10-2024

Retrofitting a legacy cutlery washing machine using computer vision

Hua Leong FWA
Singapore Management University, hlfwa@smu.edu.sg

Retrofitting a Legacy Cutlery Washing Machine using Computer Vision

Hua Leong Fwa[1[0000−0002−4472−2481]]

Singapore Management University, 81 Victoria St, Singapore 188065
hlfwa@smu.edu.sg

Abstract. Industry 4.0, the digitalization of manufacturing promises to lead to lowered cost, efficient processes and even discovery of new business models. However, many of the enterprises have huge investments in legacy machines which are not 'smart'. In this study, we thus designed a cost-efficient solution to retrofit a legacy conveyor belt-based cutlery washing machine with a commodity web camera. We then applied computer vision (using both traditional image processing and deep learning techniques) to infer the speed and utilization of the machine. We detailed the algorithms that we designed for computing both speed and utilization. With the existing operational constraints of our client, frequent re-training of the deep learning model for object detection is not feasible. Thus, we compared the generalizability of the two techniques across 'unseen' cutleries and found traditional image processing to be generalizable across 'unseen' images. Our proposed final solution uses traditional image processing for computation of utilization but a hybrid of traditional image processing and deep learning model for speed computation as it is more reliable. Our client has implemented our proposed solution for one conveyor belt-based cutlery washing machine and will be planning to scale this to multiple conveyor belt-based cutlery washing machines.

Keywords: Industry 4.0 · Computer Vision · Deep Learning · Image Processing.

1 Introduction

Industry 4.0 is characterized by digitalization of manufacturing premised on the use of technologies such as Internet of Things (IOT), Cloud Computing, Data Analytics and Artificial Intelligence (AI) [12, 11, 17]. With Industry 4.0, the advent of smarter and more autonomous machines promises to make manufacturing more efficient which further leads to optimized processes, lowered cost and even possible discovery of new business models. Notwithstanding the many benefits of Industry 4.0, the move to Industry 4.0 has been challenging for many companies.

Many of these companies have made substantial investments into the legacy machines which are not data ready or 'smart'. However, these legacy machines are still functional and continue to fulfil the operational needs of the companies. The replacement of these legacy machines with smarter ones entails high investment in equipment and technology which may not be feasible especially for small and medium enterprises with tight budgets. Retrofitting or equipping these legacy machines with sensors to enhance their functionality or performance may be a practical solution for these companies to jump on the industry 4.0 bandwagon without huge capital investments [23]. The integration of these new technologies into the companies' traditional processes in the digitalization journey in turn offers new opportunities for these companies to redesign business processes and facilitate data driven decision making.

In our study, we used computer vision techniques for enhancing a traditional conveyor belt-based cutlery washing machine with the capability of tracking the utilization and running speed of the machine. The original conveyor belt-based cutlery washing machine is a legacy machine which cleans cutleries. The various cutleries are first placed on the conveyor belt of the machine and when the machine is switched on, the conveyor belt then moves at three different levels of pre-set speeds (controlled via a knob) to transport the cutleries into a steam cleaning compartment. The actual speed of the conveyor belt varies though with the weight of the cutleries placed on it i.e. the conveyor belt slows to a speed lower than the preset one when heavier cutleries are placed on it. Sensing the actual speed of the conveyor belt however, is secondary as our client wanted instead to monitor prolonged instances of waste and careless usage. These are situations where the cutlery washing machine is running but only some or no cutleries were actually placed on it (waste usage) or where cutleries were placed on the conveyor belt but the machine was intermittently stopped (careless usage). The intermittent starting and stopping of the cutlery washing machine, according to the client, will result in higher incidences of machine failure and thus undesirable. With the use of a commodity web camera, we managed to retrofit the conveyor belt-based cutlery washing machine not only to be data ready but also to accomplish the above stated objectives of our client at a low cost.

Our client wanted to track the utilization as well as the running speed of each of their conveyor belt-based cutlery washing machines to optimize their use and lower the running cost but this cannot be achieved without the installation of additional sensors. Our proposed solution is to use computer vision techniques to process images captured through a web camera installed on top of the conveyor belt of the machine for computing the utilization and speed of the machine. With real-time tracking of utilization and speed of the conveyor belt-based cutlery washing machines, our client would be able to visualize and better optimize both the operations and running cost of the machines.

2 Related Studies

Li et al. [13] adapted the You Only Look Once (YOLO) network [19] for detection of six types of surface defects on steel strips. Using the improved network, they were able to achieve 97.55% mAP and 95.86% recall rate for the six types of defects at a detection speed of 83 FPS.

Pham et al. [18] proposed a real-time packaging defect detection system also based on YOLO network for detecting defects on package boxes moved using conveyor belts. In their study, the authors were able to achieve 78.6% mAP when evaluating the YOLO network over a test set of 40 images.

In another study by Li et al. [14], a traditional digital image processing technique was proposed to recognize congestion on conveyor belts. The authors used a static-edge detection method and then applied statistical techniques to analyse the extracted features of packages for determining whether a congestion had occurred. The authors contended that deep learning techniques would not be adequate for industrial parks with thousands of cameras due to the need for massive computing resources to train the models and the requirement for a massive set of training images. Their results also showed that they were able to achieve higher precision and recall as compared to the performance of deep learning models such as Inception [22] and YOLO.

The study by Liu and Qu [16] compared the use of traditional digital image processing technique against a deep learning based convolutional neural network for the defect classification of PCB boards. The authors compared a traditional classification algorithm based on digital image processing against one using Convolutional Neural Network [7] on a set of 1818 PCB defect images. The results indicated that the algorithm based on convolutional neural network achieved a higher classification accuracy of 95.7% which is higher than the traditional method.

The above studies demonstrated the feasibility of both traditional image processing techniques and deep learning algorithms for object detection in a factory environment. In recent years, deep learning algorithms notably are gaining popularity for use in industrial defect detection due to their ability to achieve higher accuracy. However, as highlighted by Li et al. [14], albeit the higher accuracy achieved by deep learning algorithm, deep learning algorithms may not be adequate in all industrial contexts due to their need for re-training, massive image data sets and high computing resource requirements for training of the models.

In our study, we thus compare between the use of traditional image processing technique versus deep learning algorithm in terms of their generalizability. With higher generalizability of the detection algorithm, we postulate that re-training (arising from new and unseen objects) will not need to be frequent. Finally, we also adapted and formulated algorithms to compute both utilization and running speed of the conveyor belt-based cutlery washing machine from the detected objects on the conveyor belt in this study.

3 Methodology

3.1 Hardware Setup

A commodity web camera is mounted on the ceiling on top of the conveyor belt-based cutlery washing machine. The web camera is configured to record video with resolution of 1920 by 1080 pixels covering the entire conveyor belt of the

Fig. 1: Web camera mounted on ceiling to capture top-down video of conveyor belt-based cutlery washing machine

conveyor belt-based cutlery washing machine at a rate of 30 frames per second. The web camera is in turn connected via USB to a mini-PC running Windows 11. The mini-PC is equipped with an Intel i7-1260P processor, 32 GB of RAM and a 500GB SSD drive.

The cutlery washing machine has a steam compartment within it which uses water and steam for cleaning and sterilizing the cutleries. The web camera cannot be placed on or within close proximity to the cutlery washing machine due to moisture and temperature conditions. The high temperature of steam also causes potential fogging of the camera lens due to condensation. Due to these constraints, both the web camera and mini-PC thus have to be placed a safe distance away from the cutlery washing machine and this explains our placement of the web camera and mini-PC. A picture of the mounting of the web camera is shown in Figure 1. The mini-PC is residing within the ceiling and thus hidden from view.

Our client runs several conveyor belt-based cutlery washing machines and intends to deploy this solution for all the machines. The cutlery washing machines are distributed far apart on the factory floor and connecting the web cameras mounted on top of each cutlery washing machines using usb cables to a single mini-PC is not feasible due to the distance. On top of this, the factory floor is not provisioned with adequate wireless infrastructure which rules out wireless connectivity between the cameras and mini-PCs. We thus intend to retrofit each cutlery washing machine with their own individual set of mini-PC and web camera for the final deployment.

3.2 Object Detection

The main objectives for this study are to compute the utilization of the conveyor belt-based cutlery washing machine and the speed of movement of the machine's conveyor belt. We postulate to evaluate both computer vision and deep learning objection detection techniques for the detection of the cutleries. Specifically, with the localization of the objects by bounding boxes, the utilization can be computed from the sum of areas of the predicted bounding boxes that enclose the cutleries while the speed can be derived from tracking the difference in distance in pixels (for the detected objects) across a fixed number of frames of captured video.

In recent years, deep learning has been widely adopted for object detection. Most state-of-the-art object detectors utilize deep learning networks as their backbone and detection network to extract features automatically from images for classification and localization [24, 10]. One of the popular deep learning-based object detector is YOLO [8, 1, 2]. In YOLO, convolutional layers are used to extract image features which are in turn used to predict both the bounding boxes (for locating the object) and learn the class probabilities. YOLO divides every image into a grid of S x S and every grid predicts N bounding boxes and confidence. The confidence denotes the precision of the bounding box and whether the bounding box contains an object or not. YOLO is one of the fastest detector with the capability to process images in real-time at 45 frames per second and inference of up to 300 times faster as compared to other detectors [5]. In addition, YOLO also outperforms other detectors in terms of its ability to generalize from natural images to images in other domains [19]. As the cutleries are placed on a moving conveyor belt, the speed of detecting and recognizing the objects is instrumental to the accurate calculation of utilization and speed. The real-time detection speed of YOLO is thus one of the key consideration for its use in this study.

The Faster-RCNN [20] is another popular architecture that is used for object detection. Faster-RCNN uses a backbone network for learning a convolutional feature map before using a separate network to predict the region proposals. The predicted region proposals are then reshaped using a Region of Interest (RoI) pooling layer used for both classifying the image within the proposed region and predicting the offset values for the bounding boxes.

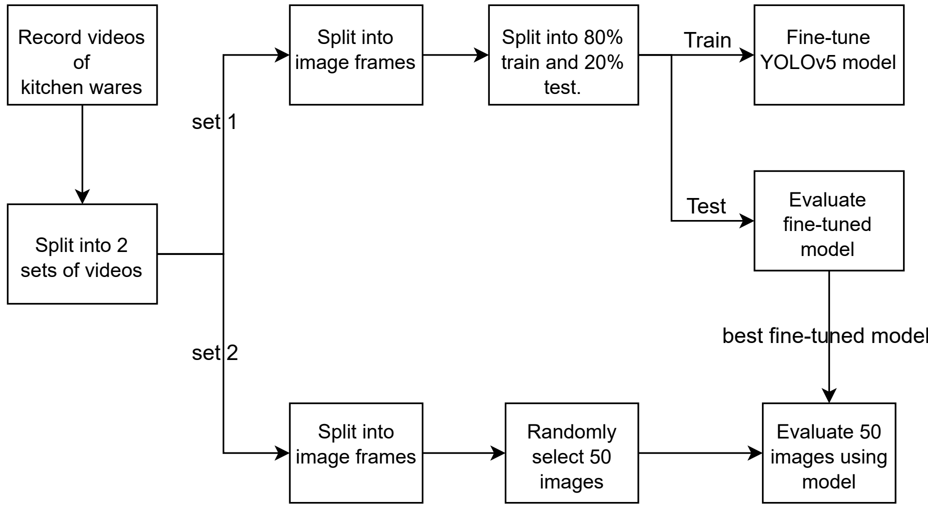


Fig. 2: Processing and evaluation workflow of videos using pre-trained YOLOv5 model

We will compare between the use and accuracies of YOLO and Faster-RCNN for the detection of the cutleries in our context.

3.3 YOLOv5 and Faster-RCNN

We first captured videos of different cutleries being cleaned on the conveyor belt-based cutlery washing machine. The processing and evaluation workflow for the videos using YOLOv5 is shown in Figure 2.

We recorded videos of various cutleries placed on the conveyor belt-based cutlery washing conveyor belt for washing. A total of 6 videos were recorded with each recording different sets of cutleries. The videos vary in lengths from 2 to 5 minutes. The videos were split into 2 sets with the second set reserved for evaluating the generalizability of the YOLOv5 model.

Using the first set of the recorded videos, we split the videos into image frames using Python OpenCV library and then manually labelled the bounding boxes in each image as the ground truth. 80% of the images were randomly selected as training image to be used for fine-tuning the model while 20% of the images (validation images) were used for evaluating the performance of the fine-tuned model. Fine-tuning or transfer learning [3] is a technique where a model pre-trained for one task is tuned or tweaked to perform a second similar task. We used pre-trained weights for YOLOv5s model from Ultralytics [9] and fine-tuned the model using our training images. YOLOv5s were pre-trained using the Microsoft Common Objects in Context (COCO) [15] dataset. We selected YOLOv5 out of the many object detection models as it offers fast inference speed of about 100 ms running on a CPU.

Using the training images, we fine-tuned the pre-trained YOLOv5s model by running the training loop for a total of 100 epochs with Stochastic Gradient Descent (SGD) as optimizer, a batch size of 64 and evaluated its performance on the validation images with early stopping. After training for 100 epochs, we were able to achieve a mean Average Precision (mAP) of 0.9655 on the validation images at an intersection-over-union (IOU) threshold of 0.5. The mAP is a metric that is commonly used for evaluating object detection models.

Using similar protocol, we fine-tuned Faster-RCNN by using MobileNet v2 [21] as the model backbone (with pre-loaded weights). The Faster-RCNN model which is trained for 100 epochs with SGD as optimizer, achieved a mAP of 0.899 for an IOU threshold of 0.5 with early stopping on the validation images. We thus selected YOLOv5 as the object detection model for our study.

In this study, we further evaluated YOLOv5 against a traditional image processing technique - Colour Image Segmentation [4, 6] for the detection of cutleries on the conveyor belt-based cutlery washing machine. The generalizability of the object detection techniques is important as the set of cutleries to be cleaned is not fixed and new collections of cutleries may be introduced from time to time. Due to operational requirements, our client would also prefer minimizing down time and effort to re-train the object detection model.

3.4 Color Image Segmentation (CIS)

We converted a sample set of 10 images from Red, Green and Blue (RGB) to Hue, Saturation and Value (HSV) colour space. Using the OpenSegment Streamlit application (https://kxborg-open-segment-hsv-segment-idubxp.streamlit.app), we first establish the HSV lower and upper boundaries color range for masking out the conveyor belt background. Having masked out the pixels within the HSV color range, the masked image is further converted to a binary image and then inverted so that the objects on the conveyor belt will be white while the conveyor belt is black. Selecting only the white pixels will then give us the boundaries or segments of the objects.

This technique works in this context as the conveyor belt is uniformly light yellow and can be differentiated in colour from the objects to be detected i.e. the cutleries. In addition, the conveyor belt-based cutlery washing machine is also deployed by our client in a large industrial kitchen floor with good lighting conditions. We recognize that the detection will deteriorate for cutleries that

Table 1: No. of objects missing detection by YOLOv5 and Color Image Segmentation

Type	No. of images	No. of Objects	YOLOv5	Color Image Segmentation (CIS)
			No. missed	No. missed
'Unseen' images	28	152	70 (46.05%)	11 (7.24%)
'Seen' images	22	155	10 (6.45%)	14 (9.03%)

are similar in colour to the conveyor belt or in poorly lit conditions. Notwithstanding the 2 constraints, an advantage of this over the deep learning technique (YOLOv5) is elimination of the need to re-train the deployed model with additional images should there be a need to detect new sets of cutleries.

4 Results and Discussion

4.1 Generalizability

To evaluate the generalizability of both techniques, we selected 50 images randomly from the second set of videos. We consider that if the object's bounding box (as detected by the algorithm) has an overlap of more than 50% with the ground truth bounding box, the object is considered as detected. It will be considered as missed detection otherwise. These images contain a mix of cutleries that were 'seen' (used in the training of the model) and 'unseen' (not used in the training of the model) by the YOLOv5s model. The generalizability results for the 2 techniques is shown in Table 1

From Table 1, we can see that for the 'unseen' images, YOLOv5 did not manage to detect 46.05% of the 152 objects while CIS only missed out 7.24% of the 152 objects. YOLOv5 performed slightly better for the 'seen' images by missing out 6.45% of the 155 objects as compared to CIS missing out 9.03% of the 155 objects. Thus, we surmise that CIS performs better in terms of generalizability and will be the preferred technique used for computation of utilization of conveyor belt-based cutlery washing machine.

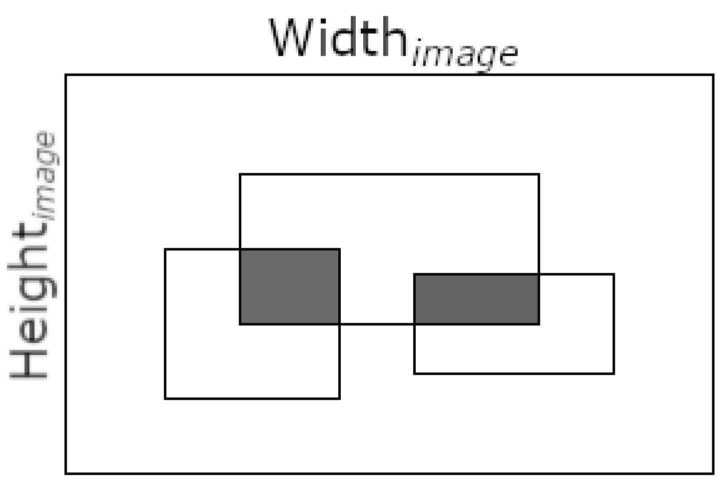


Fig. 3: Computation of utilization

4.2 Utilization and speed computation

The utilization of the conveyor belt-based cutlery washing machine which can be derived from the areas of the boundaries for each detected object using the CIS technique is given by.

$$Utilization = \frac{\sum Areas_{boundaries} - \sum Areas_{overlapped}}{Width_{image}.Height_{image}} \tag{1}$$

The overlapped areas are the gray shaded areas while the boundary areas are the areas of the whole rectangles (both shaded and unshaded) as depicted in Figure 3.

To detect the running speed of the conveyor belt-based cutlery washing machine, we employ the centroid tracker for tracking the objects.

The centroid tracker algorithm is detailed in Algorithm 1.

Algorithm 1 Centroid tracker algorithm

```
Input: currImageFrame
Output: storedObjects
Detect objects for currImageFrame as currObjects
Compute centroids of currObjects as currCentroids
while inputCentroid in currCentroids do
    if trackedObjects is empty then
        Add 1 to objectId
        Insert objectId and inputCentroid as trackedObj into trackedObjects
    else
        Compute euclidean distance between inputCentroid and centroid of
        trackedObjects as distDiff

        if min(distDiff) <threshold then
            Update centroid coordinates of object corresponding to min(distDiff) in
            trackedObjects
        else
            Assign new object id to object
            Add new object with centroid coordinate to trackedObjects
        end if
    end if
end while
```

We employ the centroid tracking algorithm for object tracking as it runs fast and works well in our context. The accuracy of object tracking is however in turn dependent on accurate detection of the objects.

With the identification and tracking of the objects, we proceed to compute the speed for every 30 frames. The distances moved by objects is calculated as the Euclidean distance between the coordinates of the tracked objects for frame 1 and frame 30 for every 30 frames of images. The speed is then computed as median of the computed distances over the 30 frames. As the webcam is set to record at 30 frames per second, the unit of measurement for computed speed will be the number of pixels moved per second. The algorithm for speed computation is detailed in Algorithm 2.

Algorithm 2 Speed computation algorithm

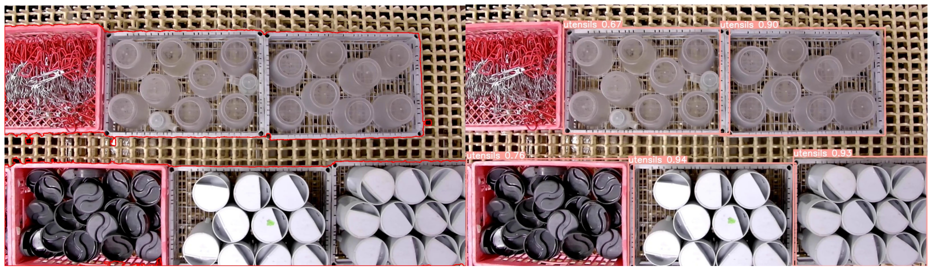
Input: *prevStoredObjects*, *currStoredObjects*
Output: *speed*

while *currObject* in *currStoredObjects* **do**
    while *prevObject* in *prevStoredObjects* **do**
        Compute euclidean distance between *prevObject* and *currObject* as *dist*
        if $dist < minDist$ **then**
            $minDist = dist$
        end if
    end while
    Insert *minDist* into *distances*
end while
Compute median(*distances*) as *speed*
return *speed*

As one of the primary objective of our client is to monitor instances of careless usage (intermittent starting and stopping of the machine), we further evaluated that our proposed speed computation algorithm is able to detect all instances where the speed is zero (where the conveyor belt slowed from a running state to a complete stop) and where the conveyor belt starts moving from a stopped state albeit with the constraint that there is at least a cutlery that is 'seen' by the camera. When combined with the utilization measure, the proposed speed algorithm is also able to detect instances of low or waste usage where speed is non-zero but utilization is low. This would be a situation where the cutlery washing machine is running but only few cutleries are placed on the conveyor belt.

4.3 Constraints

One shortcoming of the CIS detection technique is that for objects that are placed closed to one another, it tends to detect the multiple objects as a single joint object. An example of this is shown in Figure 4. The lower row of 3 baskets in Figure 4 is detected as a single object which spans the entire width of the image as opposed to YOLOv5 which managed to detect the individual baskets. This will affect the speed computation as across the image frames, the coordinates of the centroid of the tracked object will be the same and the computed speed will thus be zero. Thus, for speed computation, we switched to the YOLOv5 detection model for situations when CIS detects an object which spans the entire width of the image (within a tolerance of 5 pixels). However, we recognize that there will



(a) Detection of objects as one by CIS (b) Detection of same objects by YOLOv5

Fig. 4: Multiple objects detected as one by CIS as compared to YOLOv5

still be situations when YOLOv5 cannot detect the objects and CIS detects the objects as spanning the entire width of the image. In such situations, the speed computed by our proposed algorithm will unfortunately still be zero.

Another constraint of the CIS detection technique is for objects that are similar in color to the conveyor belt or objects that are translucent in color, either part or all of the object may not be detected. This will result in inaccurate calculation of utilization.

The final constraint is with the use of computer vision techniques for inferrring the speed of the conveyor belt-based cutlery washing machine. When no objects are placed on a moving conveyor belt, both the utilization and speed will be zero as computed using the proposed techniques. The reason for this is that no objects will be detected thus resulting in zero computed distance or speed between frames. A possible solution that we proposed to our client to tackle this issue is to install an energy socket power monitoring sensor to monitor the energy usage of the power socket that the cutlery washing machine is plugged into. For situations when the cutlery washing machine is operating but no objects are placed on the conveyor belt, the energy consumption sensor will detect that the machine is running and yet no objects are on it.

5 Conclusion

We have retrofitted the legacy conveyor belt-based cutlery washing machine using commodity web-camera and computer vision techniques to equip it with additional capabilities of tracking both utilization and speed of the machine. We proposed object detection using both traditional image processing and deep-learning techniques for the computation of utilization and speed. With the existing operational constraints of our client, frequent re-training of the deep learning model for object detection is not feasible. Thus, we also evaluated the generalizability of both object detection techniques and found colour image segmentation to be more generalizable and thus adequate for use in utilization computation.

For speed computation however, a hybrid of both colour image segmentation and YOLOv5 object detection was adopted.

With the enhanced data capability, our client will be able to visualize the automatically collected real-time utilization and speed data and optimize its operations in a data driven manner. Our client can collate and compare the utilization rate of individual conveyor belt-based cutlery washing machines across different times of the day and across the different locations for optimizing the operations of the conveyor belt-based cutlery washing machines. With the capability of detecting the speed of the machines, our client is also able to monitor instances of waste (where the machine is running but underutilized) or careless usage of the cutlery washing machine (where the machine is intermittently turned on and off) and thus minimize wastage, down-time and reduce the maintenance costs of the machines. A commodity camera was used in this proof of concept application as we were constrained by client's budget. We recognize that a higher end industrial grade camera would likely achieve better performance. Another possible future extension of this work is the investigation of newer and faster deep learning models that is more generalizable across 'unseen' images.

References

1. Adarsh, P., Rathi, P., Kumar, M.: Yolo v3-tiny: Object detection and recognition using one stage improved model. In: 2020 6th international conference on advanced computing and communication systems (ICACCS). pp. 687–694. IEEE (2020)
2. Ahmad, T., Ma, Y., Yahya, M., Ahmad, B., Nazir, S., Haq, A.u.: Object detection through modified yolo neural network. Scientific Programming **2020**, 1–10 (2020)
3. Bengio, Y.: Deep learning of representations for unsupervised and transfer learning. In: Proceedings of ICML workshop on unsupervised and transfer learning. pp. 17–36. JMLR Workshop and Conference Proceedings (2012)
4. Busin, L., Vandenbroucke, N., Macaire, L.: Color spaces and image segmentation. Advances in imaging and electron physics **151**(1), 1 (2008)
5. Diwan, T., Anirudh, G., Tembhurne, J.V.: Object detection using yolo: Challenges, architectural successors, datasets and applications. Multimedia Tools and Applications pp. 1–33 (2022)
6. Hema, D., Kannan, D.S.: Interactive color image segmentation using hsv color space. Sci. Technol. J **7**(1), 37–41 (2019)
7. Hinton, G.E., Krizhevsky, A., Sutskever, I.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25**(1106-1114), 1 (2012)
8. Jia, X., Tong, Y., Qiao, H., Li, M., Tong, J., Liang, B.: Fast and accurate object detector for autonomous driving based on improved yolov5. Scientific reports **13**(1), 9711 (2023)
9. Jocher, G.: ultralytics/yolov5: v6.0 - YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support (Oct 2021). https://doi.org/10.5281/zenodo.5563715, https://doi.org/10.5281/zenodo.5563715
10. Kaur, R., Singh, S.: A comprehensive review of object detection with deep learning. Digital Signal Processing **132**, 103812 (2023)

11. Lee, I., Lee, K.: The internet of things (iot): Applications, investments, and challenges for enterprises. Business horizons **58**(4), 431–440 (2015)
12. Lee, J., Bagheri, B., Kao, H.A.: A cyber-physical systems architecture for industry 4.0-based manufacturing systems. Manufacturing letters **3**, 18–23 (2015)
13. Li, J., Su, Z., Geng, J., Yin, Y.: Real-time detection of steel strip surface defects based on improved yolo detection network. IFAC-PapersOnLine **51**(21), 76–81 (2018)
14. Li, Y., Niu, Y., Liu, Y., Zheng, L., Wang, Z., Zhe, W.: Computer vision based conveyor belt congestion recognition in logistics industrial parks. In: 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). pp. 1–8. IEEE (2021)
15. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
16. Liu, Z., Qu, B.: Machine vision based online detection of pcb defect. Microprocessors and Microsystems **82**, 103807 (2021)
17. Molano, J.I.R., Lovelle, J.M.C., Montenegro, C.E., Granados, J.J.R., Crespo, R.G.: Metamodel for integration of internet of things, social networks, the cloud and industry 4.0. Journal of ambient intelligence and humanized computing **9**, 709–723 (2018)
18. Pham, D.L., Chang, T.W., et al.: A yolo-based real-time packaging defect detection system. Procedia Computer Science **217**, 886–894 (2023)
19. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
20. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
21. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)
22. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
23. Tran, T.A., Ruppert, T., Eigner, G., Abonyi, J.: Retrofitting-based development of brownfield industry 4.0 and industry 5.0 solutions. IEEE Access **10**, 64348–64374 (2022)
24. Zaidi, S.S.A., Ansari, M.S., Aslam, A., Kanwal, N., Asghar, M., Lee, B.: A survey of modern deep learning based object detection models. Digital Signal Processing **126**, 103514 (2022)