Electronic Component Detector Project using YOLOv11

-By Labib Tahmid

Project Concept:

Many electronics labs work with various electrical component. Usually, they are not organized when using them, so they look messy. My project, Electronic Component Detector can solve this issue by finding which component is what and the number of component present in an image, video or even live webcam feed.

Unique Objects:

For this project, as electronic component, I choose the followings:

1. Arduino
2. ESP32
3. Battery
4. DC motor
5. Sensor
6. Display

I wanted to add resistors, diode but they are very small and can cause problem, so the previously discussed components were selected.

Dataset:

As there are 6 class, I took 50 pictures of each component and make total of 300 images. Each 50 images were divided by following,

1. 20 images for single component
2. 20 images with another component
3. 8 images with 2 other components
4. 2 images with 3 or more components

Then in Roboflow, I annotated all 300 Images. After annotation, for augmentation following were used,

1. Flip: Horizontal
2. Rotation: Between -15° and +15°
3. Saturation: Between -20% and +20%
4. Brightness: Between -20% and +20%
5. Exposure: Between -10% and +10%

6. Blur: Up to 1.5px
7. Noise: Up to 0.5% of pixels

The final dataset have the following split,

1. Train set = 600 Images
2. Validation set = 80 Images
3. Test set = 20 Images

Model:

For first attempt, I used YOLOv8 model. But the dataset and model performance was not satisfying to me, so I generated previous discussed dataset and used YOLOv11 Model. Two model training was run. The findings were following,

1. For Epochs=50, batch=16,
   mAP50 = 99.5%, mAP50-95 = 95.5%
2. For Epochs=100, batch=32,
   mAP50 = 99.5%, mAP50-95 = 96.6%

Second model run was chosen as **best.pt**.

Streamlit Dashboard:

A web dashboard was developed, using streamlit. The dashboard has the following functions,

1. Image uploading
2. Video uploading
3. Webcam/Live feed

The dashboard has options to choose between them. Also, a slider is present to change confidence threshold dynamically.

Result Finding:

Initially, model performed much better on the test set from the dataset. But I used several other test images, video and also live feed to test further. Following is analysis of mine,

1. **Images:** Images were interesting because there are some distance image and some close-up image. Model performed very well on close-up images, less error there. But for distance images it most of the time classified components as motor or battery, or cannot detect at all.

The reason I think is, first the dataset is small, second images were taken same environment. That's why it performed the best in the same environment. Third is angle and lighting. Angled image performed better, also in the same lighting. Those images not containing these performed poorly.

2. **Video & Live feed:** Video and live feed surprisingly did much better than images. One reason is I took the videos in close distance, also as video is a continuous process, model fixed it error quickly.

Future Improvement:

To improve the model more, I can think of these steps,

1. I need to take more images, I think 1000 before augmentation.
2. The images should contain different distance, angle and environment.
3. These 6 classes are generic classes, I can do more and different classes too.

Important URLS:

The Roboflow Project Link: https://app.roboflow.com/labib-projects/electronic-component-detection-r2mbp/models/electronic-component-detection-r2mbp/1

Previous YOLOv8 Model Training Link: https://colab.research.google.com/drive/1GrBSlB8_sJMz802KSbALIZyqNL9_wUAT?usp=sharing

Current YOLOv11 Model Training Link: https://colab.research.google.com/drive/1I6ks_Kj34pcOR1bxVryrcTv2Yckn0UTQ?usp=sharing

GITHUB Link: https://github.com/LTNEW52/Bondstein-Project-Electronic-Component-Detection