Learning Pedestrian Detection from Virtual Worlds

Giuseppe Amato[1], Luca Ciampi[1], Fabrizio Falchi[1], Claudio Gennaro[1], and Nicola Messina[1]

Institute of Information Science and Technologies (ISTI), Italian National Research Council (CNR), Via G. Moruzzi 1, 56124 Pisa, Italy
name.surname@isti.cnr.it

Abstract. In this paper, we present a real-time pedestrian detection system that has been trained using a virtual environment. This is a very popular topic of research having endless practical applications and recently, there was an increasing interest in deep learning architectures for performing such a task. However, the availability of large labeled datasets is a key point for an effective train of such algorithms. For this reason, in this work, we introduced *ViPeD*, a new synthetically generated set of images extracted from a realistic 3D video game where the labels can be automatically generated exploiting 2D pedestrian positions extracted from the graphics engine. We exploited this new synthetic dataset fine-tuning a state-of-the-art computationally efficient Convolutional Neural Network (CNN). A preliminary experimental evaluation, compared to the performance of other existing approaches trained on real-world images, shows encouraging results.

1 Introduction

Pedestrian detection remains a very popular topic of research having endless practical applications. An important application domain of this topic is certainly video surveillance for public security, such as crime prevention, identification of vandalism, etc. A real-time response in the case of an incident, however, requires manual observation of the video stream, which is in most cases economically not feasible.

We propose a real-time CNN-based solution that is able to localize pedestrian instances in images captured by smart cameras. CNNs are a popular choice for current objects detectors since they are able to automatically learn features characterizing the objects themselves; in the last years, these solutions outperformed approaches relying instead on hand-crafted features.

The great challenge we must address using CNNs is the ability of these algorithms to generalize to new scenarios having different characteristics, like different perspectives, illuminations and object scales. This is a must when we are dealing with smart devices that should be easily installed and deployed, without the need for an early tuning phase. Therefore, the availability of large labeled training datasets that cover as much as possible the differences between

various scenarios is a key point for training state-of-the-art CNNs. Although there are some large annotated generic datasets, such as ImageNet [1] and MS COCO [2], annotating the images is a very time-consuming operation, since it requires great human effort, and it is error-prone. Furthermore, sometimes it is also problematic to create a training/testing dataset with specific characteristics.

A possible solution to this problem is to create a suitable dataset collecting images from *virtual* world environments that mimics as much as possible all the characteristics of our target real-world scenario. In this paper, we introduce a new dataset named *ViPeD* (*Vi*rtual *Pe*destrian *D*ataset), a large collection of images taken from the highly photo-realistic video game *GTA V - Grand Theft Auto V* developed by *Rockstar North*, that extends the *JTA (Joint Track Auto)* dataset presented in [3]. We demonstrate that we can improve performance and achieve competitive results compared to the state-of-the-art approaches in the pedestrian detection task.

In particular, we train a state-of-the-art object detector, YOLOv3 [4], over the newly introduced ViPeD dataset. Then, we test the trained detector on the MOT17 detection dataset (*MOT17Det*) [5], a real-world dataset suited for pedestrian detection, in order to measure the generalization capabilities of the proposed solution with respect to real-world scenarios.

To summarize, in this work we propose a real-time CNN-based system able to detect pedestrians for surveillance smart cameras. We train the algorithm using a new dataset collected using images from a realistic video game and we take advantage of the graphics engine for extracting the annotations without any human intervention. Finally, we evaluate the proposed method on a real-world dataset demonstrating his effectiveness and robustness to other scenarios.

2 Related Work

In this section, we review the most important works in object and pedestrian detection. We also analyze previous studies on using synthetic datasets as training sets. Pedestrian detection is highly related to object detection. It deals with recognizing the specific class of pedestrians, usually walking in urban environments. Approaches for tackling the pedestrian detection problem are usually subdivided into two main research areas. The first class of detectors is based on handcrafted features, such as ICF (Integral Channel Features) [6–10]. Those methods can usually rely on higher computational efficiency, at the cost of lower accuracy. On the other hand, deep neural networks approaches have been explored. [11–14] proposed some modifications around the standard CNN network [15] in order to detect pedestrians, even accounting for different scales.

Many datasets are available for pedestrian detection. Caltech [16], MOT17Det [5], INRIA [17], and CityPersons [18] are among the most important ones. Since they were collected in different living scenarios, they are intrinsically very heterogeneous datasets. Some of them [16, 17] were specifically collected for detecting pedestrians in self-driving contexts. Our interest, however, is mostly concentrated on video-surveillance tasks and, in this scenario, the recently introduced

MOT17Det dataset has proved to be enough challenging due to the high variability of the video subsets. State-of-the-art results on this dataset are reached by [13]. With the need for huge amounts of labeled data, generated datasets have recently gained great interest. [19, 20] have studied the possibility of learning features from synthetic data, validating them on real scenarios. Unlike our work, however, they did not explore deep learning approaches. [21, 22] focused their attention on the possibility to perform domain adaptation in order to map virtual features onto real ones. Authors in [3] created a dataset taking images from the highly photo-realistic video game *GTA V* and demonstrated that it is possible to reach excellent results on tasks such as people tracking and pose estimation when validating on real data.

To the best of our knowledge, [23] and [24] are the works closest to our setup. In particular, [23] also used *GTA V* as the virtual world but, unlike our method, they used Faster-RCNN [25] and they concentrated on vehicle detection.

Instead, [24] used a synthetically generated dataset to train a simple convolutional network to detect objects belonging to various classes in a video. The convolutional network dealt only with the classification, while the detection of objects relied on a background subtraction algorithm based on Gaussian mixture models (GMMs). The real-world performance was evaluated on two common pedestrian detection datasets, and one of these (MOTChallenge 2015 [26]) is an older version of the dataset we used to carry out our experimentation.

3 The ViPeD Dataset

In this section, we describe the datasets exploited in this work. First, we introduce *ViPeD* -*V*irtual *Pe*destrian *D*ataset, a new virtual collection used for training the network. Then we outline *MOT17Det* [5], a real dataset employed for the evaluation of our proposed solution. Finally, we illustrate *CityPersons* [18], a real-world dataset for pedestrian detection we used as baseline. In order to show the validity of *ViPeD* , we have compared our network trained with CityPersons against the same network trained with *ViPeD* .

3.1 ViPeD - Virtual Pedestrian Dataset

As mentioned above, CNNs need large annotated datasets during the training phase in order to learn models robust to different scenarios, and creating the annotations is a very time-consuming operation that requires a great human effort.

The main contribution of this paper is the creation of *ViPeD* , a huge collection of images taken from the highly photo-realistic video game *GTA V* developed by *Rockstar North*. This newly introduced dataset extends the *JTA (Joint Track Auto)* dataset presented in [3]. Since we are dealing with images collected from a *virtual* world, we can extract pedestrian bounding boxes for free and without the manual human effort, exploiting 2D pedestrian positions extracted from the video card. The dataset includes a total of about 500K images, extracted

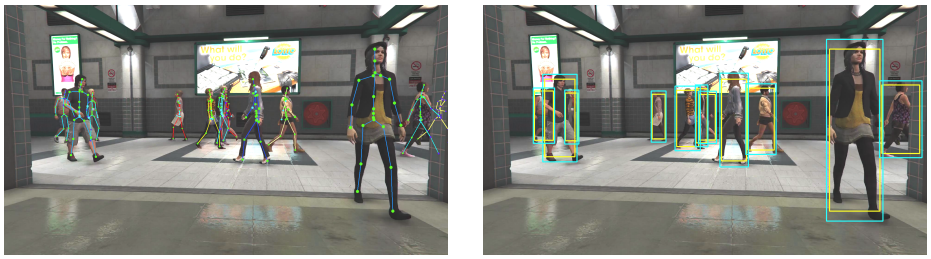
from 512 full-HD videos (256 for training and 256 for testing) of different urban scenarios.

In the following, we report some details on the construction of the bounding boxes and on the data augmentation procedure that we used to extend the *JTA* dataset for the pedestrian detection task.

A) Bounding Boxes: Since *JTA* is specifically designed for pedestrian pose estimation and tracking, the provided annotations are not directly suitable for the pedestrian detection task. In particular, the annotations included in *JTA* are related to the joints of the human skeletons present in the scene (Fig. 1a), while what we need for our task are the coordinates of the bounding boxes surrounding each pedestrian instance.

Bounding box estimation can be addressed using different approaches. The GTA graphic engine is not publicly available, so it is not easy to extract the detailed masks around each pedestrian instance; [23] overcame this issue by extracting semantic masks and separating the instances by exploiting depth information. Instead, our approach exploits the skeletons annotations already extracted by the *JTA* team in order to reconstruct the precise bounding boxes. This seems to be a more reliable solution than the depth separation approach, especially when instances are densely distributed, as in the case of crowded pedestrian scenarios.

The very basic setup consists of drawing the smallest bounding box that encloses all the skeleton joints. The main issue with this simple approach is that each bounding box perfectly contains the skeleton, but not the pedestrian mesh. Indeed, we can note that the mesh is always larger than the skeleton (Fig. 1b). We solved this problem by estimating a pad for the skeleton bounding box, exploiting another information produced by the GTA graphic engine and already present in *JTA*, i.e. the distance of all the pedestrians in the scene from the camera.



(a) (b)

Fig. 1: (a) Pedestrians in the *JTA* dataset with their skeletons. (b) Examples of annotations in the ViPeD dataset; original bounding boxes are in yellow, while the sanitized ones are in light blue.

In particular, the height of the i^{th} mesh, denoted as h_m^i, can be estimated from the height of the i^{th}skeleton h_s^i by means of the formula:

$$h_m^i = h_s^i + \frac{\alpha}{z^i} \tag{1}$$

where z^i is the distance of the i^{th} pedestrian center of mass from the camera, and α is a parameter that depends on the camera projection matrix.

Given that z^i is already available for every pedestrian, we estimated the parameter α by manually annotating 30 random pedestrians, obtaining for them the correct value for h_m^i, and then performing linear regression. We visually checked that the α parameter estimation was correct even for all the other non-manually annotated pedestrians.

We then estimated the mesh width w_m^i. Unlike the height, the width is strongly linked to the specific pedestrian pose, so it is difficult to be estimated with only the camera distance information. We decided to estimate w_m^i directly from h_m^i, assuming no changes in the aspect ratio for the original and adjusted bounding boxes:

$$w_m^i = h_m^i \frac{w_s^i}{h_s^i} = h_m^i r^i \tag{2}$$

where r^i is the aspect ratio of the i^{th} bounding box. Examples of final estimated bounding boxes are shown in Fig. 1b.

Finally, we performed a global analysis of these new annotations. As we can see in Fig. 2, in the dataset there are annotations of pedestrians farthest than 30-40 meters from the camera. However, we evaluated that humans annotators tend to avoid annotating objects farthest than this amount. We performed this analysis by measuring the height of the smallest bounding boxes in the human-annotated MOT17Det dataset [5] and catching out in our dataset at what distance from the camera the bounding boxes assume this human-limit size. Therefore, in order to obtain annotations comparable to real-world human-annotated ones, we decided to prune all the pedestrian annotations furthest than 40 meters from the camera.

From this point on, we will refer to the basic skeleton bounding boxes as *original* bounding boxes. Instead, we will refer to the bounding boxes processed by means of the previously described pipeline as *sanitized* (Fig. 1b).

B) Data Augmentation: Synthetic datasets should contain scenarios as close as possible to real-world ones. Even though images grabbed from the *GTA* game were already very realistic, we noticed some missing details. In particular, images grabbed from the game are very sharp, edges are very pronounced and common lens effects are missing. In light of this, we prepared a more realistic version of the original images.

We used *GIMP* image manipulation software, used in batch mode, in order to modify every image of the original dataset, using a set of different filters: radial blur, Gaussian blur, bloom effect, exposure/contrast. Parameters for these effects are randomly sampled from a uniform distribution.

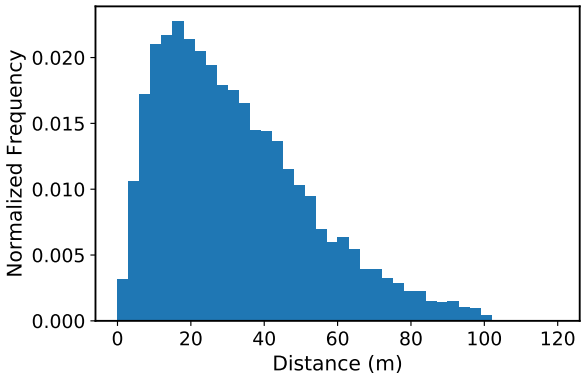


Fig. 2: Histogram of distances between pedestrians and cameras.

3.2 MOT17Det

We evaluate our solution using the recently introduced *MOT17Det* dataset [5], a collection of challenging images for pedestrian detection taken from 14 sequences with various crowded scenarios having different viewpoints, weather conditions, and camera motions. The annotations for all the sequences are generated by human annotators from scratch, following a specific protocol described in their paper. The training images are taken from sequences 2, 4, 5, 9, 10, 11 and 13 (for a total of 5,316 images), while test images are taken from the remaining sequences (for a total of 5,919 images). It should be noted that the authors released only the ground-truth annotations belonging to the training subset. The performance metrics concerning the test subset are instead available only submitting results to the *MOT17Det Challenge*[1].

3.3 CityPersons

In order to compare our solution trained using synthetic data against the same network trained with real images, we have also considered the *CityPersons* dataset [18], a recent collection of images of interest for the pedestrian detection community. It consists of a large and diverse set of stereo video sequences recorded in streets from different cities in Germany and neighboring countries. In particular, authors provide 5,000 images from 27 cities labeled with bounding boxes and divided across train/validation/test subsets.

4 Method

We use *YOLOv3* [4] as object detector architecture, exploiting the original Darknet [27] implementation. The architecture of YOLOv3 jointly performs a regression of the bounding box coordinates and classification for every proposed region.

[1] `https://motchallenge.net/data/MOT17Det/`

Unlike other techniques, YOLOv3 performs these tasks in an optimized fully-convolutional pipeline that takes pixels as input and outputs both the bounding boxes and their respective proposed categories. It is particularly robust to scale variance since it performs the detections at three different scales, down-sampling the input image by factors 32, 16 and 8.

As a starting point, we considered a model of YOLO pre-trained on the COCO dataset [2], a large dataset composed of images describing complex everyday scenes of common objects in their natural context, categorized in 80 different categories. Since this network is a generic objects detector, we then specialized it to recognize and localize object instances belonging to a specific category - i.e. the pedestrian category in our case.

Our goal is to evaluate the detector when it is trained with synthetic data. For this reason, we need to partially retrain the architecture to include new information deriving from a different domain.

In this particular work, domain adaptation between virtual and real scenarios is simply carried out by fine-tuning the pre-trained YOLOv3 architecture. In particular, we first extract the weights of the first 81 layers of the pre-trained model, since these layers capture universal features (like curves and edges) that are also relevant to our new problem. Then, we fine-tune YOLO initialing the firsts 81 layers with the previously extracted weights, and the weights associated with the remaining layers at random. In this way, we get the network to focus on learning the dataset-specific features in the last layers. All the weights are left unfrozen, so they can be adjusted by the back-propagation algorithm. With this technique, we are forcing the architecture to adjust the learned features to match those from the destination dataset.

5 Experimental Evaluation

We evaluate our solution in two different cases: first, in order to test the generalization capabilities, we train the detector using only our new synthetic dataset; then, in order to obtain best results on the MOT17Det dataset and compare them with the state-of-the-art, we evaluate detections after fine-tuning the detector also on the MOT17Det dataset itself.

Since the authors did not release the ground-truth annotations belonging to the test subset, we submitted our results to the MOT17Det Challenge in order to obtain the performance metrics. In order to prevent overfitting during the training in the second scenario, we create a validation split from the training subset considering a randomly chosen sequence. For the first scenario, instead, we validate on the full training set of MOT17Det.

Following other object detectors benchmarks, we use Precision, Recall and Average Precision (AP) as the performance metrics. A key parameter in all these metrics is the intersection-over-union threshold (IoU), which determines if a bounding box is matched to an annotation or not, i.e. if it is a true positive or a false positive.

Precision and Recall are defined as:

$$Precision = \frac{TPs}{TPs + FPs} \qquad Recall = \frac{TPs}{TPs + FNs} \tag{3}$$

where TPs are the True Positives, FPs the False Positives and FN the False Negatives. Average Precision is instead defined as the average of the maximum precisions at different recall values.

It is fairly common to observe detection algorithms compared under different thresholds, and there are often many variables and implementation details that differ between evaluation scripts which may affect results significantly. In this work, we consider only MOT17Det and COCO performance evaluators. We also use the standard IoU threshold value of 0.5.

Evaluation of the generalization capabilities Considering the first scenario, we first obtained a baseline using the original detector, i.e. the detector trained using the real-world general-purpose COCO dataset. Then, we trained the detector using our synthetic dataset, performing an ablation study over the introduced extensions.

First, we considered the original images and the original bounding boxes. Then, in order to evaluate how much the bounding-box construction policy can affect the detection quality, we considered the sanitized bounding boxes. Third, we considered also augmented images. Finally, we train the detector using the real-world dataset CityPersons, specific for the pedestrian detection task. We employ this experiment as a baseline over our ViPeD trained network. Results are reported in Table 1.

Comparison with the state-of-the-art on MOT17Det Concerning the second scenario, we obtained a baseline starting from the original detector trained with COCO and fine-tuning it with the training set of the MOT17Det dataset. Then, we considered our previous detector trained with ViPeD (the one with the sanitized bounding boxes and the augmented images) and we fine-tuned

Table 1: Results of YOLOv3 detector on MOT17Det

Training Dataset	**MOT AP**	**COCO AP**	**Precision**	**Recall**
COCO (Baseline)	0.69	0.41	87.4	72.4
CityPersons	0.58	0.37	69.0	60.5
ViPeD : Orig. BBs - Orig. Imgs	0.58	0.37	68.6	64.8
ViPeD : Sanitized BBs - Orig. Imgs	0.63	0.40	**91.1**	69.2
ViPeD : Sanitized BBs - Aug. Imgs	**0.71**	**0.48**	89.3	**73.9**

again the network with the training set of the MOT17Det dataset. Results are reported in Table 2, together with the ones obtained using the state-of-the-art approaches publicly released in the MOT17 Challenge (at the time of writing).

Table 2: Results on MOT17Det: comparison with the state-of-the-art

Method	**MOT AP**	**Precision**	**Recall**
YOLOv3 on COCO + MOT	0.80	89.9	82.8
YTLAB [13]	**0.89**	86.2	91.3
KDNT [28]	0.89	78.7	**92.1**
ZIZOM [29]	0.81	88.0	83.3
SDP [12]	0.81	**92.6**	83.5
YOLOv3 on ViPeD + MOT	0.80	90.2	84.6

Discussion Results in Table 1 show that we obtained best performances training the detector with ViPeD, using the sanitized bounding boxes and the augmented images, overtaking also the networks trained with COCO and with CityPersons. Therefore, our solution is able to generalize the knowledge learned from the virtual-world to a real-world dataset, and it is also able to perform better than the solutions trained using the real-world manual-annotated datasets.

Results in Table 2 demonstrate that our training procedure is able to reach competitive performance even when compared to specialized pedestrian detection approaches.

6 Conclusions

In this work, we propose a real-time system able to detect pedestrian instances in images. Our approach is based on a state-of-the-art fast detector, YOLOv3, trained with a synthetic dataset named *ViPeD*, a huge collection of images rendered out from the highly photo-realistic video game *GTA V* developed by *Rockstar North*.

The choice of training the network using synthetic data is motivated by the fact that a huge amount of different examples are needed in order for the algorithm to generalize well. This huge amount of data is typically manually collected and annotated by humans, but this procedure usually takes a lot of time and it is error-prone. We demonstrated that our solution is able to transfer the knowledge learned from the synthetic data to the real-world, outperforming the same approach trained instead on real-world manually-labeled datasets.

The YOLOv3 network is able to run on low-power devices, such as the NVIDIA Jetson TX2 board, at 4 FPS. In this way, it could be deployed directly on smart devices, such as smart security cameras or drones. Even if we trained YOLOv3 detector on the specific task of pedestrian detection, we think that the presented procedure could be applied at a larger scale even on other related tasks, such as object segmentation or image classification.

Acknowledgments

This work was partially supported by the AI4EU project, funded by the EC (H2020 - Contract n. 825619). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Jetson TX2 board used for this research.

References

1. J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 248–255.
2. T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014.
3. M. Fabbri, F. Lanzi, S. Calderara, A. Palazzi, R. Vezzani, and R. Cucchiara, "Learning to detect and track visible and occluded body joints in a virtual world," in *European Conference on Computer Vision (ECCV)*, 2018.
4. J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018.
5. A. Milan, L. Leal-Taixé, I. D. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," *CoRR*, vol. abs/1603.00831, 2016.
6. R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Ten years of pedestrian detection, what have we learned?" in *Computer Vision - ECCV 2014 Workshops*. Cham: Springer International Publishing, 2015, pp. 613–627.
7. S. Zhang, C. Bauckhage, and A. B. Cremers, "Informed haar-like features improve pedestrian detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.
8. S. Zhang, R. Benenson, and B. Schiele, "Filtered channel features for pedestrian detection," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1751–1760.
9. S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "How far are we from solving pedestrian detection?" in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
10. W. Nam, P. Dollar, and J. H. Han, "Local decorrelation for improved pedestrian detection," in *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, pp. 424–432.
11. Y. Tian, P. Luo, X. Wang, and X. Tang, "Deep learning strong parts for pedestrian detection," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 1904–1912.

12. F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers," in *2016 IEEE CVPR*, June 2016, pp. 2129–2137.
13. Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016, pp. 354–370.
14. P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. Lecun, "Pedestrian detection with unsupervised multi-stage feature learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.
15. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.
16. P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, April 2012.
17. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, June 2005, pp. 886–893 vol. 1.
18. S. Zhang, R. Benenson, and B. Schiele, "Citypersons: A diverse dataset for pedestrian detection," *CoRR*, vol. abs/1702.05693, 2017.
19. B. Kaneva, A. Torralba, and W. T. Freeman, "Evaluation of image features using a photorealistic virtual world," in *2011 International Conference on Computer Vision*, Nov 2011, pp. 2282–2289.
20. J. Marn, D. Vzquez, D. Gernimo, and A. M. Lpez, "Learning appearance in virtual scenarios for pedestrian detection," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 137–144.
21. D. Vazquez, A. M. Lopez, and D. Ponsa, "Unsupervised domain adaptation of virtual and real worlds for pedestrian detection," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Nov 2012, pp. 3492–3495.
22. D. Vzquez, A. M. Lpez, J. Marn, D. Ponsa, and D. Gernimo, "Virtual and real world adaptation for pedestrian detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 4, pp. 797–809, April 2014.
23. M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, and R. Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?" *CoRR*, vol. abs/1610.01983, 2016.
24. E. Bochinski, V. Eiselein, and T. Sikora, "Training a convolutional neural network for multi-class object detection using solely virtual world data," in *Advanced Video and Signal Based Surveillance (AVSS), 2016 13th IEEE International Conference on*. IEEE, 2016, pp. 278–285.
25. S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2015, pp. 91–99.
26. L. Leal-Taixé, A. Milan, I. D. Reid, S. Roth, and K. Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," *CoRR*, vol. abs/1504.01942, 2015.
27. J. Redmon, "Darknet: Open source neural networks in c," 2013.
28. F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "POI: multiple object tracking with high performance detection and appearance feature," *CoRR*, vol. abs/1610.06136, 2016.
29. C. Lin, J. Lu, G. Wang, and J. Zhou, "Graininess-aware deep feature learning for pedestrian detection," in *The European Conference on Computer Vision (ECCV)*, September 2018.