Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 00 (2019) 000–000

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

25th International Conference on Production Research Manufacturing Innovation: Cyber Physical Manufacturing
August 9–14, 2019 | Chicago, Illinois (USA)

# A Region-Based Deep Learning Algorithm for Detecting and Tracking Objects in Manufacturing Plants

Muhammad Monjurul Karim[a], David Doell[a], Ravon Lingard[a], Zhaozheng Yin[b], Ming C. Leu[c], Ruwen Qin[a,*]

[a]*Department of Engineering Management and Systems Engineering, Missouri University of Science and Technology, Rolla, MO 65409, USA*
[b]*Computer Science Department, Missouri University of Science and Technology, Rolla, MO 65409, USA*
[c]*Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO 65409, USA*

**Abstract**

In today's competitive production era, the ability to identify and track important objects in a near real-time manner is greatly desired among manufacturers who are moving towards the streamline production. Manually keeping track of every object in a complex manufacturing plant is infeasible; therefore, an automatic system of that functionality is greatly in need. This study was motivated to develop a Mask Region-based Convolutional Neural Network (Mask RCNN) model to semantically segment objects and important zones in manufacturing plants. The Mask RCNN was trained through transfer learning that used a neural network (NN) pre-trained with the MS-COCO dataset as the starting point and further fine-tuned that NN using a limited number of annotated images. Then the Mask RCNN model was modified to have consistent detection results from videos, which was realized through the use of a two-staged detection threshold and the analysis of the temporal coherence information of detected objects. The function of object tracking was added to the system for identifying the misplacement of objects. The effectiveness and efficiency of the proposed system were demonstrated by analyzing a sample of video footages.



*Keywords:* Mask RCNN; Object Detection; Transfer Learning; Temporal Coherence; Two-Staged Detection Threshold

## 1. Introduction

Visually finding an object in a complex manufacturing plant is a basic requirement of various industrial tasks like quality management, packaging, and sorting to name a few. Moreover, according to the industry 4.0 paradigm,

* Corresponding author. Tel.: +1-573-341-4493
*E-mail address:* qinr@mst.edu

monitoring objects and tracking their position in real time are needed for controlling production processes [1]. This capability also facilitates the recognition of human-object interaction, which will help to make machines and components become autonomous and self-organizing, thus reducing the manufacturing complexity [2, 3]. However, manually keeping track of objects lacks efficiency and reliability. Therefore, an automatic system of that functionality is greatly in need.

Automated object detection and tracking in complex manufacturing scenes, however, remains a very challenging task. Researchers adopted some traditional solutions for this task including the use of weight or magnetic sensors [4, 5, 6]. The radio frequency identification (RFID) technology is also commonly used to track objects [7, 8, 9]. This technology requires an RFID active tag attached with the object, a tag reader, and radio communication between them, thus requiring a large initial cost. Meanwhile, to attach tags to every tools and object in a plant is unrealistic. A more practical approach is computer vision that does not require attaching any material or sensors to objects being tracked. Computer vision detects and tracks objects through analyzing the video data, for example, using deep learning [10, 11]. Region-based convolutional neural network (RCNN) has been shown to be effective in detecting and localizing objects in images [12]. Faster RCNN proposed by He et al. [13] is a feature extractor that uses a region proposal network (RPN) to generate region proposals instead of using traditional selective search [14]. The RPN simultaneously regresses region bounding boxes and detection scores of an object. Mask RCNN [15] is an extension of faster RCNN, which performs the region segmentation at the pixel level. Recently, the YOLO [16] and SSD [17] use a single network which do not have the RPN and RoI-pooling layers, thus faster compared to Faster RCNN and Mask RCNN. However, Mask RCNN and Faster RCNN outperform YOLO and SSD in detecting small objects.

Abovementioned deep learning algorithms work well in detecting objects in static images. Yet, results may not be consistent when they are applied to videos frame-by-frame independently. Therefore, the temporal coherence of an object in successive frames has been introduced to address the issue of inconsistent detection [18, 19, 20], wherein the tubelet and optical flow are used to propagate features from one frame to another. However, the approaches in the literature are computationally expensive due to the requirement for repeated motion estimation and feature propagation, making the solution process very slow. Seq-NMS [21] has modification only in the post-processing phase and, thus, it is faster than the algorithms in [18, 19, 20]. Yet Seq-NMS tends to increase the number of false positive detections because it neither put a penalty on these detections nor add additional constraints to prevent the occurrence.

This paper presents a study that extended Mask RCNN by referring to the temporal coherence information of objects in videos and implementing a two-staged detection threshold. The temporal information includes high scoring objects in neighboring frames and their spatial locations. The two-staged detection threshold was introduced to boost up weak detections in a frame by referring to objects with high detection scores in neighboring frames. The spatial locations of these objects were used to prevent the propagation of false positive detections to other frames. This study further created the ability to track the location of any detected object and notify users if the object is not in the right place for it. In implementation of the proposed method, transfer learning [22] was used to adapt a deep learning feature extractor to the application setting. The remainder of this paper is organized as follow: Section 2 delineates the proposed method for object detection and tracking, followed by examples illustrating the implementation of the method. Results from the examples are illustrated in Section 4. Conclusions and future work are summarized at the end, in Section 5.

## 2. Methodology

The proposed framework for the object detection and tracking system is illustrated in Fig. 1. The system can use the plant's own Closed Circuit TV (CCTV) or surveillance cameras to capture videos of the work floor. Video streams of a monitored area are fed to the system. The classifier of the system uses a deep learning algorithm to semantically detect objects in that area. Then, the initial detection result is further refined by referring to the temporal coherence information of objects in videos. The system measures the distance between the location of each detected object and the location for the object in the designated zone. If the measured distance is larger than the pre-specified threshold value for the object, indicating that the object is outside the zone, a notification will be generated and sent to users through an interface. Provided with this system, users can track every object and find the location of it when the

object is misplaced. The deep learning algorithm of object detection and tracking, which is the focus of this paper, is discussed in the following.

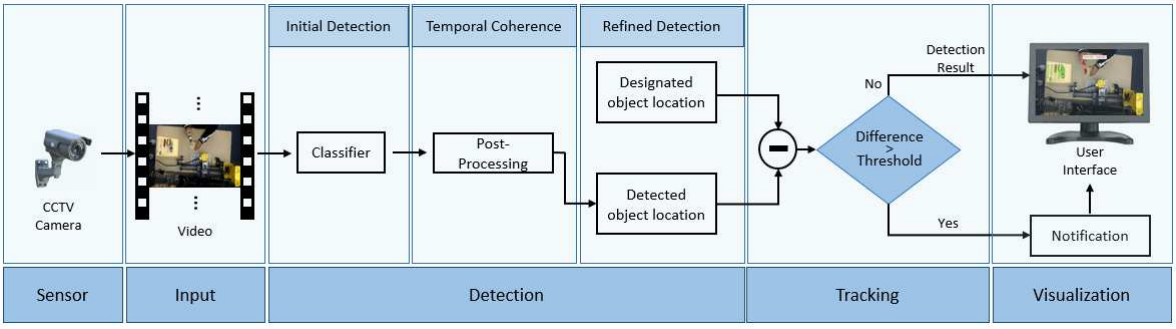


Fig. 1. Schematic diagram of the object detection and tracking system.

### 2.1. Development of a region based deep learning algorithm based on transfer learning

Region-based CNN (RCNN) has been shown to be effective in detecting and localizing objects in images. Mask RCNN, a type of RCNN performing the region segmentation at the pixel level, was chosen as the segmentation tool by this study. Fig. 2 illustrates the structure of Mask RCNN. Having an architecture of ResNet [23] based Feature Pyramid Network (FPN), the backbone of the network is a feature extractor that generates the feature map of each input image. A region proposal network (RPN) creates region of interests (ROIs) and extracts them from the feature map. The extracted feature maps are further aligned with the input image and converted into fixed size feature maps by a layer named Region of Interests Align (RoIAlign). The fixed-size feature maps of RoIs are fed into two independent branches: the network head branch performing classification and bounding box generation, and the mask branch for independently generating instance masks. Interested readers can refer to [15] for details.

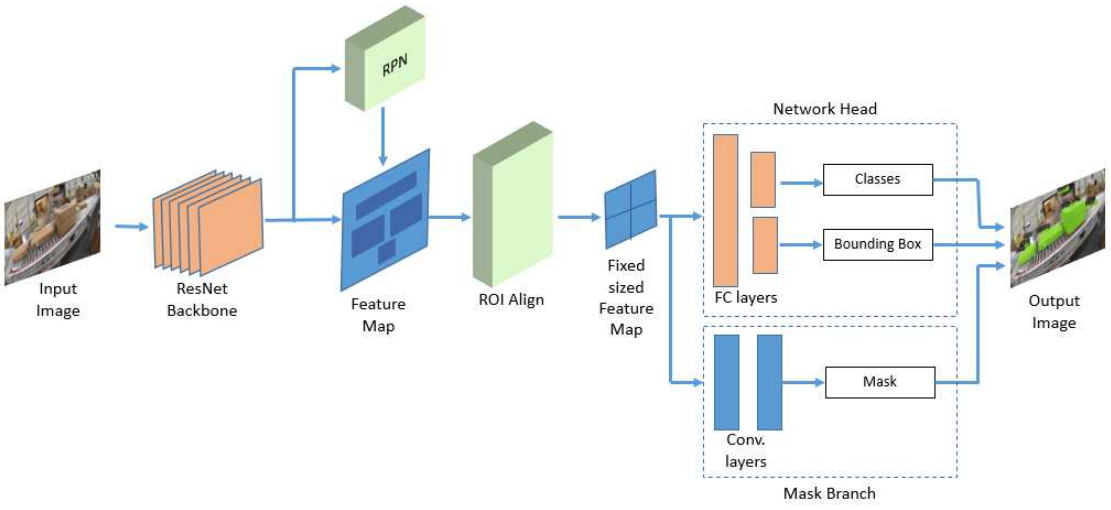


Fig. 2. Architecture of mask RCNN.

In this study, the Mask RCNN was initialized by adopting the ResNet-50 feature extractor [23] whose weights have been pre-trained on the Microsoft COCO dataset [24] that has more than 120,000 labeled images and contains around 1.5 millions of object instances in 80 categories. Then, transfer learning was used to adapt the ResNet-50 feature extractor to the specific setting of this study. Specifically, the ResNet-50 was fine-tuned using a small set of training

images collected from the intended manufacturing application. The ground truth of the training dataset was created by manually annotating the images with class labels. The training was a two-stage process. In the first stage, the network head and the mask branch were trained while all layers before the head were fixed. In the second stage, besides the network head and the mask branch the last few layers of the ResNet Backbone (C5) were trained as well.

### 2.2. Temporal coherence with a two-staged detection threshold

False detections can be reduced by incorporating the temporal coherence information of objects in successive frames. The temporal information used by this study include objects with high detection scores in preceding frames and their spatial locations. The temporal coherence of objects in videos was incorporated in the post-processing phase of the Mask RCNN.

Consider a single video clip that consists of $N$ frames, indexed by $i$. In each frame, the detector returns $M_i$ objects, indexed by $j$. An object in a frame is highly likely present in the neighboring frames within a range of displacement with similar confidence. Under this assumption, a two-staged detection threshold was introduced in this study to propagate detection results from one frame to succeeding frames. Let, $o_{i,j}$ designate object $j$ in frame $i$. The center of the bounding box for $o_{i,j}$ is specified by its coordinates $C_{i,j} := (x_{i,j}, y_{i,j})$. In $p$ frames, $C_{i,j}$ may shift to a surrounding pixel with a spatial displacement of $(p\Delta x, p\Delta y)$ where $p\Delta x$ and $p\Delta y$ are the average displacement on $x$ and $y$ axes, respectively. Fig. 3 illustrates an example wherein a hammer in frame $i-4$ was also shown in the succeeding four frames but with displacements.

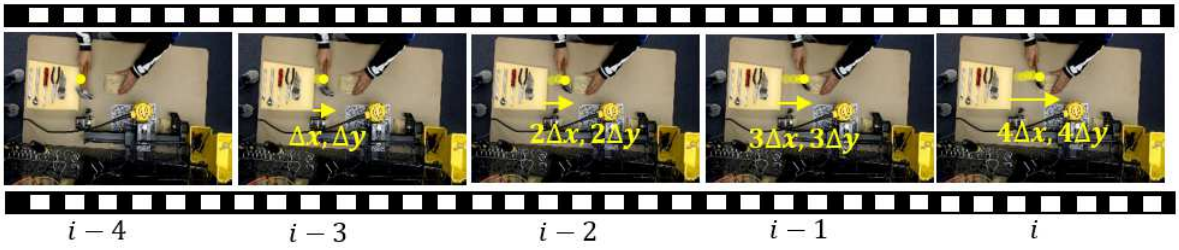


Fig. 3. An illustration of spatial displacements. The yellow dot represents the center location of the bounding box for an object. In each frame it gets displaced by $\Delta d = (\Delta x, \Delta y)$, approximately.

**Algorithm 1** two-staged detection of multiple objects in videos based on the temporal coherence information

**for** $i = 1$ *to* $N$ **do** ▷ $N$ is the number of frames
  **for** $j = 1$ *to* $M_i$ **do** ▷ $M_i$ is the number of objects in $i$ frame
    **if** $S_{i,j} \geq t_u$ **then** ▷ $S$ is the detection score and $t$ is the detection threshold
      Include object $o_{i,j}$ in the set $O_i$ with its detection score, $S_{i,j}$, and the center location, $C_{i,j}$
    **else if** $S_{i,j} \geq t_l$ **then**
      **for** $q = 1, 2, 3$ **do** ▷ $C$ is the center coordinate of detected bounding box
        **if** $o_{i,j} \in O_{i-q} \cap O_{i-(q+1)}$ & $\|C_{i,j} - C_{i-q,j}\|_2 \leq q\Delta d$ & $\|C_{i,j} - C_{i-(q+1),j}\|_2 \leq (q+1)\Delta d$ **then**
          let $S_{i,j} = (S_{i-q,j} + S_{i-(q+1),j})/2$
          Include object $o_{i,j}$ in the set $O_i$ with its $S_{i,j}$, and $C_{i,j}$
          **break**
        **end if**
      **end for**
    **end if**
    Suppress $o_{i,j}$ ▷ Eliminate low scoring object $o_{i,j}$ from the detection list
  **end for**
**end for**

Suppose the detection score for object $j$ in frame $i$ is $S_{i,j}$, and the detection threshold is a range $[t_l, t_u]$. The detector immediately returns a positive detection if $S_{i,j} \geq t_u$. Let $O_i$ be the set of detected objects in frame $i$. The detection score and the center location of these objects in frame $i$, $(S_{i,j}, C_{i,j})$, are stored for analyzing the succeeding four frames. If $t_l \leq S_{i,j} < t_u$, this weakly detected object $o_{i,j}$ is checked by referring to a pair of preceding successive frames up to three times, starting from the nearest pair (frames $i$-1 and $i$-2) to the farthest pair (frames $i$-3 and $i$-4). That is, if $o_{i,j}$ is found in both $O_{i-1}$ and $O_{i-2}$, and the spatial displacements of $o_{i,j}$ from frame $i$-1 and $i$-2 are within $(\Delta x, \Delta y)$ and $(2\Delta x, 2\Delta y)$, respectively, this weakly detected object is added to $O_i$ and the detection score of it is updated by taking the average of $S_{i-1,j}$ and $S_{i-2,j}$. Otherwise, $o_{i,j}$ is searched in $O_{i-2}$ and $O_{i-3}$ and the displacements of $C_{i,j}$ from frames $i-2$ and $i-3$ are measured to determine if it is a positive detection. $o_{i,j}$ will be searched from $O_{i-3}$ and $O_{i-4}$ if needed. If $o_{i,j}$ is not found to be a positive detection after three times of time coherence analysis, it is not reported as a positive detection. It is noticed that searching an object in pairs of successive frames will minimize the risk of progressively propagating false positive detection to succeeding frames. The algorithm of the two-staged process for detecting multiple objects from videos based on the temporal coherence information of objects is summarized as the pseudocode in Algorithm 1.

### 2.3. Object tracking

The bounding box for the designated region for object $j$ is denoted as $R_j := \{w_j,\ h_j,\ (x_j^R, y_j^R)\}$, where $w_j$ and $h_j$ are the width and height of the box, respectively, and $(x_j^R, y_j^R)$ are the coordinates of the center. If the center of the bounding box for object $j$ in frame $i$, $C_{i,j}$, is outside $R_j$, a notification label $P_{i,j}$ is generated:

$$P_{i,j} = \begin{cases} 1, & \text{if } |\ x_{i,j} - x_j^R\ | > 0.5 w_j \text{ or } |\ y_{i,j} - y_j^R\ | > 0.5 h_j \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

## 3. Implementation Details

The proposed method was evaluated through experiments in two manufacturing scenes (a workstation and a production line) and under three different lighting scenarios: normal, underexposed and overexposed lighting conditions. The illumination level at different locations of manufacturing plants such as warehouse, work area, assembly area, and inspection area, can be very different. Therefore, this study tested the impact of lighting condition to detection results.

### 3.1. Experiments and data collection

A workstation was replicated in the lab and a camera was installed on the top of the workstation. The camera captured videos of workers when they operate at that workstation with a frame rate of 30fps. A dataset (A) of 4,405 frames was acquired in this lab setup. Then, this dataset was duplicated to create a new dataset (B) by reducing the brightness of the images by 90% . Similarly, another dataset (C) was created by increasing the brightness level by 70% .

Using a video stream of an actual production line available on YouTube, three more datasets (D, E, and F) were collected. Dataset D is the original video that consists of 400 frames under a normal lighting condition. Datasets E and F were created in the same manner as datasets B and C, respectively. The size of all the image data is 1024 x 578 pixels and the resolution is 96 dpi.

### 3.2. Model training and fine-tuning

The ResNet-50 feature extractor was initialized with weights pre-trained on the Microsoft COCO dataset. The model was fine-tuned using a small set of training dataset composed of 40 images from the workstation dataset. The ground truth data of the training dataset were carefully created by manually annotating the images with 6 class labels, namely hammer, screwdriver, wrench, ratchet, plier, and allen key. In the first stage, the network head and the mask

head was trained for 30 epochs and all the parameters in the previous layers were fixed. In the second stage, in addition with the heads, the ResNet Backbone C5 were trained for 30 additional epochs, and all other layers were fixed. Each epoch consists of 100 training iterations. Stochastic gradient descent was used as the optimizer and the momentum was 0.9. The learning rate was 0.001 for the first 30 epochs of training, and it is reduced to 0.0001 for the remainder 30 epochs of training. The batch size of one image was used on a single NVIDIA Geforce GTX 1080 Ti GPU for this fine-tuning process that took about 14 hours to complete.

The model was further fine-tuned for the production line using a training dataset of 10 images. This dataset has only one class label, package. But the production line had more complex background than the workstation. This time, The network head and the mask head were trained for 25 epochs and all other layers were the same as those obtained from the first stage training for the workstation example.

## 4. Result and discussion

The object tracking system was evaluated on a workstation with the following configuration: a 2.90 GHz Intel Xeon W-2102 CPU with 4 CPU cores, 16GB of RAM and an NVIDIA Geforce GTX 1080 Ti GPU. In the evaluation, the lower boundary of detection threshold, $t_l$, was 0.5 and the upper boundary, $t_u$ was 0.8. To quantify the model effectiveness, a validation dataset of 280 images was created by taking images from the normal, overexposed, and underexposed lighting conditions. 240 out of 280 images were relevant to the workstation scene and the remaining 40 images were from the production line scene. In total 1691 ground truth labels were considered in the evaluation.

### *4.1. Quantitative result*

Intersection over Union (IoU) is the intersection between the predicted bounding box and the ground truth bounding box over the union of them. This ratio was used to determine whether a predicted object can be considered as a correct detection. In the experimental studies of this paper, the IoU value must exceed 0.60 to be considered as a correct detection.

Table 1 compares the object detection ability of Mask RCNN without temporal coherence to the one with temporal coherence under each of the three lightness conditions. Three assessment metrics are used in this comparison:

- Precision: it counts the number of correctly predicted classes out of the total number of predictions
- Recall: it counts the number of correctly predicted classes out of total number of ground-truth objects
- F1-Score: it is the harmonic mean of precision and recall

Table 1. Results on Mask RCNN model and Mask RCNN + temporal coherence model

| Architecture | Illumination level | Precision | Recall | F1 |
|---|---|---|---|---|
| Mask RCNN | Normal | 0.963 | 0.950 | 0.957 |
| | Underexposed | 0.935 | 0.922 | 0.928 |
| | Overexposed | 0.733 | 0.493 | 0.590 |
| Mask RCNN+ Temporal coherence | Normal | 0.991 | 0.979 | 0.985 |
| | Underexposed | 0.993 | 0.929 | 0.960 |
| | Overexposed | 0.727 | 0.500 | 0.593 |

From Table 1 it can be seen that the Mask RCNN model obtained a high precision (96.3% ), recall (95% ), and F1-Score (95.7% ) under the normal lighting condition. These three scores dropped by around 3% under the underexposed lighting condition, and over 20% under the overexposed condition. Adding the temporal coherence information to the Mask RCNN increased the precision for about 3% under the normal lighting condition and 6% under the underexposed condition. The improvements are due to the fact that the temporal coherence information was used for lowering the amount of false positive detections. The improvement of recall was near 3% under the normal lighting condition and only 0.5% under the underexposed condition, indicating the temporal coherence information helps improve the

ability to correctly detect more relevant objects under normal lighting condition. However, the addition of temporal coherence to the Mask RCNN did not improve either the precision or recall under the overexposed lighting condition. This is because edges of objects may not be differentiable from their background under an overexposed condition, as shown in the left column of Fig. 7(b). The detection result may get worse after applying temporal coherence to the object detection if there were false detections in several continuous frames.

To evaluate the pixel-wise accuracy of the proposed algorithm, an overlapping grid of ground truth objects and corresponding predictions was calculated under various lighting conditions. Fig. 4 shows three examples of the overlapping grid with each of them under one of the three lighting conditions. In Fig. 4 the ground truth classes are listed on the horizontal axis, and on the vertical axis the predicted classes are listed in the decreasing order of detection probability. Each grid describes the IoU value of the detected class. It can be seen from Fig. 4(a) and 4(b) that the IoU values for all detected classes are higher than 60% under the normal and underexposed lighting conditions. However, under the overexposed lighting condition, the class wrench is not listed in the vertical axis as its IoU value is lower than 60%. Moreover, the IoU value of 4 classes (wrench, plier, screwdriver, and ratchet) out of 6 classes in this condition is lower than the corresponding values under the other two lighting conditions. The result shows that adding temporal coherence to the Mask RCNN performs well in the pixel level segmentation under both normal and underexposed lighting conditions, but not under the overexposed lighting condition.

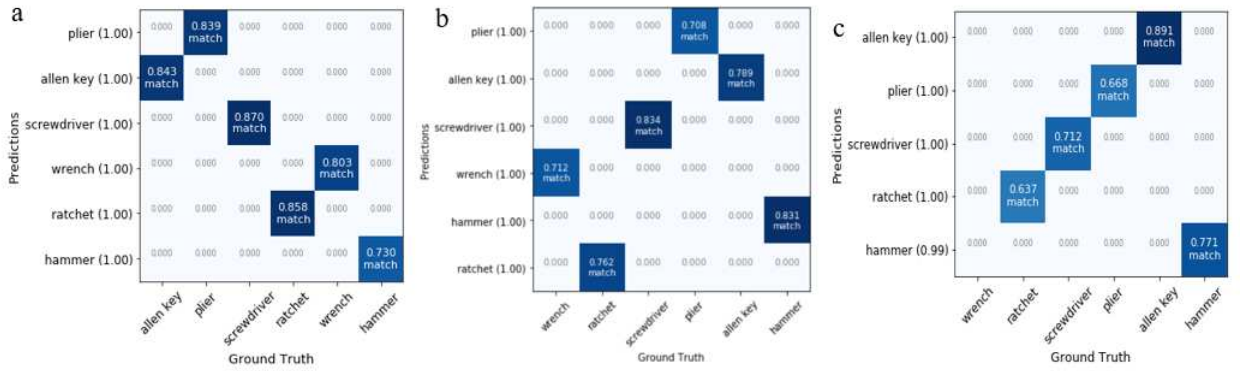


Fig. 4. Overlapping grid between ground truth and prediction under (a) normal, (b) underexposed, and (c) overexposed lighting conditions.

### 4.2. Qualitative result

Fig. 5 illustrates how notifications were generated when objects were moved out of their designated area. The system highlighted an object with a red mask when it was moved to the outside of the designated region for it.

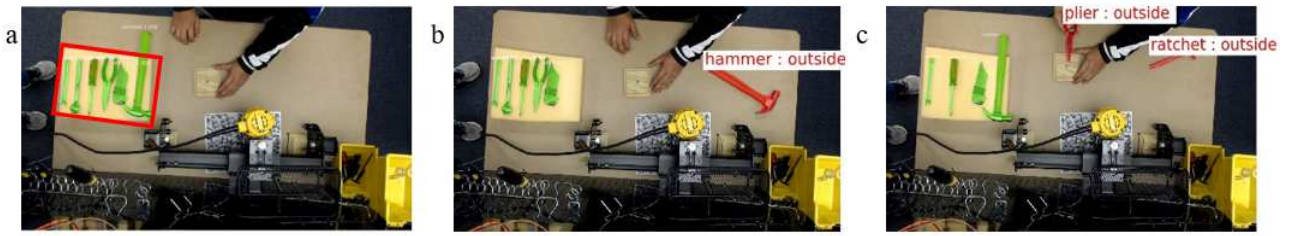


Fig. 5. Setup of the workstation. (a) Six tools are highlighted with green mask. From left to right, the tools are wrench, ratchet, screwdriver, plier, Allen key and hammer. The red box indicates the designated area for these tools; (b) a notification is generated by highlighting the hammer with red mask as it gets out of its predefined designated area; (c) notifications generated for plier and ratchet.

Fig. 6 (a), (b), and (c) show some successful examples of object detection by the proposed detecting and tracking system in the workstation scene under various lighting conditions. Under all three lighting conditions, the proposed system segmented objects successfully. Masks tightly overlapped with the corresponding objects. No obvious false positives were found in those examples. Examples in Fig. 6(d) show that there were no false positive detections in the production line scene. This is because the temporal information was used to suppress their appearance. The system also successfully generated notifications when objects were outside of the designated region.

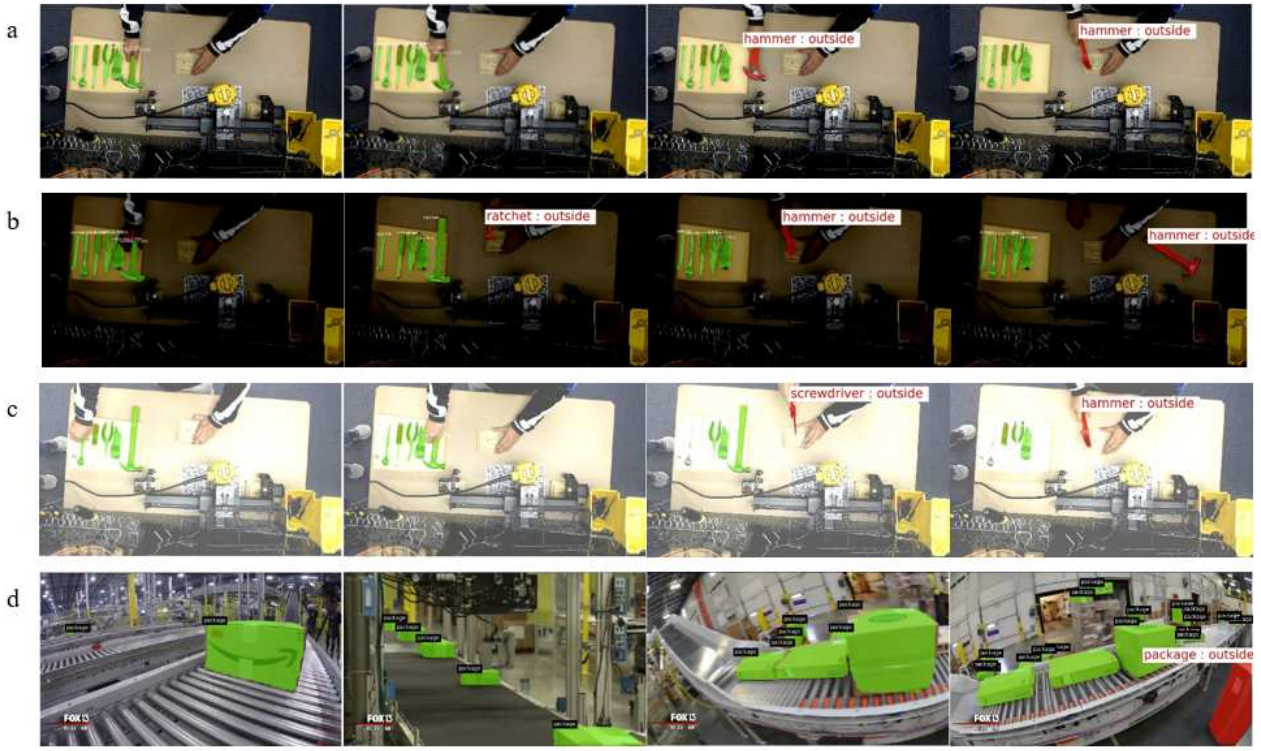


Fig. 6. Successful examples of object detection using the proposed system. (a) and (d) are under the normal lighting condition; (b) is under underexposed lighting condition; and (c) is under overexposed lighting condition.

Fig. 7 illustrates some failed examples. Fig. 7(a) and (b) are examples of false negative detections where some objects were not detected. It is found that false negative detections occurred when there was a motion blur in multiple successive frames, larger than the defined temporal window size. Fig. 7(a) shows such situations where the hammer (in the red bounding box) was not get detected because of motion blur. Moreover, the change of lighting condition may make objects unrecognizable, resulting detection failures. For example in Fig. 7(b), in the overexposed lighting condition edges of wrench and ratchet were not recognizable, and under the underexposed lighting condition the plier handle was not recognizable. As a result the detector can not detect these objects. Fig. 7(c) illustrates another two examples of false positive detections where a screwdriver and a hammer were misclassified as a wrench and a plier, respectively. This is because a single camera can not reveal the full appearance details of objects.

## 5. Conclusion

This paper presents a vision sensor based system for simultaneously detecting and segmenting industrial objects. This ability enables manufacturers to know the exact location of an object. The essence component of this system is an improved Mask RCNN developed in the study of this paper. The post-processing phase of this network was modified to further refine the initial detection result using a two-staged detection threshold and the temporal coherence information

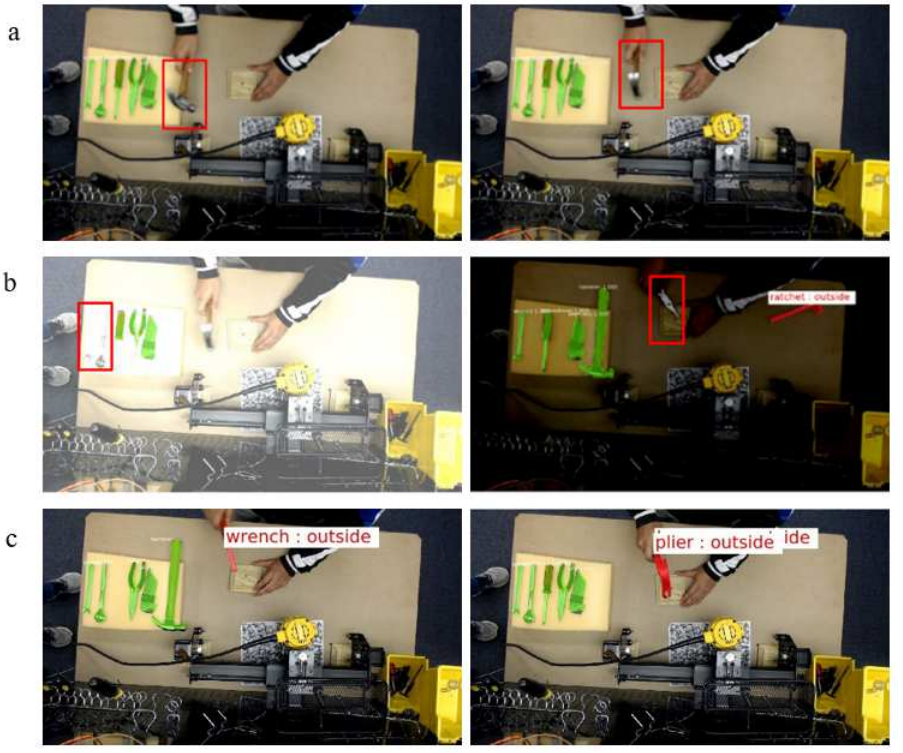


Fig. 7. Failure cases by the proposed method. (a) and (b) false negative detections; (c) false positive detections.

of objects in successive frames. The final detection result of an object was compared with its predefined location to know if a misplacement of the object from its original location was identified.

Results of the proposed algorithm are very promising to be used in real manufacturing settings. This algorithm achieved over 96% F1-score in normal and underexposed lighting conditions. Yet, detection quality needs to be improved under some challenging conditions such as: when motion blur is presented for a relatively long period of time; when the illumination level is too high; and when the camera viewpoint is limited. Future work would be focused on those matters to further refine the detection quality.

## Acknowledgments

This work was supported by the National Science Foundation (NSF) grant CMMI-1646162 on cyber-physical systems. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF. The authors also thank the project team members for their insightful comments and suggestions on this study.

## References

[1] F. Almada-Lobo, The industry 4.0 revolution and the future of manufacturing execution systems (MES), Journal of Innovation Management 3 (4) (2016) 16–21.
[2] D. Gorecky, M. Schmitt, M. Loskyll, D. Zühlke, Human-machine-interaction in the industry 4.0 era, in: 2014 12th IEEE International Conference on Industrial Informatics (INDIN), IEEE, 2014, pp. 289–294.
[3] J. Cannan, H. Hu, Human-machine interaction (HMI): A survey, Tech. rep., University of Essex (2011).
[4] W. C. Maloney, Object tracking method and system with object identification and verification, US Patent 6,707,381 (2004).
[5] T. Paulsen, H. Meyer, F. Arman, System for tracking object locations using self-tracking tags, US Patent 7,119,687 (2006).

[6] S. B. Tantry, R. U. Mashruwala, B. A. Lozier, R. L. Hess, Object-oriented architecture for factory floor management, US Patent 5,398,336 (1995).
[7] K. Ding, P. Jiang, P. Sun, C. Wang, RFID-enabled physical object tracking in process flow based on an enhanced graphical deduction modeling method, IEEE Transactions on Systems, Man, and Cybernetics: Systems 47 (11) (2017) 3006–3018.
[8] M. Liukkonen, RFID technology in manufacturing and supply chain, International Journal of Computer Integrated Manufacturing 28 (8) (2015) 861–880.
[9] J. Brusey, D. C. McFarlane, Effective RFID-based object tracking for manufacturing, International Journal of Computer Integrated Manufacturing 22 (7) (2009) 638–647.
[10] N. Cohen, O. Sharir, A. Shashua, On the expressive power of deep learning: A tensor analysis, in: JMLR: Workshop and Conference Proceedings, Vol. 49, 2016, pp. 1–31.
[11] J. Wang, Y. Ma, L. Zhang, R. X. Gao, D. Wu, Deep learning for smart manufacturing: Methods and applications, Journal of Manufacturing Systems 48 (2018) 144–156.
[12] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[13] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[14] J. R. Uijlings, K. E. Van De Sande, T. Gevers, A. W. Smeulders, Selective search for object recognition, International Journal of Computer Vision 104 (2) (2013) 154–171.
[15] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.
[16] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.
[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, SSD: Single shot multibox detector, in: European Conference on Computer Vision, Springer, 2016, pp. 21–37.
[18] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang, et al., T-CNN: Tubelets with convolutional neural networks for object detection from videos, IEEE Transactions on Circuits and Systems for Video Technology 28 (10) (2018) 2896–2907.
[19] X. Zhu, Y. Xiong, J. Dai, L. Yuan, Y. Wei, Deep feature flow for video recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2349–2358.
[20] X. Zhu, Y. Wang, J. Dai, L. Yuan, Y. Wei, Flow-guided feature aggregation for video object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 408–417.
[21] W. Han, P. Khorrami, T. L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, T. S. Huang, Seq-NMS for video object detection, arXiv preprint arXiv:1602.08465 (2016).
[22] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT press, 2016.
[23] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common Objects in Context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755.