BRINGING GENERALIZATION TO DEEP MULTI-VIEW DETECTION

A PREPRINT

Jeet Vora[1]
CVIT, KCIS
IIIT Hyderabad

Swetanjal Dutta
CVIT, KCIS
IIIT Hyderabad

Shyamgopal Karthik
CVIT, KCIS
IIIT Hyderabad

Vineet Gandhi
CVIT, KCIS
IIIT Hyderabad

September 25, 2021

ABSTRACT

Multi-view Detection (MVD) is highly effective for occlusion reasoning and is a mainstream solution in various applications that require accurate top-view occupancy maps. While recent works using deep learning have made significant advances in the field, they have overlooked the generalization aspect, which makes them *impractical for real-world deployment*. The key novelty of our work is to *formalize* three critical forms of generalization and *propose experiments to investigate them*: i) generalization across a varying number of cameras, ii) generalization with varying camera positions, and finally, iii) generalization to new scenes. We find that existing state-of-the-art models show poor generalization by overfitting to a single scene and camera configuration. We propose modifications in terms of pre-training, pooling strategy, regularization, and loss function to an existing state-of-the-art framework, leading to successful generalization across new camera configurations and new scenes. We perform a comprehensive set of experiments on the WildTrack and MultiViewX datasets to (a) motivate the necessity to evaluate MVD methods on generalization abilities and (b) demonstrate the efficacy of the proposed approach. The code is publicly available at https://github.com/jeetv/GMVD

1 Introduction

Occlusion handling remains one of the significant challenges in object detection. Multi-View Detection (MVD) is one of the mainstream solutions to deal with occlusions, especially when detecting humans/pedestrians in crowded settings. The input to MVD methods is images from multiple calibrated cameras observing the same area from different viewpoints with an overlapping field of view. The predicted output is an occupancy map [11] in the ground plane (bird's eye view). Since the cameras are calibrated, bounding boxes in each view corresponding to each point on the occupancy grid can be calculated via projection, assuming an average human width and height. More accurate estimates can be made by regressing the height on a separate branch [14]. MVD has been successfully deployed in many applications like tracking players in sports [30].

Classical methods in MVD rely on background subtraction to compute likelihood over a fixed set of anchor boxes derived using scene geometry, project them on the top view and adopt conditional random field (CRF) or mean-field inference for spatial aggregration [11, 3, 1]. While classical methods have been successful, they observe a gradual degradation in detection performance with increased crowds, as the background subtraction becomes less effective with increase in crowds and clutter.

Deep learning-based methods have replaced previous background subtraction based methods with anchor-based detections; some process each view separately [34] and some process them simultaneously [2, 6]. However, the inaccuracies in the pre-defined anchor boxes [17] limit the performance of these methods. Even if the boxes are correct, locating the exact ground point to project in each 2D bounding box presents a challenge and leads to a significant amount of errors. Moreover, some of the Deep MVD methods still rely on operations outside of Convolutional Neural Net-

[1]Corresponding author jeet.vora@research.iiit.ac.in

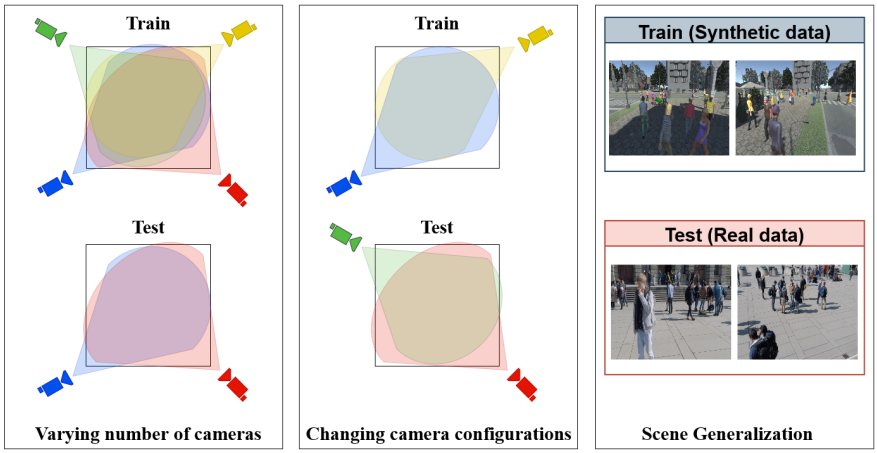

Figure 1: Three forms of generalization required in MVD: (a) varying number of cameras, (b) different camera configurations, and (c) generalizing to new scenes.

works (CNNs), requiring to work out a balance between different potential terms [2]. Hou *et al.* [14] propose a neat end-to-end trainable architecture called MVDet, eliminating the need for anchor boxes as well as CRFs or mean-field inference. It aggregates projected features from a ResNet [13] backbone using three convolutional layers to predict the final occupancy map. MVDet achieves a notable improvement over the preceding deep learning methods (over 14% improvement on the WildTrack dataset [5]).

The Deep MVD methods have made significant progress; however, the literature has overlooked the critical aspect of generalization. Ideally, three forms of generalization abilities are essential for the practical scalability and deployment of MVD methods, which is illustrated in Fig. 1:

1. *Varying number of cameras:* The model should adapt to a varying number of cameras (a network trained on six camera views, should apply to a setup with five cameras).
2. *Varying configuration:* The model should not overfit to the specific camera configuration, the performance should be similar even with altered camera positions, as long as they cover the dedicated areas together.
3. *Varying scenes:* Models trained on one scene should work on another (model trained on a traffic signal should work on a setup inside a university).

Surprisingly, all existing deep learning-based MVD methods are trained and tested with the same camera configuration, on the same scene, using the same number of cameras. Hence, the current experimental setup seriously hinders them from a deployment perspective, as models trained on one setting do not work effectively on another setting.

Our work focuses on addressing these three forms of generalization in deep MVD. We showcase the underlying limitations in the existing frameworks using a series of novel experiments: (a) we train the model using all cameras and test only using a subset of cameras to evaluate generalization with varying number of cameras, b) we train on a subset of cameras and test on another subset to evaluate generalization to new configurations and (c) we perform cross-dataset evaluations to evaluate scene generalization. We then propose simple design choices to improve the generalization abilities of the existing state-of-the-art approach. We show that minimal changes in the loss function, spatial aggregation and the use of pre-training can improve the performance and the generalization abilities. We also propose a regularization technique by randomly dropping camera views while training. More formally, we make the following contributions:

1. The key contribution of our work is to conceptualize and emphasize the importance of generalization in MVD, a crucial concern for its practical usability. Furthermore, we propose novel experimental setup to formally evaluate the generalization abilities of MVD methods.
2. We demonstrate that minimal, but thoughtful changes in terms of pre-training, feature aggregation strategy, regularization, and loss function can significantly improve the generalization of state-of-the-art MVD methods.
3. We back our claims using an extensive set of experiments and ablation studies. We show more than a staggering 20% improvement in scene and configuration generalization, paving the way for a practicable MVD.

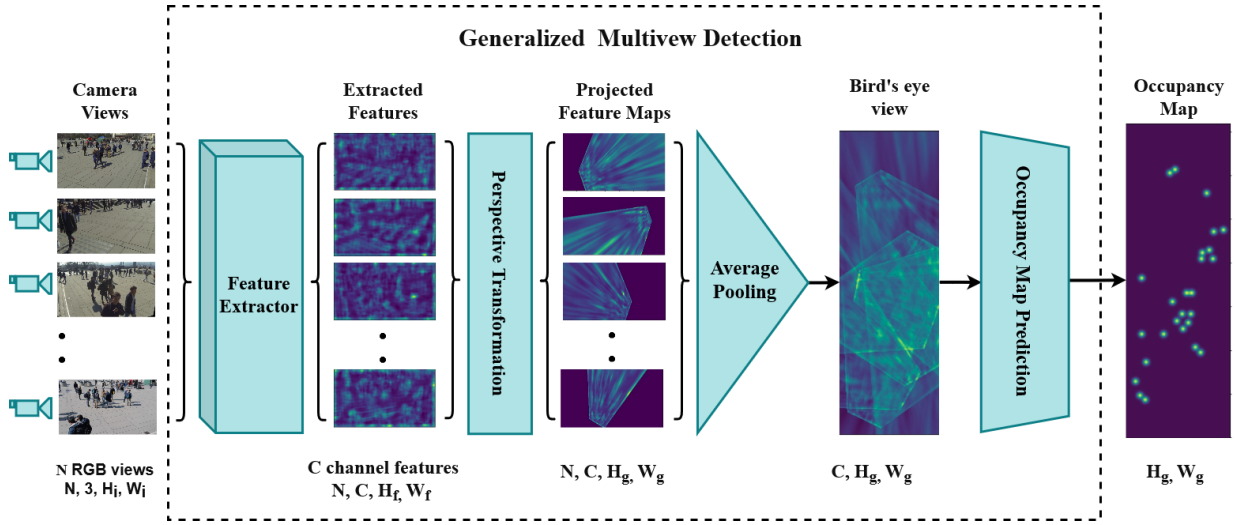

Figure 2: Our proposed architecture: ResNet features are extracted from the input views, which are then projected to the top view. Following this, the projected features across views are pooled and then the final occupancy map is predicted. The use of average pooling across views is crucial in ensuring that our proposed architecture can work for an arbitrary number of views.

2 Related Work

2.1 Monocular Object Detection

CNN-based object detectors such as Faster R-CNN [29], SSD [22], YOLO [28] which are anchor-based approaches, propose candidate bounding boxes with scores on which Non-Maximum Suppression (NMS) is performed to obtain the final set of detection candidates. Pre-defined anchor boxes have drawbacks such as tuning the anchor box hyperparameters; therefore anchor free approaches [37, 18, 9, 20] have been proposed for monocular object detection. Occlusion handling is one of the most significant challenges in monocular object detection, and varying methodologies have been explored to tackle it. The major categories include part-based detections [10, 26, 24, 35], occlusions-aware loss functions [36, 33], exploiting depth cues using RGB-D and LiDAR data [12, 7] and using multiple RGB camera views (multi-view) [11, 5]. Our work focuses on the multi-view setup.

2.2 Multi-view Pedestrian Detection

Using multiple views is a popular method to deal with occlusion and crowded scenes, especially in application settings such as sports analytics that additionally require an accurate bird's eye view localization. The setup involves multiple calibrated cameras, and the challenge is to fuse information extracted from individual views. The prior art can be categorized based on their choice of information extraction module and the fusion module.

Seminal work by Fleuret *et al.* [11] cast MVD as predicting occupancy probabilities over a discrete grid, an idea which has stood the test of time. The major limitation which remains with classical methods is their reliance on background subtraction or other handcrafted classifiers [11, 31, 34, 30, 1]. Recent methods employ deep feature extractors and detectors instead [6, 2, 25, 23, 21]; however, they also suffer from inaccuracies in the pre-defined anchor boxes and projection errors. Spatial aggregation (fusion) is often done using a Conditional Random Field (CRF) or using mean-field inference [30, 2, 11].

MVDet [14] is a recent anchor-free approach that aggregates multi-view information by perspective transformation and concatenating multi-view feature map onto the ground plane and then performs large kernel convolution for spatial aggregation. It overcomes limitations of manually tuning of CRF potentials, reliance on pre-defined 3D anchor boxes, projection errors from monocular detectors, and achieves state-of-the-art performance on WildTrack [5] dataset. Our work improves upon MVDet by proposing better design choices tailored for improved generalization and a more principled choice of the loss function [4].

3 Method

Inspired by the success of MVDet [14], we propose an anchor free deep MVD method specifically tailored to improve the generalization abilities by modifying the training objective and making use of an average pooling strategy on the projected feature maps. The overall architecture is shown in Fig. 2. The input to our pipeline are multiple calibrated RGB cameras with overlapping fields of view, and the expected output is the occupancy map for pedestrians.

3.1 Feature Extraction and Perspective Transformation

Feature Extractor: We use a ResNet18 [13] backbone as a feature extractor replacing last three strided convolutions with dilated convolutions to have a high spatial resolution of the feature maps. Given N camera views of image size $(3, H_i, W_i)$, where H_i and W_i corresponds to height and width of images, C-channel features are extracted for N camera views which corresponds to size (N, C, H_f, W_f), where H_f and W_f represents the height and width of the extracted features.

Perspective Transformation: The extracted features from the feature extractor are then projected onto the ground plane using a perspective transformation, where (H_g, W_g) corresponds to the height and width of the ground plane grid. Considering the calibrated cameras, K represents the intrinsic camera parameters and $[R|t]$ represents the extrinsic camera parameters (R is the rotation matrix and t is the translation vector).

In the world coordinate system, the ground plane corresponds to $Z = 0$, i.e., $W = (X, Y, 0, 1)^T$. A pixel of an image $I = (x, y)^T$ is transformed to the ground plane as follows:

$$I = s \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = K[R|t] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{1}$$

$$= P \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2}$$

where s is a scaling factor and P is a perspective transformation matrix.

3.2 Spatial Aggregation

Average Pooling: We first project the ResNet feature maps from each viewpoint on to the bird's eye view using the perspective transformation to obtain the projected feature maps fm_i (where, $i = 1, 2, ..., N$). Following this, we average pool the projected feature maps fm_i to obtain the final bird's eye view feature representation F of size (C, H_g, W_g), which is written as,

$$F = \frac{\sum_{i=1}^{N} fm_i}{N}. \tag{3}$$

While there can be many other alternatives to average pooling, we opt for this solution, primarily because it is permutation-invariant. Unlike MVDet, where the camera views ideally need to be input in the same order as training during inference, our proposed solution can accept arbitrary number of views in an arbitrary order. Furthermore, the average pooling solution is free from any learnable parameters which ensures that there is no overfitting introduced due to this operation. The projected feature maps for N cameras of size (N, C, H_g, W_g) after average pooling, reduces to (C, H_g, W_g), thus removing the dependency over the number of camera views thereby allowing the model to take an arbitrary number of views as input.

DropView Regularization: Inspired by Dropout [32] as well as work on self-supervised learning which drops color channels to prevent the model from memorization [15, 19], we propose the DropView regularization technique. For each sample, we randomly select one view and discard this during training. The occupancy map prediction step is done with all the remaining views. We provided a detailed analysis of the effect of this regularization strategy in our experiments.

Occupancy Map Prediction: Similar to MVDet [14], we use 3 dilated convolutional layers to predict the occupancy map of size (H_g, W_g).

3.3 Loss Function

The loss function compares the output probabilistic occupancy map (p) with the ground-truth (g). Based on the analysis provided in [4, 27] which focused on the task of predicting saliency maps, we use the combination of Kullback-Leibler Divergence (KLDiv) and Pearson Cross-Correlation (CC) metrics as a loss function. The final loss function can be written as:

$$L(p, g) = \frac{\sigma(p, g)}{\sigma(p) \times \sigma(g)} - \sum_i g_i \log\left(\frac{g_i}{p_i}\right), \tag{4}$$

where $\sigma(p, g)$ is the covariance of p and g, $\sigma(p)$ is the standard deviation of p and $\sigma(g)$ is the standard deviation of g.

4 Experiments

We test our proposed multi-view detection method on two publicly available multi-camera datasets for pedestrian detection **WildTrack** [5] and a synthetic dataset **MultiViewX** [14]. In the following subsections, we present the experiments carried out and the results obtained for the proposed MVD method.

4.1 Experimental setup

Datasets: The *WildTrack* dataset consists of 7 static calibrated cameras with overlapping fields of view and 400 synchronized frames with image resolution of 1,080 $\times$ 1,920 pixels annotated at 2 fps for ground-truth covering an area of 12×36 m^2. The ground plane grid is discretized into a $480 \times 1,440$ grid, where each grid cell is 2.5 cm square. On average, the dataset captures 23.8 persons per frame. The *MultiViewX* dataset is a synthetic dataset which has similar configurations as the *WildTrack* dataset. However, it consists of 6 static calibrated cameras with overlapping fields of view and 400 synchronized frames of resolution 1,080 $\times$ 1,920 annotated at 2 fps for ground-truth covering an area of 16×25 m^2. The ground plane grid is discretized into a $640 \times 1,000$ grid, where each grid cell is 2.5 cm square. On average, the dataset captures 40 persons per frame. For both datasets, we use the first 90% frames in training and the last 10% frames for testing, as done in previous work [14, 5].

Evaluation metrics: We use the standard evaluation metrics proposed in [5]. *Multiple Object Detection Accuracy* (MODA) is the primary performance indicator which accounts for normalized missed detections and false positives i.e., it considers both false negatives and false positives. *Multiple Object Detection Precision* (MODP) assesses the localization precision [16]. *Precision* and *Recall* is calculated by Precision = TP/(TP+FP) and Recall = TP/(TP+FN) respectively; where TP, FP and FN are True Positives, False Positives, False Negatives. A threshold of 0.5 meters is used to determine true positives.

State of the Art comparisons: We compare against 7 different methods. The set includes one monocular object detection baseline (referred to as RCNN clustering [34]); a classical probablistic occupancy map method [11]; and five deep learning approaches [14, 21, 2, 6, 23]. For generalization experiments, we only compare against MVDet, which shows far superior performance than the other methods.

4.2 Implementation Details

Down sampled images of $720 \times 1,280$ pixels serve as an input to the model. The feature extracted from ResNet-18 has $C = 512$ channel features, which is bilinearly interpolated to get the shape of 270×480. These $(N, C = 512, H_f = 270, W_f = 480)$ extracted features are projected onto top view to obtain $(N, 512, H_g, W_g)$ sized features for *N* viewpoints, which are average pooled to obtain the ground plane grid shape of $(512, H_g, W_g)$, where $(H_g, W_g) = (120, 360)$ in the case of the WildTrack dataset and $(H_g, W_g) = (160, 250)$ in the case of the MultiViewX dataset. Spatial aggregation has three layers of dilated convolution with a 3×3 kernel size and dilation factor of 1, 2, and 4. Training is done for ten epochs with early stopping; we set batch size as 2, SGD optimizer with momentum = 0.9 has been used with one-cycle learning rate scheduler. A probability of 0.4 or more on the occupancy grid is considered a detection, and a Non-Maximal Suppression (NMS) is applied with a spatial resolution of 0.5 m. All training and testing have been performed on 2 Nvidia GTX 1080 Ti GPUs. We use pre-trained ImageNet [8] weights in some of the experiments.

4.3 Results

Traditional evaluation: The comparison with the existing state-of-the-art approaches is presented in Table 1. These evaluations are done in the standard setting, where the train and testing are done with the same scene and same camera configuration. We hypothesize that the poor performance of the RCNN-clustering baseline is because of projection

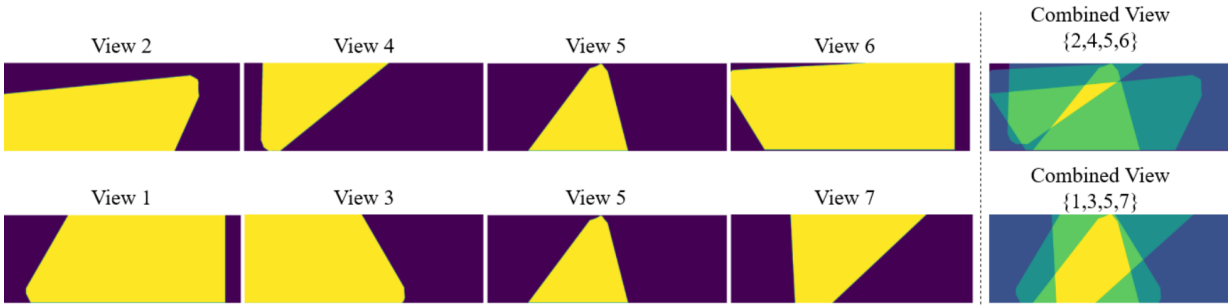

Figure 3: The figure illustrates the projection of the individual camera views onto the bird's eye view for the WildTrack dataset. On the right, we show the two selected camera subsets to measure the performance for changing camera configurations.

Method	ImageNet (pre-train)	WildTrack				MultiViewX			
		MODA	MODP	Prec	Recall	MODA	MODP	Prec	Recall
RCNN Clustering [34]	×	11.3	18.4	68	43	18.7	46.4	63.5	43.9
POM-CNN[11]	×	23.2	30.5	75	55	-	-	-	-
Lopez-Cifuentes *et al.* [23]	×	39.0	55.0	-	-	-	-	-	-
Lima *et al.* [21]	×	56.9	67.3	80.8	74.6	-	-	-	-
DeepMCD[6]	×	67.8	64.2	85	82	70.0	73.0	85.7	83.3
Deep-Occlusion [2]	×	74.1	53.8	95	80	75.2	54.7	97.8	80.2
MVDet[14]	×	88.2	75.7	94.7	93.6	83.9	79.6	96.8	86.7
MVDet(Our Implementation)	×	88.1(±0.8)	75.1(±0.4)	93.8(±0.9)	94.3(±0.6)	83.3(±0.6)	79.3(±0.3)	96.9(±0.6)	86.1 (±0.6)
MVDet + KLCC	×	**89.3**(±0.8)	75.1(±0.2)	94.7(±0.8)	94.6(±0.6)	85.3(±0.2)	79.4(±0.2)	96.5(±0.4)	88.5(±0.3)
MVDet + KLCC	✓	87.9(±0.8)	76.1(±0.2)	92.7(±0.6)	**95.4**(±0.4)	**90.3**(±0.2)	**82.6**(±0.1)	97.0(±0.2)	**93.1**(±0.3)
Ours	×	87.2(±0.6)	74.5(±0.4)	93.8(±1.6)	93.4(±1.8)	78.6(±0.9)	78.1(±0.4)	96.8(±0.5)	81.3(±0.9)
Ours(DropView)	✓	86.7(±0.4)	76.2(±0.2)	95.1(±0.3)	91.4(±0.6)	88.2(±0.1)	79.9(±0.0)	96.8(±0.2)	91.2(±0.1)
Ours	✓	85.4(±0.4)	**76.7**(±0.2)	**95.2**(±0.4)	89.9(±0.8)	86.9(±0.2)	79.8(±0.1)	**97.2**(±0.2)	89.6(±0.2)

Table 1: Comparison against the state-of-the-art methods. Our method refers to the proposed model in Section 3. We made five runs for some of the experiments and the variances are presented in the bracket.

errors which lead to inaccurate localization. Among the deep learning approaches, MVDet outperforms other methods by a notable margin.

We observe that replacing MSE with a combination of KL Divergence (KLDiv) and the Cross-Correlation (CC) loss on the MVDet framework (referred to as MVDet+KLCC) achieves a new state-of-the-art on WildTrack (MODA of 89.3%) and MultiViewX dataset (MODA of 85.3%). This is achieved without using an additional branch used in the original MVDet framework. Using KLCC loss with average pooling attains similar performance to that of concatenation with MSE loss in the MVDet framework. We surprisingly find that using pre-trained ImageNet weights leads to a drop in performance for both the methods Concat+KLCC (MODA=87.9%) and Avgpool+KLCC (MODA=85.4%) on the WildTrack dataset. In contrast, pre-training gives significant performance gains on the MultiViewX dataset.

However, this standard evaluation with a fixed set of cameras, fixed positions and the same scene in the training and test sets is highly misleading. It encourages overfitting, and the performance significantly degrades when deployed on unseen camera configuration and scenes. The rest of the experiments that we present are focused on these generalization aspects.

Performance for a varying number of cameras: Most previous deep learning-based MVD methods cannot handle a varying number of cameras because of the concatenation operation across camera views [14, 6]. Some others require non-trivial adjustment of potential terms [2]. Using average pooling instead of concatenation allows us to adapt the MVDet framework for an arbitrary number of cameras. The results are presented in Table 2. We train the model with all seven cameras and perform inference with only a subset of cameras. We start the inference with three cameras (to have sufficient coverage) and test with up to six cameras. The inference is performed on the test set (the last 10% frames of the dataset). The network already gives a MODA score of 66.8 with only four cameras, demonstrating the efficacy of the average pooling strategy. Adding more cameras allows a closer view of the pedestrians and therefore, leads to an improvement in performance. With six cameras, the MODA score is 84.1, implying that if a camera stops functioning, one can still perform MVD and get results very close to that of a seven-camera setup.

Inference on camera set	MODA	MODP	Prec	Recall
{1,2,3}	46.3	70.9	68.1	87.1
{1,2,3,4}	66.8	70.7	81.3	86.8
{1,2,3,4,5}	81.9	74.5	91.9	89.8
{1,2,3,4,5,6}	84.1	75.8	92.4	91.7

Table 2: Evaluation of our method with varying number of cameras. The model is trained on WildTrack dataset with seven cameras and tested on varying subsets of the cameras.

Train on camera set	Test on camera set	Method	With Pre-trained ImageNet weights				Without Pre-trained ImageNet weights			
			MODA	MODP	Prec	Recall	MODA	MODP	Prec	Recall
{2,4,5,6}	{2,4,5,6}	MVDet+KLCC	**85.2**	72.2	92.6	**92.5**	**82**	**73.4**	88.1	**94.9**
		Ours(DropView)	84	72.9	92.4	91.6	79.2	70.7	86.7	93.6
		Ours	81.8	**73.5**	**93.5**	87.9	79.7	72	**92.9**	86.3
	{1,3,5,7}	MVDet+KLCC	43.2	68.2	**94.6**	45.8	17.5	**73.1**	67.4	34
		Ours(DropView)	**75.1**	71.1	94.3	**79.9**	**62.4**	69	87.5	**72.8**
		Ours	66.5	**71.4**	94.3	70.8	55.9	68.3	**92.6**	60.7
{1,3,5,7}	{1,3,5,7}	MVDet+KLCC	78.2	73.6	89.5	**88.6**	**79.5**	72.7	92.5	**86.6**
		Ours(DropView)	**80.8**	74	**94.2**	86	74.3	72.8	90.5	83
		Ours	76.4	**74.6**	91.5	84.1	71.6	**73.6**	**93.5**	77
	{2,4,5,6}	MVDet+KLCC	27.8	**68.7**	**90.8**	31	5.1	**67.4**	85.5	6.2
		Ours(DropView)	**62.6**	67.4	86.7	73.9	**50.5**	63.2	**91.8**	**55.5**
		Ours	52.4	67.4	81	**68.5**	37.3	60.6	90.4	41.7

Table 3: Experiments on the WildTrack dataset with changing camera configurations; with and without using pre-trained ImageNet weights.

Performance on changing camera configurations : To evaluate the generalization ability of our model when camera positions are changed, we divide the WildTrack dataset into two subsets. The subsets are chosen such that each of them covers approximately the entire environment. Cameras one, three, five and seven form one subset and cameras two, four, five and six form the other subset. This arrangement is illustrated in Fig. 3. We train our model on one set and test on both the same set and the other set. The results are presented in Table 3. Since each set has four cameras, it is possible to train the MVDet with concatenation on both sets. Adding average pooling to MVDet + KLCC improves MODA score by more than 23% in both the cross-subset evaluations. Interestingly, adding average pooling leads to a drop in performance when testing on the same subset and highlights the importance of approaching the problem from a generalization perspective. We also observe a similar trend with the use of ImageNet pre-training. While pre-training does not significantly improve performance when evaluated on the same subset, it proves to be vital in the case of cross-subset evaluation for both MVDet+KLCC as well as our approach.

Scene Generalization: We perform cross dataset evaluation for scene generalization. Table 4 shows the results when we train on synthetic dataset (MultiViewX) and test on real dataset (WildTrack). MultiViewX has six cameras that are used for training, while WildTrack has seven cameras. MVDet concatenates the projected feature maps, so for a fair comparison, we remove one camera from WildTrack during inference. Our model achieves a MODA score of 60.1%, which is a 20.9% improvement over MVDet which has a MODA score of 39.2%. Without pre-training, the MODA drops to 35.1%. The experiment suggests that both average pooling and pre-training are vital for scene generalization. We further benchmark the cross dataset evaluation using all seven views for inference from the WildTrack dataset, and achieve a MODA score of 69.4%, which is shown in the last row of Table 4.

Choice of Loss Function: In Table 5, we present an ablation study to evaluate different loss functions on the scene generalization experiment. The models are trained on MultiviewX dataset and tested on WildTrack. All the experiments use average pooling. We average the results of five independent runs, the variances are shown in the brackets. The model trained on CC is giving good recall and the model trained on KL gives good precision. The combined model gives best performance. The results further indicate that training with KL leads to consistent results (low vari-

Method	Inference on total cameras	ImageNet (pre-train)	MODA	MODP	Prec	Recall
MVDet	6	×	17	65.8	60.5	48.8
Ours	6	×	35.1	**75.3**	**78.9**	47.9
MVDet	6	✓	39.2	68.1	70.6	67.2
Ours	6	✓	**60.1**	72.1	75.6	**88.7**
Ours(DropView)	7	✓	60.1(±0.6)	73.8(±0.1)	75.7(±0.4)	88.4(±0.1)
Ours	7	✓	69.4(±0.6)	72.96(±0.2)	83.74(±0.5)	86.14(±0.3)

Table 4: Scene Generalization : Evaluation of our method while training on synthetic dataset (MultiViewX) and testing on real dataset (WildTrack). Camera 7 of the WildTrack dataset was discarded for the experiments in the first four rows.

Method	ImageNet (pre-train)	MODA	MODP	Prec	Recall
Average pool + MSE	✓	57.3(±0.2)	72.6(±0.0)	75.6(±0.1)	84.5(±0.05)
Average pool + CC	✓	55.5(±5.5)	**74.2**(±0.4)	72.1(±4.4)	**89.5**(±2.6)
Average pool + KL	✓	62.5(±0.1)	73.4(±0.04)	**89.1**(±0.0)	71.3(±0.0)
Average pool + KLCC	✓	**69.4**(±0.6)	72.96(±0.2)	83.74(±0.5)	86.14(±0.3)

Table 5: Choice of Loss Function: we present an ablation study for our proposed method on the scene generalization experiment. Overall, the model trained with both KL-Divergence and Cross-Correlation achieves the best performance.

ance) and using CC alone shows signs of overfitting (high variance). We show further analysis in the supplementary material with synthetic images to justify the choice of KL and CC as the loss function.

Effect of DropView Regularization: The results on the traditional setting with the DropView regularization are presented in Table 1. Here, we see that DropView brings small improvements in the performance on both WildTrack and MultiViewX datasets (increase from 85.4% to 86.7% on WildTrack and from 86.9% to 88.2% on MultiViewX). However, the biggest gains for the DropView regularization is present in the varying camera configuration experiment presented in Table 3. Here, we see the drastic gains that the DropView regularization is able to provide, both when evaluated on the same set of cameras, as well as when evaluated on a different set of cameras. When the evaluation is done on a different camera set, the improvement is almost 10 percentage points on the MODA score, which clearly indicates the robustness the method introduces against changing camera configurations. However, when the DropView regularization is applied for the scene generalization task, there is a signficant drop in performance (Table 4). Overall, our analysis provides insights for an appropriate use of DropView regularization.

5 Conclusion

We find the current Multi-View Detection setup severely limited and encouraging models to overfit the training configuration. Therefore, we conceptualize and propose novel experimental setups to evaluate the generalization capabilities of MVD models in a more practical setting. We find the state-of-the-art models to have poor generalization capabilities on our proposed setups. To alleviate this issue, we introduce changes to the feature aggregation strategy, loss function, as well as a novel regularization strategy. With the help of comprehensive experiments, we demonstrate the benefits of our proposed architecture. Overall, we hope our work plays a crucial role in steering the community towards more practical Multi-View Detection solutions.

References

[1] A. Alahi, L. Jacques, Y. Boursier, and P. Vandergheynst. Sparsity driven people localization with a heterogeneous network of cameras. *Journal of Mathematical Imaging and Vision*, 41(1):39–58, 2011.

[2] P. Baqué, F. Fleuret, and P. Fua. Deep occlusion reasoning for multi-camera multi-target detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 271–279, 2017.

[3] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1806–1819, 2011.

[4] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand. What do different evaluation metrics tell us about saliency models? *IEEE transactions on pattern analysis and machine intelligence*, 41(3):740–757, 2018.

[5] T. Chavdarova, P. Baqué, S. Bouquet, A. Maksai, C. Jose, T. M. Bagautdinov, L. Lettry, P. Fua, L. Gool, and F. Fleuret. Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5030–5039, 2018.

[6] T. Chavdarova and F. Fleuret. Deep multi-camera people detection. *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 848–853, 2017.

[7] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[9] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian. Centernet: Keypoint triplets for object detection. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6568–6577, 2019.

[10] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

[11] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:267–282, 2008.

[12] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *European conference on computer vision*, pages 345–360. Springer, 2014.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[14] Y. Hou, L. Zheng, and S. Gould. Multiview detection with feature perspective transformation. In *ECCV*, 2020.

[15] S. Jenni and P. Favaro. Self-supervised feature learning by learning to spot artifacts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2733–2742, 2018.

[16] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. S. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:319–336, 2009.

[17] T. Kong, F. Sun, H. Liu, Y. Jiang, L. Li, and J. Shi. Foveabox: Beyound anchor-based object detection. *IEEE Transactions on Image Processing*, 29:7389–7398, 2020.

[18] T. Kong, F. Sun, H. Liu, Y. Jiang, and J. Shi. Foveabox: Beyond anchor-based object detector. *ArXiv*, abs/1904.03797, 2019.

[19] Z. Lai and W. Xie. Self-supervised learning for video correspondence flow. *arXiv preprint arXiv:1905.00875*, 2019.

[20] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. *ArXiv*, abs/1808.01244, 2018.

[21] J. Lima, R. Roberto, L. Figueiredo, F. Simões, and V. Teichrieb. Generalizable multi-camera 3d pedestrian detection. *ArXiv*, abs/2104.05813, 2021.

[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.

[23] A. López-Cifuentes, M. Escudero-Viñolo, J. Bescós, and P. Carballeira. Semantic driven multi-camera pedestrian detection. *ArXiv*, abs/1812.10779, 2018.

[24] J. Noh, S. Lee, B. Kim, and G. Kim. Improving occlusion and hard negative handling for single-stage pedestrian detectors. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 966–974, 2018.

[25] J. Ong, B. Vo, D. Kim, and S. Nordholm. A bayesian filter for multi-view 3d multi-object tracking with occlusion handling. *IEEE transactions on pattern analysis and machine intelligence*, PP, 2020.

[26] W. Ouyang, X. Zeng, and X. Wang. Partial occlusion handling in pedestrian detection with a deep model. *IEEE Transactions on Circuits and Systems for Video Technology*, 26:2123–2137, 2016.

[27] N. Reddy, S. Jain, P. Yarlagadda, and V. Gandhi. Tidying deep saliency prediction architectures. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10241–10247, 2020.

[28] J. Redmon, S. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[29] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

[30] G. Roig, X. Boix, H. B. Shitrit, and P. Fua. Conditional random fields for multi-camera object detection. *2011 International Conference on Computer Vision*, pages 563–570, 2011.

[31] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *2011 International conference on computer vision*, pages 137–144. IEEE, 2011.

[32] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[33] X. Wang, T. Xiao, Y. Jiang, S. Shao, J. Sun, and C. Shen. Repulsion loss: Detecting pedestrians in a crowd. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7774–7783, 2018.

[34] Y. Xu, X. Liu, Y. Liu, and S.-C. Zhu. Multi-view people tracking via hierarchical trajectory composition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4256–4265, 2016.

[35] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Li. Occlusion-aware r-cnn: Detecting pedestrians in a crowd. In *ECCV*, 2018.

[36] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li. Occlusion-aware r-cnn: detecting pedestrians in a crowd. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 637–653, 2018.

[37] C. Zhou and J. Yuan. Multi-label learning of part detectors for heavily occluded pedestrian detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3506–3515, 2017.

Inference on camera set	MODA	MODP	Prec	Recall
{1,2,3}	48.7	73.1	70.5	83.5
{1,2,3,4}	72	75.3	85.4	86.7
{1,2,3,4,5}	81.8	77.5	94.4	86.9

Table 6: Evaluation of our method with varying number of cameras. The model is trained on MultiViewX dataset with 6 cameras and tested on varying subsets of the cameras.

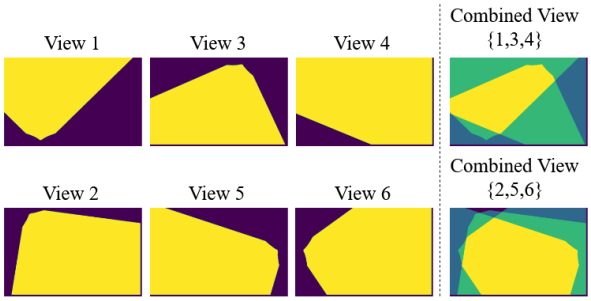

Figure 4: The figure illustrates the projection of the individual camera views onto the bird's eye view for the MultiViewX dataset. On the right, we show the two selected camera subsets to measure the performance for changing camera configurations.

Train on camera set	Test on camera set	Method	With Pre-trained ImageNet weights				Without Pre-trained ImageNet weights			
			MODA	MODP	Prec	Recall	MODA	MODP	Prec	Recall
{1,3,4}	{1,3,4}	MVDet+KLCC	**72**	76.1	93.5	**77.4**	**60.9**	**73.6**	88.4	**70.1**
		Ours(DropView)	69.5	75.2	95.1	73.3	53.2	73.2	**95**	56.2
		Ours	67.7	**76.4**	**96.2**	70.5	56.2	73.4	94.8	59.5
	{2,5,6}	MVDet+KLCC	46.3	66.4	94.5	49.1	23.4	66.1	73.8	36.2
		Ours(DropView)	**65.3**	70.3	93.3	**70.3**	**39.3**	66.3	**88**	45.5
		Ours	59.6	**73.4**	**94.7**	63.2	34.4	**68.8**	80	**45.9**
{2,5,6}	{2,5,6}	MVDet+KLCC	77.6	77.4	93.8	**83.1**	72.2	75.3	90.4	**80.7**
		Ours(DropView)	**78.6**	78.2	**96.5**	81.5	**72.8**	75.4	**92.9**	78.8
		Ours	76.1	**78.7**	95.9	79.5	71.2	**76**	92.2	77.7
	{1,3,4}	MVDet+KLCC	34.3	66.2	93.8	36.7	7.2	58.1	87.4	8.4
		Ours(DropView)	**47.7**	**72.5**	**95.5**	**50.1**	**25.2**	**69.3**	**91.4**	**27.8**
		Ours	45.8	71.8	94.5	48.6	21	65.2	86.3	24.9

Table 7: Experiments on the MultiViewX dataset with changing camera configurations; with and without using pre-trained ImageNet weights.

A Appendix

The supplementary material is organized as follows: Section A.1 contains evaluation results on MultiViewX dataset, Section A.2 contains qualitative results w.r.t to traditional evaluation and the generalization experiments from both WildTrack and MultiViewX datasets respectively and Section A.3 contains the analysis for the choice of loss function based on synthetically generated heat maps.

A.1 Results on MultiViewX Dataset

Performance for a varying number of cameras: Using average pooling instead of concatenation allows us to adapt the MVDet framework for an arbitrary number of cameras. The experiments performed on MultiViewX dataset are

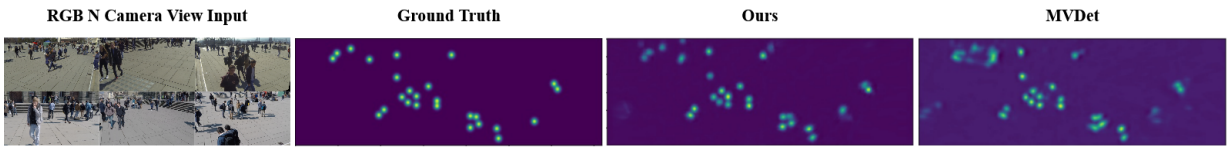

Figure 5: Sample frames from WildTrack dataset with corresponding occupancy maps of ground truth, our result and MVDet for comparison. We can see the clusters forming in the MVDet predictions, in contrast our method gives much sharper and distinct predictions.

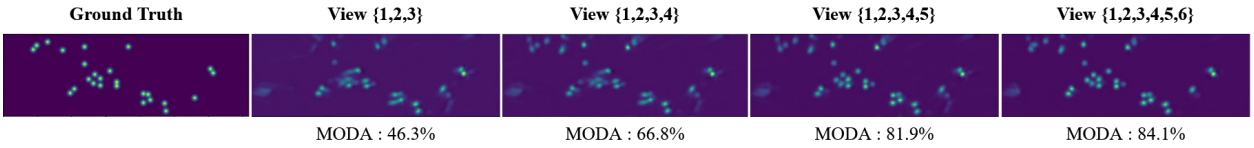

Figure 6: Occupancy maps for varying number of cameras on WildTrack dataset when trained on seven cameras and tested on varying subsets of the cameras.

presented in Table 6. We train the model with all six cameras and perform inference with only a subset of cameras. We start the inference with three cameras and test with up to five cameras. The network gives a MODA score of 72.0 with only four cameras, demonstrating the efficacy of the average pooling strategy on the MultiViewX dataset as well. With five cameras, the MODA score is 81.8, implying that use of average pooling with KLCC loss gives similar performance on MultiViewX dataset.

Performance on changing camera configurations: To evaluate the generalization ability of our model when camera positions are changed, we divide the MultiViewX dataset into two subsets. The subsets are chosen such that each of them covers approximately the entire environment. Cameras one, three, four form one subset and cameras two, five and six form the other subset. This arrangement is illustrated in Fig. 4. We train our model on one set and test on both the same set and the other set. The results are presented in Table 7. Since each set has three cameras, it is possible to train the MVDet with concatenation on both sets. Adding average pooling to MVDet + KLCC improves MODA score by more than 11% in both the cross-subset evaluations.

A.2 Qualitative results

First we show the predicted occupancy maps of MVDet and our method and compare them with the ground truth, in the traditional setting. Subsequently, qualitative results are shown w.r.t to three generalization abilities obtained from both the WildTrack and MultiViewX datasets.

A.2.1 WildTrack Dataset

The traditionally evaluated results which contains occupancy maps of ground truth, our method and MVDet are shown in Fig. 5. The occupancy map from our method which uses average pooling, KLCC loss function and ImageNet pretraining gives us more accurate localization as compared to MVDet.

Varying number of cameras: The output occupancy map for varying number of cameras are shown in Fig. 6. WildTrack consists of seven cameras, we show the results inferred with three cameras upto six cameras. As the number of views are increasing, we get an accurately localized occupancy map.

Changing camera configurations: The output occupancy map for cross subset evaluation are shown in Fig. 7. Here, we have the occupancy maps for a model trained on one set and tested on other set. For example, trained on camera views one, three, five and seven and tested on cameras two, four, five and six or vice-versa. Clearly the pre-training is improving localization in both the methods. Furthermore, our method with average pooling is better at disambiguating the occlusions and also giving brighter outputs (resulting in sharp maxima's).

A.2.2 MultiViewX Dataset

In this subsection the qualitative results for MultiViewX dataset are been shown. We consider similar configurations as in the Wildtrack dataset. The obtained results clearly indicates the improvements our method brings over the MVDet model and observations are similar to that of the Wildtrack dataset. Fig. 8 shows the traditionally evaluated results.

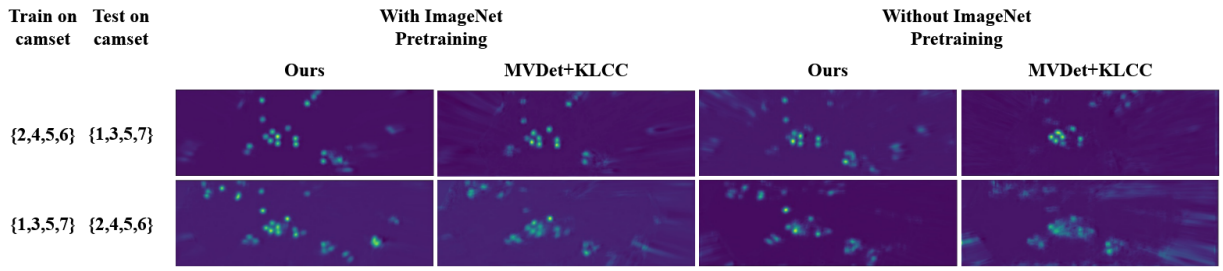

Figure 7: Result occupancy maps for cross subset evaluation from WildTrack dataset.

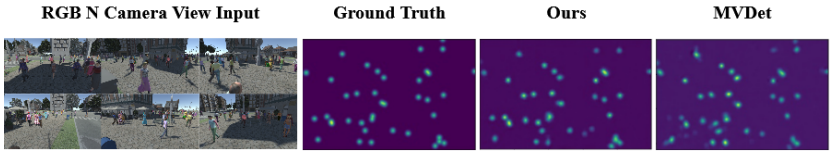

Figure 8: Sample frames from MultiViewX dataset with corresponding occupancy maps of ground truth, our result and MVDet for comparison.

Varying number of cameras: The output occupancy map for varying number of cameras are shown in Fig. 9. MultiViewX consists of six cameras, we show the results inferred with three cameras upto five cameras. As the number of views are increasing, we get an accurately localized occupancy map.

Changing camera configurations: The output occupancy map for cross subset evaluation are shown in Fig. 10. Here, we have the occupancy maps for a model trained on one set and tested on other set. For example, trained on camera views one, three, and four and tested on cameras two, five and six or vice-versa.

A.2.3 Scene Generalization

The qualitative results of output occupancy map for cross-dataset evaluation are shown in Fig. 11, when we train on synthetic dataset (MultiViewX) and test on real dataset (WildTrack). The last two occupancy maps are the outputs of our method and MVDet when tested on only 6 views of WildTrack dataset for having a fair comparison with MVDet. We also show the output occupancy map when tested on all the views of WildTrack dataset. Our method provides accurately localized occuapncy maps and disambiguate the occlusions as compared to MVDet.

A.3 Choice of Loss Function

We further motivate the choice of KLCC loss in Figure 12 using synthetic maps. MSE loss is clearly suboptimal (in the second row the loss reduces again with the predictions deteriorating). CC is symmetric and consistent on all cases. However, the loss magnitude is low in both CC and MSE (second row). KL leads to steeper loss (larger gradient magnitudes), which improves training. The combination of KL and CC merges the benefits of both these loss functions, hence the basis of our choice.

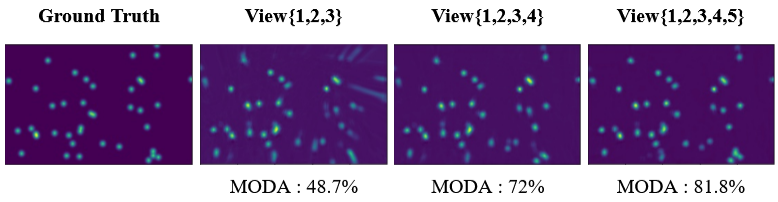

Figure 9: Occupancy maps for varying number of cameras on MultiViewX dataset when trained on seven cameras and tested on varying subsets of the cameras.

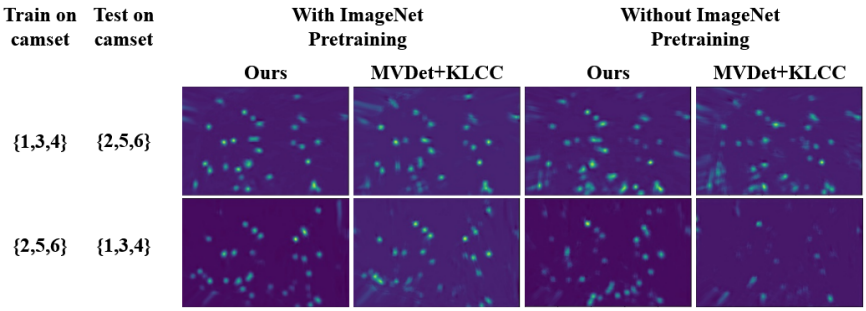

Figure 10: Result occupancy maps for cross subset evaluation from WildTrack dataset.

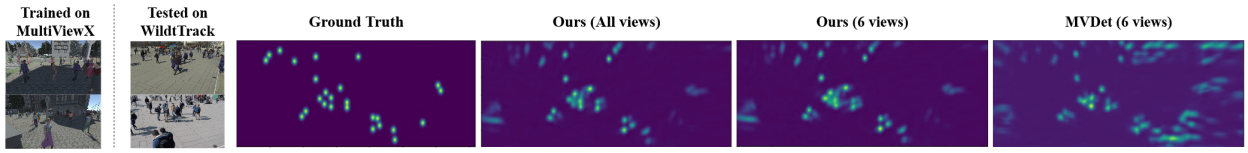

Figure 11: Occupancy maps obtained by inferencing from WildTrack dataset where the models where trained on synthetic dataset (MultiViewX).

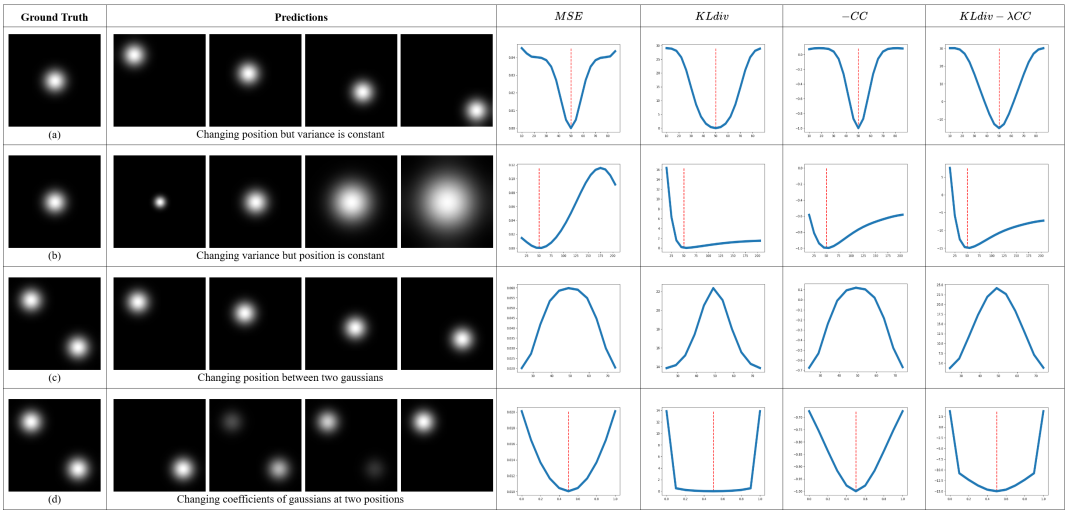

Figure 12: Inspired by the work in [4, 27], we synthetically varied heat map predictions w.r.t the ground-truth in order to quantify effects on the loss functions. The ground-truth and illustrative predictions are shown on the left. The right plots shows the values of the loss function. Each row corresponds to varying a single parameter value of the prediction: (a) location on a single mode, (b) variance, (c) location between two modes, (d) relative weights between two modes. The x-axis spans the parameter range and the dotted red line corresponds to the ground-truth (if applicable).