Camshaft's Journal Defects Detection and Classification Using Faster R-CNN Approach

[1,2] Fuad Widiatmoko
[1] Department of Engineering
Infinitigroup
[2] Master of Mechanical Engineering
Swiss German University
Tangerang City, Indonesia
fuadwidiatmoko7@gmail.com

Hanny J. Berchmans
Master of Mechanical Engineering
Swiss German University
Tangerang City, Indonesia
hannyjberchmans2020gmail.com

Henry Nasution
Master of Mechanical Engineering
Swiss German University
Tangerang City, Indonesia
henry.nasution@sgu.ac.id

[1,2] Paulus Gagat Charisma Arwidhiatma
[1] Department of Technical Support
Trias Indra Saputra
[2] Master of Mechanical Engineering
Swiss German University
Tangerang City, Indonesia
paulusgagat@gmail.com

[1,2] Eko Ari Wibowo
[1] Department of Mechanical Engineering
Astra Manufacturing Polytechnic
[2] Master of Mechanical Engineering
Swiss German University
Tangerang City, Indonesia
ari.wibo07@gmail.com

Aditya Pratama Nugroho
Department of Engineering
Infinitigroup
Bekasi, Indonesia
adit.2305@gmail.com

Abstract*—The camshaft is an essential part of a machine widely manufactured from wrought steel and nodular cast iron. One of the critical parts is a journal because this section should not have a physical defect on its surface. Several physical defects in the camshaft's journal are identified in a typical gasoline machine camshaft. This work explores different state-of-the-art object detection methods and their applicability for camshaft's journal and sprocket detection and classification tasks. Specifically, we implemented Faster R-CNN as part of Detectron2 using its base models and configurations. The results demonstrate that the X101-FPN base model for Faster R-CNN with the default configurations of Detectron2 is efficient and general enough to be applied to defect detection. This approach results in average correct detection score of 90.4% for test sets of the challenge. Though the visualizations show good prediction result, there are still some wrong defect detections. As a result, we compare the prediction results to the existing dataset and find some discrepancies.

Keywords—camshaft, journal, sprocket, defect, detectron2, faster R-CNN.

I. INTRODUCTION

A central component of the typical automotive gasoline engine and other internal combustion engines is the camshaft. Its performance and operational quality have a significant effect on the performance of the automobile sector and the growth of the whole automotive industry. Cam mechanism is commonly used in all kinds of machinery, especially automatic machinery, automatic control unit and assembly lines, which can make the follower correctly realize any anticipated motion rule. As an integral part of mechanical goods such as cars and bicycles, camshaft components are in tremendous demand every year. Since the camshaft is the engine's main transmission part, its output specifications are higher [1]. A camshaft is a rotating rod that slides against machinery to convert rotational motion into linear motion. The camshaft moves away from the axis of rotation as it is pushed by the machinery, resulting in a change of motion.

Fig. 1 shows a typical gasoline engine camshaft has 4 bearing surfaces or what is known as a journal and 1 sprocket. These journals and sprockets have a smoother surface compared to other areas which have a rough surface. if a defect occurs on this surface, the camshaft will become worn or there may be noise during operation. this is because the bearings that are attached to the surface of the journal do not rotate properly so that there are obstacles in the rotation of the bearings and have an effect on the entire combustion system or motions.

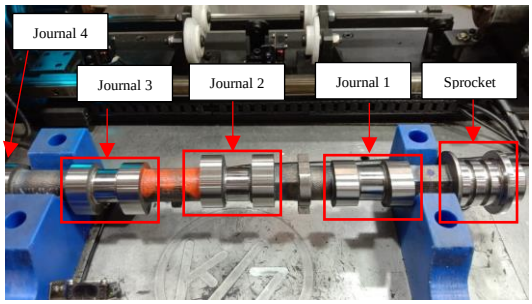

Fig. 1. Camshaft's journal and sprocket location

Camshaft are widely manufactured from wrought steel and nodular cast iron. Forged steel camshaft have a smaller number of cavities and microstructural defects compared to castings. however, defects can still occur in some parts of the camshaft which make the working performance

decrease and camshaft failure. there are various defects in the camshaft, especially in the journal and sprocket. For camshaft made in Japanese manufacturers as shown in Fig. 2 , the name defect is also a Japanese term: (1) suana or burrow on the surface of the journal, (2) Kurosawa or a very small hole that collects a lot in an area on the surface of the journal, (3) dent, (4) lied down scratch along the journal surface. in terms of detection, there are also detection items but they are not classified as defects for further processing. This detection is to see the special holes for the sprocket at the very side of the camshaft. This hole is a place for pushing rods. This detection is done to see if this hole is there. However, if this hole is not detected then the pushing rod cannot be installed there and the camshaft will be rejected or put into the remelting process.

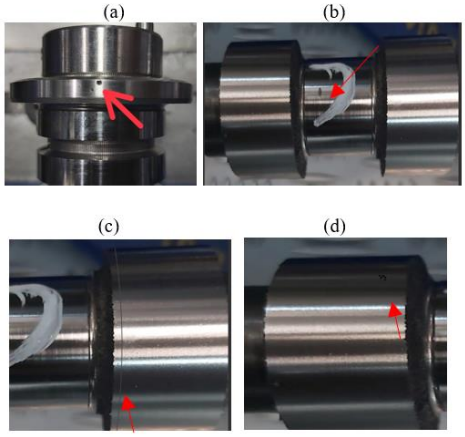

Fig. 2. Example of journal and sprocket defects: (a) suana, (b) dent, (c) scratch, (d) kurokawa

However, in most countries, detecting and classifying camshaft defects is currently done by hand or with expensive sensors. As a result, automatic detection and classification of different types of camshaft defects has recently become popular. Deep learning is also gaining traction and achieving state-of-the-art results in a variety of computer vision tasks [2-3], thanks to recent advancements. As a result, many studies in the literature employ deep learning techniques to detect and classify flaws.

Detecting and classifying defects in camshaft journals and sprockets with deep learning typically entails three steps: (1) collecting image data, (2) creating labels for the data, (3) building deep learning models from the labeled data, and (4) testing the model's performance. Furthermore, providing bounding boxes and labels for camshaft journal and sprocket defects is prone to error and necessitates a significant amount of human labor to produce accurate results. As a result, this research looks into cutting-edge object detection methods in order to find general camshaft journal and sprocket defect detection and classification models that can be used across multiple territories.

II. Visual Inspection Setup

Visual inspection setup is presented in Fig. 3. By using a typical industrial camera placed in a blue dome lighting driven by an industrial end effector's arm robot, the camera can take pictures for each journal and the sprockets at each turn. based on the camshaft circumference and the camera's field of view, 8 turns are obtained to detect all journaling surfaces in one full rotation. The camshaft tested had 4 journals and 1 sprocket, so that a total of 40 images were obtained (8 images for each journal and sprocket).

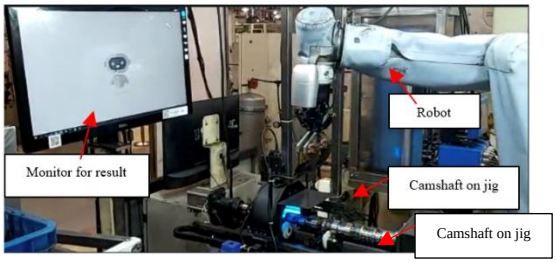

Fig. 3. Visual inspection setup

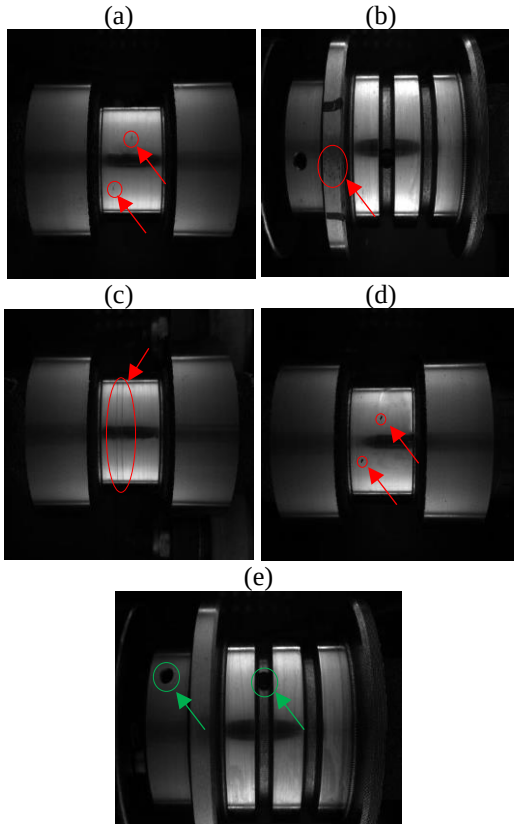

Fig. 4. Example of journal and sprocket defect captured by camera: (a) dent, (b) kurokawa, (c) scratch, (d) suana, and (e) Hole (hole is not defect, if the sprocket is lack of hole, its defect)

The visual inspection system starts with a pick and place process by the robot. Camshafts that have not been inspected will be taken by the robot and then placed in a jig that has been connected to the motor to rotate the camshaft while the camera will take pictures of each journal and sprocket for each rotation. All images captured by the camera will be entered for processing and thrown into the machine learning algorithm. all the defect detection results will be displayed on the screen. Meanwhile, the images

obtained are in the form of black and white images from the 5-megapixel camera. the defects will be seen by this image acquisition method as shown in Fig. 4.

III. Dataset and Labelling Method

To create machine learning, we must first create a model. The model is created through a training process, the goal of which is to create an accurate model that answers and predicts a variety of input data and is associated with specific categories. A dataset is required to create a model [4]. The dataset used in this work is from taking multiple sample images with the same camera setup and lighting every time. A training set, a validation set, and a testing set comprise this dataset. The training and testing sets, in particular, contain defect images from the journal and sprocket of the camshaft, as well as bounding boxes and defect types (for the training set).

It is the process of categorizing raw data (images, texts, videos, etc.) and adding one or more meaningful or informative labels to provide context so that a machine learning model can learn from it in the context of the data labelling or annotation process in machine learning. Furthermore, for this dataset, there are five types of labeled camshaft's journal and sprocket defect types namely scratch, kurokawa, dent, suana and hole. The data for the train set and validation set, respectively, are divided by the portions, while our test set uses defect images outside of the train image data and the validation set. To label the selected image, we use an open-source software, LabelImg. LabelImg is a graphical image annotation tool and label object bounding boxes in images [5]. labelImg uses the box type to annotate several types of objects that you want to name which will be passed to machine learning for training, validation and testing. As shown in Fig. 5, the label annotation results are marked according to the defect type then the annotation file is stored in the same directory. For the type of annotation used, we adapted the MS COCO json format from Microsoft [6].

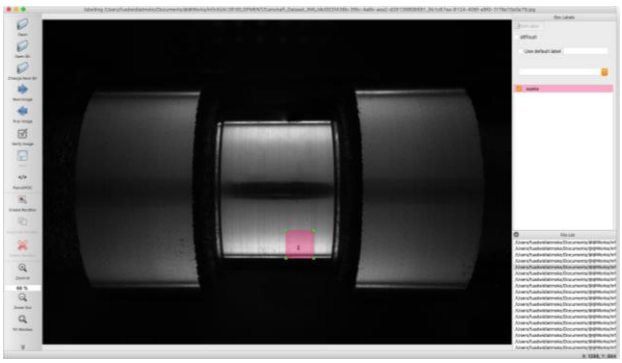

Fig. 5. Image annotation based on defect type in Labellmg

IV. Methodology

Our general methodology is to begin with data exploration to gain a better understanding of the dataset. After that, we divide the training dataset into two sets: training and evaluation. The validation allows us to quantitatively evaluate the hyper-parameters for our architectures. In terms of deep learning model architectures, we start with the Faster R-CNN base model, which is widely used. We're using the detectron2 model zoo as the baseline for a Faster R-CNN algorithm called X101-FPN [7, 8].

A. Data exploration and train, valid, and test splits

This dataset consists of one training set (train), one validation set (valid) and one test set (test). The training set contains 801 images (0, 30, 112, 220, 357, 82 for kurokawa, dent, scratch, suana, hole and random defects respectively). The validation set contain 199 images (0, 7, 17, 71, 78, 26 for kurokawa, dent, scratch, suana, hole and random defects respectively. The one test sets contain 49 images, correspondingly. Kurokawa defects do not have the number in 1 image, but this defects occur in random defects (a combination of several defects in 1 image). The dataset contains over 1000 ground-truth labels (bounding boxes and defect types). Fig. 6 depicts the defect type distributions across the entire training and validation set. In general, the scratch defect type has the most images and defects, while the dent defect type has the least.

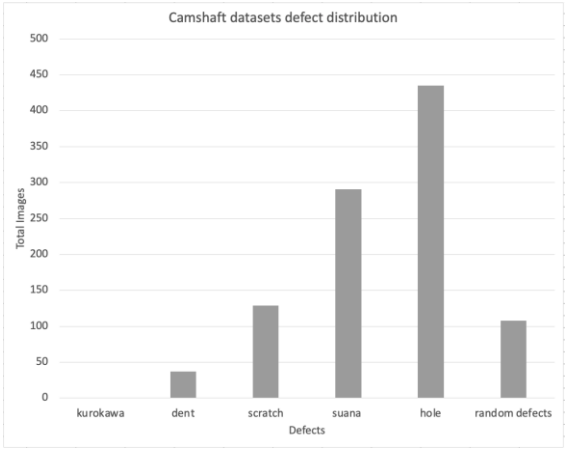

Fig. 6. Defect type distribution ovel all dataset

We split the training dataset into training, validation and test. Specifically, we keep 95% of the images for training (training and validation), and 5% for evaluation (test). Because this set has 49 images, a 5% split for evaluation is reasonable. The performance of the learned models can be evaluated using more than a dozen images. This evaluation set is used to evaluate our model's performance and the hyperparameter selection process during training in a quantitative manner (such as prediction score threshold and number of train iterations). After splitting, the damage types' distributions are shown in Fig. 7. Furthermore, because the train and evaluation distributions are similar, using the origins of damage types as a stratified field seems sufficient.

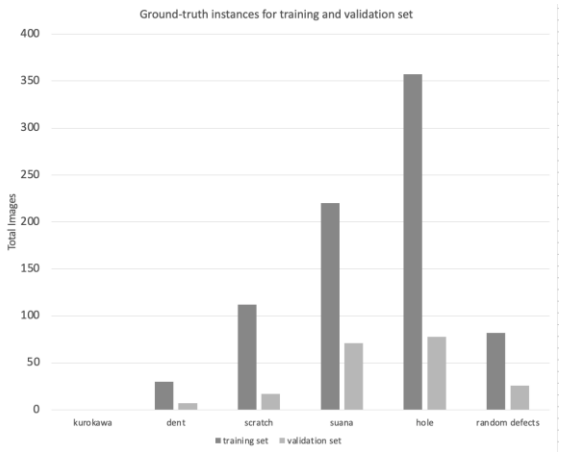

Fig. 7. Defect type distribution over all training and validation set

We ran our tests on a Google Colab notebook, which has free GPU access. It comes with an NVIDIA Tesla P100 GPU and 16 GB of RAM, as well as Python 3.x packages, PyTorch, and the Keras API with a TensorFlow backend pre-installed. They are a collection of tools or modules that are useful for several jobs specifically related to machine learning.

When training a Faster R-CNN model, there are a lot of hyper-parameters to tweak [9]. As a result, exploring all of the configurations is nearly impossible in terms of time and computational resources. As a result, we'll start with the most basic and obvious configurations. Specifically, we set 'cfg.SOLVER.WARMUP_ITERS' (number of iteration) to 1000, 'cfg.SOLVER.IMS PER BATCH' (images per batch) to 4, 'cfg.SOLVER.MAX_ITER' to 2000, 'cfg.SOLVER.BASE LR '(base learning rate) as 0.001, cfg.SOLVER.GAMMA to 0.05 and 'cfg.MODEL.ROI HEADS.NUM CLASSES '(number of classes) is 6 (as correspond to four different types of defects and add 1 for the initial setup). All other configurations are kept as default from Detectron2.

Based on the results of training and process validation, it is obtained using the help of data from the tensorboard [10], that the average precision of the training process and the loss of the training process is shown in Fig. 8. For ranked retrieval results, average precision (AP) is a measure that combines recall and precision. The average precision for a given information need is the average of the precision scores after each relevant document has been retrieved. AP can be defined as:

$$AP = \frac{\sum_r P@r}{R} \quad (1)$$

Where r, $P@r$, and R is ranked retrieval results, precision score after each relevant result, and recall [11]:

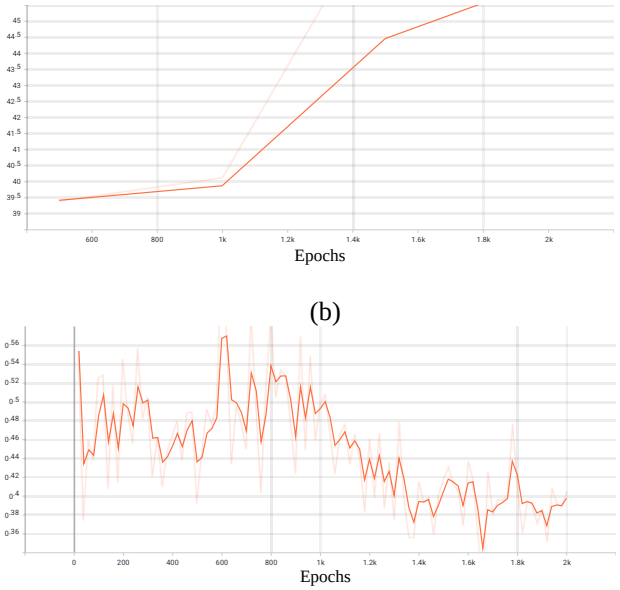

Fig. 8. Training result: (a) average precision hit 48.475 after calculated based on (1) and (b) average loss hit 0.3978

B. *Evaluate Model*

The predicted bounding boxes are also visualized with tensorboard, along with corresponding labels and results scores. Predictions and ground-truth matches are, on the whole, pretty good. The camshaft's journal and sprocket defect detection and localization correctly identifies the defect type and location in most test sets. Fig. 9 shows some sample detection and correct defect location. The results obtained show that the model built can almost detect all types of defects with a confidence level above 66% (Since we set the minimum threshold setting to be able to enter the detection defect of 50% or in detectron2 config is `cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.5` and some even have a confidence level of up to 99%. This shows that the results of the dataset and annotation process that have been carried out, as well as by using Faster R-CNN with baseline X101-FPN get quite optimal results even though there are some discrepancies. We also discovered a few inconsistencies, as well as some incorrect detection and too many ground-truth bounding boxes. Fig. 10 shows a few of these discrepancies. The built model still cannot detect a big suana defect because in the test set, the suana that occurred was indeed very large and shaped like a hole, so the detection results were not correct (recognized as a hole). This too much ground-truth bounding boxes occurs when a defect is detected by many bounding boxes among them. however, in terms of detection and classification based on test set, the predictions are always correct.

(a)

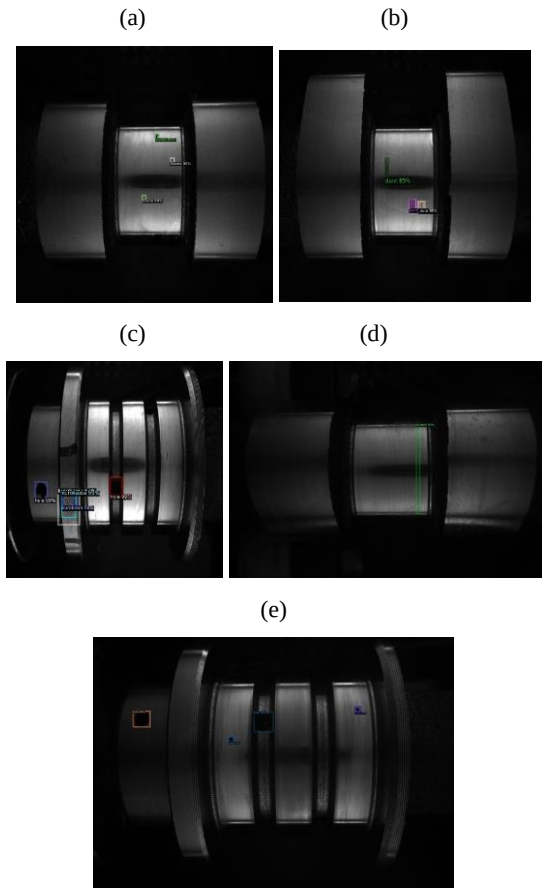

Fig. 9. Example of correct prediction detection and classification of defect in various given journal and sprocket test sets: (a) suana, (b) dent, (c) kurokawa with hole (hole is not defect), (d) scratch, and (e) combination image with multi detections (hole is not defect, if the sprocket is lack of hole, its defect)

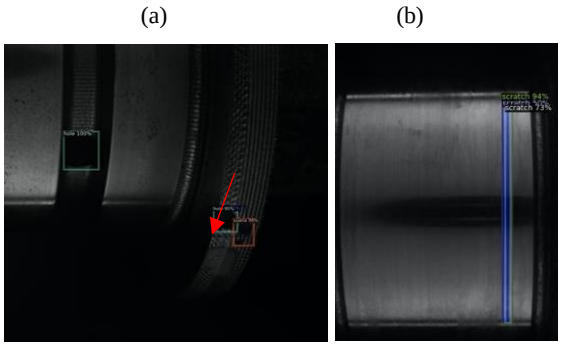

Fig. 10. (a) Example of cases where machine learning model result cannot detect suana but it is detected as a hole because they are similar in shape but different in size. (b) machine learning model result is overfit, one scratch is identified by 3 bounding boxes

V. Conclusion

This paper investigates various state-of-the-art object detection methods and their applicability to the detection and classification of camshaft journal and sprockets. We used the X101-FPN base model to test Detectron2's Faster R-CNN implementation with the X101-FPN base model. In other words, the results show that using Faster R-CNN with the X101-FPN base model and the default configurations of Detectron2 produces good prediction results for these tasks (average correct detection score of approximately 90.4 percent for test sets) and is also general enough to be used in different camshaft journal and sprocket defect detection (note: need to retrain for following discrepancies).

Acknowledgment

The authors gratefully wish to acknowledge the great support provided by Automation, Intelligent System and Robotics Department for Artificial Intelligent laboratory at Infinitigroup, Indonesia and a strong contribution from the Swiss German University department of mechanical engineering. The authors extend their sincere thanks them for the support and the constructive feedback on the work.

References

[1] Y. Zhao, K. Zhang, and J. Di, “Structural design of two-cylinder single overhead camshaft,” IOP Conf. Ser. Mater. Sci. Eng., vol. 301, no. 1, 2018, doi: 10.1088/1757-899X/301/1/012131.

[2] V. Pham and T. Dang, “ScagCNN : estimating visual characterizations of 2D scatterplots via convolution neural network,” Proceedings of the 11th International Conference on Advances in Information Technology, pp. 1-9, Bangkok, Thailand, July 2020.

[3] A. A. D. Nugroho, E. Sofyan, D. Hendriana, E. A. Wibowo, F. Widiatmoko, S. D. Panggayuh, Application of an Artificial Neural Network Model to Predict Parameter of Friction Stir Spot Welding on Aluminum Sheet, 3rd Proceeding of Conference on Management and Engineering in Industry (CMEI 2021), vol. 3 (1), pp. 7-11, 2021.

[4] S. Rostianingsih, A. Setiawan, C. I. Halim, S. Rostianingsih, A. Setiawan, and C. I. Halim, “ScienceDirect ScienceDirect COCO (Creating Common Object in Context) Dataset for Chemistry Apparatus COCO (Creating Common Object in Context) Dataset for Chemistry Apparatus,” *Procedia Comput. Sci.*, 171, pp. 2445–2452, 2020, doi: 10.1016/j.procs.2020.04.264.

[5] Lin, Tzuta. (2015) "LabelImg." *Online: https://github.com/tzutalin/labelImg/.*

[6] Lin T.-Y., Maire M., Belongie S., Hays J., Perona P., Ramanan D., et al., "Microsoft COCO: Common objects in context fleet D," 13th European Conference, Proceeding Part IV: Computer vision – ECCV 2014, pp. 740-755, Zurich, Switzerland, September 2014.

[7] *Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, “Detectron2,” 2020, [Online]. Available: https://github.com/facebookresearch/detectron2. Accessed:2020-12-03.*

[8] __, “X101-fpn,” https://dl.fbaipublicfiles.com/detectron2/COCO-Detection/faster rcnn X 101 32x8d FPN 3x/139173657/ model final 68b088.pkl, accessed: 2020-12-03.

[9] Detectron2, “Detectron2 model zoo,” https://github.com/

facebookresearch/detectron2/blob/master/MODEL ZOO.md, accessed: 2020-12-03.

[10] D. Mané, et al., (2015) TensorBoard: TensorFlow's visualization toolkit, Online: https://github.com/tensorflow/tensorboard.

[11] Zhang E., Zhang Y. (2009) Average Precision. In: LIU L., ÖZSU M.T. (eds) *Encyclopedia of Database Systems*. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-39940-9_482.