The Visual Computer (2021) 37:2127–2137
https://doi.org/10.1007/s00371-020-01974-7

ORIGINAL ARTICLE

Scale and density invariant head detection deep model for crowd counting in pedestrian crowds

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

Accepted: 4 September 2020 / Published online: 16 September 2020

Abstract

Crowd counting in high density crowds has significant importance in crowd safety and crowd management. Existing state-of-the-art methods employ regression models to count the number of people in an image. However, regression models are blind and cannot localize the individuals in the scene. On the other hand, detection-based crowd counting in high density crowds is a challenging problem due to significant variations in scales, poses and appearances. The variations in poses and appearances can be handled through large capacity convolutional neural networks. However, the problem of scale lies in the heart of every detector and needs to be addressed for effective crowd counting. In this paper, we propose a end-to-end scale invariant head detection framework that can handle broad range of scales. We demonstrate that scale variations can be handled by modeling a set of specialized scale-specific convolutional neural networks with different receptive fields. These scale-specific detectors are combined into a single backbone network, where parameters of the network is optimized in end-to-end fashion. We evaluated our framework on challenging benchmark datasets, i.e., UCF-QNRF, UCSD. From experiment results, we demonstrate that proposed framework beats existing methods by a great margin.

Keywords Dense scales · Crowd counting · Head detection · High density crowds

1 Introduction

For crowd safety and security, it is important to automatically understand high density crowd dynamics in a faster way. However, automated understanding of crowd dynamics is a challenging job. Several efforts have been done during recent years to overcome those challenges. To understand crowd dynamics, crowd counting has gained much attention from research community. Counting the number of people and estimating the distribution of people in the environment provide valuable information for crowd managers.

Considerable amount of work is reported in literature on crowd counting in high density crowds. Most of the existing methods treat the crowd counting problem as regression problem that only estimate the crowd count and avoid localization of individuals in the scene.

Pedestrian detection provides the exact location of individuals in the scene (in terms of bounding boxes), which on the one hand, provides crucial information for crowd dwellers and on the other hand, serve as useful input for other crowd applications, for example, tracking, behavior understanding and anomaly detection. Despite significant importance, limited amount of work is reported in literature to detect pedestrians in high density crowds. The task of pedestrian detection in high density crowds is extremely challenging due to severe clutter, occlusions in the scene. In high density crowds, human bodies are occluded that poses a challenge for a detector to learn consistent human-like features. Generally for crowd surveillance, camera is mounted overhead to provide better coverage of the crowded scenes. In such cases, human head is the only visible part. In this paper, we propose head detection framework that learns head-like features and provides crowd count for an input image as shown in Fig. 1.

Few strides [3,12,14,25] have been made during the recent years to learn consistent features of heads; however, the task of head detection in high density crowds is still an unsolved problem due to the following challenges:

1. Significant variations in the appearances of person heads.
2. Diversity in scales of human heads. Due to perspective distortions, human heads near to the camera appear large, while the far away heads appear small.

✉ Sultan Daud Khan
sultandaud@nutech.edu.pk

[1] National University of Technology, Islamabad, Pakistan

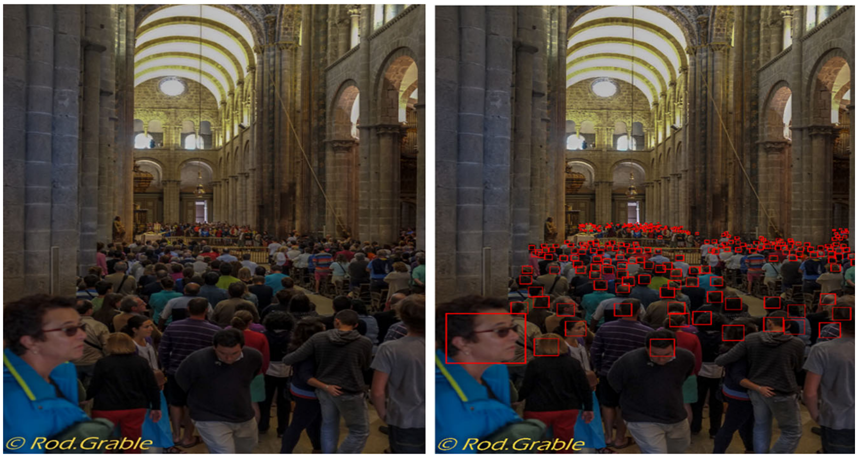

Fig. 1 Shows head detection results of proposed method. The sample input frame (left) is taken from UCF-QNRF [9] dataset and demonstrates significant variations in scales, poses and sizes of heads. Our proposed framework efficiently detects 230/236 heads and precisely estimates the bounding boxes (sizes). It is observed that proposed framework performs better by handling variations in scales, poses and sizes

3. Detecting smaller heads (composed of few pixels) in high density crowds is challenging task for existing generic detectors.
4. Most state-of-the-art detectors perform prediction on down-sampled resolution which is not applicable in fine grained detection of human heads in high density crowds.

The first challenge can be addressed by deep neural networks, since deep neural network are translation invariant and can effectively handle pose and appearance variations. However, the remaining challenges are inter-related and generally caused by perspective distortions. From the empirical evidences, we conclude that camera view point causes perspective distortions due to which the scale of the objects change drastically from one location to other location in the scene.

Currently, existing methods treat head detection problem as special case of object detection. Faster-RCNN [20] and Single Shot MultiBox [15] are the most popular and frequently adopted detectors in detection tasks. To handle scale variation of objects, Faster-RCNN uses anchor boxes of different sizes. However, Faster-RCNN, in current settings, cannot handle significant scale variations and therefore cannot be applied in head detection problem in high density crowds. Single-shot detector (SSD) [15] estimates class probabilities and bounding boxes of objects by employing multi-scale deep features. The multi-scale configuration of SSD detects multiple objects at different scales. We observed that SSD works best for large objects, however, achieved low performance when applied in high density crowds, since the size of object is extremely small.

In this paper, our goal is to precisely predict the bounding boxes of human heads in high density crowds, irrespective of the above mentioned challenges. For this purpose, we present a novel framework that captures broad range of scale variations in an image by splitting broad range of scales into small sets of sub-scales. To model each small sub-scale, we designed a separate scale-specific network that can deal with heads that corresponds to particular sub-scale. This is done by three multiple detectors with three separate region proposal networks (RPNs). After designing the network for each sub-scale, we combine all the networks into a single network and optimize the network parameters in end-to-end fashion.

Generally, our proposed framework has the following contributions:

1. A novel crowd counting framework that counts the number of people in the scene by providing fine-grained detection of human heads at high as well as low resolutions.
2. Based on superior performance achieved on benchmark datasets, the proposed framework provides an alternative solution of crowd counting to the prevalent regression-based crowd counting methods.
3. The network efficiently integrates multiple scale-specific networks into a single end-to-end network.
4. The framework provides inference at single scale and avoids the computational complexity of computing image pyramid.

The rest of this paper is organized as follows. We discusses related work in Sect. 2. Section 3 discusses the details of proposed method. Experiment results are discussed in 4. Section 6 discusses conclusion and future work.

2 Related work

Different methods and approaches are reported in literature for crowd counting and density estimation in high density crowds. Generally, we categorize these methods into two categories, (1) *detection-based methods* and (2) *regression-based methods*.

Currently, regression-based methods prevailed the counting problem that estimate the count by employing regression between crowd features and crowd count. With the tremendous success of CNNs, different models employ CNN to back propagate the regression and update the count loss. However, these do not integrate spatial information in the loss and cannot precisely localize the pedestrians in the crowded scenes. Zhang et al. [30] proposed multi-column convolutional neural network (MCNN) composed of three columns with different receptive fields to handle perspective distortions. Zhang et al. [29] estimated the count from a single image by proposing two-configuration regression model. Sam et al. [21] proposed switching network that chooses one network among multiple CNNs based on the performance. Similarly, Zhu et al. [31] proposed patch-scale discriminant regression network (PSDR) to estimate the crowd count. Sindagi et al. [26] proposed contextual pyramid network generated high-quality density maps and estimated the crowd count by integrating local and global contextual information from the image. Kang et al. [11] provided comprehensive analysis and comparisons of different crowd density estimation methods.

Generally, regression-based methods capture texture information and achieved notable performance in high density crowded scenes; however, these methods have following limitations. (1) Regression-based methods do not incorporate spatial information; therefore, these methods cannot predict the precise location and size (bounding box) of pedestrians in the scene. (2) Regression-based methods usually overestimate the count in low dense crowded scenes.

In order to address the above problems, detection-based methods, *detection-based methods* [22,23,25], train object detectors to predict the location of all pedestrians in the scene. In these methods, total number of detections represent the total count of pedestrians in the scene. Beside regression- and detection-based methods, the authors also proposed hybrid methods that combine both regression and detection-based methods. For example, Liu et al. [14] proposed a hybrid method for crowd counting, where the framework operates in two modes, i.e., regression and detection mode. The framework dynamically decides the appropriate mode depending upon the complexity.

Our propose framework is detection-based method, where we train a head detector with a notion that head is the only visible and reliable part of human body in high density crowds. Unlike other detection methods that tackle the scale problem by generating an image pyramid, we present a novel scale-adaptive framework that splits the target scales into a set of disjoint sub-scales. Each sub-scale set is modeled by separate scale-specific specialized network. These networks are then combined into a single backbone network that jointly optimized the parameters in end-to-end fashion.

Comparison and Difference. Our proposed framework is different in many aspects from the existing detection-based methods. (1) In contrast to scale-invariant detection-based methods, our framework addresses the detection problem by training set of specialized sub-networks with different RPNs. This enables our framework to capture different scale range in the input image. Single-shot detector [15] (SSD) is a cascaded framework, where predictions are generated at every stage to capture certain scales. In this way, samples that are missed in the first stage cannot be recovered in the later stages. In order to deal with this problem, each stage of SSD needs to be generalized to capture large scale variance. Unlike SSD, each scale-specific detector of proposed framework detects human heads fall within a certain scale range. In the same way, different from [2], we integrate a set of scale-specific sub-networks into a single backbone network that is optimized end-to-end. We argue that proposed configuration of framework reduces the computational complexity by sharing parameters and also enhances the detection accuracy by learning discriminating representation of human heads.

3 Proposed methodology

In this section, we discuss the architecture of our proposed framework. The overall architecture of proposed framework is shown in Fig. 2. The input to our framework is an image of arbitrary size, and output is the set of bounding boxes correspond to heads.

The backbone of the proposed framework is based on DenseNet [8] and consists of 174 layers. We use deep network to avoid the problem of gradient vanishing. The network is divided into four stages. The first stage of network consists of one convolutional layer with filter size of 7×7 and stride 2. The convolutional layer is then followed by a pooling layer with filter size of 3×3 and stride of 2. The first stage is followed by three stages, i.e., *denseblock1*, *denseblock2* and *denseblock3*. Dense block implements a set of two convolution layers. The filter size of first convolutional layer is 1×1 followed by second convolutional layer with filter size of 3×3 pixels. As illustrated in Fig. 2, *denseblock1* consists of 12 sets with total of 24 convolutional layers. The second dense block *denseblock2* consists of $24 \times 2 = 48$ layers, and *denseblock3* contains $48 \times 2 = 96$ layers. The output of each dense block passes through a Transition block. The transition block consists of one convolutional layer of filter size 1×1 followed by a pooling layer of size 2×2 with stride 2.

Deep architectures achieved significant success in object classification and detection tasks [10,13,16,20]. These detectors perform well in detecting large objects; however, the performance of these detectors degrades while detecting small objects. Generally, these detectors use feature maps of the last convolutional layer (last layer of *denseblock3*) and

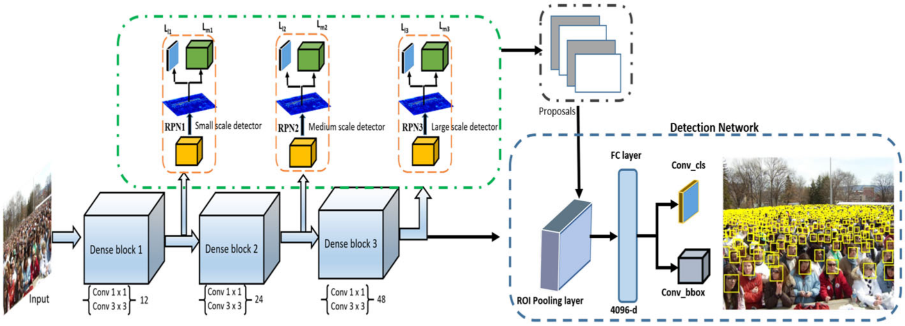

Fig. 2 Shows the pipeline of proposed framework. The framework consists of three scale-specific sub-network with different RPNs that are combined in a single backbone network. The first scale-specific sub-network detects small heads, the second detects medium size heads, and third one detects large heads. The proposals from each sub-network are accumulated and input to another detection network that makes the final prediction

employ region proposal network (RPN) to generate multi-scale object proposals. The size of the feature map of last convolutional layer is small as it is reduced step by step after passing through series of convolutional and pooling layers. Due to small resolution and large receptive field, the feature map of last convolutional layer looses information of small objects. Therefore, these detectors are not suitable for detecting small objects of size less than 32 × 32 pixels. Here, it is to be noted, according to definition of small objects in [27], objects with size smaller than 32 × 32 pixels are considered as small objects. Since we are detecting human heads in high density crowds, the size of head is less than 20 × 20 pixels (approx). These detectors are not applicable.

To detect human heads of small size, we assume that the resolution of shallow layers is large and have small receptive fields, therefore, suitable for detecting small objects. In the same way, we assume that intermediate layers contain information about the medium size objects. The receptive field of last convolutional layer (*denseblock3*) is large and helpful in detecting large objects. Unlike existing methods that use feature map of the convolutional layer of last dense block, we utilize feature map of each of three dense blocks and build three region proposal networks. The first RPN utilizes the feature map of *denseblock1*, the second RPN utilizes the feature map of *denseblock2*, and third RPN uses last dense block, i.e.,*denseblock3*. Through these three branches, RPNs generate multi-scale object proposals. Each dense block combines with RPN and implements a detector with specific receptive field size. This enables each detector to capture specific scales in an image. We set the anchor scale set of first RPN as {10, 16, 32, 56}, {64, 96, 128, 160} for second RPN and for third RPN {165, 212, 256, 512}.

For training, each RPN has its own disjoint set of training samples. Each RPN, samples regions from the input images according to pre-defined anchor scale set as mentioned above. For example, the first RPN samples positive and negative regions with the size range from $10px$ to $56px$ from the input image. We assign positive label to an anchor if the intersection-over-union (IoU) of the candidate region and ground truth is greater than 0.7. Since each RPN has its own disjoint set of samples, therefore, we ignore ground truth regions with size greater than anchor scale set of a particular RPN. It is to be noted that a single ground truth region may assign positive label to multiple anchors. Negative values are assigned to anchors with IoU less than 0.3. We also ignore those anchors that do not contribute to the training loss. Usually, these anchors belong to the region outside the boundary of the given image. However, each RPN will generate two type of outputs, i.e., bounding boxes and classification score. Therefore, we use multi-task loss function and minimize the following objective function.

$$L(l_j, m_j) = \frac{1}{M_{\text{class}}} \sum_{j=1}^{N} L_l(l_j, \hat{l_j}) + \Omega \frac{1}{M_{\text{regress}}} \sum_{j=1}^{N} L_m(m_j, \hat{m_j}) \quad (1)$$

where N is the number of samples per mini-batch. j represents the index number of an anchor. l_j and m_j represent the predicted class probability and bounding box, respectively. $\hat{l_j}$ and $\hat{m_j}$ represent the ground truth class label and bounding box, respectively. During RPN training process, $\hat{l_j}$ takes either value 1 or zero. The value “1” represents the positive class, while negative class or background is represented by “0.” L_l is the log class loss, and L_m is log regression loss. In Eq. 1, multi-task terms are normalized by M_{class} and M_{regress}, while Ω is a balancing parameter.

During training, we generate mini-batch of positive and negative samples from a single image. From empirical studies, we observed that training RPN with all samples generated from a single image cause the network bias towards negative samples, since the number of negative samples (or background) is greater than positive samples. In order to address this problem, we generate a mini-batch of 256 samples by randomly selecting positive and negative samples with the ratio of 1:1.

We employ Xavier initialization [6] to initialize all the layers. We keep learning rate of 0.001 with the decrease the learning rate by rate of 10 after every 10k iterations.

3.1 End-to-end training

In the above section, we discussed how to train multiple RPNs with different scale sets. The output of these RPNs is a set of bounding boxes of different sizes with different class labels. Now, we discuss how to utilize these scale-specific region proposals for head detection task. More precisely, we describe the algorithm that learns end-to-end network composed of multiple RPNs and detection network. Multiple RPNs and detection network trained independently will modify and update convolutional layers in their own ways. Therefore, we need to develop a method that allows the network to train end-to-end by sharing convolutional layers.

The proposals obtained from multiple RPNs are of different sizes. As per requirement of fully connected layer, the obtained proposals need to be converted to fixed size before feeding fully connected layer. Region-of-interest (ROI) pooling does this job by taking feature map from the *denseblock3* and proposals obtained from multiple RPNs as inputs as shown in Fig. 2. ROI pooling layer takes every region proposal and extracts a patch from the feature map that corresponds to that region proposal and converts it to feature map of fixed size.

The final prediction of bounding boxes and class labels is done by the detection network. The detection network has two sibling layers, one layer outputs class probability and second layer outputs a tuple that represents the offsets of predicted bounding boxes. Since we are optimizing two tasks, i.e., class label predication and prediction of bounding box offsets, therefore, we define two different loss functions. Let $\hat{c}$ represents predicted class label and c represents the ground truth class. We define class loss L_{class} as negative log-likelihood and formulated as in Eq. 2.

$$L_{\text{class}}(\hat{c}, c) = -(\log \hat{c}) \tag{2}$$

The class loss L_{class} maximizes the class probability by incentivizing the model when it predicts the positive class with higher probability and penalizing the model when it predicts the positive class with smaller probabilities. The penalizing part is done by the logarithm. The purpose of the negative sign is to make the loss value positive, since the values of class probability lie in range [0, 1] and logarithm of values in this range is negative.

The second loss $L_{\text{bbox}}(\hat{b}, b)$ for predicting the offsets of bounding boxes is defined over ground truth tuple $b = (b_x, b_y, b_w, b_h)$ and predicted tuple $\hat{b} = (\hat{b}_x, \hat{b}_y, \hat{b}_w, \hat{b}_h)$, where b_x and b_y represent the location, and b_w and b_h represent the width and height of the ground truth bounding box. The loss L_{bbox} is formulated as in Eq. 3

$$L_{\text{bbox}}(\hat{b}, b) = \sum_{j \in \{x,y,w,h\}} L_1(\hat{b_j}, b_j) \tag{3}$$

where $L_1(\hat{b_j}, b_j)$ is Huber loss and formulated as in equation

$$L_1(\hat{b}, b) = \begin{cases} 0.5(\hat{b} - b)^2, & \text{if } \left|\hat{b} - b\right| < 1 \\ \left|\hat{b} - b\right| - 0.5 & \text{otherwise} \end{cases} \tag{4}$$

We combine both losses L_{class} in Eq. 2, L_{bbox} in Eq. 3 and train detector using the following joint loss L in Eq. 4

$$L(\hat{c}, c, \hat{b}, b) = L_{\text{class}}(\hat{c}, c) + L_{\text{bbox}}(\hat{b}, b) \tag{4}$$

3.2 Significance of joint optimization

We observe that existing similar detectors [17,18] produce redundant detection by training multiple detector independently. However, our framework reduces this redundancy by sharing representation of heads among scale-specific detectors.

As discussed above, our framework combines all scale-specific detectors into a single backbone network and jointly optimize network parameters in end-to-end fashion. In order to see the significance of joint optimization, we compare the results with one of our framework variant that has similar network structure but do not jointly optimize the network parameters in end-to-end fashion. In this method, the detection obtained from each stage is accumulated to generate final detection. We call this method "straightforward" method. From experiments, we observe that "straightforward" method accumulates redundant predictions and results in low average precision. The comparison of "straightforward" method and proposed framework is shown in Fig. 3.

We also reduce the parameters of network to improve run time efficiency. We keep minimum number of filters in each layer of backbone network and initialize the network parameters by pre-training the network on ImageNet [5] dataset.

4 Experiment results

In this section, we evaluate and compare the performance of proposed framework with existing methods in both qualitative and quantitative ways. To evaluate the effectiveness of our proposed framework, we use two benchmark datasets, i.e., UCSD dataset [4] and UCF-QNRF [9]. These datasets include images collected from different scenes with varying camera view points, illumination and densities. UCSD

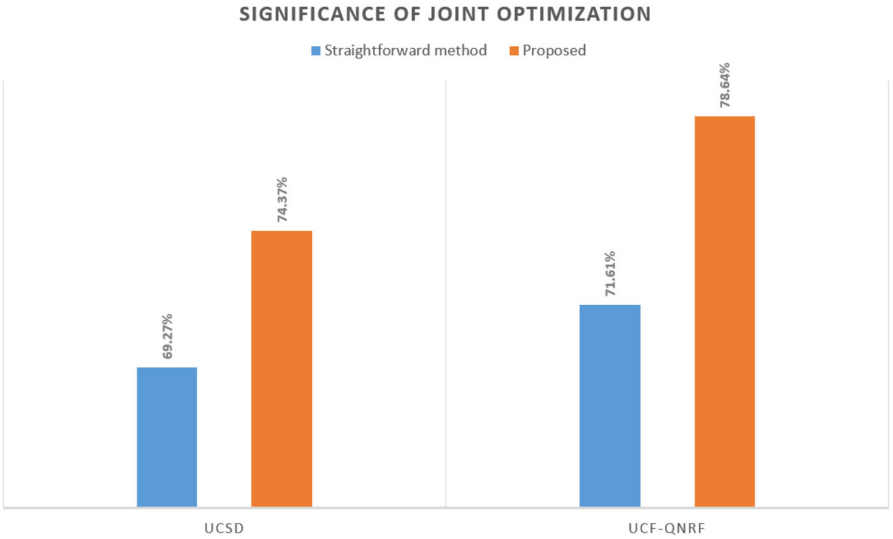

Fig. 3 Evaluation of "straightforward" method with proposed on both datasets

dataset covers low density situations, where the average count per frame is 25, while UCF-QNRF covers high density crowded scenes, where average count per frame is 815. These datasets are carefully selected among existing benchmark datasets to evaluate the performance of proposed framework in both high and low density crowded scenes. Most of existing regression-based crowd counting methods use these datasets for crowd counting. However, these dataset have never been used for head detection-based crowd counting methods. However, these datasets contain dot annotations and only suitable for evaluating regression-based models. To use these datasets for head detection problem, we annotate human heads with bounding boxes with aspect ratio of 1:1.

To comprehensively evaluate the performance of proposed method, we divide the experiment setup in two parts. The first part of the experiment discusses detection performance of proposed framework, while the second part discusses the counting performance.

4.1 Detection performance

Detection provides crucial information by precisely localizing human head in the scene. In detection performance, we measure how precisely the model detects the bounding box of objects. Therefore, it is important for a good detector to precisely localize human heads in the scene. The detection performance is usually measured by Intersection over union (IoU) which quantifies the overlap between the predicted and ground truth bounding boxes. Generally, fixed threshold value (0.5) is used for IoU. However, we observed that with fixed threshold value, the performance of the detector cannot be evaluated with different threshold values. To measure the detector performance, we use mean average precision (mAP) as evaluation metric predominately used to assess detector's performance. The detection performance of different methods is summarized in Table 1. It is to be noted that we directly use the pre-trained models of reference methods for comparisons. From the table, it is obvious that our framework achieved better results compared to existing methods.

Table 1 Detection performance of different methods in terms of mean average precision (mAP) using UCSD and UCF-QNRF datasets

Methods	UCSD [4] (%)	UCF-QNRF [9] (%)
TinyFace [2]	38.49	54.41
DecideNet [14]	34.24	50.00
SSD [15]	21.16	31.88
Faster-RCNN [20]	17.83	24.32
FCHD [28]	30.76	39.80
MCNN [30]	42.95	59.93
DenseNet63 [8]	58.62	70.20
Encoder-Decoder [1]	63.75	71.82
Idrees et al [9]	65.22	75.89
HR [7]	50.94	69.51
Proposed	68.53	79.56

To evaluate the effectiveness of each scale-specific detector, we categorize human heads into three groups based on height of image, i.e., small, medium and large. The small group corresponds to heads of size ranges from 8–60 pixels, medium (60–160) pixels and large corresponds to 160–256 pixels. We evaluate the performance of existing methods on each group in terms of mean average precision. The performance of methods is summarized in Table 2. From the table, it is obvious that all detectors achieve impressive performance on both medium and large groups. However, the performance of detectors degrades when applied on small group. From the table, it is obvious that there is a considerable gap between the performance of detectors on small and

Table 2 Performance of different detectors on small, medium and large group from UCF-QNRF dataset

Methods	Small (%)	Medium (%)	Large (%)	Average (%)
TinyFace [2]	40.29	55.64	67.29	54.41
DecideNet [14]	37.29	54.37	58.33	50.00
SSD [15]	25.23	32.76	37.65	31.88
Faster-RCNN [20]	20.76	21.39	30.82	24.32
FCHD [28]	33.64	40.29	45.48	39.80
MCNN [30]	47.83	60.34	71.62	59.93
DenseNet63 [8]	65.73	69.76	75.1	70.20
Encoder-Decoder [1]	67.47	70.58	77.42	71.82
Idrees et al. [9]	68.76	79.57	79.29	75.89
HR [7]	55.93	75.32	77.27	69.51
Proposed	72.34	82.41	83.94	79.56

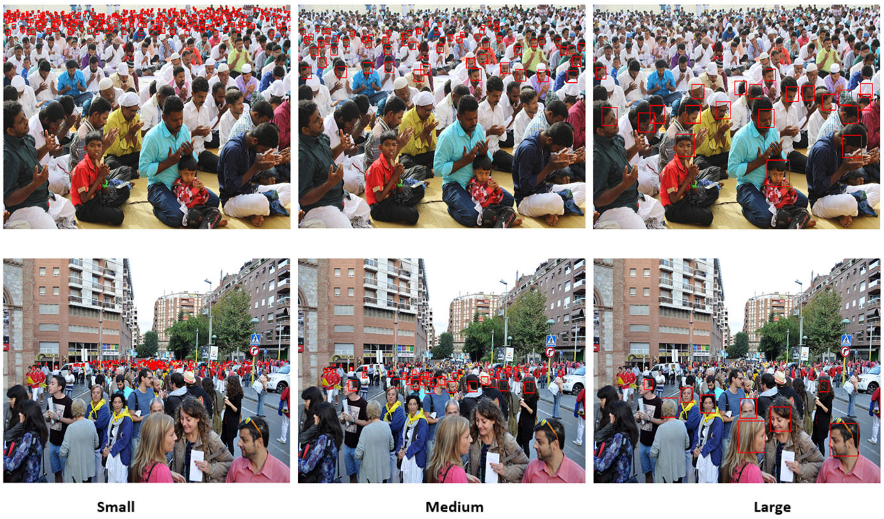

Fig. 4 Results of proposed framework at different stages using sample frames from UCF-QNRF dataset

medium/large head sizes. It attributes to small size of heads that occupy few pixels and lack of appearance information.

From Table 2, it is obvious that Faster-RCNN achieves lower performance compared to other reference methods. This is due to reason that Faster-RCNN [20] 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 [15], 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, it has less discriminating power that ultimately leads to significant amount of false positives. FCHD [28] employs fully convolutional network (FCN) that takes arbitrary size image as input and use feature map of the last convolutional layer for predicting class labels and bounding boxes. Since this method also uses last convolutional layer, therefore, it can detect heads near to camera (due to large size) and miss heads that are far from the camera. DecideNet [8] employs two sub-networks, i.e., RegNet and DetNet. The architecture of RegNet is based on FCN and DetNet follows the typical pipeline architecture of Faster-RCNN. This is due to reason that DecideNet faces difficulty in detecting small heads. HR [7] solves the multi-scale problem by using image pyramid, where image is re-scaled to different size before feeding to the network. This method achieves comparable results; however, it suffers from following limitations. (1) Processing each level of pyramid is computationally expensive. In some cases, the resolution of up-sampled image reaches to 5000 pixels per one side that significantly increase the inference time. (2) Down-sampling the image results in loss of information about the small objects. This is the reason that HR performs relatively lower than proposed method. On the other hand, proposed method achieves state-of-the-art performance on both benchmark datasets. We solve the multi-scale problem by employing scale-specific detectors that detects human heads at different range of scales and does not require image pyramid.

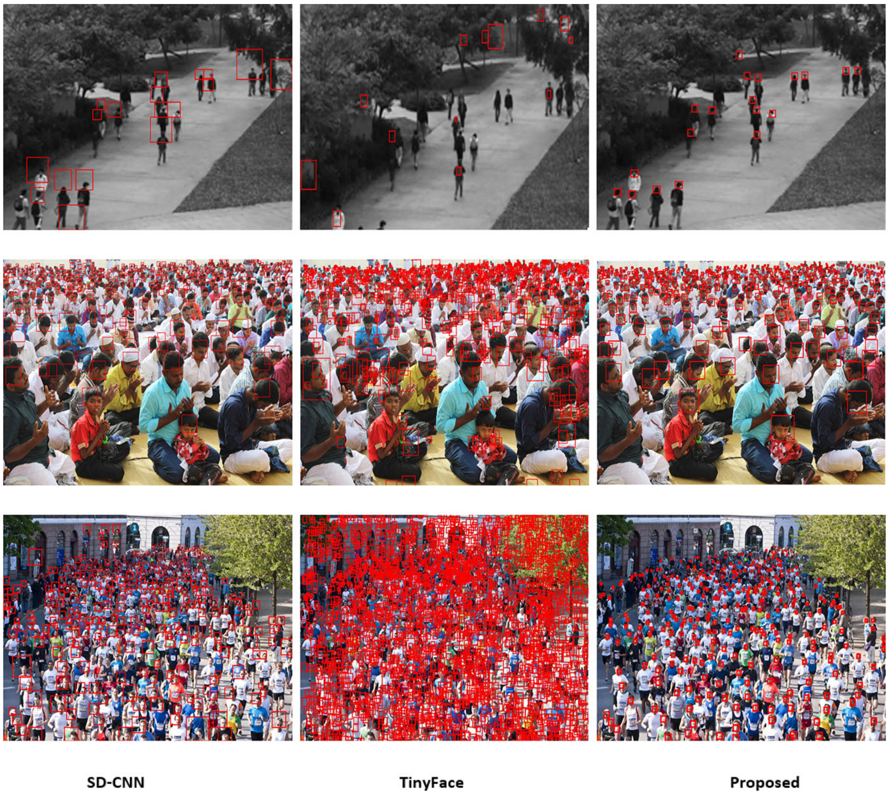

Fig. 5 Shows performance of different method on different sample frames. The first row shows the results of different methods on UCSD dataset. Second and third rows show the results of different methods using UCF-QNRF dataset

To visualize the performance of framework at different stages on small, medium and large group, we report qualitative results in Fig. 4. From the figure, it is obvious that each scale-specific detector can precisely localize and estimate the respective bounding boxes of human heads.

We also demonstrate the detection performance of different methods in Fig. 5. From the figure, it is obvious that performance of SD-CNN [3] is comparable to proposed method by predicting the location of human heads; however, the methods fails to estimate the exact bounding boxes correspond to human heads. On the other hand, TinyFace [2] accumulates large number of redundant bounding boxes around human heads. Furthermore, it also produces many false positives as obvious from the figure. Our proposed framework, on the other hand, not only precisely localize human heads but also estimate the exact sizes of bounding boxes.

4.2 Counting performance

We next evaluate the counting performance of proposed method and its comparison with other state-of-the-art methods. For evaluating counting performance, we use the same convention of mean absolute error (MAE) and mean square error (MSE) followed in state-of-the-art crowd counting methods. We report the performance of different methods in Table 3. It is obvious from the table that regression-based methods achieve comparable performance on UCF-QNRF dataset due to high density images, since these models capture regular repetitive structures (texture) in the crowd. However, the performance of regression-based methods degrades on UCSD dataset, as these methods overestimate the count in low density crowds.

Table 3 Counting performance of different methods using MAE and MSE on UCSD and UCF-QNRF datasets

Methods	UCSD [4]		UCF-QNRF [9]	
	MAE	MSE	MAE	MSE
Switching CNN [21]	1.62	2.10	228	445
MCNN [30]	1.07	1.35	227	426
Idrees et al. [9]	–	–	132	191
TinyFace [2]	1.78	2.45	120	187
SD-CNN [24]	1.01	1.28	115	175
Proposed	0.97	1.12	112	173

On the other hand, detection-based methods are unable to produce the desired results on UCF-QNRF dataset. This attributes to small head size and occlusions in high density crowded scenes. However, we notice that on UCSD dataset, detection-based methods achieve relatively high performance. However, most parts of human body were visible. Our proposed method overcome the limitations of regression based model by precisely detecting human heads in both

Table 4 Computation complexity of different methods in UCF-QNRF dataset

Methods	Average precision (%)	Inference time (ms)	Frames/s (fps)
Faster-RCNN [20]	24.32	1,200	0.83
SSD [15]	31.88	850	1.75
YOLO [19]	35.47	540	1.85
TinyFace [2]	54.41	660	1.55
SD-CNN [24]	73.19	1,600	0.62
Proposed	79.56	580	1.72

Table 5 Computation complexity of different methods in UCSD dataset

Methods	Average precision (%)	Inference time (ms)	Frames/s (fps)
Faster-RCNN [20]	17.83	827	1.20
SSD [15]	21.16	415	2.41
YOLO [19]	25.17	320	3.12
TinyFace [2]	38.49	670	1.49
SD-CNN [24]	64.19	987	1.01
Proposed	68.53	360	2.77

low and high density crowds. As obvious from the table, our framework achieves better results compared to existing related methods.

5 Computation complexity

In this section, we evaluate and compare the inference speed of proposed framework. All the models are trained and tested using NVIDIA Titan Xp GPU. We take the average inference time of randomly selected images from UCF-QNRF dataset. This dataset consists of images of varying high resolutions and densities that we believe can affect the inference time. We compare the performance with other related methods in terms of average precision and inference time. The performance of different methods is reported in Tables 4 and 5. From the Table 4, it is obvious that in UCF-QNRF dataset, proposed framework achieves 79.56% with 1.72 frames per second. On the other hand, Yolo comparatively achieved high frame rate but average precision is dropped to 35.47%, significantly lower than proposed framework. SD-CNN, on the other hand, achieved comparable performance, but with generation of large number of scale-aware proposals, inference time is very high compared to other methods.

From Table 5, it is obvious that YOLO is faster than proposed method; however, it achieved lower average precision value compared to proposed method. Furthermore, SD-CNN achieved comparable results in terms of average precision, but cause high computational cost. From Tables 4 and 5, we further observed that methods run faster and perform lower on UCSD dataset compared to UCF-QNRF dataset. From the empirical evidences, we observed that resolution of an image affects the inference time and accuracy. The images in UCSD dataset is of low resolution, where average size of head is around 8 × 8 pixels. This is due to the reason that most reference methods could not precisely localize the heads.

6 Conclusion

In this paper, we proposed a unified framework to detect human heads with wide range of scale variance. Our framework achieved better performance with minimum computational cost. We demonstrated through experiments that best performance can be achieved through integration of different scale-specific detectors. It is also demonstrated that the proposed framework achieves better performance than its counterpart, regression-based models. We also evaluate the performance of different scale-specific detectors in detecting human heads fall in their respective scale range. We hope that these encouraging results will motivate the research to adopt detection-based models instead of regression models. These results can provide a useful to many other crowd applications like tracking, crowd behavior understand and anomaly detection.

Acknowledgements We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU for this research.

References

1. Badrinarayanan, V., Kendall, A., SegNet, R.C.: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561 (2015)
2. 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)
3. Basalamah, S., Khan, S.D., Ullah, H.: Scale driven convolutional neural network model for people counting and localization in crowd scenes. IEEE Access **7**, 71576–71584 (2019)
4. Chan, A.B., Vasconcelos, N.N.: Counting people with low-level features and Bayesian regression. IEEE Trans. Image Process. **21**(4), 2160–2177 (2011)
5. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L..: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
6. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256 (2010)

7. Hu, P., Ramanan, D.: Finding tiny faces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 951–959 (2017)
8. 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)
9. 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)
10. Jin, M., Li, H.: Feature-enhanced one-stage face detector for multiscale faces. J. Electron. Imaging **29**(1), 013006 (2020)
11. Kang, D., Ma, Z., Chan, A.B.: Beyond counting: comparisons of density maps for crowd analysis tasks-counting, detection, and tracking. In: Transactions on Circuits and Systems for Video Technology (IEEE TCSVT) (2018)
12. 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)
13. Li, Y., Sun, B., Wu, T., Wang, Y.: Face detection with end-to-end integration of a convnet and a 3D model. In: European Conference on Computer Vision, pp. 420–436. Springer, Berlin (2016)
14. 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)
15. 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, Berlin (2016)
16. Mliki, H., Dammak, S., Fendri, E.: An improved multi-scale face detection using convolutional neural network. Signal Image Video Process. **14**, 1–9 (2020)
17. Qin, H., Yan, J., Li, X., Hu, X.: Joint training of cascaded CNN for face detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3456–3465 (2016)
18. Ranjan, R., Patel, V.M., Chellappa, R.: Hyperface: a deep multitask learning framework for face detection, landmark localization, pose estimation, and gender recognition. IEEE Trans. Pattern Anal. Mach. Intell. **41**(1), 121–135 (2017)
19. 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)
20. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
21. 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)
22. Saqib, M., Khan, S.D., Sharma, N., Blumenstein, M.: Crowd counting in low-resolution crowded scenes using region-based deep convolutional neural networks. IEEE Access **7**, 35317–35329 (2019)
23. Saqib, M., Khan, S.D., Sharma, N., Blumenstein, M.: Person head detection in multiple scales using deep convolutional neural networks. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–7. IEEE (2018)
24. Saqib, M., Khan, S.D., Sharma, N., Blumenstein, M.: Crowd counting in low-resolution crowded scenes using region-based deep convolutional neural networks. IEEE Access **7**, 35317–35329 (2019)
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. 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)
27. Tong, K., Wu, Y., Zhou, F.: Recent advances in small object detection based on deep learning: a review. Image Vis. Comput. **97**, 103910 (2020)
28. Vora, A., Chilaka, V.: FCHD: fast and accurate head detection in crowded scenes. arXiv preprint arXiv:1809.08766 (2018)
29. 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)
30. 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)
31. 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. **32**, 1–12 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Sultan Daud Khan received the B.Sc. degree (Hons.) in computer engineering from the University of Engineering and Technology, in 2005, the M.Sc. degree (Hons.) in electronics and communication engineering from Hanyang University, South Korea, in 2010, and the Ph.D. degree in computer science from the University of Milano-Bicocca, in 2016. He is currently an Associate Professor in the Department of Computer Science, National University of Technology, Pakistan. He has published several papers in conferences and journals, such as AVSS, IVCNZ, ICGIP, Neurocomputing, the Journal of Cellular Automata, and IEEE ACCESS. His research interests include crowd analysis, action recognition and localization, object detection, visual tracking, multi-camera, and airborne surveillance using deep learning techniques. He received the Best Reviewer Award from Pattern Recognition, in 2017. He is also an active Reviewer of prestigious journals, Neurocomputing, Pattern Recognition, the IEEE IET SIGNAL PROCESSING, ACM Multimedia, IEEE ACCESS, and ACM TOMM.

Saleh Basalamah is an Associate professor at Umm Al-Qura University. He has an M.Sc. from the University of Bristol and a Ph.D. from Imperial College London. He is currently working as an Associate professor at the Umm Al-Qura University, Kingdom of Saudi Arabia. His research interests include computer vision and multimedia.