Wide-Area Crowd Counting via Ground-Plane Density Maps and Multi-View Fusion CNNs

Qi Zhang Antoni B. Chan
Department of Computer Science, City University of Hong Kong
qzhang364-c@my.cityu.edu.hk, abchan@cityu.edu.hk

Abstract

Crowd counting in single-view images has achieved outstanding performance on existing counting datasets. However, single-view counting is not applicable to large and wide scenes (e.g., public parks, long subway platforms, or event spaces) because a single camera cannot capture the whole scene in adequate detail for counting, e.g., when the scene is too large to fit into the field-of-view of the camera, too long so that the resolution is too low on faraway crowds, or when there are too many large objects that occlude large portions of the crowd. Therefore, to solve the wide-area counting task requires multiple cameras with overlapping fields-of-view. In this paper, we propose a deep neural network framework for multi-view crowd counting, which fuses information from multiple camera views to predict a scene-level density map on the ground-plane of the 3D world. We consider 3 versions of the fusion framework: the late fusion model fuses camera-view density maps; the naïve early fusion model fuses camera-view feature maps; and the multi-view multi-scale early fusion model favors that features aligned to the same ground-plane point have consistent scales. We test our 3 fusion models on 3 multi-view counting datasets, PETS2009, DukeMTMC, and a newly collected multi-view counting dataset containing a crowded street intersection. Our methods achieve state-of-the-art results compared to other multi-view counting baselines.

1. Introduction

Crowd counting aims to estimate the number of the people in images or videos. It has a wide range of real-world applications, such as crowd management, public safety, traffic monitoring or urban planning [43]. For example, crowd counting can detect overcrowding on the railway platform and help with the train schedule planning. Furthermore, the estimated crowd density map provides spatial information of the crowd, which can benefit other tasks, such as human detection [8, 18, 27] and tracking [18, 34, 36].

Recently, with the strong learning ability of deep neural networks (DNNs), density map based crowd counting methods have achieved outstanding performance on the existing counting datasets [1, 12, 42], where the goal is to count the crowd in a single image. However, a single image view is not adequate to cover a *large* and *wide* scene, such as a large park or a long train platform. For these wide-area scenes, a single camera view cannot capture the whole scene in adequate detail for counting, either because the the scene is too large (wide) to fit within the field-of-view of the camera, or the scene is too long so that the resolution is too low in faraway regions. Furthermore, a single view cannot count regions that are still within the scene, but are totally occluded by large objects (e.g., trees, large vehicles, building structures). Therefore, to solve the wide-area counting task requires multiple camera views with overlapping field-of-views, which combined can cover the whole scene and can see around occlusions. The goal of wide-area counting is then to use multiple camera views to estimate the crowd count of the whole scene.

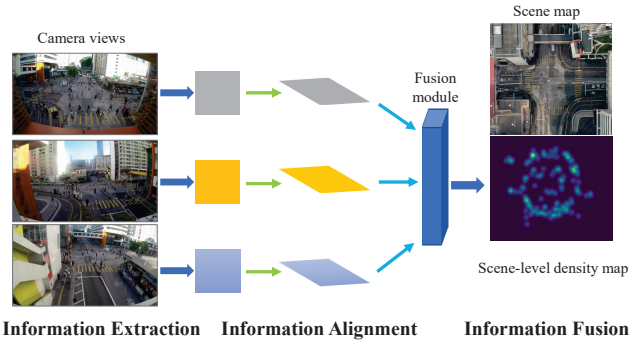

Figure 1: The pipeline of the proposed multi-view fusion framework. Feature maps are extracted from multiple camera views, aligned on the ground-plane, and fused to obtain the scene-level ground-plane density map. The scene map is shown for reference.

Existing multi-view counting methods rely on foreground extraction techniques and hand-crafted features. Their crowd counting performance is limited by the effectiveness of the foreground extraction, as well as the representation ability of hand-crafted features. Considering the strong learning power of DNNs as well as the perfor-

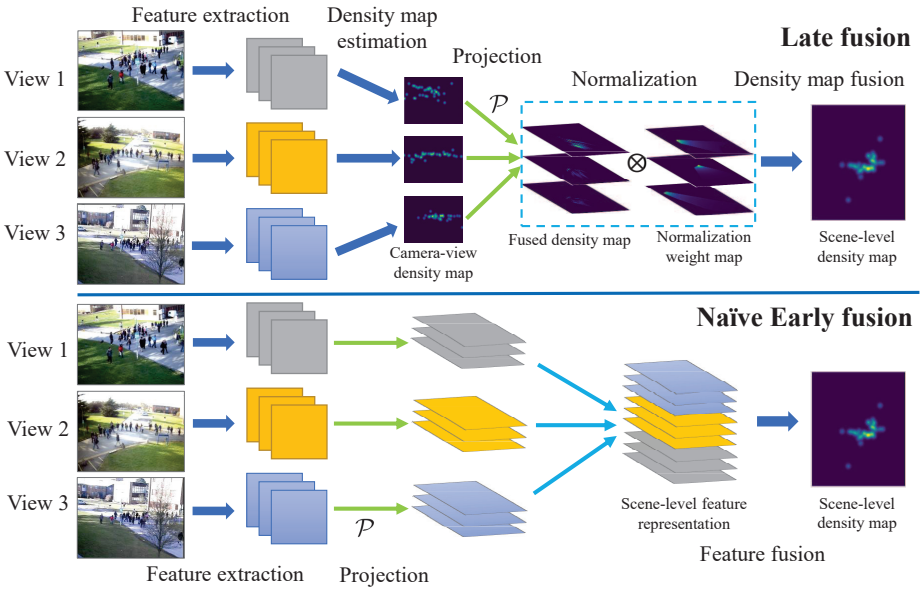

Figure 2: The pipeline of our late fusion model and naïve early fusion model for multi-view counting. In the late fusion model, single-view density maps are fused. In the naïve early fusion model, single-view feature maps are fused.

mance progress of single view counting methods using density maps, the feasibility of end-to-end DNN-based multi-view counting methods should be explored.

In this paper, we propose a DNN-based multi-view counting method that extracts information from each camera view and then fuses them together to estimate a scene-level ground-plane density map (see Fig. 1). The method consists of 3 stages: 1) *Information extraction* – single view feature maps are extracted from each camera image with DNNs; 2) *Information alignment* – using the camera geometry, the feature maps from all cameras are projected onto the ground-plane in the 3D world so that the same person's features are approximately aligned across multiple views, and properly normalized to remove projection effects; 3) *Information fusion* – the aligned single-view feature maps are fused together and used to predict the scene-level ground-plane density map.

We propose three versions of our multi-view framework that differ in the kind of information that is fused. First, in our late-fusion model (see Fig. 2 top), view-level density maps are predicted for each camera view, projected to the ground-plane, and fused for estimating the scene-level density map. We also propose a post-projection normalization method that removes the projection effect that distorts the sum of the density maps (and thus the count). Second, in our naïve early-fusion model (see Fig. 2 bottom), convolutional feature maps are extracted from each camera view, projected to the ground-plane and fused to predict the scene-level density map. Third, to handle the scale variations of the same person across camera views, our multi-view multi-scale (MVMS) early-fusion model (see Fig. 5) extracts features with consistent scale across corresponding locations in the camera views before applying projection and fusion. We consider 2 approaches for selecting the suitable scales, based on distances computed from the camera geometry.

The existing multi-view datasets that can be used for multi-view counting are PETS2009 [9] and DukeMTMC [35]. However, PETS2009 is not a wide-area scene as it focuses on one walkway, while DukeMTMC is a wide-area scene but does not contain large crowds. To address these shortcomings, we collect a new wide-area dataset from a busy street intersection, which contains large crowds, more occlusion patterns (e.g., busses and cars), and large scale variations. This new dataset more effectively tests multi-view crowd counting in a real-world scene.

In summary, the main contributions of the paper are:

- We propose an end-to-end trainable DNN-based multi-view crowd counting framework, which fuses information from multiple camera views to obtain a scene-level density map. To the best of our knowledge, this is the first study of scene-level density map estimation for multi-view counting.
- We propose 3 fusion models based on our multi-view framework (late fusion, naïve early fusion, and multi-view multi-scale early fusion), which achieve better counting accuracy compared to baseline methods.
- We collect a real-world wide-area counting dataset consisting of multiple camera views, which will advance research on multi-view wide-area counting.

2. Related Work

We briefly review methods of crowd counting from single-view and multi-view cameras.

2.1. Single-view counting

Traditional methods. Traditional single-view counting methods can be divided into 3 categories [4, 43]: detection, regression, and density map methods. Detection methods try to detect each person in the images by extracting hand-crafted features [38, 45, 48] and then training a classifier [10, 14, 46] using the extracted features. However, the detection methods do not perform well when the people are heavily occluded, which limits their application scenarios. Regression methods extract the image features [2, 6, 15, 19] and learns a mapping directly to the crowd count [3, 5, 29, 31]. But their performance is limited by the weak representation power of the hand-crafted low-level features. Instead of directly obtaining the counting number, [21] proposed to estimate density maps, where each pixel in the image contains the local crowd density, and the count is obtained by summing over the density map. Traditional density map methods learn the mapping between the hand-crafted local features and the density maps [21, 32, 47, 49].

DNN-based methods. Crowd counting with DNNs has mainly focused on density map estimation. The first networks used a standard CNN [50] to directly estimate the density map from an image. Scale variation is a critical issue in crowd counting, due to perspective effects in the image. [51] proposed the multi-column CNN (MCNN) consisting of 3 columns of different receptive field sizes, which can model people of different scales. [39] added a switch module in the MCNN structure to choose the optimal column to match the scale of each patch. [30] proposed to use the patch pyramid as input to extract multi-scale features. Similarly, [16] used an image pyramid with a scale-selecting attention block to adaptively fuse predictions on different scales. Recently, more sophisticated network structures have been proposed to advance the counting performance [1, 23, 25, 40, 41]. [42] incorporated global and local context information in the crowd counting framework, and proposed the contextual pyramid CNN (CP-CNN). [17] proposed an adaptive convolution neural network (ACNN) that uses side information (camera angle and height) to include context into the counting framework. [1] proposed the scale aggregation module to extract multi-scale features and generated high-resolution density maps by using a set of transposed convolutions.

All these methods are using DNNs to estimate a density map on the image plane of a single camera-view, with different architectures improving the performance across scenes and views. In contrast, in this paper, we focus on fusing multiple camera views of the same scene to obtain a ground-plane density map in the 3D world.

2.2. Multi-view counting

Existing multi-view counting methods can be divided into 3 categories: detection/tracking, regression, 3D cylinder methods. The detection/tracking methods first perform detection or tracking on each scene and obtain single-view detection results. Then, the detection results from each view are integrated by projecting the single-view results to a common coordinate system, e.g., the ground plane or a reference view. The count of the scene is obtained by solving a correspondence problem [7, 22, 26, 28]. Regression based methods first extract foreground segments from each view, then build the mapping relationship of the segments and the count number with a regression model [37, 44]. 3D cylinder-based methods try to find the people's locations in the 3D scene by minimizing the gap between the people's 3D positions projected into the camera view and the single view detection [11].

These multi-view counting methods are mainly based on hand-crafted low-level features and regression or detection/tracking frameworks. Regression-based methods only give the global count, while detection/tracking methods cannot cope well with occlusions when the scene is very crowded. In contrast to these works, our approach is based on predicting the ground-plane density map in the 3D world by fusing the information across camera views using DNNs. Two advantages of our approach are the abilities to learn the feature extractors and fusion stage in end-to-end training, and to estimate the spatial arrangement of the crowd on the ground plane. While the previous methods are mainly tested on PETS2009, which only contains low/moderate crowd numbers on a walkway, here we test on a newly collected dataset comprising a real-world scene of a street intersection with large crowd numbers, vehicles, and occlusions.

3. Multi-view counting via multi-view fusion

For multi-view counting, we assume that the cameras are fixed, the camera calibration parameters (both intrinsic and extrinsic) are known, and that the camera frames across views are synchronized. Given the set of multi-view images, the goal is to predict a scene-level density map defined on the ground-plane of the 3D scene (see Fig. 1). The ground-truth ground-plane density map is obtained in a similar way as the traditional camera-view density map – the ground-plane annotation map is obtained using the ground-truth 3D coordinates of the people, which is then convolved by a fixed-width Gaussian to obtain the density map.

In this section, we propose three fusion approaches for multi-view counting: 1) the *late fusion* model projects camera-view density maps onto the ground plane and then fuses them together, and requires a projection normalization step; 2) the *naïve early fusion* model projects camera-view feature maps onto the ground plane then fuses them; 3) to handle inter-view and intra-view scale variations, the *multi-view multi-scale early fusion* model (MVMS) selects features scales to be consistent across views when projecting to the same ground-plane point. We first present the com-

FCN-7	
Layer	Filter
conv 1	$16\times1\times5\times5$
conv 2	$16\times16\times5\times5$
pooling	2×2
conv 3	$32\times16\times5\times5$
conv 4	$32\times32\times5\times5$
pooling	2×2
conv 5	$64\times32\times5\times5$
conv 6	$32\times64\times5\times5$
conv 7	$1\times32\times5\times5$

Fusion	
Layer	Filter
concat	-
conv 1	$64\times n\times5\times5$
conv 2	$32\times64\times5\times5$
conv 3	$1\times32\times5\times5$

Table 1: FCN-7 backbone and fusion module. The Filter dimensions are output channels, input channels, and filter size ($w\times h$).

mon components, and then the 3 fusion models.

3.1. Backbone FCN for camera views

A fully-convolutional network (denoted as FCN-7) is used on each camera view to extract image feature maps or estimate a corresponding view-level density map. The FCN-7 settings are shown in Table 1. Although more complex DNNs, e.g., [39, 42, 51], could be applied to the camera-views, in this paper, we mainly focus on how to effectively fuse multi-view information to perform wide-area crowd counting, and thus using FCN-7 suffices.

3.2. Camera-view to scene projection

As we assume that the intrinsic and extrinsic parameters of the cameras are known, the projection from a camera's 2D image space to a 3D scene-level representation can be implemented as a differentiable fixed-transformation module (see Fig. 3). The 3D height (z-coordinate) corresponding to each image pixel is unknown. Since the view-level density maps are based on head annotations and the head is typically visible even during partial occlusion, we assume that each pixel's height in the 3D world is a person's average height (1750 mm). The camera parameters together with the height assumption are used to calculate the correspondence mapping $\mathcal{P}$ between 2D image coordinates and the 3D coordinates on the 3D average-height plane. Finally, the Sampler from the Spatial Transformer Networks [13] is used to implement the projection, resulting in the scene-level representation of the input map.

3.3. Late fusion model

The main idea of the late fusion model is to first estimate the density maps in each camera view, and then fuse them together to obtain the scene-level density map. In particular, the late fusion model consists of 3 stages (see Fig. 2 top): 1) estimating the camera-view density maps using FCN-7 on each view; 2) projecting the density maps to the ground-plane representation using the projection module; 3) concatenating the projected density maps channel-wise and then applying the Fusion module to obtain the scene-level density map. The network settings for the fusion network

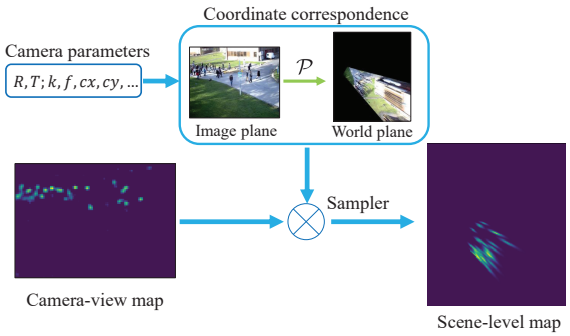

Figure 3: The projection module to transform camera-view maps to a scene-level representation. Here the camera-view map is visualized as a density map.

are presented in Table 1.

Projection Normalization. One problem is that the density map is stretched during the projection step, and thus the sum of the density map changes after the projection. Considering that the density map is composed of a sum of Gaussian kernels, each Gaussian is stretched differently depending on its location in the image plane. To address this problem, we propose a normalization method to ensure that the sum of each Gaussian kernel remains the same after projection (see Fig. 4). In particular, let (x_0, y_0) and (x, y) be the corresponding points in the image plane and the 3D world ground-plane representation. The normalization weight w_{xy} for ground-plane position (x, y) is

$$w_{xy} = \frac{\sum_{ij} D_{x_0,y_0}(i,j)}{\sum_{mn} \mathcal{P}(D_{x_0,y_0}(m,n))}, \tag{1}$$

where D_{x_0,y_0} denotes an image-space density map containing only one Gaussian kernel centered at (x_0, y_0), $\mathcal{P}$ is the projection operation from image space to ground plane, and (i, j) and (m, n) are the image coordinates and ground-plane coordinates, respectively. The normalization map $W = [w_{xy}]$ for each camera is element-wise multiplied to the corresponding projected density map before concatenation. As illustrated in Fig. 4, after normalization, the summation of the projected density map remains similar to that of the original view-level density map.

3.4. Naïve early fusion

The naïve early fusion model directly fuses the feature maps from all the camera-views to estimate the ground-plane density map. Similar to the late fusion model, we implement the early fusion model by replacing the density map-level fusion with feature-level fusion (see Fig. 2 bottom). Specifically, the naïve early fusion model consists of 3 stages: 1) extracting feature maps from each camera view using the first 4 convolution layers of FCN-7; 2) projecting the image feature maps to the ground-plane representation using the projection module; 3) concatenating the projected feature maps and applying the Fusion module to estimate

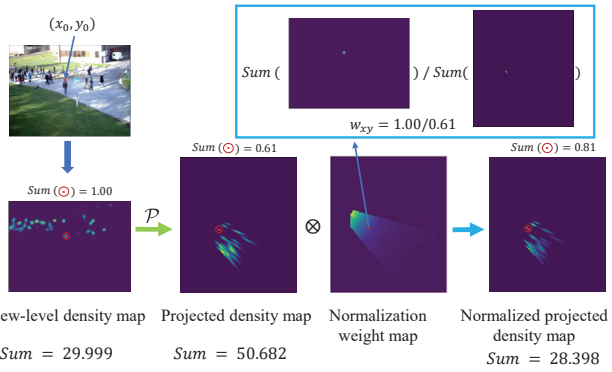

Figure 4: The projection normalization process for the late fusion model. *Sum* is the sum of the whole density map, while *Sum*(⊙) is the sum over the circled region.

the scene-level density map. Note that the projection normalization step used in the late fusion model is not required for the early fusion model, since feature maps do not have the same interpretation of summation yielding a count.

3.5. Multi-view multi-scale early fusion

Intra-view scale variations are an important issue in single-view counting, as people will appear with different sizes in the image due to perspective effects. Using multiple views increases the severity of the scale variation issue; in addition to intra-view scale variation, multi-view images have inter-view scale variations, where the same person will appear at different scales across multiple views. This inter-view scale variation may cause problems during the fusion stage as there are a combinatorial number of possible scales appearing across all views, which the network needs to be invariant to. To address this problem, we instead extract feature maps at multiple scales, and then perform scale selection so that the projected features are at consistent scales across all views (i.e., a given person appears at the same scale across all views).

Our proposed multi-view multi-scale (MVMS) early fusion architecture is shown in Fig. 5. The MVMS fusion model consists of 4 stages: 1) extracting multi-scale feature maps by applying the first 4 convolution layers of FCN-7 on an image pyramid for each camera view; 2) upsampling all the feature maps to the largest size, and then selecting the scales for each pixel in each camera-view according to the scene geometry; 3) projecting the scale-consistent feature maps to the ground-plane representation using the projection module; 4) fusing the projected features and predicting a scene-level density map using the fusion module. We consider 2 strategies for selecting the consistent scales, fixed scale selection and learnable scale selection.

Fixed scale-selection. The fixed scale selection strategy is illustrated in Fig. 5 (right-top). For a given camera, let $\{F_0, \cdots, F_n\}$ be the set of feature maps extracted from the image pyramid, and then upsampled to the same size. Here F_0 is the original scale and F_n is the smallest scale. A distance map is computed for the camera-view, where $d(x_0, y_0)$ is the distance between the camera's 3D location and the projection of the point (x_0, y_0) into the 3D-world (on the average height plane). A scale selection map S, where each value corresponds to the selected scale for that pixel, is computed using the distance map,

$$S(x_0, y_0) = s_r - \lfloor \log_z \frac{d(x_0, y_0)}{d_r} \rfloor, \tag{2}$$

where z is the zoom factor between neighboring scales in the image pyramid, and $\lfloor \cdot \rfloor$ is the floor function. d_r and s_r are the reference distance and the corresponding reference scale number, which are the same for all camera-views. In our experiments, we set the reference distance d_r as the distance value for the center pixel of the first view, and s_r as the middle scale of the image pyramid. Given the scale selection map S, the feature maps across scales are merged into a single feature map, $F = \sum_i \mathbb{1}(S = i) \otimes F_i$, where $\otimes$ is element-wise multiplication, and $\mathbb{1}$ is an element-wise indicator function.

Learnable scale-selection: The fixed scale selection strategy requires setting the reference distance and reference scale parameters. To make the scale selection process more adaptive to the view context, a learnable scale-selection model is considered (Fig. 5 (right-down)),

$$S(x_0, y_0) = b + k \log_z \frac{d(x_0, y_0)}{d_r}, \tag{3}$$

where the learnable parameter b corresponds to the reference scale, and k adjusts the reference distance. The learnable scale selection can be implemented as a 1×1 convolution on the log distance map. Then, a soft scale selection mask M_i for scale i can be obtained,

$$M_i(x_0, y_0) = \frac{e^{-(S(x_0, y_0) - i)^2}}{\sum_{j=0}^{n} e^{-(S(x_0, y_0) - j)^2}}. \tag{4}$$

The scale consistent feature map is then $F = \sum_i M_i \otimes F_i$.

3.6. Training details

A two-stage process is applied to train the model. At the first stage, the single-view density maps together with the scene-level density maps are used as supervisory information. Each single-view FCN-7 backbone is trained using the camera-view images and the corresponding single-view density maps. The learning rate is set to 1e-4. In the second stage, the supervisory information of the single-view density maps is removed. FCN-7 (either density map estimator or feature extractor) is fixed and the fusion and scale selection parts are trained. The loss function is the pixel-wise squared error between the ground-truth and predicted density maps. The learning rate is set to 1e-4, and decreases

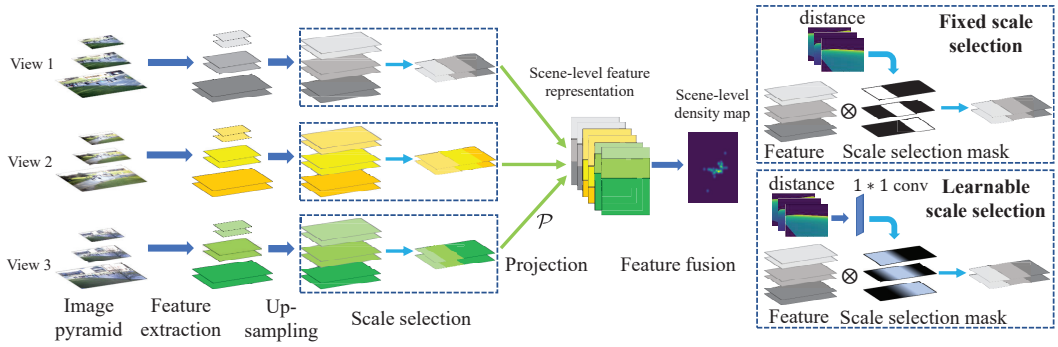

Figure 5: The pipeline of multi-view multi-scale (MVMS) early fusion model. First, multi-scale feature maps are extracted with an image pyramid. The multi-scale feature maps are up-sampled to the same size. The scale selection module (the dotted box) favors the scales of features that represent the same ground-plane point are consistent across all views. The scale-consistent features are projected to the average-height plane and then fused to obtain the scene-level density map. Two kinds of scale selection strategies (the two dotted boxes on the right) are utilized: the fixed scale selection uses the distance information relative to a reference distance, and learnable scale selection makes the reference distance a learnable parameter.

to 5e-5 during training. After training the two stages, the model is fine-tuned end-to-end. The training batch-size is set to 1 in all experiments.

4. Experiments

In this section we present our experiments on multi-view crowd counting using DNNs.

4.1. Datasets

We test on two existing datasets, PETS2009 and DukeMTMC, and our newly collected City Street dataset. Table 2 provides a summary, and Fig. 6 shows examples.

PETS2009: PETS2009 [9] is a multi-view sequence dataset containing crowd activities from 8 views. The first 3 views are used for the experiments, as the other 5 views have low camera angle, poor image quality, or unstable frame rate. To balance the crowd levels, we use sequences S1L3 (14_17, 14_33), S2L2 (14_55) and S2L3 (14_41) for training (1105 images in total), and S1L1 (13_57, 13_59), S1L2 (14_06, 14_31) for testing (794 images). The calibration parameters (extrinsic and intrinsic) for the cameras are provided with the dataset. To obtain the annotations across all views, we use the View 1 annotations provided by [20] and project them to other views followed by manual annotations to get all the people heads in the images.

DukeMTMC: DukeMTMC [35] is a multi-view video dataset for multi-view tracking, human detection or ReID. The multi-view video dataset has video from 8 synchronized cameras for 85 minutes with 1080p resolution at 60 fps. For our counting experiments, we use 4 cameras (cameras 2, 3, 5 and 8) that have overlapping fields-of-view. The synchronized videos are sampled every 3 seconds, resulting in 989 multi-view images. The first 700 images are used for training and the remaining 289 for testing. Camera extrinsic and homography parameters are provided by the dataset. In the original dataset, annotations for each view are only provided in the view ROIs, which are all non-overlapping on the ground-plane. Since we are interested in overlapping cameras, we project the annotations from each camera view to the overlapping areas in all other views. Region R2 (see Fig. 6) is excluded during the experiment, since there are no annotations provided there.

Dataset	resolution	view	train / test	crowd
PETS2009 [9]	768×576	3	1105 / 794	20-40
DukeMTMC [35]	1920×1080	4	700 / 289	10-30
City Street	2704×1520	3	300 / 200	70-150

Table 2: The comparison of three multi-view datasets.

City Street: We collected a multi-view video dataset of a busy city street using 5 synchronized cameras. The videos are about 1 hour long with 2.7k (2704×1520) resolution at 30 fps. We select Cameras 1, 3 and 4 for the experiment (see Fig. 6 bottom). The cameras' intrinsic and extrinsic parameters are estimated using the calibration algorithm from [52]. 500 multi-view images are uniformly sampled from the videos, and the first 300 are used for training and remaining 200 for testing. The ground-truth 2D and 3D annotations are obtained as follows. The head positions of the first camera-view are annotated manually, and then projected to other views and adjusted manually. Next, for the second camera view, new people (not seen in the first view), are also annotated and then projected to the other views. This process is repeated until all people in the scene are annotated and associated across all camera views. Our dataset has larger crowd numbers (70-150), compared with PETS (20-40) and DukeMTMC (10-30). Our new dataset also contains more crowd scale variations and occlusions due to vehicles and fixed structures.

Experiment settings: The image resolutions ($w\times h$) used in the experiments are: 384×288 for PETS2009, 640×360 for DukeMTMC, and 676×380 for City Street. The resolutions of the scene-level ground-plane density maps are:

Figure 6: Examples from 3 multi-view counting datasets. The first column shows the camera frames and annotations. The second column shows the camera layout and scene-level ground-plane density maps.

152×177 for PETS2009, 160×120 for DukeMTMC and 160×192 for City Street. For the detection baseline, the original image resolutions are used (Faster-RCNN will resize the images).

4.2. Experiment setup

Methods: We test our 3 multi-view fusion models, denoted as "Late fusion", "Naïve early fusion", and "MVMS" (multi-view multi-scale early fusion). The late fusion model uses projection normalization. MVMS uses learnable scale selection, and a 3-scale image pyramid with zoom factor of 0.5. These settings will be tested later in the ablation study.

For comparisons, we test two baseline methods. The first baseline is a simple approach to fusing camera-view density maps into a scene-level count, denoted as "Dmap weighted", which is an adaptation from [37]. First FCN-7 is applied to get the density map D_i for each camera-view. The density maps are then fused into a scene-level count using a weight map W_i for each view,

$$C = \sum_i \sum_{x_0, y_0} W_i(x_0, y_0) D_i(x_0, y_0), \tag{5}$$

where the summations are over the camera-views and the image pixels. The weight map W_i is constructed based on how many views can see a particular pixel. In other words, $W_i(x_0, y_0) = 1/t$, where t is the number of views that can see the projected point $\mathcal{P}(x_0, y_0)$. Note that [37] used this simple fusion approach with traditional regression-based counting (in their setting, the D_i map is based on the predicted counts for crowd blobs). Here, we are using recent DNN-based methods and crowd density maps, which outperform traditional regression-based counting, and hence form a stronger baseline method compared to [37].

The second baseline is using human detection methods and person re-identification (ReID), denoted as "Detection + ReID". First, Faster-RCNN [33] is used to detect humans in each camera-view. Next, the scene geometry constraints and the ReID method LOMO 2015 [24] are used to associate the same people across views. Specifically, each detection box's top-center point in one view is projected to other views, and ReID is performed between the original detection box and detection boxes near the projected point in other views. Finally, the scene-level people count is obtained by counting the number of unique people among the detection boxes in all views. The bounding boxes needed for training are created with the head annotations and the perspective map of each view.

Evaluation: The mean absolute error (MAE) is used to evaluate multi-view counting performance, comparing the scene-level predicted counts and the ground-truth scene-level counts. In addition, we also evaluate the MAE of the predicted counts in each camera-view. The ground-truth count for each camera-view is obtained by summing the ground-truth scene-level density map over the region covered by the camera's field-of-view. Note that people that are totally occluded from the camera, but still within its field-of-view, are still counted.

4.3. Experiment results

The experimental results are shown in Table 3. On PETS2009, our 3 multi-view fusion models can achieve better results than the two comparison methods in terms of both single-view counting and scene-level counting. Detection+ReID performs worst on this dataset because the people are close together in a crowd, and occlusion causes severe misdetection. Among our three multi-view fusion models, naïve early fusion performs worse, which suggests that the scale variations in multi-view images limits the performance. Furthermore, MVMS performs much better

Dataset	PETS 2009 [9]				DukeMTMC [35]					City Street			
Camera	1	2	3	scene	2	3	5	8	scene	1	3	4	scene
Dmap weighted	3.37	5.59	5.84	7.51	0.62	0.91	0.98	1.41	2.12	10.16	12.55	21.56	11.10
Detection+ReID	8.60	11.19	14.61	9.41	2.06	**0.25**	0.96	3.58	2.20	41.38	32.94	28.57	27.60
Late fusion (ours)	2.62	3.17	3.97	3.92	**0.49**	0.77	**0.39**	**1.15**	1.27	8.14	7.72	8.08	8.12
Naïve early fusion (ours)	2.37	4.27	4.92	5.43	0.64	0.44	0.93	1.72	1.25	8.13	**7.62**	**7.89**	8.10
MVMS (ours)	**1.66**	**2.58**	**3.46**	**3.49**	0.63	0.52	0.94	1.36	**1.03**	**7.99**	7.63	7.91	**8.01**

Table 3: Experiment results: mean absolute error (MAE) on three multi-view counting datasets. "scene" denotes the scene-level counting error, while camera numbers denote to camera-view counting error. The late fusion model uses projection normalization, and MVMS uses learnable scale selection.

Dataset	PETS2009 [9]				DukeMTMC [35]					City Street			
Camera	1	2	3	scene	2	3	5	8	scene	1	3	4	scene
Late fusion (with)	**2.62**	**3.17**	**3.97**	**3.92**	**0.49**	0.77	**0.39**	**1.15**	**1.27**	**8.14**	**7.72**	**8.08**	**8.12**
Late fusion (without)	2.75	3.86	4.37	4.22	0.63	**0.73**	0.51	1.31	1.43	9.89	9.60	9.82	9.87
MVMS (fixed)	1.74	**2.57**	3.81	3.82	0.65	**0.46**	**0.88**	1.44	1.09	8.11	7.83	8.32	**7.80**
MVMS (learnable)	**1.66**	2.58	**3.46**	**3.49**	**0.63**	0.52	0.94	**1.36**	**1.03**	**7.99**	**7.63**	**7.91**	8.01

Table 4: Ablation study comparing the late fusion model with and without projection normalization, and MVMS with fixed or learnable scale selection.

than other models, which shows the multi-scale framework with scale selection strategies can improve the feature-level fusion to achieve better performance.

On DukeMTMC, our multi-view fusion models can achieve better performance than comparison methods at the scene-level and on most camera-views. Detection+ReID achieves the best result on camera 3 because this camera is almost parallel to the horizontal plane, has low people count, and rarely has occlusions, which is an ideal operating regime for the detector. Due to lower crowd numbers in DukeMTMC, the performance gap among the 3 fusion models is not large, but MVMS still performs best.

On City Street, our 3 multi-view fusion models achieve better results than the comparison methods. Compared to PETS2009, City Street has larger crowds and more occlusions and scale variations. Therefore, the performance of the baseline methods decreases a lot, especially Detection+ReID. Our MVMS model achieves better performance than all other models. Example results of scene-level density maps and counts can be found in the supplemental.

4.4. Ablation study

We perform an ablation study on the late fusion model with and without the projection normalization step, and the results are presented in Table 4 (top). Using projection normalization reduces the error of the late fusion model, compared to not using the normalization step.

We also perform an ablation study on the scale-selection strategy of MVMS, and the results are presented in Table 4 (bottom). Most of the time the learnable scale-selection strategy can achieve lower error than fixed scale-selection. We note that even using the fixed scale-selection strategy with MVMS still outperforms the naïve early fusion, which performs no scale selection. Thus obtaining features that have consistent scales across views is an important step when fusing the multi-view feature maps.

5. Conclusion

In this paper, we propose a DNN-based multi-view counting framework that fuses camera-views to predict scene-level ground-plane density maps. Both late fusion of density maps and early fusion of feature maps are studied. For late fusion, a projection normalization method is proposed to counter the effects of stretching caused by the projection operation. For early fusion, a multi-scale approach is proposed that selects features that have consistent scales across views. To advance research in multi-view counting, we collect a new dataset of large scene containing a street intersection with large crowds. Experiments show that our proposed multi-view counting framework can achieve better counting results than other methods.

In this paper, we have assumed that the cameras are fixed and camera parameters are known. Adapting our framework to moving cameras and unknown camera parameters (using the full spatial transformer net) is interesting future work. In addition, we have trained and tested the network on each dataset individually. Another interesting future direction is on *cross-scene* multi-view counting, where the scenes in the test set are distinct from those in the training set – however, this requires more multi-view scenes to be collected.

Acknowledgments

This work was supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. [T32-101/15-R] and CityU 11212518), and by a Strategic Research Grant from City University of Hong Kong (Project No. 7004887). We are grateful for the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

References

[1] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. Scale aggregation network for accurate and efficient crowd counting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018. 1, 3

[2] Antoni B. Chan, Zhang Sheng John Liang, and Nuno Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Computer Vision and Pattern Recognition*, pages 1–7, 2008. 3

[3] Antoni B Chan and Nuno Vasconcelos. Counting people with low-level features and bayesian regression. *IEEE Transactions on Image Processing*, 21(4):2160–2177, 2012. 3

[4] Change Loy Chen, Ke Chen, Shaogang Gong, and Tao Xiang. *Crowd Counting and Profiling: Methodology and Evaluation*. Springer New York, 2013. 3

[5] K. Chen, L. C. Chen, S. Gong, and T. Xiang. Feature mining for localised crowd counting. In *BMVC*, 2012. 3

[6] Zhongwei Cheng, Lei Qin, Qingming Huang, Shuicheng Yan, and Qi Tian. Recognizing human group action by layered model with multiple cues. *Neurocomputing*, 136:124–135, 2014. 3

[7] Fabio Dittrich, Luiz ES de Oliveira, Alceu S Britto Jr, and Alessandro L Koerich. People counting in crowded and outdoor scenes using a hybrid multi-camera approach. *arXiv preprint arXiv:1704.00326*, 2017. 3

[8] Volker Eiselein, Hajer Fradi, Ivo Keller, Thomas Sikora, and Jean-Luc Dugelay. Enhancing human detection using crowd density measures and an adaptive correction filter. In *10th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 19–24. IEEE, 2013. 1

[9] James Ferryman and Ali Shahrokni. Pets2009: Dataset and challenge. In *2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 1–6. IEEE, 2009. 2, 6, 8

[10] Juergen Gall, Angela Yao, Nima Razavi, Luc Van Gool, and Victor Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 33(11):2188–2202, 2011. 3

[11] Weina Ge and Robert T. Collins. Crowd detection with a multiview sampler. In *European Conference on Computer Vision*, pages 324–337, 2010. 3

[12] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 1

[13] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2017–2025, 2015. 4

[14] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998. 3

[15] Julio Cezar Silveira Jacques Junior, Soraia Raupp Musse, and Claudio Rosito Jung. Crowd analysis using computer vision techniques. *IEEE Signal Processing Magazine*, 27(5):66–77, 2010. 3

[16] Di Kang and Antoni Chan. Crowd counting by adaptively fusing predictions from an image pyramid. In *BMVC*, 2018. 3

[17] Di Kang, Debarun Dhar, and Antoni Chan. Incorporating side information by adaptive convolution. In *Advances in Neural Information Processing Systems*, pages 3867–3877, 2017. 3

[18] Di Kang, Zheng Ma, and Antoni B Chan. Beyond counting: comparisons of density maps for crowd analysis tasks-counting, detection, and tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018. 1

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 3

[20] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*, 2015. 6

[21] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *Advances in Neural Information Processing Systems*, pages 1324–1332, 2010. 3

[22] Jingwen Li, Lei Huang, and Changping Liu. People counting across multiple cameras for intelligent video surveillance. In *IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 178–183. IEEE, 2012. 3

[23] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1091–1100, 2018. 3

[24] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z Li. Person re-identification by local maximal occurrence representation and metric learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2197–2206, 2015. 7

[25] Jiang Liu, Chenqiang Gao, Deyu Meng, and Alexander G Hauptmann. Decidenet: Counting varying density crowds through attention guided detection and density estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2018. 3

[26] Huadong Ma, Chengbin Zeng, and Charles X Ling. A reliable people counting system via multiple cameras. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(2):31, 2012. 3

[27] Zheng Ma, Lei Yu, and Antoni B Chan. Small instance detection by integer programming on object density maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3689–3697, 2015. 1

[28] L. Maddalena, A. Petrosino, and F. Russo. People counting by learning their appearance in a multi-view camera environment. *Pattern Recognition Letters*, 36:125–134, 2014. 3

[29] AN Marana, L da F Costa, RA Lotufo, and SA Velastin. On the efficacy of texture analysis for crowd monitoring. In *International Symposium on Computer Graphics, Image Processing, and Vision*, pages 354–361. IEEE, 1998. 3

[30] 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. Springer, 2016. 3

[31] Nikos Paragios and Visvanathan Ramesh. A mrf-based approach for real-time subway monitoring. In *Computer Vision and Pattern Recognition*, volume 1. IEEE, 2001. 3

[32] Viet-Quoc Pham, Tatsuo Kozakaya, Osamu Yamaguchi, and Ryuzo Okada. Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3253–3261, 2015. 3

[33] 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*, pages 91–99, 2015. 7

[34] Weihong Ren, Di Kang, Yandong Tang, and Antoni B Chan. Fusing crowd density maps and visual object trackers for people tracking in crowd scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5353–5362, 2018. 1

[35] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision workshop on Benchmarking Multi-Target Tracking*, 2016. 2, 6, 8

[36] Mikel Rodriguez, Ivan Laptev, Josef Sivic, and Jean-Yves Audibert. Density-aware person detection and tracking in crowds. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2423–2430. IEEE, 2011. 1

[37] David Ryan, Simon Denman, Clinton Fookes, and Sridha Sridharan. Scene invariant multi camera crowd counting. *Pattern Recognition Letters*, 44(8):98–112, 2014. 3, 7

[38] Payam Sabzmeydani and Greg Mori. Detecting pedestrians by learning shapelet features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 3

[39] Deepak Babu Sam, Shiv Surya, and R Venkatesh Babu. Switching convolutional neural network for crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 6, 2017. 3, 4

[40] Zan Shen, Yi Xu, Bingbing Ni, Minsi Wang, Jianguo Hu, and Xiaokang Yang. Crowd counting via adversarial cross-scale consistency pursuit. In *Computer Vision and Pattern Recognition*, pages 5245–5254, 2018. 3

[41] Zenglin Shi, Le Zhang, Yun Liu, Xiaofeng Cao, Yangdong Ye, Ming-Ming Cheng, and Guoyan Zheng. Crowd counting with deep negative correlation learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5382–5390, 2018. 3

[42] Vishwanath A Sindagi and Vishal M Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1879–1888. IEEE, 2017. 1, 3, 4

[43] Vishwanath A Sindagi and Vishal M Patel. A survey of recent advances in cnn-based single image crowd counting and density estimation. *Pattern Recognition Letters*, 107:3–16, 2018. 1, 3

[44] N. Tang, Y. Y. Lin, M. F. Weng, and H. Y. Liao. Cross-camera knowledge transfer for multiview people counting. *IEEE Transactions on Image Processing*, 24(1):80–93, 2014. 3

[45] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004. 3

[46] Paul Viola, Michael J Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2):153–161, 2005. 3

[47] Yi Wang and Yuexian Zou. Fast visual object counting via example-based density estimation. In *IEEE International Conference on Image Processing (ICIP)*, pages 3653–3657. IEEE, 2016. 3

[48] Bo Wu and Ram Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247–266, 2007. 3

[49] Bolei Xu and Guoping Qiu. Crowd density estimation based on rich features and random projection forest. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2016. 3

[50] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 833–841, 2015. 3

[51] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 589–597, 2016. 3, 4

[52] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22, 2000. 6