**Input**:

1. Detectron2: In this problem statement we capture the open-source COCO image dataset. We initialize the instance annotations using COCO\_train\_annos.json and COCO\_mul\_train\_annos.json. We receive the different categories and super-categories images in the form of damage parts. E.g., headlamp, rear\_bumper, door, hood, front\_bumper. They are then mapped using dictionary for category id and category name.
2. MaskRCNN: We specify image\_id, image, mask, class\_id, bbox. There are 2 classes 1)BG (Background), 2) Damage. We also run via\_region\_data.json file to get the details of each image we want to train for detecting damages.

**Output**:

1. Detectron2: We get the plots for the annotations on different damaged parts in the form of bounding boxes and segmentation. When we run the detectron2 libraries for training the model, we receive results like total number of damage instances, Average Precision, Average Recall, IOU score for each prediction, class\_accuracy, total\_loss. We also run the validation images to get the desired results of the damages in the form of IOU scores above the bounding box.
2. MaskRCNN: We receive the results of semantic segmentation for the damaged parts separating foreground and background along with the bounding box for annotations. Visualization in the form of histogram for the model weight matrix is carried out in the form of descriptive statistics. We also run the training and validation images to get the desired results of the scratches in the form of IOU scores above the bounding box.

**Comparative Results**:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Models Used for Car Parts Damage Detection** | **Kinds of Detection** | **Separating damaged part from the image** | **Segmentation Result** | **Bounding Box Result** | **Average Predicted IOU Score** |
| 1. | Detectron2 | Damage | Not Supported | 100% | 100% | 94.46% |
| 2. | MaskRCNN | Damage, Scratch | 100% | 100% | 100% | 0.9448 |

**Camera Mount**: