2023 IEEE International Conference on Image Processing (ICIP) | 978-1-7281-9835-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICIP49359.2023.10223161

REAL-TIME WHEEL DETECTION AND RIM CLASSIFICATION IN AUTOMOTIVE PRODUCTION

Roman Staněk [1], *Tomáš Kerepecký* [2,3], *Adam Novozámský* [3], *Filip Šroubek* [3], *Barbara Zitová* [3], *Jan Flusser* [3]

[1]Charles University, Czechia, [2]Czech Technical University in Prague, Czechia
[3]Institute of Information Theory and Automation, The Czech Academy of Sciences, Czechia

ABSTRACT

This paper proposes a novel approach to real-time automatic rim detection, classification, and inspection by combining traditional computer vision and deep learning techniques. At the end of every automotive assembly line, a quality control process is carried out to identify any potential defects in the produced cars. Common yet hazardous defects are related, for example, to incorrectly mounted rims. Routine inspections are mostly conducted by human workers that are negatively affected by factors such as fatigue or distraction. We have designed a new prototype to validate whether all four wheels on a single car match in size and type. Additionally, we present three comprehensive open-source databases, CWD1500, WHEEL22, and RB600, for wheel, rim, and bolt detection, as well as rim classification, which are free-to-use for scientific purposes.

***Index Terms*—** Detection, Classification, Automotive

1. INTRODUCTION

In 2021, global motor vehicle production was estimated to be over 80 million vehicles [1]. Most quality check tasks are performed by trained workers, who can be affected by many negative factors, which reduces the reliability of the inspection. This tedious work provides a significant opportunity for automation through computer vision, which has the potential to lower the cost of the overall process and, at the same time, achieve superior accuracy. Despite the prevalence of automated computer vision tasks, the quality control of rim mounting inaccuracies is still done manually. This paper aims to design a real-time system to ensure that all four rims on a car are of the same size and type, which is a crucial factor for maintaining car stability and passenger safety.

There are limited studies focused on wheel detection and, to the best of our knowledge, a lack of literature regarding rim classification. In this regard, we present a novel approach that constitutes the first comprehensive pipeline for joint wheel detection, classification, and size estimation.

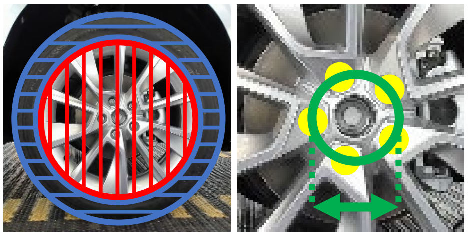

Fig. 1. Visualization of the main wheel parts. The tire is marked in blue, the rim in red, the wheel bolt centers in yellow, and the diameter of the pitch circle in green.

Contributions. (1) we propose a novel real-time rim detection, classification, and inspection approach and design a prototype ready-to-use in automotive production; **(2)** we contribute three new publicly available datasets, namely **CWD1500** for car and wheel detection, **WHEEL22** for rim classification and **RB600** for bolt detection.

2. RELATED WORK

We present an overview only in the area of wheel detection, as there is no public research on rim classification. These studies predominantly employ the *Hough transform (HT)* [2] and use machine learning to determine the presence of a wheel.

A comprehensive introduction to car and wheel detection is in [2], which presents a three-stage approach for detecting car contours from side view and identifying wheels using HT and *SURF descriptors* [3]. They use heuristics and a *Snake algorithm* [4] to improve results, but the results are inconclusive due to a small dataset (100 images) and white background.

The Master thesis [5] detects wheels utilizing real-world recordings and using *Local binary patterns* [6] and *Random forest classifier* [7]. Another work [8] identifies 14 regions of interest in vehicles from a side view, including wheels, using a classifier trained on Haar-like features [9] and HT.

The paper [10] uses *Fast HT* [11] to detect wheels in a deployed industrial vehicle classification system in Russia, filtering the Hough space to avoid false positives.

3. METHOD

Our method consists of several building blocks in which standard computer vision methods are complemented with deep

This work was partially supported by the Czech Science Foundation, grant no. GA21-03921S, and by the *Praemium Academiae* awarded by the Czech Academy of Sciences.

learning methods. We start with the creation of three datasets and then describe the car and wheel detection, rim classification, and finally, rim-size estimation. The terms *wheel*, *rim*, and *tire* can be ambiguous in the common language. In this paper, we refer to a wheel as the combination of rim and tire. A rim is a rigid core of the wheel usually made from metal. A tire is mounted on the rim and ensures good contact with the surface under the car. It is usually made of rubber-like material. For illustration, see Figure 1. The primary purpose of the rim is to provide rigid support for the tire and to transmit forces that affect the movement of the vehicle, such as the rotational force from the engine to the tire. [12].

3.1. Datasets

All data were collected at the Škoda Auto factory in Mladá Boleslav, where quality control is carried out. This company permitted us to install the monitoring equipment to gather car data and use them for research purposes. Cars are stationary on a slowly moving conveyor belt that is well-lit by multiple sources. The cameras were arranged on both sides of the conveyor belt; Camera A recorded cars from the right side, and Camera B from the left. A top-down schema of the setup is shown in Figure 2. Images of the scene taken by individual cameras simultaneously are shown in Figure 3. We employed standard Logitech BRIO 4K Stream Edition cameras with the same configuration as in [13]. The data collection script ran continuously for several days at a rate of one frame per second in FullHD quality, recording data in ten-minute bursts. Every frame was captured as Motion JPEG, and the whole time-lapse video was encoded in HEVC H.265. The data was gathered during standard work shifts; people and objects can move in the scene; see Figure 4. Thirty-four hours of collected data from both cameras were obtained after removing unusable video segments. It is necessary to provide training data with accurately labeled objects to train a neural network for object detection or tuning parameters for standard algorithms such as HT. We manually annotated the cars and wheels by drawing bounding boxes in the training images using the *Computer Vision Annotation Tool(CVAT)* [14]. The first dataset, **CWD1500**, is designed for car and wheel detection and includes 1000 training frames, 250 validation frames, and 250 testing frames. The training set also includes 91 unlabelled images to prevent false positives during training.

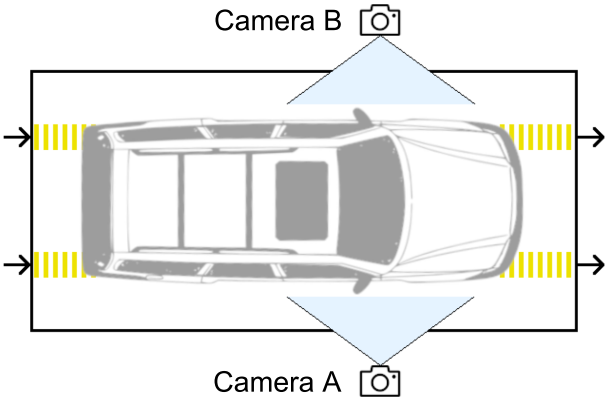

Fig. 2. Top-Down Camera Setup: The cameras were placed on either side of the conveyor belt, with Camera A capturing cars from the right and Camera B capturing them from the left.

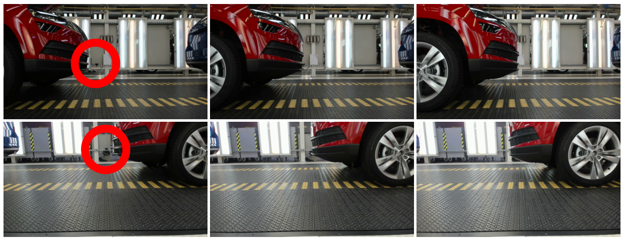

Fig. 3. Sequences of 3 subsequent frames from Camera A (top) and Camera B (bottom) illustrate the conveyor belt movement speed - approximately 11.5 cm/s. The red circles in the left column show the opposite camera.

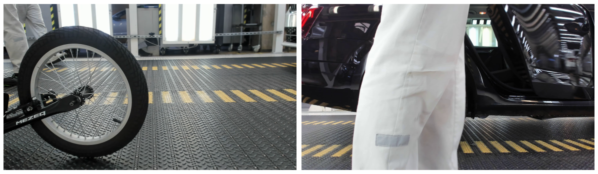

Fig. 4. Examples of problematic objects in the scene: a scooter wheel is not the object of interest, and a staff standing in front of the car completely occluding the wheel.

We employed the YOLO [15] detection network, which had been trained using the CWD1500 dataset, to identify the location of all rims in the collected videos. Subsequently, each frame was processed by cropping it to a square shape centered around the detected bounding box and resized to 256 x 256 pixels. It is important to note that some of the identified rims were not entirely captured in the pictures, which assisted in generalizing the learning process of classification, since it functioned as cropping in standard data augmentation. We manually identified 31 classes of rims based on differences in shape and color of detected wheels. For ten classes the number of representative samples was too low (less than 300). Therefore, we used only 21 classes for further analysis. We randomly selected 100 training samples for each class, 25 validation samples, and 25 test samples. It should be noted that images of one car will only appear in one set (training, validation, or test). We added one special class to include cases when the detector returns a candidate that cannot be classified even by a human. One example per class is shown in Figure 5. We refer to this comprehensive labeled dataset containing 3300 rim images as **WHEEL22**.

The last dataset was created for detecting five wheel bolts. It contains 400 images for training, 100 for testing, and 100 for validation. The rims and bolts were also manually annotated using CVAT. This dataset is called **RB600**.

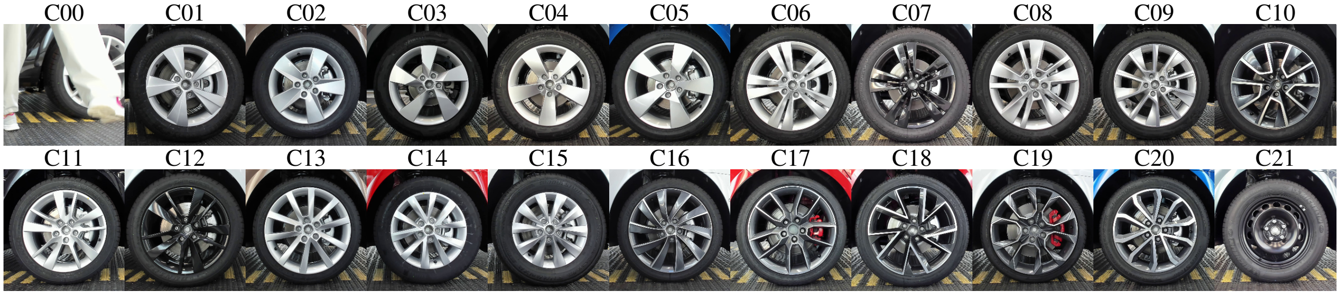

Fig. 5. WHEEL22 dataset: 1+21 rim categories. C00 category is used for handling occlusions.

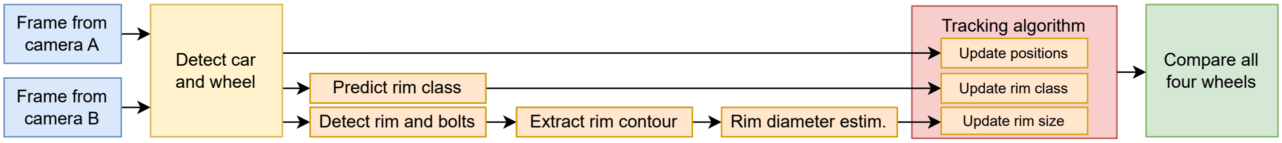

Fig. 6. Data flow in the proposed prototype.

3.2. Car and wheel detection

This section is split into two subsections. The first part deals with research utilizing traditional computer vision techniques, while the latter focuses on convolutional neural networks. A comparison of the two approaches is presented in Table 1. There are *precision* (P), *recall* (R), and *mean average precision* (mAP) to evaluate object detection performance. All of them are calculated using the *Intersection-over-Union* (IoU) threshold of 50%. If a range is specified, such as mAP@.5:.95, it indicates the average mAP over an IoU range from 0.5 to 0.95 with step size 0.05. From the results, it is evident that the deep learning method outperforms traditional methods.

a)Traditional Computer Vision: We chose HT as the most appropriate method based on the related work presented in Section 2. HT is a widely accepted technique for detecting geometric shapes that a limited number of parameters can describe. We utilized an implementation provided by [16] that was optimized for minimizing memory usage. To enhance its performance, we carried out several preprocessing steps. The image was first downscaled, converted to grayscale, and then blurred using a Gaussian filter to reduce noise and eliminate high-frequency components. The results obtained by measuring performance on the validation dataset of CWD1500 are summarized in Table 1 (first line) and are satisfactory for a baseline method. The true positives do not always match the rim contours, as the car wheels are not always perpendicular to the cameras' optical axes. The false negatives in the detection can be attributed to three reasons: dim rims, partially occluded rims, and rims partially out of the camera field of view. In the third group, the wheels are missing more than half of their area, which is hard for HT to detect.

b)Deep Learning Approach: Standard deep-learning detection methods include approaches such as U-net [17], Mask-RCNN [18], and YOLO [15]. We chose YOLOv5s specification [19] with 7M parameters and pre-trained on the COCO val2017 dataset. The model was re-trained for 50 epochs using a 2:1:1 split of the CWD1500 dataset. The estimated inference time on a single frame with YOLOv5s is around 26 ms, which is still sufficiently fast for our purposes (HT takes only 8ms).

3.3. Rim classification

We compared traditional techniques with deep learning approaches as in the previous section on wheel detection.

a)Traditional Computer Vision: As the traditional computer vision technique, we combined a *Histogram of Oriented Gradients* (HOG) [20] with a *Support Vector Machines* (SVM) [21] classifier. HOG is a feature descriptor used in computer vision for object detection. It represents the orientation of intensity gradients in an image, dividing the image into small cells and counting the gradient directions in each cell. The results of this analysis are then compiled into a histogram, which serves as a descriptor of the object's appearance. The 'orientation' parameter defines how many bins the histogram has in each cell. The parameter 'pixels per cell' describes the size of one cell in pixels. The results for three algorithm settings on WHEEL22 dataset are summarized in Table 2. In this case, the maximum achieved accuracy below 0.75 is insufficient.

b)Deep Learning Approach: Here we will concentrate on transfer learning and EfficientNet [22], a high-performing

Method		Class	Instances	P	R	mAP@.5	mAP@.5:.95
HT		wheel	182	1.000	0.703	—	—
YOLO	Sec. 3.2	wheel	182	0.983	0.970	0.993	0.962
		car	300	0.983	0.980	0.993	0.928
	Sec. 3.4	bolt	475	1.000	0.998	0.995	0.651
		rim	95	0.984	1.000	0.995	0.993

Table 1. The performance of the detection method was evaluated on test data, comparing two models: HT and YOLOv5s, using the CWD1500 dataset. Additionally, the bottom section of the results presents the outcomes of the same YOLO architecture trained and tested on the RB600 dataset.

Orientation	Pixels per cell	Accuracy	Features
9	8x8	0.643	73K
13	24x24	**0.744**	7.5K
16	24x24	0.718	9.2K

Table 2. Results of HOG with multiple variations of parameters. The column *Features* describes the number of features generated per image.

Unfrozen layers	Trainable param.	Accuracy	Train time [s]
2	2.5K	0.956	574
10	893.2K	0.989	659
25	1.5M	**0.995**	1103
237	4M	0.989	1848

Table 3. Transfer learning results using EfficientNet with various unfrozen layer configurations.

classification network. Transfer learning is a technique where a model pre-trained on a large dataset, such as ImageNet [23], is then used for training on a smaller dataset. This process accelerates and improves training by leveraging the learned representations from the pre-trained model. The layers of the pre-trained model are divided into two categories: frozen and trainable. The frozen layers remain unchanged, while the trainable layers are modified during the training. The number of trainable layers varies depending on the model, dataset size, and complexity. Individual runs with the number of unfrozen layers and achieved validation accuracy are in Table 3. The findings show that it is sufficient to train 25 layers, for which we reach a close-to-optimal model while training time is still moderate. The inference time of 60 ms per image is also favorable for our intended application. The confusion matrix for the test data shows similarity to the identity matrix, except for three classes. Specifically, C08 had two misrecognized representatives, C09 had four, and C14 had one, which resulted in an overall accuracy of 98.72%.

3.4. Rim size estimation

Since real rim dimensions are not known, we can either compare only relative sizes within a single car or use objects of known size to estimate the real rim diameter. All cars in our scenario have a pitch circle diameter of 112 mm, so the object of known size, which we have to detect, is the circle on which five bolts lie. The cameras were not calibrated precisely and had slight discrepancies in tilt and mounting positions. The car position on the conveyor belt is not fixed, and wheels may not be perfectly perpendicular to the camera. Therefore, the rim circumference may not be a perfect circle but an ellipse, as shown in Figure 7. The second detection network was trained on the RB600 set to detect the rim and bolts. The same YOLOv5s architecture as described in Section 3.2b was used. The input to the second YOLO network was the bounding box of the wheel detected by the first YOLO network. The performance of the bolt and rim detection, including the overall performance, is summarized in Table 1. Then we calculate the ellipse [24] on which the centers of the bolts lie.

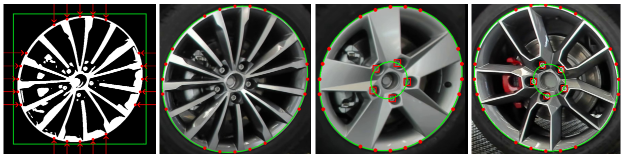

Fig. 7. Detection of rim and bolt ellipses.

To extract the outline of the rim, we applied Otsu's method [25] for thresholding. To find stable sample points, we cast rays from the center of each edge, spaced at 10% of the edge length. When a ray hits a white pixel, that location is added as a sample point. We use these points to fit the second ellipse that matches the contour of the rim. Visualization of the entire procedure using rays is presented in Figure 7. From these two ellipses we are able to estimate the real size of the rim.

3.5. Tracking

The tracking algorithm for car parts is of paramount significance as it improves accuracy and enhances detection speed. We use only Camera A and apply the same tracking information to Camera B due to the minor differences in horizontal coordinates. The IoU of bounding boxes conducts the data association for tracking in consecutive frames. The wheels are tracked similarly to the car and then assigned to the currently tracked car. Using the tracking information, calculating the median class for a particular rim achieves 100% accuracy on tested videos of the total length of 10 hours containing 500 cars.

The prototype design follows the order described in Section 3. A visual representation of the primary modules and their inter-module data flow can be seen in Figure 6. The procedure begins by acquiring frames from both cameras. Next, vehicle and wheel detection is performed (3.2b). Subsequently, the rim class is predicted (3.3b), and the diameter of the rim is estimated from the bounding box surrounding the wheel (3.4). The final step involves comparing all four wheels. The average processing time for a single input consisting of two frames, one from each camera, is 0.4 seconds for the entire pipeline. All experiments were performed on GeForce RTX 2070.

4. CONCLUSION

We proposed a real-time, fully automated system for rim size inspection of cars moving on the assembly line. The system consists of three steps: car and wheel detection, rim classification, and estimation of real rim dimensions. Traditional computer vision methods, such as Hough Transform and SVM with HOG features, were compared with deep learning techniques. When deep learning techniques are selected, the success rate in each intermediate step is approximately 99 percent. For the purpose of learning and testing, three datasets were prepared, which are publicly available for scientific purposes on Kaggle:
https://www.kaggle.com/datasets/adamnovozmsk/cawdec

5. REFERENCES

[1] OICA, "Produciton statistics," https://www.oica.net/, January 2023 [Online].

[2] Allam Shehata Hassanein, Sherien Mohammad, Mohamed Sameer, and Mohammad Ehab Ragab, "A survey on hough transform, theory, techniques and applications," *arXiv*, 2015.

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "Surf: Speeded up robust features," in *Computer Vision – ECCV 2006*, Aleš Leonardis, Horst Bischof, and Axel Pinz, Eds., Berlin, Heidelberg, 2006, pp. 404–417, Springer Berlin Heidelberg.

[4] Michael Kass, Andrew Witkin, and Demetri Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, Jan 1988.

[5] Karin Hultström, "Image based wheel detection using random forest classification," Master's thesis, Lund university, Faculty of Engineering Centre for Mathematical Sciences Mathematics, 2013.

[6] Timo Ojala, Matti Pietikäinen, and David Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[7] Tin Kam Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 1995, vol. 1, pp. 278–282 vol.1.

[8] Alberto Chávez-Aragón, Robert Laganière, and Pierre Payeur, "Vision-based detection and labelling of multiple vehicle parts," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 1273–1278.

[9] Paul Viola and Michael Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001, vol. 1, pp. I–I.

[10] Anton Grigoryev, Dmitry Bocharov, Arseniy Terekhin, and Dmitry Nikolaev, "Vision-based vehicle wheel detector and axle counter," *Proceedings - 29th European Conference on Modelling and Simulation, ECMS 2015*, pp. 521–526, 05 2015.

[11] Hungwen Li, Mark A Lavin, and Ronald J Le Master, "Fast hough transform: A hierarchical approach," *Computer Vision, Graphics, and Image Processing*, vol. 36, no. 2, pp. 139–161, 1986.

[12] Günter Leister, *Passenger Car Tires and Wheels*, 1st edition. Springer, Cham, 2018.

[13] Adam Novozámský et al., "Automated object labeling for cnn-based image segmentation," in *2020 IEEE International Conference on Image Processing (ICIP)*, Oct 2020, pp. 2036–2040.

[14] CVAT.ai Corporation, "Computer vision annotation tool (cvat)," Sept. 2022.

[15] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015.

[16] HK. Yuen, J. Princen, J. Illingworth, and J. Kittler, "Comparative study of hough transform methods for circle finding," *Image and Vision Computing*, vol. 8, no. 1, pp. 71–77, 1990.

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017.

[19] Glenn Jocher et al., "ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference," Feb. 2022.

[20] 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)*, 2005, vol. 1, pp. 886–893.

[21] Corinna Cortes and Vladimir Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep 1995.

[22] Mingxing Tan and Quoc V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *CoRR*, vol. abs/1905.11946, 2019.

[23] Olga Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[24] Radim Halır and Jan Flusser, "Numerically stable direct least squares fitting of ellipses," in *Proc. 6th International Conference in Central Europe on Computer Graphics and Visualization. WSCG*. Citeseer, 1998, vol. 98, pp. 125–132.

[25] Nobuyuki Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.