Towards One-Shot PCB Defect Detection with YOLO

G. Spadaro\textsuperscript{1} \hspace{1cm} G. Vetrano\textsuperscript{2} \hspace{1cm} B. Penna\textsuperscript{2} \hspace{1cm} A. Serena\textsuperscript{2} \hspace{1cm} A. Fiandrotti\textsuperscript{1}

\textsuperscript{1}Computer Science Department  \\
University of Turin  \\
\textsuperscript{2}SPEA Company

Workshop - AI per l’Industria

Ital-IA

AI per l’Industria
World leader in the sector.

Automatic machinery for testing electronic boards.
World leader in the sector.

Automatic machinery for testing electronic boards.

Tester SPEA equipped with mobile probes.
World leader in the sector.

Automatic machinery for testing electronic boards.

Tester SPEA equipped with mobile probes.

SPEA machines are equipped with imaging systems.
World leader in the sector.
Automatic machinery for testing electronic boards.
Tester SPEA equipped with mobile probes.
SPEA machines are equipped with imaging systems.
World leader in the sector.

Automatic machinery for testing electronic boards.
Tester SPEA equipped with mobile probes.
SPEA machines are equipped with imaging systems.
World leader in the sector.

Automatic machinery for testing electronic boards.
Tester SPEA equipped with mobile probes.
SPEA machines are equipped with imaging systems.

✗ Bottleneck of the testing system
World leader in the sector.
Automatic machinery for testing electronic boards.
Tester SPEA equipped with mobile probes.
SPEA machines are equipped with imaging systems.

✗ Bottleneck of the testing system

Proposed workflow
YOLO is a one-stage detector.

Suitable for real-time application.
YOLO - You Only Look Once

YOLO is a **one-stage** detector.

Suitable for **real-time** application.

YOLO is a Deep Convolutional Neural Network designed to perform **object detection tasks**.

It consists of:

- **Backbone** extracts relevant features from the input image
- **Neck** combines these features
- **Head** is where the detection happens

*Figure: YOLO model [1]*

*Figure: YOLOv5 architecture*
Table of Contents

1. Dataset Generation

2. Experimental Results
Dataset Generation

We used images provided by the tester SPEA.

We defined a set of **39 classes** and we annotate the **central component** of each acquired image.

Why: Lack of information about the other components.

Class defined as Device Type and Case size.
Dataset Generation

We used images provided by the tester SPEA.

We defined a set of 39 classes and we annotate the central component of each acquired image.

Why: Lack of information about the other components.

Class defined as Device Type and Case size.

Images from the acquisition of different PCBs
Dataset Generation

We used images provided by the tester SPEA.
We defined a set of 39 classes and we annotate the central component of each acquired image.
Why: Lack of information about the other components.
Class defined as Device Type and Case size.

Images from the acquisition of different PCBs
Dataset Generation

We used images provided by the tester SPEA.

We defined a set of 39 classes and we annotate the central component of each acquired image. Why: Lack of information about the other components. Class defined as Device Type and Case size.

Images from the acquisition of different PCBs

Using CAD data of acquired PCBs we automatically labeled the central component.
Dataset Generation

We used images provided by the tester SPEA.

We defined a set of **39 classes** and we annotate the **central component** of each acquired image.

**Why:** Lack of information about the other components.

Class defined as Device Type and Case size.

Images from the acquisition of **different PCBs**

Using CAD data of acquired PCBs we **automatically** labeled the **central component**.
Dataset Generation

We used images provided by the tester SPEA.

We defined a set of **39 classes** and we annotate the **central component** of each acquired image. Why: Lack of information about the other components.

Class defined as Device Type and Case size.

Images from the acquisition of **different PCBs**

Using CAD data of acquired PCBs we automatically labeled the **central component**.

Our Dataset is then composed of images having the label of only the **central components** of different PCBs.
**Problem:** Images are annotated only with the central component, i.e. we have a lot of **False Negative**.
Problem: Images are annotated only with the central component, i.e. we have a lot of False Negative.

Solution: Considered these “crops” to automatically reconstruct and annotate the overall image of each PCB.
**Problem:** Images are annotated only with the central component, i.e. we have a lot of False Negative.

**Solution:** Considered these “crops” to automatically reconstruct and annotate the overall image of each PCB.

For each PCB of the Dataset:
**Problem**: Images are annotated only with the central component, i.e. we have a lot of False Negative.

**Solution**: Considered these “crops” to automatically reconstruct and annotate the overall image of each PCB.

For each PCB of the Dataset:

- We filled an empty images with its crops.
**Problem:** Images are annotated only with the central component, i.e. we have a lot of False Negative.

**Solution:** Considered these “crops” to automatically **reconstruct** and annotate the overall image of each PCB.

For each PCB of the Dataset:

- We filled an empty images with its crops.
- Position obtained converting the machine coordinates into pixel coordinates.
Problem: Images are annotated only with the central component, i.e. we have a lot of False Negative.

Solution: Considered these “crops” to automatically reconstruct and annotate the overall image of each PCB.

For each PCB of the Dataset:

- We filled an empty images with its crops.
- Position obtained converting the machine coordinates into pixel coordinates.
- Bounding box and class of the central component were also reported.
Problem: Images are annotated only with the central component, i.e. we have a lot of False Negative.

Solution: Considered these “crops” to automatically reconstruct and annotate the overall image of each PCB.

For each PCB of the Dataset:

- We filled an empty images with its crops.
- Position obtained converting the machine coordinates into pixel coordinates.
- Bounding box and class of the central component were also reported.

We recreated and annotated 11 boards.
Dataset Generation

Figure: 60000x20000 image of the Top side of the CPE010 PCB reconstructed using 354 crops
Dataset Generation

Figure: 60000x20000 image of the Top side of the CPE010 PCB reconstructed using 354 crops

We took crops of these images to create a dataset with 5,490 images correctly annotated (i.e. without False Negative) and with all of the 39 classes of components.
### Summary Table

<table>
<thead>
<tr>
<th>Component class</th>
<th># Samples</th>
<th>$\mu m^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resistor_0402</td>
<td>511</td>
<td>756.81</td>
</tr>
<tr>
<td>Resistor_0603</td>
<td>967</td>
<td>1.884.59</td>
</tr>
<tr>
<td>Resistor_0805</td>
<td>472</td>
<td>3.885.82</td>
</tr>
<tr>
<td>Resistor_1206</td>
<td>47</td>
<td>7.076.86</td>
</tr>
<tr>
<td>Resistor_1210</td>
<td>2</td>
<td>6.584.29</td>
</tr>
<tr>
<td>Resistor_RMINIMELF</td>
<td>3</td>
<td>7.698.42</td>
</tr>
<tr>
<td>Resistor_Array</td>
<td>92</td>
<td>7.940.38</td>
</tr>
<tr>
<td>Resistor_2010</td>
<td>9</td>
<td>18.547.78</td>
</tr>
<tr>
<td>Resistor_2512</td>
<td>20</td>
<td>30.649.76</td>
</tr>
<tr>
<td>Capacitor_0402</td>
<td>958</td>
<td>794.03</td>
</tr>
<tr>
<td>Capacitor_0603</td>
<td>886</td>
<td>1.710.48</td>
</tr>
<tr>
<td>Capacitor_0805</td>
<td>404</td>
<td>3.155.46</td>
</tr>
<tr>
<td>Capacitor_1206</td>
<td>93</td>
<td>6.296.04</td>
</tr>
<tr>
<td>Capacitor_1210</td>
<td>39</td>
<td>13.096.62</td>
</tr>
<tr>
<td>Capacitor_Polar_0603</td>
<td>13</td>
<td>3.990.02</td>
</tr>
<tr>
<td>Capacitor_Polar_CMKTA</td>
<td>20</td>
<td>8.554.51</td>
</tr>
<tr>
<td>Capacitor_Polar_1411P</td>
<td>3</td>
<td>16.262.89</td>
</tr>
<tr>
<td>Capacitor_Polar_CMKTB</td>
<td>1</td>
<td>29.971.18</td>
</tr>
<tr>
<td>Capacitor_Polar_CMKTD</td>
<td>20</td>
<td>48.392.77</td>
</tr>
<tr>
<td>Inductor_1210</td>
<td>4</td>
<td>69.504.67</td>
</tr>
<tr>
<td>Inductor_IND-XAL4020</td>
<td>4</td>
<td>27.742.08</td>
</tr>
<tr>
<td>Inductor_INDIHL2525CZ01</td>
<td>4</td>
<td>67.996.82</td>
</tr>
<tr>
<td>Fuse_0603</td>
<td>8</td>
<td>2.121.57</td>
</tr>
<tr>
<td>Fuse_FUSESM</td>
<td>6</td>
<td>21.973.29</td>
</tr>
<tr>
<td>Fuse_FUSE-SMDC020</td>
<td>2</td>
<td>24.644.09</td>
</tr>
<tr>
<td>Led_0805</td>
<td>56</td>
<td>4.483.81</td>
</tr>
<tr>
<td>Led_TEKTONE.LED.1411</td>
<td>4</td>
<td>13.329.49</td>
</tr>
<tr>
<td>Connector_CMIMA4VFD_SM</td>
<td>2</td>
<td>59.097.38</td>
</tr>
<tr>
<td>Connector_CMIMA6VFD</td>
<td>2</td>
<td>76.665.90</td>
</tr>
<tr>
<td>Potentiometer_SMRVAR1</td>
<td>1</td>
<td>33.786.26</td>
</tr>
<tr>
<td>Relay_RLPICK-117-1A</td>
<td>52</td>
<td>42.563.62</td>
</tr>
<tr>
<td>Switch_Array_PULSOMRON</td>
<td>1</td>
<td>56.310.86</td>
</tr>
<tr>
<td>Diode_DMELF</td>
<td>2</td>
<td>18.398.49</td>
</tr>
<tr>
<td>Cylindrical_dioode</td>
<td>71</td>
<td>7.481.30 - 7.538.37</td>
</tr>
<tr>
<td>Metallic_packaging</td>
<td>6</td>
<td>23.934.04 - 52.777.34</td>
</tr>
<tr>
<td>Plastic_packaging</td>
<td>706</td>
<td>878.41 - 70.537.68</td>
</tr>
</tbody>
</table>

Table: The PCB component classes considered in this work with number of samples and packaging area over the 11 PCB images we were provided.
Table of Contents

1. Dataset Generation

2. Experimental Results
Experimental Results

Results

We followed a leave-one-out approach

- All available boards as a training set, leaving one PCB out as a test set

<table>
<thead>
<tr>
<th>Test set</th>
<th>mAP@0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPE010</td>
<td>0.775</td>
</tr>
<tr>
<td>JPAMA30-256K SN</td>
<td>0.704</td>
</tr>
<tr>
<td>KDBRLYCMDR3</td>
<td>0.850</td>
</tr>
<tr>
<td>KEXANADUX70V1</td>
<td>0.952</td>
</tr>
<tr>
<td>LI122SM-2_CB533_009</td>
<td>0.819</td>
</tr>
<tr>
<td>MPSDRV608</td>
<td>0.882</td>
</tr>
<tr>
<td>SPE010-2</td>
<td>0.994</td>
</tr>
<tr>
<td>Z010500 SN</td>
<td>0.524</td>
</tr>
<tr>
<td>ZCPU7Z0</td>
<td>0.787</td>
</tr>
<tr>
<td>ZPROMEA50_SN_02680</td>
<td>0.783</td>
</tr>
<tr>
<td>ZPROMEA50_SN_01115</td>
<td>0.812</td>
</tr>
<tr>
<td><strong>Mean</strong></td>
<td><strong>0.808</strong></td>
</tr>
</tbody>
</table>

**Table:** mAP@0.5 for the board left out of the training set (all board images are reconstructed from patches).
Experimental Results

Last experiment

We automatically annotate a real complete image of the ZPROMEA board from which we took larger crops composed of a large number of components.


Experimental Results

Last experiment

We automatically annotate a real complete image of the ZPROMEA board from which we took larger crops composed of a large number of components.

Labeled image of the produced Test Set
Experimental Results

Last experiment

We automatically annotate a real complete image of the ZPROMEA board from which we took larger crops composed of a large number of components.

Labeled image of the produced Test Set

[Graph showing Precision-Recall curve with various components and their recall values.]

Resistor_1206 0.995
Capacitor_0402 0.991
Fuse_FUSESM 0.604
Plastic_packaging 0.807
Inductor_1210 0.975
Resistor_0402 0.640
Capacitor_1210 0.995
Cylindrical_diode 0.595
Led_0805 0.995
Resistor_0603 0.994
Capacitor_0603 0.990
Resistor_0805 0.994
all classes 0.881 mAP@0.5
Conclusions

We created a dataset containing images annotated only with the central component.
Conclusions

We created a dataset containing images annotated only with the central component.

We have extended these annotations to reconstructed PCB images.

Future works:

➣ Consider a new detection module in the head of the network.

➣ Acquire new boards to balance the distribution of components.
Conclusions

We created a dataset containing images annotated only with the central component.

We have extended these annotations to reconstructed PCB images.

We created a dataset correctly annotated to avoid false negative.
Conclusions

We created a dataset containing images annotated only with the central component.

We have extended these annotations to reconstructed PCB images.

We created a dataset correctly annotated to avoid false negative.

We considered a component detection task in the wild creating a Test set having larger images with several components.
Conclusions

We created a dataset containing images annotated only with the central component.

We have extended these annotations to reconstructed PCB images.

We created a dataset correctly annotated to avoid false negative.

We considered a component detection task in the wild creating a Test set having larger images with several components.

Future works:
Conclusions

We created a dataset containing images annotated only with the central component.

We have extended these annotations to reconstructed PCB images.

We created a dataset correctly annotated to avoid false negative.

We considered a component detection task in the wild creating a Test set having larger images with several components.

Future works:

➢ Consider a new detection module in the head of the network.
Conclusions

We created a dataset containing images annotated only with the central component.

We have extended these annotations to reconstructed PCB images.

We created a dataset correctly annotated to avoid false negative.

We considered a component detection task in the wild creating a Test set having larger images with several components.

Future works:

➢ Consider a new detection module in the head of the network.
➢ Acquire new boards to balance the distribution of components.
Thank you for your attention.

Questions?
You only look once: Unified, real-time object detection.