Convolutional Neural Networks of Density Maps for Crowd Counting

Alberto Schiabel
University of Padova
alberto.schiabel@gmail.com

Linpeng Zhang
University of Padova
zhanglinpeng1998@gmail.com

August 20, 2020

Abstract

The crowd counting problem aims to estimate the number of people within an image or a video-frame from surveillance cameras. Accurate crowd counting is a challenging problem due to scale variations and the lack of a big dataset of images labelled with the exact number of people depicted. This problem is usually solved by estimating the density map generated from the people's location annotations or by leveraging deep convolutional networks. In this paper, we propose an alternative model that combines relevant features of other models [1, 16] recently introduced in the literature. To demonstrate the effectiveness of the proposed method, we conduct extensive experiments on 2 public crowd counting datasets. Through extensive experiments, we were able to get 92.8 and 16.9 as MAE, 148.2 and 28.1 as MSE in two difficult datasets: ShanghaiTech part A and B respectively.

I. Introduction

Crowd counting is a significant topic for crowd scene analysis and video surveillance, which have gained considerable attention in recent years. Overcrowding can represent a public safety issue, as it may cause crowd crushes. Moreover, due to the recent Covid-19 pandemic, overcrowding is particularly inadvisable, due to the social-distancing rules introduced by many countries. Crowd counting can also be generalized to the biological image processing and vehicle counting fields, in which the task is to count semantic image features.

Crowd counting is a very challenging problem, because of heavy occlusions due to the surrounding environment and large scale or perspective variations in crowd images. Other major problems are the non-uniform distribution of people, complex illumination scenes, and the limited size of existing datasets for crowd counting.

II. Related Work

Many approaches have been proposed to study the difficulties of solving the crowd counting problem [9, 10, 13, 19, 23]. Here we present a compact review of the contributions related to this problem. For a more detailed survey on crowd counting, we refer the readers to [18, 28, 29].

i. Traditional approaches

Traditional methods of crowd counting can be generally divided into two categories: detection-based methods and regression-based methods.

Detection-based methods usually assume that each object of interest (i.e. a person) in a frame can be located by using a visual object detector and that the total crowd count can be obtained by accumulating each detected person. However, these methods are limited by person occlusions and background clutters.

Researchers also attempted regression-based methods to directly learn a relation from the crowd-relevant feature of image patches to the count in the region [2, 6, 21]. Crowd-relevant features may include segment-based features, structural-based features and local texture features. Pham et al. [20] observed the difficulty of learning a linear mapping and used random forest regression to learn a non-linear mapping between local patch features and density maps.

ii. CNN-based approaches

Recently, convolutional neural networks (CNNs) have been found to be effective in crowd counting.

Zhang et al. [8] proposed a CNN alternatively trained by crowd density and crowd count. Wang et al. [30] used directly a CNN-based approach to map an image frame to its corresponding crowd count value.

These approaches were initially limited to scale-relevant features and did not generalize well with scale variations on crowd frames.

Recent research improved the spatial resolution of this type of networks and reduced the number of parameters required to be learned by the CNN. Multi-column CNN (MCNN) approaches introduced by Zhang et al. [31] allows to tackle the large scale variation in crowd scenes. Sindagi et al. [26, 27] explored methods to incorporate the contextual information by learning various density levels and generate high-resolution density maps. To improve the quality of density maps, they use an adversarial loss to overcome the limitation of Euclidean loss. Li et al. [15] proposed CSRNet by combining VGG-16 [25] and dilated convolution layers to aggregate multi-scale contextual information.

III. Dataset

We used two publicly available datasets for our experiments: the Mall dataset[1] [3–5, 17] and the Shanghai Tech dataset[2][31].

i. Mall Dataset

The Mall dataset was collected at the Nanyang Technological University of Singapore from a single publicly accessible webcam. It contains 2000 video frames with over 60.000 manually labelled pedestrians. The head position of every pedestrian is also annotated in all frames. The images have size 640x480 and use the RGB color format.

ii. ShanghaiTech Dataset

The crowd dataset collected by the ShanghaiTech University contains 1198 RGB images with around 330.000 manually labelled heads. To date, it is the largest publicly available crowd counting dataset in terms of number of annotated heads. This dataset actually consists of two parts: Part A and Part B. Images in Part A have been randomly crawled from the Internet, and most of them depict a high density of people. Images in Part B are taken from busy streets in Shanghai. Both dataset parts are already split in training and validation folders: 300 and 400 images for training, 182 and 316 for validation respectively.

Unlike the Mall dataset, images vary considerably in size and conditions. The main strength of this dataset is that every image is taken from a different viewpoint.

iii. Preprocessing

We had to preprocess the Mall images to remove the top 16 pixels, which were completely black and thus didn't present any feature useful for learning. We also had to split the dataset to perform cross-validation. We used 75% of the images for training and 25% for validation.

Furthermore, we converted the ground truth provided by both datasets used for our experiments in formats readable by Python (NPY and CSV), since the crowd count and heads location points were only available as Matlab's MAT-file objects.

iv. Data Augmentation

The major drawback of crowd-counting datasets is the very limited number of labelled images. Even if we merged the two datasets used for our experiments, we would only have a total of about 3200 samples. We then decided to augment the data at our disposal by introducing random vertical and horizontal flipping, rotation by a multiple of 90 degrees, and random resizing up to half the sizes in both width and height. Adding small variation to the input data helps to prevent overfitting and usually provides better model generalization. For a review on Image Data Augmentation techniques, we refer the readers to [24].

IV. Method

First of all we tried a baseline convolutional neural network to directly regress the head-count. This simple model consists of 4 convolutional layers,

[1] http://personal.ie.cuhk.edu.hk/~ccloy/downloads_mall_dataset.html

[2] https://www.kaggle.com/tthien/shanghaitech

each one endowed with an efficient implementation of a multi-kernel pooling, the so-called *stacked pooling*[11]. This technique relies on the fact that scale-invariance is a main feature of crowd counting, hence pooling with multiple kernels and averaging the results can capture more local relations between pixels. However, we realized that the performances were bad even in the simpler Mall dataset: in particular, it overfits and returns a MAPE (4) of over 100%. Hence, we modelled a CNN for regression that takes images of 3-channels and of any width/height, and returns a crowd density map of the same resolution of the input image. The sum of all the pixels represents the head-count. This approach exploits all information available in the dataset, i.e. not only the head-count, but also the specific head points. Moreover, this method captures scale invariance through data augmentation (random scaling of images) rather than the computationally intensive multi-kernel pooling technique described in [11].

Our final approach, however, is mainly inspired by Cao et al's paper[1] that introduced the SANet architecture. His team had a radical idea: rather than using the actual head-count as ground truth for guiding the learning process, they used density maps of heads and counted those instead.

i. Density Maps

The paper [16] uses a convolutional neural network which predicts crowd density maps. Since in the Shanghai Dataset we have only the coordinate of the headpoints, to get a ground truth for the density map, we placed a Gaussian distribution with standard deviation of 15 pixels for each head-point to represent the ground truth. The density map is normalized in such a way that the sum of all values of the density map is equal to the ground truth of the head-count.

ii. Architecture

Our model's first 14 layers use a pre-trained VGG16 model (as suggested by Liu et al), which is then chained to an original architecture that performs upsampling via transposed convolution operations.

The main idea is that the VGG16 model downsamples an approximation of the density map of the head in the images, whereas the transposed convolution section upsamples the results of VGG16. This architecture can be compared to a convolutional autoencoder where VGG16 represents the encoder that learns how to represent a density map of the input in a latent space, and the rest of the network tries acts as a decoder.

VGG16 is a 16-layer deep convolutional neural network architecture introduced in 2014 for the ImageNet challenge classification. Some notable drawbacks of the VGG16 or VGG19 architectures are that they're slow to converge and their weight are quite large in terms of disk occupation (a pre-trained VGG16 model is about 533MB).

For all convolutional layers we use ReLU as activation function, a kernel size of 3×3 and 1×1 stride with padding. This means that our convolutional layers do not change dimensions; instead, we use the max-pooling of VGG16 for the downsampling. For each transpose convolutional layer, we use Relu as activation function and a kernel size of 2×2 with 2×2 stride. These kernels allow us to upsample the dimensions by a factor 2 at each transpose convolutional layer, without overlappings of the kernels. We point out the fact that bigger kernels may improve the performances since they may capture more local relations between pixels. However, our choices are a trade-off between resource usage and the quality of the estimation. The full architecture is represented in Figure 1.

Some authors propose to use simpler architectures for achieving similar results to VGG16 in less time, such as SqueezeNet[12] or Xception[7]. We did not experiment on them because we did not find any paper on Crowd-Counting that used them, so we would not have had metrics to compare our work.

V. Experiments

i. Affine transformations

The Shanghai Tech dataset defines the ground truth in terms not only of head-count per frame, but also in terms of (x, y) coordinates of the heads in the pictures. Due to the shortage of samples in the dataset, we performed data augmentation as previously stated. Since we transformed the images to perform data augmentation, we also needed to adapt the ground truth to the new coordinate system. Such a transformation can be interpreted as an affine transformation, i.e. a transformation that can be expressed in the form

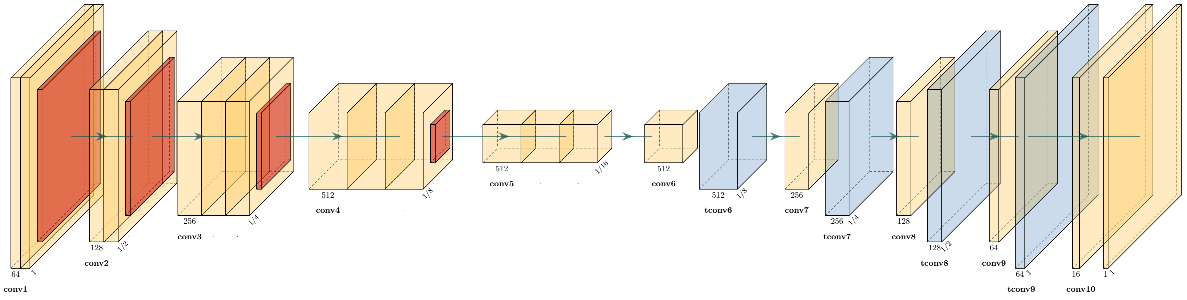

Figure 1: *Our architecture. The size of the images (1, 1/2, etc.) represent the proportion respect to the input image. The input and output layers are omitted.*

of a matrix multiplication (linear transformation) followed by a vector addition (translation)[3].

OpenCV's `getAffineTransform` returns the 2x3 `map_matrix` that transforms 3 points (x_i, y_i) in the original images to the corresponding points (x'_i, y'_i) in the resized image space such that:

$$\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \texttt{map\_matrix} \cdot \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{1}$$

We selected 3 extreme points for the transformation: the top-left point, the bottom-right and the top-right.
We needed to add the $\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$ row vector to the bottom of the diagonal matrix `map_matrix` to transform the original ground truth into the new scale using OpenCV's `perspectiveTransform` method.

ii. Training process

For training, we take into account the Euclidean Distance between ground truth and predictions, which are both density maps. This is exactly the so-called counting loss function used in [16], i.e.:

$$L_c = \frac{1}{M} \sum_{i=1}^{M} (y_i - \hat{y}_i)^2$$

where y_i and $\hat{y}_i$ represent the ground truth and the predictions of the i-th image in the batch. However, for time and space issues we use stochastic gradient descent, which may converge faster since it does more weight updates and consumes less memory. Hence in our case we have $M = 1$.
We evaluate the MAE and (root) MSE metrics on the estimated head-count which is calculated by summing every pixel of the density map, but we do not leverage them to guide training since our purpose is to predict a precise density map. We leverage a pre-trained VGG16NET model to perform transfer learning since it reduces the time required for training and increase the quality of the results, as reported in the experiment of [16].
We adapt the network to regress to high-resolution density maps, in the following way:

- we removed the fully connected layers and the last max-pooling layer to avoid further downsampling;
- to predict a crowd density map of the same resolution of the input we added 4 transposed convolutional layers to recover the spatial resolution, increasing the size by a factor 2 each; all such layers were alternated by convolutional layers;
- as last layer we put a convolutional layer of 1 unit to directly regress to the crowd density map.

All the process of data augmentation is performed online since we train one image for each batch. Moreover, the random resized input images have widths and heights which are multiple of 16: this ensures that when the image is downsampled and upsampled again (up to a factor 16), the input shape is preserved (except for the number of channels, which is only one for the output image).
We use mean absolute error (MAE), (root) mean square error (MSE) and (MAPE) as evaluation metrics. The three metrics are defined as follows:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |\hat{\mathbf{z}}_i - \mathbf{z}_i| \tag{2}$$

[3] `https://docs.opencv.org/3.4/d4/d61/tutorial_warp_affine.html`

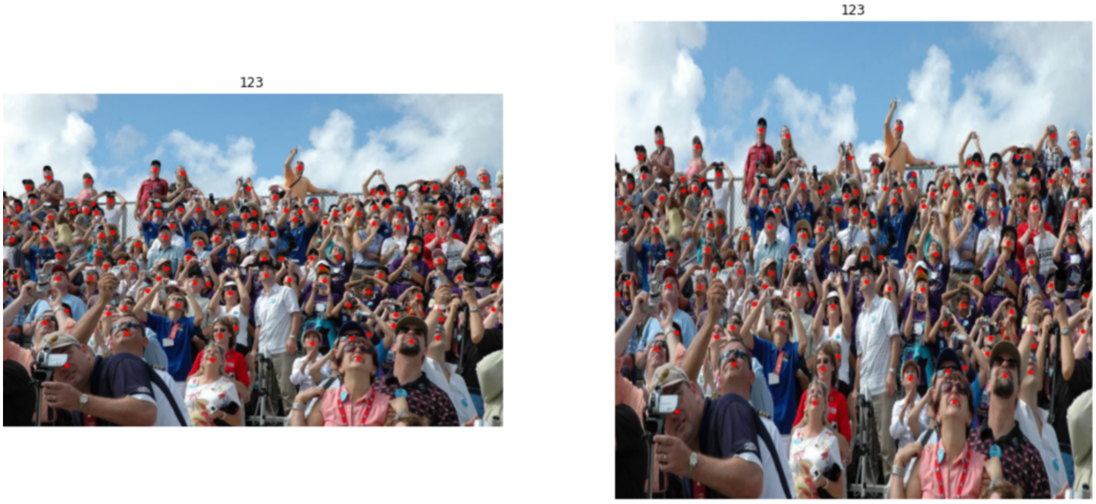

Figure 2: *Example of affine transformation of ground truth points (represented by red dots in the frames).*

$$\mathrm{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{\mathbf{z}}_i - \mathbf{z}_i)^2} \tag{3}$$

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{\mathbf{z}}_i - \mathbf{z}_i|}{\mathbf{z}_i} * 100 \tag{4}$$

where N is the number of images in the validation set, $\hat{\mathbf{z}}_i$ denotes the predicted crowd count obtained from the model for the i-th image, and $\mathbf{z}_i$ denotes the ground truth count of the i-th image.

iii. Experiments results

We run our experiments on a Google Cloud virtual machine with 8 CPUs, 52 GB of memory and 1 NVIDIA Tesla T4. The GPU has 2.560 CUDA cores, 320 Turing Tensor cores and 16GB GDDR6 of memory. The GPU memory bandwidth is 300 GB/s. Our models are implemented using Python 3.8, Tensorflow 2.2 and CUDA driver 10.1. We get the best result at the epoch 126 (i.e., passing through the entire dataset of both parts 63 times). The results of MSE and MAE metrics are shown in Table 1. As MAPE we got 28.2% and 22.80% respectively in ShanghaiTechA and ShanghaiTechB.

VI. Conclusion

i. Our contribution

Our major contribution to the Computer Vision area are: (1) a clear tutorial on how to convert ground truth head-points coordinates to a different 2-dimensional space, which can be useful for resizing large inputs or for performing data augmentation; (2) a novel CNN architecture that combines ideas from [1] and [16] and that is inspired by convolutional autoencoders to predict a high--resolution density map.

ii. What we learned

Writing this paper helped us realize how important it is to be able to quickly try out different ideas and gather feedback about experiments. As expected, model training was the activity that took us the longest time, which prevented us from trying out some ideas that came up during the experiments. We expanded our knowledge of OpenCV for performing geometrical transformations, and we learned about some inconsistencies in its API the hard way. Most importantly, we experienced what it means to read papers written by others, extract their novel ideas and apply them in different settings.

iii. Future work

Starting from some observations we made in this paper, future works may investigate on the following ideas:

- The pixel-wise Euclidean loss function we used, albeit fast and easy to compute and used in SANet, may not be sufficiently robust and suitable for Crowd Counting. It assumes

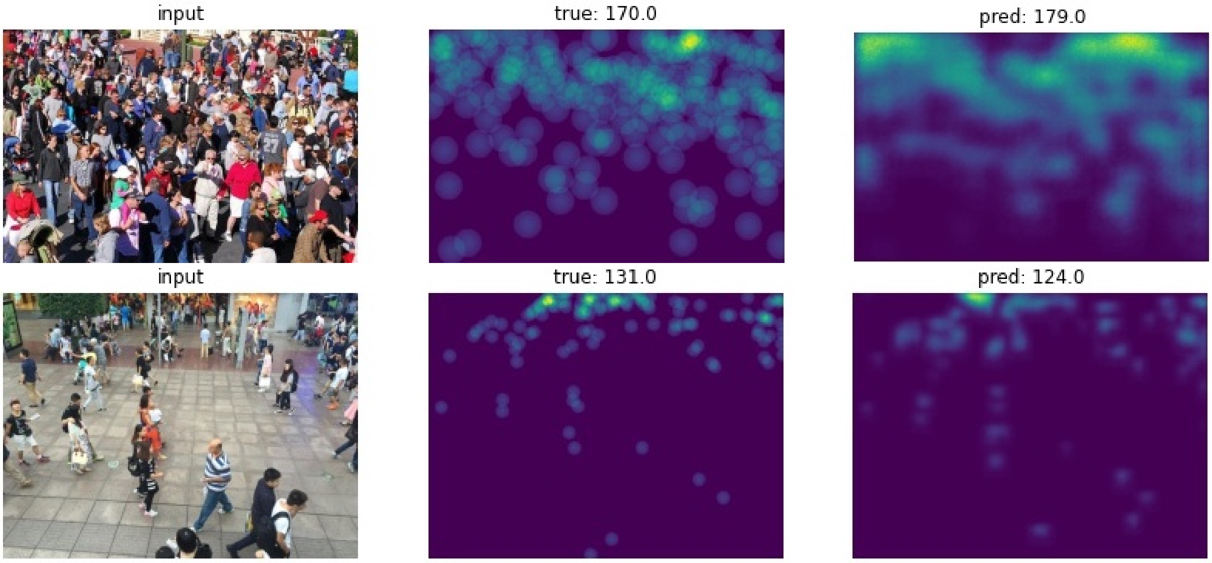

Figure 3: *Example of predictions from the validation sets of both ShanghaiTechA (upper row) and ShanghaiTechB (lower row).*

	Part A		Part B	
Method	**MAE**	**MSE**	**MAE**	**MSE**
Zhang et al. (Cross-scene CNN, 2015) [8]	181.8	277.7	32.0	49.8
Zhang et al. (Multi-column CNN, 2016) [31]	110.2	173.2	26.4	41.3
Babu Sam et al. (Switching CNN, 2017) [22]	90.4	135.0	21.6	33.4
Sindagi et al. (CP-CNN, 2017)[27]	73.6	106.4	20.1	30.1
Liu et al. (2018) [16]	72.0	106.6	14.4	23.8
Liu et al. (2018) [16]	73.6	112.0	13.7	21.4
Ours	92.8	148.2	16.9	28.1

Table 1: *MAE and MSE on ShanghaiTech dataset.*

each pixel is independent (which in general is not true) and may result in blurry images in image generation problems[14]. An enhancement probably worth trying would be a custom weighted loss function that considers both how different the learned density map is from the ground truth, and how far the predicted head-counts are from the actual count.

- The small size of the datasets is the main problem of such model architectures. Despite our novel approach for data augmentation, which uses affine transformation to adapt head-points, the images are still not different enough to obtain a good generalization. A good approach to augment the dataset could be to use the ranking loss proposed by [16]. In particular, they collect an additional dataset where no head points are available and rely on the fact that, given patches of an image, one is contained in another only if the former depicts less people than the latter.
- As noted in IV, VGGNet suffers from having plenty of parameters and being slow to train. Experiments on substituting the VGGNet part of our proposed architecture with simpler pretrained models (e.g. SqueezeNet[12], Xception[7]) could be conducted to hopefully obtain similar results in less time.

References

[1] Xinkun Cao et al. "Scale Aggregation Network for Accurate and Efficient Crowd Counting: 15th European Conference, Munich, Germany, September 814, 2018, Proceedings, Part V". In: Sept. 2018, pp. 757–773. ISBN: 978-3-030-01227-4. DOI: 10.1007/978-3-030-01228-1_45.

[2] A. B. Chan and N. Vasconcelos. "Bayesian Poisson regression for crowd counting". In: *2009 IEEE 12th International Conference on Computer Vision*. 2009, pp. 545–551.

[3] L. Chen Change et al. "Crowd Counting and Profiling: Methodology and Evaluation". In: *Modeling, Simulation and Visual Analysis of Crowds*. Vol. 11. 2013, pp. 347–382.

[4] K. Chen et al. "Cumulative Attribute Space for Age and Crowd Density Estimation". In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 2467–2474.

[5] K. Chen et al. "Feature Mining for Localised Crowd Counting". In: *Proceedings of the British Machine Vision Conference*. BMVA Press, 2012, pp. 21.1–21.11. ISBN: 1-901725-46-4. DOI: `http://dx.doi.org/10.5244/C.26.21`.

[6] Ke Chen et al. "Feature mining for localised crowd counting". In: *In BMVC*. Vol. 1. 2012.

[7] Francois Chollet. "Xception: Deep Learning with Depthwise Separable Convolutions". In: July 2017, pp. 1800–1807. DOI: `10.1109/CVPR.2017.195`.

[8] Cong Zhang et al. "Cross-scene crowd counting via deep convolutional neural networks". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 833–841.

[9] W. Ge and R. T. Collins. "Marked point processes for crowd counting". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 2913–2920.

[10] W. Ge, R. T. Collins, and R. B. Ruback. "Vision-Based Analysis of Small Groups in Pedestrian Crowds". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.5 (2012), pp. 1003–1016.

[11] Siyu Huang et al. *Stacked Pooling: Improving Crowd Counting by Boosting Scale Invariance*. 2018. arXiv: `1808.07456 [cs.CV]`.

[12] Forrest Iandola et al. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and textless1MB model size". In: Feb. 2016.

[13] H. Idrees, K. Soomro, and M. Shah. "Detecting Humans in Dense Crowds Using Locally-Consistent Scale Prior and Global Occlusion Reasoning". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.10 (2015), pp. 1986–1998.

[14] Phillip Isola et al. "Image-to-Image Translation with Conditional Adversarial Networks". In: July 2017, pp. 5967–5976. DOI: `10.1109/CVPR.2017.632`.

[15] Y. Li, X. Zhang, and D. Chen. "CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1091–1100.

[16] Xialei Liu, Joost van de Weijer, and Andrew D. Bagdanov. "Leveraging Unlabeled Data for Crowd Counting by Learning to Rank". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[17] C. C. Loy, S. Gong, and T. Xiang. "From Semi-supervised to Transfer Counting of Crowds". In: *2013 IEEE International Conference on Computer Vision*. 2013, pp. 2256–2263.

[18] Chen Change Loy et al. *Crowd Counting and Profiling: Methodology and Evaluation*.

[19] D. Oñoro-Rubio and R.J. López-Sastre. "Detecting Humans in Dense Crowds Using Locally-Consistent Scale Prior and Global Occlusion Reasoning". In: *Lecture Notes in Computer Science* 9911 (2016).

[20] V. Pham et al. "COUNT Forest: CO-Voting Uncertain Number of Targets Using Random Forest for Crowd Density Estimation". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 3253–3261.

[21] D. Ryan et al. "Crowd Counting Using Multiple Local Features". In: *2009 Digital Image Computing: Techniques and Applications*. 2009, pp. 81–88.

[22] Deepak Sam, Shiv Surya, and R. Babu. "Switching Convolutional Neural Network for Crowd Counting". In: July 2017, pp. 4031–4039. DOI: `10.1109/CVPR.2017.429`.

[23] J. Shao et al. "Crowded Scene Understanding by Deeply Learned Volumetric Slices". In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.3 (2017), pp. 613–623.

[24] Connor Shorten and Taghi Khoshgoftaar. "A survey on Image Data Augmentation for Deep Learning". In: *Journal of Big Data* 6 (Dec. 2019). DOI: 10.1186/s40537-019-0197-0.

[25] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *arXiv 1409.1556* (Sept. 2014).

[26] V. A. Sindagi and V. M. Patel. "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)*. 2017, pp. 1–6.

[27] V. A. Sindagi and V. M. Patel. "Generating High-Quality Crowd Density Maps Using Contextual Pyramid CNNs". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 1879–1888.

[28] Vishwanath A. Sindagi and Vishal M. Patel. "A survey of recent advances in CNN-based single image crowd counting and density estimation". In: *Pattern Recognition Letters* 107 (May 2018), pp. 3–16. DOI: 10.1016/j.patrec.2017.07.007. URL: https://doi.org/10.1016%5C%2Fj.patrec.2017.07.007.

[29] Gaurav Tripathi, Kuldeep Singh, and Dinesh Vishwakarma. "Convolutional neural networks for crowd behaviour analysis: a survey". In: *The Visual Computer* (Mar. 2018), pp. 1–24. DOI: 10.1007/s00371-018-1499-5.

[30] Chuan Wang et al. "Deep People Counting in Extremely Dense Crowds". In: Oct. 2015, pp. 1299–1302. DOI: 10.1145/2733373.2806337.

[31] Y. Zhang et al. "Single-Image Crowd Counting via Multi-Column Convolutional Neural Network". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 589–597.