Going Beyond the Regression Paradigm with Accurate Dot Prediction for Dense Crowds

Deepak Babu Sam* Skand Vishwanath Peri* Mukuntha N. S. R. Venkatesh Babu
Indian Institute of Science, Bangalore, India-560012

Abstract

We present an alternative to the paradigm of density regression widely being employed for tackling crowd counting. In the prevalent regression approach, a model is trained for mapping images to its crowd density rather than counting by detecting every person. This framework is motivated from the difficulty to discriminate humans in highly dense crowds where unfavorable perspective, occlusion and clutter are prevalent. Though regression methods estimate overall crowd counts pretty well, localization of individual persons suffers and varies considerably across the entire density spectrum. Moreover, individual detection of people aids more explainable practical systems than predicting blind crowd count or density map. Hence, we move away from density regression and reformulate the task as localized dot prediction in dense crowds. Our dot detection model, DD-CNN, is trained for pixel-wise binary classification to detect people instead of regressing local crowd density. In order to handle severe scale variation and detect people of all scales with accurate dots, we use a novel multi-scale architecture which does not require any ground truth scale information. This training regime, which incorporates top-down feedback, helps our model to localize people in sparse as well as dense crowds. Our model delivers superior counting performance on major crowd datasets. We also evaluate on some additional metrics and evidence superior localization of the dot detection formulation.

1. Introduction

Crowd counting from images, especially dense crowds, has acquired a lot of academic as well as practical interest. While the pragmatic interest is driven by the need to quickly analyze large gatherings for security and planning reasons, the academic involvement is due to the great challenge posed by the problem. The major difficulty is attributed to the extreme variation in appearance of people ranging from large faces to heads occupying a few pixels in dense regions. Add to these the pervasive occlusions, pose variations and background clutter. Consequently, counting by detecting humans is perceived difficult [9], especially to scale satisfactorily across the whole density spectrum seen in typical crowd scenes. As a relatively easy solution, dense crowd counting is normally posed as a density regression problem. Here the idea is to annotate the location of every head, which are then converted to crowd density by convolving with a Gaussian kernel of a fixed spread. The kernel is chosen such that the integration of the map directly gives out crowd count. Since modern crowd counting approaches [4, 11, 16, 20, 28] employ Convolutional Neural Networks (CNN), density maps ease the training process as the task of predicting exact point of head annotation is reduced to regressing local density.

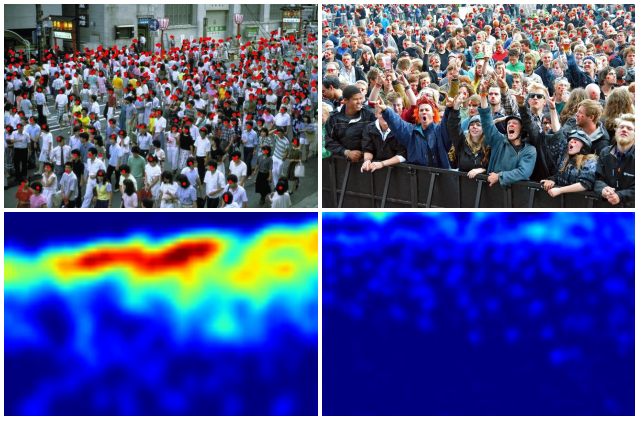

Figure 1. Dot Detection Vs Density Regression. The top row shows crowds with dot predictions from the proposed DD-CNN, while bottom row has corresponding density maps. The dot detection has better localization of individuals across density ranges.

Recent works have made huge progress in devising new architectures and algorithms to improve performance of density regression based approaches. The major metric for performance evaluation of counting models only considers overall count estimation and does not account localization of prediction on to individual humans. Though these methods deliver good count accuracy for a given crowd scene,

*equal contribution

the localization seems poor for further downstream applications. This is because the density map describes the people count in local regions and hence the focus is not to accurately locate each person. Moreover, the notion of density makes more sense when people are relatively closer as in highly dense crowds. The density surface in sparser crowds have frequent discontinuities and the values over human heads are mostly near zero. This is evident in Figure 1, where the density peaks on large faces are spread out, indicating practically almost no detection in sparse region. Also note that one could not consistently find local peaks in these regions to localize persons. This is largely true irrespective of Gaussian kernel parameters used for ground truth density map creation. Furthermore, the local peaks might not accurately correspond to the location of persons (except in certain density ranges) as it is trained for regressing density in a local region rather than to pinpoint people. Hence, any simple method to post-process density maps for better localization might not scale equally across the entire density range (See Section 5.2). In contrast, ideally one would expect spot-on predictions on people at all scales. Such a system facilitates applications other than computing mere counts. From accurate dot detections, faces and features can be extracted for other purposes, which is cumbersome with density maps. Above all, individual detection of people facilitates a more explainable and practical AI system.

Hence, in this work, we try to break the 'traditional' paradigm of training for density regression and replace it with accurate dot detection framework. We define the problem statement as to predict localized dots over the head of any person irrespective of the scale, pose or other variations. Additionally, this has to be done without any bounding box annotations, but only with point annotations available with crowd datasets. There are many challenges in achieving such a goal; the major one being the extreme scale and density variation in crowd scenes. In normal detection scenarios, this is trivially done using a multi-scale architecture, where images are fed to the model at different scales and trained. However, such a naive approach is not possible in our case since there is no ground truth scale information (through bounding boxes) available with the crowd dataset, instead only point annotation are present. Furthermore, the multi-scale architecture has to deal with large variation in appearance of people across scale. A lower scale person simply is not a rescaled version of a large face, but looks drastically different. In sparse crowd, facial features may be visible, but in highly dense crowd people are only seen as blobs. These pose challenges for dense dot detection.

We devise a Dot Detection CNN model, named DD-CNN for the proposed challenging problem. The basic idea is to train the CNN model for pixel-wise binary classification task of detecting people. Cross entropy loss is used instead of l_2 regression employed in density estimation. DD-CNN is optimized in a multi-scale architecture which does not require ground truth scale information, but uses only point supervision. In summary, this work contributes:

- A new training paradigm of dot detection for crowd counting, dropping the prevalent density regression.
- A unique multi-scale fusion architecture that facilitates highly localized detection of people in dense crowds.
- A novel training regime that only requires point supervision and delivers impressive performance.

2. Previous Work

Many early works in crowd counting rely on detection based frameworks. For instance, works like [22, 25, 26], use motion and appearance features to detect individual persons. The recurrent network of [21], sequentially count and detect people. Though there are numerous works on face detection like [8, 13, 15, 29], they are not well suited for crowd counting, which is characterized by people at any pose with high chance of faces being occluded. Moreover, these methods fail in highly dense crowds. The features needed for discrimination in a dense crowd is completely different from a sparse gathering. Above all, most of the face detection approaches require bounding box annotation, which is not available with counting datasets. Consequently, density regression based methods took the stage with significantly better performance. Idrees et al. [9] regress crowd count using a combination of head detection and features from interest points along with frequency analysis. Soon regression models are adapted to deep learning framework with the work of [27], where a CNN is trained to predict crowd density map with an additional task of estimating direct count. Though there are models which directly regress crowd count [24], they fail to learn enough good features due to lack of spatial information (available in density map) and delivers inferior performance than training for density map prediction. Handling diversity in the crowd images is one of the key to improve performance as evident from body of works leveraging multiple CNNs. The cascade of regressors by [23], tries to correct density prediction made by the previous network. Onoro et al. [16] use multiple networks, each trained with images of different scales and the outputs are fused. These multi-scale techniques are outperformed by multi-column architectures. The idea is to employ multiple CNN columns with different receptive fields tuned for different scales [5, 28]. The performance of multi-column approaches are further improved by specializing the CNN columns aggressively through a differential training procedure [2, 4]. Continuing the trend, having auxiliary information indicating the scale or density of the crowd at a local as well as global level (through dedicated classifiers) leads to better prediction as shown

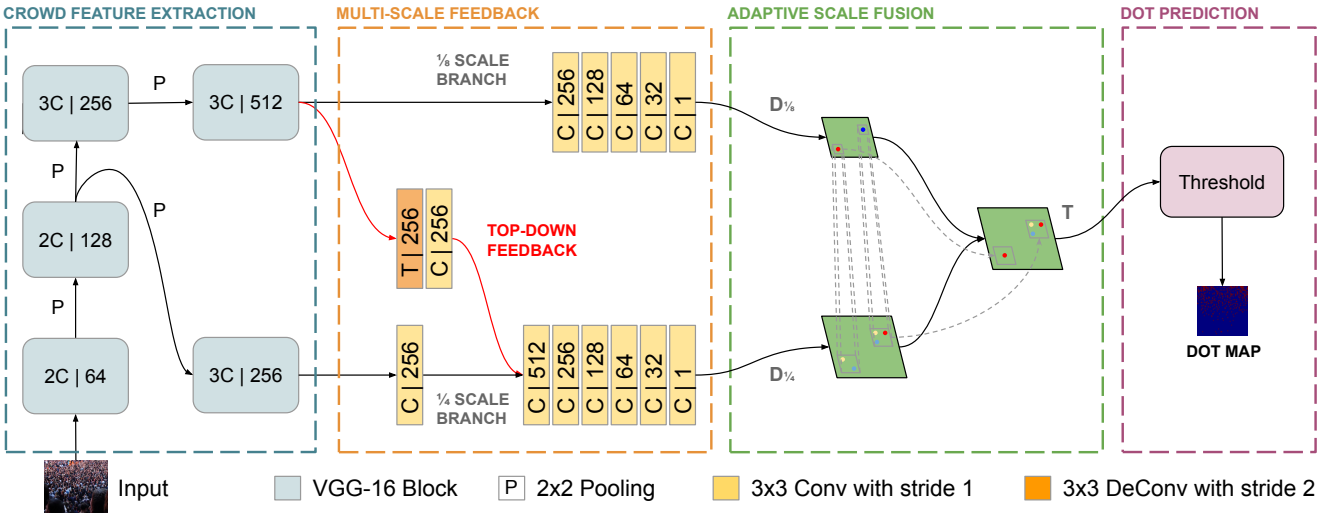

Figure 2. The architecture of the proposed dot detection network. DD-CNN has a multi-scale architecture with dot predictions at different resolutions, which are combined through *Adaptive Scale Fusion*. The networks are trained with pixel-wise binary cross-entropy loss.

in [19, 20]. The architecture by [6] combines multi-scale features and is trained with additional local pattern consistency loss. In contrast to these approaches, Babu Sam et al. [1] develop a top-down feedback mechanism that can iteratively improve density prediction made by a CNN regressor. On similar lines, iterative density estimation is done at increasing resolution using features and prediction of previous networks [17]. Liu et al. [14] try to address the issue of annotation difficulty by leveraging unlabeled data with an additional task of count ranking in a multitask framework. The Grid Winner-Take-All autoencoder in [3] trains almost 99% of the model parameters without using any crowd annotation. Furthermore, VGG based networks with dilated convolution layers are shown to be better by [11]. DecideNet [12] model combines density regressor with a Faster R-CNN and tries to improve density prediction by adaptively switching between the two. But the performance is evident only on sparse crowd and is not evaluated in dense datasets. Note that this is not a detection work, but improves regression with a detector and requires some bounding box annotation for training the detectors as well. Idrees et al. [10] use DenseNet with composition loss and train to predict densities at different resolution. They threshold the predicted density maps to get dot detections, which leads to a drop in counting performance (Sect 5.2). Contrary to all these approaches, we completely eliminate regression loss and train the model for per-pixel binary classification.

3. Our Approach

In Section 1, we have motivated the paradigm shift from density regression to dot detection. The basic objective is to predict highly localized points on heads of people. At a high-level view, this is a dense classification task, where at each pixel the model has to predict the presence of a person irrespective of the scale, pose or other variations. Figure 2 illustrates our proposed solution, the dot detection framework DD-CNN. DD-CNN is composed of four functional modules; the first *Crowd Feature Extraction* network converts the input crowd scene to rich features at multiple resolutions. Then this feature set is processed by *Multi-Scale Feedback* module, which correlates multi-scale information to generate predictions at multiple resolutions. Subsequently, the novel *Adaptive Scale Fusion* module combines the multi-scale predictions into single map, where each value indicates the confidence of person detection. A threshold is applied on this map to generate the final accurate dot predictions. The following sections describes in detail each functional modules as well as the training regime.

3.1. Crowd Feature Extraction

Good features form backbone of any vision systems. It has been recently shown that VGG-16 [18] based networks work well for crowd feature extraction and achieve state-of-the-art performance [11]. Following the trend, we employ the first four 3×3 convolutional blocks from VGG-16, which are initialized with ImageNet trained weights. The input to the network is a three channel image of fixed size 224×224. Due to max-pooling, the resolution of feature maps halves every block. After the second max-pooling, the network branches into two, with the third block being replicated in both. The third block is copied so that the two branches specialize by sharing low-level features without any conflict. The two branches give out feature map sets at different resolutions. One set has size one-fourth that of the input image and is meant to resolve relatively dense crowd

features. The other one-eighth resolution feature maps are for discriminating sparse crowd and large faces as they have higher receptive field. These multi-scale feature sets are used by the subsequent modules to make dot prediction.

3.2. Multi-Scale Module

The feature extraction blocks are followed by two columns of CNN for processing the multi-scale feature maps. As shown in Figure 2, each feature set is passed through a block of 3×3 convolution layers to finally make per-pixel binary classification for presence of a person. These layers have ReLU non-linearity, except for the last, which has Sigmoid to predict pixel-wise confidence. Since the one-eighth scale feature set is computed with a larger receptive field, it could have global context information regarding crowd regions in the image. The one-fourth counterpart, though has a higher resolution, its predictions are based on limited global context and could result in false detection on crowd like patterns. Hence, we leverage the context information from one-eighth scale set through a top-down feedback connection. Basically, a transpose convolution layer is used to upsample the one-eighth feature maps followed by a normal 3×3 convolution to extract feedback feature maps. The feedback maps are then concatenated with the one-fourth scale column features. This helps the scale column block to receive high-level context information and achieve better prediction at higher resolution.

Apart from handling drastic variation in scale of appearance of people, such a multi-scale architecture is also motivated from the need to predict at the exact location in the output map. Note that there is inherent inconsistency in ground truth annotation of heads. The location of annotations vary widely in sparse crowds, where the point could be any where on the face or head. This issue is relatively less for dense regions owing to small heads, but requires prediction at smaller resolution for sparse crowds like the one-eighth. At this size, there is a high chance that the predicted and ground truth location closely match. But predicting at 1/8th resolution causes one pixel in the output to represent multiple people in a dense region. This calls for progressive prediction at increasing resolutions for better performance at all densities. However, we empirically find that two scales are sufficient to capture this variability for existing benchmark datasets. Now the challenge is to combine the multi-resolution predictions, which can have overlapping predictions with no scale information being available.

3.3. Multi-Scale Pretraining

The training of DD-CNN is done in two stages; the first is the *Multi-Scale Pretraining* and the second is *Adaptive Scale Training* (Section 3.5). Here we discuss the pretraining of the multi-scale network. The multi-scale module outputs per-pixel confidences at two different resolutions and we train each scale with per-pixel binary cross entropy loss. The loss is defined as,

$$\mathcal{L}(X, Y, \lambda) = \frac{1}{N} \sum_{x,y} \begin{matrix} \lambda Y'[x,y] \log X[x,y] + \\ (1 - Y'[x,y]) \log(1 - X[x,y]) \end{matrix} \quad (1)$$

where X is network prediction for a given input image and Y is the point ground truth map. $Y'[x, y] = \min(Y[x, y], 1)$ simply represents the binarized version of Y, where value 1 at pixel (x, y) indicates the presence of a person and 0 for background. Note that the summation runs over the spatial dimensions of the output, making the objective per-pixel. Since there are significantly less points with persons than without in training images, class imbalance might arise. So while training, we weigh the person class more by a factor λ (typically 2 or 4) and is observed to improve the performance.

Let $\mathcal{D}_{\frac{1}{4}}$ and $\mathcal{D}_{\frac{1}{8}}$ be respectively the one-fourth and one-eighth scale prediction maps. We train the individual scale columns with ground truth binary maps of same resolution. These maps are created from the head annotations available with crowd datasets. If $\mathcal{D}^{GT}_{\frac{1}{4}}$ and $\mathcal{D}^{GT}_{\frac{1}{8}}$ represents the ground truth maps, they are generated as,

$$\mathcal{D}^{GT}_{\frac{1}{s}}[x, y] = \sum_{x',y'} \mathbf{1}_{(x,y)=(\lfloor \frac{x'}{s} \rfloor, \lfloor \frac{y'}{s} \rfloor)} \quad (2)$$

where (x', y') are the annotated locations of people and s is either 4 or 8 for the two scales. Note that $\mathbf{1}$ is indicator function and $\lfloor . \rfloor$ denotes floor operation. The expression evaluates to the number of people being annotated at any location (x, y) in the downsampled resolution. For pretraining, we optimize parameters of one-eighth scale branch by minimizing the loss $\mathcal{L}(\mathcal{D}_{\frac{1}{8}}, \mathcal{D}^{GT}_{\frac{1}{8}}, \lambda_{\frac{1}{8}})$. Standard mini-batch gradient descent with momentum is employed (learning rate is fixed to 1e-3). Once the training is saturated, the weights updated for one-eighth branch are frozen and then remaining one-fourth network blocks are optimized. This is done by backpropagating one-fourth loss $\mathcal{L}(\mathcal{D}_{\frac{1}{4}}, \mathcal{D}^{GT}_{\frac{1}{4}}, \lambda_{\frac{1}{4}})$. Note that this scale is trained with the top-down feedback features and outputs dot map with a higher resolution. Thus, we have dot predictions at two different resolutions for the same crowd scene and can have inconsistent or inconclusive detections which needs to be faithfully combined.

3.4. Adaptive Scale Fusion and Dot Detection

A multi-scale architecture in the dot detection framework offers some unique challenges. The important one is the absence of scale information of the crowd. For a given person in a crowd image, there is no information regarding the size of the person in order to train with the correct scale. Hence we propose a novel *Adaptive Scale Fusion* (ASF) strategy, which does not require bounding box annotation, but delivers accurate dot prediction across drastic

scale and density variations. ASF essentially combines the predictions from multi-scale module and forms one output at the higher one-fourth resolution. For any given point in one-eighth prediction map ($\mathcal{D}_{\frac{1}{8}}$), corresponding region in the next higher resolution scale is taken (2×2 region in one-fourth scale $\mathcal{D}_{\frac{1}{4}}$) and the scale in which the maximum response occurs is the winning candidate. This is conceptually similar to scale pyramids, but adapted for resolving dot detections from multi-resolution predictions. To be more precise, let $p(x) = \lfloor \frac{x}{2} \rfloor$ evaluates to the coordinate in $\mathcal{D}_{\frac{1}{8}}$ for a pixel at location x of $\mathcal{D}_{\frac{1}{4}}$. Now for every pixel in ASF output $\mathcal{T}$, we compute an indicator variable $\mathcal{I}[x, y]$ to identify the scale and correct detections are filtered out. The ASF operation is expressed mathematically as,

$$\mathcal{I}[x,y] = \begin{cases} 1 & \text{if } \mathcal{D}_{\frac{1}{8}}[p(x),p(y)] \geq \max\limits_{\substack{(p(x'),p(y')) \\ =(p(x),p(y))}} \mathcal{D}_{\frac{1}{4}}[x',y'] \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

$$\mathcal{T}[x,y] = \begin{cases} \mathcal{D}_{\frac{1}{4}}[x,y] & \text{if } \mathcal{I}[x,y] = 0 \\ \mathcal{D}_{\frac{1}{8}}[p(x),p(y)] & \text{if } (\frac{x}{2},\frac{y}{2}) = (p(x),p(y)) \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where $\mathcal{T}$ has one-fourth resolution. Note that the max operation is applied over all (x', y') pairs that maps to the same coordinates in $\mathcal{D}_{\frac{1}{8}}$ as that of point (x, y).

In a nutshell, ASF merges the dot maps from multiple scales; a point in one scale is selected if it is maximum in its scale neighbourhood. This framework helps to select the scale which is giving higher prediction confidence. A threshold is applied on the output of ASF to generate the final highly localized binary dot map.

3.5. Adaptive Scale Training

After the *Multi-Scale Pretraining* of individual scale branches, we perform joint training to fine-tune the columns on two specialties. Ideally, we would like the $\mathcal{D}_{\frac{1}{8}}$ network to specialize on sparse crowds (or people appearing large) and $\mathcal{D}_{\frac{1}{4}}$ in dense crowds corresponding to their receptive fields. Such a division is enforced with the ASF architecture through a special training procedure. Note that straightforward training of ASF is not trivial due to absence of any scale information. For example, a person may have detections in all the scales. One cannot simply take the scale with maximum confidence, because $\mathcal{D}_{\frac{1}{8}}$ scale predictions are seen to dominate in confidence value as it aggregates more information regarding a point than scale $\mathcal{D}_{\frac{1}{4}}$. Hence, we device *Scale Adaptive Training* which fine-tunes the two scale columns such that each responds more to its own specialties and the ASF can then be done faithfully at test time.

To aid better training with ASF architecture, we leverage on the observation that some scale information can be obtained from ground truth point annotation. For example, at one-eighth resolution prediction, people in dense crowds would merge as one point (happens if there are multiple people in a region of 8×8). This provides a clear signal that these people could not be resolved at one-eighth scale and has to be in the other scale. So for the *Adaptive Scale* training, we incorporate this *Overlap Criteria (OLC)* on top of ASF to selectively fine-tune scale columns and achieve better specialization. For every point in $\mathcal{D}_{\frac{1}{8}}$ map, a check for overlap of ground truth points is performed. If there is an overlap in $\mathcal{D}_{\frac{1}{8}}$, it means that the point under consideration has to be trained in $\mathcal{D}_{\frac{1}{4}}$. This is done by setting the loss for the location to be zero in one-eighth $\mathcal{D}_{\frac{1}{8}}$ and allowing $\mathcal{D}_{\frac{1}{4}}$ network branch to be updated. Such an adaptive training causes the two scale networks to specialize on crowds of different types. However, OLC does not indicate anything about the scale of the majority non-overlapping points. For these points, the ASF module selects a scale, which is the scale corresponding to the point having the highest confidence. This acts like promoting the "winner" and updating the selected scale network. The exact loss formulation is:

$$\mathcal{M}_{\frac{1}{4}}[x,y] = \begin{cases} 1 & \text{if } \mathcal{D}^{GT}_{\frac{1}{8}}[p(x),p(y)] > 1 \\ 1 - \mathcal{I}[x,y] & \text{otherwise} \end{cases} \tag{5}$$

$$\mathcal{M}_{\frac{1}{8}}[x,y] = 1 - \mathcal{M}_{\frac{1}{4}}[p(x),p(y)] \tag{6}$$

$$\begin{aligned} \mathcal{L}_M \quad = \quad & \mathcal{L}(\mathcal{D}_{\frac{1}{4}} \odot \mathcal{M}_{\frac{1}{4}}, \mathcal{D}^{GT}_{\frac{1}{4}} \odot \mathcal{M}_{\frac{1}{4}}, \lambda_{\frac{1}{4}}) \\ & + \mathcal{L}(\mathcal{D}_{\frac{1}{8}} \odot \mathcal{M}_{\frac{1}{8}}, \mathcal{D}^{GT}_{\frac{1}{8}} \odot \mathcal{M}_{\frac{1}{8}}, \lambda_{\frac{1}{8}}), \end{aligned} \tag{7}$$

where $\mathcal{L}_M$ is joint loss for *Adaptive Scale* training and $\mathcal{M}_{\frac{1}{s}}$ represent the mask variables to indicate selected points for backpropagation. We train DD-CNN by minimizing $\mathcal{L}_M$ in same way as in pretraining. The branches of DD-CNN progressively get specialized possibly for different crowd densities. This results in the columns to respond more for its own specialties and facilitate ASF at test time. Note that OLC is crucial for the optimization as it acts as tie breaker.

At test time, only ASF is performed (as in Figure 2) and the points are selected based on the confidence. This adaptive architecture helps in predicting highly localized dots on people ranging in sparse to dense crowds. Finally, the threshold value (typically ∼0.5) for dot detection is selected so as to minimize the MAE (Sect 4.1) over a validation set.

4. Experiments

4.1. Evaluation Scheme and Metrics

Primarily two metrics are employed to evaluate any crowd counting system. The most important measure is the MAE or Mean Absolute Error and is defined as MAE $= \frac{1}{N}\sum_{n=1}^{N}|C_n - C_n^{GT}|$, where C_n is the count predicted for input n while its actual count is C_n^{GT}. This metric is a direct indicative of count accuracy of the model. To mea-

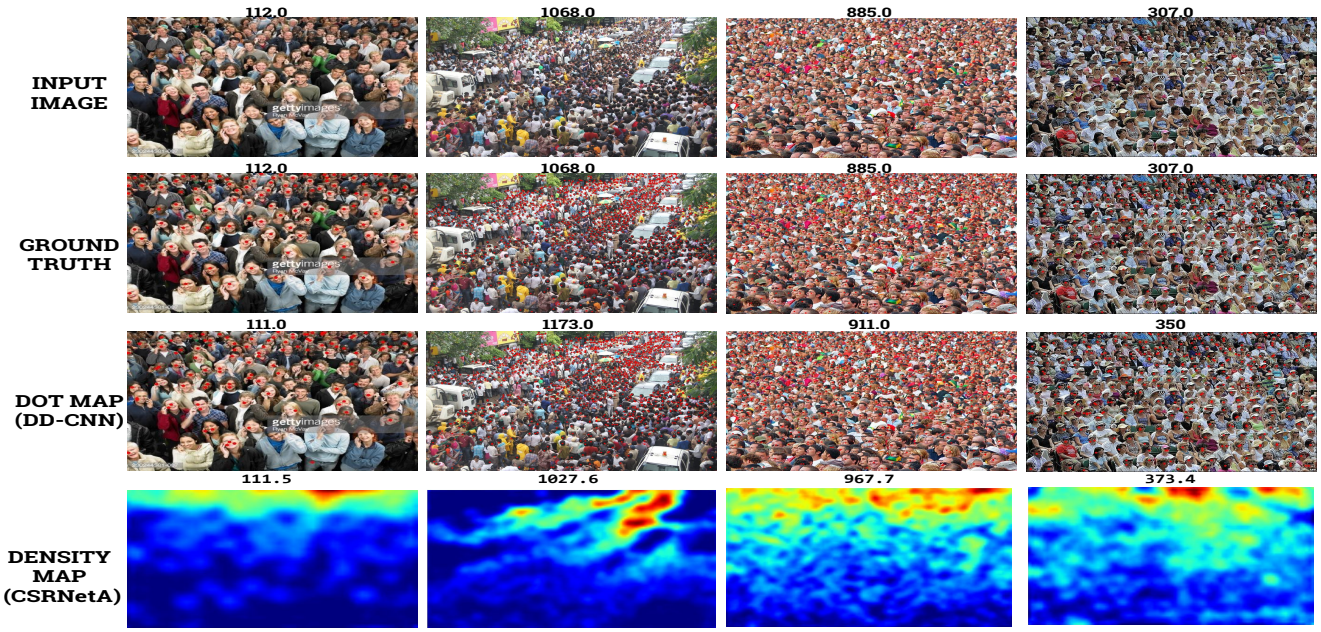

Figure 3. Predictions made by DD-CNN on images of Shanghaitech dataset [28]. The results emphasize the ability of our dot detection approach to localize people in crowds (zoom in the dot maps to see the difference).

sure the variance and hence robustness of the count estimate, Mean Squared Error or MSE is used. It is given by MSE $= \sqrt{\frac{1}{N}\sum_{n=1}^{N}(C_n - C_n^{GT})^2}$. However, there are some severe drawbacks with these metrics. The major limitation is that the metrics do not consider localization of the predictions. The MAE only measure the accuracy of overall count prediction and hence we evaluate our model on some localization metrics in Section 5.3. Note that except for UCF-QNRF dataset, for all other experiments the weighting hyper-parameters are set as $\lambda_{\frac{1}{4}} = 2$ and $\lambda_{\frac{1}{8}} = 1$.

4.2. UCF-QNRF Dataset

UCF-QNRF is introduced by [10] and by far the largest dense crowd counting dataset. There are 1201 images for training and 334 for the test set. The density of crowd varies between 49 to as high as 12,865. For this dataset, the class weighting factors are set as $\lambda_{\frac{1}{4}} = 4$ and $\lambda_{\frac{1}{8}} = 2$. Table 1 benchmarks DD-CNN with other regression models. DD-CNN obtains an MAE of 120.6, which is 12.6 lower than that of [10]. This shows that our approach is quite adaptable to highly diverse crowd scenario with relatively low MAE.

Method	MAE	MSE
Idrees et al. [9]	315	508
MCNN [28]	277	426
CMTL [19]	252	514
SCNN [4]	228	445
Idrees et al. [10]	132	191
DD-CNN (Ours)	**120.6**	**161.5**

Table 1. Performance of DD-CNN along with other methods on UCF-QNRF dataset [9]. Our model has better count estimation than all other methods.

4.3. UCF_CC_50 Dataset

UCF_CC_50 dataset [9] is a dataset of 50 images of highly diverse and dense crowds. The dataset poses a severe challenge to crowd counting models due to the small size and the drastic density variation, which ranges from 94 to 4543 people per image. A five fold cross-validation testing is performed on the dataset for evaluation. From Table 2, it is seen that DD-CNN delivers an impressive MAE of 215.4 and even beats the SA-Net [6] regression model by a margin of 43. Despite being a small dataset with drastic diversity, the state-of-the-art counting performance of our model, well evidence the effectiveness of dot detection.

4.4. Shanghaitech Dataset

The Shanghaitech dataset is introduced by [28] and consists of two sets, Part_A and Part_B. Part_A is quite diverse with large variations in crowd density ranging from 33 to 3139 people per image. But Part_B has relatively sparser crowds with maximum density of 578 and is less diverse. We train our DD-CNN on the dataset and Part_A results are reported in Table 2. Note that all other models in the table are based on density regression and is not exactly fair to compare DD-CNN with just MAE. DD-CNN achieves a

Figure 4. Dot predictions made by individual scale columns of DD-CNN on Shanghaitech dataset [28]. The outputs clearly shows that the multi-scale training improves significantly the dot prediction quality. (Zoom in to see the difference)

	UCF_CC_50		ST Part_A	
Model	MAE	MSE	MAE	MSE
Zhang et al. [27]	467.0	498.5	181.8	277.7
MCNN [28]	377.6	509.1	110.2	173.2
SCNN [4]	318.1	439.2	90.4	135.0
CP-CNN [20]	295.8	320.9	73.6	106.4
IG-CNN [2]	291.4	349.4	72.5	118.2
Liu et al. [14]	279.6	388.9	72.0	106.6
IC-CNN [17]	260.9	365.5	68.5	116.2
CSR-Net [11]	266.1	397.5	68.2	115.0
SA-Net [6]	258.4	334.9	**67.0**	**104.5**
DD-CNN	**215.4**	**295.6**	71.9	111.2

Table 2. Comparison of DD-CNN performance on UCF_CC_50 [9] and Shanghaitech Part_A dataset [28]. DD-CNN delivers very competitive count accuracy relative to other regression models.

detection MAE of 71.9 in Part_A, which is very close to the count error of best regression methods, with the difference being just 4.9. This again indicates that our approach has competitive performance along with all the merits of being a detection model. Figure 3 displays some dot predictions results of DD-CNN.

5. Analysis and Ablations

5.1. Effect of Multi-Scale Architecture

As described in Section 3.2, the proposed DD-CNN employs a multi-scale architecture with dot predictions at two different resolutions. This is motivated so as to address the drastic scale variation across sparse to dense crowds. We require localized dot prediction for both large faces/heads as well as for people in dense regions. A single network prediction would be biased to frequently appearing crowd type and would give lower confidence for large faces and fail to cross detection threshold. This problem of almost no response for people appearing large is severe with density regression (see Figure 3). To empirically establish the usefulness of the proposed DD-CNN architecture, we ablate our model in Table 3. We train a regression model, CSRNet-A [11] (CSR-A-reg) which is similar to the network used for one-eighth branch of DD-CNN. The count errors for individual scale columns are also listed in Table 3 and outputs are shown in Figure 4. As expected, the individual scale MAEs are higher than the combined multi-scale count error. We also see that MAE drops significantly without the *Overlap Criteria* (OLC) for training. Further, we run DD-CNN without the top-down feedback (TDF) connection. The performance with feedback is higher than without, indicating a possible propagation of high-level context to the next scale.

5.2. Dot vs Density Maps

We emphasize that the dot map framework is fundamentally different from density map in terms of the approach, philosophy and benefits. Here we show that the dot maps cannot be easily obtained by post-processing density maps. The CSR-A-reg model trained for regression in Section 5.1, is evaluated and density predictions are converted to dot

	ST Part_A		UCF-QNRF	
Method	MAE	MSE	MAE	MSE
CSR-A-reg	73.65	120.06	173.45	203.27
CSR-A-reg-dot	84.61	142.03	198.34	248.43
CSR-A-thr	309.8	513.4	384.42	566.98
CSR-A-thr-dot	167.09	218.22	164.38	204.97
DD-CNN $\mathcal{D}_{\frac{1}{8}}$ only	75.67	109.36	165.78	254.71
DD-CNN $\mathcal{D}_{\frac{1}{4}}$ only	125.19	190.77	234.7	511.9
DD-CNN (no TDF)	81.34	136.21	341.3	422.4
DD-CNN (no OLC)	104.95	151.26	346.81	406.59
DD-CNN	71.9	111.2	120.6	161.5

Table 3. Results for DD-CNN model ablative experiments. The results evidence the effectiveness of the design choices.

maps by thresholding. The threshold value is selected over a validation set to minimize the detection MAE. However, we find it difficult to threshold density maps without loss of counting performance. Lower the sigma (σ) of Gaussian used for density map generation, lower is the MAE drop. Though CSR-A-reg is trained with sigma as small as 1.0 (at prediction resolution), the MAE after thresholding is above 300, labeled as CSR-A-thr entry in Table 3. We even go to the extreme of dot map regression ($\sigma = 0$), for which the normal MAE is reasonable (CSR-A-reg-dot). But again, the thresholded MAE is very high (CSR-A-thr-dot). Some outputs of this model are shown in Figure 5, which clearly indicates hardly any detection in sparse regions and spurious or multiple predictions in remaining areas. Since no scale information regarding the detected person (like bounding boxes) is available, simple non-maximal suppression techniques do not work well across density ranges. Hence it is clear that these thresholding methods suffer from poor detections and results in much higher MAE than DD-CNN.

5.3. Localization of Detections

In this section, we analyse the localization of dot detection framework through some additional metrics. The MAE metric popularly being used in crowd counting, does not take into account prediction localization. It simply checks whether the overall crowd count of the scene matches with the ground truth. In other words, it is not necessary to detect people to get good MAE scores, but spurious responses could be counted as well. Hence, we propose a new metric named Mean Offset Error (MOE). MOE is defined as the distance in pixels between the predicted and ground truth dot averaged over test set. This is evaluated at the model prediction resolution and directly accounts for dot localization. A fixed penalty of 12 pixels is added for absent or spurious dot detections. Next, we follow [10] and consider a detection correct if the prediction is within a threshold distance. The threshold is varied to evaluate localization with average precision (L-AP), recall (L-AR) and Area under ROC (L-AuC). Furthermore, the Grid Average Mean absolute Error or GAME [7] metric, which is indicative of local count prediction accuracy, is also considered. GAME divides the prediction map into a grid of cells and the cell crowd counts are averaged.

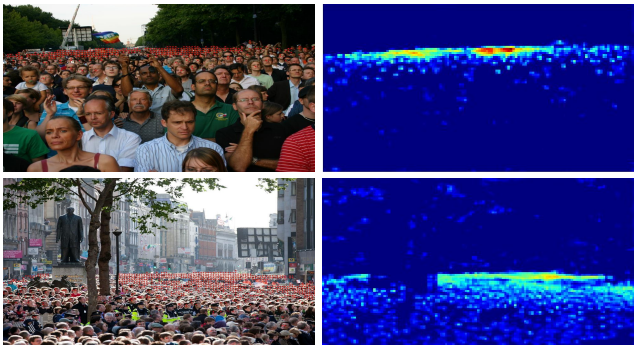

Figure 5. Detection by thresholding density maps of CSR-A-thr-dot net; results show almost no detections in sparse regions.

	ST Part_A		UCF-QNRF	
Metric	CSR-A	DD-CNN	CSR-A	DD-CNN
MOE ↓	5.13	4.93	4.16	2.91
L-AP ↑	0.61	0.65	0.72	0.82
L-AR ↑	0.76	0.81	0.77	0.83
L-AuC ↑	0.45	0.69	0.68	0.78
GAME(0) ↓	167.09	71.9	176.43	120.6
GAME(1) ↓	214.87	86.08	185.76	123.54
GAME(2) ↓	241.44	91.05	194.36	134.79
GAME(3) ↓	263.0	105.12	216.26	141.68

Table 4. Evaluation of DD-CNN and baseline regression on the localization metrics to analyse the dot prediction performance. Our model seems to achieve better localization of predictions.

Table 4 lists the performance of our model relative to the regression baseline on the metrics specified above. We use CSR-A-thr-dot model defined in Section 5.2 as baseline and compute localization metrics on detections from thresholded density maps. Clearly, DD-CNN outperforms the regression model in localization as evident from MOE and L-AUC scores. The same trend is observed in different levels of GAME metric as well. These experiments demonstrate that the proposed dot detection framework delivers superior localization, while still maintaining high count accuracy.

6. Conclusion

We propose a novel change to the framework of density regression employed for dense crowd counting. The density maps typically generated by existing regression models suffer from poor localization among other limitations. We address these issues by reformulating the counting task as a localized dot prediction problem. The proposed model, DD-CNN, is trained for per-pixel binary classification task of predicting a person. DD-CNN employs a multi-column multi-scale architecture to handle the drastic scale variations. Extensive evaluations indicate that the model achieves better or competitive performance compared to the state-of-the-art methods, despite providing the merits of a dot detection system. In the future, we hope that the community would move from the current regression paradigm to the dot detection framework and hence have more practical benefits of accurate localization.

Acknowledgment

This work was supported by SERB, Dept. of Science and Technology, Govt. of India (SB/S3/EECE/0127/2015).

References

[1] D. Babu Sam and R. V. Babu. Top-down feedback for crowd counting convolutional neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[2] D. Babu Sam, N. N. Sajjan, R. V. Babu, and M. Srinivasan. Divide and grow: Capturing huge diversity in crowd images with incrementally growing CNN. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[3] D. Babu Sam, N. N. Sajjan, H. Maurya, and R. V. Babu. Almost unsupervised learning for dense crowd counting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[4] D. Babu Sam, S. Surya, and R. V. Babu. Switching convolutional neural network for crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] L. Boominathan, S. S. Kruthiventi, and R. V. Babu. CrowdNet: A deep convolutional network for dense crowd counting. In *Proceedings of the ACM international conference on Multimedia (ACMMM)*, 2016.

[6] X. Cao, Z. Wang, Y. Zhao, and F. Su. Scale aggregation network for accurate and efficient crowd counting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[7] R. Guerrero-Gmez-Olmedo, B. Torre-Jimnez, R. Lpez-Sastre, S. M. Bascn, and D. Ooro-Rubio. Extremely overlapping vehicle counting. In *Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, 2015.

[8] P. Hu and D. Ramanan. Finding tiny faces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[9] H. Idrees, I. Saleemi, C. Seibert, and M. Shah. Multi-source multi-scale counting in extremely dense crowd images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[10] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, and M. Shah. Composition loss for counting, density map estimation and localization in dense crowds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[11] Y. Li, X. Zhang, and D. Chen. CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[12] J. Liu, C. Gao, D. Meng, and A. G. Hauptmann. DecideNet: Counting varying density crowds through attention guided detection and density estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016.

[14] X. Liu, J. van de Weijer, and A. D. Bagdanov. Leveraging unlabeled data for crowd counting by learning to rank. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[15] M. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis. SSH: Single stage headless face detector. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[16] D. Onoro-Rubio and R. J. López-Sastre. Towards perspective-free object counting with deep learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016.

[17] V. Ranjan, H. Le, and M. Hoai. Iterative crowd counting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[18] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[19] V. A. Sindagi and V. M. Patel. CNN-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017.

[20] V. A. Sindagi and V. M. Patel. Generating high-quality crowd density maps using contextual pyramid CNNs. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[21] R. Stewart and M. Andriluka. End-to-end people detection in crowded scenes. *arXiv preprint arXiv:1506.04878*, 2015.

[22] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision (IJCV)*, 2005.

[23] E. Walach and L. Wolf. Learning to count with CNN boosting. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016.

[24] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao. Deep people counting in extremely dense crowds. In *Proceedings of the ACM international conference on Multimedia (ACMMM)*, 2015.

[25] M. Wang and X. Wang. Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[26] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2005.

[27] C. Zhang, H. Li, X. Wang, and X. Yang. Cross-scene crowd counting via deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[28] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[29] C. Zhu, Y. Zheng, K. Luu, and M. Savvides. CMS-RCNN: Contextual multi-scale region-based CNN for unconstrained face detection. In *Deep Learning for Biometrics*. Springer, 2017.