Understanding the impact of mistakes on background regions in crowd counting

Davide Modolo, Bing Shuai, Rahul Rama Varior, Joseph Tighe
Amazon, AWS Rekognition
dmodolo,bshuai,-,tighej@amazon.com

Abstract

In crowd counting we often observe wrong predictions on image regions not containing any person. But how often do these mistakes happen and how much do they affect the overall performance? In this paper we analyze this problem in depth and present an extensive analysis on five of the most important crowd counting datasets. We present this analysis in two parts. First, we quantify the number of mistakes made. Our results show that (i) mistakes on background are substantial and they are responsible for 18-49% of the total error, (ii) models do not generalize well to different kinds of backgrounds and perform poorly on completely background images, and (iii) models make many more mistakes than those captured by the standard Mean Absolute Error (MAE) metric, as counting on background compensates substantially for misses on foreground. And second, we quantify the performance change gained by helping the model better deal with this problem. We enrich a popular crowd counting network with a segmentation branch trained to suppress background predictions. This simple addition (i) reduces background error by 10-83%, (ii) reduces foreground error by up to 26% and (iii) improves overall crowd counting performance up to 20%. When compared against the literature, this simple technique achieves very competitive results on all datasets, showing the importance of tackling the background problem.

1. Introduction

Crowd counting has attracted a lot of attention in the last few years, thanks to its applications in real-world use cases. Despite recent successes, it remains a difficult task, as models need to work well across different scenarios, from dense crowds to sparse scenes, and on any person, independently on what they are wearing or how they appear. One of the most important challenges is the problem of *scale*, which causes people on the far end of the image to appear much smaller compared to those closer to the camera. Most recent works [2, 3, 12, 14, 16–18, 20, 23–25, 28–30, 35–37, 39, 40] tackle this problem by proposing new models that attempt to achieve invariance to scale variations.

Figure 1: *Crowd counting networks output an important amount of wrong predictions on regions not containing any person, especially when these resemble crowds (e.g., foliage, roudish objects, stones, logos, etc.).*

Instead, we investigate an orthogonal problem: errors crowd counting networks make on image regions that contain no people (i.e., the background). While this problem is evident in the density predictions produced by state-of-the-art crowd counting networks (fig. 1), it remains unexplored in the literature. Only a few previous works [1, 7, 9, 27, 31, 41] have suggested that crowd counting models should be aided to attend to foreground regions only. Here we go a step further and perform an extensive quantitative evaluation that addresses the following questions: *"how much do mistakes on background affect crowd counting performance?"* In order to answer it, we experiment with five of the most popular crowd counting datasets: Shanghai Tech (Part A & B) [40], WorldExpo'10 [38], UCF-QNRF [10] and GCC [33]. Additionally, we also experiment on ADE20k [42], a semantic segmentation dataset from which we remove people and use as 100% background.

In the first part of this paper we focus on understanding how many mistakes are actually made on background regions. As the concept of background is undefined for crowd counting (i.e., each person is annotated solely with a 2D point), in this work we define background as a function of a person's head size, which we estimate automatically similar to [30]. In detail, we first enlarge each head to twice its size and then set all the pixels inside these areas as foreground and everything outside as background. Despite these favorable conditions that relax the foreground considerably, our results show that background mistakes are responsible for

18-49% of the total crowd counting error (depending on the dataset), On the most challenging datasets (ShanghaiTechA and UCF-QNRF), mistakes on background are almost as frequent as those on foreground (roughly 1 for every 2). Moreover, our analysis also shows that models do not generalize well to different kinds of background. For example, a model trained on ShanghaiTechB that achieves a MAE of 5.0 on its background, achieves a much larger MAE of 18.5 on a dataset not containing any person instance (ADE20k). Finally, by experimenting on background and foreground independently we show that the standard crowd counting MAE computed on the full image hides a lot of mistakes, as wrong predictions on the background are used to compensate for under-predictions on the foreground. Importantly, this difference is substantial and we hope that our results will encourage the community to report MAE on background and foreground independently (e.g., the aforementioned ShanghaiTechB model that achieves a MAE of 9.1 on whole images, achieves an MAE of 5.0 on the background and a MAE of 10.7 on the foreground: $5 + 10.7 \gg 9.1$).

In the second part on this work, we investigate how crowd counting performance changes when the network learns to tackle this problem. We propose to enhance a classic crowd counting network with a simple foreground segmentation branch used to suppress background mistakes. In a thorough analysis we show that this addition brings many benefits: (i) it reduces errors on background regions by 10-83% on all datasets; (ii) it improves predictions on foreground by up to 26%, and (iii) it increases crowd counting performance by up to 20%. Interestingly, these improvements enable such a simple approach to achieve performance on par with the state-of-the-art methods, which use much more complex architectures. This shows the importance of addressing the background problem.

We outline the paper as follow: in sec. 2 we summarize related works; in sec. 3 we present our first contribution: an in-depth analysis on the impact of errors on background regions in crowd counting; in sec. 4 we present our second contribution: an analysis on how teaching a crowd counting model about the background changes its performance; finally, in sec. 5 we present our conclusions.

2. Related work

Crowd counting. Approaches in the literature can be categorized into two high level groups: counting-by-detection [15, 21, 32, 34] and counting-by-regression [1–6, 12–14, 17, 19, 20, 22–25, 27–30, 35–37, 39, 40]. The former group employs person/head detectors to localize and count all the instances in an image, while the latter regresses a feature representation of the image into a count number [4, 5, 22] or a density map [1–3, 6, 12–14, 17, 19, 20, 23, 24, 27–30, 35, 37, 39, 40]. Most of the recent approaches belong to the latter group and focus on learning new and more accurate image representations.

Challenges in crowd counting. One of the most prominent challenges is the issue of *scale*, which causes people on the far end of the image to appear smaller than those closer to the camera. This problem originates from the perspective effect and most of the recent works in the literature have addressed this with new multi-scale models [2, 3, 12, 14, 17, 20, 23–25, 28–30, 35–37, 39, 40]. Some works adopted multi-column architectures [2, 12, 17, 20, 23, 24, 28, 40], where each column is dedicated to a specific scale; others [3, 14, 29, 30, 35, 37, 39] proposed single-column architectures that learn multi-scale features within the network itself (e.g., by combining feature maps from different layers [29, 30, 37, 39]); finally, [25, 36] proposed perspective-aware networks. On a different direction, [6, 13, 19] focused at improving *spatial awareness* in counting. Differently from all these works, we explore yet another important problem: crowd counting networks wrongly *count on background* regions not containing any person's instance. While this issue was briefly mentioned in [1, 27], to the best of our knowledge, we are the first to quantitatively evaluate the magnitude of this problem and present an extensive analysis on how this affects crowd counting performance.

Reducing errors on background. Only a few methods in the literature [1, 7, 9, 27, 31, 41] have, to a certain extent, tried to address this problem by using semantic segmentation branches trained to separate the foreground from the background. For example, Arteta et al. [1] employed a body segmentation branch, along with numerous other supervisions, like multiple point annotations from different annotators on each entity (penguin), uncertainty maps that capture the annotators agreement and depth density maps that capture the perspective change. Huang et al. [9] combined the features of a body-parts segmentation branch with those of a structured density map and used those to regress to the final count. Similarly, [7, 27, 31, 41] combined a head segmentation branch with: a density map estimator and a random high-level density classifier [7], a depth estimator and a count regressor [41], an appearance and residual branches [31] and a global density branch and per-pixel density one [27]. Note how all these methods employ many cues and auxiliary tasks in their designs, as they focus on achieving the best possible counting performance. Instead, in this work we employ a foreground/background segmentation branch only for analysis purposes, where we use it to quantify how crowd counting performance changes as the model learns about the background. We hope that our discoveries can shed some lights on this problem and inspire new research directions.

3. Wrong predictions on background regions

In this section we present the first analysis on the problem of predicting counts on regions not containing any person instance. We present an extensive analysis on five of the most popular crowd counting datasets, on which we quantify the number of mistakes popular crowd counting approaches make on these regions.

3.1. Our baseline: CSRNet+

In our analysis we experiment with the popular CSRNet architecture [14]. However, in our re-implementation of this network we made few small changes to better fit it to the task of crowd counting. More specifically, we remove the `pool3` layers of VGG16 and set the dilation rates of convolution layers in the 4^{th} and 5^{th} block to be 2 and 4 respectively. This leads to higher-resolution features maps that are key to predicting very small people covering just a few pixels. Moreover, we also adopt a sub-pixel convolutional layer [26] for upsampling the predicted density map to the original input image size. From our experiments, these small changes slightly, but consistently, improve crowd counting performance over the settings of the original CSRNet (table 1). Finally, we follow the implementation details of [14] and we use the classic method proposed by Zhang et al. [40] to generate ground truth density maps: we convolve each head ground truth point annotation with a fixed Gaussian kernel of $\sigma = 15$ pixels.

3.2. Datasets

We experiment with five of the largest and most popular crowd counting datasets: UCF-QNRF [10], Shanghai Tech (Part A & B) [40], WorldExpo'10 [38] and GCC [33]. As these datasets capture quite different scenarios, they provide the best mix of background for this analysis. Finally, in order to understand how crowd counting models perform on general background images, we also test on a large-scale semantic segmentation dataset: ADE20K [42].

- **UCF-QNRF** [10] is one the newest crowd counting dataset and it contains large diversity both in scenes, as well as in background types. It consists of 1535 images high-resolution images from Flickr, Web Search and Hajj footage. The number of people (i.e., the count) varies from 50 to 12,000 across images. In order to fit images as large as 6000×9000 pixels in memory, at inference we downsample these to a maximum side of 1920 pixels.
- **ShanghaiTech** [40] consists of two parts: A and B. *Part A* contains 482 images of mostly crowded scenes from stadiums parades and its count averages >500 people per image. *Part B* consists of 716 images of less-crowded street scenes taken from fixed cameras and counts varying from 2 to 578.
- **WorldExpo'10** [38] contains 3980 frames from 1132 video sequences. These are split into 5 scenes and we report their average performance. The dataset is commonly evaluated by masking out images with some regions of interest (ROIs) provided by the creators, which are meant to suppress both (some) background and small non-annotated people in the far end of the image. We follow this standard procedure.
- **GCC** [33] is the newest dataset and it consists of 15212 synthetic images with more than 7.5 million people. The dataset contains a large variety of computer generated scenes, from very dense to very sparse. It contains three slips: Random, Camera and Location. We limit our analysis to the last set (as it is the most challenging one), but we compare against the literature on all three.
- **ADE20k** [42] is a semantic segmentation dataset containing images picturing more diverse and challenging scenes compared to those for crowd counting. For example, these scenes range from natural to man-made and from outdoor to indoor, and they provide an excellent test use case. We evaluate on the 1468 background validation images (i.e., those that do not contain any person).

3.3. Metrics

We report our results using the standard crowd counting metrics: Mean Absolute Error (MAE) and Root Mean Squared Error (MSE). In details, given the predicted count $\mathbf{C}^p$ and ground truth count $\mathbf{C}^{gt}$:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\mathbf{C}_i^p - \mathbf{C}_i^{gt}|, \quad \text{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\mathbf{C}_i^p - \mathbf{C}_i^{gt})^2} \quad (1)$$

where N refers to the number of test images.

In order to better analyze the behavior of crowd counting models and how they deal with background regions, we report our results on three MAE/MSE adaptations, each one evaluating the error on a particular region of the image. More specifically, we evaluate on background only, foreground only and full images. We compute these as in eq. 1, but only count on specific regions:

$$\mathbf{C}_i^p = \sum_{j}^{H}\sum_{k}^{W}\mathbf{D}_i^p(j,k)\cdot\mathbf{M}_i^{gt} \quad (2)$$

where H, W indicate the spatial resolution of the image, $\mathbf{D}_i^p$ is the predicted density map and $\mathbf{M}_i^{gt}$ corresponds to a ground truth mask that specifies what region to evaluate on. The computation of $\mathbf{C}_i^{gt}$ is analogous. For **Full Image**, we set every element in $\mathbf{M}_i^{gt}$ to 1, meaning that every pixel in the image is considered in the error estimation. Note how this is the standard case used in the crowd counting literature. For **Foreground**, instead, we only set the

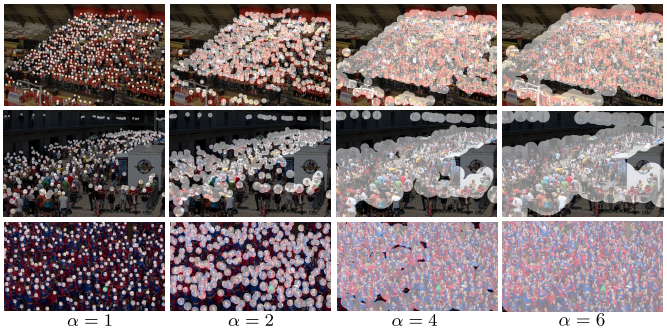

Figure 2: *Three images and three foreground masks $\mathbf{M}^{gt}$ obtained by dilating the head size d_i with α (sec. 3.4).*

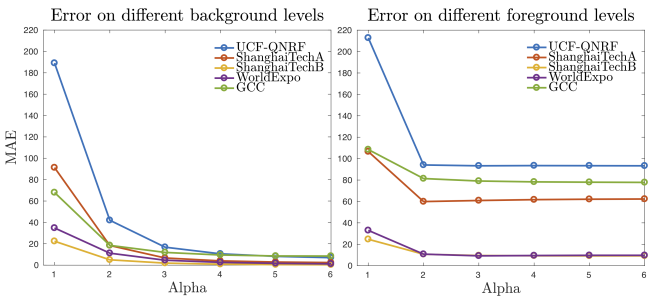

Figure 3: *Errors on background and foreground regions as a function of α, which is used to dilate each head d_i and define different background/foreground boundaries (i.e., the larger α is, the less the amount of background in an image, fig. 2 for examples).*

foreground elements in $\mathbf{M}_i^{gt}$ to 1, and the rest 0. This estimates count error on foreground regions only and it does not penalize for false positive predictions on background. Finally, **Background** has a mask complementary to Foreground (i.e., ones and zeros swapped). In the next section we explain how to estimate $\mathbf{M}_i^{gt}$.

3.4. Background analysis

In this section we present a series of experiments that investigate if and by much crowd counting models wrongly predict people on background regions.

What is background in crowd counting? In crowd counting datasets, each person is annotated only with its head point (x_i, y_i), which is not sufficient to estimate good boundaries between foreground and background and to generate accurate foreground masks for evaluation. We overcome this limitation by augmenting each point annotation with a value d_i, corresponding to the diameter of the head. We estimate this similarly to the bounding-box technique of Rama Varior et al. [30]: first, we run a head detector, then we associate its detections (of size s_i) to the annotated head points, and finally we set the size of the remaining heads (the tiny ones that the detector failed to localize) to 15 pixels, which is the common size estimate used in crowd counting. This can be summarized as $d_i = \max(s_i, 15)$.

Next, we obtain the foreground mask $\mathbf{M}_i$ by setting all the pixels inside each head blob centered at (x_i, y_i) to 1. In order to understand how the performance changes with respect to the definition of foreground, we experiment with different head blob sizes by dilating the estimated head size by a factor $\alpha = [1, \ldots, 6]$: $d'_{i,j} = d_i \cdot \alpha_j$ (fig. 2 and fig. 3).

Among these, $\alpha = 1$ is the stricter definition, as each head corresponds to foreground and any non-head region is mapped to background. Under this setting, all models surprisingly achieve a very large MAE on both background and foreground. We attribute this phenomena to three factors: (i) there is uncertainty in our estimation of s_i, (ii) there is inconsistency in the exact location of the annotated point (i.e., sometimes the point lies on the chin of a person, other times on the forehead, etc.) and, more importantly, (iii) crowd counting models are good at counting, but less accurate at localizing each individual person: they tend to output density predictions that are less peaky than the Gaussian kernels used to convolve each point during training, resulting sometimes in predictions just outside a head region.

Model	Train & Test dataset	Background % Surface	Background MAE	Foreground % Surface	Foreground MAE	Full Image MAE
CSRNet+	ShanghaiTechA	39%	18.4	61%	60.0	64.9
CSRNet+	ShanghaiTechB	76%	5.0	24%	10.7	9.1
CSRNet+	UCF-QNRF	51%	42.0	49%	94.1	95.1
CSRNet+	WorldExpo	89%	10.2	11%	11.2	8.7
CSRNet+	GCC	88%	17.7	12%	79.9	81.2
MCNN [40]	ShanghaiTechA	39%	56.9	61%	96.1	110
MCNN [40]	ShanghaiTechB	76%	18.3	24%	31.5	26.4
CSRNet [14]	ShanghaiTechA	39%	37.1	61%	68.4	66.5
CSRNet [14]	ShanghaiTechB	76%	24.2	24%	24.9	9.6
SFCN [33]	UCF-QNRF	51%	41.6	49%	114	102

Table 1: *MAE results of different models on five crowd counting datasets, split into background, foreground and aggregated over the full image. While the community has mostly focused on reducing Foreground error, the unexplored Background error is also important and worth addressing in the future.*

For all the other values of α, the performance is much more consistent: while foreground MAE increases slowly, background MAE continues to decrease as the background shrinks. From these results we can see that $\alpha = 2$ (fig. 2 mid) is a good choice to define the foreground/background boundary, as it provides a good trade-off between a too strict foreground (causing the issues mentioned above) and a too relax foreground (causing important background regions to be considered as foreground). In the remaining of the paper we evaluate on background and foreground using this definition.

Is background a problem for crowd counting? Now that we have defined what foreground and background mean

in crowd counting, we investigate how many mistakes are made in these regions with respect to the full image. We present our results in table 1, along with the percentage of background and foreground for each dataset. While not as frequent as the mistakes on foreground, the errors of CSRNet+ on background regions are still substantial: on all datasets Background MAE is responsible for around 18-49% of the total error. This is especially problematic on very crowded datasets, where the areas belonging to background and foreground are very similar (ShanghaiTechA and UCF-QNRF), meaning that the errors on background are almost as frequent as those on foreground (i.e., 1:2 and 1:2.4 when normalized by the surface area). On the much less dense datasets, results are less severe, but this is the case because ShanghaiTechB and GCC contain similar backgrounds in their train and test sets, and WorldExpo uses ROIs to suppress difficult regions (sec. 3.2).

In table 1 we also report the results we obtained by running the code and models (available online) of some popular crowd counting approaches[1]. Their results show similar behavior of those of our CSRNet+ baseline. Interestingly, CSRNet achieves a substantially higher MAE on foreground and background compared to CSRNet+, even though their full image MAEs are very similar. Upon investigatation we noticed that CSRNet is not particularly good at localizing people and it tends to output less peaky density maps (due to its lower resolution feature maps).

Observation about MAE for crowd counting. Finally, we want to highlight how MAE computed on full images is not equal to the sum of the MAEs computed over background and foreground (e.g., in first row of table 1, $64.9 \neq 60.0 + 18.4$). This is surprising, considering that the union of these two mutually-exclusive pixel sets are equivalent to that of the full image. This behavior is due to the fact that MAE computed on full images uses wrong predictions on background regions to compensate for missed predictions on foreground areas. This is an important concern, especially considering the large discrepancies reported in table 1. Going forward, we hope that these results will encourage the community to report more accurate estimates than MAE computed on the whole image, like MAE split into background and foreground, or GAME [8].

Do models generalize to different backgrounds? Here we investigate how models trained on a specific dataset generalize to different kinds of background (i.e., to other datasets). Results are presented in table 2. The best performing model (i.e., lowest background MAE) on each dataset is, except for WorldExpo, the model trained on that same dataset. This is a domain gap problem and it is substantial; for example, a very good model trained on ShanghaiTechB makes 50% more mistakes than one trained on UCF-QNRF on the background of UCF-QNRF (62.1 vs 42.0). This problem is even more evident in the results on the ADE20k dataset, which does not contain any person: the best model trained on real data (UCF-QNRF) outputs an average count of 8.4 per image, while the worse (WorldExpo) more than 45. These are equivalent to 12,000 and 66,000 people predicted in a dataset not containing any person. This poor generalization to different backgrounds is an important limitation towards applying crowd counting techniques to real world use cases.

Train \ Test	Shanghai Tech A	Shanghai Tech B	UCF QNRF	World Expo	GCC	ADE20k
ShanghaiTechA	**18.4**	7.7	57.3	6.7	143.2	27.6
ShanghaiTechB	21.3	**5.0**	62.1	9.0	19.9	18.5
UCF-QNRF	20.5	8.8	**42.0**	19.1	38.6	8.4
WorldExpo	98.8	13.5	118.1	11.2	73.6	45.1
GCC	24.9	7.9	45.2	**5.9**	**17.7**	**3.2**

Table 2: *Background MAE for CSRNet+ across datasets.*

Finally, the results suggest that the model trained on synthetic data (GCC) is, on average, the best performing model on background. However, upon investigation we observed that this model undercounts significantly on any real image, leading to good background MAE, but terrible foreground MAE. For example, it achieves and MAE of 235 on ShanghaiTeachA, of 25 on ShanghaiTechB, of 274 on UCF-QNRF and of 46 on WorldExpo, which are significantly higher then the results presented in table 1 (MAE 60, 10.7 94.1 and 11.2, respectively). These results are a bit discouraging, as they show that significant work is needed before we can use synthetic data for crowd counting.

Conclusions. Crowd counting models occasionally make mistakes on background regions and every researcher on this topic is likely aware of this behavior. However, this problem appears to be much more severe that what people may have anticipated: our analysis quantitatively showed that crowd counting models produce an important number of wrong predictions on background regions, which flactuates from 18 to 49%, depending on the dataset. Moreover, our analysis also showed why these mistakes have not been clearly captured before: the MAE metric computed on the full image hides these mistakes behind underpredictions on foreground regions, fooling us in believing that crowd counting models perform better than they actually do. Finally, our analysis also showed that crowd counting datasets do not contain enough diversity in terms of background, which lead to poor generalization when evaluated on pure background images. Given all these discoveries, we believe that wrongly predicting people on background regions is an important issue in crowd counting and we hope that these results will inspire more works to solve this problem.

[1]Note: the CSRNet models the authors released online achieve slightly better performance compared to the results they report in their paper.

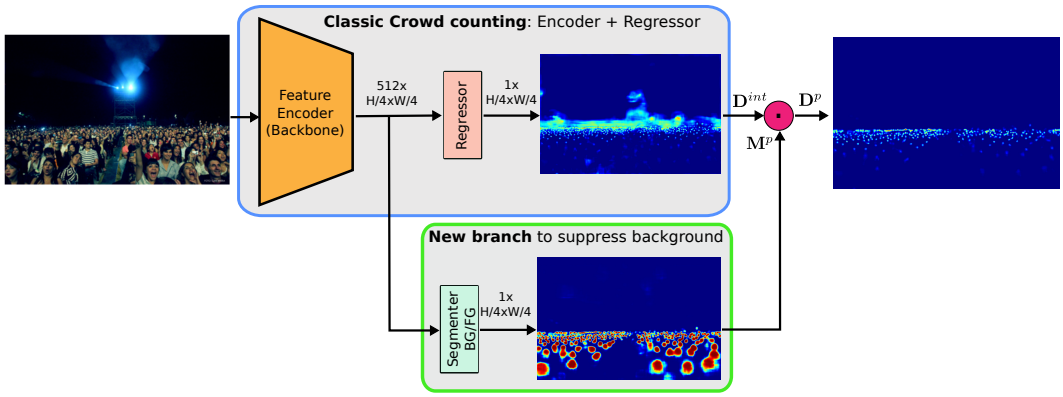

Figure 4: *In the top row we show a classic crowd counting methods that consists of a feature encoder and a count regression module. In our approach we enrich this design with a segmentation branch that is used to suppress predictions on background regions.*

4. Teaching the network about background

In this section we present the second part of our analysis, where we investigate how crowd counting performance changes when a network learns to suppress wrong counts on background regions. Towards this, we propose a simple change to the typical crowd counting network that aims at reducing background mistakes from the final density map, leading to cleaner outputs and more accurate counting. We propose to enrich the final regression block that maps the backbone's features to a density map for crowd counting, with a new head that is trained specifically to suppresses predictions on background regions (fig. 4). We model this head as a shallow foreground/background segmentation branch that has low impact in terms of computational cost with respect to the overall model. This branch outputs a mask that is used to modulate the density map outputted by the regression head. This mechanims has two benefits: while the segmentation head reliably suppresses background predictions, the regression head can now better specialize on foreground patterns and improve its counting accuracy (due to being trained on gradients from foreground pixels only, eq. 5).

Finally, note that the idea of using a segmentation branch to attend to foreground regions was first introduced by Arteta et al. [1] to count penguins and very recently also by [7, 9, 27, 31, 41] to count people (sec. 2). However, all these works used a segmentation branch within a much more complex design, as their goal was to achieve the best counting performance. Instead, we use it as part of our analysis on understanding the impact of suppressing background mistakes on the final count. Towards this, we are purposely making our model as simple and as specialized to this problem as possible. In the next paragraphs we explain how to train this simple approach.

Loss functions. Given a training image of size (W, H), we train the segmentation head with a pixel-wise binary cross entropy loss between the sigmoided predicted mask $\mathbf{M}^p$ and its corresponding ground truth $\mathbf{M}^{gt}$:

$$\mathcal{L}_{bce} = \frac{1}{HW}\sum_{i}^{H}\sum_{j}^{W} -\mathbf{M}^{gt}_{i,j}\cdot log(\mathbf{M}^{p}_{i,j}) - (1-\mathbf{M}^{gt}_{i,j})\cdot log(1-\mathbf{M}^{p}_{i,j}) \tag{3}$$

Moreover, following the literatures [2,3,12,14,17,20,23–25, 28–30, 35–37, 39, 40], we train our regression head with a pixel-wise ℓ_2 distance loss between the predicted density map $\mathbf{D}^p$ and its corresponding ground truth map $\mathbf{D}^{gt}$:

$$\mathcal{L}_{\ell_2} = \sum_{i}^{H}\sum_{j}^{W}\sqrt{(\mathbf{D}^{p}_{i,j} - \mathbf{D}^{gt}_{i,j})^2} \tag{4}$$

where the predicted density map $\mathbf{D}^p$ is obtained by modulating the intermediate density map $\mathbf{D}^{int}$ with the predicted foreground mask $\mathbf{M}^p$ as follows: $\mathbf{D}^p = \mathbf{D}^{int} \odot \mathbf{M}^p$, in which $\odot$ represents the Hadamard operation. Importantly, note how the regression module only aggregates counting contribution for foreground regions, as the segmentation head takes the responsibility for recognizing background regions. In an end-to-end fashion (fig. 4), we train our model (including the backbone) with the following dual task loss:

$$\mathcal{L} = \mathcal{L}_{\ell_2}(\mathbf{D}^p, \mathbf{D}^{gt}) + \lambda\mathcal{L}_{bce}(\mathbf{M}^p, \mathbf{M}^{gt}) \tag{5}$$

where λ regulates the importance of the segmentation loss. From these losses one can see how separating foreground and background predictions (to regressor and segmenter, respectively) not only helps reducing mistakes on background, but it also helps the regressor becoming more accurate, as it is now entirely dedicated on counting on foreground regions only.

Implementation details. We use the backbone (CSRNet+) introduced in sec. 3.1 and 3 additional fully convolutional layers for the segmentation head (an exact copy of the 3 fully convolutional layers used for regression). Moreover, we generate $\mathbf{M}^{gt}$ as explained in sec. 3.4, with $\alpha = 1$ (sec. 4.2).

Train & Test dataset	Model	Background MAE		Foreground MAE		Full Image MAE	
ShanghaiTechA	CSRNet+	18.4		60.0		64.9	
	CSRNet+ w/BS	14.9	↓↓19%	58.9	↓1.6%	62.6	↓3.5%
ShanghaiTechB	CSRNet+	5.0		10.7		9.1	
	CSRNet+ w/BS	3.2	↓↓36%	7.9	↓↓26%	7.2	↓↓20.1%
UCF-QNRF	CSRNet+	42.0		94.1		95.1	
	CSRNet+ w/BS	31.9	↓↓24%	85.5	↓9.1%	86.3	↓9.2%
WorldExpo	CSRNet+	11.2		10.2		8.7	
	CSRNet+ w/BS	10.1	↓10%	9.5	↓7%	8.1	↓6.9%
GCC	CSRNet+	17.7		79.9		81.2	
	CSRNet+ w/BS	10.1	↓↓43%	66.1	↓↓17.3%	65.6	↓↓19.2%

Table 3: *We compare CSRNet+ w/o and w/ our background suppression branch (BS) on five crowd counting datasets. Adding the background suppression branch brings many benefits: (i) errors on background reduce considerably, (ii) errors on foreground also reduce, though less and (iii) the final performance is always better.*

Model \ Train	Shanghai TechA		Shanghai TechB		UCF-QNRF		World Expo		GCC	
CSRNet+	27.6		18.5		8.4		45.1		3.2	
CSRNet+ w/BS	19.7	↓↓29%	3.1	↓↓83%	5.2	↓↓38%	36.0	↓↓20%	0.8	↓↓75%

Table 4: *We compare CSRNet+ w/o and w/ our background suppression branch (BS) on the pure background dataset ADE20k. Errors on background reduces substantially.*

4.1. Validation of our approach

In this section, we experiment with this simple approach and evaluate its impact on the task of crowd counting, especially on background regions. We compare the CSRNet+ baseline model presented in sec. 3.1 with the same CSRNet+, but enhanced with a segmentation branch. For simplicity, in the remaining of the paper we will refer to our approach as CSRNet+ w/BS (background suppression). First, we compare against the CSRNet+ results presented in table 1 and investigate if this new branch can improve its performance. These are presented in table 3.

Background mistakes decrease (Background MAE). Results validate our hypothesis that the segmentation branch can help reduce mistakes on background and show that our approach can consistently reduce these errors by an important 10-40% on all datasets, over the baseline. Importantly, this improvement generalizes well to other background types, like on the background dataset ADE20k (table 4), where MAE always decreases, from 30% to more than 80%. Finally, in the supplementary material we also report the improvement across datasets (baseline in table 2), further showing the improvement in background generalization. For example, MAE for the model trained on Shanghai Tech B and tested on UCF-QNRF drops from 62.1 to 26.2 and for the model trained on WorldExpo and tested on Shanghai Tech A from 98.8 to 47.9.

Foreground errors decrease (Foreground MAE). In sec. 4, we argued that adding the segmentation mask and using it to modulate the final output can, in theory, help the regressor better specialize on foreground (as it is lifted of the responsibility of the background) and produce more accurate predictions. Results validate this and show that in practice MAE on foreground regions always reduces, sometimes marginally (+1.6% on Shanghai Tech A), but other times substantially (+26% on Shanghai Tech B).

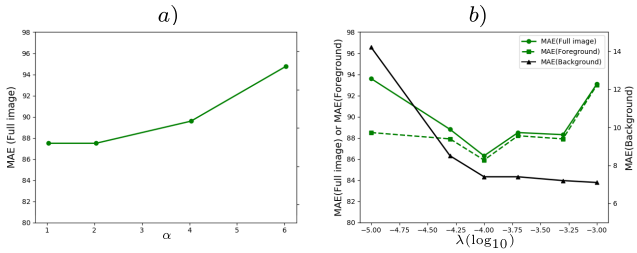

Figure 5: *Results for different values of α (a) and λ (b).*

Overall performance improves (Full Image MAE). Finally, results also show that improving both background and foreground MAEs leads to a consistent improvement in the overall image performance, up to 20% better.

4.2. Sensitivity analysis of hyper-parameters

In this section, we evaluate some of our choices for the paraneters if CSRNet+ w/BS. We experiment on the UCF-QNRF dataset only, as it is the largest and most diverse crowd counting dataset and it is a good benchmark for this study.

Lambda λ. We investigate how sensitive our model is to different values of λ in eq. 5, which regularizes the importance of the segmentation head. As shown in fig. 5b, the model performs best on full images when λ is in the range of 0.5×10^{-4} to 5×10^{-4}. This provides a good trade-off between not using the segmentation head ($\lambda = 0$) and relying on it too much (λ too large). Moreover, MAE on foreground is the lowest when λ is around 10^{-4}, while MAE on background consistently decreases as the segmentation head gets more and more importance (i.e., λ increases). These results show the importance of training a model that is well balanced for both foreground and background. In our experiments, we use λ to 10^{-4}.

Training foreground mask $\mathbf{M}^{gt}$. In our experiments we chose $\alpha = 1$. Our intuition is twofold: (i) d should be at least as large as the Gaussian kernel $\sigma = 15$ used to define GT maps (otherwise non-zero pixels' count would be wrongly assigned to background) and (ii) it should be as close as possible to the true size of the head (s_i). To verify this hypothesis, we experiment here with different values of α and train with different foreground/background definitions. Results in fig. 5a show that the best performance is indeed achieved by setting α to 1. Nevertheless, note that our model is quite robust and still achieves great performance until the point where d_i becomes too large and almost every pixel is labeled as foreground (i.e., $\alpha > 4$).

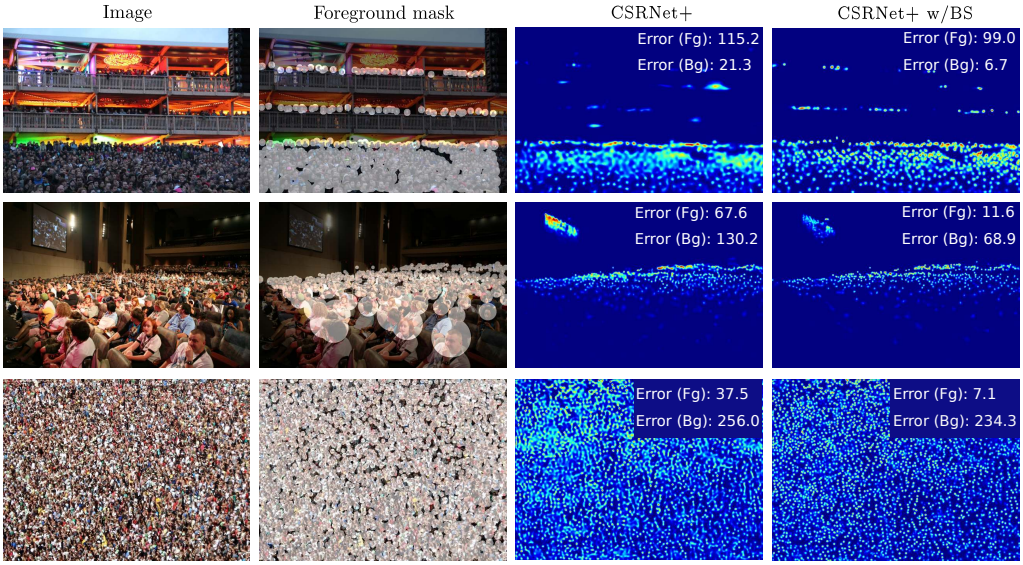

Figure 6: *Enhancing CSRet+ with the ability to suppress background (w/BS) produces more accurate density maps with less errors on background regions (Error Bg) and significantly sharper foreground estimates (Error Fg).*

			UCF-QNRF		Shanghai Tech A		Shanghai Tech B		WorldExpo	GCC (MAE)		
Method	**Venue & Year**		MAE	MSE	MAE	MSE	MAE	MSE	Avg MAE	Rand	Cam	Loc
MCNN [40]	CVPR	2016	277.0	426.0	110.2	173.2	26.4	41.3	11.6	100.9	110.0	154.8
SwitchCNN [24]	CVPR	2017	228.0	445.0	90.4	135.0	21.6	33.4	9.4	-	-	-
CP-CNN [28]	ICCV	2017	-	-	73.6	106.4	20.1	30.1	8.9	-	-	-
CSRNet [14]	CVPR	2018	-	-	68.2	115.0	10.6	16.0	8.6	38.2	61.1	92.2
CL-CNN [10]	ECCV	2018	132.0	191.0	-	-	-	-	-	-	-	-
CSRNet+PACNN [25]	CVPR	2019	-	-	62.4	102.0	7.6	11.8	-	-	-	-
SFCN [33]	CVPR	2019	102.0	171.4	64.8	107.5	7.6	13.0	9.4	**36.2**	**56.0**	**89.3**
BL [19]	ICCV	2019	88.7	154.8	62.8	101.8	7.7	12.7	-	-	-	-
PGCNet [36]	ICCV	2019	-	-	57.0	**86.0**	8.8	13.7	8.1	-	-	-
SANet+SPANet [6]	ICCV	2019	-	-	59.4	92.5	**6.5**	**9.9**	7.7	-	-	-
ASNet [11]	CVPR	2020	91.6	159.7	57.8	90.1	-	-	6.6	-	-	-
AMRNet [18]	ECCV	2020	**86.6**	152.2	61.6	98.4	7.0	11.0	-	-	-	-
LibraNet [16]	ECCV	2020	88.1	**143.7**	**55.9**	97.1	7.3	11.3	-	-	-	-
CSRNet+ w/BS	-		**86.3**	**153.1**	62.6	103.3	7.2	11.5	8.1	**30.2**	**39.3**	**65.6**
CSRNet+ w/BS (pre-trained)	-		-	-	**58.3**	**100.1**	**6.7**	**10.7**	**7.9**	32.6	40.2	69.8

Table 5: *Quantitative results of CSRNet+ enriched with a segmentation branch, on five popular datasets, against several approaches in the literature. "pre-trained" refers to models pre-trained on the large-scale UCF-QNRF dataset.*

4.3. Comparison with the state-of-the-art

In the previous sections we evaluated the effect of reducing background mistakes for crowd counting by enriching a model with a segmentation branch (sec. 4.1). For completeness, we now compare this architecture against other works in the literature. Results are presented in table 5. Despite its simplicity, our approach achieves remarkably competitive performance on all the five datasets. We find these results very encouraging, as they show that sometimes there is no need for complex architectures, but rather for simple solutions that tackle the right problem. Finally, we present some qualitative results of this approach in fig. 6.

5. Conclusions

We presented an extensive analysis on a problem that has been overlooked by the literature, yet it plays a fundamental part in the overall crowd counting performance and the applicability of crowd counting approaches to real world applications. Our results showed that the problem of counting on background regions is significant and in it is responsible for 18-49% of the total count error. Furthermore, we showed that this problem can be substantially mitigated by teaching the counting network the concept of background. By simply enriching a crowd counting network with a background segmentation branch we were able to reduce these mitakes by up to 83%, leading to better crowd counting performance (up to 20%). Finally, such a simple architectural change led to results on par with the state-of-the-art, on all the evaluated datasets. We find these results remarkable and a clear indication that future research should start addressing this problem more directly.

References

[1] Carlos Arteta, Victor Lempitsky, and Andrew Zisserman. Counting in the wild. In *ECCV*, 2016.

[2] Lokesh Boominathan, Srinivas SS Kruthiventi, and R Venkatesh Babu. Crowdnet: A deep convolutional network for dense crowd counting. In *ACM*, 2016.

[3] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. Scale aggregation network for accurate and efficient crowd counting. In *ECCV*, 2018.

[4] Antoni B Chan, Zhang-Sheng John Liang, and Nuno Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *CVPR*, 2008.

[5] Antoni B Chan and Nuno Vasconcelos. Bayesian poisson regression for crowd counting. In *CVPR*, 2009.

[6] Zhi-Qi Cheng, Jun-Xiu Li, Qi Dai, Xiao Wu, and Alexander Hauptmann. Learning spatial awareness to improve crowd counting. In *ICCV*, 2019.

[7] Junyu Gao, Qi Wang, and Xuelong Li. Pcc net: Perspective crowd counting via spatial convolutional network. *TCSVT*, 2019.

[8] Ricardo Guerrero-Gómez-Olmedo, Beatriz Torre-Jiménez, Roberto López-Sastre, Saturnino Maldonado-Bascón, and Daniel Onoro-Rubio. Extremely overlapping vehicle counting. In *ibPRIA*, 2015.

[9] Siyu Huang, Xi Li, Zhongfei Zhang, Fei Wu, Shenghua Gao, Rongrong Ji, and Junwei Han. Body structure aware deep crowd counting. *TIP*, 27(3):1049–1059, 2017.

[10] 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 *ECCV*, 2018.

[11] Xiaoheng Jiang, Li Zhang, Mingliang Xu, Tianzhu Zhang, Pei Lv, Bing Zhou, Xin Yang, and Yanwei Pang. Attention scaling for crowd counting. In *CVPR*, 2020.

[12] Di Kang and Antoni Chan. Crowd counting by adaptively fusing predictions from an image pyramid. In *BMVC*, 2018.

[13] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *NIPS*, 2010.

[14] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *CVPR*, 2018.

[15] Jiang Liu, Chenqiang Gao, Deyu Meng, and Alexander G Hauptmann. Decidenet: Counting varying density crowds through attention guided detection and density estimation. In *CVPR*, 2018.

[16] Liang Liu, Hao Lu, Hongwei Zou, Haipeng Xiong, Zhiguo Cao, and Chunhua Shen. Weighing counts: Sequential crowd counting by reinforcement learning. In *ECCV*, 2020.

[17] Lingbo Liu, Zhilin Qiu, Guanbin Li, Shufan Liu, Wanli Ouyang, and Liang Lin. Crowd counting with deep structured scale integration network. In *ICCV*, 2019.

[18] Xiyang Liu, Jie Yang, and Wenrui Ding. Adaptive mixture regression network with local counting map for crowd counting. In *ECCV*, 2020.

[19] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Bayesian loss for crowd count estimation with point supervision. In *ICCV*, 2019.

[20] Daniel Onoro-Rubio and Roberto J López-Sastre. Towards perspective-free object counting with deep learning. In *ECCV*, 2016.

[21] Mikel Rodriguez, Ivan Laptev, Josef Sivic, and Jean-Yves Audibert. Density-aware person detection and tracking in crowds. In *ICCV*, 2011.

[22] David Ryan, Simon Denman, Clinton Fookes, and Sridha Sridharan. Crowd counting using multiple local features. In *DICTA*, 2009.

[23] Deepak Babu Sam, Neeraj N Sajjan, R Venkatesh Babu, and Mukundhan Srinivasan. Divide and grow: Capturing huge diversity in crowd images with incrementally growing cnn. In *CVPR*, 2018.

[24] Deepak Babu Sam, Shiv Surya, and R Venkatesh Babu. Switching convolutional neural network for crowd counting. In *CVPR*, 2017.

[25] Miaojing Shi, Zhaohui Yang, Chao Xu, and Qijun Chen. Revisiting perspective information for efficient crowd counting. In *CVPR*, 2019.

[26] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016.

[27] Zenglin Shi, Mattes Pascal, and Cees G. M. Snoek. Counting with focus for free. In *ICCV*, 2019.

[28] Vishwanath A Sindagi and Vishal M Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In *ICCV*, 2017.

[29] Vishwanath A Sindagi and Vishal M Patel. Multi-level bottom-top and top-bottom feature fusion for crowd counting. In *ICCV*, 2019.

[30] Rahul Rama Varior, Bing Shuai, Joseph Tighe, and Davide Modolo. Scale-aware attention network for crowd counting. In *arXiv preprint arXiv:1901.06026*, 2019.

[31] Jia Wan, Wenhan Luo, Baoyuan Wu, Antoni B Chan, and Wei Liu. Residual regression with semantic prior for crowd counting. In *CVPR*, 2019.

[32] Meng Wang and Xiaogang Wang. Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In *CVPR*, 2011.

[33] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. Learning from synthetic data for crowd counting in the wild. In *CVPR*, 2019.

[34] Bo Wu and Ram Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *ICCV*, 2005.

[35] Chenfeng Xu, Kai Qiu, Jianlong Fu, Song Bai, Yongchao Xu, and Xiang Bai. Learn to scale: Generating multipolar normalized density maps for crowd counting. In *ICCV*, 2019.

[36] Zhaoyi Yan, Yuchen Yuan, Wangmeng Zuo, Xiao Tan, Yezhen Wang, Shilei Wen, and Errui Ding. Perspective-guided convolution networks for crowd counting. In *ICCV*, 2019.

[37] Anran Zhang, Lei Yue, Jiayi Shen, Fan Zhu, Xiantong Zhen, Xianbin Cao, and Ling Shao. Attentional neural fields for crowd counting. In *ICCV*, 2019.

[38] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *CVPR*, 2015.

[39] Lu Zhang and Miaojing Shi. Crowd counting via scale-adaptive convolutional neural network. In *WACV*, 2018.

[40] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *CVPR*, 2016.

[41] Muming Zhao, Jian Zhang, Chongyang Zhang, and Wenjun Zhang. Leveraging heterogeneous auxiliary tasks to assist crowd counting. In *CVPR*, 2019.

[42] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.