Towards One-Shot PCB Defect Detection with YOLO

G. Spadaro[1] G. Vetrano[2] B. Penna[2] A. Serena[2] A. Fiandrotti[1]

[1]Computer Science Department
University of Turin

[2]SPEA Company

Workshop - AI per l'Industria

Ital-IA
ITALIA INTELLIGENZA ARTIFICIALE
Terzo Convegno Nazionale
Pisa 29-31 Maggio 2023
cini National Lab AIIS

SPEA

EIDOSLAB

SPEA

World leader in the sector.

Automatic machinery for testing electronic boards.

SPEA

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.

SPEA

EIDOSLAB

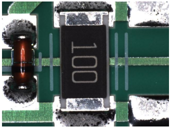
SPEA

World leader in the sector.

Automatic machinery for testing electronic boards.

Tester SPEA equipped with mobile probes.

SPEA machines are equipped with imaging systems.



SPEA

EIDOSLAB

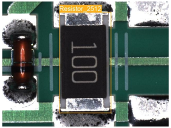
SPEA

World leader in the sector.

Automatic machinery for testing electronic boards.

Tester SPEA equipped with mobile probes.

SPEA machines are equipped with imaging systems.



Current workflow

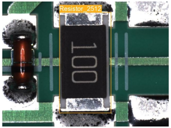
SPEA

World leader in the sector.

Automatic machinery for testing electronic boards.

Tester SPEA equipped with mobile probes.

SPEA machines are equipped with imaging systems.



Current workflow

✗ Bottleneck of the testing system

SPEA

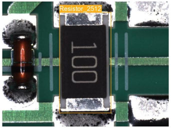
EIDOSLAB

World leader in the sector.

Automatic machinery for testing electronic boards.

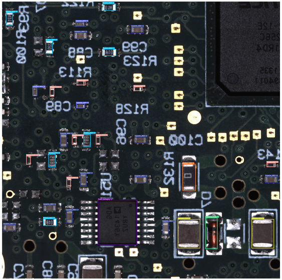
Tester SPEA equipped with mobile probes.

SPEA machines are equipped with imaging systems.



Current workflow

✗ Bottleneck of the testing system



Proposed workflow

SPEA

EIDOSLAB

YOLO - You Only Look Once

YOLO is a one-stage detector.

Suitable for real-time application.

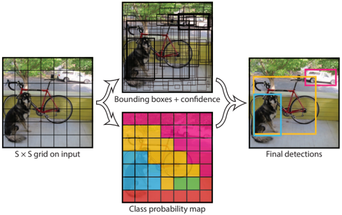


Figure: YOLO model [1]

YOLO - You Only Look Once

YOLO is a one-stage detector.

Suitable for real-time application.

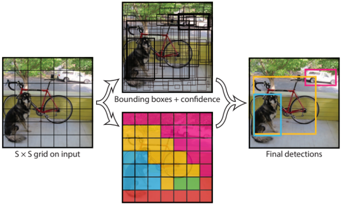


Figure: YOLO model [1]

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

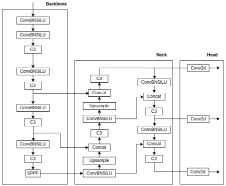


Figure: YOLOv5 architecture

It consists of:

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

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

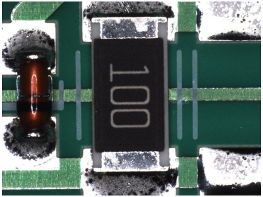
We used images provided by the tester SPEA.

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



SPEA

EIDOSLAB

Dataset Generation

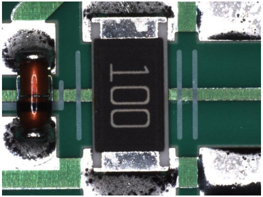
We used images provided by the tester SPEA.

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

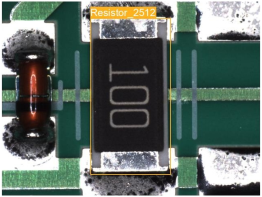
We used images provided by the tester SPEA.

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

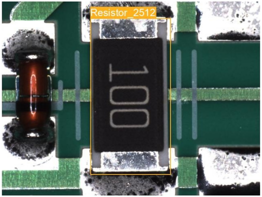
We used images provided by the tester SPEA.

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.

Dataset Generation

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.

Dataset Generation

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.

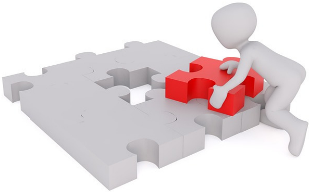
For each PCB of the Dataset:

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.

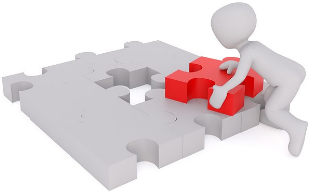


Problem: Images are annotated only with the central component, i.e. we have a lot of False Negative.

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.



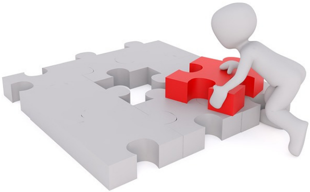
Dataset Generation

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.



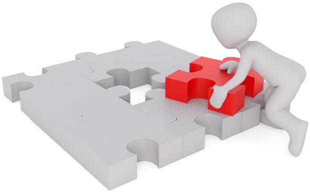
Dataset Generation

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

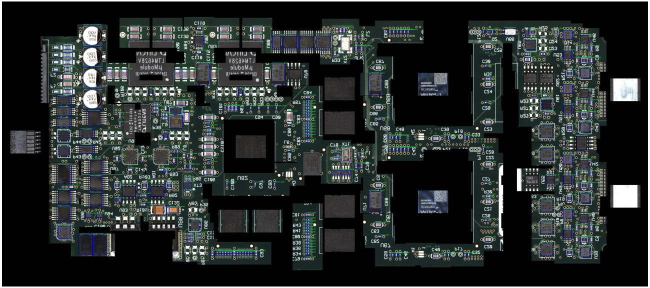


Figure: 60000x20000 image of the Top side of the CPE010 PCB reconstructed using 354 crops

Dataset Generation

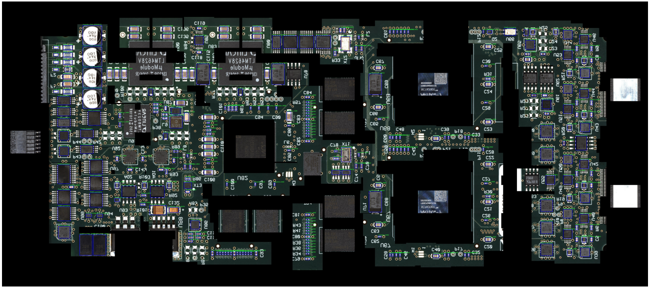
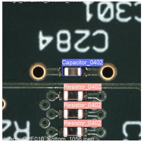


Figure: 60000x20000 image of the Top side of the CPE010 PCB reconstructed using 354 crops

We 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

Component class	# Samples	μm^2
Resistor_0402	511	756,81
Resistor_0603	967	1.884,59
Resistor_0805	472	3.885,82
Resistor_1206	47	7.076,86
Resistor_1210	2	6.584,29
Resistor_RMINIMELF	3	7.698,42
Resistor Array	92	7.940,38
Resistor_2010	9	18.547,78
Resistor_2512	20	30.649,76
Capacitor_0402	958	794,03
Capacitor_0603	886	1.710,48
Capacitor_0805	404	3.155,46
Capacitor_1206	93	6.296,04
Capacitor_1210	39	13.096,62
Capacitor_Polar_0603	13	3.990,02
Capacitor_Polar_CMKTA	20	8.554,51
Capacitor_Polar_1411P	3	16.262,89
Capacitor_Polar_CMKTB	1	29.971,18
Capacitor_Polar_CMKTD	20	48.392,77

Component class	# Samples	μm^2
Capacitor_Polar_CEVPA8X10	4	69.504,67
Inductor_1210	4	13.988,71
Inductor_IND-XAL4020	4	27.742,08
Inductor_INDIHLP2525CZ01	4	67.996,82
Fuse_0603	8	2.121,57
Fuse_FUSESM	6	21.973,29
Fuse_FUSE-SMDC020	2	24.644,09
Led_0805	56	4.483,81
Led_TEKTONE_LED_1411	4	13.329,49
Connector_CMIMA4VFD_SM	2	59.097,38
Connector_CMIMA6VFD	2	76.665,90
Potentiometer_SMRVAR1	1	33.786,26
Relay_RLPICK-117-1A	52	42.563,62
Switch Array_PULSOMRON	1	56.310,86
Diode_DMELF	2	18.398,49
Cylindrical_diode	71	7.481,30 - 7.538,37
Metallic_packaging	6	23.934,04 - 52.777,34
Plastic_packaging	706	878,41 - 70.537,68

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

Table of Contents

1. Dataset Generation

2. 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

Test set	mAP@0.5
CPE010	0.775
JPAMA30-256K SN	0.704
KDBRLYCMDR3	0.850
KEXANADUX70V1	0.952
LI122SM-2_CB533_009	0.819
MPSDRV608	0.882
SPE010-2	0.994
Z010500 SN	0.524
ZCPU7Z0	0.787
ZPROMEA50_SN_02680	0.783
ZPROMEA50_SN_01115	0.812
Mean	**0.808**

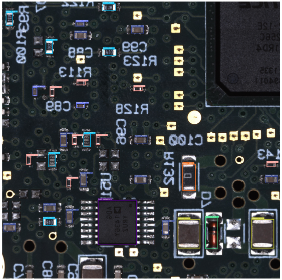
Table: mAP@0.5 for the board left out of the training set (all board images are reconstructed from patches).

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

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

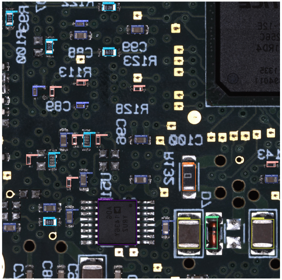


Labeled image of the produced Test Set

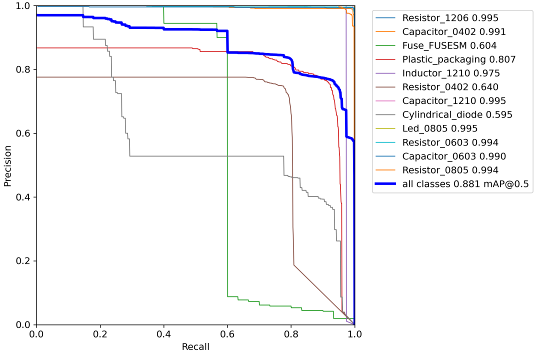
Experimental Results

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



SPEA EIDOSLAB

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.

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?

References

[1] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi.
You only look once: Unified, real-time object detection.
CoRR, abs/1506.02640, 2015.