Colour-assisted PCB Inspection System with Hardware Support for Real-time Environment

H. S. Lim[1(✉)], Y. L. Lee[1], K. M. Yap[1], M. H. Lin[1], and T. K. Lian[2]

[1] Research Centre for Human-Machine Collaboration (HUMAC), Sunway University, Subang Jaya, Malaysia
18124693@imail.sunway.edu.my
[2] Deng Kai Sdn Bhd, Klang, Malaysia

Abstract. Printed Circuit Boards (PCBs) are crucial functional components in electrical devices that require inspection during production to prevent faulty products. However, Automated Optical Inspection (AOI) machines for PCB inspection are costly, especially for small production lines. Previous studies showed that image processing and computer vision techniques could provide cheaper alternatives for PCB optical inspection, albeit with the respective limitations in the research field. The proposed project combines the development of a colour-assisted image subtraction algorithm with the use of hardware support to produce a complete system with PCB counting and major missing component detection functions for the real-time environment. The hardware aspect included the use of the Raspberry Pi, a USB camera, a sensor-equipped conveyor and a custom camera stand to address several limitations in the research field. Testing and validation procedures were performed on the complete system such as function validation and comparative testing. The results of the experimental test procedures showed promising results with the average accuracy exceeding 90% for both the PCB counting and the major missing component detection functions. Overall, the project explored the use of colour in computer vision for PCB inspection, with hardware support for the real-time environment that could be expanded in future work.

Keywords: image processing · image subtraction · defect detection · colour segmentation

1 Introduction

Printed Circuit Boards (PCBs) are electronic boards embedded with internal wirings to connect various mounted electrical components as a complete circuit. Nearly all electronic products use PCBs since they are compact systems with high reliability and relatively low cost for mass production [1]. A typical PCB production line consists of a sequence of processes including schematic design, wire tract etching, components mounting and PCB inspection. The latter is a crucial step to detect any form PCB defects that occurred during production [2]. A fatal defect is a category of defects that leads to

S.-C. Haw and K. Sonai Muthu (Eds.): CITIC 2022, 10, pp. 313–330, 2022.
https://doi.org/10.2991/978-94-6463-094-7_25

the malfunction or failure of a PCB. Missing components are considered fatal defects because each component on the PCB has its functions and should all be present for the board to function as intended. These defects should be identified during PCB production to reduce wastage of cost and materials for reproduction and to avoid producing faulty products for customers.

Conventionally, trained employees would manually inspect PCBs for defects [1]. However, human inspection is not completely reliable due to the presence of human errors, fatigue and subjectivity. Moreover, as production lines grew larger, this task also grew tedious and unmanageable. Subsequently, Automated Optical Inspection (AOI) Machines were introduced to automate the PCB inspection process, eventually becoming an expensive industry standard for PCB inspection [1]. While the inspection of PCBs with AOI machines may not be an issue for larger corporations, smaller production companies such as start-up companies and Small-Medium Enterprises (SMEs) might face challenges in obtaining them. An AOI machine imposes a huge initial cost and with a relatively low volume of PCB production and sales, the SMEs and start-up companies might take a long time to recover from this initial investment.

Fortunately, AOI machines use cameras to conduct visual inspection to identify defects on a PCB, which can also be achieved via image processing and computer vision techniques. This could offer lower-cost alternatives to AOI machines. Papers such as [1] have reviewed numerous methods proposed by previous researchers to address PCB inspection using these techniques while also highlighting the limitations and possible future work in the field. This indicates that computer vision for PCB inspection is an established research topic to be further explored even though they still encounter limitations across the board. A literature review was conducted to explore different computer vision techniques used for PCB inspection and to identify the limitations faced by these techniques in an attempt to address them in the proposed system.

The proposed system combines the development of an image processing algorithm with a hardware setup to produce a compact and low-cost inspection system that can operate in a natural factory setting.

2 Related Work

Image subtraction is a referential, pixel-based technique used to find the differences between a reference image and an input image. In theory, this technique should be effective and feasible to detect missing, misaligned or misplaced components on a defective PCB. The general steps in image subtraction presented in [3] are as follows: Firstly, a reference image and an input image are captured and read as grayscale images. Then, the images are converted to binary images consisting of only black (0) and white (1) pixels. After that, the subtraction operation is performed with these images where corresponding pixels of both images are compared and the difference between corresponding pixels are highlighted in the output. These highlighted differences represent the defects and background noise, shown as white pixels on a black image. Finally, spatial filtering techniques were applied to the output to reduce unwanted noise. As mentioned, these steps are general steps in performing image subtraction, which were also applied in a similar manner by [4–6] in their PCB inspection algorithms.

Being a simple and versatile technique on its own, image subtraction allows for modification to better suit different purposes and to develop unique techniques. For example, for noisy or low-quality images encourages researchers to apply layers of pre-processing to enhance the process and results of image subtraction. In [7], after obtaining the colour reference and input images, the authors performed median filtering to reduce noise. Then, they zoomed into the images and sharpened specific regions of the PCBs. Although image subtraction itself is already a fast process with relatively low computational time, [7] also explored options to speed up the process through image compression. After the filtering and sharpening procedures, the authors compressed the images using wavelet transform before performing the subtraction. The results showed that this algorithm produced results in shorter computation time as compared to using non-compressed images, with the defect detection accuracy of 80–85% for mounted PCBs.

In another approach, other features of an image can also be used as the subject for image subtraction such as the edges in an image. In [8], instead of using the full image for image subtraction, the authors first smoothened the reference and input PCB images using Gaussian filtering. This reduces the interference from noise, which allows the Canny edge detector to work more effectively. After the edges are extracted, they are refined through non-maximum suppression and hysteresis procedures to enhance the quality of detected edges. The results showed that the edges are fairly effective in detecting the missing components from their test PCB images. Overall, this research paper supported the idea that other components of an image could also be used for image subtraction algorithms, such as the colour of the subject.

Reference [2] used an industrial-grade CCD camera that connected directly to the computer with a rigid setup of hardware for their study, where the camera and a coaxial ring-shaped lighting were positioned at a 90° angle above the PCB bed to simulate an industrial system. In this setup, the camera was controlled by computer while the ring-lighting was controlled and positioned manually. This showed, that a rigid setup effectively controls the image-capture environment, which solved image alignment and provided a consistent image capture environment. With this setup, [2] proposed using a different form of image subtraction for PCB inspection known as background subtraction. This technique uses the background learning functions available in the OpenCV library. The Mixture of Gaussian (MoG) method is a common background learning method that computes a mathematical model after learning the features of the background images supplied to it. In an input image or video feed, this model allows the program to calculate the probability of a pixel belonging to the background, effectively highlighting objects in the foreground or new objects that entered the field of vision. An example application of background subtraction is motion detection in surveillance cameras. With an innovative use of background subtraction, [2] supplied live video frames of a non-defective reference PCB to the MoG method as the "background". Then, since a defective PCB would have differences compared to the background, those differences will be highlighted in the final output. The average error rate in 20 rounds of testing was between 5 and 8%, even under varying light conditions in live images. Although [2] has been referenced in later researches, there is still a lack of research expanding on background subtraction for PCB defect detection.

Template matching is a basic brute-force method in computer vision used to find exact matches of a template in an input image. Reference [9] is one of the few papers that discusses about using template matching for PCB defect detection. In this method, the author manually cropped out individual components from a reference image of a PCB to use as the templates. Then, the PCB in a pre-processed input image was isolated using background subtraction. This is so that the program would not try to look for template matches in irrelevant locations on the input image. Hough transform is then performed on the isolated PCB image to straighten and align the PCB image. After pre-processing, the template matching process was performed in two stages: A rough matching stage and a precise matching stage. In rough matching, the algorithm starts scanning the grayscale input image with a sliding window the size of the templates and attempts to calculate the similarity between the window region and the template itself. If the similarity exceeds a defined threshold, the precise matching stage is triggered. Precise matching is performed on the colour images where the colour components were included to measure similarity between the template and the region on the input. If the similarity between the supposedly similar regions falls below a threshold value, it indicates that a defect is detected. These defects include missing components and soldering errors.

In theory, this technique might be effective for defect detection. However, it may not be suitable for a real-time system since template images of each PCB component need to be manually cropped out before the matching can be done. This would be tedious, especially if the PCB models are swapped frequently during operation. Even so, template matching could still be a viable technique for PCB defect detection with further exploration.

Machine learning (ML) is a subdivision of Artificial Intelligence (AI) that involves the creation of predictive models based on the features and patterns learned from a training dataset. These models are then applied to input data to make predictions and classifications. Note that there are numerous forms of machine learning techniques that could be applied for image defect detection and this literature review only covered two: a decision tree and a neural network.

A decision tree is a fundamental ML model that uses a tree structure to simulate human decision-making. Reference [10] proposed an algorithm using a ML technique called Random Forests, which is a prediction model that generates multiple decision trees and averages the predictions from the trees as the final output. The training dataset used in this model comprises of features obtained using a popular feature extraction algorithm known as the Speeded-Up Robust Features (SURF) algorithm. The SURF algorithm is able to detect points of interest in a PCB image and generate descriptors with 64 dimensions, each describing the characteristics of the detected region. These descriptors or features are then used to algorithmically generate the decision criteria for each decision tree in the Random Forests algorithm. In this model, decisions are made based on the features of the input as compared with the patterns of extracted features from the training dataset. The expected output of the model is the probability of defect in a detected region. Reference [11] also proposed the usage of a statistical map called Weighted Kernel Density Estimation (WKDE) to map the points for better probability estimation of defects. Regions with a high probability of defects, primarily scratches, would be circled on the PCB image in the final results.

In [11], a neural network algorithm was developed to detect missing components on a PCB. Within the field of ML, a neural network is an established Deep Learning model that tries to mimic the neural connections and operations of a human brain when making decisions and classifications. In the paper, the authors prepared a dataset of 147 manually-taken images of a PCB, 104 of which are used in the training dataset and 43 of them used in the testing dataset. To prepare the training dataset, they performed image subtraction with binary images of the reference and the training images. Then, they used a geometric feature extraction algorithm to calculate the area, perimeter and compactness of each object in the subtracted image. These objects are treated as missing components and the area, perimeter and compactness describe their characteristics. These descriptors are passed into the neural network to train the model before it is applied to the testing data set. The results showed that the model was able to recognise missing components on the testing images with 97% accuracy. Even though [11] performed their training and testing with datasets of bare PCBs, which are PCBs without components, the paper proves that a neural network is viable for PCB defect and missing components detection as long as the training data is sufficient. Unfortunately, the biggest challenge of using machine learning for PCB inspection is that the training dataset should contain a considerable number of images for the model to be effective. On top of that, different PCBs may vary in many different aspects including shapes, sizes, design, components, and wire tracts based on client needs. This implies that new training datasets have to be created for each model of PCB to be inspected and a new machine learning model should be generated and trained with each dataset.

2.1 Limitations of Overall Techniques

Regardless of computer vision techniques used, there are limitations that exist across the board. These limitations affect the overall performance and reliability of the algorithms for PCB inspection and their effects were proven through experiments in past research papers. These limitations are discussed in this section, along with their potential solutions as proposed by several papers.

2.1.1 Illumination Issues

Illumination is not a challenge in AOI machines since they have a controlled environment with built-in illumination for visual inspection. However, the issue of illumination is a common challenge for a vast majority of computer vision projects including PCB inspection. This issue is difficult to resolve since too much illumination or too little illumination will both lead to unreliable and inaccurate results. The issue is not as simple as adding a controlled light source, since the light brings up additional considerations, such as colour variations, reflections and cast shadows on the PCB. With more focus on illumination, several research papers experimented and addressed the issue through hardware setup.

Based on the literature review, several authors proposed using external light sources with their setup to manage illumination. Reference [2] stated that their illumination is generally provided by a coaxial ring-shaped light positioned at 90° above the PCB bed. This lighting was manually controlled while the author captured reference and input

images. Reference [4] on the other hand, used red LED lights to manually illuminate their PCB images. Although these options might produce considerable results, manual positioning of lighting might cause inconsistencies. Thus, a consistent and fixed lighting setup is required for the proposed project to reduce lighting discrepancies.

Reference [12] attempted to mimic an industrial environment by supporting their hardware with a "light box" structure. The structure was a Perspex cuboid with multiple LED strips lining the edges of the box. The smartphone camera was placed on top of the box with the PCB bed positioned in the middle of the camera frame. Although this light box provided more consistent lighting and attempted to minimise shadows, the authors pointed out that it does not replicate an industrial environment well enough. Overall, these illumination techniques imply that it is possible to control the environment to an extent and reduce the effect of illumination issues with a consistent lighting setup.

2.1.2 Image Alignment

Reference [2] stated that the alignment of PCBs for inspection in the industrial environment is enabled through positioning lasers, which is not accessible for many researchers. The issue of image alignment affects most computer vision techniques for PCB inspection, but is most significant in pixel-based operations such as image subtraction and its derivatives. These techniques require that the position and photo angle of reference images and input images be exactly aligned. This is because the image subtraction methods compare the reference image and input image in a strict pixel-by-pixel manner. Any offset between the alignment of images ends up producing inaccurate results.

To manage alignment issues, research papers such as [4, 9] suggested using an image alignment algorithm such as the Hough transformation to pre-process images before proceeding with defect detection. These algorithms attempt to match the alignment of the reference and input images as closely as possible. Unfortunately, pre-processing steps may not be suitable for the proposed system in this project as it imposes additional processing and there is no one-size-fits-all algorithm to align images. These alignment algorithms might require custom input and adjustments for each image to accurately align the images.

2.1.3 Image Quality

In this case, image quality refers to the noise level and the resolution of an image. Several research papers have experimented with different image quality and discussed about their impact on their otherwise working algorithm.

Reference [3] developed an algorithm using image subtraction and experimented with different levels of noise. Through their experiment, the authors stated that noise levels above certain threshold will drown out the detected defects and lead to algorithm failure. To reduce the effect of noise, different forms of spatial filtering techniques have been suggested. References [3, 4, 8] in particular, suggests that median filtering is most effective in removing noise and is said to prevent blurring and retain the significant details in images. Although, it is worth noting that the defects detected during image subtraction processes are also a form of noise and too much filtering would instead remove the detected defects leading to false negative results [3].

Besides noise, the resolution of images is also considered an image quality factor. In [6], the authors performed the testing of their image subtraction algorithm using two sets of images, one set taken with a low-resolution camera (less than 720p) and another set taken with a high-resolution camera (more than 720p). Their results proved that resolution affects the accuracy of the defect detection, showing 80% accuracy with the high-resolution images but only 40% accuracy with the low-resolution images.

It is also worth noting that low image quality can also affect ML techniques. The training dataset for ML models is crucial to create an accurate and reliable model. A low-quality (LQ) training dataset implies that the information learned and extracted are also of low quality, which ultimately leads to a low-accuracy model. Reference [13] tested their genetic programming algorithm using two datasets of images, one of high quality (HQ) and another of LQ. In this experiment, quality refers to the resolution, sharpness and illumination of the PCB images. These datasets were used to train two separate models. The model trained with HQ images was tested with HQ input images and the model trained with LQ images was tested with LQ input images. The results showed that the HQ-trained model produced a 92.9% accuracy while the LQ-trained algorithm produced an 87.7% accuracy. Although both results can be considered good, the experiment showed the difference between using HQ images and LQ images for PCB inspection.

3 Methodology

3.1 Hardware Requirements

This proposed system ran on a Raspberry Pi with an industrial-grade camera supported by a custom stand. The image subtraction algorithm benefitted from the stand, a sensor-equipped conveyor and studio lights to address the limitations of image subtraction, namely alignment and illumination. The custom stand holds the camera in a fixed position while the conveyor ensures each PCB follows a fixed path, stopping each PCB in an exact position within the camera frame. The stand setup is shown in Fig. 1, while the operation of the conveyor is shown in Fig. 2, where sensor A automatically starts the conveyor and sensor B automatically stops the conveyor.

The sensor-equipped conveyor used is a buffer-linking conveyor that connects the path between the component assembly machine and the component mounting machine in the factory. The former machine picks and places components on empty PCBs, while the latter mounts the components permanently on the PCB. The proposed inspection system is position in between these machines to spot defects with the assembly and alerts employees before the components are permanently mounted on the PCBs.

The experimental setup also uses diffused studio lights to evenly illuminate the PCBs that passed through the camera frame. The full experimental setup is shown in Fig. 3.

3.2 PCB Counting and Inspection Algorithm

The image subtraction algorithm is a referential method in which a reference image has to be captured as a basis of comparison to identify defects in the PCB. After a reference

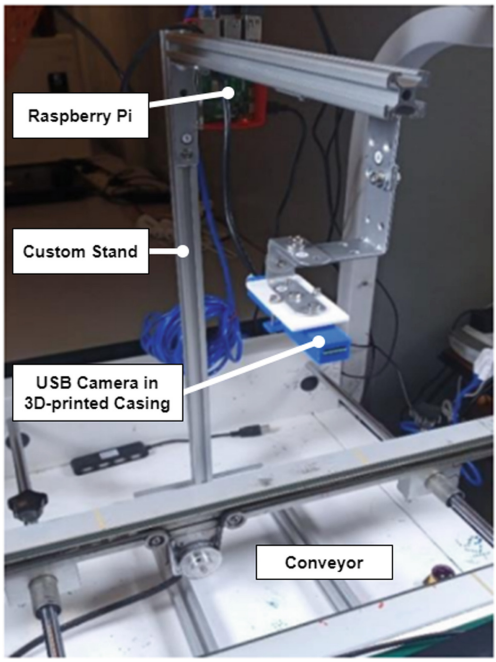

Fig. 1. Custom adjustable stand supporting the Raspberry Pi and camera aligned with conveyor tracts

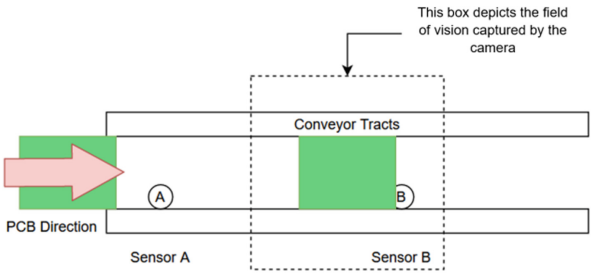

Fig. 2. Graphical Representation of Sensor-equipped Conveyor Operation

image of a non-defect PCB is captured, the proposed algorithm generates a colour mask to segment the coloured regions of the PCB from the components as shown in Fig. 4. These images are saved for the image subtraction process. The region-of-interest (ROI) is also marked on the user interface as seen in Fig. 5 as the red box, so that the PCB inspection only occurs within the region to avoid false detections.

After the reference image capture, the PCB Counting and inspection algorithm is initiated. The live feed is supplied to the algorithm for the generation of its colour mask for image subtraction. Once the detected PCB moves into the ROI, a colour mask is similarly generated for the PCB, as shown in Fig. 6. This colour mask is then subtracted with the colour mask of the reference image.

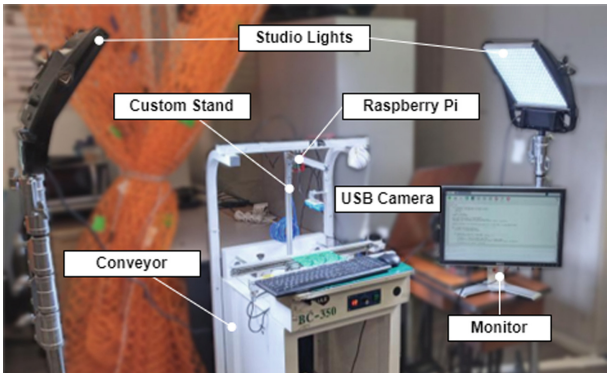

Fig. 3. Complete Experimental Setup

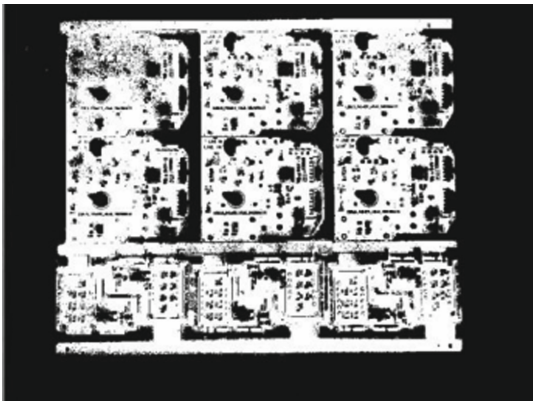

Fig. 4. Example of the Colour Mask Generated

At the same time, the PCB from the live feed is also being tracked for a PCB counting function using a line-crossing algorithm.

After the subtraction, morphological operations are used to enhance the detected defects and the buzzer is activated as necessary to alert users to manually remove the PCB from the conveyor.

There are three start-up options for the system. First one is to initiate by capturing a reference image, second is to use a read-saved image and third is for an expert to manually adjust the range of red-green-blue (RGB) values to effectively segment the PCB in the images. As shown in Fig. 7, the latter option creates a sliding trackbar to set the lower range and upper ranges for each of the RGB values and displays the mask generated for the PCB on the camera feed.

The general process flow of the algorithm and its start-up options are shown in Fig. 8.

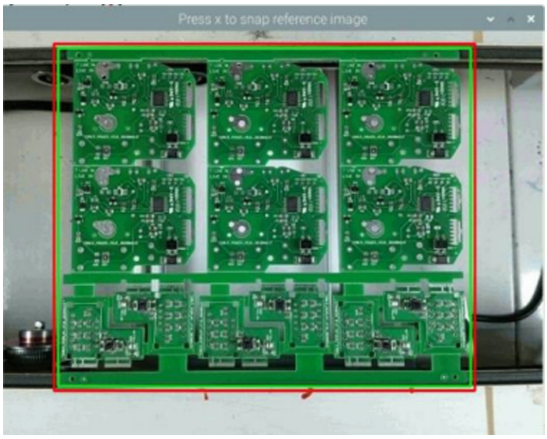

Fig. 5. Capturing Reference Image and ROI

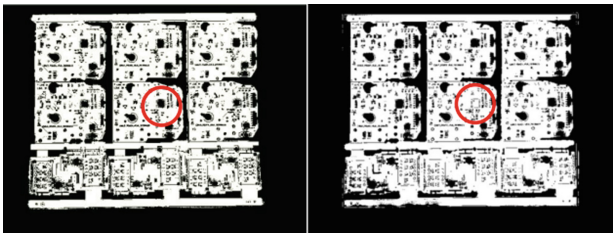

Fig. 6. Example of Reference Colour Mask and Live Feed Colour Mask with Defect on PCB

3.3 Testing Procedure

Based on the flowchart in Fig. 9, the testing procedure started with a pool of sample PCBs provided by the client. For each testing and validation phase, a fixed number of rounds were set for each set of testing. After capturing a reference image, a PCB was randomly selected from the pool and its condition (defect or non-defect) was manually recorded. Then, the PCB was placed at the start sensor of the conveyor, which moves the PCB to the stop sensor. The system was allowed to perform its task and the key result of the test was observed and recorded. For example, during the missing component detection module validation, the result of whether a missing component was detected or not was observed and recorded. After a successful test, the PCB was removed from the conveyor. The removed PCB was placed back into the pool of PCBs for reuse and the procedure repeated itself until the appropriate number of tests was completed.

4 Results and Discussion

Initial tests with the system were successful for both PCB counting and missing component detection. Figure 10 shows the output for the PCB counting module while Fig. 11

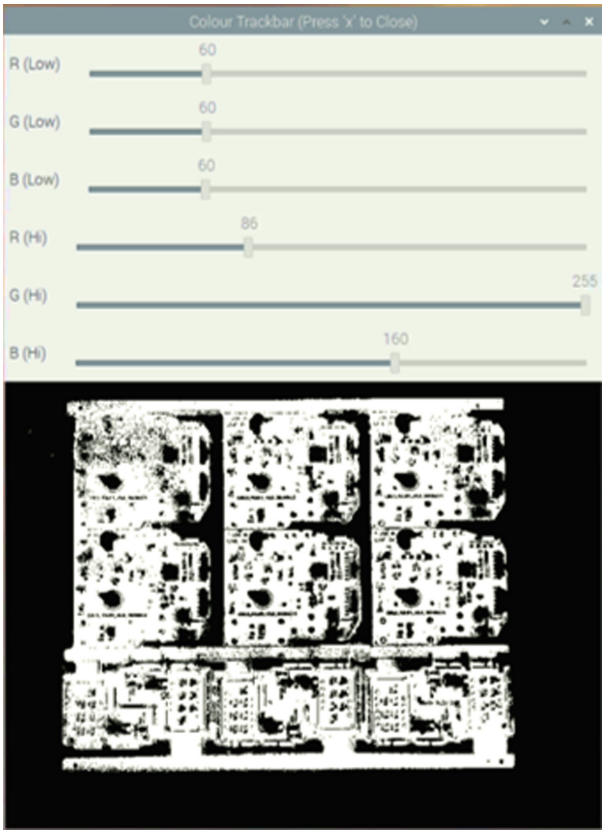

Fig. 7. Colour Trackbar to Adjust RGB Values for PCB Segmentation

shows an example of a PCB with a missing component with its corresponding output shown in Fig. 12. The missing component is detected as a white spot in the output.

After the integration of the system, the complete system was tested to prove its concept, functionalities and practicalities. The testing process tested both the counting system and the inspection system as an integrated system. The testing procedure was previously elaborated in the Methodology. Table 1 summarises the missing component detection results in a confusion matrix. In this test, 4 rounds with 30 PCBs per round were conducted, with a total of 120 samples. The results for each round are shown in Table 2.

Based on the confusion matrix in Table 2, the average accuracy for the PCB counting function was 99.17% while the average accuracy of the missing component detection function was 95%. Overall, these results were excellent and showed that both of the major functions tested individually also worked well after their integration in the complete system.

One of the limitations of the counting module is the reliance on the colour masking module to track the PCB on the camera feed. The proposed system does not recognise

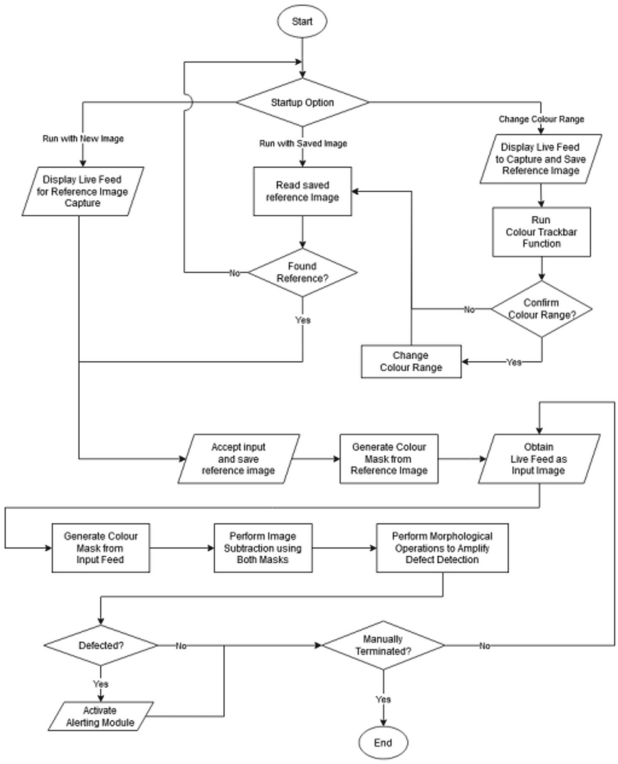

Fig. 8. Process Flow of the Colour-assisted Image Subtraction algorithm

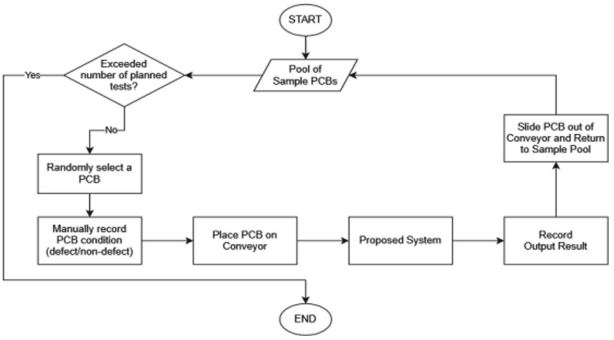

Fig. 9. Process Flow of Testing Procedure

PCBs, but only sees it as object of a specified colour range and a specified area size inside the feed. The system does not know what this object is, but it will just track the object. If the specified colour range is incompatible with the PCB coloured regions, the mask would not accurately represent the PCB on the feed and the PCB would not be detected. Hence, since no PCB is detected, the line-crossing algorithm would not trigger.

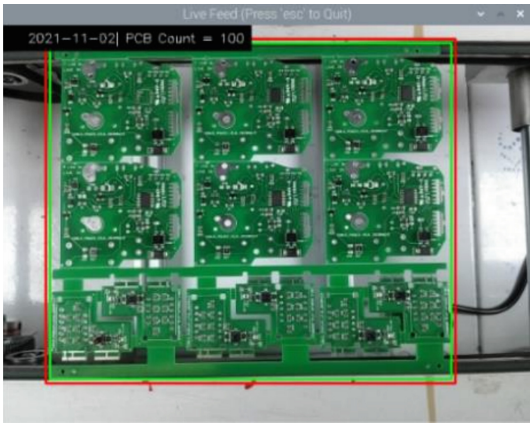

Fig. 10. Example Output for Counting Function

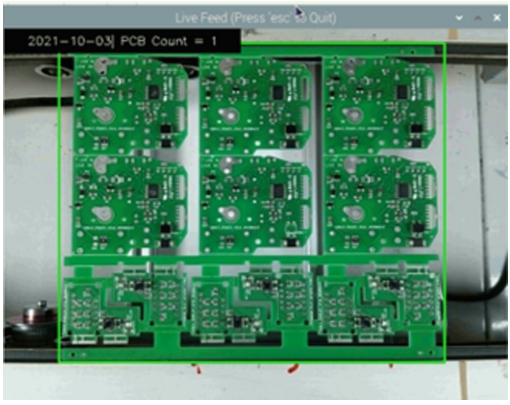

Fig. 11. Example PCB with Missing Component

Besides that, illumination remains a factor that affects the performance of the colour-masking module, which in turn affects both the counting module and the missing component detection module. Low illumination or change in the colour of illumination can alter the perceived colour of the PCB in the camera feed. For example, a low light environment would darken the colour of a PCB, which would make the PCB tracking unstable. In some cases, the PCB might not be detected at all which also means that it would not be counted and the components cannot be properly segmented for image subtraction.

Besides the system testing, a comparative testing was also performed to compare the results of a general image subtraction algorithm against the proposed algorithm with the assistance of colour segmentation. One round of testing with 30 PCBs was performed for each algorithm with their results shown in Tables 3 and 4.

As shown in the comparative testing results, the proposed approach managed to obtain up to 90% accuracy, whereas the general approach only obtained 53.33% accuracy.

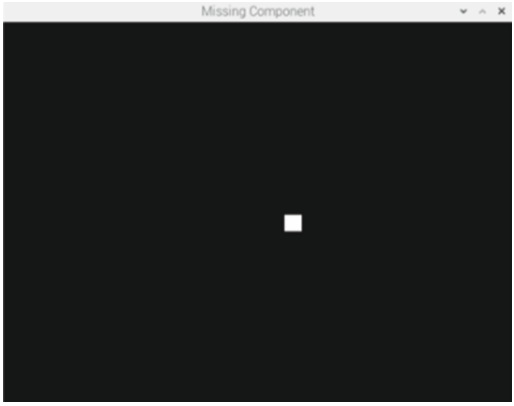

Fig. 12. Example Output for Missing Component Detection Function

Table 1. Confusion Matrix for Missing Component Detection Testing

		Actual PCB condition	
	n = 120	Non-defective	Defective
Result recorded	Non-defective	69	4
	Defective	2	45

Table 2. PCB Conditions with the Actual and Recorded Defect and Count for Each Testing Round

Round	PCB condition	PCB count	Correctly detected	False Positive (FP)/False Negative (FN)	Counting module result
1	Non-defective	21	21	FP: 0	30
	Defective	9	9	FN: 0	
2	Non-defective	16	15	FP: 1	30
	Defective	14	13	FN: 1	
3	Non-defective	24	22	FP: 2	29
	Defective	6	5	FN: 1	
4	Non-defective	12	11	FP: 1	30
	Defective	18	18	FN: 0	

This shows that the colour segmentation was shown to improve the detection of the missing components. In many cases, the general image subtraction algorithm used was unable to detect the difference between defective and non-defective PCBs.

Table 3. Confusion Matrix for Proposed Algorithm with Colour Segmentation

		Actual PCB condition	
	n = 30	Non-defective	Defective
Result recorded	Non-defective	14	1
	Defective	2	13

Table 4. Confusion Matrix for General Image Subtraction Algorithm

		Actual PCB condition	
	n = 30	Non-defective	Defective
Result recorded	Non-defective	9	7
	Defective	7	7

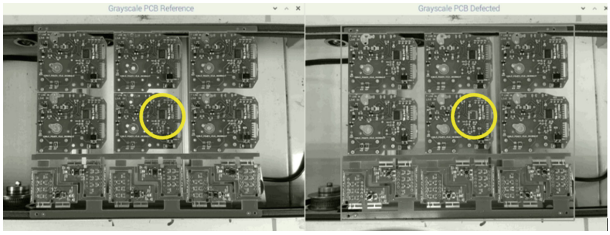

Fig. 13. Example of Grayscale Images for General Image Subtraction Algorithm

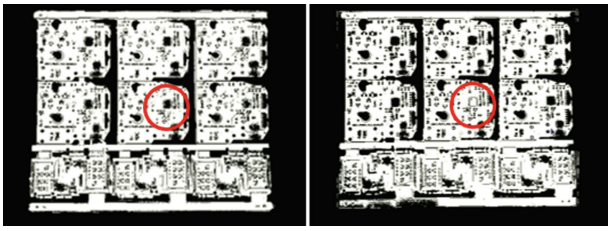

Fig. 14. Example of Colour-segmented Images for Proposed Algorithm

This is mainly because most image subtraction algorithms would first convert images into grayscale images before converting them into binary images for the subtraction. This might reduce the contrast between different regions of the same image. In this case, as shown in Fig. 13, the grayscale image shows that the components have less contrast with the PCB as compared to the colour-segmented binary images in Fig. 14.

Although this testing showed that the general image subtraction did not have a good performance in the comparative testing against the proposed algorithm, it does

not dispute image subtraction algorithm as the go-to method for defect detection. Image subtraction is simple, yet powerful and highly customisable for its applications. The discrepancies between the results of the comparative test could due to the lack of pre-processing, post-processing and experimentation with the general image subtraction algorithm.

5 Conclusion and Future Work

The testing and validation procedures performed showed that the proposed system achieved relatively high standard of accuracy under ideal conditions. Both the PCB counting and the major missing component detection functions managed to achieve an average accuracy exceeding 90%.

Overall, the inclusion of colour segmentation for image subtraction in the proposed algorithm was shown to improve the accuracy of the subtraction process. This colour segmentation function is a novel pre-processing approach meant to enhance image subtraction, primarily with subjects that has elements with clear colour differentiation, such as the PCB boards and their components.

However, as expected, the proposed system still faced a variety of limitations that could be addressed in future work. One of which was the lack of robustness in the PCB tracking function used within the PCB counting module. As mentioned, the PCB was tracked on a camera feed based only on the boards' colour. The system does not actually recognise them as a PCB, but rather as an object within the colour range specified. A machine learning model trained to identify PCBs specifically would provide more robustness and sophistication to this function.

Besides that, the missing component detection function was only able to detect defects of a certain size in the real-time environment. A possible future work to address this could be through the use of image zooming via algorithm or with more advanced machinery that can control the movement of the PCB, the conveyor and the camera. Another aspect that could be explored is the use of machine learning for real-time PCB inspection. As shown in the literature review, numerous computer vision algorithms could still be explored, developed and tested for the research topic of PCB optical inspection.

Additionally, there were many limitations with the resources available, namely the lack of variety in the PCB test sample available. Testing using a variety of PCBs can also be included in future work to ensure that the system is adaptable to changing PCBs in a production line. Another limitation is the absence of an AOI machine for comparative testing to gauge the proposed system as compared with the industry standard.

Acknowledgments. As part of the Public-Private Research Network (PPRN), this project was initially proposed by Deng Kai Sdn. Bhd., who supported the project by providing PCB samples and the sensor-equipped conveyor, as well as providing insight through their engineers and the PCB production line at their factory.

References

1. A. D. B. and M. Rao, "A survey on defect detection in bar PCB and assembled PCB using image processing techniques", 2017 Int. Conf. Wireless Comms Signal Processing and Networking (WiSPNET), pp. 39–43, Mar. 2017, doi: https://doi.org/10.1109/WiSPNET.2017.8299715.
2. K. Sundaraj, "PCB inspection for missing or misaligned components using background subtraction", WSEAS Transactions Info. Sci. Appl., vol. 6, no. 5, pp. 778 – 787, 2009.
3. S. Kaushik and J. Ashraf, "Automatic PCB defect detection using image subtraction method", IJCSN, vol. 1, no. 5, Oct. 2012.
4. I. Ibrahim, S. A. R, S. Abu Bakar, M. M. Mokji, J. A. A. Mukred, Z. M. Yusof, Z. Ibrahim, K. Khalil and M. S. Mohamad, "A printed circuit board inspection system with defect classification capability", IJIMIP, vol. 3, no. 1, pp. 82–87, Mar. 2012.
5. F. Raihan and W. Ce, "PCB defect detection using OpenCV with image subtraction method" in ICIMTech, September 15 – 17, 2017, pp. 204–209.
6. A. D. B. and M. Rao, "SMT component inspection in PCBA's using image processing techniques", IJITEE, vol. 8, no. 12, pp. 541–547, Oct. 2019, doi: https://doi.org/10.35940/ijitee.L3422.108121.
7. M. Borthakur, A. Latne and P. Kulkami, "A comparative study of automated PCB defect detection algorithms and to propose an optimal approach to improve the technique" IJCA, vol. 114, no. 6, pp. 27 – 33, Mar. 2015, doi: https://doi.org/10.5120/19985-1938.
8. R. R. Chavan, A. A. Chavan, G. D. Dokhe, M. B. Wagh, A. S. Vaidya, "Quality control of PCB using image processing", IJCA, vol. 141, no. 5, pp. 28 – 32, 2016, doi: https://doi.org/10.5120/ijca2016909623
9. H. Yin, "A template-matching-based fast algorithm for PCB components detection", Adv. Materials Research, vol. 690 – 693, pp. 3205 – 3208, 2013, doi: https://doi.org/10.4028/www.scientific.net/AMR.690-693.3205.
10. E. H. Yuk, S. H. Park, C. S. Park and J. G. Baek, "Feature-learning-based printed circuit board inspection via Speeded-Up Robust Features and Random Forest", Appl. Sci., vol. 8, no. 932, Jun. 2018, doi: https://doi.org/10.3390/app8060932.
11. M. Mogharrebi, A. S. Prabuwono, S. Sahran and A. Aghamohammadi, "Missing component detection on PCB using neural networks", Adv. Electr. Eng. Electr. Machines, vol. 134, pp. 387–394, doi: https://doi.org/10.1007/978-3-642-25905-0_51.
12. S. Ur Rehman, K. F. Thang and N. S. Lai, "Automated PCB identification and defect-detection system", IJECE, vol. 9, no. 1, pp. 297–306. Feb. 2019, doi: https://doi.org/10.11591/ijece.v9i1.pp297-306
13. F. Xie, A. H. Dau, A. L. Uitdenbogerd and A. Song, "Evolving PCB visual inspection programs with genetic programming" in 28th Int. Conf. Image and Vision Computing, New Zealand, 2013, pp. 406–411, doi: https://doi.org/10.1109/IVCNZ.2013.6727049.