Computer Science and Information Systems 19(3):1177–1198 https://doi.org/10.2298/CSIS210925023T

Re-evaluation of the CNN-based State-of-the-art Crowd-counting Methods with Enhancements

Matija Teršek*, Maša Kljun*, Peter Peer, and Žiga Emeršič

Faculty of computer and information science
Večna pot 113, SI-1000 Ljubljana
{matija.tersek, masa.kljun}@student.uni-lj.si,
{peter.peer, ziga.emersic}@fri.uni-lj.si

Abstract. Crowd counting has a range of applications and it is an important task that can help with the accident prevention such as crowd crushes and stampedes in political protests, concerts, sports, and other social events. Many crown counting approaches have been proposed in the recent years. In this paper we compare five deep-learning-based approaches to crowd counting, reevaluate them and present a novel CSRNet-based approach. We base our implementation on five convolutional neural network (CNN) architectures: CSRNet, Bayesian Crowd Counting, DM-Count, SFA-Net, and SGA-Net and present a novel approach by upgrading CSRNet with application of a Bayesian crowd counting loss function and pixel modeling. The models are trained and evaluated on three widely used crowd image datasets, ShanghaiTech part A, part B, and UCF-QNRF. The results show that models based on SFA-Net and DM-Count outperform state-of-the-art when trained and evaluated on the similar data, and the proposed extended model outperforms the base model with the same backbone when trained and evaluated on the significantly different data, suggesting improved robustness levels.

Keywords: Crowd counting, convolutional neural networks, deep learning.

1. Introduction

Automatic estimation of a number of people in a crowd as illustrated in Figure 1 is an important technique with applications in many fields. Political protests, rallies, concerts, religious events, etc., are just some of the situations that can benefit from the automatic crowd counting, since having a good estimate of the crowd can help prevent crowd crushes, stampedes, and other accidents. Furthermore, in the light of the recent pandemic of the COVID-19, crowd counting and crowd analysis can help prevent the spread of the virus by ensuring enough physical distance between people in some usually crowded public places, such as stores, cinemas, recreational areas, etc.

In addition to the mentioned applications, crowd counting is popular as it can be easily extended to a task of counting objects in other fields. Some of them include counting vehicles for traffic control [37, 49], monitoring discarded fish catch and counting animals for environmental control [2, 13, 52, 50], counting leafs for plant phenotyping [1], estimating the number of cells in microscopic images [27] or more generally detecting moving objects [6, 17, 23] Counting of objects is crucial in such tasks as it automates and speeds up otherwise tedious processes.

* Both authors contributed equally

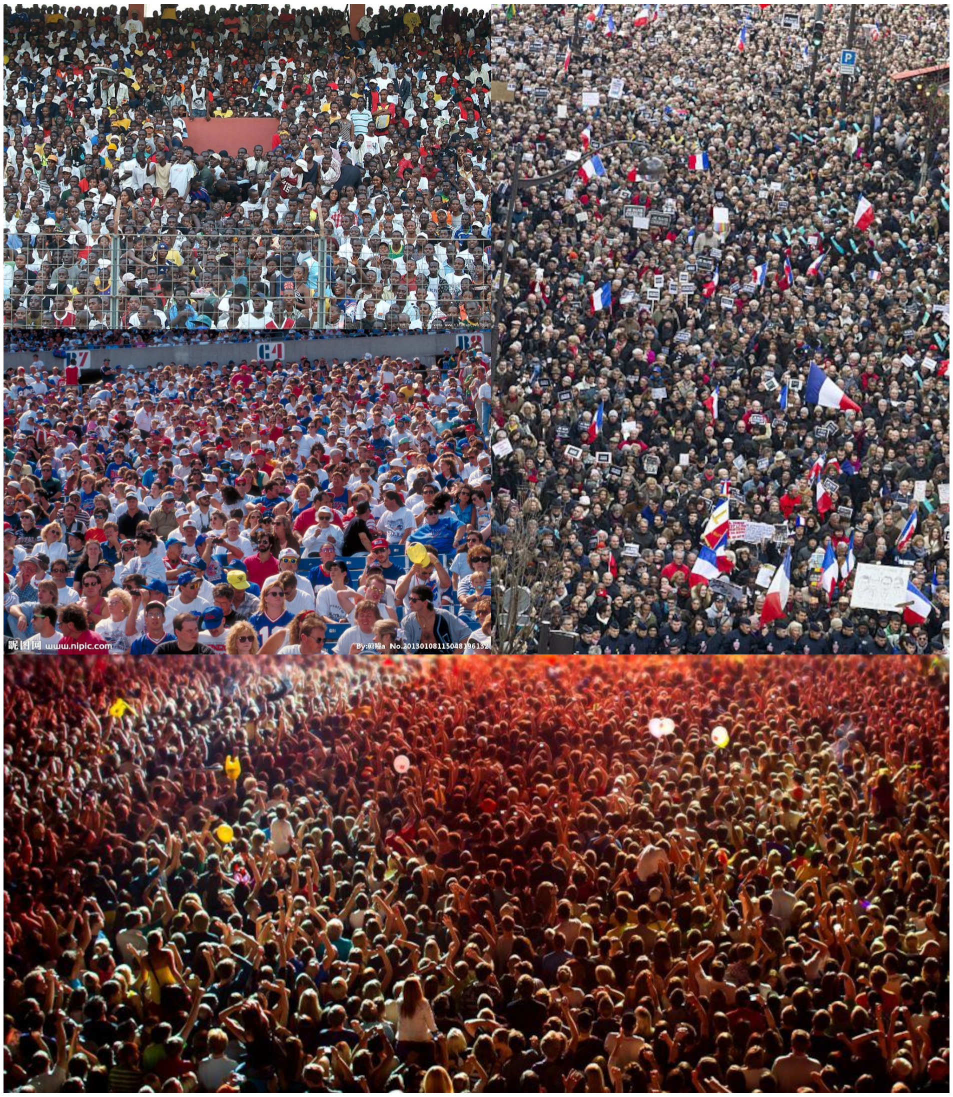


Fig. 1. Figure shows 4 randomly chosen images from the ShanghaiTech part A train set. We can see that the images are cropped to contain dense crowds only.

Because of the wide variety of applications of crowd counting methods, a lot of research has been made and many different algorithms have been proposed. Different approaches to crowd counting exist, and they can be roughly divided into 3 groups– detection, regression, and density based. While some related works include overviews of existing crowd analysis methods [16, 32, 44, 48, 65], the other focus more on discovering the new approaches [29, 33, 57, 59, 71].

In this paper we focus on CNN-based approaches, as they recently began to gain in the popularity. We briefly describe and provide key features of five state-of-the-art models.

Unlike some of the related works, which only gather the results from authors' papers, we try to train and evaluate the models ourselves on three popular crowd counting datasets. Furthermore, we propose an improvement for one of the models and compare it to the others. Source code and pretrained weights are available at our GitHub[1]. To summarize our key contributions:

- Direct comparison of some the most popular state-of-the-art models and their re-implementation. To the best of our knowledge, no comparison to this extent has not been made in literature yet.
- Presentation and comparison of our own model. The model's architecture is based on CSRNet with dilated convolutions, with added pixel modeling and enhanced Bayesian loss function.
- We make our implementations freely available for other researchers to use and modify.

This paper is organized as follows: In Chapter 2 we provide the most common approaches to crowd counting. In Chapter 3 we describe five state-of-the-art CNN models and our suggested improvement. In Chapter 4 we describe the three datasets on which we evaluate the models and discuss the results of our evaluation.

2. Crowd counting approaches

The goal of crowd (of people) counting methods is to determine the number of people present in a particular area. There exist many different approaches of doing this and we can divide the traditional approached into 3 main categories - detection, regression, and density based approaches [16, 48]. CNNs dominate the more recent approaches, which can be categorized into its own group. We would like to emphasize that despite the fact that the term crowd could be used for any type of crowd of objects, all the mentions of a crowd refer to the crowd of people.

For the most comprehensive overview of the area we refer the reader to some of the surveys on crowd counting approaches, such as [11]. Here we only skim through the more popular approaches from the recent years.

Detection based approaches: This is the most straight-forward approach that can use whole bodies (Monolithic detection [9, 10, 15, 26, 43, 53–55, 63]) or just parts of it (Part-based detection [12, 28, 30, 62]), e.g., the combination of head and hands. Approaches in the first group use features such as Haar wavelets or histogram of oriented gradient (HOG) to represent the body, and then use a classifier with the sliding window approach across the image to detect person candidates. Models can be then learned using support vector machines, boosting, random forests, etc.

In the recent years many object detectors based on CNNs were also presented. YOLO network [40] applies a single neural network to the full image, divides it into region, and predicts bounding boxes and probabilities for each region. Other CNN approaches include Fast R-CNN [19] and Faster R-CNN [41].

[1] https://github.com/tersekmatija/crowd-counting-cnns

Another approach uses shape learning, where humans are modeled with 3d shapes composed of ellipsoids. A stochastic process is then employed to estimate the number and shape configuration which best explains a given foreground mask in a scene [18, 69]. The drawback of detection based approaches is that they fail in high occlusion situations or in highly crowded spaces [32].

Regression based approaches: The idea of this group's methods is not to count individuals, but to estimate the crowd density, which is specifically useful in more crowded places [4, 5, 7–9, 20, 21, 31, 34, 36, 38, 42, 51]. Methods in this group first encodes low-level information with the help of foreground, edge, texture, and/or gradient features. Then, with the help of a regression model, a mapping between low-level features and people count is made. Different regressions, such as linear regression, ridge regression, neural network, etc., can be used. The drawback of regression based approaches is that when the same object is placed in different depths in the image, the values of features extracted from those objects can vary upon the depth of where the object was placed. However, this problem can be tackled by geometric correction [32].

Density based approaches: The idea of this group's methods, such as [39, 61, 64, 67] in its most simplistic form is to obtain a density map from an image and then integrate it in order to get the estimation of people in the image. Contrary to the previous approaches, these also consider the spatial information. The pioneering work include [27], who suggest learning a linear mapping between local patch features and corresponding object density maps. The methods differ in the choice of a training loss function (e.g., squared error between the predicted density values and the ground truth) and in the choice of a density map prediction method (e.g., with the help of a linear model) [24].

CNNs: In 2015 the pioneering work with deep networks in crowd counting was introduced in [58], introducing CNN approaches to the crowd counting. Since then many of CNN based approaches were proposed. The basic idea behind CNN based approaches is that they normally try to predict the density map from the input image and infer the count from it. This also means they are the most similar to the traditional density based approaches. Models that are based on CNNs differ in the usage of different backbones (e.g., VGG-16, VGG-19, Inception v3), loss functions, additional maps (e.g., attention map), and model structure (e.g., single or multi column).

In recent surveys [48, 11] authors classify CNN-based approaches into four categories, based on the property of the networks: *Basic CNNs* include networks with basic CNN layers and represent initial deep learning approaches for crowd counting [14, 35, 56, 58, 67], *scale-aware models* that leverage multi-column or multi-resolution architectures to achieve scale robustness [3, 25, 37, 68], *context-aware models* that incorporate global and local contextual information to improve performance [45, 46], and *multi-task frameworks* that combine crowd counting with tasks such as crowd velocity estimation, etc. [2, 47, 66, 70] Based on the inference methodology, they also classify them into *patch-based*, where models are trained using patches from the image and the inference is done using sliding window approach [2, 3, 14, 25, 35, 37, 56, 58, 66, 70], and whole image-based [45–47, 68, 60].

We describe five CNN models for crowd counting along with their key features and one improved model in the next chapter.

3. CNN models

In this section we shortly describe each of the models. Note that we put the main focus on their features, where they differ the most from each other. The models were chosen as each of them made a significant contribution in the field, as well as based on their popularity in literature, at the time of writing.

3.1. CSRNet

The architecture of this model is divided into 2 parts: a CNN at the front-end and a dilated CNN at the back-end. The basis of CSRNet front-end is build on VGG-16 model with the fully-connected layers removed [29]. Ten layers of VGG-16 are kept, with only three pooling layers instead of five. The back-end consists of six dilated convolutional layers, for which the authors suggest that it represents a good alternative to the pooling layers. Dilated convolution can be used instead of the pooling layer, since it maintains the resolution of feature map and contains more detailed information. Another 1×1 convolutional layer is added as the output layer.

Authors suggest different models, which are determined by different back-end settings that vary in the dilation rate. We use model B in our experiments, as it is the most successful [29], where the dilation rate is set to 2 for all the back-end layers.

Dilated convolution. The idea of the dilated convolution is that it uses sparse kernels, which enlarge the receptive field. The same can be achieved by adding more convolutional layers, however, that increases the computational cost.

For input $x(m,n)$ and filter $w(i,j)$, of length M, width N, and the dilation rate r ($r = 1$ results in a normal convolution), we can define output $y(m,n)$ of the dilated convolution as

$$\sum_{i=1}^{M}\sum_{j=1}^{N} x(m + r \times i,\ n + r \times j) w(i,j). \tag{1}$$

Loss function and training. Loss function is derived from the Euclidean distance between the ground truth and estimated density map. The loss function is defined as

$$\mathcal{L} = \frac{1}{2N}\sum_{i}^{N} ||D_i^{est} - D_i^{gt}||_2^2, \tag{2}$$

where N is the size of the training batch, D_i^{est} the density map generated by the CSRNet, and D_i^{gt} the ground truth density map of the input image.

In training the first 10 convolutional layers are fine-tuned from a trained VGG-16. Initial settings for other layers are set with the help of a Gaussian distribution with 0.01 standard deviation, and stochastic gradient descent (SGD) with rate $1e - 6$ is applied during training.

3.2. Bayesian crowd counting

The Bayesian model uses VGG-19 as the backbone, with the last pooling and the subsequent fully connected layers removed. The output of the backbone is upsampled to $\frac{1}{8}$ of the input image size by bilinear interpolation and fed to a regression header. The regression header consists of two 3×3 convolutional layers, one with 256 and the other with 128 channels, and one 1×1 convolutional layer. The produced output is a density map [33].

Bayesian crowd counting model differs from other CNN based models in the utilization of a loss function. Opposed to the previous models, which use a Gaussian kernel to obtain the ground truth density map and define loss function as a sum of pointwise distances between ground truth and estimated density maps, it uses a novel Bayesian loss function.

Bayesian Loss function and training. We can derive the loss function as follows. Let x be a random variable describing the spatial location, and y be a random variable representing the annotated head point. Let $m = 1, ..., M$ where M is the number of pixels in the density map and let $n = 1, ..., N$, where N is the total crowd count. Let z_n be a head position and y_n be a corresponding label. The likelihood function of location x_m given the label y_n can be defined as

$$p(x_m|y_n) = N(x_m; z_n, \sigma^2 1_{2\times 2}), \tag{3}$$

where $N(x_m; z_n, \sigma^2 1_{2\times 2})$ is a 2D Gaussian distribution evaluated at x_m, with the mean at the annotated point z_n and an isotropic covariance matrix $\sigma^2 1_{2\times 2}$.

Using Bayes we can then compute

$$p(y_n|x_m) = \frac{p(x_m|y_n)p(y_n)}{p(x_m)} = \frac{N(x_m; z_n, \sigma^2 1_{2\times 2})}{\sum_{n=1}^{N} N(x_m; z_n, \sigma^2 1_{2\times 2})}. \tag{4}$$

The Bayesian loss function can be defined as

$$\mathcal{L}^{\text{Bayes}} = \sum_{n=1}^{N} \mathcal{F}(1 - E[c_n]), \tag{5}$$

where $\mathcal{F}$ is a distance function (ℓ_1) and $E[c_n]$ is the expected value of a total count associated with y_n, that can be computed as

$$E[c_n] = \sum_{m=1}^{M} p(y_n|x_m) D^{est}(x_m). \tag{6}$$

When inferring, the total count is just a sum over an estimated density map.

Additionally, authors introduce the background pixel modeling for background pixels that are far away from any of the annotation points. They introduce an additional background label $y_0 = 0$ in addition to the head labels, as it makes no sense to assign the background pixels to any of the head labels. The posterior label probability is then rewritten and additional expected count for the entire background $E[c_0]$ is introduced. Pixel modeling defines a new, enhanced loss function, as also described in Equation 7.

MSRA initializer is used for the initialization of the regression header, whereas the backbone is pre-trained on ImageNet. Parameters are updated with the help of the Adam optimizer with an initial learning rate $1e - 5$.

3.3. Our proposed model

Concepts such as dilated convolution, use of Bayesian loss instead of Gaussian kernel and using pixel modeling to suppress background pixels have been shown to improve crowd couting performance [29]. We infer that combining all of these key concepts should result in a more robust and better performing crowd counting model. Therefore we base our proposed model on the CSRNet and Bayesian crowd counting loss function and pixel modeling [33, 33], with the goal of improving performance.

The basic structure of our model is the same as the one of the CSRNet, described in Subsection 3.1. We use the first ten layers of the VGG-16 with 3 pooling layers for the front-end, 6 convolutional layers with the dilation rate set to 2 as the back-end, and an additional 1×1 layer as the output layer.

Loss function and training. Instead of CSRNet's loss function provided in Equation 2, we use the enhanced loss function defined as:

$$\mathcal{L}^{Bayes+} = \sum_{n=1}^{N} \mathcal{F}(1 - E[c_n]) + \mathcal{F}(0 - E[c_0]). \tag{7}$$

The weights are initialized in the same way as in the CSRNet. The first 10 convolutional layers are fine-tuned from a trained VGG-16, whereas initial settings for other layers are obtained with the help of a Gaussian distribution with 0.01 standard deviation. Parameters are updated with the help of the Adam optimizer with an initial learning rate of $1e - 6$.

3.4. SFANet

The next model we analyse is SFANet [71]. It uses the first 13 layers of a pre-trained VGG-16-bn (VGG-16 with batch normalization) as the front-end feature map extractor. It is suitable as it has a strong ability to represent features and can be easily concatenated by the back-end dual path networks. Four source layers (conv2-2, conv3-3, conv4-3, and conv5-3) are then connected to a dual multi-scale fusion networks with attention (density map path and attention map path), which represent the back-end. Attention map path is incorporated to tackle the background noise and non-uniformity of crowd distributions.

Loss function and training. In most models an Euclidean loss is used for measuring estimation error, which is defined as:

$$\mathcal{L}^{\text{DEN}} = \frac{1}{N} \sum_{i=1}^{N} \|D_i^{est} - D_i^{gt}\|^2, \tag{8}$$

where D_i^{est} is the estimated density map of i-th input image, D_i^{gt} represents the ground truth density map, and N is the batch size. SFANet also uses the described loss function. In addition, the model uses the attention map loss function, a binary class entropy defined as

$$\mathcal{L}^{\text{ATT}} = -\frac{1}{N}\sum_{i=1}^{N}(A_i^{gt}\log(P_i) + (1 - A_i^{gt})\log(1 - P_i)), \tag{9}$$

where A_i^{gt} is the attention map ground truth, and P_i probability of each pixel in predicted attention map activated by sigmoid function.

The unified loss function is then defined as

$$\mathcal{L} = \mathcal{L}^{\text{DEN}} + \alpha\mathcal{L}^{\text{ATT}}, \tag{10}$$

with α weighting weight set to 0.1.

The first 13 layers of a pre-trained VGG-16-bn are applied as the front-end feature extractor. Other parameters are randomly initialized with a Gaussian distribution with a standard deviation 0.01. Parameters are updated with the help of Adam optimizer with learning rate of $1e - 4$ and weight decay of $5e - 3$.

Ground truth. Density map ground truth D^{gt} is obtained similarly as in most models, with the use of Gaussian kernels.

Attention map ground truth is obtained from D^{gt} and Gaussian kernel as

$$\begin{aligned} \mathbb{Z} &= D_i^{gt} \times G_{\mu,\sigma^2}(x), \\ A_i^{gt}(x) &= \begin{cases} 0, & x < thresh \\ 1, & x \geq thresh \end{cases}, \forall x \in \mathbb{Z}, \end{aligned} \tag{11}$$

with $thresh$ set to 0.001.

3.5. DM-Count

DM-Count model considers crowd counting as a distribution matching problem [57]. The architecture of the model is based on the VGG-19 and is the same as in the Bayesian Crowd Counting model (see Subsection 3.2). Different to the previous models, who use density map estimations that are computed with the help of Gaussian kernels, DM-Count can preprocess ground truth annotations without the use of a Gaussian. Instead it uses Optimal Transport (OT) to measure the similarity between the normalized predicted density map and the normalized ground truth density map. OT computation is then stabilized with the help of a Total Variation (TV) loss.

Loss function and training. The loss function is the combination of the counting loss, optimal transport loss, and the total variation loss. Let $h \in \mathbb{R}_+^n$ be a vectorized binary map for dot annotation, and $\hat{h} \in \mathbb{R}_+^n$ a vectorized predicted density map.

Counting loss.

$$\mathcal{L}^{\text{COUNT}}(h, \hat{h}) = |\|h\|_1 - \|\hat{h}\|_1|, \tag{12}$$

where $\| \cdot \|$ denotes the L_1 norm.

Optimal transport loss. Since h and $\hat{h}$ are both unnormalized density functions, they can be turned into the probability density functions (PDF) with dividing them by their respective total mass. Optimal transport loss is then defined as

$$\begin{aligned}\mathcal{L}^{\mathrm{OT}}(h,\hat{h}) &= \mathcal{W}(\frac{h}{\|h\|_1},\frac{\hat{h}}{\|\hat{h}\|_1}) \\ &= \left\langle \alpha^*, \frac{h}{\|h\|_1} \right\rangle + \left\langle \beta^*, \frac{\hat{h}}{\|\hat{h}\|_1} \right\rangle,\end{aligned} \tag{13}$$

where $\mathcal{W}$ is a Monge-Kantorovich's Optimal Transport cost (see [57] for the definition), with α^* and β^* being solutions of the optimal transport problem.

Authors suggest the use of OT instead of some other measure of similarity between two PDFs, such as Kullback-Leibler divergence or Jensen-Shannon divergence, as it provides a valid gradient to train a network. The gradient with respect to $\hat{h}$ can be obtained as

$$\frac{\partial \mathcal{L}^{OT}(h,\hat{h})}{\partial \hat{h}} = \frac{\beta^*}{||\hat{h}||_1} - \frac{\langle \beta^*, \hat{h} \rangle}{||\hat{h}||_1^2}, \tag{14}$$

which can be back-propagated to learn the parameters of the density estimation network.

Total variation loss. OT loss is optimized with Sinkhorm algorithm for approximating α^* and β^* in each training iteration. Due to this optimization, OT loss approximates well more dense areas, but it performs poorer for the low density areas. To cope with that, Total variation loss is additionally used and can be defined as

$$\mathcal{L}^{\mathrm{TV}}(h,\hat{h}) = \frac{1}{2}\left\| \frac{h}{\|h\|_1} - \frac{\hat{h}}{\|\hat{h}\|_1} \right\|_1. \tag{15}$$

3.6. SGANet

The SGANet model is the first model that investigates Inception-v3 as a backbone network instead of VGG-16, VGG-19, or ResNet, as in the most state-of-the-art models [59]. Fully-connected layers and two maxpooling layers are removed. Before the last Inception Module an upsampling layer is added, which is connected to both, the attention layer and the last Inception Module. Attention layer's output is then applied to the feature maps generated by the last Inception Module.

Loss function and training. In SGANet a novel curriculum loss strategy to address the issues caused by extremely dense regions was used. This is a strategy of model learning where easy examples are selected at the beginning of the training and more difficult ones are added to the training set gradually. A threshold is used for determining the difficulty score, where density map pixels with higher values than the threshold have higher difficulty scores, since such pixels are within the regions of denser crowds. The whole training set is used throughout the training process, however, the threshold is first set to a low value and then gradually increased, which turns difficult pixels into easy ones so that they contribute more to the training.

The loss function is defined as a sum of two loss functions:

$$\mathcal{L} = \mathcal{L}^{\text{DEN}} + \lambda \mathcal{L}^{\text{SEG}}, \tag{16}$$

where λ is a hyper-parameter set to 20. The density map loss can be calculated as

$$\mathcal{L}^{\text{DEN}} = \frac{1}{2N} \sum_{i=1}^{N} \|\hat{M}_i^{den} - M_i^{den}\|_F^2 \tag{17}$$

and segmentation map loss is defined as the cross-entropy loss as

$$\begin{aligned} \mathcal{L}^{\text{SEG}} = -\frac{1}{N} \sum_{i=1}^{N} \|M_i^{seg} \odot \log(\hat{M}_i^{seg}) \\ + (1 - M_i^{seg}) \odot \log(1 - \hat{M}_i^{seg})\|_1, \end{aligned} \tag{18}$$

where $\|\cdot\|_1$ denotes the element-wise matrix norm, $\odot$ denotes elementwise multiplication of two same-size matrices, and M^{seg} and M^{den} represent ground truth (without hat) and estimated (with hat) segmentation and density maps.

Ground truth. Ground truth density map M^{den} is obtained using a Gaussian kernel with fixed σ. Segmentation map ground truth is obtained similarly, but as

$$M^{seg}(x) = \sum_{i=1}^{N} \delta(x - x_i) * J_n(x), \tag{19}$$

where $J_n(x)$ is an all-one matrix of size $n \times n$ centered at the position x. [59] set $n = 25$.

Model uses the Adam optimizer for updating the parameters, where the initial learning rate is set to $1e - 4$ and reduced by a factor of 0.5 after every 50 epochs. The weights of the Inception layers are loaded from a pre-trained Inception-v3 model.

4. Experiments and Results

Here we describe our experiments, including data used, evaluation protocol, and present results and findings. We relied on implementations provided by the authors. All models were implemented in Pytorch and trained with provided default parameters.

4.1. Data

We test the described models on the three publicly available datasets, described below.

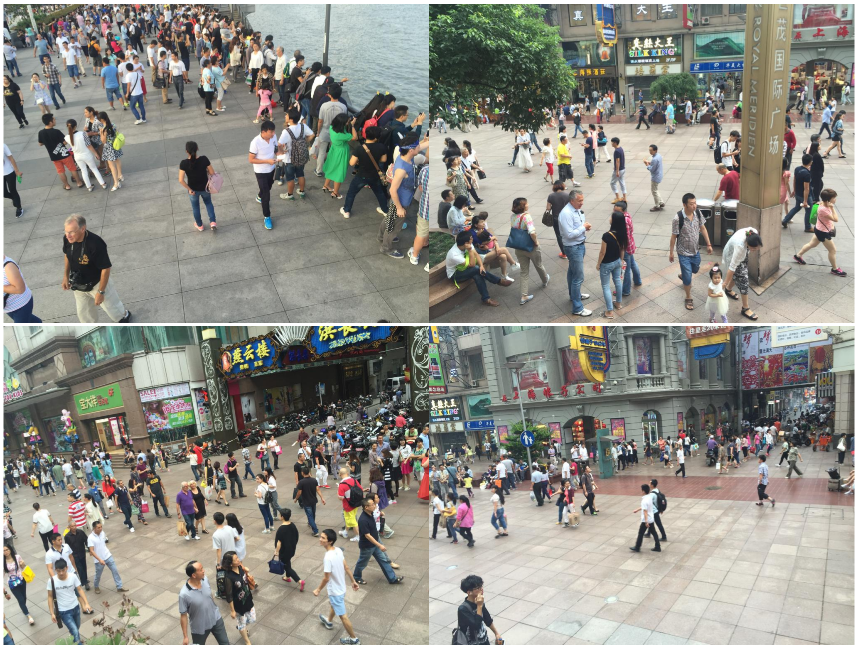


Fig. 2. Figure shows 4 randomly chosen images from the ShanghaiTech part B train set. We can see that the images contain relatively sparse crowds. The background often consists of buildings and vegetation, but can also include rivers as seen in the top left image

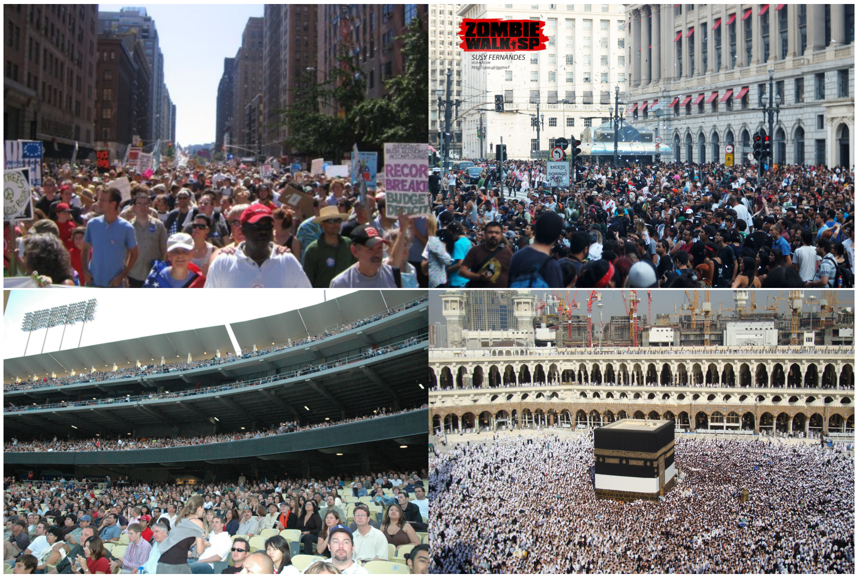


Fig. 3. Figure shows 4 randomly chosen images from the UCF-QNRF train set. We can see that the images are more realistic than the images from the ShanghaiTech part A, as they include not only crowds but also buildings, sky, and vegetation

ShanghaiTech Dataset. ShanghaiTech consists of two parts – part A and part B [68]. Part A contains 482 images downloaded from the internet, containing highly congested scenes. It contains a total of $241,667$ annotated people, with a 501 average per image, and 3139 maximum. It comes split into a train and a test set, containing 300 and 182 images, respectively. Images in this dataset are challenging to count, as they contain extremely congested scenes, varied perspective, and unfixed resolution. Figure 1 shows some examples of ShanghaiTech part A train set images.

Part B contains 716 images that are taken from the busy streets of metropolitan areas of Shanghai. Images are of fixed size and contain total of 88,488 annotated people, with a 124 average per image, and 578 maximum. Same as the part A, it is already split into a train and a test set, containing 400 and 316 images, respectively. As images are captured in metropolitan areas, they contain relatively sparse crowds and include streets, buildings, vegetation, and sometimes rivers as well. In Figure 2 we show some examples of the ShanghaiTech part B train set images.

UCF-QNRF Dataset. UCF-QNRF is among the newest and the largest datasets for crowd counting problems [22]. It consists of 1525 images and contains a total of $1,251,642$ annotated people, with a 815 average, and $12,865$ maximum. It is split into a train and a test set, containing 1201 and 334 images, respectively. Dataset contains images with congested scenes with a diverse set of viewpoints, densities, and lighting variations. Different from the ShanghaiTech part A, which contains images with dense crowds that are cropped to contain crowds only, images from this set also contain buildings, vegetation, sky, and roads, as they are present in realistic scenarios captured in the wild, making the dataset more realistic but also more difficult to count. Figure 3 shows some examples of UCF-QNRF train set images. We summarize the datasets in Table 1.

Table 1. A summary of used datasets. For each we show a number of images, average and maximum people count per image, and a total number of annotated people

Dataset	Images	Avg count	Max count	Annotations
ShanghaiTech A [68]	482	501.4	$3,139$	$241,677$
ShanghaiTech B [68]	716	123.6	578	$88,488$
QNRF [22]	$1,535$	815	$12,865$	$1,251,642$

4.2. Evaluation metrics

We use Mean Absolute Error (MAE) and Mean Squared Error (MSE) for the evaluation and they are defined as follows

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|C_i - C_i^{GT}|, \tag{20}$$

$$\text{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}|C_i - C_i^{GT}|^2}, \tag{21}$$

where n is a number of images, C_i represents the inferred count, and C_i^{GT} represents the ground truth count.

4.3. Evaluation

We train and test the models on the three mentioned datasets – ShanghaiTech part A, part B, and UCF-QNRF. In addition to training and evaluating models separately for the three datasets, we also include the results of training the model with ShanghaiTech part A train set and evaluating it with UCF-QNRF test set, to inspect how well the models learn to generalize when trained on a similar dataset. We show the obtained MAE and MSE in Table 2 and also provide some qualitative evaluation of the results. Furthermore, in Table 3 we also compare the sizes (in millions of trainable parameters) of the evaluated models, showing that the performance of the proposed model is increased without increasing the training complexity.

Quantitative Analysis The best results in general are obtained on the ShanghaiTech part B (SHB) dataset, which is expected due to the low average count per image and relatively sparse crowds. We see that the best results are obtained by the SFA-Net (7.05 MAE) and SGA-Net (11.48 MSE), followed closely by the DM-Count (7.68 MAE). The worst performance is given by the CSRNet (11.27 MAE), which is outperformed by our

Table 2. In this table we show the evaluation of the models in terms of MAE and MSE on different datasets. The best results are marked in bold. We see that SFA-Net and DM-Count perform the best on ShanghaiTech part A (SHA), with the first giving the best performance on ShanghaiTech part B (SHB), and the latter giving the best performance also on the UCF-QNRF (QNRF). In terms of MSE, SGA-Net outperforms the SFA-Net on the SHB dataset. Bayesian Crowd Counting yields the best results when trained on SHA and evaluated on QNRF. We also show that our combination of Bayesian Crowd Counting model and CSRNet, Bayesian CSRNet, is in fact an improvement of the original CSRNet model. "/" denotes situations where we could not execute the training due to our hardware limitations. However, in these cases, where possible, we report values from models' papers – denoted by a *

Datasets	SHA		SHB		QNRF		QNRF on SHA	
Method	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
CSRNet	75.44	113.55	11.27	19.32	/	/	199.54	319.09
Bayesian CSRNet	69.46	111.73	8.48	13.55	103.94	186.22	139.83	260.59
Bayesian Crowd Counting	66.92	112.07	8.27	13.56	90.43	161.41	**138.39**	**256.81**
DM-Count	61.39	**98.56**	7.68	12.66	**88.97**	**154.11**	141.43	260.23
SFA-Net	**59.58**	99.43	**7.05**	12.18	$\mathit{100.8}^*$	$\mathit{174.5}^*$	170.29	365.59
SGA-Net	61.58	101.59	7.60	**11.48**	$\mathit{89.1}^*$	$\mathit{150.6}^*$	/	/

Table 3. A size comparison of models. For each model we show a number of trainable parameters (in table denoted by # of TP) in millions (M)

	CSRNet	Bay. CSRNet	Bay. Crowd Count.	DM-Count	SFA-Net	SGA-Net
# of TP [M]	**16.3**	**16.3**	21.5	21.5	17.0	18.1

improved model Bayesian CSRNet (8.48 MAE) and Bayesian Crowd Counting model (8.27 MAE).

The results on ShanghaiTech part A (SHA) are better than those on the UCF-QNRF due to the smaller dataset and smaller and less complicated images. The best results on SHA dataset are obtained by SFA-Net and DM-Count. While the first has a lower MAE (59.58), the second has lower a MSE (98.56). They are closely followed by SGA-Net (61.58 MAE). The worst performance is obtained by CSRNet (75.44 MAE), however, we show that our Bayesian CSRNet model is in fact an improvement of the original CSRNet, with MAE of 66.92.

Due to the bigger size of the images from the QNRF dataset and our hardware limitations, we were not able to train and evaluate all of the models. In cases like these modifying the models in order to be able to retrain them on our limited settings could result in falsely lower results. In order to avoid that, we either omit reporting results in these cases (denoted as "/" in Table 2) or show results as reported in the models' respective papers (written in italic and denoted by a star after the number in Table 2). However, since we could not verify the procedure we do not consider them in the analysis. Nevertheless, we see that DM-Count once again performs the best (88.97 MAE), and is closely followed by Bayesian Crowd Counting (90.43 MAE). Out of the three, our improved Bayesian CSRNet performs the worst (103.94 MAE).

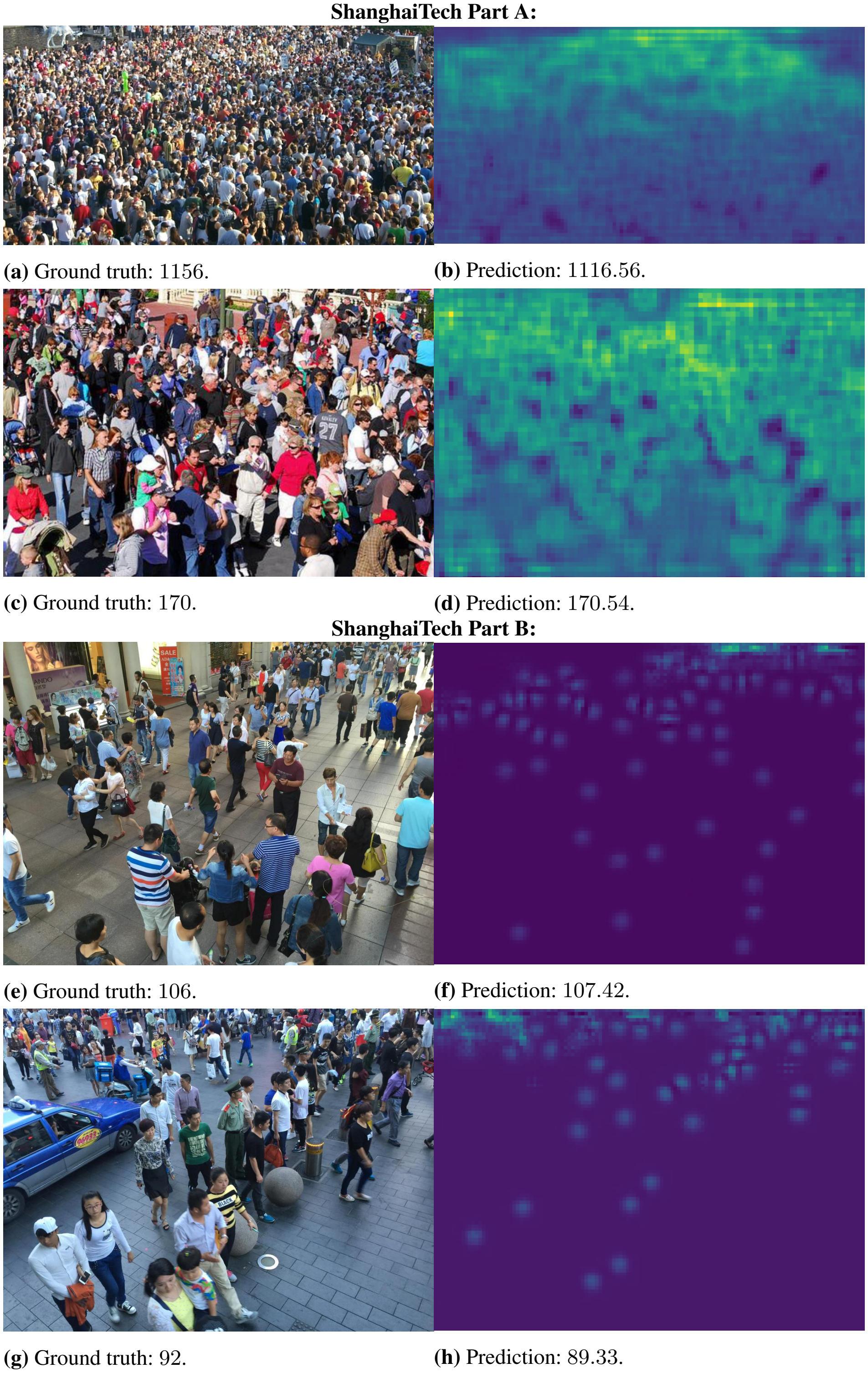
Due the problems with the QNRF dataset, we, in addition to the evaluation of the models on SHA, SHB, and QNRF, also show the results of training the model on SHA train set and evaluating it with QNRF test set, since they both contain relatively dense crowds. The idea behind this experiment is also to see how well the models can learn to generalize, when trained trained on similar, but slightly different images. We see that the overall results here are significantly worse due to the models being trained on images that are cropped to contain crowds only, not including buildings and vegetation in the background. As images in the test set include those objects in the backgrounds, models could misinterpret them and count them as a crowd. The best results are given by the Bayesian Crowd Counting model (138.39 MAE), followed relatively closely by DM-Count (141.43 MAE) and our Bayesian CSRNet (145.03 MAE). The worst performance is once again achieved by the CSRNet (199.54 MAE).

Note that some results differ from the results reported in the author's papers. We argue, that the primary reason for this is that some authors use different implementations in their papers (such as CSRNet, whose authors provide two official implementations – one in Pytorch and one in Caffe). Furthermore, we were unable to train some models due to the computational limitations (and our limited hardware) on the QNRF dataset. In cases like these modifying the models in order to be able to retrain them on our limited settings could result in falsely lower results. In order to avoid that, we either omit reporting results in these cases (denoted as "/" in Table 2) or show results as reported in the original papers (denoted with "*" after the number in Table 2).

Qualitative Analysis We show the results of our improved model in Figures 4 and 5. In the first figure we show the input images from the ShanghaiTech part A and part B test set, and predicted density maps and inferred counts on a model trained on ShanghaiTech datasets. In the second figure we show the input image from the UCF-QNRF test set and predicted density maps and inferred counts on models trained on UCF-QNRF and ShanghaiTech datasets.

5. Conclusion

We reviewed definitions and provided concise descriptions of 5 CNN based models – CSRNet, Bayesian Crowd Counting, DM-Count, SFA-Net and SGA-Net. In addition we trained and evaluated the models ourselves, contrary to many other related works who just provided evalution results from author's papers. We evaluated the models on ShanghaiTech part A dataset, ShanghaiTech part B dataset, and UCF-QNRF dataset. Additionally, we wanted to see how good the results are when training the model on one dataset (ShanghaiTech part A) and evaluating it on another (UCF-QNRF). We saw that the best overall results are those obtained on ShanghaiTech part B dataset, as models work better on images that are less complicated or have less dense crowds. The best results in terms of MAE on the ShanghaiTech part A were obtained with the SFA-Net model, followed closely by the DM-Count model. The first also performed best on the ShanghaiTech part



ShanghaiTech Part A:

(a) Ground truth: 1156.

(b) Prediction: 1116.56.

(c) Ground truth: 170.

(d) Prediction: 170.54.

ShanghaiTech Part B:

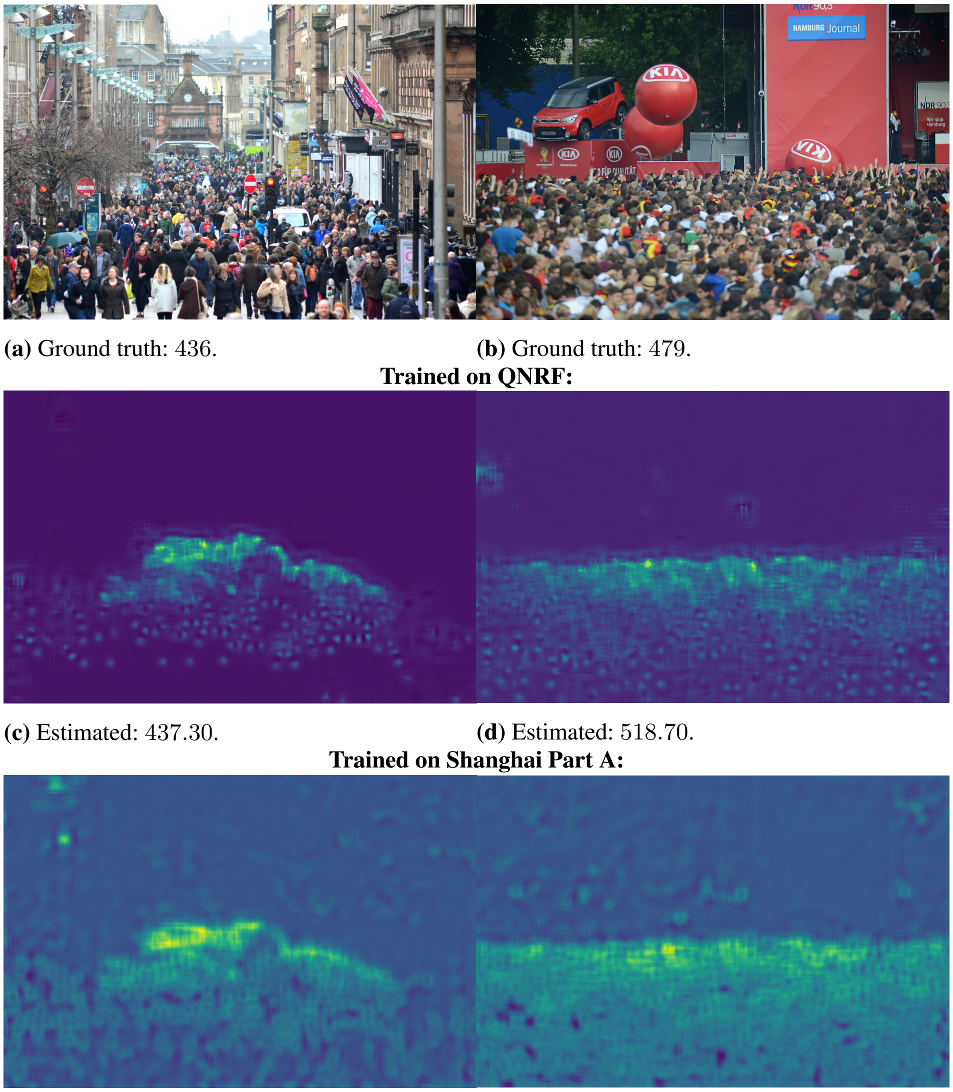
(e) Ground truth: 106.

(f) Prediction: 107.42.

(g) Ground truth: 92.

(h) Prediction: 89.33.

Fig. 4. Images in the left column represent input images from the ShanghaiTech part A (a – d) and ShanghaiTech part B (e – h) test sets, with 1156, 170, 106, and 92 annotated people, respectively. Images in the right column represent the predicted density maps obtained by our improved model Bayesian CSRNet. Estimated counts are 1116.56, 170.54, 107.42, and 89.33, respectively. We use the weights trained on the ShanghaiTech part A train images for the first two density maps (b and d) and weights trained on the ShanghaiTech part B train images for the bottom two density maps (f and h)



(a) Ground truth: 436.

(b) Ground truth: 479.

Trained on QNRF:

(c) Estimated: 437.30.

(d) Estimated: 518.70.

Trained on Shanghai Part A:

(e) Estimated: 444.24.

(f) Estimated: 466.95.

Fig. 5. The upper two images show input images from UCF-QNRF test set with 436 and 479 annotated people. The bottom 4 figures (c – f) show the predicted density maps obtained by our Bayesian CSRNet trained on UCF-QNRF (middle row) and on ShanghaiTech part A (bottom row) train sets. We see that the density maps in the middle row are clearer, as the model is trained on similar images that also contain buildings and streets, and it can better distinguish between them and the crowds. We also see that the inferred result is slightly better on the model trained on UCF-QNRF for the left input image, but the one trained on SHA performs slightly better for the input image from the right column

B, and the latter also performed best on the UCF-QNRF dataset. In terms of MSE, SGA-Net outperforms the SFA-Net on ShanghaiTech part B. The results of training the models on one dataset and evaluating them on the other were less good, however, that was expected due to the smaller train set with images that were cropped to contain crowds only, whereas the images from the test set also included buildings, sky, and vegetation.

In addition to the evaluation of the 5 mentioned models, we also suggested an improvement of the CSRNet. We implemented a new model based on the CSRNet and a Bayesian crowd counting loss function and pixel modeling. We showed that the new model is in fact an improvement of the original model.

Due to the computational limitations we were unable to train/evaluate some models on the QNRF dataset. For the future work we suggest the investigation of possible solutions. Since many datasets exist, we also suggest the evaluation of the models on other datasets (e.g., NWPU). SGA-Net also shows a possible investigation field, as it uses Inception-v3 model instead of VGG-16 or VGG-19, and yet still shows very promising results.

References

1. Aich, S., Stavness, I.: Leaf counting with deep convolutional and deconvolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 2080–2089 (2017)
2. Arteta, C., Lempitsky, V., Zisserman, A.: Counting in the wild. In: European conference on computer vision. pp. 483–498. Springer (2016)
3. Boominathan, L., Kruthiventi, S.S., Babu, R.V.: Crowdnet: A deep convolutional network for dense crowd counting. In: Proceedings of the 24th ACM international conference on Multimedia. pp. 640–644 (2016)
4. Chan, A.B., Liang, Z.S.J., Vasconcelos, N.: Privacy preserving crowd monitoring: Counting people without people models or tracking. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–7. IEEE (2008)
5. Chan, A.B., Vasconcelos, N.: Bayesian poisson regression for crowd counting. In: 2009 IEEE 12th international conference on computer vision. pp. 545–551. IEEE (2009)
6. Chapel, M.N., Bouwmans, T.: Moving objects detection with a moving camera: A comprehensive review. Computer science review 38, 100310 (2020)
7. Chen, K., Gong, S., Xiang, T., Change Loy, C.: Cumulative attribute space for age and crowd density estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2467–2474 (2013)
8. Chen, K., Loy, C.C., Gong, S., Xiang, T.: Feature mining for localised crowd counting. In: Bmvc. vol. 1, p. 3 (2012)
9. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). vol. 1, pp. 886–893. Ieee (2005)
10. Enzweiler, M., Gavrila, D.M.: Monocular pedestrian detection: Survey and experiments. IEEE transactions on pattern analysis and machine intelligence 31(12), 2179–2195 (2008)
11. Fan, Z., Zhang, H., Zhang, Z., Lu, G., Zhang, Y., Wang, Y.: A survey of crowd counting and density estimation based on convolutional neural network. Neurocomputing 472, 224–251 (2022), https://www.sciencedirect.com/science/article/pii/S0925231221016179
12. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE transactions on pattern analysis and machine intelligence 32(9), 1627–1645 (2009)

13. French, G., Fisher, M., Mackiewicz, M., Needle, C.: Convolutional neural networks for counting fish in fisheries surveillance video (2015)
14. Fu, M., Xu, P., Li, X., Liu, Q., Ye, M., Zhu, C.: Fast crowd density estimation with convolutional neural networks. Engineering Applications of Artificial Intelligence 43, 81–88 (2015)
15. Gall, J., Yao, A., Razavi, N., Van Gool, L., Lempitsky, V.: Hough forests for object detection, tracking, and action recognition. IEEE transactions on pattern analysis and machine intelligence 33(11), 2188–2202 (2011)
16. Gao, G., Gao, J., Liu, Q., Wang, Q., Wang, Y.: Cnn-based density estimation and crowd counting: A survey. arXiv preprint arXiv:2003.12783 (2020)
17. Garcia-Garcia, B., Bouwmans, T., Silva, A.J.R.: Background subtraction in real applications: Challenges, current models and future directions. Computer Science Review 35, 100204 (2020)
18. Ge, W., Collins, R.T.: Marked point processes for crowd counting. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2913–2920. IEEE (2009)
19. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
20. Haralick, R.M., Shanmugam, K., Dinstein, I.H.: Textural features for image classification. IEEE Transactions on systems, man, and cybernetics (6), 610–621 (1973)
21. Idrees, H., Saleemi, I., Seibert, C., Shah, M.: Multi-source multi-scale counting in extremely dense crowd images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2547–2554 (2013)
22. Idrees, H., Tayyab, M., Athrey, K., Zhang, D., Al-Maadeed, S., Rajpoot, N., Shah, M.: Composition loss for counting, density map estimation and localization in dense crowds. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 532–546 (2018)
23. Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., Qu, R.: A survey of deep learning-based object detection. IEEE access 7, 128837–128868 (2019)
24. Kang, D., Ma, Z., Chan, A.B.: Beyond counting: comparisons of density maps for crowd analysis tasks—counting, detection, and tracking. IEEE Transactions on Circuits and Systems for Video Technology 29(5), 1408–1422 (2018)
25. Kumagai, S., Hotta, K., Kurita, T.: Mixture of counting cnns: Adaptive integration of cnns specialized to specific appearance for crowd counting. arXiv preprint arXiv:1703.09393 (2017)
26. Leibe, B., Seemann, E., Schiele, B.: Pedestrian detection in crowded scenes. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1, pp. 878–885. IEEE (2005)
27. Lempitsky, V., Zisserman, A.: Learning to count objects in images. Advances in neural information processing systems 23, 1324–1332 (2010)
28. Li, M., Zhang, Z., Huang, K., Tan, T.: Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In: 2008 19th international conference on pattern recognition. pp. 1–4. IEEE (2008)
29. Li, Y., Zhang, X., Chen, D.: Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1091–1100 (2018)
30. Lin, S.F., Chen, J.Y., Chao, H.X.: Estimation of number of people in crowded scenes using perspective transformation. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 31(6), 645–654 (2001)
31. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the seventh IEEE international conference on computer vision. vol. 2, pp. 1150–1157. Ieee (1999)
32. Loy, C.C., Chen, K., Gong, S., Xiang, T.: Crowd counting and profiling: Methodology and evaluation. In: Modeling, simulation and visual analysis of crowds, pp. 347–382. Springer (2013)
33. Ma, Z., Wei, X., Hong, X., Gong, Y.: Bayesian loss for crowd count estimation with point supervision. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6142–6151 (2019)

34. Marana, A., Costa, L.d.F., Lotufo, R., Velastin, S.: On the efficacy of texture analysis for crowd monitoring. In: Proceedings SIBGRAPI'98. International Symposium on Computer Graphics, Image Processing, and Vision (Cat. No. 98EX237). pp. 354–361. IEEE (1998)
35. Mundhenk, T.N., Konjevod, G., Sakla, W.A., Boakye, K.: A large contextual dataset for classification, detection and counting of cars with deep learning. In: European Conference on Computer Vision. pp. 785–800. Springer (2016)
36. Ojala, T., Pietikäinen, M., Mäenpää, T.: Gray scale and rotation invariant texture classification with local binary patterns. In: European Conference on Computer Vision. pp. 404–420. Springer (2000)
37. Onoro-Rubio, D., López-Sastre, R.J.: Towards perspective-free object counting with deep learning. In: European Conference on Computer Vision. pp. 615–629. Springer (2016)
38. Paragios, N., Ramesh, V.: A mrf-based approach for real-time subway monitoring. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. vol. 1, pp. I–I. IEEE (2001)
39. Pham, V.Q., Kozakaya, T., Yamaguchi, O., Okada, R.: Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3253–3261 (2015)
40. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
41. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems 28, 91–99 (2015)
42. Ryan, D., Denman, S., Fookes, C., Sridharan, S.: Crowd counting using multiple local features. In: 2009 Digital Image Computing: Techniques and Applications. pp. 81–88. IEEE (2009)
43. Sabzmeydani, P., Mori, G.: Detecting pedestrians by learning shapelet features. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2007)
44. Saleh, S.A.M., Suandi, S.A., Ibrahim, H.: Recent survey on crowd density estimation and counting for visual surveillance. Engineering Applications of Artificial Intelligence 41, 103–114 (2015)
45. Shang, C., Ai, H., Bai, B.: End-to-end crowd counting via joint learning local and global count. In: 2016 IEEE International Conference on Image Processing (ICIP). pp. 1215–1219. IEEE (2016)
46. Sheng, B., Shen, C., Lin, G., Li, J., Yang, W., Sun, C.: Crowd counting via weighted vlad on a dense attribute feature map. IEEE Transactions on Circuits and Systems for Video Technology 28(8), 1788–1797 (2016)
47. Sindagi, V.A., Patel, V.M.: Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). pp. 1–6. IEEE (2017)
48. Sindagi, V.A., Patel, V.M.: A survey of recent advances in cnn-based single image crowd counting and density estimation. Pattern Recognition Letters 107, 3–16 (2018)
49. Sooraj, P., Kollerathu, V., Sudhakaran, V.: Real-time traffic counter using mobile devices. Journal of Big Data Analytics in Transportation 3(2), 109–118 (2021)
50. Tian, M., Guo, H., Chen, H., Wang, Q., Long, C., Ma, Y.: Automated pig counting using deep learning. Computers and Electronics in Agriculture 163, 104840 (2019)
51. Tian, Y., Sigal, L., Badino, H., De la Torre, F., Liu, Y.: Latent gaussian mixture regression for human pose estimation. In: Asian Conference on Computer Vision. pp. 679–690. Springer (2010)
52. Tseng, C.H., Kuo, Y.F.: Detecting and counting harvested fish and identifying fish types in electronic monitoring system videos using deep convolutional neural networks. ICES Journal of Marine Science 77(4), 1367–1378 (2020)
53. Tuzel, O., Porikli, F., Meer, P.: Pedestrian detection via classification on riemannian manifolds. IEEE transactions on pattern analysis and machine intelligence 30(10), 1713–1727 (2008)

54. Viola, P., Jones, M.J.: Robust real-time face detection. International journal of computer vision 57(2), 137–154 (2004)
55. Viola, P., Jones, M.J., Snow, D.: Detecting pedestrians using patterns of motion and appearance. International Journal of Computer Vision 63(2), 153–161 (2005)
56. Walach, E., Wolf, L.: Learning to count with cnn boosting. In: European conference on computer vision. pp. 660–676. Springer (2016)
57. Wang, B., Liu, H., Samaras, D., Nguyen, M.H.: Distribution matching for crowd counting. Advances in Neural Information Processing Systems 33 (2020)
58. Wang, C., Zhang, H., Yang, L., Liu, S., Cao, X.: Deep people counting in extremely dense crowds. In: Proceedings of the 23rd ACM international conference on Multimedia. pp. 1299–1302 (2015)
59. Wang, Q., Breckon, T.P.: Segmentation guided attention network for crowd counting via curriculum learning. arXiv preprint arXiv:1911.07990 (2019)
60. Wang, Y., Ma, Z., Wei, X., Zheng, S., Wang, Y., Hong, X.: ECCNAS: Efficient crowd counting neural architecture search. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 18(1s), 1–19 (2022)
61. Wang, Y., Zou, Y.: Fast visual object counting via example-based density estimation. In: 2016 IEEE International Conference on Image Processing (ICIP). pp. 3653–3657. IEEE (2016)
62. Wu, B., Nevatia, R.: Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. International Journal of Computer Vision 75(2), 247–266 (2007)
63. Wu, B., Nevatia, R.: Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. vol. 1, pp. 90–97. IEEE (2005)
64. Xu, B., Qiu, G.: Crowd density estimation based on rich features and random projection forest. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1–8. IEEE (2016)
65. Zhan, B., Monekosso, D.N., Remagnino, P., Velastin, S.A., Xu, L.Q.: Crowd analysis: a survey. Machine Vision and Applications 19(5-6), 345–357 (2008)
66. Zhang, C., Li, H., Wang, X., Yang, X.: Cross-scene crowd counting via deep convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 833–841 (2015)
67. Zhang, S., Li, H., Kong, W.: A cross-modal fusion based approach with scale-aware deep representation for rgb-d crowd counting and density estimation. Expert Systems with Applications 180, 115071 (2021), https://www.sciencedirect.com/science/article/pii/S0957417421005121
68. Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 589–597 (2016)
69. Zhao, T., Nevatia, R., Wu, B.: Segmentation and tracking of multiple humans in crowded environments. IEEE transactions on pattern analysis and machine intelligence 30(7), 1198–1211 (2008)
70. Zhao, Z., Li, H., Zhao, R., Wang, X.: Crossing-line crowd counting with two-phase deep neural networks. In: European Conference on Computer Vision. pp. 712–726. Springer (2016)
71. Zhu, L., Zhao, Z., Lu, C., Lin, Y., Peng, Y., Yao, T.: Dual path multi-scale fusion networks with attention for crowd counting. arXiv preprint arXiv:1902.01115 (2019)

Matija Terček holds a bachelor's degree from Computer Science and Mathematics and is obtaining his master's degree in Data Science. His main research areas are time series analysis, computer vision and lightweight convolutional neural networks.

Maša Kljun holds a bachelor's degree and master's degree from Computer Science and Mathematics. Her main research areas include predictive maintenance, time series analysis, and computer vision.

Peter Peer is a full professor at University of Ljubljana, Faculty of Computer and Information Science and holds PhD in computer and information science. As the head of the Laboratory of Computer Vision his latest research is focused mostly on biometrics with an emphasis on deep learning. He co-authored over 100 research papers in international conferences and journals.

Žiga Emeršič is a teaching assistant at University of Ljubljana, Faculty of Computer and Information Science and holds PhD in computer and information science. Within the Laboratory of Computer Vision, he is mostly dealing with biometrics with an emphasis on deep learning. He co-authored over 40 research papers in international conferences and journals.

Received: September 25, 2021; Accepted: June 05, 2022.