Parts Recognition
By Computer Vision Intelligent Service

(Ron, April 2019)

Project Summary

Computer vision, AI and Big Data technologies from Cloud, HD cameras, High resolution imagery, digital CAD designs, IOT devices & sensors to Machine Learning techniques across industry verticals provide a great opportunity to enable Field Service Technicians with improved productivity and efficiency with minimal resolution time by autodetecting defective parts in near real time with great customer experience.

This invention collects, processes and analyzes signals related to defective subcomponents of machines and equipment across different ISVs (e.g. Semiconductor, Oil and Gas, Smart Buildings, Manufacturing etc.) and builds 'parts (sub-components) recognition' machine learning model(s) with actionable recommendations for field service experts to improve technician productivity and provide great customer experience.

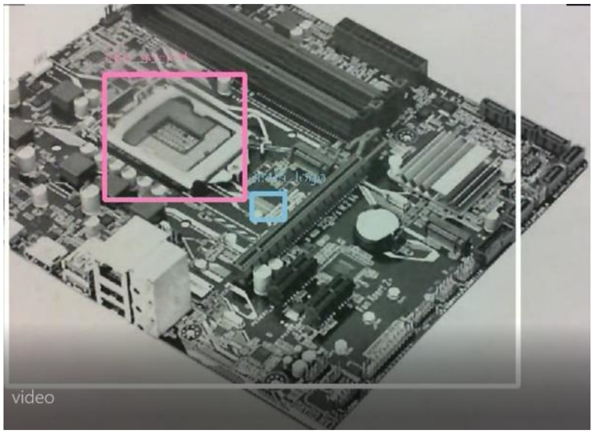
Signals used for this invention include, but not limit to, the following:

1. Localization - Locate the position of the detected subcomponent by X, Y coordinates
2. Depth – Granularity of subcomponent
3. Distance/Angle – Subcomponents image from different camera angles
4. Light/Shade – Modalities of light/grey/dark/color/tone
5. Texture – RGB spectrum
6. Parts/Serial number of subcomponent - Texts
7. Dynamic image inferences – moving images for RT image feed
8. Associating Image of a subcomponent to customer asset properties and entities

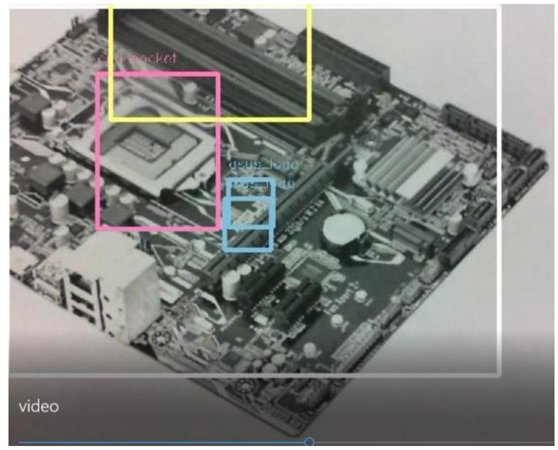
In computer vision technology, a typical "object detection" model detects a single object or multiple objects from any given static image at a higher granularity i.e. objects but not sub-components or parts of an object. To maximize the success rate of detecting parts, we propose using camera to scan the target and capture the best frame when many parts are recognized. The client application may need to programmatically capture several frames with a high number of detected objects, and then let the user choose the best one from them.

Alternatively, from a series of frames, every frame may have different confidence scores for the same parts (i.e. objects) being detected. We can programmatically capture this information and compute by their means or percentile to generate the best prediction for parts detection.

In the following pictures, the colored bounding boxes indicate detected parts from the motherboard inside the picture. As you can see, our model was able to recognize more parts in figure 1 than in figure 2. Obviously using the static image from figure 1 will miss at least two parts of information that were successfully captured in figure 2.



(Figure 1)



(Figure 2)

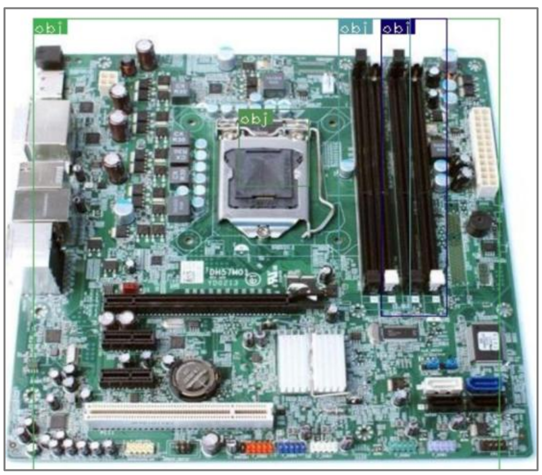
This “parts-recognition” machine learning service uses an autonomous system agent to automatically capture, process and integrate signals from image, sensors and IoT devices. The intelligent ML system will then automatically auto-label those subcomponents and associate with customer asset properties in the system. The service also allows customer feedback to the service will continue to iterate and get better & more intelligent with autonomous parts (subcomponents) detection with higher accuracy. This intelligent service in near real time will not only help Field Technicians across multiple ISVs with preventative repair and predictive maintenance scenarios but also allow enterprises to potentially have millions of dollars in savings.

Pretrain and Retrain Models

Given the mission of our "Parts Recognition" model, we decouple the training process into two major steps

1. We build a pretrained model with the capability of detecting key parts in each image. We will provide this model to customers. It will help customers to create their models much more efficiently.
2. Our Dynamic 365 customers will use our pretrained model to get suggested areas for possible parts from any images they scan. Customers then label the recognized parts from the images with their domain knowledge and retrain the model with these labeled data.

As an example, the pretrained model needs to be able to tell some key parts on the motherboard (see figure 3), while our customers will leverage the suggested "obj" area (see figure 3) for labeling and use it to retrain the model. The resulting model will be able to tell both the exact names and the exact locations of the detected objects (see figure 4).



(Figure 3)

(Figure 4)

<u>Pretrained Model Steps:</u>

1. We want to ensure all the training labels cover as many reasonable sub-parts or sub-areas as possible, to maximize the object localization performance after training.

2. In the "Object Detection", the loss function is formed to calculate loss from both classification and localization. But, in the "Parts Recognition", when we build pretrained model, we care about localization, rather than classification. There are two steps to compute loss here:

Step 1: Once we detect all candidate bboxes (based on RPN, or anchor boxes techniques), we apply IOU computation between each candidate bbox and the annotated bbox. We then pick the candidate with the highest IOU.

Step 2: We use the loss function to compute MSE.

Let’s say the target variable: $y_t = [p_c \ \ b_x \ \ b_y \ \ b_h \ \ b_w \ \ c_1 \ \ c_2 \ \ \ldots \ \ c_n]$

and candidate bbox pool: y_i,

the best candidate bbox: $y_d = IOU_{max}(y_i, y_t)$, $y_d = [p_{dc} \ \ b_{dx} \ \ b_{dy} \ \ b_{dh} \ \ b_{dw} \ \ c_{d1} \ \ c_{d2} \ \ \ldots \ \ c_{dn}]$

the loss function will be:

$$L(y\hat{},y) = \alpha\,(b_x - b_{dx})^2 + \alpha\,(b_y - b_{dy})^2 + \beta\,(b_h - b_{dh})^2 + \beta\,(b_w - b_{dw})^2$$

$\alpha > \beta$ are weights, it indicates that the accuracy of locating the centroid of an object is more important, therefore it should have higher weights, but we don’t have quantified numbers for them at this point.

Step 3: Because of the focus on this training step is about “object localization”, we need to use a more accurate performance measure to truly represent how well the model detects parts. The formula we use is:

$$\mathrm{mAP} = (1/n)\sum_{k=1}^{n} \mathrm{IOU}(\mathrm{Yt(k)}, \mathrm{Ya(k)})$$

Yt(k) is the target variable for the Kth bbox
Ya(k) is the actual value for the Kth bbox
(Each bbox represents a part, or an object.)

These changes should be able to reduce computation pressure in loss function, improve training efficiency and accuracy of the object localization.

Retraining Model

Customers will be provided with an out-of-the-box pretrained model. Also, the service offering will allow customers to customize the pretrained model pertinent to their use case or specific ISVs to retrain the pretrained model with their specific domain data.

Customization features will include the following functionalities to allow customers with feedback/input:

1. Customers will get all suggested areas for parts (out-of-the-box capability) from an image.
2. Customers have the flexibility to pick and choose the detected area as a targeted part for labeling.
3. Customers can also adjust the size of the area.
4. We also want to provide a bunch of similar open image data programmatically for customers to decide whether they want to include them in their retraining process.