



Computer Generated Training Data

Jashua Luna

Accurately segmented, labeled data takes a long time to produce. For certain use-cases, it may be sufficient to train the bulk of a network on computer generated images, then finish training on a smaller set of real-world images. Computer generated images can be automatically labeled and segmented so large training sets are easy to obtain. In Blender, a series of random rooms with different furniture arrangements and lighting conditions were rendered on a frame-by-frame basis and metadata saved to txt files. Object vertices (three dimensional coordinates relative object origin) were projected onto the camera lens before saving to text file. Projected coordinates were consolidated into just the vertices which described the object in the image; vertices not relevant to the object's silhouette in the image plane were discarded. Parallax distortion was a major factor here and some object silhouettes do not perfectly align with the object in the image-plane. Silhouettes were exported to binary pngs on a per-object basis. Several scripts were written to convert binary-pngs to different annotation formats. For example: COCO_V3 JSON. Several network models were trained on this data, however technical limitations resulted in inconclusive results. Training and image-segmentation model like MaskRCNN was infeasible due to time-constraints. Decent results were obtained when training FastRCNN; data was converted to bounding boxes from the png silhouettes. In summary, computer-generated data seems like a viable aggregate to real-world training-data.
