

## **Task Documentation**

The given dataset has a variety of noise included into it. Few observations we can see by looking at the image are :

1. The image sizes are different in some cases.
2. Some of the images are clear and few are blurred.
3. Most of the images have check signs enclosed into the checkbox but few have outside the walls as well.
4. Some images consist of adjacent checkboxes with partial parts visible in the image.

### **The Approach :**

1. Load the Image
2. Convert it to grayscale.
3. Apply thresholding on the image to convert it into binary format.
4. We then find the contours of the image.
5. Applying "approxPolyDP" to approximate the shapes of the contours iteratively.
6. Get the X,Y coordinate of the rectangle and also the Width and Height of the rectangle using "boundingRect".
7. Then we draw a rectangle using these X,Y,W,H over the image.

### **Step By Step Working:**

- Step 1 : First of all we create a for loop to get the images one by one. We can also use the listdir of the OS module with the directory path to store all of them all together. Then we create a copy of our image to retain the original image.
- Step 2 : Here we convert the image into grayscale, then we apply the thresholding using "threshold" . It converts the image into binary and also to get the areas that are of our interest.
- Step 3 : After converting it into binary format, we apply the "findcontours" method to get the contours. It works generally by getting the pixel intensity difference where it has the high difference. Then it creates boundary lines and these are generally called contours.
- Step 4 : Then we iteratively apply "approxPolyDP" and then "boundingRect" on each contour to approximate the shape as quadrilateral and then find the x,y,w,h for each rectangle.
- Step 5: After getting the x,y and width and heights we initiate a loop looking for the biggest square we can draw using the contours. It is because of the fear of losing information from the image if our model does not identify the squares properly.
- Step 6: Then at the end we draw a rectangle over the original image which we retained as the copy in our choice of colour, which is red in this case.