OPEN ACCESS
International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 35, No. 16 (2021) 2160012 (31 pages)

DOI: 10.1142/S0218001421600120

World Scientific
www.worldscientific.com

Impact of Labeling Schemes on Dense Crowd Counting Using Convolutional Neural Networks with Multiscale Upsampling

Greg Olmschenk*

Department of Computer Science, The Graduate Center, CUNY
365 Fifth Avenue, New York, NY 10016, USA

NASA Goddard Space Flight Center
8800 Greenbelt Rd, Greenbelt, MD 20771, USA
golmschenk@gradcenter.cuny.edu

Xuan Wang*

Department of Computer Science, The Graduate Center, CUNY
365 Fifth Avenue, New York, NY 10016, USA
xwang4@gradcenter.cuny.edu

Hao Tang

Department of Computer Information Systems
Borough of Manhattan Community College, CUNY
199 Chambers Street, New York, NY 10007, USA
htang@bmcc.cuny.edu

Zhigang Zhu

Department of Computer Science
The City College of New York, CUNY
160 Convent Avenue, New York, NY 10031, USA

Department of Computer Science
The Graduate Center, CUNY
365 Fifth Avenue, New York, NY 10016, USA
zzhu@ccny.cuny.edu

Received 21 June 2020
Accepted 3 February 2021
Published 15 September 2021

Gatherings of thousands to millions of people frequently occur for an enormous variety of educational, social, sporting, and political events, and automated counting of these high-density

* Corresponding authors with equal contributions.

crowds is useful for safety, management, and measuring significance of an event. In this work, we show that the regularly accepted labeling scheme of crowd density maps for training deep neural networks may not be the most effective one. We propose an alternative inverse k-nearest neighbor (ikNN) map mechanism that, even when used directly in existing state-of-the-art network structures, shows superior performance. We also provide new network architecture mechanisms that we demonstrate in our own MUD-ikNN network architecture, which uses multi-scale drop-in replacement upsampling via transposed convolutions to take full advantage of the provided ikNN labeling. This upsampling combined with the ikNN maps further improves crowd counting accuracy. We further analyze several variations of the ikNN labeling mechanism, which apply transformations on the kNN measure before generating the map, in order to consider the impact of camera perspective views, image resolutions, and the changing rates of the mapping functions. To alleviate the effects of crowd density changes in each image, we also introduce an attenuation mechanism in the ikNN mapping. Experimentally, we show that inverse square root kNN map variation (iRkNN) provides the best performance. Discussions are provided on computational complexity, label resolutions, the gains in mapping and upsampling, and details of critical cases such as various crowd counts, uneven crowd densities, and crowd occlusions.

Keywords: Crowd counting; convolutional neural network; k-nearest neighbor; upsampling.

1. Introduction

Every year, gatherings of thousands to millions occur for protests, festivals, pilgrimages, marathons, concerts, and sports events. For any of these events, there are countless reasons to desire to know how many people are present. For those hosting the event, both real-time management and future event planning are dependent on how many people are present, where they are located, and when they are present. For security purposes, knowing how quickly evacuations can be executed and where crowding might pose a threat to individuals is dependent on the size of the crowd. For public health reasons, especially during and after the COVID-19 pandemic, measuring real-time crowd densities with surveillance cameras is valuable in enforcing social distancing while maintaining privacy. In journalism, crowd size information is frequently used to measure the significance of an event, and systems which can accurately report on the event size are important for a rigorous evaluation.

Many systems have been proposed for crowd counting purposes, with most recent state-of-the-art methods being based on convolutional neural networks (CNNs). To the best of our knowledge, every CNN-based dense crowd counting approach in recent years relies on using a density map of individuals, primarily with a Gaussian-based distribution of density values centered on individuals labeled in the ground truth images. Often, these density maps are generated with the Gaussian distribution kernel sizes being dependent on a k-Nearest Neighbor (kNN) distance to other individuals.[27] In this work, we explain how this generally accepted density map labeling is lacking and how an alternative inverse kNN (ikNN) labeling scheme, which does not explicitly represent crowd density, provides improved counting accuracy (Fig. 1). We will show how a single ikNN map provides information similar to the accumulation of many density maps with different Gaussian spreads, in a form which is better suited for neural network training. This labeling provides a significant gradient spatially across the entire label while still providing precise location

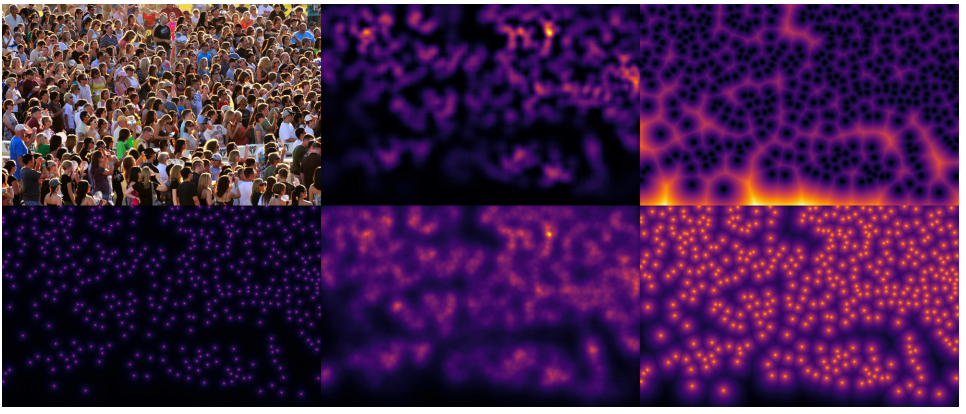


Fig. 1. An example of a crowd image and various kinds of labelings. From left to right, on top: the original image, the density map, the kNN map with $k = 1$. On bottom: the inverse kNN map with $k = 1$, $k = 3$, and $k = 1$ shown with a log scaling (for reader insight only). Note that in the case of the density map, any values a significant distance from a head labeling are very small. In contrast, the inverse kNN map has a significant gradient even a significant distance from a head position.

information of individual pedestrians (excluding exactly overlapping head labelings). We show that by simply replacing density map training in an existing state-of-the-art network with our ikNN map training, the testing accuracy of the network improves. This is the first major contribution of the paper.

Furthermore, coupling multi-scale drop-in replacement upsampling with densely connected convolutional networks[9] and our proposed ikNN mapping, we provide a new network structure, MUD-ikNN, which integrates Multi-scale Upsampling with transposed convolutions[24] in the DenseBlock structures from DenseNet201,[9] and utilizes our ikNN labeling scheme. The transposed convolutions are used to spatially upsample intermediate feature maps to the ground truth label map size for comparison. This approach provides several benefits. First, it allows the features of any layer to be used in the full map comparison, where many existing methods require a special, architecture-specific network branch for this comparison. Notably, this upsampling, comparison, and following regression module can be used at any point in any CNN, with the only change being the parameters of the transposed convolution. This makes the module useful not only in our specific network structure, but also applicable in future state-of-the-art, general-purpose CNNs. Second, as this allows features which have passed through different levels of convolutions to be compared to the ground truth label map, this intrinsically provides a multi-scale comparison without any dedicated additional network branches, thus preventing redundant parameters which occur in separate branches. Third, because the transposed convolution can provide any amount of upsampling (with the features being used to specify the upsampling transformation), the upsampled size can be the full ground truth label size. In contrast, most existing works used a severely reduced size label

map for comparison. These reduced sizes remove potentially useful training information. Although some recent works use full-size labels, they require specially crafted network architectures to accomplish this comparison. Our proposed upsampling structure can easily be added to most networks, including widely used general-purpose networks, such as DenseNet. This proposed multi-scale network structure is the second major contribution of the paper.

To validate the effectiveness and robustness of the ikNN mapping scheme, we additionally evaluate several variants of the ikNN mapping/labeling scheme. We analyze if additional considerations in designing mapping schemes impact the performance of crowd counting and examine the cause of the impact. Experiments are performed to test if these variant approaches improve the performance of crowd counting. For example, one such experiment tests how the variants affect performance detecting crowds with various per-person pixel resolutions. In this experiment, we divide the images of a dataset into the near half and far half with different crowd densities per pixel, in order to perform comparisons of the original ikNN mapping and its variants. Experimental results demonstrate that the inverse square root kNN mapping (iRkNN) achieves in the lowest errors, while the ikNN is close in performance. The mappings designed to account for considerations in both perspective views and resolutions performed worse than those which do not implement special mechanisms for these purposes. Details and discussion of the reasoning will be provided. The analysis of the various potential alternative mapping schemes is the third major contribution of the paper.

To alleviate the effects of crowd density changes in each image, we also introduce an attenuation mechanism to the ikNN mapping, which we propose to mitigate the estimation errors in far empty regions of the image. We performed experiments on original ikNN mapping, inverse square root kNN mapping and normalized ikNN mapping. We choose iRkNN mapping because our evaluation results show iRkNN has the lowest error. The original ikNN map has performance close to the iRkNN map. Since the normalized ikNN mapping can apply extra weight on pedestrians at farther distances in an effort to "correct" the perspective distortion, especially when further region is empty, this may result in overestimation. We apply our attenuation approach to adjust weights of these far empty regions, and further, we evaluate how our attenuation method can balance the extra weight that normalized the ikNN mapping applied in those regions.

The paper is organized as follows. Section 2 discusses related work. Section 3 proposes our overall new multi-scale network architecture for crowd counting, MUD-ikNN, to motivate the design of the ikNN mapping. Section 4 details the proposed inverse k-nearest neighbor map labeling method and its justification. Section 5 provides further proposed variations of the ikNN mapping schemes and the attenuation mechanism. Section 6 presents experimental results on several crowd datasets and analyzes the findings. Section 7 provides a few concluding remarks.

2. Related Work

Many works use explicit detection of individuals to count pedestrians.[23,16,22] However, as the number of people in a single image increases and a scene becomes crowded, these explicit detection methods become limited by occlusion effects. Early works to solve this problem relied on global regression of the crowd count using low-level features.[4,6,5] While many of these methods split the image into a grid to perform a global regression on each cell, they still largely ignored detailed spatial information of pedestrian locations. Ref. 14 introduced a methods of counting objects using density map regression, and this technique was shown to be particularly effective for crowd counting by Ref. 25. Since then, to the best of our knowledge, every CNN-based crowd counting method in recent years has used density maps as a primary part of their cost function.[11,18,21,25,27,19,15,17,20]

A primary advantage of the density maps is the ability to provide a useful gradient for network training over large portions of the image spatially, which helps the network identify which portion of the image contains information signifying an increase in the count. These density maps are usually modeled by representing each labeled head position with a Dirac delta function, and convolving this function with a 2D Gaussian kernel.[14] This forms a density map where the sum of the total map is equal to the total count of individuals, while the density of a single individual is spread out over several pixels of the map. The Gaussian convolution allows a smoother gradient for the loss function of the CNN to operate over, thereby allowing slightly misplaced densities to result in a lower loss than significantly misplaced densities.

In some works, the spread parameter of the Gaussian kernel is often determined using a k-nearest neighbor (kNN) distance to other head positions.[27] This provides a form of pseudo-perspective which results in pedestrians which are more distant from the camera (and therefore smaller in the image) having their density spread over a smaller number of density map pixels. While this mapping will often imperfectly map perspective (especially in sparsely crowded images), it works well in practice. Whether adaptively chosen or fixed, the Gaussian kernel size is dependent on arbitrarily chosen parameters, usually fine-tuned for a specific dataset.

We compare with adaptive version in this work, due to its success and being more closely related to our method. For example, in a recent work,[11] the authors used multiple scales of these kNN-based, Gaussian convolved density maps to provide levels of spatial information, from large Gaussian kernels (allowing for a widespread training gradient) to small Gaussian kernels (allowing for precise localization of density). While this approach effectively integrates information from multiple Gaussian scales, thus providing both widespread and precise training information, the network is left with redundant structures and how the various scales are chosen is fairly ad hoc. Our alternative ikNN labeling method supersedes, these multiple scale density maps by providing both a smooth training gradient and precise label locations (in the form of steep gradients) in a single label. Our new network structure

utilizes a single branch CNN structure for multi-scale regression. Together with the ikNN labeling, it provides the benefits of numerous scales of these density maps.

Though most CNN-based approaches use a reduced label size, some recent works[19,15,3,13] have begun using full resolution labels. In contrast even to these works, we provide a generalized map module which can be added to existing network structures. Specifically, the map module can be used as a drop-in replacement for the density map comparisons. This map module can be added to most dense crowd counting architectures with little or no modification to the original architecture. In this paper, our proposed network is based off the DenseNet201,[9] with our map module added to the end of each DenseBlock.

Our ikNN mapping is obliquely related to a distance transform, which has been used for counting in other applications.[1] However, the distance transform is analogous to a kNN map, rather than our ikNN. Notably, the ikNN crowd labeling presents the network with a variable training gradient to the network, with low values far from head labelings and cusps at a head labeling. In contrast, a kNN or distance transform provides constant training gradients everywhere. To our knowledge, neither the distance transform nor a method analogous to our ikNN labeling has been used for dense crowd counting.

To deal with uneven crowd densities in each image, some works[8,26,12] use attention mechanism to count pedestrians. A most recent work[12] uses attention scaling network to overcome the prediction bias of regions with different density levels. The authors proposed an approach to improve the performance by generating scaling factors and attention masks related to the far regions with different density levels, and generating attention-based density maps above the original density maps. Instead of training attention networks to generate new maps, we add an attenuation function to our ikNN mapping to reduce the map values to the far regions where there are little or no people to mitigate the estimation errors and improve the crowd estimation performance.

3. MUD-ikNN: A New Multi-Scale Network Architecture

We propose a new network structure, MUD-ikNN, with both multi-scale upsampling using DenseBlocks[9] and our ikNN mapping scheme. For providing a context of our proposed ikNN mapping scheme, we will describe the network structure first, before the detailed description of the ikNN mapping in Sec. 4. We show that the new MUD-ikNN structure performs favorably compared with existing state-of-the-art networks. In addition to the use of ikNN maps playing a central role, we also demonstrate how features with any spatial size can contribute to the prediction of ikNN maps and counts through the use of transposed convolutions. This allows features of various scales from throughout the network to be used for the prediction of the crowd. Throughout this section, a "label map" may refer to either our ikNN map or a standard density map, as either can be used with our network.

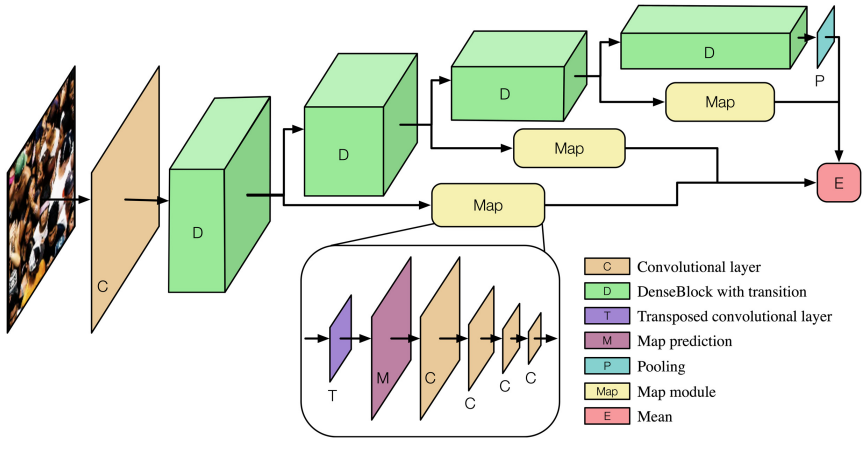


Fig. 2. A diagram of the proposed network architecture MUD-ikNN: multiscale regression with DenseBlocks and ikNN mapping. Best viewed in color.

3.1. *MUD-ikNN architecture*

The proposed MUD-ikNN network structure is shown in Fig. 2. Our network uses the DenseBlock structures from DenseNet201[9] in their entirety. DenseNet has been shown to be widely applicable to various problems. The output of each DenseBlock (plus transition layer) is used as the input to the following DenseBlock, just as it is in DenseNet201. However, each of these outputs is also passed to a map module (excluding the final DenseBlock output), which includes a transposed convolutional layer, a map prediction layer, and a small count regression module with four convolution layers. For each transposed convolution, the kernel size and stride are the same value, resulting in each spatial input element being transformed to multiple spatial output elements. The kernel size/stride value is chosen for each DenseBlock such that resulting map prediction is the size of the ground truth label. This form of upsampling using transposed convolutions allows the feature depth dimensions to contribute to the gradients of the map values in the predicted label map. Both the stride and kernel size of the transposed convolutions of our network are 8, 16, and 32 for the first three Denseblocks, respectively.

The label map generated at after each DenseBlock is individually compared against the ground truth label map, each producing a loss which is then summed,

$$\mathcal{L}_m = \sum_j \mathrm{MSE}(\hat{M}_j, M_j), \tag{1}$$

where j is the index of the DenseBlock that the output came from, M is the ground truth label map, and $\hat{M}$ is the predicted map labeling.

Each predicted label map is then also used as the input to a small count regression module. This module is a series of four convolutional layers, shown in the inset of

Table 1. A specification of the map module layers. This module is used at 3 points throughout our network as shown in Fig. 2, so the initial input size varies. However, the transposed convolution always produces a predicted map label which is uniform size (1×224×224).

Layer	Output size	Filter
Input from DenseBlock	$128 \times 28 \times 28$	
	$256 \times 14 \times 14$	
	$896 \times 7 \times 7$	
Transposed convolution	$1 \times 224 \times 224$ (map prediction)	$(8, 16, 32) \times (8, 16, 32)$stride $= (8, 16, 32)$
Convolution	$8 \times 112 \times 112$	2×2stride $= 2$
Convolution	$16 \times 56 \times 56$	2×2stride $= 2$
Convolution	$32 \times 28 \times 28$	2×2stride $= 2$
Convolution	$1 \times 1 \times 1$	28×28

Fig. 2. The sizes of these layers are specified in Table 1. The regression module then has a singleton output, corresponding to the predicted crowd count.

The mean of all predicted crowd counts from the regression modules, three in Fig. 2, and the output of the final DenseBlock is used as the final count prediction

$$\mathcal{L}_c = \mathrm{MSE}\left(\frac{\hat{C}_{\mathrm{end}} + \sum_{j=1}^{m} \hat{C}_j}{m+1}, C\right) \tag{2}$$

with C being the ground truth count, $\hat{C}_{end}$ being the regression count output by the final DenseBlock, and $\hat{C}_j$ being the count from the jth map regression module ($j = 1, 2, \ldots, m$; $m = 3$ in Fig. 2). This results in a total loss given by $\mathcal{L} = \mathcal{L}_m + \mathcal{L}_c$.

3.2. *MUD-ikNN benefits*

This approach has multiple benefits. First, if an appropriately sized stride and kernel size are specified, the transposed convolutional layer followed by label map prediction to regression module can accept any sized input. For example, an additional DenseBlock could be added to either end of the DenseNet, and another of these map modules could be attached. Second, each label map is individually trained to improve the prediction at that layer, which provides a form of intermediate supervision, easing the process of training earlier layers in the network. At the same time, the final count is based on the mean values of the regression modules. This means that if any individual regression module produces more accurate results, its results can individually be weighted as being more important to the final prediction.

We note that the multiple Gaussian approaches by Ref. 11 has some drawbacks. The spread of the Gaussians, as well as the number of different density maps, is arbitrarily chosen. Additionally, without upsampling, a separate network branch is required to maintain spatial resolution. This results in redundant network parameters and a final count predictor which is largely unconnected to the map prediction optimization goal. Our upsampling approach allows the main network to retain

a single primary branch and connects all the optimization goals tightly to this branch.

With both the Gaussian density maps and our ikNN maps, it is worth noting the importance of the label resolution. Taken to an extreme, one might reduce the label resolution to 1×1. Of course, this is equivalent to the global count label used by works prior to density map labels.[4,6,5] As the resolution increases, finer details of label training gradients emerge, allowing for the network to take advantage of the label training gradients. To take full advantage of the ikNN map label features (e.g. precise head position label cusps), a label resolution matching the original image resolution is ideal.

3.3. *MUD-ikNN implementation details*

The input to the network is 224×224 image patches. At evaluation time, a 224×224 sliding window with a step size of 128 was used for each part of the test images, with overlapping predictions averaged. The label maps use the same size patches, and predictions from the network are of the same resolution. Each count regression module contains the same four layers, as specified in Table 1.

For each experiment, the network was trained for 10^5 training steps. The network was designed and training process carried out using PyTorch (v0.4.0). The network was trained on an Nvidia GTX 1080 Ti.

Computational complexity in training and testing. As is typical for deep neural networks methods, training a neural network is time-consuming, while using the trained network for inference on new examples is computationally efficient. Using the above configuration, training the model to convergence takes several days. Using the trained model, the network can perform inference on a batch of image patches in 63 ms. Such inference speeds enable real-time inference of video data.

Complete details of the network code and hyperparameters can be found at https://github.com/golmschenk/sr-gan.

4. Inverse k-Nearest Neighbor Map Labeling

We propose using full image size ikNN maps as an alternative labeling scheme from the commonly used density map explained in Sec. 2. Formally, the commonly used density map[11,18,21,25,27] is provided by

$$D(\boldsymbol{x}, f(\cdot)) = \sum_{h=1}^{H} \frac{1}{\sqrt{2\pi} f(\sigma_h)} \exp\left(-\frac{(x - x_h)^2 + (y - y_h)^2}{2 f(\sigma_h)^2}\right), \tag{3}$$

where H is the total number of head positions for the example image, σ_h is a size determined for each head position (x_h, y_h) using the kNN distance to other heads positions (a fixed size is also often used), and f is a manually determined function for scaling σ_h to provide a Gaussian kernel size. We use this adaptive Gaussian

label as the baseline in our experiments. For simplicity, in our work, we define f as a simple scalar function given by $f(\sigma_h) = \beta\sigma_h$, with β being a hand-picked scalar. Though they both apply to head positions, the use of kNN for σ_h in the density map is not to be confused with the full kNN map used in our method, which is defined by

$$K(\boldsymbol{x}, k) = \frac{1}{k} \sum \min_k (\sqrt{(x - x_h)^2 + (y - y_h)^2}, \forall\, \boldsymbol{h} \in \mathcal{H}), \tag{4}$$

where $\mathcal{H}$ is the list of all head positions. In other words, the kNN distance from each pixel, $\boldsymbol{x} = (x, y)$, to each head position, $\boldsymbol{x}_h = (x_h, y_h)$, is calculated.

To produce the inverse kNN (ikNN) map, we use

$$M_{\mathrm{i}k\mathrm{NN}}(\boldsymbol{x}, k) = \frac{\alpha_m}{K(\boldsymbol{x}, k) + 1}, \tag{5}$$

where $M_{\mathrm{i}k\mathrm{NN}}$ is the resulting ikNN map, with the addition and inverse being applied element-wise for each pixel $\boldsymbol{x} = (x, y)$, K is defined in Eq. (4), and α_m is a constant scalar parameter to control the magnitude of the values in $M_{\mathrm{i}k\mathrm{NN}}$. Note that a term $+1$ is added in the denominator to prevent division by zero.

4.1. *Justifying ikNN map labeling*

To understand the advantage of an ikNN map over a density map, we can consider taking the generation of density maps to extremes with regard to the spread parameter of the Gaussian kernel provided by f. At one extreme, is a Gaussian kernel with zero spread. Here the delta function remains unchanged, which in practical terms translates to a density map where the density for each pedestrian is fully residing on a single pixel. When the difference between the true and predicted density maps is used to calculate a training loss, the network predicting density 1 pixel away from the correct labeling is considered just as incorrect as 10 pixels away from the correct labeling. This is not desired, as it both creates a discontinuous training gradient, and the training process is intolerant to minor spatial labeling deviations. The other extreme is a very large Gaussian spread. This results in inexact spatial information of the location of the density. At this extreme, this provides no benefit over a global regression, which is the primary purpose for using a density map in the first place. The extreme cases are shown for explanatory purposes, yet any intermediate Gaussian spread has some degree of both these issues. Using multiple scales of Gaussian spread, Ref. 11 tries to obtain the advantage of both sides. However, the size of the scales and the number of scales are then arbitrary and hard to determine. A similar explanation is illustrated in Fig. 3.

In contrast, a single ikNN map provides a substantial gradient everywhere while still providing steep gradients in the exact locations of individual pedestrians. Notably, near zero distance, the ikNN mapping clearly has a greater slope, and in comparison, for any Gaussian, there exists a distance at which all greater distances have a smaller slope than the equivalent position on the ikNN mapping. This means

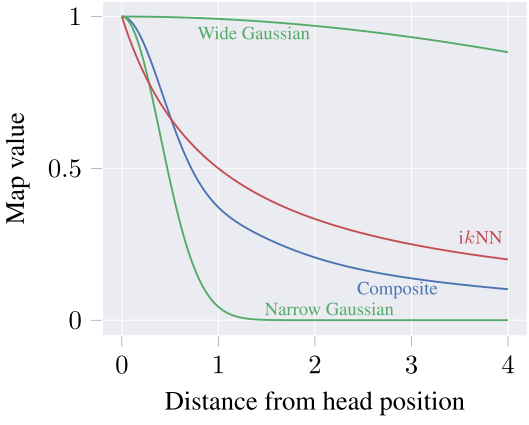


Fig. 3. (Color online) A comparison of the values of map labeling schemes with respect to the distance from an individual head position (normalized for comparison). Two Gaussians are shown in green. The blue line shows a composite of several Gaussians with spread parameters between those of the two extremes (The work[11] uses three Gaussian spreads in their work). This provides both precise and distant training losses. Our approach of the ikNN map shown in red (with $k = 1$) approaches a map function with a shape similar to the integral on the spread parameter of all Gaussians for a spread parameter range from 0 to some constant.

the slope of the Gaussian is only greater than the slope of the ikNN mapping for a middle range arbitrarily determined by the Gaussian spread. The ikNN curve and its derivative's magnitude (the inverse distance squared) monotonically increase toward zero. We want to note here that directly using a kNN map does not have the advantage of using an inverse kNN map, since a kNN or distance transform provides constant training gradients everywhere. This was further verified in our preliminary experiments. An example of our ikNN map compared with a corresponding density map labeling can be seen in Fig. 1. Reference[11] uses three density maps with different Gaussian spread parameters, with the Gaussian spread being determined by the kNN distance to other head positions multiplied by one of the three spread parameters. For a single head position, all Gaussian distributions integrated on β from 0 to an arbitrary constant result in a form of the incomplete gamma function. This function has a cusp around the center of the Gaussians. Similarly, the inverse of the kNN map also forms a cusp at the head position and results in similar gradients at corresponding distances as the integrated Gaussian function. In our experiments, we found that an inverse kNN map outperformed density maps with ideally selected spread parameters.

4.2. *Implementing ikNN map labeling*

In one experiment, we first use an existing network architecture[11] which uses DenseNet[9] as the backbone architecture, although we replace the density maps with ikNN maps and show there is an improvement in the prediction performance of the trained model. This demonstrates a direct improvement gained using our ikNN

method on an existing state-of-the-art network. Note that the regression module from ikNN map to count is then also required to convert from the ikNN map to a count. The difference in error between the original approach in Ref. 11 and the network in Ref. 11 with our ikNN maps, though improved, is relatively small. We suspect this is because the density maps (or ikNN maps) used during training are downsampled to a size of 28×28 (where the original images and corresponding labels are 224×224). This severe downsampling results in significant binning of pixel information, and this seems to reduce the importance of which system is used to generate the label. Taken to the extreme, when downsampled to a single value, both approaches would only give the global count in the patch (where the ikNN map gives the inverse of the average distance from a pixel to a head labeling which can be translated to an approximate count). This downsampling is a consequence of the network structure only permitting labels of the same spatial size as the output of the DenseBlocks of the DenseNet. Our MUD-ikNN network described in Sec. 3 remedies this through transposed convolutions, allowing for the use of the full-size labels.

Label generation computational complexity. The generation of the ikNN labels occurs as a one-time data preprocessing step before the training process, and thus, the label generation method does not have an impact on the speed of training steps, or the testing step. The same is true for all the variations of the mapping approaches as described below.

Discussions on occlusion. The primary goal of the network is a statistical prediction of the number of people in a crowd. As such, resolving individual occlusions is not a critical issue, so long as the network accurately predicts the overall crowd size. That said, inspecting how ikNN maps handle occlusions differently than Gaussian-based density maps may provide interesting insights differentiating the methods. In the Gaussian-based density map, two overlapping individuals will provide density totals summing to 2, taking into account the fact that two individuals are overlapping in the image. Notably, two perfectly overlapping individuals (center head locations being selected as the same pixel) will not be resolved by the ikNN map when $k = 1$. However, such overlaps leading to an overall count increase are still captured by the global count label used in ikNN approach. In such a case of perfect occlusion, we expect little or no visual information to suggest the presence of another person. In which case, the position of the additional person count is not useful to training the network. It can actually be detrimental, as without visual evidence, there is no way for the network to determine whether any particular person has a prefect occlusion, and it must guess between 1 and 2 individuals being at each. In this way, only including the additional count in the global count, and not in the map label (as the ikNN approach does), may be beneficial. However, perfect occlusions are a rare occurrence (indeed, perfectly overlapping heads likely cannot have been marked by the original human-created ground-truth). When occlusion is not a perfect overlap, the ikNN map has an advantage over the density maps. When the head positions are 2 pixels away, the ikNN forms two distinct peaks in the map label

(one for each head location). Typically, this is not the case for the Gaussian-based labeling, where the Gaussians form a mixture model, often containing only a single peak. As explained in Sec. 4.1, the value of the ikNN maps comes from providing improved training gradients. These distinct peaks in the map label demonstrate a case where the ikNN map provides useful training information which may be lost in the density map, as implied by our experimental results in Sec. 6.

5. Generalization of ikNN Mapping, Variations, and Attenuation

To motivate the generalization of the ikNN mapping, additional analysis was performed on the training samples of the UCF-QNRF dataset. This training dataset has 1201 images, with a average image resolution of 2013×2902. The dataset contains images which are viewed from a variety of perspectives, most of which results in a significant perspective distortion, which results in wider coverage of the physical area on the far end of an image than the near end. This results in denser crowd counts per pixel in more distance locations in the image (such as can be seen in Fig. 4(a)). To analyze whether additional mechanisms in the labeling scheme which account for these distortions improve crowd counting performance, we generate statistics of the full images as well as the far half ($y > 0$) and the near half ($y < 0$) of the images

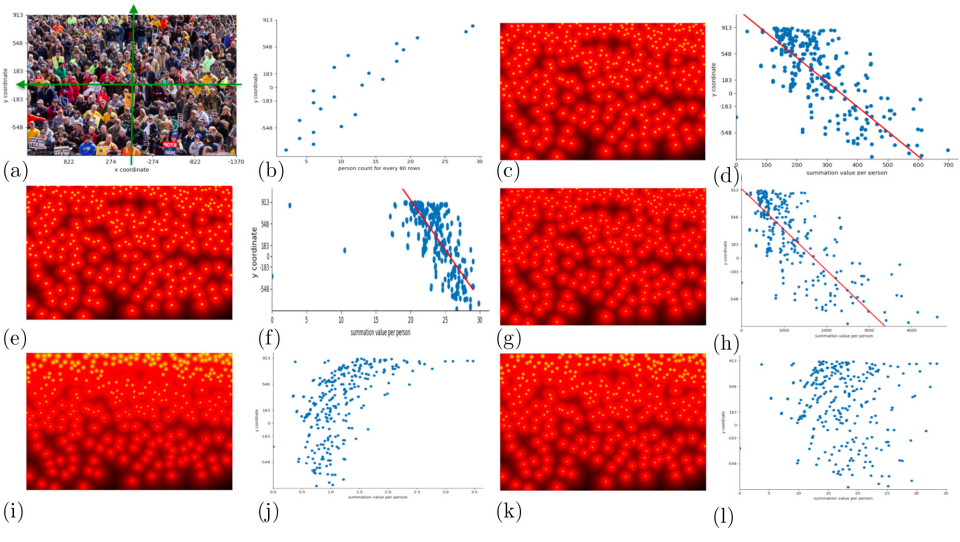


Fig. 4. (Color online) A crowd image (a) and its counts (b) and various maps (c–l). In (a), the origin of the image is at the center where the optical axis of the camera points to, the positive y goes up, and the positive x goes to the left, aligning with the camera coordinate system. In (b), the count numbers of every 80 rows are shown in the horizontal coordinates. Map values of four different maps (with a log scaling) and the plots of their summation-per-person versus the y-coordinate are shown from (c) to (l): (c) and (d) — i1NN; (e) and (f) — iS1NN; (g) and (h) — iR1NN; (i) and (j) — n-i1NN; (k) and (l) — w-i1NN. In each plot, the vertical axis is the y coordinate aligned with the image, and the horizontal coordinate value of each blue dot shows the summation of map values of a person at that y coordinate. The red line in each of (d), (f) and (h) is a linear fitting of the summation plot.

Table 2. Statistics of mean and median counts and summation-per-person values on various maps for the UCF-QNRF dataset.

	Count		i1NN sum		iS1NN sum		iR1NN sum		n-i1NN sum		w-i1NN sum	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median
Full image	838	421	147	105	18	17	668	360	1.71	1.36	12	11
Near half	337	136	300	180	21	20	1994	714	1.46	0.90	16	12
Far half	501	228	131	77	17	16	591	201	3.02	1.52	16	11
Near/far ratio	0.67	0.60	2.29	2.34	1.24	1.25	3.37	3.55	0.48	0.59	1.00	1.09

(with the origin of the coordinate system xoy defined in the center of the image where the optical axis of the camera points to; see Fig. 4(a)), and their ratios in Table 2. As many images do not have uniformly distributed crowd in the more distant part of the image, we show both the mean and median values. The mean (median) count of the ground truth person labeling shows a ratio of 0.67 (0.60) in the near half compared to the far half; the smaller ratio of the median indicates that images often do not have a significant crowd density in the far end of the images. Details of other columns will be explained in the following.

Examples of crowd images are shown in Fig. 5. To further analyze the impact of uneven crowding in the images, we first subcategorized the images into different categories based on the location of the people for both training samples and test samples of UCF-QNRF dataset. These categories consist of the cases when the far half is empty, the far half is not empty, and the far half has less than 5% of the head counts (Table 3). The proportion of each category for both the training set and the test set are roughly the same. Next, we analyze the subcategories based on the total count of the images. In Table 4, we show the that proportion of each category in the

Fig. 5. Examples of crowd images from UCF-QNRF dataset. On top: examples of dataset images which contain head positions in far half. On bottom: examples of dataset images in which the far half is empty.

Table 3. Number of images based on people location of UCF-QNRF dataset.

	No. of images	Far half empty	Far half not empty	Far half count less than 5%
Train	1201	85 (7%)	1116 (93%)	131 (11%)
Test	334	25 (7%)	309 (93%)	34 (10%)

Table 4. Statistics for images based on total count of UCF-QNRF dataset. We categorize the dataset into five subsets based on total count of the image: 0–250, 251–500, 501–750, 751–1000, 1001–2500, 2501–5000, 5001–10000 and $> 10\,000$.

	No. of images	0–250	251–500	501–750	751–1000	1001–2500	2501–5000	5001–10 000	10 000+
Train	1201	340 (28%)	359 (30%)	159 (13%)	88 (7%)	175 (16%)	61 (5%)	16 (1.3%)	4 (0.3%)
Test	334	88 (26%)	100 (30%)	43 (14%)	35 (10%)	54 (16%)	14 (4%)	0 (0%)	0 (0%)

test set is similar to the training set. However, there are no images which have total count over 5000 in test set, where there is a small portion of images (1.3%) that have a count of over 5000 in the training set.

To produce variants of the inverse kNN (ikNN) map, we define

$$M(\boldsymbol{x}, k) = \frac{\alpha_m}{R(\boldsymbol{x})(K(\boldsymbol{x}, k)^q + 1)}, \tag{6}$$

where M is the resulting generalized ikNN map, q is the distance normal order of the kNN map, R is the weighting factor considering perspective views (ranges) and/or image resolutions, and the operations are applied element-wise for each pixel $\boldsymbol{x} = (x, y)$.

5.1. *The original ikNN mapping and its variations*

For each of the below mappings, the statistics of the means and medians of the summation-per-person values of the full, near half, and far half images, as well their ratios, are shown in Table 2.

5.1.1. *The original ikNN mapping*

When $R(\boldsymbol{x}) = 1$ and $q = 1$ in Eq. (6), the mapping is the original ikNN in Eq. (5),

$$M_{ik\mathrm{NN}}(\boldsymbol{x}, k) = \frac{\alpha_m}{K(\boldsymbol{x}, k) + 1}, \tag{7}$$

To analyze the distributions of the map values, we approximate the summation of ikNN map values per labeled person, $S_K(\boldsymbol{x}_h, k)$, by summing up the ikNN map values with a circular region whose radius r is half of the kNN distance of that head position $\boldsymbol{x}_h = (x_h, y_h)$:

$$S_K(\boldsymbol{x}_h, k) = \sum_r M_{ik\mathrm{NN}}(\boldsymbol{x}_h \pm r, k). \tag{8}$$

Figure 4(c) shows the i1NN map (i.e. $k = 1$) with a log scaling for the image in Fig. 4 (a), and Fig. 4(d) shows the distribution of the summations per person along their y

coordinates. We can see that in the i1NN mapping, for people from far to near ranges, the summation values increase with the increase of image resolution due to the perspective view, with almost six times from the far end to the near end (roughly from 100 to 600 on the fitted red line). This implies that the ikNN mapping might favor the near parts of the images more, as more weight is assigned per person compared to the far regions of the image (Fig. 4(b), "counts per 80 rows" from 5 to 30 from near to far). The statistics shown in the "i1NN Sum" column of Table 2 shows that the summation values decrease from near to far, by about 2.3 times.

5.1.2. *Inverse squared kNN mapping*

To reduce the effect of unbalanced map values due to perspective views, a potential solutions is an inverse squared kNN mapping. This method turns distance-related measures into area-related measures in the mapping function, which increases the gradient of values in the resulting maps near head positions. When $R(\boldsymbol{x}) = 1$ and $q = 2$ in Eq. (6), the mapping is the inverse squared kNN (iSkNN):

$$M_{\text{iS}k\text{NN}}(\boldsymbol{x}, k) = \frac{\alpha_m}{K(\boldsymbol{x}, k)^2 + 1}. \tag{9}$$

Since the values of the iSkNN map decrease more quickly from a head position than in the ikNN map, and individuals closer to the camera have a relatively large pixel area before reaching the boundary of the kNN, this mapping will tend to provide a smaller map summation per individual for those in the nearer regions of the image. Figure 4(e) shows the iS1NN map (iSkNN when $k = 1$) with a log scaling of the image Fig. 4(a), and Fig. 4(f) shows its distribution of the summations per person. We can see that in the iS1NN mapping, for people from far to near ranges, summation values do not increase as dramatically as in the i1NN mapping. Instead, the rate is about 1.5 times from the far end to the near end (summation values are roughly from 20 to 30 on the fitted red line). The statistics shown in Table 2 shows that the summation values per person do not decrease as fast as in the i1NN mapping from near to far (about 1.2 times compared to 2.3 times). Since the far end has more counts than the near end of images, the squared ikNN map may incentivize the network to learn how to count more distant individuals which occupy a smaller pixel space.

5.1.3. *Inverse square root kNN mapping*

The next mapping to compare is an inverse square root kNN mapping (denoted as iRkNN), which turns distance-related measures into square root measures in the mapping function, thus making the changing rates of the mapping function slower — the opposite direction of the iSkNN. When $R(\boldsymbol{x}) = 1$ and $q = 1/2$ in Eq. (6), the mapping is the iRkNN:

$$M_{\text{iR}k\text{NN}}(\boldsymbol{x}, k) = \frac{\alpha_m}{\sqrt{K(\boldsymbol{x}, k)} + 1}. \tag{10}$$

Since the values of the iRkNN map as a function of pixel distance from a head position decrease at a slower rate than does the ikNN map, the mapping will tend to favor the near parts of an image than the far parts compared to the ikNN map. Figure 4(g) shows the iR1NN map (iRkNN when $k = 1$) with a log scaling of the image Fig. 4(a), and Fig. 4(h) shows its distribution of the summations per person. We can see that in the iR1NN mapping, for people from far to near portions of the image, summation values change more dramatically than the i1NN mapping, with the increase of image resolutions due to the perspective view, with a rate of about 10 times from the far end to the near end (summation values roughly from 300 to 3000 on the fitted red line). The statistics shown in Table 2 show that the summation values per person decrease faster than i1NN from near to far (about 3.4 times versus 2.3 times). Since the near end has much higher resolutions than the far end of images, the iRkNN map may improve the performance of counting at least to the near end.

5.1.4. *Normalized ikNN mapping*

The third method is a normalized ikNN mapping, which makes the sum of values for each person (almost) invariant to the distance of the person to the camera, thus reducing the effects of camera perspective changes. When $R(\boldsymbol{x})$ in Eq. (6) is a function of the ranges, given by fitting a line to the point set $(S_K(\boldsymbol{x_h}, k), y_h)$ (the red line in Fig. 4(d) of the corresponding ikNN map (Fig. 4(d)):

$$N_K(\boldsymbol{x}, k) = ay + b, \tag{11}$$

where a and b and the slope and the intercept of the fitted line, only subject to the change of y, and $q = 1$ in Eq. (8), the mapping is the normalized ikNN (denoted as n-ikNN):

$$M_{\text{n-i}k\text{NN}}(\boldsymbol{x}, k) = \frac{\alpha_m}{N_K(\boldsymbol{x})(K(\boldsymbol{x}, k) + 1)}. \tag{12}$$

Figure 4(i) shows the n-i1NN map ($k = 1$) with a log scaling of the image Fig. 4(a), and Fig. 4(j) shows its distribution of the summations per person. We can see that in the n-i1NN mapping, for people from far to near ranges, summation values are almost normalized to 1, which is about invariant to the image resolution due to the perspective view, from the far end to the near end. The statistics shown in the "n-i1NN Sum" column of Table 2 show that the mean/median summation values per person are scaled to a single digit (ideally it would be 1), which shall not change much from near to far. Since the far end has much more counts than the near end of images, the normalized ikNN map may improve the performance of counting at least to the far end. However, due to the approximation of the summation per person in Eq. (8), the actual "normalized" results favor more toward the far end, which may cause significant suppression of the map values in the near range with high image resolution, and enlargement of the map values of the far range with low image resolution and sparse crowd for some of the images and thus high noise level.

This might have negative affect, which will be validated this in our experiment section.

5.1.5. *Weighted ikNN mapping*

As mentioned above, the n-ikNN mapping may potentially apply too much weight on people in distance portions of the image, with lower image resolutions than people in near ranges, thus our final mapping is a weighted ikNN, which means to strike a balance between perspective views and image resolutions, and calibrating the bias of the approximation of summation-per-person calculation in Eq. (8). When $R(\boldsymbol{x})$ is a weighted function to consider both the perspective and resolution, and $q = 1$ in Eq. (6), the mapping is the weighted ikNN (denoted as w-ikNN):

$$M_{\text{w}-i k\text{NN}}(\boldsymbol{x}, k) = \frac{\alpha_m}{\sqrt{N_K(\boldsymbol{x})}(K(\boldsymbol{x}, k) + 1)}, \tag{13}$$

where we use the square root of the mean summation per person along the y coordinates as a weight function. Figure 4(k) shows the weighted i1NN map ($k = 1$) with a log scaling of the image Fig. 4(a), and Fig. 4(l) shows its distribution of the summations per person. We can see that in the weighted i1NN mapping, for people from far to near ranges, the summation values from the far end to the near end actually are equally distributed, around 20, which hopefully balance the consideration of different image resolutions and distance ranges. The statistics shown in the "n-i1NN Sum" column of Table 2 show that the summation values per person from near to far behave somewhere between the original i1NN and the normalized i1NN, which is actually a ratio of about 1:1.

5.2. *ikNN mapping with attenuation*

A recent work[12] proposed an approach to improve the performance by applying scaling factors and attention masks related to the far regions with different density levels, thus generating attention-based density maps. We add an attenuation function to our ikNN mapping, applied to the far regions which often contain little or no people. This may help mitigate the estimation errors and improve the performance. This is similar to the attention masks, but applied during the preprocessing step of the label maps rather than within the network itself. To generalize the inverse kNN (ikNN) map with attenuation, we define an attenuation function $A(\boldsymbol{x})$:

$$A(\boldsymbol{x}) = \begin{cases} \dfrac{y_{\min} - y}{y_{\max} - y_{\min}} \mathcal{D} + 1, & \boldsymbol{y} \in [y_{\min}, y_{\max}] \\ 1, & \boldsymbol{y} < y_{\min} \end{cases}, \tag{14}$$

where $\boldsymbol{x} = (x, y)$, $[y_{\min}, y_{\max}]$ represents range of the far empty region in the y direction where has no people in the image, and $\mathcal{D}$ is a scale factor. 1 is added in the

denominator to prevent division by zero in the following equation:

$$M_{\text{attenuation}}(\boldsymbol{x}, k) = \frac{\alpha_m}{A(\boldsymbol{x})R(\boldsymbol{x})(K(\boldsymbol{x}, k)^q + 1)}. \tag{15}$$

Here $M_{\text{attenuation}}$ is the resulting generalized ikNN map with attenuation.

5.2.1. *Original ikNN mapping with attenuation*

When $R(\boldsymbol{x}) = 1$ and $q = 1$ in Eq. (15), the mapping is the original ikNN with attenuation:

$$M_{\text{attenuation}-i\text{kNN}}(\boldsymbol{x}, k) = \frac{\alpha_m}{A(\boldsymbol{x})(K(\boldsymbol{x}, k) + 1)}. \tag{16}$$

As discussed in Sec. 5.1.1, the original ikNN mapping may favor the nearer parts of the images where there is greater pixel resolution per person. In further consideration of the uneven crowd distribution, we experimented with decreasing the weights of the farther parts by applying an attenuation function to these far empty regions.

5.2.2. *Inverse square root kNN mapping with attenuation*

An inverse square root kNN mapping (denoted as iRkNN) reduces the relative impact of distances within the mapping label. When $R(\boldsymbol{x}) = 1$ and $q = 1/2$ in Eq. (15), the mapping is defined by

$$M_{\text{attenuation}-\text{iR}k\text{NN}}(\boldsymbol{x}, k) = \frac{\alpha_m}{A(\boldsymbol{x})(\sqrt{K(\boldsymbol{x}, k)} + 1)}. \tag{17}$$

In our experiments, iRkNN provided the best performance of the nonattenuation variations. As such, we compare iRkNN with attenuation to base ikNN with attenuation (Sec. 6.5).

5.2.3. *Normalized ikNN mapping with attenuation*

With $R(\boldsymbol{x})$ described for normalization as was presented in Sec. 5.1.4 and $q = 1$ for Eq. (15), the mapping is the normalized ikNN with attenuation (attenuation-n-i1NN) and is given by

$$M_{\text{attenuation}-\text{n}-i\text{kNN}}(\boldsymbol{x}, k) = \frac{\alpha_m}{A(\boldsymbol{x})N_K(\boldsymbol{x})(K(\boldsymbol{x}, k) + 1)}. \tag{18}$$

Although the unattenuated n-i1NN reduces the effects of camera perspective changes, the normalization may over emphasize map values areas of the image where no individuals exist. By adding the attenuation function to the n-i1NN mapping, we may reduce the weights in these farther parts of the image, with the potential for improved performance.

Figure 6 shows the attenuation mapping results for three i1NN mapping variations. We can see our attenuation function reduced map values for the farther, empty

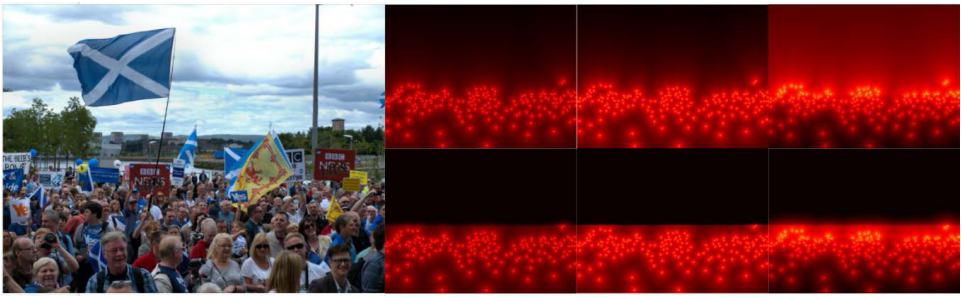


Fig. 6. An example of a crowd image and three mapping variations without and with attenuation (all shown in a log scale). From left to right, i1NN mapping, iR1NN mapping and n-i1NN mapping (top); i1NN mapping, iR1NN mapping and n-i1NN mapping after applying attenuation (bottom). The scale factor $\mathcal{D}$ is set as 100 in our experiments.

regions of the image, with the potential to lower the estimation errors. These performance improvements are evaluated in Sec. 6.5.

6. Experimental Results

6.1. *Evaluation metrics*

For each dataset that we evaluated our method on, we provide the mean absolute error (MAE), normalized absolute error (NAE), and root mean squared error (RMSE). These are given by the following equations:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{C}_i - C_i|, \tag{19}$$

$$\text{NAE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{C}_i - C_i|}{C_i}, \tag{20}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{C}_i - C_i)^2}. \tag{21}$$

In the first set of experiments, we demonstrate the improvement of the ikNN labeling scheme compared to the density labeling scheme. We trained our network using various density maps produced with different Gaussian spread parameters, β (as described in Sec. 4) and compared these results to the network using ikNN maps with varying k. We also analyze the advantage of upsampling the label for both density and ikNN maps. In the second set of experiments, we provide comparisons to the state-of-the-art on standard crowd counting datasets. In these comparisons, the best ikNN map and density map from the first set of experiments is used. Most works provide their MAE and RMSE results. Reference[11] provided the additional metric of NAE. Though this result is not available for many of the datasets, we provide our own NAE on these datasets for future works

Table 5. General statistics for the tested datasets.

Dataset	No. of images	Total count	Mean count	Max count	Average resolution
UCF-QNRF	1535	1 251 642	815	12 865	2013 × 2902
ShanghaiTech Part A	482	241 677	501	3139	589 × 868
ShanghaiTech Part B	716	88 488	123.6	578	768 × 1024
UCF-CC-50	50	63 974	1279	4633	2101 × 2888

to refer to. The most directly relevant work[11] has only provided their results for their latest dataset, UCF-QNRF. As such, their results only appear with regard to that dataset. Finally, we design an experiment to test if the more sophisticated variations of ikNN mapping would improve the performance of crowd counting.

General statistics about the datasets used in our experiments is shown in Table 5. Results show that (1) overall ikNN mapping has better performance than the density mapping; (2) i1NN mapping has the best performance among all; and (3) ikNN mapping is better than the density mapping when k is not larger than 3.

6.2. *Impact of labeling approach and upsampling*

6.2.1. *Density maps versus ikNN maps*

We used the ShanghaiTech dataset[27] part A for this analysis. The results of these tests are shown in Table 6. The density maps provide a curve, where too large and too small of spreads perform worse than an intermediate value. Even when choosing

Table 6. Results using density maps versus ikNN maps with varying k and β, as well as the various upsampling resolutions on the ShanghaiTech Part A dataset. If a resolution is not shown, it is the default 224 × 224. Multiple β correspond to a different Gaussian density map for each of the three map module comparisons.

Method	MAE	NAE	RMSE
MUD-densityβ0.3 28×28	79.0	0.209	120.5
MUD-densityβ0.3 56×56	74.8	0.181	121.0
MUD-densityβ0.3 112×112	73.3	0.176	119.1
MUD-i1NN 28×28	75.8	0.180	120.3
MUD-i1NN 56×56	72.7	0.181	117.4
MUD-i1NN 112×112	70.8	0.166	117.0
MUD-densityβ0.05	84.5	0.233	139.9
MUD-densityβ0.1	76.8	0.189	120.3
MUD-densityβ0.2	75.3	0.175	124.2
MUD-densityβ0.3	72.7	0.174	120.4
MUD-densityβ0.4	75.7	0.176	130.5
MUD-densityβ0.5	76.3	0.182	130.0
MUD-density$\beta_1$0.5, $\beta_2$0.3, $\beta_3$0	78.5	0.205	124.2
MUD-density$\beta_1$0.5, $\beta_2$0.3, $\beta_3$0.05	77.8	0.207	124.9
MUD-density$\beta_1$0.4, $\beta_2$0.2, $\beta_3$0.1	76.7	0.202	122.7
MUD-density$\beta_1$0.1, $\beta_2$0.2, $\beta_3$0.4	75.1	0.191	119.0

Table 6. (*Continued*)

Method	MAE	NAE	RMSE
MUD-density$\beta_1 0.2$, $\beta_2 0.3$, $\beta_3 0.4$	76.0	0.196	122.1
MUD-i1NN	68.0	0.162	117.7
MUD-i2NN	68.8	0.168	109.0
MUD-i3NN	69.8	0.169	110.7
MUD-i4NN	72.2	0.173	116.0
MUD-i5NN	74.0	0.182	119.1
MUD-i6NN	76.2	0.188	120.9

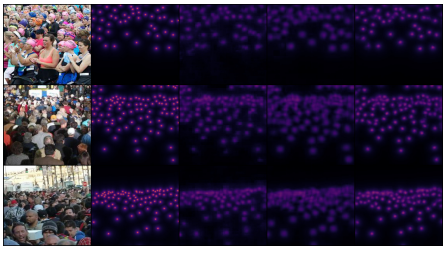
the best value (where $\beta = 0.3$), which needs to manually determined, the i1NN label significantly outperforms the density label.

Included in the table are experiments, in the fashion of Ref. 11, with density maps using three different β values. Here β_1 denotes the spread parameter used as the label map for the first map module, while β_2 and β_3 are for the second and third modules. Contrary to findings of Ref. 11, we only gained a benefit from three density labels when the first output had the smallest spread parameter. Even then, the gain was minimal. Upon inspection of the weights produced by the network from the map to the count prediction, the network reduces the predictions from the nonoptimal β maps to near zero and relies solely on the optimal map (resulting in a reduced accuracy compared to using the optimal map for each map module).

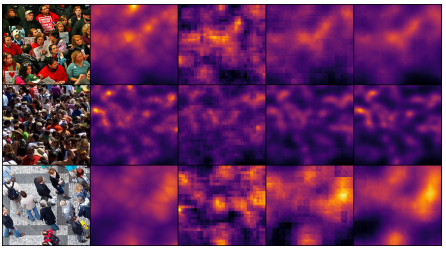
With varying k, we find that an increased k results in lower accuracy. This is likely due to the loss of precision in the location of an individual. The most direct explanation for this can be seen in the case of $k = 2$. Every pixel on the line between two nearest head positions will have the same map value, thus losing the precision of an individual location.

6.2.2. *Upsampling analysis*

Most existing works use a density map with a reduced size label for testing and training. Those that use the full label resolution design specific network architectures for the high-resolution labels. Our map module avoids this constraint by upsampling the label using a trained transposed convolution, which can be integrated into most existing architectures. Using the ShanghaiTech part A dataset, we tested our network using various label resolutions to determine the impact on the predictive abilities of the network. These results can be seen in Table 6. Experiments without no label resolution given are 224×224. In the top section of this table, we see the performance of the network when using output labels with sides of size 28, 56, 112. The corresponding comparison with size of 224 is seen in the third section of the table. In each case, a higher resolution results in a higher accuracy. Note that this results in a minor change to the map module structure, as the final convolution kernel needs to match the remaining spatial dimension. A set of predicted ikNN map labels can be seen in Fig. 7, where a grid pattern due to the upsampling can be identified in some cases.



(a) Input patches with corresponding i1NN labels and predictions.



(b) Input patches with corresponding i3NN labels and predictions.

Fig. 7. A set of randomly selected input image patches, along with their corresponding ground truth map labels and map predictions. Figure 7(a) shows the i1NN case and Fig. 7(b) shows the i3NN case. In each subfigure, there are three rows, each corresponding to a randomly selected input image. In each subfigure, the five columns from left to right are the original image patch, the ground truth label, and the patches from the three map modules in order through the network.

6.3. *Comparisons on standard datasets*

The following demonstrates our network's predictive capabilities on various datasets, compared to various state-of-the-art methods. Again, we note that our improvements are expected to complementary to the existing approaches, rather than alternatives.

For these experiments, we used the best k, 1, and best β, 0.3, from the first set of experiments.

The first dataset we evaluated our approach on is the UCF-QNRF dataset.[11] The results of our MUD-ikNN network compared with other state-of-the-art networks are shown in Table 7. Our network significantly outperforms the existing methods. Along with a comparison of our complete method compared with the state-of-the-art, we compare with the network of Ref. 11, but replace their density map predictions and summing to count with our ikNN map prediction and regression to count. Using the ikNN maps, we see that their model sees improvement in MAE with ikNN maps, showing the effect of the ikNN mapping. This experiment shows that the improvement of the performance of the ikNN mapping without the use of the MUD model, a

Table 7. Results on the UCF-QNRF dataset.

Method	MAE	NAE	RMSE
Idrees *et al.*[10]	315	0.63	508
MCNN[27]	277	0.55	426
Encoder–Decoder[2]	270	0.56	478
CMTL[21]	252	0.54	514
SwitchCNN[18]	228	0.44	445
Resnet101[7]	190	0.50	227
DenseNet201[9]	163	0.40	226
Idrees *et al.*[11]	132	0.26	191
Idrees *et al.*[11] with i1NN maps	122	0.252	195
MUD-i1NN	104	0.209	172

Table 8. Results on the ShanghaiTech Part A, ShanghaiTech Part B and UCF-CC-50 datasets.

	ShanghaiTech Part A			ShanghaiTech Part B			UCF-CC-50		
Method	MAE	NAE	RMSE	MAE	NAE	RMSE	MAE	NAE	RMSE
ACSCP	75.7	—	102.7	18.7	—	26.0	291.0	—	404.6
D-ConvNet-v1	73.5	—	112.3	17.2	—	27.4	288.4	—	404.7
ic-CNN	68.5	—	116.2	10.7	—	16.0	266.1	—	397.5
CSRNet	68.2	—	115.0	10.6	—	16.0	260.9	—	365.5
MUD-densityβ0.3	72.7	0.174	120.4	16.6	0.130	26.9	246.44	0.188	348.1
MUD-i1NN	68.0	0.162	117.7	13.4	0.107	21.4	237.76	0.191	305.7

reduction of MAE error of 10 counts; with the MUD model (including upsampling), the performance is further improved, a further reduction of the MAE error of 18 counts.

The second dataset we evaluated our approach on is the ShanghaiTech dataset.[27] The dataset is split into two parts, Part A and Part B. For both parts, we used the training and testing images as prescribed by the dataset provider. The results of our evaluation on part A are shown in Table 8. Our MUD-ikNN network slightly outperforms the state-of-the-art approaches on this part. The results of our evaluation on part B are also shown in Table 8. Here our network performs on par or slightly worse than the best-performing methods. Notably, our method appears perform better on denser crowd images, and ShanghaiTech Part B is by far the least dense dataset we tested.

The third dataset we evaluated our approach on is the UCF-CC-50 dataset.[10] We followed the standard evaluation metric for this dataset of a five-fold cross-evaluation. The results of our evaluation of this dataset can be seen in the last portion of Table 8.

Overall, our network performed favorably compared with existing approaches. An advantage to our approach is that our modifications can be applied to the architectures we're comparing against. The most relevant comparison is between the ikNN version of the MUD network, and the density map version of the same MUD network. Here, the ikNN approach always outperformed the density version. We speculate that the state-of-the-art methods we have compared with, along with other general-purpose CNNs, could be improved through the use of ikNN labels and upsampling map modules. Note that the overall mean absolute errors (MAE) across all the five datasets are still relatively high compared to the latest state-of-the-art network architectures Table 9. Applying ikNN labels to more powerful crowd counting architectures would be an interesting future direction.

6.4. *Evaluating ikNN mapping variations*

Experiments were also performed on our additional mapping mechanisms that explicitly account for image perspective and mapping change rates to analyze their impact on crowd counting performance. To test how they perform on various image

Table 9. Summary of statistics (crowd data and MAE results) for the tested datasets.

Dataset	No. of images	Total count	Mean count	Max count	Resolution	MAE	Relative MAE
UCF-QNRF	1535	1 251 642	815	12 865	2013×2902	104	12.8%
ShanghaiTech Part A	482	241 677	501	3139	589×868	68	13.6%
ShanghaiTech Part B	716	88 488	123.6	578	768×1024	13.4	10.9%
UCF-CC-50	50	63 974	1279	4633	2101×2888	238	18.6%

Table 10. Performance statistics of counts errors (MAEs) using various maps (ikNN, iSkNN, iRkNN, n-ikNN and w-ikNN, where $k = 1$) for the UCF-QNRF dataset. The first column of values are the three mean counts of the ground truth labels. Inside the parentheses after each MAE, the relative error over its corresponding mean count is listed.

	Mean count	i1NN (1e-3)	iS1NN(1e-3)	iR1NN(1e-3)	n-i1NN(1e-3)	w-i1NN(1e-3)	w-i1NN(1e-2)
Full Image	838	104.9 (12.5%)	107.0 (12.8%)	100.6 (12.0%)	125.7 (15.0%)	120.3 (14.3%)	122.7 (14.6%)
Near Half	337	44.4 (13.2%)	43.6 (12.9%)	47.0 (13.9%)	62.1 (18.4%)	58.0 (17.2%)	58.1 (17.2%)
Far Half	501	77.3 (15.5%)	78.8 (15.6%)	73.3 (14.6%)	89.8 (17.9%)	85.8 (17.1%)	85.5 (17.1%)

resolutions, we divide the images of the UCF-QNRF dataset[11] into the near halves and far halves, which typically have different crowd densities per pixel, in order to compare the performance of the original ikNN mapping and its four variations. Since it was also observed that the sums of the various mapping mechanisms are different (as shown in Table 2), up to several orders of magnitude, we have also have used various map multipliers (i.e. the constant scalar parameter α_m) so that the averages of the map values are normalized to the same order of magnitude.

Table 10 shows the results of these comparisons. In the following experiments, by default, $\alpha_m = 10^{-3}$. $\alpha_m = 10^{-2}$ was used for w-i1NN, to bring its map values to the same order of magnitude as the baseline i1NN.

From these experiments, we note the following observations:

- The iRkNN mapping ($k = 1$) results in the best performance of all the approaches. Note that this approach increases the map values of near end more than the far end of the image by flattening the map value distribution curve for each person. However, the results show that it not only increases the accuracy of the overall count, but also the accuracy of the far half. This likely because it reduces the noise of the far end of the image where they is either very sparse crowding or very dense crowding (resulting in more noise). This may be reduced by using a flattened curve so it focuses on the overall counts rather than individuals.
- The iSkNN mapping ($k = 1$) improves the accuracy of the higher resolution end of the images (near half) slightly. Note that the iSkNN mapping of each person has more significant gradients near head locations. Due to the high-resolution nature of the near half, the rapidly decreasing may increase the accuracy of the models with respect to head locations.

- Consistently, the accuracy of counts for the near halves are better than the far half of images, across all the three (ikNN, iRkNN, iSkNN) mapping approaches without normalization or weighting. This indicates image resolution significantly impacts the performance. Compared to these approaches, the weighted and normalized approaches appear to balance the performance of the near and far halves. This is most apparent for the normalized version.
- The attempts to improve the performance of the far end by increasing the map values as functions of ranges did not produce a benefit. Instead, a decrease in performance in the near halves of the images was observed, likely due to suppressing the map values. A decrease in performance in the far halves was also observed, likely due to increases in the noise level of in the low resolution. It is possible these losses in performance would not occur if we used the true camera geometry as opposed to estimates. However, this camera geometry is unavailable.
- Although the performance appears to increase as the orders of magnitudes of various maps increase, the multipliers actually do not significantly impact the overall performance. This is observed in the similar results of different multipliers for the weighted mapping approach.

6.5. *Evaluating ikNN mapping with attenuation*

We further performed experiments on the effectiveness of our attenuation approach using the UCF-QNRF dataset.[11] We group the images of the UCF-QNRF dataset based on the uneven crowd densities in the far half (categorized into cases where the far half is empty, the far half is not empty, and the far half has less than 5% of total count of an image). We also group the images based on the total crowd counts in five categories (0–250, 251–500, 501–750, 751–1000) for images which have total counts under 1000, and 2 categories (1001–2500, 2501–5000) for images which have total counts over 1000. In order to compare the performance of our attenuation mechanism, we chose the original ikNN mapping and two other variations: the iRkNN mapping and the n-ikNN mapping. The iRkNN mapping demonstrated the best performance prior to attenuation. The n-ikNN mapping seems to emphasize the farther regions of the images, and thus has the potential for significant improvements via the attenuation. We perform experiments to evaluate the effectiveness of our attenuation method on these three mappings. Table 11 shows the results.

For each case, $\alpha_m = 10^{-3}$. From these experiments, we draw the following observations:

- Of the three attenuated mappings, the attenuation-i1NN mapping results in the best performance. With attenuation, ikNN further decreases the weight of far, empty regions of the image, and the overall count accuracy has slight improvements. iR1NN performs significantly worse with attenuation. As the iR1NN variant already reduces the value of sparsely crowded areas, we speculate that the additional attenuation results in under-emphasized sparse areas. While

Table 11. Performance comparison of counts errors (MAEs) using various maps (i1NN, iR1NN and n-i1NN) and attenuation maps (attenuation-i1NN, attenuation-iR1NN, and attenuation-n-i1NN) for the UCF-QNRF dataset. The first column of values are the mean counts of the ground truth labels for each category. We use the default multiplier which is $\alpha_m = 10^{-3}$.

	Mean count	i1NN	Attenuation-i1NN	iR1NN	Attenuation-iR1NN	n-i1NN	Attenuation-n-i1NN
Overall	838	104.9	**103.8**	100.6	110.9	125.7	**116.6**
Far half empty	626	109.2	**103.4**	104.3	134.2	134.9	**93.1**
Far half not empty	726	104.3	**103.8**	100.3	109	124.9	**118.5**
Far half less than 5%	413	106	**91.4**	121.1	124.8	150.5	**85.1**
0–250	173	94.7	109.7	91.6	124.1	150.5	**146.6**
251–500	360	124	**98.3**	116.8	**110.6**	139.9	**107**
501–750	617	92.2	**77.6**	104.5	**97**	91.7	109.7
751–1000	867	111	**107.2**	92	106.8	97.9	108.1
1001–2500	1610	94.4	115.8	92.7	108.7	112.1	**106**
2501–5000	3206	145	**131.5**	80.9	92.1	94.9	**80.2**

the attenuation improved the average n-i1NN performance, it was still out-preformed by both attenuated and nonattenuated versions of the other two mapping variants.

- Both the attenuation-ikNN mapping and the attenuation-n-ikNN mapping surpass their original mappings' performance. Attenuation-ikNN mapping only slightly improves overall performance, likely due to the original ikNN mapping already favoring the near regions, thus our attenuation approach may less of an impact. Comparatively, the attenuation provides a significant improvement for the performance of the n-ikNN mapping. This is likely due to the normalization favoring the far end, which may result in overemphasized values. The attenuation reduces these far region values. The iRkNN mapping performs worse with attenuation, likely for the reasons explained above.
- The performance of both the ikNN and n-ikNN mappings improve with attenuation not only in cases of uneven density distributions (far half empty, far half not empty and far half less than 5%), but also on most of the total count grouping categories. The attenuation-ikNN mapping performs worse on 0–250 and 1001–2500 categories than original ikNN mapping, likely due to the various density distributions in these two categories. The attenuation-n-ikNN mapping performs worse than the n-ikNN mapping for the 501–750 and 751–1000 categories. This may be due to the denser near end of the images, in which our normalization suppresses the map values and may cause larger errors.

7. Conclusions

We have presented a new form of labeling for crowd counting data, the ikNN map. We have compared this labeling scheme to commonly accepted labeling approach for crowd counting, the density map. We show that using the ikNN map with an existing

state-of-the-art network improves the accuracy of the network compared to density map labelings.

We have also provided a new network architecture MUD-ikNN, which uses multi-scale drop-in replacement upsampling via transposed convolutions to take full advantage of the provided ikNN labeling. This upsampling combined with the ikNN maps further improves crowd counting accuracy.

We have further studied several variations of the ikNN labeling mechanism, including the inverse squared kNN, the inverse square root kNN, the normalized ikNN and the weighted ikNN to analyze the impact of camera perspective views, image resolutions, and the changing rates of the mapping functions. Experiments on a dataset show that the inverse square root kNN has the best performance, with the original ikNN being a close second.

In addition, we have investigated an attenuation mechanism to handle uneven crowd distributions in an image, especially when the far end of the image is (approximately) empty. We further study the impact of weighting and attenuation to various cases of the crowd distributions and have found that the attenuation mechanism helps in cases of uneven crowd distributions, thus improving the overall performance. Critical discussions are provided for future studies in terms of perspective distortions, crowd occlusions, and label resolutions.

Statistically, the normalized and weighted approaches do correct the perspective distortion crowd mapping values as expected, but the preliminary experiments on one dataset show that the overall performance is degraded when using these distortion corrections. This presents an avenue for further future investigation. For example, the use of real camera geometry with respect to the ground plane, which could be available for real-world applications such as surveillance or transportation cameras, could be used to correct the perspective distortion, rather than use the crowd count labeling of individual images, which can often be inaccurate or even erroneous for some of the images due to the varying distributions of the crowd.

Finally, we want to note here the mean absolution error (MAE) is still relatively large (over 10%) and therefore there is still space to improve. Using our mapping approaches with the latest state-of-the-art crowd counting architectures may provide further developments. We have demonstrated the improvements gained by using increased label resolutions and provide an upsampling map module which in principle can be generally used by other crowd counting architectures. These approaches can be used a drop-in replacement in other crowd counting architectures, as we have done for DenseNet, which resulted in a network which performs favorably compared with the state-of-the-art.

Acknowledgments

The research is supported by the US National Science Foundation (NSF) through a Partnerships for Innovation Award #1827505 and Smart and Connected

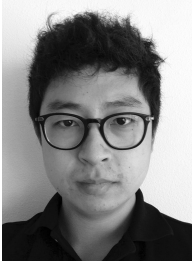
Community Planning Award #1737533, Air Force Office of Scientific Research (AFOSR) via Award #FA9550-21-1-0082, and Bentley Systems, Incorporated, through a CUNY-Bentley Collaborative Research Agreement (CRA) 2017-2020. Additional support is provided by the Office of the Director of National Intelligence (ODNI) via the Intelligence Community Center of Academic Excellence (IC CAE) at Rutgers University (Awards #HHM402-19-1-0003 and #HHM402-18-1-0007).

References

1. C. Arteta, V. Lempitsky and A. Zisserman, Counting in the wild, in *European Conf. Computer Vision* (Amsterdam, The Netherlands, 2016), pp. 483–498.
2. V. Badrinarayanan, A. Kendall and R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.* **39** (12) (2017) 2481–2495.
3. X. Cao, Z. Wang, Y. Zhao and F. Su, Scale aggregation network for accurate and efficient crowd counting, in *Proc. European Conf. Computer Vision (ECCV)* (Munich, Germany, 2018), pp. 734–750.
4. A. B. Chan, Z.-S. J. Liang and N. Vasconcelos, Privacy preserving crowd monitoring: Counting people without people models or tracking, in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conf.* (Anchorage, Alaska, USA, 2008), pp. 1–7.
5. K. Chen, S. Gong, T. Xiang and C. Change Loy, Cumulative attribute space for age and crowd density estimation, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (Portland, Oregon, USA, 2013), pp. 2467–2474.
6. K. Chen, C. C. Loy, S. Gong and T. Xiang, Feature mining for localised crowd counting, in *BMVC*, Vol. 1 (2012), p. 3.
7. K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (Las Vegas, Nevada, USA, 2016), pp. 770–778.
8. M. Hossain, M. Hosseinzadeh, O. Chanda and Y. Wang, Crowd counting using scale-aware attention networks, in *2019 IEEE Winter Conf. Applications of Computer Vision (WACV)* (Waikoloa Village, Hawaii, USA, 2019), pp. 1280–1288.
9. G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, Densely connected convolutional networks, in *CVPR*, Vol. 1 (2017), p. 3.
10. H. Idrees, I. Saleemi, C. Seibert and M. Shah, Multi-source multi-scale counting in extremely dense crowd images, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (Portland, Oregon, USA, 2013), pp. 2547–2554.
11. 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 *Proc. European Conf. Computer Vision (ECCV)* (Munich, Germany, 2018), pp. 532–546.
12. X. Jiang, L. Zhang, M. Xu, T. Zhang, P. Lv, B. Zhou, X. Yang and Y. Pang, Attention scaling for crowd counting, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (Seattle, Washington, USA, 2020), pp. 4706–4715.
13. I. H. Laradji, N. Rostamzadeh, P. O. Pinheiro, D. Vazquez and M. Schmidt, Where are the blobs: Counting by localization with point supervision, in *Proc. European Conf. Computer Vision (ECCV)* (Munich, Germany, 2018), pp. 547–562.
14. V. Lempitsky and A. Zisserman, Learning to count objects in images, in *Advances in Neural Information Processing Systems* (2010), pp. 1324–1332.

15. Y. Li, X. Zhang and D. Chen, CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (Salt Lake City, Utah, USA, 2018), pp. 1091–1100.
16. Z. Lin and L. S. Davis, Shape-based human detection and segmentation via hierarchical part-template matching, *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(4) (2010) 604–618.
17. V. Ranjan, H. Le and M. Hoai, Iterative crowd counting, in *Proc. European Conf. Computer Vision (ECCV)* (Munich, Germany, 2018), pp. 270–285.
18. D. B. Sam, S. Surya and R. V. Babu, Switching convolutional neural network for crowd counting, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Vol. 1 (Honolulu, Hawaii, USA, 2017), p. 6.
19. Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu and X. Yang, Crowd counting via adversarial cross-scale consistency pursuit, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (Salt Lake City, Utah, USA, 2018), pp. 5245–5254.
20. Z. Shi, L. Zhang, Y. Liu, X. Cao, Y. Ye, M.-M. Cheng and G. Zheng, Crowd counting with deep negative correlation learning, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (Salt Lake City, Utah, USA, 2018), pp. 5382–5390.
21. V. A. Sindagi and V. M. Patel, CNN-based cascaded multi-task learning of high-level prior and density estimation for crowd counting, in *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE Int. Conf.* (Lecce, Italy, 2017), pp. 1–6.
22. M. Wang and X. Wang, Automatic adaptation of a generic pedestrian detector to a specific traffic scene, in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conf.* (Colorado Springs, Colorado, USA, 2011), pp. 3401–3408.
23. B. Wu and R. Nevatia, Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors, in *Tenth IEEE Int. Conf. Computer Vision (ICCV'05)* (Beijing, China, 2005), pp. 90–97.
24. M. D. Zeiler, D. Krishnan, G. W. Taylor and R. Fergus, Deconvolutional networks, in *2010 IEEE Computer Society Conf. Computer Vision and Pattern Recognition* (Sanfransico, California, USA, 2010), pp. 2528–2535.
25. C. Zhang, H. Li, X. Wang and X. Yang, Cross-scene crowd counting via deep convolutional neural networks, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (Boston, Massachusetts, USA, 2015), pp. 833–841.
26. A. Zhang, J. Shen, Z. Xiao, F. Zhu, X. Zhen, X. Cao and L. Shao, Relational attention network for crowd counting, in *Proc. IEEE Int. Conf. Computer Vision* (Seoul, Korea, 2019), pp. 6788–6797.
27. Y. Zhang, D. Zhou, S. Chen, S. Gao and Y. Ma, Single-image crowd counting via multi-column convolutional neural network, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (2016), pp. 589–597.

Greg Olmschenk is a NASA Postdoctoral Fellow at the NASA Goddard Space Flight Center. His research focuses on neural networks to detect and characterize various astrophysical phenomena. He received his Ph.D. from the Graduate Center of the City University of New York. The presented work was primarily performed during his Ph.D. studies.



Xuan Wang received his BE degree in Computer Science in China. He received his MS degree in Computer Science from Stevens Institute of Technology, New Jersey. He is currently a second-year Ph.D. student at the CUNY Graduate Center and a Research Assistant at the City College Visual Computing Laboratory (CcvcL), supervised by Professor Zhigang Zhu. Before he joined the Ph.D. Program at the CUNY Graduate Center, he worked in industry for a number of years as a software engineer. His research interests include computer vision and machine learning, focusing on context-based scene and activity understanding.

Hao Tang is an Associate Professor of Computer Science at The Borough of Manhattan Community College, City University of New York. He earned his Ph.D. degree in Computer Science, concentrating in the Computer Vision, at the Graduate Center of CUNY. His research interests are in the fields of virtual and augmented reality, human-computer interaction, mobile vision and navigation and the applications in surveillance, assistive technology, and education. His research paper was selected as the best paper finalist of International Conference on Multimedia and Expo.

Zhigang Zhu received his BE, ME and Ph.D. degrees, all in computer science, from Tsinghua University, Beijing. He was an Assistant Professor, Lecturer, and then Associate Professor at Tsinghua University, Beijing, and a Research Fellow/Visiting Professor at the University of Massachusetts in Amherst. He is current Herbert G. Kayser Chair Professor of Computer Science, at the City College of New York (CCNY) and the CUNY Graduate Center, where he directs the City College Visual Computing Laboratory (CcvcL). His research interests include 3D computer vision, multimodal sensing, human-computer interaction (HCI), virtual/augmented reality, and various applications in assistive technology, robotics, surveillance and transportation. He has published over 190 peer-reviewed technical papers in the related fields. He is an Associate Editor of the Machine Vision Applications Journal, Springer.