Real-Time Component Identification on Printed Circuit Boards

Aakash Hegde*
Rakshana Jayaprakash*
aakashh2@illinois.edu
rj17@illinois.edu
University of Illinois
Urbana-Champaign, Illinois, USA

ACM Reference Format:
Aakash Hegde and Rakshana Jayaprakash. 2024. Real-Time Component Identification on Printed Circuit Boards. In *Proceedings of Smart-X '22: Fake ACM Symposium on Smart cities, homes, phones, and beyond (Smart-X '22).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 ABSTRACT

This study addresses the issue of growing electronic waste by focusing on the real-time detection and classification of electronic components on printed circuit boards which will enable extraction and reuse of these components. We explore and implement deep-learning models to facilitate detection and identification of components with a focus on speed, low complexity, and accuracy. We primarily focus on using YOLO (You Only Look Once) [10] as the backbone for our models. Unlike traditional detection methods that process an image multiple times, YOLO's unified architecture enables it to detect objects in a single pass, making it highly suitable for this use case. By utilizing techniques such as image segmenting and Slicing Aided Hyper Inference (SAHI) [2], we are able to effectively detect smaller and more densely placed components.

2 INTRODUCTION

With the rapid electrification and digital development that the world is experiencing, there is also an accompanying rise in the amount of electronic waste (e-waste) generated. At the same time, the global e-waste collection and recycling rate is not keeping pace with this growth. A recent report by the Global E-waste Monitor stated that in 2022 the world generated 62 billion kg of e-waste, and that only 22.3 percent (13.8 billion kg) of the e-waste generated was documented as properly collected and recycled [3].

Waste printed circuit boards (WPCBs), which contain a large number of high-value components and toxic substances, are a critical component of all e-waste generated. These WPCBs are often recycled for the raw materials that can be extracted from them such as copper, fiber, etc. They also consist of electronic components (ECs) such as resistors, capacitors, integrated circuits (ICs), etc. which are generally extracted from the printed circuit board (PCB) before it is recycled. These ECs are usually in potentially reusable conditions and can be reused in making other PCBs and appliances. A recent survey has shown that these ECs are not effectively being reused and recycled in this way partly due to the limited attention that has been given to this field [13].

*Both authors contributed equally to this research.

Smart-X '22, Jan 18–May 05, 2022, Urbana, IL
2024. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

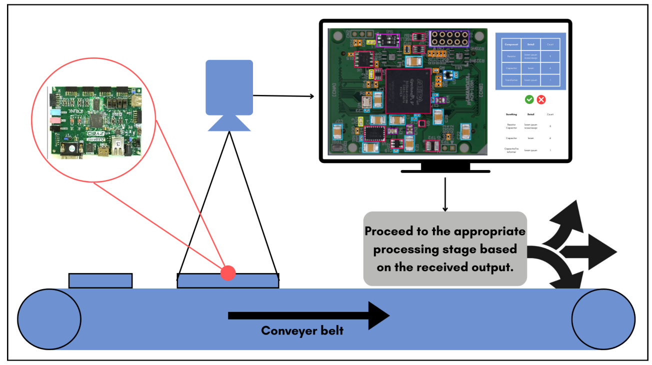

Figure 1: Proposed system model with the camera capturing the input and the PCBs on the conveyer belt.

In this paper, we identify one of the key problems that contribute to the low degree of reuse of ECs on WPCBs – effective detection and identification of ECs. We work on developing a system that can identify components in real-time which will then help in identifying which WPCBs are worth extracting ECs from and also help in segregating the ECs once they have been extracted. Figure 9 describes the system at a high level – a camera looks down on a conveyor belt that has WPCBs on it. The video/frames are processed by our algorithm and the ECs are identified. Based on these results, actions can be taken further down the conveyor belt regarding how the ECs are extracted and segregated. The latter topic is out of scope for our study.

We aim to design this for computationally weaker devices (eg: a Raspberry Pi) as we believe that it is important to ensure that the proposed system does not become a financial burden for industries trying to adopt it into their existing systems. Since models that are powerful enough for this task can be run on such systems nowadays, it would also make sense to maximize the utilization of computational power and get the best performance per dollar.

Two important factors that make this task challenging can be better explained given a brief understanding of how WPCB recycling plants function. The usual WPCB recycling plant with WPCBs laid out on conveyor belts is shown in Figure 2. For the scope of this project, we can focus on the initial sections of the workflow. The WPCBs are first laid out on a moving conveyor belt that feeds into machines that carry out the process of extracting ECs and recycling the bare PCB. The first challenge for our system is to identify components using images that are rendered slightly blurry due to noise from movement. The second challenge is to be able

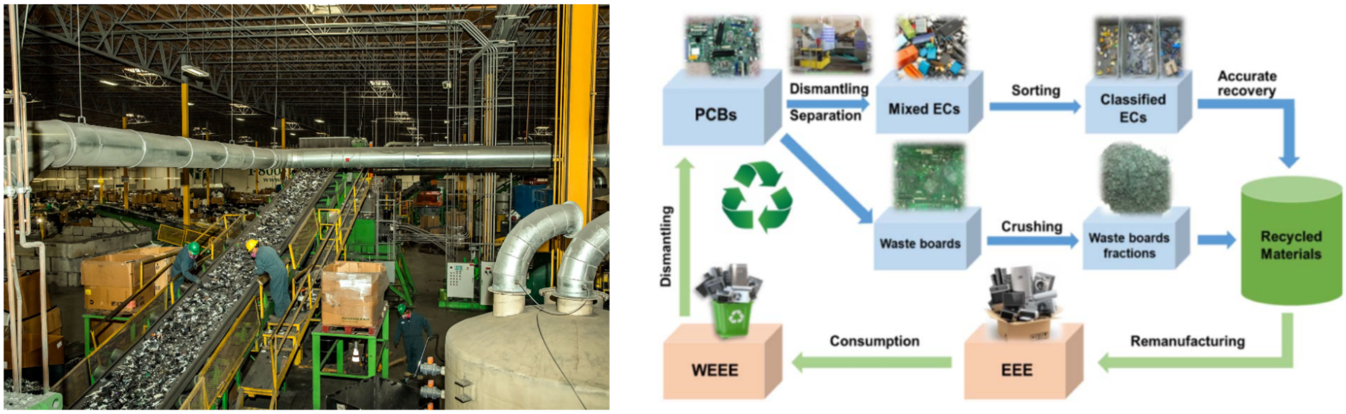

Figure 2: E-waste on a conveyor belt in a recycling plant and recycling flow chart of a PCB in WEEE (from [13]).

Table 1: Comparing existing features vs. proposed

Feature	Component detection	Embedded deployment	Uses regular camera images	Real-time testing
PCB component identification ([4] [6])	✓	×	✓	×
IC chip identification ([11])	-	×	✓	×
Defect detection ([8])	×	×	×	✓
PCB Dataset creation ([9], [7])	✓	×	-	×
Our System	✓	✓	✓	✓

to process in real-time so as to keep up with the large volume of WPCBs. This project aims to address these challenges as the current methods and research in this area are not focused on this matter (discussed in the next section).

Some other challenges that are unaddressed in this study include:

- The system's performance when the WPCBs are not clean. Considering that recycling plants often get old, dusty, grimy electronic devices, our image processing algorithm might not perform as well as it would.
- Identifying if the ECs are in working condition. There are dedicated devices that test for the correct functioning of ICs that may be used for this.
- Arrangement of PCBs on the conveyor belt. Our system requires that the PCBs are not haphazardly piled on one another on the conveyor belt.

3 RELATED WORK

Prior work on this topic has usually focused on the detection and classification of components on PCBs. In the following paragraphs, we discuss these works and indicate how they are inadequate for our particular problem statement.

The study in [11] develops a deep neural network, IC-ChipNet, for learning an embedding from images of IC chips to facilitate fine-grained retrieval, recognition, and verification tasks in the domain of microelectronics. By leveraging machine learning and computer vision techniques, it seeks to enhance the identification and validation of IC packages that is important for ensuring the integrity, security and legitimacy of integrated circuits. This work relates to our use case with respect to the identification of ICs using certain embeddings in the images. However, it only detects a single component (ICs) and does not have any metrics on the performance of the machine learning models used.

In [4] a novel deep learning approach is presented for detecting and classifying components on PCBs more efficiently. By leveraging graph embedding techniques and a three-stage object detection pipeline the research addresses challenges that would have otherwise made the task of identifying and labeling the various components in a densely laid out PCB very difficult. This study does a great job with respect to labeling and classification of various components on a board, but the multi-staged approach and the machine learning models used are not resource friendly nor computationally light enough for real-time performance on weaker systems. We intend to explore the method from this paper for our work as the basic implementation closely relates to ours.

There are several works that are focused on obtaining datasets for the purpose of component detection on PCBs [7] [6] [9]. In [9], the lack of publicly available high quality dataset of labeled PCB images for recycling related purposes is identified and is remedied by the contribution of one such resource. It also discusses an approach that can help identify PCBs as a whole which may prove to be useful when implemented in conjunction with our system if required. Datasets such as these help us train our machine learning model and improve upon the works identified in this section.

Remotely related to our work are those that focus on microscopic or highly zoomed-in images of ECs for fault/defect detection at a very fine grain. One such work in [8] identifies the challenges of the high density of components on modern PCBs and proposes a real-time capable deep learning model that is able to extract the required

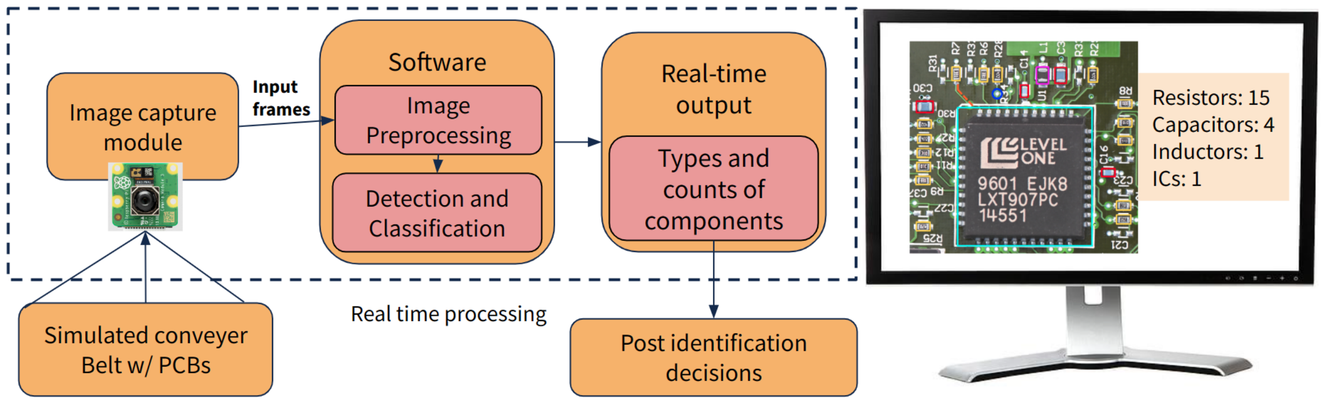

Figure 3: System overview highlighting the three modules.

features for the purpose of defect detection. The experimental setup proposed in this study may also serve as an inspiration for building a dedicated setup for our system as part of future work.

4 SYSTEM OVERVIEW

This section describes the proposed system for the real-time detection and classification of electronic components in PCBs on moving conveyor belts. First, the selected component detection model, trained on our dataset, is loaded into the system. Frames of images of the components in motion on the conveyor are captured and processed. These images are input to the detection model. The system outputs include bounding boxes showing the detected components, their corresponding types and the count of each. The conveyor belt's movement invariably leads to motion blur and varying degrees of noise in the captured images. This blur effect can vary in intensity depending on the speed of the conveyor belt and the camera's shutter speed, posing a challenge to the object detection model's ability to accurately identify and classify components. Achieving successful identification of components in real-time poses is the most significant challenge, emphasizing the need for robust algorithms and efficient processing mechanisms.

At its core, the system comprises three primary modules: the Image Capture Module, the Image Preprocessing Module, and the Object Detection and Classification Module. Figure 10 provides a visual representation of the system architecture, illustrating the flow of data and the functionality of each module.

4.1 Image Capture Module

- **Input:** Real-time video feed from a camera overlooking the conveyor belt.
- **Output:** Raw frames captured at high frequency.
- **Functionality:** This module is responsible for continuously capturing images of the boards as they move along the conveyor belt.

4.2 Image Pre-processing Module

- **Input:** Raw frames from the Image Capture Module.
- **Output:** Enhanced frames ready for object detection.
- **Functionality:** Given the challenges of detecting small ECs, this module applies pre-processing techniques on the input images and divides the images into multiple segments of the same size. Segmenting the image helps the subsequent detection module to identify these small features, thereby facilitating more accurate object detection.

4.3 Object Detection and Classification Module

- **Input:** Preprocessed images from the Image Preprocessing Module.
- **Output:** Detection results, including bounding boxes, component types, and counts.
- **Functionality:** This module employs a deep learning-based object detection model to identify and classify various electronic components in the preprocessed images. It generates bounding boxes around each detected component, identifies the component type, and counts the number of each type of component present. The model is expected to operate in real-time, ensuring swift processing to keep pace with the movement of the conveyor belt.

4.4 Evaluation Approach

To evaluate the performance of our system, we will employ several key evaluation metrics and criteria. We will calculate the average precision (AP) score for each component. This AP score will provide insights into the accuracy of our system's detection capabilities for individual components. We utilize the mean average precision (mAP) score to assess the overall performance of our system in detecting all four components collectively.

5 PROPOSED APPROACH

In this section, we describe the primary modules in the system and give a deeper insight into the techniques we used for processing images and enabling the detection of even the smallest components on the PCBs.

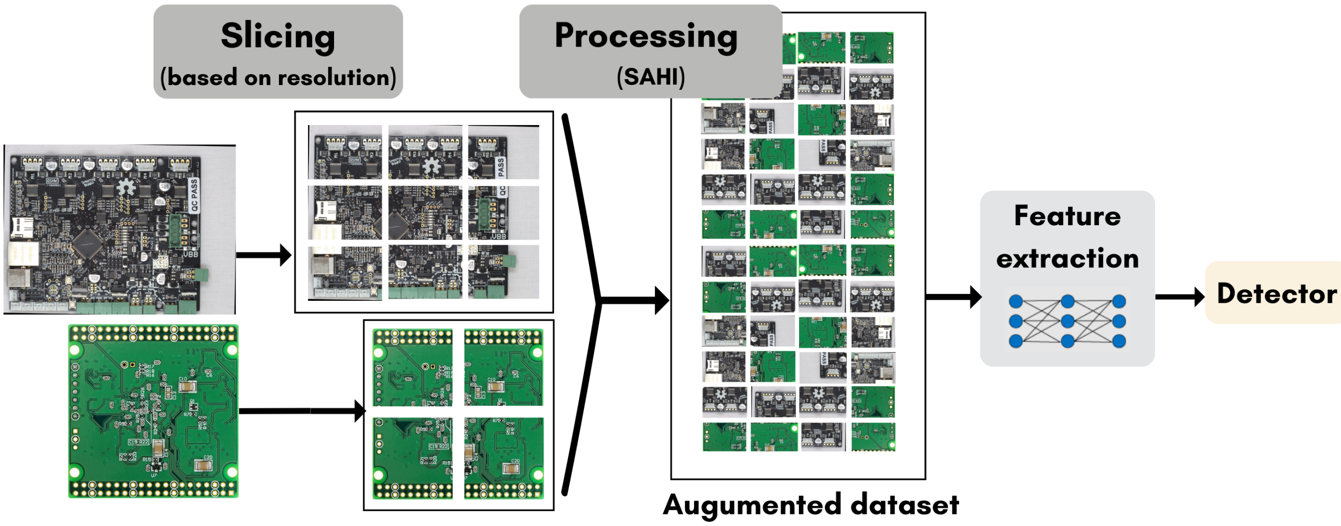

Figure 4: Generation of fine tuned dataset.

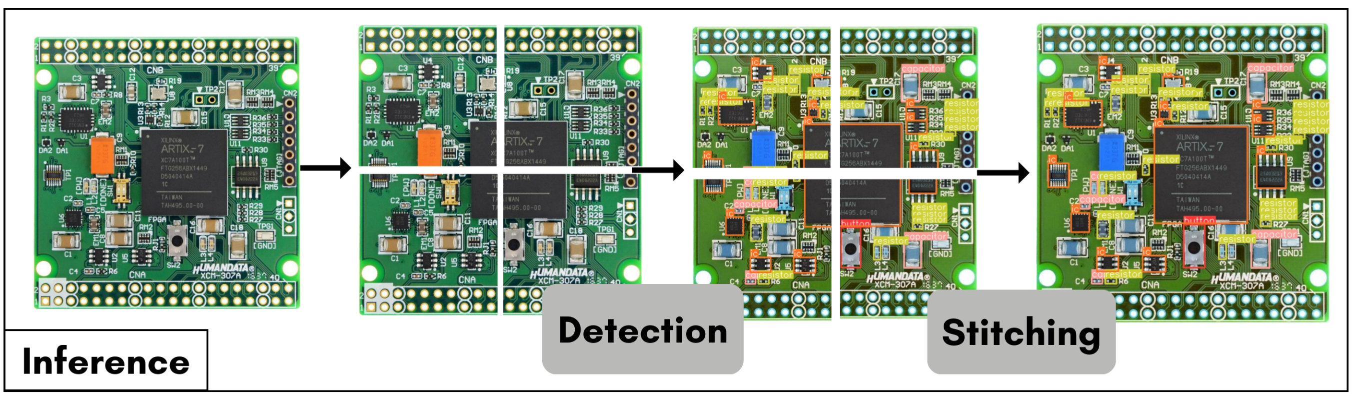

Figure 5: Inference of each frame form video stream.

5.1 Image Capture

One of the goals of this study was to implement and evaluate the functionality without using expensive or complex hardware. Therefore, the image capture module uses an ordinary camera that can provide a resolution in the range of 10s of megapixels, provides basic auto-focus or manually adjustable focus, and can capture images at a frame-rate of at least 30 frames per second. The captured frames will then be pre-processed and fed to the detection model. Since there is no guarantee for the rate at which the image frames might be consumed by the subsequent stages, we do not buffer the captured frames. This ensure that the frame that is operated on is always the latest image.

5.2 Image Pre-processing

5.2.1 Training: During the training process, we used the Slicing Aided Hyper Inference (SAHI) technique to prepare our dataset for optimal performance in small object detection. This method [2] focuses on enhancing the detectability of small-sized objects within high-resolution images by slicing these images into smaller, more manageable segments.

This step involves slicing each image into overlapping patches. This approach is designed to augment the apparent size of small electronic components, making them more detectable by the deep learning model. We chose patch sizes such as 640x640pixels, which align with our model's input requirements. To ensure that no object is partially missed due to slicing near the edges, we ensured an overlap between adjacent patches, at 20%.

The SAHI technique ensures that once the images are sliced, each object's annotation is recalculated to fit within its respective patch and if an object spans across multiple patches, it is annotated in each one, ensuring complete coverage. Patches that do not contain any objects are discarded, focusing the model's training on valuable data. This speeds up the training process and also prevents the model from learning irrelevant features.

To further improve the model's ability to generalize across various scenarios, we applied additional image augmentations such as rotation, scaling, and color adjustment to these patches. We think

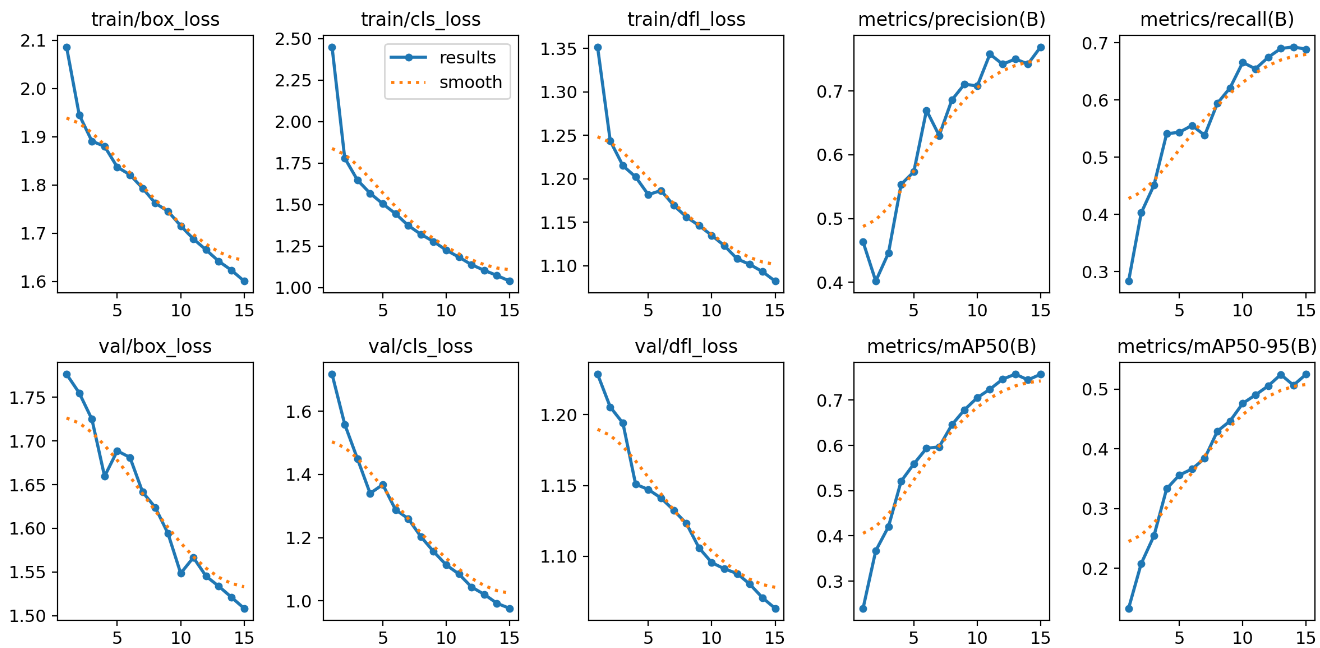

Figure 6: Train loss, val loss, metrics(precision, recall, mAP) per epoch for YOLOv8+SAHI model

such augmentations can simulate different operational conditions and result in a more robust model.

5.2.2 Inference: During inference, the image frame that is obtained from the image capture module is segmented into equal sized image frames (sub-frames) based on the resolution. These sub-frames are then fed into the detection model described in the next section. This step is essential to enable the detection of ECs that are very small (less than 2mm in one dimension) and often found in dense clusters on the PCB.

5.3 Object Detection and Classification

The object detection and classification model is the most important module in our system. This model is tasked with detecting the ECs on the PCB, identifying the type of EC detected, and keeping a count of each component for every unique PCB that is seen.

In our study, we experimented with multiple models and evaluated their performance for this task. Some of the factors that we considered while selecting the model were the inference speed and accuracy.

6 IMPLEMENTATION

In this section we describe the software and hardware components we used to implement and test out approach. We also describe how the models were trained and the datasets that were used.

6.1 Hardware

We implemented our approach on two different systems -

- A Raspberry Pi 4 Model B with 1.5GHz, 64-bit, quad-core ARMv8 CPU; an 8MP, 30FPS, auto-focus enabled camera compatible with the Raspberry Pi; a monitor.
- A laptop with 11th Gen Intel Core i7-1165G7 2.70GHz with 16GB RAM, a 20MP, 30FPS, auto-focus enabled webcam;

Each of these systems have the following common components required for our implementation - a camera, a computer, and a monitor/screen to display and analyze the results.

Testing our approach on these two systems allowed us to compare performance between a more powerful (yet typical) system like a laptop and a Raspberry Pi. The two aspects that must be considered when choosing the system are cost and performance. A more powerful system, which is inadvertently more expensive, can run more complex models while ensuring the same frame rate, and provide better component identification accuracy.

6.2 Software

We implemented our approach in Python3.8. We utilized vendor provided and community maintained libraries to initialize and configure the camera. The image obtained from the camera would then be segmented as per the process mentioned in the previous section and fed into the component detection model sequentially.

The YOLO based deep-learning modules were trained on our datasets using Python APIs on powerful machines to speed up the training process. After training the models, the resulting weights were transferred to the system to be used, and imported into the corresponding model for use during inference.

Once detection, identification and component counting is complete, we also had non-essential functions to stitch the segmented images back together to for visualization purposes. Here, we would

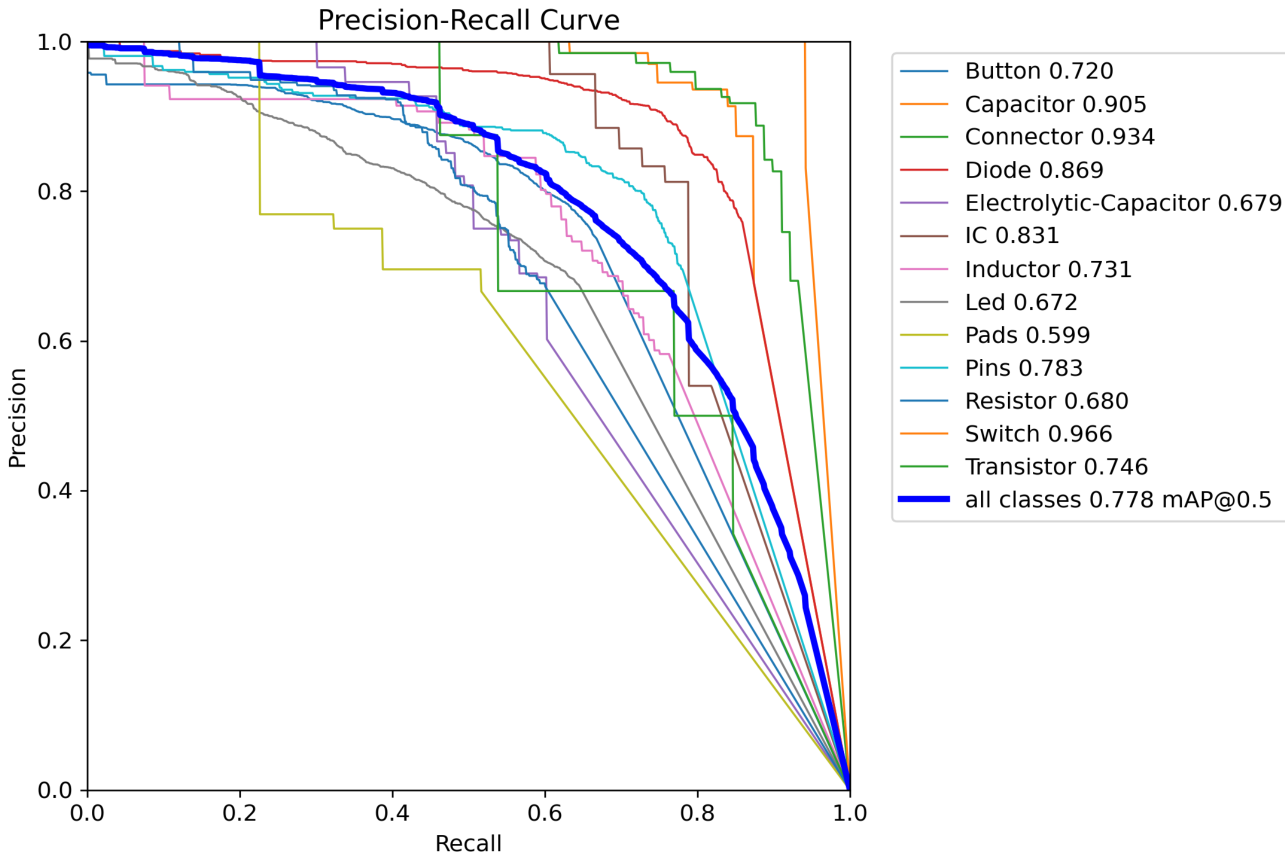

Figure 7: Precision-Recall Curve for test set

see bounding boxes drawn around all the detected ECs and also get the type and total count of ECs on the entire board.

6.3 Datasets and Training

Popular object detection frameworks like Detectron2, MMDetection, and YOLOv5 are equipped with models trained on datasets like ImageNet and MS COCO. Using these pretrained models enables efficient fine-tuning on smaller datasets, drastically reducing training durations compared to starting from scratch with extensive datasets. The datasets typically utilized for pretraining consist of low-resolution images (640x480) showcasing relatively large objects that occupy a substantial portion of the image (about 60% of the image height on average). Trained on such datasets, these models exhibit excellent detection accuracy for similar input types. But their performance declines when tasked with detecting smaller components in high-resolution images, as often encountered in advanced drone and surveillance camera footage, and in our case, densely packed components on a PCB.

Due to their exceptional capabilities in real-time object detection, we chose to use the family of YOLO models over the others for our task. YOLO's architecture enables rapid image processing, which is essential for maintaining high accuracy when detecting densely packed and diminutive components typical in PCBs. The models' ease of integration, ability to fine-tune with specific datasets, and reliable performance on low-computational platforms were key factors in its suitability for our project. We used a combination of high res and low res images sourced from the following datasets:

Dataset 1: *PCB Component Dataset* [5]
This dataset consists of 47 high-resolution images capturing various electronic components commonly found on PCBs, with annotations for 31 different component types totaling approximately 62,000 instances. The dataset exhibits a skewed distribution, with resistors, capacitors, and connectors dominating, alongside high intra-class variance and low inter-class variance, posing challenges and reflecting real-world complexities.

Dataset 2: *RF100 PCB Object Detection Benchmark* [1]
We have also utilized the RF100 benchmark, a publicly available dataset aimed at establishing object detection benchmarks for model generalizability. This dataset, sourced from the Roboflow platform, provides standardized images and annotations for PCB components. The dataset is split into a train set (87% - 672 images), a validation set (8% - 64 images), and a test set (5% - 36 images). We enhanced our dataset through fine-tuning using standard data augmentation

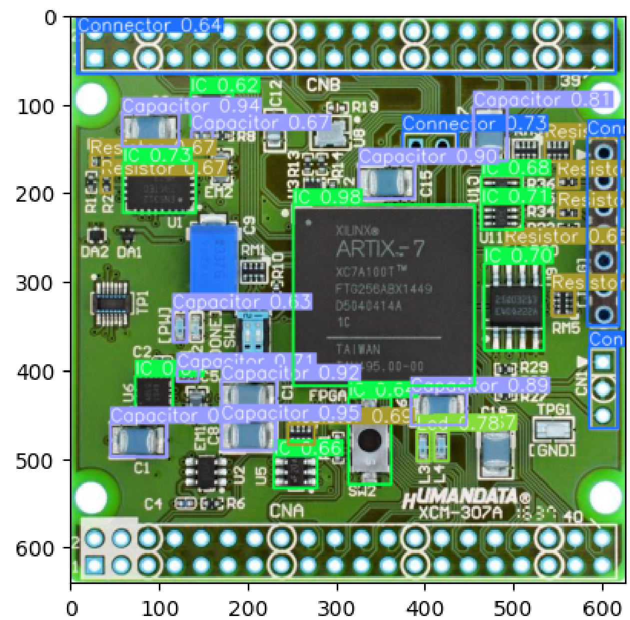

Figure 8: Inference of test image by fine tuned YOLOv5 model on initial dataset

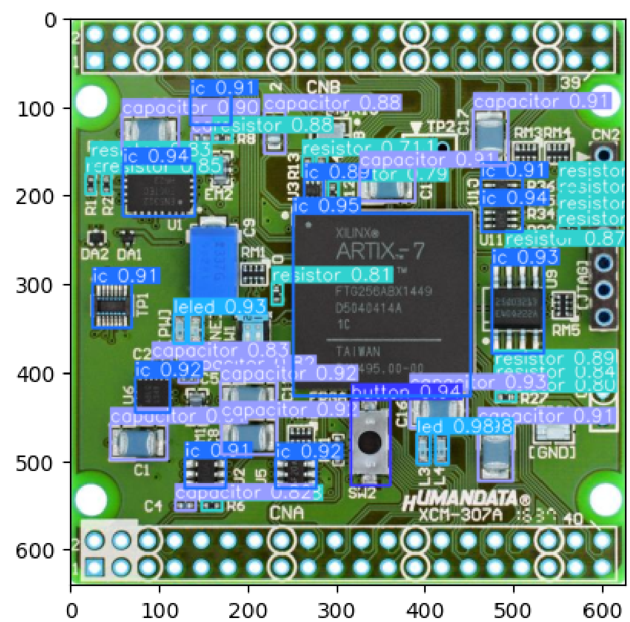

Figure 9: Inference of test image by fine tuned YOLOv8+SAHI model on augmented dataset

techniques outlined by Shorten et al. (2019) [12]. This involved preprocessing steps: auto-orientation and resizing to a standardized dimension of 640x640 pixels, tiling based on image resolution. To enhance dataset variability, these augmentations also encompassed horizontal and vertical flips, as well as 90° rotations in both clockwise and counter-clockwise directions, with the option for rotations within a range of -15° to +15°.

Table 2: Comparing existing features vs. proposed

Setup	mAP50	mAP50-95
YOLOv5	53.0	30.5
YOLOv8	52.9	33.8
YOLOv5 + SAHI	53.6	34.2
YOLOv8 + SAHI	56.2	38.4

7 EVALUATION

The results outlined in Table 2 illustrate the potential of our proposed approach. In our evaluation, we assessed performance using the mean Average Precision (mAP) at Intersection over Union (IoU) thresholds of 50% and 95%, consistent with established benchmarks in object detection accuracy. Although the improvements are modest, they are notable. Specifically, the integration of the SAHI with the YOLO models shows promising results. The YOLOv8+SAHI model reached a mAP50 of 56.2 and a mAP50-95 of 38.4.

These outcomes indicate that the SAHI adaptation, tailored for small objects like the components on a PCB, enhances detection capabilities. Future investigations will need to consider the balance between improved detection accuracy and operational speed, particularly given the efficient real-time processing capabilities of YOLO. This approach allows us to continue refining our model to optimize both performance and speed, particularly in practical, real-world applications.

8 CONCLUSION

In this study, our primary focus has been on developing a system aimed at addressing the growing concern of electronic waste by facilitating the detection and categorization of potentially reusable electronic components present on waste printed circuit boards. One of the reasons such recycling is not widespread is because of the difficulty in being able to tell if a board is worth recycling for its components. We have identified some of the challenges that make it a difficult task to identify components such as the density of placement and small size of the components. We have also ensured that the solution is not financially burdening which would encourage faster adoption into the current recycling ecosystem.

Our system is primarily composed of an ordinary camera for image capturing and a simple computational device such as a laptop or Raspberry Pi for processing. During operation, the images are pre-processed by being segmented into equal sized sub-frames which are then fed to the detection and identification deep-learning model. The model is then able to extract the minute features in this sub-frame, detect the presence of components, and identify them successfully. A count of components is maintained by the algorithm for every PCB. This information can be used to determine whether it is worth to extract the components on that board.

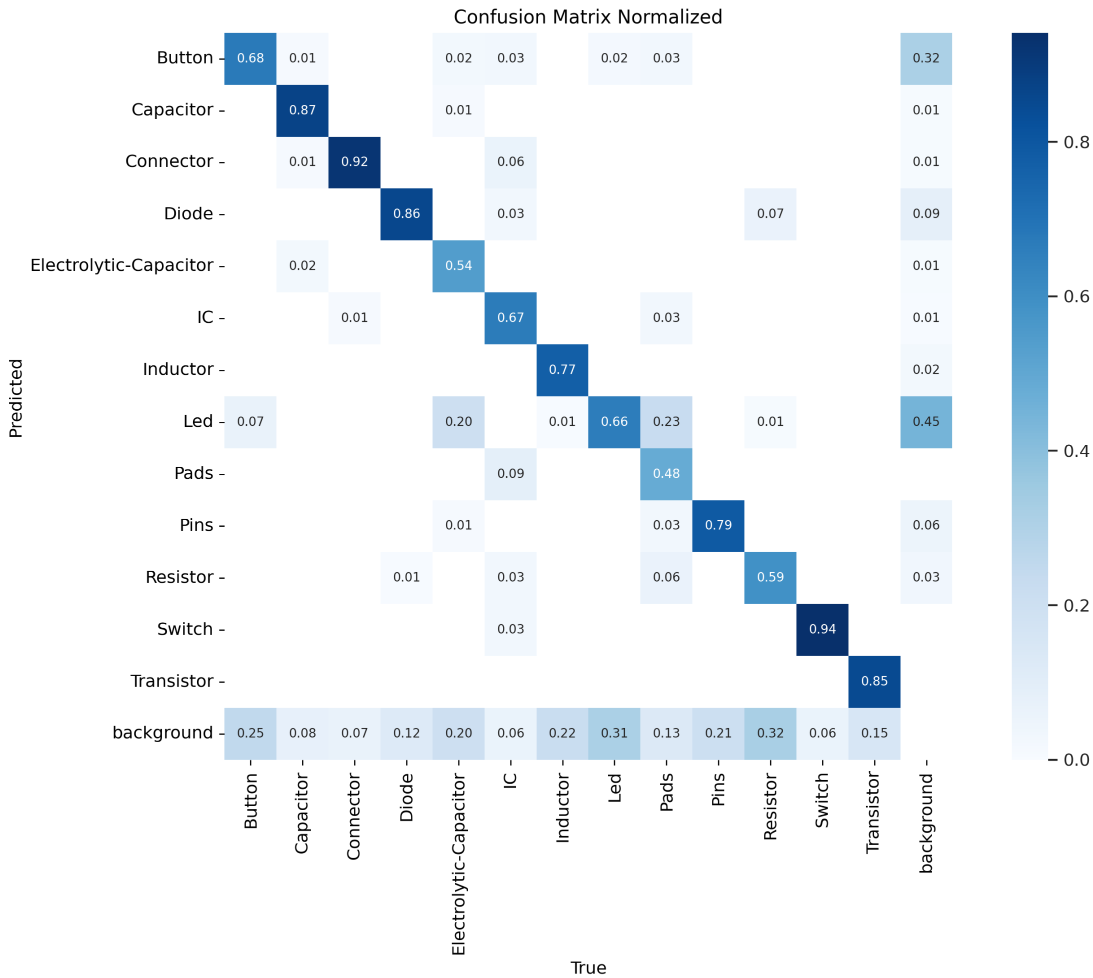

Figure 10: Confusion Matrix for test set

REFERENCES

[1] Roboflow 100. 2023. printed circuit board Dataset. https://universe.roboflow.com/roboflow-100/printed-circuit-board. https://universe.roboflow.com/roboflow-100/printed-circuit-board visited on 2024-05-10.

[2] Fatih Cagatay Akyon, Sinan Onur Altinuc, and Alptekin Temizel. 2022. Slicing Aided Hyper Inference and Fine-tuning for Small Object Detection. *2022 IEEE International Conference on Image Processing (ICIP)* (2022), 966–970. https://doi.org/10.1109/ICIP46576.2022.9897990

[3] Tales Yamamoto Rosie McDonald Elena D'Angelo Shahana Althaf Garam Bel Otmar Deubzer Elena Fernandez-Cubillo Vanessa Forti Vanessa Gray Sunil Herat Shunichi Honda Giulia Iattoni Deepali S. Khetriwal Vittoria Luda di Cortemiglia Yuliya Lobuntsova Innocent Nnorom Noémie Pralat Michelle Wagner Cornelis P. Baldé, Ruediger Kuehr. 2024. Global E-waste Monitor 2024. In *International Telecom-munication Union (ITU) and United Nations Institute for Training and Research (UNITAR)*.

[4] Chia-Wen Kuo, Jacob Ashmore, David Huggins, and Zsolt Kira. 2018. Data-Efficient Graph Embedding Learning for PCB Component Detection. arXiv:1811.06994 [cs.CV]

[5] Chia-Wen Kuo, Jacob Ashmore, David Huggins, and Zsolt Kira. 2019. Data-Efficient Graph Embedding Learning for PCB Component Detection. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE.

[6] Hangwei Lu, Dhwani Mehta, Olivia P. Paradis, Navid Asadizanjani, Mark Mohammad Tehranipoor, and D. Woodard. 2020. FICS-PCB: A Multi-Modal Image Dataset for Automated Printed Circuit Board Visual Inspection. *IACR Cryptol. ePrint Arch.* 2020 (2020), 366. https://api.semanticscholar.org/CorpusID:215796830

[7] Gayathri Mahalingam, Kevin Marshall Gay, and Karl Ricanek. 2019. PCB-METAL: A PCB Image Dataset for Advanced Computer Vision Machine Learning Component Analysis. In *2019 16th International Conference on Machine Vision Applications (MVA)*. 1–5. https://doi.org/10.23919/MVA.2019.8757928

[8] Van-Truong Nguyen and Huy-Anh Bui. 2022. A real-time defect detection in printed circuit boards applying deep learning. *EUREKA: Physics and Engineering* 2 (Mar. 2022), 143–153. https://doi.org/10.21303/2461-4262.2022.002127

[9] Christopher Pramerdorfer and Martin Kampel. 2015. A dataset for computer-vision-based PCB analysis. In *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*. 378–381. https://doi.org/10.1109/MVA.2015.7153209

[10] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. arXiv:1506.02640 [cs.CV]

[11] Md Alimoor Reza and David J. Crandall. 2020. IC-ChipNet: Deep Embedding Learning for Fine-grained Retrieval, Recognition, and Verification of Microelectronic Images. In *2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*.

[12] Connor Shorten and Taghi M. Khoshgoftaar. 2019. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data* 6 (2019), 1–48. https://api.semanticscholar.org/CorpusID:195811894

[13] Wenting Zhao, Junqing Xu, Wenlei Fei, Ziang Liu, Wenzhi He, and Guangming Li. 2023. Correction: The reuse of electronic components from waste printed circuit boards: a critical review. *Environ. Sci.: Adv.* 2 (2023), 827–827. Issue 5. https://doi.org/10.1039/D3VA90014B