Bringing Generalization to Deep Multi-View Pedestrian Detection

Jeet Vora[1] Swetanjal Dutta[1] Kanishk Jain[1] Shyamgopal Karthik[2] Vineet Gandhi[1]
[1]CVIT, IIIT Hyderabad [2]University of Tübingen
{jeet.vora, swetanjal.dutta, kanishk.j}@research.iiit.ac.in
shyamgopal.karthik@uni-tuebingen.de, vgandhi@iiit.ac.in

Abstract

Multi-View Detection (MVD) is highly effective for occlusion reasoning in a crowded environment. 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 evaluate them*: generalization with i) a varying number of cameras, ii) varying camera positions, and finally, iii) to new scenes. We find that existing state-of-the-art models show poor generalization by overfitting to a single scene and camera configuration. To address the concerns: (a) we propose a novel Generalized MVD (GMVD) dataset, assimilating diverse scenes with changing daytime, camera configurations, and a varying number of cameras, and (b) we discuss the properties essential to bring generalization to MVD and propose a barebones model incorporating them. We present comprehensive set of experiments on WildTrack , MultiViewX and the GMVD datasets to motivate the necessity to evaluate the generalization abilities of MVD methods and to demonstrate the efficacy of the proposed approach. The code and dataset are available at* `https://github.com/jeetv/GMVD`.

1. Introduction

"Essentially all models are wrong, but some are useful."

— George E. P. Box

In this work, we pursue the problem of Multi-View Detection (MVD), a mainstream solution for dealing 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 [9] in the ground plane (bird's eye view). The solutions of MVD has evolved from classical methods [9, 3, 1], to hybrid approaches [18] to end-to-end trainable deep learning architectures [13]. Expectedly, the current landscape of MVD is dominated by end-to-end trainable deep learning methods [13, 12, 27]. We argue that by *training and testing on homogeneous data*, current deep MVD methods have overlooked critical fundamental concerns, and to render them *useful*, the focus should shift towards their generalization abilities.

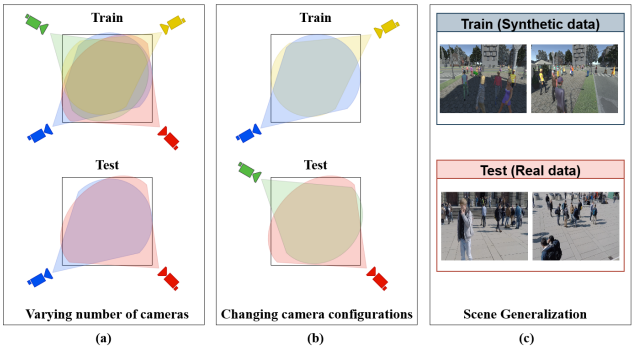

Figure 1. Three forms of generalization required in MVD: (a) varying number of cameras, (b) different camera configurations, and (c) generalizing to new scenes.

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 work on 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 span the dedicated area.

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, the existing deep learning-based MVD methods are primarily trained and tested with the same camera

Figure 2. The train and test sets of WildTrack (first row) and MultiViewX datasets (second row) have significant overlap. We show the last image of the training set (left) and the first image of the test set (right). In both datasets, the appearance of several pedestrians is already seen in the training set. In WildTrack , there are many static pedestrians as well.

configuration, on the same scene, using the same number of cameras. Even the environmental conditions (time, weather, etc.) are similar across train and test splits. For instance, the most commonly used WildTrack dataset [6] includes a 200-second recording from all cameras, where the first 3 minutes are used for training and the rest of the 20 seconds are used for testing. We argue that the current state-of-the-art (SOTA) methods are seriously hindered from the deployment perspective. The current models [13, 12, 27] break if a camera malfunctions and is unavailable during inference. Additionally, they require retraining if a camera needs to be added to the setup. Furthermore, our experiments show that the performance significantly drops if the camera positions or the scene is varied. The SOTA models also overfit to the order in which the cameras are sent to the model (i.e they are not *permutation-invariant*).

The absence of a diverse dataset is a major shortcoming. The available datasets: WildTrack (real) and MultiViewX (synthetic), comprise a single short sequence, where initial frames are used for training and later for testing. In Fig. 2, we show that the evaluation strategy in both datasets is unreliable and prone to overfitting. To this end, we propose a novel Generalized MVD (GMVD) dataset. Given the privacy concerns, COVID restrictions, hardware setup difficulties, the requirement of manual annotations, etc., we believe curating a sizeable synthetic dataset is the right way forward. Hence, we use Unity and the Grand Theft Auto (GTA) game environment to capture the GMVD dataset. It includes about 53 sequences captured in 7 different scenes with significant variations in camera configuration, weather, lighting conditions, pedestrian appearance, etc. The number of cameras also varies across scenes. We use 6 scenes for training and 1 scene for testing. The proposed GMVD dataset sets up a new benchmark for evaluating MVD with generalization. It further allows reserving valuable real-world footage [6] directly for testing.

Furthermore, we suggest design guidelines to ensure the practical usability of Deep MVD methods. We demonstrate that permutation invariance, transfer learning, and regularization are vital for generalization. We improve the baseline architecture [13] with appropriate changes and establish SOTA generalization for MVD. We want to emphasize that our work pivots around the barebone baseline architecture and does not claim any significant architectural novelty. The focus of our work is to address the critical limitations of Deep MVD models from an application perspective. Overall, our work makes the following contributions:

1. We conceptualize and emphasize the importance of generalization in MVD and propose a novel GMVD dataset for the same.

2. We highlight the shortcomings of the current evaluation methodology and propose novel experimental setups on existing datasets.

3. We adapt the baseline architecture to bring generalization to deep MVD. We show that *permutation invariance* is crucial for MVD and average pooling is one minimal way to achieve it. We propose a novel *drop view regularization*.

4. We back our claims using an extensive set of experiments and ablation studies. We show staggering improvements in scene and configuration generalization, paving the way for a practicable MVD.

2. Related Work

2.1. Classical Methods

Seminal work by Fleuret *et al*. [9] casts MVD as predicting occupancy probabilities over a discrete grid, an idea which has stood the test of time. The 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 aggregation [9, 3, 1]. The classical methods, however, observe a gradual degradation in detection performance with increased crowds, as the background subtraction becomes less effective with an increase in crowds and clutter. Some methods do away with background subtraction and rely on handcrafted classifiers [26] instead.

2.2. Anchor-based MVD

Anchor-based MVD methods replace background subtraction with anchor-based deep pedestrian detectors like Faster R-CNN [25], SSD [21] and YOLO [24]. Some of

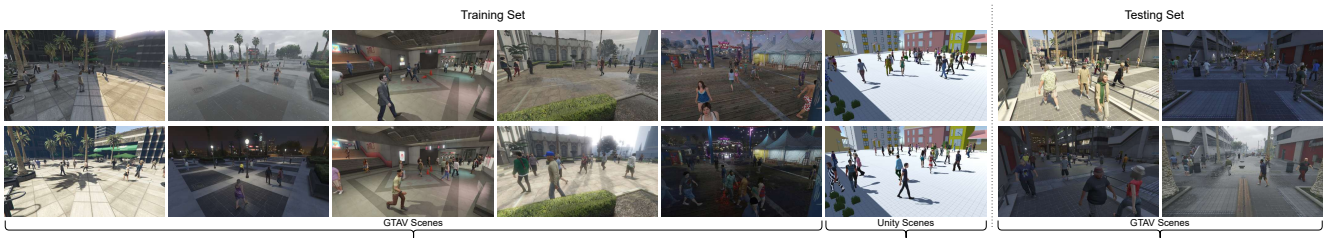

Figure 3. The proposed GMVD Dataset includes seven scenes. Each column illustrates frames from one of the views from two different sequences of the same scene. The first six scenes are used for training, and the last scene with two configurations are reserved for testing. Additionally, there are noticeable lighting and weather variations within each scene.

Table 1. Dataset Statistics for various MVD datasets. Our proposed GMVD dataset is the largest and most diverse dataset on a variety of metrics. Avg. coverage refers to the average number of cameras that cover each point on the ground plane.

Dataset	Track Labels	IDs	# Scenes	# Training Frames	# Testing Frames	# Cameras	# Sequences	Avg. Coverage
WildTrack	✓	313	1	360	40	7	1	3.74
MultiViewX	✓	350	1	360	40	6	1	4.41
GMVD (Ours)	✓	2800	7	4983	1012	3, 5, 6, 7, 8	53	2.76 - 6.4

these methods process each view separately [30] and some process them simultaneously [2, 7]. The inaccuracies in the pre-defined anchor boxes [18] 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 significant errors. Moreover, some of the anchor-based methods still rely on operations outside of Convolutional Neural Networks (CNNs), requiring working out a balance between different potential terms [2].

2.3. End-to-end Deep MVD

MVDet [13] 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 manual tuning of CRF potentials, reliance on pre-defined 3D anchor boxes and projection errors from monocular detectors. It aggregates projected features from a ResNet [11] backbone using three convolutional layers to predict the final occupancy map. MVDet achieves notable improvement over the preceding anchor-based methods (over 14% improvement on the WildTrack dataset [6]). The idea from [13] was further enhanced by using deformable transformers [32] to improve the feature aggregation in MVDeTr [12]. More recently, SHOT [27] introduced a combination of homographies at multiple heights to improve the quality of the projections.

3. Proposed Dataset

We propose a new MVD dataset incorporating the three forms of generalization discussed above (Fig. 1). Some example frames from the proposed Generalized Multi-View Detection (GMVD) dataset are illustrated in Fig. 3. The GMVD dataset contains diverse non-overlapping scenes within and across training and test sets. In contrast, the existing MVD datasets WildTrack and MultiViewX include noticeable overlap across train and test splits (single scene, pedestrians appearance, and location), encouraging existing MVD methods to overfit the dataset-specific aspects and thus hindering their practicality. The GMVD dataset, by its design, prevents overfitting from happening by keeping a clear separation in train and test splits.

Capturing a real-world MVD dataset is difficult, primarily because of privacy concerns. The COVID restrictions also restrict crowded human capture. Moreover, such a dataset requires significant manual annotation effort. Consequently, we curate the GMVD dataset using synthetic environments. The GMVD dataset is curated using Grand Theft Auto V (GTAV) and Unity Game Engine. We employ two different environments to avoid overfitting to a single synthetic data generation source. This reasoning is aligned with recent works [10, 31] which utilize multi-source datasets to improve generalization performance. The GMVD dataset includes seven different scenes, one indoor (subway) and six outdoors. One of the scenes are reserved for the test split. We vary the number of total cameras in each scene and provide different camera configurations within a scene.

Additional salient features of GMVD include daytime variations (morning, afternoon, evening, night) and weather variations (sunny, cloudy, rainy, snowy). We generate multiple short sequences for each scene while randomly varying the daytime and the weather. The generation of multiple random sequences ensures diversity, as different pedestrians (with different clothing and appearance) are picked in each case. The dataset also includes significant variations in lighting conditions. Local illumination sources come into

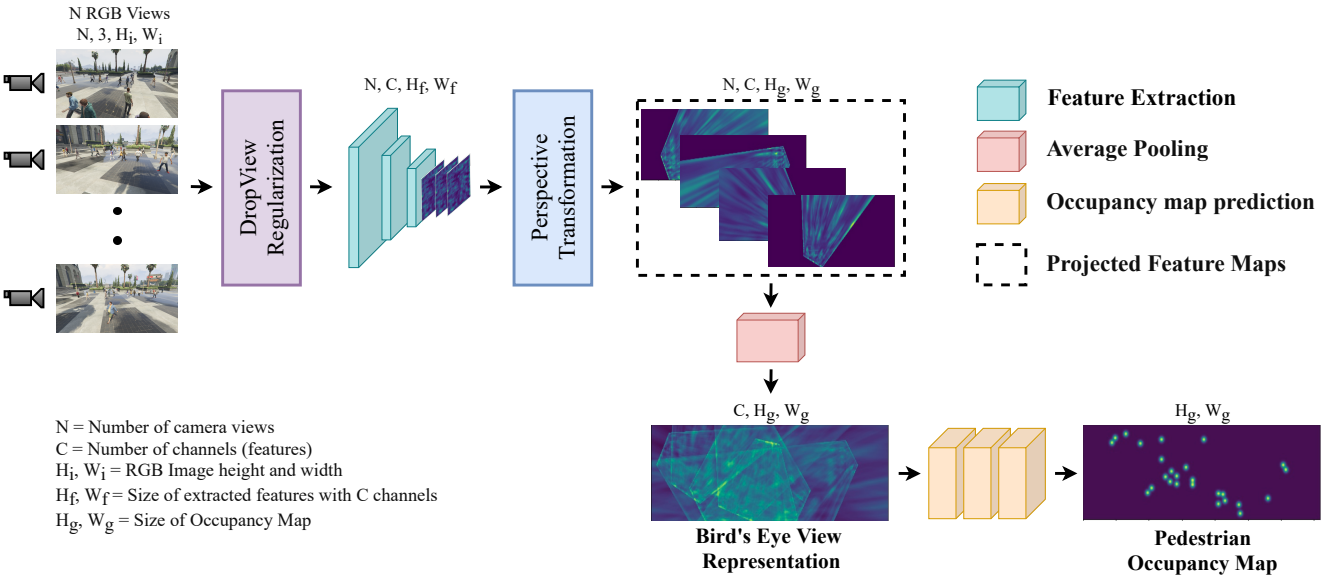

Figure 4. Our proposed architecture: ResNet features are extracted from the input views, which are then projected to the top view. Following this, the projected features across views are pooled and then the final occupancy map is predicted. The use of average pooling across views is crucial in ensuring that our proposed architecture can work for an arbitrary number of views.

play due to the presence of indoor and night scenes. We compare our dataset with the existing ones in Table 1. avg. coverage represents the average amount of cameras observing each location. For GMVD, avg. coverage varies from 2.76-6.4 cameras depending on the scene. In addition to the discussed variations, GMVD is advantageous due to the dataset size, especially in terms of the total number of individual sequences.

We further encourage future methods to train on the GMVD dataset and test their performance on sparsely available, difficult to capture real-world datasets like WildTrack .

Dataset Generation: We used Script Hook V [4] library to interface with the GTAV environment. For each scene, camera positioning and orientation were determined manually so as to increase the camera coverage. All the cameras were positioned above the humans' average height. Due to hardware limitation, it is commonplace to have a small synchronization delay in real-world multi-camera setups. To emulate such realistic scenario, we induce a small synchronization error (20-100 ms) between different camera views [17]. A ground plane was defined for each location, partially overlapping with each camera's field of view. Only pedestrians inside the ground plane were considered for multi-view detection. We relied on the GTA's navigational AI engine to avoid collision and to obtain realistic pedestrian behavior.

In the Unity environment, scenes are manually curated by putting together 3D models of street, buildings and other props. We used the PersonX [29] 3D human models to create the pedestrians. To avoid collision errors (which are present in MultiViewX dataset), for each frame, pedestrians were spawned at random locations within the region of interest.

Since both the environments are synthetic, the 3D-2D correspondences were directly available from the game engines. We use similar procedure as [13] for camera calibration.

Track Labels: Our work focuses on a comprehensive analysis of the problem of Multi-View Detection. However, the proposed dataset can also be useful for the task of multi-view pedestrian tracking. To this end, for the sequences generated from the GTAV environment, we collect the track labels while capturing the data. While we do not use track labels in this work, we provide them with the dataset, which will be beneficial for the community in the future. We provide a total of 125000 frames with track labels. The GTAV frames for the GMVD dataset are regularly sampled from these densely annotated sequences.

4. Proposed Method

We propose an anchor-free deep MVD method along the lines of [13, 12, 27] 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. 4. 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.

4.1. Feature Extraction and Perspective Transformation

Feature Extractor: We use a ResNet18 [11] 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:

$$I = s \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = K[R|t] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = P \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (1)$$

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

4.2. 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 $(\text{where}, i = 1, 2, ..., N)$. 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,

$$F = \frac{\sum_{i=1}^{N} fm_i}{N}. \quad (2)$$

While there can be many other alternatives to average pooling, we opt for this solution, primarily because it is *permutation-invariant*. Unlike existing methods [13, 12, 27], 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.

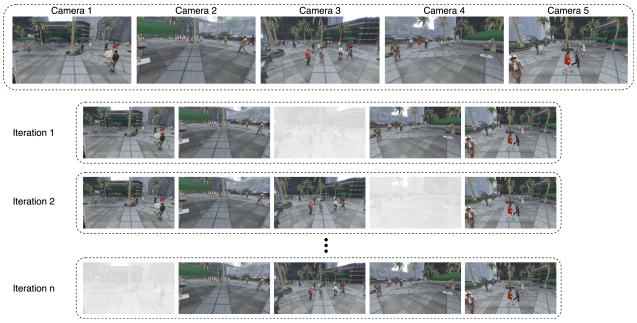

Figure 5. An illustration of our proposed DropView regularization

DropView Regularization: Inspired by Dropout [28] 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 to discard during training iterations, as illustrated in Fig. 5. The occupancy map prediction step is done with all the remaining views. We provide a detailed analysis of the effect of this regularization strategy in our experiments.

Occupancy Map Prediction: Similar to MVDet [13], we use 3 dilated convolutional layers to predict the occupancy map of size (H_g, W_g).

4.3. Loss Function

The loss function compares the output probabilistic occupancy map (p) with the ground-truth (g). Inspired by the work on saliency estimation in images and videos [5, 23, 14], 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:

$$L(p, g) = \frac{\sigma(p, g)}{\sigma(p) \times \sigma(g)} - \sum_i g_i \log\left(\frac{g_i}{p_i}\right), \quad (3)$$

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. The loss function was selected empirically using the scene generalization experiment, i.e. training on MultiViewX and testing on WildTrack , where using KLDiv+CC gave best results (compared with MSE, CC or KLDiv alone).

5. Experiments

5.1. Experimental setup

Datasets: In addition to our proposed GMVD dataset, we use the WildTrack and MultiViewX datasets. The *WildTrack* dataset consists of 7 static calibrated cameras with

Table 2. Comparison against the state-of-the-art methods. Our method refers to the proposed model in Section 4. We made five runs for some of the experiments and the variances are presented in the bracket.

Method	ImageNet (pre-train)	WildTrack				MultiViewX			
		MODA	MODP	Prec	Recall	MODA	MODP	Prec	Recall
RCNN Clustering [30]	×	11.3	18.4	68.0	43.0	18.7	46.4	63.5	43.9
POM-CNN [9]	×	23.2	30.5	75.0	55.0	-	-	-	-
Lopez-Cifuentes *et al*. [22]	×	39.0	55.0	-	-	-	-	-	-
Lima *et al*. [20]	×	56.9	67.3	80.8	74.6	-	-	-	-
DeepMCD [7]	×	67.8	64.2	85.0	82.0	70.0	73.0	85.7	83.3
Deep-Occlusion [2]	×	74.1	53.8	95.0	80.0	75.2	54.7	97.8	80.2
MVDet [13]	×	88.2	75.7	94.7	93.6	83.9	79.6	96.8	86.7
MVDeTr [12]	✓	91.5	82.1	97.4	94.0	93.7	91.3	99.5	94.2
SHOT [27]	×	90.2	76.5	96.1	94.0	88.3	82.0	96.6	91.5
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	✓	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)
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)

overlapping fields of view, covering an area of 12×36 m^2. The dataset comprises a single 200 second sequence annotated at 2 fps. The image resolution is 1080×1920 pixels. The ground plane grid is discretized into a 480×1440 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 1080×1920 annotated at 2 fps for ground-truth covering an area of 16×25 m^2. The ground plane grid is discretized into a 640×1000 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 [13, 6].

Evaluation metrics: We use the standard evaluation metrics proposed in [6]. *Multiple Object Detection Accuracy* (MODA) is the primary performance indicator that 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 the true positives.

SOTA comparisons: We compare against nine different methods. The set includes one monocular object detection baseline (referred to as RCNN clustering [30]); a classical probabilistic occupancy map method [9]; four anchor-based methods [20, 2, 7, 22] and three recent end-to-end trainable deep MVD approaches [13, 12, 27]. For generalization experiments, we only compare against the recent state-of-the-art methods MVDet [13], MVDeTr [12] and SHOT [27].

5.2. 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 $(512, H_g, W_g)$. H_g and W_g vary from scene-to-scene, depending on the area of ground plane.

The 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 1, SGD optimizer with momentum = 0.9 has been used with one-cycle learning rate scheduler. A probability of τ or more on the occupancy grid is considered a detection. For GMVD experiments, τ is determined using MultiViewX as a validation set, and for other experiments, we use $\tau = 0.4$ in alignment with the previous works. Non-Maximal Suppression (NMS) is applied with a spatial resolution of 0.5m. All training and testing have been performed on a single Nvidia GTX 1080 Ti GPU. Unless specifically mentioned, we always use pre-trained ImageNet [8] weights while training our proposed model.

5.3. Results

Like prior works, we evaluate our approach on the WildTrack and MultiViewX datasets in Table 2. We find that our proposed models attains satisfactory performance on the test sets of both WildTrack (best MODA score of 87.2) and MultiViewX (best MODA score of 88.2). This is slightly worse than the recently proposed methods [12, 27], but is far superior to the performance of the classical and the anchor-based MVD methods. However, we would like to highlight that the traditional evaluation protocol is highly misleading since the train and test sets have significant overlap, thereby

Table 3. Results for evaluating with a varying number of cameras. The model is trained on all 7 cameras on WildTrack , and is tested on 2 different sets of 4 cameras each.

	Inference on {1,3,5,7}				Inference on {2,4,5,6}			
Method	MODA	MODP	Prec	Recall	MODA	MODP	Prec	Recall
MVDet	38.9	71.5	**93.8**	41.6	16.2	47.6	80.3	21.4
MVDeTr	55.8	**76.7**	80.8	73.2	34.6	69.2	68.6	63.8
SHOT	66.6	75.1	91.0	73.9	46.3	67.8	88.2	53.5
Ours	76.5	74.0	91.7	84.0	**79.3**	71.4	**91.1**	87.9
Ours (DropView)	**77.0**	74.5	90.3	**86.2**	79.2	72.5	88.6	**90.9**

Table 4. Experiments on the WildTrack dataset with changing camera configurations

		Inference on {2,4,5,6}				Inference on {1,3,5,7}			
	Method	MODA	MODP	Prec	Recall	MODA	MODP	Prec	Recall
Trained on camera set {2,4,5,6}	MVDet	**85.2**	72.2	92.6	**92.5**	43.2	68.2	**94.6**	45.8
	MVDeTr	75.4	**79.5**	**96.9**	77.9	41.7	**73.7**	92	45.7
	SHOT	81.9	74.1	94.1	87.4	51.4	72.5	94.4	54.6
	Ours	81.8	73.5	93.5	87.9	66.5	71.4	94.3	70.8
	Ours (DropView)	84.0	72.9	92.4	91.6	**75.1**	71.1	94.3	**79.9**
Trained on camera set {1,3,5,7}	MVDet	27.8	**68.7**	**90.8**	31.0	78.2	73.6	89.5	**88.6**
	MVDeTr	5.6	65.5	62.4	14.0	72.5	**78.9**	95	76.5
	SHOT	15.3	62.9	89.2	17.4	79.7	76.4	**95.7**	83.5
	Ours	52.4	67.4	81	68.5	76.4	74.6	91.5	84.1
	Ours (DropView)	**62.6**	67.4	86.7	**73.9**	**80.8**	74.0	94.2	86

encouraging overfitting. Therefore, we emphasize the evaluation across a varying number of cameras, changing camera configurations, and on new scenes.

Generalization to Varying Number of Cameras: An interesting scenario that can potentially occur in practical scenarios is the loss of some camera feeds due to various issues. In this case, a model trained with 7 cameras, may need to be able to perform inference with just 4 cameras. To simulate this setting, we train all the models (MVDet, MVDeTr, SHOT and Ours) on all 7 cameras and test them on 2 different sets of 4 cameras ({1,3,5,7},{2,4,5,6}) in Table 3. Our proposed model is able to naturally work in this setting without any issues. For MVDet, MVDeTr, and SHOT, we randomly duplicate 3 of these views to ensure that 7 views are available. We observe that the performance of MVDet, MVDeTr, and SHOT degrades drastically when evaluated in this setting. When trained with the DropView regularization, our proposed model outperforms these methods by a huge margin (MODA of 77.0 vs 66.6 and 79.2 vs 46.3). This experiment clearly illustrates the need for the architectures to automatically work with an arbitrary number of views. Furthermore, since MVDet, MVDeTr, and SHOT learn a separate spatial aggregation module for each view, the spatial aggregation module overfits to the order of input cameras (indicated by the significant performance variations across the two sets). Future works should ensure that the model has *permutation invariance* to the order of input views in addition to working with an arbitrary number of views.

Generalization to New Camera Configurations: Another practical scenario that we explore is when the camera positions are varied between the train and test sets. We train all the models on two sets of camera views and then test the trained models on both sets. The results are provided in Table 4. When the models are evaluated on the same camera configuration, all the models have satisfactory performance. However, when evaluated on the different camera configuration, MVDet, MVDeTr, and SHOT see a huge degradation in performance. Our model is fairly robust to the changing camera configuration. Especially when trained with DropView regularization, the resulting model outper-

Table 5. 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 five rows.

Method	Inference on total cameras	ImageNet (pre-train)	MODA	MODP	Prec	Recall
MVDet	6	×	17.0	65.8	60.5	48.8
MVDeTr	6	✓	50.2	69.1	74.0	77.3
SHOT	6	×	53.6	72.0	75.2	79.8
Ours	6	✓	60.1	72.1	75.6	**88.7**
Ours (DropView)	6	✓	66.1	72.2	82.0	84.7
Ours	7	✓	69.4	72.96	**83.7**	86.14
Ours (DropView)	7	✓	**70.7**	**73.8**	**89.1**	80.6

Table 6. Changing configuration and scene generalization experiment on the setting introduced in [27]

Method	MODA	MODP	Prec	Recall
MVDet	33.0	76.5	64.5	73.4
MVDeTr	56.5	70.8	85.0	68.6
SHOT	49.1	**77.0**	73.3	**77.1**
Ours	57.8	76.5	88.7	66.3
Ours (DropView)	**66.1**	75.8	**89.3**	75.2

Table 7. Comparison and evaluation of our method when trained on GMVD training set: first column shows the result on GMVD test set and second column is when tested on WildTrack dataset.

Method	GMVD				WildTrack			
	MODA	MODP	Prec	Recall	MODA	MODP	Prec	Recall
MVDet	50.5	72.8	83.6	64.7	69.0	71.1	88.4	79.5
Ours	**68.2**	**76.3**	**91.5**	**75.5**	**80.1**	**75.6**	**90.9**	**89.1**

forms all other models by over 20 percentage points.

Scene Generalization: Finally, an important concern with the practical utility of MVD methods is that since real-world data is scarce, a trained model should be able to generalize to new scenes. We first evaluate the scene generalization abilities of the MVD methods by training them on MultiViewX and evaluating them on WildTrack in Table 5. Our proposed model is able to utilize the extra camera present in the WildTrack dataset and achieves a MODA score of 70.7. This further highlights the benefits of an architecture that works with arbitrary number of views, since the performance during inference can be enhanced by adding more view. However, even without the additional view, our model achieves a MODA score of 66.1, which is much higher than SHOT which only achieves a MODA score of 53.6.

In addition to this, we perform the scene generalization experiment proposed in [27] where the MultiViewX scene is split into two halves, and each half is covered using 3 cameras each. In this setting as well (Table 6), our proposed approach with DropView regularization has a MODA score of 66.1, which is significantly higher than both SHOT (49.1) and MVDeTr (56.5).

GMVD Benchmark: Having shown that our proposed model is capable of comprehensive generalization abilities, we benchmark our proposed approach on the GMVD dataset (Table 7). We train our model on the training set of the GMVD dataset and use MultiViewX dataset for validation. Since each sequence in the training set has a different number of cameras, *none* of the existing methods can be applied to this setting, since they can be trained only on a *fixed* set of cameras. However, to have a comparison with prior work, we adapt MVDet to work in this setting by duplicating frames. When evaluated on WildTrack , our model is able to achieve a MODA score of 80.1, which is a significant improvement over the results from training on MultiViewX (70.7). This is only marginally less than our best result on the WildTrack dataset (87.2), despite the WildTrack train set heavily overlapping with the test set. This indicates that training on synthetic data alone might be sufficient for this task when coupled with unsupervised domain adaptation techniques. When evaluated on GMVD test set, our model achieves a MODA score of 68.2. This highlights the difficulty of the GMVD test set, compared to WildTrack and MultiViewX , resulting from a distinct train-test split and the presence of extensive variations. We believe that our dataset can serve two important purposes. The first is as a diverse, synthetic dataset from which a model can be adapted to real-world data. The second is that the GMVD dataset itself can be a challenging benchmark to evaluate the generalization capabilities of MVD methods. In this setting, MultiViewX being used for validation is ideal, since this ensures that no information from the test set is leaked during training.

6. Discussion and Future work

The biggest limitation in the field of Multi-View Detection is that real-world capture of data is extremely challenging due to the difficulty in collecting a dataset with people in addition to the challenges involved in the hardware setup and annotations. The absence of a large, diverse benchmark significantly hampers the progress of this topic. Therefore, the existing WildTrack dataset is extremely valuable for the community. However, due to its limited size and variety, it is not suitable for training and should only be used to evaluate the generalization abilities of the models. In this regard, we hope that our proposed dataset and our barebone model serves as a useful tool in bridging the gap between the theory and real-world application of MVD methods. In our work, we have not explored the use of unsupervised domain adaptation techniques to bridge the gap between the feature distributions of the synthetic and real datasets and the direction is left for exploration in the future work.

7. 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. In addition to this, we propose a diverse, synthetic, but realistic dataset which can be used both as an evaluation benchmark, as well as a training dataset for various MVD methods. Overall, we hope our work plays a crucial role in steering the community towards more practical Multi-View Detection solutions.

References

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

[2] Pierre 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] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(9):1806–1819, 2011.

[4] Alexander Blade. Script Hook V. `http://www.dev-c.com/gtav/scripthookv/`, 2008. [Online; accessed 19-July-2008].

[5] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 41(3):740–757, 2018.

[6] Tatjana Chavdarova, Pierre Baqué, Stéphane Bouquet, Andrii Maksai, Cijo Jose, Timur M. Bagautdinov, Louis Lettry, Pascal V. Fua, Luc Van Gool, and François Fleuret. Wildtrack: A multi-camera hd dataset for dense unscripted pedestrian detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5030–5039, 2018.

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

[8] Jia Deng, W. Dong, R. Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[9] 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 (TPAMI)*, 30:267–282, 2008.

[10] Rui Gong, Dengxin Dai, Yuhua Chen, Wen Li, and Luc Van Gool. mdalu: Multi-source domain adaptation and label unification with partial datasets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8876–8885, 2021.

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

[12] Yunzhong Hou and Liang Zheng. Multiview detection with shadow transformer (and view-coherent data augmentation). In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1673–1682, 2021.

[13] Yunzhong Hou, Liang Zheng, and Stephen Gould. Multiview detection with feature perspective transformation. In *European Conference on Computer Vision (ECCV)*, 2020.

[14] Samyak Jain, Pradeep Yarlagadda, Shreyank Jyoti, Shyamgopal Karthik, Ramanathan Subramanian, and Vineet Gandhi. Vinet: Pushing the limits of visual modality for audio-visual saliency prediction. *arXiv preprint arXiv:2012.06170*, 2020.

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

[16] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, John 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 (TPAMI)*, 31:319–336, 2009.

[17] Philipp Kohl, Andreas Specker, Arne Schumann, and Jurgen Beyerer. The mta dataset for multi-target multi-camera pedestrian tracking by weighted distance aggregation. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[18] Tao Kong, Fuchun Sun, Huaping Liu, Yuning Jiang, Lei Li, and Jianbo Shi. Foveabox: Beyond anchor-based object detection. *IEEE Transactions on Image Processing*, 29:7389–7398, 2020.

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

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

[21] W. Liu, Dragomir Anguelov, D. Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and A. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision (ECCV)*, 2016.

[22] Alejandro López-Cifuentes, Marcos Escudero-Viñolo, Jesús Bescós, and P. Carballeira. Semantic driven multi-camera pedestrian detection. *ArXiv*, abs/1812.10779, 2018.

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

[24] Joseph Redmon, S. Divvala, Ross 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.

[25] Shaoqing Ren, Kaiming He, Ross 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 (TPAMI)*, 39:1137–1149, 2015.

[26] G. Roig, X. Boix, Horesh Ben Shitrit, and P. Fua. Conditional random fields for multi-camera object detection. *2011 International Conference on Computer Vision (ICCV)*, pages 563–570, 2011.

[27] Liangchen Song, Jialian Wu, Ming Yang, Qian Zhang, Yuan Li, and Junsong Yuan. Stacked homography transformations for multi-view pedestrian detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6049–6057, 2021.

[28] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[29] Xiaoxiao Sun and Liang Zheng. Dissecting person re-identification from the viewpoint of viewpoint. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[30] Yuanlu Xu, Xiaobai Liu, Yang Liu, and Song-Chun Zhu. Multi-view people tracking via hierarchical trajectory composition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4256–4265, 2016.

[31] Yuyang Zhao, Zhun Zhong, Fengxiang Yang, Zhiming Luo, Yaojin Lin, Shaozi Li, and Nicu Sebe. Learning to generalize unseen domains via memory-based multi-source meta-learning for person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6277–6286, June 2021.

[32] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.