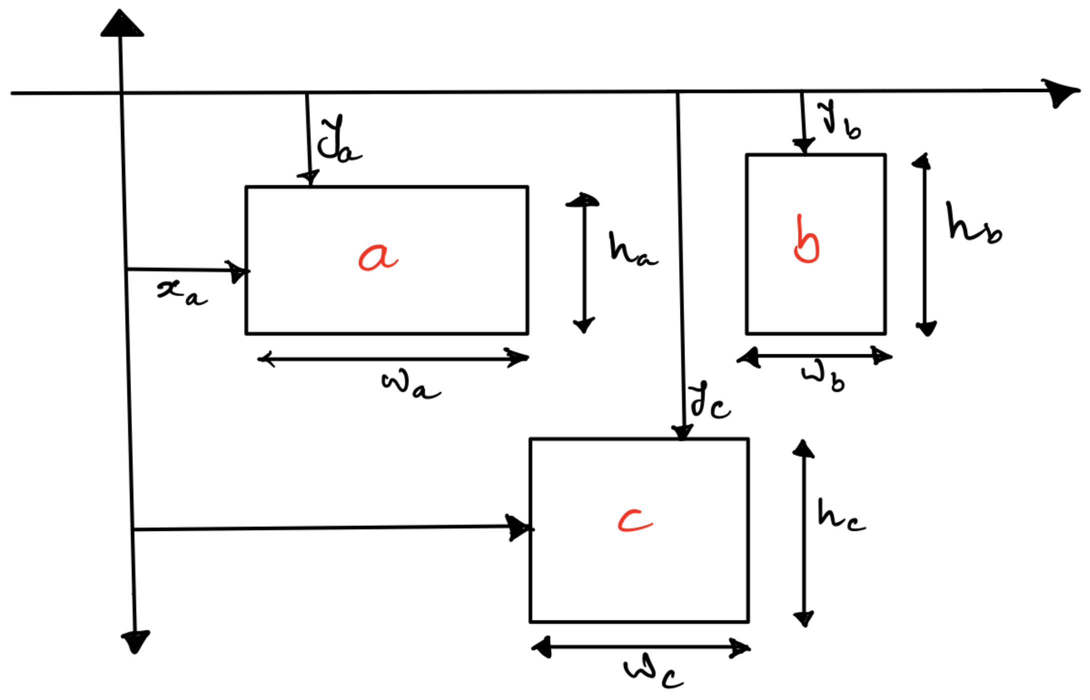
Object Coordinate Extraction

Goal: Detect relative object coordinates based on the object frequencies in each row.

- According to the position of bounding bounding boxes group them in their respective rows.
- Count object (bounding box) frequencies in each rows and assign row number and object number to each bounding box.

General Idea: Idea is to group objects in a way such that objects with similar vertical distance from the origin groups together. As shown in the image below the vertical distance is measured by ‘y’ value. We have annotations in the format of (x_min, y_min, x_max, y_max) which is also referred as a Pascal VOC challenge annotations format. y_min is the vertical distance of an object from the origin. These grouped objects would form a row, and we order these objects from a single row according to the x_min, which is horizontal distance from the origin. Next step is to assign a count to these row wise sorted objects in increasing order (Which will act as a bounding box number in that row.)



Grouping Formula:

$$round\left(\frac{y}{threshold}\right)$$
$$threshold = maximum\_height * threshold\_constant$$

By dividing each object’s vertical distance from origin with a threshold value and rounding that off, in a way we are saying that we want to group all those objects in a single row which has height unto to some particular portion of the threshold value. For example the division results 0.1, 0.3, 0.5 would all result in a single group after round operation.

Important thing is to design the appropriate threshold value so that it can address various height difference of objects in a single row.

Here the threshold is taken as **maximum possible height of an object in an image multiplied by threshold constant value** which can be seen as a space between two possible rows (threshold adjusting value), moreover it can use to adjust threshold according to various situations, more like a buffer.

Maximum height of an object has been taken because, maximum possible distance between two shelf would be equal to the maximum possible height of an object.

Code Solution:

To Run: python shelf-coordinate.py --img_dir (absolute path)
--ann_path (absolute path) --output_dir (absolute path)
--threshold_constant (float value, default: 0.97)
-- annotation_type (string value, default: 'true')

Inputs:

```
 —img_dir: Input_Images_Directory
                       the absolute path to images directory
 —ann_path: input_annotations_file (PascalVOC style annotations)
                       the absolute path to annotations file
 —output_dir: Output_Result_dir
                       the absolute path to output results
 —threshold_constant: [threshold constant] desired threshold constant
                    to be multiplied with max height, default 0.97
 —annotations_type: 'which type of annotations you would like to
                    experiment on, true or pred?' Default 'true'
```

Outputs:

```
—Output images folder
—Json annotations file, containing row number and object number as a
last and second last element of each bounding box annotation.
```

Files:

- **shelf-coordinate.py (main file)** -
  - command line arguments parser function.
  - function to create shelf objects coordinate system (program logic)
- **utils.py** -
  - function to convert the pascalVOC style annotations to coco style annotations
  - to save output images.
- **relative_coordinates.py** - (all the functions necessary to extract relative coordinates of objects in a shelf)
  - function to group bounding boxes in appropriate rows.
  - Function to count objects in each row sequentially and assign an object number in that row.
- **drawing_box.py** -
  - To draw the bounding boxes from annotations data and assign the calculated [row, object#] label to each of them.

Code Flow:

- Code flow for build_coordinate_system function.

For each image,

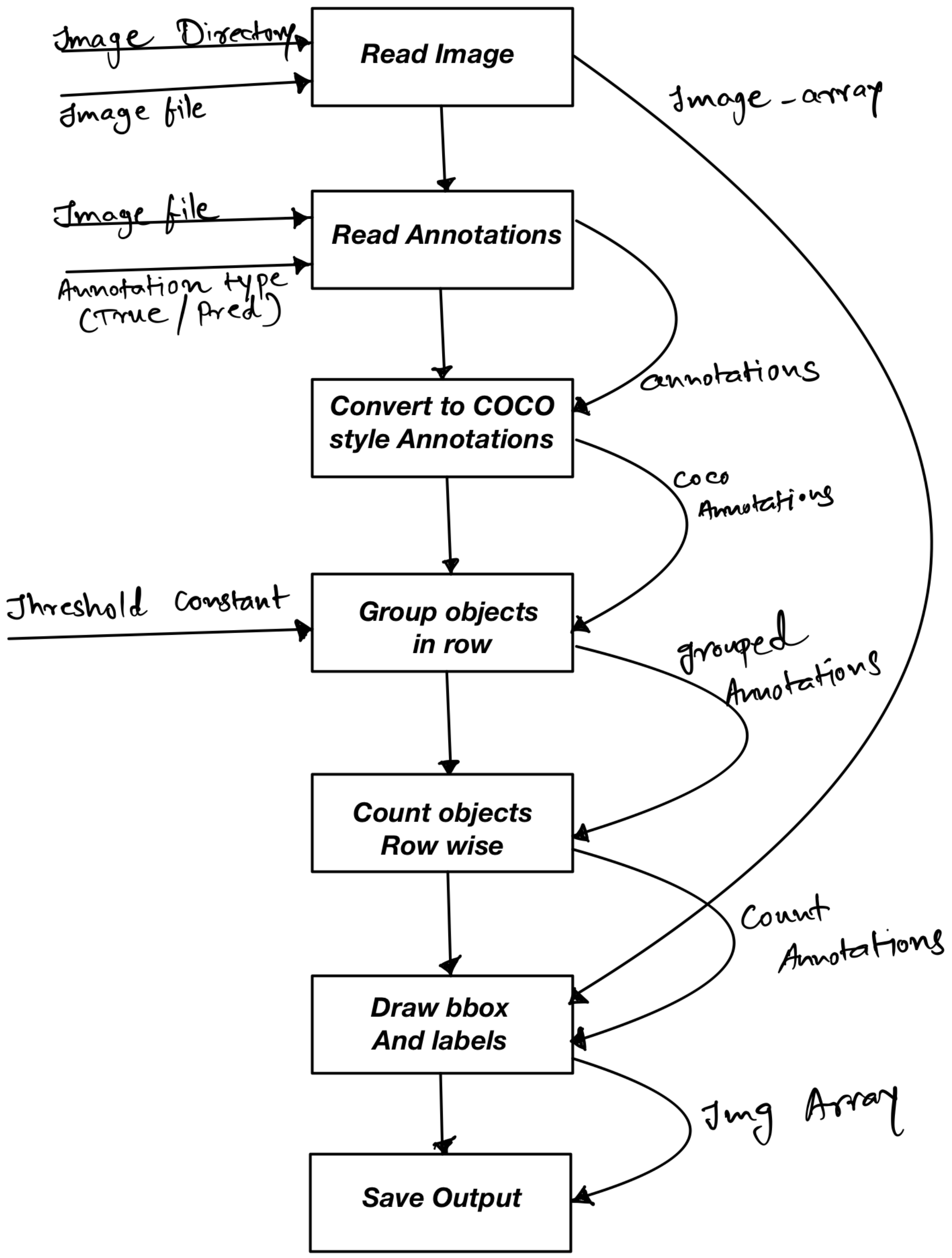


Image Directory
Image file
Read Image
Image-array
Image file
Annotation type (True / Pred)
Read Annotations
annotations
Convert to COCO style Annotations
coco Annotations
Threshold Constant
Group objects in row
grouped Annotations
Count objects Row wise
Count Annotations
Draw bbox And labels
Img Array
Save Output