Vision-Based Quality Control Check of Tube Shaft using DNN Architecture

Uday Kulkarni[1]*, *Abhishek* Patil[1], *Rohit* Devaranavadagi[1], *Shreya B* Devagiri[1], *Sneha K* Pamali[1] and *Raunak* Ujawane[2].

[1]Computer Science, KLE Technological University, Hubballi, Vidyanagar, 580031, Karnataka, India.
[2]Dana Anand India Private Limited, Dharwad, Karnataka, India

Abstract. Quality control is the process of ensuring that a product or service meets certain predetermined standards of quality. This can involve testing, inspection, and other methods to ensure that the product or service is fit for its intended use. The tube shaft is a component used in the drive shaft of a vehicle. It undergoes several stages from raw material to final product to increase its structural properties. Following the first step, which is hardening, preliminary quality control is done by cutting the tube shaft into two parts lengthwise to check the intensity of hardening and decide whether to accept or reject the part. We present a machine vision-based quality control system that uses You Only Look Once (YOLO) v5 to assess hardening intensity by analyzing the pattern formed on the cut piece's surface.

1 Introduction

Quality Control (QC) is the process by which products or services are tested and measured to ensure they meet a standard. To use a procedure called QC [1], an organization or an industry can ensure that a product's quality is upheld, improved, and has a longer lifespan. This is achieved by testing the product at various stages to ensure there are no errors. As one of the increased abilities in industrial automation, Computer Vision (CV) [2] aids in our ability to complete the task. In this particular field, they have made a lot of achievements. Many industries conduct quality control [1] testing manually by humans, who have a tendency for making mistakes or errors. Lack of training, poor communication, distractions, and overconfidence are among the most common reasons for human errors. The industry will lose nearly billions of dollars as a result of the errors, and product quality and dependability will suffer. The industry adopted computer vision-based quality control [1] to ensure that errors don't occur. It has given us many advantages, such as accuracy, reliability, and a

*Corresponding Author: shreyadevagiri16@gmail.com

reduction in downtime. Modern technology is moving humans away from the repetitive inspection task and more towards automation. Major advancements in hardware and software have already been developed, which have largely automated the QC [1] inspection process. CV [2] an image-based technology that aims to extract information from images to simulate human inspection, is a major tool for QC [1]. In a typical CV [2] system, pictures of a product are analyzed using at least one camera and a processing unit. A product image may be subjected to various analytics, such as surface, color, and dimension analysis. A desired characteristic of quality is outputted with quantified data from each analysis. A CV [2] system, as opposed to humans, can deliver consistent performance over extended periods of time at a comparatively low cost. Additionally, Computer Visions [2] systems can frequently complete checks in a split second, whereas a human may result in a much longer delay. Manufacturers trying to boost efficiency and reliability can greatly benefit from this.

The Tube shaft [3] is a component used in the drive shaft of a vehicle, and this drive shaft is used to transmit the torque from the transmission to the differential of the vehicle. The tube shaft [3] undergoes several batch-based stages from the raw material stage to the final stage. Our research focuses on the hardened tube shaft part which is classified as an OK-Part indicating an approved part and a NOT-OK-Part indicating a rejected part. If the first part of the batch is approved in the hardening stage, then the whole batch is sent to the next stage for processing, else if the first part of the batch is rejected then the whole batch is rejected and the hardening parameters of the tube shaft are modified. To approve the tube shaft as OK-Part and NOT-OK-Part it is cut into two pieces lengthwise. The cut piece of the hardened part is classified as an OK-Part or NOT-OK-Part by looking at the pattern developed on its surface due to the intensity of the hardening which has a dark pattern on the outer region as shown in Figure 1(a), if these dark regions approach towards the middle of the cut piece, then it signifies that excessive hardening has been done or in other terms, the cut piece has become brittle and that batch is rejected. If the dark regions are below the range, then also the batch is rejected because the hardening done is not up to the requirement. The tube shaft [3] has two major parts called the neck and spine as represented in Figure 1(b). Our focus is on the spine portion as this is the region where the effect of hardening is seen. The red colored boundary box is indicated to denote our region of interest which is used to determine the quality of the tube shaft [3] and classify it as an OK-Part or NOT-OK-Part.

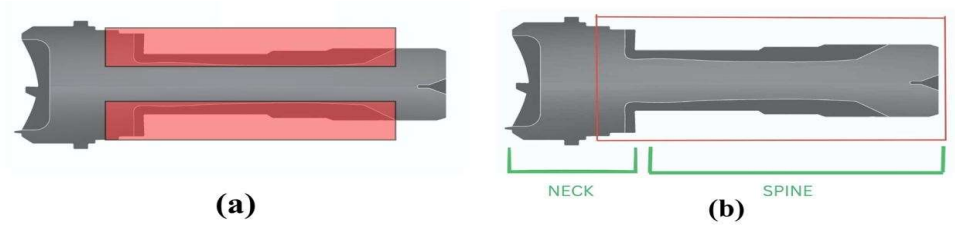

Fig. 1. (a) The dark region shows the intensity of hardness of the tube shaft and (b)The Red boundary box indicates the region of interest of the tube shaft from where the features will be extracted.

In this paper, YOLOv5's [4] application of object detection tasks is presented. YOLOv5 [4] is a deep learning-based object detection model from the YOLO (You Only Look Once) [5] model family. It is intended to be fast and accurate, making it appropriate for a wide range of object detection jobs. YOLOv5 [4] predicts the bounding boxes and class labels of objects in an image using a single convolutional neural network (CNN) [6]. The CNN [6] is trained on a large dataset of annotated images and can learn to recognize objects and their locations in images. When applied to a fresh image, YOLOv5 divides it into a grid of cells first. The

model predicts the presence of each object class in each cell, as well as the bounding box coordinates and confidence scores for any objects present in the cell, for each cell. YOLOv5 [4] then refines the predictions and produces the final object detections through a variety of post-processing procedures such as non-maximum suppression and intersection over the union. YOLOv5 [4] is an object detection model that is both fast and accurate.

This paper is divided into five sections. The introduction of the study is presented in the first section. The second section discusses the methodologies used in product identification. The third section discusses the research methodology. The fourth section contains the results and analysis, followed by the fifth section, which contains the conclusions.

2 Related Work

2.1 One-Stage Object Detector

The one-stage Detector contains only a single feed-forward neural network. In one stage object detector, the bounding box is predicted with the classification label on it without using the region of the proposal. On the image, it performs the dense sampling on several points by using different aspects of ratios and sales. During training dense sampling is challenging because it is unbalanced between the positive and negative values in the background. To extract the features CNN [6] is used i.e. one of the advantages of one stage detector in terms of speed. The simplicity and accuracy of stage detectors are in high demand. You Only Look Once (YOLO) [5] and Single Shot Multibox Detector (SSD) [7] are the famous architectures of one-stage object Detectors.

2.1.1 Single Shot MultiBox Detector (SSD)

SSD [7] is developed on VGG-16 with additional structures included to boost the performance. These additional convolution layers which are inserted at the end of the model to decrease the size of the model. SSD [7] features are to detect the smaller objects from large-scale feature maps then these are given to the deep layers i.e. convolution layers for detection of the object. SSD [7] uses the Faster-RCNN [8] anchor to generate a priori boxes with varying scales or aspect ratios for each unit and these priori boxes are used to predict the bounding boxes. In general, each unit will create a large number of priori boxes with different scales and aspect ratios. This reduces the difficulty of training. The advantages of SSD [7] include that it is faster than YOLO [5] and Faster-RCNN [8], although it had difficulty detecting smaller objects, which was rectified by selecting a better backbone architecture, namely ResNet.

2.1.2 YOU ONLY LOOK ONCE (YOLO) Family

(i) YOLOv1: It is the first one-stage object detection technique. It is faster when compared to a two-stage object detector. The advantages were capacity to recognize the objects fast and efficiently was the first successful step for object identification. Its disadvantages are that it can't detect the smaller objects in images with a large group of objects in it because in YOLO [5] architecture each grid is built for single object identification. It is incapable of detecting new shapes and their class. Loss function use approximate detection performance to for the errors from large and small bounding boxes resulting inaccurate localizations. It is not accurate as RCNN [9] detection approach.

(ii) YOLOv2 and YOLO9000: YOLOv2 [10] is a YOLO [5] improvement that maintains an easy balance between accuracy and speed. In a real-time environment, the YOLO9000 model [10] predicts 9000 object classes. YOLOv2 [10] uses DarkNet-19 architecture [11] instead of GoogleNet's core architecture [12] and its use's normalization to improve convergence and cooperative training of classification and detection systems to increase detection classes, which Vision Based Quality Control Check of Tube Shaft using DNN Architecture 5 is accomplished by eliminating Fully Connected layers to increase speed and applying learned anchor boxes to have better priors and improve recall score. With YOLOv2 [10], we may choose the model based on speed and accuracy because this architecture has fewer parameters.

(iii) YOLOv3: It has incremental enhancements when compared to the previous versions. Darknet-53 has replaced the feature extractor network. They also use various approaches like batch normalization, data augmentation, and multi-scale training. Instead of the SoftMax classifier, a logistical classifier is utilized. YOLOv3 [13] is faster than YOLOv2 [10], however when it comes to precision, YOLOv3 [13] falls short of YOLOv2 [10]. YOLOv4: The YOLOv4 [14] architecture is a convolutional neural network (CNN) [6] architecture designed for fast and precise object detection. It comprises a feature extractor as the primary body, followed by several detection layers. The primary portion of YOLOv4 [14] extracts high-level information from the input image using a combination of residual blocks and dense blocks. The bounding boxes and class probabilities for the items in the image are then predicted by the detection layers using these features. The bounding box predictions are more accurate because of the addition of anchor boxes and a novel loss function in YOLOv4 [14]. However, YOLOv4 [14] is significantly easier to train on a single GPU than most other detection algorithms, which typically require multiple GPUs. It is twice as rapid and performs comparably to EfficientDet [15]. YOLOv4's [14] architecture aims to strike a balance between efficiency and precision, making it suitable for real-time object recognition applications.

3 Proposed Methodology

3.1 Dataset

We used a custom dataset prepared by us from KLE Technological University via dark room in our project. It includes 325 RGB photos of the cut piece of tube shaft components. The image aspect ratio is 1:1. The image resolution is 3024x3024 JPG format without compression and with auto-light balancing. The dataset was numbered 1 through 325. The photos were then labeled using Roboflow. The dataset was randomly split into 70%, 20%, and 10% for training, validation, and testing, respectively shown in
Table 1.

3.2 Data Preprocessing

The dark region on the surface of the tube shaft's cut piece developed during the hardening stage wasn't visible enough even after dipping the cut piece into an acidic [16] solution, due to which testing the images in the industry setup was challenging. To overcome this, the raw images are enhanced by increasing the brightness of the overall image and highlighting the dark regions for better feature extraction. Figure 2(a) and Figure 2(b) represent the image

before and after enhancement. We enhanced the images by computing the average brightness, contrast, and darkness value that will work in all lighting conditions. After enhancing the images, we divided the whole dataset into OK and NOT-OK categories and drew bounding boxes indicating the region of interest which is the spine of the tube shaft. While testing the image in the industry setup, negligence in keeping the part in the correct position may have affected the prediction output. Also, the vibrations produced by heavy machinery may affect the quality of the image to overcome this data augmentation is performed by generating blurred images, images with a slight change in the angle, and images with a horizontal flip, vertical shear, or horizontal shear with bounding box exposure from the range -24% to 24% and blur up to 3.25 px. Now dataset includes a total of 900 RGB photos and Table 2 describes the split of data after augmentation.

Table 1. The number of images in training, testing and validation in the original dataset

Dataset	Part Type	Number of Images
Training	OK	153
	NOT-OK	120
Testing	OK	16
	NOT-OK	18
Validation	OK	29
	NOT-OK	34

Table 2. The number of images in training, testing, and validation after labelling and data augmentation

Dataset	Part Type	Number of Images
Training	OK	309
	NOT-OK	322
Testing	OK	43
	NOT-OK	45
Validation	OK	89
	NOT-OK	92

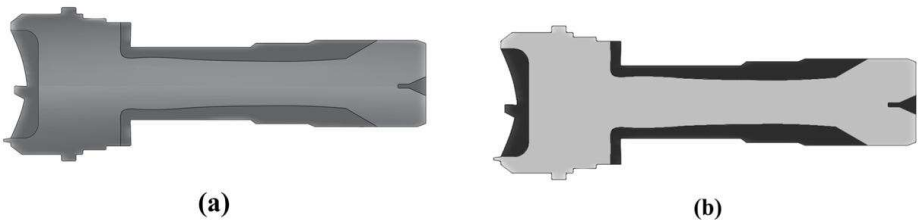

Fig. 2. (a) OK part before data Pre-processing and (b) OK Part after data Pre-processing (Enhanced)

3.3 Object Detection Model

Single-stage object detectors and two-stage object detectors are the two stages that object detection is typically divided into. We focused on the single-stage object detector, which is the first category. In a single-stage object detector, the region of interest is eliminated before the item is divided into classes using boundary boxes. YOLO family members are examples of single-stage object detectors. The YOLO object detection architecture, also known as YOU ONLY LOOK ONCE, consists of a single neural network that processes the input to predict bounding boxes and class labels for each bounding box. We used the YOLOv5 [4] architecture to achieve our goals because it surpasses the other models based on accuracy, detection, and speed.

3.4 YOLOv5 Model Architecture

This section outlines our procedures for determining whether a product is OK or NOT-OK. We employ the two-step strategy, where the first step involves doing target identification using some established techniques like YOLOv5 [4], and the second step involves using an image classifier to carry out two classifications.

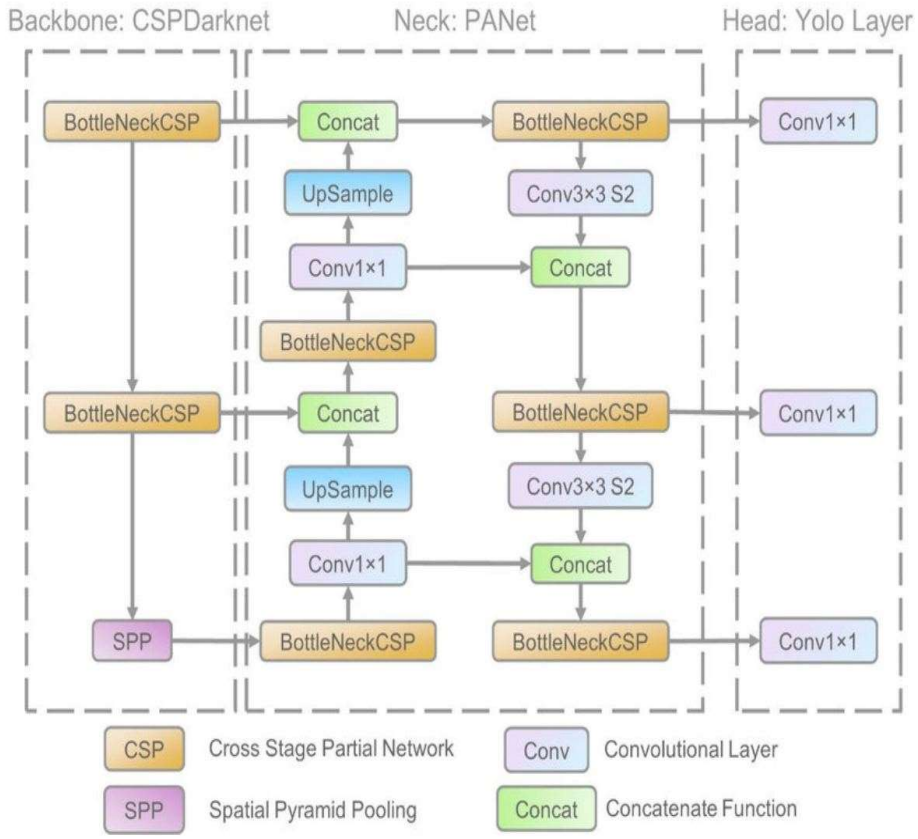

Fig. 3. The diagrammatic representation of the Yolov5 model architecture that consists of the backbone, neck and head section of the architecture.

3.4.1 Detection of pattern on tube shaft

The entire YOLOv5 architecture [4] is shown in Figure 2. The Backbone, Neck, and Head make up the three basic structural components of the YOLO series of models.

(i) YOLOv5 Backbone: For feature extraction from photos made up of cross-stage partial (CSP) Networks (CSPNet) [17], it uses CSPDarknet [18] as the backbone. CSPNet [17] is fast and extracts all the features from the input images.

(ii) YOLOv5 Neck: The neck's role is to merge the features extracted from the backbone, that is CSPNet [17]. The flexibility of input image size is enabled via Spatial Pyramid Pooling (SPP) [19]. YOLOv5 [4] generates feature pyramid vectors from any image size, aspect ratio, and scale. The combined feature pyramids are then sent to the detection head via the Path Aggregation Network (PANet) [20], which shortens the path with precise localization information and establishes a broken path between each object proposal and object level. PANet [20] increased the network's speed and flexibility.

(iii) YOLOv5 Head: The convolution network [6] serves as the detecting head. The detecting head provides object information such as confidence score, location, size and class. Based on the information provided by the detecting head, a bounding box is drawn.

YOLOv5 [4] also includes the following options for training:

(i) Activation function: The sigmoid activation function and LeakyRelu. The LeakyRelu function is used with convolution operations in the hidden or middle layers, while the sigmoid function is used with convolution operations in the output layer.

(ii) Loss function: Logistic loss is used along with binary cross-entropy.

(iii) Optimizers: SGD and Adam are the optimization options. In this paper, we have used Adam as the optimizer.
As we can see above, YOLOv5 [4] contains a variety of pre-trained models. The difference between them is the trade-off between model size and inference time. YOLOv5 includes several illumination spots as compared to the YOLO series:
(i) Multiscale: instead of PAN [20], utilize FPN [21] to improve the feature extraction network, making the model simpler and faster.
(ii) Target overlap: identify nearby places using the rounding method, to ensure that the target is mapped to many center grid points around it.

3.4.2 ResNet50 for Classification of as OK or NOT-OK part

ResNet is an acronym for Residual Network. One of the key components of YOLOv5 is the use of ResNet50 [22] as the backbone network. The use of ResNet50 [22] in YOLOv5 [4] allows the model to benefit from the pre-trained weights and knowledge of the ImageNet [23] dataset, as well as the ability to learn complex patterns and perform well on image classification tasks. This helps to improve the accuracy and performance of YOLOv5 [4] as an object detection model.

3.4.3 Complete Structure and Working of Our Detection Model

We developed a novel hybrid model for product detection by integrating the YOLOv5 [4] and ResNet50 [22]. The YOLOv5 [4] algorithm is widely used to detect objects, and the photos that are detected are preprocessed. The preprocessed image is then transferred to ResNet50 [22], a convolutional neural network [6], to perform the classification of OK-Part and NOT-OKAY-Part. After the hardening stage, the tube shaft is sliced into two pieces to determine the level of hardening. If the part is hardened more than required, it becomes brittle and is rejected. If the part is hardened less than required, it is also rejected since it requires strong structural properties. Although the features were not visible enough to make an accurate prediction. To address this, we improved the dataset during preprocessing by increasing brightness and dark region intensity to interpret the surface pattern of the sliced piece. Figure 2(a) is the cut piece of an accepted or OK-Part, and the intensity of hardening done is as per the requirement, whereas Figure 4(a) is the cut piece of a rejected part, hardening done is more than the requirement and thus the dark regions have exceeded the limit and came towards the center of the cut piece. In Figure 4(b) hardening done is not uniform, the dark patterns on the cut piece are irregular. No hardening has been done to the part in Figure 4(c) thus there are no dark patterns on the cut piece's surface.

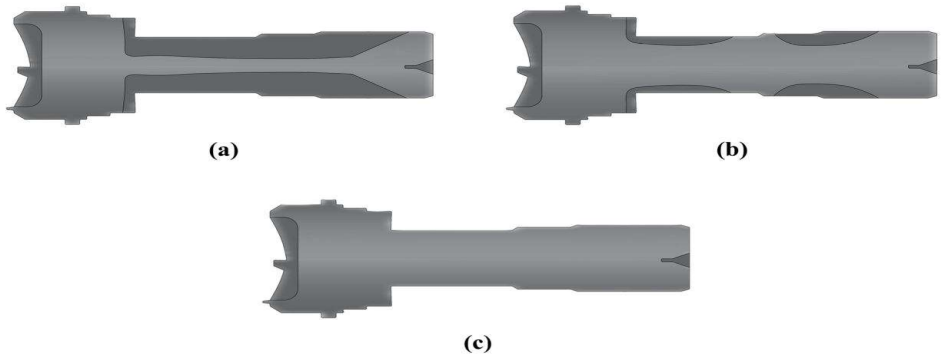

Fig. 4. NOT-OK Part Types: (a)Image of over-hardened part, (b)Image of non-uniformly hardened part and (c)Image of the unhardened part.

3.5 Loss

The formula for estimating the position of the target's regression box is:

$$CS_p^q = T_{p,q} * \text{IOU}_{\text{prediction}}^{\text{ground}} \quad (1)$$

In the above parameters, p and q indicate the p^{th} grids of q^{th} regression box, CS_{pq} and represents the p^{th} grid's confidence score for the q^{th} bounding box. $T_{p,q}$ denotes whether or not there is a target. $T_{p,q}$ equals 1 when the target is in the q^{th} bounding box; otherwise, $T_{p,q}$ equals 0. The IOU parameter represents the intersection over the union of the predicted box and the ground truth box There are 3 types of losses in YOLOv5 Architecture i.e., box loss, classification loss and object loss.

(i) Box loss: It indicates how efficiently the algorithm can detect the object in the center and check whether the predicted bounding box covers the object or not. x, y which is the location of the anchor box's B centroid. The width and height of the anchor box are specified as w, h. The width and height of the anchor box are specified as w_i, h_i.

$$L_{\text{box}} = \lambda_{\text{coordinates}} \sum_{p=0}^{S^2} \sum_{q=0}^{B} I_{p,q}^{\text{ojb}} bq\left(2 - w_p \times h_p\right) \left[\left(x_p - \hat{x}_p^q\right)^2 + \left(y_p - \hat{y}_p^q\right)^2 + \left(w_p - \hat{w}_p^q\right)^2 + \left(h_p - \hat{h}_p^q\right)^2\right] \quad (2)$$

(ii) Classification loss: It indicates how efficiently the algorithm can predict the correct class i.e. "L" for the given object

$$L_{\text{classification}} = \lambda_{\text{class}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{p,q}^{\text{obj}} \sum_{C \in \text{ classes}} p_p(c) \log(\hat{p}_L(c)) \quad (3)$$

(iii) Object loss:

$$l_{\text{object}} = \lambda_{\text{noobject}} \sum_{p=0}^{S^2} \sum_{q=0}^{B} I_{p,q}^{\text{noobject}} \left(c_p - \hat{c}_l\right)^2 + \lambda_{\text{object}} \sum_{p=0}^{s^2} \sum_{q=0}^{B} I_{p,q}^{\text{obj}} \left(c_p - \hat{c}_l\right)^2 \quad (4)$$

where $\lambda_{\text{coordinates}}$ is the location of loss coefficient,
λ_{class} is the type of loss coefficient,
$\hat{x}$, $\hat{y}$ is the true central coordinate of the target, and
$\hat{w}$, $\hat{h}$ is the width and height of the target.
If the anchor box at (p, q) contains targets, then the value $I_{p,qobj}$ is 1; otherwise, the value is 0. The p(c) represents the category probability of the target, and $\hat{p}_{L(c)}$ is the true value of the category. The length of the two is equal to the total number of categories C.
The loss function of YOLOv5 architecture is

$$\text{loss} = L_{\text{box}} + L_{\text{classification}} + L_{\text{object}} \quad (5)$$

3.6 GUI (Graphical user interface) of the designed Tkinter-based app

A GUI (Graphical user interface) is created using Python's Tkinter library, making it convenient and simple to test the tube shaft's quality shown in Figure 5. A web camera is integrated with the GUI to capture images in real time that are input to the DL model, the raw images captured for testing are then enhanced for better feature extraction and provided to the model, which predicts the quality of the tube shaft as OK-Part or NOT-OK-Part. Additionally, the option to use an existing image is provided, allowing the user to choose any image to check its quality. Fields like part number, machine number, station-number and shift type are provided for documenting captured image.

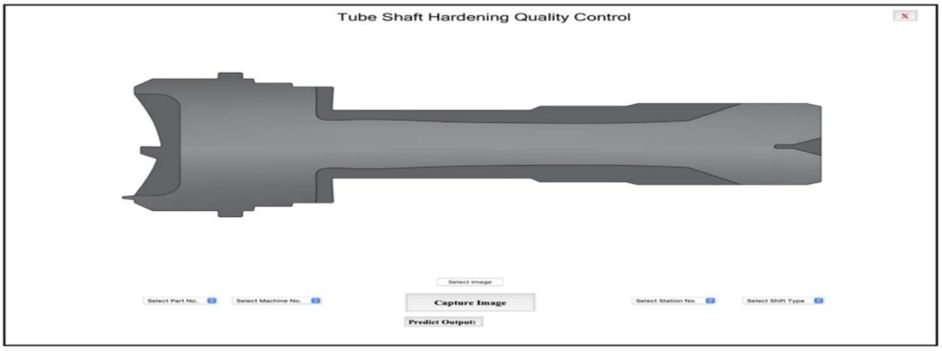

Fig. 5. Snapshot of the GUI designed to carry out the experiment at industry

4 Results and Analysis

4.1 Experimental Setup

YOLOv5 is implemented on Pytorch 1.12.1. For training and testing, we have used NVIDIA DGX-1 on GPU 0 Tesla V100-SXM2 with 3 cores. We employ a part of a pre-trained model from YOLOv5 in the training phase. Our customized YOLOv5 model needed over 60 minutes to be trained on the aforementioned system. The model size is 14.3 MB, there are 157 layers overall, 7015519 parameters, and 15.8 GFLOPs.

4.2 Evaluation Metrics

Precision measures in terms of percentage for correct prediction i.e. it measures how accurately the predictions are done. Recall measures the correct predictions of all the true positive. Precision and Recall value ranges from 0 to 1.

$$\text{Precision}(P) = \frac{TP}{TP + FP} \tag{6}$$

$$\text{Recall}(R) = \frac{TP}{TP + FN} \tag{7}$$

TP, FP, and FN denote true positive, false positive, and false negative respectively. The Intersection over Union is another essential metric (IOU). It is a phrase commonly used in object detection. IOU is used to determine whether or not the bounding box was accurately predicted.

$$\text{IOU} = \frac{\text{Areaofoverlap}}{\text{AreaofUnion}} \tag{8}$$

$$\text{IOU} = \frac{\text{area}(\text{ predicted } - \text{ area } \cap \text{ ground } - \text{ truth })}{\text{area}(\text{ predicted } - \text{ area } \cup \text{ ground } - \text{ truth })} \tag{9}$$

A threshold is used to objectively determine whether the model successfully predicted the box position or not.

$$\text{class(IoU)} = \begin{cases} \text{Positive} \rightarrow \text{IOU} \geq \text{Threshold} \\ \text{Negative} \rightarrow IOU < \text{Threshold} \end{cases} \tag{10}$$

Average Precision (AP) it is defined as the area under the precision-recall curve. Mean Average Precision(mAP) computes the score by comparing the detected bounded box to the ground truth. Higher the score, morethe accurate the model predictions. mAP 0.5 denotes the average of AP when the IOU exceeds 50%, and mAP 0.5:0.95 denotes the average of AP from 50% to 95% IOU with a 5% interval.

$$AP = \int_0^1 P(R)dR \tag{11}$$

$$mAP = \frac{1}{i}\sum_{k=1}^{k=i} AP_k \tag{12}$$

Here i denotes the number of classes. The f1 metric evaluates the trade-off between precision and recall. When f1 is high, it indicates that both precision and recall are high. A lower f1 score indicates a greater difference in precision and recall.

$$f1 = 2\frac{\mathrm{P} * \mathrm{R}}{\mathrm{P} + \mathrm{R}} \tag{13}$$

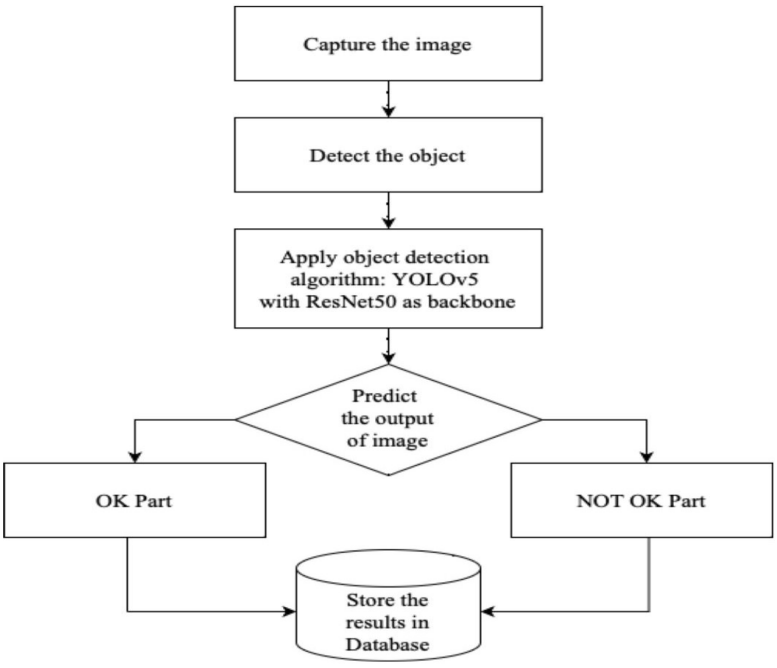

Fig. 6. Proposed Model: Proposed Deep Learning Model for Object detection and storing the ouptut in the

database

4.3 Implementation of object detection model

In the training process, we use the training dataset to learn about the object and detect and the validation dataset is used for model performance during the training. The following are the default hyper-parameters in the supervised training stage: The training epochs are set to 150. The batch size is 16. Adam is used as the optimizer, with a learning rate of 0.01. During the pre-training stage, all 900 images from the tube shaft dataset are used, including the labels. It can take up to 65 minutes to train the model for 150 epochs. The graphs depict the evolution of our model, which exhibits several outcome metrics for both the training and validation sets. We test on three different setups after training: photoshoot, normal, and industry. The following are the default hyper-parameters in the supervised training stage: The training epochs are set to 150. The batch size is 16. Adam is used as the optimizer, with a learning rate of 0.01. During the pre-training stage, all 325 images from the tube shaft dataset are used, including the labels. It can take up to 65 minutes to train the model for 150 epochs. The graphs depict the evolution of our model, which exhibits several outcome metrics for both the training and validation sets as shown in Figure 7 where x axis represents the number of epochs and y axis represents either box loss, objectness loss, training class loss, precision, recall and Mean Average Precision (mAP 0.5) for training and validation dataset.

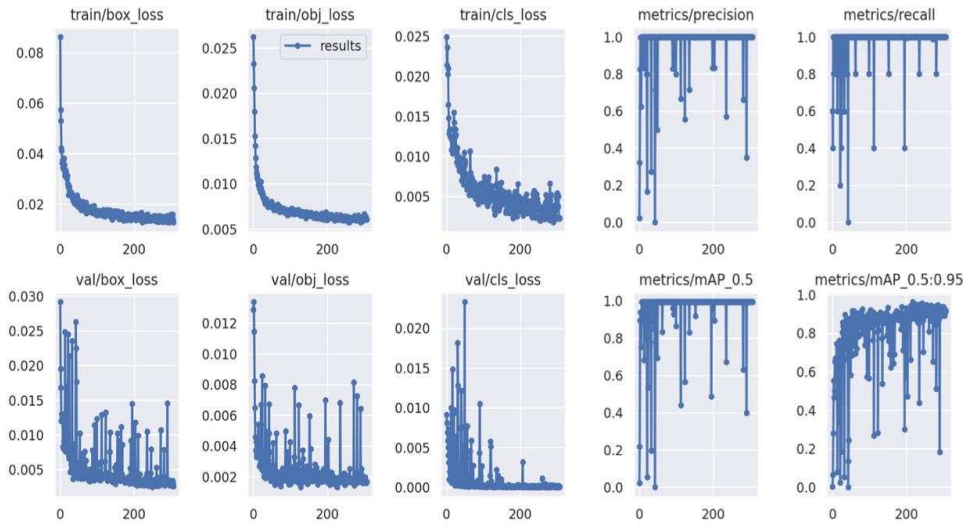

Fig. 7. Plots of box loss, objectness loss, training class loss, precision, recall and Mean Average Precision(mAP)- Y axis for training and validation dataset over the training epochs - X axis.

4.4 Implementation of Industry Setup

The industrial setup is used for performing the preliminary quality checking of the tube shaft. The testing arrangement consists of a single camera that takes images to test the part, our model is integrated with a graphical user interface and loaded onto the PC. Industry workers can examine the part by simply clicking on the button “Capture” and then “Predict”. The predicted output shown in Figure 9 and this is stored in the database. The database contains

the following information: captured image, part number, station number, batch number, and prediction output. The quality control department managers have access to this database.

Fig. 8. Industry Setup: Custom setup at Dana Anand India for testing purpose

4.5 Results of experiments conducted

The final implementation is done on the industry setup shown in the figure 8. The tube shaft pictures captured in the industry are then classified as Ok-part or as Not-Ok-part. The tube shaft in the figure 9(a) is classified as Not-Ok part because the tube shaft has been overheated. The tube shaft in the figure 9(b) is classified as Not-Ok part because the heating is done partially here. The tube shaft in the figure 9(c) is classified as Not-Ok part because there is no heating done. The tube shaft in figure 9(d) is classified as Ok-part because the heating is optimal.

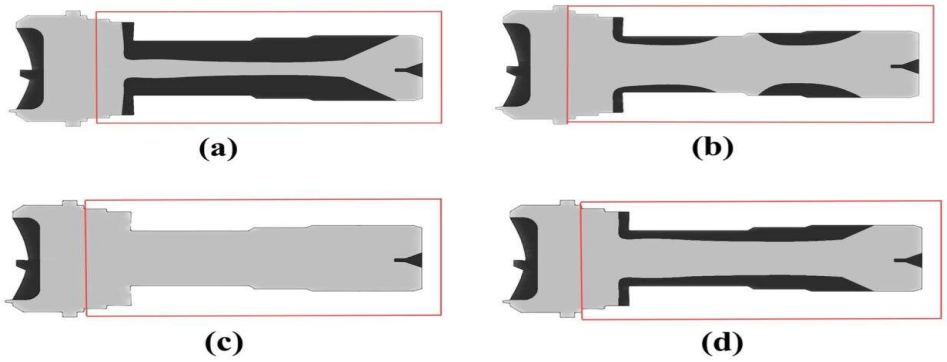

Fig. 9. Tube Shaft Pictures Captured in the Industry Setup

5 Conclusion

The proposed computer vision-based quality control increases the efficiency of quality checking of tube shafts and reduces human error by analyzing the intensity of hardening done on the surface of the tube shaft's cut piece. The images captured for testing the quality of the

tube shaft's cut piece are stored in the database along with part details. In the future, on-device DNN model can be developed by training the model with images captured in the industry setup and result in increased accuracy. This Project was conceptualized and implemented jointly by KLE Tech and Dana Anand India Private Limited.

References

1. Akundi, A., Reyna, M.: A machine vision based automated quality control system for product dimensional analysis. Procedia Computer Science 185, 127–134 (2021). https://doi.org/10.1016/j.procs.2021.05.014. Big Data, IoT, and AI for a Smarter Future
2. Lawaniya, H.: Computer vision. IET Computer Vision (2020)
3. Li, B.: Research on geometric dimension measurement system of shaft parts based on machine vision. EURASIP Journal on Image and Video Processing 2018 (2018). https://doi.org/10.1186/s13640-018-0339-x
4. 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)
5. 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)
6. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. CoRR abs/1409.4842 (2014) 1409.4842
7. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European Conference on Computer Vision, pp. 21–37 (2016). Springer
8. 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)
9. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. CoRR abs/1311.2524 (2013) 1311.2524
10. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271 (2017)
11. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. CoRR abs/1612.08242 (2016) 1612.08242
12. Wu, C., Wen, W., Afzal, T., Zhang, Y., Chen, Y., Li, H.: A compact dnn: Approaching googlenet-level accuracy of classification and domain adaptation. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 761–770 (2017). https://doi.org/10.1109/CVPR.2017.88
13. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
14. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: Yolov4: Optimal speed and accuracy of

object detection. arXiv preprint arXiv:2004.10934 (2020)
15. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10781–10790 (2020)
16. Siddiki, S.Y., Afroz, M., Munib, G., Amin, M.: Simulation of production of nitric acid. (2015)
17. Wang, C.-Y., Mark Liao, H.-Y., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., Yeh, I.-H.: Cspnet: A new backbone that can enhance learning capability of CNN. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1571–1580 (2020). https://doi.org/10.1109/CVPRW50498.2020.00203
18. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. CoRR abs/1804.02767 (2018) 1804.02767
19. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence 37(9), 1904–1916 (2015)
20. Yang, J., Fu, X., Hu, Y., Huang, Y., Ding, X., Paisley, J.: Pannet: A deep network architecture for pan-sharpening. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 1753–1761 (2017). https://doi.org/10.1109/ICCV.2017.193
21. Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 936–944 (2017). https://doi.org/10.1109/CVPR.2017.106
22. Koonce, B.: ResNet 50, pp. 63–72 (2021). https://doi.org/10.1007/978-1-4842-6168-2 6