Real-time Object Detection for Smart Connected Worker in 3D printing

Shijie Bian[1,3][0000−0003−2814−9854], Tiancheng Lin[1][0000−0002−1026−6342], Chen Li[1,3][0000−0002−4678−1418], Yongwei Fu[1,2][0000−0003−2136−9352], Mengrui Jiang[1][0000−0001−7557−0997], Tongzi Wu[1][0000−0001−5292−9469], Xiyi Hang[4][0000−0001−5292−9469], and Bingbing Li*[1,2][0000−0002−4010−2621]

[1] Autonomy Research Center for STEAHM, California State University, Northridge, CA 91324, U.S.
[2] Department of Manufacturing Systems Engineering and Management, California State University, Northridge, CA 91330, U.S.
[3] Department of Mathematics, University of California, Los Angeles, CA 90095, U.S.
[4] Department of Electrical and Computer Engineering, California State University, Northridge, CA 91330, U.S.

Abstract. IoT and smart systems have been introduced into the advanced manufacturing, especially 3D printing with the trend of the fourth industrial revolution. The rapid development of computer vision and IoT devices in recent years has led the fruitful direction to the development of real-time machine state monitoring. In this study, computer vision technology was adopted into the Smart Connected Worker (SCW) system with the use case of 3D printing. Specifically, artificial intelligence (AI) models were investigated instead of discrete labor-intensive methods to monitor the machine state and predict the errors and risks for the advanced manufacturing. The model achieves accurate supervision in real-time for twenty-four hours a day, which can reduce human resource costs significantly. At the same time, the experiments demonstrate the feasibility of adopting AI technology to more aspects of the advanced manufacturing.

Keywords: Object detection · Machine state monitoring · Smart connected worker · 3D printing

1 Introduction

1.1 Background and Motivation

New computer chips are constantly refreshing the computing efficiency per unit area within the framework of Moore's Law [1]. At the same time, GPUs designed for parallel computing have accelerated the development of artificial intelligence (AI). Among these achievements in AI algorithms, the development of computer vision models has been particularly remarkable. With the advent of computer

* The corresponding author's email: bingbing.li@csun.edu

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_42

vision models, numerous real-time object detection and classification algorithms have been proposed. In particular, real-time image processing, due to its ability to extract high-level information from digital inputs with great efficiency, is widely used in realms ranging from facial recognition to self-driving cars. Since the processing of visual data is mainly performed by human labor in the current manufacturing realm, computer vision-based real-time monitoring was developed to replace the traditional discrete labor-intensive monitoring. The filtering algorithm with a trained vision model was combined into an automated system capable of monitoring a 3D printer through an internal camera in real-time to recognize objects. This paper describes in detail the comprehensive process including object analysis, data collection, algorithm implementation, model training, and result evaluation. Utilizing this system, the defined 9 working states of the 3D printer were successfully identified with high accuracy and low cost.

1.2 Related Works

Computer vision enables the extraction of features from image and video content for the purpose of identification and inference. Recent advances in this field sparked the development of various algorithms and pushed the manufacturing processes to be more efficient and intelligent [2]. An intelligent system using the computer vision method was tested to raise production efficiency under a cloud-based additive manufacturing setting [3]. Using the computer vision method combined with machine learning techniques, an autonomous system was established to perform powder classification from images containing different powder features [4].

Object detection is one of the techniques within the computer vision field, and much effort has been centered around developing algorithms and methods for its efficient applications in the manufacturing of various kinds. Recent endeavors have exhibited the potential of applying deep learning algorithms in object detection. For example, Convolutional Neural Network (CNN) was demonstrated to outperform traditional methods in extracting features from raw data with promising accuracy [5]. Deep Convolutional Neural Networks (DCNN) is a state-of-art approach for object detection, but it is suggested that further models need to be established to allow real-time scenario applications [4]. To improve the performance of object detection models, a technique of generating synthetic training data to train CNN was proposed, which significantly reduced the training time [6]. An algorithm was introduced to generate high accuracy object detection based on joint impedance control, which resulted in high flexibility of manufacturing devices [7]. In addition, remarkable progress was also made in making the object detection model robust to domain shifting and obtaining high training accuracy and efficiency [8]. A Hybrid Smart Region-Based Detection (SRBD) combined several existing object detection algorithms such as YOLO, R-CNN, to accommodate their shortcomings and obtained a high detection accuracy [9].

In recent years scholars and professionals have also made efforts to deploy object detection methods in smart systems through IoT devices. For instance, an object detection algorithm was employed in IoT-based embedded devices

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_42

[10], while maintaining the minimal impact that comes from the variation of environmental conditions. Automated object detection in urban surveillance systems extracted vehicle license plates from images accurately while reducing data storage in the systems [11]. A deep learning-based object detection system [12] successfully sent keywords through Raspberry Pi to Alexa smart speakers using JSON script, thus realizing the implementation of a camera-based smart speaker system. Object detection was partnered with a robot activity support system to enable everyday activity monitoring and assessment in a smart home environment [13].

1.3 Research Contribution

This paper proposes an automated system for the real-time monitoring of 3D printers. The major contributions in this work are:

Object detection and image processing techniques were adopted from the realm of computer vision and integrated into the advanced manufacturing systems with a use case in 3D printing. Using a carefully selected dataset from diverse experimental configurations, a YOLO-based model was trained and is capable of identifying the exact positions of major components in real-time with promising efficiency and accuracy.

Based on the results acquired from the pre-trained machine learning model, a real-time filtering algorithm for both the identification and classification of the 3D printer's machine states was developed. By constantly recording the positions of 3 major components of the 3D printer, the algorithm serves as a supervisor that checks for faulty behaviors in the 3D printing processes. By recognizing the start of each printing section, and by remembering the past actions of the printer's components, the algorithm can not only identify the current machine status but also predict the future ones.

This work encapsulated the pre-trained model and the filtering algorithm into one automated system for the behavior-supervision and status-monitor of 3D printers. Since all components are working at low cost and in real-time, the proposed work would fit smoothly into the advanced manufacturing systems that are related to 3D printing and may serve as either a replacement of human labor or as an auxiliary unit.

2 Background

2.1 3D Printer Monitoring

For a traditional 3D printer, there are three essential interior components that work together during printing: the extruder that ejects the material for printing, the build plate that serves as the supporting platform, and the motor axis that controls the movement of the extruder.

Meanwhile, during the process of printing a part, a 3D printer must go through the following main states in sequential order: the **initialized state**,

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_42

where the printer starts up and processes the printing command(s); the **testing state**, where all components ready their positions; the **calibration state**, where the extruder calibrates its location and finds the starting position; the **heating state**, where the nozzle and chamber heat up; the **printing state**, where the extruder ejects the model and support materials onto the build plate; the **ending state**, where all components are reset to their original positions. The goal is to predict the 3D printer's machine state in real-time through the position of three essential interior components by analyzing the interior camera image via an object detection algorithm.

2.2 Object Detection Algorithms

Region Based Convolutional Neural Networks (R-CNN) Traditional methods which utilize the CNN [14] are suited for classifying images by extracting features. However, real-world inputs that compress numerous objects with distinct characteristics will render these methods computationally expensive. Therefore, the R-CNN architecture [15] combines a regional selective search with the CNN model to solve this problem. In the R-CNN's architecture, the input image is first segmented into numerous small regions. Based on certain features, such as the similarity in color and texture, these small regions are combined together into larger pieces via a greedy algorithm. After warping these pieces into a single region, Graph Neural Networks (GNN) is applied to extract feature vectors that are later classified by a Support Vector Machine (SVM) [16]. Even though the R-CNN method serves as a promising baseline for object detection, the slow testing process greatly limits it in real-time image analysis. Therefore, other methods, such as the Single-shot Detectors (SSD) have been proposed.

Single-shot Detectors (SSD) Instead of generating proposed regions with a dedicated algorithm, SSD [17] utilize predetermined bounding boxes for training, thus eliminating the time spent on region proposals, and achieving much higher inference speed as compared to the R-CNN model. In terms of structure, SSD generally consists of two main components, as shown in Figure 1: a base neural network for performing general feature extractions, and auxiliary network layers with decreasingly sized filters for the final classification.

During the training process of the SSD, digital images with predetermined bounding boxes around each target object are fed into the first component of the model. By passing through the convolutional and pooling layers of a pre-trained classification neural network, such as the Visual Geometry Group from Oxford (VGG16) [18], features of the input image can be extracted into mappings. For each portion of a feature mapping, default bounding boxes (anchor boxes) with diverse shapes and scales are assigned. Using matching strategies such as the maximization of Intersection over Union (IoU), the anchor boxes most similar to the ground truth are filtered out and treated as positive samples, while the rest are identified as negatives. Finally, by passing the processed data through

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_42

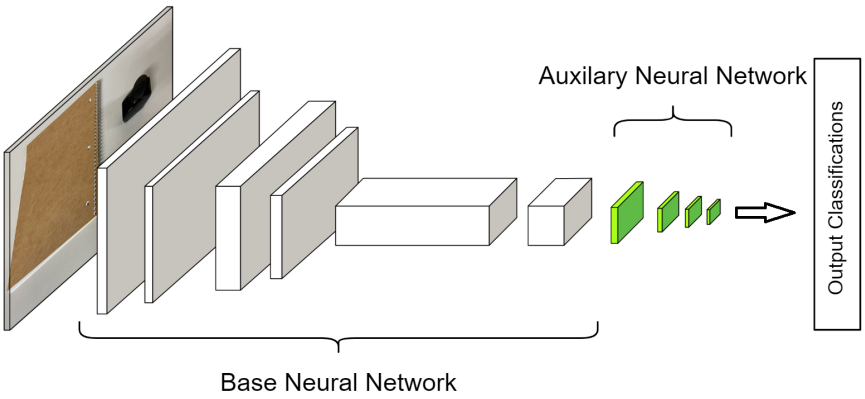


Fig. 1. Illustration of the SSD Structure

an auxiliary set of convolution layers with decreasing sizes, the SSD model is able to produce final classification results for objects of multiple scales.

As a single-shot method, the SSD is able to achieve object detection with high accuracy and competent inference speed. Therefore, the SSD model, namely the You Only Look Once (YOLO) algorithm, was adopted for the construction of a real-time monitoring system.

3 Methodologies

3.1 An YOLO-based Object Detection Model

YOLO [19] is a state-of-the-art real-time object recognition algorithm based on the SSD method. By applying a single neural network to the full input image, YOLO is able to predict both the class and position of bounding boxes within one evaluation, hence achieving great accuracy and efficiency. Therefore, the proposed object detection model is based on the YOLO algorithm for acquiring the locations of three critical components in the 3D printer in real-time.

Data Preprocessing Alongside the digital image that serves as the input, predetermined class labels, as well as bounding boxes, are also fed into the model as the ground truth. Specifically for each ground-truth bounding box, four descriptors were recorded: the center x-coordinate, the center y-coordinate, the width (b_w), and the height (b_h). For each input image encoded in the RGB color model, the scale was standardized by resizing it into 416 $\times$ 416 (unit: pixel) before plugging it into the model. During data preprocessing as shown in Figure 2, a one-hot encoding of the class labels was concatenated after the four descriptors of the ground-truth bounding boxes, and arrive at a single vector consisting of all the information for each of the input images. In the next step,

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_42

the input image was split into 19 × 19 grid cells for identifying the exact location of each detection item. An object of interest is only considered as belonging to a certain cell if its center is within the boundaries of that cell.

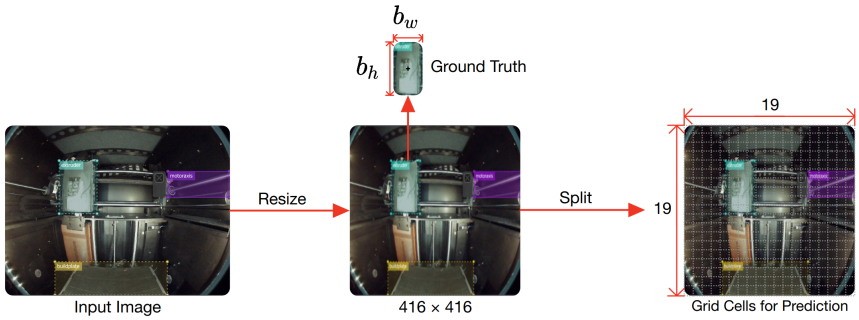


Fig. 2. Illustration of the preprocessing stage

Training For the training process, 3 predefined anchor bounding boxes are assigned to each of the grid cells. Subsequently, the anchor boxes were passed with the predetermined input information into deep CNN architecture of YOLO version 3 [20]. The 13 × 13 scaled feature map was achieved by down-sampling the input image with a stride of 32. For each of the 3 anchor boxes, the probability of containing a certain class of item (i.e., the bounding box probability) using the Intersection over Union (IoU) was extracted. Similar to the previously mentioned steps of an SSD, the anchor boxes most similar to the ground truth are filtered out, while the others are omitted. The same procedure was applied to the 16 × 16 and 8 × 8 down-sampled sizes to capture information of various scales.

Data Output After being fed into the model for inference, the original input image, resized into 416 × 416, is delivered as the output, followed by the predicted bounding boxes acquired from the network. For each object of interest, the following parameters were acquired: the center x-coordinate, the center y-coordinate, the width, the height, and a confidence score measured as a probability that indicates how certain the model believes that the bounding box contains the correctly predicted object. Specifically, the confidence score is calculated by taking the product of the IoU (Intersection over Union between the ground truth and the bounding box) and the probability that an object is in the bounding box (Pr(Object)). This detected information serves as the input to the filtering algorithm for the proceeding process of machine state identification.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_42

3.2 A Filtering Algorithm for Machine State Identification

With the output gained from the YOLO-based object detection model, a filtering algorithm was developed for identifying the machine states of the 3D printer. By comparing the positions of the machine components against all possible combinations, the filtering algorithm is able to filter out the most likely current machine state from all possible machine states. In particular, the following 3 critical components of the 3D printer were primarily focused on the extruder, the build plate, and the motor axis.

Defining the Variables In order to predict the state of the 3D printer, both the precise and relative positions of each and every major component need to be located. Therefore, the following numerical variables for locating the components inside the 3D printer were defined: e_x is the vertical-coordinate of the center of the extruder (identifying its horizontal position), e_y is the horizontal-coordinate of the bottom of the extruder (identifying its height and relative position to the camera), b_x is the horizontal coordinate of the top of the build plate (identifying its height). All of these three variables can be easily calculated from the bounding box outputs of the object detection algorithm.

Predicting the States With the object detection's output, the predefined variables were calculated to predict the 3D printer's machine states. As introduced in the background, 6 main states of the 3D printer (listed in logical order) were mainly predicted: the initialized state, the testing state, the calibration state, the heating state, the printing state, and the ending state. In particular, the printing state, being the most complex one, has four sub-states: the printing and wiping of the support material, as well as the printing and wiping of the model material. For ease of representation, the printing state was considered as one unified state in the subsequent sections.

Since the printing process of almost any 3D model has to go through the same order of steps as introduced in the background, the detailed logic on how to predict the machine states based on the positions of the printer components obtained from the YOLO-based object detection model was provided as follows:

Initialized State : the extruder is at the left-back corner, while the build plate is at the bottom;
Testing State : the extruder remains at the same position, the build plate elevates to the top;
Calibration State : the extruder moves forward, while the build plate remains at the same position;
Heating State : the extruder returns to the left back corner, while the build plate remains at the same position;
Printing State : the extruder moves at the front positions, while the build plate gradually descends;
Ending State : the extruder returns to the left back corner, while the build plate descends to the bottom.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_42

Even though the position of the motor axis is not directly included in the prediction of the machine state, it can be used as an auxiliary piece of information for validation.

For a real-time image acquired from the camera and processed by the object detection algorithm, whether the positions of the printer components match the prior-listed combinations aided us in identifying the machine state of the 3D printer at that exact time. This algorithm was considered as a filter that extracts the machine state from only looking once at the position results obtained from the object detection algorithm.

After predicting the current machine state, the information was stored in a record to retrieve the past states that a 3D printer went through for further processing and analysis, such as energy disaggregation. In addition, by storing the past machine states and following the logical order of the printing steps, the proposed model is able to check whether an unexpected error occurs by predicting the future machine states. Since both our object detection model, as well as our filtering algorithm, runs in real-time, a single workflow can be combined for predicting the machine state of a 3D printer in real-time.

4 Experimental Evaluation

In this section, the YOLO-based object detection model was trained and validated on a carefully selected dataset to assess its feasibility to be utilized in real-time machine state predictions of the realm of manufacturing systems. By testing on real-time data acquired during a real-world working scenario, the accuracy and efficiency of the proposed model were further evaluated.

4.1 Dataset

Camera Setting For collecting image frames from the 3D printer, an SVPRO Fisheye Lens 180 degree USB heat-resistant camera with resolution 1080P and frame rate 30 frames per second (fps) was installed inside the Stratasys uPrint SE 3D Printer. To capture critical objects as needed, the camera was fixed on the interior door of the 3D printer. To increase the image quality, the LED lights inside the 3D printer were kept on with the lights on in the laboratory during image collection. Finally, a python program was developed using the OpenCV package [21] to get frames from the camera at a rate of 5 fps, and then saved as the initial training dataset. This python program also implements "start capturing" and "stop capturing" as a graphical user interface to facilitate data collection.

Image Collection After setting up the dataset collection process, the 3D printer was set up to print a sample model within approximately 8 minutes to finish. When the 3D printer showed "finished", image capturing was stopped right away. As a result, there were a total of around 2400 image frames collected.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_42

However, as shown in Figure 5, there were around 100 images among this training set that was very obscure because of the extremely fast movement of the extruder during the printing process, which made them invaluable for training. To filter a valid training dataset, these images were dismissed.

4.2 Training and Experimental Configurations

Image Labeling For each output image from the training set, the YOLO Visual Object Tagging Tool (VoTT) v1 software was utilized to give the bounding boxes as well as labels that serve as the ground truths. As shown in Figure 3, for each of the 2300 training images, rectangle bounding boxes were drawn for each object on observation of our eyes. The output folder from VoTT contains three parts: the first part is a folder including original images with corresponding text files storing each class's bounding box coordinates; the second part has two text files, which separates the original training dataset into training data and validation data (specifically after performing random shuffle on the entire dataset, the first 80% of the dataset was split and selected for training, while the remaining was selected for validation); the third part is a text file containing the reference of the first two parts.

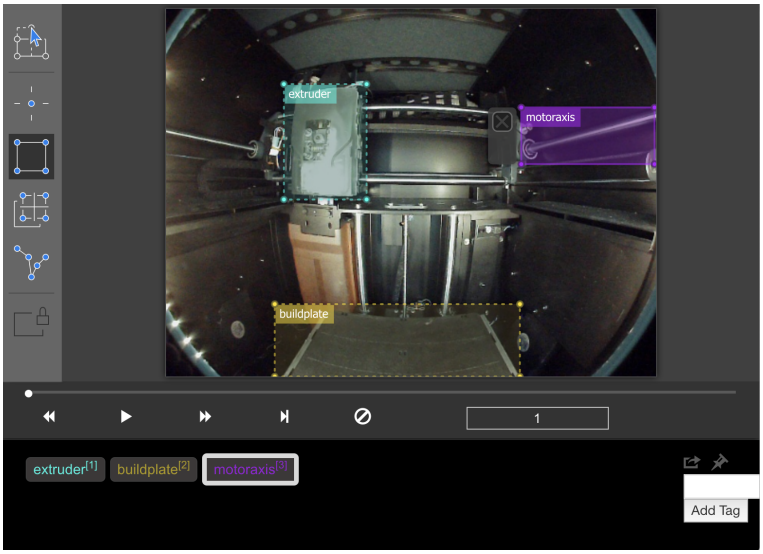


Fig. 3. Training images with the ground-truth bounding boxes and labels.

Model Training Configurations For the training process of the YOLO-based object detection algorithm, the configurations as listed in Figure 4 were adopted.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_42

Furthermore, 4 GPUs from the Pacific Research Platform (PRP) Kubernetes Nautilus cluster cloud platform [22] were deployed to increase the computing speed during training.

Number of Classes: 3	Image Height: 416 pixels	Image Width: 416 pixels
Momentum: 0.9	Batch Size: 64	Decay Rate: 0.0005
Learning Rate: 0.0001	Steps: 100, 25000, 35000	Scales: 10, 0.1, 0.1

Fig. 4. Model Configurations

4.3 Results and Discussion

Object Detection After a training time of 3 hours using the Nautilus GPU cluster cloud platform, the object detection model converged after 10000 iterations. The average class detection accuracy is 0.999 and the average IoU is 0.922. The average inference speed is approximately 0.010 seconds per frame with GPU, which is a sufficient efficiency for real-time object detection.

For testing and validation, a high-resolution sample video from the internal camera of the 3D printer was recorded, which consists of all the stages of the printing process. A total of 2122 image frames were exacted from the video and fed into the trained object detection model. Among the 2122 testing images, the model accurately predicted the bounding box positions of 2011 frames with an average confidence score of 0.87, thus achieving an average prediction accuracy of approximately 94.8%. This accuracy was considered to be sufficient for the subsequent machine state prediction.

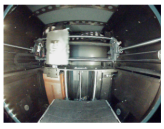
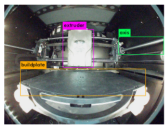
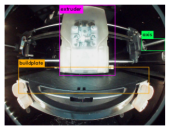
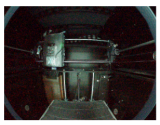
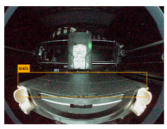
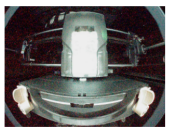
	Training data	Detection result comparison 1	Detection result Comparison 2	Training performance and result:
Light environment training				Converged in 10000 epoches IoU: 0.922
Dark environment training				Converged in 40000 epochs IoU: 0.702

Fig. 5. Training environment comparison

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_42

Machine State Prediction A web-based Graphical User Interface (GUI) was developed for displaying the results obtained from the machine state prediction algorithm, as shown in Figure 6. Specifically, dark blue marks the past machine states that have been recorded, green marks the current machine state, while light blue marks the machine state that should follow the current one. In a well-lit testing environment that is similar to the training configurations of the object detection model, our algorithm achieves a perfect prediction of the machine states, provided that the object detection results are accurate.

Discussion The YOLO-based object detection model achieved an average prediction accuracy of approximately 94.8% as well as a 100 fps frame rate during real-time evaluation. Other object detection models that may serve as candidates for the proposed approach include the traditional R-CNN methods. As demonstrated by Redmon et al. [23] with the Common Objects in Context (COCO) dataset [24], even though R-CNN models may achieve satisfactory or even better detection accuracy, their inference speed seldom exceed 20 fps and is significantly slower than YOLO, which can achieve an inference speed up to 220 fps.

In order to further evaluate the performance of the proposed model, Mask R-CNN [25], a state-of-the-art R-CNN object detection model, was adopted as the baseline. Specifically, the same 2300 training images were labeled and split into the training and validation dataset and were trained in the Mask R-CNN with default configurations. After 50 epochs, the model converged and was tested with the same 2122 image frames that were collected during the experiment. Among the total 2122 test images, 2032 frames predicted all bounding box positions correctly, achieving an average accuracy of approximately 95.8%, which is only slightly higher than the accuracy of the proposed model (94.8%). However, the average inference speed for Mask R-CNN on the test dataset was approximately 0.17 seconds per frame (equivalent to 6 fps), which is significantly slower than the frame rate of the proposed model (100 fps).

As shown in Figure 7, even though Mask R-CNN was able to achieve slightly higher accuracy by performing both bounding box detection and instance segmentation, the actual locations of the bounding boxes produced by both models were very similar. Since the filtering algorithm for machine state prediction only requires accurate bounding box coordinates, the two models' accuracy is approximately equivalent. However, Mask R-CNN's slow inference speed (6 fps) is insufficient for real-time monitoring, whereas the proposed model's competent accuracy and efficiency are more suitable for integration into smart manufacturing systems.

5 Conclusion

In this paper, the object detection model, as well as the filtering algorithm, were developed and serve as a real-time workflow for monitoring the 3D printer's machine states. By feeding real-time image frames directly into a pre-trained YOLO-based object detection model, the positions of the critical components

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_42

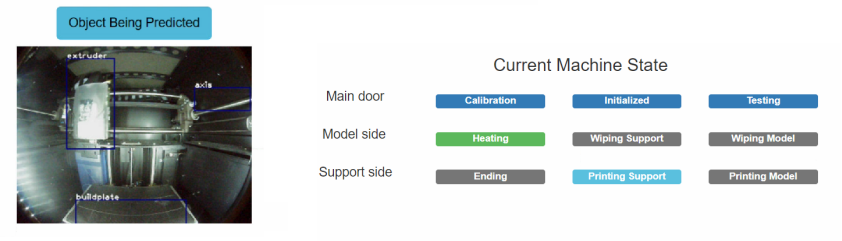


Fig. 6. Machine state predictions as shown on GUI

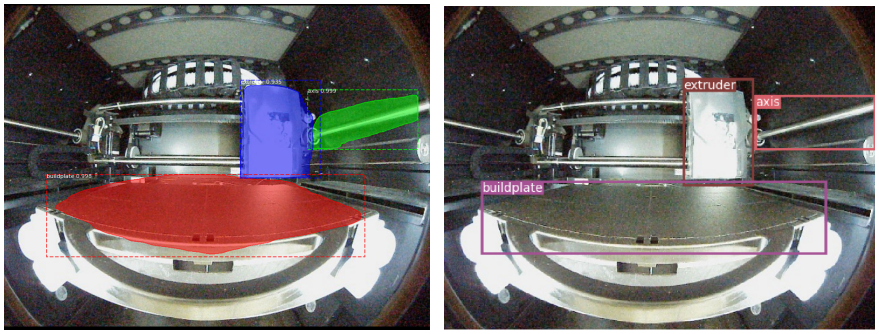


Fig. 7. Test results of Mask R-CNN (left) and the proposed model (right)

of the 3D printer were acquired to predict and record the machine states via the filtering algorithm. The proposed model and algorithm achieved exceptional accuracy and efficiency while testing in an environment that is similar to the training set. Future efforts in this direction would include detecting the machine states of more sophisticated machinery, as well as under more diverse environments. Our work suggests the possibility of adopting state-of-the-art computer vision algorithms for more advanced manufacturing systems, such as the smart connected worker.

6 Acknowledgement

This research was part of the Technical Roadmap Project "Establishing Smart Connected Workers Infrastructure for Enabling Advanced Manufacturing: A Pathway to Implement Smart Manufacturing for Small to Medium Sized Enterprises (SMEs)" supported by the Clean Energy Smart Manufacturing Innovation Institute (CESMII) sponsored through the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Advanced Manufacturing Office Award Number DE-EE0007613.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_42

References

1. R. R. Schaller. Moore's law: past, present and future. *IEEE Spectrum*, 34(6):52–59, 1997.
2. Ben G. Weinstein. A computer vision for animal ecology. *Journal of Animal Ecology*, 87(3):533–545, 2018.
3. Yuanbin Wang, Pai Zheng, Xun Xu, Huayong Yang, and Jun Zou. Production planning for cloud-based additive manufacturing—a computer vision-based approach. *Robotics and Computer-Integrated Manufacturing*, 58:145–157, 2019.
4. Ajeet Ram Pathak, Manjusha Pandey, Siddharth Rautaray, and Karishma Pawar. Assessment of object detection using deep convolutional neural networks. In Subhash Bhalla, Vikrant Bhateja, Anjali A. Chandavale, Anil S. Hiwale, and Suresh Chandra Satapathy, editors, *Intelligent Computing and Information and Communication*, pages 457–466, Singapore, 2018. Springer Singapore.
5. Binbin Zhang, Prakhar Jaiswal, Rahul Rai, Paul Guerrier, and George Baggs. Convolutional neural network-based inspection of metal additive manufacturing parts. *Rapid Prototyping Journal*, 25(3):530–540, 2019.
6. J. Li, P. Götvall, J. Provost, and K. Åkesson. Training convolutional neural networks with synthesized data for object recognition in industrial manufacturing. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1544–1547, 2019.
7. M. Beschi, E. Villagrossi, L. Molinari Tosatti, and D. Surdilovic. Sensorless model-based object-detection applied on an underactuated adaptive hand enabling an impedance behavior. *Robotics and Computer-Integrated Manufacturing*, 46:38–47, 2017.
8. Mehran Khodabandeh, Arash Vahdat, Mani Ranjbar, and William G. Macready. A robust learning approach to domain adaptive object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.
9. R. Anitha and S. Jayalakshmi. A systematic hybrid smart region based detection (srbd) method for object detection. In *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, pages 139–145, 2020.
10. Faisal Mehmood, Israr Ullah, Shabir Ahmad, and Dohyeun Kim. Object detection mechanism based on deep learning algorithm using embedded iot devices for smart home appliances control in cot. *Journal of Ambient Intelligence and Humanized Computing*, 2019.
11. L. Hu and Q. Ni. Iot-driven automated object detection algorithm for urban surveillance systems in smart cities. *IEEE Internet of Things Journal*, 5(2):747–754, 2018.
12. B. Sudharsan, S. P. Kumar, and R. Dhakshinamurthy. Ai vision: Smart speaker design and implementation with object detection custom skill and advanced voice interaction capability. In *2019 11th International Conference on Advanced Computing (ICoAC)*, pages 97–102, 2019.
13. Garrett Wilson, Christopher Pereyda, Nisha Raghunath, Gabriel de la Cruz, Shivam Goel, Sepehr Nesaei, Bryan Minor, Maureen Schmitter-Edgecombe, Matthew E. Taylor, and Diane J. Cook. Robot-enabled support of daily activities in smart home environments. *Cognitive Systems Research*, 54:258–272, 2019.
14. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_42

15. Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, 2014.
16. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.
17. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *Lecture Notes in Computer Science*, page 21–37, 2016.
18. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.
19. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016.
20. S. L. Chen, S. C. Lin, Y. Huang, C. W. Jen, Z. L. Lin, and S. F. Su. A vision-based dual-axis positioning system with yolov4 and improved genetic algorithms. In *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*, pages 127–134, 2020.
21. G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
22. San Diego Pacific Research Platform University of California. Nautilus.
23. Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.
24. Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.
25. Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77970-2_42