Crowd Counting in Low-Resolution Crowded Scenes Using Region-Based Deep Convolutional Neural Networks

MUHAMMAD SAQIB[1], SULTAN DAUD KHAN[2], NABIN SHARMA[1], AND MICHAEL BLUMENSTEIN[1]

[1]Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, School of Software, University of Technology Sydney, Ultimo, NSW 2007, Australia
[2]University of Hail, Ha'il 2440, Saudi Arabia

Corresponding author: Muhammad Saqib (muhammad.saqib@student.uts.edu.au)

ABSTRACT Crowd counting and density estimation is an important and challenging problem in the visual analysis of the crowd. Most of the existing approaches use regression on density maps for the crowd count from a single image. However, these methods cannot localize individual pedestrian and therefore cannot estimate the actual distribution of pedestrians in the environment. On the other hand, detection-based methods detect and localize pedestrians in the scene, but the performance of these methods degrades when applied in high-density situations. To overcome the limitations of pedestrian detectors, we proposed a motion-guided filter (MGF) that exploits spatial and temporal information between consecutive frames of the video to recover missed detections. Our framework is based on the deep convolution neural network (DCNN) for crowd counting in the low-to-medium density videos. We employ various state-of-the-art network architectures, namely, Visual Geometry Group (VGG16), Zeiler and Fergus (ZF), and VGGM in the framework of a region-based DCNN for detecting pedestrians. After pedestrian detection, the proposed motion guided filter is employed. We evaluate the performance of our approach on three publicly available datasets. The experimental results demonstrate the effectiveness of our approach, which significantly improves the performance of the state-of-the-art detectors.

INDEX TERMS Deep convolutional neural networks, crowd counting and density estimation, Motion Guided Filter, faster R-CNN.

I. INTRODUCTION

Crowd scene understanding is an important and challenging problem in computer vision. The phenomenon of crowd is commonly observed in sports, festivals, social, political and religious gatherings which tends to attract and gather a huge number of people in a constrained environment. Such mass gatherings pose serious challenges to crowd safety and raise security concerns for the participant as well as organizers. Therefore, crowd analysis is one of most important and challenging task in video surveillance due to complex behavior of pedestrians. Crowd analysis can be used for detecting critical crowd levels, detecting and counting of people and also for detecting anomalies in crowded scenes. Moreover, it can be used for tracking individuals or group of people in crowds. Among these applications, estimating the number of people from a single image becomes extremely important for crowd control and crowd safety. In public gatherings, it is important to know the number of people attending the event which can provide useful piece of information for future event planning and public space design.

Traditionally regression-based techniques are extensively used for crowd counting and density estimation. However, recent advancement in deep learning has shown outstanding results in detection using CNN. In a typical CNN based approach, there is local connectivity of a region in the input image to the output image as compared to the traditional feedforward neural network. In a feedforward neural network, every input layer is fully connected with the output layer. Deep CNN is a compositional model, in which features are extracted ranging from low-level to high-level along the pipeline of CNN towards the final layers. The lower layers represent low-level features such as edges, and the subsequent layers represent abstract features such as shapes, etc. We have used Faster R-CNN for the detection of pedestrians in the low-to-medium density crowd videos. In Faster-RCNN [54], a small network namely Region Proposal Network (RPN) is used on top of the feature-map to extract object candidates or region proposals in contrast to other approaches like Selective Search [65], CPMC [7], MCG [1], Edge boxes [85], etc. The advantage of RPN make Faster R-CNN an end-to-end pipeline for the detection and also does not add to the computation of the network. To detect the objects at multiple scales, region proposals at various scale and aspect ratios are extracted using anchor boxes. The anchor boxes of different aspect ratios and sizes are considered to capture scale variation. The center of the anchor box coincides with the center of the sliding window.

The models mentioned above achieved a considerable improvement in object detection in particular and pedestrian detection in general when applied to static images. However, the performance of a detector on videos is limited due to the following reasons;

- Pedestrians in videos pass through a wide range of variations in pose, clothing, lighting and occlusions. This wide range of intra-class variability has a negative effect on the detector's performance. In some cases, the detector missed detection for a particular person in subsequent frames of video. In other cases, the detector ends up with many false positives which results in low recall rates and high Mean Absolute Error (MAE) of a detector.
- CNN based detectors are designed to learn features from raw image pixels and cannot leverage the temporal information existed across the frames of the video.

In this paper, we propose an approach to estimate crowd count by improving the detection performance of a generic detector when applied to videos. Compared with the existing methods, the main contributions of this paper are as follows:

- We leverage temporal information between the subsequent frames of video by proposing Motion Guided Filter (MGF), which utilizes energy function to estimate the displacement vector based on brightness, gradient constancy and spatio-temporal smoothness.
- We utilize MGF and propose a refinement algorithm 1 for low-level tracking that exploits temporal correspondence and suppresses false alarms.
- We recover missed detection by allowing the tracker to operate in two modes: 1) *detection mode*, 2) *low-level tracking mode*.
- We evaluate our approach on three datasets, PETS2009 [22], UCSD dataset [12] and Mall dataset [14]. From experimental results, we observe that the performance of a generic detector is improved by incorporating temporal information.

Fig 1 and 3 show the effectiveness of our approach. We first apply state-of-the-art object detection techniques to detect people in low-density crowds, and then the performance of a detector is improved by leveraging the spatio-temporal information between the frames of video. In general, our method takes predicted detection of a detector as input and generates refined detection as an output with higher recall rate.

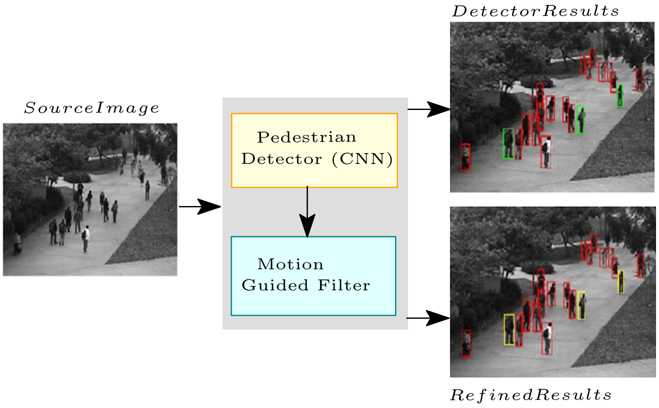


FIGURE 1. The performance of a detector is improved by employing our approach as depicted in the Fig 1. The left image is the input sample frame. The upper right image shows the pedestrians in red bounding box are detected by the detector while the pedestrians in green bounding boxes are the missed detection. The lower right image shows that the missed pedestrians are recovered by our proposed approach and highlighted in yellow color. (Best viewed in color).

The remainder of the paper is organized as follows. In Section II, a comprehensive overview of literature is discussed. In Section III, we present the proposed methodology. The detail analysis of experiment is presented in Section IV followed by a conclusion in Section V.

II. RELATED WORK

Estimating crowd density or estimating the number of people attending the event can substantially reduce the cost by deploying an exact number of security personnel required for public safety and security. Various methods for estimating the crowd count are proposed in literature. Generally, we can classify these methods into two major categories, 1) *Regression based methods*, 2) *Detection based methods*.

A. REGRESSION BASED METHODS

These employ machine learning techniques like Support Vector Regressor [72], Gaussian Process Regression(GPR) [8], linear regression [17], K-Nearest Neighbor [78], and neural network [47] are employed to estimate the crowd count by performing regression between the image features and crowd size. Regression based crowd counting algorithms can be further categorized into two groups: *Holistic* and *Local*.

In *holistic approaches*, image features like size, shape, edges, keypoints, and texture are extracted from the entire image and regression is then applied to estimate the size of crowd. In [8], edge and texture features are extracted from the whole image, and the correspondence between the number of people and features is learned through Gaussian Process Regression. Reference [44] extracts shape, color, size, texture features from the image and self-organizing neural network is employed for crowd density estimation. The neural network is employed by [28] to learn the correspondence between the foreground pixels extracted from the whole image and number of people. A method is proposed by [74] that transforms an image into multiple scales using wavelet transform and then the first and second order features are extracted a as density character vector. A Support Vector Machine classifier is trained that classify density character into different density levels. In [37], edge and blob size histogram features are extracted and neural network is trained to find the relationship between the number of people and extracted features. Reference [35] proposed crowd flow segmentation as the first step and then applying counting framework to count the number of people in each flow segment.

In *local approaches*, image features are extracted from the local patches of image and regression is applied locally to each patch of image and estimate the number of people in each patch. In this case, crowd count is the direct sum of these local estimates. The size and shape features are extracted from the local patches of the image [36] and linear (cylinder model) is employed to estimate crowd count. Reference [39] proposed pixel based density function for counting problem. The density function is a mapping between the feature vector associated with every single pixel value and its ground-truth density value. The ground-truth density value of the pixel is approximated by fitting the normalized Gaussian kernel to the pixel dotted annotation. The multi-output regression model is proposed by [14] for crowd counting by extracting size, shape, edge and texture features from the local regions of the image. Their proposed regression model is able to estimate people count in spatially localized regions. Reference [15] extracts SURF features from the local region of the image and support vector regression are employed to learn the correspondence between the features and the count of people. Reference [32] proposed a counting framework based on multi-source multi-scale approach, which used multiple features extracted from the local regions of an image. These features are taken from different sources like HOG, Local Binary Pattern (LBP) and Fourier analysis. These features are computed at different scales for accurate and reliable counting. This work is extended by [4] which added more features like wavelets, SIFT, and GLCM. Reference [45] proposed a local Histogram-of-Orientation Gradient, in contrast to the standard Histogram-of-Orientation-Gradient, used to describe the parts of the person independently and therefore helps in extraction of features even in partial occlusions.

A great deal of work in [2], [9], [12], and [56] for crowd counting and density estimation has focused on local features such as edges and blobs extracted from the foreground. Typically, regression techniques such as Ridge Regression (RR) [14], Bayesian Poison Regression (BPR) [12], Gaussian Process Regression (GPR) [9] are used to learn the model between the local features and count. However, a significant amount of information is lost in the calculation of such features. The accuracy and performance of such features heavily rely on the segmentation of foreground. The foreground segmentation is a challenging problem especially because of varying lighting conditions and shadowing effect [66]. Reference [57] considered texture features that are directly related to the crowd density and counting. The more texture means high crowd density which is not always true because of the incorrect foreground segmentation. Foreground segmentation of crowd only caters for moving crowd, but it performs poorly in the case of a static or very slow moving crowd. Furthermore, these features cannot be used for contextual crowd scene understanding. Reference [23] proposed the simplified version of the previous work by estimating the object density using regression random forest improving the training accuracy. Similarly, [3] proposed an interactive and iterative density estimation technique. In this technique, user annotates the object with the dot for object and line segment for its diameter to estimate the density. The low-level features extracted are mapped to the density value. The learned mapping can be visualized intuitively by the user for error. The error indicates the need for further annotations to refine results in the next iteration.

The performance of regression-based methods are improved further by employing Convolution Neural Networks (CNN) [5], [34], [48], [49], [59], [69], [82]. In these methods, density maps are generated from the image patches, where count for each patch is obtained by performing the integration over the density map. Zang *et al.* [77] proposed a CNN model which can generate both crowd count and density maps using switchable alternative learning for counting and density map. The training and testing require a perspective map for perspective normalization which might not be available in practice. A Multi-column Convolutional Neural Network (MCNN) is proposed in [82], which utilizes three columns with filter size of a different receptive field is used to compensate for perspective distortion. MCNN is trained to estimate crowd density at only three different scales in extremely crowded still images. Boominathan *et al.* [5] and Zhang *et al.* [82] proposed multicolumn CNN approaches, in which different columns with different filter sizes are used to capture multiple scales variation along with perspective. The final prediction obtained from columns is averaged to get a density map. Finally, the integral of density map gives crowd count. Similarly, Switch-CNN [59] proposed switching architecture which intelligently switches appropriate regressor for particular crowd patch based on variation in density within the single image. However, these approaches are highly scene specific and may perform poorly on cross scene analysis. Reference [69] used shallow CNN architecture in the framework of ensemble learning where new models are added to the ensemble to fix the error from the previous model. These ensembles were used to estimate the density map. The density map is then spatially integrated to count. However, the research did not clearly explain the stopping criteria for adding new models to the ensemble. Therefore, ultimately adding new models to the ensemble might end up in overfitting. Reference [34] used contextual information such as perspective weights, camera tilt angle and camera height to estimate crowd density. The contextual information is an auxiliary input to the Filter Manifold Network (FMN) to produce filter weights for the convolutional layer according

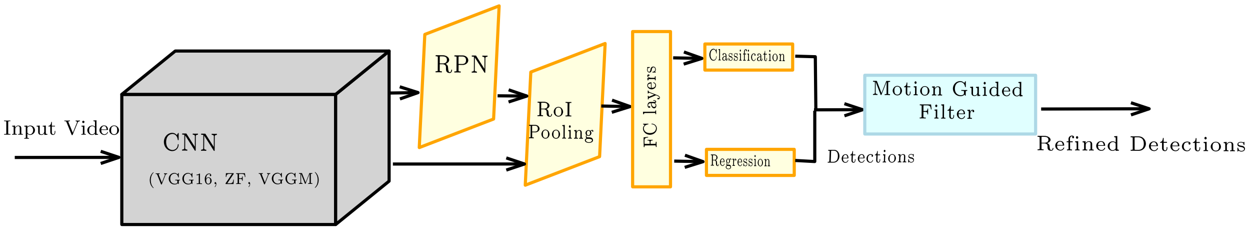


FIGURE 2. Proposed framework.

to the scene context. Thus convolutional layer adapts itself to the context of the scene in contrast to the fine-tuning and transfer learning. The current datasets do not provide contextual information, therefore, comparison with current datasets is not possible. The convolution is made adaptive to only parameters related to perspective. There might be more complex parameters related to the scene which are ignored.

Most recently, Shen *et al.* [61] proposed Adversarial Cross-Scale Consistency Pursuit (ACSCP) approach using adversarial loss instead of traditional Euclidean loss to mitigate the blurry effect due to l_2 regularization in the generation of density maps from crowd patches. Moreover, a new regularize is proposed to enforce the scale consistency such that the number of crowd count in the large patch is coherent with the sum of crowd counts in the corresponding smaller non-overlapping patches. Similarly, Cao *et al.* [6] proposed Scale Aggregation Network (SANet) for accurate and efficient high-resolution of density maps using new training loss called local pattern loss. Sindagi and Patel [63] proposed Contextual Pyramid CNN (CP-CNN) in which local and global contextual information is incorporated with Density Map Estimator (DME) to generate high-quality density maps. Finally, all the maps are fused to estimate the crowd count and density. Crowd counting is formulated as a semantic scene model [30]. The pedestrian, head, and their context are three key factors, that are considered as a composite body-part semantic structure for two types of scene semantic models. These models are turned into different sub-tasks to train deep CNN for counting and scene semantic analysis. Xiong *et al.* [75] proposed Convolutional LSTM (convLSTM) to exploit temporal correlation along with spatial dependencies to boost the count accuracy in a complex scene. However, most of the datasets of the high-density crowd are still images of the crowd and therefore do not carry temporal information.

Regression-based methods work well in high-density situations since they can capture generalized density information but suffer from following limitations. 1) The performance of these methods degrades when applied to low-density situations due to overestimating the count. 2) These methods cannot localize pedestrian in the scene and thus provide no information about the distribution of pedestrians in the environment which is sometimes very crucial for the crowd managers and security personnel.

B. DETECTION-BASED METHODS

On the other hand, *detection based methods* [16], [21], [24], [38], [68], [83], train object detectors to localize the position of each person, where crowd count is the number of detections in the scene. Detection based methods can be further divided into two categories, 1) hand-crafted feature based models [16], [19]–[21], [68], [70], [80] and 2) deep features models [29], [31], [40], [43], [46], [50], [51], [64], [71], [79], [81]. In the first category, hand-crafted features like edges [10], [11], texture [10], [60], [74], and shape [10] are extracted from image to train SVM or boosting classifiers. After training, learned weights of the classifier are considered as a template for the entire human body. These hand-crafted features have low representation of human body and performance of classifier degrades when applied to complex crowded scenes. In order to model complex poses of pedestrians, DPM [21], [41], [84] learn mixture of local templates for each body part. Although DPM is robust to complex poses but feature representation and classifier cannot be jointly optimized to improve performance.

III. PROPOSED METHODOLOGY

We proposed a framework for crowd counting using state-of-the-art deep CNN as shown in Fig. 2. According to the framework, input frames are given to the Region-based CNN. We have used Faster R-CNN [54] with Caffe [33] for the detection of pedestrians. The datasets that we have used contain few numbers of frames which are not sufficient enough for training deep CNN. Therefore, we have used transfer learning from ImageNet [18] to fine-tune our models. These fine-tuned models are used in the testing phase to test on unseen frames. We have used various network architectures such as ZF [76], VGG16 [62], and VGGM [62] to train the system and evaluate the performance on the test dataset. ZF is a 8 layered architecture containing 5 convolutional layers and 3 fully-connected layers. Similarly, VGG16 is a 16 layered architecture that has 13 convolutional layers and 3 fully connected layers.

In Fast R-CNN [25] the order of the extracting region of proposals and running the CNN is exchanged as compared to RCNN [26] architecture. In this architecture whole image is passed once through the CNN and the regions are now extracted from convolutional feature map using ROI pooling.

This change in architecture reduces the computation time by sharing the computation of convolutional feature map between region proposals. The region proposal is projected to the corresponding spatial part of convolutional feature volume. Finally, the fully connected layer expects the fixed-size feature vector, and therefore the projected region is divided into a grid and Spatial Pyramid Pooling (SPP) is performed to get fixed-size vector. SPP deals with the variable window size of pooling operation and thus end-to-end training of the network is very hard. The generation of the region proposals is the bottleneck at the test time. In the above-mentioned approaches, CNN was used only for regression and classification. The idea was further extended to use CNN also for region proposals. The latest offspring from the RCNN family, the Faster R-CNN [54] proposed the idea of a small CNN network called Region Proposal Network (RPN), build on top of the convolutional feature map. RPN is two-layered network which does not add to the computation of overall network. A sliding window is placed over a feature map in reference to the original image. The notion of anchor box is used to capture object at multiple scales. The center of the anchor box having a different aspect ratio and size coincide with the center of the sliding window. RPN generates region proposals of different sizes and aspect ratios at various spatial locations. Finally, regression provides finer localization with reference to the sliding window position. The complete architecture is shown in Fig. 2.

A. MOTION GUIDED FILTER

Convolution neural networks like SSD [42], YOLO [53], and Squeezedet [73] showed a significant improvement in domain of real-time object detection using a single image. However, the performance of these networks can be improved further by leveraging the temporal information available in real-time videos. Leveraging temporal information in object detection is not a trivial problem. Usually, end-to-end learning is a sophisticated way of solving computer vision problems, but in the case of videos, this approach cannot be applied. Feeding multiple frames to the CNN is not possible due to the limitation of memory. Therefore, as a solution, we propose a Motion Guided Filter that recovers the missed detection in the frames by using the flow estimation. It is observed that pedestrian detected in the first frame travels a few pixels in the next frame. For estimating the displacement, we utilize an energy function which is based on three assumptions: brightness constancy assumption, gradient constancy, and spatio-temporal smoothness constraint.

1) BRIGHTNESS CONSTANCY

For estimating the displacement, it is assumed that the gray value of a pixel does not change [27]

$$\Omega(x, y, t) = \Omega(x + u, y + v, t + 1) \tag{1}$$

Ω: $\lambda \subset \mathbb{R}^3 \to \mathbb{R}$ denotes bounding box sequence, and $w := (u, v, 1)$ is the displacement vector between an image at time t and another image at time $t + 1$. Here it is to be noted that bounding box Ω is a 4-D vector. For estimating the displacement, we use only the spatial coordinates of pixels while we assume the size of the bounding box (width and height) is the same. Therefore we omit the size of the bounding box in the equations.

2) GRADIENT CONSTANCY

It is also assumed that the gray value of the pixel does not change instantaneously. However, this assumption is weak since a slight change in the environment or change in illumination may change gray values of the image, therefore, in this case, we allow some small variations and determine the displacement vector by a criterion that is invariant to gray value changes. We, therefore, use the gradient of gray value instead of considering the gray values directly. We then assumed that gradient of gray value does not vary due to the displacement [67] and is given by

$$\nabla\Omega(x, y, t) = \nabla\Omega(x + u, y + v, t + 1) \tag{2}$$

where ∇ is the gradient. Equation (2) deals with translatory motion while (1) is best suited for complicated motions.

3) SPATIO-TEMPORAL SMOOTHNESS

Up till now, the model estimates the displacement of one pixel from one frame to another without taking into account neighboring pixels. Therefore, the model runs into problems as soon as the gradient disappears somewhere. Furthermore, we also expect some outliers in the estimates. Therefore it is very useful to use smoothness assumption. This constraint can be either applied only in the spatial domain if we want to compute flow between two images or to the spatio-temporal domain, if the displacement in the whole sequence of images is needed. Here, since we are recovering the bounding box of the next frame from the current frame, therefore we use only spatial smoothness constraint.

With this discussion, we now derive an energy function that will penalize deviations from these aforementioned assumptions. Let $x := (x, y, t)$ is the pixel of frame at t and $w := (u, v, 1)$ is its displacement vector. Then deviations from the grey value constancy and gradient constancy are measured by the following energy function

$$E_d = \int_\lambda (\mid \Omega(x+w) - \Omega(x) \mid^2 + \gamma \mid \nabla\Omega(x + w) - \nabla\Omega(x) \mid^2) dx \tag{3}$$

where γ is a balancing parameter between brightness and gradient constancies.

Finally, we write the smoothness term which penalizes the total variations in the flow field [55] and can be expressed as

$$E_s = \int_\lambda (\mid \nabla u \mid^2 + \mid \nabla v \mid^2) dx \tag{4}$$

The total energy function is the weighted sum of the above two equations and is given by

$$E(u, v) = E_d + \alpha E_s \tag{5}$$

with some regularization parameter the value of $\alpha > 0$. The goal is to find displacement vector (u, v) that minimizes the energy function given by (5)

For every pixel $x_t \in \Omega_t$ in a frame at t, we compute its corresponding pixel in a frame at $t+1$ by using the following equation

$$x_{t+1} = x_t + w \tag{6}$$

4) DETECTION REFINEMENT

After detecting pedestrians in each frame of the analyzed video sequence, we then leverage temporal information across the multiple frames to further refine the detection results. For this purpose, we use (6) for low-level tracking to establish temporal correspondence across multiple frames. Exploiting temporal information can suppress false alarms generated due to the noise and other random distortions. We integrate temporal information across the frames to re-score detections and suppress the false positives. Let $\Omega_t = \{\omega_1, \omega_2, \ldots, \omega_n\}$ represents a set of n bounding boxes (or detections) in a frame at t. We then represent $D = \{\Omega_1, \Omega_1, \ldots, \Omega_N\}$ as a container of all sets of bounding boxes for a video sequence containing N number of frames. In order to refine Ω_t for the current frame at t, we employ a matching hypothesis based on overlap area between the current bounding box $\omega_i \in \Omega_t$ and $\omega_j \in \Omega_{t+1}$ in the subsequent frame. We propose a refinement Algorithm 1 which takes Ω_t as input and gives the corresponding refined Ω_R as an output. Given a set of bounding boxes Ω_t in the current frame, we define a temporal window of the size W. For each bounding box $\omega_i \in \Omega_t$ in a frame at t, we first predict its location in the next frame at $t+1$ by computing displacement vector w as in (6). We then compute Δ between ω_i and set of bounding boxes Ω_{t+1} in the next frame at $t+1$ and select the best match (maximum value of Δ). We compute Δ between two detections ω_i and ω_j as Intersection over Union and formulated as $\frac{\omega_i \cap \omega_j}{\omega_i \cup \omega_j}$. Final confidence score σ is computed for each ω_i by accumulating confidence score over temporal window W, as in line 8 of the Algorithm 1. We then delete the bounding box for which confidence score σ is less than ϵ. We set the value of $\epsilon = 0.5$ in all our experiments. We refine the container D in the same way. Let $R = \left\{\Omega_1{}', \Omega_2{}', \ldots, \Omega_N{}'\right\}$ is a container of refined sets of bounding boxes for a video sequence containing N frames. The bounding boxes obtained after this step are refined and trusted detections.

In some cases, a given set of bounding boxes Ω_t may not contain a detection for a particular person due to occlusion, or missed detection, etc. In order to address this issue, we integrate temporal information by reliably tracking pedestrian through time and use it to find the missed detection. Our tracking approach operates in two modes: 1) *detection mode*, 2) *Low-level tracking mode*. We initialize a tracker for each detection in a frame at t. Whenever the tracker finds and matches a detection in the next frame at $t+1$, it follows the detection mode. This mode enables tracking more robust to variations in scale, appearances and pose.

Algorithm 1 Refinement of Detection Results

Input: Sets of Bounding Boxes Ω_t

Output: Refined Bounding Boxes Ω_R

```
 1: function Refinement($\Omega_t$)
 2:     $T = \{\Omega_{t+1}, \Omega_{t+2}, \ldots, \Omega_{t+W}\}$
 3:     Initialize evidence accumulator $\sigma$ to zero
 4:     for each bounding box $\omega_i$ in $\Omega_t$ do
 5:         for each $\Omega_j$ in $T$ do
 6:             Compute displacement vector $\mathbf{w}$ using (5)
 7:             Predict next location $\omega_i{}'$ as $\omega_i + \mathbf{w}$
 8:             $\sigma = \sigma + \arg\max_{j \in T} \Delta(\omega_i{}', \Omega_j)$
 9:             Update $\omega_i$ as $\omega_i \longleftarrow \omega_i{}'$
10:         end for
11:         if $\frac{1}{W}\sum_1^W \sigma > \varepsilon$ then
12:             Insert $\omega_i$ in tail of $\Omega_R$
13:         end if
14:     end for
15:     return $\Omega_R$
16: end function
```

If for some reasons, tracker cannot find detection in the next frame, the tracker relies on low-level tracking. In low-level tracking mode, the tracker estimates the displacement vector and predicts the next location by using (5) and (6). It is to be noted that we are not interested in long-range tracking, instead our goals is to use low-level tracking to fill in the gap by recovering the missed detection. Let $\{x_t, s_t\}$ be the position and size of a pedestrian being tracked. Let $\ddot{x_t}$ and $\ddot{s_t}$ are the observations of x_t and s_t, with Gaussian noises of co-variance R_x and R_s. For each track in a frame at t, we have predictions $\left\{\hat{x}_{t|t-1}, \hat{s}_{t|t-1}\right\}$ and we search for pedestrian detection around position $\hat{x}_{t|t-1}$ and size $\hat{s}_{t|t-1}$. For any pedestrian detection P will be assigned to a track if $\parallel \hat{x}_{t|t-1} - x_f \parallel < \alpha$ and $\parallel \hat{s}_{t|t-1} - s_f \parallel < \alpha$, where x_f is the position and s_f is the size of P. We set $\alpha = 0.3$ in our experiments. We adopt a greedy strategy of data association and score each track by N_d, where N_d represents the number of detections it has matched. During the detection mode, we maintain a pedestrian template $P_{template} = I(x_t, s_t)$ at location x_t and of size s_t. We use normalized correlation to search for best match in the image. In case, a tracker cannot find and match a detection, then tracker switch to low-level tracking mode and continue tracking. In this case we update the template linearly as in (7)

$$P_{template} = (1 - \beta_{update} P_{template} + \beta_{update} I(x_t, s_t)) \tag{7}$$

where β_{update} is set to 0.1 in our experiments. For every track, we keep track of N_d and N_{llt}, where N_d is the number of step that a track follows the detection mode and N_{llt} is the number steps in which track follows low-level tracking mode. We terminate a track if $N_{llt} \; / \; N_d > 1.5$.

IV. EXPERIMENTS

In this section, we discuss the qualitative and quantitative analysis of the results obtained from the experiments.

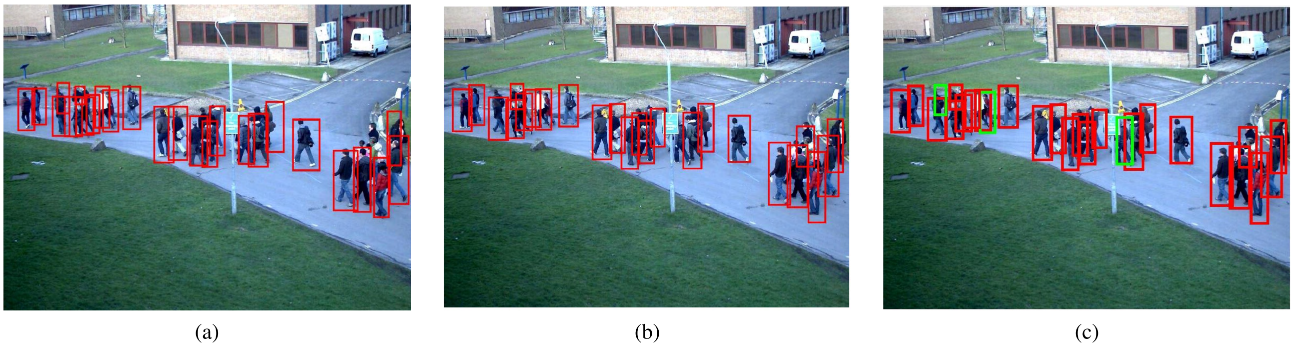


FIGURE 3. (a) Detections in the first frame. (b) Detections in the second frame. (c) Recovered detection missed in second frame.

TABLE 1. Crowd datasets.

Dataset	Resolution	Color	Location	Test frames	Train frames	Crowd Size
PETS2009 [23]	$768X576$	RGB	Outdoor	800	1200	8 to 26
UCSD [12]	$238X158$	Grayscale	Outdoor	800	1200	11 to 50
Mall [14]	$640X480$	RGB	Indoor	800	1200	15 to 60

We evaluate our approach using three publicly available datasets, PETS2009 [22], UCSD dataset [12] and Mall dataset [14]. These datasets include indoor and outdoor scenes with varying densities. Traditionally, regression-based methods are evaluated on these datasets. Therefore the available annotations are only suitable for regression-based analysis and not for detection base methods. Typically, there is a dot annotation for every person in the scene. These annotations also include perspective map used for the normalization of perspective distortion. Such dot annotations are not suitable for training a CNN model for pedestrian detection. Therefore, for the first time, we annotated each pedestrian with a bounding box that covers the whole body of the pedestrian. The complete details of the datasets are given in the Table. 1. The sample images along with overlaid ground-truth annotations from three datasets are shown in the Fig. 7 (d) (e) (f).

After annotating all video sequences, we then trained different models, i.e, *ZF* [76], *VGGM* [62] and *VGG16* [62] on Nvidia Quadro P6000 GPU with a learning rate of 0.0001 and batch size of 64. The RPN batch size is kept constant at 128 for region based proposal networks.

We then evaluate and compare the performance of our method with other reference methods. For the sake of a comprehensive evaluation, we divide the experiment setup into two phases. In the first phase, we evaluate and compare the detection/localization performance while in the second phase, we evaluate and compare the crowd counting performance.

A. LOCALIZATION PERFORMANCE

In this section, we evaluate and compare the localization performance of different models. The purpose of evaluating localization performance is to measure how well the model localized the pedestrian in the given scene. Precise localization of pedestrians is very crucial for the crowd managers and security personnel to effectively respond to the anomalous situations.

The localization accuracy by which a model can predict the bounding box of a pedestrian is typically judged by Intersection over Union (IoU) between predicted and ground-truth. In most of the cases, IoU is used with fixed threshold value 0.5 for deciding whether a bounding box is successfully detected. However, with the fixed threshold value, one cannot overview the range of performance with varying the thresholds. Therefore, we use mean Average Precision (mAP) as an evaluation metric that averaged the performance over a wide range of IoU thresholds.

We evaluate the localization performance of these models in two ways, i.e., pre-trained and fine-tuning. In the pre-training phase, these networks are trained from scratch by using *ImageNet* dataset and then the learned models are directly used for detecting pedestrians during the testing phase. In fine-tuning case, we fine-tuned these pre-trained models by using the images from PETS2009, UCSD and Mall datasets.

We analyzed the performance of each network architecture at a different iteration during the fine-tuning phase. During training, the snapshot of trained models are saved at the interval of $10k$ as shown in the Fig. 4 for Mall dataset [14]. All the network architectures were able to converge after $20k$ iterations. The best-trained model obtained at iteration $90k$ of $VGG16$ having mAP of .701 was used for evaluation on the testing sequence of Mall dataset [14]. Similarly, trained model based on ZF architecture for USCD [12] with high mAP of 0.783 is obtained at $80k$ iteration as shown in Fig. 6. The reason for high mAP can be attributed to

TABLE 2. Localization performance at different steps.

Methods	Pre-Training			Fine Tuning			Proposed		
Models	VGG16	VGGM	ZF	VGG16	VGGM	ZF	VGG16+MGF	VGGM+MGF	ZF+MGF
UCSD	0.45	0.25	0.40	0.67	0.65	0.59	0.73	0.71	0.63
PETS 2009	0.40	0.15	0.20	0.75	0.73	0.69	0.82	0.78	0.75
Mall	0.41	0.25	0.30	0.68	0.63	0.57	0.71	0.65	0.58

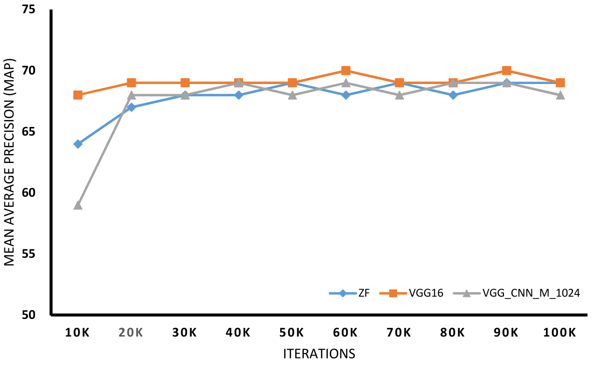


FIGURE 4. Performance at different iteration for Mall dataset [14].

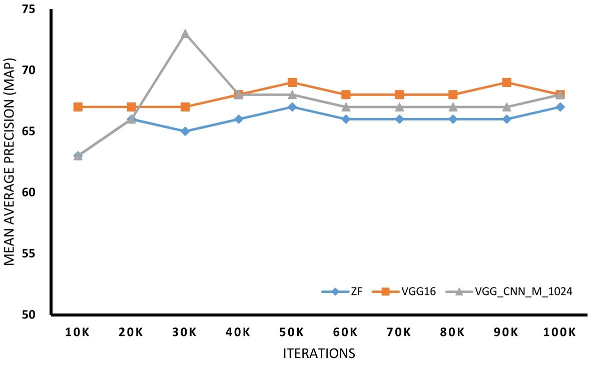


FIGURE 5. Performance at different iteration for PETS dataset [22].

the low-resolution of the dataset as well as the smaller filter size used in *ZF* architecture. Thus *ZF* shows stable performance throughout all the iterations. Furthermore, a *VGG*16 model with high mAP of 0.692 is obtained for PETS2009 dataset [22] as shown in Fig. 5.

After fine tuning the models, we then employ spatio-temporal filtering approach discussed in the section, which further refines the detection by exploiting spatial and temporal information between the consecutive frames.

Table. 2 shows the performance of these models obtained during pre-training, fine-tuning and after employing a Motion Guided Filter. It is obvious from the table that all the base models show poor performance during the pre-trained phase. The reason for poor performance is the models were trained on ImageNet dataset and not on pedestrian datasets. However, these models are generic enough to be fine-tuned for pedestrian detection. We have used 50% samples of pedestrian datasets during fine-tuning phase. The models fairly learn

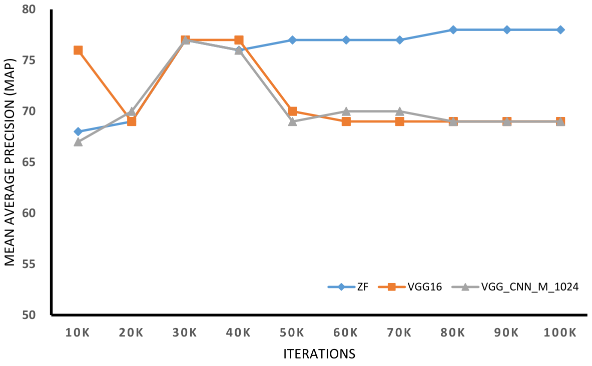


FIGURE 6. Performance at different iteration for UCSD dataset [12].

the representation of pedestrians and as a result, mAP is increased up to 72% on average for all the datasets. Among the fine-tuned CNN models, VGG16 outperforms other methods. Even after fine-tuning, there is still room for improvement since the information like consistent brightness and color pattern of pedestrian existed between the subsequent frames are not exploited. For example, the detector detects pedestrians in one frame while detector completely missed that detection in subsequent frames. Therefore, by exploiting the spatio-temporal relationship and brightness consistency constraint, our proposed Motion Guided Filter is able to recover the detection which are missed due to occlusions. As a result, our proposed methodology is able to improve the mAP for all the models.

B. COUNTING PERFORMANCE

In this section, we evaluate the performance of different crowd counting methods. In addition to CNN based methods, we used four different regression-based models, i.e Gaussian Process Regression [8], linear regression [17], K-Nearest Neighbor [78] (K=4) and neural network [47] with sigmoid activation function for crowd counting. These regression models are trained on local features, i.e., size, shape, edge and keypoints. The features are extracted from the local regions of image by first dividing the image into patches. We then apply a regression technique to each patch of an image. The local features are extracted in the following ways.

Size refers to the area of foreground object. The area of object is measured as the count of foreground pixels. In order to compensate for perspective distortions, we assign weight $W(x, y)$ to each foreground pixel as in [8] based on the

TABLE 3. Evaluation of crowd counting methods.

Methods	Models	UCSD		PETS 2009		Mall	
		MAE	MSE	MAE	MSE	MAE	MSE
Regression(on hand-crafted features)	GPR [8]	1.46	6.23	1.78	16.97	2.58	8.86
	Linear [18]	1.56	6.48	1.77	17.75	2.58	9.65
	KNN [79]	2.72	9.63	3.00	18.69	2.89	9.23
	NN [48]	8.13	33.08	4.11	30.42	26.06	163.41
CNN	VGG16 [63]	2.89	9.25	2.67	18.53	3.52	10.25
	VGGM [63]	3.92	10.47	2.63	17.56	4.85	13.65
	ZF [77]	3.55	11.26	2.64	16.28	3.65	11.46
CNN + MGF (proposed)	VGG16+MGF	1.27	5.62	1.21	5.43	1.89	7.29
	VGGM+MGF	1.38	6.29	1.28	5.66	2.32	8.35
	ZF+MGF	1.41	6.35	1.36	6.48	2.56	8.78

TABLE 4. Comparative analysis with other techniques on UCSD [12] dataset.

Method	**MAE Test**
Density + MESA [40]	1.7
Crowd CNN Model with global regression [78]	1.6
COUNT forest [53]	1.6
CNN Model with no boosting [70]	1.63
Boosted CNN (1 boost) [70]	1.35
Boosted CNN (2 boost) [70]	1.29
Boosted CNN (3 boost) [70]	1.28
Fine-tuned (2 boost) [70]	2.01
Twice as deep [70]	1.82
Thrice as deep [70]	2.42
Ensemble of 2 CNNs [70]	1.55
Ensemble of 3 CNNs [70]	1.53
Zhang2015 [78]	1.60
MCNN [83]	1.07
Hydra-CNN	1.65
Switching CNN [60]	1.62
ConvLSTM [76]	1.30
BSAD [31]	1.0
ACSCP [62]	1.04
SANet [6]	1.02
Proposed Method (VGG16 + MGF)	**1.27**

relative size of reference object in the scene. The weighted area A of blob B is computed as follows

$$A = \sum_{(x,y)\in B} W(x,y)$$

Shape is computed by measuring the orientation of perimeter pixels. Perimeter pixels contain important and useful information about the shape of the object. For computing shape feature, we generate a histogram of orientations with four bins. Each bin corresponds to the orientations of pixels. The four bins correspond to four shape features and denoted by S(h), where h ∈ [1, 4].

Edge is computed by taking the histogram of edge pixels of the foreground object. We divide edge orientation histogram into six bins over the range of [0, 180°]. In this case, for perspective normalization, each edge pixel assigns a weighted vote of $\sqrt{W(x,y)}$ to a corresponding histogram bin h as follows.

$$E(h) = \sum_{(x,y)\in K} \begin{cases} \sqrt{W(x,y)}, & \text{if } \theta_h \leq \theta_{x,y} \leq \theta_{h+1} \\ 0 & \text{otherwise} \end{cases}$$

where K is set of edge pixels of a blob of the foreground object and $\theta_{x,y}$ is the orientation of edge pixel. Upper and lower bound of bin h is represented by θ_h and θ_{h+1} respectively.

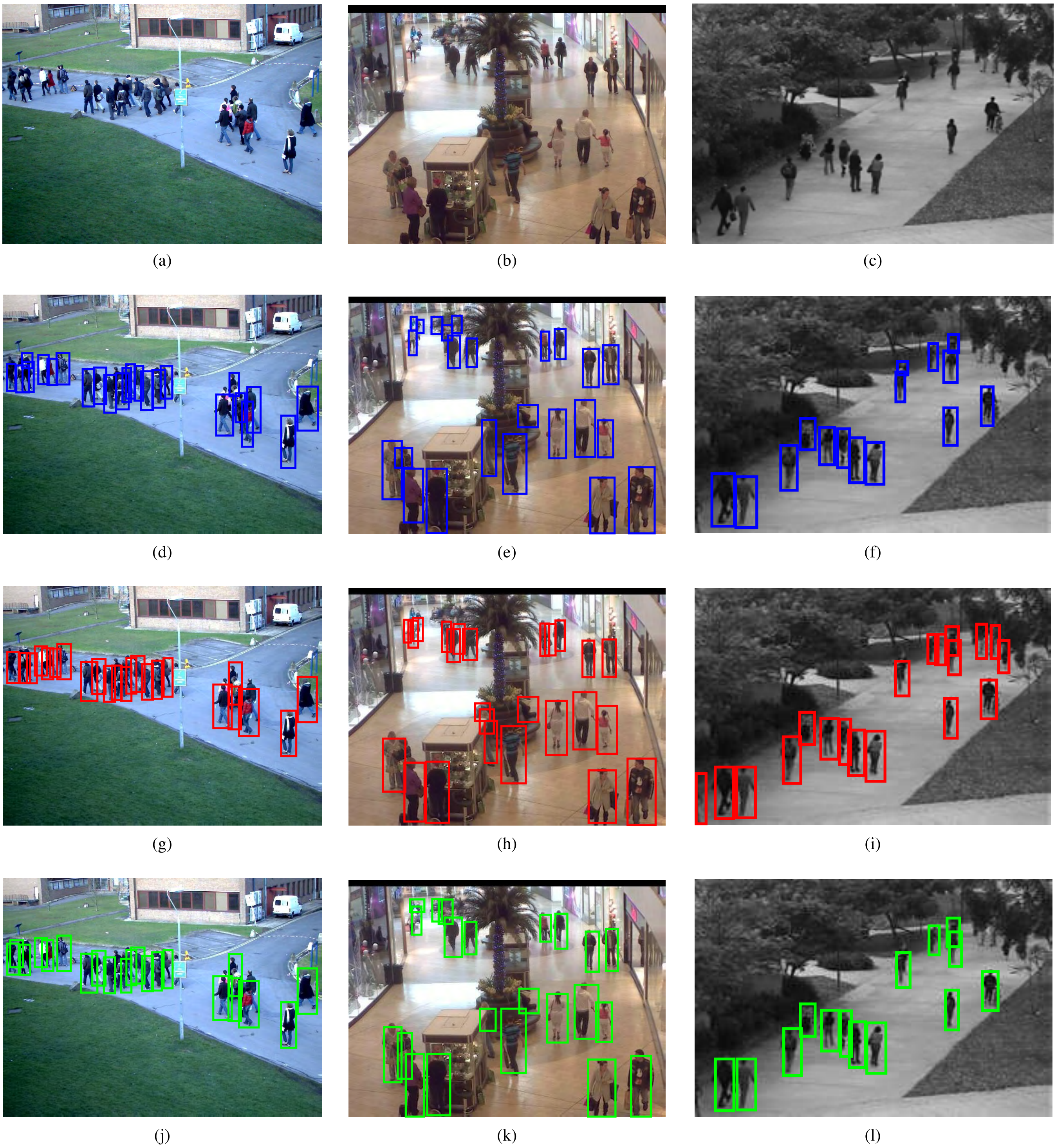
Interest points refers to keypoints in the scene and provide useful information about the human crowding. We extract two types of features, i.e. FAST and SURF from the blob and denoted as P_F and P_S respectively and computed as follows.

$$P_F = \sum_{(x,y)\in M} \sqrt{W(x,y)}$$

$$P_S = \sum_{(x,y)\in N} \sqrt{W(x,y)}$$

where M represents *FAST* features extracted from the blob, and N represents *SURF* features extracted from foreground blobs. In this case, we also assign weights W to each interest point to compensate for perspective distortions.

We train four different regression models using these local features and the results of regression and CNN based methods are reported in Table. 3 in terms of Mean Absolute



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

FIGURE 7. Qualitative results: Row: 1 (a-c) Sample images from datasets (a) PETS2009 (b) Mall dataset (c) UCSD dataset. Row: 2 (d-f) Ground-truth annotations overlaid on samples images. Row 3: (g-i) detection results Row 4: (j-l) Refine results after spatio-temporal smoothing.

Error (MAE) and Mean Square Error (MSE). MAE and MSE are mostly used evaluation measures for counting and formulated as

$$MAE = \frac{1}{|T|}\sum_{t\in T}(\mu_t - G_t)^2 \tag{8}$$

$$MSE = \frac{1}{|T|}\sum_{t\in T}|\mu_t - G_t| \tag{9}$$

where T is the total number of testing frames. While μ_t and G_t are the predicted and ground-truth count of pedestrian respectively in a frame at t. From the Table. 3, it is obvious that regression-based methods perform well than CNN based detection methods. It is attributed to the fact that in high-density situations, regression models perform well in approximating the count by leveraging the rich context in crowded patches while CNN based detection models are unable to localize and detect pedestrians

TABLE 5. Comparative analysis with other techniques on Mall [14] dataset.

Method	**MAE Test**
CA-RR [13]	3.43
COUNT forsest [53]	2.50
CNN Model with no boosting [70]	9.54
Boosted CNN (1 boost) [70]	2.43
Boosted CNN (2 boost) [70]	2.08
Boosted CNN (3 boost) [70]	2.13
Fine-tuned (2 boost) [70]	2.01
Twice as deep [70]	10.41
Thrice as deep [70]	15.37
Ensemble of 2 CNNs [70]	6.52
Ensemble of 3 CNNs [70]	6.57
ConvLSTM-nt [76]	2.53
ConvLSTM [76]	2.24
Bidirectional LSTM [76]	2.10
Proposed Method (VGG16 + MGF)	**1.89**

due to the small size of the head, occlusion, and perceptive distortions. We observed from the experiments that detection based methods provide a reliable estimation in sparse crowds where the pedestrians are fully visible. Based on our experiments and as obvious from the table we find that detection and regression-based counting methods show different performances depending on the densities of the crowd. The regression-based methods provide reliable estimates when applied to congested scenes. However, these methods cannot provide localization information for persons in the scene and tend to overestimate the count when applied in low-density situations. The detection based methods, on the other hand, can localize each person precisely in low dense situations. However, the performance of detection based methods improves in all situations after employing our proposed Motion Guided Filter.

The average time for processing each frame for detection was 0.044 seconds, 0.130 seconds and 0.048 seconds, for ZF, VGG16 and VGG M, respectively. On average the frame was processed in 0.130 sec/frame.

We also compare our method with other crowd counting methods using UCSD dataset, and the results are reported in Table. 4. We use MAE metric as an evaluation measure. From the table, it is obvious that our proposed method outperforms most of the state-of-the-art methods. Our approach out-performs most of the state-of-the-art methods in UCSD data set. However, our approach shows lower performance in comparison to few state-of-the-art methods. The low performance attributes to the following reasons: (1) UCSD data set consists of extremely low-resolution videos. Each video frame has to be re-sized and padded before input to the network. As a result, frame lost most spatial information and become too coarse to describe pedestrians. (2) Faster R-CNN lacks the ability to detect small objects lies in various scales.

TABLE 6. Comparative analysis with other techniques on PETS [22] dataset.

Method	**MAE Test**
Shape+Edges+Keypoints [59]	1.77
Proposed Method(VGG16 + MGF)	**1.21**

In the same way, we compare our method with other methods using Mall and PETS datasets, and the results are reported in Table. 5 and 6 respectively. In this case, we use VGG16 as best model for comparison with other methods. As obvious from tables, our proposed method produced superior results as compared to the state-of-the-art methods.

V. CONCLUSION

In this work, we proposed a framework for counting of crowd in a low-to-medium density crowd videos. The framework use state-of-the-art detector Faster-RCNN to detect pedestrian in crowd video. We the used Motion Guided Filter to recover misdetections and therefore improve mean Average precision of the overall detections. The improvement in the accuracy of detection also lead to the improvement in the counting and density estimation of crowd. The proposed approach can be used easily incorporated in the real-time monitoring and surveillance applications and as well as high-level scene understanding of crowd.

REFERENCES

[1] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 328–335.

[2] M. Arif, S. Daud, and S. Basalamah, "Counting of people in the extremely dense crowd using genetic algorithm and blobs counting," *IAES Int. J. Artif. Intell.*, vol. 2, no. 2, p. 51, 2013.

[3] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Interactive object counting," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 504–518.

[4] A. Bansal and K. S. Venkatesh. (2015). "People counting in high density crowds from still images." [Online]. Available: https://arxiv.org/abs/1507.08445

[5] L. Boominathan, S. S. Kruthiventi, and R. V. Babu, "Crowdnet: A deep convolutional network for dense crowd counting," in *Proc. 24th ACM Int. Conf. Multimedia*, 2016, pp. 640–644.

[6] X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale aggregation network for accurate and efficient crowd counting," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2018, pp. 757–773.

[7] J. Carreira and C. Sminchisescu, "CPMC: Automatic object segmentation using constrained parametric min-cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1312–1328, Jul. 2012.

[8] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–7.

[9] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–7.

[10] A. B. Chan, M. Morrow, and N. Vasconcelos, "Analysis of crowded scenes using holistic properties," in *Proc. IEEE Int. Workshop Perform. Eval. Tracking Surveill.*, 2009, pp. 101–108.

[11] A. B. Chan and N. Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 5, pp. 909–926, May 2008.

[12] A. B. Chan and N. Vasconcelos, "Counting people with low-level features and Bayesian regression," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2160–2177, Apr. 2012.

[13] K. Chen, S. Gong, T. Xiang, and C. C. Loy, "Cumulative attribute space for age and crowd density estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2467–2474.

[14] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localised crowd counting," in *Proc. BMVC*, vol. 1, 2012, p. 3.

[15] D. Conte, P. Foggia, G. Percannella, F. Tufano, and M. Vento, "A method for counting moving people in video surveillance videos," *EURASIP J. Adv. Signal Process.*, vol. 2010, no. 1, 2010, Art. no. 231240.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 886–893.

[17] A. C. Davies, J. H. Yin, and S. A. Velastin, "Crowd monitoring using image processing," *Electron. Commun. Eng. J.*, vol. 7, no. 1, pp. 37–47, Feb. 1995.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.

[19] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[20] P. Dollár, Z. Tu, P. Perona, and S. Belongie, *Integral Channel Features*. BMVC Press, 2009.

[21] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[22] J. Ferryman and A. Shahrokni, "PETS2009: Dataset and challenge," in *Proc. 12th IEEE Int.Workshop Perform. Eval. Tracking Surveill. (PETS-Winter)*, Dec. 2009, pp. 1–6.

[23] L. Fiaschi, U. Koethe, R. Nair, and F. A. Hamprecht, "Learning to count with regression forest and structured labels," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 2685–2688.

[24] W. Ge and R. T. Collins, "Marked point processes for crowd counting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 2913–2920.

[25] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.

[26] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.

[27] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, nos. 1–3, pp. 185–203, Aug. 1981.

[28] Y.-L. Hou and G. K. H. Pang, "Automated people counting at a mass site," in *Proc. IEEE Int. Conf. Automat. Logistics (ICAL)*, Sep. 2008, pp. 464–469.

[29] Q. Hu, P. Wang, C. Shen, A. van den Hengel, and F. Porikli, "Pushing the limits of deep CNNs for pedestrian detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 6, pp. 1358–1368, Jun. 2018.

[30] S. Huang *et al.*, "Body structure aware deep crowd counting," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1049–1059, Mar. 2018.

[31] S. Huang and D. Ramanan, "Expecting the unexpected: Training detectors for unusual pedestrians with adversarial imposters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jul. 2017, pp. 4664–4673.

[32] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2547–2554.

[33] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

[34] D. Kang, D. Dhar, and A. B. Chan. (2016). "Crowd counting by adapting convolutional neural networks with side information." [Online]. Available: https://arxiv.org/abs/1611.06748

[35] S. Khan, G. Vizzari, S. Bandini, and S. Basalamah, "Detecting dominant motion flows and people counting in high density crowds," *J. WSCG*, vol. 22, no. 1, pp. 21–30, 2014.

[36] P. Kilambi, E. Ribnick, A. J. Joshi, O. Masoud, and N. Papanikolopoulos, "Estimating pedestrian counts in groups," *Comput. Vis. Image Understand.*, vol. 110, no. 1, pp. 43–59, 2008.

[37] D. Kong, D. Gray, and H. Tao, "A viewpoint invariant approach for crowd counting," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 3, Aug. 2006, pp. 1187–1190.

[38] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 878–885.

[39] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1324–1332.

[40] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-aware fast R-CNN for pedestrian detection," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 985–996, Apr. 2018.

[41] Z. Lin and L. S. Davis, "Shape-based human detection and segmentation via hierarchical part-template matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 604–618, Apr. 2010.

[42] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2016, pp. 21–37.

[43] P. Luo, Y. Tian, X. Wang, and X. Tang, "Switchable deep network for pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 899–906.

[44] A. N. Marana, S. A. Velastin, L. F. Costa, and R. A. Lotufo, "Estimation of crowd density using image processing," in *Proc. IEE Colloq. Image Process. Secur. Appl.*, Mar. 1997, pp. 11/1–11/8.

[45] Z. Ma and A. B. Chan, "Crossing the line: Crowd counting by integer programming with local features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2539–2546.

[46] J. Mao, T. Xiao, Y. Jiang, and Z. Cao, "What can help pedestrian detection?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6034–6043.

[47] A. N. Marana, S. A. Velastin, L. F. Costa, and R. A. Lotufo, "Estimation of crowd density using image processing," in *Proc. IEE Colloq. Image Process. Secur. Appl.*, Mar. 1997, pp. 11/1–11/8.

[48] M. Marsden, K. McGuinness, S. Little, and N. E. O'Connor. (2016). "Fully convolutional crowd counting on highly congested scenes." [Online]. Available: https://arxiv.org/abs/1612.00220?context=cs

[49] D. Oñoro-Rubio and R. J. López-Sastre, "Towards perspective-free object counting with deep learning," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 615–629.

[50] W. Ouyang and X. Wang, "Joint deep learning for pedestrian detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2056–2063.

[51] W. Ouyang, X. Zeng, and X. Wang, "Modeling mutual visibility relationship in pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3222–3229.

[52] V.-Q. Pham, T. Kozakaya, O. Yamaguchi, and R. Okada, "Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3253–3261.

[53] J. Redmon and A. Farhadi. (2017). "YOLO9000: Better, faster, stronger." [Online]. Available: https://arxiv.org/abs/1612.08242

[54] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[55] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D, Nonlinear Phenomena*, vol. 60, nos. 1–4, pp. 259–268, 1992.

[56] D. Ryan, S. Denman, C. Fookes, and S. Sridharan, "Crowd counting using multiple local features," in *Proc. Digit. Image Comput., Techn. Appl. (DICTA)*, Dec. 2009, pp. 81–88.

[57] D. Ryan, S. Denman, C. Fookes, and S. Sridharan, "Scene invariant multi camera crowd counting," *Pattern Recognit. Lett.*, vol. 44, pp. 98–112, Jul. 2014.

[58] D. Ryan, S. Denman, S. Sridharan, and C. Fookes, "An evaluation of crowd counting methods, features and regression models," *Comput. Vis. Image Understand.*, vol. 130, pp. 1–17, Jan. 2015.

[59] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1, Jul. 2017, pp. 4031–4039.

[60] M. Saqib, S. D. Khan, and M. Blumenstein, "Texture-based feature mining for crowd density estimation: A study," in *Proc. Int. Conf. Image Vis. Comput. New Zealand (IVCNZ)*, Nov. 2016, pp. 1–6.

[61] Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu, and X. Yang, "Crowd counting via adversarial cross-scale consistency pursuit," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5245–5254.

[62] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[63] V. A. Sindagi and V. M. Patel, "Generating high-quality crowd density maps using contextual pyramid CNNs," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1861–1870.

[64] Y. Tian, P. Luo, X. Wang, and X. Tang, "Pedestrian detection aided by deep learning semantic tasks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5079–5087.

[65] J. R. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.

[66] H. Ullah, M. Ullah, M. Uzair, and F. Rehman, "Comparative study: The evaluation of shadow detection methods," *Int. J. Video Image Process. Netw. Secur.*, vol. 10, no. 2, pp. 1–7, 2010.

[67] S. Uras, F. Girosi, A. Verri, and V. Torre, "A computational approach to motion perception," *Biol. Cybern.*, vol. 60, no. 2, pp. 79–87, 1988.

[68] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 734–741.

[69] E. Walach and L. Wolf, "Learning to count with CNN boosting," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 660–676.

[70] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 32–39.

[71] W. Ouyang and X. Wang, "A discriminative deep model for pedestrian detection with occlusion handling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3258–3265.

[72] Y. Wang, H. Lian, P. Chen, and Z. Lu, "Counting people with support vector regression," in *Proc. 10th Int. Conf. Natural Comput. (ICNC)*, Aug. 2014, pp. 139–143.

[73] B. Wu, F. Iandola, P. H. Jin, and K. Keutzer, "SqueezeDet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 446–454.

[74] L. Xiaohua, S. Lansun, and L. Huanqin, "Estimation of crowd density based on wavelet and support vector machine," *Trans. Inst. Meas. Control*, vol. 28, no. 3, pp. 299–308, 2006.

[75] F. Xiong, X. Shi, and D.-Y. Yeung, "Spatiotemporal modeling for crowd counting in videos," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5161–5169.

[76] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 818–833.

[77] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 833–841.

[78] J. Zhang, B. Tan, F. Sha, and L. He, "Predicting pedestrian counts in crowded scenes with rich and high-dimensional features," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1037–1046, Dec. 2011.

[79] L. Zhang, L. Lin, X. Liang, and K. He, "Is faster R-CNN doing well for pedestrian detection?" in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 443–457.

[80] S. Zhang, C. Bauckhage, and A. B. Cremers, "Informed haar-like features improve pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 947–954.

[81] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li. (2018). "Occlusion-aware R-CNN: Detecting pedestrians in a crowd." [Online]. Available: https://arxiv.org/abs/1807.08407

[82] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 589–597.

[83] T. Zhao, R. Nevatia, and B. Wu, "Segmentation and tracking of multiple humans in crowded environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 7, pp. 1198–1211, Jul. 2008.

[84] L. Zhu, Y. Chen, and A. Yuille, "Learning a hierarchical deformable template for rapid deformable object parsing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 6, pp. 1029–1043, Jun. 2010.

[85] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 391–405.

MUHAMMAD SAQIB received the bachelor's degree in computer systems engineering from the University of Engineering and Technology, Peshawar, Pakistan, in 2007, and the master's degree in electronics and communication engineering from Hanyang University, South Korea, in 2010. He is currently pursuing the Ph.D. degree with the School of Software, University of Technology Sydney, Australia.

He started as a Lecturer at Umm Al-Qura University, Saudi Arabia, where his main responsibilities include demonstration of labs and mentoring of undergraduate students in their projects. He is also involved in various research projects related to image processing and computer vision. His research interests include pedestrian crowd analysis, object detection, and deep convolutional networks. He is a Professional Member of the Pakistan Engineering Council and an Active Member of the Centre for Artificial Intelligence, University of Technology Sydney.

SULTAN DAUD KHAN received the M.Sc. degree in electronics and communication engineering from Hanyang University, South Korea, and the Ph.D. degree in computer vision from the University of Milano-Bicocca, Italy. He is currently an Assistant Professor with the College of Computer Science and Engineering, University of Hail, Saudi Arabia. His research interests mainly include computer vision application to pedestrian and crowd analysis.

NABIN SHARMA received the Ph.D. degree from the School of ICT, Griffith University, QLD, Australia. He is currently a Senior Lecturer with the School of Software, Faculty of Engineering and IT, UTS. His research interests include video and image processing, pattern recognition, and machine learning techniques for object detection and recognition. He has more than 14 years of experience in research and development and academia. He has substantial industry experience in software design and development while working on various projects at IBM India Private Ltd. He has published over 45 papers in refereed books, conferences, and journals. He secured research grants for projects with funds exceeding AUD$341K, in collaboration with industry and academia.

MICHAEL BLUMENSTEIN is currently the Associate Dean (Research Strategy and Management) of the Faculty of Engineering and IT, UTS, where he recently concluded his role as the Head of the School of Software. Previously, he was with Griffith University, QLD, Australia, where he has accumulated over a decade of experience in leadership roles, including portfolio Dean (Research) of the Sciences Group and the Head of the School of ICT.

Dr. Michael served as the Chair of the Queensland Branch, IEEE Computational Intelligence Society. He was also the Gold Coast Chapter Convener and a Board Member of the Australian Computer Society's Queensland Branch Executive Committee. He is the immediate past Chairman of the IT Forum Gold Coast and a former Board Member of IT Queensland. He has served on the Australian Research Council's (ARC) College of Experts on the Engineering, Mathematics and Informatics (EMI) Panel. In addition, he was elected as the Executive of the Australian Council of Deans of Information and Communication Technology (ACDICT). He currently serves on the Australian Information Industry Association (AIIA) NSW Council and is the elected Director of the Australian Computer Society (ACS) Technical Advisory Board.