Kayaker Detection with Mask R-CNN

Nikola Dakić
University of Novi Sad
Faculty of Technical Science
Novi Sad, Serbia
nikoladakic@uns.ac.rs

***Abstract* - This paper shows a methodological approach with a transfer learning technique for kayakers detection and instance segmentation using the mask region proposal convolutional neural network (Mask R-CNN). Custom dataset used in this project is consist of 42 kayaker images with pixel-by-pixel polygon annotation for the automatic segmentation task. The proposed transfer learning technique makes use of a Mask R-CNN model pre-trained on Microsoft Coco dataset. The pre-trained model is later fine-tuned on custom dataset. Generated model has achieved 0.859 mean average precision (mAP) on test set. The results indicate promising application of deep learning detecting kayakers on video.**

Keywords* – *kayaker, Object detection, Mask R-CNN, Transfer learning, Convolutional neural network

1. INTRODUCTION

Object detection is a challenging computer vision task that involves predicting both where the objects are in the image and what type of objects were detected. The Mask Region-based Convolutional Neural Network, or Mask R-CNN, model is one of the state-of-the-art approaches for object recognition tasks. The Matterport Mask R-CNN project provides a library that allows you to develop and train Mask R-CNN Keras models for your own object detection tasks. This library allows fast training via transfer learning with top performing models trained on challenging object detection tasks, such as MS COCO.

1.1 DATASET

The dataset is comprised of 42 images that contain kayakers, and annotation file in .json format which contains the coordinates of all the polygons. Images for dataset are collected manually from google search engine. For each dataset (training and test) .json annotation file is created manually using VGG Image Annotator (VIA).

The Mask R-CNN is designed to learn to predict both bounding boxes for objects as well as masks for those detected objects. So for image segmentation capabilities of the model, dataset needs to have coordinates of polygons. There are a few steps required in order to prepare the dataset for modeling:

- Download images
- Create and parse annotation files
- Create KayakerDataset object that can be used by Mask_RCNN library

1.2 PROBLEM STATEMENT

The goal of this project is to do image classification, object detection and image segmentation on the input video and to produce output video with detected kayaker. Example of video frame can be seen on Fig 1.

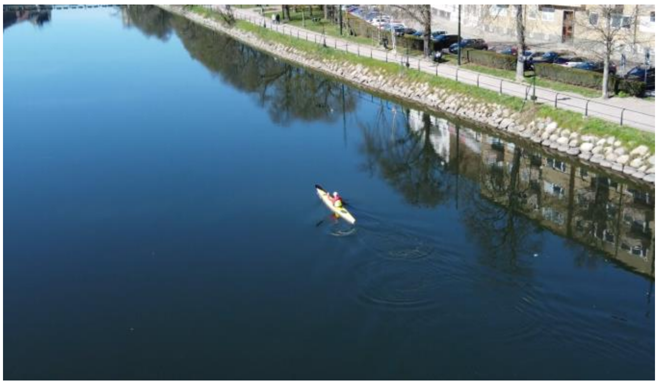

Fig. 1. Video frame example

- **Input**: .mp4 video, resolution=1920x1080, frame_rate=30fps
- **Output**: video with instance segmentation mask, bounding boxes around kayaker, class and score labels

2. BACKGROUND

The object segmentation task has evolved over the years in the field of computer vision. The recent advancements in deep neural networks have outperformed traditional machine learning models in the object classification and detection tasks.

Deep Learning in the field of computer vision developed from image classification to object localization, object segmentation, and to instance segmentation [2] (Fig.2).

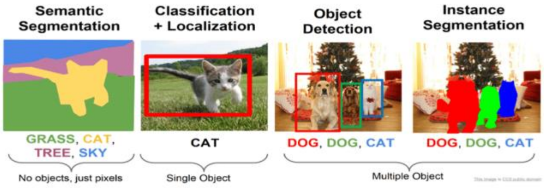

Fig. 2. Comparison of semantic segmentation, classification and localization, object detection and instance segmentation. (Figure adapted from [5])

There are two type of object detectors: two stage detectors-Region based R-CNN family and single stage detectors: such as you look only once (YOLO) [4] and single shot detector (SSD) [5].

Two-Stage Detectors The region proposed methods of object detection are from the R-CNN family. The model pipeline detects objects in two stages: (1) generating proposal of region of interest (RoI) and (2) classification of objects in the proposed regions. The region proposal algorithms have evolved in the following order [6]: R-CNN, Fast R-CNN, Faster R-CNN and Mask R-CNN.

Single-Stage Detectors The single-stage detectors take an approach of a simple regression problem predicting the bounding box coordinates and the class probabilities from images in one evaluation. The YOLO [4] and SSD [4] families belong to this category of detector, and are designed for speed and real-time use while they compromise on accuracy.

2.1 Mask R-CNN

This algorithm belongs to the two-stage detectors that uses feature pyramid network (FPN), and region proposal network (RPN) for the object segmentation. It is an extension of the Faster R-CNN [19] along with a new pipeline for masking the detected objects for instance segmentation [8] (cf. Fig. 2). In Mask R-CNN, the spatial layout of input object encoded by a mask and predicted by using fully connected network (FCN) pixel to pixel convolutions instead of the fully connected (FC) layers that will flatten into vectors lacking spatial dimensions as in Faster R-CNN. The RoI-align (instead of RoI-pooling) calculated with bi-linear interpolation to improve segmentation accuracy. This technique had gained popularity as it improves object instance segmentation precision [2,9,6]. In this work, for instance segmentation of kayaker at pixel-level, it is opted a two-stage detector: Mask R-CNN.

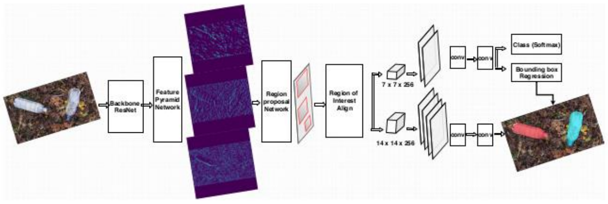

Fig. 3. Mask R-CNN Architecture

3. Technical Approach

The implementation of kayaker segmentation model started with the datasets preparation. The Mask R-CNN model was built with Python 3, TensorFlow, Keras, and OpenCV libraries by adapting the code from open-source Mask R-CNN implementation [1]. The annotated custom dataset along with pre-trained model weights were fed into the training pipeline consisting of several stages.

Model configuration: The ResNet-50 and ResNet-101 were the feature backbone network options for Mask R-CNN algorithm. ResNet-101 network is chosen. The images were resized to dimension 512 × 512. Detection maximum instance was updated from 100 to 5 that facilitates the maximum detection of object instances. The detection minimum confidence parameter value was set to 0.9 for tuning model's performance.

Dataset description: The dataset consists of custom images downloaded from the internet. Though there are many state-of-the-art datasets such as COCO and the PASCAL VOC that have general images for training and research purpose, it was noted that the availability of images pertaining to the kayaker class was limited. To overcome this problem, a discrete set of images were downloaded from the internet. The set contains single and multiple kayaker images of normal and deformed features possessing various backgrounds. Detecting objects in multi-kayaker images would be a challenge due to occlusion and lighting conditions. This will be a challenge for the generalization of the kayaker instance segmentation for the model.

Data pre-processing: Image annotation of each object is a time-consuming and tedious task. However, it is an essential initial step for an instance segmentation task. It is necessary to define the pixel-wise ground truth for the target objects for the models to perform segmentation and mask generation. Unlike other object detectors, the Mask R-CNN model requires pixel-wise annotation for training. There are many publicly available tools for data annotation such as VIA-VGG [3] and LabelMe [10].

For this work, pixel-wise polygon annotation of the images was performed for instance segmentation training as per the COCO dataset format. Custom dataset images were of varied sizes and formats. VIA-VGG annotator tool was used for the pixel-wise polygon graphical annotation. The tool generates the output in json format for the annotated images. The segmented ground truth mask represents a region-wise spatial position and axis of each target object.

The images were resized to dimension 512 × 512 × 3 and to preserve the aspect ratio, each image was padded with zero to match the one-size training requirements (square format of same dimension).

All the images are stored in the system in two folders with the training images and the pixel annotation and labeling information as arrays for each image in the .json file. These image files are read one-by-one from the system drive for resizing. Video files input for testing is pre-processed with OpenCV libraries in python to extract the images frame-by-frame to be fed into the detection pipeline for segmentation. The segmented output image frames were saved into the set directory path with the OpenCV video.

Transfer learning: The custom dataset training was initialized with the Mask R-CNN model pre-trained on MS-COCO dataset with 80 classes. The model weights in the initial layers representing low-level features will be useful in many classification tasks. While deeper layers learn the high-level features, this can be altered and retrained according to the problem definition. In the initial phase, the models were trained with backbone ResNet-50 and ResNet-101 for the kayaker instance segmentation using stochastic gradient descent (SGD) and Adam optimizer. The deeper network of ResNet-101 and SGD optimizer performed better comparatively and it was opted for training on the custom dataset.

4. Results

Models performances were compared against loss metrics and were evaluated by using COCO evaluation metric mAP [8,14]. Image segmentation models being far too expensive for the cross-validation method, Mask R-CNN hyper-parameter tuning was based on the configuration parameters.

Table 1. shows models performance.

TABLE I. PERFORMANCE MEASURE OF THE MODEL

Model	Starting Weights	Training Layers	Augment.	Epochs	Training mAP	Test mAP
M1	COCO	heads	N	10	0.834	0.857
M2	M1 (10.h5)	all	N	10	0.864	0.859

Table 1. Model performance

Table 1. shows models mean average precision (mAP). Tuned model for ALL layers without augmentation performed better than M1 model which is initialized with pre-trained weights of MS COCO dataset.

Final test was done on input video and result are shown on Figure 5. As we can seen in this example, model successfully detected kayaker in each frame.

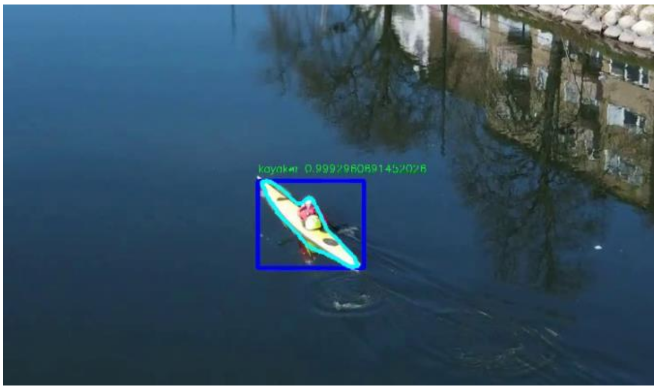

Fig 5. Detected kayaker on video

5. Conclusions

The models trained initially with *head* layers on the pre-trained model and later fine-tuned with *all layers* training.

In future work, next step is to fine-tuned model with *select layers* training with and without augmentation and to do more of hyper-parameters tuning to improve results. Collected dataset can also be expanded to achieve even better performances.

Test of images and video data produced a qualitatively noticeable good performance, in instance segmentation.

References

[1] *Abdulla, W.: Mask R-CNN for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask RCNN (2017), accessed 07-Oct-2020*

[2] Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Martinez-

Gonzalez, P., Garcia-Rodriguez, J.: A survey on deep learning techniques for image and video semantic segmentation. Applied Soft Computing 70, 41–65 (2018)

[3] Dutta, A., Zisserman, A.: The VGG image annotator (VIA). CoRR abs/1904.10699 (2019), http://arxiv.org/abs/1904.10699

[4] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)

[5] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. Lecture Notes in Computer Science p. 21–37 (2016). https://doi.org/10.1007/978-3-319-46448-0 2, http://dx.doi.org/10.1007/978-3-319-46448-0_2

[6] Zhao, Z.Q., Zheng, P., Xu, S.t., Wu, X.: Object detection with deep learning: A review. IEEE Transactions on Neural Networks and Learning Systems 30(11), 3212–3232 (2019)

[7] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)

[8] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2961–2969 (2017)

[9] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al.: Speed/accuracy trade-offs for modern convolutional object detectors. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 7310–7311 (2017)

[10] Torralba, A., Russell, B.C., Yuen, J.: Labelme: Online image annotation and applications. Proceedings of the IEEE 98(8), 1467–1484 (2010)

[11] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P.,Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision. pp. 740–755. Springer (2014)