On Density Map Estimation and Crowd Counting with Convolutional Neural Networks

Giacomo Di Prima, Giuseppe Viterbo, Marco Zenari
{giacomo.diprima, giuseppe.viterbo, marco.zenari.2}@studenti.unipd.it

Abstract

Crowd counting has become a major focus in computer vision for purposes of crowd control and public safety. The density map enables the user to get more accurate and comprehensive information about the spatial distribution considering the perspectives of the pictures, which could be critical for making correct decisions in high-risk environments. In the past, numerous approaches have been proposed, the most successful ones involve the use of CNN to reconstruct density maps of crowded images. In this paper we analyze the well establish CSRNet, proposed by Li et al. [12], challenging the use of dilation in convolutional layers as a possible solution for the task. We also investigate the relationship between the resolution of the training images and the use of dilation. Moreover we propose a new technique to create density maps with asymmetric kernel that improve the MRE score by approximately 13 *%. Finally we introduce a promising flux base approach to estimate the number of people entering a scene over temporal consecutive frames. Our code is available here [5].*

1. Introduction

Being able to accurately estimate crowds from images or videos has become an increasingly important task in computer vision for purposes of crowd control and public safety. Furthermore, knowing the number of people in a commercial area or playground could be used to determine their business capabilities, and it may also be used for economic research. Moreover, the study of crowd counting and density estimate can also be adapted in various different fields, such as psychological effects of people gathering groups [1], animal migration [13] and bacterial activity [20]. Although several techniques have been used to tackle such a complex task, researchers have since 2015 begun using CNNs to predict density maps starting from images [4][16]. This approach has become the main framework for crowd counting; however, many challenges remain to be faced. First, the geometry of the space, how much crowded the scene is and the estimation of heads sizes. Second, the different image quality will affect the effect of feature extraction. While for the image resolution used one must rely on the quality of the available datasets, on the other hand, it is of great importance to design an appropriate density mapping that is able to capture the most information possible about the spatial distribution considering the perspectives of the picture. In this paper, we introduce a variation on the procedure for generating distribution maps presented in [21]. Our approach involves considering different contributions with respect to the x and y axes to estimate the proximity between heads. This solution is successful, improving the results by $\approx 13\%$ for the MRE value 5, over the standard methodology. For evaluating our results, we introduce a metric based on the percentage error made in the total count estimate. Alongside it, we use the MAE and RMSE, which are the standard metrics found in the literature. We find these metrics to be limited, since they do not take into account how many heads are present in the image. Furthermore, we use the CSRNet model as base model [12], which is an end-to-end convolutional network, to study whether the dilated convolution strategy presented in the original paper is really effective. Our experiments show how proper initialization of model weights can compensate for the absence of dilated layers when considering the ShangaiTech dataset. We test the modified model on the DroneCrowd dataset, which presents high-resolution pictures with a fixed point of view to simulate a realistic scenario when considering emergency situations in which crowd control is crucial. The results improve greatly with respect to the one obtained with ShangaiTech. Lastly, we introduce a method to compute the fluxes of people moving through a line in space. This might be helpful in estimating the number of people inside a building in case of emergency.

2. Related Works

The use of CNN applied in the density estimation approach has outperformed the traditional methods (like regression and detection base approach), as reported by Li et al. [12]. The main idea is to take advantage of the full spatial information and train and end-to-end CNN to reconstruct not only the counts of people present in the sample, but also their spatial distribution. The creation of the density map has been proposed in Zhang et al. [21], and it is described in section 4.4. The two main proposed architecture are based on Multi-Column CNN (MCNN), proposed in Zhang et al. [21], and the Single-Column CNN architecture proposed by Li et al. [12], also known as CSRNet. CSRNet has been a standard in the field, as described by Dent et al. [2], taking advantage of transfer learning, with a front-end architecture composed of the first 10 pretrained layers of VGG-16, and a backend of convolutional layers that uses dilatation to allow for flexible aggregation of the multi-scale contextual information while keeping the same resolution. In our implementation we decided to challenge the use of the dilated convolution by introducing a more suitable parameter inizialization trought the use of the *He inizialization*, described in He et al. [6].

3. Datasets

We train our model on two different datasets: the *Shangaitech* [21] and the *DroneCrowd* [18], [17], [3]. For the evaluation we use also the *UCF-QNRF* dataset [7] and the *JHU-CROWD++* dataset [15] . In table 1 we report a summary of the used datasets.

3.1. ShangaiTech

The *Shangaitech* contains 1198 images with annotation of the positions of the centers of the heads. The total number of people is $330,165$. The dataset is divided in two parts: Part_A contains 482 images crawled from Internet while Part_B contains 716 images taken from busy streets of Shangai. Part_A images are in average more crowded than Part_B images and since the density map approach works in the assumption of high density we use the former for the training while we use Part_B only for the evaluation of the model. For the training set we sample randomly 300 images that with the data augmentation and the density map computation described in section 4 leads to a training set of a total of 1200 density maps. The test set is composed of the 736 density maps obtained applying the same processing to the remaining 184 images of the dataset. In section 5 we evaluate our model on the 184 images of the test set of Part_A and the 316 images of the test set of Part_B.

3.2. DroneCrowd

The *DroneCrowd* dataset is captured by a drone-mounted camera covering different scenarios, like campus, street, park etc. The resolution of the videos is 1920×1080 pixels and are recorded at 25 frames per seconds (FPS). The average number of people in the videos is 144.8, with a minimum of 25 and a maximum of 455. For this dataset the annotation consists in the trajectories of heads, therefore for every frame the position of the head is registered together with a number that identifies the individual. The total number of frames that we select is 9000 and our training set is composed of frames sampled with a rate of 1 every 30 from 30 video clips. The subsampling is introduced to reduce the correlation between images and the density maps are created as described in section 4 discarding the information of the label of the individual. The frames have also being augmented as described in section 4, giving a total of 1200 images in the training set. For the test set we have applied the same procedure on 12 video clips obtaining a total of 480 images. In section 5 we evaluate our model on the 300 images subsampled from the test set.

3.3. UCF-QNRF

The UCF-QNRF dataset consists of 1535 images 2902×2013 collected from Flickr, Web and Hajj footage accounting for 90%, 7% and 3% respectively. The average number of annotations is 815.4 per image, with a minimum of 49 and a maximum of $12,865$. The crowd distribution is therefore on average dense. We use this dataset for evaluation of our model in section 5, but due to the high resolution we evaluate our metrics only on a subsample of 30 randomly selected images.

3.4. JHU-CROWD++

The *JHU-CROWD++* dataset is a collection of 4372 images of resolution 900×1450 pixels. The images come from the internet, so they have different scenes. Images from different scenes increase the diversity of data. This dataset covers images in different weathers such as rain, snow, haze etc. Moreover, every annotation in the image contains more information, including the location of the head, the size of the head, corresponding occlusion level, and the blur level. This dataset is used only for evaluation in section 5 and we subsample 300 images randomly, using only the annotation of the position of the head.

4. Methods

In this section we present the specifics that will be later applied for defining the experiments we conduct.

4.1. Density maps

The desired output of our model is the density map of an image and therefore we need to produce them for the training starting from annotations of the positions of the heads. We follow initially the procedure described in [21] and then propose a slight variation to take into account the perspective of the camera with respect to the scene. If the pixel $\mathbf{x_i}$ correspond to the center of an head, then an image with N heads can be represented as

$$H(\mathbf{x}) = \sum_{i=1}^{N} \delta(\mathbf{x} - \mathbf{x_i}). \tag{1}$$

Dataset	Type	Resolution	Frames	Max count	Min Count	Average count
Shangaitech Part_A [21]	image	-	482	$3,139$	33	501.4
Shangaitech Part_B [21]	image	768×1024	716	578	9	123.6
DroneCrowd [18] [17] [3]	video	1920×1080	$33,600$	455	25	144.8
UCF-QNRF [7]	image	2902×2013	$1,535$	$12,865$	49	815.4
JHU-CROWD++ [15]	image	900×1450	4250	7286	0	262.3

Table 1. Comparison between the different datasets we use. The “-” in Resolution means that the frames have different resolutions.

CSRNet	
No Dilation	Dilation
input (unfixed-resolution color image)	
front-end (fine-tuned from VGG-16)	
back-end	
conv3-512-1	conv3-512-2
conv3-512-1	conv3-512-2
conv3-512-1	conv3-512-2
conv3-256-1	conv3-256-2
conv3-128-1	conv3-128-2
conv3-64-1	conv3-64-2
conv1-1-1	

Table 2. CSRnet configurations. All the convolutional layers use padding to mantain the previouse size and adopt the ReLu activation function. The convolutional layers’ parameter are denoted as “conv-(kernel size) - (number of filters) - (dilation rate)”. The max-pooling in the VGG-16 backend are conducted over 2 × 2 pixel windows with stride 2.

The density map of the image is given by the convolution of $H(\mathbf{x})$ with a Gaussian kernel G_σ [11]

$$D(\mathbf{x}) = H(\mathbf{x}) * G_\sigma(\mathbf{x}). \tag{2}$$

In order to consider the perspective of the image and in particular the distortion caused by the homography between the ground plane and the image plane the standard deviation σ of the Gaussian kernel is not the same at every pixel but it is, in principle, determined by the geometry of the scene that typically we do not know. A possibility is to compute it based on a density argument, assuming that if the crowd is somewhat evenly distributed then the average distance between the head and its k-nearest neighbours is a reasonable estimate of the distortion. For every head $\mathbf{x_i}$ we compute its average distance d_i with respect to the $k = 3$ nearest neighbours and assume that the standard deviation σ_i is proportional to it, $\sigma_i = \beta \cdot d_i$ where the parameter $\beta = 0.3$ is chosen empirically. In conclusion, the density map is obtained as

$$D(\mathbf{x}) = H(\mathbf{x}) * G_{\sigma_i}(\mathbf{x}), \tag{3}$$

with $\sigma_i = \beta \cdot d_i$. Our variation to this procedure is to use different standard deviations along the x-axis and the y-axis, in order to take into account the azimuth angle of the camera. Instead of the average distance from the neighbours d_i we compute the average distance along x, $d_{i,x}$ and along y, $d_{i,y}$ and use the covariance matrix

$$\begin{pmatrix} \sigma_{i,x}^2 & 0 \\ 0 & \sigma_{i,y}^2 \end{pmatrix}, \tag{4}$$

where $\sigma_{i,x} = \beta \cdot d_{i,x}$ and $\sigma_{i,y} = \beta \cdot d_{i,y}$. In figure 1 we present an image of the *Shangaitech* dataset and the density maps obtained with same standard deviation for x and y and with different standard deviation along the two axis.

4.2. Data augmentation

To augment the data we crop 4 patches of $1/8$ of the initial dimension from each image at random locations with the constraint that the patches are completely contained in the original image.

4.3. Architecture

Following the work of Li et al. [12] we use two different implementations of the CSRNet. Its main components are a front-end and a back-end. The front-end is composed of the first ten pretrained layers of VGG-16 (Simonyan et al. [14]) with only three pooling layers instead of five to suppress the detrimental effects on output accuracy caused by the pooling operation. The back-end is composed of two possible configurations, one with Gaussian layers with a dilation parameter of 2 and one with no dilation. The front-end is set up to take full advantage of transfer learning from the well establish VGG-16 model, while the back-end is designed with the goal of testing if the dilation can be used in the context of crowd counting to overcome more challenging environments, characterized by high-density, occlusion, scale changes, uneven crowd distribution, background confusion, diverse illumination and weather, perspective distortion.

4.4. Parameters Initialization

Recent deep CNNs are mostly initialized by random weights drawn from Gaussian distributions [10]. Using fixed standard deviations, very deep models (e.g.$>$ 8 convolutional layers) have difficulties to converge, as reported by the VGG team [14]. A proper initialization method should avoid reducing or magnifying exponentially the magnitudes of input signals, as described by He et al. [6]. This issue is exactly what happens following the scheme used in [12] to instantiate CSRnet, where the back-end of the network was initialized by using a gaussian distribution $\mathcal{N}(0, 0.01)$. In

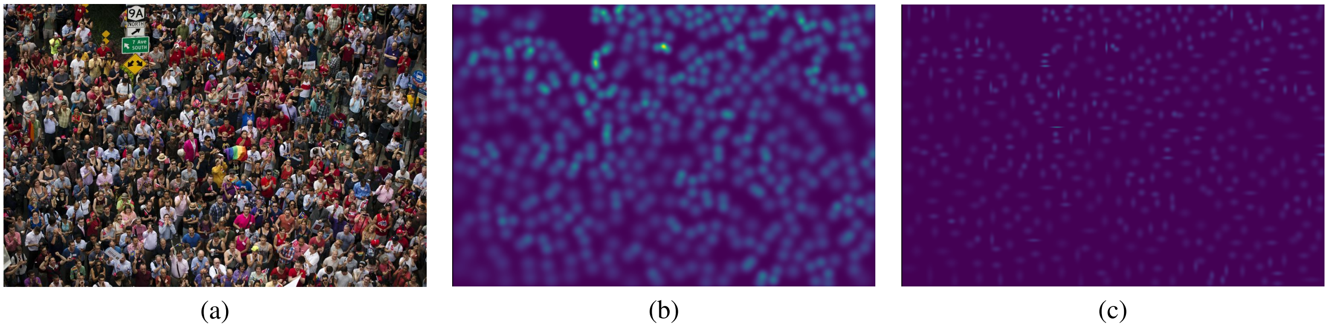

(a) (b) (c)

Figure 1. An image from *ShangaiTech* dataset (a), the corresponding density map obtained with symmetric Gaussian kernel (b) and density map obtained with different standard deviation along x and y axis (c).

fact, while the average input has an order of magnitude of 10^{-1}, the related average values taken by the output pixels is around 10^{10}. In [6] the authors propose an initialization method (*He initialization*) which takes into account the non-linearity of activation functions, such as ReLU activations. They worked out the following condition:

$$\frac{1}{2} n_l Var[\omega_l] = 1 \tag{5}$$

Where n_l is the number of connections of a response in layer l. This equation implies the following initialization scheme for ω_l weights in layer l:

$$\omega_l \sim \mathcal{N}(0, 2/n_l) \tag{6}$$

Biases are initialized to 0. By adopting this method, the mean values taken by the pixel coming from the input and output images are about the same order of magnitude.

4.5. Optimization

Due to our limited computational resources, it was not possible to train our model for as many epochs as in [12], where the authors implemented the Stochastic Gradient Descend (SGD) optimizer for 400 epochs. To allow our implementation to achieve the best possible result, we benchmark three different optimization algorithms: SGD [8], Adam [9], and Adan [19]. The last algorithm has been developed to consistently improve the model training speed across deep networks by reformulating the vanilla Nesterov acceleration to develop a new Nesterov momentum estimation (NME) method, which avoids the extra overhead of computing gradient at the extrapolation point. Then Adan adopts NME to estimate the gradient's first- and second-order moments in adaptive gradient algorithms for convergence acceleration.

5. Experiments

To evaluate the results of our experiments we use as metrics the mean absolute error (MAE):

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |counts_i - output_i| \tag{7}$$

and the root mean squared error (RMSE):

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (counts_i - output_i)^2} \tag{8}$$

Where N is the total number of images, $counts$ corresponds to the ground truth total number of people in the picture and $output$ the one computed from the model's density maps. Alongside the MAE and RMSE, which are standards in the field, we decide to use a metric which compute the average percentage counting error made by the model with respect to the ground truth. The mean relative error (MRE) is defined as follow:

$$MRE = \frac{1}{N} \sum_{i=1}^{N} \frac{|counts_i - output_i|}{counts_i} \tag{9}$$

and takes into account how the miscounting relates to how much crowded the images considered are.

5.1. Optimizer Selection

We train CSRnet with dilation and *He initialization* for 100 epochs, using the 3 different optimizer previously described and the ShangaiTech Part_A training set. The results are reported in table 3, where the metrics are evaluated over the test set of ShangaiTech Part_A. Note that the optimizer yielding the lowest metrics values is Adam. We try to further improve the result by training the best model thus obtained for 50 more epochs using SGD. As can be seen in table 3 this procedure yields better results across all metrics. Due to our computational limitations, we carry out our experiments using models trained for 100 epochs using the Adam algorithm. For all the experiment the learning rate is $l_r = 10^{-6}$.

Algorithms	MAE	RMSE	MRE
SGD	243.5	317.0	88.6%
Adam	113.6	180.8	38.0%
Adan	165.6	244.7	53.9%
Adam+SGD	**109.3**	**174.9**	**36.3%**

Table 3. Algorithm selection: each training lasted 100 epochs, except for Adam+SGD which lasted $100 + 50$ epochs, respectively.

5.2. Dilation in Training

We report in table 4 the metrics for the two training dataset, ShangaiTech Part_A and DroneCrowd, and the two configurations for CSRNet in order to evaluate if there is a meaningful improvement in the use of dilation in the convolutional layers. These results show that the use of dilation in the analysis has improved the performance on the DroneCrowd dataset but it has lowered the performances on ShangaiTech Part_A, suggesting that for highly challenging environments it is better to not use dilation. This result dose not agree with Li et al. [12], but we also did not manage to achieve their value of MAE and RMSE. We are probably limited by the lower number of epochs in the training.

5.3. New Density map approach

In table 5 we report the results of using our implementation of density maps compared to the standard ones on the ShangaiTech Part_A dataset. From these results we can confirm that using our implementation of the density map can better capture the complex distortion introduced by the various camera angles. We use data augmentation as described in 4.2.

5.4. Resolution in Training

In order to test if the resolution could be considered a limiting factor for the training procedure, we test if training the CSRNet implementations on only the High Resolution images of ShangaiTech Part_A has any meaningful influence on the training. The results are reported in table 6. For the training, we subsample only the images with a resolution higher than (900 × 600) pixels. We use data augmentation as described in 4.2. The performances are degraded by training non-dilated model only with the high resolution images, but they are improved when training the dilated model. From this we can conclude that dilated networks are negatively influenced by lower resolution images.

5.5. Model Robustness

We decided to test the robustness of the models trained on two different level of environmental challenges: ShangaiTech Part_A dataset highly challenging, especially due to the various camera angles, and the DroneCrwod dataset, less challenging thanks to the fixed camera angle. The results are reported in the table 7, that shows the performances of trained models over all the datasets presented in section 3. It is clear that the robustness of these model is quite bad in both cases, but training on a low level of environmental challenges seems to be, on average, more performant.

5.6. Flux of density to compute people back-and-forth

One possible application of our model is to compute the number of people entering or exiting from a particular place, *e.g.* a building or a street, by computing the flux of the density maps obtained with a camera that frames the gateway of such places. Taking a frame of the scene with a given rate, for example 1 frame per second, we can use our model to predict the density maps $D(x, y, t)$, where x and y are the location of the pixels and t is the time at which the frame is taken. We can compute the current as minus the gradients of the densities

$$\mathbf{J}(x, y, t) = -\left(\frac{\partial D}{\partial x}, \frac{\partial D}{\partial y}\right), \tag{10}$$

that in the discrete domains of images can be computed applying derivative filters along x and y axis. Then if we are interested in the number of people entering into the scene from the boundaries of the image or from a part of it we can compute the integral over time of the flux

$$\Phi_A(t) = \int_A \mathbf{J}(x, y, t) \cdot d\mathbf{A}, \tag{11}$$

where A is the region of interest and $d\mathbf{A}$ is the unitary vector normal to A exiting form the image. The number of people entering from the bound region A in a time interval Δt is given by

$$N_{A,\Delta t} = \int_{\Delta t} dt\, \Phi_A(t), \tag{12}$$

where the integrals need to be replaced by summations in the discrete domain of images. We applied this idea to a video of the *DroneCrowd* dataset whose scene is an horizontal street as in figure 2. Annotating the video we have estimated that the 9 people have exited from the scene from the right bound and 8 people have entered, with a net change of people in the scene of $N = -1$ in a time interval of 12 s. Our estimation of N given by the computation of equation 12 with discrete density maps taken at $t = 0, 1, 2, ..., 12$ s is equal to $N_{ext} = -1.02$. Due to the lack of statistics we cannot conclude about the robustness of this result and we leave such study for future works.

	Dilation			No Dilation		
Dataset	MAE	RMSE	MRE	MAE	RMSE	MRE
ShangaiTech Part_A	113.6	180.8	38.0%	**104.2**	**154.2**	**34.8 %**
DroneCrowd	**17.03**	**20.24**	**15.9%**	23.28	26.54	21.2%

Table 4. Evaluation Metrics: the datasets have been used for both training and testing.

	Dilation			No Dilation		
Density map	MAE	RMSE	MRE	MAE	RMSE	MRE
Standard	113.6	180.8	38.0%	104.2	154.2	34.8 %
New	**89.2**	**127.7**	**25.4%**	**103.2**	**149.9**	**33.0 %**

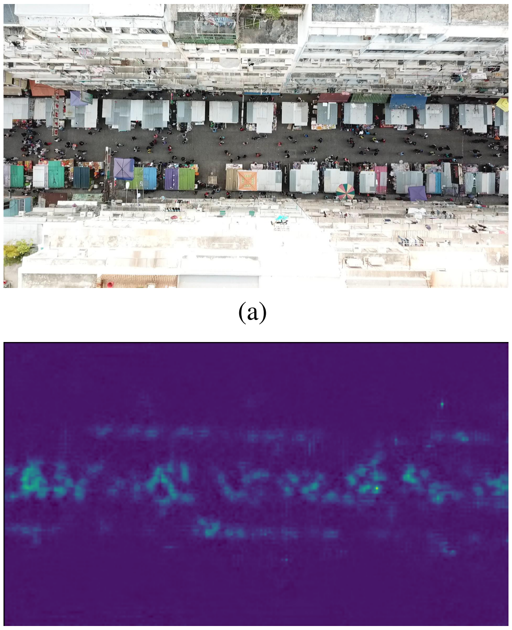
Table 5. New density performance for the ShangaiTech Part_A dataset. The Standard implementation row is equivalent to the first row of table 4.

	Dilation			No Dilation		
Resolution	MAE	RMSE	MRE	MAE	RMSE	MRE
All	113.6	180.8	38.0%	**104.2**	**154.2**	**34.8 %**
HighRes	**93.73**	**131.94**	**30.0 %**	138.36	198.68	48.3%

Table 6. Performances in training depending on the resolutions of the image for the ShangaiTech Part_A dataset. The resolution All is equivalent to the first row of table 4.

		Dilation			No Dilation		
Dataset training	Dataset evaluation	MAE	RMSE	MRE	MAE	RMSE	MRE
ShangaiTech Part_A	DroneCrowd	221.58	390.29	233.0 %	233.90	338.94	239.6%
ShangaiTech Part_A	JHU-CROWD++	297.21	643.53	342.7 %	330.97	681.13	367.7%
ShangaiTech Part_A	ShangaiTech Part_B	44.90	66.21	72.4%	59.61	76.78	93.9%
ShangaiTech Part_A	UCF-QNRF	773.03	1553.05	516.1%	755.73	1462.11	480.6 %
DroneCrowd	ShangaiTech Part_A	386.89	508.82	86.9 %	371.15	488.36	83.1%
DroneCrowd	JHU-CROWD++	258.15	699.28	79.3%	255.34	693.62	79.9%
DroneCrowd	ShangaiTech Part_B	110.19	144.13	83.8 %	102.19	137.43	74.0 %
DroneCrowd	UCF-QNRF	310.51	434.18	81.1%	293.67	426.31	84.6%

Table 7. Evaluation Metrics: in this table are reported the performances of the two implementations for CSRNet. The evaluation has been performed on > 300 samples of the original datasets, with the exception of UCF-QNRF, for which only 30 samples are used due to the computational cost of evaluating its high resolution images.



(a)

(b)

Figure 2. An image from *DroneCrowd* dataset (a), and the corresponding density map predicted with our model (b).

6. Conclusions

The adoption of dilation in convolutional layers seems to be less straightforward than how it is described in [12]. We conclude that it is a parameter to fine tune depending on the dataset and especially on the resolution of the available images. The procedure we introduce for generating density maps yields consistently better scores with respect to the one proposed in [16]. Taking into account independently x and y contributions to the distortion of the images appear to be a winning strategy. Regarding the robustness of these models, we can conclude that training on less challenging datasets, like the DroneCrowd, let the model capture better the representation of the head's positions. In all cases we report low metrics scores, that can probably be also imputed to the length of our training process and the low variability of our training datasets. Our fluxes based approach to determine the variation of the number of people in the scene shows promising results that may be inspected and broaden in the future, together with expanding the applications of our new density map definition.

References

[1] Adrian F Aveni. The not-so-lonely crowd: Friendship groups in collective behavior. *Sociometry*, pages 96–99, 1977.

[2] Lijia Deng, Qinghua Zhou, Shuihua Wang, Juan Manuel Górriz, and Yudong Zhang. Deep learning in crowd counting: A survey. *CAAI Transactions on Intelligence Technology*, 2023.

[3] Dawei Du, Longyin Wen, Pengfei Zhu, Heng Fan, Qinghua Hu, Haibin Ling, Mubarak Shah, and Junwen Pan. Visdrone-cc2020: The vision meets drone crowd counting challenge results. In *ECCVW*, volume 12538, pages 675–691, 2020.

[4] Min Fu, Pei Xu, Xudong Li, Qihe Liu, Mao Ye, and Ce Zhu. Fast crowd density estimation with convolutional neural networks. *Engineering Applications of Artificial Intelligence*, 43:81–88, 2015.

[5] Marco Zenari Giacomo Di Prima, Giuseppe Viterbo. Crowd counting. `https://github.com/MarcoZenari/crowd_counting.git`, 2024.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[7] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–546, 2018.

[8] Jack Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, pages 462–466, 1952.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[11] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. *Advances in neural information processing systems*, 23, 2010.

[12] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1091–1100, 2018.

[13] Julia K Parrish and Leah Edelstein-Keshet. Complexity, pattern, and evolutionary trade-offs in animal aggregation. *Science*, 284(5411):99–101, 1999.

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[15] Vishwanath A Sindagi, Rajeev Yasarla, and Vishal M Patel. Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method. *Technical Report*, 2020.

[16] Chuan Wang, Hua Zhang, Liang Yang, Si Liu, and Xiaochun Cao. Deep people counting in extremely dense crowds. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1299–1302, 2015.

[17] Longyin Wen, Dawei Du, Pengfei Zhu, Qinghua Hu, Qilong Wang, Liefeng Bo, and Siwei Lyu. Detection, tracking, and counting meets drones in crowds: A benchmark. In *CoRR*, 2019.

[18] Longyin Wen, Dawei Du, Pengfei Zhu, Qinghua Hu, Qilong Wang, Liefeng Bo, and Siwei Lyu. Drone-based joint density map estimation, localization and tracking with space-time multi-scale attention network. *CoRR*, abs/1912.01811, 2019.

[19] Xingyu Xie, Pan Zhou, Huan Li, Zhouchen Lin, and Shuicheng Yan. Adan: Adaptive nesterov momentum algorithm for faster optimizing deep models. *arXiv preprint arXiv:2208.06677*, 2022.

[20] He-Peng Zhang, Avraham Be'er, E-L Florin, and Harry L Swinney. Collective motion and density fluctuations in bacterial colonies. *Proceedings of the National Academy of Sciences*, 107(31):13626–13630, 2010.

[21] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 589–597, 2016.