**Problem Statement:**

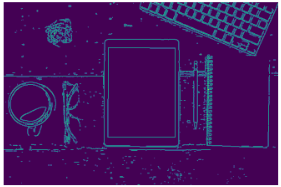
Find the bounding boxes of all the objects in the given image. No Deep Learning architecture can be used.



The idea used to solve the object detection problem is by **detecting the edges** and **finding the contours** which can help us detect the object perimeter**.**

**Challenges to solving the problems:**

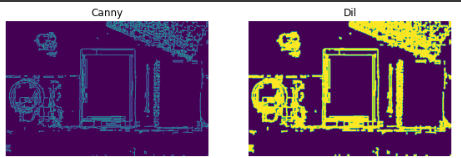
* **Non-Uniformity of the background:** In the given image, the background is a table top. The color of the table is not quite uniform. There are some sudden black shades in the lower segment of the image. This may lead to some noisy edge detection as the pixel value gradient is high in some of these regions. Similarly, there is wide brighter line that runs horizontally. The gradient of the pixels again is very high in these region, leading to edge detection. These noisy edges mix up with the edges of objects in the image. These creates a problem in the contour definition of the objects.

****

* **Distance between the objects:** Some of the objects like the pen notepad are a bit close to each other. Dilating the detected edges (for better contours of the objects), may cause problem. The edges of separate object may get mixed. This might cause failure to identify close objects as separate entities.

**Solution to the challenges:**

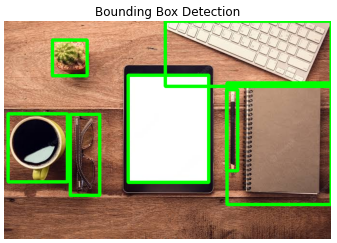
* **Canny Edge detection** is applied to the problem, and then the contours are generated. The Hysteresis Thresholds in Canny edge detection, decides the actual edges from the bunch, based on maximum gradient and minimum gradient threshold. These threshold values are adjusted for detection of the objects in the image.
* The canny detected image is then dilated for producing better contours of object detection. The shape of the contour is estimated and bounding boxes are generated. To ignore irrelevant contours generated, the area of the contour is adjusted.

****

* In spite of tuning all the Parameters like **canny thresholds, dilation iterations** and **area of contours,** no particular set of value for these parameters is able to detect all the objects. So we use different sets of parameter configuration for detecting all the objects. **Three canny edge detected images are obtained from three configurations.**
* In these 3 configuration, different **color information** is also used to detect some objects. **RGB value** as well as **Hue** value from **HSV** format is used for edge detection. Hue is able to distinguish the close prominent edges of the background from that of the object. Each configuration produces a bounding boxes for a set of objects in the image. In this way bounding box is obtained for all the objects are produced.

****

* In this method, more than one bounding box for the same objects is accumulated, from all the configuration. **Redundant bounding boxes** needed to be eliminated. **Non-Maximum Suppression** does this job. It removes bounding boxes whose **IOU** is above a threshold, defined by us. Thus, finally the resultant bounding boxes obtained after Non-Maximum Suppression is shown below.

****

**Thoughts on Improvement:**

Image masking techniques can be used to detect the noisy background effectively. Background detection algorithms can be tested for distinguishing the foreground and background. And finally, parameter tuning can always be improved for better results.