Inductor Image Classification using Detectron2

01

Idea & Implementation

- We developed an approach using component coordinate points to create a training dataset that enables machine learning models to differentiate between various electronic components.
- Since direct coordinate export from KiCad PCB Editor wasn't possible, we utilized Labelme to manually annotate component coordinates, creating a comprehensive training dataset.
- The annotation process using Labelme allowed us to precisely mark component locations and classify them. This detailed annotation enabled effective training and testing, with optimized data loading for rapid iteration.
- We focused on two distinct classes - "Inductor" and "Donut" - to train the model in distinguishing between visually similar objects. The key differentiating factor in annotation was the inclusion of vertices in "Inductor" images, helping the model learn distinctive features between the classes.
- We implemented Detectron2's Faster R-CNN architecture with a ResNet-101 backbone and Feature Pyramid Network (FPN) for accurate object detection and classification. The model, initially pre-trained on the COCO dataset, was fine-tuned on our custom annotated dataset to specifically identify electronic components.

Working

Input Image
(Pre-processed image for model input)

→

Feature Extraction
Resnet 101 + FPN
(Extracts multi-scale feature maps)

→

Region Proposal Network
(Generates candidate bounding boxes)

→

ROI Pooling
(Pools regions for object classification)

→

Classification & Box Refinement
(Assigns class labels and refines box positions)

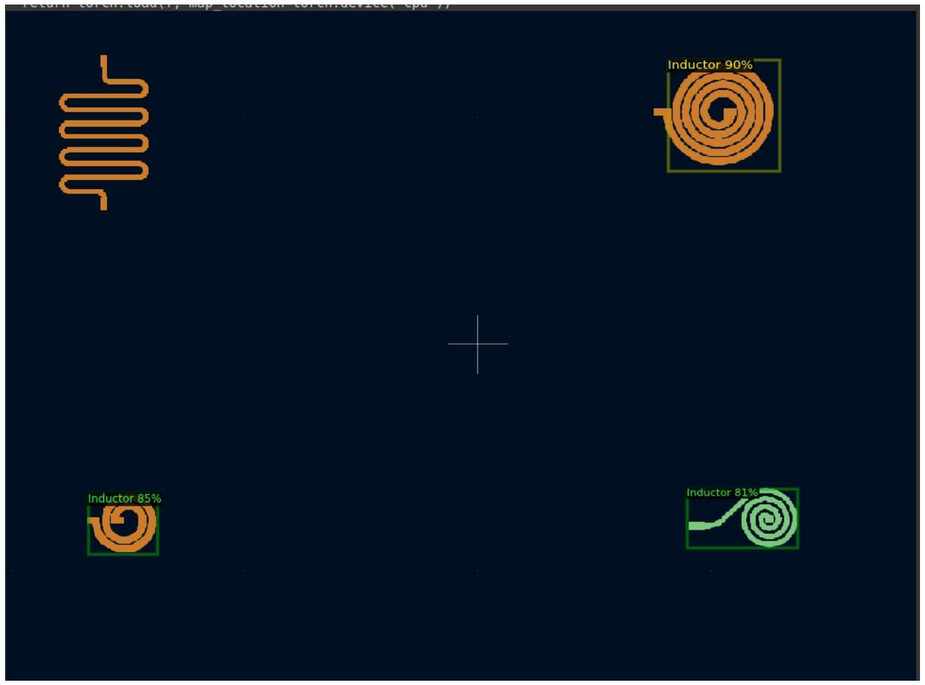
→

Output
(Assigns class labels and refines box positions)

Annotated Images using coordinates of components

(Used for training the model)

Output



Group Members -
Navneet Mishra (MT23049)
Monika Singh (2021169)

Code Link - [Github]