

Analysis of Instance Segmentation using Mask-RCNN

Aniket Satishchandra Paste

Student, School of Computer Science & Engineering
K.L.E Technological University
Hubballi, India
pasteaniket111@gmail.com

Dr. Satyadhyan Chickerur

Professor, School of Computer Science Engineering
K.L.E Technological University
Hubballi, India
chickerursr@gmail.com

Abstract -Object detection has been one of the greatest achievement in the field of Machine learning. Most of the models in the domain carry out the process of identifying the character or object using a bounding box. In the recent years, detection of the object using Instance segmentation has been in limelight. However, no interest has been showed towards the data present in the field of Entertainment. The focus of this paper is to carry out pixel level comparison and to study the behaviour of the model by analysing the result obtained at various different instances.

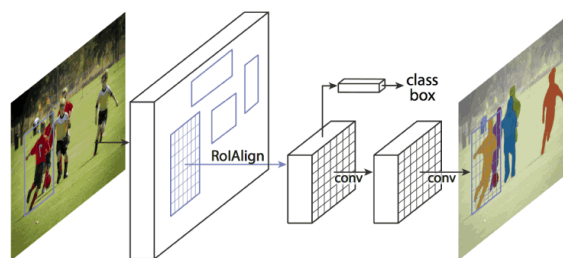
Keywords – Machine Learning, Bounding Box, Instance Segmentation, Entertainment field.

I. INTRODUCTION

There has been an exponential growth in the field of Vision community which specifically involves Object classification and detection. Models of many different variations and flavours are available which allows the end user to get the desired output in the required time. Entertainment is another great source of data which has great potential in the future. There is surplus amount of data available in this domain, which can provide a great significant contribution to the community of Deep Learning. On combining the present state of art architecture in Machine Learning with the field of Entertainment, many new techniques and methodologies can be derived. In this paper, we will observe the training of model and analyse the result obtained after carry out the Instance Segmentation on the images of Tom and Jerry using Mask RCNN. The objective is to study the behaviour of the model while carry out semantic segmentation. It will allow us to deduce some insights on how the model is getting trained and analyse what type of results are achieved at various different instances during training phase of model.

II. RELATED WORK

For the last 10 years, there has been wide variety of approaches formulated and applied in order to achieve better performance and accuracy. Initially the basic brute force approach was to apply the Convolutional Neural Network [12],



The Mask R-CNN framework for instance segmentation

Figure.1. Methodology

Where the High resolution images is divided into many different sections and is used to train the model. But such approaches had its own limitations, like large memory requirements and computational power requirements. With the course of time, much better approaches were developed and applied to make the learning process of the model much easier and faster. One of such approaches is Regional Convolutional Neural Network, R-CNN [1] where High capacity CNN are used to propose the desired regions. Such regions can be used to segment and identify the objects in the image. Further methodologies which were proposed were based upon the previous findings, enhancing their performance. Fast R-CNN [3] is one such approach which is the enhanced version of R-CNN. In this model, in place of normal CNN, Fast R-CNN [3] makes use of VGG16 network which is able to yield great result and performance. Faster R-CNN [4] is another model whose working is based upon Fast R-CNN. This is level 3 improvisation of R-CNN, which makes use a novel approach named Regional Proposal Network (RPN). RPN is based upon convolutional neural network that simultaneously predicts object bounds and objectness scores at each position. On using this approach, it has greatly improved the ability of detecting the required regions of the model. In this paper we have made use of Faster R-CNN whose architecture will be used in Backbone architecture.

III. MASK R-CNN

The main ideology upon which the working of Mask RCNN is called Instance Segmentation. This type of classification is considered only when the location of the object is identified where the pixels are also differentiated among each objects in the frame. The working of Mask RCNN can be divided among two parts, one which scans the whole image and identify the regions of interest. The second part of the model is responsible to generate the bounding boxes and add masks to the proposed regions. The first part of the model is implemented using the concepts of Faster R-CNN. Apart from Faster R-CNN, the authors has improvised most of its previous works, and have combined them into Mask RCNN.

A. Backbone

For Backbone of the model, any convolutional Neural Network which serves the purpose of extracting features from the image can be used. For the following analysis, Resnet50 [13] has been used. Other models like Resnet101 which is the successor of the Resnet50 can also be used. The main function of this segment is to identify the low level features (edges and corners) and further detect high level features. These extracted features is then feed to FPN (Feature Pyramid Network) [4]. The main functionality of FPN is to improve the standard and performance of high level feature extraction, which is done by Resnet50 or any other respective model.

B. Region Proposal Network (RPN)

The main objective of RPN is to implement the functionality of Region Proposal. This is done by scanning the image and identifying the regions which may contain the object. The speed at which RPN operated is very high in case when done with GPU. One of the major contributing factors to this speed is the usage of weights stored in the FPN. This allows RPN to reuse the extracted features efficiently and avoid duplicate calculations.

There are two outputs obtained from RPN. One is called Anchor Class which acts as the actual bounding box for the object in the image. The second output is Bounding Box Refinement, which is not exactly encloses the object in the frame, but RPN uses it to calculate the delta value(% change in x, y, width and height) which is then further used to refine the anchor box to fit in the object.

C. ROI Classifier & Bounding Box Regressors:

The operations of ROI Classifier and Bounding Box Regressor depends upon the output obtained from RPN in the form of ROIs (Region of Interest). The two outputs obtained from this stage are Class and Bounding Box Refinement. Identification of class is basically the part Multi-Class Identification in where the detected object is assigned to a particular class to which it belongs. Bounding Box Refinement is further refinement of the results obtained from RPN where the location and the bounding box size is redefined to capture the object in the image.

D. ROI Pooling:

One of the major challenges faced in object detection and classification is that the Classifier do not handle the variable input size properly. This issue arises due to the

fact that the classifier can handle only fixed input size. However, in Mask R-CNN, due to the involvement of RPN, different Regions of Interest of different sizes are proposed. This result of RPN is then used by a special function called ROI Pooling. ROI Pooling is implemented using Bilinear Interpolation, which allows the model to select the size of the box.

E. Segmentation Masks:

One of the biggest change introduced by Mask R-CNN is the Classification of Images using Masks through Instance Segmentation. In this approach, pixel level comparison is done in order to get the exact layout of the object present in the image.

IV. PROPOSED APPROACH

Mask RCNN is a deep neural network aimed to solve instance segmentation problem in machine learning or computer vision. In other words, it can separate different objects in an image or a video. On feeding an image as input, it gives you the object bounding boxes, classes and masks. There are two stages in the working of the Mask RCNN. First, it generates proposals about the regions where there might be an object based on the input image. Second, it predicts the class of the object refines the bounding box and generates a mask in pixel level of the object based on the first stage proposal. Both stages are connected to the backbone structure. The first stage is achieved by using existing model; in this case, we have used Faster RCNN. The principles of Mask RCNN is similar to Faster RCNN, we have two output for each image, one is class label and the other is the bounding box. Whereas in case of Mask RCNN, we have an additional third branch that gives the mask of the identified object as output. In order to calculate Pixel level comparison, Mask RCNN make use of RoIAlign. In Faster RCNN in place of RoIAlign, RoIPool is been used. The main purpose of RoIPool is to allow one forward/backward pass for multiple RoIs in one input image. Though RoIPool is very fast, the major drawback is its process of dividing a large resolution image into rather small map by quantization, which results in misaligned results on the boundaries. The above drawback is taken care by RoIAlign by using bilinear interpolation.

V. NETWORK ARCHITECTURE

Depending upon the availability of the good performing model, the Mask RCNN can be implemented with any object detection model. In this case, we have used convolutional backbone architecture used which is used for feature extraction over an image. The Mask RCNN makes use of residual networks using the nomenclature Network-Depth-Features. In Residual Networks we have ResNet and ResNetXt. The Residual Networks can also be replaced by another model known as Feature Pyramid Network. FPN follows top-down architecture with lateral connections to build an in-built network feature pyramid from a single scale input. By using FPN, we can extract RoI features from different levels of the feature pyramid according the data present in the pyramid. Apart from this, the rest of the

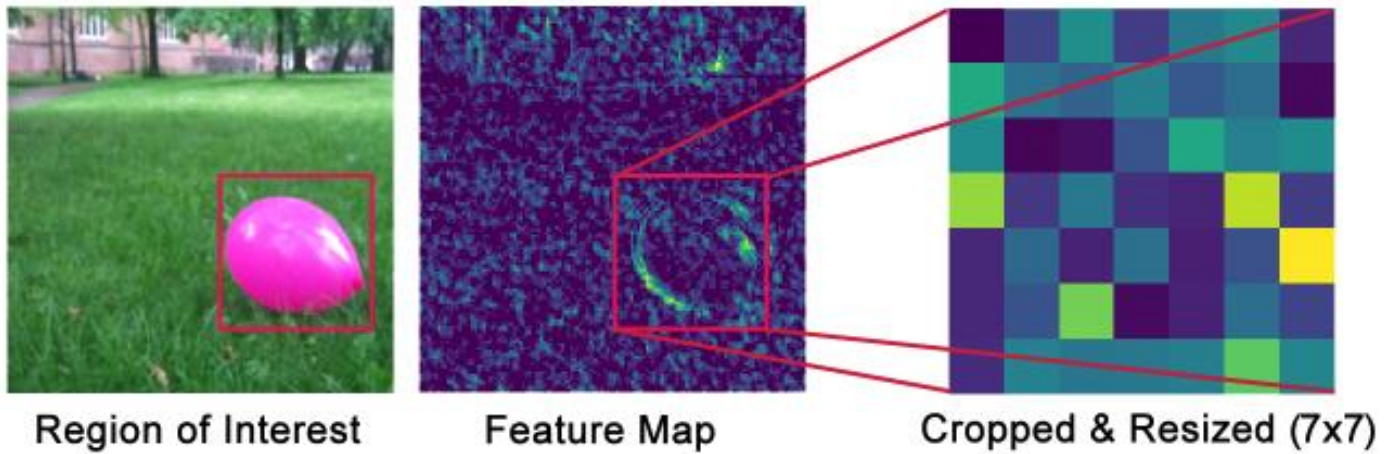


Figure.2. Instance Segmentation

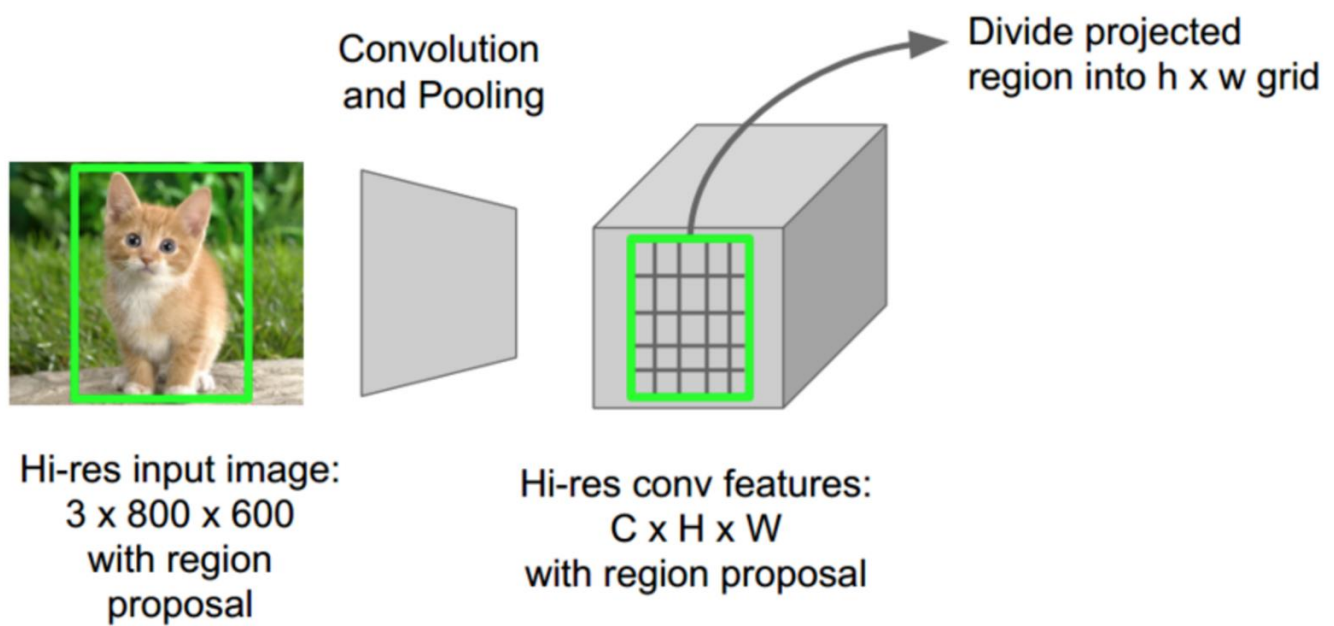


Figure.3. ROI Pooling

Approach is similar to Vanilla ResNet. Both these approaches gives excellent gains in both accuracy and speed to Mask RCNN.

VI. IMPLEMENTATION DETAILS

To achieve better results, dataset size plays a very crucial role. Many different datasets are available, using which people today are trying to develop new approaches and methods. However, there is no significant role of Animated and cartoon movies in the field of Deep Learning. Using this huge amount of data, amazing feats can be achieved. In this research we have made use of the cartoon characters 'Tom & Jerry', in where different variety of episodes were collected and converted into frames. Around 1500 images were selected, which were arranged in such a way that no two images were similar in any form.

All the images were then segmented into two different formats: Masks and XMLs. Masks are used for processing at the pixel level comparison, which allows the model to generate the results with very high precision. The XML files are the files which comprises of the coordinates of the bounding box which will be the part of the final results. In case of Tensorflow, during the conversion of dataset into TF Record files, there will be need of separate XML and Mask files in the form of .PNG. Whereas in case of frameworks like Keras, along with Mask files, there is a need of csv file which includes the full path of the actual image, mask image and the coordinates of the bounding box which is stored in the XML files. The backbone model which we have used to support this architecture is Resnet50. We have trained this model with the value of epochs set to 50 and number of steps set to 2000. The output of the model at different instances has been captured and is displayed in the Table no 1 and 2.

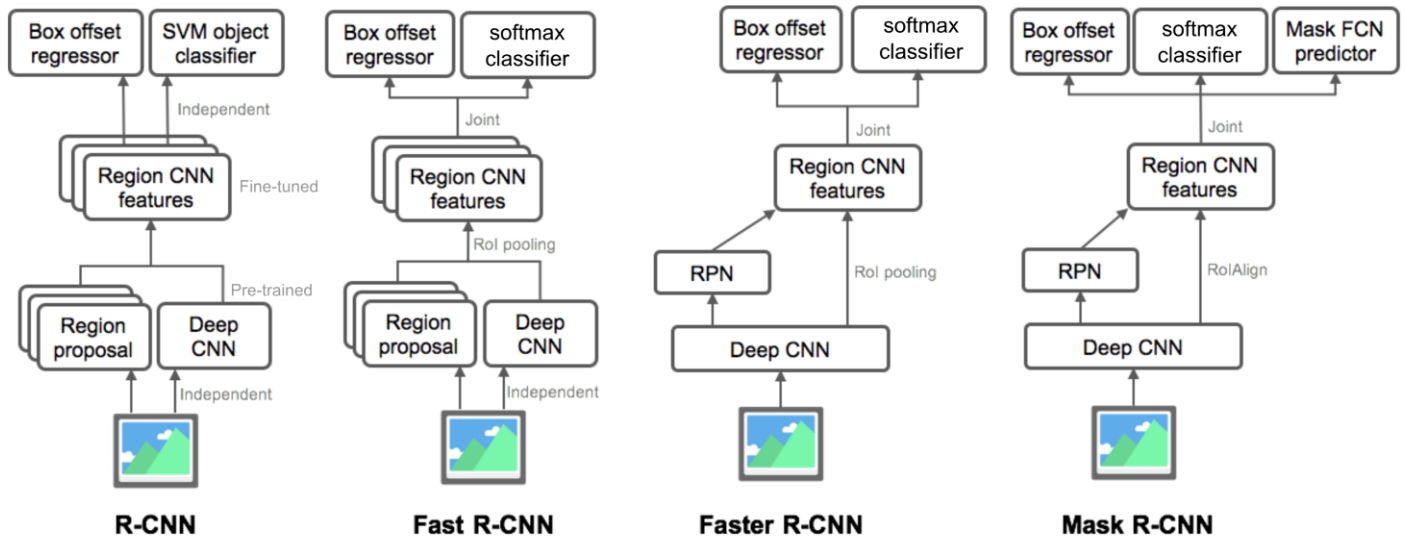


Figure 4. Network Architecture

| method | proposals | training data | COCO val | | COCO test-dev | |
|----------------------------------|-----------|---------------|----------|---------------|---------------|---------------|
| | | | mAP@.5 | mAP@[.5, .95] | mAP@.5 | mAP@[.5, .95] |
| Fast R-CNN [2] | SS, 2000 | COCO train | - | - | 35.9 | 19.7 |
| Fast R-CNN [impl. in this paper] | SS, 2000 | COCO train | 38.6 | 18.9 | 39.3 | 19.3 |
| Faster R-CNN | RPN, 300 | COCO train | 41.5 | 21.2 | 42.1 | 21.5 |
| Faster R-CNN | RPN, 300 | COCO trainval | - | - | 42.7 | 21.9 |

Figure 5. Comparison of different Regional Proposal Networks [11]

VII. RESULTS

In order to understand the results obtained, the output has been classified into three sections. The first section demonstrates the results accumulated by the Resnet50 and RPN (Regional Proposal Network), which constitutes the internal architecture. After training the model for 50-100 epochs, the model begins proposing the Region of interest which can be depicted from the Table no. 1. According to the images, the ROIs proposed are randomly scattered all across the frame. Since the ROIs proposed by the model are not accurate, we can concur that the model is still at its initial stage. The second section shows the initial attempts of the model to capture the object using bounding boxes. This result is obtained after training the model for 400-600 epochs. Images of Table no 1. represents the ROIs in the form of bounding boxes. Even though the object is not captured, we can easily conclude that the RPN and Resnet50 are partially trained. The results of third section is represented with diagrams in Table no 2. These images shows the partial success of the model to capture the object in the image using the bounding boxes. Model is trained up to 800-1500 epochs, in order to achieve the results. Results of 800 and 1200 epochs do not exactly capture the object in the frame, but the results of model up to 1500 epochs successfully captures the object using multiple bounding boxes. At this point, we can say that the Resnet50 and RPN are sufficiently trained to identify the object. After this point of time, it is the work of ROI Pooling and Segmentation Masking to exactly capture the object into a single bounding box.

The results of the fourth section can be tracked using the results of 1800 and 2000 epochs represented in the Table no 2. In model with 1800 epochs training, the model attempts to capture the object with few number of Bounding Boxes. The most optimal result is achieved after training the model for 2000 epochs. In the case of performance, out of all the available Regional Proposal Networks Faster RCNN tends to perform better. Figure no. 5 gives the quantitative details of different model's performance. The second column of the table represents the number of RoIs made by the region proposal network. The Third column represents the training dataset. In this case COCO dataset is being used. The fourth column represents the mean average precision(mAP) in measuring accuracy.

VIII. CONCLUSION

In this paper, we have demonstrated the usage of Mask R-CNN in the field of Entertainment and show how the model is expected to behave at different instances during training process. The above model has been trained on Tesla K80 which has a computing power of 3.8. However the results of the model will always be directly dependent on the type of GPU and Data used for the training process. It is recommended to have a huge dataset (1000-2000 images) so that wide variety of data images can be used to train the model, improving the capability of capturing the object from any kind of image. While training the model, it is observed that during the initial phase the model tries to identify the different Regions of interest from the image. Hence initially

the internal models like Resnet50 and Region Proposal Network is getting trained with the images. After 200 epochs when the internal models start giving suitable results, ROI Pooling and Segmentation Masking comes into picture

where aggregation of all the ROIs takes place, to exactly identify the object.








| EPOCHS | IMAGE 1 | IMAGE 2 | IMAGE 3 |
|--------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| 50 |  |  |  |
| 150 |  |  |  |
| 250 |  |  |  |
| 400 |  |  |  |
| 600 |  |  |  |

TABLE 1

| EPOCHS | IMAGE1 | IMAGE2 | IMAGE3 |
|--------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| 800 |  |  |  |
| 1200 |  |  |  |
| 1500 |  |  |  |
| 1800 |  |  |  |
| 2000 |  |  |  |

TABLE 2

ACKNOWLEDGEMENT

I would like to express my special thanks to Dr.S.Chickerur for his valuable guidance which inspired me greatly to do my best. I am immensely grateful to him for providing all the necessary resources for my research. I would also like to thank K.L.E Technological University for providing me opportunity to carry out this research.

REFERENCES

- [1] *R-CNN Rich feature hierarchies for accurate object detection and semantic segmentation*, Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, CVPR' 14.
- [2] *MR-CNN, Object detection via a multi-region & semantic segmentation-aware CNN model*, Spyros Gidaris, Nikos Komodakis, ICCV' 15.
- [3] *Fast R-CNN*, Ross Girshick, ICCV' 15
- [4] *Faster R-CNN, RPN Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, Shaoqing Ren, et al. NIPS' 15
- [5] *YOLO v1, You Only Look Once: Unified, Real-Time Object Detection*, Joseph Redmon, et al., CVPR' 16
- [6] *HyperNet: Towards Accurate Region Proposal Generation and Joint Object Detection*, Tao Kong, et al., CVPR' 16.
- [7] *R-FCN: Object Detection via Region-based Fully Convolutional Networks*, Jifeng Dai, et al, NIPS' 16.
- [8] *FPN, Feature Pyramid Networks for Object Detection*, Tsung-Yi Lin, et al. CVPR' 17.
- [9] *YOLO9000: Better, Faster, Stronger*, Joseph Redmon, Ali Farhadi, CVPR' 17.
- [10] *RetinaNet, Focal Loss for Dense Object Detection*, Tsung-Yi Lin, et al, ICCV' 17.
- [11] *Mask R-CNN*, Kaiming He, et al. ICCV' 17
- [12] *Gradient-Based Learning Applied to Document Recognition*, Yann Lecun, Leon Bottou, Yoshua Bengio, and Patrick Haffner, Proc. Of the IEEE 1998.
- [13] *Deep Residual Learning for Image Recognition*, Kaiming he, Xiangyu Zhang, Shaoqing Ren, Jian Sun.