**Steps followed :**

* First uses a Resnet to extract feature maps from the images.
* These feature maps are then passed through a Region Proposal Network (RPN) which returns the candidate bounding boxes.
* We then apply an RoI pooling layer on these candidate bounding boxes to bring all the candidates to the same size.
* And finally, the proposals are passed to a fully connected layer to classify and output the bounding boxes for objects .

**Detail steps :**

#### 1. Backbone Model

We use the ResNet 50 architecture to extract features from the images in Mask R-CNN. So, the first step is to take an image and extract features using the ResNet 50 architecture. These features act as an input for the next layer.

#### 2. Region Proposal Network (RPN)

Now, we take the feature maps obtained in the previous step and apply a region proposal network (RPM). This basically predicts if an object is present in that region (or not). In this step, we get those regions or feature maps which the model predicts contain some object.

#### 3. Region of Interest (RoI)

The regions obtained from the RPN might be of different shapes, right? Hence, we apply a pooling layer and convert all the regions to the same shape. Next, these regions are passed through a fully connected network so that the class label and bounding boxes are predicted. It generates the segmentation mask.

For that, we first compute the region of interest so that the computation time can be reduced. For all the predicted regions, we compute the Intersection over Union (IoU) with the ground truth boxes. We can computer IoU like this:

IoU = Area of the intersection / Area of the union

Now, only if the IoU is greater than or equal to 0.5, we consider that as a region of interest. Otherwise, we neglect that particular region. We do this for all the regions and then select only a set of regions for which the IoU is greater than 0.5.

#### 4. Segmentation Mask

Once we have the RoIs based on the IoU values, we can add a mask branch to the existing architecture. This returns the segmentation mask for each region that contains an object. It returns a mask of size 28 X 28 for each region which is then scaled up for inference.

## Step by Step Prediction and why it is used :

#### Backbone Model

We use the ResNet 50 architecture to extract features from the images in Mask R-CNN. So, the first step is to take an image and extract features using the ResNet 50 architecture. These features act as an input for the next layer.

**1. Stage 1: Region Proposal Network**

The Region Proposal Network (RPN) runs a lightweight binary classifier on a lot of boxes (anchors) over the image and returns object/no-object scores. Anchors with high objectness score (positive anchors) are passed to the stage two to be classified.

Often, even positive anchors don't cover objects fully. So the RPN also regresses a refinement (a delta in location and size) to be applied to the anchors to shift it and resize it a bit to the correct boundaries of the object.

### 1.a RPN Targets

The RPN targets are the training values for the RPN. To generate the targets, we start with a grid of anchors that cover the full image at different scales, and then we compute the IoU of the anchors with ground truth object. Positive anchors are those that have an IoU >= 0.7 with any ground truth object, and negative anchors are those that don't cover any object by more than 0.3 IoU. Anchors in between (i.e. cover an object by IoU >= 0.3 but < 0.7) are considered neutral and excluded from training.

To train the RPN regressor, we also compute the shift and resizing needed to make the anchor cover the ground truth object completely.

### 1.b RPN Predictions

Here we run the RPN graph and display its predictions.

## Stage 2: Proposal Classification

This stage takes the region proposals from the RPN and classifies them.

### 2.a Proposal Classification

Run the classifier heads on proposals to generate class propbabilities and bounding box regressions.

### 2.c Step by Step Detection

Here we dive deeper into the process of processing the detections.

1. Apply Bounding Box Refinement

2. Filter Low Confidence Detections

3. Per-Class Non-Max Suppression

**3. Stage 3: Generating Masks**

This stage takes the detections (refined bounding boxes and class IDs) from the previous layer and runs the mask head to generate segmentation masks for every instance.

### 3.a Mask Targets

These are the training targets for the mask branch.

### 3.b Predicted Masks