Young Defence Scientists Programme

DSTA Defence Science & Technology Agency

DSO National Laboratories

ECOD: EMBEDDING CLUSTERING FOR OBJECT DETECTION IN AUTOMATING PRINTED CIRCUIT BOARD ANALYSIS

Member:
Wayne William Loo Shyan (Raffles Institution)

Mentors:
Shen Bingquan, Lim Qin Zhi Eddie
(DSO National Laboratories)

1 Motivation and Scope

Hardware TROJANS are not a new problem. They are **modifications made for malicious intent**, most often suspected to be performed at the production stage of Printed Circuit Boards (PCBs) before they are shipped to consumers. Attacks through hardware trojans often involve **Limited Performance**, **Data Leakage** and **Denial of Service** hence PCB analysis is needed by operators of critical systems, such as the military, to ensure security.

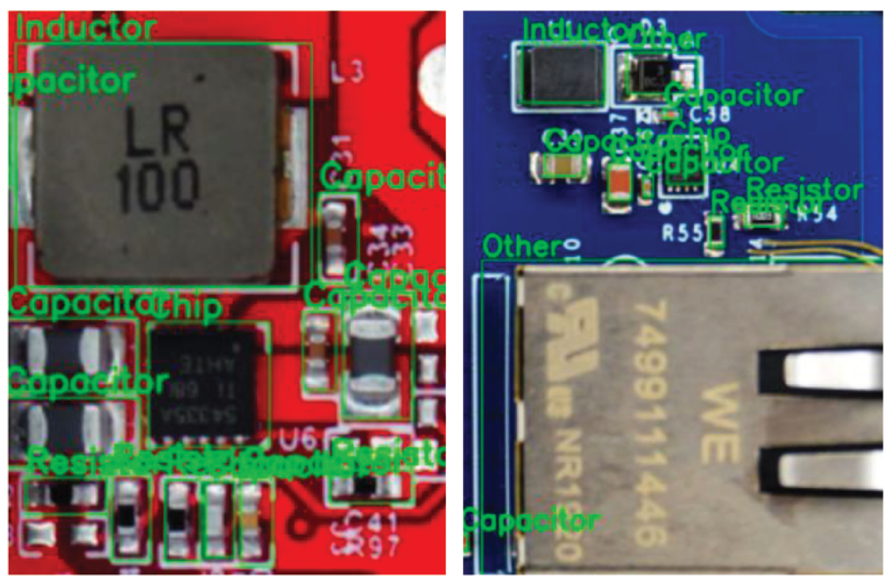
Current ANALYSIS Techniques: PCB analysis is needed, but current techniques are largely manual, thus labour-intensive, time-consuming, and costly. Object detection (OD) is a **promising method of automation** for **visual inspection.**

Current OD Techniques: Current OD techniques [1, 2, 3, 4] for PCBs either use large labelled datasets or are not fully automatic during their usage.

Our Research Objective: **To find out if alternative methods can improve data-efficient object detection of components on PCBs.**

Hypotheses:

1. The proposed pipeline YOLO-ResNet-Cluster would improve the classification accuracy of the base YOLO model.
2. The proposed pipeline SAM-ResNet[N]-Cluster would increase the recall of the pipeline to near 100%, though its precision would fall despite selection of objects with ResNet and clustering.



Above: Example ground-truth labels of small sections of two PCBs in the validation set, illustrating the correct boxes and labels that we aim for a model to predict accurately. These also illustrate the high intra-class variance for the "Inductor", "Other" and "Capacitor" classes, which is a key challenge in this specific OD task.

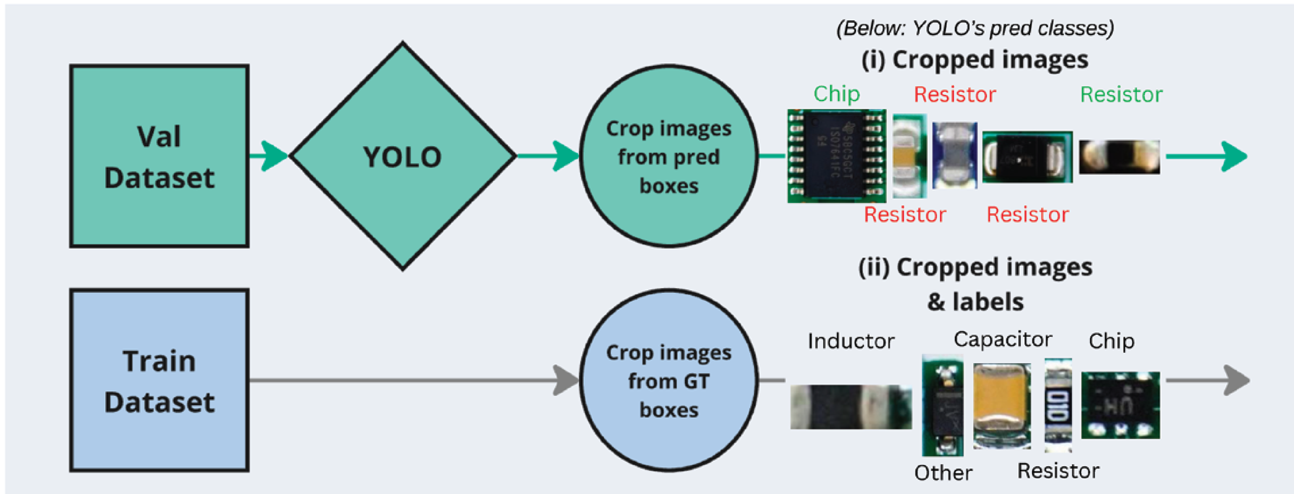
2 Methodology

We propose two novel pipelines, which, while they work rather differently (see pipeline Part A's), share a similar flow at the clustering stage (see (Part B) which is shared between 2.1 and 2.2).

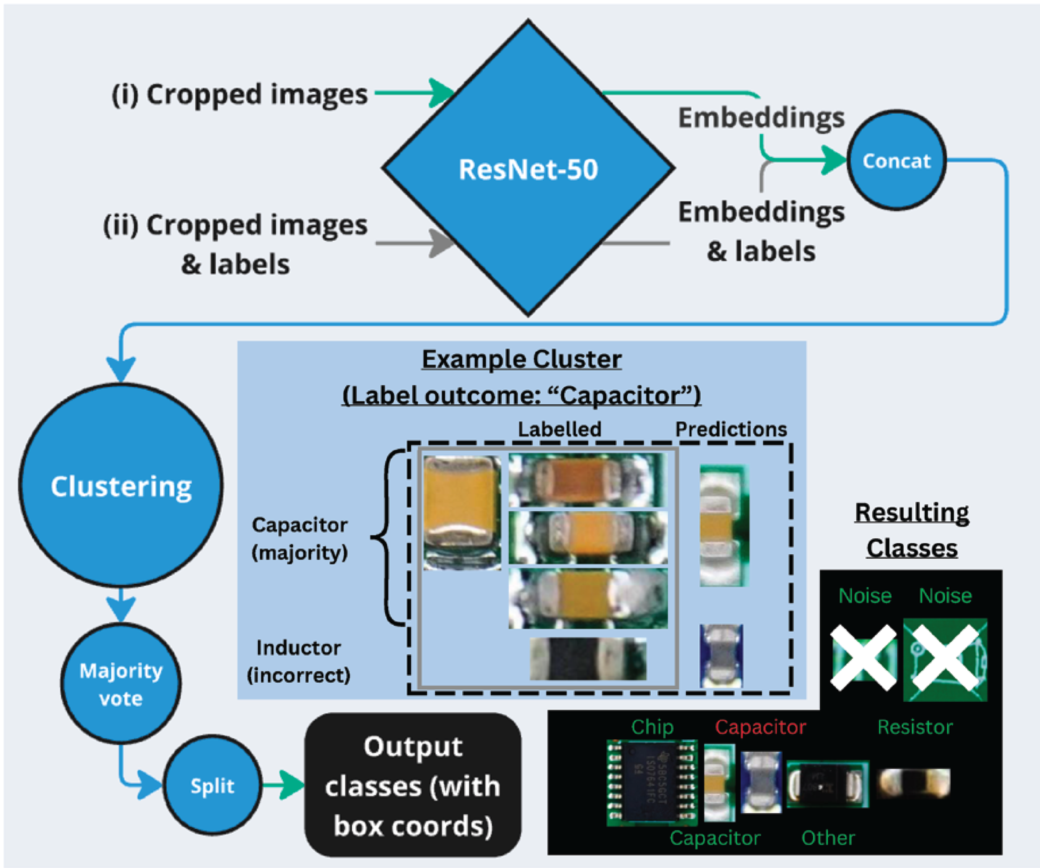
2.1 Pipeline 1: YOLO-ResNet-Cluster (Part A)

Key Features

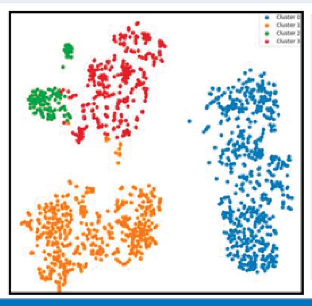
- YOLO for region proposal
- Replaces classification output of YOLO
- Clustering to group similar object crops together **(Part B)**
- Reduces reliance on large data for classification
- Fully automatic; no human labelling to indicate clusters during inference unlike some previous methods



(Part B)



- ResNet classification head removed
- Cluster val+train together
- Output: labels+box coords



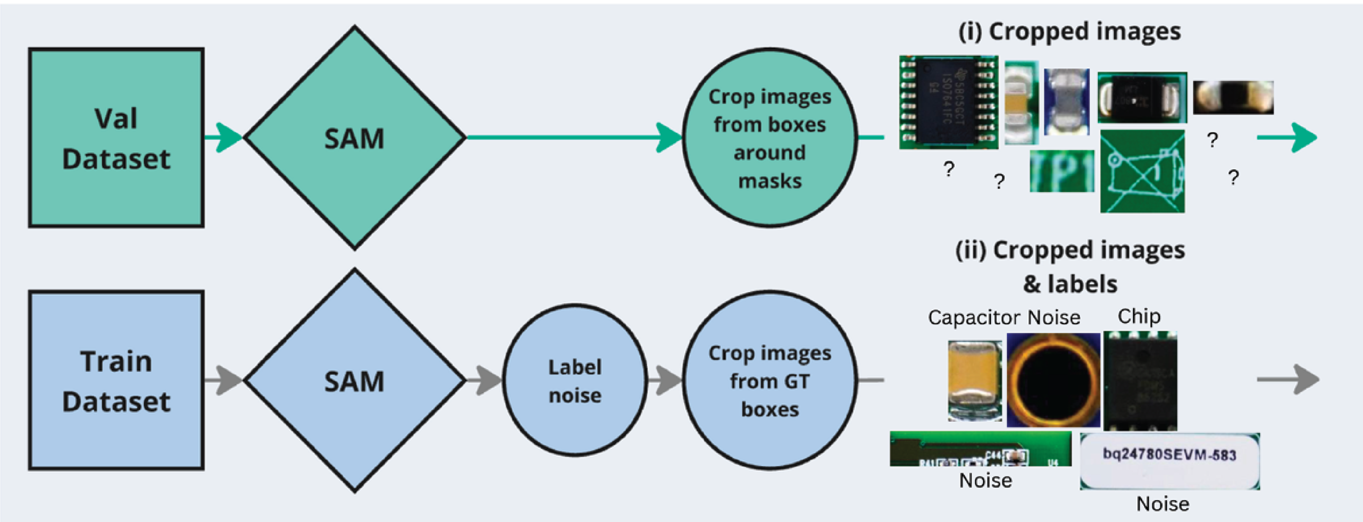
Right: Example t-SNE plot of clusters. Each point is one crop. n_clusters = 5

Legend:
- : Unlabelled data only
- : Labelled data only
- : Both
- ☐ : Dataset
- ◇ : Model, inference
- ○ : Some operation
- ☐, ■ : Output

2.2 Pipeline 2: SAM-ResNet[N]-Cluster (Part A)

Key Features

- SAM to generate masks (of almost everything)
- Clustering to group similar object crops together **(Part B)**
- Separation from noise [N] is performed during classification
- Purpose is to prioritise high recall
- Opens possibility of segmentation rather than merely boxes

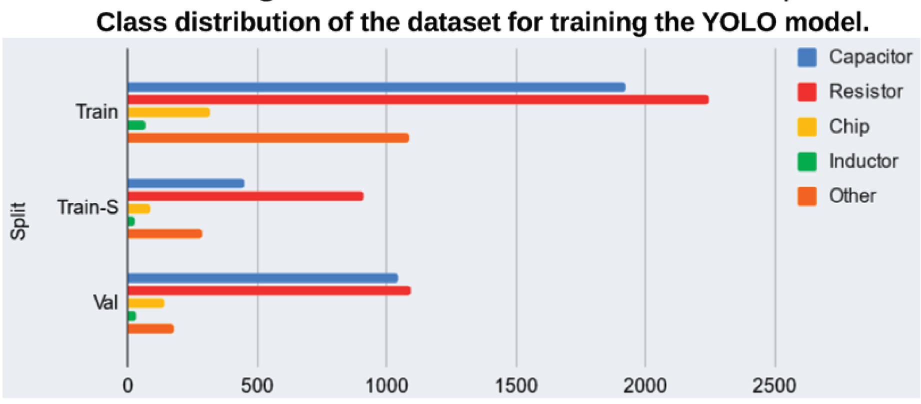


2.3 Datasets and Model Training

Total: **32** PCB images **Split:** **26** Train, **6** Validation (Val)
5 Classes *(listed below)* **5** Train-S (***Train-Small** subset*)

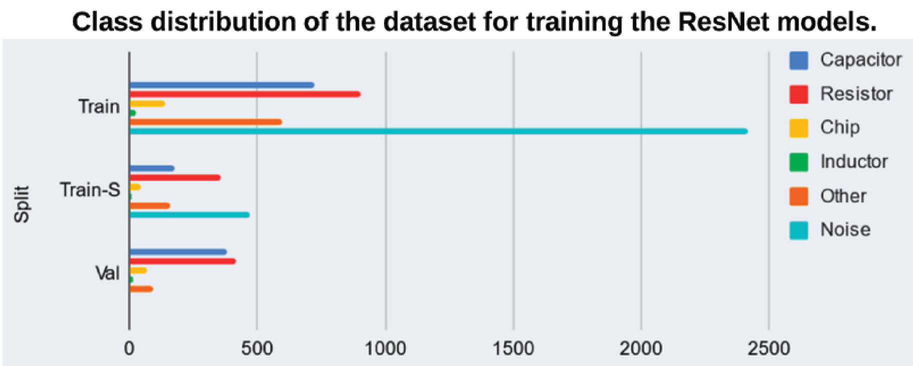
YOLOv8x
Dataset: Bounding boxes with classes, sliding window
Training: "Train" or "Train-S", 60 and 157 epochs

Class distribution of the dataset for training the YOLO model.



ResNet-50
Dataset: Cropped objects (+ SAM noise for ResNet[N])
Training: "Train" or "Train-S", 100 epochs, 1e-4 learning rate. Total 4 fine-tuned ResNet models.

Class distribution of the dataset for training the ResNet models.



3 Results and Discussion

Quantitative results of each pipeline on validation set, for each train set

Datasets	Full dataset (26 train)				Reduced dataset (5 train)			
Method	**P**	**R**	**F1**	**Cls**	**P**	**R**	**F1**	**Cls**
YOLO	<u>0.592</u>	0.492	<u>0.537</u>	0.659	<u>0.760</u>	0.448	<u>0.564</u>	0.722
YOLO-ResNet-Cluster				<u>0.847</u>				<u>0.825</u>
SAM-ResNet[N]-Cluster	0.202	<u>0.613</u>	0.304	0.802	0.264	<u>0.613</u>	0.369	0.782

Metrics

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

$$F1\ Score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

$$Class\ Accuracy\ (Cls) = \frac{Correct\ class\ for\ TP\ boxes}{Total\ TP\ boxes}$$

<u>YOLO-ResNet-Cluster:</u>

- **Successfully improved** classification accuracy from YOLO
- Irregularity:
  - YOLO F1 score and classification accuracy increased despite reduction in data. Possible reasons:
    - Reduced dataset happened to be very representative
    - Incorrect training hyperparameters

<u>SAM-ResNet[N]-Cluster:</u>

- Higher recall, but **extremely poor precision**
- Ignoring information about the object's size seems to result in many non-objects looking like objects even to a human after processing
- 'Reflect' padding could highlight unwanted edge features
- Could be improved through better image processing

4 Conclusion

In summary, we propose two novel pipelines for data-efficient detection of PCB components. The former focuses on enhancing an existing model's performance on a small training dataset and succeeds, while the latter focuses heavily on high recall and working with even less training data. Both seek to exploit the similarity in appearance between PCB components to enhance performance. We hope that our proposition of using clustering in object detection leads to further research of its use for data-efficient methods, even beyond the application of automatic hardware trojan detection on PCBs.

<u>Acknowledgements</u>

All images of PCBs and their components are taken from the provided dataset. All other images in the poster above are original. For authoring words used in the poster, Generative Artificial Intelligence was only applied once in paraphrasing the Abstract before heavy edits were made to it. Hence, its use in this poster is almost negligible, as the poster includes only minor portions that were influenced by the original Abstract.

<u>References</u>

[1] G. Piliposyan and S. Khursheed, "Computer Vision for Hardware Trojan Detection on a PCB Using Siamese Neural Network," 2022 IEEE Physical Assurance and Inspection of Electronics (PAINE), Huntsville, AL, USA, 2022, pp. 1-7, doi: 10.1109/PAINE56030.2022.10014967
[2] G. Mahalingam, K. M. Gay, and K. Ricanek, "PCB-METAL: A PCB Image Dataset for Advanced Computer Vision Machine Learning Component Analysis," IEEE Xplore, May 01, 2019. https://ieeexplore.ieee.org/document/8757928?denied=
[3] C.-W. Kuo, J. Ashmore, D. Huggins, and Z. Kira, "Data-Efficient Graph Embedding Learning for PCB Component Detection," arXiv.org, 2018. https://arxiv.org/abs/1811.06994 (accessed Dec. 22, 2024).
[4] W. Zhao, S. Gurudu, S. Taheri, S. Ghosh, Sathiaseelan, Mukhil Azhagan Mallaiyan, and N. Asadizanjani, "PCB Component Detection using Computer Vision for Hardware Assurance," arXiv.org, 2022. https://arxiv.org/abs/2202.08452 (accessed Dec. 22, 2024).