* Placed Depth camera on Transform (same as RGB camera)
* Synchronised tick rate of the depth camera with the RGB camera
* Every new data will be processed as raw data
* Implemented a formula to calculate the distance out of the value of the BGR pixel array
* Get RGB Image with coordinates/pixels of recognized transforms by YOLO
* Cut out the frame of the transform out of the RBG image
* Edited the RGB frame with an adaptive threshold filter to find abnormal changes in colour gradients
* The filter will change every pixel to black (0) or white (255)
* Transforms in given frame should be coloured black
* Recognize every black pixel and calculate distance of this specific pixel in the depth image
* Calculate length of the transform out of results
* Calculate the reference point of the transform
* To get the middle of the transform the half of the calculated length combined with the angel of the transform will be added to the distance
* Calculate the middle of the object to get useful results for comparison
* Implemented a tracker based on the same approach as YOLO and changed the code to function in our script
* Send values of every bounding box to the tracker while filtering obvious wrong detected objects
* Detect the centroid of every bounding box
* Calculate the distance of every centroid to all other centroid points in the next frame, select the version with the least movement
* Return an ID for every bounding box/centroid