An Evaluation of Human Detection Methods on Camera Images in Heavy Industry Environments

Nico Zengeler
Computer Science Institute
Hochschule Ruhr West
Bottrop, Germany
nico.zengeler@hs-ruhrwest.de

Matthias Grimm
Computer Science Institute
Hochschule Ruhr West
Bottrop, Germany
matthias.grimm@hs-ruhrwest.de

Colja Borgmann
Computer Science Institute
Hochschule Ruhr West
Bottrop, Germany
colja.borgmann@stud.hs-ruhrwest.de

Marc Jansen
Computer Science Institute
Hochschule Ruhr West
Bottrop, Germany
marc.jansen@hs-ruhrwest.de

Sabrina Eimler
Computer Science Institute
Hochschule Ruhr West
Bottrop, Germany
sabrina.eimler@hs-ruhrwest.de

Uwe Handmann
Computer Science Institute
Hochschule Ruhr West
Bottrop, Germany
uwe.handmann@hs-ruhrwest.de

***Abstract*—In this paper we evaluate different machine learning models for human body detection in heavy industry environments. Contributing a framework to asses the reliability of a detection system in industrial environments, we compare techniques of feature extraction for support vector machines to artificial neural networks. To accommodate for common environmental challenges in heavy industry, such as dust, difficult light conditions and partially covered persons, we apply programmatic changes to our test image set and evaluate the accuracy of person detection, foot point estimation and the tendency of erroneous detections.**

***Index Terms*—heavy industry, image processing, human detection, neural networks, Industry 4.0**

I. Introduction

The emerging economic trend of industry 4.0 requires software to support heavy manufacturing structures. To increase productivity, flexibility and work safety, an automatic facility management strategy needs applications that gather and analyse information about the production process in realtime. The project *DamokleS 4.0* [5] aims to develop a system which aids employees of heavy industry sides using modern hardware and software. For example, augmented reality glasses and other smart mobile devices may provide information to the workers at any moment in time, which helps them to make more efficient use of time or access fast evacuation routes in case of an emergency. These systems require reliable detection of humans in factory sites to provide information. Putting the information into a context model allows for online information assistance and improved production planning. An essential part of such a software concerns processing of images provided by cameras.

In this paper we examine different methods to detect human bodies in a heavy industry setting. Our contributions include a simulated industrial work video data set, which we recorded in a laboratory setting, and various image perturbation methods to seek an accurate and fail-safe human detection system.

We begin this paper by delineating the context of this contribution, relating to other work within the *DamokleS 4.0* project and previous projects. We then illuminate our evaluation process by explaining how we collected and preprocessed our image data, which evaluation criteria we assessed, which person detection methods with what parameters we used and the results we obtained that way. In our conclusion we end this paper with a short discussion of our final results and a statement about future contributions and which improvements we might expect from them.

II. State of the art

Current state of the art knows a variety of methods to detect human bodies on camera images [3, 4, 7, 9, 15, 16, 17, 18, 20, 23, 24, 25, 27, 26]. We decided to focus our investigations on the well researched *Histogram of Oriented Gradients* (HOG) [2, 4, 13, 20, 22, 29], the neural architecture *You Only Look Once* (YOLO) [15, 16, 17] and the *OpenPose* system (OP) [3, 18, 23].

In a related project we introduced a video surveillance system to protect critical infrastructures using only HOG method combined with a Kalman tracking algorithm [10]. In this project, we designed our software architecture such that it supports human operators who detect, track and recognize suspicious subjects in case of an alert. We implemented our system at two reference airports in order to gather intelligence about arising challenges in real world applications. We found the huge amount of image data, recorded on a network of non-overlapping cameras, impeded the recovery of a once detected person. The camera-based data analysis consisted of several image processing modules like a salient-based people detection and a HOG algorithm based on the implementation of [13]. We decided to use a GPU-based implementation to speed up the HOG algorithm and make it fulfill our realtime requirements. The scenarios described in [10] resemble those in the context of heavy industries with respect to challenges

978-1-5386-9490-9/19/$31.00

introduced by different light conditions and the high need for fast algorithms.

Concerning the *DamokleS 4.0* project [5], [14] describes the overall software architecture underlying our context model [6]. Also [14] sketched the essential ideas that drive our test scenarios as well as the associated processes for implementation in mobile devices. The suggested scenarios concern workplace safety, production and maintenance applications. The proposed approach provides context-based support for factory employees during all these scenarios. For context recognition, [14] proposes the usage of mobile device sensors and external sensors devices mounted in the factory building, for example cameras and beacons.

III. Implementation

Our human detection system must meet different requirements. First, we need to consider difficulties in industrial environments; we thus need to consider image perturbations by dust, light reflections and moving objects that may hide workers from sight. Second, we want to perform an accurate translation of the human position into a world coordinate frame and thus need a precise estimation of the foot point location. And for a third requirement, our system must run in real time on affordable hardware.

In order to assess a most reliable human detection system, we determine three important quality factors: the accuracy in terms of human body detection, the precision of foot point localisation and the tendency of erroneous detections, which may either falsely detect persons or miss persons in the image. We evaluate these qualities on a set of simulated working task video sequences recorded in our laboratory, which we programmatically vary with increasing noise, light reflections and partial occultation via bulks of noise.

Figure 1 shows exemplary outputs of our three investigated methods on typical frames of our recorded data. The HOG and YOLO detections lack anticipation of hidden body parts, such that the location of the foot point remains rather imprecise. An exemplary pose estimation by OP on one of our simulated work task frames shows a good anticipation of the hidden body parts; we may reliably locate the foot point of the person standing behind an obstacle where the other methods would merely detect the upper body part.

In the light of these requirements, we expect the best results from the OP system, as it incorporates knowledge of a human body model and should thus resist to perturbations by noise, lighting conditions and partial occultation by design. We expect difficulties in the foot point detection performed by the HOG algorithm and the YOLO architecture, as they would only the visible body part. From the HOG method we expect good performance under normal conditions but a rapid decay in accuracy as our systematic perturbations proceed.

Fig. 1: Exemplary detections of our three investigated methods on different frames of our simulated working task. Left: OpenPose yields a human body skeleton of a person partially hidden by an object. Center: YOLO detects multiple persons but misses a person in the crowd, also it does not anticipate the hidden body parts. Right: HOG detects rough bounding boxes of three persons in a crowd.

A. Histogram of Oriented Gradients

The Histogram of Oriented Gradients (HOG) method yields feature descriptions of an image. Any machine learning model may then use these features in order to perform object detection, for example to perform human detection [4]. As elaborated by [22], this method first computes the gradient of an image, for example by central differences, then divides the image into adjacent, non-overlapping cells and for each cell computes the gradient orientations and bins them into a histogram. Grouping these histograms into larger blocks yields a concatenated block feature b and allows a block feature normalisation by the Euclidean norm:

$$b = \frac{b}{\sqrt{||b||^2 + \epsilon}}$$

The method then concatenates the normalized block features into a single HOG feature, which it then normalises again. Upon these features a learning algorithm may detect or classify objects in the image as done by [2]. For our evaluation, we used the OpenCV implementation with a support vector machine (SVM) [11, 21].

B. You Only Look Once

The *You Only Look Once* (YOLO) network performs multi-class object detection on images. YOLO runs in real-time and may detect humans as well as any other objects that it has learned. The neural architecture basically consists of a deep cascade of convolutional and max pooling layers with different filter sizes and pooling regions. The output layer of this network yields a grid of $S \times S$ cells; for each cell it predicts a class probability and a number of bounding boxes with a confidence score [17]. The method multiplies the conditional class probabilities and the individual box confidence scores, obtaining a class-specific confidence score for each cell, using the intersection over union (IOU) measure between predicted bounding boxes and ground truth:

$$P(class|obj) \cdot P(obj) \cdot IOU_{pred}^{truth} = P(class) \cdot IOU_{pred}^{truth}$$

The authors incrementally improve on their neural architecture [16]. For our experiments we use the first version of YOLO as implemented in the Darknet C++ library [15].

C. OpenPose

The OpenPose (OP) method reliably estimates human body poses by employing body model knowledge [3, 18, 23]. Depending on the choice of the body model (COCO, BODY25 or MPI) the runtime and pose estimation quality may vary within overall reasonable performance ranges. OP estimates two-dimensional poses of multiple people in an image, combining multiple stages of learned Part Affinity Fields (PAF) and Part Confidence Maps (PAC) with a bipartite graph matching procedure [3]. The neural architecture learns to associate body parts with individuals in the image, encoding a global context and jointly learning part locations and their association within one prediction process. The real-time applicability allows us to detect multiple persons in a live video stream and use this information for precise foot point location.

IV. Evaluation

We recorded video data upon which we tested our methods in our laboratory. We have mounted four AVT Prosilica GE1650C video cameras with a resolution of 1600×1200 pixels, from which we use the recordings of two cameras (C_1, C_2) for our evaluation. We set up a simulated work task scene in three modalities: a single person walking through the room and working on a computer, a group of two persons walking the same path and a group of six persons doing random walk and performing arbitrary work. For the first two modalities, the actors wore workwear. In the group footage three persons in everyday clothing enter the scene. We recorded about $4,000$ frames with two cameras C_1 and C_2, from which we manually labelled a total of $3,557$ images, each with bounding boxes around the individual persons and around their feet. We then preprocessed these images with our systematic perturbations as explained below and obtained a test set containing $142,311$ images, upon which we measured the three methods qualities.

A. Noise

To evaluate the methods resistances against noise, we perturbate the input images with noise, which we generate according to the normal distribution [12]:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

We set $\mu = 0$ and increase σ in a range of $[0, 250]$ with a step size of 25, such that we generate eleven images with a different noise level for each video frame, starting with $\sigma = 0$ and ending with $\sigma = 255$. We did not try other kinds of noise as we make no further assumptions on the nature of the noise that may perturbate the whole camera image [1].

Fig. 2: Example images of our preprocessed video frames. Top row: Gaussian noise increases from $\sigma = 0$ (left) to $\sigma = 250$ (right). Second row: the strength of our blinding light increases from 0% of light map addition (left) to 100% (right). Third row: our six occultation modalities. Bottom row: the combined perturbations, from $i = 0$ (left) to $i = 10$ (right).

B. Light

In order to investigate the performance under different light conditions, we simulated a blinding light effect on a shining surface. We therefore added a blinding light effect [8] on the background image and subtracted the original background, such that only the light maps shown in figure 3 prevailed. We then added this light map to each frame recorded by the respective camera, with a strength varying from 0% to 100% in steps of 10%, such that we obtained eleven blinding light images per original video frame. We did not try darkened conditions as we assume the factory site to provide enough illumination for cameras to work proper.

C. Occultation

To simulate partial occultation, we partially covered our manually labelled bounding boxes with noise. We covered

Fig. 3: Top row: background images of camera C_1 (right) and C_2 (left). Bottom row: corresponding light maps added to the images of camera C_1 (right) and C_2 (left).

all the bounding boxes in an image simultaneously in six modalities: the horizontal left, center and right part as well as the vertical top, center and bottom part. Each occultation modality covers a third of the bounding box along the respective axis. Within each bounding box, we cover the occulted region with a bulk of black and white pixels randomly generated according to the normal distribution. We did not add real occlusion blocks as our simulated environment also provides natural occultation by tables and other obstacles.

D. Combined perturbations

We combine the perturbations by noise, blinding light and occultation to obtain a fourth, most challenging test case of hardship. To avoid a combinatorial explosion, we decided to generate eleven images per original frame, to each of which we add noise and blinding light as previously explained, but increasing in the same pace: for $i \in \{0...10\}$ we set the noise $\sigma = 25 \cdot i$ and the light strength to $(10 \cdot i)\%$ We also add a combination of bounding box occultation by covering each person in the bottom third in horizontal direction and along his or her vertical central axis. The bottom row of figure 2 shows the results of this combined perturbation process.

E. Results

To evaluate the methods we consider their performances for our three requirements under varying conditions as shown in figure 4 and their computational resource usage as listed in table I. We evaluate the detection qualities via the accuracies of person and foot point detections as well as the overall tendency of erroneous detections. As we average the erroneous detections, which amounts to < 0 for missing persons and > 0 for false positives detections, a value of 0 means that the methods tends to produce as many false positives as it misses persons on average.

	HOG	YOLO	OP
FPS	33.5	18.4	11.3
GPU Memory	251 MB	1293 MB	1313 MB

TABLE I: Speed and computational resource usage of the three investigated methods measured on a video file.

Our manually labelled data consists of bounding boxes for persons and their feet; to obtain a score for performance measures, we test whether the methods estimate correct locations of the persons and their feet by checking if predicted points lie within our manually labelled boxes. Given the predicted bounding boxes, we use their central point for person location and the center of the bounding boxes bottom line for foot point estimation to evaluate the HOG and YOLO method. To evaluate the OP system, we used the BODY25 model with default parameters. Using this human body model, we evaluated the neck key point for person location and the center between right and left ankle key points for foot point location. As shown in figure 4, the OP system shows a clear advantage in noise resistance, light anticipation and partially covered body completion, especially considering foot point location. Concerning partial coverages, we can state that the YOLO detector especially misses persons covered along their vertical central axis. Looking at the erroneous detections tendencies, we may state that, on average, OP tends to falsely detect as many persons as it misses while YOLO tends to miss persons in the image. Considering the hardship of combined perturbations, all methods rapidly decay in performance, yet OP excels all other methods. Evaluating the detailed individual results, as published under [19], the tendency of erroneous detections hints that OP falsely detects persons in the stack of chairs on the background of C_1. Unsurprisingly, the HOG method struggles with all our image perturbations most.

V. Conclusion

We contributed a method to evaluate person detection models for heavy industry environments. We have published all our results, the source code and raw data mentioned in this paper under [19]. Our investigations focus on human detection on RGB cameras. From our investigated detectors, OpenPose proves most robust to partial coverages, noisy images and blinding light conditions. Yet the YOLO architecture may provide useful auxiliary scene information, which we may use to track non-human objects. The HOG method may still prove useful in cases with little computational resources but perfect image conditions.

A. Discussion

In our evaluations we used default parameters of all our investigated methods. Fine-tuning these hyper parameters may change results, also it may make sense to use means of Transfer Learning in order to fine-tune the models on the concrete industrial situation. As for YOLO, we have used the pre-trained weights from the first version. We did not implement any sophisticated human body completion techniques nor did

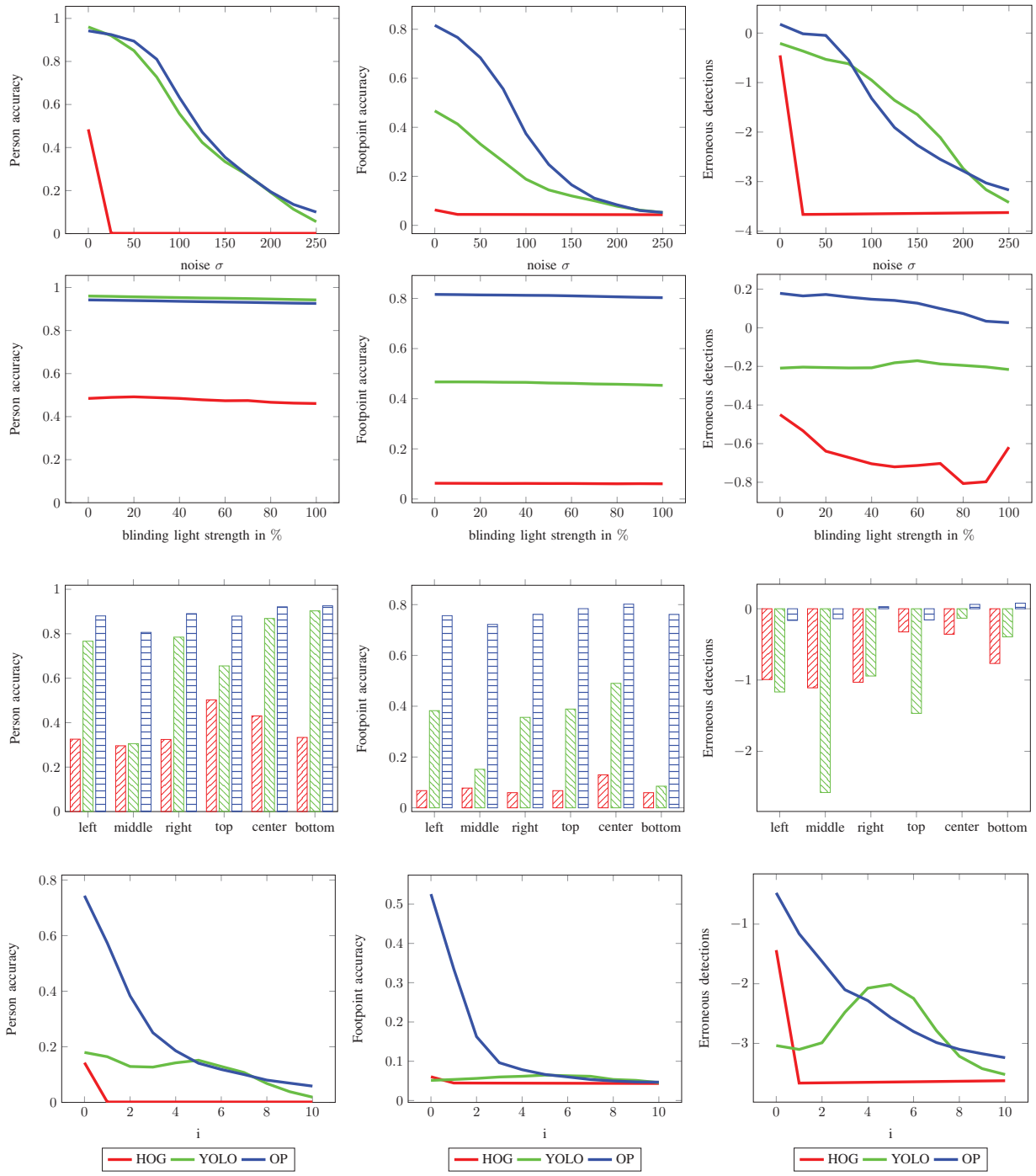

Fig. 4: Overall results averaged over all sequences. Top row: resistances against noise. Second row: anticipation of the blinding light effect. Third row: partially covered persons in our six occultation modalities. Bottom row: the hardship of combined perturbations. Left column: accuracy of person detection. Middle column: accuracy of foot point detection. Right column: number of erroneous detections (< 0: missing persons, > 0: false positives, 0: as many missing persons as false positives on average).

we use any information of temporal correlation between the video frames but worked on each frame independently. Doing so, we discard advantages that may come from recurrent neural network architectures or visual flow information. As we could not find any related work that describes challenges for human detection in heavy industrial environments, we cannot make a statement on how well our dataset simulated the real scenarios.

B. Future work

Future work may evaluate recent development in object detection, for example YOLOv3 and through-wall human pose estimation using radio signals, as described by [28]. We also aim to implement techniques to track individual persons through the factory site. This requires additional identification features, such as face recognition, gait recognition or smart device identification.

Acknowledgement

This work was supported by the *DamokleS 4.0* project [5] project funded by the European Regional Development Fund (ERDF), the European Union (EU) and the federal state North Rhine Westphalia.

References

[1] A. K. Boyat and B. K. Joshi. "A Review Paper: Noise Models in Digital Image Processing". In: *CoRR* abs/1505.03489 (2015). arXiv: 1505.03489. URL: http://arxiv.org/abs/1505.03489.

[2] H. Bristow and S. Lucey. "Why do linear SVMs trained on HOG features perform so well?" In: *CoRR* abs/1406.2419 (2014). arXiv: 1406.2419. URL: http://arxiv.org/abs/1406.2419.

[3] Z. Cao et al. "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields". In: *CVPR*. 2017.

[4] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *CVPR*. Vol. 1. June 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[5] *DamokleS 4.0 - IKT für Cyber Physical Systems*. https://www.damokles40.eu/. Accessed: 2018-10-31.

[6] A.K. Dey. "Understanding and Using Context". In: *Personal Ubiquitous Comput.* 5.1 (Jan. 2001), pp. 4–7. ISSN: 1617-4909. DOI: 10.1007/s007790170019. URL: http://dx.doi.org/10.1007/s007790170019.

[7] P. Dollar et al. "Pedestrian Detection: An Evaluation of the State of the Art". In: *TPAMI* 34.4 (Apr. 2012), pp. 743–761. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2011.155.

[8] *GIMP Documentation*. https://docs.gimp.org/2.10/en/plug-in-lighting.html. Accessed: 2018-11-09.

[9] B. Girish and S. Venkata. "METHOD AND APPARATUS FOR HUMAN DETECTION IN IMAGES". 20180157904. June 2018. URL: http://www.freepatentsonline.com/y2018/0157904.html.

[10] U. Handmann et al. "APFel - Fast multi camera people tracking at airports, based on decentralized video indexing". In: 2 (Jan. 2014), pp. 48–55.

[11] *OpenCV*. https://github.com/itseez/opencv. 2015.

[12] P. Z. Peebles. *Probability, random variables, and random signal principles*. Vol. 3. McGraw-Hill New York, NY, USA: 2001.

[13] V. Prisacariu and I. Reid. *fastHOG - a real-time GPU implementation of HOG*. Tech. rep. 2310/09. Department of Engineering Science, 2009.

[14] Z. Qasem et al. "Dynamic, Adaptive, and Mobile System for Context-Based and Intelligent Support of Employees in Heavy Industry". In: (Oct. 2018). DOI: 10.1109/ES.2018.00021.

[15] J. Redmon. *Darknet: Open Source Neural Networks in C*. http://pjreddie.com/darknet/. 2013–2016.

[16] J. Redmon and A. Farhadi. "YOLOv3: An Incremental Improvement". In: *CoRR* abs/1804.02767 (2018). arXiv: 1804.02767. URL: http://arxiv.org/abs/1804.02767.

[17] J. Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *CoRR* abs/1506.02640 (2015). arXiv: 1506.02640. URL: http://arxiv.org/abs/1506.02640.

[18] T. Simon et al. "Hand Keypoint Detection in Single Images using Multiview Bootstrapping". In: *CVPR*. 2017.

[19] *Source code and data used in this paper*. https://gitlab.hs-ruhrwest.de/nico.zengeler/human-detection. Accessed: 2018-11-30.

[20] T. Surasak et al. "Histogram of oriented gradients for human detection in video". In: *ICBIR*. May 2018, pp. 172–176. DOI: 10.1109/ICBIR.2018.8391187.

[21] *The OpenCV Reference Manual*. 2.4.9.0. Itseez. Apr. 2014.

[22] C. Tomasi. "Histograms of oriented gradients". In: *Computer Vision Sampler* (2012), pp. 1–6.

[23] S. Wei et al. "Convolutional pose machines". In: *CVPR*. 2016.

[24] B. Wu et al. "SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving." In: *CVPR Workshops*. 2017, pp. 446–454.

[25] S. Wu et al. "Exploiting Target Data to Learn Deep Convolutional Networks for Scene-Adapted Human Detection". In: *TIP* 27.3 (Mar. 2018), pp. 1418–1432. ISSN: 1057-7149. DOI: 10.1109/TIP.2017.2779271.

[26] S. Zhang, J. Yang, and B. Schiele. "Occluded Pedestrian Detection Through Guided Attention in CNNs". In: *CVPR*. June 2018.

[27] S. Zhang et al. "Towards Reaching Human Performance in Pedestrian Detection". In: *TPAMI* 40.4 (Apr. 2018), pp. 973–986. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2017.2700460.

[28] M. Zhao et al. "Through-Wall Human Pose Estimation Using Radio Signals". In: *CVPR*. June 2018.

[29] Z. Qiang Zhu et al. "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients". In: *CVPR*. Vol. 2. June 2006, pp. 1491–1498. DOI: 10.1109/CVPR.2006.119.