Estimating People Flows to Better Count Them in Crowded Scenes

Weizhe Liu[1], Mathieu Salzmann[1,2], Pascal Fua[1]

[1]CVLab, EPFL, Switzerland [2]ClearSpace, Switzerland
{weizhe.liu, mathieu.salzmann, pascal.fua}@epfl.ch

Abstract. Modern methods for counting people in crowded scenes rely on deep networks to estimate people densities in individual images. As such, only very few take advantage of temporal consistency in video sequences, and those that do only impose weak smoothness constraints across consecutive frames.

In this paper, we advocate estimating people flows across image locations between consecutive images and inferring the people densities from these flows instead of directly regressing. This enables us to impose much stronger constraints encoding the conservation of the number of people. As a result, it significantly boosts performance without requiring a more complex architecture. Furthermore, it also enables us to exploit the correlation between people flow and optical flow to further improve the results.

We will demonstrate that we consistently outperform state-of-the-art methods on five benchmark datasets.

Keywords: Crowd Counting, Grid Flow Model, Temporal Consistency

1 Introduction

Crowd counting is important for applications such as video surveillance and traffic control. Most state-of-the-art approaches rely on regressors to estimate the local crowd density in individual images, which they then proceed to integrate over portions of the images to produce people counts. The regressors typically use Random Forests [16], Gaussian Processes [7], or more recently Deep Nets [55, 59, 30, 34, 49, 41, 36, 26, 17, 33, 40, 22, 14, 32, 5].

When video sequences are available, some algorithms use temporal consistency to impose weak constraints on successive density estimates. One way is to use an LSTM to model the evolution of people densities from one frame to the next [49]. However, this does not explicitly enforce the fact that people numbers must be strictly conserved as they move about, except at very specific locations where they can move in or out of the field of view. Modeling this was attempted in [24] but, because expressing this constraint in terms of people densities is difficult, the constraints actually enforced were much weaker.

In this paper, we propose to regress people flows, that is, the number of people moving from one location to another in the image plane, instead of densities.

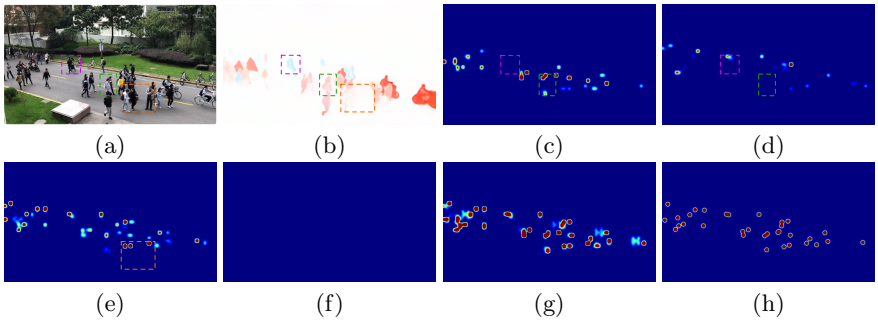


Fig. 1. From people flow to crowd density. (a) Original image. (b) Optical flow. Red denotes people moving right and blue moving left. The overlaid orange box encloses people moving slowly or not at all, the pink box people moving left, and the green box people moving right. (c) Estimated flow of people moving right. People moving left, such as those in the pink box, do not contribute to it, whereas those in the green box do. (d) Flow of people moving left. The situations within the pink and green box are reversed. (e) Estimated flow of people staying within the same grid location from one time instant to the next, such as those within the orange box. They are not necessarily static. They may simply not have had time to change location between the two time instants. (f) Estimated flow of people moving up. As no one does, it is almost zero everywhere. (g) Density map inferred by summing all the flows incident on a particular location. (h) Ground truth density map.

To this end, we partition the image into a number of grid locations and, for each one, we define ten potential flows, one towards each neighboring location, one towards the location *itself*, and the last towards regions outside the image plane. In practice, the last one is only used at boundary locations. The flow towards the location itself enables us to account for people who stay in the same location from one instant to the next and the final flow to account for people who enter or exit the field of view. Fig. 1 depicts some of the ten flows we compute. All the flows incident on a grid location are summed to yield an estimate of the people density in that location. The network can therefore be trained given ground-truth estimates only of the local people densities as opposed to people flows. In other words, even though we compute flows, our network only requires ground-truth density data for training purposes, like most others.

We will show that this formulation allows us to effectively impose people conservation constraints—people do not teleport from one region of the image to another—much more effectively than earlier approaches. This increases performance using network architectures that are neither deeper nor more complex than state-of-the-art ones. Furthermore, regressing people flows instead of densities provides a scene description that includes the motion direction and magnitude. This enables us to exploit the fact that people flow and optical flow should be highly correlated, as illustrated by Fig. 1, which provides an additional regularization constraint on the predicted flows and further enhances performance.

We will demonstrate on five benchmark datasets that our approach to enforcing temporal consistency brings a substantial performance boost compared to state-of-the-art approaches. Furthermore, if the cameras can be calibrated, we can apply our approach in the ground plane instead of the image plane, which further improves performance, as shown in the supplementary material. Our contribution is therefore a novel formulation of regressing people densities from video sequences that enforces strong temporal consistency constraints without requiring complex network architectures.

2 Related Work

Given a single image of a crowded scene, the currently dominant approach to counting people is to train a deep network to regress a people density estimate at every image location. This density is then integrated to deliver an actual count [48, 23, 27, 37, 25, 24, 15, 60, 57, 47, 18, 19, 52, 29, 21, 50, 51, 39, 42, 8, 46, 53, 54].

Enforcing Temporal Consistency. While most methods work on individual images, a few have nonetheless been extended to encode temporal consistency. Perhaps the most popular way to do so is to use an LSTM [13]. For example, in [49], the ConvLSTM architecture [38] is used for crowd counting purposes. It is trained to enforce consistency both in the forward and the backward direction. In [58], an LSTM is used in conjunction with an FCN [28] to count vehicles in video sequences. A Locality-constrained Spatial Transformer (LST) is introduced in [11]. It takes the current density map as input and outputs density maps in the next frames. The influence of these estimates on crowd density depends on the similarity between pixel values in pairs of neighboring frames.

While effective these approaches have two main limitations. First, at training time, they can only be used to impose consistency across annotated frames and cannot take advantage of unannotated ones to provide self-supervision. Second, they do not explicitly enforce the fact that people numbers must be conserved over time, except at the edges of the field of view. The recent method of [24] addresses both these issues. However, as will be discussed in more detail in Section 3.1, because the people conservation constraints are expressed in terms of numbers of people in neighboring image areas, they are much weaker than they should be.

Introducing Flow Variables. Imposing strong conservation constraints when tracking people has been a concern long before the advent of deep learning. For example, in [3], people tracking is formulated as multi-target tracking on a grid and gives rise to a linear program that can be solved efficiently using the K-Shortest Path algorithm [44]. The key to this formulation is the use as optimization variables of people flows from one grid location to another, instead of the actual number of people in each grid location. In [31], a people conservation constraint is enforced and the global solution is found by a greedy algorithm that sequentially instantiates tracks using shortest path computations on a flow network [56].

T	number of time steps
K	number of locations in the image plane
I^t	image at t-th frame
m_j^t	number of people present at location j at time t
$f_{i,j}^{t-1,t}$	number of people moving from location i to location j between times $t-1$ and t
$N(j)$	neighborhood of location j that can be reached within a single time step

Table 1. Notations.

Such people conservation constraints have since been combined with additional ones to further boost performance. They include appearance constraints [1, 10, 2] to prevent identity switches, spatio-temporal constraints to force the trajectories of different objects to be disjoint [12], and higher-order constraints [4, 9].

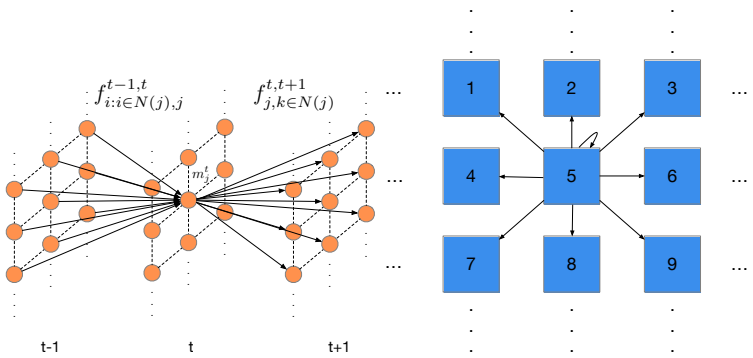
However, all these works predate deep learning. These kind of flow constraints have never been used in a deep crowd counting context and are designed for scenarios in which people can still be tracked individually. In this paper, we demonstrate that this approach can also be brought to bear in a deep pipeline to handle dense crowds in which people cannot be tracked as individuals anymore.

3 Approach

We regress *people flows* from images. We take these flows to be counts between two consecutive time instants of people either moving from their current location to a neighboring one, staying at the same location, or moving in or out of the field of view. They are depicted by Fig. 2 and summarized in Table 1. People flows incident on a specific location are then summed to derive the number of people per location or *people count* per location. The *crowd density* then simply is the *people count* divided by the location area. Our key insight is that this formulation enables us to impose much tighter *people conservation constraints* than earlier approaches. By this, we mean that we can accurately model the fact that all people present in a location at a given instant either were already there at the previous one or came from a neighboring location. This assumes the image frequency to be high enough for people not being able to move beyond neighboring locations in the time that separates consecutive frames. This is a common assumption that has proved both valid and effective in many earlier works.

3.1 Formalization

Let us consider a video sequence $\mathbf{I} = \{\mathbf{I}^1, \dots \mathbf{I}^T\}$ and three consecutive images $\mathbf{I}^{t-1}$, $\mathbf{I}^t$, and $\mathbf{I}^{t+1}$ from it. Let us assume that each image has been partitioned



(a) Grid model (b) Neighborhood of each location

Fig. 2. People flows. (a) The crowd density at time t at a given location can only come from neighboring grid locations at time $t-1$ and flow to neighboring grid locations at time $t+1$, in both cases including the location itself. (b) For each location not at the boundary of the image plane, there are nine locations reachable within a single time step, including the location itself. For locations at the edge of the image plane, we add a tenth location that represents the rest of the world. It allows for flows of people who either leave the image or enter it from outside.

into K rectangular grid locations. In our implementation, a location is one spatial position in the final convolutional feature map, corresponding to an 8×8 neighborhood in the image. However, other choices are possible.

The main constraint we want to enforce is that the number of people present at location j at time t is the number of people who were already there at time $t-1$ and stayed there plus the number of those who walked in from neighboring locations between $t-1$ and t. The number of people present at location j at time t also equals the sum of the number of people who stayed there until time $t+1$ and of people who went to a neighboring location between t and $t+1$.

Let m^t_j be the number of people present at location j at time t, or *people count* at that location. Let $f^{t-1,t}_{i,j}$ be the number of people who move from location i to location j between times $t-1$ and t, and $N(j)$ the neighborhood of location j that can be reached within a single time step. These notations are illustrated by Fig. 2 (a) and summarized in Table 1. In practice, we take $N(j)$ to be the 8 neighbors of grid location j plus the grid location itself to account for people who remain at the same place, as depicted by Fig. 2 (b). Our people conservation constraint can now be written as

$$\sum_{i\in N(j)} f^{t-1,t}_{i,j} = m^t_j = \sum_{k\in N(j)} f^{t,t+1}_{j,k} \,. \tag{1}$$

for all locations j that are *not* on the edge of the grid, that is, locations from which people cannot appear or disappear without being seen elsewhere in the image.

Most earlier approaches [30, 59, 5, 17, 20, 25, 23] regress the values of m_j^t, which makes it hard to impose the constraints of Eq. 1 because many different values of the flows $f_{i,j}^{t-1,t}$ can produce the same m_j^t values. For example, in [24], the equivalent constraint is

$$\forall j \quad m_j^t \leq \sum_{i \in N(j)} m_i^{t-1} \text{ and } m_j^t \leq \sum_{k \in N(j)} m_k^{t+1} \; . \tag{2}$$

It only states that the number of people at location j at time t is less than or equal to the total number of people at neighboring locations at time $t-1$ and that the same holds between times t and $t+1$. These are much looser constraints than the ones of Eq. 1. They guarantee that people cannot suddenly appear but do not account for the fact that people cannot suddenly disappear either. Our formulation lets us remedy this shortcoming. By regressing the $f_{i,j}^{t-1,t}$ from pairs consecutive images and computing the values of the m_j^t from these, we can impose the tighter constraints of Eq. 1.

3.2 Regressing the Flows

We now turn to the task of training a regressor that predicts flows that correspond to what is observed while obeying the above constraints and properly handling the boundary grid locations. Let us denote the regressor that predicts the flows from $\mathbf{I}^{t-1}$ and $\mathbf{I}^t$ as $\mathcal{F}$ with parameters Θ to be learned during training. In other words, $f^{t-1,t} = \mathcal{F}(I^{t-1}, I^t; \Theta)$ is the vector of predicted flows between all pairs of neighboring locations between times $t-1$ and t. In practice, $\mathcal{F}$ is implemented by a deep network. The predicted local people counts m_j^t, that is number of people per grid location j and at time t, are taken to be the sum of the incoming flows according to Eq. 1, and the predicted count for the whole image is the sum of all the m_j^t. As the flows are not directly observable, the training data comes in the form of *people counts* $\bar{m}_j^t$ per grid location j and at time t.

During training, our goal is therefore to find values of Θ such that

$$\bar{m}_j^t = \sum_{i \in N(j)} f_{i,j}^{t-1,t} = \sum_{k \in N(j)} f_{j,k}^{t,t+1} \text{ and } \quad f_{i,j}^{t-1,t} = f_{j,i}^{t,t-1} \; . \tag{3}$$

for all i, j, and t, except for locations at the edges of the image plane, where people can appear from and disappear to unseen parts of the scene. The first constraint is the people conservation constraint introduced in Section 3.1. The second accounts for the fact that, were we to play the video sequence in reverse, the flows should have the same magnitude but in the opposite direction. As will be discussed below, we enforce these constraints by incorporating them into the loss function we minimize to learn Θ. Finally, we impose that all the flows be non-negative by using ReLu normalization in the network that implements $\mathcal{F}$. Note that we only require the people flow to be non-negative, the fact that a location may contain less than 1 person simply means that the flow value will be less than 1.

Regressor Architecture. Recall that $f^{t-1,t} = \mathcal{F}(\mathbf{I}^{t-1}, \mathbf{I}^t; \Theta)$ is a vector of predicted flows from neighboring locations between times $t-1$ and t. In practice, $\mathcal{F}$ is implemented by the encoding/decoding architecture shown in Fig. 3, and $f^{t-1,t}$ has the same dimension as the image grid and 10 channels per location. The first are the flows to the 9 possible neighbors depicted by Fig. 2 (b) and the tenth represents potential flows from outside the image and is therefore only meaningful at the edges. The fifth channel denotes the flow towards the location itself, which enables us to account for people who stay in the same location from one instant to the next.

To compute $f^{t-1,t}$, consecutive frames $\mathbf{I}^{t-1}$ and $\mathbf{I}^t$ are fed to the CAN encoder network of [25]. This yields deep features $s^{t-1} = \mathcal{E}_e(I^{t-1}; \Theta_e)$ and $s^t = \mathcal{E}_e(I^t; \Theta_e)$, where $\mathcal{E}_e$ denotes the encoder with weights Θ_e. These features are then concatenated and fed to a decoder network to output $f^{t-1,t} = \mathcal{D}(s^{t-1}, s^t; \Theta_d)$, where $\mathcal{D}$ is the decoder with weights Θ_d. $\mathcal{D}$ comprises the back-end decoder of CAN [25] with an additional final ReLU layer to guarantee that the output is always non-negative. The encoder and decoder specifications are given in the supplementary material.

Grid Size. In all our experiments, we treated each spatial location in the output people flow map as a separate location. Since our CAN [25] backbone outputs a down-sampled density map, each output grid location represent an 8×8 pixel block in input image. This down-sampling rate is common in crowd counting models [25, 24, 17] because it represents a good compromise between high-resolution of the density map and efficiency of the model. In the supplementary material, we will confirm this by showing that changing the down-sampling rate degrades performance.

Loss Function and Training. To obtain the ground-truth maps $\bar{m}^t$ of Eq. 3, we use the same approach as in most previous work [30, 59, 5, 17, 20, 25, 23]. In each image $\mathbf{I}^t$, we annotate a set of c^t 2D points $P^t = \{P_i^t\}_{1 \leq i \leq c^t}$ that denote the positions of the human heads in the scene. The corresponding ground-truth density map $\bar{m}^t$ is obtained by convolving an image containing ones at these locations and zeroes elsewhere with a Gaussian kernel $\mathcal{N}(\cdot|\mu, \sigma^2)$ with mean μ and standard deviation σ. We write

$$\bar{m}_j^t = \sum_{i=1}^{c^t} \mathcal{N}(p_j|\mu = P_i^t, \sigma^2) \ , \ \forall j \ . \tag{4}$$

where p_j denotes the center of location j. Note that this formulation preserves the constraints of Eq. 3 because we perform the same convolution across the whole image. In other words, if a person moves in a given direction by n pixels, the corresponding contribution to the density map will shift in the same direction and also by n pixels.

The final ReLU layer of the regressor guarantees that the estimated flows are non-negative. To enforce the constraints of Eq. 3, we define our combined loss

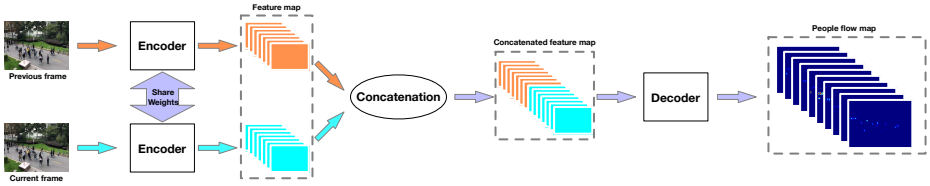


Fig. 3. Model Architecture: Two consecutive RGB image frames are fed to the same encoder network that relies on the CAN scale-aware feature extractor of [25]. These multi-scale features are further concatenated and fed to a decoder network to produce the final people flow maps.

function L_{combi} as the weighted sum of two loss terms. We write

$$L_{combi} = L_{flow} + \alpha L_{cycle} \ , \tag{5}$$

$$L_{flow} = \sum_{j \in I^t} \left[(\bar{m}_j^t - \sum_{i \in N(j)} f_{i,j}^{t-1,t})^2 + (\bar{m}_j^t - \sum_{k \in N(j)} f_{j,k}^{t,t+1})^2 \right] ,$$

$$L_{cycle} = \sum_{j \in I^t} \left[\sum_{i \in N(j)} (f_{i,j}^{t-1,t} - f_{j,i}^{t,t-1})^2 + \sum_{k \in N(j)} (f_{j,k}^{t,t+1} - f_{k,j}^{t+1,t})^2 \right] .$$

where $\bar{m}_j^t$ is the ground-truth crowd density value, that is, the *people count* at time t and location j of Eq. 4 and α is a scalar weight we set to 1 in all our experiments.

Although the variant of L_{combi} can be computed from only two consecutive frames, at training time we always use three to enforce the temporal consistency constraints of Eq. 1. Algorithm 1 describes our training scheme in more detail. Note that we do *not* assume all training frames to be annotated. Only frames V, $2V$, $3V$ need be with $V \geq 1$. To evaluate the loss function for frame kV, where k is an integer, we then use frames $kV-1$, kV, and $kV+1$, where one of the three is annotated. In practice, we could also use frames $kV-n$, kV, and $kV+n$ with $n \geq 1$.

3.3 Exploiting Optical Flow

When the camera is static, both the people flow discussed above and the optical flow that can be computed directly from the images stem for the motion of the people. They should therefore be highly correlated. In fact, this remains true even if the camera moves because its motion creates an apparent flow of people from one image location to another. However, there is no simple linear relationship between people flow and optical flow. To account for their correlation, we therefore introduce an additional loss function, which we define as

$$L_{optical} = \sum_j \delta(m_j > 0)(\mathbf{O}_j - \bar{o}_j^{t-1,t})^2 \ , \tag{6}$$

$$\text{where} \quad \mathbf{O} = \mathcal{F}_o(m^{t-1}, m^t; \Theta_o) \ .$$

Algorithm 1 Three-Frames Training Algorithm

Require: Training image sequence $\{\mathbf{I}^1, \ldots, \mathbf{I}^T\}$ with an interval V between annotated frames.

Require: Ground-truth density maps $\{\bar{m}^V, \bar{m}^{2V} ..., \bar{m}^{(T//V)V}\}$ computed by convolving the annotations according to Eq. 4.

procedure TRAIN($\{\mathbf{I}^1, .., \mathbf{I}^T\}$,$\{\bar{m}^V, .., \bar{m}^{(T//V)V}\}$)
  Initialize the weights Θ of regressor network $\mathcal{F}$
  for # of gradient iterations **do**
    Pick 3 consecutive frames $(\mathbf{I}^{t-1}, \mathbf{I}^t, \mathbf{I}^{t+1})$, where t is a multiple of V, meaning that only $\mathbf{I}^t$ is annotated
    Set $f^{t-1,t} = \mathcal{F}(I^{t-1}, I^t, \Theta)$
    Set $f^{t,t+1} = \mathcal{F}(I^t, I^{t+1}, \Theta)$
    Set $f^{t,t-1} = \mathcal{F}(I^t, I^{t-1}, \Theta)$
    Set $f^{t+1,t} = \mathcal{F}(I^{t+1}, I^t, \Theta)$
    Reconstruct density map m_1^t from $f^{t-1,t}$
    Reconstruct density map m_2^t from $f^{t,t+1}$
    Reconstruct density map m_3^t from $f^{t,t-1}$
    Reconstruct density map m_4^t from $f^{t+1,t}$
    Minimize L_{combi} of Eq. 5 w.r.t. Θ using Adam
  end for
end procedure

where m^{t-1} and m^t are density maps inferred from our predicted flows using Eq. 1, $\mathbf{O}_j$ denotes the corresponding predicted optical flow at grid location j by a pre-trained regressor $\mathcal{F}_o$, $\bar{o}^{t-1,t}$ is the optical flow from frames $t-1$ to t computed by a state-of-the-art optical flow network [43], and the indicator function term $\delta(m_j > 0)$ ensures that the correlation is only enforced where there are people. This is especially useful when the camera moves to discount the optical flows generated by the changing background. We also use CAN [25] as the optical flow regressor $\mathcal{F}_o$ with 2 input channels, one for m^{t-1} and the other m^t. This network is pre-trained separately on the training data and then used to train the people flow regressor. We refer the reader to the supplementary material for implementation details.

Pre-training the regressor $\mathcal{F}_o$ requires annotations for consecutive frames, that is, $V = 1$ in the definition of Algorithm 1. When such annotations are available, we use this algorithm again but replace L_{combi} by

$$L_{all} = L_{combi} + \beta L_{optical} \ . \tag{7}$$

In all our experiments, we set β to 0.0001 to account for the fact that the optical flow values are around 4,000 times larger than the people flow values.

4 Experiments

In this section, we first introduce the evaluation metrics and benchmark datasets used in our experiments. We then compare our results to those of current state-

of-the-art methods. Finally, we perform an ablation study to demonstrate the impact of individual constraints.

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 [59, 55, 30, 34, 49, 41]. 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. In practice, $\hat{z}_i$ is taken to be $\sum_{p\in I_i} m_p$, that is, the sum over all locations or people counts obtained by summing the predicted people flows.

4.2 Benchmark Datasets and Ground-truth Data

For evaluations purposes, we use five different datasets, for which the videos have been released along with recently published papers. The first one is a synthetic dataset with ground-truth optical flows. The other four are real world videos, with annotated people locations but without ground-truth optical flow. To use the optional optical flow constraints introduced in Section 3.3, we therefore use the pre-trained **PWC-Net** [43], as described in that section, to compute the loss function $L_{optical}$ of Eq. 6. Please refer to the supplementary material for additional details.

CrowdFlow [35]. This dataset consists of five synthetic sequences ranging from 300 to 450 frames each. Each one is rendered twice, once using a static camera and the other a moving one. The ground-truth optical flow is provided as shown in the supplementary material. As this dataset has not been used for crowd counting before, and the training and testing sets are not clearly described in [35], to verify the performance difference caused by using ground-truth optical flow vs. estimated one, we use the first three sequences of both the static and moving camera scenarios for training and validation, and the last two for testing.

FDST [11]. It comprises 100 videos captured from 13 different scenes with a total of 150,000 frames and 394,081 annotated heads. The training set consists of 60 videos, 9000 frames and the testing set contains the remaining 40 videos, 6000 frames. We use the same setting as in [11].

UCSD [6]. This dataset contains 2000 frames captured by surveillance cameras on the UCSD campus. The resolution of the frames is 238 $\times$ 158 pixels and the framerate is 10 fps. For each frame, the number of people varies from 11 to 46. We use the same setting as in [6], with frames 601 to 1400 used as training data and the remaining 1200 frames as testing data.

Venice [25]. It contains 4 different sequences and in total 167 annotated frames with fixed 1,280 × 720 resolution. As in [25], 80 images from a single long sequence are used as training data. The remaining 3 sequences are used for testing purposes.

WorldExpo'10 [55]. It comprises 1,132 annotated video sequences collected from 103 different scenes. There are 3,980 annotated frames, 3,380 of which are used for training purposes. Each scene contains a Region Of Interest (ROI) in which the people are counted. As in previous work [55, 59, 34, 33, 17, 5, 20, 41, 36, 32, 40] on this dataset, we report the *MAE* of each scene, as well as the average over all scenes.

4.3 Comparing against Recent Techniques

Model	*MAE*	*RMSE*
MCNN [59]	172.8	216.0
CSRNet[17]	137.8	181.0
CAN[25]	124.3	160.2
OURS-COMBI	97.8	112.1
OURS-ALL-EST	96.3	111.6
OURS-ALL-GT	**90.9**	**110.3**

(a)

Model	*MAE*	*RMSE*
MCNN [59]	3.77	4.88
ConvLSTM [49]	4.48	5.82
WithoutLST [11]	3.87	5.16
LST [11]	3.35	4.45
OURS-COMBI	2.17	2.62
OURS-ALL-EST	**2.10**	**2.46**

(b)

Model	*MAE*	*RMSE*
MCNN [59]	145.4	147.3
Switch-CNN [34]	52.8	59.5
CSRNet[17]	35.8	50.0
CAN[25]	23.5	38.9
ECAN[25]	20.5	29.9
GPC[24]	18.2	26.6
OURS-COMBI	**15.0**	**19.6**

(c)

Model	MAE	RMSE
Zhang *et al.* [55]	1.60	3.31
Hydra-CNN [30]	1.07	1.35
CNN-Boosting [45]	1.10	-
MCNN [59]	1.07	1.35
Switch-CNN [34]	1.62	2.10
ConvLSTM [49]	1.30	1.79
Bi-ConvLSTM [49]	1.13	1.43
ACSCP [36]	1.04	1.35
CSRNet [17]	1.16	1.47
SANet [5]	1.02	1.29
ADCrowdNet [23]	0.98	1.25
PACNN [37]	0.89	1.18
SANet+SPANet [8]	1.00	1.28
OURS-COMBI	0.86	1.13
OURS-ALL-EST	**0.81**	**1.07**

(d)

Model	Scene1	Scene2	Scene3	Scene4	Scene5	**Average**
Zhang *et al.* [55]	9.8	14.1	14.3	22.2	3.7	12.9
MCNN [59]	3.4	20.6	12.9	13.0	8.1	11.6
Switch-CNN [34]	4.4	15.7	10.0	11.0	5.9	9.4
CP-CNN [41]	2.9	14.7	10.5	10.4	5.8	8.9
ACSCP [36]	2.8	14.05	9.6	8.1	2.9	7.5
IG-CNN [33]	2.6	16.1	10.15	20.2	7.6	11.3
ic-CNN[32]	17.0	12.3	9.2	8.1	4.7	10.3
D-ConvNet [40]	**1.9**	12.1	20.7	8.3	**2.6**	9.1
CSRNet [17]	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 [20]	2.0	13.14	8.9	17.4	4.75	9.23
CAN [25]	2.9	12.0	10.0	**7.9**	4.3	7.4
ECAN [25]	2.4	**9.4**	8.8	11.2	4.0	7.2
PGCNet [52]	2.5	12.7	8.4	13.7	3.2	8.1
OURS-COMBI	2.2	10.8	**8.0**	8.8	3.2	**6.6**

(e)

Table 2. Comparative results on different datasets. (a) **CrowdFlow**. (b) **FDST**. (c) **Venice**. (d) **UCSD**. (e) **WorldExpo'10**.

We denote our model trained using the combined loss function L_{combi} of Section 3.2 as **OURS-COMBI** and the one using the full loss function L_{all} of Section 3.3 with ground-truth optical flow as **OURS-ALL-GT**. In other words, **OURS-ALL-GT** exploits the optical flow while **OURS-COMBI** does not. If the ground-truth optical flow is not available, we use the optical flow estimated by **PWC-Net** [43] and denote this model as **OURS-ALL-EST**.

Synthetic Data. Fig. 4 depicts a qualitative result, and we report our quantitative results on the **CrowdFlow** dataset in Table 2 (a). **OURS-COMBI** outperforms the competing methods by a significant margin while **OURS-ALL-EST** delivers a further improvement. Using the ground-truth optical flow values in our L_{all} loss term yields yet another performance improvement, that points to the fact that using better optical flow estimation than **PWC-Net** [43] might help.

Real Data. Fig. 5 depicts a qualitative result, and we report our quantitative results on the four real-world datasets in Tables 2 (b), (c), (d) and (e). For **FDST** and **UCSD**, annotations in consecutive frames are available, which enabled us to pre-train the $\mathcal{F}_o$ regressor of Eq. 6. We therefore report results for both **OURS-COMBI** and **OURS-ALL-EST**. By contrast, for **Venice** and **WorldExpo'10**, only a sparse subset of frames are annotated, and we therefore only report results for **OURS-COMBI**.

For **FDST**, **UCSD**, and **Venice**, our approach again clearly outperforms the competing methods, with the optical flow constraint further boosting performance when applicable. For **WorldExpo'10**, the ranking of the methods depends on the scene being used, but ours still performs best on average and on Scene3. In short, when the crowd is dense, our approach dominates the others. By contrast, when the crowd becomes very sparse as in Scene1 and Scene5, models that comprise a pool of different regressors, such as [40], gain an advantage. This points to a potential way to further improve our own method, that is, to also use a pool of regressors to estimate the people flows.

4.4 Ablation Study

Model	*MAE*	*RMSE*
BASELINE	124.3	160.2
IMAGE-PAIR	125.7	164.1
AVERAGE	128.9	174.6
WEAK [24]	121.2	155.7
OURS-FLOW	113.3	140.3
OURS-COMBI	97.8	112.1
OURS-ALL-EST	**96.3**	**111.6**

(a)

Model	*MAE*	*RMSE*
BASELINE	2.44	2.96
IMAGE-PAIR	2.48	3.10
AVERAGE	2.52	3.14
WEAK [24]	2.42	2.91
OURS-FLOW	2.31	2.85
OURS-COMBI	2.17	2.62
OURS-ALL-EST	**2.10**	**2.46**

(b)

Model	*MAE*	*RMSE*
BASELINE	0.98	1.26
IMAGE-PAIR	1.02	1.40
AVERAGE	1.01	1.31
WEAK [24]	0.96	1.30
OURS-FLOW	0.94	1.21
OURS-COMBI	0.86	1.13
OURS-ALL-EST	**0.81**	**1.07**

(c)

Table 3. Ablation study. (a) **CrowdFlow**. (b) **FDST**. (c) **UCSD**.

To confirm that the good performance we report really is attributable to our regressing flows instead of densities, we performed the following set of experiments. Recall from Section 3.2, that we use the CAN [25] architecture to regress the flows. Instead, we can use this network to directly regress the densities, as in the original paper. We will refer to this approach as **BASELINE**. In [24], it was suggested that people conservation constraints could be added by incorporating a loss term that enforces the conservation constraints of Eq. 2 that are weaker than those of Eq. 1, which are those we use in this paper. We will refer to this

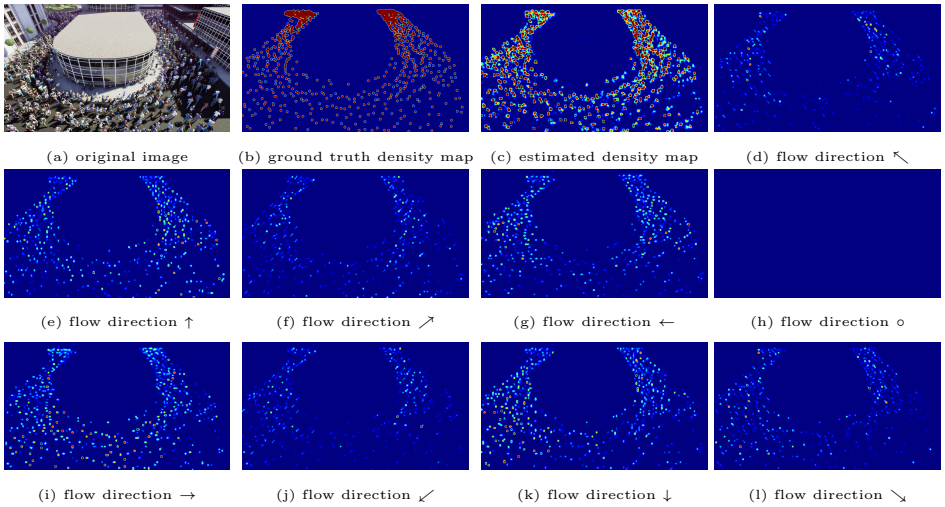


Fig. 4. Density estimation in CrowdFlow. People are running counterclockwise. The estimated people density map is close to the ground-truth one. It was obtained by summing the flows towards the 9 neighbors of Fig. 2 (b). They are denoted by the arrows and the circle. The latter corresponds to people not moving and is, correctly, empty. Note that the flow of people moving down is highest on the left of the building, moving right below the building, and moving up on the right of the building, which is also correct. Inevitably, there is also some noise in the estimated flow, some of which is attributable to body shaking while running.

approach relying on weaker constraints while still using the CAN backbone as **WEAK**. As **OURS-COMBI**, it takes two consecutive images as input. For the sake of completeness, we implemented a simplified approach, **IMAGE-PAIR**, which takes the same two images as input and directly regresses the densities. To show that regressing flows does not simply smoothe the densities, we implement one further approach, **AVERAGE**, which takes three images as input, uses CAN to independently compute three density maps, and then averages them. To highlight the importance of the forward-backward constraints of Eq. 3, we also tested a simplified version of our approach in which we drop them and that we refer to **OURS-FLOW**.

We compare the performance of these five approaches on **CrowdFlow**, **FDST**, and **UCSD** in Table 3. Both **IMAGE-PAIR** and **AVERAGE** do worse than **BASELINE**, which confirms that temporal averaging of the densities is not the right thing to do. As reported in [24], **WEAK** delivers a small improvement. However, using our stronger constraints brings a much larger improvement, thereby confirming the importance of properly modeling the flows as we do here. As expected **OURS-FLOW** improves on **IMAGE-PAIR** in all three datasets, with further performance increase for **OURS-COMBI** and **OURS-ALL-EST**. This confirms that using people flows instead of densities is a win and the additional constraints we impose all make positive contributions.

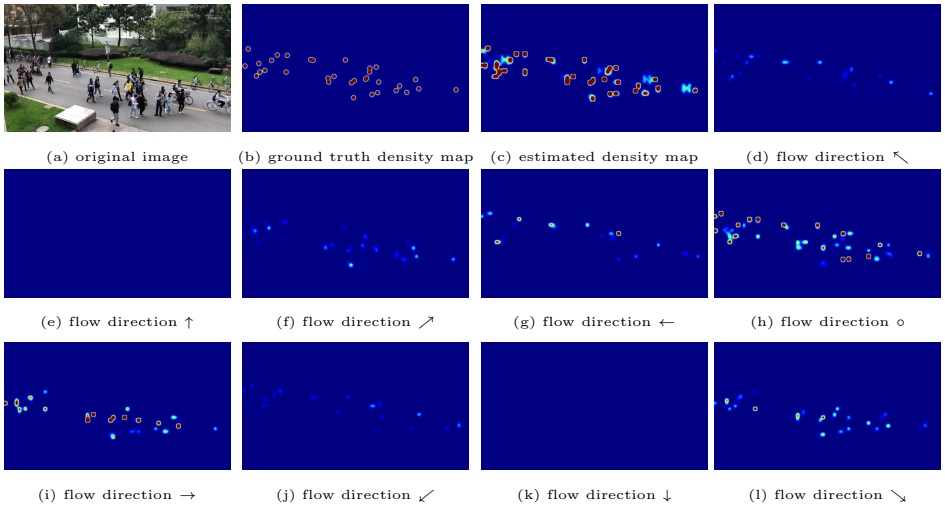


Fig. 5. Density estimation in FDST. People mostly move from left to right. The estimated people density map is close to the ground-truth one. It was obtained by summing the flows towards the 9 neighbors of Fig. 2 (b). They are denoted by the arrows and the circle. Strong flows occur in (g),(h), and (i), that is, moving left, moving right, or not having moved. Note that the latter does not mean that the people are static but only that they have not had time to change grid location between the two time instants.

5 Conclusion

We have shown that implementing a crowd counting algorithm in terms of estimating the people flows and then summing them to obtain people densities is more effective than attempting to directly estimate the densities. This is because it allows us to impose conservation constraints that make the estimates more robust. When optical flow data can be obtained, it also enables us to exploit the correlation between optical flow and people flow to further improve the results.

In this paper, we have focused on performing all the computations in image space, in large part so that we could compare our results to that of other recent algorithms that also work in image space. We have nonetheless shown in the supplementary material, that modeling the people flows in the ground plane yields even better performance. A promising application is to use drones for people counting because their internal sensors can be directly used to provide the camera registration parameters necessary to compute the homographies between the camera and the ground plane. In this scenario, the drone sensors also provide a motion estimate, which can be used to correct the optical flow measurements and therefore exploit the information they provide as effectively as if the camera was static.

Acknowledgments This work was supported in part by the Swiss National Science Foundation.

References

1. BenShitrit, H., Berclaz, J., Fleuret, F., Fua, P.: Tracking Multiple People Under Global Apperance Constraints. In: International Conference on Computer Vision (2011)
2. BenShitrit, H., Berclaz, J., Fleuret, F., Fua, P.: Multi-Commodity Network Flow for Tracking Multiple People. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(8), 1614–1627 (2014)
3. Berclaz, J., Fleuret, F., Türetken, E., Fua, P.: Multiple Object Tracking Using K-Shortest Paths Optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(11), 1806–1819 (2011)
4. Butt, A., Collins, R.: Multi-Target Tracking by Lagrangian Relaxation to Min-Cost Network Flow. In: Conference on Computer Vision and Pattern Recognition. pp. 1846–1853 (2013)
5. Cao, X., Wang, Z., Zhao, Y., Su, F.: Scale Aggregation Network for Accurate and Efficient Crowd Counting. In: European Conference on Computer Vision (2018)
6. Chan, A., Liang, Z., Vasconcelos, N.: Privacy Preserving Crowd Monitoring: Counting People Without People Models or Tracking. In: Conference on Computer Vision and Pattern Recognition (2008)
7. Chan, A., Vasconcelos, N.: Bayesian Poisson Regression for Crowd Counting. In: International Conference on Computer Vision. pp. 545–551 (2009)
8. Cheng, Z., Li, J., Dai, Q., Wu, X., Hauptmann, A.G.: Learning Spatial Awareness to Improve Crowd Counting. In: International Conference on Computer Vision (2019)
9. Collins, R.: Multitarget Data Association with Higher-Order Motion Models. In: Conference on Computer Vision and Pattern Recognition (2012)
10. Dicle, C., Camps, O.I., Sznaier, M.: The Way They Move: Tracking Multiple Targets with Similar Appearance. In: International Conference on Computer Vision (2013)
11. Fang, Y., Zhan, B., Cai, W., Gao, S., Hu, B.: Locality-Constrained Spatial Transformer Network for Video Crowd Counting. International Conference on Multimedia and Expo (2019)
12. He, Z., Li, X., You, X., Tao, D., Tang, Y.Y.: Connected Component Model for Multi-Object Tracking. IEEE Transactions on Image Processing **25**(8) (2016)
13. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation **9**(8), 1735–1780 (1997)
14. Idrees, H., Tayyab, M., Athrey, K., Zhang, D., Al-Maadeed, S., Rajpoot, N., Shah, M.: Composition Loss for Counting, Density Map Estimation and Localization in Dense Crowds. In: European Conference on Computer Vision (2018)
15. Jiang, X., Xiao, Z., Zhang, B., Zhen, X.: Crowd Counting and Density Estimation by Trellis Encoder-Decoder Networks. In: Conference on Computer Vision and Pattern Recognition (2019)
16. Lempitsky, V., Zisserman, A.: Learning to Count Objects in Images. In: Advances in Neural Information Processing Systems (2010)
17. Li, Y., Zhang, X., Chen, D.: CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes. In: Conference on Computer Vision and Pattern Recognition (2018)
18. Lian, D., Li, J., Zheng, J., Luo, W., Gao, S.: Density Map Regression Guided Detection Network for RGB-D Crowd Counting and Localization. In: Conference on Computer Vision and Pattern Recognition (2019)

19. Liu, C., Weng, X., Mu, Y.: Recurrent Attentive Zooming for Joint Crowd Counting and Precise Localization. In: Conference on Computer Vision and Pattern Recognition (2019)
20. Liu, J., Gao, C., Meng, D., Hauptmann1, A.: Decidenet: Counting Varying Density Crowds through Attention Guided Detection and Density Estimation. In: Conference on Computer Vision and Pattern Recognition (2018)
21. Liu, L., Qiu, Z., Li, G., Liu, S., Ouyang, W., Lin, L.: Crowd Counting with Deep Structured Scale Integration Network. In: International Conference on Computer Vision (2019)
22. Liu, L., Wang, H., Li, G., Ouyang, W., Lin, L.: Crowd Counting Using Deep Recurrent Spatial-Aware Network. In: International Joint Conference on Artificial Intelligence (2018)
23. Liu, N., Long, Y., Zou, C., Niu, Q., Pan, L., Wu, H.: Adcrowdnet: An Attention-Injective Deformable Convolutional Network for Crowd Understanding. In: Conference on Computer Vision and Pattern Recognition (2019)
24. Liu, W., Lis, K., Salzmann, M., Fua, P.: Geometric and Physical Constraints for Drone-Based Head Plane Crowd Density Estimation. International Conference on Intelligent Robots and Systems (2019)
25. Liu, W., Salzmann, M., Fua, P.: Context-Aware Crowd Counting. In: Conference on Computer Vision and Pattern Recognition (2019)
26. Liu, X., Weijer, J., Bagdanov, A.: Leveraging Unlabeled Data for Crowd Counting by Learning to Rank. In: Conference on Computer Vision and Pattern Recognition (2018)
27. Liu, Y., Shi, M., Zhao, Q., Wang, X.: Point In, Box Out: Beyond Counting Persons in Crowds. In: Conference on Computer Vision and Pattern Recognition (2019)
28. Long, J., Shelhamer, E., Darrell, T.: Fully Convolutional Networks for Semantic Segmentation. In: Conference on Computer Vision and Pattern Recognition (2015)
29. Ma, Z., Wei, X., Hong, X., Gong, Y.: Bayesian Loss for Crowd Count Estimation with Point Supervision. In: International Conference on Computer Vision (2019)
30. Onoro-Rubio, D., López-Sastre, R.: Towards Perspective-Free Object Counting with Deep Learning. In: European Conference on Computer Vision. pp. 615–629 (2016)
31. Pirsiavash, H., Ramanan, D., Fowlkes, C.: Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects. In: Conference on Computer Vision and Pattern Recognition. pp. 1201–1208 (June 2011)
32. Ranjan, V., Le, H., Hoai, M.: Iterative Crowd Counting. In: European Conference on Computer Vision (2018)
33. Sam, D., Sajjan, N., Babu, R., M., S.: Divide and Grow: Capturing Huge Diversity in Crowd Images with Incrementally Growing CNN. In: Conference on Computer Vision and Pattern Recognition (2018)
34. Sam, D., Surya, S., Babu, R.: Switching Convolutional Neural Network for Crowd Counting. In: Conference on Computer Vision and Pattern Recognition. p. 6 (2017)
35. Schröder, G., Senst, T., Bochinski, E., Sikora, T.: Optical Flow Dataset and Benchmark for Visual Crowd Analysis. In: International Conference on Advanced Video and Signal Based Surveillance (2018)
36. Shen, Z., Xu, Y., Ni, B., Wang, M., Hu, J., Yang, X.: Crowd Counting via Adversarial Cross-Scale Consistency Pursuit. In: Conference on Computer Vision and Pattern Recognition (2018)
37. Shi, M., Yang, Z., Xu, C., Chen, Q.: Revisiting Perspective Information for Efficient Crowd Counting. In: Conference on Computer Vision and Pattern Recognition (2019)

38. Shi, X., Chen, Z., Wang, H., Yeung, D., Wong, W., Woo, W.: Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In: Advances in Neural Information Processing Systems. pp. 802–810 (2015)
39. Shi, Z., Mettes, P., Snoek, C.G.M.: Counting with Focus for Free. In: International Conference on Computer Vision (2019)
40. Shi, Z., Zhang, L., Liu, Y., Cao, X.: Crowd Counting with Deep Negative Correlation Learning. In: Conference on Computer Vision and Pattern Recognition (2018)
41. Sindagi, V., Patel, V.: Generating High-Quality Crowd Density Maps Using Contextual Pyramid CNNs. In: International Conference on Computer Vision. pp. 1879–1888 (2017)
42. Sindagi, V., Patel, V.: Multi-Level Bottom-Top and Top-Bottom Feature Fusion for Crowd Counting. In: International Conference on Computer Vision (2019)
43. Sun, D., Yang, X., Liu, M., Kautz, J.: Pwc-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In: Conference on Computer Vision and Pattern Recognition (2018)
44. Suurballe, J.: Disjoint Paths in a Network. Networks (1974)
45. Walach, E., Wolf, L.: Learning to Count with CNN Boosting. In: European Conference on Computer Vision (2016)
46. Wan, J., Chan, A.B.: Adaptive Density Map Generation for Crowd Counting. In: International Conference on Computer Vision (2019)
47. Wan, J., Luo, W., Wu, B., Chan, A.B., Liu, W.: Residual Regression with Semantic Prior for Crowd Counting. In: Conference on Computer Vision and Pattern Recognition (2019)
48. Wang, Q., Gao, J., Lin, W., Yuan, Y.: Learning from Synthetic Data for Crowd Counting in the Wild. In: Conference on Computer Vision and Pattern Recognition (2019)
49. Xiong, F., Shi, X., Yeung, D.: Spatiotemporal Modeling for Crowd Counting in Videos. In: International Conference on Computer Vision. pp. 5161–5169 (2017)
50. Xiong, H., Lu, H., Liu, C., Liu, L., Cao, Z., Shen, C.: From Open Set to Closed Set: Counting Objects by Spatial Divide-And-Conquer. In: International Conference on Computer Vision (2019)
51. Xu, C., Qiu, K., Fu, J., Bai, S., Xu, Y., Bai, X.: Learn to Scale: Generating Multipolar Normalized Density Maps for Crowd Counting. In: International Conference on Computer Vision (2019)
52. Yan, Z., Yuan, Y., Zuo, W., Tan, X., Wang, Y., Wen, S., Ding, E.: Perspective-Guided Convolution Networks for Crowd Counting. In: International Conference on Computer Vision (2019)
53. Zhang, A., Shen, J., Xiao, Z., Zhu, F., Zhen, X., Cao, X., Shao, L.: Relational Attention Network for Crowd Counting. In: International Conference on Computer Vision (2019)
54. Zhang, A., Yue, L., Shen, J., Zhu, F., Zhen, X., Cao, X., Shao, L.: Attentional Neural Fields for Crowd Counting. In: International Conference on Computer Vision (2019)
55. Zhang, C., Li, H., Wang, X., Yang, X.: Cross-Scene Crowd Counting via Deep Convolutional Neural Networks. In: Conference on Computer Vision and Pattern Recognition. pp. 833–841 (2015)
56. Zhang, L., Li, Y., Nevatia, R.: Global Data Association for Multi-Object Tracking Using Network Flows. In: Conference on Computer Vision and Pattern Recognition (2008)

57. Zhang, Q., Chan, A.B.: Wide-Area Crowd Counting via Ground-Plane Density Maps and Multi-View Fusion CNNs. In: Conference on Computer Vision and Pattern Recognition (2019)
58. Zhang, S., Wu, G., Costeira, J., Moura, J.: FCN-rLSTM: Deep Spatio-Temporal Neural Networks for Vehicle Counting in City Cameras. In: International Conference on Computer Vision (2017)
59. Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-Image Crowd Counting via Multi-Column Convolutional Neural Network. In: Conference on Computer Vision and Pattern Recognition. pp. 589–597 (2016)
60. Zhao, M., Zhang, J., Zhang, C., Zhang, W.: Leveraging Heterogeneous Auxiliary Tasks to Assist Crowd Counting. In: Conference on Computer Vision and Pattern Recognition (2019)