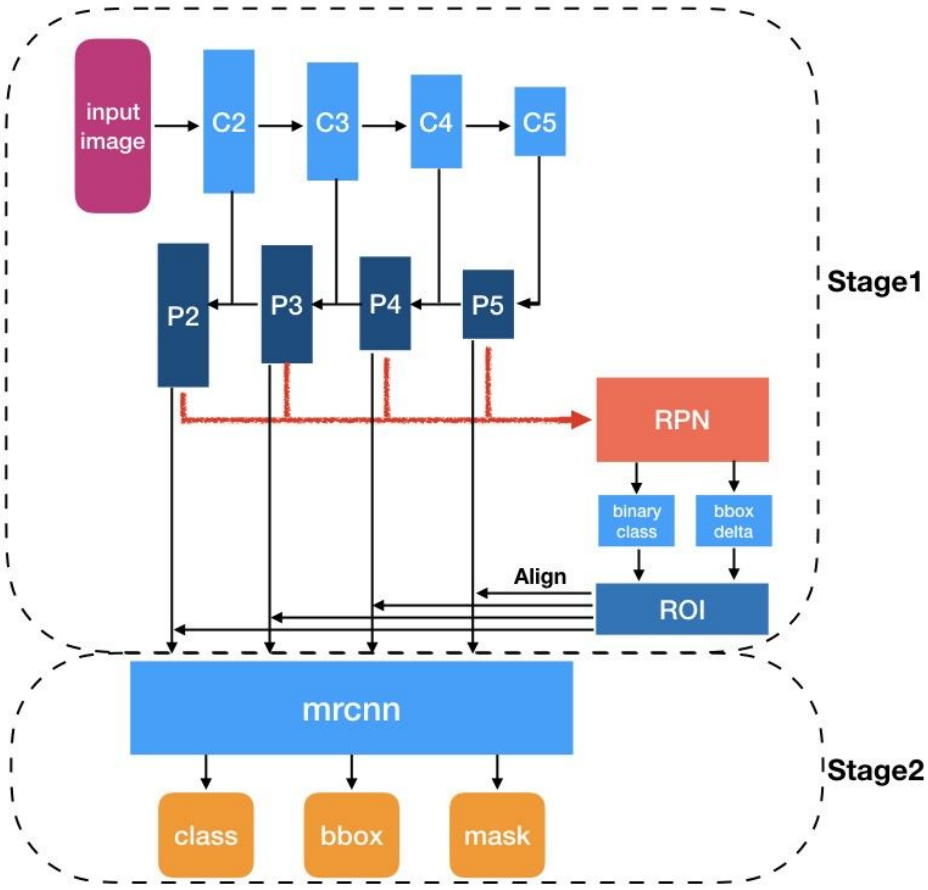
Mask Rcnn

My Understanding of Architecture

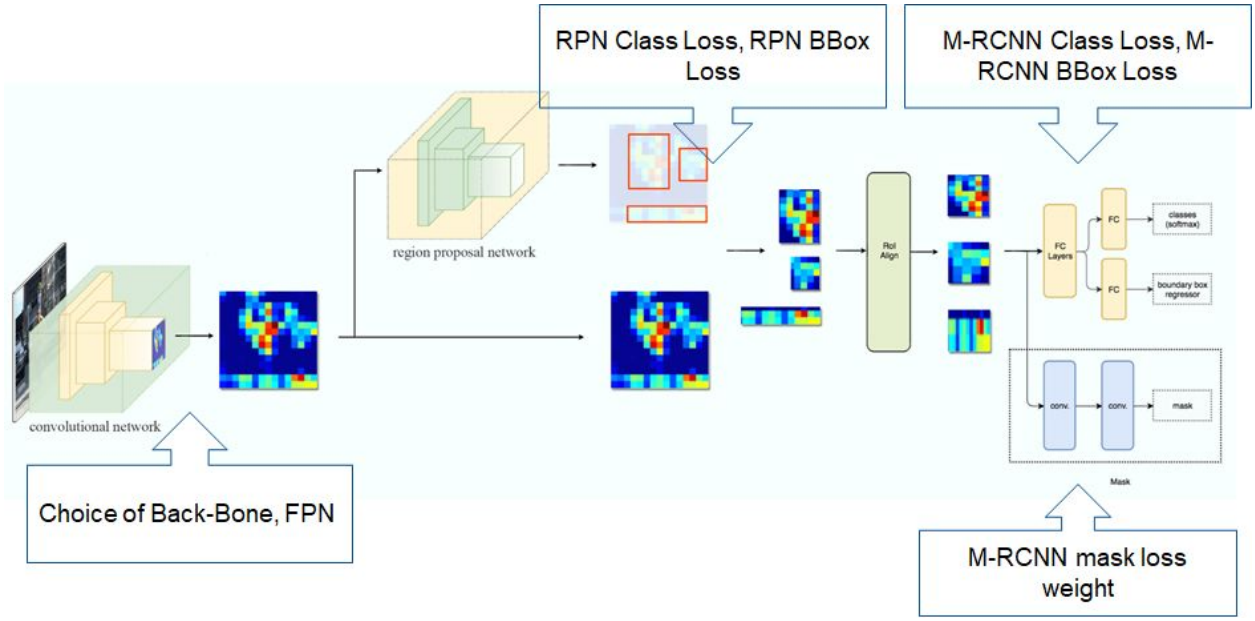
It is a kind of widely used in Instance segmentation.
Mask Rcnn has two stages :

1. In Frist Stage it scans the input Image and Generate proposals in which the area's likely to contain the object or not
2. In second stage it classifies image, Bounding Boxes and Masks of objects

Architecture of MasRcnn:



Credits :https://miro.medium.com/max/1024/1*dYb3w2iVxkN7Ifx-eA8ZRg.jpeg



1. Frist step(Feature Extraction):

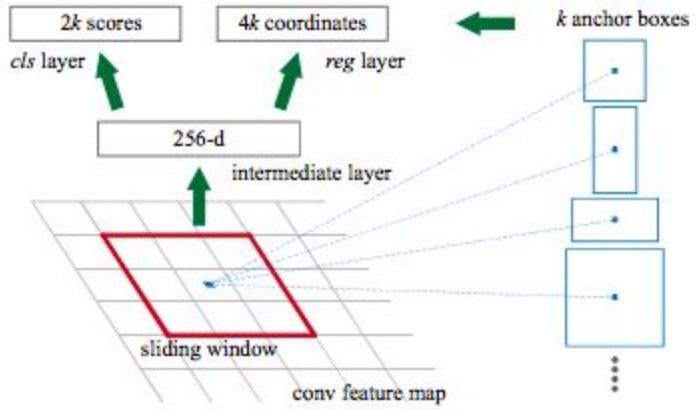
-> First the image is passed to any one of the standard CNN(Convolution neural network) typically Resnet50, resnet101 or Feature pyramid network are used for Feature extraction. FPN(Feaute pyramid network) is mostly used. Because in the bottom-up path way the output of the last layer of each stage is referenced as a set of feature maps in top-down path way as shown in below figure so it combines low-resolution, semantically strong features with high resolution from bottom-up path and semantically weak features from top-down path with connections. In short " It takes high level feature from frist pyramid and passes them to another pyramid which is top-down path way so doing this it has both high level and low level features

Next, these features are fed to RPN.

2 . Regional proposal network (RPN):

The RPN is a lightweight neural network that scans images in a sliding fashion and finds areas that contain objects. The regions that generate when RPN scans are called anchors. These anchors are generated on Feature maps send in First step.

These anchors are of different scales and different aspect ratio's.



Here **'k'** is the no of anchors. RPN network does two things: It will detect whether the object is **foreground** or **background** and it detects the **boundary boxes** of the objects. For each location in RPN it Makes **K** guesses so we have 2*k location scores for classification of foreground or background and 4*k coordinates for bounding boxes. So RPN has two has two kind's of problems to do one is **Binary Classification** and second is **Regresssion** for boundary boxes . If several anchors overlap we pick the anchors with highest overlapping score i.e IOU(intersection over union) score with some threshold usually will 0.5 to0.8(your choice) This process is called **Non-Max-Supression.** Finally the goal of this RPN is to identify which boxes are good, may contain foreground and to produce target regression coefficients

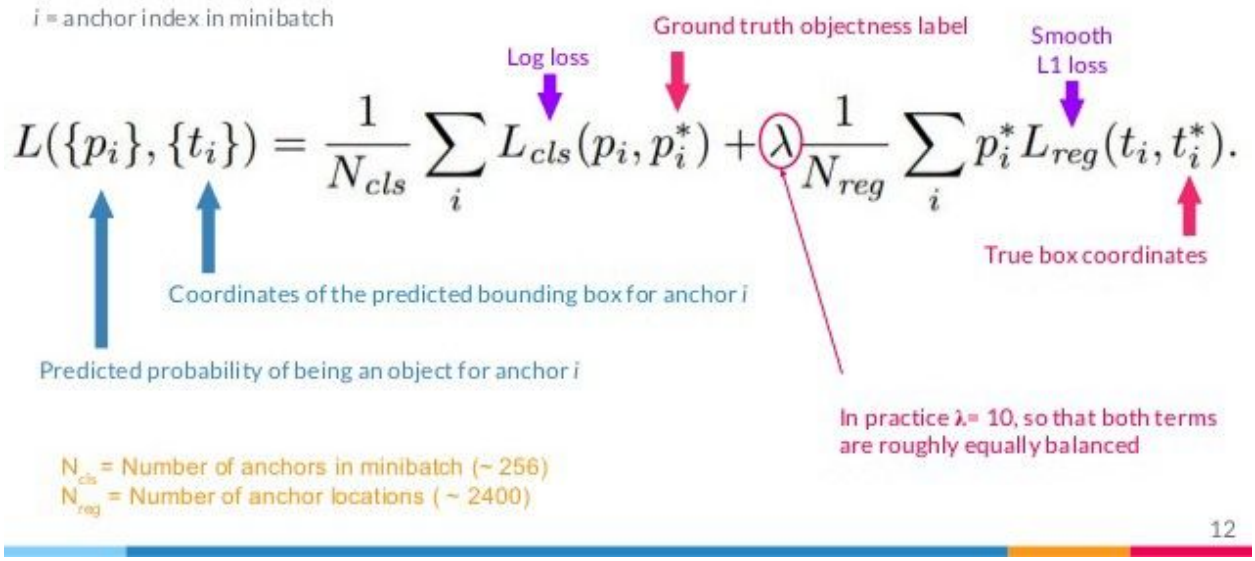
Finally It will output (1,M,N,k*2) matrix that indicate class of each position:-=> Here M*N is the Feature Map and k*2 is the probability of each class
And also it will output (1,M,N,K*4) indicating the position of each area.
These coordinate values are calculated as follows, where Tx , Ty , Tw , and Th respectively represent the x , y coordinates, width, and height of the top left corner of the target box, and Ox , Oy , Ow and Oh represent the x , y coordinates, width and height of the top left corner of the ground-truth box, respectively.

$$t_x = \frac{(T_x - O_x)}{O_x} \quad t_y = \frac{(T_y - O_y)}{O_y} \quad t_w = \log \frac{T_w}{O_w} \quad t_h = \log \frac{T_h}{O_h}$$

This function is easily reversible, given the regression coefficients and coordinates of the upper left corner and the width and height of the original bounding box, the upper left corner and width and height of the target box can be easily calculated.

Loss Function of RPN:
RPN is the sum of binary classification loss and boundary box regression loss.

Classification Loss = Binary Cross-Entropy
BB loss = Distance loss (Smooth L1 Loss)



After getting the 4 values of the object area then we use **Non-Max-Supression** discussed above
After this we send this to ROI align

3. ROI Align:

Before moving to this i will try to explain why we need this step, The classifiers don't accept the input of different size so they require a fixed size of input, but due to the bounding boxes refinement in RPI the ROI(region of Intrests) can have differents sizes.

So here **ROI_Align** comes into play here sample the feature map at different points and apply a bilinear interpolation (for me it sounds like it takes the cropping part of feature map and resize with interpolation).

4. Segmentation mask, classification Bounding box regression

Segmentataion Mask :

This mask branch consists of convolution networks which that take positive regions selected and given by ROIAlign and generate masks for them. Loss used : BInary cross-entropy

Classification:These ROI align features are send to Fully connected layer to softmax layer for classification. Loss : mean cross-entropy

Bound_Box loss : As similarly to classification ROI feature maps are send to Fully connected layer and a linear regression to find the Bounding box of object
Loss used : distance loss : Smooth_L1-Loss

In This project the input is Image and output is masks these are given in Encoded_pixel format
Given input with fixed size of(512,512) and loss are
 rpn_class loss=>RPN anchor classifier loss is it foreground or background binary cross-entroy
Rpn_box_loss => Smooth L1 loss of BBox co-ordinates
Mrcnn_class_loss: It is a average of cross-entropy loss of all the classes
Mrcnn_bbox_loss: It is also the mean of distance loss or Smooth L1 loss of each class
Mrcnn_mask_loss: Mean of cross-entropy loss of each class
Loss in logs =.> It is the sum of all the losses above

Here in Maskrcnn we don't explicitly give bounding boxes co-ordinates in training The matterport library I used, has a function for the bounding boxes
https://github.com/matterport/Mask_RCNN/blob/3deaec5d902d16e1daf56b62d5971d428dc920bc/mrcnn/utils.py#L34
So we have Input Image and output as Class_id, Bouningboxes as explained above and masks which is encoded_pixel format and also we have loss function to back propagate for training

The output predicted is The bounding boxes with respect to category,Class_id, masked_object

References:
https://medium.com/@nabil.madali/demystifying-region-proposal-network-rpn-faa5a8fb8fce
https://medium.com/@fractaldle/mask-r-cnn-unmasked-c029aa2f1296
https://jonathan-hui.medium.com/image-segmentation-with-mask-r-cnn-ebe6d793272
https://jonathan-hui.medium.com/what-do-we-learn-from-region-based-object-detectors-faster-r-cnn-r-fcn-fpn-7e354377a7c9
https://towardsdatascience.com/review-fpn-feature-pyramid-network-object-detection-262fc7482610
https://github.com/matterport/Mask_RCNN/

Thank you:)