CS 419/619
Minor Project
Mask Rcnn image segmentation

Under the guidance of Dr. Surya Prakash

Namani Sreeharsh(180001032)

Rapolu Pulakitha(180001041)

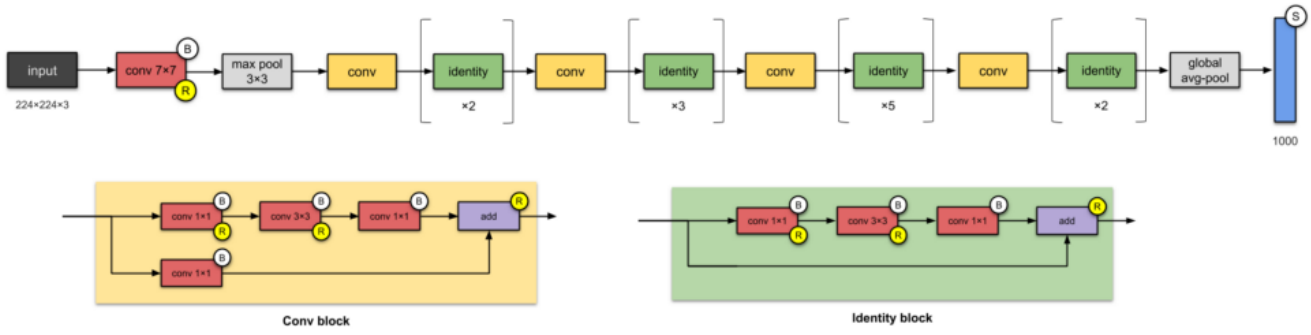
Ruchir Mehta(180001044)

Introduction

The aim is to create a mask around the entities/objects bearing information in the given image. This way, we can get rid of the portion of useless information or contain no data. We plan to implement it in a recorded video and then extend it to real-time video capture (future enhancement).
We have tried to train a model to perform image segmentation on dogs. We have used the Mask-R CNN algorithm, the state-of-the-art algorithm, to perform Image segmentation. We started with learning CNN; then we learned RCNN. Further, we referred to Fast-RCNN and Faster-RCNN.

Important Terms:

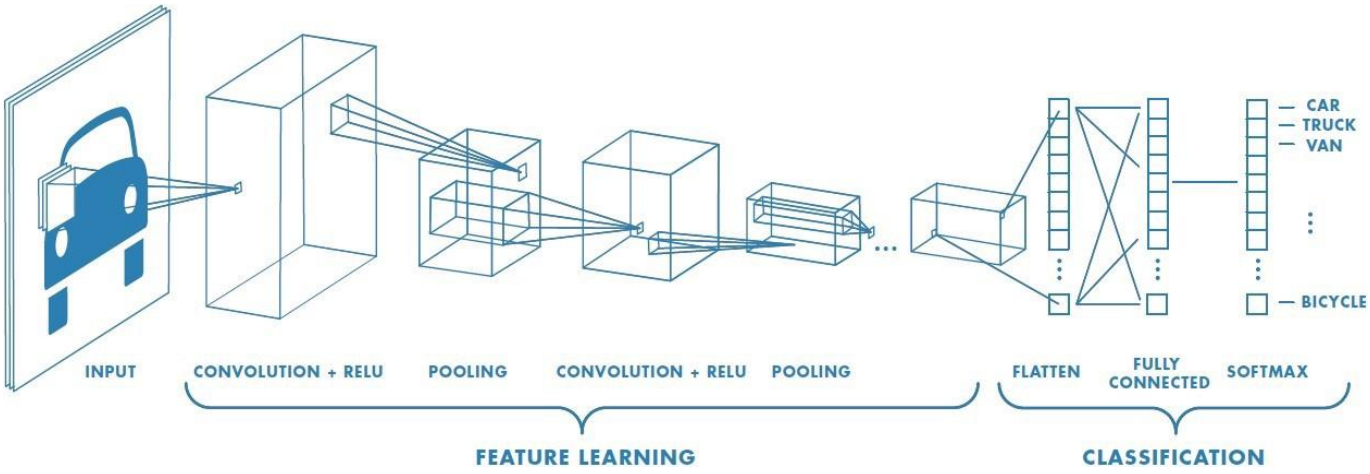
a. Image segmentation
The technique of splitting an image into segments based on the objects present and their relevance is known as image segmentation. Because we can acquire the particular pixel-wise area of the items instead of an approximate placement from a rectangle box, it makes it easier to evaluate the image.
b. Region-based segmentation
For this situation, we can set a threshold. The pixel esteem falling beneath or over that threshold can be characterized in like manner (like an object or the background).



c. Edge detection segmentation
We can utilise this discontinuity to detect edges and so determine an object's border. This aids in the detection of the forms of several objects in a single image.

d. CNN

A CNN is a class of artificial neural network, most commonly applied to analyze visual imagery. The name "convolutional neural network" refers to the network's use of the convolution mathematical procedure. Convolutional neural networks are a type of neural network that uses convolution rather than standard matrix multiplication in at least one layer.



A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning method for assigning relevance (learnable weights and biases) to various aspects/objects in a picture and distinguishing between them. The amount of pre-processing required by a ConvNet is much less than that required by other classification methods.While basic approaches require hand-engineering of filters, ConvNets can learn these filters/characteristics with enough training.

The ConvNet's job is to compress the images into a format that is easier to process while preserving elements that are important for obtaining a decent prediction.

A ConvNet is able to **successfully capture the Spatial and Temporal dependencies** in an image through the application of relevant filters.

There are pooling layers to reduce the dimension of image. There are two types of Pooling: Max Pooling and Average Pooling.

At the end, adding a Fully-Connected layer is a (usually) cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space.

Region Proposal Network

RPN has a classifier and a regressor. The chance of a proposal including the target object is determined by the Classifier. Regression regresses the proposal coordinates. Its goal is to suggest many things that can be identified inside a single image.

Region of Interest (ROI pooling):
After finding probable Region of Interest(RoIs), we pass them into a Fully connected neural network. Since the shape of each RoI would be different, it would be impossible to construct a neural network which trains each RoI. So there should be a layer which normalises the shape of all predicted RoIs into the shape which is suitable for the neural network i.e., into the shape of the inputs which are accepted by the neural network and RoI pooling layer serves this purpose.

e. RCNN
R-CNN generated region proposals based on selective search and then processed each proposed region, one at time, using Convolutional Networks to output an object label and its bounding box.

f. Fast RCNN
Fast R-CNN made the R-CNN algorithm much faster by processing all the proposed regions together in their CNN using a ROIPool layer.

g. Faster R-CNN
Faster R-CNN pushed it even further by performing the region proposal step using a ConvNet called Region Proposal Network(RPN). Both the RPN, and the classification and bounding box prediction network worked on common feature maps, thus making inference faster.

- Faster R-CNN starts by extracting feature maps from the images with a ConvNet.
- The potential bounding boxes are then returned after passing these feature maps through a Region Proposal Network (RPN).
- Then, on these candidate bounding boxes, we apply a RoI pooling layer to make all of the candidates the same size.
- Finally, the proposals are sent to a fully connected layer that classifies and outputs the object's bounding boxes.

h. Mask R-CNN
Mask R-CNN is essentially a "Faster R-CNN" extension. For object detection tasks, "Faster R-CNN" is commonly used. It returns the class label and bounding box coordinates for each object in a supplied image.
In Mask R-CNN, we utilise the ResNet 101 architecture to extract features map from the images, same to how we use the ConvNet in Faster R-CNN to extract feature maps from the image. So, the first step is to use the ResNet 101 architecture to extract features from an image. These features serve as an input to the next layer.

Mask R-CNN also generates the segmentation mask.

Region of Interest (RoI)
The areas acquired from the RPN may be of various shapes. Thus, we apply a pooling layer and convert every one of the areas to a similar shape. Then, these regions are gone through a completely connected network with the goal that the class label and bounding boxes are anticipated.

Till this point, the means are practically like how Faster R-CNN functions. Presently comes the distinction between the two systems. Likewise, Mask R-CNN additionally produces 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.

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.

Methodology

The pipeline of Faster RCNN :

Raw Image ---> CNN(Eg Resnet101) --(feature extraction)-- RPN --(bounding boxes aka extracting ROIs)--> ROI Pooling ---> Fully Connected Neural Network

Classification + Localization(bounding box) = Object Detection(with only one object in the image)

Bounding box= (x,y,w,h), (x,y) is the center of bounding box and w and h are width and height of the bounding box

Object Detection

- finding out exact location of prolly multiple object present in the image
- Each image needs a different number of outputs

Object detection + Semantic segmentation = Instance segmentation (Mask-RCNN)

Object Detections (by Faster RCNN)
Semantic segmentation (Fully Convolutional networks (FCN) = CNN + Fully Connected Neural Network)

1. RCNN (ROI-region of interest) ~2000 boxes
    a. Selective search is an example of region proposal network
2. Faster RCNN (region proposal network)
    a. Stage 1: Determine the bounding box (ROI)
    b. Stage 2: Determine the class label for each ROI by ROI pooling
    c. ROI pooling, RoIPool has data loss
    d. So, RoIAlign is used that preserves spatial orientation of features with no loss of data
3. FCN-To predict the mask for each ROI
    a. Convolutional layers retain special orientation

Dataset Used = Image + Segmentation Mask
Assumption: Each image contains only one object
We have used a data set that is linked in the github repository here

This dataset has segmented pictures of dogs in different postures. It contains the (x, y) coordinates of the mask in all the images around the object.
Training dataset has 50 labelled samples and testing has 6 labelled samples.

Before running notebook, we need to store the dataset images.zip file under the 'dataset' folder:

Zip file structure:

```
images.zip
|- "train" directory
  |- jpg image files of training data
  |- "via_region_data.json" annotations file of training data
|- "val" directory
  |- jpg image files of validation data
  |- "via_region_data.json" annotations file of validation data
```

Results

Every segmented mask in an image is represented as a binary mask with the same size as that of image.

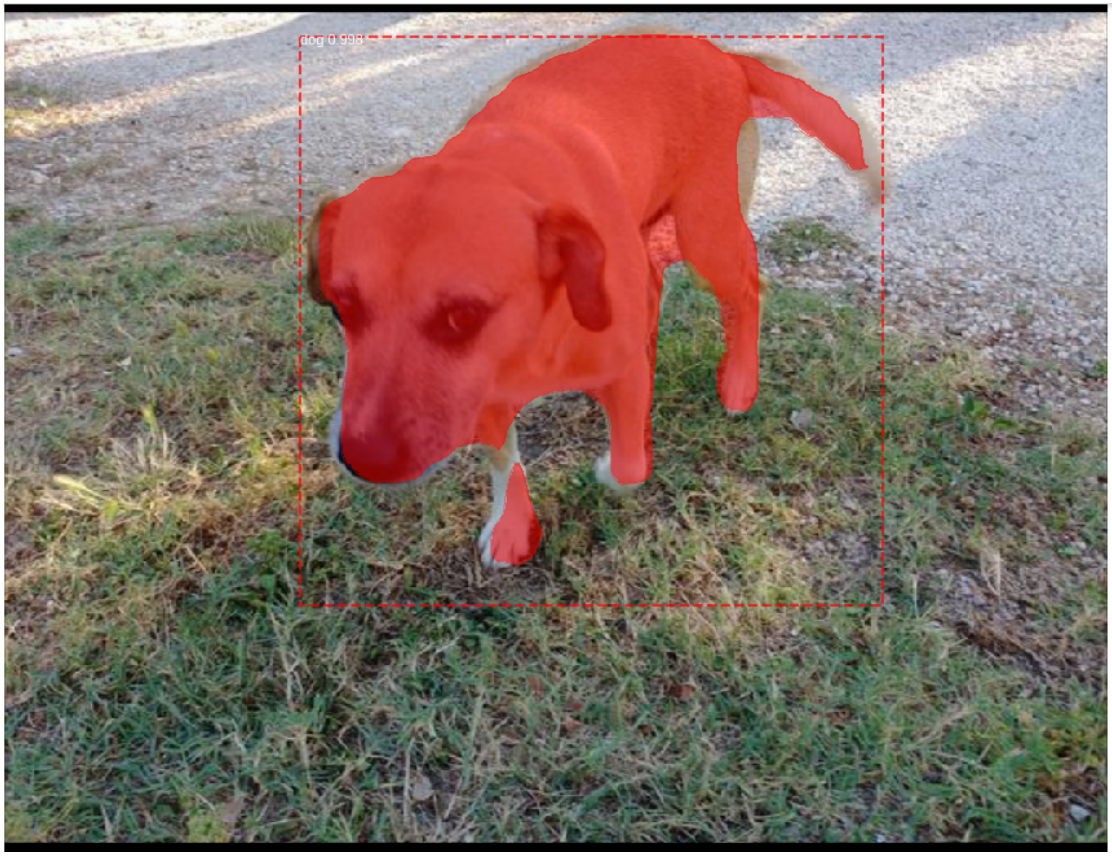
```
Processing 1 images
image                    shape: (1024, 1024, 3)       min:    0.00000  max:  255.00000  uint8
molded_images            shape: (1, 1024, 1024, 3)    min: -123.70000  max:  151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max: 1024.00000  int64
anchors                  shape: (1, 261888, 4)        min:   -0.35390  max:    1.29134  float32
gt_class_id              shape: (1,)                  min:    1.00000  max:    1.00000  int32
gt_bbox                  shape: (1, 4)                min:  283.00000  max:  850.00000  int32
gt_mask                  shape: (1024, 1024, 1)       min:    0.00000  max:    1.00000  bool
```

Image size is 1024 x 1024 x 3(channels)

gt_mask is a binary mask of the segment(dog) with the same size as a that of image


dog 0.998


dog 0.999


dog 0.998


dog 0.998

Conclusions

We successfully implemented the mask-rcnn algorithm on images to predict the segmentation binary mask on the object i.e. dog present in the image.

4. References

a. https://www.pyimagesearch.com/2020/09/28/image-segmentation-with-mask-r-cnn-grabcut-and-opencv

b. https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/?utm_source=blog&utm_medium=computer-vision-implementing-mask-r-cnn-image-segmentation

c. https://www.analyticsvidhya.com/blog/2019/07/computer-vision-implementing-mask-r-cnn-image-segmentation/

d. https://www.analyticsvidhya.com/blog/2021/03/introduction-to-image-segmentation-for-data-science/

e. https://en.wikipedia.org/wiki/Convolutional_neural_network

f. https://learnopencv.com/deep-learning-based-object-detection-and-instance-segmentation-using-mask-rcnn-in-opencv-python-c/