Training a Convolutional Neural Network for Multi-Class Object Detection Using Solely Virtual World Data

Erik Bochinski, Volker Eiselein and Thomas Sikora
Communication System Group, Technische Universität Berlin
Einsteinufer 17, 10587 Berlin
bochinski, eiselein, sikora@nue.tu-berlin.de

Abstract

Convolutional neural networks are a popular choice for current object detection and classification systems. Their performance improves constantly but for effective training, large, hand-labeled datasets are required. We address the problem of obtaining customized, yet large enough datasets for CNN training by synthesizing them in a virtual world, thus eliminating the need for tedious human interaction for ground truth creation. We developed a CNN-based multi-class detection system that was trained solely on virtual world data and achieves competitive results compared to state-of-the-art detection systems.

1. Introduction

The task of object detection in images and video is among the most fundamental ones in computer vision. There is a broad range of applications which are aided by these detection systems, like assisted driving [4, 20, 19], industrial applications [34, 36] or video surveillance [45, 11, 43]. The basic principle is to extract image features like HOG, SIFT, SURF [6, 30, 1] which are useful for a generic description of instances of specific object classes. They are then utilized to identify and localize those instances. For convolutional neural networks (CNNs) [27, 25], those feature extractors are learned directly from a training dataset. The great challenge is to find the common characteristics of these features to separate multiple object classes that are robust to different scales, rotations or pose, illumination and camera properties. Several methods have been proposed to solve this task [13, 8]. In the case of CNNs, these classifiers can be learned together with the respective features [29, 33] or using additional machine learning methods [16, 18, 47].

For object detection in video, spatio-temporal information can be utilized as well. In the case of static video sources which are typical for surveillance scenarios, efficient techniques like background subtraction [37, 9] can be utilized to find possible objects. To do so, no a priori knowledge about these objects is necessary [40, 35].

The availability of large labeled datasets is crucial for training state-of-the-art CNNs [25, 16, 38]. There are datasets available, like the ImageNet dataset [7] with over 14M images in over 21K categories or the Cityscapes dataset [5] with 30 categories and 5K fine/ 20K coarse grained pixel-wise annotated images. The first consists of all kinds of images gathered from the internet, the latter targets urban scene understanding for automotive applications. Creating the annotations requires great effort and there is always a certain susceptibility to errors if performed by humans. Automatic annotation methods are not generally applicable because every technique learned on this data could implicitly inherit the flaws of the method used to create it [26]. Sometimes it is also problematic to create a training/testing dataset with specific characteristics, like the presence of certain objects which are rare in real world environments or hard to control, e.g. animals. It is also not always possible to record data from specific views (aerial etc.) or with an arbitrary number of cameras.

A possible solution to these problems is the usage of synthetic datasets created in virtual world environments. Once a system to generate these synthetic datasets is made, almost unlimited data can be generated. Camera count and position are obviously no longer an issue. Rare events or object occurrences can be staged at any frequency. Most important, the ground truth can be generated with the used video engine and is most certainly complete and pixel accurate. Additionally, every scene can be rendered multiple times with different illumination and weather conditions. This expands the variety of the generated data.

In this paper we describe a system which is capable of generating such synthetic datasets and we use it for training a multi-class object detector based on background subtraction and a CNN. We show that it is possible to detect real world pedestrians, vehicles and animals in a reliable manner, even without the usage of any real world data at the training stage. The system shows competitive results

IEEE AVSS 2016, August 2016, Colorado Springs, CO, USA

to state-of-the-art object detectors on standard real world datasets.

2. Related Work

Current state-of-the-art methods are the deformable parts model (DPM) [13] or aggregate channel features (ACF) [8] detectors. For the DPM, different parts of an object and their positional relation are trained. The individual parts are not part of the training data, only bounding boxes of the whole object are used so that the latent model parts need to be learned as well. The ACF detector calculates various features on octave-spaced scale intervals and approximates them on additional scales via extrapolation. Thereby substantial computational resources can be saved while losing only a negligible amount of accuracy. These features are then used to detect objects with an AdaBoost [15] classifier.

Some systems using virtual world data have been deployed in the past. ObjectVideo virtual video (OVVV) [39] is a visual surveillance simulation test bed which allows one to test algorithms on real time camera data from a virtual world. It supports multi-view scenarios with different camera types (e.g. PTZ) and related distortions. Pedestrians can be controlled either by the user, scripts or AI. The authors report evaluations for tracking and background subtraction methods on this data. In [31], a pedestrian detector using HOG features and a linear SVM is developed. To train the detector, synthetic training data from a simulated car drive trough a virtual city is used. The authors also trained their detector on real world data and came to the conclusion that the performance of both versions is similar. It is shown in [44, 46] that a pedestrian detector trained on many virtual world and few real world samples achieves the same accuracy compared to a detector trained on real world data exclusively. In [23] different image similarity features were examined. The authors used photo realistic virtual world scenes in order to obtain the same images with different camera and lighting conditions.

The method presented in this paper is an extension of the aforementioned systems but uses solely virtual world training data and performs multi-class object detection using a CNN.

3. Dataset Synthesis

To generate scenes in a virtual world environment, the game *garry's mod* [12] is used. It is a sandbox game based on Valve's source engine [42] and enables the player to generate arbitrary scenarios or even game modes. To do so, the resources from installed games based on the source engine are available. This involves maps, characters, props and so on. There are also lots of user-made resources available for free [41].

The different object class instances in our system are implemented as non-player characters (NPCs), also called bots. These are characters which are controlled not by humans but by server scripts. Using the NextBot AI system [3], three types of bots were implemented: persons, vehicles and animals. They are initialized with random models (visual appearances) and can roam over the map or move on predefined tracks.

(a) Original (b) Bounding Box (c) Visible Mask

Figure 1. Illustration of the bounding boxes and pixel-accurate masks: (a) raw image, (b) object bounding box (green) computed using the model's part specific hitboxes (yellow), (c) visibility mask.

We implemented an add-on that allows to record and render arbitrary scenes as well as to create and control the NPCs in it. The ground truth is generated automatically and contains the bounding boxes and pixel-wise masks of all objects for each frame (see Figure 1).

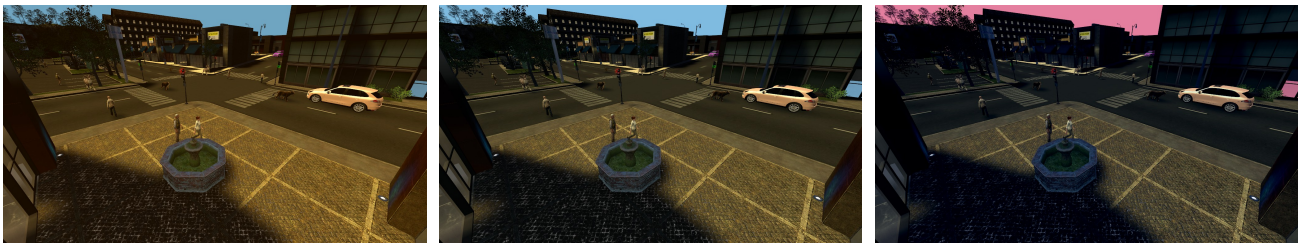
3.1. Dataset creation

A publicly available dataset[1] for training and testing was generated on two different maps with a total of 200 NPCs (100 each) and 15 static cameras in typical surveillance positions. Additionally, a set of negative frames without any objects of interest was generated. Instead of using static camera perspectives, we recorded a "flight" through the map and rendered the resulting images. Then, the training images were extracted according to the following rules:

1. the bounding box around the visibility mask of an object has an area of at least 35×35 pixel,
2. all object bounding box corners are inside the frame,
3. the overlap of both boxes is at least 50%.

The first constraint rejects objects which are too small. The second rule ensures that the object is completely in the field of view. The third rule discards all objects which are occluded by more than 50%. All other objects are cropped in

[1] available at: http://www.nue.tu-berlin.de/mocat



(a) Dawn (b) Day (c) Dusk

Figure 2. Sample images of the virtual world with different illumination settings.

(a) Animals (b) Pedestrians (c) Vehicles (d) Negatives

Figure 3. Examples of extracted training images

square patches with an extra border of 10%. The negative samples were cropped randomly from the negative frames with a size between 35×35 and 200×200. Examples are shown in Figure 3.

4. Proposed Object Detector

In our object detection system we use the common combination of background subtraction and subsequent classification [22]. The summary of the method is as follows: Firstly, the frames of an input video are segmented into foreground and background pixels. Using this information, potential objects are cropped from the frame and fed into a CNN for classification. The object class with the maximum activation on the output layer of the network is then assigned to the object while a minimal threshold t_c ensures a high confidence level.

This system is based on the assumption that the classes are non-overlapping, meaning that an object cannot have more than one class assigned. Typically only one or none of the outputs is greater than t_c.

4.1. Background Subtraction

Background subtraction is a popular tool in many video surveillance applications in order to improve the runtime of an analysis. A common choice is the use of gaussian mixture models (GMMs) [37]. The history of every pixel is described by K gaussian distributions and updated by an EM algorithm like approximation. If a new pixel value falls within 2.5 standard deviations of one of the K distributions it is considered background, otherwise foreground. Many improvements were made to this algorithm, like the splitting of over-dominant modes [10]. In our research we used the SGMM-SOD algorithm [9] which handles two GMMs in parallel. By a sophisticated initialization and updating scheme of both models it can be prevented that newly introduced objects fade into the background if becoming stationary. Therefore standing and waiting objects like pedestrians can be detected continuously. The mask of foreground pixels is then post-processed using morphological open and close operations.

4.2. Classification Using CNN

Over the last few years, CNNs attracted significant attention in the field of image classification. They show a great ability to find and utilize common class specific characteristics of images if given a large enough training dataset. CNNs were introduced first to recognize handwritten digits [27] in the early 90s, but a major breakthrough was achieved 2012 with the release of the AlexNet [25]. The basic principle can be understood as a special case of the common multilayer perceptron (MLP) where every neuron is only connected to a receptive field in front of it. Additionally, all neurons of a specific layer share the same weights. The weighted input of a neuron with N inputs before applying the activation function is:

$$v = \sum_{i=1}^{N} a_i w_i \tag{1}$$

where a, w denotes the input from the previous layer and weights respectively. With a two-dimensional constrained receptive field and weight sharing, the formula can be adopted for v at position x, y as follows:

$$v_{x,y} = \sum_{i=x-\frac{N}{2}}^{x+\frac{N}{2}} \sum_{j=y-\frac{N}{2}}^{y+\frac{N}{2}} w_{i,j} a_{x-i,y-j} \tag{2}$$

which can be implemented as the name-giving discrete, two-dimensional convolution $v = a * w$. To reduce the amount of data and providing translation invariance, subsampling layers are used. Most common is the max-pooling, where the maximum value of the pooling area is chosen.

Layer	Dimension
Conv 1	$24 \times 6 \times 6$
Pool 1	3×3
Conv 2	$64 \times 3 \times 3$
Pool 2	2×2
Conv 3	$96 \times 3 \times 3$
Conv 4	$64 \times 2 \times 2$
Pool 3	2×2
FC 1	256
FC 2	256
FC 3	4

Table 1. Net structure

Our net consists of 4 convolutional, 3 pooling and 3 fully connected (FC) layers as shown in Table 1. It was trained using the caffe framework [21]. The input dimension is 40×40 pixel, random crops were taken from 48×48 pixel sized training images. For further data augmentation, random mirroring was done. The stochastic gradient descent solver was used, the weights were initialized gaussian with $\sigma = 0.01, \mu = 0$. The training was done over 3 epochs with a momentum of 0.9 and a base learn rate of 0.01, decreased by a factor of 10 after each epoch. To prevent overfitting, 50% dropout was used at FC1 and FC2.

4.3. Improvements

Bounding Box Splitting Since the CNN is only used to classify the detected object candidates, it is crucial that they describe individual objects separately. It is impossible to distinguish between multiple visually overlapping objects in the foreground mask, which may occur when people are moving in groups. As a remedy, in our system bounding boxes with atypical aspect ratios are split into smaller subboxes with typical aspect ratios of e.g. pedestrians and the classification is performed on these.

Temporal Filtering All objects with distinct overlapping bounding boxes in consecutive frames are considered identical. Thus, the maximum activation and the class assignment can be inherited from other frames if the current classification results are below the classification threshold t_c. This allows an increase in t_c which results in a better false positive rate. Also, "gaps" in the detection of the same object over time are avoided, lowering the false negative rate as well. To determine identical objects over time, the intersection over union (IOU) is used:

$$0.2 \leq \frac{B_i^t \cap B_{i'}^{t+1}}{B_i^t \cup B_{i'}^{t+1}} \tag{3}$$

where B_i^t describes the bounding box of the i^{th} object at the t^{th} frame. If the threshold of 0.2 is met for exactly one bounding box pair (i, i') it is assumed that they both define the same object.

These two improvements reduce greatly the occurrence of false negatives as presented in the following experiments.

Figure 4. Sample of ground truth (yellow) vs. detection results (green). The detection bounding boxes represent the persons accurately but show poor matching results to the ground truth.

5. Experiments

In order to assess the detection performance of the proposed method, experiments have been conducted on both virtual and well-known real world datasets. The presented results are computed with the development kit of the MOT challenge [28]. The pedestrian detection ground truth for the AVG-TownCentre sequence is taken from [2], for PETS09-S2L1 it is included in the development kit. We noticed a slight displacement noise in the ground truth (examples shown in Figure 4) which is probably due to its partial generation using interpolation techniques (PETS09-S2L1) and the simple position estimation based on the head positions (AVG-TownCentre). We therefore propose to use a different IOU threshold $t_{IOU} = 0.2$ instead of $t_{IOU} = 0.5$ for computing the correct matches in the evaluation. This parameter has also been used in the well-known CLEAR performance evaluation protocol [24] and effectively allows matches with inaccurate ground truth data. However, in order to facilitate a comparison with other methods, we show the results for both values in the evaluation.

5.1. Results on Virtual World Data

Table 2 shows the detection results on virtual world data for multiple object classes. We conducted our experiments on the same sequences for three different illumination settings: dawn, day and dusk as shown in Figure 2. The performance in dawn and day conditions are similarly good while during dusk a lot more false positives are generated. A manual inspection of the sequences suggests that they are generally too dark for the background subtractor and CNN to work properly.

In all cases, the reduced t_{IOU} enhances the results due to deviations in the region proposals of the background subtraction. The shadow detection seems less effective than on real world data, thus rendering the extracted regions bigger. Also, during processing, the legs of pedestrians and animals tend to be removed by morphological operations, again changing the size of the bounding boxes. Hence, a more relaxed t_{IOU} improves the results in these cases.

	Class	FN	FP	Rec	Prec	MODA	MODP	FN	FP	Rec	Prec	MODA	MODP
		$t_{IOU} = 0.5$						$t_{IOU} = 0.2$					
dawn	Person	12058	4256	54.8	77.5	38.9	83.8	10945	3143	59.0	83.4	47.2	59.1
	Vehicle	10805	8932	82.3	84.9	67.6	87.5	7833	5960	87.1	89.9	77.3	69.9
	Animal	6199	2165	64.8	84.1	52.5	84.1	5735	1701	67.4	87.5	57.8	61.6
	All	29062	15353	72.4	83.2	57.8	86.3	24513	10804	76.7	88.2	66.4	66.6
day	Person	10126	4676	62.0	77.9	44.4	84.2	9256	3806	65.3	82.0	51.0	61.5
	Vehicle	11705	8824	80.8	84.8	66.3	87.6	8779	5898	85.6	89.8	75.9	70.6
	Animal	7270	1847	58.8	84.9	48.4	81.4	6620	1197	62.5	90.2	55.8	51.0
	All	29101	15347	72.3	83.2	57.8	86.0	24655	10901	76.6	88.1	66.2	65.9
dusk	Person	13093	11934	50.5	52.8	5.3	72.2	9535	8376	63.9	66.9	32.3	21.7
	Vehicle	15922	9856	73.6	81.9	57.3	82.7	12146	6080	79.9	88.8	69.8	54.9
	Animal	12919	3505	27.0	57.7	7.2	71.2	11213	1799	36.7	78.3	26.5	17.6
	All	41934	25295	59.9	71.2	35.7	79.6	32894	16255	68.5	81.5	53.0	43.7

Table 2. Class specific results on virtual world data for the same scenes with different illumination settings.

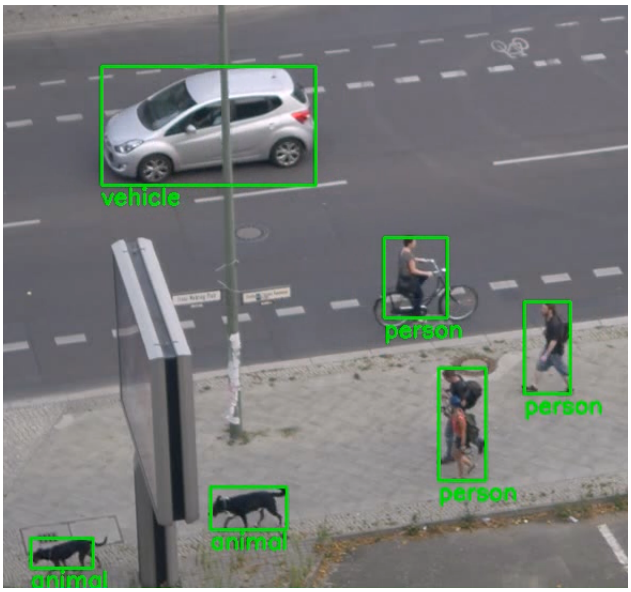


Figure 5. Example of a multi-class detection on real world data. The morphological operations ensure that the vehicle is detected as a whole and not split by the lamppost. The people in front can not be distinguished because they form a singular moving block, although they are detected as a person. The person on the bike shows that people in unusual poses that are not covered by the training data can be detected as well.

5.2. Results on Real World Data

The real-world performance was evaluated on two common pedestrian detection datasets. Table 3 shows the comparison between our approach and the state-of-the-art methods DPM [13] (version 5 from [17] and pretrained on the INRIA dataset [6]) and ACF [8] (results taken directly from MOT 2015 challenge).

For the PETS09-S2L1 sequence with $t_{IOU} = 0.5$, our approach shows the lowest false positive count and lies in between the reference methods with regard to precision. However, the false negative count is higher and the recall lower. This can be explained by the ground truth properties, as mentioned before, and changes drastically with a reduced $t_{IOU} = 0.2$. The reduced threshold improves also the reference methods but especially in the false positive rate, the proposed system shows a favorable performance (up to 8-21 times fewer false positives compared to DPM and ACF). The MODA measure (which includes false positive and false negative detections) for our baseline method shows a performance between the two reference methods.

The bounding box splitting and temporal filtering steps especially reduce the false negative count. When both are applied, the false negative count is reduced by over 35% without a significant computational overhead. The bounding box splitting introduces additional false positives, mainly because single pedestrians might also be split and both splits are positively classified. The effect of these few false positives is less important compared to the improvement as can be seen by looking at the MODA values.

The AVG-TownCentre sequence poses additional problems as many overlapping detections can be found. This reduces the performance for all methods considerably compared to PETS. The main findings, however, are similar to the PETS evaluation. The proposed system shows a reduced number of false positives and a higher number of false negatives which can be reduced using bounding box splits and temporal filtering. The MODA metric in case of $t_{IOU} = 0.2$ shows that the proposed system performs better than ACF but worse than DPM.

For a more complete comparison we exchanged our CNN, that was trained solely on the virtual world data with the BVLC reference implementation of the R-CNN that ships with the caffe framework. We see the lowest rate for false positives for all experiments at the expense of the highest false negative counts. Indeed, these results are biased because the R-CNN is a magnitude greater with regard to net structure, parameter size and computational effort. Therefore its classification performance can not be compared to our net directly.

	Method	FN	FP	Rec	Prec	MODA	MODP	FN	FP	Rec	Prec	MODA	MODP
		$t_{IOU} = 0.5$						$t_{IOU} = 0.2$					
PETS09-S2L1 [14]	DPM [13]	756	639	83.1	85.3	**68.8**	**75.9**	628	511	86.0	88.3	74.6	**32.3**
	ACF [8]	**392**	1494	**91.2**	73.2	57.9	71.7	**162**	1264	**96.4**	77.3	68.1	15.6
	ours	1734	629	61.3	81.3	47.2	75.0	1165	**60**	74.0	98.2	72.6	24.6
	ours + BS	1308	521	70.8	85.9	59.1	74.8	935	148	79.1	96.0	75.8	25.6
	ours + TF	1613	664	64.0	81.2	49.1	74.9	1011	62	77.4	98.2	76.0	24.2
	ours + BS + TF	1125	557	74.9	85.7	62.4	74.7	729	161	83.7	95.9	**80.1**	25.3
	ours + BS + TF + BVLC R-CNN	1217	**357**	72.8	**90.1**	64.8	75.1	923	63	79.4	**98.3**	78.0	27.0
AVG-TownCentre [2]	DPM [13]	**1873**	1312	**73.8**	80.1	**55.4**	**72.4**	**1583**	1022	**77.8**	84.5	**63.5**	**18.8**
	ACF [8]	3382	1649	52.7	69.5	29.6	71.2	2914	1181	59.2	78.2	42.7	12.6
	ours	4237	1756	40.7	62.3	16.1	67.7	3040	559	57.4	88.0	49.6	6.7
	ours + BS	4064	1956	43.1	61.2	15.7	67.5	2901	793	59.4	84.3	48.3	6.7
	ours + TF	4161	1861	41.7	61.6	15.7	67.7	2938	638	58.9	86.8	49.9	6.7
	ours + BS + TF	3978	2021	44.3	61.0	16.0	67.5	2801	844	60.8	83.7	49.0	6.7
	ours + BS + TF + BVLC R-CNN	4809	**450**	32.7	**83.8**	26.4	68.7	4372	**13**	38.8	**99.5**	38.6	9.2

Table 3. Results on real world data for two common datasets. DPM and ACF are state-of-the-art detectors for reference. BS denotes the bounding box split improvement, TF the temporal filtering. BVLC R-CNN denotes the usage of the R-CNN reference model shipping with caffe instead of our CNN trained solely on the virtual world data.

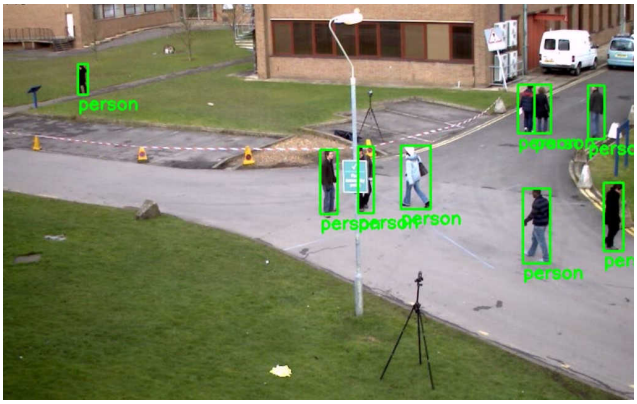


Figure 6. Sample detections on the PETS09-S2L1 sequence. Note that the two people in the back are detected as a singular block but are split correctly. The person behind the lamppost, although partly covered, is detected completely.

Exemplary detection results for the proposed system are shown in Figure 6. Due to the unavailability of related detection videos for the multi-class detection, no numerical results can be given, but samples are shown in Figure 5.

Table 4 shows the achieved detection speeds of the proposed system. For 768×576 pixel resolution, 13.5 Hz can be obtained and for 1080p videos still 3.4 Hz are possible. A great share of the detection time is consumed by the SGMM-SOD background subtraction algorithm which handles two background models in parallel. The classification is mainly performed on the GPU and the background subtraction on the CPU. Thus, a pipelined evaluation of both steps could improve the performance to a factor of ~ 2.

Seq.	Resolution	#Fr.	Method	Time	Hz
PETS	768×576	795	SGMM-SOD	30s	26.5
			Main	59s	13.5
			R-CNN	3m43s	3.5
AVG	1920×1080	450	SGMM-SOD	81s	5.5
			Main	1m43s	3.4
			R-CNN	5m32s	1.6

Table 4. Performance overview for the test sequences: SGMM-SOD shows the run-time for background subtraction only. Main denotes the runtime with our CNN, R-CNN in conjunction with the BVLC reference model of the R-CNN.

6. Conclusion

In this paper we proposed a system for automated dataset generation for object detection algorithms using virtual world data and showed that a CNN-based approach trained on this data can achieve similar results to state-of-the-art detection methods trained on real-world images. The automated annotation system avoids the previously tedious task of manual ground truth creation and can be extended easily to multiple object classes. The work shows that a CNN-based object classifier can perform competitively in the detection of multiple object classes like pedestrians, vehicles or animals without using any real-world training samples. This contributes to solving the need of evermore labeled training data to train new and bigger classification methods.

7. Acknowledgement

The research leading to these results has received funding from the European Community's FP7 under grant agreement number 607480 (LASIE).

References

[1] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Computer vision–ECCV 2006*, pages 404–417. Springer, 2006.

[2] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3457–3464. IEEE, 2011.

[3] M. Booth. The ai systems of left 4 dead. In *Keynote, Fifth Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE09)*, 2009.

[4] H. Cho, Y.-W. Seo, B. Vijaya Kumar, and R. R. Rajkumar. A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 1836–1843. IEEE, 2014.

[5] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops*, volume 3, 2015.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[8] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(8):1532–1545, 2014.

[9] R. H. Evangelio, M. Patzold, I. Keller, and T. Sikora. Adaptively splitted gmm with feedback improvement for the task of background subtraction. *Information Forensics and Security, IEEE Transactions on*, 9(5):863–874, 2014.

[10] R. H. Evangelio, M. Pätzold, and T. Sikora. Splitting gaussians in mixture models. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 300–305. IEEE, 2012.

[11] M. Evans, C. J. Osborne, and J. Ferryman. Multicamera object detection and tracking with object size estimation. In *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*, pages 177–182. IEEE, 2013.

[12] Facepunch Studios. Garrys mod. http://www.garrysmod.com.

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

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

[15] J. Friedman, T. Hastie, R. Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[17] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. http://people.cs.uchicago.edu/ rbg/latent-release5/.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(9):1904–1916, 2015.

[19] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–8. IEEE, 2013.

[20] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, R. Cheng-Yue, F. Mujica, A. Coates, et al. An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*, 2015.

[21] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[22] S. Johnsen and A. Tews. Real-time object tracking and classification using a static camera. In *Proceedings of IEEE International Conference on Robotics and Automation, workshop on People Detection and Tracking*. Citeseer, 2009.

[23] B. Kaneva, A. Torralba, and W. T. Freeman. Evaluation of image features using a photorealistic virtual world. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2282–2289. IEEE, 2011.

[24] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, M. Boonstra, and V. Korzhova. Performance evaluation protocol for face, person and vehicle detection & tracking in video analysis and content extraction (vace-ii). *Computer Science & Engineering University of South Florida, Tampa*, 2006.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[26] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[27] B. B. Le Cun, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*. Citeseer, 1990.

[28] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler. MOTChallenge 2015: Towards a benchmark for multi-target tracking. *arXiv:1504.01942 [cs]*, Apr. 2015. arXiv: 1504.01942.

[29] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[30] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[31] J. Marin, D. Vázquez, D. Gerónimo, and A. M. López. Learning appearance in virtual scenarios for pedestrian detection. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 137–144. IEEE, 2010.

[32] Rockstar Games, Inc. Grand theft auto v. http://www.rockstargames.com/V/.

[33] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[34] A. Sharma, R. P. P. Singh, and P. Lehana. Evaluation of the accuracy of genetic algorithms for object detection in industrial environment. *International Journal of Scientific and Technical Advancements*, 1(3):105–111, 2015.

[35] A. Sobral and A. Vacavant. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding*, 122:4–21, 2014.

[36] N. Somani, A. Perzylo, C. Cai, M. Rickert, and A. Knoll. Object detection using boundary representations of primitive shapes. In *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, China*, 2015.

[37] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[39] G. R. Taylor, A. J. Chosak, and P. C. Brewer. Ovvv: Using virtual worlds to design and evaluate surveillance systems. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[40] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 255–261. IEEE, 1999.

[41] Valve Cooperation. Steam workshop. https://steamcommunity.com/app/4000/workshop/.

[42] Valve Cooperation. Valve developer community. https://developer.valvesoftware.com.

[43] D. Varga, L. Havasi, and T. Szirányi. Pedestrian detection in surveillance videos based on cs-lbp feature. In *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2015 International Conference on*, pages 413–417. IEEE, 2015.

[44] D. Vazquez, A. M. Lopez, J. Marin, D. Ponsa, and D. Geronimo. Virtual and real world adaptation for pedestrian detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(4):797–809, 2014.

[45] X. Wang, M. Wang, and W. Li. Scene-specific pedestrian detection for static video surveillance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(2):361–374, 2014.

[46] J. Xu, D. Vazquez, A. M. Lopez, J. Marin, and D. Ponsa. Learning a part-based pedestrian detector in a virtual world. *Intelligent Transportation Systems, IEEE Transactions on*, 15(5):2121–2131, 2014.

[47] W. Y. Zou, X. Wang, M. Sun, and Y. Lin. Generic object detection with dense neural patterns and regionlets. *arXiv preprint arXiv:1404.4316*, 2014.