AUTOMATED PCB DEFECT DETECTION USING MACHINE LEARNING

[1] Dr. Prakash R, [2]Bhoomika B, [3]Chinmayi V N, [4] Meghana,

[1] Professor, [234]B.E. Student [1234]NDepartment of Electrical & Electronics Engineering, **[1234]Acharya Institute of Technology, Bengaluru, India**

Abstract: *Printed Circuit Boards have been utilized for the necessary requirements in the electronic sectors; if there are any problems regarding the PCB production process, it can result in the generation of functional problems and costly production processes. This manual vision inspection and traditional rule-based automated optical inspections have also been confronted with certain difficulties, including less consistency, less flexibility, and less accuracy for different conditions. This paper introduces the Automated PCB Defect Detection and Real-Time Segregation System by using the YOLOv8 Deep Learning and Hardware Approach. Pictures of the PCB samples can be captured by the USB camera, and the manual segregations by utilizing the Roboflow tool have been employed for training the YOLOv8n for the open circuit, the short circuit, and the hole defects in the PCB samples. This resultant model can also be utilized for edge computing by the Raspberry Pi Microcontroller, and the Arduino segregation of the PCB samples corresponding to the open circuit, the short circuit, and the hole can be established by the servo motors and the conveyor systems along the different routes for the storages and removal of the samples. The result validated the feasibility of the research work with an accuracy of 93.2% and an approximate runtime of around 78 ms per image by utilizing the Raspberry Pi Microcontroller. This aforementioned system provided the low-cost and feasible approaches for the inspections and segregation of the PCB samples for the Small and Medium Scale Manufacturing Units.*

Keywords: PCB inspection-YOLOv8, Machine Learning, Embedded Systems, Raspberry Pi, Automation

I. INTRODUCTION

In each of these devices, currently used, from a simple calculator to the most advanced computer, the Printed Circuit Boards (PCBs) do not only provide the mechanical support on which the components are mounted; rather, the PCBs also provide the support on which the components are electrically connected. With the ever-increasing need for miniaturization of devices, the process employed for the PCBs has equally presented an increased complexity; hence, errors during the process of PCBs production due to the absent hole error might emerge.

Open Circuit:It is manufactured by the junction of unused and isolated conductors.

Conventional approaches to verification of PCBs involve the process of manual verification and rule AOI verification. Manual verification requires enormous human resources, and human knowledge or human fatigue, in general, is involved, as stated by. The important characteristics in rule AOI verification remain human-engineered, with no capacity to accommodate some variations which might be involved in the design of PCBs and in light arrangement settings for newly designed PCBs.

Nevertheless, with the emergence of new developments in deep learning, even image features can by themselves be found out from images by the automated inspection system. You Only Look Once object detection techniques, also known as YOLO, may be applied for identification and for classification of object location, with detection speeds in mere milliseconds. On this consideration, a real-time process for identification and for carrying out image segmentation for identification of defects in a PCB by YOLOv8 shall be developed.

II. Related Work

The usual procedure for image processing, such as edge detection, template matching, and morphological analysis, was employed for tracing the defects in the PCB from the previous developed technologies [4]. Though it was highly effective and implemented successfully, it is identified that it is highly sensitive to noised image patterns, environmental patterns, and patterns that exist in the PCB image patterns. The convolutional neural network technology reinstated the glory of the machine learning technology for achieving enhanced robustness and accuracy for tracing the defects in the PCB image patterns [5]. Few authors attempted to develop the tracing of defects in the PCB image patterns using the machine learning technology for enhanced accuracy improvement than the previous developed techniques [6]. The most recent authors focused more on the object detection methods, namely Faster R- CNN and YOLO for tracing the defects in the PCB for real-time action [7], [8]. YOLO especially requires a special mention for the trace of defects for its deep understanding about the accuracy and time consumed.

>Most of the authors have clearly described the structure of the GPU and did not focus on other implementation procedures, such as that of the embedded system.

III. SYSTEM OVERVIEW

The process, besides the aforementioned, in the system includes image acquisition, defect detection, decision-making, and separation. This has been made possible by acquiring images from the USB camera mounted on the PCB conveyor belt. These images have been processed with the help of the algorithm supported by the Raspberry Pi, which is known as YOLOv8. The decisions were made with the help of the Arduino processor decision, after which control signals were generated.

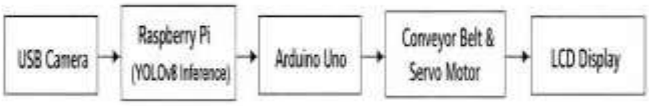
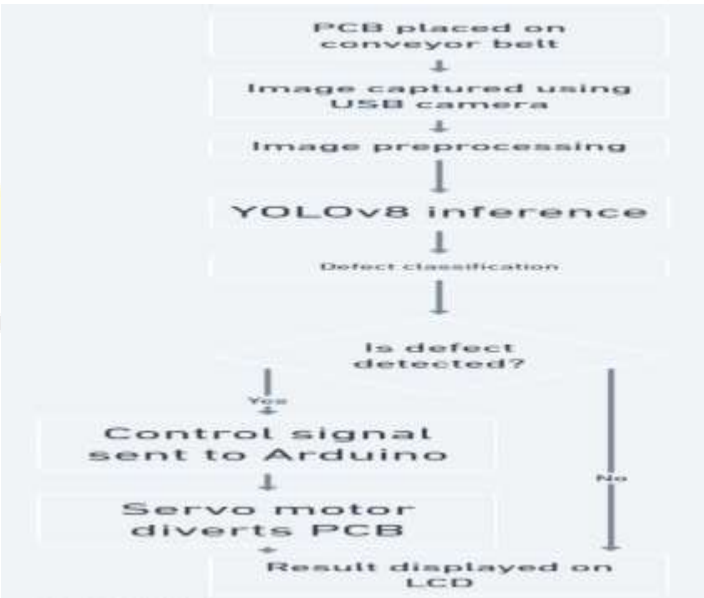


Fig. 1. Overall block diagram of the proposed PCB defect detection and segregation system

Fig. 1 Overall Design PCB Defect Detection and Segregation System Design. Moreover, the design will incorporate computer vision technology to segregate the PCB modules with an electromechanical module and differentiate them by means of computer vision technology to make it totally automatic. The USB Camera will be placed above the conveyor belt to capture the images of the PCB modules at a predetermined speed, and the immediate analysis will be done by means of the Raspberry Pi. In the proposed design, the YOLOv8 learned method will be utilized to segregate the defective modules by means of decision-making by the Raspberry Pi, resulting in nearly a zero latency factor from the image detection process to the identification process.

According to the type of defects, the control signals will be produced by means of the serial interface to the Arduino Microcontrollers. In the proposed design, the Arduino Microcontrollers will be utilized to control and manage the low-level hardware components of the processing level of the project like the motion of the conveyor belt by means of the motor driver and the Servo Motor, which will be switched on to segregate the PCB modules. In the analysis process, the values will be added to the modularity of the processing level of the project by means of the diagnosis process of the PCB where the processing of the analysis process will be done independently from the motion of the conveyor belt by means of the motor driver and the activation of the Servo Motor to segregate the PCB modules subsequent to the immediate analysis process by means of the Raspberry Pi..

IV. Operational View



A Dataset Preparation

The labeling of the PCB images was done using Roboffow. This dataset contains defect-free and defective images. Defects in this dataset are classified into three classes: open circuits, short circuits, and missing holes. Data augmentation techniques were followed to increase generalization capability by rotation, brightness alteration, and ffipping.

Parameter	**Value**
Total images	1,200
Training images	840
Validation images	180
Test images	180
Defect classes	Open, Short, Hole

Table I. Dataset Description

B . YOLOv8 Detection Model

YOLOv8 is a single-stage object detection model that predicts bounding boxes and class probabilities in one forward pass. The detection process is defined :

$$\hat{y} = f_{\{\theta\}}(x)$$

where x is the input PCB image, fθ represents the YOLOv8 network, and ŷ denotes the predicted bounding boxes and class labels.

The training objective is given by [10]:

$$L = \lambda_{\{box\}} L_{\{box\}} + \lambda_{\{cls\}} L_{\{cls\}} + \lambda_{\{obj\}}$$

C. Training Configuration

Parameter	**Value**
Model variant	YOLOv8n
Image size	640 × 640
Epochs	100
Optimizer	Adam
Learning rate	0.001
Framework	Ultralytics (PyTorch)

Table II. YOLOv8 Training Parameters

V. Hardware Implementation

Inference: This could be achieved by using the Raspberry Pi 4.

Motor Control: This will be achieved through the employment of an Arduino Uno microcontroller board.

DC Motor Based Conveyor Belt System: This will also be done using Arduino Uno.

The hardware devices expected to be involved in the implementation process of the proposed system are expected to include Raspberry Pi, which is expected to involve the role of carrying out functions pertaining to the process of imagining and the detection of the process of imagination. The consumption of minimal space is also expected, and it is also expected to take the role of carrying out some processes by making use of its capabilities, as well as taking the role of carrying out processes pertaining to the processing of edge devices, which are also expected to involve carrying out processes pertaining to the process of performing deep learning. The device is also expected to involve being connected to the camera, which is expected to take the role of imagining processes pertaining to the joints of SMD, as well as involving having the required resolution, which is also expected to take the role of detecting.

Since the Raspberry Pi and other hardware components shall be perfect, the process concerning the concept of controlled and managed power supply shall also be considered while incorporating the constant voltage. Since the temperature or heat issue shall not hamper the working of the Raspberry Pi and other hardware components because the passive components concerning the heat issue shall also be considered while cooling the heat that could have been developed. Since all the hardware components shall also increase the flexibility of the system because all the hardware components concerning the elimination of vibration and images, which could hamper the clearness of the image, shall also be placed.

I.Component	Specification
Camera	USB Webcam (1080p)
Processor	Raspberry Pi 4
Controller	Arduino Uno
Motor driver	L268N
Actuators	DC motor, Servo
Display	16×2 LCD

Table III. Hardware Component

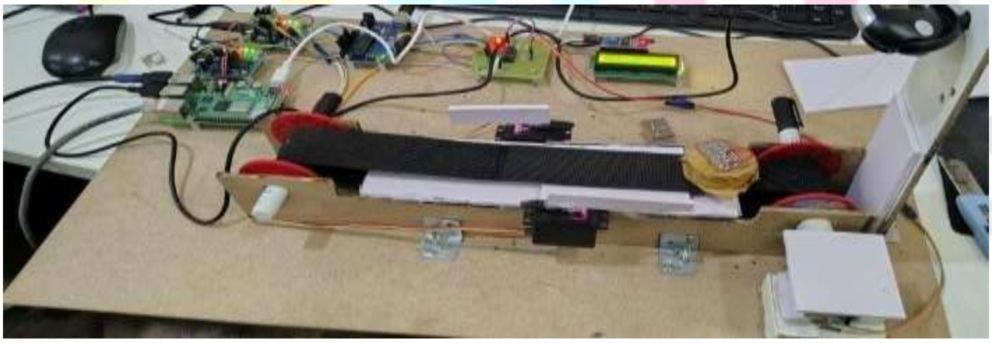


Fig. 3. Photograph of the experimental hardware setup.

VI. Performance Evaluation and Results

A. Quantitative Detection Performance

Detection performance was evaluated using precision, recall, and mean Average Precision (mAP):

$$Precision = \frac{TP}{(TP + FP)}$$

$$Recall = \frac{TP}{(TP + FN)}$$

$$mAP = \frac{1}{N} \sum AP$$

where TP, FP, and FN denote true positives, false positives, and false negatives, respectively.

Defect Type	**Precision (%)**	**Recall (%)**
Open circuit	94.1	92.5
Short circuit	93.6	91.8
Missing hole	92.2	90.9
Overall mAP@0.5	**G3.2**	—

Table IV. Detection Performance

B.Qualitative Detection Results

Fig. 6. Sample PCB defect detection results using YOLOv8.

Fig. 6(a): Missing-hole defect

Fig. 6(b): Short-circuit defect

Fig. 6(c): Open-circuit defect

Fig. 6. Example YOLOv8 detection results for PCB defects. Figure 6(a) - Def Fig.6(b): Short circuit To Fig.6 (c): Open circuit From Figure 6, the PCB images corresponding to the results of open-circuit, short circuit, and missing hole faults localization by the proposed method using the YOLOv8 algorithm are shown. In the image, the method is able to locate the open circuit, short circuit, and missing hole faults and the normal PCB with precise bounding boxes and corresponding labels. The method is able to locate normal PCBs without any false positives.confirm the model's capability to reliably detect and localize PCB defects under varying layouts and visual conditions.

C. ***Confusion Matrix Analysis***

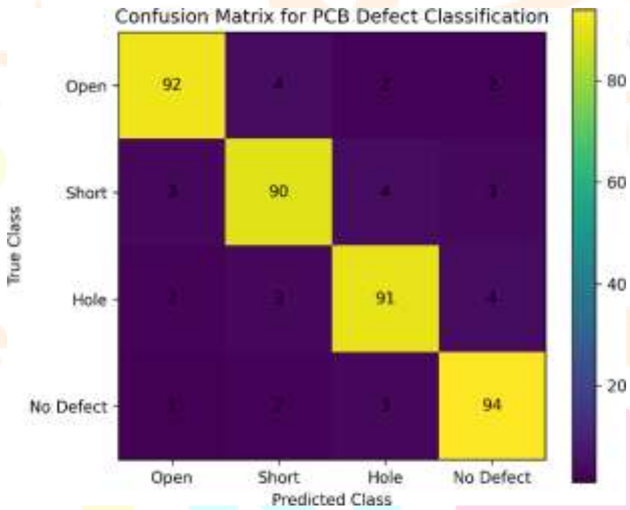


Fig. 7. Confusion Matrix for Classifying PCB Defects. In the confusion matrix, there is high diagonal dominance, and it signifies an effective classification without much confusion among Open Circuit, Short Circuit, Missing Hole, and Defect-Free classes of PCBs.

The confusion matrix in Fig. 7 is a detailed class-by-class analysis of the newly designed system for PCB defect detection. The strong dominance of the diagonal elements shows that a significant percentage of instances from every class were detected correctly. This verifies that YOLOv8 has been successful in identifying different classes of PCB defects, despite them closely resembling each other.

Open-circuit faults have the highest value of true positive rate because open-circuit faults have visible structural differences that can be easily captured by the model. There is a slight degree of confusion between short-circuit and open-circuit faults, and the reason is that short-circuit faults and open-circuit faults have a similarity in tracing characteristics when PCB is of high density. Missing hole faults have a stable level of classification, with low misclassification.

The No Defect class presents the highest classification accuracy, which clearly indicates the capability of the model to reduce the false positive rate and avoid misclassifying the defects. The confusion matrix proves the effectiveness of the proposed system in classifying the PCB defects efficiently.

D. Precision-Recall Curve Analysis

PR curves are used to investigate the balance between detection precision and recall over varying confidence thresholds. Complementing accuracy alone, PR analysis offers a richer understanding of model performance in tasks such as defect detection, where there might be issues of class imbalance. The PR curves for each category of PCB defects, generated using YOLOv8, are presented in Fig. 8.

In the detection of PCB defects, it is very important to maintain high precision with increasing recall so as not to miss the true defects in order to minimize false alarms. Precision–recall analysis gives insight into the sensitivity of the confidence threshold of the detection model. This is particularly important in the evaluation of real-time inspection systems deployed in manufacturing environments.

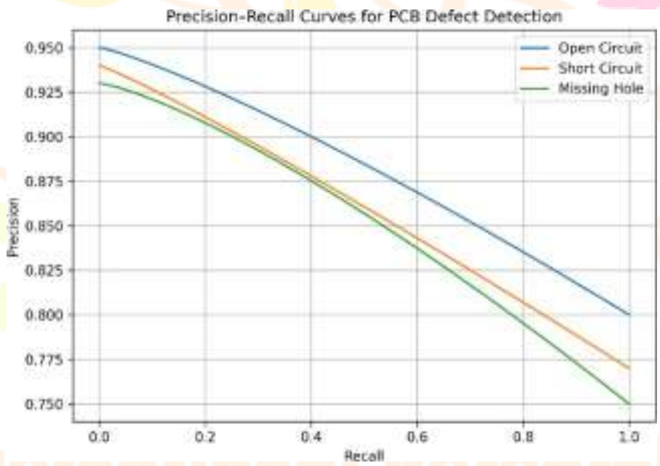


Fig. 8. Precision-Recall curves for PCB defect detection.

As reflected in Fig. 8, all classes of defects maintain high precision over a wide range of recall, indicating stable and reliable performance in detection. Among these, open-circuit defects represent the strongest PR characteristics, while the precision for the short-circuit and missing-hole defects is slightly low for high recall owing to some visual similarities and very minute features of the defect. In general, the PR curves confirm that the proposed system provides a good balance between precision and recall, making it more suitable for real-time PCB inspection applications.

E. Real-Time Performance

Metric	**Value**
Inference time	78 ms Frames per second 12.8 Sorting
accuracy	96.4%

Table V. Real-Time System Performance

Real-Time System Performance The performance of the automatic PCB image analysis system is of prime importance with respect to the real-time system processes. The effectiveness of the proposed technique has been explained by understanding the possible benefits that could have been put to fruitful use with respect to the application of Raspberry Pi technology and its effectiveness to address the continuous processing system. As already depicted in relation to the afore-mentioned statement with respect to the figure, there would be no words about the afore-mentioned result with respect to the afore-mentioned problem, where there would be the afore-mentioned quotation with respect to the number of the figure '9'.
where it has been depicted that due to the afore-mentioned inference time in relation to the afore-mentioned computation method to address the afore-mentioned consecutive images of PCB, it has been explained by taking into account the afore-mentioned application of YOLOv8 method to address the afore-mentioned neural network levels, where it has been explained that due to taking into account the afore-mentioned application of YOLOv8 method, there would be a need of approximately 78ms with respect to the afore-mentioned inference time to address the afore-mentioned consecutive images, which would be much more optimized in order to be able to address the afore-mentioned continuous processing system in some other different subject of expertise, where this subject of expertise would be the afore-mentioned PCB image analysis system.

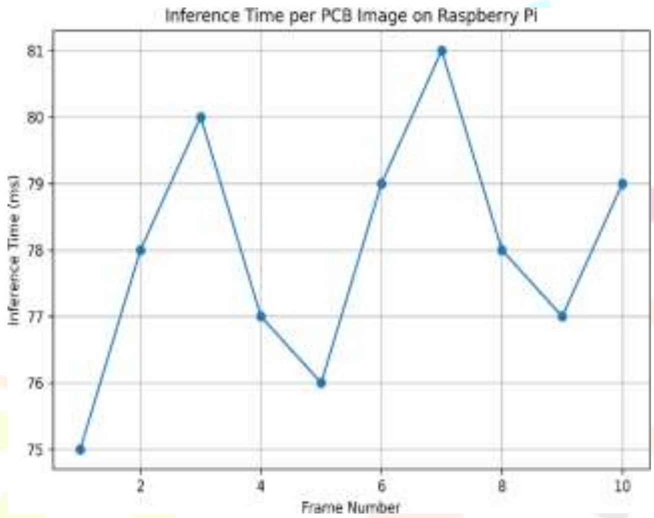


Fig. 9. Inference time per PCB image during real-time operation on Raspberry Pi.

*VII.*Conclusion

The relevance of the topic is because it justified the need for an automated system of defects and their segregation done in real time for using YOLOv8 in embedded systems. The technique employed in the research is quite efficient. This technique is able to process in real time using Raspberry Pi. Improvements are expected in the future with new defects and improved speeds.

With the results of the experiment above, it is known how the system is able to achieve the same level of accuracy along the line even when the conveyor is running along the non-stop system. To conclude, the whole process with the use of computer vision technology and the intervention of the hardware is capable of showing that the deep learning approach is indeed able to support the checking process of the PCB through the use of a lower-cost system, with or without the use of the GPU.

This modularity of the system, which will emerge from the components of hardware and software, will ensure that there is a clear advantage in relation to the division that relates to the identification of defects and the control of the mechanics that relates to the components of the system. It also seems to have a high possibility of there being a relationship between the modularity of the system and the ease of maintenance that relates to the components of the system. Moreover, the matter of incorrect identification of defects will also decline.

Conclusion

In conclusion, the emerging strategy makes it feasible to enable the automation process of the inspection and separation of the PCB through vision systems. This thus bridges the gap between the lab where most vision models are built and the solutions needed. The strategy makes sure that the vision system application within the lab is feasible.

VIII.References

[1] R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed., Springer, 2022.

[2] D. M. Tsai and J. Y. Luo, “Defect inspection in PCB manufacturing using machine vision,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 8, pp. 3380–3388, Aug. 2011.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, real-time object detection,” in *Proc. IEEE CVPR*, 2016, pp. 779–788.

[4] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in *Proc. IEEE CVPR*, 2017, pp. 7263–7271.

[5] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020.

[6] U. Farooq, S. Khan, and A. Hafeez, “Automated visual inspection of PCB using deep learning,” *IEEE Access*, vol. 9, pp. 144562–144574, 2021.

[7] Y. C. Wu and C. H. Lin, “PCB defect detection using image processing,” *International Journal of Advanced Manufacturing Technology*, vol. 25, no. 3–4, pp. 363–369, 2005.

[8] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,”
IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE CVPR*, 2016, pp. 770–778.

[10] T. Lin et al., “Microsoft COCO: Common objects in context,” in *Proc. ECCV*, 2014, pp. 740–755.

[11] M. Everingham et al., “The Pascal Visual Object Classes (VOC) challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[12] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[13] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. IEEE CVPR*, 2015,
pp. 3431–3440.

[14] H. Chen, Y. Pang, and J. Li, “PCB defect detection using deep convolutional neural networks,” *Journal of Manufacturing Systems*, vol. 56, pp. 456–468, 2020.

[15] Z. Huang et al., “Automatic surface defect detection using deep learning,” *Applied Sciences*, vol. 8, no. 7, pp. 1–15, 2018.

[16] S. Mittal, “A survey of FPGA-based accelerators for convolutional neural networks,” *Neural Computing and Applications*, vol. 32, pp. 1109–1139, 2020.

[17] X. Wang et al., “Embedded deep learning for industrial inspection,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4006–4017, Jul. 2019.

[18] N. O’Mahony et al., “Deep learning vs traditional computer vision,” *Machine Vision and Applications*, vol. 31, no. 1, 2020.

[19] S. Zhai, Z. Zhang, and Y. Zhang, “Real-time defect detection using YOLO-based models,” *IEEE Access*, vol. 8, pp. 176065– 176074, 2020.

[20] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, 2015.

[21] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proc. IEEE CVPR*, 2001, pp. 511–518.

[22] C. Szegedy et al., “Going deeper with convolutions,” in *Proc. IEEE CVPR*, 2015, pp. 1–9.

[23] R. Girshick, “Fast R-CNN,” in *Proc. IEEE ICCV*, 2015, pp. 1440–1448.

[24] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling," in *Proc. ICML*, 2019, pp. 6105–6114.

[25] Ultralytics, "YOLOv8 documentation," 2023. [Online]. Available: https://docs.ultralytics.com

[26] Roboflow Inc., "Roboflow: Dataset management for computer vision," 2023.

[27] Raspberry Pi Foundation, "Raspberry Pi 4 Model B datasheet," 2022.

[28] A. Sinha and S. Mishra, "Automation of inspection systems using embedded controllers," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 2, pp. 857–866, 2019.

[29] S. Kumar et al., "Real-time industrial inspection using embedded vision," *Procedia Computer Science*, vol. 167, pp. 1922– 1931, 2020.

[30] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.