Arabian Journal for Science and Engineering (2021) 46:3051–3065
https://doi.org/10.1007/s13369-020-04990-w

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

Sparse to Dense Scale Prediction for Crowd Couting in High Density Crowds

Sultan Daud Khan[1] · **Saleh Basalamah**[2]

Received: 9 April 2020 / Accepted: 28 September 2020 / Published online: 27 October 2020

Abstract
Head detection-based crowd counting is of great importance and serves as a preprocessing step in many visual applications, for example, counting, tracking, and crowd dynamics understanding. Despite significant importance, limited amount of work is reported in the literature to detect human heads in high-density crowds. The problem of detecting heads in crowded scenes is challenging due to significant scale variations in the scene. In this paper, we tackle this problem by exploiting contextual constraints offer by the crowded scenes. For this purpose, we propose two networks, i.e., sparse-scale convolutional neural network (SS-CNN) and dense-scale convolutional neural network (DS-CNN). SS-CNN detects human heads with coarse information about the scales in the image. DS-CNN utilizes detection obtained from SS-CNN and generates dense scalemap by globally reasoning the coarse scales of detections obtained from SS-CNN via Markov Random Field (MRF). The dense scalemap has unique property that it captures all scale variations in image and provides an aid in generating scale-aware proposals. We evaluated our framework on three challenging state-of-the-art datasets, i.e., UCF-QNRF, WorldExpo'10, and UCF_CC_50. Experiment results show that proposed framework outperforms existing state-of-the-art methods.

Keywords Crowd counting · Head detection · High-density crowds · Crowd analysis

1 Introduction

Ensuring crowd safety and providing security to the participants of mass events is challenging problem and receiving great attention from the scientific community. With the growing population and increasing urbanization, mass events like marathons, sports, religious festivals, concerts, and carnivals organized frequently. In order to ensure crowd safety and security at these mass events, adequate safety measures must be adopted by the event organizers and security personnel. Crowd disasters still occur frequently, for example, during Love Parade [1] and Hajj [2], despite all safety measures. Crowd disasters usually attribute to critically high densities in a constrained environment. To avoid crowd disasters and in order to ensure crowd safety, it is important to analyze crowd dynamics. Understanding crowd dynamics has numerous applications, for example, anomaly detection [3–6], congestion detection [7], crowd counting, tracking [8,9] and many others. Among these applications, crowd counting has achieved tremendous attention from the computer vision community during recent years [10–14]

The goal of crowd counting is to estimate total number of pedestrians in the scene. Crowd counting has numerous applications in video surveillance, traffic monitoring [15], public space design and event planning. With precise crowd count and localization of pedestrians in the environment can substantially reduce the cost. However, crowd counting is challenging due to non-uniform and complex distribution of the people in the environment. Significant variations in human head scales have further made the counting problem challenging. Several strides have been made during recent years to tackle these challenges. Most of traditional crowd counting methods use different regression techniques like linear regressor [16], support vector regressor (SVM) [17], Gaussian process regression (GPR) [18], K-nearest neighbor (KNN) [19] and neural network (NN) [20] to estimate the crowd count. With the recent advancement in computer vision technology, and with success of Convolutional neural networks (CNNs), recent crowd counting methods

✉ Sultan Daud Khan
sultandaud@nutech.edu.pk

Saleh Basalamah
smbasalamah@uqu.edu.sa

[1] National University of Technology, Islamabad, Pakistan

[2] Umm Al-Qura University, Makkah, Saudi Arabia

employ various CNNs methods to estimate crowd count by employing regression on the density maps [13,14,21–23]. Regression-based models work well in high-density situations since these models capture the global and generalized density information. However, these methods over estimate the count in low-density situations. Moreover, these methods are blind and cannot estimate the location of individuals in the scene which provide crucial information for crowd managers. On contrary, detection-based crowd counting methods detect individuals in the scene by training an object detector. In detection base approaches, crowd count is the number of detection in image. These methods extract discriminating features that best describe the human body.

The performance of detection-based crowd counting methods in low-density crowds is high as compared to high-density situations. In high-density crowds, due to limited space, pedestrians stand very close to each other. Most parts of human body are not visible due to occlusion and it is challenging for a detector to precisely detect pedestrians. In high-density crowd, as large part of human body is occluded, human head is reliable part. Although few steps have been made in detecting human heads in high-density crowds [11,12,24,25], head detection in high-density crowds is challenging job and there is still room for improvement.

Variations in intra-class scales, appearances and poses of heads have further worsen the problem to detect human head in cluttered scenes. Thanks to the translation invariance property of CNNs which has enabled large capacity networks to efficiently handle the problem of pose and appearance variations in the scene. However, the problem of intra-class scale variance is still an open issue.

From empirical analysis, we observe that scale variations is naturally caused by perspective distortions in images. The perspective distortion is imposed by camera view point [26] due to which scale of the object changes from near to far end in an image. Perspective information embeds the distance between the object and camera and provides better estimate of the different scales in the scene. In high-density crowd scenes, traditional CNN models with a single scale can not detect human heads with significant scale variations.

In order to handle scale variations, Zhang et al. [27] propose multi-column CNN (MC-CNN) that uses multiple branches, where each branch corresponds to a CNN with different receptive field. MC-CNN consider only limited scales and cannot handle the significant scales variations in high-density crowded scenes. Moreover, MCNN is hard to train due to multi-column architecture and computational complexity increases with the increase in the number of columns. Sam et al. [15] propose Switch-CNN using same intuition of multiple branches but instead of concatenating features from multiple branches, Switch-CNN predict a scale class of an input image. The predicted scale class is then used to select one of the branches and used its features to estimate density. Similarly, Liu et al. [24] introduce DecideNet that operates in two modes: (1) regression mode and (2) detection mode. The network switch between the modes for different location of image based on the real density. Both Switch-CNN and DecideNet uses binary decision in selecting a mode based on classifier's output. This kind of hard decision may cause wrong selection of mode that will ultimately lead to incorrect results. In order to solve this problem, Hossain et al. [28] propose scale attention network that "softly" selects the scales based on the density. Zhang et al. [23] propose multi-resolution attention CNN (MCA-CNN) that generate a score map, where high response in score map represents high probability of head that will guide the network to focus on head areas. Li et al. [29] proposes CSRNet that uses dilated convolutions with fixed receptive fields. The model cannot handle high scale variations in the scene due to fixed receptive field sizes. It is observed that CSRNet works well for medium size scales while performance degrades at smaller and larger scales. Basalamah et al. [12] propose SD-CNN and achieved state-of-the-art performance by tackling the scale variations by encoding perspective information in scalemap. However, estimating scalemap requires human efforts to manually annotate human heads for each input image. Recently, Deepak et al. [30] propose multi-column model (LSC-CNN) that fuses top-down features and produces detection at multiple resolutions. Yancheng et al. [31] proposed a model (Tiny Face) that generates high-resolution faces from low-resolution and blurry one by employing generative adversarial network.

In this paper, we proposed a framework that handles significant scale variations by predicting dense scales in an image. Generally, our framework follows the following pipeline.

1. At first stage, our Sparse-Scale CNN (SS-CNN) takes whole image as input and outputs multiple feature maps corresponding to each sparse scale. Our SS-CNN is similar to MC-CNN, yet we change receptive fields of branches by changing the number of filters and filter sizes to capture as much scales of heads as possible in images. We then apply non-maximum suppression method to each feature map that suppresses low confidence pixels and detects human heads. We then accumulate detection from multiple branches. The obtained detection provides coarser information about the location and scale of human heads in the scene.
2. In the second stage, we use detection obtained from the first stage and generate dense scalemap by globally reasoning via Markov Random Field that captures the scale of head at each pixel of the image.

3. In the third stage, a uniform gird of points is initialized over dense scalemap. Taking grid point as a center, we then extract perspective-aware patches (proposals).
4. In the fourth stage, we classify each input proposal into head/background and obtain a response map. Finally, we apply non-maximum suppression on resultant response map and final detection are obtained. The detection performance of proposed method compare to related methods is shown in Fig. 1.

Our framework has the following contribution in comparison to other state-of-the-art methods:

- Proposed framework introduces a paradigm shift in crowd counting methods and replaces prevalent regression-based methods by detecting human heads in high-density crowds.
- We introduce a method of dense scale prediction (at each pixel of image) by exploiting information from locally consistent sparse scales.
- With precise detection in high-density crowds, our framework provides the distribution of people in the environment.
- Proposed framework achieves state-of-the-art performance on existing challenging benchmark datasets, i.e., UCF-QNRF [34], WorldExpo'10 [35], UCF_CC_50 [32]

2 Related Work

Many methods are reported in the literature for crowd counting and density estimation. Generally, we categories the literature in two major classes, i.e., *Regression methods* and *Detection methods*.

(a) LSC-CNN [30] **(b)** TinyFace [32] **(c)** Proposed

Fig. 1 Sample frame from UCF_CC_50 [32] shows the significance of our proposed method compare to most recent methods. Sample frame has 1046 annotated human heads. **a** LSC-CNN [30] (1st image) underestimates the count and detects 873 heads. **b** Tiny Face [33] (2nd image) trained on face dataset [31] over estimates the count and detects 1537 heads and produces many false positives. **c** Our proposed framework (3rd image), on the other hand, detects 1029 (close to ground truth count) heads and also precisely detects bounding boxes (best view in zoom)

Regression-based methods, for example, support vector regressor [17], linear regression [16], K-nearest neighbor [19] Gaussian Process Regression(GPR) [18], and neural network [20] predict the count by performing regression between features of image and count. For example, GPR is employed in [18] using edge and texture features to estimate the count in an image. Crowd density is estimated in [20] by employing self-organizing map. Correspondence between crowd count and foreground pixels is learned by neural network in [36]. Neural network is trained on histograms of edges and blob size to find crowd count [37]. Support vector regressor is trained in [38] to estimate the count by using SURF features.

With the tremendous success of deep learning in classification and segmentation problems, several CNN models are proposed to estimate the crowd count. In CNN-based models, density map is generated by learning hierarchical features from the raw image, and count is obtained by integrating the count from patches of the image. A Multi-column Convolutional Neural Network (MCNN) [27] estimates the crowd count by using three branches with different filter sizes to compensate perspective distortions. CNN-based switching architecture is proposed that efficiently switches regressors for a particular crowd patch based on density level. Scale Aggregation Network (SANet) [13] is proposed for estimating high-resolution density maps.

Detection-based methods detect pedestrians in crowds and final count is the sum of detection. We further divide these methods into two categories, i.e., *hand-crafted features* [39–42] and *hierarchical features* [43–47]. The first category trains a classifier based on hand-crafted features, for example, edges, texture, and shape. The performance of these methods is relatively low in complex scenes, since hand-crafted features provide weak representation of human body. Deformable Part Model (DPM) is proposed [48] to model more generic representation and learn different poses and parts of humans. DPM is efficient and robust to complex scenes; however, learning parameters cannot be optimized to improve performance.

In second category, hierarchical features are learned from the raw images using CNN. The first step in this direction is taken in [49]. Although the network achieved success in early years, it lost its popularity in the following years due its dependence on Selective Search strategy [50] for object proposals generation. In order to overcome this limitation, Faster-RCNN proposed two-stage pipeline that uses Region Proposal Network (RPN) for object proposal generation. You-Only-Look-Once (YOLO) [51], instead of generating object proposals, use regression model to classify bounding boxes of different sizes and scales. Single shot detector (SSD) [52] uses Fully Convolutional network to produce limited number of bounding boxes. Class probabilities are assigned to each bounding box. Non-maximum suppression method (NMS) is then applied to suppress low confidence bounding boxes and final detection are obtained.

To summarize the shortcomings of the existing methods, we argue that above mentioned regression-based methods are blind and cannot detect/localize humans in the scene. On the other hand, detection base methods cannot effectively handle the scale and small object detection problem. Previous approaches adopt different way to solve the multi-scale and small object detection problem. For example, Faster-RCNN fails to detect small objects. It attributes to the fact that Faster-RCNN uses feature map of the high level layer for object detection. These high level layers have large receptive fields sizes and do not contain information about the small objects. Therefore, Faster-RCNN misses heads during inference stage. SSD [35] on the other hand uses feature maps of top and shallow to tackle scale in variance problem. Features maps from the top layers have small resolution that lack details of small objects. Moreover, the resolution of shallow layers is large, however, have less discriminating power that ultimately leads to significant amount of false positives. YOLO follows anchor box-based network structure and uses bounding box regression. YOLO performs well in general object detection tasks; however, in crowded scenes, the size and shape of heads change significantly as compared to generic large objects, it requires much more complex design of anchor boxes to capture wide range of scales. Therefore, YOLO (anchor box-based method) is inefficient in such case. Moreover, YOLO has difficulty in detecting objects that are small and close to each other due to only two anchor boxes in a grid predicting only one class of object. Furthermore, we observed from our experiments that the performance of YOLO is lower than traditional Faster RCNN and significantly lower than our proposed framework.

We address above shortcomings by proposing head detection base crowd counting framework that detect heads in low as well as in high-density crowds. Proposed framework addresses multi-scale and small object detection problem by generating dense scale map that captures wide range of scale variations in input image. We then exploit dense scale map to generate scale-aware proposals that are classified by DS-CNN.

3 Proposed Methodology

In this section, we discuss the methodology of proposed framework. The pipeline of proposed framework is shown in Fig. 2. The framework has two main networks. (1) Sparse Scale Convolutional Neural Network (SS-CNN) and (2) Dense Scale Convolutional Neural Network (DS-CNN). The architecture of SS-CNN is different than DS-CNN. SS-CNN is a multi-branch architecture that detects human heads in limited range of scales. SS-CNN cannot detect all heads in

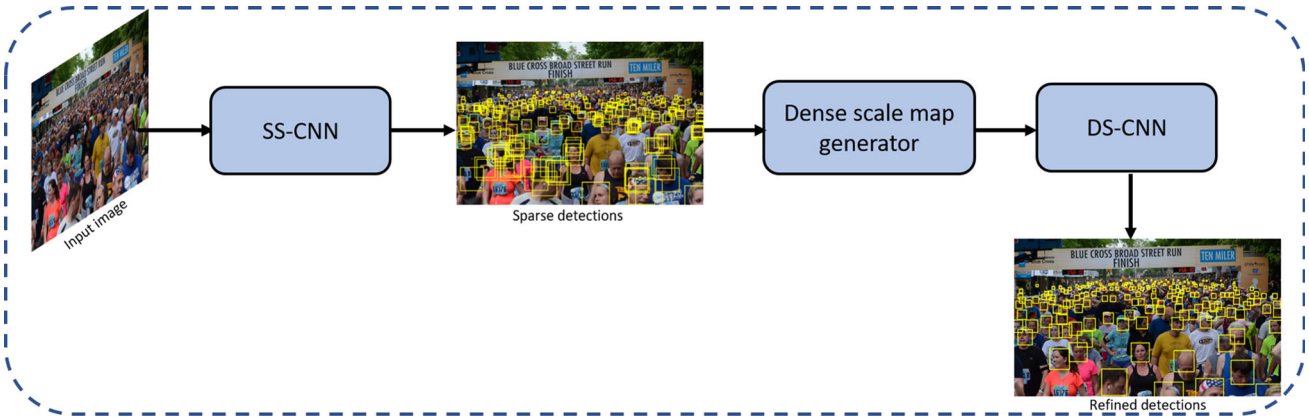

Fig. 2 Pipeline of proposed framework. An arbitrary size image is provided as input to sparse scale convolutional neural network (SS-CNN) that generates sparse detections. SS-CNN due to sparse nature missed many detection and accumulated false positives. These sparse detection are then utilized by dense scale generator that generates dense scale map. The dense scale map is then utilized by dense scale convolutional neural network to generate scale-aware proposals by exploiting dense scale map and obtain refined detection as final output

the image since the network cannot capture wide range of scales of human heads. To capture wide range of scales of human heads (dense scale), we utilized the detection obtained by SS-CNN and generate dense scale map by globally reasoning the coarse scales of detections obtained from SS-CNN via Markov Random Field (MRF). The dense scale map is then utilized by DS-CNN to generate scale-aware object proposal which are then classified by into two classes, i.e., head/background and obtain a response map. Finally, we apply non-maximum suppression on resultant response map and obtain final detections.

3.1 Sparse-Scale Convolutional Neural Network

In high-density situations, as discussed in Sect. 1, head is most reliable and visible part. Due to perspective distortions, there is significant variations in scales of people heads. Scale is defined as the size of the image patch (corresponds to head) in the original image, corresponds to the pixel in feature map of the last convolutional layer of the network. Conventionally, scale problem is handled by using multi-scale pyramid. Multi-scale pyramid has been extensively used to handle the scale problem in object detection tasks. However, in high-density crowd images, processing multi-scale image pyramid incurs huge computation cost. Furthermore, information about the smaller heads is lost due to image re-sizing to a smaller scale. On the other hand, convolutional neural network with fixed strides and filter sizes unable to handle scale variations at large extent. With the smaller scale, the network have small receptive field and more susceptible to smaller heads in the image. On the other hand, larger scale will focus on large heads and skip smaller heads.

To address this problem, we propose Sparse-Scale Convolutional Neural Network (SS-CNN) which consists of three branches. Each branch consists of convolutional layers with different filter sizes and strides to capture the different characteristics of crowd at different scales. Our SS-CNN is similar to [53], which was basically proposed for image classification task. In our case, we modify the network in a way to handle detection problem. The overall architecture of our network is shown in Fig. 3. We keep the same network structure for all branches but filter sizes and strides are changed to enable the network to capture different sizes of heads. We then adopt a fusion strategy to combine feature maps from three branches.

The feature maps of the last convolutional layer from different braches of SS-CNN are of different in sizes; however, the number of channels is same. Due to unique pattern in sizes of convolutional layers of three branches of SS-CNN, the size of feature map of 1st branch (Feature map 1) is half of the size of feature map of feature map (Feature map 2) from 2nd branch layer. Similarly, the size of feature map of 3rd branch is half of feature map of 2nd branch. In order to fuse these feature maps together, we need to bring all of three feature maps to the same size. For this purpose, we up-sample feature maps of 2nd and 3rd branches to match the size of feature maps of 1st branch. For up-sampling, we use deconvolution layer which adopts a top-down approach and makes the feature semantically stronger for detecting small objects. In order to make feature map of 2nd and 3rd branch equal to the feature map size of 1st branch, we apply one 2×2 deconvolution with 512 channels to feature map of 2nd branch and two 2×2 deconvolution layers (one after the other) to feature map of 3rd branch. After applying these operations, the feature maps from different layers are summarized point to point with equivalent weights. We then employ 1×1 convolution layer to further suppress aliasing and generate the final fusion map. The final fusion map contains both seman-

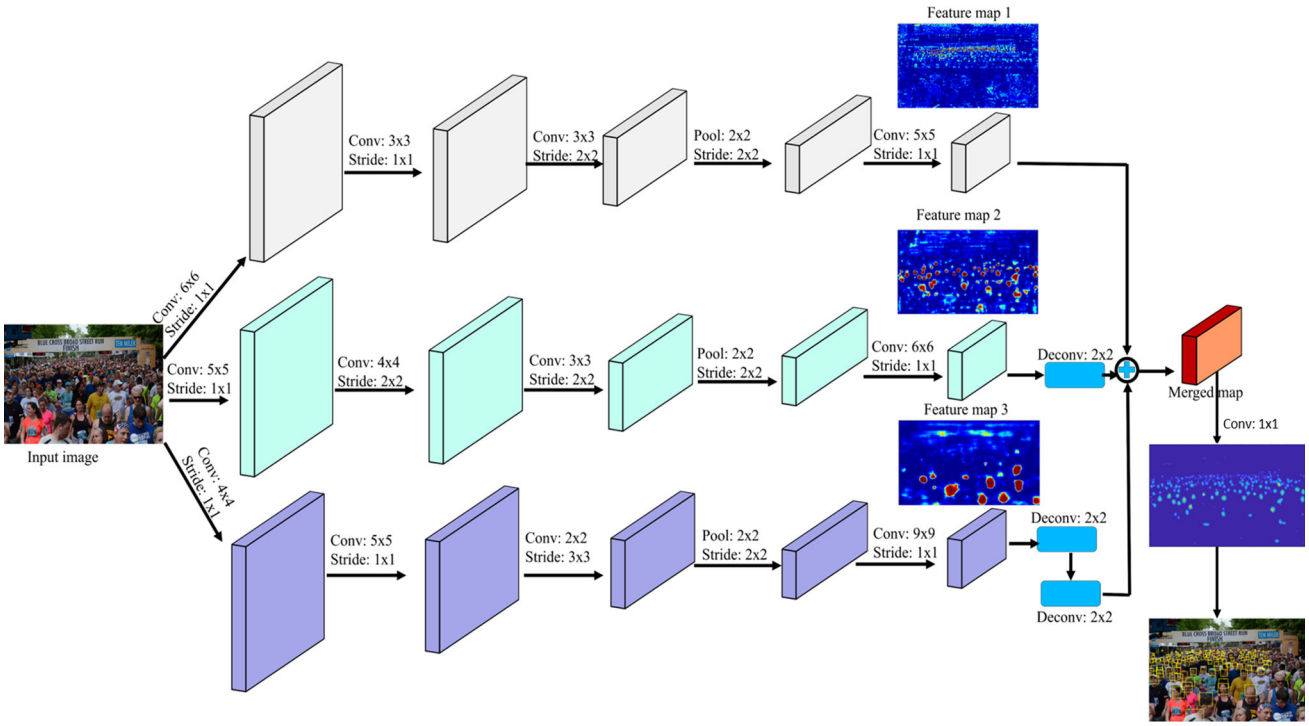

Fig. 3 SS-CNN: sparse scale convolutional neural network for detecting human head in sparse scales. Branches of the network having different receptive fields are encoded with different colors

tic knowledge from higher layers and fine-grained details of small objects.

We optimize loss function by computing the Euclidean distance between the estimated merged feature map and ground truth. We define the loss function as follows

$$L(\Phi) = \frac{1}{2K} \sum_{j=1}^{K} ||M(I_j : \Phi) - M_j|| \tag{1}$$

where K is the number of training images, Φ is the learnable weights of SS-CNN, I_j is the input image, $M(I_j : \Phi)$ is the estimated stacked feature map for image I_j and M_j is the ground truth density map.

For training the network, we follow the process adopted in [54]. We train each single CNN (corresponds to branch) of the SS-CNN separately and use these pre-trained CNNs to initialize all the branches of SS-CNN and then fine-tuned all the parameters simultaneously. The branched structure makes SS-CNN more efficient to model the characteristics of crowd density with different head sizes. This model once trained on large dataset (contains millions of heads of different sizes) can easily be adapted to another dataset which contains human heads of the different sizes.

For localization of human heads, we post-process the estimated stacked feature map and find the local peaks by employing non-maximal suppression method. Due to the sparse configuration (three branches) of the SS-CNN, the network has following limitations, (1) predicts human heads in limited range of scales, (2) cannot provide accurate localization, (3) misses heads with different scales. For example, the receptive field of first branch of SS-CNN is 28. Each pixel in output feature map of first branch covers a window of region of fixed size (28). Similarly, the receptive field sizes of other two branches are 56 and 112, respectively. With this arrangement, SS-CNN cannot detect heads with a scale less than 28 and more than 112. Therefore, SS-CNN cannot handle dense scale variations in real-life high-density crowded images. To handle dense scale variations, intuitive solution is to increase the number of branches; however increasing the number of branches will incur high computational complexity during training and testing phase.

In order to address dense scale problem, we use scale and confidence of detection information obtained from SS-CNN. We embed this information in discontinuity preserving Markov Random Field that estimates dense scales of an entire input image.

3.2 Generation of Dense Scale Map

In high-density crowded scenes, head detection is a challenging task due to smaller head size and significant scale variations. As discussed above, SS-CNN detects human heads in limited range of scales. For accurate detection, we need dense scales that provides full coverage of all scale

variations in the image. In order to achieve this, we utilize the scale and confidence information from the detection provided by SS-CNN and propose a strategy that predicts dense scales of an entire image.

From empirical studies, we observe that scale of heads in the neighborhood is same but different in different locations of the image due to perspective distortions. However, this change is gradual across the image. Scale of human head in crowded scene provides a cue about immediate scales in the neighborhood of associated detection. We utilize the scale and confidence information of a particular detection and transfer this knowledge to the surrounding neighborhood and then to the entire image as in [55]. The pipeline of dense scale prediction is shown in Fig. 4.

Let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$ is a set of detection obtained from SS-CNN. Each detection $\omega_i \in \Omega$ is represented by $\{x_i, y_i, s_i, c_i\}$, where (x_i, y_i) denotes the position of detection, s_i and c_i represent the scale and confidence value, respectively. Now, given a set of detection Ω, our goal is to infer scales for each pixel of the input image. As discussed above, each detection have similarity relationship in terms of scales and confidence with its neighborhood; therefore we define *Relationship Function* to capture this similarity. It is observed that detections with larger scales effect the neighborhood at large distance while smaller detections have influence of limited range [55]. Since the scale consistency is valid only in the neighborhood and detection with high confidence values provide more reliable information, therefore, our *Relationship Function* depends on scales, confidence and distance information. Given a set of detection Ω, we compute dense scale map Υ that has maximum value at any location (x, y) and formulated as follows

$$\Upsilon_{x,y}(\Omega) = \arg\max_{i=1}^{n} \left(c_i . \exp\left(- \frac{||x - x_i||^2 + ||y - y_i||^2}{\sigma^2 . (1 + \frac{s_i}{\max(w,h)})^2} \right) \right) \tag{2}$$

where σ is the standard deviations along the x-axis and y-axis and (w, h) represent the width and height of the input image.

After obtaining scale map Υ, we then employ Markov Random Field to enforce smoothness across the image. This step is important to incorporate perspective effects due to which the scale of human heads gradually change from one pixel to another. Human heads near to the camera appear large as compared to far end. In order to enforce this consistency, we treat scales as random variables and use Markov Random Field to enforce smoothness across the image. Let ν represents set of pixels in the input image and L represents set of labels. We assign l_p to each pixel $p \in \nu$. Here, we assume that labels should vary smoothly across the image. We model this by using the following energy function [56]:

$$E(l) = \sum_{p \in \nu} D_p(l_p) + \sum_{(p,q) \in \chi} C_x(l_p, l_q) \tag{3}$$

where the first term $D_p(l_p)$ in equation represents the cost of assigning label l_p to pixel p. In the second term, χ shows the edges of four-connected pixels of image in the graph and $C_x(l_p, l_q)$ is the cost of assigning labels l_p and l_q to two neighboring pixels and calculated as $C_x(l_p, l_q) = C_x(l_p - l_q)$.

3.3 DS-CNN: Dense Scale Convolutional Neural Network

In this section, we discuss our DS-CNN that classify proposals into head/background. The pipeline of our DS-CNN is shown in Fig. 5. As shown in the Figure, the first preprocessing step is to generate scale friendly object proposals. We utilize dense scalemap to generate scale-friendly proposals. Let G represents uniform grid of points overlaid over the image. Let $P = \{p_1, p_2, \ldots p_n\}$ is set of n points belong to grid G, where $p_i = (x, y)$ represents location in the image. We then generate n number of object proposals of size $\Upsilon(P)$. Ideally, the resolution of the gird G is same as the input image; however, it will incur high computation cost. We reduce the resolution of the grid by a parameter λ with range of $0 < \lambda \leq 1$. Let I_x, I_y represents the resolution of input image. Let resolution of grid $G = \{g_x, g_y\}$ is given by: $g_x = \lambda I_x$, $g_y = \lambda I_y$. We observe a trade-off in selecting the value of λ. Higher values of λ, increase the number of proposals which leads to high accuracy at the cost of computational complexity. On contrary, lower values of λ result in lower recall rates due to less number of proposals. From empirical evidences, we found that 0.65 is the optimum choice, so we use $\lambda = 0.65$ in all our experiments.

We pre-process each object proposal before feeding to the network using the following steps: (1) Crop image patch corresponding to each proposal. (2) Re-size image patch according to the size of input layer of the network. The network then classifies each proposal by assigning a confidence value. After feed-forwarding all proposal, we then generate *response map* (equal to size of input image), where each pixel of the map represents the confidence value of the corresponding proposal. In order to precisely localize heads, we employ non-maximal suppression method (NMS). NMS finds local peaks using fixed threshold. For performance evaluation, we match predicted location with the ground truth locations.

The backbone of the proposed framework is based on Densenet-169 [57]and consists of 169 layers. The network consists of four dense blocks. Each dense block consists of set of convolutional layers densely connected together.

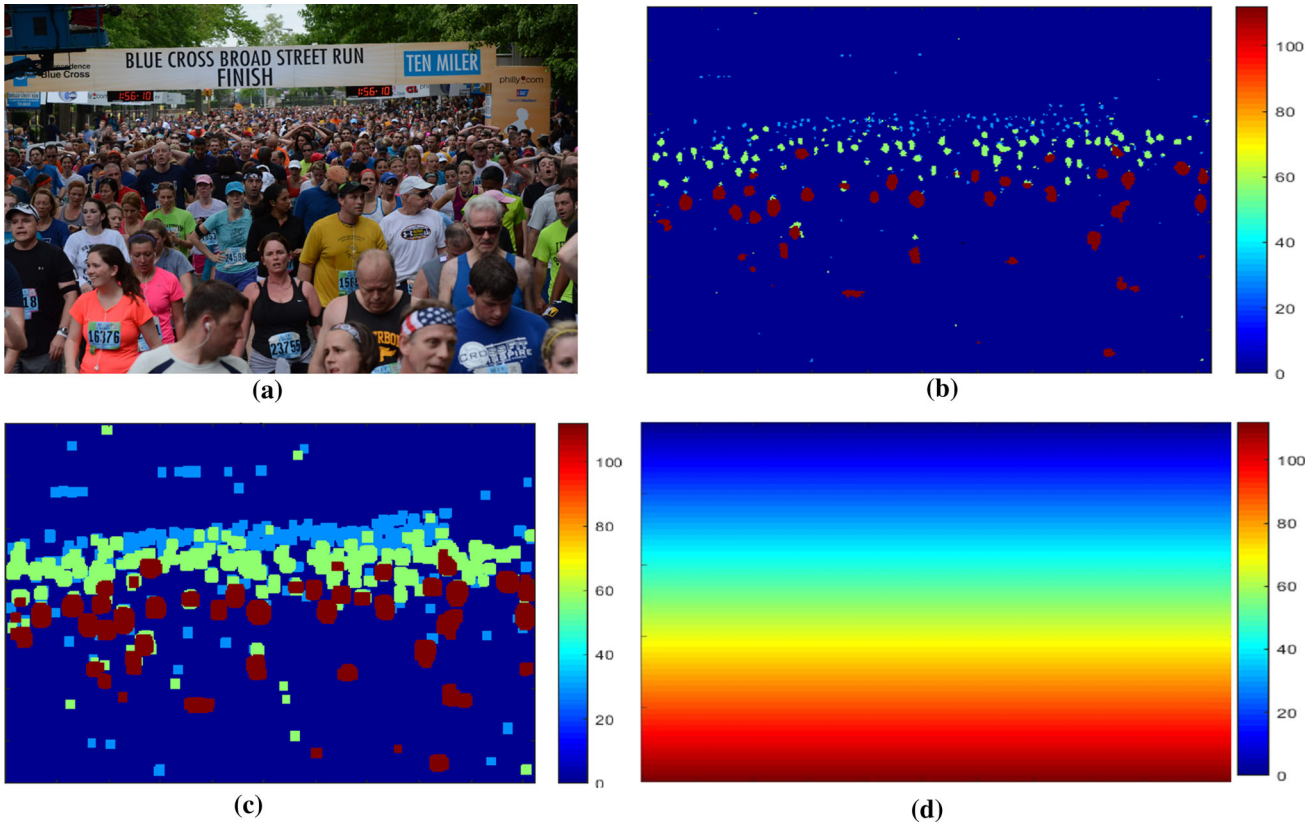

Fig. 4 Computation of dense scales: **a** input sample image. **b** The scales from detections obtained using DSCNN. **c** The scale obtained using Eq. 2 **d** Dense scale map obtained using Eq. 3. Heat maps in (**b**)–(**d**) used to represent the size of heads, where larger heads are indicated by red and small heads are represented by blue color

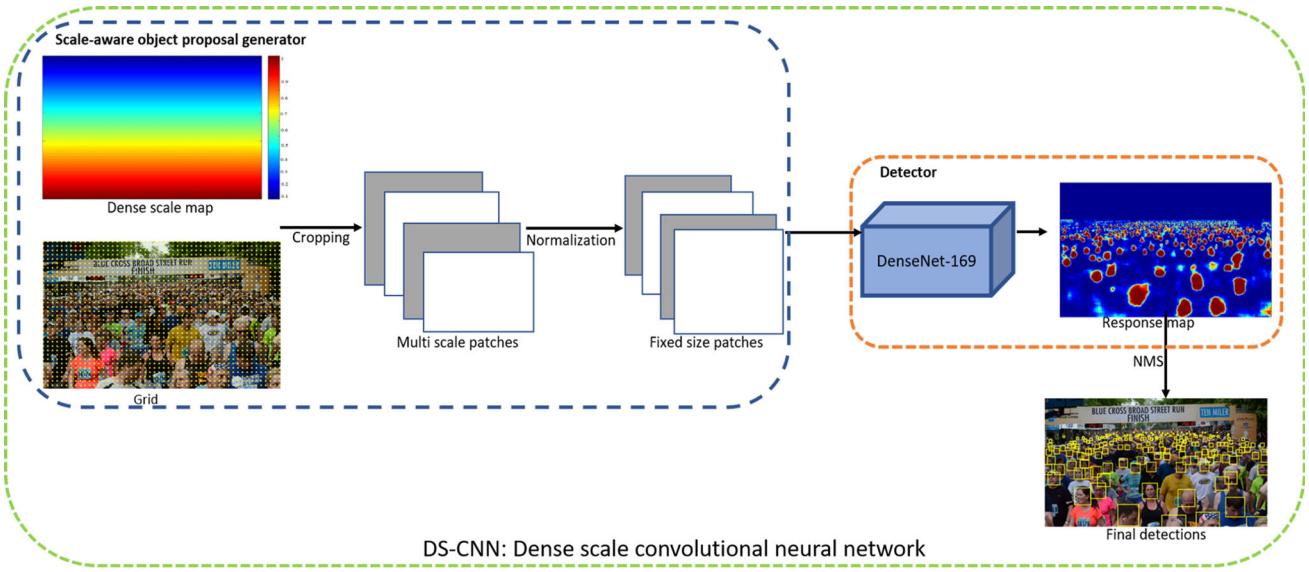

Fig. 5 DS-CNN: dense scale convolutional neural network for detecting human heads using dense scale map

Each dense block receives the input from previous block and outputs a feature map half of the size of feature map of the previous dense block. This significant reuse of residuals implements a deep supervision, since every layer receives more supervision from the previous layer that results in a powerful network that solves the problem of vanishing gradient.

The input image is first passed through a convolutional layer that has filter size of 7×7 and stride of 2, followed by a max pooling layer with size of 3×3 and stride of 2. The resultant feature map is then passed through a first dense block. The first dense block consists of six sets of two convolutional layers. The filter size of the first convolutional layer is 1×1 and filter size of the second convolutional layer is 3×3. In this way, the first dense block consists of $6 \times 2 = 12$ convolutional layers. In the same way, second dense block consist of $12 \times 2 = 24$ convolutional layers. Third dense block consists of $32 \times 2 = 64$ convolutional layers and fourth dense block consists of $32 \times 2 = 64$ layers. Each dense block is followed by transition layer. Transition layer consists of one convolutional layer of filter size 1×1 and one average pooling layer with filter size of 2×2 and stride of 2.

For training, we label bounding boxes (correspond to patches of image) to head/background. We use intersection-over-union (IoU) to decide the label of bounding box. IoU represents the overlap ratio of candidate bounding box to ground truth. We set threshold to 0.5 and label any bounding box as "head" if the overlap ratio is greater than threshold. The remaining bounding boxes are labeled as background. We use Adam solver with step learning rate. We set initially the learning rate to 0.001, and reduce it by a factor of 2 after every 20 epoch. We trained the entire network for 100 epoch with a batch size of 64.

UCF_CC_50 data is a challenging dataset and firstly introduced by H. Idrees et al. [32]. This dataset contains 50 images collected from different sources with different resolutions, viewpoints and varying densities. The density varies from 94 people/image to 4543 people/image with total of 63,974 annotations in 50 images. In our experiments, we follow the same fivefold cross-validation strategy proposed by [32].

WorldExpo'10 was introduced by Zhang et al. [35]. The dataset contains 225,216 head annotations collected from 3980 images. These images were captured from 108 cameras with different viewpoints and sampled from 1132 video sequences. For training, we use 3380 frames while remaining images are used for testing.

UCF-QNRF dataset is the most recent and comprehensive dataset introduced by H. Idrees et al. [34]. The dataset consist of 1535 images with 1,251,642 head annotations. Dataset has large number of high-density images with diverse set of viewpoints, resolutions, and lighting variations. The images were collected from three sources: Flickr, Web Search and the Hajj videos. The resolution of images is large as compared to other datasets which makes it more suitable for head detection task in high-density crowds. The summary of these datasets is given in Table 1.

For training, we also augment the training data and cropped 9 patches from different locations of each image. We keep size of each patch 1/4 of the original image. We use all these patches for training our model.

For comprehensive evaluation and comparison, we split experimental setup into two stages. In the first stage, we discuss counting performance of the framework while we evaluate and compare detection performance of our proposed framework in the second stage.

4 Experiment Results

In this section, we evaluate and compare our proposed with other state-of-the-art methods both in quantitative and qualitative way. We use three publicly available benchmark datasets for crowd counting, UCF-QNRF [34], WorldExpo'10 [35], UCF_CC_50 [32].

4.1 Evaluations and Comparisons

In this section, we evaluate and compare the counting performance of our proposed method with other state-of-the-art methods. For crowd counting, we follow the same convention of existing methods and use absolute error (MAE) and the mean squared error (MSE) as evaluation metrics defined

Table 1 Summary of the datasets. The second column shows that total number of images per dataset, the third column shows total number of annotations, fourth column shows the average count, CD represents crowd density in fifth column, Res represents the average resolution, and last column shows the total storage required by each dataset on hard drive

Dataset	No. of imgs	No. of Ann	Avg Cnt	CD	Res	Memory
WorldExpo'10 [35]	3980	225,216	56	3.5	576 × 238	325 MB
UCF-CC-50 [32]	50	63,974	1279	5.7	2101 × 2888	44 MB
UCF-QNRF [34]	1535	1,251,642	815	4.9	2013 × 2902	4.33 GB

Table 2 Comparative analysis with other techniques on UCF_CC_50 [32] dataset

Methods	MAE	MSE
Rodriguez et al. [58]	655.7	697.8
Idrees et al. [32]	419.5	590.3
Zhang et al. [35]	467.0	498.5
Liping et al. [59]	302.3	411.6
Lempitsky et al. [60]	493.4	487.1
MCNN [27]	377.6	509.1
SD-CNN [12]	235.7	345.6
MRA-CNN [23]	240.0	352.6
LSC-CNN [30]	243.1	374.8
Tiny face [33]	237.4	354.3
Proposed	229.4	325.6

as follows:

$$\text{MAE} = \frac{1}{K}\sum_{k=1}^{K}|\mu_k - G_k| \tag{4}$$

$$\text{MSE} = \frac{1}{K}\sum_{k=1}^{K}(\mu_k - G_k)^2 \tag{5}$$

where K is the total number of testing frames, μ_k is the predicted count and G_k is ground-truth count of pedestrians at frame k.

Using the above evaluation metric, we report the results of our method and other existing related methods in Table 2 using UCF_CC_50 dataset.

Rodriguez et al. [58] leverage global structure of the scene by estimating density map to improve the head detector performance. Idrees et al. [32] fuses feature from multi-sources, including SIFT, head detection, and Fourier to estimate crowd count. Zhang et al. [35] proposed CNN model to estimate crowd count. Lempitsky et al. [60] extracted SIFT features from randomly selected patches and use MESA distance to learn density map. MCNN [27] proposed multi-column CNN to estimate density map. MRA-CNN [23] estimate crowd density by adopting multi-resolution CNN to address the problem of scale variations in image.

These state-of-the-art methods produce higher MAE and MSE value compare to proposed framework. The lower performance attributes to the following reasons. (1) These methods exploit global texture information to capture generalized crowd density information. These methods work well in high-density crowds but overestimate the crowd count in low-density situations. (2) Since these methods are regression based, therefore, these methods cannot precisely localize human in the scene. On the other hand, SD-CNN [12] used scale-aware proposal to detect heads in high-density crowd. The method is based on generating perspective map that captures different scales in the image. The method performs well in detecting heads in high- and low-density images; however, the acquisition of perspective information required human efforts. From Table 2, it is obvious that our proposed model outperforms existing methods. It is due to reason that proposed framework precisely detect human heads in both low and high-density crowds, where the scale problem is effectively handled by proposed framework using dense scale map.

In Table 3, we compare the results of state-of-the-art methods with proposed method on WorldExpo'10 dataset using MAE evaluation metric. Zhang et al. [35] extract local features by cropping patches of different sizes from different parts of the image and train a model to estimate crowd count and density. However, the model relies on perspective information that generally is hardly available. MCNN [27] estimates the density map by capturing multiple scales of objects using multi-column CNN. The performance of MCNN is reduced in high-density crowd images as it covers limited scales. Switching CNN [15] is the extension of MCNN that predicts the density map by choosing appropriate regressor for input patch. ACSCP [14] proposed patch-to-density prediction network by employing cross-scale regularization scheme. CP-CNN [61] incorporates local and global contexts by proposing contextual pyramid CNN.

Table 3 Comparative analysis with other techniques on WorldExpo'10 [35] dataset using MAE metric

Methods	S1	S2	S3	S4	S5	Average
Zhang et al. [35]	9.80	14.10	14.30	22.20	3.70	12.90
MCNN [27]	3.40	20.60	12.90	13.00	8.10	11.60
Switching CNN [15]	4.40	15.70	10.00	11.00	5.90	9.40
ACSCP [14]	2.80	14.05	9.60	8.10	2.90	7.50
CP-CNN [61]	2.90	14.70	10.50	10.40	5.80	8.90
Lingbo et al. [62]	2.60	11.80	10.30	10.40	3.70	7.76
SD-CNN [12]	2.90	10.80	10.10	9.40	3.90	7.42
LSC-CNN [30]	2.90	11.30	9.40	12.30	4.30	8.00
Proposed	2.10	10.47	8.78	9.14	3.54	6.83

Table 4 Comparative analysis with other techniques on UCF-QNRF [34] dataset

Methods	MAE	MSE
MCNN [27]	277	426
Switching CNN [15]	228	445
Zhang et al. [35]	227	426
Idrees et al. [34]	132	191
Idrees et al. [32]	315	508
LSC-CNN [30]	120.5	218.3
Tiny face [33]	336.8	741.6
Proposed	115.2	175.7

Lingbo et al. [62] produces density map by employing regression using recurrent spatial-aware network.

WorldExpo'10 is low dense dataset and state-of-the-art regression-based methods over estimate the crowd count. On contrary, proposed method achieves high performance by precisely detecting the heads. Using dense scale map for generating high-quality scale-aware proposals, the proposed method also precisely localize human heads in low-density scenes. SD-CNN [12] produces comparable results by capturing dense scales in the input image using scale map; however the generation of scale map is manual and requires human efforts. To avoid generating scale map with human efforts, proposed methods utilize the information from multi-scale object detector, i.e., SS-CNN to automatically generate dense scale map.

In Table 4, we compare our results with other methods using UCF-QNRF [34] dataset. From the Table, it is obvious that Tiny Face [33] performs comparatively lower than other state-of-the-art methods. Tiny Face [33] is basically a face detector trained on wide range of face samples. UCF-QNRF contains images, where features of face are hardly visible due to occlusion and severe clutter; therefore it is challenging for Tiny Face detector to detect faces. On the other hand, our proposed method focus on head regions and out performs other methods by precisely detecting heads in both crowded and less dense images of UCF-QNRF dataset. We also observed that state-of-the-art methods produce higher MAE and MSE value compare to proposed framework. The lower performance attributes to the following reasons. (1) These methods exploit global texture information to capture generalized crowd density information. These methods work well in high-density crowds but overestimate the crowd count in low-density situations. (2) Since these methods are regression based, therefore, these methods cannot precisely localize human in the scene.

From above results, it is obvious that our dense prediction module effectively provide coverage of various scales in both high- and low-density crowded images.

We now evaluate the detection performance of our method and compare results with other state-of-the-art methods. For detection performance, we follow the same convention used in [34]. We compute precision and recall rates with various threshold and compute area under the curve to quantify the performance. We use the output of DS-CNN and employ non-maximal suppression method with a fixed threshold. For other methods, we directly use their models to generate density map followed by non-maximal suppression method. We report detection performance of each method in Table 5. From the table, it is obvious that our proposed method out performs other state-of-the-art methods. We also report the visualization of qualitative results of different methods in Fig. 6.

It is also to be noticed that the performance of detection is based on threshold value. Different threshold values change the detection performance and finding the optimal threshold value is hard to find. Therefore, finding optimal strategy of detection for the output of CNN is an important direction for future research.

To evaluate the performance of proposed head detection on other different CNN architectures, we performed experiments on different CNN models in Table 6. Table 6 summarizes the results of different CNN models, i.e., AlexNet [63], ZFNet [64], VGGNet [65], ResNet [66] on three benchmark datasets. From the experiment Table, it is obvious that all CNN models performed comparatively lower than DenseNet. It attributes to the deep architecture of the DenseNet that allows the reuse of residuals and enables smooth flow of gradients throughout the network. This makes the network easy to train with limited number of parameters and achieves high precision rate. On the other hand, AlexNet, VGGNet, ZF are shallow and have large number of parameters (due to FC layers). We observed that these CNN models accumulates many false positives that results in lower precision rates.

We also analyze the computational complexity of our proposed method using all three data sets. We observed that our proposed framework incur computational cost. This is due to reason that our framework processes large number of proposals to generate response map. We found that our framework takes 0.37 s (on average) to process a single image from World-Expo10 dataset. However, the computational cost increase with increase in resolution of image. UCF_CC_50 and UCF-QNRF contain images of high resolutions and it takes 1.34 and 1.76 s (on average) to process images from UCF_CC_50 and UCF-QNRF, respectively.

We also demonstrate the performance of proposed framework using images of different resolutions and the results are reported in Fig. 7. From the Figure, it is obvious that proposed framework precisely localize human heads in all scenes.

Table 5 Localization performance of different methods in terms of Average Precision (Avg), Average Recall (AvR) and Area Under Curve (AUC). The values of AvP and AvR are represented in percentages

Methods	WorldExpo'10			UCF-QNRF			UCF_CC_50		
	AvP	AvR	AUC	AvP	AvR	AUC	AvP	AvR	AUC
MCNN [27]	55.24	52.28	0.51	59.93	63.50	0.59	33.27	35.64	0.31
Liping et al. [59]	65.72	47.91	0.58	71.73	68.68	0.72	34.28	31.19	0.31
SD-CNN [12]	69.46	67.65	0.69	71.27	67.29	0.73	45.67	40.12	0.45
Idrees et al. [34]	–	–	–	75.5	59.75	0.71	–	–	–
Proposed	71.24	68.65	0.71	76.27	63.29	0.73	48.32	42.18	0.47

Fig. 6 Depicts qualitative comparison of proposed method with other state-of-the-art methods. Column represents the prediction of different methods. Rows shows samples frames of benchmark datasets. **1st** row shows the sample frame from UCF_CC_50 dataset. **2nd** shows the sample frame from UCF-QNRF and **3rd** row shows the sample frames from WorldExpo'10 dataset

Table 6 Summary of different CNN architecture and performance comparison on different datasets

CNN model	Model details			Avg Performance		
	# of Params(10^6	# of Conv Layer	# of FC layer	WorldExpo'10	UCF-QNRF	UCF_CC_50
AlexNet	57	5	2	50.43	52.07	15.29
VGGNet	134	13	2	56.78	60.39	25.16
ZF	58	5	2	52.62	54.71	19.93
ResNet	23.4	49	1	64.19	67.46	39.70
DenseNet	18	169	1	71.24	76.27	48.32

(a) **(b)** **(c)**

Fig. 7 Shows the outputs of our proposed framework. **a** shows a sample frame of size 600 × 900 pixels from UCF-QNRF dataset. **b** shows a sample frame of size 2304 × 3456 pixels from UCF_CC_50 dataset and **c** shows a sample frame of size 576 × 720 pixels from WorldExpo'10 dataset

5 Conclusion

In this paper, we proposed a framework to detect human heads with significant large variance in high-density crowds. From experimental results, we observed that our framework shows a trade-off between the detection performance and speed for head detection. The proposed framework uses multibranch SS-CNN to obtain initial detection. We then generate dense scalemap using the information obtained from SS-CNN. The dense scalemap provides full coverage of significant scale variations in the image. Though we utilize point annotations of human heads during training, DS-CNN precisely predict the bounding boxes on human heads. Experiments indicate that the proposed model not only achieves better performance compared to regression methods but also precisely predict the location of human heads in all benchmark datasets. We hope that the proposed method will encourage research community to use detection base approaches instead of regression base approaches.

Acknowledgements This work is supported by National University of Science and Technology, Islamabad, Pakistan. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU for this research.

References

1. Helbing, D.; Mukerji, P.: Crowd disasters as systemic failures: analysis of the love parade disaster. EPJ Data Sci. **1**(1), 7 (2012)
2. Salamati, P.; Rahimi-Movaghar, V.: Hajj stampede in mina, 2015: need for intervention. Arch. Trauma Res. **5**(2), e36308 (2016)
3. Shine, L.; Edison, A.; Jiji, C.: A comparative study of faster R-CNN models for anomaly detection in 2019 AI city challenge. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 306–314 (2019)
4. Mahadevan, V.; Li, W.; Bhalodia, V.; Vasconcelos, N.: Anomaly detection in crowded scenes. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1975–1981. IEEE (2010)
5. Ullah, H.; Altamimi, A.B.; Uzair, M.; Ullah, M.: Anomalous entities detection and localization in pedestrian flows. Neurocomputing **290**, 74–86 (2018)
6. Sultani, W.; Chen, C.; Shah, M.: Real-world anomaly detection in surveillance videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6479–6488 (2018)
7. Khan, S.D.: Congestion detection in pedestrian crowds using oscillation in motion trajectories. Eng. Appl. Artif. Intell. **85**, 429–443 (2019)
8. Dehghan, A.; Shah, M.: Binary quadratic programing for online tracking of hundreds of people in extremely crowded scenes. IEEE Trans. Pattern Anal. Mach. Intell. **40**(3), 568–581 (2017)
9. Idrees, H.; Warner, N.; Shah, M.: Tracking in dense crowds using prominence and neighborhood motion concurrence. Image Vis. Comput. **32**(1), 14–26 (2014)

10. Marsden, M.; McGuinness, K.; Little, S.; O'Connor, N.E.: Resnetcrowd: a residual deep learning architecture for crowd counting, violent behaviour detection and crowd density level classification. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–7. IEEE (2017)
11. Khan, S.D.; Ullah, H.; Uzair, M.; Ullah, M.; Ullah, R.; Cheikh, F.A.: DISAM: density independent and scale aware model for crowd counting and localization. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 4474–4478. IEEE (2019)
12. Basalamah, S.; Khan, S.D.; Ullah, H.: Scale driven convolutional neural network model for people counting and localization in crowd scenes. In: IEEE Access (2019).
13. Cao, X.; Wang, Z.; Zhao, Y.; Su, F.: Scale aggregation network for accurate and efficient crowd counting. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 734–750 (2018).
14. Shen, Z.; Xu, Y.; Ni, B.; Wang, M.; Hu, J.; Yang, X.: Crowd counting via adversarial cross-scale consistency pursuit. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5245–5254 (2018).
15. Sam, D.B.; Surya, S.; Babu, R.V.: Switching convolutional neural network for crowd counting. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4031–4039. IEEE (2017).
16. Davies, A.C.; Yin, J.H.; Velastin, S.A.: Crowd monitoring using image processing. Electron. Commun. Eng. J. **7**(1), 37–47 (1995)
17. Wang, Y.; Lian, H.; Chen, P.; Lu, Z.: Counting people with support vector regression. In: 2014 10th International Conference on Natural Computation (ICNC), pp. 139–143. IEEE (2014)
18. Chan, A.B.; Liang, Z.-S.J.; Vasconcelos, N.: Privacy preserving crowd monitoring: counting people without people models or tracking. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–7. IEEE (2008)
19. Zhang, J.; Tan, B.; Sha, F.; He, L.: Predicting pedestrian counts in crowded scenes with rich and high-dimensional features. IEEE Trans. Intell. Transp. Syst. **12**(4), 1037–1046 (2011)
20. Marana, A.N.; Velastin, S.; Costa, L.; Lotufo, R.: Estimation of crowd density using image processing (1997)
21. Arteta, C.; Lempitsky, V.; Zisserman, A.: Counting in the wild. In: European Conference on Computer Vision, pp. 483–498. Springer (2016).
22. Onoro-Rubio D.; López-Sastre, R.J.: Towards perspective-free object counting with deep learning. In: European Conference on Computer Vision, pp. 615–629. Springer (2016).
23. Zhang, Y.; Zhou, C.; Chang, F.; Kot, A.C.: Multi-resolution attention convolutional neural network for crowd counting. Neurocomputing **329**, 144–152 (2019)
24. Liu, J.; Gao, C.; Meng, D.; Hauptmann, A.G.: Decidenet: counting varying density crowds through attention guided detection and density estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5197–5206 (2018)
25. Shami, M.; Maqbool, S.; Sajid, H.; Ayaz, Y.; Cheung, S.-C.S.: People counting in dense crowd images using sparse head detections. IEEE Trans. Circuits Syst. Video Technol. **29**, 2627–2636 (2018)
26. Gao, X.-S.; Hou, X.-R.; Tang, J.; Cheng, H.-F.: Complete solution classification for the perspective-three-point problem. IEEE Trans. Pattern Anal. Mach. Intell. **25**(8), 930–943 (2003)
27. Zhang, Y.; Zhou, D.; Chen, S.; Gao, S.; Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 589–597 (2016).
28. Hossain, M.; Hosseinzadeh, M.; Chanda, O.; Wang, Y.: Crowd counting using scale-aware attention networks. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1280–1288. IEEE (2019).
29. Li, Y.; Zhang, X.; Chen, D.: CSRNET: Dilated convolutional neural networks for understanding the highly congested scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1091–1100 (2018).
30. Sam, D.B.; Peri, S.V.; Kamath, A.; Babu, R.V.; et al.: Locate, size and count: accurately resolving people in dense crowds via detection. arXiv preprint arXiv:1906.07538 (2019).
31. Yang, S.; Luo, P.; Loy, C.-C.; Tang, X.: Wider face: a face detection benchmark. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5525–5533 (2016).
32. Idrees, H.; Saleemi, I.; Seibert, C.; Shah, M.: Multi-source multi-scale counting in extremely dense crowd images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2547–2554 (2013).
33. Bai, Y.; Zhang, Y.; Ding, M.; Ghanem, B.: Finding tiny faces in the wild with generative adversarial network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 21–30 (2018).
34. Idrees, H.; Tayyab, M.; Athrey, K.; Zhang, D.; Al-Maadeed, S.; Rajpoot, N.; Shah, M.: Composition loss for counting, density map estimation and localization in dense crowds. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 532–546 (2018).
35. Zhang, C.; Li, H.; Wang, X.; Yang, X.: Cross-scene crowd counting via deep convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 833–841 (2015).
36. Hou, Y.-l.; Pang, G.K.: Automated people counting at a mass site. In: 2008 IEEE International Conference on Automation and Logistics, pp. 464–469. IEEE (2008).
37. Kong, D.; Gray, D.; Tao, H.: A viewpoint invariant approach for crowd counting. In: 18th International Conference on Pattern Recognition (ICPR'06), 3, pp. 1187–1190. IEEE (2006).
38. Conte, D.; Foggia, P.; Percannella, G.; Tufano, F.; Vento, M.: A method for counting moving people in video surveillance videos. EURASIP J. Adv. Signal Process. **2010**(1), 231240 (2010)
39. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection (2005).
40. Dollár, P.; Appel, R.; Belongie, S.; Perona, P.: Fast feature pyramids for object detection. IEEE Trans. Pattern Anal. Mach. Intell. **36**(8), 1532–1545 (2014)
41. Wang, X.; Han, T.X.; Yan, S.: An HOG-LBP human detector with partial occlusion handling. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 32–39. IEEE (2009).
42. Zhang, S.; Bauckhage, C.; Cremers, A.B.: Informed haar-like features improve pedestrian detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 947–954 (2014).
43. Hu, Q.; Wang, P.; Shen, C.; van den Hengel, A.; Porikli, F.: Pushing the limits of deep cnns for pedestrian detection. IEEE Trans. Circuits Syst. Video Technol. **28**(6), 1358–1368 (2017)
44. Huang, S.; Ramanan, D.: Expecting the unexpected: training detectors for unusual pedestrians with adversarial imposters. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2243–2252 (2017).
45. Luo, P.; Tian, Y.; Wang, X.; Tang, X.: Switchable deep network for pedestrian detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 899–906 (2014).
46. Mao, J.; Xiao, T.; Jiang, Y.; Cao, Z.: What can help pedestrian detection? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3127–3136 (2017).
47. Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; Li, S.Z.: Occlusion-aware R-CNN: detecting pedestrians in a crowd. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 637–653, (2018).

48. Lin, Z.; Davis, L.S.: Shape-based human detection and segmentation via hierarchical part-template matching. IEEE Trans. Pattern Anal. Mach. Intell. **32**(4), 604–618 (2010)
49. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014).
50. Uijlings, J.R.; Van De Sande, K.E.; Gevers, T.; Smeulders, A.W.: Selective search for object recognition. Int. J. Comput. Vis. **104**(2), 154–171 (2013)
51. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016).
52. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C.: SSD: single shot multibox detector. In: European Conference on Computer Vision, pp. 21–37. Springer (2016).
53. Cireşan, D.; Meier, U.; Schmidhuber, J.: Multi-column deep neural networks for image classification. arXiv preprint arXiv:1202.2745 (2012).
54. Hinton, G.E.; Osindero, S.; Teh, Y.-W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)
55. Idrees, H.; Soomro, K.; Shah, M.: Detecting humans in dense crowds using locally-consistent scale prior and global occlusion reasoning. IEEE Trans. Pattern Anal. Mach. Intell. **37**(10), 1986–1998 (2015)
56. Felzenszwalb, P.F.; Huttenlocher, D.P.: Efficient belief propagation for early vision. Int. J. Comput. Vis. **70**(1), 41–54 (2006)
57. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017).
58. Rodriguez, M.; Laptev, I.; Sivic, J.; Audibert, J.-Y.: Density-aware person detection and tracking in crowds. In: 2011 International Conference on Computer Vision, pp. 2423–2430. IEEE (2011).
59. Zhu, L.; Li, C.; Yang, Z.; Yuan, K.; Wang, S.: Crowd density estimation based on classification activation map and patch density level. J. Neural Comput. Appl. 1–12 (2019).
60. Lempitsky, V.; Zisserman, A.: Learning to count objects in images. In: Advances in Neural Information Processing Systems, pp. 1324–1332 (2010).
61. Sindagi, V.A.; Patel, V.M.: Generating high-quality crowd density maps using contextual pyramid CNNs. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1861–1870 (2017).
62. Liu, L.; Wang, H.; Li, G.; Ouyang, W.; Lin, L.: Crowd counting using deep recurrent spatial-aware network. arXiv preprint arXiv:1807.00601 (2018).
63. Krizhevsky, A.; Sutskever, I.; Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012).
64. Zeiler, M.D.; Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision, pp. 818–833. Springer (2014).
65. Simonyan, K.; Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).
66. He, K.; Zhang, X.; Ren, S.; Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016).