex. That is, they are attracted to other ones regardless of their sex. There is a positive correlation between attractiveness and brightness. Therefore, for any two fireflies, the brighter firefly attracts the less bright one. Otherwise it attracts randomly. The landscape of the objective function determines the brightness of a firefly. The algorithm is described as follows:

Input: Feature vectors x

1. Initialization: For each population member known as firefly, input vectors are labeled randomly with cluster numbers while there is no empty cluster. The mean value of each cluster is considered as the centroid of it. That is, a firefly is the center of clusters which is formulated as follows:

$$w_{ij}=\begin{cases} 1, & x_i \in \text{cluster}_j \\ 0, & x_i \notin \text{cluster}_j \end{cases} \quad (6)$$

$$\text{cen}_j=\frac{\sum_{i=1}^{S} w_{ij}\, x_i}{\sum_{i=1}^{S} w_{ij}} \; ; \; j = 1,2,\dots,N \quad (7)$$

where $x_i$ is the input data i. S is the number of input vectors (i.e. input images), and N is the number of founded clusters which is equal to or lower than the number of clusters specified by the user.

2. Fitness calculation: For each firefly, the fitness is determined based on inter-cluster similarity. For each firefly, the fitness is calculated as follows:

$$\text{Fitness}=\left(\sum_{j=1}^{N}\sum_{i=1}^{S} w_{ij}(x_i - \text{cen}_j)^2\right)^{-1} \quad (8)$$

3. Best firefly selection: The firefly with maximum inter-cluster similarity or maximum fitness is selected as the best solution.

4. Centroid mutation: For each two fireflies, the firefly with lower fitness (l) is absorbed to the firefly with higher fitness (h):

$$\text{cen}_l' = \text{cen}_l^{old} + \beta e^{-\gamma r_{l,h}^2}(\text{cen}_h^{old} - \text{cen}_l^{old}) + \alpha\epsilon \quad (9)$$

$$r_{l,h} = \|\text{cen}_l - \text{cen}_h\| \quad (10)$$

where α controls the step size, and ϵ is a vector drawn from the uniform distribution [-0.5 0.5]. The attractiveness is defined by γ which plays an important role to determine the convergent speed and FA behavior.

5. Cluster determination: For each firefly, the cluster of each input data is determined as:

$$\text{Cluster}(x_i) = \text{cen}_j \text{ with } j = \arg\min_{n\in N}\{\| x_i - \text{cen}_n \|\} \quad (11)$$

6. Center calculation: For each firefly, new centers are calculated as in step 1.

7. Repeat step 2 to 6 until the termination condition (number of generations) is satisfied.

We considered 100, 0.3, 1, and 1 as generation limit, alpha, gamma and beta, respectively.

## C. The Second Proposed Technique (GLCM-Firefly)

The second proposed system is shown in Fig. 3. First, the RGB input images are converted to gray level images and are resized to 320x240 to speed up the proposed algorithm. Then, we applied GLCM to the given image to extract a feature vector of size 20 (5 different features with parameters $(1,0^\circ), (1,45^\circ), (1,90^\circ)$ *and* $(1,135^\circ)$). Next we clustered the extracted feature vectors using FA. At this stage, each input image is assigned to one cluster. Finally the label of each cluster is specified by majority voting between its members according to the known class labels of them. The clusters of test images are specified according to the minimum distance approach.

# 3. EXPERIMENTS

## 3.1 Experimental Setting

The proposed approach for crowd density estimation is evaluated within the R1 PETS 2009 dataset [20] with the resolution of 420x271 pixels as shown in Fig. 4. We manually selected 600 images of different sets of View 001 to cover all crowd density classes, brightness levels and shadows. Based on the manual estimation, the image were separated in group of low (lower than or equal to 10 people), moderate density (between 11 and 20 people), and high density (higher than 20 people).

The proposed algorithms are implemented using Matlab language on an Intel® core ™ i5 M460 2.53 GHz PC with 4 GB RAM. To use all data for both training and testing, we have used the 10-fold cross validation. We randomly broke data into 10 sets while each set is made up of the same number of crowd levels.

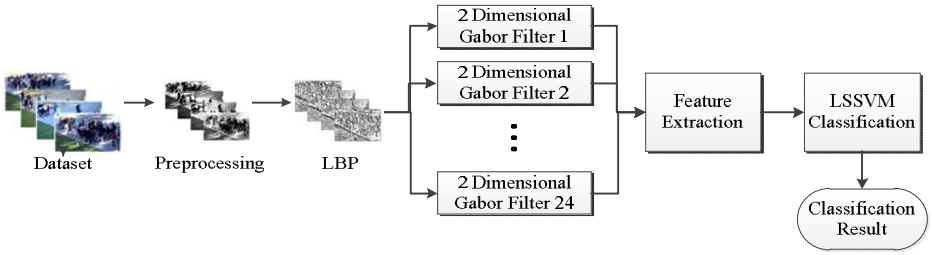


**Figure 2. LBPG schematic.**

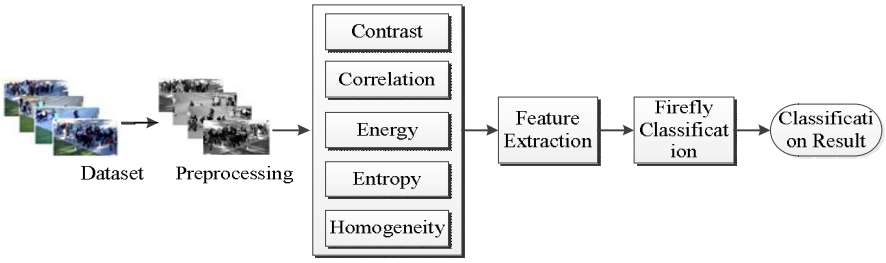


**Figure 3. GLCM-Firefly schematic.**

## 3.2 Experimental Result

In this section, we present the results of our experiments. Table 1 compares the classification accuracy of LBPG density estimation technique corresponding to RBF and Polynomial kernels. The performance of RBF kernel in 7 folds was better than or similar to that of the Polynomial kernel. RBF kernel has a higher mean and lower standard deviation in 10-folds compared to the polynomial kernel. The actual and predicted classes of these two kernels are also visualized in detail in part (a) and (b) of Table 2. According to Table 2 (a), 95%, 87.5%, and 97% of low, moderate and high density images are guessed correctly. Moreover, the RBF kernel has a better performance in estimating low and high densities. Moreover, the average classification accuracy is improved by 1.33% for the RBF kernel as compared to the Polynomial kernel. According to Table 2 (a, b), the RBF kernel was able to decrease non-neighbor errors. That is, it resulted in lower misclassifications between high and low density classes.

In the proposed GLCM-firefly approach, we applied different number of fireflies to find the best setting, the results of which are shown in Fig. 5. The best result is achieved for about 50 fireflies. It shows a trade-off between the number of fireflies and the learning quality. The more fireflies, the more learning. However, a large number of fireflies results in the over-fitting problem and decreases the generalization.


**Figure 4. Regions in PETS view 001 dataset**

The confusion matrix of GLCM-firefly is also shown in Table 2 (c). In this method, the wrongly estimated densities are always assigned to neighbor classes. Finally, we implemented several crowd density estimation algorithms and used them to estimate crowd density level on our data set described in the previous part. The algorithms we used are GLCM, TIOCM [10, 12], and Gabor, in which Gabor filter is applied to the gray level input image instead of the LBP image. Table 3 shows the performance of the proposed feature compared to other crowd density classification methods. The proposed methods have achieved the best mean and standard derivation of classification accuracy. Comparing Gabor and LBPG showed that LBP plays an important role in decreasing the effects of gray value changes. Therefore, LBP significantly increased the robustness of the algorithm.

There is a trade–off between accuracy and running time in the proposed methods. LBPG has a better performance than others. The GLCM-Firefly algorithm is the fastest one and its accuracy is higher than the others except for the LBPG algorithm.

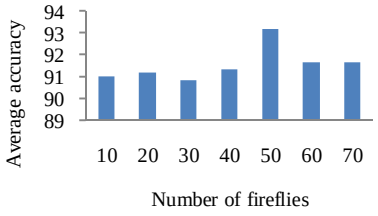


**Figure 5. The mean of the percentage of the accuracy in 10-folds.**

**Table 1. Comparing the classification accuracy of LBPG with different classification kernels.**

| | | Kernel Type | |
|---|---|---|---|
| | | RBF | Polynomial |
| Fold Number | 1 | 95 | 90 |
| | 2 | 91.66 | 88.33 |
| | 3 | 93.33 | 95 |
| | 4 | 98.33 | 96.66 |
| | 5 | 88.33 | 95 |
| | 6 | 96.66 | 96.66 |
| | 7 | 90 | 93.33 |
| | 8 | 93.33 | 86.66 |
| | 9 | 95 | 91.66 |
| | 10 | 90 | 85 |
| Mean | | 93.16 | 91.83 |
| St. Dev. | | 3.18 | 4.19 |

## 4. CONCLUSIONS

In this paper, the LBPG and GLCM-Firefly algorithms were proposed for crowd density estimation based on texture descriptors. These algorithms were evaluated using 600 selected images on R1 PETS 2009 dataset. We assessed and compared the proposed algorithms with three other methods, namely, GLCM, TIOCM, and multi-channel Gabor filter. As shown in the results, the proposed algorithms outperform the other algorithms significantly. The rates of true classification in LBPG and GLCM-Firefly are 93.16% and 92.99%, respectively. The GLCM-Firefly approach has the best time complexity and the wrongly estimated densities are always assigned to the neighboring classes.

**Table 2. Confusion matrixes of a) LBPG algorithm using RBF kernel, b) LBPG algorithm using Polynomial kernel, and c) of GLCM-Firefly algorithm.**

| | | | Predicted Density Level | | |
|---|---|---|---|---|---|
| | | | Low | Moderate | High |
| True Density Level | (a) | Low | 95% | 5% | 0% |
| | | Moderate | 9.5% | 87.5% | 3% |
| | | High | 2% | 1% | 97% |
| | (b) | Low | 91.5% | 8.5% | 0% |
| | | Moderate | 7.5% | 91% | 1.5% |
| | | High | 5.5% | 1.5% | 93% |
| | (c) | Low | 94% | 6% | 0% |
| | | Moderate | 7% | 86.5% | 6.5% |
| | | High | 0% | 1.5% | 98.5% |

**Table 3. Classification accuracy of different algorithms.**

| | | Algorithm | | | | |
|---|---|---|---|---|---|---|
| | | GLCM | TIOCM | Multi-channel Gabor | LBPG | GLCM-Firefly |
| Fold Number | 1 | 86.66 | 85 | 78.33 | 95 | 96.66 |
| | 2 | 83.33 | 85 | 71.67 | 91.66 | 91.66 |
| | 3 | 91.66 | 78.33 | 61.67 | 93.33 | 88.33 |
| | 4 | 86.66 | 85 | 75 | 98.33 | 91.66 |
| | 5 | 83.33 | 85 | 70 | 88.33 | 98.33 |
| | 6 | 91.66 | 83.33 | 66.67 | 96.66 | 91.66 |
| | 7 | 86.66 | 81.66 | 78.33 | 90 | 96.66 |
| | 8 | 85 | 86.66 | 73.33 | 93.33 | 91.66 |
| | 9 | 90 | 91.66 | 78.33 | 95 | 91.66 |
| | 10 | 90 | 85 | 65 | 90 | 91.66 |
| Mean | | 87.5 | 84.66 | 71.83 | 93.16 | 92.99 |
| St. Dev. | | 3.16 | 3.4 | 5.95 | 3.18 | 3.12 |
| Time (ms) | | 0.0856 | 17.1452 | 0.8509 | 0.9645 | 0.0733 |

## REFERENCES


[1] Swets, D. L., Punch, B., & Weng, J. (1995, October). Genetic algorithms for object recognition in a complex scene. In Image Processing, 1995. Proceedings., International Conference on (Vol. 2, pp. 595-598). IEEE.

[2] Yoshinaga, S., Shimada, A., & Taniguchi, R. I. (2010). Real-time people counting using blob descriptor. Procedia-Social and Behavioral Sciences, 2(1), 143-152.

[3] Merad, D., Aziz, K. E., & Thome, N. (2010, August). Fast people counting using head detection from skeleton graph. In Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on (pp. 233-240). IEEE.

[4] Yuan, X., Lu, Y. J., & Sarraf, S. (1993, October). A computer vision system for measurement of pedestrian volume. In TENCON'93. Proceedings. Computer, Communication, Control and Power Engineering. 1993 IEEE Region 10 Conference on (pp. 1046-1049). IEEE.

[5] Kong, D., Gray, D., & Tao, H. (2006, August). A viewpoint invariant approach for crowd counting. In Pattern Recognition, 2006. ICPR 2006. 18th International Conference on (Vol. 3, pp. 1187-1190). IEEE.

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

[7] Fradi, H., Zhao, X., & Dugelay, J. L. (2013, July). Crowd density analysis using subspace learning on local binary pattern. In Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on (pp. 1-6). IEEE.

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

[9] Marana, A. N., da Fontoura Costa, L., Lotufo, R. A., & Velastin, S. A. (1999, March). Estimating crowd density with Minkowski fractal dimension. InAcoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on (Vol. 6, pp. 3521-3524). IEEE.

[10] Rahmalan, H., Nixon, M. S., & Carter, J. N. (2006, June). On crowd density estimation for surveillance. In Crime and Security, 2006. The Institution of Engineering and Technology Conference on (pp. 540-545). IET.

[11] Hou, Y. L., & Pang, G. K. (2008, September). Automated people counting at a mass site. In Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on (pp. 464-469). IEEE.

[12] Li, W., Wu, X., Matsumoto, K., & Zhao, H. A. (2010, October). Crowd density estimation: An improved approach. In Signal Processing (ICSP), 2010 IEEE 10th International Conference on (pp. 1213-1216). IEEE.

[13] Wen, Q., Jia, C., Yu, Y., Chen, G., Yu, Z., & Zhou, C. (2011). People number estimation in the crowded scenes using texture analysis based on gabor filter.Journal of Computational Information Systems, 7(11), 3754-3763.

[14] Wang, Z., Liu, H., Qian, Y., & Xu, T. (2012, July). Crowd density estimation based on local binary pattern co-occurrence matrix. In Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on (pp. 372-377). IEEE.

[15] Tan, T. N. (1992). Texture feature extraction via visual cortical channel modelling. In Pattern Recognition, 1992. Vol. III. Conference C: Image, Speech and Signal Analysis, Proceedings., 11th IAPR International Conference on (pp. 607-610). IEEE.

[16] Ojala, T., Pietikäinen, M., & Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. Pattern recognition, 29(1), 51-59.

[17] Suykens, J.A.K . (2011, August 16). Least Squares Support Vector Machines [online]. Available: http://www.esat.kuleuven.be/sista/lssvmlab/

[18] Haralick, R. M. (1979). Statistical and structural approaches to texture.Proceedings of the IEEE, 67(5), 786-804.

[19] Yang , X. S. (2010). Nature-Inspired etaheuristic Algorithms. Luniver Press.

[20] PETS2009 (2009, July 6). PETS 2009 [online]. Available FTP: ftp.cs.rdg.ac.uk/pub/PETS2009/