People Counting System Based on Head Detection using Faster RCNN from Both Images and Videos

N. BABY RANI[1], J.LAVANYA [2], K. SATHWIKA[2], M.LIKHITHA[2], N. SOWJANYA[2]

[1]Assistant Professor, Department of Information Technology, Mallareddy Engineering College for Women, (UGC-Autonomous), Hyderabad, India, rani.nuvvula@gmail.com.

[2]Student, Department of Information Technology, Mallareddy Engineering College for Women, (UGC-Autonomous), Hyderabad, India.

Abstract

The People Counting System is an advanced computer vision application with the primary goal of accurately counting people in crowded or monitored areas. It has been designed to address the need for real-time and reliable head detection to count individuals in various environments like shopping malls, transportation hubs, stadiums, and public gatherings. The motivation behind developing this system stems from the growing demand for automated and efficient crowd monitoring in security, retail analytics, and crowd management scenarios. Traditional people counting methods often rely on simpler techniques such as motion detection or background subtraction. However, these methods may not deliver precise results in complex scenarios with crowded environments or occlusions. Their accuracy suffers due to the lack of fine-grained object recognition and tracking, making them time-consuming, error-prone, and costly. To overcome these limitations, automating the counting process with computer vision techniques proves to be a significant improvement in terms of accuracy and efficiency. The demand for an accurate and efficient people counting system becomes essential as traditional manual counting methods and existing automated techniques struggle to handle crowded and occluded scenarios effectively. In response to this need, the proposed system offers an effective and reliable solution for automated people counting in crowded environments. Leveraging computer vision technology, this system utilizes the cutting-edge Faster R-CNN, an object detection model, to detect and count individual heads in real-time. Focusing on head detection ensures precise counting while minimizing common errors related to double counting, often encountered in simpler counting methods. This system represents a substantial advancement over traditional approaches by accurately identifying and localizing heads even in densely crowded scenes where heads may be partially obscured or overlapping with other objects. Its high accuracy ensures minimal counting errors, which proves crucial in applications where precise counting is necessary for decision-making processes, such as occupancy management, security, or retail analytics. One of the key advantages of the system is its real-time capability enabled by Faster R-CNN, allowing continuous and instantaneous counting. This feature ensures an immediate response to changing crowd conditions, making it a valuable tool for effective crowd management and decision-making in various scenarios.

1. Introduction

Counting people is one of the most important tasks in intelligent video surveillance systems and it has a wide range of applications and business value in many places, such as banks, railway stations, shopping malls, schools, etc. However, counting people in a crowded surveillance environment is a challenge task due to low resolution, occlusion, illumination change, imaging viewpoint variability, background clutter, etc. There are two main types of people detection methods: traditional and deep learning methods. Traditional learning methods extract features by the histogram of oriented gradients (HOG) and classify the features by a linear SVM classifier [1–3]. The advances in deep learning greatly bring single-image people detection to an unprecedented performance level. While many people detection algorithms have been designed for standard side-mounted cameras, the best performance to date has been achieved by deep-learning object detection algorithms [4]. Deep learning methods include single-stage approaches such as YOLO, and SSD. While two-stage approaches reach higher accuracy, single-stage approaches take up fewer computing resources and fast speed.

2. Literature Survey

In the past decades, various methods of people counting have been proposed. These methods can be divided into three groups:

Counting by trajectory clustering

This kind of methods count people by adopting tracking techniques. Antonini and Thiran [5] applied a multilayer clustering technique to trajectories which are obtained from a tracking system. This method can reduce the bias between the number of tracks and the real number of persons. Topkaya et al. [6] improved the clustering method using a generic person detector. This method fused different types of features, including color, spatial and temporal features into clustering. For some applications, these approaches can get comparative good performance. However, they are usually time-consuming and computation resource-consuming since they tightly depend on tracking techniques. Besides, the performance would degrade a lot when applying these methods to motion imaging platforms since this situation usually makes the track algorithms unstable.

Counting by regression

This type of methods adopt regression-based techniques to learn a mapping between low-level features and the people count in a scene. Lempitsky and Zisserman [7] introduced an object counting method through pixel-level object density map regression whose integral over any image region gives the count of objects within that region and learning to infer such density can be formulated as a minimization of a regularized risk quadratic cost function. Chan and Vasconcelos [8] extracted a set of holistic low-level features from each segmented region, and learned a function that maps features into the number of people with the Bayesian regression. Morerio et al. [9] proposed a global group-based approach to estimate the count of people relying on accurate camera calibration. Idrees et al. [10] estimated the number of individuals in extremely dense crowds based on multi-source and multi-scale features and regress using SVR from images. Zhang et al. [11] alternatively trained a CNN regression model for crowd counting with two related learning objectives, namely the crowd density and the crowd count. The obvious advantage of these methods is that they can circumvent explicit object segmentation and detection. However, their performance is heavily influenced by the occlusion and perspective effect. Besides, these methods just provide the information of the people count, and fail to locate the individuals. Actually, the location information of individuals is usually very important for video surveillance systems. For example, we can use this information to automatically obtain the spatial distribution of people in a scene.

Counting by detection

The basic idea of this type of methods is to design a detector to detect each individual to obtain people count. The adopted detection methods usually include body detection [12], shoulders detection [12, 13], and head detection [15]. Comparatively speaking, head detection methods usually outperform the other two types of methods in crowded applications with heavy occlusion since only the heads of many persons may be visible in this situation. In order to effectively detect heads, many methods have been presented, such as skeleton graph [16], the head template matching and learning-based methods [17], [18]. These methods can be applied to some specific scenes and would fail for some real surveillance applications.

3. Proposed System Design

Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

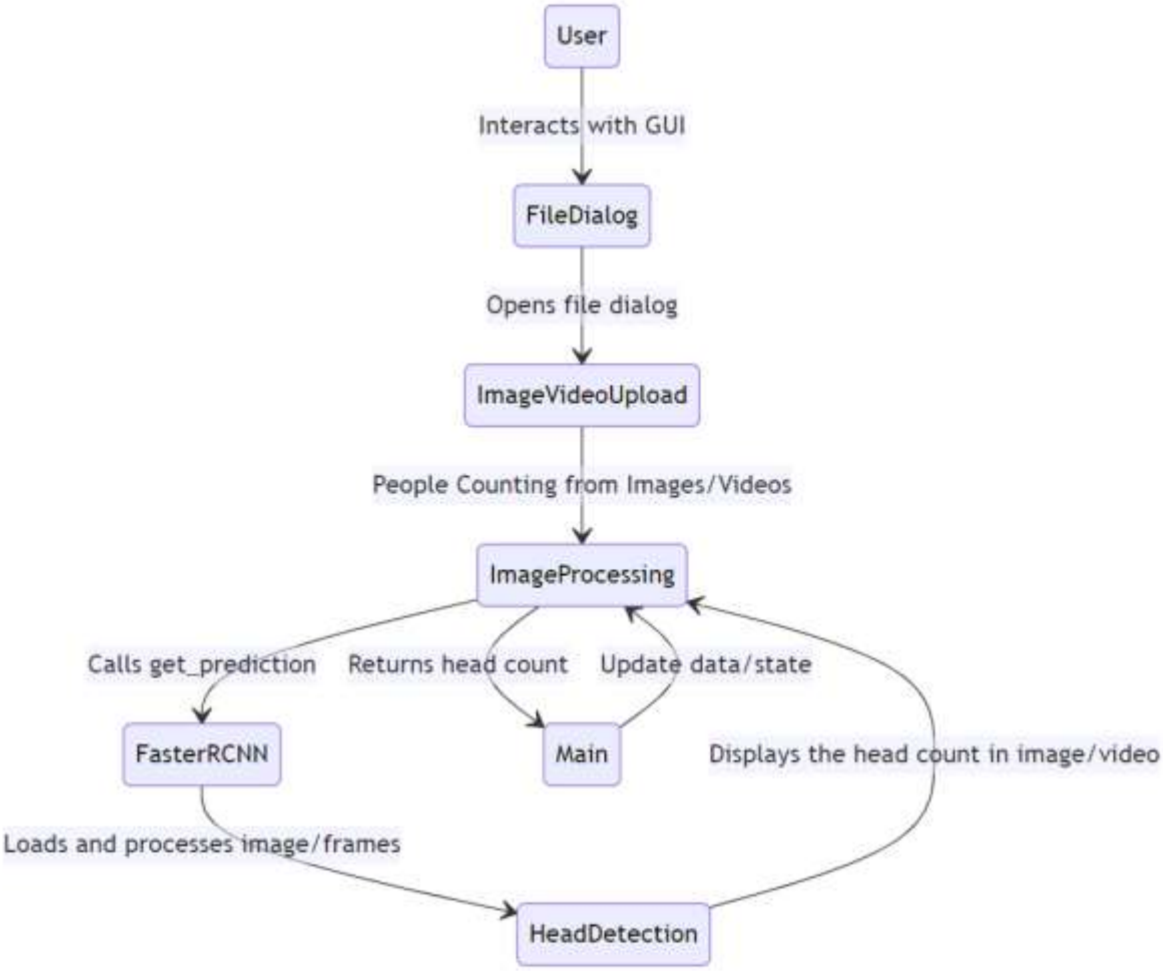


Figure 1. Proposed system design.

3.1 CNN Model

Currently, CNNs are the most researched machine learning algorithms in medical image analysis. The reason for this is that CNNs preserve spatial relationships when filtering input images. As mentioned, spatial relationships are of crucial importance in radiology, for example, in how the edge of a bone joins with muscle, or where normal lung tissue interfaces with cancerous tissue. As shown in Fig. 4.1, a CNN takes an input image of raw pixels, and transforms it via Convolutional Layers, Rectified Linear Unit (RELU) Layers and Pooling Layers. This feeds into a final Fully Connected Layer which assigns class scores or probabilities, thus classifying the input into the class with the highest probability.

Convolution Layer: A convolution is defined as an operation on two functions. In image analysis, one function consists of input values (e.g., pixel values) at a position in the image, and the second function is a filter (or kernel); each can be represented as array of numbers. Computing the dot product between the two functions gives an output. The filter is then shifted to the next position in the image as defined by the stride length. The computation is repeated until the entire image is covered, producing a feature (or activation) map. This is a map of where the filter is strongly activated and 'sees' a feature such as a straight line, a dot, or a curved edge. If a photograph of a face was fed into a CNN, initially low-level features such as lines and edges are discovered by the filters. These build up to progressively higher features in subsequent layers, such as a nose, eye or ear, as the feature maps become inputs for the next layer in the CNN architecture.

Convolution exploits three ideas intrinsic to perform computationally efficient machine learning: sparse connections, parameter sharing (or weights sharing) and equivariant (or invariant) representation. Unlike some neural networks where every input neuron is connected to every output neuron in the subsequent layer, CNN neurons have sparse connections, meaning that only some inputs are connected to the next layer. By having a small, local receptive field (i.e., the area covered by the filter per stride), meaningful features can be gradually learnt, and the number of weights to be calculated can be drastically reduced, increasing the algorithm's efficiency. In using each filter with its fixed weights across different positions of the entire image, CNNs reduce memory storage requirements. This is known as parameter sharing. This is in contrast to a fully connected neural network where the weights between layers are more numerous, used once and then discarded. Parameter sharing results in the quality of equivariant representation to arise. This means that input translations result in a corresponding feature map translation. The convolution operation is defined by the $*$ symbol. Convolution layer is the primary layer to extract the features from a source image and maintains the relationship between pixels by learning the features of image by employing tiny blocks of source data. It's a mathematical function which considers two inputs like source image $I(x, y, d)$ where x and y denotes the spatial coordinates i.e., number of rows and columns. d is denoted as dimension of an image (here d=3 since the source image is RGB) and a filter or kernel with similar size of input image and can be denoted as $F(k_x, k_y, d)$..

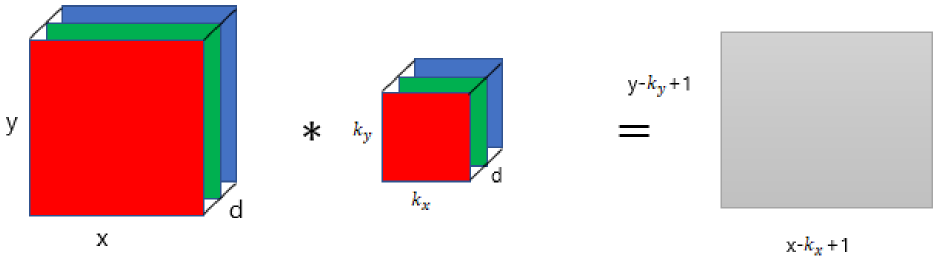
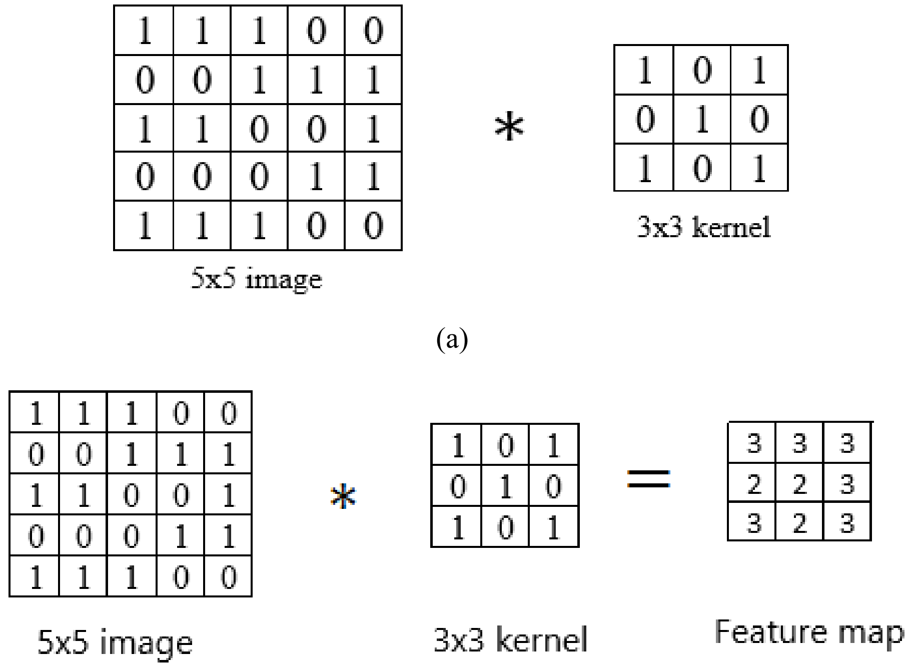


Figure 2: Representation of convolution layer process.

The output obtained from convolution process of input image and filter has a size of $C\left((x - k_x + 1), (y - k_y + 1), 1\right)$, which is referred as feature map. Let us assume an input image with a size of 5×5 and the filter having the size of 3×3. The feature map of input image is obtained by multiplying the input image values with the filter values.



(a)

(b)

Figure 3: Example of convolution layer process (a) an image with size 5×5 is convolving with 3×3 kernel (b) Convolved feature map.

ReLU layer: Networks those utilizes the rectifier operation for the hidden layers are cited as rectified linear unit (ReLU). This ReLU function $\mathcal{G}(\cdot)$ is a simple computation that returns the value given as input directly if the value of input is greater than zero else returns zero. This can be represented as mathematically using the function $max(\cdot)$ over the set of 0 and the input x as follows:

$$\mathcal{G}(x) = \max\{0, x\}$$

Max pooing layer: This layer mitigates the number of parameters when there are larger size images. This can be called as subsampling or down sampling that mitigates the dimensionality of every feature map by preserving the important information. Max pooling considers the maximum element form the rectified feature map.

Fully Connected Layer: The final layer in a CNN is the Fully Connected Layer, meaning that every neuron in the preceding layer is connected to every neuron in the Fully Connected Layer. Like the convolution, RELU and pooling layers, there can be 1 or more fully connected layers depending on the level of feature abstraction desired. This layer takes the output from the preceding layer (Convolutional, RELU or Pooling) as its input, and computes a probability score for classification into the different available classes. In essence, this layer looks at the combination of the most strongly activated features that would indicate the image belongs to a particular class. For example, on histology glass slides, cancer cells have a high DNA to cytoplasm ratio compared to normal cells. If features of DNA were strongly detected from the preceding layer, the CNN would be more likely to predict the presence of cancer cells. Standard neural network training methods with backpropagation and stochastic gradient descent help the CNN learn important associations from training images.

4. Results and description

Figure 4 depicts a visual representation of the GUI of the people counting system described in the project. It showcases how the application's main window looks to the user. This includes buttons, labels, and other graphical elements that allow users to interact with the system. It provides an overview of the user interface's design and layout.

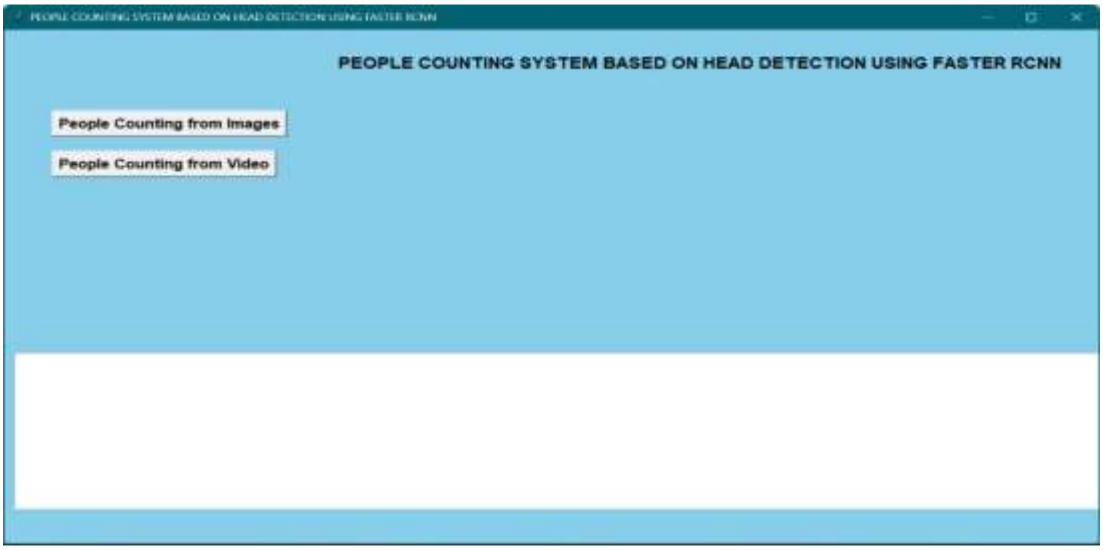

PEOPLE COUNTING SYSTEM BASED ON HEAD DETECTION USING FASTER RCNN
PEOPLE COUNTING SYSTEM BASED ON HEAD DETECTION USING FASTER RCNN
People Counting from Images
People Counting from Video

Figure 4: Illustrating the main GUI application of proposed people counting system based on head detection.

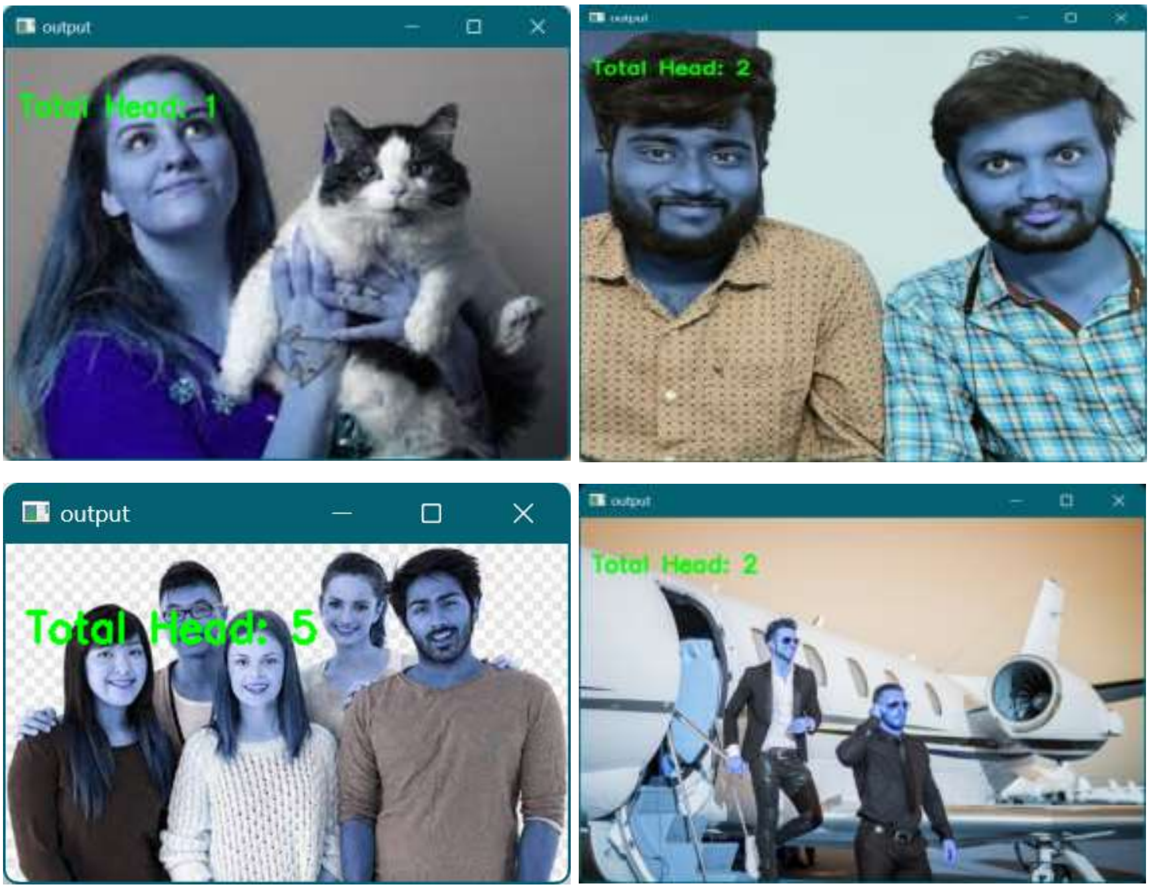


Figure 5: Sample prediction on test images using proposed FRCNN model.

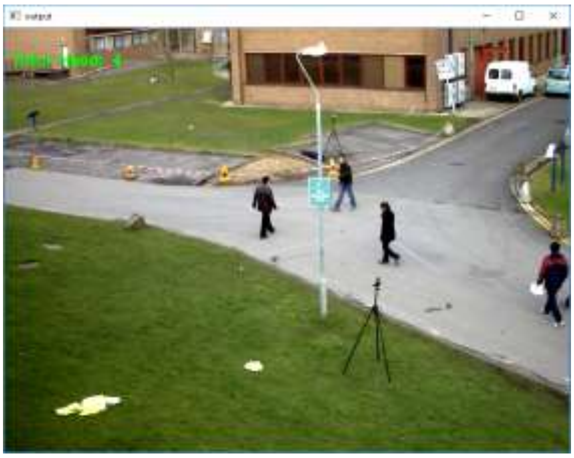


Figure 6: Sample prediction on test video using proposed FRCNN model.

5. Conclusion

The People Counting System presented in this project successfully uses a pre-trained Faster R-CNN model for detecting and counting human heads in both images and videos. The system incorporates a user-friendly graphical interface, making it accessible to individuals with varying levels of technical expertise. Real-time processing capabilities enhance its utility, particularly for applications like crowd management, security surveillance, and occupancy analysis.

References

[1] M. Andriluka, S. Roth, B. Schiele, Pictorial structures revisited: People detection and articulated pose estimation, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1014–1021.

[2] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 1, 2005, pp. 886–893.

[3] D.T. Nguyen, W. Li, P.O. Ogunbona, Human detection from images and videos: A survey, Pattern Recognit. 51 (2016) 148–175.

[4] S. Li, M.O. Tezcan, P. Ishwar, J. Konrad, Supervised people counting using an overhead fisheye camera, in: 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2019, pp. 1–8.

[5] G. Antonini, J. P. Thiran, Counting pedestrians in video sequences using 330 trajectory clustering, Circuits and Systems for Video Technology, IEEE Transactions on 16 (8) (2006) 1008–1020.

[6] I. S. Topkaya, H. Erdogan, F. Porikli, Counting people by clustering person detector outputs, in: Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on, IEEE, 2014, pp. 313–318.

[7] V. Lempitsky, A. Zisserman, Learning to count objects in images, in: Advances in Neural Information Processing Systems, 2010, pp. 1324–1332.

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

[9] P. Morerio, L. Marcenaro, C. S. Regazzoni, People count estimation in small crowds, in: Advanced video and signal-based surveillance (AVSS), 2012 IEEE Ninth International Conference on, IEEE, 2012, pp. 476–480.

[10] H. Idrees, I. Saleemi, C. Seibert, M. Shah, Multi-source multi-scale counting in extremely dense crowd images, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 2013, pp. 2547– 2554.

[11] C. Zhang, H. Li, X. Wang, X. Yang, Cross-scene crowd counting via deep convolutional neural networks, in: Proc. CVPR, 2015.

[12] W. Ouyang, X. Wang, Joint deep learning for pedestrian detection, in: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE, 2013, pp. 2056–2063.

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

[14] S. Wang, J. Zhang, Z. Miao, A new edge feature for head-shoulder detection, in: Image Processing (ICIP), 2013 20th IEEE International Conference on, IEEE, 2013, pp. 2822–2826.

[15] T. Van Oosterhout, S. Bakkes, B. J. Kr¨ose, Head detection in stereo data for people counting and segmentation., in: VISAPP, 2011, pp. 620–625.

[16] K.-E. Aziz, D. Merad, B. Fertil, N. Thome, Pedestrian head detection and tracking using skeleton graph for people counting in crowded environments., in: MVA, 2011, pp. 516–519.

[17] V. B. Subburaman, A. Descamps, C. Carincotte, Counting people in the crowd using a generic head detector, in: Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on, IEEE, 2012, pp. 470–475.

[18] F. Conti, A. Pullini, L. Benini, Brain-inspired classroom occupancy monitoring on a low-power mobile platform, in: Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on, IEEE, 2014, pp. 624–629.