Keypoint RCNN is an extension from traditional mask RCNN, due to the latter’s generality that allows it to be adapted to human pose estimation tasks. In order to explore its performance more holistically, we experiment with keypoint RCNN by applying it to CCTV footage dataset, which features a low resolution image that is not the norm for keypoint detection.

First, we begin by exploring the architecture of keypoint RCNN, which simply differs from mask RCNN in the last stage. We have previously understood mask RCNN and explained its architecture. Specifically, in its last layer, ROIs are fed into the box prediction and mask branch. On the other hand, keypoint RCNN removes the mask branch and introduces the keypoint branch instead, which produces a one-hot keypoint mask. This means that each keypoint is represented by a single pixel. Moreover, most keypoint RCNN runs a keypoint predict on ROI that has the class label determined as the person class, to gain higher accuracy. Another potential difference is that the keypoint branch could benefit from incorporating a class-agnostic prediction since we are merely focusing on getting the keypoint on human instances, which will save computation time.

In our experiment, we imported the pytorch built-in keypoint RCNN. This model is pre-trained so no training is necessary. Plus, the dearth of data, merely 700 CCTV footage, is not enough to train custom keypoint RCNN with great accuracy. The provided keypoint RCNN here has ResNet 50 as its backbone. The prediction model returns the corresponding bounding box, its confidence score, and the keypoint within that bounding box. In the code, we excluded detection results with a confidence level lower than 0.6 and we only ran predictions on the first 50 CCTV footage.

The result of the prediction is then visualized by drawing the label on the original image. Though the dataset provided ground truth annotations for reference, it is computationally expensive to calculate accuracy by comparing your predictions. Since we need to check the distance between keypoint to determine if it is being accurately categorized, this means we will have a loop comparing all these key points. The problem is that these key point coordinates are not aligned so we have no idea which set of points to compare to. Thus, we resort to a more qualitative evaluation of the result.

On average, the keypoint RCNN performed well, with most person instances correctly identified. However, due to the low resolution quality of these images, a few of them have misclassified keypoints. However, this is mostly due to the box prediction error since bounding boxes were applied to the area of background. Here is a few example:



In some other cases, people are not even detected:





Another caveat is that these keypoint RCNN models do not group associative keypoint together. In other words, it will be hard to connect key points together to reflect arms, legs, etc., because we do not know which points are in the same group. Plus, the keypoint detection is mainly 2-dimensional, meaning it will be harder to understand the pose through certain angles, such as sideways. If resources allow, 3-dimensional custom keypoint detection will be preferred.