Two-level Data Augmentation for Calibrated Multi-view Detection

Martin Engilberge* Haixin Shi* Zhiye Wang Pascal Fua
EPFL, Lausanne, Switzerland
firstname.lastname@epfl.ch

Abstract

Data augmentation has proven its usefulness to improve model generalization and performance. While it is commonly applied in computer vision application when it comes to multi-view systems, it is rarely used. Indeed geometric data augmentation can break the alignment among views. This is problematic since multi-view data tend to be scarce and it is expensive to annotate.

In this work we propose to solve this issue by introducing a new multi-view data augmentation pipeline that preserves alignment among views. Additionally to traditional augmentation of the input image we also propose a second level of augmentation applied directly at the scene level. When combined with our simple multi-view detection model, our two-level augmentation pipeline outperforms all existing baselines by a significant margin on the two main multi-view multi-person detection datasets WILDTRACK and MultiviewX.

1. Introduction

In recent years deep learning models have been widely adopted in the computer vision fields. One of the reasons for this wide adoption is the generalization ability of gradient-based models [12]. While such models generalize well, they are still subject to overfitting their training data. Multiple methods have been proposed to combat overfitting. Some focus on the model design *e.g.* dropout layer [19] or batch normalization [10], while others such as data augmentation [16] directly tackle one of the root causes of overfitting: overparametrization due to limited data. While data augmentation has been widely used and studied in a variety of fields, it is rarely used in the multi-view context. Indeed in a multi-view setup geometric data augmentation can easily break the alignment among views.

Most multi-view people detection methods do not employ data augmentation [23, 5, 3]. While less than ideal, this wasn't the main limiting factor in those earlier methods, since they only used deep learning models for initial monocular predictions which could be pre-trained using monocular data augmentation. However recent approaches [18, 9] have adopted end-to-end architectures directly predicting detections in the ground plane (top view) from multi-view inputs. When trained from scratch, such methods can greatly benefit from data augmentation.

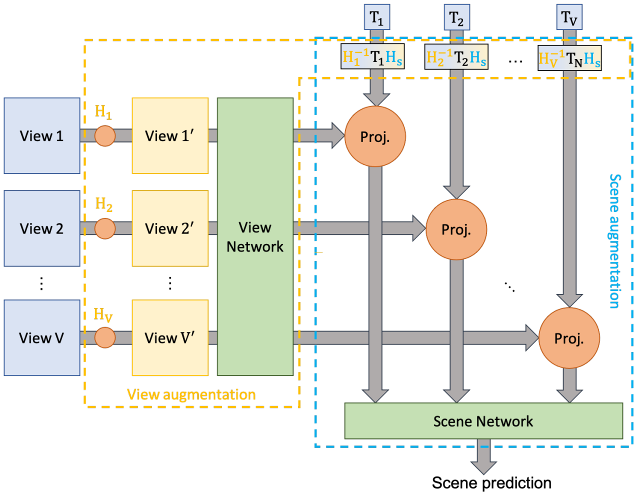

Figure 1: **Data augmentation in a multi-view setting** Illustration of a multi-view model combined with our multi-view data augmentation pipeline. The input of the model consists of multiple images coming from different viewpoints. Each view is associated with a transformation T_v that projects the corresponding view to a common scene representation where the different views are aligned. In yellow, our view based augmentation mechanism applies data augmentation (H_v) independently on each view and updates the original transformation to preserve alignment. It helps reduce the view network from overfitting the training data. Circled in light blue, our new scene augmentation is applied directly in the aligned scene representation by updating the projection transformation T_v with a scene augmentation H_S.

[*] These authors contributed equally to this work.
Project code at https://github.com/cvlab-epfl/MVAug

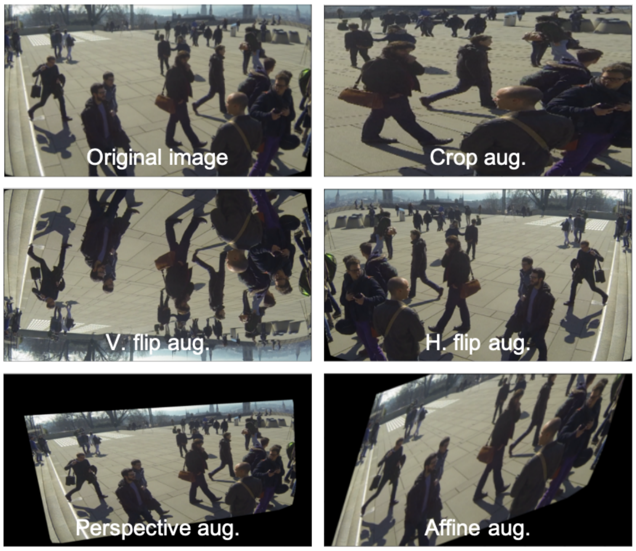

Figure 2: **Geometric data augmentation** Visualization of the different kinds of geometric data augmentations, top left is the un-augmented image.

In this paper we propose to address the issue by introducing a data augmentation pipeline for multi-view model illustrated in Fig. 1. Our pipeline is able to augment each view independently while preserving the overall alignment among views (view augmentation). Additionally we introduce a new type of multi-view augmentation, applied directly at the scene level, we call it scene augmentation. Each type of augmentation helps reduce overfitting of different parts of the network. We demonstrate the benefit of both types of augmentation on the multi-view multi-object detection task and show that when combined with our model it outperforms state-of-the-art multi-view methods [9, 18, 8] on the challenging WILDTRACK [2] and MultiviewX [9] datasets.

2. Related works

In this section we briefly introduce previous work on multi-view detection and discuss existing approaches for data augmentation.

Multi-view multi-person detection Over the year multiple monocular detection methods have been proposed such as the R-CNN family of models [6, 14] that predict bounding boxes from single input image using a two-stage architecture. More recently single stage anchor-free approaches [24, 20] have yielded promising results.

However when it comes to detecting people in crowded scene they tend to miss heavily occluded people. To remedy this problem multiple works have proposed to detect people in a multi-view setup. Using multiple calibrated cameras [21] reduces the likelihood to suffer from occlusion in every view. To aggregate multiple views most existing methods predict pedestrian occupancy map on the ground plane [5, 1, 3, 9, 8, 18]. While final detections are done on the ground plane, some models first predict monocular detections [23, 5] before projecting and aggregating the results. Others choose to combine view aggregation and prediction in a single step by for example learning jointly a CNN and Conditional Random Fields (CRF) [1, 15]. More recent approaches learn end-to-end neural networks, where projections on the ground plane are part of the networks. One such approach [8] proposes a view aggregation network that leverage an attention mechanism as part of a transformer network to select the most relevant part of each view to generate the final detection map. In [18] instead of a single projection on the ground plane, they propose to use multiple projections onto planes at different heights in order to approximate a 3d world coordinate system.

Data Augmentation Data augmentation is widely used to improve generalization of neural networks [16]. During training it provides artificial samples generated by altering original data in multiple ways. Traditional methods can be roughly divided into two sorts. First, the geometric transformations methods, including flipping, cropping, rotation and translation, tackle positional bias in training data. The other is photometric transformations which performs augmentations in the color channels space or injects noise into images[17]. With the booming of deep learning, many image data augmentation methods combined with deep learning have been developed. Feature space augmentation proposed by DeVries and Taylor [4] extracts vectors from low-dimensional feature maps and adds noise, interpolates, and extrapolates. Adversarial training uses the samples generated by the rival network for augmentation [17].

Image augmentation in detection setting faces multiple challenges. When bounding boxes ground truth are used, data augmentation cannot be applied directly, special augmentation is required to correctly preserve ground truth boxes [25]. Anchor-free detection models are free of such limitation, however, when used in a multi-view setting combining them with data augmentation can be the cause of inconsistency among views. For such reasons traditional geometric image augmentations are rarely used in multi-view settings[22, 9]. To ensure the alignment among different views in multi-view pedestrian detection, Hou *et al.* [8] proposed to augment each view individually via geometric transformation and then reverse the augmentation before projection. Having to reverse the data augmentation is one drawback of such approach, by doubling the number of projection happening in the network it introduces noises in the features due to repeated bilinear interpolation.

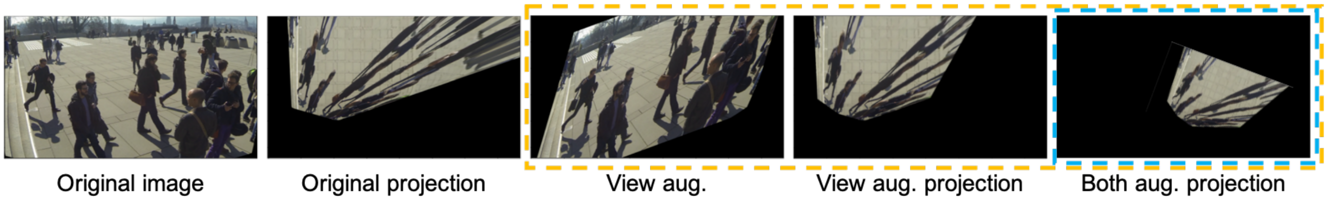

Figure 3: **Visualization of view and scene augmentation, and their effect on the ground plane projection** The two left images correspond to the original image and its corresponding projection on the ground plane. The third and fourth images visualize the effect of affine view augmentation, and the projection on the ground plane of the augmented image. Note that between second and forth image the alignment is preserved. The last image visualizes the effect of adding affine scene augmentation to the view augmented image on the ground plane projection.

3. Approach

We tackle the multi-view multi-person detection problem. In this section we introduce the problem formalism. Then we present our multi-view data augmentation framework. Finally, we show how it is combined with our multi-view network.

3.1. Multiview detection formalism

Let us consider a scene containing V different cameras with partially overlapping fields of view. Each camera is calibrated [21], yielding the calibration $\mathbf{C_v} = \{\mathsf{K}_v, \mathsf{R}_v, \mathbf{t}_v\}$. Where K_v is the intrinsic camera matrix, and R_v and $\mathbf{t}_v$ are the extrinsic camera parameters.

A set of frames $\mathbf{I} = \{\mathbf{I}_1, \dots \mathbf{I}_V\}$ coming from the different cameras can be projected to a common ground plane using a top view reprojection. The top view projection matrix T_v for view v can be derived from the calibration as follows $\mathsf{T}_v = \mathsf{K}_v[\mathsf{R}_v|\mathbf{t}_v]$ assuming that the ground plane has a zero z-coordinate ($z = 0$) in the world coordinate system. The projection of an image onto the ground plane is then written as $\mathbf{I}_v^{ground} = P(\mathbf{I}_v, \mathsf{T}_v)$ where P is the projection function.

3.2. Geometric data augmentation

Applying data augmentation in a multi-view context is not trivial, when applying geometric transformation to an image it invalidates its calibration and with it the projection on the common ground plane. We propose to solve this issue by extending the augmentation process to include the ground plane projection matrix T_v.

We focus our attention on the following geometric data augmentation: flipping, cropping, affine transformation and perspective transformation. An example of each transform is visible in Fig. 2. It is possible to express all these geometric data augmentation in the form of a homography H. The Appendix Section 1 contains the detailed homographies for each augmentation.

View augmentation Our approach consists of two types of augmentation, first the one we call view augmentation, which is applied to the input image. This is similar to standard data augmentation. However we also update the ground plane projection in order to preserve the alignment among views. Note that each view is augmented independently and can be transformed by different augmentations.

Given a homography H_v characterizing a view augmentation and the image $\mathbf{I}_v$ for view v we write the augmented image as $\mathbf{I'}_v = P(\mathbf{I}_v, \mathsf{H}_v)$. The augmented ground plane projection is written as $\mathsf{T}'_v = \mathsf{H}_v^{-1}\mathsf{T}_v$. In a single step the augmented projection, inverse the effect of the image data augmentation before doing the original ground plane projection.

Scene augmentation The second type of augmentation is novel and specific to multi-view training, we call it scene augmentation. Scene augmentation only changes the ground plane projection matrix, it modifies the projection of all the views in a similar manner. Intuitively it can be seen as applying augmentation on the ground plane directly. In practice it consists only in a modification to the ground plane projection, and doesn't require any additional projection step. We visualize the effect of scene augmentation in Fig. 4.

Given a scene augmentation characterized by the homography H_S the ground plane projection is augmented as follows $\mathsf{T}'_v = \mathsf{T}_v\mathsf{H}_S$. Note that scene augmentation is independent of the viewpoint, all views are augmented with the same H_S.

Both types of augmentation modify the ground plane projection matrix but they do it independently, it is possible to apply each type of augmentation on their own, or to combine the two. When both are applied, the augmented ground plane projection can be written as $\mathsf{T}'_v = \mathsf{H}_v^{-1}\mathsf{T}_v\mathsf{H}_S$. Fig. 3 contains a visualization of both types of augmentation.

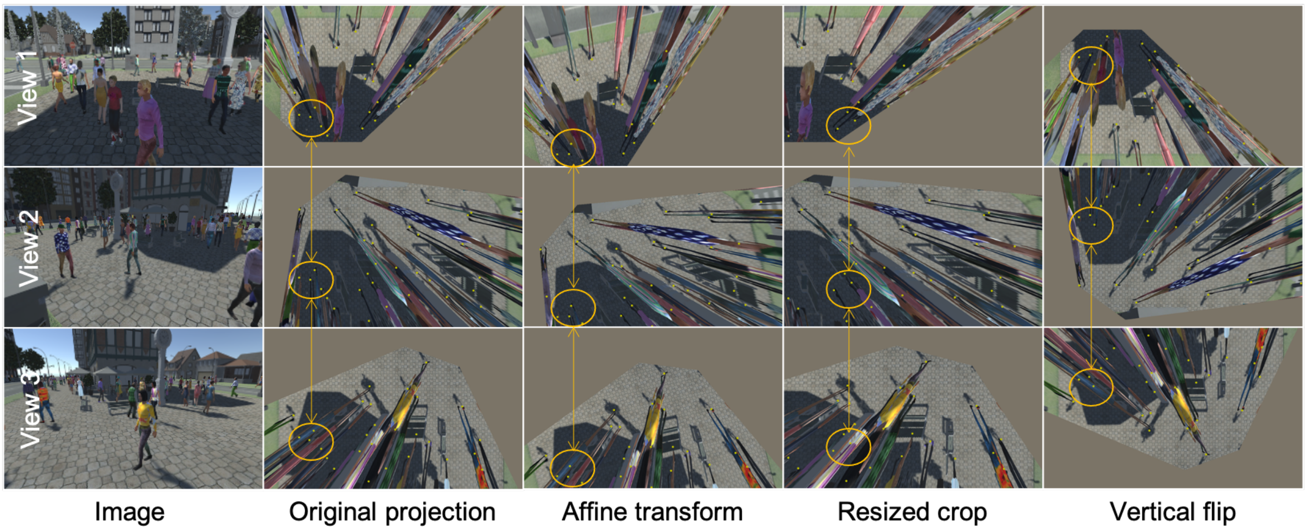

Figure 4: **Visualization of scene augmentation** We visualize the effect of different scene augmentation by projecting the original image onto the ground plane using ground plane homography augmented with scene augmentation. Note that the ground plane augmentation is the same for all the views which guarantees to preserve the alignment between views. The orange circle highlight the same ground truth points across the two views.

3.3. Model Architecture and Training

In this section we describe the architecture and training procedure of our multi-view detection model. The overall architecture can be seen in Fig. 5

Multi-view detection The proposed multi-view model consists of three learnable modules, first the feature extractor based on a truncated ResNet 34 [7] process each image independently. Each feature is then projected on the ground plane using its associated projection matrix. After projection the features are concatenated and goes through the scene detector which outputs the final scene detection map. More formally, it reads as follows:

$$\mathbf{I}' \xmapsto{f_{\boldsymbol{\theta}_0}} \mathbf{F} \xmapsto{P(\mathbf{F},\mathbf{T}')} \mathbf{G} \xmapsto{d^S_{\boldsymbol{\theta}_1}} X \tag{1}$$

Where $\mathbf{I}'$ is the set of augmented input images and $\mathbf{T}'$ is its corresponding set of augmented top view projection matrices. $\mathbf{F} = \{\mathbf{F}_1, \dots \mathbf{F}_V\}$ is the output of the ResNet parameterized by weights θ_0, with view features $\mathbf{F}_v = f_{\boldsymbol{\theta}_0}(\mathbf{I}'_v)$. $\mathbf{G}$ corresponds to the projected features in the ground plane $\mathbf{G} = \{\mathbf{G}_1, \dots \mathbf{G}_V\}$ with $\mathbf{G}_v = P(\mathbf{F}_v, \mathsf{T}'_v)$. Finally, the ground features are concatenated and goes through the scene detector parameterized by weights θ_1 which outputs the final detection heatmap on the ground plane. We denote $X = F(\mathbf{I}', \mathbf{T}', \theta_0, \theta_1)$ for short this scene detection pipeline.

In parallel, the image features $\mathbf{F}$ of all the views go into the view detector which processes them independently and outputs a view detection map for each of them. It reads as follows:

$$\mathbf{F}_v \xmapsto{d^V_{\boldsymbol{\theta}_2}} R_v \tag{2}$$

Where $\mathbf{F}_v$ is the ResNet output defined above. R_v corresponds to the detection heatmap in the image plane for view v. It is the output of the view detector parameterized by weights θ_2. We denote $R_v = F(\mathbf{I}', \theta_0, \theta_2)$ for short this view detection pipeline.

Loss function The aim of the training is to learn the weights $\theta_{0:2}$ of the model. Given the model inputs $\mathbf{I}'$ and $\mathbf{T}'$ and their corresponding scene detection ground truth $\hat{X}$ and view detection ground truth $\hat{R}_v$ our model is trained with two loss functions.

For the scene detection loss, we use the Mean Squared Error (MSE) defined as follows:

$$\mathcal{L}_{\text{ground}}(X, \hat{X}) = \left(X - \hat{X}\right)^2 \tag{3}$$

Similarly to [9] we also supervised the training directly in the image plane with the following loss.

$$\mathcal{L}_{\text{image}}(R, \hat{R}) = \frac{1}{V}\sum_{v=1}^{V}\left(R_v - \hat{R}_v\right)^2 \tag{4}$$

As opposed to [9] we only apply this loss for detection at feet level instead of both feet and head. We found empirically no benefit for adding additional supervision at head

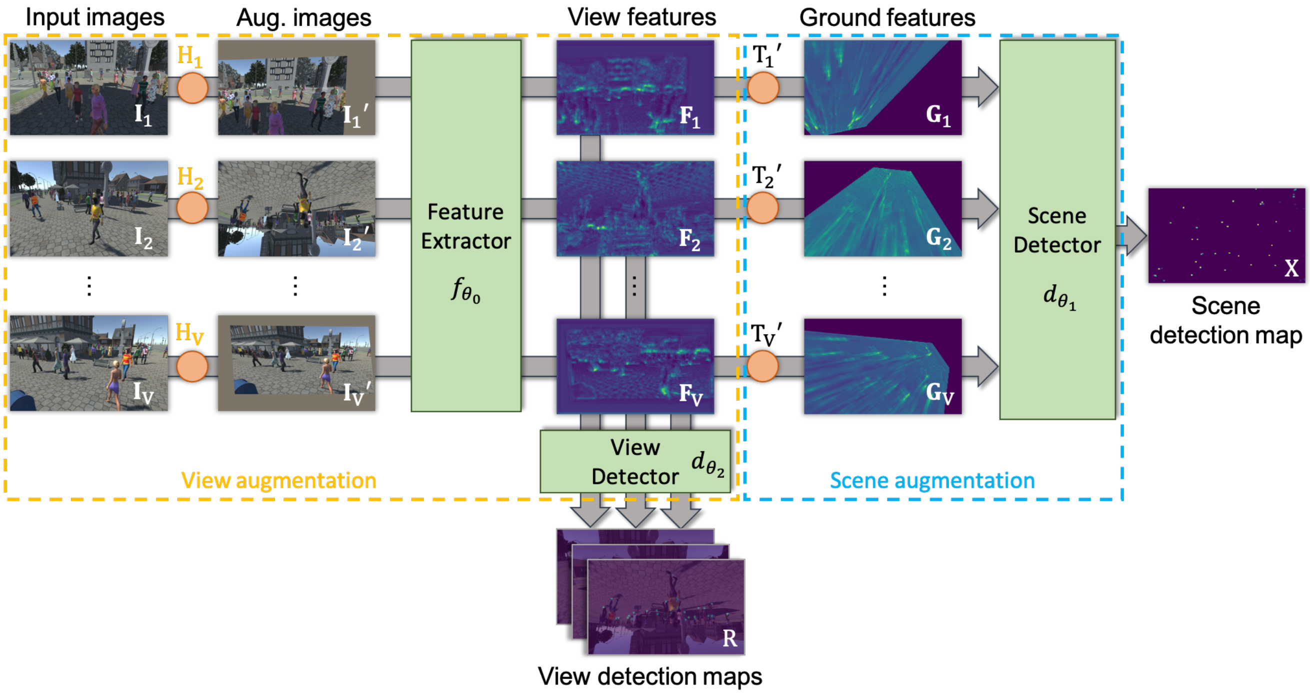

Figure 5: **Details of the proposed multi-view multi-person detection architecture** A set of input images $\mathbf{I}$ is augmented with the proposed view augmentation. Each view augmentation is reflected on the set of ground plane projection homographies $\mathbf{T}$ to form the augmented homographies $\mathbf{T}'$ preserving the alignment on the ground plane. The augmented images go through a feature extractor module, then the features are being projected onto the ground plane where they are aggregated by the scene detector which outputs the final scene detection heatmap. In parallel, features from individual images are fed to the view detector to generate view detection used for regularization purposes. Additionally ground plane projection homographies $\mathbf{T}$ can be extended with our second type of augmentation, scene augmentation, which applies augmentation directly in the ground plane. Boxes with green background correspond to learnable modules, arrows going through a module represent elements being processed independently by that module. Orange discs represent projection operation, the letter above each disc corresponds to the homography used by the projection.

level. $\mathcal{L}_{\text{image}}$ serves two purposes, first it acts as a regularizer pushing the feature extractor to generate relevant features independently for each view. Secondly, when combined with view augmentation, it helps reduce overfiting in the feature extractor part of the model.

Both losses are summed to form the training loss

$$\mathcal{L} = \mathcal{L}_{\text{ground}} + \mathcal{L}_{\text{image}}.$$

4. Experiments

We validate our approach on the multi-view multi-person detection task using the WILDTRACK and MultiviewX datasets.

4.1. Experimental setup

Datasets To train our model we use two multi-view pedestrian datasets: The WILDTRACK dataset has 7 cameras that focus on an area of 12m × 36m in the real world. It contains 400 synchronized frames per view with a resolution of 1080×1920. Each person is annotated with a bounding box. Fig. 3 shows an image from the WILDTRACK dataset.

The MultiviewX dataset has 6 cameras that focus on an area of 16m × 25m. It is a synthetic dataset representing a virtual world. It also contains 400 synchronized frames per view with a resolution of 1080×1920. For both dataset the images are resized to 536×960 before being augmented and fed to the model. Three images coming from different views can be seen in Fig. 4.

The aggregation of multiple views is done in the ground plane. In WILDTRACK we discretize the ground plane such that one cell correspond to 20 cm resulting in ground plane map of dimensionality 180×80. For MultiviewX the ground plane map has a dimensionality of 160×250 with cell corresponding of 10 cm. The scales of the ground plane map have been chosen to minimize computational cost.

	WILDTRACK dataset				MultivievX dataset			
model	MODA	MODP	Prec.	Rec.	MODA	MODP	Prec.	Rec.
DeepOcclusion [2]	74.1	-	95.0	80.0	-	-	-	
MVDet [9]	88.2	75.7	94.7	93.6	83.9	79.6	96.8	86.7
SHOT [18]	90.2	76.5	96.1	94.0	88.3	82.0	96.6	91.5
MVDeTr [8]	91.5	**82.1**	**97.4**	94.0	93.7	**91.3**	**99.5**	94.2
MVAug (Ours)	**93.2**	79.8	96.3	**97.0**	**95.3**	89.7	99.4	**95.9**

Table 1: **Multi-view multi-person detection** Detection performance of our proposed model on the WILDTRACK and MultiviewX datasets. We report MODA, MODP, precision and recall [11]. The proposed approach outperforms all existing baseline in terms of MODA on both datasets. In general this performance gain can be explained by a increase in recall.

Evaluation Metrics We adopt similar evaluation metrics as previous work [2, 9, 8], we report Precision, Recall, MODA, and MODP [11]. A threshold equivalent to 0.5 meters is used to determine true positives. We use the matlab MOTChallenge Evaluation toolkit.

Implementation details Our model is implemented in Pytorch, and runs on a single Nvidia v100 GPU. The data augmentation pipeline wraps the original Torchvision augmentation in order to extract their parameters and generate the corresponding homography. During training random affine transformations are used for both view and scene augmentation and in both cases, a proportion of 50% of the training data is augmented.

The feature extractor is based on a ResNet 34 with its last four layers removed. It outputs feature of dimensionality 128.

The view detector consists of two pairs of ReLu and a 1×1 convolutional layer followed by a sigmoid function. First convolution layer contains 128 filters and the second one a single filter. The output of the view detector is only used for regularization purposes, hence the minimal architecture of the view detector allows for greater regularization effect on the feature extractor.

For the scene detector, we adopt a multi-scale architecture, this detector is responsible for aggregating the ground plane features coming from multiple views. Therefore it needs to be able to handle slight misalignment among them due to calibration error. The scene detector consists of four scales where the spatial resolution of the features is halved in among each scale using adaptive average pooling. Each scale consists of four blocks of convolutional layer - batch normalization [10] - ReLu [13]. The output of the four scales are bilinearly interpolated back to their original dimension, concatenated and fed into a final 1×1 convolutional layer followed by a sigmoid function to produce the final scene detection heatmap.

Our model outputs probabilistic detection heatmaps, to compute evaluation metrics we extract detection points from those heatmaps. We apply Non Maximum Suppression (NMS), then select the top 200 detections and use K-Means clustering on detection score with K=2 to separate true detection from noise.

Augmentation parameters We list the parameters used for each type of geometric transforms. For random affine augmentation, the rotation can be up to 45 degrees, the translation up to 20% on both directions, the scaling up to 20% up or down, and the shearing up to 10 degrees. For the random resized crop, the crop covers an area of 80% to 100% of the original image with an aspect ration between 0.75 and 1.33 before being resized to the original image size. The perspective transformation uses a distortion scale of 0.5. Horizontal and vertical flips don't require any parameters.

4.2. Comparing to the State-of-the-Art

On the multi-view people detection task, we compare our model to 4 baselines. Results can be found in Table 1. On both WILDTRACK and MultiviewX the proposed model using our two-level augmentation scheme outperforms all previous baseline on MODA with a significant margin. In particular it outperforms MVDeTr which uses a simpler form of view based augmentation combined with a more complex transformer based architecture. The improvement in MODA can be explained by an increase in recall, in general our model detects people that were missed by other models. It confirms the better generalization of our model due to our data augmentation pipeline.

Note that our model underperforms on the MODP metric when compared to MVDeTr this can be explained by our choice of ground plane discretization strategy. MVDeTr uses much smaller cells of 2.5 cm. Even though the metric threshold has been adjusted to account for this, rounding error from the change of scale remains and mostly affect MODP which is directly computed from distances in the discretized space. As stated above the coarser grid was chosen for computational reason due to the large number of experiments needed to evaluate the proposed data augmentation pipeline.

4.3. Further Analysis

We conduct additional experiments to justify the design choice of our method, and we evaluate the contribution of each of its components. In an effort to stay as close as possible to a real-life scenario, all the following experiments are conducted on the WILDTRACK dataset.

Optimal combination of view and scene augmentation We propose to investigate which combination of view and scene augmentation is optimal. In Table 2 we report the MODA metric for multi-view people detection on WILDTRACK. When only scene augmentation is used, the affine augmentation is most beneficial. When only view augmentation is used, affine augmentation and crop augmentation perform very well. We can see that when only one type of augmentation is used, view augmentation generates greater improvement than scene augmentation. Overall when compared to no augmentation at all, most augmentation strategies are beneficial. Finally, the best pairwise combination of augmentation consists of using random affine for both view and scene augmentation.

Ablation study We conduct an ablation study to measure how each of the loss $\mathcal{L}_{\text{image}}$, the view augmentation and the scene augmentation contribute to the overall performance of the system. We report MODA and MODP on the WILDTRACK dataset in Table 3. On their own each component improves MODA, both augmentation generate greater improvement than $\mathcal{L}_{\text{image}}$. When combined with view augmentation, $\mathcal{L}_{\text{image}}$ improves MODA by almost a point, whereas it has detrimental effect when combined with scene augmentation alone. When scene augmentation is used, it systematically improves MODP. The best result is obtained when everything is combined.

		View augmentation					
		No aug	H-Flip	V-Flip	Affine	Persp.	Crop
Scene augmentation	No aug	90.86	91.28	91.49	92.65	92.23	92.45
	H-Flip	91.39	90.65	91.81	92.44	91.70	92.54
	V-Flip	90.55	91.60	90.97	91.91	91.07	92.02
	Affine	91.49	91.91	92.02	**93.17**	91.49	92.44
	Persp.	91.28	90.86	90.44	90.86	90.44	91.49
	Crop	90.76	92.54	91.91	91.49	91.81	92.44

Table 2: **Combination of view and scene augmentation** We report MODA metric on the WILDTRACK dataset for all pairwise combination of view and scene augmentations. For view augmentation crop, affine, and perspective augmentation perform best. For scene augmentation affine and horizontal flip augmentation are best. Best results is obtained by combining affine view augmentation with affine scene augmentation.

Model component			Metrics	
$\mathcal{L}_{\text{image}}$	View aug.	Scene aug.	MODA	MODP
			90.65	76.92
✓			90.86	79.59
	✓		91.28	77.52
		✓	91.28	79.20
✓	✓		92.12	77.67
✓		✓	90.65	78.41
✓	✓	✓	**93.17**	**79.83**

Table 3: **Ablation results on WILDTRACK** We report MODA and MODP metric on the WILDTRACK dataset, we evaluate the contribution of the proposed view augmentation, scene augmentation and the image prediction loss. Without $\mathcal{L}_{\text{image}}$ loss, both view and scene augmentation perform similarly. When $\mathcal{L}_{\text{image}}$ is added view augmentation perform significantly better. The best result is obtained with the combination of the three components.

Augmentation proportion We propose to evaluate how the proportion of augmentation impacts detection results. To do so we vary the percentage of training data that is subjected to either view or scene augmentation. We report MODA and MODP on the WILDTRACK dataset in Table 4. When percentage of augmentation is kept identical for both view and scene augmentation, the best result is obtained when 50% of the training data is augmented. We also test the effect of having different percentages for view and scene augmentation, better detection results are obtained when the proportion of view augmentation is larger than the proportion of scene augmentation (See line two and three of Table 4).

		Metrics	
View aug.	Scene aug.	MODA	MODP
0%	0%	90.86	79.59
25%	25%	91.49	78.24
25%	50%	90.86	79.41
50%	25%	92.12	79.66
50%	50%	**93.17**	**79.83**
75%	75%	90.65	78.47
100%	100%	90.44	78.24

Table 4: **Varying proportion of augmentation** We report MODA metric on the WILDTRACK dataset. We evaluate the effect of varying the proportions of input images affected either by view or scene augmentation. Best result is obtained with 50% of the training image augmented with both view and scene augmentation. When using unbalanced proportion, it is beneficial to have of view augmentation rate higher than the scene augmentation rate.

Effect on overfitting The main goal of data augmentation is to increase model generalization by reducing overfitting to the training dataset. We propose to measure how each component of our approach helps reduce overfitting. On the WILDTRACK dataset, we measure overfitting by computing the ratio of validation loss over training loss. Ideally, the overfitting ratio should be one, meaning that the model performs similarly on the training and validation dataset.

We can see in Fig. 6 that for our baseline model which uses neither $\mathcal{L}_{\text{image}}$ nor any kind of augmentation, it is not the case and the overfitting ratio quickly grow over 5. Adding $\mathcal{L}_{\text{image}}$ to the baseline help reduce the ratio, additionally using scene augmentation or view augmentation further reduces overfitting. Note that view augmentation has more impact on overfitting than scene augmentation. Finally, when the three components are used together, the overfitting ratio is reduced the most and very close to an ideal ratio of one.

5. Limitations

In our experiment we limited ourselves to using only one type of geometric augmentation for both scene and view augmentation. It might be possible to further improve performances by combining multiple type of augmentation. However with more than one type of augmentation the number of possible combinations becomes quite large and therefore computationally expensive to evaluate systematically.

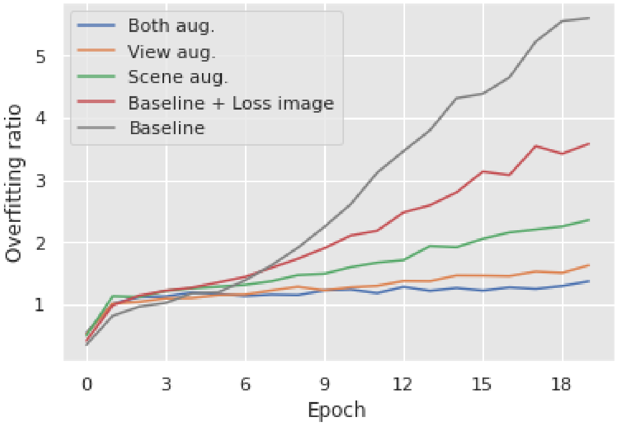

Figure 6: **Effect of data augmentation on training overfitting** We visualize the evolution of the overfitting ratio over training epochs. It is computed by dividing the validation loss by the training loss. Each component of our method contributes in reducing overfitting, the best result is obtained when both augmentations are combined with the image loss.

Similarly due to limited computational resources and large number of experiments we were only able to run each training once, ideally we would like to average the results over multiple runs. Nonetheless, with the current results we were able to observe general trends when it comes to data augmentation in a multi-view detection system.

6. Conclusion

In this paper, we proposed a new two-level augmentation pipeline for multi-view multi-person detection. When combined with our simple multi-view end-to-end trainable model, it outperforms all existing baselines.

Through extensive ablation studies, we show the contribution of each component of our model and their interaction with each other. We systematically evaluate all pairwise combination of scene and view augmentation. Furthermore we confirm that the proposed approach is effective on real data, by obtaining state-of-the-art results on both the WILDTRACK and MultiviewX datasets.

Acknowledgments This work was funded in part by the Swiss Innovation Agency.

References

[1] P. Baqué, F. Fleuret, and P. Fua. Deep Occlusion Reasoning for Multi-Camera Multi-Target Detection. In *International Conference on Computer Vision*, 2017.

[2] T. Chavdarova, P. Baqué, S. Bouquet, A. Maksai, C. Jose, L. Lettry, P. Fua, L. Van Gool, and F. Fleuret. The Wildtrack Multi-Camera Person Dataset. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[3] T. Chavdarova and F. Fleuret. Deep Multi-Camera People Detection. pages 848–853, 2017.

[4] Terrance DeVries and Graham W. Taylor. Dataset augmentation in feature space. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017.

[5] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multi-Camera People Tracking with a Probabilistic Occupancy Map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):267–282, February 2008.

[6] R.B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *arXiv Preprint*, 2013.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[8] Y. Hou and L. 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.

[9] Y. Hou, L. Zheng, and S. Gould. Multiview Detection with Feature Perspective Transformation. In *European Conference on Computer Vision*, pages 1–18, 2020.

[10] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning*, 2015.

[11] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, 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(2):319–336, 2009.

[12] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. *arXiv Preprint*, 2017.

[13] V. Nair and G. E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *International Conference on Machine Learning*, 2010.

[14] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems*, 2015.

[15] G. Roig, X. Boix, H. Ben Shitrit, and P. Fua. Conditional Random Fields for Multi-Camera Object Detection. In *International Conference on Computer Vision*, 2011.

[16] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 2019.

[17] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *J. Big Data*, 6:60, 2019.

[18] Liangchen Song, Jialian Wu, Ming Yang, Qian Zhang, Yuan Li, and Junsong Yuan. Stacked homography transformations for multi-view pedestrian detection. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[20] Z. Tian, C. Shen, H. Chen, and T. He. FCOS: Fully Convolutional One-Stage Object Detection. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[21] R.Y. Tsai. A Versatile Cameras Calibration Technique for High Accuracy 3D Machine Vision Metrology Using Off-The-Shelf TV Cameras and Lenses. *Journal of Robotics and Automation*, 3(4):323–344, 1987.

[22] Kaixuan Wang and Shaojie Shen. Mvdepthnet: Real-time multiview depth estimation neural network. In *2018 International Conference on 3D Vision, 3DV 2018, Verona, Italy, September 5-8, 2018*, pages 248–257. IEEE Computer Society, 2018.

[23] Y. Xu, X. Liu, Y. Liu, and S.C. Zhu. Multi-View People Tracking via Hierarchical Trajectory Composition. In *Conference on Computer Vision and Pattern Recognition*, pages 4256–4265, 2016.

[24] X. Zhou, D. Wang, and P. Krähenbühl. Objects as Points. In *arXiv Preprint*, 2019.

[25] Barret Zoph, Ekin D. Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V. Le. Learning data augmentation strategies for object detection. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVII*, volume 12372 of *Lecture Notes in Computer Science*, pages 566–583. Springer, 2020.