Context-Aware Crowd Counting

Weizhe Liu Mathieu Salzmann Pascal Fua
Computer Vision Laboratory, École Polytechnique Fédérale de Lausanne (EPFL)
{weizhe.liu, mathieu.salzmann, pascal.fua}@epfl.ch

Abstract

State-of-the-art methods for counting people in crowded scenes rely on deep networks to estimate crowd density. They typically use the same filters over the whole image or over large image patches. Only then do they estimate local scale to compensate for perspective distortion. This is typically achieved by training an auxiliary classifier to select, for predefined image patches, the best kernel size among a limited set of choices. As such, these methods are not end-to-end trainable and restricted in the scope of context they can leverage.

In this paper, we introduce an end-to-end trainable deep architecture that combines features obtained using multiple receptive field sizes and learns the importance of each such feature at each image location. In other words, our approach adaptively encodes the scale of the contextual information required to accurately predict crowd density. This yields an algorithm that outperforms state-of-the-art crowd counting methods, especially when perspective effects are strong.

1. Introduction

Crowd counting is important for applications such as video surveillance and traffic control. In recent years, the emphasis has been on developing *counting-by-density* algorithms that rely on regressors trained to estimate the people density per unit area so that the total number can be obtained by integration, without explicit detection being required. The regressors can be based on Random Forests [18], Gaussian Processes [7], or more recently Deep Nets [41, 42, 26, 31, 40, 36, 32, 24, 19, 30, 33, 22, 15, 28, 5], with most state-of-the-art approaches now relying on the latter.

Standard convolutions are at the heart of these deep-learning-based approaches. By using the same filters and pooling operations over the whole image, these implicitly rely on the same receptive field everywhere. However, due to perspective distortion, one should instead change the receptive field size across the image. In the past, this has been addressed by combining either density maps extracted from image patches at different resolutions [26] or feature maps obtained with convolutional filters of different sizes [42, 5]. However, by indiscriminately fusing information at all scales, these methods ignore the fact that scale varies continuously across the image. While this was addressed in [31, 30] by training classifiers to predict the size of the receptive field to use locally, the resulting methods are not end-to-end trainable; cannot account for rapid scale changes because they assign a single scale to relatively large patches; and can only exploit a small range of receptive fields for the networks to remain of a manageable size.

In this paper, we introduce a deep architecture that explicitly extracts features over multiple receptive field sizes and learns the importance of each such feature at every image location, thus accounting for potentially rapid scale changes. In other words, our approach adaptively encodes the scale of the contextual information necessary to predict crowd density. This is in contrast to crowd-counting approaches that also use contextual information to account for scaling effects as in [32], but only in the loss function as opposed to computing true multi-scale features as we do. We will show that it works better on uncalibrated images. When calibration data is available, we will also show that it can be leveraged to infer suitable local scales even better and further increase performance.

Our contribution is therefore an approach that incorporates multi-scale contextual information directly into an end-to-end trainable crowd counting pipeline, and learns to exploit the right context at each image location. As shown by our experiments, we consistently outperform the state of the art on all standard crowd counting benchmarks, such as ShanghaiTech, WorldExpo'10, UCF_CC_50 and UCF_QNRF, as well as on our own Venice dataset [1], which features strong perspective distortion.

2. Related Work

Early crowd counting methods [39, 38, 20] tended to rely on *counting-by-detection*, that is, explicitly detecting

[1] https://sites.google.com/view/weizheliu/home/projects/context-aware-crowd-counting

individual heads or bodies and then counting them. Unfortunately, in very crowded scenes, occlusions make detection difficult, and these approaches have been largely displaced by *counting-by-density-estimation* ones, which rely on training a regressor to estimate people density in various parts of the image and then integrating. This trend began in [7, 18, 10], using either Gaussian Process or Random Forests regressors. Even though approaches relying on low-level features [9, 6, 4, 27, 7, 14] can yield good results, they have now mostly been superseded by CNN-based methods [42, 31, 5], a survey of which can be found in [36]. The same can be said about methods that count objects instead of people [1, 2, 8].

The people density we want to measure is the number of people per unit area *on the ground*. However, the deep nets operate in the image plane and, as a result, the density estimate can be severely affected by the local scale of a pixel, that is, the ratio between image area and corresponding ground area. This problem has long been recognized. For example, the algorithms of [41, 17] use geometric information to adapt the network to different scene geometries. Because this information is not always readily available, other works have focused on handling the scale implicitly within the model. In [36], this was done by learning to predict pre-defined density levels. These levels, however, need to be provided by a human annotator at training time. By contrast, the algorithms of [26, 32] use image patches extracted at multiple scales as input to a multi-stream network. They then either fuse the features for final density prediction [26] without accounting for continuous scale changes or introduce an *ad hoc* term in the training loss function [32] to enforce prediction consistency across scales. This, however, does not encode contextual information into the features produced by the network and therefore has limited impact. While [42, 5] aim to learn multi-scale features, by using different receptive fields, they combine all of these features to predict the density.

In other words, while the previous methods account for scale, they ignore the fact that the suitable scale varies smoothly over the image and should be handled adaptively. This was addressed in [16] by weighting different density maps generated from input images at various scales. However, the density map at each scale only depends on features extracted at this particular scale, and thus may already be corrupted by the lack of adaptive-scale reasoning. Here, we argue that one should rather extract *features* at multiple scales and learn how to adaptively combine them. While this, in essence, was also the motivation of [31, 30], which train an extra classifier to assign the best receptive field for each image patch, these methods remain limited in several important ways. First, they rely on classifiers, which requires pre-training the network before training the classifier, and thus is not end-to-end trainable. Second, they typically assign a single scale to an *entire* image patch that can still be large and thus do not account for rapid scale changes. Last, but not least, the range of receptive field sizes they rely on remains limited in part because using much larger ones would require using much deeper architectures, which may not be easy to train given the kind of networks being used.

By contrast, in this paper, we introduce an end-to-end trainable architecture that adaptively fuses multi-scale features, without explicitly requiring defining patches, but rather by learning how to weigh these features for each individual pixel, thus allowing us to accommodate rapid scale changes. By leveraging multi-scale pooling operations, our framework can cover an arbitrarily large range of receptive fields, thus enabling us to account for much larger context than with the multiple receptive fields used by the above-mentioned methods. In Section 4, we will demonstrate that it delivers superior performance.

3. Approach

As discussed above, we aim to exploit context, that is, the large-scale consistencies that often appear in images. However, properly assessing what the scope and extent of this context should be in images that have undergone perspective distortion is a challenge. To meet it, we introduce a new deep net architecture that adaptively encodes multi-level contextual information into the features it produces. We then show how to use these scale-aware features to regress to a final density map, both when the cameras are not calibrated and when they are.

3.1. Scale-Aware Contextual Features

We formulate crowd counting as regressing a people density map from an image. Given a set of N training images $\{I_i\}_{1\leq i\leq N}$ with corresponding ground-truth density maps $\{D_i^{gt}\}$, our goal is to learn a non-linear mapping $\mathcal{F}$ parameterized by θ that maps an input image I_i to an estimated density map $D_i^{est}(I_i) = \mathcal{F}(I_i, \theta)$ that is as similar as possible to D_i^{gt} in L^2 norm terms.

Following common practice [25, 29, 23], our starting point is a network comprising the first ten layers of a pre-trained *VGG-16* network [34]. Given an image I, it outputs features of the form

$$\mathbf{f}_v = \mathcal{F}_{vgg}(I) \ , \tag{1}$$

which we take as base features to build our scale-aware ones.

As discussed in Section 2, the limitation of $\mathcal{F}_{vgg}$ is that it encodes the same receptive field over the entire image. To remedy this, we compute scale-aware features by performing *Spatial Pyramid Pooling* [11] to extract multi-scale context information from the VGG features of Eq. 1. Specifically, as illustrated at the bottom of Fig. 1, we compute these

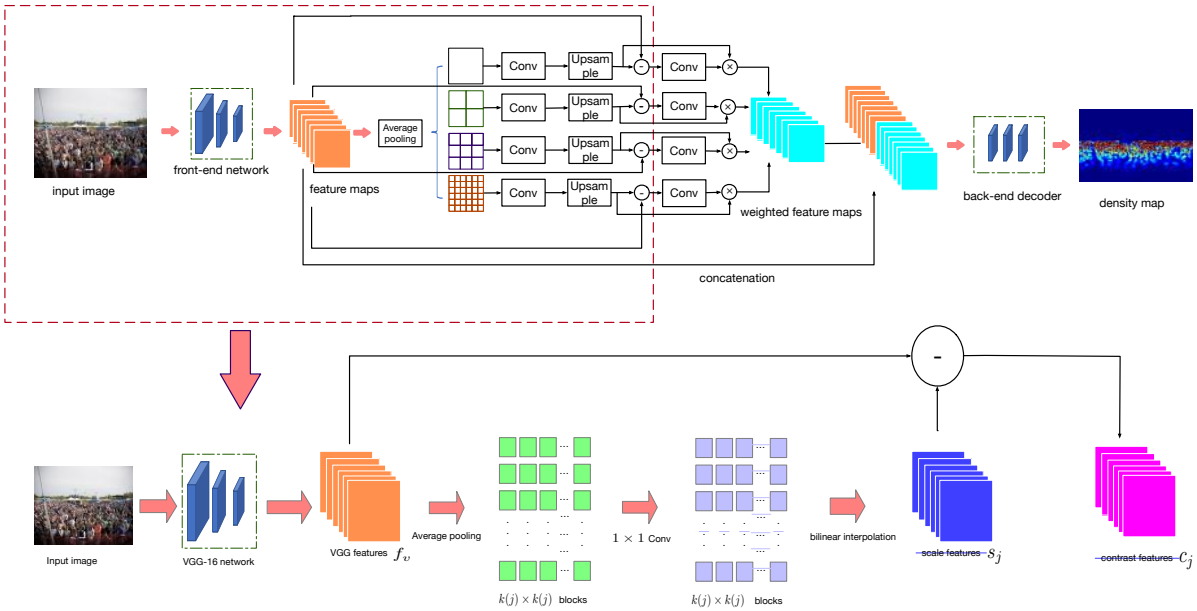


Figure 1: **Context-Aware Network. (Top)** RGB images are fed to a font-end network that comprises the first 10 layers of the *VGG-16* network. The resulting local features are grouped in blocks of different sizes by average pooling followed by a 1×1 convolutional layer. They are then up-sampled back to the original feature size to form the contrast features. Contrast features are further used to learn the weights for the scale-aware features that are then fed to a back-end network to produce the final density map. **(Bottom)** As shown in this expanded version of the first part of the network, the contrast features are the difference between local features and context features.

scale-aware features as

$$\mathbf{s}_j = U_{bi}(\mathcal{F}_j(P_{ave}(\mathbf{f}_v, j), \theta_j)) \ , \tag{2}$$

where, for each scale j, $P_{ave}(\cdot, j)$ averages the VGG features into $k(j) \times k(j)$ blocks; $\mathcal{F}_j$ is a convolutional network with kernel size 1 to combine the context features across channels without changing their dimensions. We do this because SPP keeps each feature channel independent, thus limiting the representation power. We verified that without this the performance drops. This is in contrast to earlier arthitectures that convolve to reduce the dimension [37, 43]; and U_{bi} represents bilinear interpolation to up-sample the array of contextual features to be of the same size as $\mathbf{f}_v$. In practice, we use $S = 4$ different scales, with corresponding block sizes $k(j) \in \{1, 2, 3, 6\}$ since it shows better performance compared with other settings.

The simplest way to use our scale-aware features would be to concatenate all of them to the original VGG features $\mathbf{f}_v$. This, however, would not account for the fact that scale varies across the image. To model this, we propose to learn to predict weight maps that set the relative influence of each scale-aware feature at each spatial location. To this end, we first define contrast features as

$$\mathbf{c}_j = \mathbf{s}_j - \mathbf{f}_v \ . \tag{3}$$

They capture the differences between the features at a specific location and those in the neighborhood, which often is an important visual cue that denotes saliency. Note that, for human beings, saliency matters. For example, in the image of Fig. 2, the eye is naturally drawn to the woman at the center in part because edges in the rest of the image all point in her direction and that edges at her location do not. In our context, these contrast features provide us with important information to understand the local scale of each image region. We therefore exploit them as input to auxiliary networks with weights θ^j_{sa} that compute the weights ω_j assigned to each one of the S different scales we use. Each such network outputs a scale-specific weight map of the form

$$\omega_j = \mathcal{F}^j_{sa}(\mathbf{c}_j, \theta^j_{sa}) \ . \tag{4}$$

$\mathcal{F}^j_{sa}$ is a 1×1 convolutional layer followed by a sigmoid function to avoid division by zero. We then employ these weights to compute our final contextual features as

$$\mathbf{f}_I = \left[\mathbf{f}_v | \frac{\sum_{j=1}^{S} \omega_j \odot \mathbf{s}_j}{\sum_{j=1}^{S} \omega_j} \right] \ , \tag{5}$$

where $[\cdot|\cdot]$ denotes the channel-wise concatenation operation, and $\odot$ is the element-wise product between a weight map and a feature map.

Altogether, as illustrated in Fig. 1, the network $\mathcal{F}(I, \theta)$ extracts the contextual features $\mathbf{f}_I$ as discussed above, which are then passed to a decoder consisting of several dilated convolutions that produces the density map. The specific architecture of the network is described in Table 1. As shown

Figure 2: **Context and saliency.** People's gaze tends be drawn to the person in the center, probably because most the image edges point in that direction.

by our experiments, this network already outperforms the state of the art on all benchmark datasets, without explicitly using information about camera geometry. As discussed below, however, these results can be further improved when such information is available.

3.2. Geometry-Guided Context Learning

Because of perspective distortion, the contextual scope suitable for each region varies across the image plane. Hence, scene geometry is highly related to contextual information and could be used to guide the network to better adjust to the scene context it needs.

We therefore extend the previous approach to exploiting geometry information when it is available. To this end, we represent the scene geometry of image I_i with a *perspective map* M_i, which encodes the number of pixels per meter in the image plane. Note that this perspective map has the same spatial resolution as the input image. We therefore use it as input to a truncated *VGG-16* network. In other words, the base features of Eq. 1 are then replaced by features of the form

$$\mathbf{f}_g = \mathcal{F'}_{vgg}(M_i, \theta_g) \ , \tag{6}$$

where $\mathcal{F'}_{vgg}$ is a modified *VGG-16* network with a single input channel. To initialize the weights corresponding to this channel, we average those of the original three RGB channels. Note that we also normalize the perspective map M_i to lie within the same range as the RGB images. Even though this initialization does not bring any obvious difference in the final counting accuracy, it makes the network converge much faster.

To further propagate the geometry information to later stages of our network, we exploit the modified VGG features described above, which inherently contain geometry information, as an additional input to the auxiliary network of Eq. 4. Specifically, the weight map for each scale is then computed as

layer	front-end($\mathcal{F}_{vgg}$)	layer	back-end decoder
1 - 2	3×3×64 conv-1	1	3×3×512 conv-2
	2 × 2 max pooling	2	3×3×512 conv-2
3 - 4	3×3×128 conv-1	3	3×3×512 conv-2
	2 × 2 max pooling	4	3×3×256 conv-2
5 - 7	3×3×256 conv-1	5	3×3×128 conv-2
	2 × 2 max pooling	6	3×3×64 conv-2
8 - 10	3×3×512 conv-1	7	1×1×1 conv-1

Table 1: **Network architecture of proposed model** Convolutional layers are represented as "(kernel size) × (kernel size) × (number of filters) conv-(dilation rate)".

$$\omega_j = \mathcal{F}^j_{gc}([\mathbf{c}_j|\mathbf{f}_g] \, , \theta^j_{gc}) \ . \tag{7}$$

These weight maps are then used as in Eq. 5. Fig. 3 depicts the corresponding architecture.

3.3. Training Details and Loss Function

Whether with or without geometry information, our networks are trained using the L^2 loss defined as

$$L(\theta) = \frac{1}{2B} \sum_{i=1}^{B} \|D_i^{gt} - D_i^{est}\|_2^2 \ , \tag{8}$$

where B is the batch size. To obtain the ground-truth density maps D_i^{gt}, we rely on the same strategy as previous work [19, 31, 42, 30]. Specifically, to each image I_i, we associate a set of c_i 2D points $P_i^{gt} = \{P_i^j\}_{1 \leq j \leq c_i}$ that denote the position of each human head in the scene. The corresponding ground-truth density map D_i^{gt} is obtained by convolving an image containing ones at these locations and zeroes elsewhere with a Gaussian kernel $\mathcal{N}^{gt}(p|\mu, \sigma^2)$ [21]. We write

$$\forall p \in I_i, D_i^{gt}(p|I_i) = \sum_{j=1}^{c_i} \mathcal{N}^{gt}(p|\mu = P_i^j, \sigma^2) \ , \tag{9}$$

where μ and σ represent the mean and standard deviation of the normal distribution. To produce the comparative results we will show in Section 4, we use the same σ as the methods we compare against.

To minimize the loss of Eq. 8, we use Stochastic Gradient Descent (SGD) with batch size 1 for various size dataset and Adam with batch size 32 for fixed size dataset. Furthermore, during training, we randomly crop image patches of $\frac{1}{4}$ the size of the original image at different locations. These patches are further mirrored to double the training set.

4. Experiments

In this section, we evaluate the proposed approach. We first introduce the evaluation metrics and benchmark

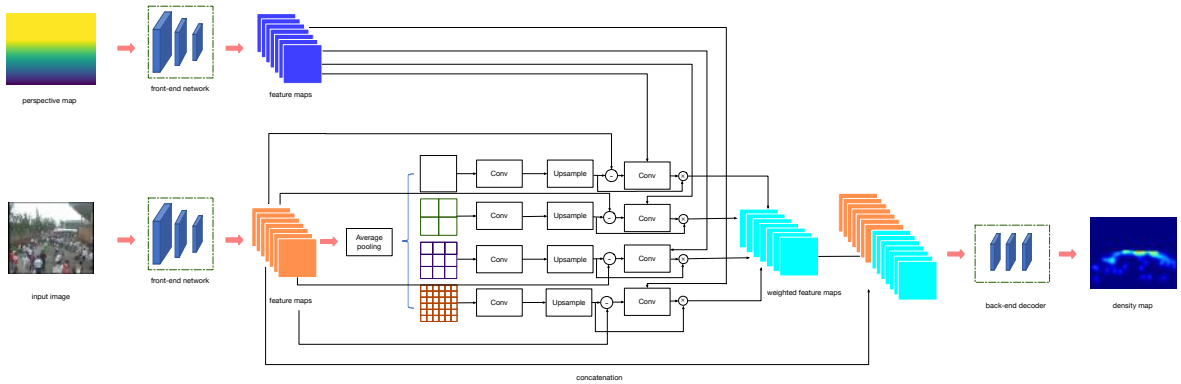


Figure 3: **Expanded Context-Aware Network.** To account for camera registration information when available, we add a branch to the architecture of Fig. 1. It takes as input a perspective map that encodes local scale. Its output is concatenated to the original contrast features and the resulting scale-aware features are used to estimate people density.

datasets we use in our experiments. We then compare our approach to state-of-the-art methods, and finally perform a detailed ablation study.

4.1. Evaluation Metrics

Previous works in crowd density estimation use the mean absolute error (MAE) and the root mean squared error ($RMSE$) as evaluation metrics [42, 41, 26, 31, 40, 36]. They are defined as

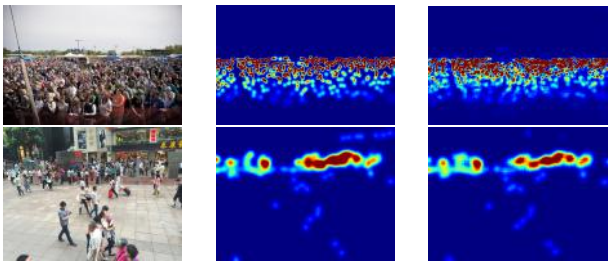
$$MAE = \frac{1}{N}\sum_{i=1}^{N}|z_i - \hat{z}_i| \text{ and } RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(z_i - \hat{z}_i)^2}\ ,$$

where N is the number of test images, z_i denotes the true number of people inside the ROI of the ith image and $\hat{z}_i$ the estimated number of people. In the benchmark datasets discussed below, the ROI is the whole image except when explicitly stated otherwise. Note that number of people can be recovered by integrating over the pixels of the predicted density maps as $\hat{z}_i = \sum_{p \in I_i} D_i^{est}(p|I_i)$.

4.2. Benchmark Datasets and Ground-truth Data

We use five different datasets to compare our approach to recent ones. The first four were released along with recent papers and have already been used for comparison purposes since. We created the fifth one ourselves and will make it publicly available as well.

ShanghaiTech [42]. It comprises 1,198 annotated images with 330,165 people in them. It is divided in part A with 482 images and part B with 716. In part A, 300 images form the training set and, in part B, 400. The remainder are used for testing purposes. For a fair comparison with earlier work [42, 32, 19, 33], we created the ground-truth density maps in the same manner as they did. Specifically, for Part A, we used the geometry-adaptive kernels introduced in [42], and for part B, fixed kernels. In Fig. 4, we show one image from each part, along with the ground-truth density maps and those estimated by our algorithm.



(a) Input image (b) Ground truth (c) Our prediction

Figure 4: **Crowd density estimation on ShanghaiTech**. First row: Image from Part A. Second row: Image from Part B. Our model adjusts to rapid scale changes and delivers density maps that are close to the ground truth.

UCF-QNRF [15]. It comprises 1,535 jpeg images with 1,251,642 people in them. The training set is made of 1,201 of these images. Unlike in **ShanghaiTech**, there are dramatic variations both in crowd density and image resolution. The ground-truth density maps were generated by adaptive Gaussian kernels as in [15].

UCF_CC_50 [14]. It contains only 50 images with a people count varying from 94 to 4,543, which makes it challenging for a deep-learning approach. For a fair comparison again, the ground-truth density maps were generated using fixed kernels and we follow the same 5-fold cross-validation protocol as in [14]: We partition the images into 5 10-image groups. In turn, we then pick four groups for training and the remaining one for testing. This gives us 5 sets of results and we report their average.

	Part_A		Part_B	
Model	MAE	$RMSE$	MAE	$RMSE$
Zhang *et al.* [41]	181.8	277.7	32.0	49.8
MCNN [42]	110.2	173.2	26.4	41.3
Switch-CNN [31]	90.4	135.0	21.6	33.4
CP-CNN [36]	73.6	106.4	20.1	30.1
ACSCP [32]	75.7	102.7	17.2	27.4
Liu *et al.* [24]	73.6	112.0	13.7	21.4
D-ConvNet [33]	73.5	112.3	18.7	26.0
IG-CNN [30]	72.5	118.2	13.6	21.1
ic-CNN[28]	68.5	116.2	10.7	16.0
CSRNet [19]	68.2	115.0	10.6	16.0
SANet [5]	67.0	104.5	8.4	13.6
OURS-CAN	**62.3**	**100.0**	**7.8**	**12.2**

Table 2: **Comparative results on the ShanghaiTech dataset.**

WorldExpo'10 [41]. It comprises 1,132 annotated video sequences collected from 103 different scenes. There are 3,980 annotated frames, with 3,380 of them used for training purposes. Each scene contains a Region Of Interest (ROI) in which people are counted. The bottom row of Fig. 5 depicts three of these images and the associated camera calibration data. We generate the ground-truth density maps as in our baselines [31, 19, 5]. As in previous work [41, 42, 31, 30, 19, 5, 21, 36, 32, 28, 33] on this dataset, we report the *MAE* of each scene, as well as the average over all scenes.

Venice. The four datasets discussed above have the advantage of being publicly available but do not contain precise calibration information. In practice, however, it can be readily obtained using either standard photogrammetry techniques or onboard sensors, for example when using a drone to acquire the images. To test this kind of scenario, we used a cellphone to film additional sequences of the Piazza San Marco in Venice, as seen from various viewpoints on the second floor of the basilica, as shown in the top two rows of Fig. 5. We then used the white lines on the ground to compute camera models. As shown in the bottom two rows of Fig. 5, this yields a more accurate calibration than in **WorldExpo'10**. The resulting dataset contains 4 different sequences and in total 167 annotated frames with fixed 1,280 $\times$ 720 resolution. 80 images from a single long sequence are taken as training data, and we use the images from the remaining 3 sequences for testing purposes. The ground-truth density maps were generated using fixed Gaussian kernels as in part B of the **ShanghaiTech** dataset.

4.3. Comparing against Recent Techniques

In Tables 2, 3, 4, and 5, we compare our results to those of the method that returns the best results for each one of the 4 public datasets, as currently reported in the literature.

Model	MAE	$RMSE$
Idrees *et al.* [14]	315	508
MCNN [42]	277	426
Encoder-Decoder [3]	270	478
CMTL [35]	252	514
Switch-CNN [31]	228	445
Resnet101 [12]	190	277
Densenet201 [13]	163	226
Idrees *et al.* [15]	132	191
OURS-CAN	**107**	**183**

Table 3: **Comparative results on the UCF_QNRF dataset.**

Model	MAE	$RMSE$
Idrees *et al.*[14]	419.5	541.6
Zhang *et al.* [41]	467.0	498.5
MCNN [42]	377.6	509.1
Switch-CNN [31]	318.1	439.2
CP-CNN [36]	295.8	320.9
ACSCP [32]	291.0	404.6
Liu *et al.* [24]	337.6	434.3
D-ConvNet [33]	288.4	404.7
IG-CNN [30]	291.4	349.4
ic-CNN[28]	260.9	365.5
CSRNet [19]	266.1	397.5
SANet [5]	258.4	334.9
OURS-CAN	**212.2**	**243.7**

Table 4: **Comparative results on the UCF_CC_50 dataset.**

Model	Scene1	Scene2	Scene3	Scene4	Scene5	Average
Zhang *et al.* [41]	9.8	14.1	14.3	22.2	3.7	12.9
MCNN [42]	3.4	20.6	12.9	13.0	8.1	11.6
Switch-CNN [31]	4.4	15.7	10.0	11.0	5.9	9.4
CP-CNN [36]	2.9	14.7	10.5	10.4	5.8	8.9
ACSCP [32]	2.8	14.05	9.6	8.1	2.9	7.5
IG-CNN [30]	2.6	16.1	10.15	20.2	7.6	11.3
ic-CNN[28]	17.0	12.3	9.2	8.1	4.7	10.3
D-ConvNet [33]	**1.9**	12.1	20.7	8.3	**2.6**	9.1
CSRNet [19]	2.9	11.5	**8.6**	16.6	3.4	8.6
SANet [5]	2.6	13.2	9.0	13.3	3.0	8.2
DecideNet [21]	2.0	13.14	8.9	17.4	4.75	9.23
OURS-CAN	2.9	12.0	10.0	**7.9**	4.3	**7.4**
OURS-ECAN	2.4	**9.4**	8.8	11.2	4.0	**7.2**

Table 5: **Comparative results in MAE terms on the WorldExpo'10 dataset.**

They are those of [5], [15], [5], and [32], respectively. In each case, we reprint the results as given in these papers and add those of **OURS-CAN**, that is, our method as described in Section 3.1. On the first three datasets, we consistently and clearly outperform all other methods. On the **WorldExpo'10** dataset, we also outperform them on average, but

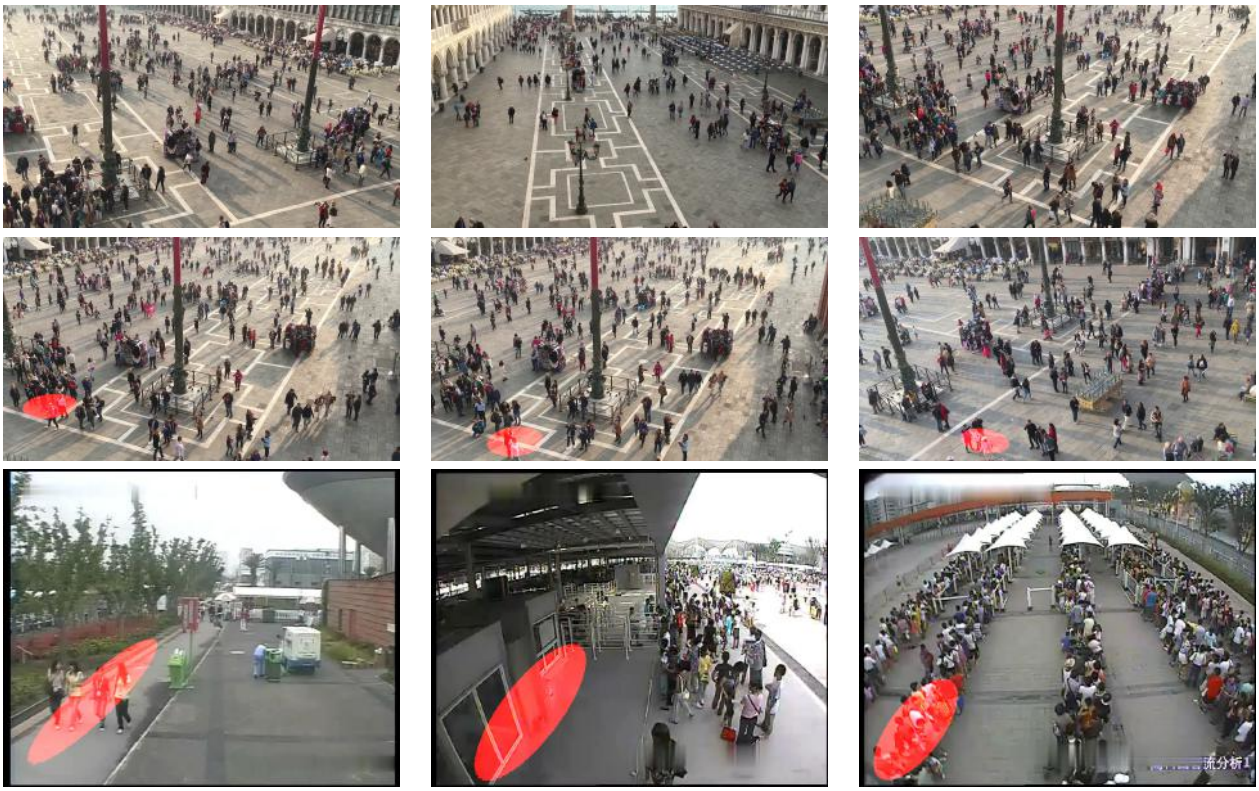


Figure 5: **Calibration in Venice and WorldExpo'10.** (Top row) Images of Piazza San Marco taken from different viewpoints. (Middle row) We used the regular ground patterns to accurately register the cameras in each frame. The red ellipse overlaid in red is the projection of a 1m radius circle from the ground plane to the image plane. (Bottom row) The same 1m radius circle overlaid on three WorldExpo'10 images. As can be seen in the bottom right image, the ellipse surface corresponds to an area that could be filled by many more people that could realistically fit in a 1m radius circle. By contrast, the ellipse deformations are more consistent and accurate for Venice, which denotes a better registration.

not in every scene. More specifically, in Scenes 2 and 4 that are crowded, we do very well. By contrast, the crowds are far less dense in Scenes 1 and 5. This makes context less informative and our approach still performs honorably but looses its edge compared to the others. Interestingly, as can be seen in Table 5, in such uncrowded scenes, a detection-based method such as DecideNet [21] becomes competitive whereas it isn't in the more crowded ones. In Fig. 6, we use a **Venice** image to show how well our approach does compared to the others in the crowded parts of the scene.

The first three datasets do not have any associated camera calibration data, whereas **WorldExpo'10** comes with a rough estimation of the image plane to ground plane homography and **Venice** with an accurate one. We therefore used these homographies to run **OURS-ECAN**, our method as described in Section 3.2. We report the results in Tables 5 and 6. Unsurprisingly, **OURS-ECAN** clearly further improves on **OURS-CAN** when the calibration data is accurate as for **Venice** and even when it is less so as for **WorldExpo**, but by a smaller margin.

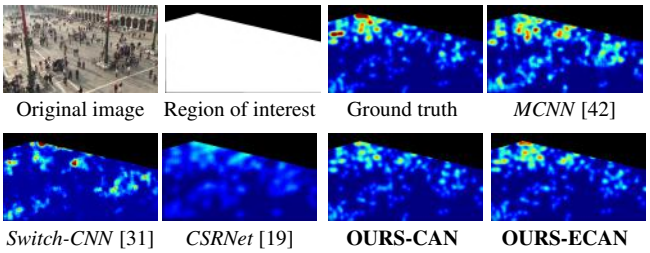
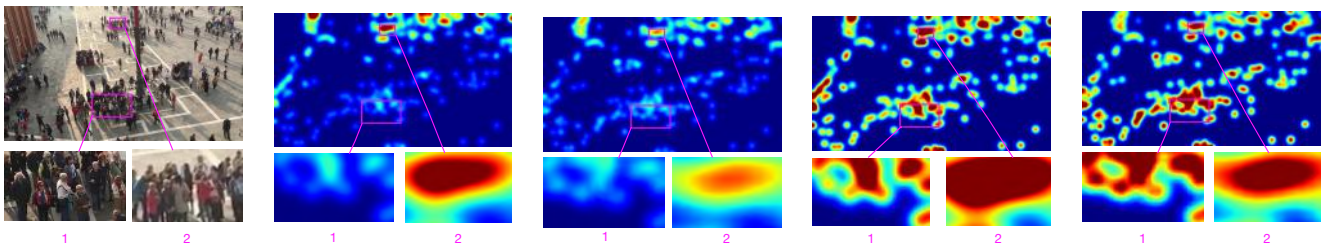


Figure 6: **Density estimation in Venice.** Original image, ROI, ground truth density map within the ROI, and density maps estimated both by the baselines and our method. Note how much more similar the density map produced by **OURS-ECAN** is to the ground truth than the others, especially in the upper corner of the ROI, where people density is high.

4.4. Ablation Study

Finally, we perform an ablation study to confirm the benefits of encoding multiple level contextual information and of introducing contrast features.



(a) (b) (c) (d) (e)

Figure 7: **Image-plane density vs ground-plane density.** **(a)** The two purple boxes highlight patches in which the crowd density per square meter is similar in the top image. **(b)** Ground-truth *image density* obtained by averaging the head annotations in the image plane as is done in *all* the approaches discussed in this paper, including ours. The bottom two patches are expanded versions of the same two purple boxes. The density appears much larger in one than in the other due to perspective distortion that increases the image density further away from the camera. **(c)** The density estimation returned by **OURS-ECAN**. **(d)** The ground-truth density normalized for image-scale variations so that it can be interpreted as a density per square meter. **(e)** The **OURS-ECAN** density similarly normalized. Note that the estimated densities in the two small windows now fall in the same range of values, which is correct.

Model	MAE	$RMSE$
MCNN [42]	145.4	147.3
Switch-CNN [31]	52.8	59.5
CSRNet[19]	35.8	50.0
OURS-CAN	23.5	38.9
OURS-ECAN	**20.5**	**29.9**

Table 6: **Comparative results on the Venice dataset.**

Model	MAE	$RMSE$
VGG-SIMPLE	68.0	113.4
VGG-CONCAT	63.4	108.7
VGG-NCONT	63.1	106.4
OURS-CAN	**62.3**	**100.0**

Table 7: **Ablation study on the ShanghaiTech part A dataset.**

Concatenating and Weighting VGG Features. We compare our complete model without geometry, **OURS-CAN**, against two simplified versions of it. The first one, **VGG-SIMPLE**, directly uses VGG-16 base features $\mathbf{f}_v$ as input to the decoder subnetwork. In other words, it does not adapt for scale. The second one, **VGG-CONCAT**, concatenates all scale-aware features $\{\mathbf{s}_j\}_{1\leq j\leq S}$ to the base features instead of computing their weighted linear combination, and then passes the resulting features to the decoder.

We compare these three methods on the **ShanghaiTech** Part A, which has often been used for such ablation studies [36, 5, 19]. As can be seen in Table 7, concatenating the VGG features as in **VGG-CONCAT** yields a significant boost, and weighing them as in **OURS-CAN** a further one.

Contrast Features. To demonstrate the importance of using contrast features to learn the network weights, we compare **OURS-CAN** against **VGG-NCONT** that uses the scale features $\mathbf{s}_j$ instead of the contrast ones to learn the weight maps. As can be seen in Table 7, this also results in a substantial performance loss.

5. Conclusion and Future Perspectives

In this paper, we have shown that encoding multi-scale context adaptively, along with providing an explicit model of perspective distortion effects as input to a deep net, substantially increases crowd counting performance. In particular, it yields much better density estimates in high-density regions.

This is of particular interest for crowd counting from mobile cameras, such as those carried by drones. In future work, we will therefore augment the image data with the information provided by the drone's inertial measurement unit to compute perspective distortions on the fly and allow monitoring from the moving drone.

We will also expand our approach to process consecutive images simultaneously and enforce temporal consistency, which among other things implies correcting ground-truth densities to also account for perspective distortions and be able to properly reason in the terms of ground-plane densities instead of image-plane densities, which none of the approaches discussed in this paper do. We did not do it either so that our results could be properly compared to the state of the art. However, as shown in Fig. 7, the price to pay is that the estimated densities, because they are close to this image-based ground truth, need to be corrected for perspective distortion before they can be treated as ground-plane densities. An obvious improvement would therefore be to directly regress to ground densities.

Acknowledgments This work was supported in part by the Swiss Federal Office for Defense Procurement.

References

[1] Carlos Arteta, Victor Lempitsky, J.Alison Noble, and Andrew Zisserman. Interactive Object Counting. In *European Conference on Computer Vision*, 2014.

[2] Carlos Arteta, Victor Lempitsky, and Andrew Zisserman. Counting in the Wild. In *European Conference on Computer Vision*, 2016.

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *arXiv Preprint*, 2015.

[4] Gabriel J. Brostow and Roberto Cipolla. Unsupervised Bayesian Detection of Independent Motion in Crowds. In *Conference on Computer Vision and Pattern Recognition*, pages 594–601, 2006.

[5] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. Scale Aggregation Network for Accurate and Efficient Crowd Counting. In *European Conference on Computer Vision*, 2018.

[6] Antoni B. Chan, Zhang-Sheng John Liang, and Nuno Vasconcelos. Privacy Preserving Crowd Monitoring: Counting People Without People Models or Tracking. In *Conference on Computer Vision and Pattern Recognition*, 2008.

[7] Antoni B. Chan and Nuno Vasconcelos. Bayesian Poisson Regression for Crowd Counting. In *International Conference on Computer Vision*, pages 545–551, 2009.

[8] Prithvijit Chattopadhyay, Ramakrishna Vedantam, Ramprasaath R. Selvaju, Dhruv Batra, and Devi Parikh. Counting Everyday Objects in Everyday Scenes. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[9] Ke Chen, Chen Change Loy, Shaogang Gong, and Tao Xiang. Feature Mining for Localised Crowd Counting. In *British Machine Vision Conference*, page 3, 2012.

[10] Luca Fiaschi, Rahul Nair, Ullrich Koethe, and Fred A. Hamprecht. Learning to Count with Regression Forest and Structured Labels. In *International Conference on Pattern Recognition*, pages 2685–2688, 2012.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In *European Conference on Computer Vision*, 2014.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[13] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[14] Haroon Idrees, Imran Saleemi, Cody Seibert, and Mubarak Shah. Multi-Source Multi-Scale Counting in Extremely Dense Crowd Images. In *Conference on Computer Vision and Pattern Recognition*, pages 2547–2554, 2013.

[15] 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 *European Conference on Computer Vision*, 2018.

[16] Di Kang and Antoni B. Chan. Crowd Counting by Adaptively Fusing Predictions from an Image Pyramid. In *British Machine Vision Conference*, 2018.

[17] Di Kang, Debarun Dhar, and Antoni B. Chan. Incorporating Side Information by Adaptive Convolution. In *Advances in Neural Information Processing Systems*, 2017.

[18] Victor Lempitsky and Andrew Zisserman. Learning to Count Objects in Images. In *Advances in Neural Information Processing Systems*, 2010.

[19] Yuhong Li, Xiaofan Zhang, and Deming Chen. CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[20] Zhe Lin and Larry S. Davis. Shape-Based Human Detection and Segmentation via Hierarchical Part-Template Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):604–618, 2010.

[21] Jiang Liu, Chenqiang Gao, Deyu Meng, and Alexander G. Hauptmann1. Decidenet: Counting Varying Density Crowds through Attention Guided Detection and Density Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[22] Linbo Liu, Hongjun Wang, Guanbin Li, Wanli Ouyang, and Liang Lin. Crowd Counting Using Deep Recurrent Spatial-Aware Network. In *International Joint Conference on Artificial Intelligence*, 2018.

[23] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single Shot Multibox Detector. In *European Conference on Computer Vision*, 2016.

[24] Xialei Liu, Joost van de Weijer, and Andrew D. Bagdanov. Leveraging Unlabeled Data for Crowd Counting by Learning to Rank. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2015.

[26] Daniel Onoro-Rubio and Roberto J. López-Sastre. Towards Perspective-Free Object Counting with Deep Learning. In *European Conference on Computer Vision*, pages 615–629, 2016.

[27] Vincent Rabaud and Serge Belongie. Counting Crowded Moving Objects. In *Conference on Computer Vision and Pattern Recognition*, pages 705–711, 2006.

[28] Viresh Ranjan, Hieu Le, and Minh Hoai. Iterative Crowd Counting. In *European Conference on Computer Vision*, 2018.

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems*, 2015.

[30] 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 *Conference on Computer Vision and Pattern Recognition*, 2018.

[31] Deepak Babu Sam, Shiv Surya, and R. Venkatesh Babu. Switching Convolutional Neural Network for Crowd Counting. In *Conference on Computer Vision and Pattern Recognition*, page 6, 2017.

[32] Zan Shen, Yi Xu, Bingbing Ni, Minsi Wang, Jianguo Hu, and Xiaokang Yang. Crowd Counting via Adversarial Cross-Scale Consistency Pursuit. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[33] Zenglin Shi, Le Zhang, Yun Liu, and Xiaofeng Cao. Crowd Counting with Deep Negative Correlation Learning. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[34] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*, 2015.

[35] Vishwanath A. Sindagi and Vishal M. Patel. CNN-based Cascaded Multi-task Learning of High-level Prior and Density Estimation for Crowd Counting. In *International Conference on Advanced Video and Signal Based Surveillance*, 2017.

[36] Vishwanath A. Sindagi and Vishal M. Patel. Generating High-Quality Crowd Density Maps Using Contextual Pyramid CNNs. In *International Conference on Computer Vision*, pages 1879–1888, 2017.

[37] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. In *Conference on Computer Vision and Pattern Recognition*, pages 1–9, June 2015.

[38] Xin Wang, Bin Wang, and Liming Zhang. Airport Detection in Remote Sensing Images Based on Visual Attention. In *International Conference on Neural Information Processing*, 2011.

[39] Bo Wu and Ram Nevatia. Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors. In *International Conference on Computer Vision*, 2005.

[40] Feng Xiong, Xinjian Shi, and Dit-Yan Yeung. Spatiotemporal Modeling for Crowd Counting in Videos. In *International Conference on Computer Vision*, pages 5161–5169, 2017.

[41] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-Scene Crowd Counting via Deep Convolutional Neural Networks. In *Conference on Computer Vision and Pattern Recognition*, pages 833–841, 2015.

[42] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-Image Crowd Counting via Multi-Column Convolutional Neural Network. In *Conference on Computer Vision and Pattern Recognition*, pages 589–597, 2016.

[43] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid Scene Parsing Network. In *Conference on Computer Vision and Pattern Recognition*, 2017.