Crane safety system with monocular and controlled zoom cameras

A. Vierling, T. Sutjaritvorakul and K. Berns

Technische Universität Kaiserslautern, Germany
E-mail: {vierling,tanittha,berns}@cs.uni-kl.de

Abstract -

In this paper, we propose an approach and workflow in order to detect humans in the environment around a crane with Monocular Images. The considered area is split up into a zone around the crane truck and one around the load. The load will be monitored with an optical zoom camera where we can control the zoom. We discretize the zoom levels and a Convolutional Neural Network for each zoom level is trained. Afterwards a Meta Convolutional Neural Network is trained in order to select the next zoom level. Since there are no public datasets available for this kind of task we propose to generate the needed data with a photorealistic simulation.

Keywords -

Crane; Safety; Human Detection; Deep Learning; CNN; Simulation

1 Introduction

In the construction site environment each human is exposed to many potentially dangerous objects. Common sources for injuries are heavy-duty commercial vehicles. The distribution of accidents in the example of an excavator can be seen in figure 1. As one can see most accidents happen in an area which is not visible to the operator, i.e., behind the vehicle or at the opposite site of the operator.

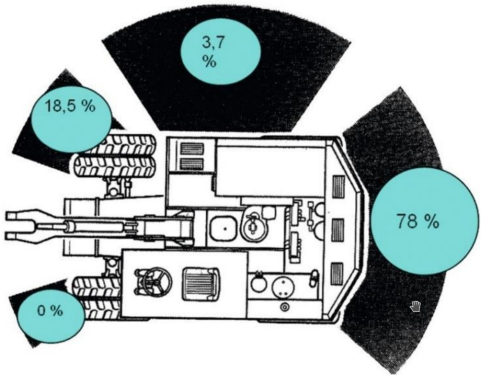


Figure 1. Distribution of fatal and critical accidents with excavators shown in the blue circles[14].

It is, therefore, necessary to support the operator of heavy-duty machines when the area around it is not closed off and humans can appear unexpectedly. Though the problem gets even worse with bigger machines where even bigger areas are occluded by the machine. Therefore there is a need to detect humans in the potential risk areas of a crane, which is definitely one of the biggest machines in a construction site environment. The work presented in this paper is focused on mobile cranes and caterpillar cranes, so specifically tower cranes are not considered. Universal usability for cranes of different manufactures is nonetheless important. We therefore do not make us of the internal crane data. When operating a crane there are two main areas which include a potential threat to humans, one is the crane truck itself and the other is the load of the crane. It is necessary to consider both parts for a reasonable approach.

We therefore propose a methodology which can be used to detect humans around the crane. It mostly relies on already tested *Deep Learning* approaches for object detection. Our methodology does not only detect humans around the crane but also around the load, which is more complicated due to some further restrictions. Additional challenging elements are due to the construction site environment. In road scenarios one can assume that the vehicles are in either a highway, city, urban or metropolitan area, while a construction machinery can operate in a mining field, building site, forest or even rough terrain. The environment effects type and activity of each object that the system is going to encounter during the detection task. One further handicap is that standard approaches for detecting humans around the load are not suitable. Since the shape and material of the load is not known in advance and the load can potentially be moved in a big area, which is also not necessarily planned beforehand. The only remaining possibility is to mount cameras on the crane itself. Because of the sheer size of the crane, off the shelf solutions for detecting humans are not applicable due to unusual perspectives and distances .

In the following sections we describe our approach in detail. The rest of the paper is organized as follows: Section 2 discusses the state-of-the-art of human detection algorithms split it up into the different perspectives which are used in the proposed approach. In section 3 the general

structure of the proposed approach is described and again split up in the different zones of interest. Section 4 is considered with the generation of the needed datasets with a simulation tool. Section 5 takes a look at the network architectures chosen in section 3 to solve the detection tasks. Finally in section 6 we recap the proposed methodology.

2 Related work

In construction machines, safety systems mostly conduct without any automatic brake when possibility of a collision occurs. The system simply alarms the operator to brake manually. In some systems, the operator needs to acknowledge the alarm by e.g. touching the monitor screen. Traditionally, safety in construction sites is done by direct manual observation [17]. It mainly uses non vision-based sensors, e.g., RFID tag, ultrasonic or infrared sensors. The sensors can be attached on the workers' helmets or wristbands and inform the crane operator about their position. Due to the characteristics of proximity sensors, these safety systems are unable to predict a person's trajectory and absolute position due to an insufficient detection range. Most vision-based safety systems take advantage of specific cues of a persons appearance. In the construction site environment many such visual features can be used, e.g., hard helmet detection [26], detecting workers from reflective vest [19]. Even though dress code in construction site is strict, some people could unintentionally ignore it or do not realize the risk this poses.

2.1 Crane Truck View

For the detection of humans around the crane most state of the art Convolutional Neural Networks(CNNs) can be used as a basis. The perspectives are quite similar, even though the appearance of most humans is a lot more defined due to the presence of reflective vests and helmets. As a basis we refer for example to [24, 15, 23], which of course does not cover the whole extend of work in the field of human detection.

2.2 Load View

Most Detection algorithms for top view camera systems can be found in surveillance systems. Many different proposed methodologies have been proposed, however, these are not really suited to our needs. These approaches often times make use of the fact that either the camera itself is static [31, 33] or uses distinct features of the background, like a street [3]. Some methodologies use features which are prevalent even in top view images, like the head-shoulder shape [34], which may be different due to helmets or carried objects in an construction site environment. Some also simply try to estimate the number of persons in an image [13, 6]. Most are quite often not build to deal with images from a distance as far as in a crane scenario [22, 6, 30].

Another approach is to use a human detection algorithm which works for the front view case and adapt it in order to deal with different views and scenes [32, 1]. We do not only have to deal with the fact, that there is no fitting approach for our use-case but also no public available dataset. Recent advancements in the usage of simulated data for machine learning have been made in reinforcement learning [18], object detection via CNNs [21], viewpoint estimation via CNNs [20] and segmentation tasks [25]. Hence, there are many possibilities to tackle the introduced task, but additionally to the so far presented state-of-the-art it may be necessary to use domain adaptation techniques in order to improve the results as in [4, 16, 29, 35, 2]

3 Proposed Approach

The application seems to encourage an approach which is separated into the detection around the crane and around the load. In our approach we will follow this idea and deal with both problems in a separate manner. We start with the setup of the sensor system in each case. Afterwards we shortly talk about the used CNNs and how to train them specifically for our application scenario.

3.1 Human Detection Crane Truck

In this subpart of our Application scenario we use a camera setup which roughly looks like presented in figure 2.

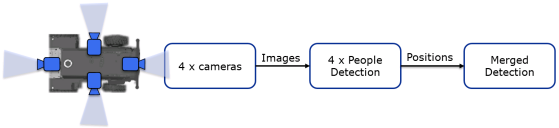
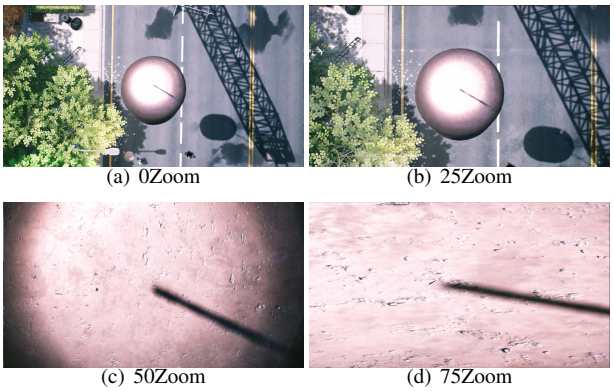


Figure 2. Sketch of camera setup at the crane truck

We have a couple of cameras mounted on the crane truck. They look into different directions in order to have the possibility to detect humans in the blind spot of the operator. In the crane scenario the blind spots are not static, due to the revolving superstructure. We are then able to work on each camera image individually and then merge the results. In order to detect humans in each of the images we use a Neural Network. Like already elaborated in section 2, CNNs yield solid performances when dealing with human detection for such frontal views. As for the explicit architecture of the used CNN we refer to [15], since we do not adapt the actual structure to much, still some additional work is needed. Off the shelf trained networks will not yield perfect results since the construction site environment differs from most available train-

ing data. Going into a construction site environment and collecting enough labeled data of typical workers to train such a network from scratch is very time consuming. We therefore use a pretrained Network as initialization. Afterwards we apply transfer learning with some examples from construction site environment workers. We also will use simulated data since we will need to generate it anyway, as explained further below. The workflow will be similar to figure 4, but keep in mind that we start with a pretrained network and do not need to train from scratch. With this in hand we have all the necessary parts to combine it into a human detection algorithm around the crane truck.



(a) 0Zoom (b) 25Zoom (c) 50Zoom (d) 75Zoom

Figure 3. View of different levels of zoom

3.2 Human Detection Load

The detection of humans around the load is more involved. The first problem is how and where to mount the camera system. The possibility of putting, e.g., cameras on tripods on the ground and then merging the images is not feasible due to the fact that the exact trajectory the load will take is often times adapted during the hoisting process. Therefore constant rearrangement of the tripods would be needed, which is not feasible. Another possibility would be to mount a mobile camera below the load, which is only practical with a wireless connection. This is also not feasible due to most loads containing lots of metal parts and therefore greatly interfering with the data transmission. Additionally one needs to consider the fact that a complete hoisting cycle may take up to several hours, which further solidifies the need for a wired connection because of the battery capacity of wireless cameras. So the only possibility which is left is to mount a camera on the crane itself. Of course many cameras could be mounted on the crane boom. We selected the boom top as a first choice, since for other locations at the boom additional need for an alignment with the load is present. This is not the case for the boom top due to the load being approximately perpendicular to the ground.

There are however some other problems arising from mounting the camera on the boom top. A crane can easily reach a height of $100m$, but the boom top may also be close to the ground depending on the hoisting process. So we have to deal with many different perspectives and distances to the ground. This greatly affects the size of the humans in the image. Therefore there is a reasonable necessity for a camera with controllable optical zoom in order to deal with these differing camera heights.

If however one has such a controllable zoom additional effort needs to be made in order to chose the zoom value accordingly. Of course a high zoom value is desirable since then the size of a human in the images is bigger. On the other hand a low zoom value gives us a wider view of the scene. Specifically for the crane scenario an additional limitation is the occlusion due to the position of the load. This occlusion will always be present, so we need to try and optimize the occlusion by adjusting the zoom level accordingly. The optimal zoom level is not only dependent on the pose of the boom but also on the position of the load, see figure 3. We therefore propose the following approach. We split up the zoom levels in discrete parts, for simplicity we here take the zoom levels of $0\%, 25\%, 50\%, 75\%$. For each of the zoom levels we want to individually train a CNN, called $Zoom0$, $Zoom25$, $Zoom50$, $Zoom75$. Each of the networks take images as inputs and we choose the same architecture for each of the networks. Training for each network will be done with its one appropriate dataset. We so to speak need a $DatasetZoom0$, $DatasetZoom25$, $DatasetZoom50$ and $DatasetZoom75$. Since such datasets are not publicly available we follow the workflow proposed in figure 4 for each network individually.

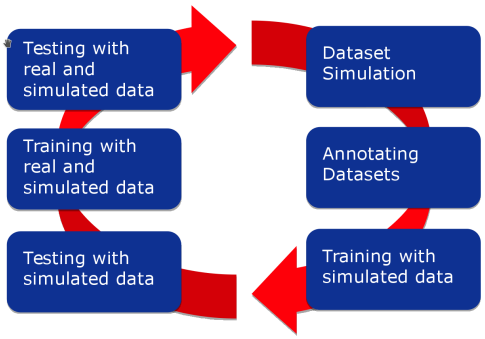


Figure 4. Proposed workflow for each individual network

Assuming this training is finished we take an additional network which we will call MetaNetwork, which differs in its architecture. See section 5 for a closer look at the architecture and the training process. This MetaNetwork

will take the current image and the corresponding zoom level as input. Its output is then the proposed number of human each of the four Networks will detect. The idea here is, that the MetaNetwork can, e.g., due to a uniform texture, see figure 3, detect when the zoom level is too high and we have a lot of occlusion. On the other hand the MetaNetwork may notice that only a small amount of occlusion is present, but we are far away from the ground and therefore may say that a bigger zoom level may detect more pedestrians due to bigger humans. Then a deciding algorithm should determine what the appropriate zoom level is and also select the network accordingly. The whole architecture is presented in figure 5.

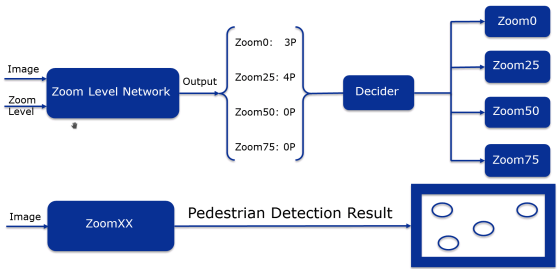


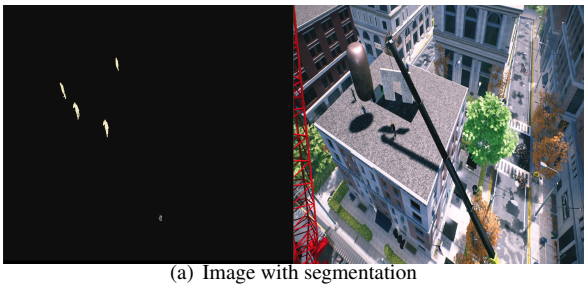
Figure 5. Complete architecture

In its easiest form such a Decider could just take the network with the highest proposed number of detected humans. Also more sophisticated versions are possible, e.g., only switch after a certain amount of time has past to prevent constant switching or do not directly switch between *Zoom*0 and *Zoom*75. If all this is put into place we have an algorithm which gives an educated guess which zoom level to choose and then puts a detection network in charge which is specialized to exactly this zoom level.

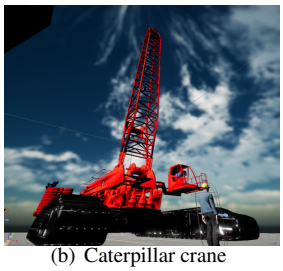
4 Dataset Generation

A common problem for Machine Learning in the context of commercial vehicles is the very small number of public datasets. A huge volume of image data is needed. The common datasets and benchmarks which are used in learning process of classification or object detection tasks, e.g., KITTI [12], Daimler [8, 11, 27], Caltech [7], INRIA [5], PASCAL [10], ETHZ [9] are not really applicable to our situation. So in order to generate the needed data we exploit a simulation tool, Unreal Engine 4. The idea of the approach is similar to [28]. Instead of training the human detector using only real-world image data. The Unreal Engine can produce realistic images of a construction site environment e.g different workers, different kinds of weather conditions such as snow, rain, cloud. For such effects see figure 6.

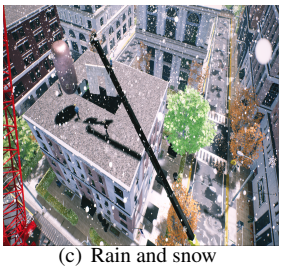
Also different times of day and therefore different lightning conditions are easily simulated. In the case of the



(a) Image with segmentation



(b) Caterpillar crane



(c) Rain and snow

Figure 6. Examples of synthetic data generated from UE4: (a) Hoisting with the appropriate segmentation by the object instance mask, (b) and (c) are scene in construction site.

construction worker, the tool allows us to create desired character appearance, motion or posture, based on motion capture data. Also different perspectives of the same situation, at the same time are possible. As we will see in section 5 we need to have the exact same situation with different zoom levels in order to get an objective ground truth for training the MetaNetwork. Data gathered in real-life would not be able to fulfill this constraint, since either the timing or the perspective of each zoom level would be slightly off.

The annotation of the synthetic data can be automatically achieved by an object instance mask plugin. In a dataset which is annotated by hand there is always a concern left, that the data may be annotated in a wrong manner. Manual annotation is a tedious task so one can not be sure to, e.g., not miss a human in the image or have slight errors in the localization of an object. All these concerns vanish with the use of automatic annotated data, since the simulation environment knows the ground truth and is therefore inherently objective.

5 Learning Framework

In the proposed approach a couple of networks are involved. For the detection in each of the zoom levels and around the crane we just use a state of the art object detection network and adjust the trained weights with transfer learning, see [15] for a detailed look on the base architecture. The MetaNetwork needs to be constructed in a dif-

ferent manner. Since it does not only take the image but also the current zoom level as an input we can not simply use a standard architecture but have to adapt to this format. The proposed architecture can be seen in figure 7

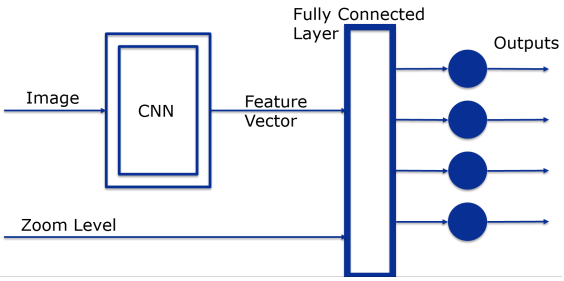


Figure 7. Network architecture of the MetaNetwork

So as we can see we first have a CNN, with nearly the same architecture as in the detection networks for each zoom level. We remove the last layers in order to just pass on the feature vector which is normally evaluated in the last layers of a network. We then connect this feature vector and the zoom level, encoded simply as an integer value, with a fully connected layer. Then the result of this fully connected layer will be given to our four desired outputs. The idea is that this MetaNetwork will deduce similar information as the detection networks but take the current zoom level into account and by that help to chose a suitable network. The training process will differ slightly from a normal CNN. Since the ground truth is the actual number of humans each of the networks detects, the dataset needs to reflect that. Therefore the dataset needs to contain 4 images, one for each zoom level, while the ground truth is the actual evaluation of each image by the corresponding zoom network. Then during training one of the four contained images can be chosen arbitrarily and the zoom level input needs to be passed on accordingly. This means that before we can train the MetaNetwork each of the specialized ZoomNetworks has to be trained sufficiently.

6 Conclusion and Future Work

Let us recap the approach presented in this paper. We defined a sensor setup which is able to detect humans around the crane truck and around the load. This camera setup contains standard monocular cameras and a zoom camera with a controllable zoom. Additionally we presented a CNN structure which is used for each of the mounted cameras individually. Then a meta algorithm, which is itself again a neural network yields a first estimation what an appropriate zoom level would be. According to this estimate the zoom level and the specialized network is chosen. After the proposed approach is evaluated a prototype system which is able to inform the crane operator about humans or other potentially endangered objects entering the working area is being developed

7 Acknowledgement

The work at hand was funded from the Federal Ministry of Education and Research (BMBF) under grant agreement number 01|16SV7738 and name SafeguARd.

References

[1] Shervin Ardeshir and Ali Borji. Ego2top: Matching viewers in egocentric and top-view videos. CoRR, abs/1607.06986, 2016.

[2] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. CoRR, abs/1612.05424, 2016.

[3] H. Y. Cheng, C. C. Weng, and Y. Y. Chen. Vehicle detection in aerial surveillance using dynamic bayesian networks. IEEE Transactions on Image Processing, 21(4):2152–2159, April 2012.

[4] Brian Chu, Vashisht Madhavan, Oscar Beijbom, Judy Hoffman, and Trevor Darrell. Best practices for fine-tuning visual classifiers to new domains. In ECCV Workshops, 2016.

[5] Navneet Dalal and Bill 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.

[6] B. K. Dan, Y. S. Kim, Suryanto, J. Y. Jung, and S. J. Ko. Robust people counting system based on sensor fusion. IEEE Transactions on Consumer Electronics, 58(3):1013–1021, August 2012.

[7] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. PAMI, 34, 2012.

[8] Markus Enzweiler and Dariu M Gavrila. Monocular pedestrian detection: Survey and experiments. IEEE transactions on pattern analysis and machine intelligence, 31(12):2179–2195, 2009.

[9] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. A mobile vision system for robust multi-person tracking. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8. IEEE, 2008.

[10] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision, 111(1):98–136, January 2015.

[11] Fabian Flohr, Dariu Gavrila, et al. Pedcut: an iterative framework for pedestrian segmentation combining shape models and multiple data cues. In BMVC, 2013.

[12] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. International Journal of Robotics Research (IJRR), 2013.

[13] Y. L. Hou and G. K. H. Pang. People counting and human detection in a challenging situation. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 41(1):24–33, Jan 2011.

[14] Horst Leisering. Rückfahrkameras an erdbaumaschinen, 2012.

[15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In ECCV, 2016.

[16] Xingjun Ma, James Bailey, Sudanthi N. R. Wijewickrema, Shuo Zhou, Zakaria Mhammedi, Yun Zhou, and Stephen O'Leary. Extracting real-time feedback with neural networks for simulation-based learning. CoRR, abs/1703.01460, 2017.

[17] Milad Memarzadeh, Mani Golparvar-Fard, and Juan Carlos Niebles. Automated 2d detection of construction equipment and workers from site video streams using histograms of oriented gradients and colors. Automation in Construction, 32:24–37, 2013.

[18] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, February 2015.

[19] Rafael Mosberger and Henrik Andreasson. Estimating the 3d position of humans wearing a reflective vest using a single camera system. In Field and Service Robotics, pages 143–157. Springer, 2014.

[20] Yair Movshovitz-Attias, Takeo Kanade, and Yaser Sheikh. How useful is photo-realistic rendering for visual learning? CoRR, abs/1603.08152, 2016.

[21] Weichao Qiu and Alan Yuille. Unrealcv: Connecting computer vision to unreal engine. In Gang Hua and Hervé Jégou, editors, Computer Vision – ECCV 2016 Workshops, pages 909–916, Cham, 2016. Springer International Publishing.

[22] M. Rauter. Reliable human detection and tracking in top-view depth images. In 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 529–534, June 2013.

[23] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. arXiv preprint arXiv:1612.08242, 2016.

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems (NIPS), 2015.

[25] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, Computer Vision – ECCV 2016, pages 102–118, Cham, 2016. Springer International Publishing.

[26] Abu HM Rubaiyat, Tanjin T Toma, Masoumeh Kalantari-Khandani, Syed A Rahman, Lingwei Chen, Yanfang Ye, and Christopher S Pan. Automatic detection of helmet uses for construction safety. In Web Intelligence Workshops (WIW), IEEE/WIC/ACM International Conference on, pages 135–142. IEEE, 2016.

[27] Nicolas Schneider and Dariu M Gavrila. Pedestrian path prediction with recursive bayesian filters: A comparative study. In German Conference on Pattern Recognition, pages 174–183. Springer, 2013.

[28] Mohammad Soltani, Zhenhua Zhu, and Amin Hammad. Automated annotation for visual recognition of construction resources using synthetic images. Automation in Construction, 62:14–23, 02 2016.

[29] Baochen Sun and Kate Saenko. Deep CORAL: correlation alignment for deep domain adaptation. CoRR, abs/1607.01719, 2016.

[30] Ting-En Tseng, An-Sheng Liu, Po-Hao Hsiao, Cheng-Ming Huang, and Li-Chen Fu. Real-time

people detection and tracking for indoor surveillance using multiple top-view depth cameras. 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4077–4082, 2014.

[31] Kristof Van Beeck and Toon Goedemé. Pedestrian detection and tracking in challenging surveillance videos. In José Braz, Julien Pettré, Paul Richard, Andreas Kerren, Lars Linsen, Sebastiano Battiato, and Francisco Imai, editors, Computer Vision, Imaging and Computer Graphics Theory and Applications, pages 356–373, Cham, 2016. Springer International Publishing.

[32] M. Wang, W. Li, and X. Wang. Transferring a generic pedestrian detector towards specific scenes. In 2012 IEEE Conference on Computer Vision and Pattern Recognition, pages 3274–3281, June 2012.

[33] J. Xiao, H. Cheng, H. Sawhney, and F. Han. Vehicle detection and tracking in wide field-of-view aerial video. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 679–684, June 2010.

[34] Huazhong Xu, Pei Lv, and Lei Meng. A people counting system based on head-shoulder detection and tracking in surveillance video. In 2010 International Conference On Computer Design and Applications, volume 1, pages V1–394–V1–398, June 2010.

[35] Donggeun Yoo, Namil Kim, Sunggyun Park, Anthony S. Paek, and In-So Kweon. Pixel-level domain transfer. CoRR, abs/1603.07442, 2016.