Towards Generalization in Multi-View Pedestrian Detection

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science in ***Computer Science and Engineering*** *by Research*

by

Jeet Vora
2019701006
jeet.vora@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
April 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled **"Towards Generalization in Multi-View Pedestrian Detection"** by Jeet Vora, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: Prof. Vineet Gandhi

To my family, my guide and my friends

Acknowledgments

First, I would like to thank my research advisor, Prof. Vineet Gandhi, for the freedom and flexibility he gave and the experience I had while working on this thesis. None of the work in this thesis would have been possible without the continuous encouragement and enthusiasm he shared for the work. I also thank him for providing me the opportunity of working under him, grooming me on exploring, pursuing, and formulating concrete ideas, and conducting rigorous analyses, which helped me get a lot of insights into how to proceed with research. I also thank him for being my research advisor and his role in shaping and giving a new direction to my career.

I would like to thank my colleagues Swetanjal Dutta, Kanishk Jain for their valuable contribution; without them, this work would not have been possible. I want to thank Shyam Gopal Karthik for his constant support and guidance, which helped me finish the work in time, and for valuable discussions which gave me a fresh perspective always. His understanding and intuition of research inspire me, and I hope to continue working with him in the future. I am privileged to have worked with all of them. I also thank Varun Chhangani for helping whenever I have been stuck with any systems-related issue

I must also thank to my colleagues at CVIT lab for creating such a motivated and inspiring ecosystem. I also thank the institute and CVIT lab workspace for providing unparalleled access to resources without it this work was not possible.

I want to thank Dr C.V. Jawahar, Dr Anoop Namboodiri, Dr Madhava Krishna, Dr Avinash Sharma, Dr Ravi Kiran Sarvadevabhatla, Dr Pawan Kumar whose coursework at IIIT-H were a great learning experience and without it the work in this thesis would not be possible. I would also like to thank my batchmates and friends I made in IIIT-H for making my MS journey an enjoyable and productive experience. I should also thank K L Bhanu and Sarath Sivaprasad for sharing their experiences and having valuable discussions, which helped in my work.

Finally, I thank my family because they have always helped and provided constant support, motivation, and encouragement throughout my life and my career.

Abstract

Detecting humans in images and videos has emerged as an essential aspect of intelligent video systems that solve pedestrian detection, tracking, crowd counting, etc. It has many real-life applications varying from visual surveillance and sports to autonomous driving. Despite achieving high performance, the single camera-based detection methods are susceptible to occlusions caused by humans, which drastically degrades the performance where crowd density is very high. Therefore multi-camera setup becomes necessary, which incorporates multiple camera views for detections by computing precise 3D locations that can be visualized and transformed to Top View also termed as Bird's Eye View (BEV) representation and thus permits better occlusion reasoning in crowded scenes.

The thesis, therefore, presents a multi-camera approach that globally aggregates the multi-view cues for detection and alleviates the impact of occlusions in a crowded environment. But it was still primarily unknown how satisfactorily the multi-view detectors generalize to unseen data. In different camera setups, this becomes critical because a practical multi-view detector should be usable in scenarios such as i) when the model trained with few camera views is deployed, and one of the cameras fails during testing/inference or when we add more camera views to the existing setup, ii) when we change the camera positions in the same environment and finally iii) when deploying the system on the unseen environment; an ideal multi-camera setup system should be adaptable to such changing conditions.

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. We formalized three critical forms of generalization and outlined the experiments to evaluate them: generalization with i) a varying number of cameras, ii) varying camera positions, and finally, iii) to new scenes. We discover 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 generated a novel Generalized MVD (GMVD) dataset, assimilating diverse scenes with changing daytime, camera configurations, varying number of cameras, and (b) we discuss the properties essential to bring generalization to MVD and developed a barebones model to incorporate them. We performed a series of experiments on the 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 developed approach.

Contents

Chapter Page

List of Figures

Figure Page

List of Tables

Table Page

Chapter 1

Introduction

1.1 Problem Statement

1.1.1 Context

In computer vision, humans as target objects are attracting more attention. Researchers have developed algorithms to help detect, track, and identify humans, which benefits various applications, such as sports analytics, robotics, machine interaction, surveillance, self-driving cars, etc. In-person tracking is achieved by identifying the same person from every frame in video recordings; motion captured by person tracking algorithms would aid in the detection and identification of the individual. Studies of learning human motion can date back to the period before digital video systems and the internet were globally commercialised. Frameworks were designed to teach computers to see a walking person or to recognise the gestures of a person. Pedestrians were usually simply modelled using connected cylinders and sticks representing the topology of body parts. The symmetry of the human body was employed to separate humans from background. In recent years, imaging technology has advanced dramatically. Cameras are now more affordable, smaller, and of higher quality than they have ever been. Simultaneously, computational power has skyrocketed. Computing platforms such as multicore processing and graphical processing units have been geared toward parallelization (GPU). This hardware version enables the real-time implementation of Computer Vision algorithms for pedestrian detection and tracking. Rapid advancements in deep learning, convolution neural network (CNN), and GPU computing power are the primary reasons for CV-based pedestrian detection and tracking evolution.

Classification, Localization, Detection, and Segmentation are the four main types of tasks in computer vision as shown in Figure 1.1 from [36].

- **Classification :** determines which object categories (such as humans, dogs, or cars) are represented in an image or video.

- **Localization :** object localization i.e it gives us an information of object but also tells us with a bounding box which is a position of the object within the image.

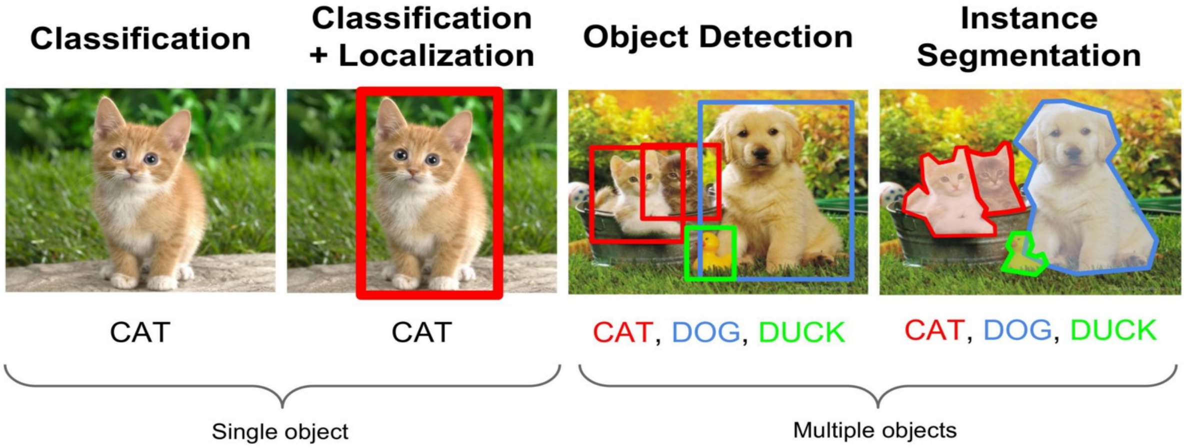

Figure 1.1: Image classification, object detection, and instance segmentation comparison figure.

- **Detection :** it detects semantic objects of a specific category in a image or captured video sequence.
- **Segmentation :** it solves the problem of ”which object or scene each pixel belongs to” also categorized as semantic segmentation and instance segmentation.

Detecting pedestrians is the primary goal of pedestrian detection in an image and video sequence as well as localizing their positions and sizes. The problem of pedestrian detection is approached in multiple ways. The classical computer vision based methods use hand crafted features such as HOG [32] features or randomly generated low level features such as Haar [48] like features. These attributed features are being used to train the model that performs the classification. Other approaches include using several simple classifiers to make a strong classifier, for example Ada boost. Recently the interest for CNN has increased since this method has managed to achieve high performance in several different fields of computer vision. The most notable results are for general classifications tasks such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [43]. Deep CNNs serve as the foundation for Deep Learning based pedestrian detection to extract featues from image or video frames and then classify as pedestrians. These approaches are classified into two types.

- **Two Stage Detectors :** In two-stage detectors such as R-CNN [18], Fast R-CNN [17], Faster R-CNN [40], initially, deep features are used to propose approximate object regions, which are then used for classification and bounding box regression for the candidate. This results in two stage detectors with high detection accuracy.
- **One Stage Detectors :** On the other hand one-stage detectors such as SSD [33], YOLO [39]. Without the region proposal step, bounding boxes are predicted over the images. This process takes less time and can thus be used in real-time devices which achieves high speed.

Person detection is also divided into two categories based on the amount of cameras used for detection, which are as follows:

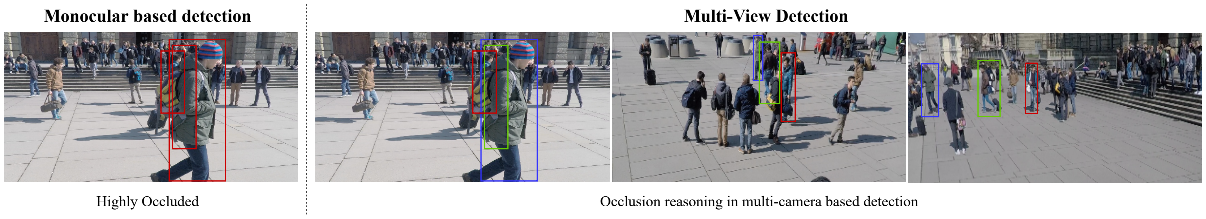
Monocular based detection
Multi-View Detection
Highly Occluded
Occlusion reasoning in multi-camera based detection

Figure 1.2: Severe occlusion in monocular detection (left). Multi-camera detection resolves the difficulty arising from high occlusion (right).

- **Monocular based detection :** To perform detections, monocular methods depend on the input feed of a single camera. These methods offer a simple and straightforward setup but provide no 3D information.

- **Multi-camera detection :** In multi-camera setups, we have multiple cameras placed at certain positions and orientation in the environment. The cameras are placed such that their FOV's are either non-overlapping or overlapping. The major benefit is we get 3D information of environment.

1.1.2 Challenges in monocular camera detection and Motivation for multi-camera detection

Since monocular approaches are based on the input of a single camera for detections. Over recent years these methods has achieved state-of-the-art results. This class of algorithms typically proposes potential bounding boxes candidate with scores. They then use Non-Maximum Suppression (NMS) to generate a final set of candidates. A feature vector of random dimension can then be computed for any variable size 2D bounding box in that image using Region Of Interest (ROI) pooling and then it is fed to a classifier to determine whether the bounding box resembles a true detection. While this algorithm has proven its worth on numerous benchmarks, it may fail in cluttered scenes shown in Figure 1.2. This shows the problem of monocular detectors when people obstruct each other severely.

The multi-camera based methods use images from multiple calibrated cameras observing the same area from different viewpoints with an overlapping field of view to take full advantage of appearance or geometrical uniformity throughout views to resolve ambiguities in cluttered scenes and acquire accurate 3D localisation as shown in the Figure 1.2 . Basically, it globally aggregates the multi-view cues for detections and thus motivates the need for multi-camera setup with overlapping FOV's to resolve the difficulties arising from high occlusion and crowdedness.

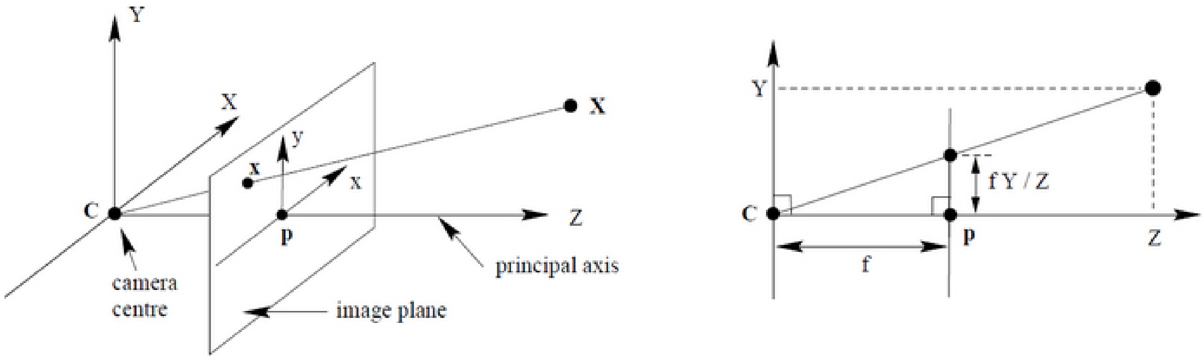

Figure 1.3: Camera geometry and pinhole camera model

1.2 Background

In this thesis since we are looking into Multi-Camera Detection, in subsequent sections will discuss about few terminologies in multi-view detection literature.

1.2.1 Camera Geometry and the Pinhole Model

The pinhole camera model [20] describes the projection of points in 3D space to an image plane as the mathematical relationship. Let the origin of a Euclidean coordinate system be the centre of projection, the plane $Z = f$, which is known as the focal plane or image plane. A point in space with coordinates $(X, Y, Z)^T$ under the pinhole camera model is mapped to the points on image plane $(\frac{fX}{Z}, \frac{fY}{Z}, f)^T$ as shown in Figure 1.3. Neglecting the final image coordinate, camera geometry is the central projection mapping from 3D world space to 2D image coordinate. The projection centre is also known as the optical centre or the camera centre. The line perpendicular to the image plane from the *camera centre* is known as the *principal ray* or *principal axis*. The *principal point* is the intersection of the principal axis and the image plane. The *principal plane of the camera* is the plane that runs through the centre of the camera and parallel to the image plane. Camera centre is denoted by **C** and principal point by **p**. The camera is centred at the coordinate origin.

$$(X, Y, Z)^T \longrightarrow (\frac{fX}{Z}, \frac{fY}{Z})^T \tag{1.1}$$

Assuming homogeneous coordinates representation of the world as well as image points, central projection can be simply expressed in terms of matrix multiplication as a linear mapping between their

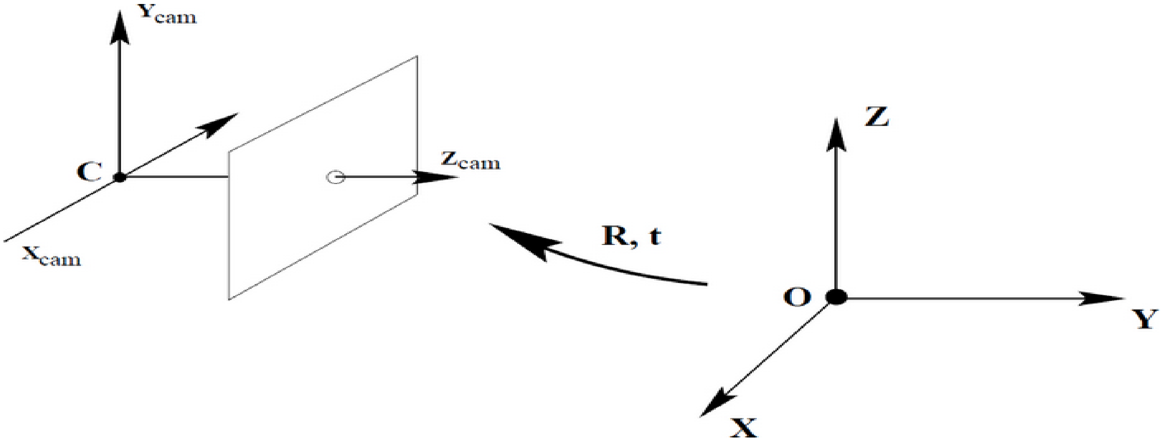

Figure 1.4: The world coordinate and camera coordinate frames are transformed using Euclidean geometry.

homogeneous coordinates,

$$\begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix} \tag{1.2}$$

Principal Point : In the image plane origin of coordinates is assumed to be at the principal point in theory. In practise, this may not be the case, therefore, the Eq. 1.2 is expressed as,

$$\begin{bmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix} \tag{1.3}$$

First matrix in right side of the Eq. 1.3 is called ***camera calibration matrix*** expressed by K.

Camera Rotation and Translation : Generally, the world coordinate frame determines points in space. *Rotation* and *Translation* connect the camera coordinate and world coordinate frames. As shown in Figure 1.4, if the coordinate of the point in the world coordinates is $X_{world} = (X, Y, Z, 1)^T$, then X_{cam} is transformed by,

$$X_{cam} = \begin{bmatrix} R & t \end{bmatrix} X_{world} \tag{1.4}$$

where R is rotation matrix of 3 X 3 and t is translation vector of 3 X 1. Accumulating together we get,

$$x = K \begin{bmatrix} R & t \end{bmatrix} X_{world} \tag{1.5}$$

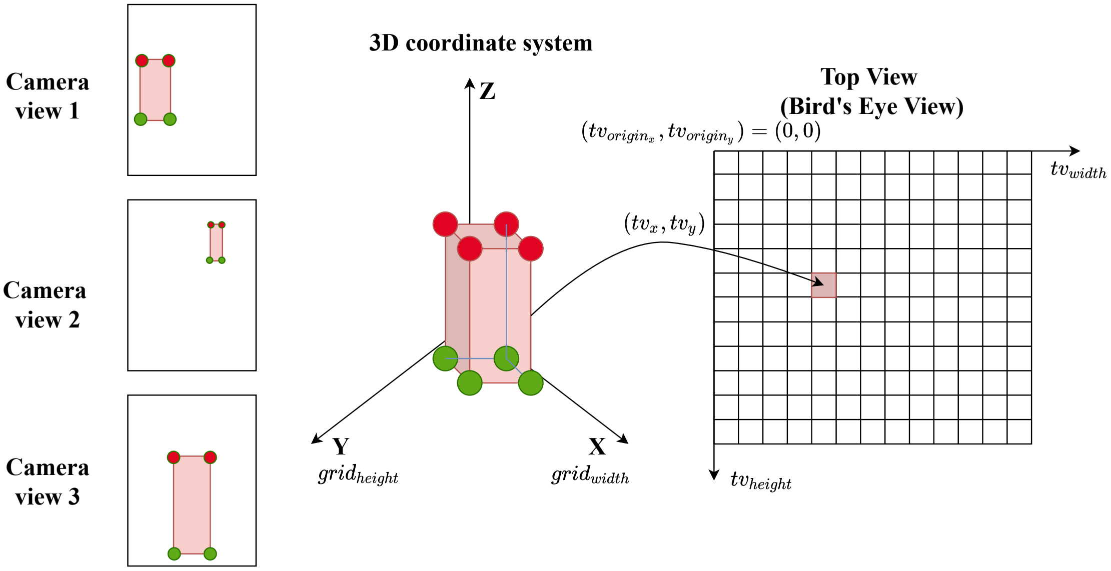

Figure 1.5: The discretized ground plane i.e $(Z = 0)$ and representation of presence of human in terms of cuboid in 3D coordinate system, its corresponding human silhouette in each camera views (i.e camera view 1, 2 and 3) and the representation of cuboid from 3D coordinates space to Top View coordinates (Bird's Eye View representation). Red and Green points in 3D coordinate and its corresponding 2D points in each camera view are used for synchronized calibration of multiple cameras.

x which is the pinhole camera's mapping in the world coordinate frame. The pinhole camera matrix, P, is denoted by,

$$x = K\begin{bmatrix} R & t \end{bmatrix} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r1 & r2 & r3 & t1 \\ r4 & r5 & r6 & t2 \\ r7 & r8 & r9 & t3 \end{bmatrix} \tag{1.6}$$

It consists of nine degrees of freedom, three from $K(f, p_x, p_y)$; three from the rotation matrix R and three from the translation vector t. *Internal camera parameters* K shows camera's internal orientation, which is fixed. *External parameters* R and t represent the camera's orientation and position in relation to a world coordinate system.

1.2.2 Camera Calibration and Top View Representation

To detect a person's presence in the environment, the ground plane are discretized at $Z = 0$ in 3D coordinate system and based on the assumption of average human height (i.e 1.8m) and average human width (0.25m), the presence of a person in 3D is represented as cuboid as shown in Figure 1.5. When this cuboid is back-projected to respective camera views, the person is localized in the camera view and is represented using bounding box coordinates.

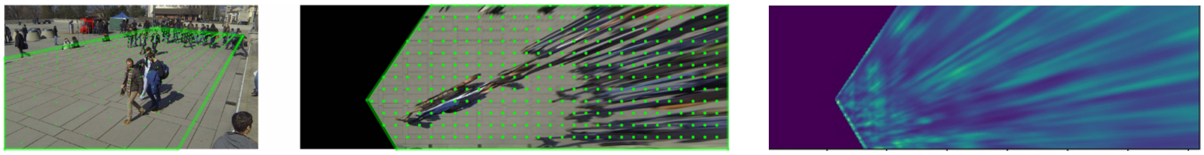

Figure 1.6: Illustrating Perspective Transformation by assuming every pixels on the ground plane i.e $(Z = 0)$. Image(left) is projected to the ground plane(center). Similarly, feature maps can be projected to the ground plane(right).

We require 2D-3D correspondences to calibrate camera i.e to estimate intrinsic and extrinsic parameters of camera. The cameras in WildTrack , MultiViewX and GMVD dataset are calibrated and synchronized based on the above assumption of grid structure and obtained annotated 2D and 3D correspondences. To calibrate the cameras we used the Pinhole camera model as mentioned in the section 1.2.1, due to its widespread usage and support in multiple libraries, including OpenCV.

The Top View or BEV representation of ground plane means looking to 3D scene from the top perspective. In Figure 1.5 the discretized ground plane location observed from top view perspective is shown. This rectangular area (Top View) can also be used for tracking. Let's consider 3D coordinate $(X, Y, Z = 0)$ and Top View coordinate, where area is of dimensions $(tv_{width}, tv_{height})$ and origin of top view grid is $(tv_{origin_x}, tv_{origin_y}) = (0, 0)$. Based on the defined grid in 3D system $(grid_{width}, grid_{height})$ we can obtain the top view coordinates (tv_x, tv_y).

Figure 1.6 shows the perspective transformation of an image to the discretized ground plane (top view). In the same way features are being transformed to the top view, this type of projection suffers less from spatial structure break as compared to projection of an image, because 2D spatial information in feature maps has already been concentrated into individual pixels .

1.2.3 Probabilistic Occupancy Maps (POM)

The Probabilistic Occupancy Map is a method for estimating the marginal probabilities of individual presence at each location in a given area of interest. In another words, given the evidence provided by the background subtraction, it estimates the probability that someone is standing at each location. Figure 1.7 and Table 1.1 refers to some common notations and representations used in this subsection. The ground plane of an environment is discretized and every location represents the presence of individuals.

They are written as,

$$P(X_i^o = 1 | B_i), \qquad \text{for every } o \text{ and } i \tag{1.7}$$

Given such a $P(B_i|X_i)$ model, which is the result of background subtraction provided the true occupancy of the scene, estimating $P(X_i|B_i)$ becomes a Bayesian Computation. This cannot be done with the generic method because of the complexity of the non-trivial $P(B_i|X_i)$ model and due to the

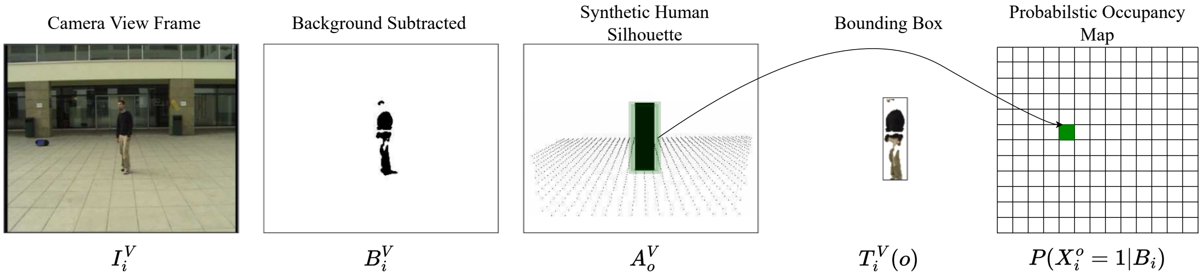

Figure 1.7: POM

dimensionality of B_i and X_i. This problem is being addressed by representing humans as rectangles, which are then used to generate ideal synthetic images A_i^V as shown in Figure 1.7 and determine whether or not people are present at specified locations. We approximate the occupancy probabilities as the marginals of a product law Q minimising the Kullback-Leibler divergence from the ”true” conditional posterior distribution.

More specifically, in 1.2.3.1 two assumptions of independence are mentioned from which analytical results are derived. In 1.2.3.2 we discuss about the generative $P(B|X)$ model, which entails calculating the distance between the actual images B and the synthetic image which is a function of X. Based on the model and the assumptions, in subsection 1.2.3.3 an analytical relationship between estimates of the marginal probabilities of occupancy $P(X_i^1 = 1|B_i), ..., P(X_i^L = 1|B_i)$ is been derived by minimizing the Kullback-Leibler divergence between the corresponding true posterior and the product law.

1.2.3.1 Independence Assumptions

The two independence assumptions stated below will allow us to derive the relationship between the optimal q_os analytically:

First assumption is that people in the environment do not consider the presence of other people when moving around while avoidance strategies are ignored. This can be formally written as,

$$P(X^1, ..., X^L) = \prod_o P(X_o) \tag{1.8}$$

Second assumption considers all statistical dependencies between bounding box views to be caused by the presence of individuals in the environment. This is the same as defining the bounding box views as vector functions $X = (X^1, ..., X^L)$ plus some noise which is independent. This means that once the presence of all individuals is determined, the bounding box views become independent. This is valid till we ignore other hidden variables such as morphology or garments, which may influence multiple views at the same time. This assumption are written as,

$$P(B^1, ..., B^V|X) = \prod_V P(B^V|X) \tag{1.9}$$

Table 1.1: POM Notations

WXH	image dimension.
V	total number of camera views.
L	total locations when the ground plane is discretized.
T	total number of bounding boxes for a frame.
t	bounding box index for frames.
Q	product law for approximation, the posterior distribution $P(\|B_i)$ for a fixed i,.
E_Q	Expectation of $X \sim Q$.
q_o	is marginal probability for Q, i.e $Q(X_o = 1)$.
ϵ_o	is the prior probability at location i, $P(X_o = 1)$
A_o^V	the image of 1's inside a rectangle for the silhouette representation at location o observed from camera view V, and 0's elsewhere.
I_i	images from camera views $I_i = (I_t^1, ..., I_i^V)$.
B_i	the background subtracted binary images $B_i = (B_i^1, ..., B_i^V)$.
X_i	is the boolean random variable vectors $X^1, ..., X^L$ for occupying the location o on the ground plane $X_i^o = 1$.

1.2.3.2 Generation of Image Model

Let the synthetic image A^V be obtained by putting rectangles at positions where $X_o = 1$, hence $A^V = \otimes_o X_o A_o^C$, where $\otimes$ is the "union" of two images. An image like this is a function of X and thus a random quantity. The background subtracted image B^V is modelled as if it were an ideal image with some noise. According to empirical evidence, it appears that the noise increases as the area of the A^V ideal image, pseudo-distance $\Psi(B, A)$ is introduced to account for this asymmetry. Ψ is written as,

$$\Psi(B, A) = \frac{1}{\sigma} \frac{|B \otimes (1 - A) + (1 - B) \otimes A|}{|A|} \tag{1.10}$$

The background subtraction quality is accounted by the parameter σ. Smaller the σ more B^V is picked nearer to its ideal value A

Given the true hidden state, a conditional distribution $P(B^V|X)$ of the background subtracted images is modelled as a density decreasing with pseudo-distance $\Psi(B^V, A^V)$ between the background subtracted image and an synthetic image A^V of rectangular shapes where people are present according to X. The model is defined as,

$$P(B|X) = \prod_V P(B^V|X) = \prod_V P(B^V|A^V) = \frac{1}{Z} \prod_V e^{-\Psi(B^V, A^V)} \tag{1.11}$$

1.2.3.3 The relationship between q_os

The expectation under $X \sim Q$ is denoted by E_Q. Because we want to minimise the Kullback-Leibler divergence between the "true" posterior $P(|B)$ and the approximation Q, the following form of derivative with respect to the unknown q_o is been used (see [14] for more detailed derivations).

$$\frac{\partial}{\partial q_o} KL(Q, P(.|B)) = \log \frac{q_o(1-\epsilon_o)}{(1-q_o)\epsilon_o} + E_Q(\sum_V \Psi(B^V, A^V)|X^o = 1) - E_Q(\sum_V \Psi(B^V, A^V)|X^o = 0) \tag{1.12}$$

Hence, if solved as,

$$\frac{\partial}{\partial q_o} KL(Q, P(.|B)) = 0 \tag{1.13}$$

we get the following,

$$q_o = \frac{1}{1 + exp(\lambda_o + \sum_V (E_Q(\Psi(B^V, A^V)|X^o = 1) - E_Q(\Psi(B^V, A^V)|X^o = 0)))} \tag{1.14}$$

where, $\lambda_o = \log \frac{1-\epsilon_o}{\epsilon_o}$, $E_Q(\Psi(B^V, A^V)|X_o = \xi)$ is the untractable computation. However, under $X \sim Q$, the image A^V is focused around B^V, we approximate, $\forall \xi \in 0, 1$.

$$E_Q(\Psi(B^V, A^V)|X^o = \xi) \simeq \Psi(B^V, E_Q(A^V|X^o = \xi)) \tag{1.15}$$

leading to,

$$q_o = \frac{1}{1 + exp(\lambda_o + \sum_V (\Psi(B^V, E_Q(A^V|X^o = 1)) - \Psi(B^V, E_Q(A^V|X^o = 0))))} \tag{1.16}$$

1.2.4 Evaluation Metrics

The standard evaluation metrics used for Multi-View Pedestrian Detection proposed in [26, 8] are as follows:

1.2.4.1 Multi Object Detection Accuracy ($MODA$)

To evaluate system performance accuracy, it is the primary performance indicator that accounts for missed detection and false positive counts i.e. it considers both false positives and false negatives. Assuming m_t denotes the number of misses and fp_t the number of false positives for each frame t, the Multiple Object Detection Accuracy ($MODA$) is computed as,

$$MODA(t) = 1 - \frac{c_m(m_t) + c_f(fp_t)}{N_G^t} \tag{1.17}$$

where c_f and c_m are the cost functions for the false positives and the missed detections and N_G^t is the number of ground truth in the t^{th} frame; c_f and c_m are used as scalar weights that can be changed depending on the application. For example, if missed detections are more important than false positives, we can raise c_m and reduce c_f. c_f and c_m are both equal$(= 1)$ in this evaluation. We compute *Normalized MODA (N-MODA)* as,

$$MODA(t) = 1 - \frac{\sum_{t=1}^{N_{frames}} (c_m(m_t) + c_f(fp_t))}{\sum_{t=1}^{N_{frames}} N_G^t} \tag{1.18}$$

1.2.4.2 Multi Object Detection Precision $(MODP)$

To evaluate the localization precision the spatial overlap information between the ground truth and the system output was used. The *Mapped Overlap Ratio* is computed as,

$$\textit{Mapped Overlap Ratio} = \sum_{i=1}^{N_{mapped}^t} \frac{|G_i^t \cap D_i^t|}{|G_i^t \cup D_i^t|} \tag{1.19}$$

where i^{th} ground-truth object in the t^{th} frame is denoted by G_i^t, for G_i^t the D_i^t denotes the detected object, and the number of mapped object pairs in the frame t is denoted by N_{mapped}^t. The Multiple Object Detection Precision $(MODP)$ for frame t is computed as,

$$MODP(t) = \frac{\textit{Mapped Overlap Ratio}}{N_{mapped}^{(t)}} \tag{1.20}$$

This gives us the detection precision in any given frame and by considering total number of relevant evaluation frames the measure is being normalized. If $N_{mapped}^t = 0$, then $MODP$ is forced to zero for that frame. We compute the *Normalized MODP (N-MODP)* that gives the precision of detection for the entire sequence,

$$N\text{-}MODP(t) = \frac{\sum_{t=1}^{N_{frames}} MODP(t)}{N_{frames}} \tag{1.21}$$

1.2.4.3 $Precision$ and $Recall$

$Precision$ refers to the proportion of your results that are relevant. $Recall$, on the other hand, is the percentage of total relevant results correctly classified by the algorithm . They both are calculated as follows.

$$Precision = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Positive}} \tag{1.22}$$

$$Recall = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Negative}} \tag{1.23}$$

1.3 Contributions

A multi-camera approach aggregates the multi-view cues for detections and alleviates the impact of occlusions in the crowded environment. But it was still unknown how well the multi-view detectors generalize to unseen data and in different camera setups, this becomes significant because a practical multi-view detector should be ready to use in a variety of scenarios. Therefore, in this thesis the following contributions are:

1. Conceptualizing and emphasizing the importance of generalization in Multi-View Detection and developed a novel GMVD (Generalized Multi-View Detection) dataset using GTA-V and Unity game engines for the same.
2. Highlighting the shortcomings of the current evaluation methodology and propose novel experimental setup on existing datasets such as WildTrack and MultiviewX.
3. Adapting the baseline architecture to bring generalization to deep MVD. Showing that ***permutation invariance*** of multiple-cameras as an input to the model is crucial for multi-view detection and average pooling is one minimal way to achieve it. We developed a novel ***drop view regularization***, where one of the camera view is dropped randomly while training. Usage of the more effective loss function like KLdiv (KL Divergence) and Cross Correlation (CC).
4. Demonstrated extensive set of experiments and ablation studies. Showing staggering improvements in scene and configuration generalization, paving the way for a practicable Multi-View Detection.

1.4 Thesis outline

The rest of the thesis is organized as follows: In Chapter 2 we investigate and formulate the generalization settings required for real-time or deployabale multi-view detection systems. In addition to this, propose a method and evaluation strategies to incorporate the generalization aspects. In Chapter 3 we identify the shortcomings of the benchmarking datasets and generate a generic and diverse synthetic dataset for multi-view pedestrian detection task using game engines.

Chapter 2

Generalization in Deep Multi-view Pedestrian Detection

2.1 Related Work

The first step in multiview pedestrian detection is to aggregate information from multiple RGB camera views. In 2.1.1 and 2.1.2, given a fixed assumption of human height and width, researchers fused multiple sources of information for multi-view 2D anchors. The corresponding multi-view 2D anchor boxes and all the ground plane locations are calculated first. In 2.1.3, anchor-free approach is been used which replaces inaccurate anchor boxes by sampling feature vectors from feature maps at corresponding points to represent ground plane locations.

2.1.1 Classical Methods

Seminal work by Fleuret [14] cast 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 use mean-field inference or conditional random field (CRF) for spatial aggregration [14, 5, 2]. The classical methods, however observe a gradual degradation in detection performance with increased crowds, as the background subtraction becomes less effective with increase in crowds and clutter. Some methods do away with background subtraction and rely on handcrafted classifiers [42] instead.

2.1.2 Anchor based MVD

Anchor based MVD methods replace background subtraction with anchor-based deep pedestrian detectors like Faster R-CNN [40], SSD [33] and YOLO [39]. Some of these methods process each view separately [49] and some process them simultaneously [4, 9]. The inaccuracies in the pre-defined anchor boxes [28] 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 Anchor based methods still rely on operations outside of Convolutional Neural Networks (CNNs), requiring to work out a balance between different potential terms [4].

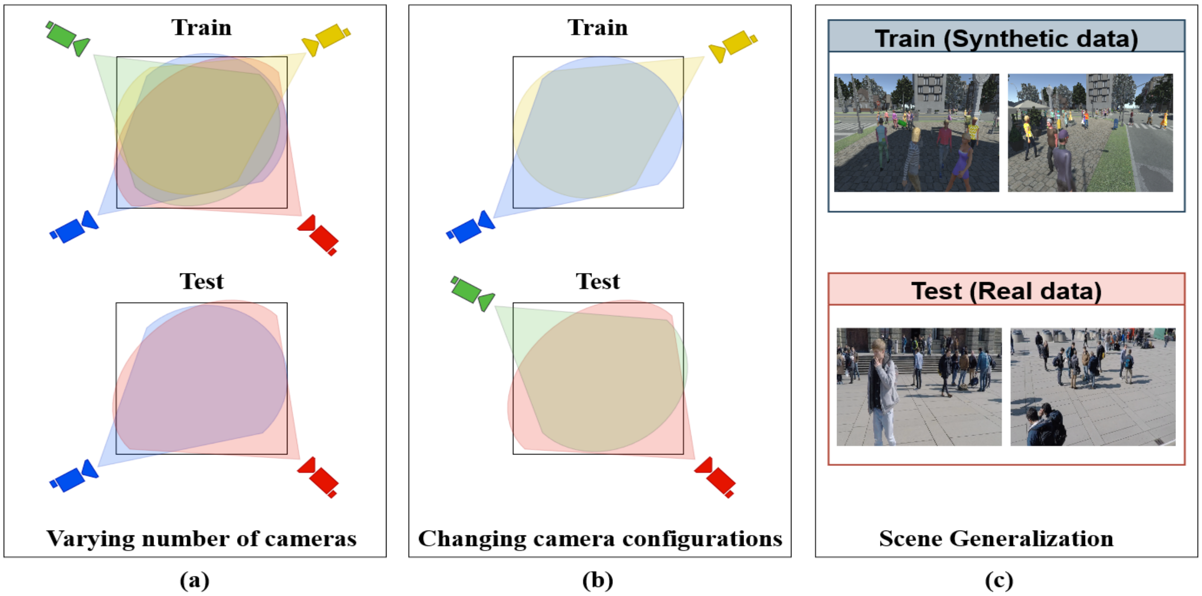

Figure 2.1: Three forms of generalization required in MVD: (a) varying number of cameras, (b) different camera configurations, and (c) generalizing to new scenes.

2.1.3 End-to-end Deep MVD

MVDet [23] 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 [21] 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 [8]). The idea from [23] was further enhanced by using deformable transformers [52] to improve the feature aggregation in MVDeTr [22]. More recently, SHOT [44] introduced a combination of homographies at multiple heights to improve the quality of the projections.

2.2 Generalization in Multi-View Pedestrian Detection

The solutions of Multi-View Detection (MVD) has evolved from classical methods to hybrid approaches and finally to end-to-end trainable deep learning architectures. Expectedly, the current landscape of MVD is dominated by end-to-end trainable deep learning methods. 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. Ideally, three forms of generalization abilities are essential for the practical scalability and deployment of MVD methods, which is illustrated in Figure 2.1.

- **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). The model need not to be re-trained again in such situations.
- **Varying camera configuration :** The model should not overfit to the specific camera postion and orientation. The performance should be similar even with altered camera positions, as long as they span the same dedicated ROI in the environment.
- **Scene Generalization :** Models trained on one scene should work on another scene (eg:- model trained on a traffic signal should work on a setup inside a university).

The most important property to be considered for generalization in an end-to-end trainable model of multi-camera system is ***Permutation Invariance*** property. The solutions proposed in the entire Section 2.1 from classical to End-to-end Deep MVD did not consider the *permutation invariance* property while designing the multi-view detection models. We need to understand, why *permutation invariance* is important for generalization? The simple anwer to this is, the order in which camera views are given as an input to the model should have this property of *permutation invariance*. If we provide camera inputs in the same sequence every time, the model learns the sequence of inputs, thus we need to change the sequence of inputs every time and make the model to be invariant to the order in which the inputs are given . This property also ensures the above mentioned three generalization aspects i.e varying number of cameras, varying camera configuration and scene generalization.

2.3 Our Developed Method

We developed an anchor free deep MVD method along the lines of [23, 22, 44] 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 of model is shown in Fig. 2.2. 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.

2.3.1 Feature Extraction and Perspective Transformation

Feature Extractor: We use a ResNet18 [21] 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.

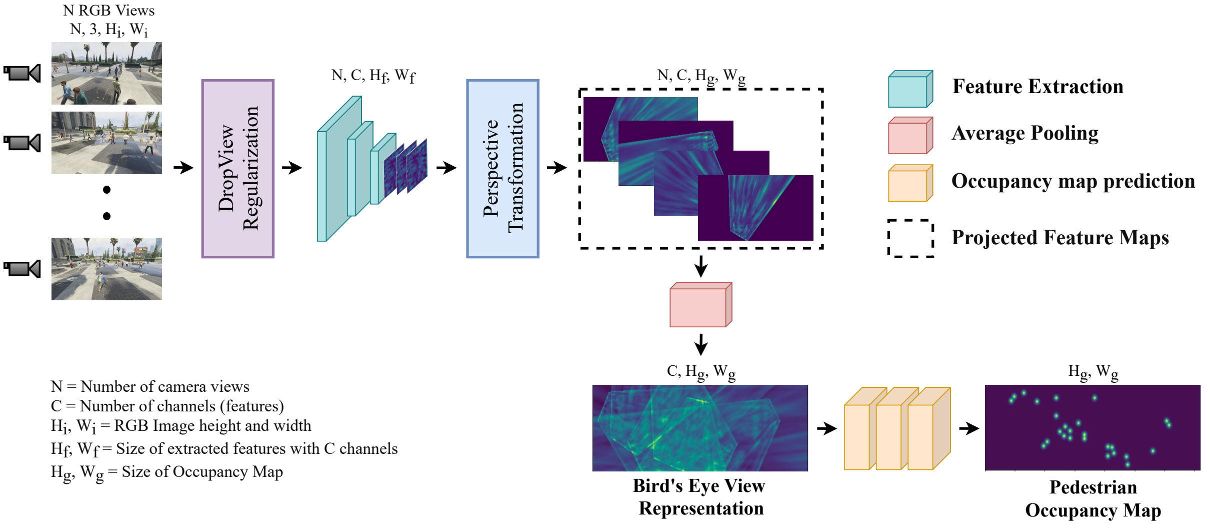
N RGB Views
N, 3, H_i, W_i
DropView Regularization
N, C, H_f, W_f
Perspective Transformation
N, C, H_g, W_g
Feature Extraction
Average Pooling
Occupancy map prediction
Projected Feature Maps
N = Number of camera views
C = Number of channels (features)
H_i, W_i = RGB Image height and width
H_f, W_f = Size of extracted features with C channels
H_g, W_g = Size of Occupancy Map
C, H_g, W_g
Bird's Eye View Representation
H_g, W_g
Pedestrian Occupancy Map

Figure 2.2: Our developed 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 architecture can work for an arbitrary number of views.

Perspective Transformation: The extracted features from the feature extractor are then projected to 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:

$$I = s \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = K[R|t] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = P \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.1}$$

where perspective transformation matrix is denoted by P and scaling factor by s.

2.3.2 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

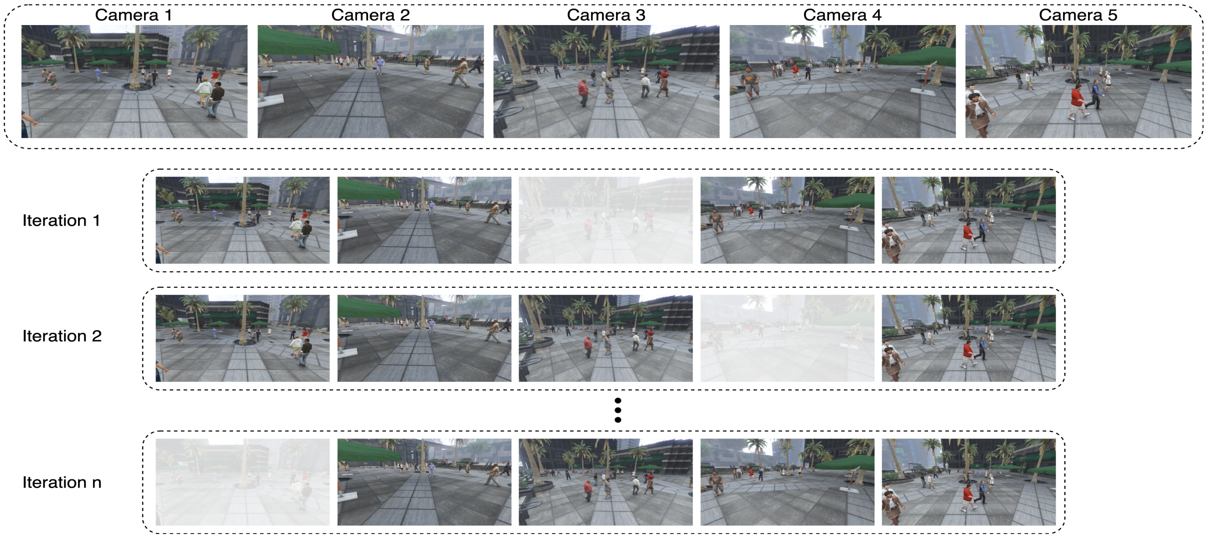

Figure 2.3: An illustration of DropView regularization

eye view feature representation F of size (C, H_g, W_g), which is written as,

$$F = \frac{\sum_{i=1}^{N} fm_i}{N}. \tag{2.2}$$

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, the developed 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 [45] as well as work on self-supervised learning which drops color channels to prevent the model from memorization [25, 29], we use the DropView regularization technique. For each sample, we randomly select one view to discard during training iterations, as illustrated in Fig 2.3. 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 [23], we use 3 dilated convolutional layers to predict the occupancy map of size (H_g, W_g).

2.3.3 Loss Function

The loss function is the comparison of the output probabilistic occupancy map (p) and the ground-truth (g). Inspired by the work on saliency estimation in images and vidoes [7, 38, 24], the combination of Kullback–Leibler Divergence (KLDiv) and Pearson Cross-Correlation (CC) metrics is used as a loss function. The combined loss function can be written as:

$$L(p, g) = \frac{\sigma(p, g)}{\sigma(p) \times \sigma(g)} - \sum_i g_i \log \left(\frac{g_i}{p_i} \right), \tag{2.3}$$

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

2.4 Experiments

2.4.1 Experimental setup

Datasets: In addition to our GMVD dataset, we use the WildTrack and MultiViewX datasets. The *WildTrack* dataset consists of 7 static calibrated cameras with 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 synthetic dataset *MultiViewX* 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, 90% frames are used in training and the last 10% frames for testing, as done in previous work [23, 8].

Evaluation metrics: We use the standard evaluation metrics proposed in [8]. *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 [26]. *Precision* and *Recall* is calculated as shown in Chapter 1 and subsection 1.2.4. A 0.5 meters threshold is used to determine the true positives.

State of the Art comparisons: We compare against nine different methods. The set includes one monocular object detection baseline (referred to as RCNN clustering [49]); a classical probabilistic occupancy map method [14]; four anchor based methods [30, 4, 9, 34] and three recent end-to-end trainable deep MVD approaches [23, 22, 44]. For generalization experiments, we only compare against the recent state-of-the-art methods MVDet [23], MVDetr [22] and SHOT [44].

2.4.2 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

Table 2.1: Comparison against the state-of-the-art methods. Our method refers to the developed model in Section 2.3. 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 [49]	×	11.3	18.4	68.0	43.0	18.7	46.4	63.5	43.9
POM-CNN [14]	×	23.2	30.5	75.0	55.0	-	-	-	-
Lopez-Cifuentes [34]	×	39.0	55.0	-	-	-	-	-	-
Lima [30]	×	56.9	67.3	80.8	74.6	-	-	-	-
DeepMCD [9]	×	67.8	64.2	85.0	82.0	70.0	73.0	85.7	83.3
Deep-Occlusion [4]	×	74.1	53.8	95.0	80.0	75.2	54.7	97.8	80.2
MVDet [23]	×	88.2	75.7	94.7	93.6	83.9	79.6	96.8	86.7
MVDeTr [22]	✓	91.5	82.1	97.4	94.0	93.7	91.3	99.5	94.2
SHOT [44]	×	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)

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 [11] weights while training our model.

2.4.3 Results

Like prior works, we evaluate our approach on the WildTrack and MultiViewX datasets in Table 2.1. We find that our developed 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 [22, 44], 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

Table 2.2: 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**

protocol is highly misleading since the train and test sets have significant overlap, thereby 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 2.2. Our 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 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 2.4. 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 outperforms all other models by over 20 percentage points.

Table 2.3: 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

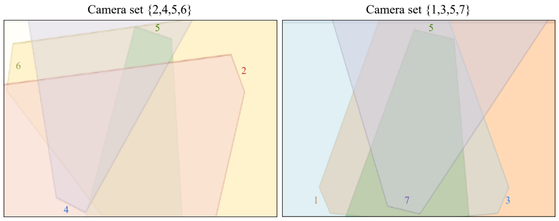

Figure 2.4: Camera splits of WildTrack dataset for changing camera configuration experiment.

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 2.3. Our 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 [44] where the MultiViewX scene is split into two halves, and each half is covered using 3 cameras each. In this setting as well (Table 2.5), our approach with DropView regularization has a MODA score of 66.1, which is significantly higher than both SHOT (49.1) and MVDeTr (56.5).

Table 2.4: Experiments on the WildTrack dataset with changing camera configurations

Trained on camera set			Inference on {2,4,5,6}				Inference on {1,3,5,7}			
		Method	MODA	MODP	Prec	Recall	MODA	MODP	Prec	Recall
	{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	72.9	92.4	91.6	**75.1**	71.1	94.3	**79.9**
	{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

Table 2.5: Changing configuration and scene generalization experiment on the setting introduced in [44]

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

2.5 Choice of Loss Function

We ablate the choice of the loss function in Table 2.6 for the scene generalization experiment. We consider the Mean Squared Error (MSE), KL-Divergence(KL), Pearson Cross-Correlation (CC), as well as our chosen loss function (KL+CC). We find that the combination of KL-Divergence and Pearson Cross-Correlation achieves significantly better results than any other loss function.

2.6 Qualitative results

First we show the predicted occupancy maps of MVDet, MVDeTr, SHOT 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.

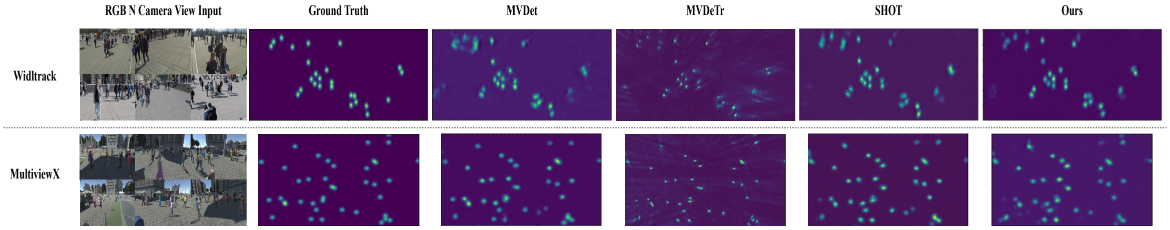

Figure 2.5: Sample frames from WildTrack and MultiViewX dataset with corresponding occupancy maps of ground truth, our result MVDet, MVDeTr and SHOT for comparison. We can see the clusters forming in the MVDet predictions, in contrast our method gives much sharper and distinct predictions.

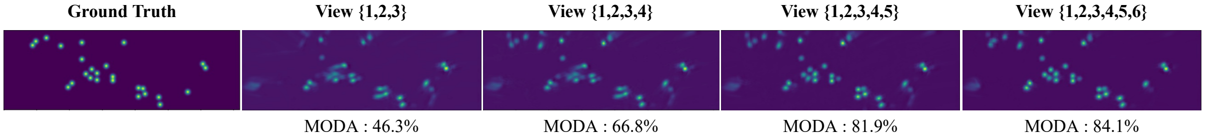

Figure 2.6: Occupancy maps for varying number of cameras on WildTrack dataset when trained on seven cameras and tested on varying subsets of the cameras.

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

Table 2.6: Choice of Loss Function: we present an ablation study for our method on the scene generalization experiment. Overall, the model trained with both KL-Divergence and Cross-Correlation achieves the best performance.

2.6.1 WildTrack Dataset

The traditionally evaluated results which contains occupancy maps of ground truth, our method, MVDet, MVDeTr and SHOT are shown in Fig. 2.5. The occupancy map from our method which uses average pooling, KLCC loss function and ImageNet pretraining gives us more accurate localization as compared to the base MVDet architecture. The results (maps) are competitive when compared to SHOT and MVDeTr. The maps obtained using MVDeTr are sharper and focused, however, it also has more false positives.

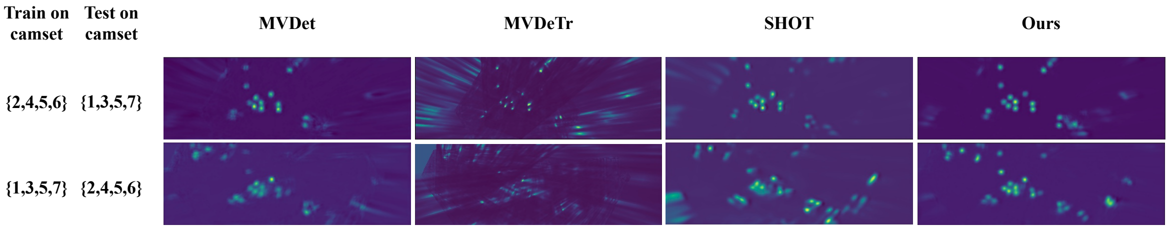

Figure 2.7: Result occupancy maps for cross subset evaluation from WildTrack dataset.

Varying number of cameras: The output occupancy map for varying number of cameras are shown in Fig. 2.6. 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. 2.7. 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 like the camera splits shown in Figure 2.4. 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).

2.6.2 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, MVDeTr and SHOT model and observations are similar to that of the Wildtrack dataset. Fig. 2.5 shows the traditionally evaluated results.

Varying number of cameras: The output occupancy map for varying number of cameras are shown in Fig. 2.9. 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. 2.10. 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, the camera splits are shown in Figure 2.8 and their results are shown in Table 2.7.

2.6.3 Scene Generalization

The qualitative results of output occupancy map for cross-dataset evaluation are shown in Fig. 2.11, when we train on synthetic dataset (MultiViewX) and test on real dataset (WildTrack). First four occupancy maps are the outputs of MVDet, MVDeTr, SHOT and our method when tested on only 6 views of WildTrack dataset for having a fair comparison with other methods. We also show the output

Trained on camera set		Inference on {1,3,4}				Inference on {2,5,6}			
	Method	MODA	MODP	Prec	Recall	MODA	MODP	Prec	Recall
{1,3,4}	MVDet	72	76.1	93.5	77.4	46.3	66.4	94.5	49.1
	MVDeTr	**77.4**	**85.1**	97.9	79	60.4	71.3	95.4	63.5
	SHOT	74.3	76.3	94.1	**79.3**	37.3	67	67.5	**72.1**
	Ours	67.7	76.4	96.2	70.5	59.6	73.4	94.7	63.2
	Ours (DropView)	67.3	75.3	**98.4**	68.5	**62.9**	**73.6**	**96.3**	65.4
{2,5,6}	MVDet	34.3	66.2	93.8	36.7	77.6	77.4	93.8	83.1
	MVDeTr	51.1	72.1	**94.9**	54	**83.1**	**87.1**	**97.8**	**85**
	SHOT	47.3	**73**	94.2	50.3	80.7	78.7	96.1	84.1
	Ours	45.8	71.8	94.5	48.6	76.1	78.7	95.9	79.5
	Ours (DropView)	**53.4**	71.6	88.2	**61.6**	75.2	77.4	92.8	81.5

Table 2.7: Experiments on the MultiViewX dataset with changing camera configurations

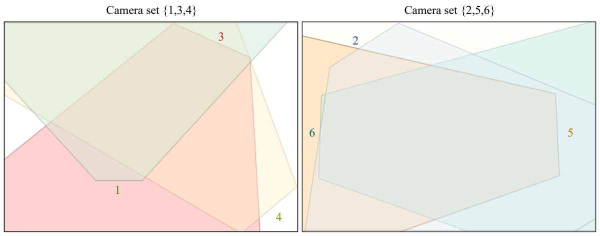

Figure 2.8: Camera splits of MultiViewX dataset for changing camera configuration experiment shown in Table 2.7.

occupancy map when tested on all the views of WildTrack dataset. Our method provides accurately localized occupancy maps and disambiguate the occlusions as compared to other methods.

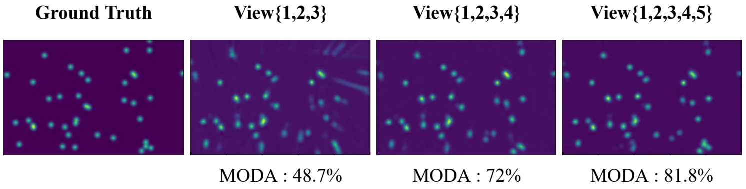

Figure 2.9: Occupancy maps for varying number of cameras on MultiViewX dataset when trained on seven cameras and tested on varying subsets of the cameras.

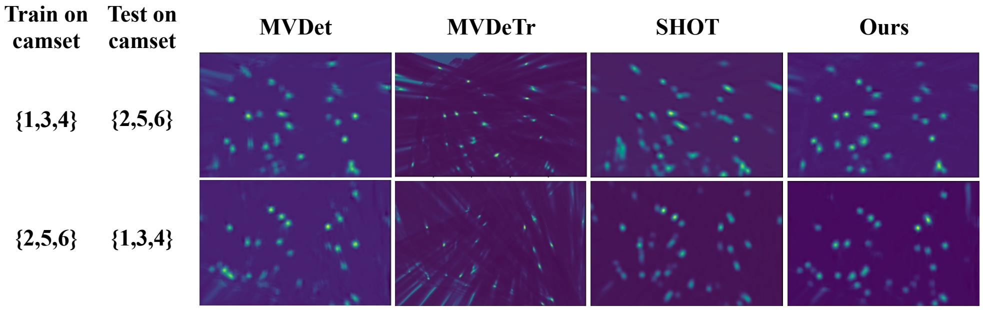

Figure 2.10: Result occupancy maps for cross subset evaluation from MultiViewX dataset.

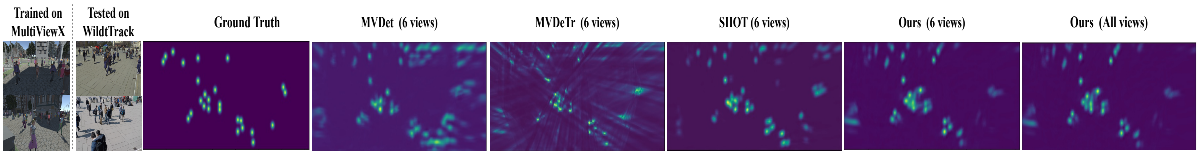

Figure 2.11: Occupancy maps obtained on inference from WildTrack dataset where the models where trained on the synthetic dataset (MultiViewX).

Chapter 3

Multi-View Detection Dataset

3.1 Introduction

In recent years, AI and ML have had a significant impact on a wide range of applications ranging from images, videos to text understanding and speech recognition. Many of the recent successes can be attributed to improved computation and a large amount of training data. Data collection has emerged as a critical bottleneck amongst many challenges in machine learning. It is well known that the majority of the time spent on running machine learning end-to-end is spent on data preparation, which includes collecting, cleaning, visualising, analysing and feature engineering. While all of these steps take time and because of insufficient training data, data collection becomes very important and has recently become a challenge in newer applications. Furthermore, as deep learning has grown in popularity, there is an increased demand for training data. Feature engineering is one of the most difficult steps in traditional machine learning because the user must understand the application and provide features used for training the models. On the other hand, Deep learning can generate features automatically, saving us from the need for feature engineering, which is an essential part of data preparation. But deep learning requires more training data to perform well. As a result, accurate, scalable, and large amounts of data for training AI models have become the necessity. Figure 3.1. shows an overview of the data collection research landscape for machine learning and AI tasks. Labeling data has traditionally been a natural topic of research for machine learning problems. Semi-supervised learning, for example, is a classic problem where model training is performed on a small amount of labeled data and significantly high amount of unlabeled data. However, because AI models must be trained on massive volumes of training data, data management issues such as acquiring large datasets, performing data labeling at scale, and improving the quality of huge amounts of existing data become increasingly important.

While traditional computer vision methods did not require any training data, making them unsupervised techniques, the performance of methods based on the supervised learning paradigm heavily depends on the size and quality of training datasets. Deep learning's enormous potential, in particular, can only be fully realised with large datasets. Significant research effort has gone into developing such datasets, for

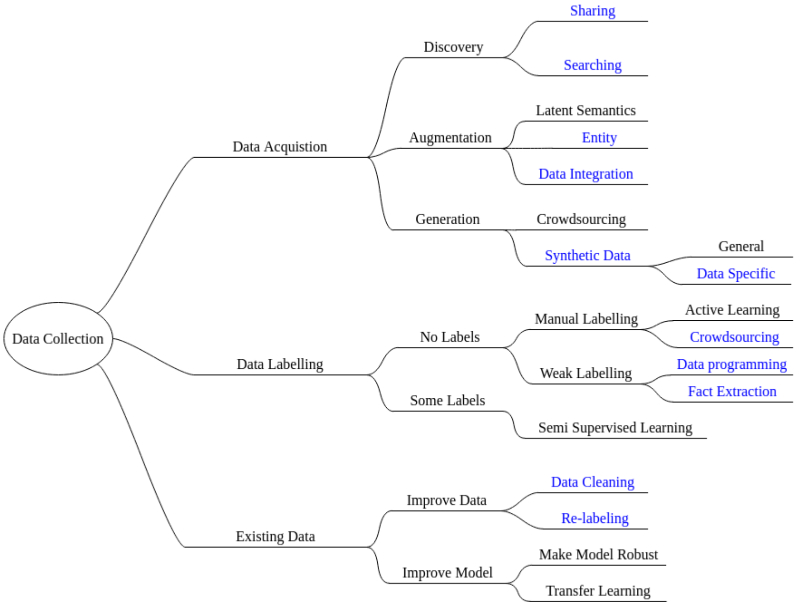

Figure 3.1: Research landscape of data collection for ML and AI tasks

example some famous datasets are ImageNet [11], MNIST [12], MS COCO [31], CityScapes [10], or the NYU dataset [35]. These datasets have enabled the majority of recent advances in computer vision. But there are various challenges to collect such large scale real datasets and there becomes a need for synthetic dataset. In subsequent subsections 3.1.1 and 3.1.2 will discuss about the key challenges for collecting real dataset and the importance of synthetic dataset.

3.1.1 Key Challenges of collecting Real Dataset

3.1.1.1 Biased Data

The undesirable patterns or behaviors that learned from data are termined as bias, these are highly influential in the model decisions but are not aligned with the ideal decision of the society in which they operate. Bias in models can occur due to a variety of factors, including age, gender, race, or even the intersection of these and other characteristics. Such bias occurs when model is deployed and when training data underrepresents some subset of the population as input.

3.1.1.2 Consent and Privacy

These are the concepts that computer vision practitioners do not adequately address in data collection. Human subjects research is exempt from the review of Research Ethics Board when it is based on publicly available information and the recognized individuals who have no expectation of privacy. To what extent do people give up their right to privacy when they post content online or when others post content about them without their permission?. For example, in face recognition, researchers frequently rationalise data collection by restricting datasets to identities of celebrities, assuming that these individuals have lower privacy expectations. Some researchers allow individuals, celebrity or not, to opt-out of being included in face datasets, indicating an appreciation for the collection's non-consensual nature.

3.1.1.3 Annotation and Labelling Cost and Time

Labels or annotations in are often termed as "ground truth". To label the data, the annotation and labelling process for computer vision requires a great deal of skill, knowledge, and effort. While labelling various problems are been encountered which makes the labelling tasks more time taking and ineffective. The challenges which are faced during this process are; *first*, to manually annotate label data images, we require a large workforce capable of producing a large volume of training data; *second*, it is not enough to simply generate data; maintaining high quality is also necessary; otherwise, the models will not be trained with the appropriate inputs, therefore, requires labelers to have domain expertise; *third*, to generate high quality data selection of right tools and techniques is also vital; *fourth*, the most important part in annotation and labelling process is the cost involved for generation of such massive data.

3.1.1.4 Other Challenges

Recently, because of COVID-19 pandemic and restictions in place, collecting real data was a bigger challenge which involves humans, other living beings, vehicles, other non-living objects for applications such as multi-object detection and tracking, sports, self-driving cars, surveillance, etc. Hardware setup in computer vision applications such as, placements and positions of multiple cameras in multi-camera setup, camera calibrations, etc. are also one of the critical challenges faced in real data collection process.

3.1.2 Importance of Synthetic Dataset

Synthetic data is validated information generated by computer simulations or algorithms in place of real-world data. For decades, synthetic data has existed in some form or another. It can be found in computer games such as flight simulators and scientific simulations. Gartner 3.2 predicted in a June 2021 report on synthetic data that by 2030, most of the data used in AI will be generated artificially by rules, statistical models, simulations, or other techniques. In dealing with privacy concerns and reducing bias; synthetic data are important by ensuring information diversity to accurately represent the real world. Everything in a synthetic data simulations can be controlled, i.e they are fully user controlled. They are perfectly annotated, a variety of annotations are generated automatically, which is one of the main reasons why synthetic data is so inexpensive when compared to real data. The primary cost of synthetic data is the initial investment in developing the simulation. Following that, generating data is exponentially less expensive than real data. Multi-spectral data can be generated from synthetic data, i.e autonomous vehicle manufacturers have realized that collection and annotation of non-visible or real data is more difficult.

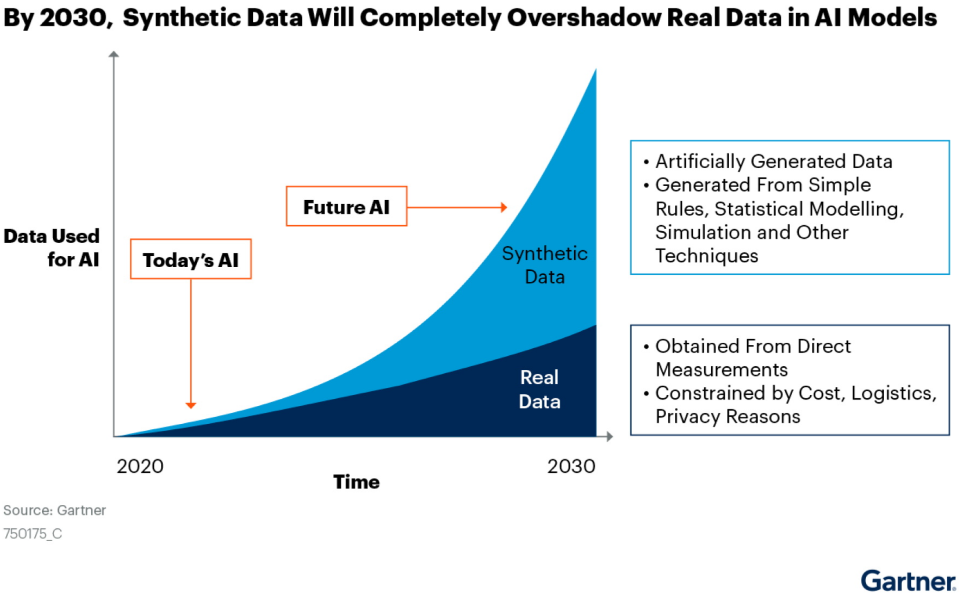

Figure 3.2: Gartner prediction for synthetic dataset usage in AI tasks.

This is the reason they have been among the most vocal supporters of synthetic data. Simulations are used by companies such as Alphabet's Waymo and General Motors' Cruise to generate synthetic LiDAR data. Because this data is synthetic, the ground truth is known, and the data is labelled automatically. Similarly, synthetic data works well in computer vision applications involving infrared or radar imagery, where humans cannot fully understand the imagery.

Given the challenges of collecting real dataset and benefits for generating synthetic dataset. The formulation and evaluation strategies of three generalization settings discussed in previous Chapter 2 and the shortcomings of current benchmark datasets from Figure 3.3 motivates to curate the generic and diverse dataset. So, in thesis, we are focussing our work on curating a synthetic multi-view pedestrian detection dataset which is generic and diverse and can be used for evaluating the generalization capabilities of the MVD methods. In section 3.2 we will discuss about the multi-camera pedestrian detection datasets been used in the literature. In Section 3.3, the steps involved in dataset generation and in Section 3.4 we define the charachterisitcs of the curated dataset and finally in Section 3.5 we show the epxeriments performed and the state-of-the art results.

3.2 Related Work

We list the commonly used pedestrian datasets with a focus on the multi-camera ones. As overlapping we refer to multi-camera datasets whose camera's have strictly overlapping fields of view.

Table 3.1: Commonly used multi-camera person detection and tracking datasets.

Dataset	Resolution	Overlapping	IDs	# Cameras	Ground Truth
Duke MTMC	1920×1080	×	2000	8	-
PETS 2009 S2.L1	720×576	✓	19	7	2D
Laboratory	320×240	✓	6	4	3D
Terrace	320×240	✓	9	4	3D
Passageway	320×240	✓	13	4	3D
SALSA	1024×768	✓	18	4	3D
Campus	1920×1080	✓	25	4	2D
EPFL-RLC	1920×1080	✓	-	3	3D
WildTrack	1920x1080	✓	313	7	2D, 3D
MultiViewX	1920x1080	✓	350	6	2D, 3D
GMVD (Ours)	1920x1080	✓	2800	3, 5, 6, 7, 8	2D, 3D

Duke MTMC Duke MTMC [41] dataset does not belong to this category, as only 2 of its camera's fields of view slightly overlap. Being a real dataset, in 2019 original authors have terminated the Duke MTMC dataset.

PETS 2009 S2.L1 The most widely used dataset with an overlapping camera setup is the PETS 2009 S2.L1 [13] sequence. In part due to the slope in the scene, the provided calibration poses large homography mapping deterioration and inconsistencies in the projection of 3D points across views (as noted also in [[37], p. 10], [[16], p. 10], [[9], p. 3]). Besides being a small scale dataset, the PETS 2009 S2.L1 is acquired in an actor setup. Hence it does not allow for good generalization and fair benchmarking of appearance based methods. Recently PETS 2009 dataset also has been taken down.

EPFL campus The three sequences which are been shot at the EPFL campus [14]: Laboratory, Terrace and Passageway is overlapping multi-camera datasets, they have a small total number of identities and are relatively sparsely populated. From Table 3.1 we can see that, Laboratory, Terrace and Passageway are of very small size and has low image quality.

SALSA The SALSA [3] also has overlapping multi-camera setup, a cocktail party of 30 minutes is recorded, where the people are not moving most of the time i.e static, making this dataset less difficult.

Campus The Campus [49, 50] has multi-camera setup with overlapping fields of view, but does not provide 3D annotations to localize people.

EPFL-RLC The EPFL-RLC [9] dataset outperforms PETS in terms of joint-calibration accuracy and synchronisation. However, rather than providing a complete groundtruth, this dataset represents a collection of a balanced set of positive and negative multi-view annotations and is used for classification of a position as occupied by a pedestrian. Entire ground-truth annotations are provided for a small subset of the last 300 of the total 8000 frames, originally used for testing [12]. Moreover, it is acquired with only three cameras which have a much more limited field of view. This results in a approx. 10-fold smaller number of detections on average per frame.

Wildtrack The Wildtrack [8] improves upon other multi-camera person datasets because of (i) the high precision calibration and synchronisation between the cameras; and (ii) the large number of annotations that allows for developing deep learning based multi-view detectors. It exceeds the total number of annotations and the regions of interest (ROI's) are of significantly larger resolution. It consists of 7 static cameras with overlap FOV's and 400 synchronized frames. Wildtrack being a real dataset is highly acceptable dataset for benchmarking the multi-camera pedestrian detection models. But the dataset have only one camera configuration, with same scene and using same number of cameras. Even the environmental conditions (time, weather, etc.) are similar across train and test splits i.e its comprises of single short sequence.

MultiviewX The synthetic dataset MultiviewX [23] is curated using Unity engine [1] and which uses models of human from PersonX [46]. It has 6 cameras and slighlty smaller ROI's and rest it has same configuration as Wildtrack dataset such only one camera configuration, wih same scene, with same number of cameras and same environment conditions across train and test splits.

Given the absence of diverse dataset and shortcomings of benchmark dataset such as Wildtrack (real) and MultiviewX (synthetic). From Figure 3.3 it shows that the evaluation strategy in both these datasets is unreliable and prone to overfitting. Therefore in this thesis, we have curated a diverse synthetic dataset known as Generalized Multi-View Detection (GMVD) dataset to exploit and benchmark the generalization abilities of the multi-view detection systems.

3.3 Generalized Multi-View Detection Dataset Generation

In the thesis, using Grand theft Auto V (GTA V) and Unity Game Engine we are curating a synthetic dataset for multi-view detection task and to support our generalization claims as mentioned in Chapter 2. Figure 3.4 demonstrates general data collection pipeline adopted in this thesis.

3.3.1 GTA V

Grand Theft Auto V (GTA V) [15] is a 2013 action-adventure game developed by Rockstar North and published by Rockstar Games. Which is made up state of San Andreas i.e it is fictional. The commercial video game Grand Theft Auto V (GTA V) has detailed all aspects of the world with realistic graphics, and provides a diverse environment for data collection. Grand Theft Auto V's publisher allows non-commercial usage of the frames and video sequences of the game with certain restrictions such as spoilers should not be distributed.

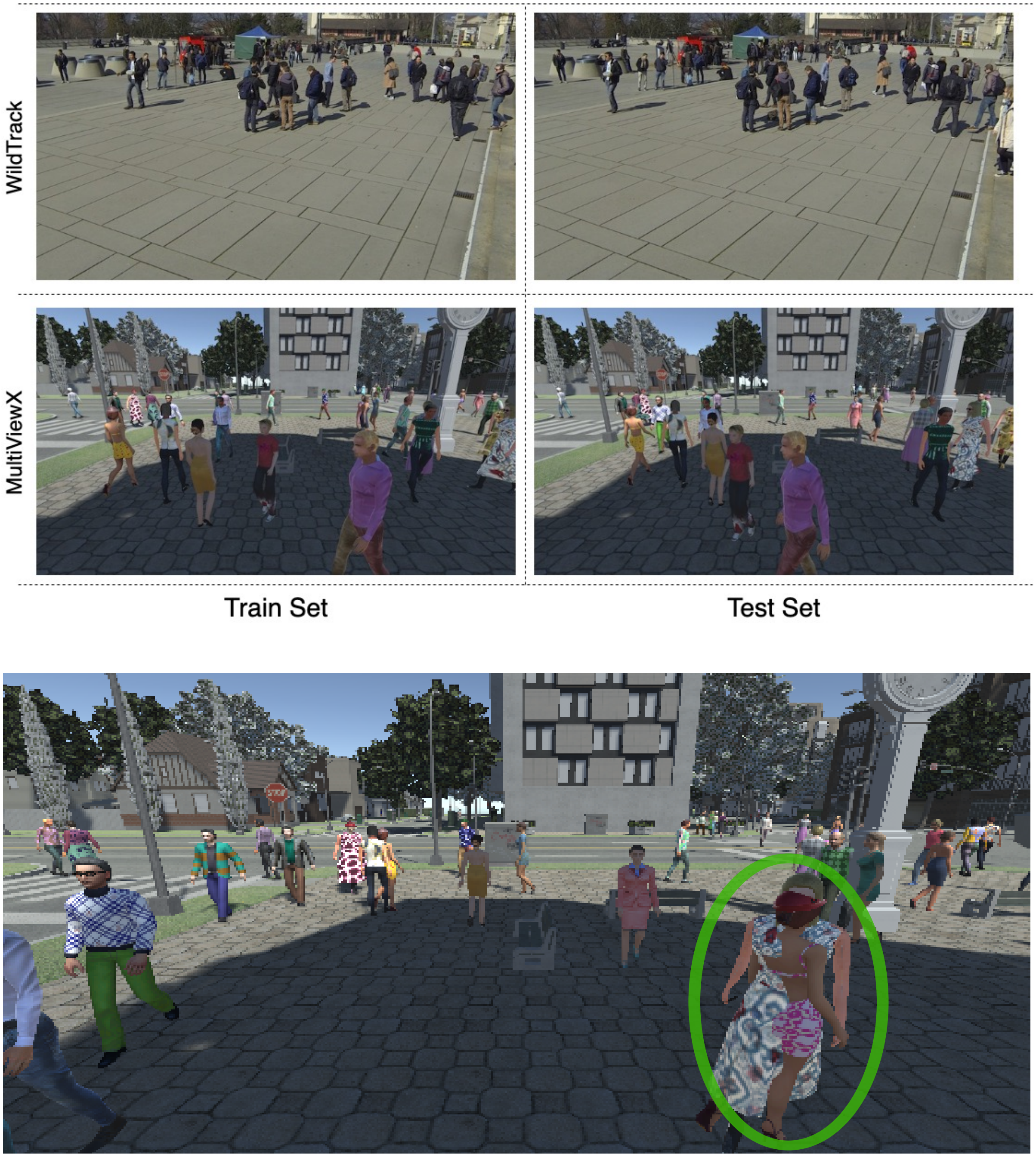

Figure 3.3: **Shortcomings of benchmark datasets :** The train and test sets of Wildtrack (first row) and MultiViewX datasets (second row) have significant overlap. We show the last image of the training set (left) and the first image of the test set (right). In both datasets, the appearance of several pedestrians is already seen in the training set. In Wildtrack, there are many static pedestrians as well. The MultiviewX dataset contains frames which shows the collision of two human models as highlighted with green in the (last row) sample frame.

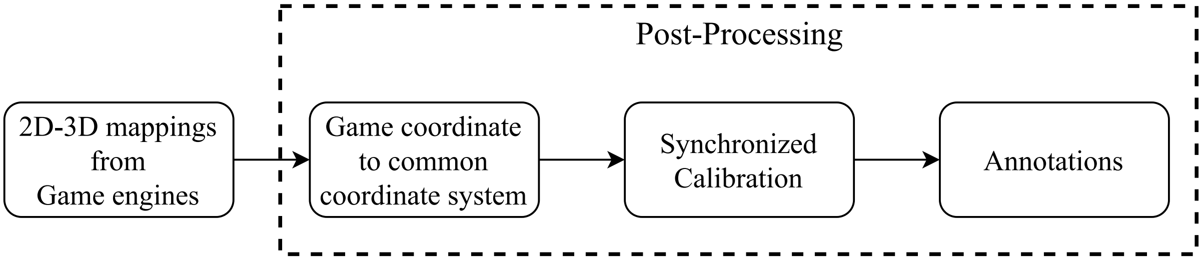
Post-Processing
2D-3D mappings from Game engines
Game coordinate to common coordinate system
Synchronized Calibration
Annotations

Figure 3.4: Data collection flowchart

3.3.2 Scripthook V

Script Hook V [6] is a free utility for the video game which allows you to use script native functions for GTA V with custom ASI plugins. This mod, created by Alexander Blade, is a plugin library that also includes the most recent Native Trainer and ASI Loader. This allows your mods to function and interact properly with the popular game Grand Theft Auto V. The tool doesn't work for online version of Grand Theft Auto.

3.3.3 Unity Engine

Unity [1] is a game engine which can be created on multiple platforms. In 2005, Unity Technologies released Unity engine. Unity's primary focus is the creation of 3D and 2D games, as well as interactive content. It is used as a framework that allows you to create once and publish everywhere. Time and cost are reduced when we use some Unity features such as asset tracking, rendering, and scripting. Recently unity has developed a toolkit for generating computer vision dataset on a large scale using Perception package [47]. It is currently focused on a few camera-based use cases, but will eventually expand to other types of machine learning tasks.

3.3.4 Data Acquisition using GTA V and Unity game engines

We used Script Hook V library to interface with the GTA V environment and used MTA-Mod [27], this repository contains two Grand Theft Auto V Mods which were used for creating the GMVD Dataset. In Unity environment, we used PersonX [46] 3D human models to create the pedestrians. Below are the the common steps required in curating Multi-view dataset either using GTA or Unity engine.

1. **Identifying scene :** Exploration of the GTA V game manually to identify the locations where actual pedestrians movement happens; indoor locations such as subways, and outdoor locations such as malls, parks, railway station compounds, the airport area, etc. In Unity, the scene is manually created by putting together 3D models for street, buildings and other props.

2. **Camera positioning and orientation :** Once the location is identified, we need to understand the scene to where we can place the cameras and what would be its orientation, since we want the cameras to be strictly overlapping.

3. **Identifying the ground plane origin and ROI :** As we are discretizing the ground plane, we need to identify the origin of the ground plane and define the Region of Interest (ROI) where we need to perform detections.

4. **Spawning Pedestrians :** We can spawn pedestrians at random positions in the scene after placing cameras and identifying the ROI and origin.

5. **Changing time and weather :** GTA V provides the control to change the time to morning, afternoon, night, evening, etc and similarly we can change weather to sunny, cloudy, rainy, snowy, etc. In Unity, we did not had any time and weather changes.

6. **Assigning Tasks to Pedestrians :** We can assign tasks/actions to be performed by spawned pedestrians in the given scene. Tasks such as wander in an area of certain radius, move from one location to another at certain speed, etc. We require these task to capture random movements of the pedestrians in the scene.

7. **Recording the sequences :** We can start recording the sequences once everything is set for a particular scene. Recording will capture the frames at 41 fps in GTA V and Unity we capture at 30 fps and we store the 2D and 3D positions of the pedestrians at each frame for every camera view for post-processing step.

8. **Camera Calibration :** Calibration of cameras to compute intrinsic and extrinsic parameters need to be performed based on 2D and 3D locations we obtain from the GTA V game and Unity for obtaining proper annotations w.r.t synchronized multiple cameras. If we don't get proper calibrations in post-processing step we need to repeat again the steps 2,4,6 and 7.

In GTA V, 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 [27]. 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 Unity, just to avoid collision errors (which are present in MultiViewX 3.3 dataset), pedestrians were spawned at random locations within the region of interest. Steps 1, 2, 3 and 8 being the most time consuming process since there is no proper documentation for scripthook to obtain the desired location and its corresponding co-ordinates, placement of camera on the ROI and changing its orientation. We need to re-verify by calibration process, if desired calibration and annotations are not generated then need to repeat the steps the above steps.

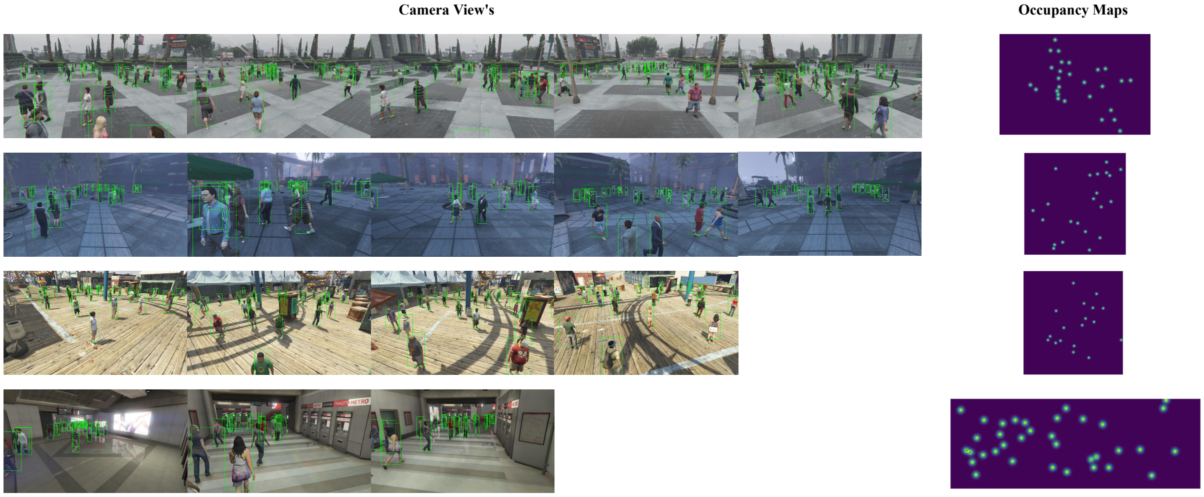

Figure 3.5: Synchronized camera calibration and sample ground truth annotations generated after post-processing step are shown in terms of bounding boxes in respective camera view's and the top view occupancy maps for GMVD Dataset.

3.3.4.1 Post-processing

Once we get 2D and 3D matchings from the game engines, we do a post-processing step to convert the 3D game coordinates to a common coordinate system i.e 3D game coordinate of grid plane origin (-800, -900, 0) gets converted to (0, 0, 0) as the origin which can be further used for re-calibration and generating annotations. After that, we use similar procedure as mentioned in [23] for synchronised camera calibrations and generating annotations as a post-processing step. Some of the samples of annotations in terms of bounding boxes and ground truth occupancy map is being shown in Figure 3.5

3.4 Generalized Multi-view Detection Dataset Characteristics

We generated a new MVD dataset incorporating the three forms of generalization discussed above (Figure 2.1). Some example frames from the generated Generalized Multi-View Detection (GMVD) dataset are illustrated in Figure 3.6. 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

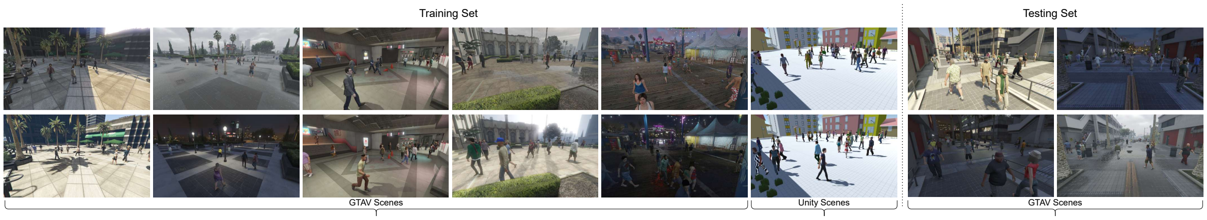

Figure 3.6: The generated GMVD Dataset includes seven scenes. Each column illustrates frames from one of the views from two different sequences of the same scene. The first six scenes are used for training and the last scene with two configurations are reserved for testing. Additionally, there are noticeable lighting and weather variations within each scene.

Table 3.2: 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

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 [19, 51] which utilize multi-source datasets to improve generalization performance. The GMVD dataset includes seven distinct 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, there are approximately 2800 person indentities as shown in Figure 3.7. The dataset also includes significant variations in lighting conditions. Local illumination sources come into play due to the presence of indoor and night scenes. We compare our dataset with the existing ones in Table 3.2. 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.

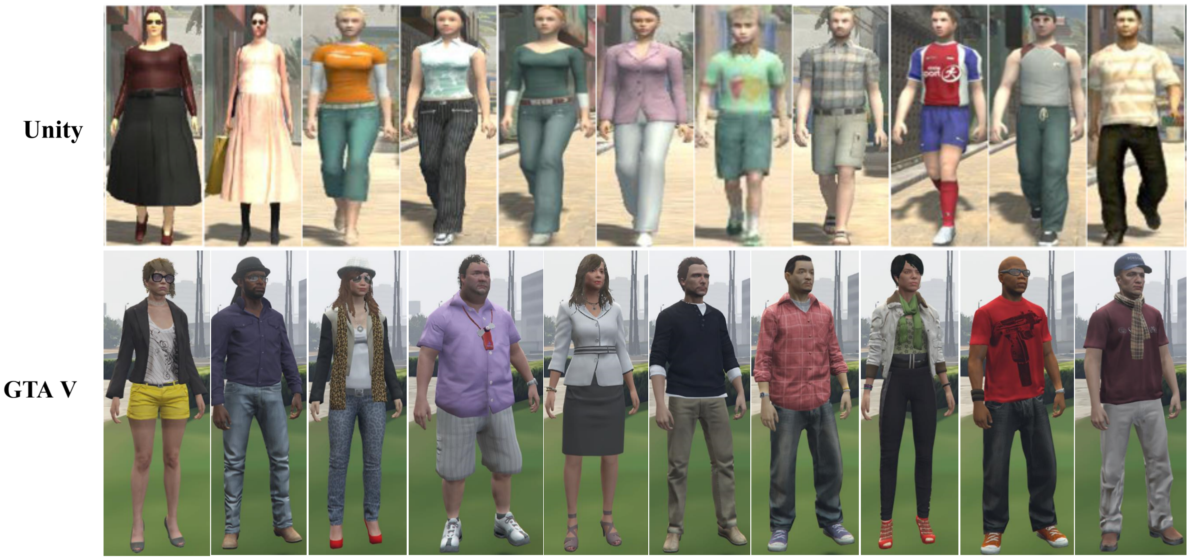

Figure 3.7: Samples of various person identities are shown from both Unity and GTA V, which are included in GMVD Dataset.

Table 3.3 shows the comparison of our dataset with exisiting ones based on the ground plane grid area (ROI) in meters being used for multi-camera detection, the dimensions to generate Top View (Bird's Eye View) represenation of the ROI and the density of the pedestrians in the scene per frame basis (defined by crowdedness).

Our work focuses on a comprehensive analysis of the problem of Multi-View Detection. However, the 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.

Thereby, we propose the GMVD dataset as a new benchmark for MVD. 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 .

3.5 Experiments and Results

Having shown in 2 that our model is capable of comprehensive generalization abilities, we show comparison with other methods and benchmark our developed approach on the GMVD dataset (Table 3.4). 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

Table 3.3: Region of Interest (top view area) for various scenes of GMVD Dataset compared with Wildtrack and MultiviewX

Dataset	Grid Area	Top View Dimensions	Crowdedness
WildTrack	12 × 36 m^2	480 × 1440	20 person/frame
MultiViewX	16 × 25 m^2	640 × 1000	40 person/frame
GMVD(ours)			
GTA scene 1	20 × 30 m^2	800 × 1200	20-50 person/frame
GTA scene 2	30 × 12 m^2	1200 × 480	20-50 person/frame
GTA scene 3	25 × 25 m^2	1000 × 1000	20-50 person/frame
GTA scene 4	29 × 19 m^2	1160 × 760	20-50 person/frame
GTA scene 5	28 × 27 m^2	1120 × 1080	20-50 person/frame
GTA scene 6	33 × 31 m^2	1320 × 1240	20-50 person/frame
Unity scene 1	16 × 25 m^2	640 × 1000	40 person/frame
Unity scene 2	16 × 25 m^2	640 × 1000	40 person/frame

Table 3.4: 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**

can be adapted to this setting, since they can be trained only on a *fixed* set of cameras. We stack dummy top view map to the existing methods to be adapted on our generalization settings and can be trained on our GMVD dataset. MVDet method was easily adaptable and trained on GMVD dataset by adding dummy top view map but SHOT and MVDeTr cannot be trained on our GMVD dataset due to their single view dependency for computation and their significant impact on the loss function. 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. Notably, this shows that training on our synthetic dataset, we can nearly attain the same performance as training on WildTrack itself. When evaluated on GMVD test set, our model achieves a MODA score of 68.2. The results empirically suggest 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.

3.6 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 curated 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.

Chapter 4

Conclusions

In this thesis, we explored alleviating some challenges in practical scenarios in the Multi-View Detection system. In particular, much emphasis is on the generalization and evaluation strategies of the MVD systems, which need to be adopted for benchmarking.

We find that the existing Multi-View Detection setup are severely limited and encourages models to overfit the training configuration. We identified and showed the importance of *Permutation Invariance* property to be considered for MVD systems. Therefore, we conceptualized and formalized three critical forms of generalization and outlined the experiments to evaluate them in more practical settings: generalization with i) a varying number of cameras, ii) varying camera positions, and finally, iii) to new scenes. We find the state-of-the-art models to have poor generalization capabilities and on this evaluation 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 architecture.

In addition to this, we show the shortcomings of the existing multi-view detection datasets and the challenges of curating the real dataset. Therefore, we generated a synthetic but diverse and realistic dataset using GTA-V and Unity game engines that can be used for both evaluations as well as training MVD methods. We demonstrated our developed method and benchmarked the state-of-the-art results on our GMVD Dataset and on a real dataset i.e. WildTrack , which gives comparable results when performing synthetic to real transfer. Overall, we hope our work plays a crucial role in steering the community towards more practical Multi-View Detection systems.

Related Publications

- **Bringing Generalization to Deep Multi-View Pedestrian Detection**. Jeet Vora, Swetanjal Dutta, Kanishk Jain, Shyamgopal Karthik, Vineet Gandhi. **Accepted at WACV 2023** IEEE/CVF Winter Conference on Applications of Computer Vision - 3rd Workshop on Real-World Surveillance, Applications and Challenges.

Bibliography

[1] Unity: Unity technologies. `https://unity.com/`.

[2] 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.

[3] X. Alameda-Pineda, J. Staiano, S. Ramanathan, L. M. Batrinca, E. Ricci, B. Lepri, O. Lanz, and N. Sebe. Salsa: A novel dataset for multimodal group behavior analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38:1707–1720, 2016.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[11] 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.

[12] L. Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[13] J. Ferryman and A. Shahrokni. Pets2009: Dataset and challenge. In *2009 Twelfth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 1–6, 2009.

[14] 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.

[15] R. Games:. Policy on posting copyrighted Rockstar Games material. `http://tinyurl.com/pjfoqo5`.

[16] W. Ge and R. T. Collins. Crowd detection with a multiview sampler. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision - ECCV 2010 - 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part V*.

[17] R. B. Girshick. Fast r-cnn. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[18] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[19] R. Gong, D. Dai, Y. Chen, W. Li, and L. Van Gool. mdalu: Multi-source domain adaptation and label unification with partial datasets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8876–8885, 2021.

[20] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[21] 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.

[22] 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.

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

[24] S. Jain, P. Yarlagadda, S. Jyoti, S. Karthik, R. Subramanian, and V. Gandhi. Vinet: Pushing the limits of visual modality for audio-visual saliency prediction. *arXiv preprint arXiv:2012.06170*, 2020.

[25] 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.

[26] 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.

[27] P. Kohl, A. Specker, A. Schumann, and J. 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.

[28] 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.

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

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

[31] T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[32] Z. L. Lin and L. S. Davis. A pose-invariant descriptor for human detection and segmentation. In *ECCV*, 2008.

[33] 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.

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

[35] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[36] A. Ouaknine. Review of deep learning algorithms for object detection. `https://medium.com/`, 2018.

[37] P. Peng, Y. Tian, Y. Wang, J. Li, and T. Huang. Robust multiple cameras pedestrian detection with multi-view bayesian network. *Pattern Recognit.*, 48:1760–1772, 2015.

[38] 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.

[39] 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.

[40] 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.

[41] E. Ristani, F. Solera, R. S. Zou, R. Cucchiara, and C. Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV Workshops*, 2016.

[42] 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.

[43] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015.

[44] L. Song, J. Wu, M. Yang, Q. Zhang, Y. Li, and J. Yuan. Stacked homography transformations for multi-view pedestrian detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6049–6057, 2021.

[45] 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.

[46] X. Sun and L. Zheng. Dissecting person re-identification from the viewpoint of viewpoint. In *CVPR*, 2019.

[47] Unity Technologies. Unity Perception package. `https://github.com/Unity-Technologies/com.unity.perception`, 2020.

[48] P. A. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57:137–154, 2004.

[49] 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.

[50] Y. Xu, X. Liu, L. Qin, and S.-C. Zhu. Cross-view people tracking by scene-centered spatio-temporal parsing. In *AAAI*, 2017.

[51] Y. Zhao, Z. Zhong, F. Yang, Z. Luo, Y. Lin, S. Li, and N. 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.

[52] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.