**BaggageAI Assignment Solution**

* **Notebook name**: Assignment\_CV.ipynb
* **Input:** All images from threat\_images and background\_images directory
* **Output:** All combinations of threat images pasted on different background images as per the given specifications
* **Sample Output:** 5 threat images and 5 background images were given. It generated total 25 output images. (Combination of all threat with all background images)
* **Note:** To test program on other images, add more threat images and background images in respective directories and run the script. All output combinations of those images will be generated in result\_images directory.

**PROGRAM DESCRIPTION**

**boundingBox() function**

* Input: Image path
* Outputs:
  + Top left and bottom right coordinates of rectangle around the largest object in the input image
  + Centre points of the largest object in the input image
  + Area of the largest object in the image
* Description:
  1. Read image in grayscale and apply gaussian filter to supress noise
  2. Use Binary inverse thresholding to get binary image with black background
  3. Find biggest contour and get top left and bottom right coordinates of bounding rectangle for that contour
  4. Fit ellipse around that contour to find true centre of largest object in the image
  5. Return rectangle coordinates, ellipse centre points and contour area

**translucent() function**

* Input: PIL image object
* Output: Translucent mask (PIL image object)
* Description:
  1. Convert PIL image object to cv2 image object then convert it to grayscale
  2. Use gaussian filter and inverse binary thresholding to find binary image mask (0 for black-background,255 for white-object), so that background pixels will become completely transparent when it is used as a mask in *Image.paste()* method
  3. Change all non-zero element (white-object) in the mask to 150 value, so that it will become translucent when it is used as a mask in *Image.paste()* method
  4. Convert mask image to PIL image object and return it

**Finding image files**

* *os.walk()* is used for preparing two lists
  1. *threat\_img\_list*: list of the names of threat images from *threat\_images* directory
  2. *bg\_img\_list*: list of the names of background images from *background\_images* directory
* Results will be saved in *result\_images* directory

**Image processing**

Nested for loops

* Outer loop for every background image
  + 1. Center points and Area of background object is found by using *boundingBox(image\_path)* function for every background image.
    2. *resize\_l* is new length and width for threat object such that area of threat object will be 8 times smaller than respective background image. (So that threat object will look proportionate to background object, for large background object: large threat object and likewise for small size)
* Inner loop for every threat image corresponding to outer loop background image
  + 1. Top left and bottom right coordinates of rectangle surrounding threat object is found by using *boundingBox(image\_path*) function.
    2. Random x,y coordinates are calculated such that threat object pasted on background image does not get outside the background object
    3. Threat image is cropped in rectangle as per coordinates found above
    4. Threat image is rotated by 45 degree and white color is used to fill in the background. *expand=True* argument is passed so that edges of image do not get cut due to rotation. Image size will expand to fit whole image.
    5. Rotated image is resized using *resize\_l* to get proportion according background image
    6. Translucent mask such that background of threat object becomes transparent and threat object itself becomes translucent id found using *translucent(image\_object)* function
    7. Finally, Threat object is pasted on background image with top left corner of the threat object is at *(random\_x,random\_y)* so that it does not go outside the background object. Translucent mask is used to give translucent effect and to remove background of threat object.

**Future changes:**

* Object oriented programming concepts can be added
* Background class and threat class
* Overloading of boundingBox() function
* Better method for translucent mask