Proc. International Conference on Mechanical, Automotive and Mechatronics Engineering (ICMAME 2023)
29-30 April 2023, Dubai, UAE

A Novel Approach to Generate Dataset for Object Detection in Assembly Lines

Ramesh Kaki
Department of Mechanical Engineering
BITS Pilani, Hyderabad Campus
Hyderabad, India
p20200056@hyderabad.bits-pilani.ac.in

Samarth Soni
Department of Computer Science & Information Systems
BITS Pilani, Hyderabad Campus
Hyderabad, India
f20191274@hyderabad.bits-pilani.ac.in

Sandip Deshmukh
Department of Mechanical Engineering
BITS Pilani, Hyderabad Campus
Hyderabad, India
ssd@hyderabad.bits-pilani.ac.in

Tathagata Ray
Department of Computer Science & Information Systems
BITS Pilani, Hyderabad Campus
Hyderabad, India
rayt@hyderabad.bits-pilani.ac.in

Chandu Parimi
Department of Civil Engineering
BITS Pilani, Hyderabad Campus
Hyderabad, India
parimi@hyderabad.bits-pilani.ac.in

***Abstract* — Quality Assurance (QA) is required to ensure precision in the vehicle Assembly Unit process. The most significant challenge is that manual work is error-prone, and even minor errors can be a problem for a vehicle. This study aims to explore suitable Deep Learning (DL) models to automate various parts of the task well. For the current work, the aim is focused on predicting/detecting points on the chassis accurately. In this article, we elaborated on the process to generate the dataset and the proposed model 'You Look Only Once-v5' (YOLOv5) to identify cross marks on the vehicles. The model architecture and parameters are discussed in-depth and changed to detect and classify marked objects against the chassis background. The accuracy and efficiency evaluations show that the model achieved the top performance in average precision (mAP) of ≥ 98%.**

Keywords — Machine Learning, Deep Learning, Computer Vision (CV), YOLO, Object Detection

I. INTRODUCTION

Nick Bostrom once said, "Machine Intelligence is the last invention humanity will ever need to make." The assembly lines in the Indian industry are usually dependent on humanitarian aid and are thus prone to human error. Therefore, there is an immense requirement for a system that can automate various parts of the assembly unit in the industry. While automating these tasks, we focus on Quality Assurance (QA), i.e., ensuring that the task is performed accurately. This is required because the end product of any assembly line must solve the purpose, notwithstanding whether or not it was built manually or not. If automation results in lower quality, it clearly cannot and won't be used [1]. Within an assembly process, a lot of drilling work is involved in making the chassis frame ready for installation. However, before drilling, the operator will mark the positions of the drill. To automate this process, a system must give assurance by identifying the 'X' marks and their position concerning the origin or baseline [2].

Hence, we are currently building a framework that predicts 'X' marks on the chassis. As explained, the framework to detect 'X' marks will later be used for predicting where to drill holes for fitting nuts and bolts or other components [3-5]. This is a Machine Learning (ML) problem in the domain of Computer Vision (CV). ML is the process of using code to solve a task by making it learn how to solve it rather than providing all the rules. In short, ML involves building a code that takes data as input, learns the context-specific mappings, and uses those to produce a desired output rather than relying on the human-supplied algorithm. The domain of ML has many algorithms suited to learning the mappings for various contexts [6-7]. Deep Learning is one of the major algorithms used for any ML research. [8] Deep Learning is so prevalent partly because the techniques to solve any problem can be learned with enough tweaking of its hyperparameters and structure. [9] The problem of detecting shapes in images comes under Object Detection, which is part of the umbrella field of Computer Vision. For building an effective Object Detection or CV model, 1000 images per class are required [10]. In the case of the current problem, only one class is present: the spatial location of 'X'. Here enters the problem of dataset collection and annotation, which is essential before any ML or DL work can be performed. Doing this process manually is no easy task, as it requires collecting images by making various 'X's on the chassis, clicking pictures from various angles, and then annotating each 'X' in each image with good precision. There is a need to automate even the data collection process.

This need for data motivates any user of ML to find new, effective, and accurate data generation techniques in CV. For this automatic data collection task, two methods can be used, i.e., Generative Artificial Intelligence or a Rule-Based Approach. Generative AI consists of Generative Adversarial

Networks (GANs) and Variational Autoencoders (VAEs) [11], which are used in complex problems like Face Detection where data is scarce, and features are complicated. However, since the required data at present is fairly simplistic, requiring the detection of a single type of object with minimal variation in its shape, it is best to go with Occam's Razor and employ the straightforward strategy of generating images using a rule-based approach. Occam's razor is the problem-solving principle explaining how the simplest solution is the most correct or appropriate solution [12]. Hence, the proposal in this paper can be summarized as follows:

Build a rule-based script that can generate appropriate data (here, images) capturing the variations in the vehicle assembly context and generate annotations for that data. Employing this script, cook a set of 10000 images alongwith their labels, since DL algorithms can scale better as the amount of data increases. [13] Then, build a Deep Learning (DL) model by using images and their corresponding annotations generated in the format required by the model to predict the position of 'X' marks for the domain of Assembly Quality Assurance.

The study in this paper aims to predict 'X' marks on a metal chassis with high accuracy. Manufacturers struggle greatly due to the loss in precision resulting from human error when machines are manually controlled. Human involvement in system safety and the errors on the part of operators and managers have played a significant role in accidents. [14] It becomes difficult to assess and measure these human errors' impact accurately. [15] Also, increased reliance on technology enhances manufacturing efficiency. [16] Based on the principles, structures, and procedures proposed in this framework, it will be helpful to predict the appropriate place to drill automatically.

The composition of the remaining paper is categorized by sections, wherein the "Related work" mentions the research done in the respective area. "Dataset description" and "Object detection" mentions the dataset's particulars, the methodology of our work, and the details of the DL training. "Results" and "Conclusion and Future Analysis" discuss our results, conclude our work and give insights into the possible continuation. "Citations" provides references and citations; later, it may be used to compare our work with some homologous prevailing work.

II. RELATED WORK

Some of the related work is illustrated in the TABLE I 1. From this work we got motivated and implemented a new idea in generating a large dataset.

TABLE I: Some of the work utilized ML algorithms in Manufacturing applications

Authors	**Application Area**	**Type of data used**	**Type of ML algorithm used**	**Key findings/ Conclusions drawn**
Shuhui et al. (2018)	Fault identification /Manufacturing/Assembly	image format data	CNN, t-SNE Technique (distributed stochastic neighbor embedding)	This model achieved a diagnostic accuracy of 98 percent when diagnosing rolling element bearing faults [17].
Vedang et al. (2015)	Fault detection/ Assembly	Video format data	Machine Vision Inspection Method, Gaussian Mixture Models (GMMs) with blob analysis, optical flow, running average	The average running method was chosen as the best because it processed video data sets quickly and detected faults immediately [18].
Ducournau et al. (2004)	Monitoring in Agricultural industry	No Labeled data (cluster)	USL, Image acquisition (computer-aided analysis)	To detect and count germinated seeds on a single picture, the system takes no more than a second or two. An hour between camera shots can be reduced to a few seconds with a computer running at 1 GHz, which is much faster than that [19].
Michael et al. (2019)	Unknown object tracking	image format data	Synthetic depth (SD) Mask R-CNN, Fine-Tuned Mask R-CNN, Euclidean Clustering	SD Mask R-CNN achieves a high-performance rate with a minimum number of iterations [20].
Sergio et al. (2011)	Fruits and Vegetable quality Inspections	image format data	Artificial Neural Network (ANN), ANOVA (Analysis of variance)	Real-time computer vision processing of fruits and vegetables is articulated utilizing CCD cameras for improved performance [21].
Sajjad et al. (2019)	Automatic damage inspection of the fasteners	labeled and cluster data	Mask R-CNN, Resnet 101, Auto encoder , Ensemble	Using supervised and unsupervised ML techniques, this model effectively detects damage to various fastener types during overhaul processes [22].
CHEOL et al., (2020)	Prediction for Manufacturing Factors	labeled and cluster data	ANN, Follows Data Clustering-based Machine Learning (DC-ML), common regression models	DC-ML is an ML technique /algorithm that uses a high-scoring algorithm to predict roll force as well as plate thickness during each pass of rolling. This ensures the highest quality steel plates [23].

Chan et al. ()	Manufacturing	synthetic data	ARENA	creating synthetic manufacturing datasets using a discrete event simulation software named ARENA, and the data size was relatively smaller as compared to our image data [24].
Otto et al. (2013)	Assembly lines	synthetic data	SALBPGen	They were able to gain important insights, and introduced some qualitative metrics for a good generator like flexibility, diversified, ease of description, and randomized[25].
Hafizi et al (2019)	Automotive Manufacturing	image format data	ML and Six Sigma	provided an overview of the quality control method in automotive manufacturing industry, which is related to our domain [26].

A. Pre – Requisites

I. For generating images using logic- and rule- based approaches, we need knowledge of the following:

a. OpenCV, Pandas, Numpy, XML/TXT editing in Python
b. Encoding format of YOLO (which saves in txt format)

II. For implementing Computer Vision algorithms in python, we need knowledge of the following:

a. OpenCV, Pandas, Numpy, PyTorch, Keras, and algorithms-specific libraries
b. Maths behind CV and DL algorithms
c. Knowledge of how to implement the above algorithms using custom data

III. DESCRIPTION OF DATASET

The dataset is generated with a rule-based approach, as illustrated in Fig. 1. It involves critical analysis and planning to design an algorithm before it generates synthetic data. It requires testing various methodologies of importing and editing images in Python. Since we used a Python script being executed on a cloud kernel, the dataset generation was of O(n) time complexity, i.e., the total time linearly increased with the increase in the number of required output images. Although a Python script has speed limitations due to its being a general-purpose, interpreted, dynamically typed, high-level, garbage-collected language, this did not affect our study because speed is not a focal point for our study. Things like using C-language inbuilt Python libraries and reducing I/O operations decrease running time. Also, even though there can be some design restrictions and increased runtime errors in Python due to its being dynamically typed, we avoided this partly due to our experience in programming in Python and partly by following best practices such as planning on the requirements and architecture before coding and having a Functional and Object Oriented approach. As a proof of concept, the time the script took for 1000 images was ~32 minutes, which is affordable and not a huge overhead.

Hence using this approach, we have drawn X marks of appropriate size and dimension at random places, as shown in Fig. 3, on a pre-set background image as illustrated in Fig. 2, and also noted the coordinates of mark positions.

The background image must have enough variations covering all the cases found in real-life scenarios. Hence, we use multiple real-life images captured from different angles of the chassis prototype we fabricated. Within a real-life chassis, drilling would be performed only in a sub-portion of the entire surface area, so we fix the boundaries within which to generate 'X' marks. We then generate a random set of (x,y) tuples for each image and use Python libraries to draw 'X' marks centered at those pixel locations. Once these images with 'X' marks were generated, annotation files were generated corresponding to each image, as shown in Fig. 4. Each DL algorithm designed for Object Detection uses a specific annotation format that captures the coordinates of rectBoxes and the object's class. Hence, we designed the script correspondingly by correctly calculating the values of the annotation coordinates for each X mark in the generated images. Then, text files have been appropriately edited by entering the calculated values at the corresponding locations in the file from where the model will read them. Once these annotation files have been generated, a manual verification has been conducted to ensure that all annotations properly label the X marks as expected. Likewise, a total of 10000 images were generated for training.

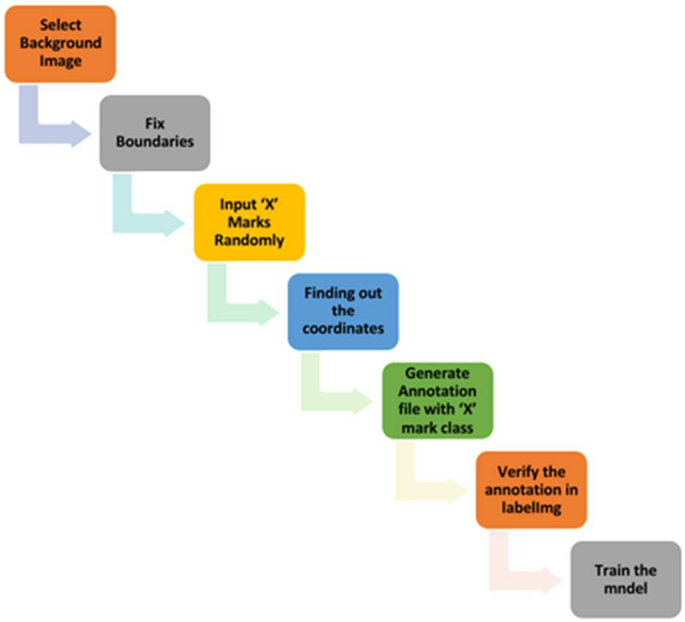

Fig. 1. Process to Generate the Dataset

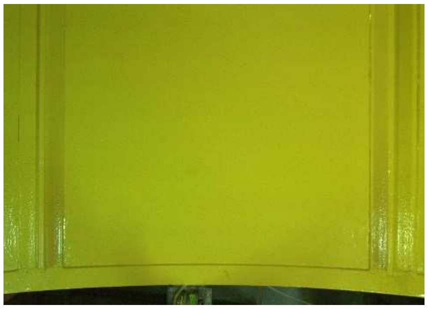

Fig. 2: Background image, it replicates most of the vehicles

Fig. 3: Randomly placed ‘X’ Marks on a plain background

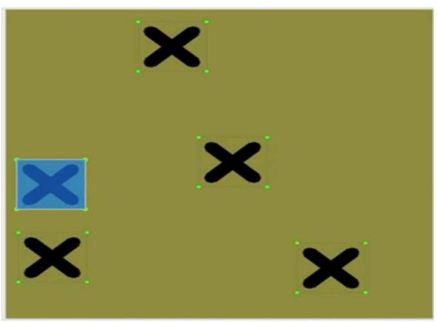

Fig. 4: Annotation Object with the help of saved coordinates

IV. OBJECT DETECTION ALGORITHM

Traditional classification algorithms still fall short of object identification algorithms’ ability to identify mechanical components in assembly environments. When all factors, such as dataset sizes, image sizes, processing times, etc. were taken into account. There are several reasons why YOLOv5 algorithm may have been chosen over other performing object detection algorithms such as R-CNN, Fast R-CNN, etc., for a problem of single-class object detection suggested by researchers [28]. The YOLOv5 algorithm was likely chosen for its speed, accuracy, simplicity, and flexibility, making it an ideal choice for single-class object detection problems [29]. However, YOLO (You Only Look Once) outperformed other approaches in some cases, as suggested by the authors. Due to its speed and accuracy, YOLO is one of the most well-known object detection techniques. The initial release of YOLO, made by Redmond and others in 2016, was hailed as a significant advancement in the real-time detection and tracking of objects. Ultralytics has just released the most recent version of YOLO. Even though its name has been the subject of discussion, it is frequently referred to as YOLOv5.

Regarding the issue of small items in groups, we acknowledge that this is a challenging problem in object detection, and it is an active area of research in the computer vision community. However, we would like to emphasize that our study focuses on detecting cross marks on vehicle chassis, which are not typically small items in groups. Therefore, that the limitations of YOLO in detecting small items in groups are not directly relevant to our study. The backbone of this work is CSPNet (Cross Stage Patial Network), which has a quick processing speed. Path Aggregation Network (PANet) is utilized as a model neck to construct feature pyramids that can fit various object sizes. Class probabilities, bounding boxes, and objectiveness scores are calculated using anchor boxes in the same manner as previous versions. All relevant versions of the robust Deep Learning-based object detection model YOLOv5 (YOLOv5n,s,m,l,x) depend on the system configurations to complete early training and accuracy. So based on the system configurations, we have used YOLOv5m for training. To train the model, HPC machines of BITS Pilani, Hyderabad, running Linux with 32GB GPUs, were utilized [27]. The model was set to 1000 epochs with a momentum of 0.937%. The learning rate was set to 0.01, optimal for small batch sizes and quick convergence, and the weight decay was set at 0.0005. An RoI was deemed positive if its IoU ground-truth box exceeded 0.2. If not, it was considered negative. After each training cycle, the weights from the previous epoch are preserved, allowing training to be paused and resumed at any time.

V. RESULTS AND DISCUSSION

Since YOLO was trained on the HPC Sharanga with a much more available GPU, it could process a batch size of 8, i.e., eight images in parallel. Hence, it completed 400/999 epochs in 24 hours of continuous training. Fig. 5 shows the mAP (mean Average Precision) achieved by YOLOv5 in predicting the ‘X’ marks. YOLO calculates the accuracy by comparing the predicted bounding boxes with the ground truth bounding boxes. The intersection over union (IoU) metric is used to determine the overlap between the predicted and ground truth bounding boxes. If the IoU is greater than a threshold value (usually 0.5), the predicted bounding box is considered a correct detection. The accuracy is then calculated as the ratio of correct detections to the total number of ground truth objects. Fig. 6 shows the loss observed by the model concerning the epochs.

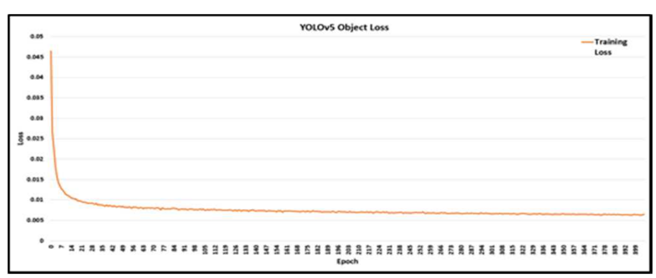

Fig. 5: Training Accuracy vs Epochs

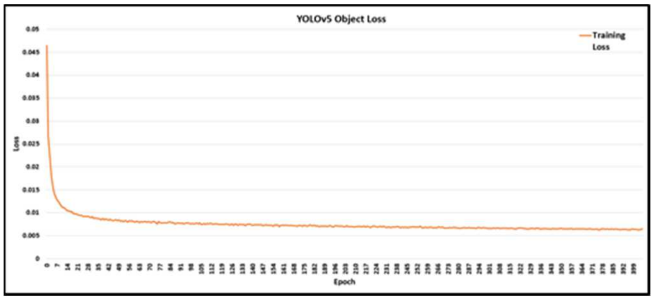

Fig. 6: Training Loss vs Epochs

Since the data we used was very simplistic (just X marks on a fairly plain background), our models could quickly learn the representations and were able to detect the crosses on test data correctly with a 10FPS real-life video feed of the unit, as shown in Fig. 7. The model is also tested on real-life data, as shown in the Fig. 8, and has performed well, with the ‘X’ Marks being identified with acceptable confidence. This was likely owing to the similarity between these test images to those the model trained on, notwithstanding that the training points are generated randomly, whereas the test points were physically drawn on the chassis. Moreover, a sufficient amount of variability in our synthetic dataset was incorporated in the background images. We also ensured that the annotation boxes were tightly fitted to the objects of interest and avoided including any unnecessary space within the boxes. These measures helped to reduce the risk of overfitting in YOLO model.

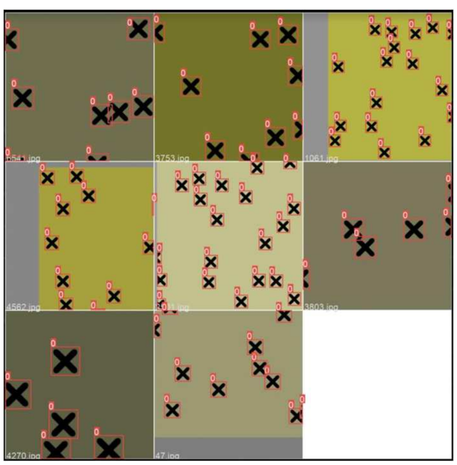

Fig. 7: Scatter collages provided by the model

Fig. 8: Predicted images with trained model on real life problem

VI. CONCLUSIONS AND FUTURE SCOPE

This paper presents a novel method for generating synthetic data for a computer vision problem using a rule-based approach, which can be a practical alternative to more complex data generation methods. The YOLO model was used to train the generated dataset and achieved a test accuracy of over 98%. This work showcases that automated data collection and annotation can be a less laborious process when coupled with human monitoring and manipulation, potentially saving research hours and effort in custom-built applications. Based on these results, it can be said that the chosen YOLO model offers object detection with both speed and accuracy. Also, 98% accuracy shows a good indication of the model's ability to detect objects accurately.

Moreover, YOLO is known for its fast inference speed, making it a popular choice for real-time object detection applications. However, still, future work includes finding ways to make synthetic data more unpredictable and closer to real data by including a more varying number of background images with shapes similar to the objects to be detected. Annotating objects more tightly and creating multiple classes can also help the model identify and detect objects more accurately. These improvements will enable the model to perform better even when entirely new shapes are present, preventing false detections and ensuring high accuracy in real-world scenarios. Further research can explore using a hybrid of rule-based and generative deep learning-based methodologies for more complex data generation tasks.

DECLARATIONS

Conflict of Interest: The authors declare that they have no conflict of interest.

ACKNOWLEDGMENT

The authors gratefully acknowledge the computing time provided on the high performance computing facility, Sharanga, at the Birla Institute of Technology and Science - Pilani, Hyderabad Campus.

REFERENCES

[1] J.P. Russell, J.P. What are quality assurance and quality control? Learn about Quality 2017 [cited 2017 15/08]; Available from: (http://asq.org/learn-about-quality/quality-assurance-quality-control/overview/overview.html)

[2] P. Tanisha, John V. Kennedy, and Johan Potgieter. "A comparison of traditional manufacturing vs additive manufacturing, the best method for the job." Procedia Manufacturing 30 (2019): 11-18. (https://doi.org/10.1016/j.promfg.2019.02.003)

[3] S. Lakshmisri. "Machine learning-future of quality assurance." International Journal of Emerging Technologies and Innovative Research (www. jetir. org), ISSN (2019): 2349-5162. (https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3785661)

[4] T. Seiji, Noriyuki Kadoya, Yoshiki Takayama, Tomohiro Kajikawa, Katsumi Shima, Kakutarou Narazaki, and Keiichi Jingu. "A deep learning-based prediction model for gamma evaluation in patient-specific quality assurance." Medical Physics 45, no. 9 (2018): 4055-4065. (https://doi.org/10.1002/mp.13112)

[5] K. Fotios, Vasiliki Balaska, Symeon Symeonidis, Dimitrios Tsilis, Spyridon G. Mouroutsos, Loukas Bampis, Athanasios Psomoulis, and Antonios Gasteratos. "Automating dairy production lines with the yoghurt cups recognition and detection process in the Industry 4.0 era." Procedia Computer Science 217 (2023): 918-927. (https://doi.org/10.1016/j.procs.2022.12.289)
[6] F. Rita, João Barroso, and Vítor Filipe. "Conformity Assessment of Informative Labels in Car Engine Compartment with Deep Learning Models." In Journal of Physics: Conference Series, vol. 2278, no. 1, p. 012033. IOP Publishing, 2022. (https://iopscience.iop.org/article/10.1088/1742-6596/2278/1/012033/meta)
[7] L. Qinghua, Junmeng Lin, Lufeng Luo, Yunzhi Zhang, and Wenbo Zhu. "A supervised approach for automated surface defect detection in ceramic tile quality control." Advanced Engineering Informatics 53 (2022): 101692. (https://doi.org/10.1016/j.aei.2022.101692)
[8] B. Mahesh, "Machine learning algorithms-a review." International Journal of Science and Research (IJSR).[Internet] 9 (2020): 381-386. (10.21275/ART20203995)
[9] K. Kenji, Leslie Pack Kaelbling, and Yoshua Bengio. "Generalization in deep learning." arXiv preprint arXiv:1710.05468 (2017). (https://doi.org/10.48550/arXiv.1710.05468)
[10] K. Ramesh, E. V. Ramana, L. Srikanth, C. Sri Harsha, and N. Kiran Kumar. "Identification of SMAW Surface Weld Defects Using Machine Learning." In Recent Advances in Materials Processing and Characterization, pp. 339-350. Springer, Singapore, 2023. (https://doi.org/10.1007/978-981-19-5347-7_28)
[11] M. Faisel, Kaki Ramesh, Sandip Deshmukh, Tathagata Ray, Chandu Parimi, Praveen Tandon, and Pramod Kumar Jha. "Nuts&bolts: YOLO-v5 and image processing based component identification system." Engineering Applications of Artificial Intelligence 118 (2023): 105665. (https://doi.org/10.1016/j.engappai.2022.105665)
[12] D. Pedro. "The role of Occam's razor in knowledge discovery." Data mining and knowledge discovery 3 (1999): 409-425. (https://doi.org/10.1023/A:1009868929893)
[13] H. Joel, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. "Deep learning scaling is predictable, empirically." arXiv preprint arXiv:1712.00409 (2017). (https://doi.org/10.48550/arXiv.1712.00409)
[14] R. Jens. "Human error and the problem of causality in analysis of accidents." Philosophical Transactions of the Royal Society of London. B, Biological Sciences 327, no. 1241 (1990): 449-462. (https://doi.org/10.1098/rstb.1990.0088)
[15] L. Hunszu, Sheue-Ling Hwang, and Thu-Hua Liu. "Economic assessment of human errors in manufacturing environment." Safety Science 47, no. 2 (2009): 170-182. (https://doi.org/10.1016/j.ssci.2008.04.006)
[16] M. Anil, and Arunkumar Pennathur. "Advanced technologies and humans in manufacturing workplaces: an interdependent relationship." International journal of industrial ergonomics 33, no. 4 (2004): 295-313. (https://doi.org/10.1016/j.ergon.2003.10.002)
[17] W. Shuhui, Jiawei Xiang, Yongteng Zhong, and Yuqing Zhou. "Convolutional neural network-based hidden Markov models for rolling element bearing fault identification." Knowledge-Based Systems 144 (2018): 65-76. (https://doi.org/10.1016/j.knosys.2017.12.027)
[18] C. Vedang, and Brian Surgenor. "A comparative study of machine vision based methods for fault detection in an automated assembly machine." procedia manufacturing 1 (2015): 416-428. (https://doi.org/10.1016/j.promfg.2015.09.051)
[19] D. Sylvie, A. Feutry, P. Plainchault, P. Revollon, Bertrand Vigouroux, and M. H. Wagner. "An image acquisition system for automated monitoring of the germination rate of sunflower seeds." Computers and Electronics in Agriculture 44, no. 3 (2004): 189-202. (https://doi.org/10.1016/j.compag.2004.04.005)
[20] D. Michael, Matthew Matl, Saurabh Gupta, Andrew Li, Andrew Lee, Jeffrey Mahler, and Ken Goldberg. "Segmenting unknown 3d objects from real depth images using mask r-cnn trained on synthetic data." In 2019 International Conference on Robotics and Automation (ICRA), pp. 7283-7290. IEEE, 2019. (https://doi.org/10.1109/ICRA.2019.8793744)
[21] C. Sergio, Nuria Aleixos, Enrique Molt√≥, Juan G√≥mez-Sanchis, and Jose Blasco. "Advances in machine vision applications for automatic inspection and quality evaluation of fruits and vegetables." Food and bioprocess technology 4, no. 4 (2011): 487-504. (https://doi.org/10.1007/s11947-010-0411-8)
[22] T. Sajjad, Ralf Schoenfeld, and Bernd Bruegge. "Automatic damage detection of fasteners in overhaul processes." In 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE), pp. 1289-1295. IEEE, 2019. (https://doi.org/10.1109/COASE.2019.8843049)
[23] P. Cheol Young, Jin Woog Kim, Bosung Kim, and Joongyoon Lee. "Prediction for manufacturing factors in a steel plate rolling smart factory using data clustering-based machine learning." IEEE Access 8 (2020): 60890-60905. (https://doi.org/10.1109/ACCESS.2020.2983188)
[24] K.C. Chan, Marsel Rabaev, and Handy Pratama. "Generation of synthetic manufacturing datasets for machine learning using discrete-event simulation." Production & Manufacturing Research 10, no. 1 (2022): 337-353. (https://doi.org/10.1080/21693277.2022.2086642)
[25] O. Alena, Christian Otto, and Armin Scholl. "Systematic data generation and test design for solution algorithms on the example of SALBPGen for assembly line balancing." European Journal of Operational Research 228, no. 1 (2013): 33-45. (https://doi.org/10.1016/j.ejor.2012.12.029)
[26] Md. Hafizi, Siti Nur Sakinah Jamaludin, and A. H. Shamil. "State of the art review of quality control method in automotive manufacturing industry." In IOP Conference Series: Materials Science and Engineering, vol. 530, no. 1, p. 012034. IOP Publishing, 2019. (10.1088/1757-899X/530/1/012034)
[27] M. Steven, Bram Vanherle, Joris de Hoog, Taoufik Bourgana, Abdellatif Bey-Temsamani, and Nick Michiels. "CAD2Render: A Modular Toolkit for GPU-accelerated Photorealistic Synthetic Data Generation for the Manufacturing Industry." In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 583-592. 2023. (https://doi.org/10.48550/arXiv.2211.14054)
[28] M. Bharat, Navjot Singh, and K. K. Mishra. "Road object detection: a comparative study of deep learning-based algorithms." Multimedia Tools and Applications 81, no. 10 (2022): 14247-14282. (https://doi.org/10.1007/s11042-022-12447-5)
[29] Wang, Xu Annie, Julie Tang, and Mark Whitty. "Data-centric analysis of on-tree fruit detection: Experiments with deep learning." Computers and Electronics in Agriculture 194 (2022): 106748. (https://doi.org/10.1016/j.compag.2022.106748).

.