Toronto Metropolitan University

Department of Aerospace Engineering
Faculty of Engineering & Architectural Science

Semester (Term, Year)	F23
Course Code	AER850
Course Section	01
Course Title	Intro to Machine Learning
Course Instructor	Prof. Reza Faieghi
Submission	Project
Submission No.	01
Submission Due Date	17/12/23
Title	Project 3
Submission Date	14/12/23

Submission by (Name):	**Student ID (XXXX1234)**	**Signature**
Emily Birkenhead	6788	Emily Birkenhead

By signing the above you attest that you have contributed to this submission and confirm that all work you contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, and "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Academic Integrity Policy 60, which can be found at www.torontomu.ca/senate/policies/

Table of Contents

Table of Figures

GitHub

For the purposes of this report, Step 1 was run using Spyder, while Steps 2-3 were run using Google Colab. Therefore, some formatting may not be consistent depending which platform was used to run the code. Both files have been uploaded to GitHub, with all section included.

GitHub link: https://github.com/ebirkenhead/AER850_Project3

Introduction

This project aims to use machine learning modules to enhance the manufacturing efficiency of printed circuit boards. Using machine learning modules reduces the need for manual inspection, thus improving the reliability of the circuits. This project report uses machine learning to replicate this process, through the use of computer vision systems, using the object detection algorithm, YOLO.

An image of a motherboard will be used, and OpenCV put in place to extract the required components from the image using object masking processes. Next, YOLOv8, will be used to classify the components on a circuit board, train the module to correctly identify these components, and to assign the correct component names to test and validation images. Three validation images will be used to test the training modules abilities.

Step 1 - Object Masking

Object masking is used to allow the machine to identify and isolate specific objects within an image. To correctly mask an object in a machine learning program, the following steps must be followed:

- Thresholding
- Edge Detection
    - Corner Detection or
    - Contour Detection
- Line Filtering
- Mask Extraction

For this report, the following image underwent the Object Masking process:

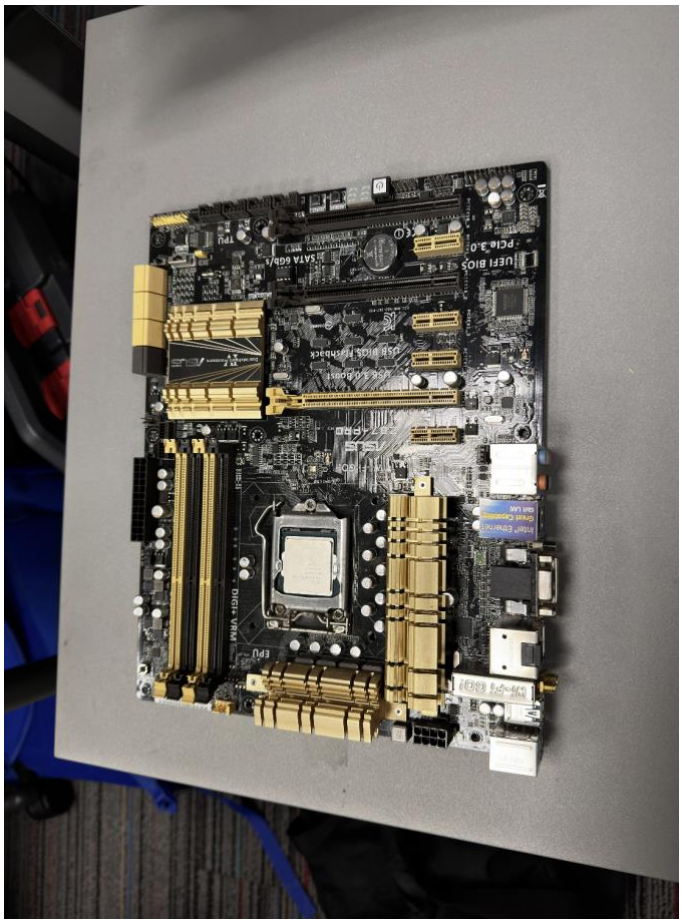

Figure 1 - Original Image

Thresholding

Thresholding is the process of creating a binary image from a grey scale image. The thresholding function compares the intensity of each pixel, with a predefined threshold value. If the pixel is higher than the threshold, it is set to 255 (white), if it is lower it is set to 0 (black). Thresholding is done to create a clear distinction between the foreground and background of an image, allowing for further edge detection to take place.

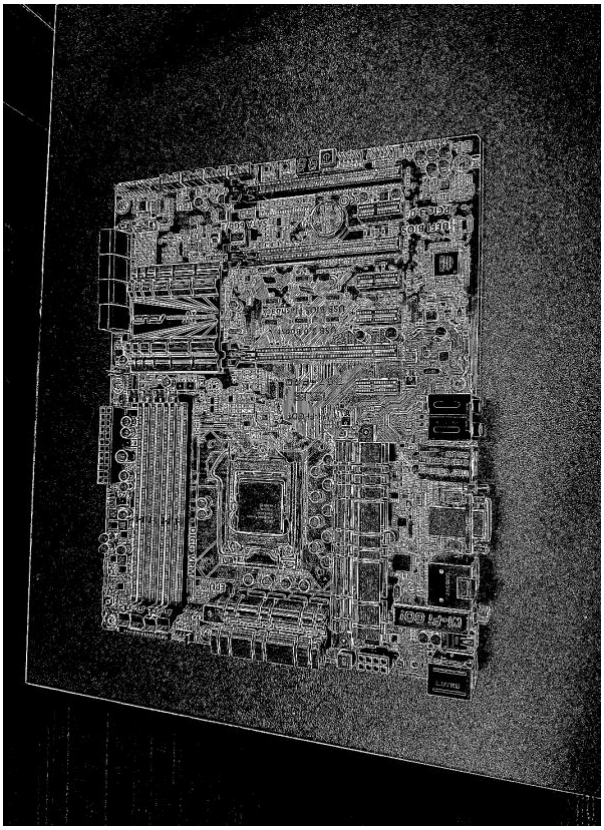

Figure 2 - Thresholding Image

Edge Detection

Edge detection is used to identify points in a digital image where the image brightness is not consistent. The points are usually the edges of an object. Edge detection can be done using two methods, either corner detection or contour detection.

Corner detection refers to the technique of using the point where two edges meet (or where one edge ends) to extract the significant features and information from the image. It identifies points that are significant for an image analysis task, regardless of the size of the components. If many small corners are detected but are not relevant for a particular application, additional processing steps would be required to filter them out.

Contour detection identifies the points that form a boundary between different regions based on color, intensity, or texture discontinuities. Contour detection is most useful when the shape of the object is of interest, such as for identification or calculations, but when the other details of the object are not as important.

For the purposes of this report, both methods were attempted, however the contour detection revealed the better results, which was as expected.

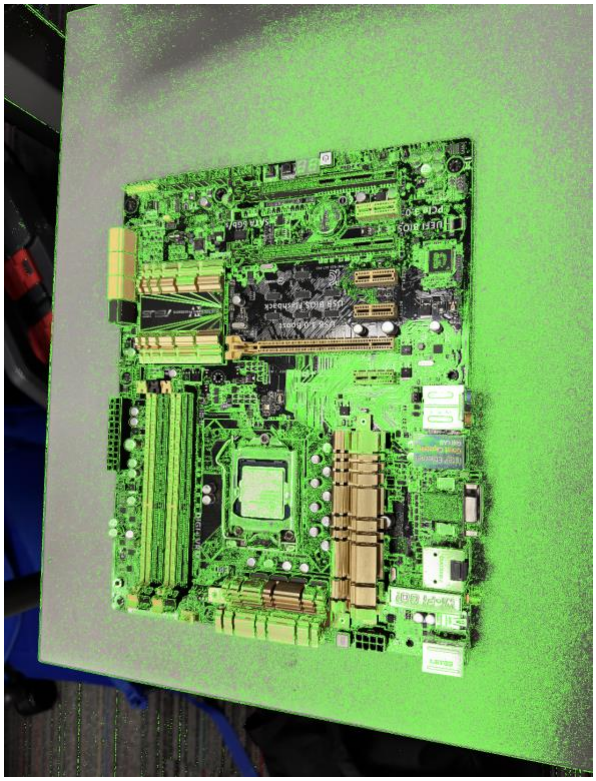

Figure 3- Edge Detection Image

Line Filtering

Line filtering is the process of identifying and removing irrelevant lines from an image. It looks at characteristics such as length, orientation, and intensity, and decides whether they are needed. Line filtering is a useful tool for removing noise from an image being processed.

Mask Extraction

Mask extracting involves isolating one part of an image to separate an object from its background. A mask is created, and applied to the original image. For this report, this was doen using the bitwise operation “cv2.bitwise_and()”. This operation cuts the intended object from its background.

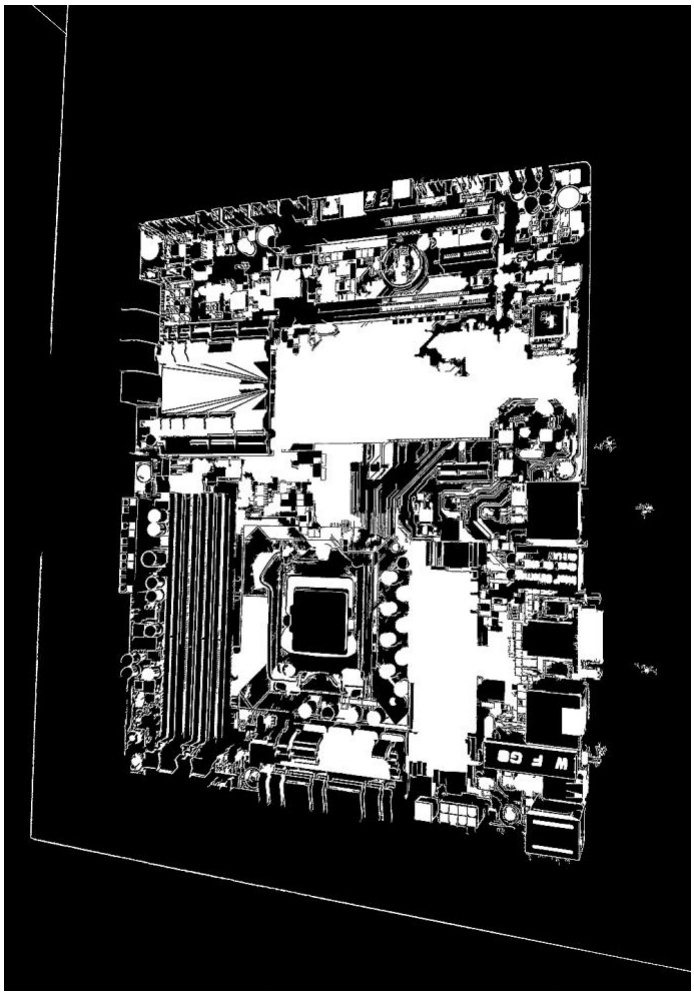

Figure 4 - Mask Image

Final Result

Figure 5 - Final Extracted Image

The final extracted image shows the PCB on a black background. The object masking process was effective in isolating the image and removing the unnecessary elements from its background. The components on the PCB are still very visible, which means any further analysis taking place on the image can still occur.

There are some issues with the lighting and reflections visible across the PCB. However, these do not significantly subtract from the visibility of the components, therefore it is not too problematic.

Therefore, overall, the object masking process was successful in isolating the PCB from its background.

Step 2 – YOLOv8 Training

The collected training data may be found in the Appendix.

Normalized Confusion Matrix

Confusion Matrices are used to describe the performance of a model on a given set of data, when the true values are known. The rows represent the predictions the model has made, and the columns represent the instances of a predicted class. The matrix makes it very easy to see if one component has been confused for another.

The training model developed for this report resulted in the following confusion matrix:

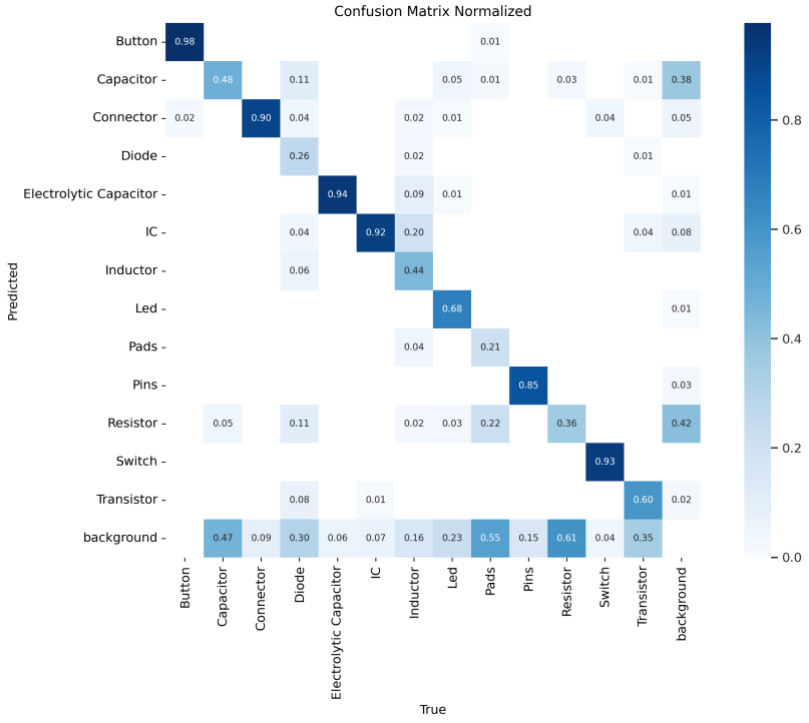

Figure 6 - Normalized Confusion Matrix

The diagonal line through the middle shows the number of times the model accurately predicted a component. It can be seen that “Button” and “Electrolytic Capacitor” had the highest likelihood of being accurately predicted, at 98% and 94% respectively, while “Resistor” had the lowest rate, at 36%. The elements that are not shown along the diagonal, are the elements that were often confused for another. For example, the model predicted "Capacitor" 11% of the time when it was actually "Resistor".

This model also shows how frequently each component was mistaken for the “Background” of the image, in the right most column of the graph. The matrix shows these values to be very low,

meaning that it was very rare for this to happen, with “Resistor” being most often mistaken, 42% of the time. The bottom row of the matrix shows how often the background of the image was classified as a component. This ranges from Never (“Button”) to 61% of the time (“Resistor”).

From this matrix, due to the diagonal dominance, it can be concluded that the model was very good at correctly classifying the components. However, the darker cells off the centre, do indicate that there was some confusion between the components. “Button” had the least difficulty with being identified as something else, while “Inductor”, “Resistor”, and “Capacitor” offered the most confusion.

Precision-Confidence Curve

A precision-confidence curve is used to visualise the difference between the precision of a model and its confidence that the decisions made are correct. The y-axis of a Precision-Confidence graph shows the percentage of true positives among the instances that the model predicted as positive. The x-axis represents the confidence the model has in these predictions. The higher the confidence level, the higher the model's certainty that its predictions are correct. The lines on the graph represent different components, demonstrating how the model's confidence threshold adjusts. If the confidence threshold is high, the model will make fewer, but more accurate, predictions. When the confidence threshold is low, the model makes more predictions, usually including false positives which in turn result in a lower precision.

The training model developed for this report resulted in the following precision-confidence curve:

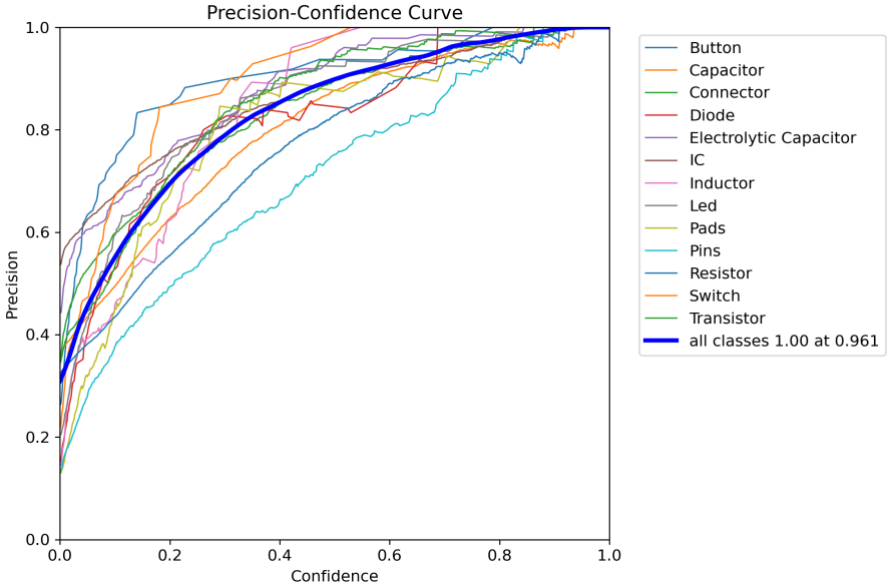

Figure 7 - Precision-Confidence Curve

A line that stays high as confidence decreases indicates that the model maintains high precision even though it is less certain about its predictions. When there is a steep drop off with the decreasing confidence, the model will only have high precision when it has high confidence about its predictions. It can be seen from this graph therefore, that the model maintains a high precision, despite low confidence, until it reaches a confidence of approximately 25%. After that, its precision does begin to drop.

The legend to the right of the graph shows "all classes 1.00 at 0.961". This means that when all classes are considered together, the model achieves a precision of 1.00 at a confidence threshold of 0.961. When the model is 96.1% confident or more about its predictions, it is extremely precise across all classes.

Precision-Recall Curve

A Precision-Recall (PR) Curve is used as a graphical representation of the performance of a model at a variety of threshold settings. PR Curves are used most often when there is a signific difference between the number of instances of each component in a system.

The y-axis of the graph represents it precision (the ratio of true positives predicted to the total number of **positive** predictions made). The x-axis is the recall (the ratio of true positive predictions to the total number of **actual** predictions made).

The training model developed for this report resulted in the following PR Curve:

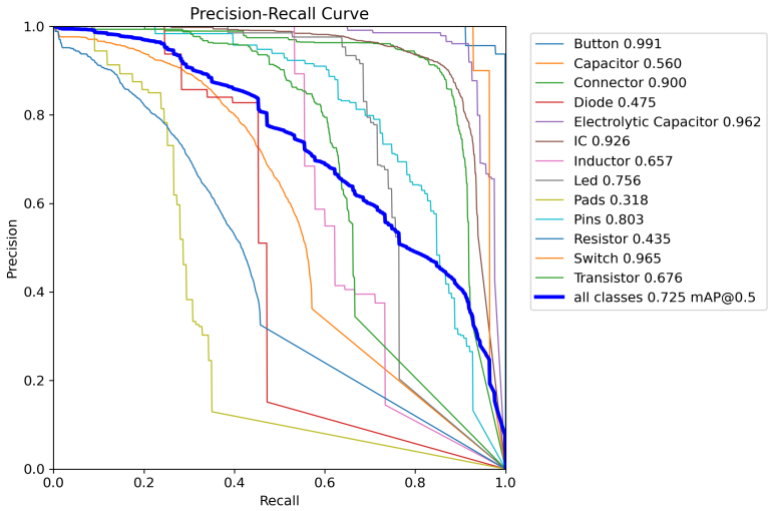

Figure 8 - Precision-Recall Curve

The lines on the graph represent the PR curve for each component. A curve that remains close to the top right corner implies a very high performance, meaning very few false positives or negatives. As the curves approach the bottom left corner, however, this implies a higher number of either false positives (low precision) or false negatives (low recall).

The legend to the right of the graph shows the Average Precision score for each class. The higher the AP value, the better the model's performance is for that class. Classes like "Button" and "Electrolytic Capacitor" have very high AP scores (near 1), suggesting the model performs very well in detecting them. Others, like "Pads" and "Resistor", have lower AP scores, indicating that the model is less effective in these cases.

The legend shows that "all classes 0.725 mAP@0.5" which indicates the mean Average Precision across all classes at a threshold of 0.5. It is an average of the AP scores for each class and represents a single performance metric for the model's ability to correctly classify all classes. A value of 72.5% shows a decent ability to correctly classify the components at a threshold of 0.5.

Step 3 – YOLOv8 Evaluation

Result

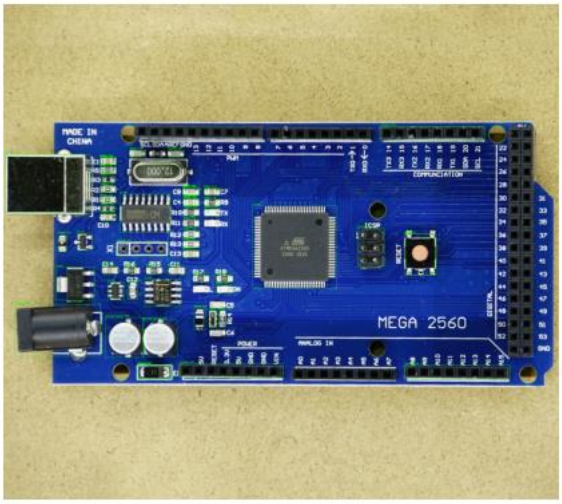

Figure 9 - Ardmega Resulting Image

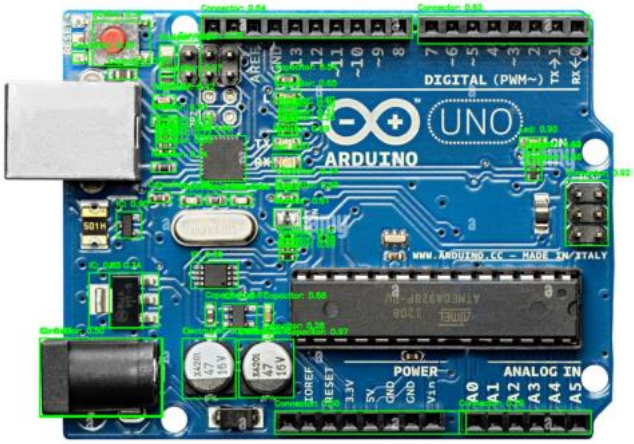

Figure 10 - Arduno Resulting Image

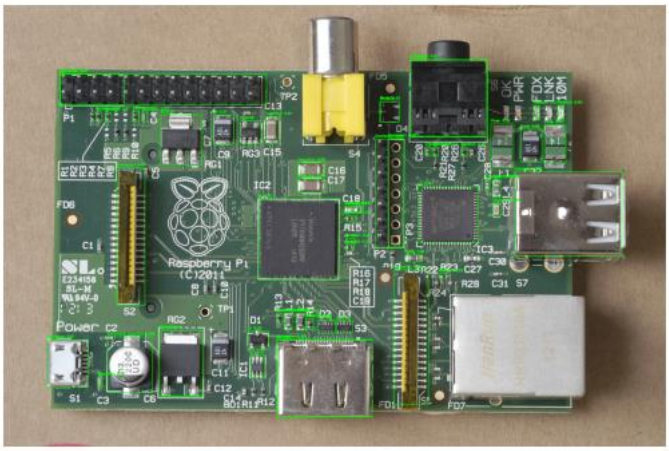

Figure 11 - Rasppi Resulting Image

The results of the YOLOv8 Training model visibly show the name and location of each component to a relatively high degree of accuracy. The boxes shown around each component are very tight fitting to the component, demonstrating high precision. The training model was run at a epoch of 150. To improve the accuracy of the component identification, this could be increased to 200, but this would not have any extreme improvement on the current results.

Therefore, the YOLOv8 training model was successful in its component identification task of the three given test images.

Conclusion

This project was successful in its use of machine learning concepts to identify components on a circuit board. Object masking process were used to extract an image from its background. This demonstrated the process used to keep the essential parts of an image for further processing, while removing the unnecessary components. Next, YOLOv8 was used to train a model to correctly identify the names and locations of the components found on a circuit board. The training program was then evaluated on three images of circuit boards.

Both tasks were completed to high levels of accuracy. To improve, the number of epochs used in the YOLOv8 training could be increased, as could the size of the images analysed. However, overall, this report was successful in its use of machine learning ideas and processes to run a training module on circuit board components.

References

Model Prediction with Ultralytics YOLO. (n.d.). Retrieved from Ultralytics: https://docs.ultralytics.com/modes/predict/

Appendix

Training Data

```
Validating runs/detect/yolov8n_pcb_components3/weights/best.pt...
Ultralytics YOLOv8.0.227 🚀 Python-3.10.12 torch-2.1.0+cu118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 3008183 parameters, 0 gradients, 8.1 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 14/14 [00:47<00:00,  3.39s/it]
                   all        105      19108      0.817      0.668      0.725      0.525
                Button        105         45      0.904          1      0.991      0.832
             Capacitor        105       7251       0.76      0.432       0.56      0.294
             Connector        105        659        0.8      0.892        0.9      0.687
                 Diode        105         53      0.824      0.443      0.475      0.347
Electrolytic Capacitor        105        160      0.852      0.938      0.962      0.674
                    IC        105       1322      0.833      0.904      0.926      0.701
              Inductor        105         45      0.869      0.556      0.657      0.519
                   Led        105        127      0.862      0.685      0.756      0.548
                  Pads        105        143       0.84       0.22      0.318      0.217
                  Pins        105        151      0.618      0.828      0.803      0.574
              Resistor        105       8600      0.696      0.304      0.435      0.209
                Switch        105         28      0.908      0.929      0.965      0.825
             Transistor        105        524      0.854      0.557      0.676      0.393
```

Ardmega Data

image 1/1 /content/drive/MyDrive/AER850 Project 3/data/evaluation/ardmega.jpg: 832x928 1 Button, 9 Capacitors, 6 Connectors, 2 Electrolytic Capacitors, 8 ICs, 27 Resistors, 14.8ms
Speed: 11.3ms preprocess, 14.8ms inference, 2.7ms postprocess per image at shape (1, 3, 832, 928)
Detected: IC - confidence: 0.92
Detected: IC - confidence: 0.92
Detected: IC - confidence: 0.90
Detected: Connector - confidence: 0.90
Detected: IC - confidence: 0.88
Detected: Resistor - confidence: 0.86
Detected: IC - confidence: 0.86
Detected: Resistor - confidence: 0.86
Detected: IC - confidence: 0.83
Detected: Connector - confidence: 0.83
Detected: Resistor - confidence: 0.81
Detected: Capacitor - confidence: 0.80
Detected: Capacitor - confidence: 0.78
Detected: IC - confidence: 0.78
Detected: Resistor - confidence: 0.77
Detected: Resistor - confidence: 0.76
Detected: Capacitor - confidence: 0.73
Detected: Resistor - confidence: 0.73
Detected: Resistor - confidence: 0.73
Detected: Resistor - confidence: 0.71
Detected: Resistor - confidence: 0.71
Detected: Resistor - confidence: 0.70
Detected: Resistor - confidence: 0.70
Detected: Capacitor - confidence: 0.70
Detected: Electrolytic Capacitor - confidence: 0.70
Detected: Resistor - confidence: 0.69
Detected: Resistor - confidence: 0.68
Detected: Capacitor - confidence: 0.66
Detected: Electrolytic Capacitor - confidence: 0.65
Detected: Resistor - confidence: 0.63
Detected: Resistor - confidence: 0.62
Detected: Capacitor - confidence: 0.62
Detected: Connector - confidence: 0.62
Detected: Resistor - confidence: 0.61
Detected: Button - confidence: 0.61
Detected: Resistor - confidence: 0.60
Detected: Connector - confidence: 0.57
Detected: Capacitor - confidence: 0.55
Detected: Resistor - confidence: 0.55
Detected: Connector - confidence: 0.53
Detected: Resistor - confidence: 0.44
Detected: Resistor - confidence: 0.36
Detected: Capacitor - confidence: 0.35
Detected: Capacitor - confidence: 0.34
Detected: Resistor - confidence: 0.33
Detected: Resistor - confidence: 0.32
Detected: Resistor - confidence: 0.31
Detected: Connector - confidence: 0.30
Detected: Resistor - confidence: 0.29
Detected: IC - confidence: 0.29
Detected: Resistor - confidence: 0.27
Detected: Resistor - confidence: 0.26
Detected: Resistor - confidence: 0.25

Arduno Data

image 1/1 /content/drive/MyDrive/AER850 Project 3/data/evaluation/arduno.jpg: 640x928 1 Button, 15 Capacitors, 7 Connectors, 2 Electrolytic Capacitors, 7 ICs, 3 Leds, 20 Resistors, 13.7ms
Speed: 5.9ms preprocess, 13.7ms inference, 2.6ms postprocess per image at shape (1, 3, 640, 928)
Detected: Electrolytic Capacitor - confidence: 0.97
Detected: Electrolytic Capacitor - confidence: 0.97
Detected: IC - confidence: 0.93
Detected: Connector - confidence: 0.92
Detected: Led - confidence: 0.90
Detected: IC - confidence: 0.90
Detected: IC - confidence: 0.90
Detected: IC - confidence: 0.90
Detected: Resistor - confidence: 0.89
Detected: Resistor - confidence: 0.86
Detected: IC - confidence: 0.83
Detected: Resistor - confidence: 0.79
Detected: Resistor - confidence: 0.77
Detected: Capacitor - confidence: 0.75
Detected: Resistor - confidence: 0.74
Detected: IC - confidence: 0.74
Detected: Resistor - confidence: 0.69
Detected: Capacitor - confidence: 0.68
Detected: Resistor - confidence: 0.68
Detected: Capacitor - confidence: 0.67
Detected: Capacitor - confidence: 0.66
Detected: Capacitor - confidence: 0.66
Detected: Resistor - confidence: 0.66
Detected: Capacitor - confidence: 0.65
Detected: Capacitor - confidence: 0.65
Detected: Connector - confidence: 0.64
Detected: Connector - confidence: 0.62
Detected: Resistor - confidence: 0.61
Detected: Resistor - confidence: 0.55
Detected: Capacitor - confidence: 0.54
Detected: Resistor - confidence: 0.52
Detected: Connector - confidence: 0.50
Detected: Capacitor - confidence: 0.50
Detected: Connector - confidence: 0.50
Detected: Capacitor - confidence: 0.44
Detected: Resistor - confidence: 0.43
Detected: Capacitor - confidence: 0.42
Detected: Resistor - confidence: 0.40
Detected: Resistor - confidence: 0.39
Detected: IC - confidence: 0.39
Detected: Button - confidence: 0.37
Detected: Resistor - confidence: 0.37
Detected: Led - confidence: 0.37
Detected: Led - confidence: 0.37
Detected: Resistor - confidence: 0.36
Detected: Capacitor - confidence: 0.35
Detected: Resistor - confidence: 0.35
Detected: Resistor - confidence: 0.34
Detected: Resistor - confidence: 0.34
Detected: Capacitor - confidence: 0.34
Detected: Resistor - confidence: 0.31
Detected: Connector - confidence: 0.29
Detected: Capacitor - confidence: 0.28
Detected: Capacitor - confidence: 0.27

Detected: Connector - confidence: 0.26

Rasppi Data

image 1/1 /content/drive/MyDrive/AER850 Project 3/data/evaluation/rasppi.jpg: 640x928 10 Capacitors, 10 Connectors, 1 Diode, 1 Electrolytic Capacitor, 13 ICs, 4 Padss, 16 Resistors, 1 Switch, 10.5ms
Speed: 4.4ms preprocess, 10.5ms inference, 2.3ms postprocess per image at shape (1, 3, 640, 928)
Detected: Electrolytic Capacitor - confidence: 0.96
Detected: IC - confidence: 0.95
Detected: Connector - confidence: 0.92
Detected: IC - confidence: 0.92
Detected: Connector - confidence: 0.90
Detected: IC - confidence: 0.90
Detected: IC - confidence: 0.87
Detected: Capacitor - confidence: 0.85
Detected: Capacitor - confidence: 0.84
Detected: Capacitor - confidence: 0.83
Detected: IC - confidence: 0.82
Detected: Connector - confidence: 0.81
Detected: Capacitor - confidence: 0.81
Detected: IC - confidence: 0.76
Detected: IC - confidence: 0.71
Detected: Connector - confidence: 0.70
Detected: Switch - confidence: 0.70
Detected: Connector - confidence: 0.68
Detected: Connector - confidence: 0.66
Detected: Pads - confidence: 0.65
Detected: Resistor - confidence: 0.61
Detected: Resistor - confidence: 0.58
Detected: Capacitor - confidence: 0.57
Detected: Pads - confidence: 0.56
Detected: Capacitor - confidence: 0.56
Detected: Resistor - confidence: 0.50
Detected: Resistor - confidence: 0.48
Detected: Capacitor - confidence: 0.48
Detected: Connector - confidence: 0.47
Detected: Capacitor - confidence: 0.45
Detected: Resistor - confidence: 0.42
Detected: Connector - confidence: 0.41
Detected: Resistor - confidence: 0.40
Detected: Resistor - confidence: 0.40
Detected: IC - confidence: 0.38
Detected: IC - confidence: 0.37
Detected: Capacitor - confidence: 0.36
Detected: IC - confidence: 0.36
Detected: Connector - confidence: 0.35
Detected: Resistor - confidence: 0.35
Detected: Pads - confidence: 0.33
Detected: Diode - confidence: 0.33
Detected: Capacitor - confidence: 0.32
Detected: IC - confidence: 0.32
Detected: Resistor - confidence: 0.32
Detected: Resistor - confidence: 0.31
Detected: Pads - confidence: 0.31
Detected: Connector - confidence: 0.31
Detected: IC - confidence: 0.30
Detected: Resistor - confidence: 0.29
Detected: Resistor - confidence: 0.28
Detected: IC - confidence: 0.28
Detected: Resistor - confidence: 0.27
Detected: Resistor - confidence: 0.27

Detected: Resistor - confidence: 0.26
Detected: Resistor - confidence: 0.25