LEARNING FACTORIZED CROSS-VIEW FUSION FOR MULTI-VIEW CROWD COUNTING

Liangfeng Zheng[1], *Yongzhi Li*[1] *and Yadong Mu*[1*]

[1]Wangxuan Institute of Computer Technology, Peking University

{zhengliangfeng,yongzhili,myd}@pku.edu.cn

2021 IEEE International Conference on Multimedia and Expo (ICME) | 978-1-6654-3864-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICME51207.2021.9428284

ABSTRACT

Crowd counting has been a long-standing task in surveillance video analysis. Most of existing methods focus on a single-view setting. Crowd counting with multiple views can provide richer and complementary information across views. However, the task is still inadequately explored in the literature. Previous works have attempted either to project each camera view onto a common geometric 2-D ground-plane and estimate crowd density map through aggregation [1], or set up connections among all pixel pairs [2]. However, registering a local view to the global ground-plane is error-prone and fails to explicitly model the critical inter-view correlation. Full inter-pixel connections inevitably lead to explosion of parameters. To solve these problems, in this paper, we propose an efficient module that effectively does the job of cross-view fusion by directly modeling the correlation between each pair of views. More specifically, to distill and transfer all useful information from multiple sources views to a target camera view, we factorize the full transformation into a generic-fusion component that encodes all geometric / semantic information of this target view, and a view-specific affine-transform component that encodes the scene geometry / semantics cues of specific source view. This factorization significantly reduces the parameter redundancy and enables plug-and-play of new cameras. Extensive experiments on three multi-view counting datasets (PETS2009, DukeMTMC, and CityStreet) clearly and consistently demonstrate the superiority of the proposed method.

Index Terms— Scene Analysis, Multi-View Crowd Counting, Cross-View feature fusion, Neural network

1. INTRODUCTION

Crowd counting aims to count people in surveillance images or videos. It is of great practical importance in many applications such as crowd management, public safety, traffic monitoring or urban planning [3]. A popularly-adopted method in crowd counting is estimating a crowd density map using deep neural networks, as demonstrated in [4, 5, 6]. Nonetheless, most of existing works operate in a single-view setting, ignoring the correlation among different cameras. Indeed, cross-view crowd counting is highly demanded in many scenarios, since single-view images are intrinsically not suitable for tackling large wide scenes or scenes with many occlusions.

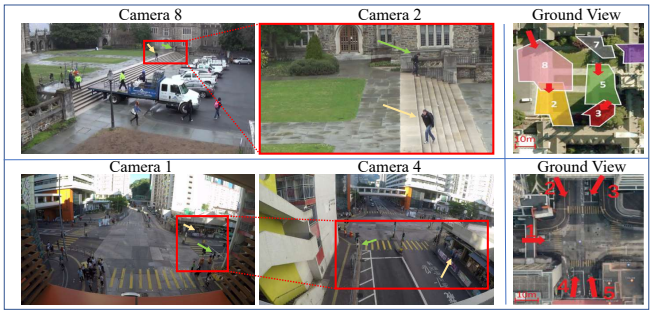

Fig. 1. Illustration of the benefit of the common region from another camera view. Arrow with the same color indicated the same pedestrians in different views. It can be seen that areas that are difficult to see clearly in one view are easy to see in another view. Zoom in and viewed in color for more details.

This work aims to develop a new method for fusing multiple camera views with overlapped field-of-views for improved estimation of crowd count in complicated scenes. Early development of multi-view counting mostly relies on foreground extraction and hand-crafted features, leading to inferior performance. More Recent works adopt convolutional neural networks (CNNs) as the main workhorse. For example, the work in [1] projects features from all views into a common 2-D ground plane and conducts feature fusion for obtaining a 2-D density map. [7] takes height into account and instead utilizes a 3-D space for fusing different camera views. In above methods, partial supervision will stem from the consistency among the projected camera views.

We here argue that widely-used common ground-planes in existing methods may not be optimal choices for conducting cross-view fusion. Registering a local view to the global ground-plane is prone to errors and does not explicitly model the critical inter-view correlation. Instead, this work tries to directly harness the correlation between a pair of views. Some motivating cases are found in Fig. 1. As seen, in many cases the crowd information is vague in a view and clearly observed in another view. The fusion of both views can intuitively bring enhanced counting performance.

The idea of explicit view-to-view fusion has rarely been explored in the task of crowd counting. There are tightly-

*Corresponding Author.

978-1-6654-3864-3/21/$31.00 ©2021 IEEE

related works in some other computer vision tasks, such as cross-view human pose estimation [2]. In specific, [2] established parameter-controlled connections between any two pixels of two views, which amounts to tremendous parameters to be learned. In this work, we propose a significantly more efficient view-to-view fusion scheme. For each camera view (*i.e.* a target view), our model is comprised of a generic-fusion component only dependent on the target view and some affine transformations which vary across other views to be fused (*i.e.* source views). Intuitively, the generic-fusion component encodes all geometric / semantic information of a target view. The affine-transform component parameterizes the conversion across views, encoding scene geometry, semantics and all other relevant cues. Such decomposition enables the plug-and-play of new cameras. Namely, when more cameras are added, they can be immediately fused into other views once the affine-transform components are learned.

To validate the proposed model, we conduct comprehensive evaluations on three crowd-counting benchmarks: PETS2009, DukeMTMC, and CityStreet. Our method consistently outstrips all competitors by large margins, re-calibrating the state-of-the-art performance. It is also worth mentioning that our proposed method achieves a huge improvement in both the single view and the ground plane.

2. RELATED WORK

The task of multi-view counting [8, 3] has been studies for years. Conventional relevant methods can be roughly divided into three categories: *3D cylinder based methods [9]* try to find the position of people in the 3D scene by minimizing the gap between the 3D position of the people projected into the camera view and the single view detection, *detection / tracking based methods [10, 11, 12]* that first perform detection or tracking on each scene respectively. Afterward, the view-specific detection results are aggregated via geometric projection to a ground plane, and *regression based methods [13, 14]* that first extract foreground segments from each view and then use a regression model to predict the count number according to segment-level features. The aforementioned methods often lead to limited performance, partly owing to an error-prone step that separates the crowd from scene background and the use of hand-crafted features (rather than modern deep model-induced neural features) in human detection or crowd count regression.

Recent drastic advance of deep neural networks has inspired several multi-view counting methods that have the traits of end-to-end parameter optimization and the use of pre-trained deep features. For example, Zhang and Chan proposed a DNN-based two-stage multi-view counting method dubbed as MVMS [1]. In the first stage of MVMS, each camera view separately extracts view-specific information (*e.g.*, density map) and projects it onto a common ground plane in the 3-D scene, using calibrated camera parameters. Next, features from multiple views are pooled (or concatenated) and go through some additional neural blocks for obtaining scene-level density maps. More recently, Zhang *et.al* [15] further proposed a new method by taking multi-height projection into account to improve the geometric correspondence across views. Unlike [1], this method predicts a 3-D crowd density map that encodes the distribution of the crowd in a 3-D space. Compared with conventional methods, both [1] and [15] demonstrated large performance leaps. However, these ground-plane based models heavily hinge on the local-to-global geometric registration and are often ineffective in tackling view-dependent occlusions, scale variation across views etc. In this work, we learn to directly explore overlapped regions between two camera views.

3. OUR PROPOSED METHOD

In this section, we first describe the problem setting of multi-view crowd counting, then overviews our proposed model. In the last sub-section, we give a detailed description of cross-view fusion, the key module of our proposed solution.

3.1. Multi-view Crowd Counting

We follow the setting of multi-view crowd counting as described in [1]. For multi-view counting, the cameras are fixed and the camera calibration parameters are known. Given a set of multi-view images, the task aims to predict the number of people in the entire 3D scene composed of these images. [1] directly predicts a scene-level density map defined on the ground-plane of the 3D scene. 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.

3.2. Our Proposed Model

Fig. 2 illustrates the computational flow of our proposed cross-view crowd counting model. Images of each camera view first pass through a convolutional neural network (CNN) to obtain the crowd density maps. A second module in the pipeline interacts with two camera views for cross-view enhancement. As a key improvement and technical contribution of this work, we propose a new direct view-to-view fusion scheme, which will be discussed in detail later. Finally, the enhanced density maps of all views are projected into a common ground plane, where a learnable predictor is optimized to generate the final crowd count. Below we detail the proposed model except for deferring the cross-view fusion scheme to Section 3.3.

Backbone network. In conventional single-view crowd counting models, the feature-extracting backbone network is often designed to be very complicated [16, 6, 17]. Since the design of such backbone network is not the main focus of this work, we follow previous common practice and directly use a light-weight FCN-7 as our backbone network universally for all views. Detailed neural architecture can be found in [1] and omitted here. We only slightly tailor the input to be RGB color images, rather than grayscale images as in [1].

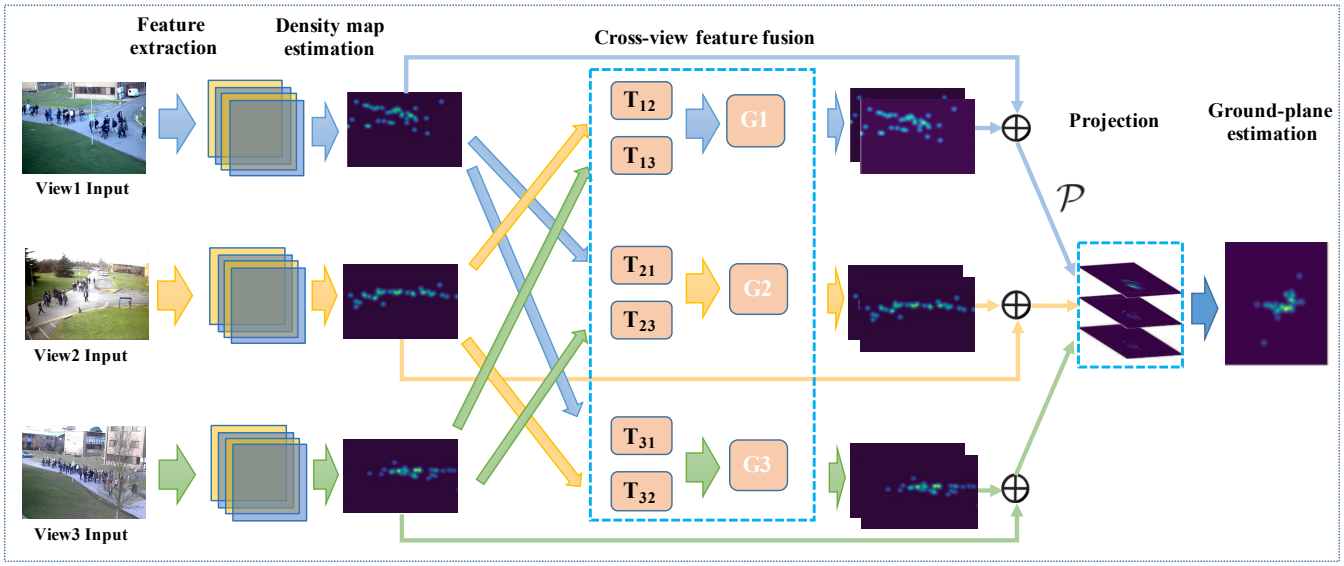

Fig. 2. The pipeline of our proposed model for the multi-view crowd counting problem. Each view corresponds to a unique camera, with all intrinsics known. The pipeline is comprised of three sequential steps: view-specific density map estimation (via a light-weight CNN that reads each image), cross-view fusion that enhances the initial estimations, and projection from each individual view to a common ground-plane, where a crowd count is finally estimated.

Projection between camera-views and ground-plane. After getting the density map of each view, we need to do a projection between the camera view and the scene. Since we already know the intrinsic and extrinsic parameters of the cameras, the conversion process can be implemented by a differentiable fixed-transformation module. Each pixel's height in the 3D world is assumed as a person's average height (1750 mm). The camera parameters with the height assumption are used to calculate the correspondence mapping P between 2D image coordinates and the 3D coordinate, which can be implemented as Spatial Transformer Networks [18].

Ground-plane density map predictor. Since the density map is stretched during the projection step, an additional normalization as described in [1] is employed to keep the sum of the density map before and after the projection unchanged. Normalized projected density maps are then concatenated and fed to a small CNN to obtain the eventual scene-level density map. For this small CNN, we follow the design of 3 convolutional layers as in [1]

3.3. Cross-View Fusion

The idea of view-to-view fusion has not been explored in the context of multi-view crowd counting yet. However, there exist a few such endeavors in similar tasks, such as human pose estimation [2]. In specific, [2] proposes a cross-view fusion strategy that naively connects each pixel on view 1 with all pixels on view 2. Intuitively, this results in incredibly large number of parameters and high computational cost. To comprehensively demonstrate the superiority of our proposed cross-view fusion scheme, we adapt [2] to the task of multi-view crowd counting for comparison. Below we present both [2] and our new idea that conducts improved cross-view fusion by joining generic and affine-transformation components.

Naive fusion strategy in [2]. Denote the point sets under different cameras 1 and 2 as Z_1 and Z_2, respectively. The features of view 1 and 2 at different locations are denoted as $F^1 = \{x_1^1, ..., x_{|Z_1|}^1\}$ and $F^2 = \{x_1^2, ..., x_{|Z_2|}^2\}$. The key to establishing the feature interconnection between view1 and view2 is the correspondence between the two views:

$$x_i^1 = x_i^1 + \sum_{j=1}^{|Z_2|} w_{j,i} * x_j^2, \forall i \in Z_1, \tag{1}$$

where $w_{j,i}$ is a scalar parameter. In [2], determining the values for all $w_{j,i}$ for each pair of cameras is accomplished by FCLs (Fully Connect Layers).

Geometrically, when a pixel i in view 1 and another pixel j in view 2 correspond to the same 3D point in the scene, they shall follow the constraint of epipolar geometry. Namely, $w_{j,i}$ is generally positive for entries on the epipolar line and zero otherwise, which indeed affects the feasible set of $w_{j,i}$. Nonetheless, our experiments reveal that the enforcement of epipolar constraints suffers from high sensitivity to the precision of camera calibration. Therefore we abandon all geometry-induced constraints and let all $w_{j,i}$ freely optimized in the implementation.

Critically this strategy adopts a full pixel-to-pixel connection and entails tremendous number of parameters (quadratic with respect to image pixels) to be learned. When the input image has high spatial resolution, the strategy will severely suffer from the explosion of parameter count and also the over-fitting issue on small data.

Improved cross-view fusion strategy. To reduce the number of total learnable parameters, we factorize the cross-

Table 1. Experiment results: mean absolute error (MAE) on three multi-view counting datasets. We list out the MAE of each camera view as well as the whole scene. "scene" denotes the scene-level counting error. In addition to comparison to state-of-the-art methods, we also implemented FCLs [2] and performed experiments on the multi-view crowd counting dataset for comparison. Best performances are highlighted in bold.

Dataset	PETS2009				DukeMTMC					CityStreet			
Camera	1	2	3	scene	2	3	5	8	scene	1	3	4	scene
Dmap weighted	3.37	5.59	5.84	8.32	0.62	0.91	0.98	1.41	2.12	10.16	12.55	21.56	11.10
Detection+ReID	8.60	1.19	14.61	9.41	2.06	**0.25**	0.96	3.58	2.20	41.38	32.94	28.57	27.60
Late fusion [1]	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
Naive early fusion [1]	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 [1]	**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
3D counting [7]	-	-	-	3.15	-	-	-	-	1.37	-	-	-	7.54
FCLs [2]	1.77	2.74	3.56	3.40	0.52	0.78	**0.36**	1.12	1.02	7.67	7.49	5.50	7.71
Our method	**1.66**	**2.36**	**3.41**	**3.08**	**0.46**	0.73	0.42	**1.10**	**0.87**	**7.38**	**6.85**	**5.18**	**7.08**

view fusion module into two sub-modules: the generic fusion model and light-weight affine transformations.

We set the current view as the target view and all other views as the source view. Let $w^{base} \in \mathbb{R}^{k\times(H\times W)}$ be the parameter matrix of a generic fusion model, which connects k pixels in a target view and all $H \times W$ pixels in a source view (w^{base} only varies with respect to the target view and is the same for different source views. We thus omit the view-related indices unless otherwise notified). The information transfer on the source view can pass the w^{base} to the k pixels of the target view. Because some regions in a view may not be visible in another view or uninformative for the crowd counting task, we choose k pixels instead of all $H\times W$ pixels. It is worth mentioning that we randomly select k pixels from all the pixels where people have appeared. For the choice of k, we do a thorough ablation study in Section 4.3. Since the geometry (position, orientation angle etc.) and scene semantics of each camera are typically unique, an independent generic fusion model is learned for each camera. To reduce the number of total learnable parameters, we factorize the cross-view fusion module into two sub-modules: the generic fusion model and light-weight affine transformations.

Suppose the ith selected pixel in view 1 can find its correspondence in an epipolar line I in view 2, which is encoded by $w_i^{base} \in \mathbb{R}^{H\times W}$. If view 2 changes to 3, we can obtain the epipolar line by applying an appropriate affine transformation to I. This is equivalent to affine transformation to w_i^{base}. We can compute the corresponding fusion model for all k pixels by applying affine transformations:

$$w_i^{adv} = T^{\theta_i}(w_i^{base}), i = 1, ..., k, \tag{2}$$

where T is the affine transformation function, θ_i is the parameter of each selected pixel which can be learned from data. Note that T is implemented by Spatial Transformer Networks [18], which only requires six parameters. In other words, the affine transform of all selected k pixels only needs $6\times k$ parameters, which greatly reduces the number of parameters compared to the naive fusion strategy. A more detailed analysis is conducted later. We learn different θ for different camera pairs. It is worth noting that both w^{base} and θ are invariant to a specific image. When a new view is added, we can learn θ using very little data and transfer the information of that view to other views, because the number of parameters of θ is very small. Since all cameras are assumed to be fixed once mounted, all above-mentioned parameters need to be calibrated once only.

After obtaining w^{adv}, we can pass the information from view 3 to selected k pixels on view 1. Other views will also do the affine transformation, but the parameters used are different. After getting all the information from other views, we concatenate them with the information of the current view. The new feature is used to generate density maps on each single-view and ground-plane, which are supervised by mean-square error (MSE) loss.

Parameter Analysis. We set the number of cameras as N, and the image on each camera has $H \times W$ pixels. It can be seen that the parameter of generic fusion is $k \times H \times W$, and the parameter of each affine transform is $6 \times k$, so the parameter of our method is $N \times k \times H \times W + C_N^2 \times 6 \times k$. In contrast, the number of parameters of the original naive model is $C_N^2 \times (H \times W) \times (H \times W)$, which is much larger especially when N is relatively large.

4. EVALUATIONS AND EXPERIMENTS

4.1. Datasets and Evaluation

Datasets. To evaluate the proposed method, we follow the dataset settings in [1] and compare our methods on 3 public multi-view counting datasets, PETS2009 [19], DukeMTMC [20] and CityStreet [1]. PETS2009 [19] contains 3 views. There are 1105 and 794 images for training and testing respectively. The input image is resized to 384×288 and the ground-plane density maps resolution is 152×177. DukeMTMC [20] contains 4 views. The first 700 images and the remaining 289 images are used for training and testing respectively. The resolution of input images is 640×360, and

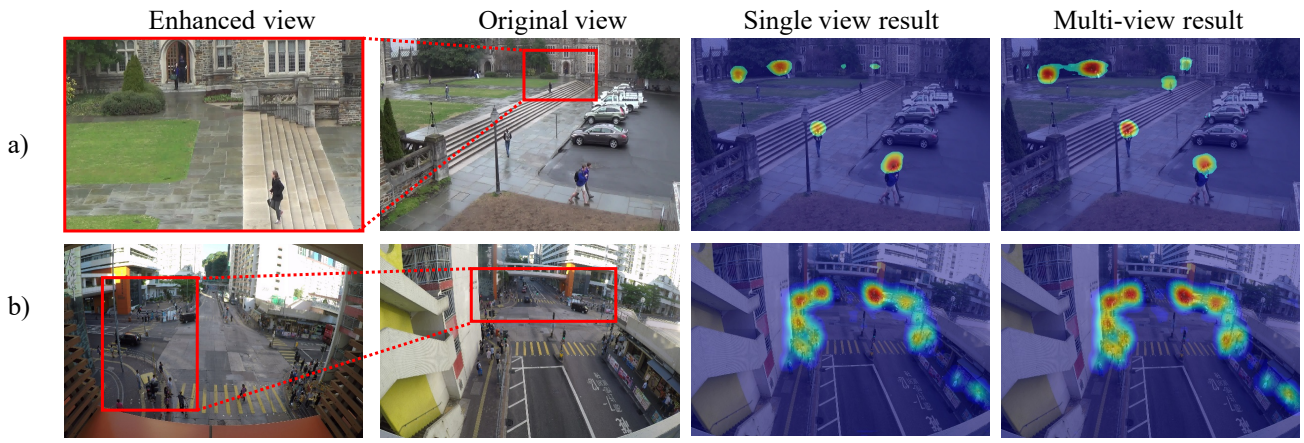

Fig. 3. Multi-view enhanced visualization. It is difficult to predict accurately in the original view relying on a single view. But with the enhancement of the features from the enhanced view, the model can accurately predict even people with several pixels. Zoom in and view in color for more details.

Table 2. Ablation study on CityStreet dataset comparing the different choice of k.

Camera	1	3	4	scene
$k = 10$	7.81	7.66	5.56	7.67
$k = 50$	7.46	6.80	5.26	7.45
$k = 100$	**7.38**	6.85	5.18	**7.08**
$k = 200$	7.41	**6.68**	**4.95**	7.32

160×120 for the ground-plane density maps. CityStreet [1] consists of 3 views and 500 images in which the first 300 is for training and the rest 200 is for testing. The resolution of input images is 676×380 and the resolution of ground-plane density maps is 160×192. The intrinsic and extrinsic parameters are estimated using a calibration algorithm from [21].

Evaluation Metric. Mean absolute error (MAE) is used to evaluate the scene-level counting performance as well as compare scene-level predicted counts and ground truth counts. In addition, we also evaluate the MAE of the predicted counts in each camera-view.

4.2. Implementation Details

We add the cross-view fusion module to the entire model for training. In the training process, the generic fusion model remains the same for each view, while the affine transformations are different in different camera pairs. We use multi-view data to train the cross-view fusion module in an end-to-end fashion, whose parameters are updated together with other parameters of the network.

A two-stage training process is applied to train the model. In the first stage, the model is trained under the supervision of density maps in each view together with the ground-plane density map. All density maps are obtained by applying a fixed-sized Gaussian kernel convolution on the ground truth. In the second stage, the supervisory information of the single-view density maps is removed and the backbone network is fixed except the cross-view fusion and final crowd count predictor. The learning rate and batch size are set to 1e-4 and 1 without further empirical tuning in all experiments.

4.3. Quantitative Results

We design different comparison experiments to explore the effectiveness of the module, including comparison with the state-of-the-art methods and ablation study.

Comparison with the state-of-the-art. Table 1 shows all comparisons conducted on three multi-view crowd counting datasets. "Dmap weighted" fuses single-view density map into a scene-level count with a view-specific weighted map, which is adapted from [13]. "Detection + ReID" first detects all humans in each camera-view and then associates the same people across views. "Late fusion" model fuses density maps to generate ground plane density maps. "Naive early fusion" model fuses feature maps to generate ground plane density maps; "MVMS" model fuses feature maps with a scale selection module. "3D counting" model projects feature maps into 3D space and supervises the training with projection consistency measure loss. Besides, we also implement FCLs [2] proposed in the multi-view human pose estimation task and adapt it into our multi-view crowd counting task, denoted as FCLs in Table 1.

As can be seen from the table, our method achieves superior results on all three datasets. Especially in the City Street dataset and PETS2009 dataset where people are relatively dense, our method surpasses the state-of-the-art methods for both single views and ground plane. Since the CityStreet dataset has more people and a denser population than the PETS2009 dataset, our model achieves significant improvement compared with state-of-the-art. On the DukeMTMC dataset, in which cameras 2, 3, and 5 are not related to each other, the cross-view fusion method has almost no improve-

ment in these three views compared to the original late fusion method. However, on camera 8, the areas that are originally difficult to be observed on the picture are supplemented by other cameras, thus the performance of the model is significantly improved. Detection+ReID achieves the best results on camera 3 in the DukeMTMC dataset because this camera is almost parallel to the horizontal and has a low count, which is an ideal scene for the detector.

Ablation study. We conduct an ablation study on the CityStreet dataset, which is also the most densely populated dataset. The columns of Table 2 show the results of using different k in generic fusion model. Because the input image resolution is small, and the image is down-sampled three times by the model, the value of k we choose is also relatively small. It can be seen that as k increases, the crowd counting performance of the model becomes better. When k reaches a certain level, the performance of the model does not continue to become better as k becomes larger but even worse on part of the view and ground plane. However, with an increase of k, the parameter amount and calculation amount of the model increases. For the trade-off of model speed and accuracy, we need to choose an appropriate k. In our experiments, we choose $k = 100$.

4.4. Qualitative Results

To prove that the model has learned relevant information from other views to improve the crowd counting prediction of the current view after employing the cross-view fusion module we designed, we visualize the density map output from the last layer of the network for several representative examples, as shown in Fig. 3.

As seen, some people are only visible at limited number of pixels in the video frames. It is insufficient to use single-view image for precisely counting such people. Through performing cross-view fusion, features that are obtained from other angles of view are used to enhance the current view, helping improve the accuracy of the prediction under the current view. When the prediction on each view is more accurate, it is obvious that more accurate information can be obtained on the ground-plane.

5. CONCLUDING REMARKS

In this work, we propose a novel method for learning human crowd counting in a multi-view scenario. Our key idea is utilizing the common region constraint between different views to solve the problems of occlusion and long-distance blurring in cameras, which enhances the feature maps in every single view and further improves the performance on the ground plane. We evaluate the proposed method on three public datasets and conduct several ablation studies. Strong evidence is observed to demonstrate its effectiveness and superiority. *Acknowledgement:* This work is supported by National Key R&D Program of China (2020AAA0104400), National Natural Science Foundation of China (61772037) and Beijing Natural Science Foundation (Z190001).

6. REFERENCES

[1] Qi Zhang and Antoni B Chan, "Wide-area crowd counting via ground-plane density maps and multi-view fusion cnns," in *CVPR*, 2019.

[2] Haibo Qiu, Chunyu Wang, Jingdong Wang, Naiyan Wang, and Wenjun Zeng, "Cross view fusion for 3d human pose estimation," in *ICCV*, 2019.

[3] Vishwanath A. Sindagi and Vishal M. Patel, "A survey of recent advances in cnn-based single image crowd counting and density estimation," *Pattern Recognit. Lett.*, vol. 107, pp. 3–16, 2018.

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

[5] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Máadeed, Nasir M. Rajpoot, and Mubarak Shah, "Composition loss for counting, density map estimation and localization in dense crowds," in *ECCV*, 2018.

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

[7] Qi Zhang and Antoni B. Chan, "3d crowd counting via multi-view fusion with 3d gaussian kernels," in *AAAI*, 2020.

[8] Chen Change Loy, Ke Chen, Shaogang Gong, and Tao Xiang, "Crowd counting and profiling: Methodology and evaluation," in *Modeling, Simulation and Visual Analysis of Crowds - A Multidisciplinary Perspective*, Saad Ali, Ko Nishino, Dinesh Manocha, and Mubarak Shah, Eds., vol. 11 of *The International Series in Video Computing*, pp. 347–382. Springer, 2013.

[9] Weina Ge and Robert T Collins, "Crowd detection with a multiview sampler," in *ECCV*, 2010.

[10] 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.

[11] Jingwen Li, Lei Huang, and Changping Liu, "People counting across multiple cameras for intelligent video surveillance," in *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*, 2012.

[12] Huadong Ma, Chengbin Zeng, and Charles X Ling, "A reliable people counting system via multiple cameras," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 2, pp. 1–22, 2012.

[13] David Ryan, Simon Denman, Clinton Fookes, and Sridha Sridharan, "Scene invariant multi camera crowd counting," *Pattern Recognition Letters*, vol. 44, pp. 98–112, 2014.

[14] Nick C Tang, Yen-Yu Lin, Ming-Fang Weng, and Hong-Yuan Mark Liao, "Cross-camera knowledge transfer for multiview people counting," *IEEE Transactions on image processing*, vol. 24, no. 1, pp. 80–93, 2014.

[15] Qi Zhang and Antoni B Chan, "3d crowd counting via multi-view fusion with 3d gaussian kernels.," in *AAAI*, 2020.

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

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

[18] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu, "Spatial transformer networks," in *NIPS*, 2015.

[19] James Ferryman and Ali Shahrokni, "Pets2009: Dataset and challenge," in *2009 Twelfth IEEE international workshop on performance evaluation of tracking and surveillance*, 2009.

[20] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *ECCV*, 2016.

[21] Zhengyou Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, 2000.