Task-5.2 HD Report on Mask R-CNN Technique

1. Introduction:

In recent years deep learning techniques have improved the accuracy of object detection, semantic detection and human pose estimation in computer vision. These advances were possible due to the conceptually intuitive, robust, and fast methods like Fast/Faster RCNN and Fully Convolutional Network (FCN). Of all these methods Mask R-CNN is the only high-performance algorithm for the instance segmentation i.e. it can detect all the objects in an image or video in least amount of time.

This report summarizes the research work published in [1] to develop a framework for the instance segmentation. In this report, initially basics of object detection and image segmentation are described, followed by the architecture of previously developed frameworks and their challenges. This report provide overview of Mask-RCNN architecture, results from its implementation and suggestions on further improvements.

2. Object Detection and Segmentation:

Object detection task involves identifying the presence, location, and type of one or more objects in an image or video. Based on these attributes a bounding box is drawn around that object. But this method does not provide the information about the true shape of object. Image segmentation technique creates a pixel-wise mask for each object in the image, through which we get the granular information of the object. It involves dividing a visual input into objects or parts of objects to simplify image analysis. Image segmentation is further classified into two categories; sematic segmentation and instance segmentation. Semantic segmentation classifies all the pixels of an image into semantically meaningful classes of objects. These classes correspond to the real-world objects. On the other hand, instance segmentation identifies each object instance for every know object within the image by labelling each pixel of the image. For example, in an image of multiple cars sematic segmentation will classify all cars as one instance while instance segmentation will identify each individual car. Instance segmentation technique requires detection of all objects in the image and segmentation of each instance. Instance segmentation has high importance in the self-driving cars, where this technique helps classify all the cars and person in the vicinity.

3. Image Segmentation Techniques:

There are various classic image segmentation techniques which could be used to classify the objects in the image. Thresholding technique is region-based image segmentation algorithm. It divides the image into foreground and background based on the specified threshold value to separate pixels and isolate objects. In the histogram-based image segmentation technique, histogram is used to group the pixels into object and background based on the gray levels value. K-Means clustering algorithm is also used to group the pixels based on the feature similarity. Edge detection technique involves arranging the pixels into curved line segments based on the sharp changes in the brightness.

As these classic methods require human intervention and expertise, they have less accuracy and low performance compared to the deep learning techniques. Deep learning technology has provided many algorithms for image segmentation which use neural network for object classification. In the Convolutional Neural Networks (CNNs) based image segmentation, the CNN feeds the segments of the image to a neural network for labelling the pixels in the image. This technique does not provide good performance for the images with different aspect ratios and spatial locations, as it takes huge amount of computational time for processing these large number of regions. Region-Based Convolutional Neural Network (RCNN) resolves this issue by proposing number of boxes (regions) and checking for object in these regions using selective search. RCNN is a combination of region proposals with CNN model and provides excellent object detection accuracy with deep ConvNet. In RCNN, last layer of CNN model is retrained on the target object classes. Then region of interest (ROI) feature vectors are obtained from CNN and passed to SVM classifier for object detection. RCNN training is a multi-stage pipeline, expensive and slow.

Fast RCNN resolves the drawbacks of the CNN and RCNN techniques by using a single model for feature extraction, object detection and bounding boxes. In this method convolutional feature maps are generated for images using CNN. Region of proposals are extracted from these maps using ConvNet and resized using ROI pooling layer, before feeding it to a fully connected network. A softmax layer and liner regression layer is used to output the bounding boxes for the predicted object classes. As Fast RCNN also uses the selective search for ROI, it is slow and time consuming on the real-world image datasets. "Faster R-CNN object detection algorithm utilizes the Region Proposal Network (RPN) that shares full-image convolutional features with the detection network in a cost-effective manner than R-CNN and Fast R-CNN."[2] RPN is a fully convolutional network used to predict the object bounds and high-quality region proposals. These region proposals are passed through detection network to generate bounding boxes. RPN utilizes the feature maps to generate the region proposals. Feature maps are generated using the ConvNet from an image. Like the Fast RCNN, this technique also uses ROI pooling layer for re-sizing, softmax layer and linear regression layer for object classification and bounding box. As the Faster RCNN process all the regions of image sequentially it takes multiple passes to extract all the objects from the image. Region-Based Fully Convolution Networks (R-FCN) have convolutional layers to process varying input sizes in small amount of time. R-FCN like the FCN is used to classify all the pixels of the image to determine context and location of the objects in short time as it shares computation across the entire image.

4. Mask R-CNN:

Mask R-CNN is successor of Faster R-CNN technique, with a two-stage framework. In the first stage object proposals are generated, while in the second proposals are classified to generate bounding boxes and masks. In the Mask-RCNN technique detection (classification and bounding box) task and masking task executes in parallel to each other. Unlike Faster R-CNN, Mask R-CNN uses FPN, have replaced ROI Pooling layer with ROI Align and have additional branch to generate the segmentation masks on each Region of Interest (RoI). Mask R-CNN uses feature pyramid networks as backbone to generate the ROI features, which has increased the object detection accuracy. As ROI Pool involved the coarse quantization steps, it was replaced with the ROI Align in the Mask R-CNN for generating the masks. [1]

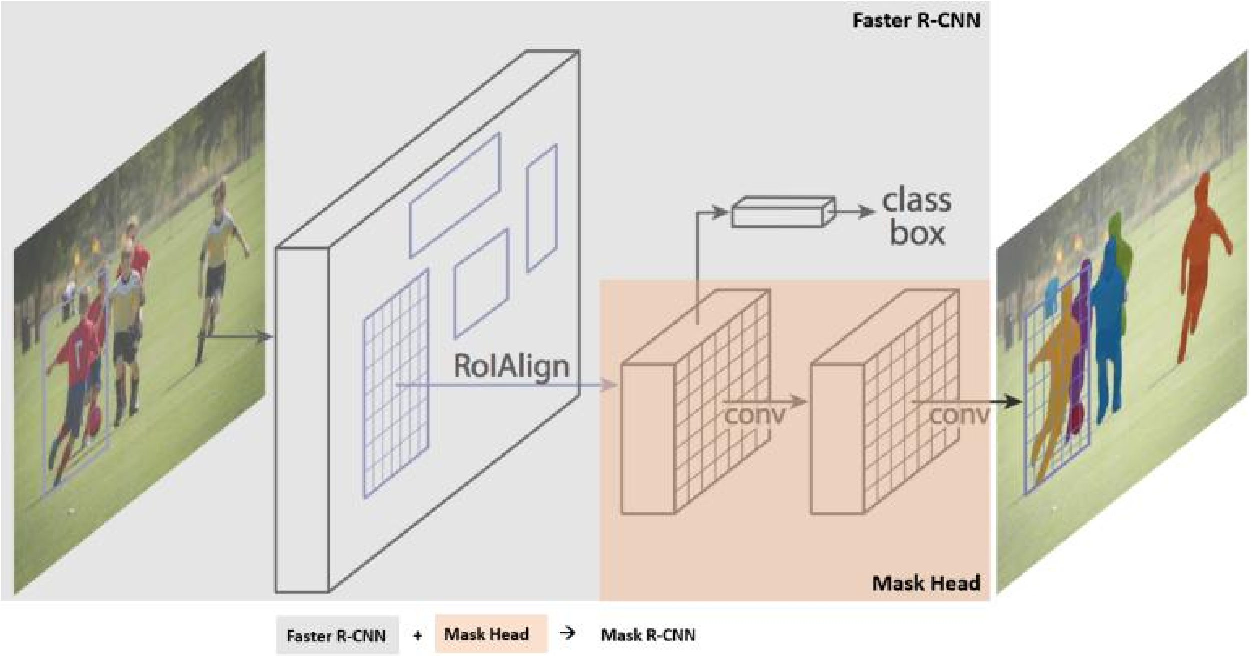

Fig 1. Mask R-CNN architecture image from [1].

4.1 Feature Pyramid Network (FPN)

In the paper [1], researcher used Feature Pyramid Network (FPN) with ResNet architecture of the Faster-RCNN in the Mask R-CNN to extract the RoI features from different levels of the feature pyramid according to their scale. "ResNet-FPN backbone for feature extraction with Mask RCNN gives excellent gains in both accuracy and speed" [1]. Better quality information could be extracted from the multiple feature layers generated by the FPN compared to the regular pyramid networks. "FPN provides a top-down pathway to construct higher resolution layers from a semantic rich layer.” [3] FPN is a feature extractor and is used with the object detector like RPN for object detection and classification. To help the detector predict the location of object in the image, lateral connections are added between the pyramid layer and feature map. This also help reduce the training time. In the top-down pathway 1x1 convolutional filter is applied to obtain the first feature layer. This layer is up sampled by 2 nearest neighbors up sampling. Again, 1x1 convolution is applied with element wise addition. A 3x3 convolution is applied on all the merged layers.

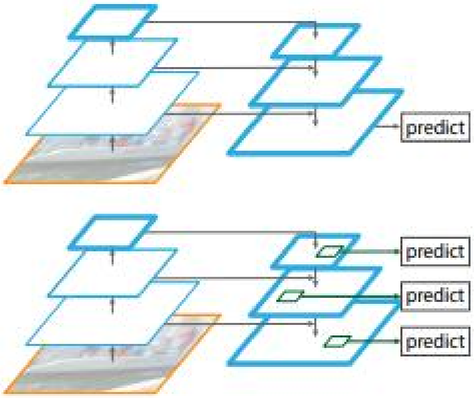

Fig 2. Image showing the top-down architecture of FPN with skip connections [3]

4.2 Region Proposals

For generating the region proposals, Mask R-CNN selects anchors based on their objectness score. These anchor boxes coordinates are modified according to the values obtained from the RPN head. After the coordinates are modified, invalid bounding boxes whose coordinates lie outside the image are removed from the proposal. After this non maximum suppression is applied on these bounding boxes to remove the boxes having overlap more than the specified threshold. After this all the anchor boxes are grouped together from different layers of the FPN.

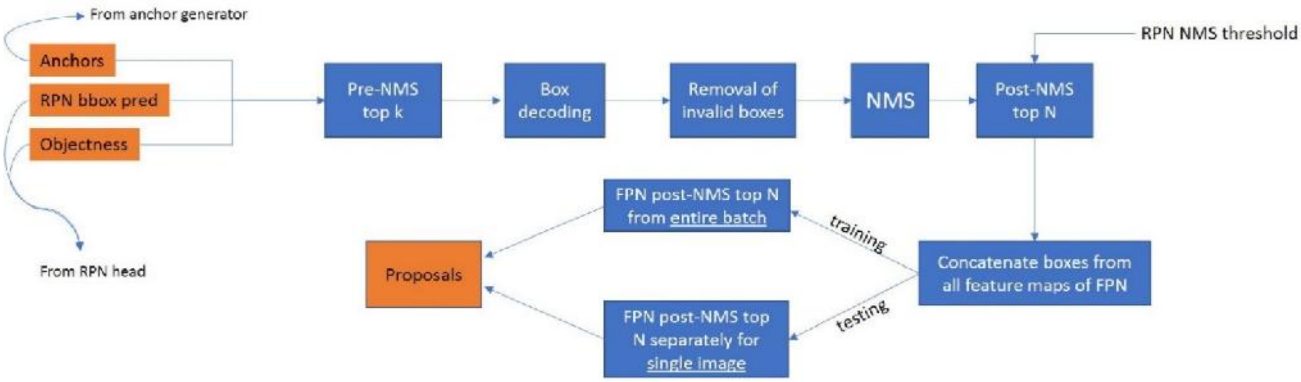

Fig 3. Above image shows the process of generating region proposals. [Source]

4.3 ROI Align and Bounding Boxes

After the features are obtained from the FPN, they are mapped to the ROI based on its area. Bounding box with coordinates based on the original image size can be obtained from the RPN proposals. To get the new coordinates relative to the feature map size, ROI pooling divides each coordinate by K (relative size of feature with original image) and take integer part. Based on these new coordinates parts from the image are cropped. For getting the fixed size output from the ROI pooling, quantization process is used. In this process, cropped parts based on new coordinates are divided into bins (N x N grid). From these grid max and average values are taken. Due to the large number of quantization steps ROI Pool fails in instance segmentation as it impacts the generations of masks for the image. Because of this reason ROI Pool is replaced with the ROI Align in the Mask R-CNN. In ROI align each coordinate is also divided by the K, but it does not take integer part i.e. all new coordinates are float values. This process also divides the cropped parts into bins, but for defining concrete values in these bins' ROI algin chooses regularly 4 points in each bin using bilinear interpolation. Max pooling is done on these 4 points to obtain the result.

Both ROI Pool and ROI Align create same PxP matrix for the features. The output of each ROI is reshaped before passing through a fully connected layer. This generates same length of vectors for different ROI. These ROI vectors of same length are passed through 2 branches of fully connected layers. One branch predicts the object class and the other branch bounding box coordinates for the object. Regression model predicts many boxes (anchors) to predict multiple objects in the image. Anchor boxes are selected based on the confidence score IoU. (IoU) computes Intersection over Union for ground truth box and predicted box. Non max suppression (NMS) removes boxes with IoU score less than threshold value. Threshold for the detecting the class of the object and limit on the number of classes for object could also be specified as parameter to these layers.

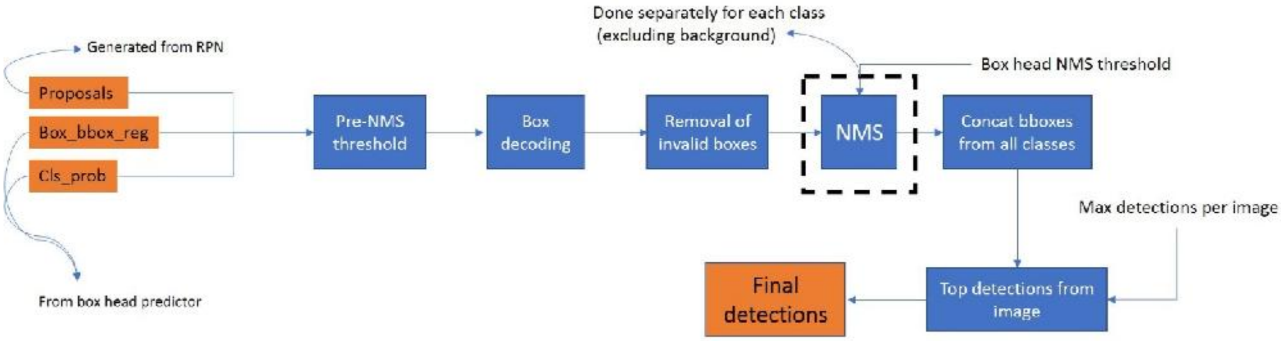

Fig 4. Above image shows the process of generating final bounding boxes from the 2 fully connected layers [Source]

4.4 Mask generation

In the process of feature extraction of masks, fully connected layers are not used as the spatial structure information is lost in these layers while generating the masks. For feature extraction of masks bounding boxes are provided as input to the feature extractor. These ROI align output are passed through the convolutional layer followed by the ReLU which produces tensor of same dimension as ROI align output. On these extracted features deconv operation is performed inside the predictor. Predictor is a 1x1 convolutional layer, with the number of out channels being equal to the number of classes i.e. one mask for each class detected in the image. The masks obtained from the predictor could be resized according to the image size with bi-linear interpolation. Bounding boxes are also re-sized according to the masks obtained. Based on the mask threshold final masks coordinates are obtained for all the objects detected in the image.

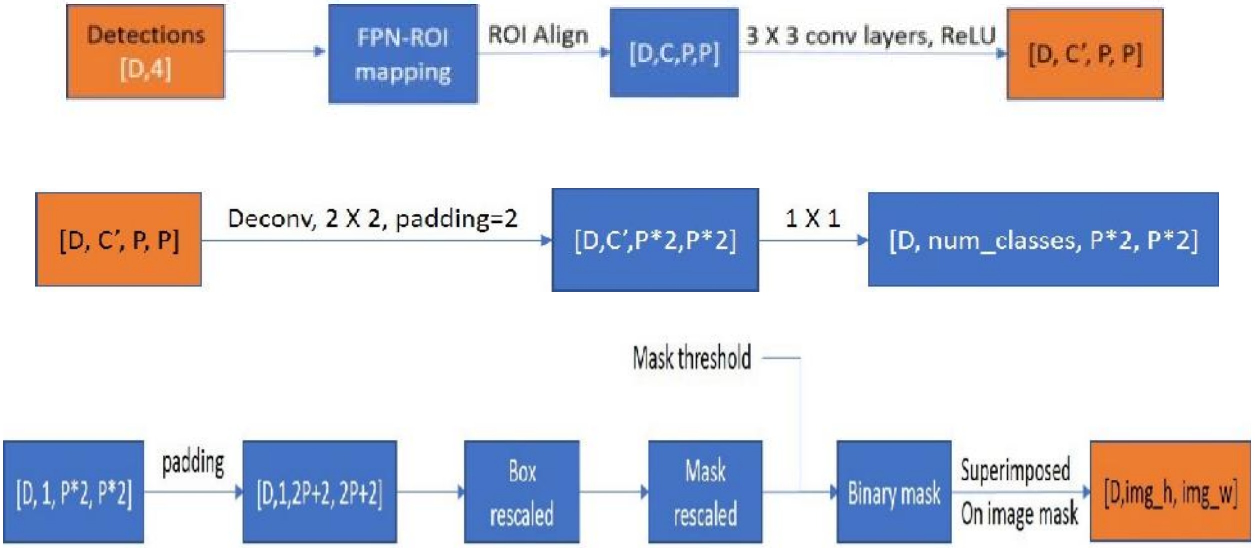

Fig 5. Above images show the process of generating the masks for objects detected in the input image. D denotes the number of detections from mask head, C Number of feature layer channels, P for pooler resolution and C` denotes the out channels of feature extractor. [Source]

4.5 Loss Function

Loss function for the Mask R-CNN is given in the below equation. Lcls denotes the loss generated in the classification of the objects, Lbox denotes the loss for bounding box regressor and Lmask denotes the average loss mask generation. Lcls and Lbox are same as the Faster R-CNN. Lmask is defined as the average binary cross entropy loss obtained after applying per pixel sigmoid on the masks obtained for each ROI. Lmask is defined only for the ground truth class, other class does not contribute to this loss. This loss function decouples the mask and class prediction, due to which network generate masks for each class without competition.

$$L = L_{cls} + L_{box} + L_{mask}$$

Eq 1. Loss function of Mask R-CNN

5. Mask R-CNN for Human Pose Estimation:

“Human pose estimation refers to the process of inferring poses in an image.” [4] This method is used to localize the human joints in an image or video. The goal of human pose estimation to extract the positions of the joints (key points) from the image or video, based on which movement of a human can be tracked. As Mask R-CNN is flexible technique, it could be easily modified for the human pose estimation. Like the segmentation problem, key points for every person in image could be extracting by modelling each key point as a distinct class. For the pose estimation key point’s location is modelled as one-hot mask. Mask R-CNN then predicts K masks for each of the key points.

6. Implementation Results:

Below are the some of the visual results obtained from the Mask R-CNN instance segmentation technique. For these implementation results code from public GitHub repository was re-utilized which is modified on initial implementation to work on latest version of TensorFlow. Weights from the model trained on the COCO image dataset were used for image classification. The test images for this task are mixture of image provided in the GitHub repository and image collected over the internet. These set of test images includes images of different size, color and resolution. Images also have different number and type of objects. Code for this implementation is available on Google Colab notebook.

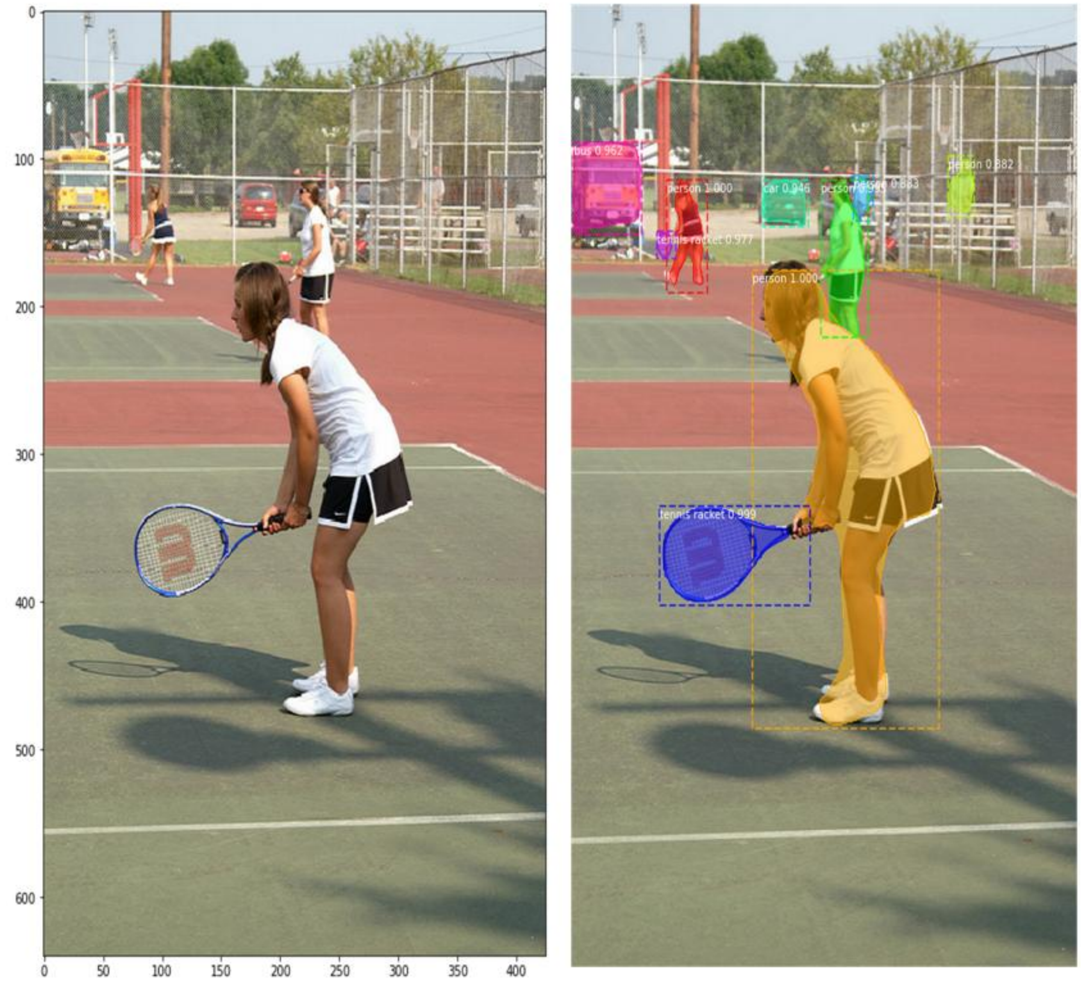

Fig 6. In the above figures, left image is the original image and right image has object type with confidence, bounding box and mask from Mask R-CNN. For this image 4 ROI's were generated, 8 classes (Person, Car, Bus, Tennis Racket, Ball) were detected, 8 masks were generated with confidence scores [0.99987245, 0.9995534, 0.99899143, 0.99464536, 0.977487, 0.9616527, 0.9455735, 0.88254076, 0.882192].

image	shape: (640, 425, 3)	min: 0.00000	max: 255.00000	uint8
molded_images	shape: (1, 1024, 1024, 3)	min: -123.70000	max: 151.10000	float64
image_metas	shape: (1, 93)	min: 0.00000	max: 1024.00000	float64
anchors	shape: (1, 261888, 4)	min: -0.35390	max: 1.29134	float32

Table 1. In the above tabular data, we could observe the original shape of the input image, min/max value of pixels and type of the data. Table also shows the number of anchors used to generate the masks and share of the feature vector.

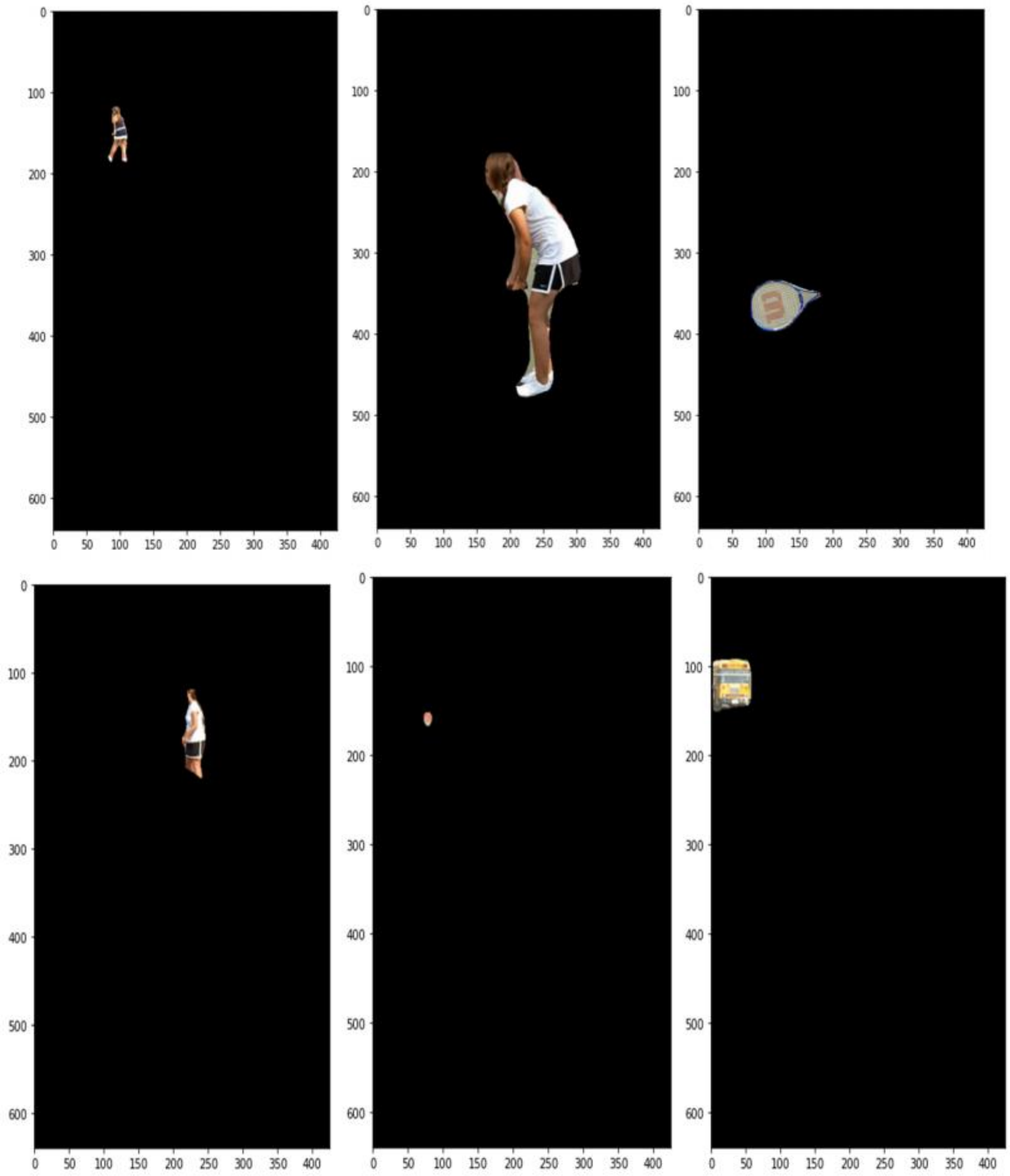

Fig 7. Above images show the individual objects detected in the input image provided to the Mask R-CNN

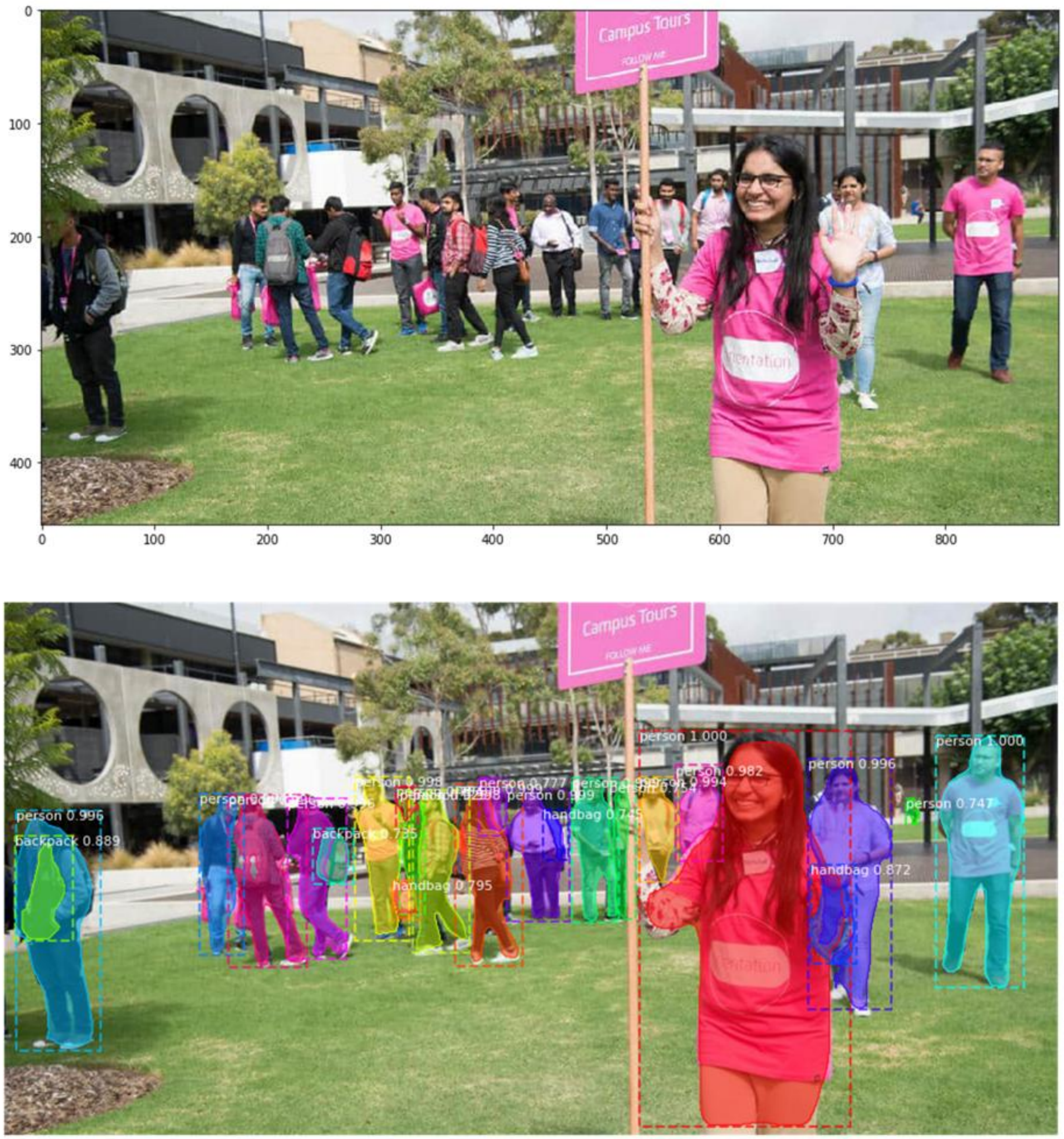

Fig 8. In the above results for instance segmentation, we could observe that the Mask R-CNN was able to correctly classify most of the objects in the image. But it incorrectly classified the hand of girl in pink t-shirt as the handbag with confidence of 0.872. As we had restricted the classes for object detection, other objects like trees, board were not detected in the image. In this image 21 masks were generated for each class (Person, Handbag, Backpack) detected by the Mask RCNN.

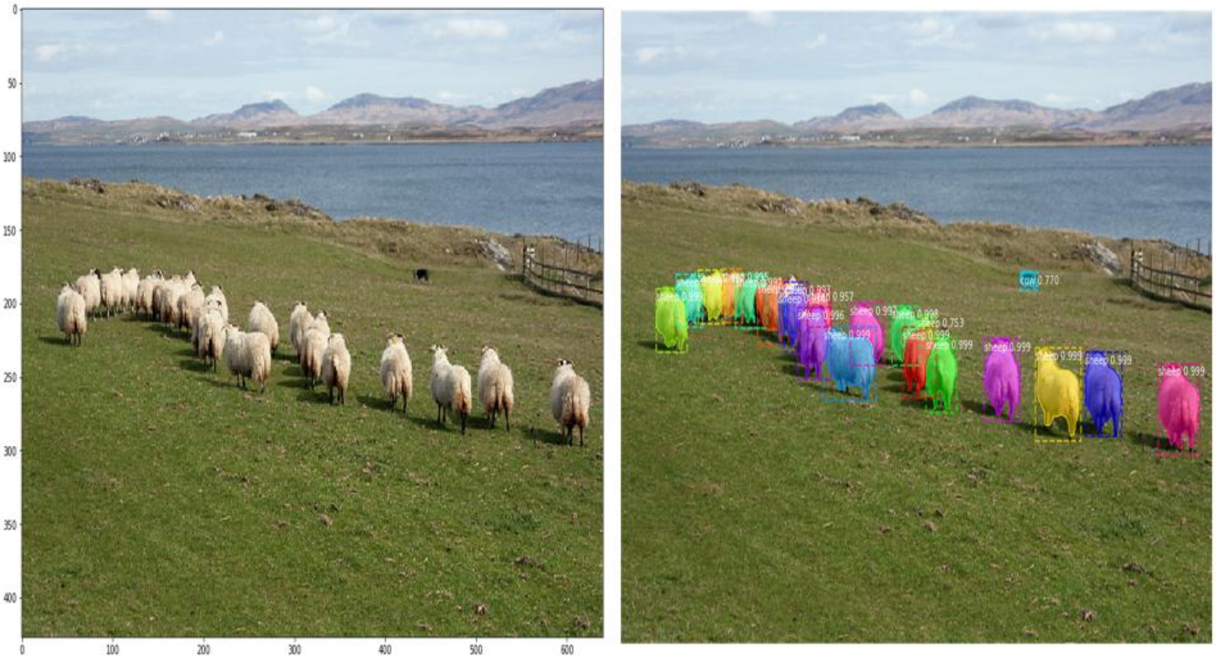

Fig 9. For this image 21 classes were detected, and 21 masks were generated with confidence score [0.9994179 0.9989911 0.99892175 0.9988507 0.99884284 0.9987815 0.99873704 0.99863535 0.99854773 0.99779177 0.99747974 0.99718696 0.9969003 0.99637526 0.9952414 0.99460196 0.99341124 0.9568936 0.93427294 0.7699169 0.7531292]

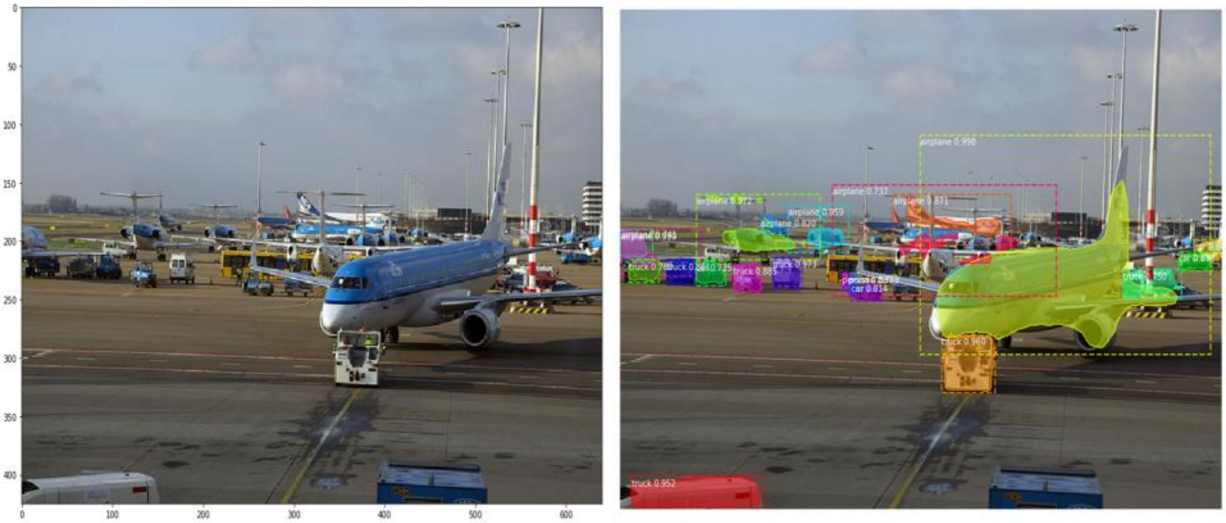

Fig 10. For this image 21 classes were detected, and 21 masks were generated with max score of 0.99

We could observe that in first image all the sheep classified by the Mask R-CNN have same confidence score. While in the second image multiple objects like Airplane, Trucks were classified with bounding box and mask with different scores.

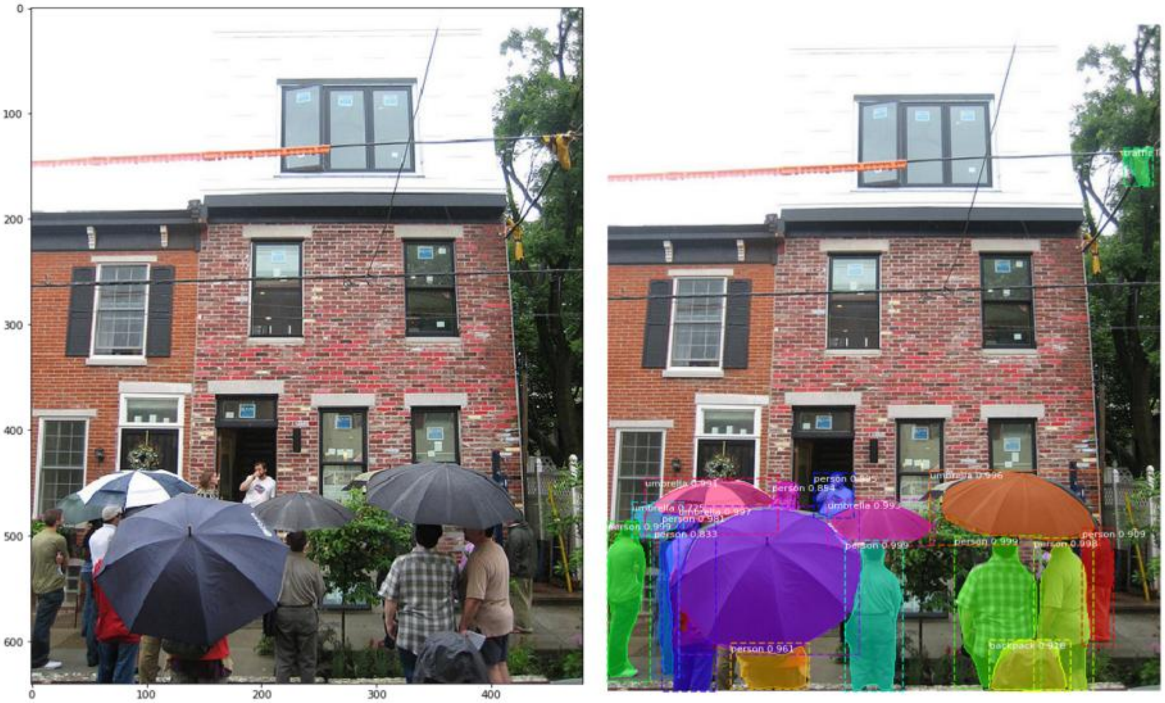

Fig 11. For the above image 17 classes and 17 masks were generated.

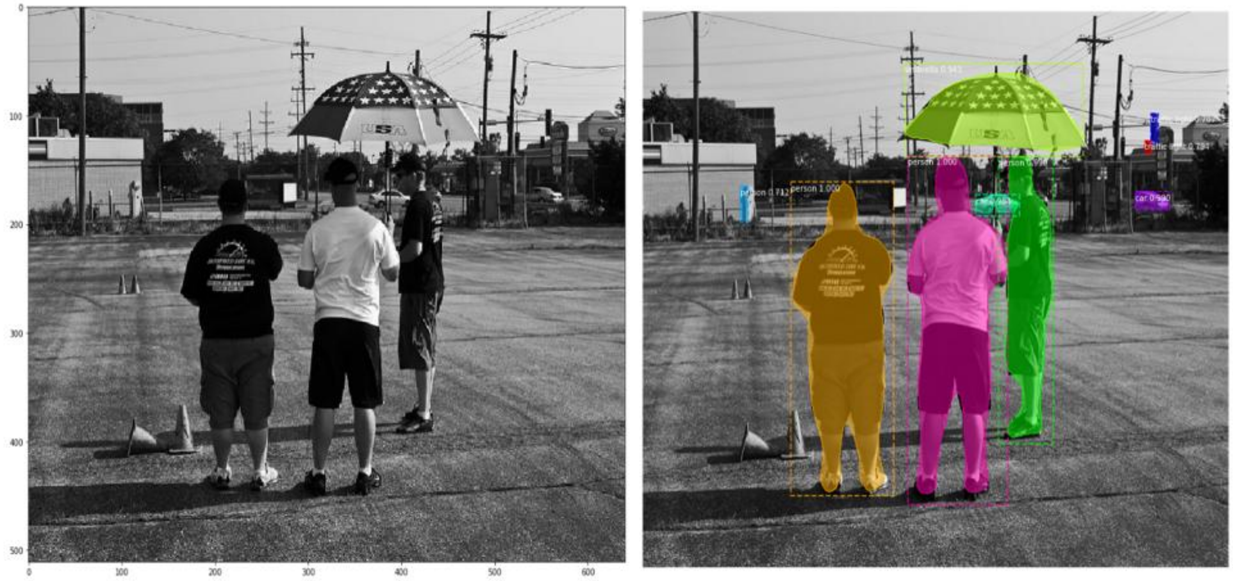

Fig 12. For this image 9 classes were detected, and 9 masks were generated (Person, Umbrella, Car, Traffic Light) with max confidence score of 0.99

In the above images we could observe that Mask R-CNN is correctly classifying images with multiple object overlapping (first image) and image in a grayscale.

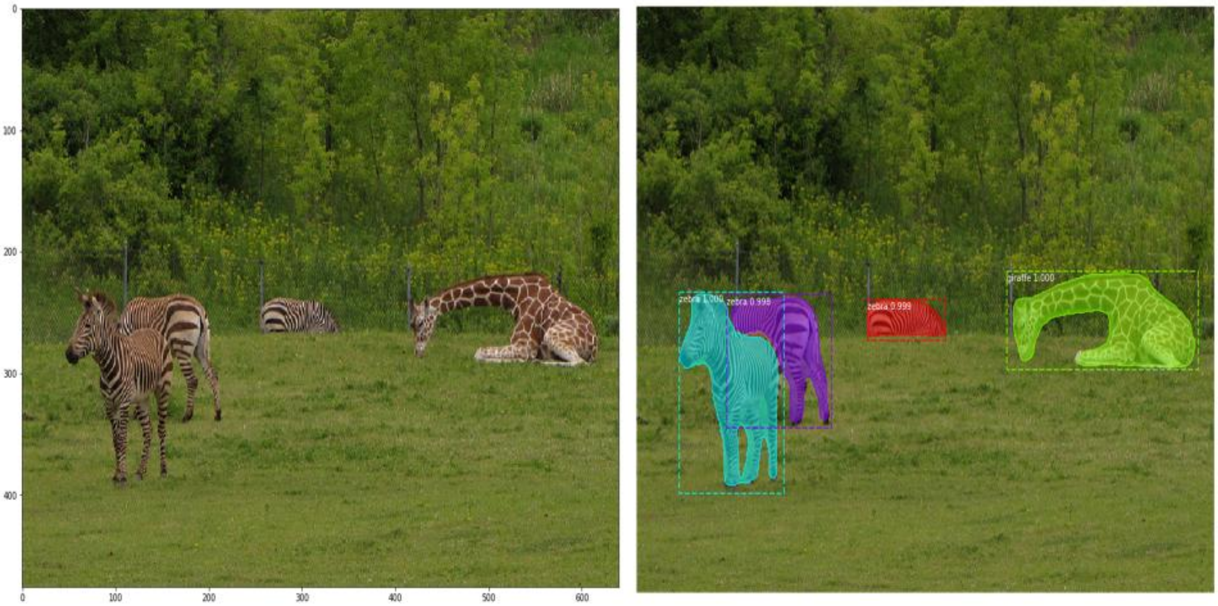

Fig 13. For the above image 4 classes were detected with the confidence score [0.9996276 0.9996008 0.99933416 0.99845934]

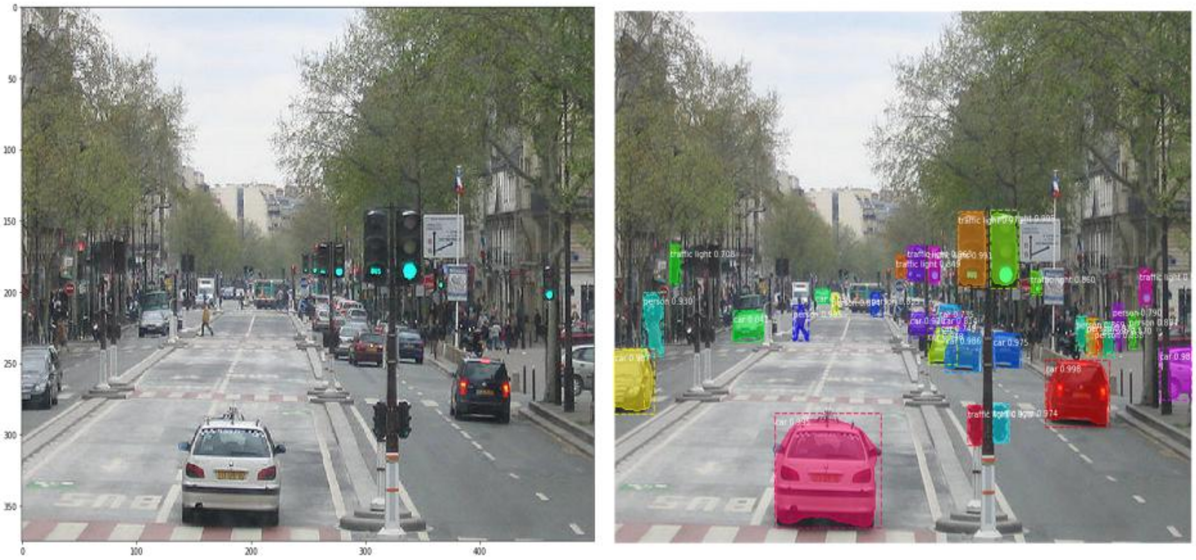

Fig 14. For the above image 33 masks were generated for each class detected by the Mask RCNN.

Below are the visualization results from the implementation of the Human Pose Estimation using Mask R-CNN. For this task, code was reutilized from the public repository with pre-trained weights on the COCO dataset. These weights were utilized for human pose estimation on the test images collected from the internet. Test images set includes images of different sizes, colors, and number of objects.

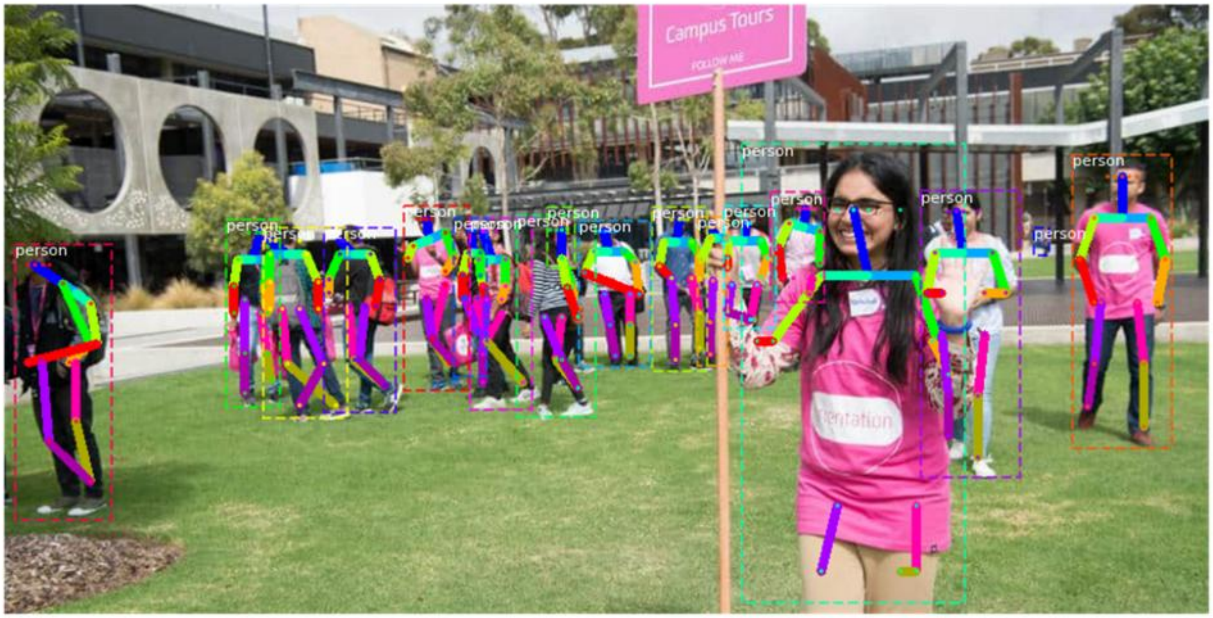

Fig 15. In the above image we could see the pose estimated for the human in the input image using Mask RCNN.

image	shape: (455, 900, 3)	min: 0.00000	max: 255.00000
mrcnn_mask	shape: (1, 100, 28, 28, 2)	min: 0.00002	max: 0.99949
rois	shape: (18, 4)	min: 7.00000	max: 872.00000
keypoints	shape: (18, 17, 3)	min: 0.00000	max: 867.00000
class_ids	shape: (18,)	min: 1.00000	max: 1.00000
masks	shape: (455, 900, 18)	min: 0.00000	max: 1.00000
scores	shape: (18,)	min: 0.77681	max: 0.99962

Table 2. From the above tabular data results, we could observe the original shape of the input image with min and max value for the pixels. In the table we could see that 18 classes were detected by the Mask R-CNN. For each of these 18 classes in the image, 18 keypoints were detected and 18 masks were generated. We could also observe that min confidence score for the object detected was 0.77 and max was 0.99, which is above the default IoU score of 0.5.

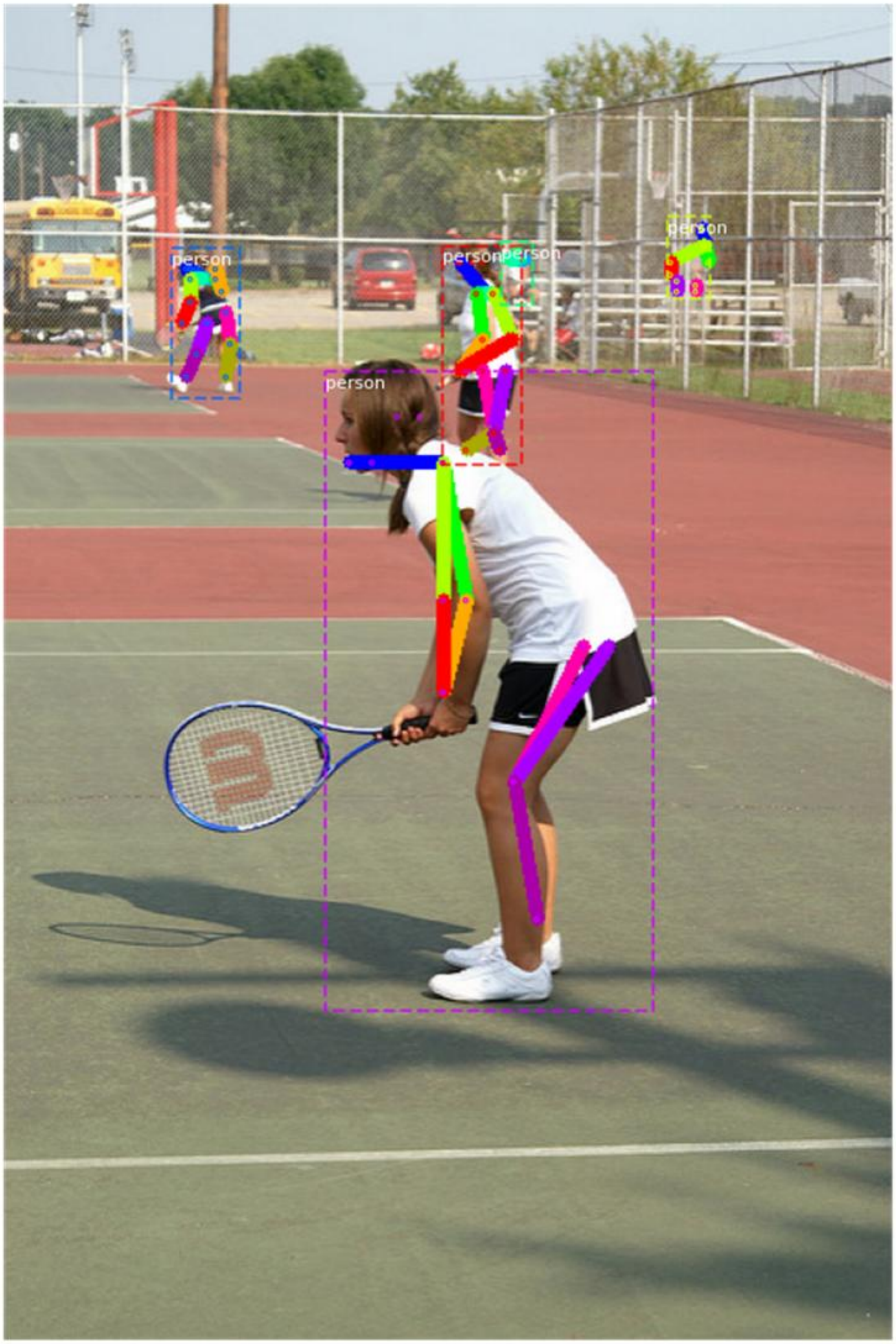

Fig 16. In the above image 5 classes and 3 keypoints were detected. For each of the 5 classes, 5 masks were generated with IoU confidence score between 0.77 – 0.99.

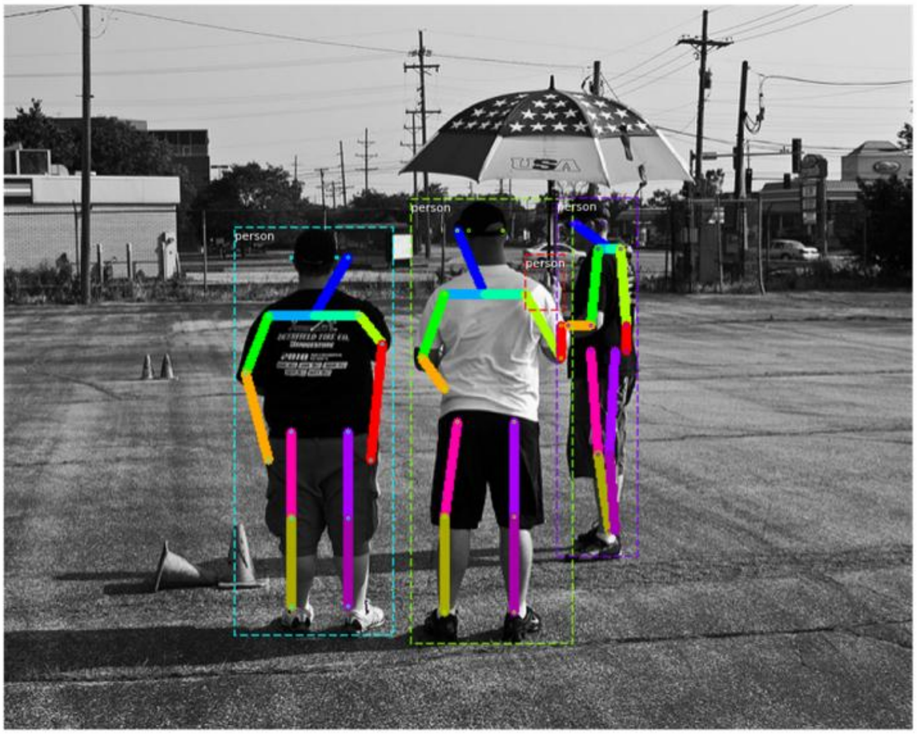

Fig 17. For the above image 4 classes and 3 keypoints were detected.

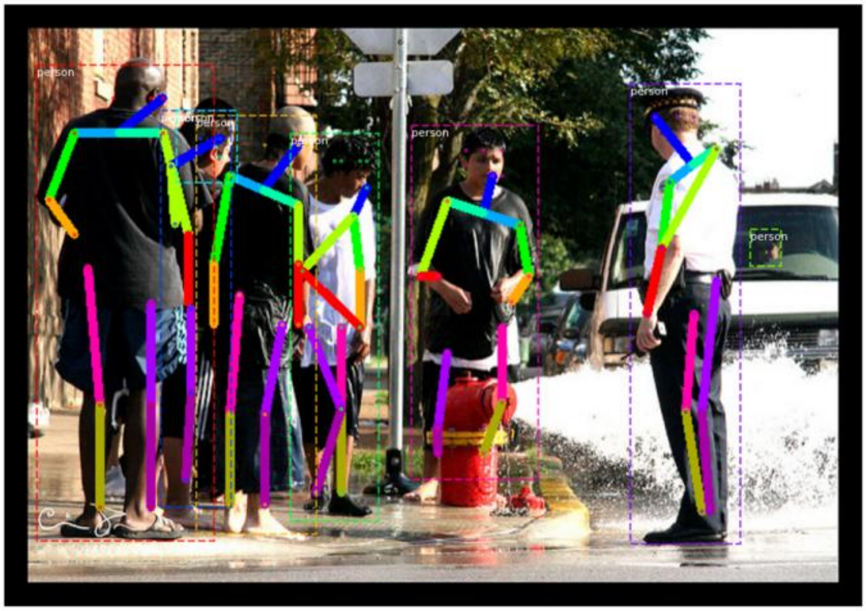

Fig 18. In the above image 8 classes were detected and 8 masks were generated by the Mask RCNN

7. Suggested Improvements:

“It is difficult for the Mask R-CNN method to distinguish different objects with similar features of the shape because it extracts a tight region proposal including only single object without gaps, and uses only texture features within the area in the input image corresponding to the region proposal for learning” [4]. For example, Mask R-CNN cannot strictly distinguish humans’ left and right eyes whose shapes are very similar but have different positions. To solve this issue research paper [4] proposes a method to expand the region proposal output from the Mask-RCN’s RPN. This method will extract the features from both original regional proposal and expanded ones and will use all these features for learning and classification. This method is based on the idea that features from the expanded proposal gives the information about the object surrounding and its relative position.

Also, the Mask-RCNN segmentation algorithm has a high time overhear and a poor accuracy of image segmentation edge. This method uses the background as the foreground at the edge causing excessive segmentation and object edge contouring. Research paper [5] proposes a method to improve the target segmentation by combining the Mask-RCNN algorithm and the Grabcut image segmentation technique which uses the Gaussian model.

8. Conclusion:

In this report, I summarized the architecture of Mask R-CNN and findings presented in the paper [1]. Through this research paper it was observed the Mask R-CNN technique is high-performance and flexible algorithm for the instance segmentation and human pose estimation. Mask R-CNN could be used for the face detection, medical imaging, autonomous vehicles and iris recognition. This report also provided the overview of the other methods used for image segmentation and their drawbacks. In this report, I have provided the results from the code implementation for Mask R-CNN with explanations on a sample image dataset. Finally, in this report improvement suggestion were illustrated which could be implemented to enhance the performance of Mask R-CNN.

9. References:

[1] Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick. Mask R-CNN. In ICCV 2017

[2] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. arXiv:1506.01497, 2016

[3] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie. Feature Pyramid Networks for Object Detection. arXiv: 1612.03144, 20187

[4] Naoki Degawa, Xin Lu, Akio Kimura. A performance improvement of Mask R-CNN using region proposal expansion. In Proc. SPIE 11049, 2019

[5] Xin Wu, Shiguang Wen, Yuan-Ai Xie. Improvement of Mask-RCNN Object Segmentation Algorithm. In book: Intelligent Robotics and Applications (pp.582-591), 2019