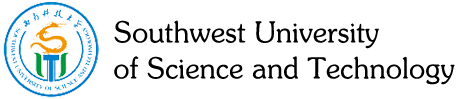
****

**MD MAHMUDUL HASAN TAREQ (阿里)**

**STUDENT ID: 4420190083**

Academic Report for

Image Segmentation

Instruction Teacher: He Gang

Table of Contents

[1. Abstract 3](#_Toc93008884)

[2. Introduction 3](#_Toc93008885)

[Image Segmentation Technique 4](#_Toc93008886)

[Thresholding 4](#_Toc93008887)

[Watershed 4](#_Toc93008888)

[Region-Based Segmentation 4](#_Toc93008889)

[Edge Segmentation 4](#_Toc93008890)

[Clustering-based Segmentation 5](#_Toc93008891)

[Deep Learning-Based Methods 5](#_Toc93008892)

[Convolutional Encoder-Decoder Architecture 6](#_Toc93008893)

[Applications of Image Segmentation 7](#_Toc93008894)

[Robotics (Machine Vision) 7](#_Toc93008895)

[Medical imaging 7](#_Toc93008896)

[Smart Cities 8](#_Toc93008897)

[Self-Driving Cars 8](#_Toc93008898)

[3. Methodology 8](#_Toc93008899)

[Mask-RCNN 9](#_Toc93008900)

[Advantages of Mask R-CNN 9](#_Toc93008901)

[4. Result and Discussion 10](#_Toc93008902)

[5. Conclusion 11](#_Toc93008903)

[6. References 12](#_Toc93008904)

# 1. Abstract

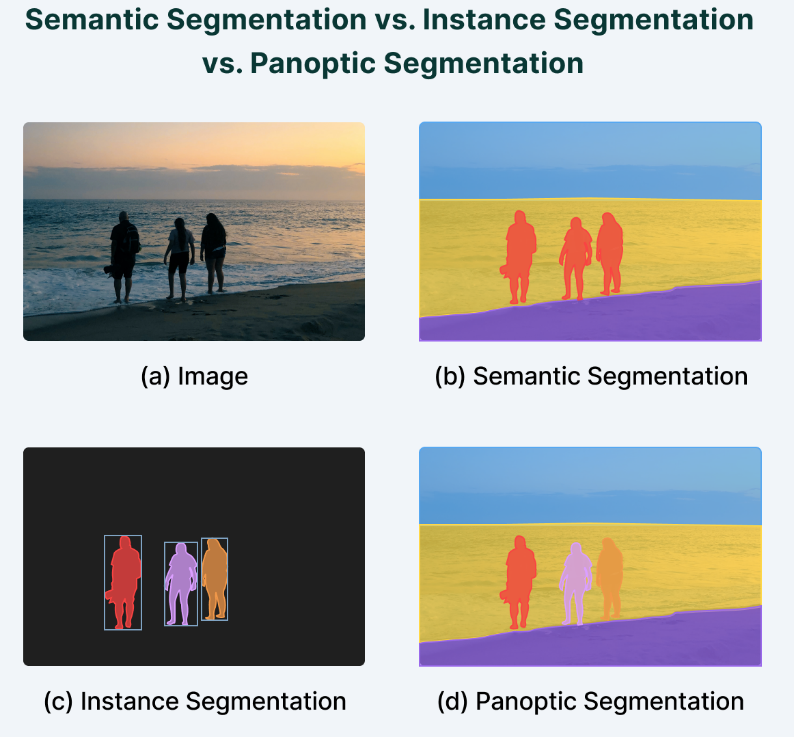
Image segmentation is the process of separating a [digital image](https://en.wikipedia.org/wiki/Digital_image) into multiple image segments in [digital image processing](https://en.wikipedia.org/wiki/Digital_image_processing). The parts into which the image is divided are called Image Objects. It is done based on the image properties like similarity, discontinuity, etc. Image Segmentation helps to obtain the region of interest (ROI) from the image. The goal of image segmentation is to simplify the image for better analysis. It is the process of assigning labels to every pixel in an image. Image segmentation has wide applications in Machine Learning, Computer Vision, AI, Medical imaging, Recognition tasks, Video surveillance, Object detection, etc. It impacts several domains, from healthcare to space science. There are two major types of Image Segmentation technique one is Semantic Segmentation and another one is Instance Segmentation. In this paper I have implemented Instance Segmentation by using mask\_rcnn\_resnet50\_fpn model with pytorch.

# 2. Introduction

Image segmentation is a branch of computer vision and digital image processing that seeks to classify related regions or segments of an image. Because the entire process is digital, a pixel representation of the analog image is provided, making the work of constructing segments the same as grouping pixels.

Image segmentation is an extension of image classification in which we do localization in addition to classification. Image segmentation thus is a superset of image classification with the model pinpointing where a corresponding object is present by outlining the object's boundary.  In computer vision, most image segmentation models consist of an encoder-decoder network as compared to a single encoder network in classifiers. The encoder encodes a latent space representation of the input which the decoder decodes to form segment maps, or in other words maps outlining each object’s location in the image.

Image segmentation tasks can be classified into three groups based on the amount and type of information they convey.

While semantic segmentation segments out a broad boundary of objects belonging to a particular class, instance segmentation provides a segment map for each object it views in the image, without any idea of the class the object belongs to. Panoptic segmentation is by far the most informative, being the conjugation of instance and semantic segmentation tasks. Panoptic segmentation gives us the segment maps of all the objects of any particular class present in the image.

## Image Segmentation Technique

There are so many common approaches have appeared in the recent literature on image segmentation. Some image segmentation method and techniques are, “Thresholding, Region based segmentation, Watershed Method, EDGE detection, Clustering, Markov random field” etc. We define each method, provide an overview of its implementation, and discuss its advantages and disadvantages.

### Thresholding

Thresholding is one of the easiest methods of image segmentation where a threshold is set for dividing pixels into two classes. Pixels that have values greater than the threshold value are set to 1 while pixels with values lesser than the threshold value are set to 0.

The image is thus converted into a binary map, resulting in the process often termed binarization. Image thresholding is very useful in case the difference in pixel values between the two target classes is very high, and it is easy to choose an average value as the threshold.

Thresholding is often used for image binarization so that further algorithms like contour detection and identification that work only on binary images can be used.

### Watershed

The watershed is a classical algorithm used for segmentation. That is, for distinguishing between different objects in an image. The gradient magnitude is calculated using a picture topographical surface. Watershed lines, which define region boundaries, correspond to pixels with the highest gradient magnitude intensities (GMIs). Any pixel encompassed by a common watershed line receives water that flows downward to a common local intensity minimum (LIM). A catch basin is formed by pixels draining to a common minimum, which indicates a segment. "Results are more consistent, detected borders are continuous," says one proponent of the watershed method. "Complex gradient calculation" is a disadvantage.

### Region-Based Segmentation

Region-based segmentation algorithms find commonalities between neighboring pixels and group them together into a single class. Typically, the segmentation procedure starts with some pixels set as seed pixels, and the algorithm works by [detecting the immediate boundaries](https://www.v7labs.com/blog/object-detection-guide) of the seed pixels and classifying them as similar or dissimilar.

The immediate neighbors are then treated as seeds and the steps are repeated till the entire image is segmented. An example of a similar algorithm is the popular watershed algorithm for segmentation that works by starting from the local maxima of the euclidean distance map and grows under the constraint that no two seeds can be classified as belonging to the same region or segment map.

### Edge Segmentation

Edge segmentation, also called edge detection, is the task of detecting edges in images. From a segmentation-based viewpoint, we can say that edge detection corresponds to classifying which pixels in an image are edge pixels and singling out those edge pixels under a separate class correspondingly.

Edge detection is generally performed by using special filters that give us edges of the image upon convolution. These filters are calculated by dedicated algorithms that work on estimating image gradients in the x and y coordinates of the spatial plane.

### Clustering-based Segmentation

Clustering algorithms are commonly used in modern segmentation procedures that rely on image processing techniques. Clustering algorithms perform better than their counterparts and can provide reasonably good segments in a small amount of time. Popular algorithms like the K-means clustering algorithms are unsupervised algorithms that work by clustering pixels with common attributes together as belonging to a particular segment.

K-means clustering, in particular, takes all the pixels into consideration and clusters them into “k” classes. Differing from region-growing methods, clustering-based methods do not need a seed point to start segmenting from.

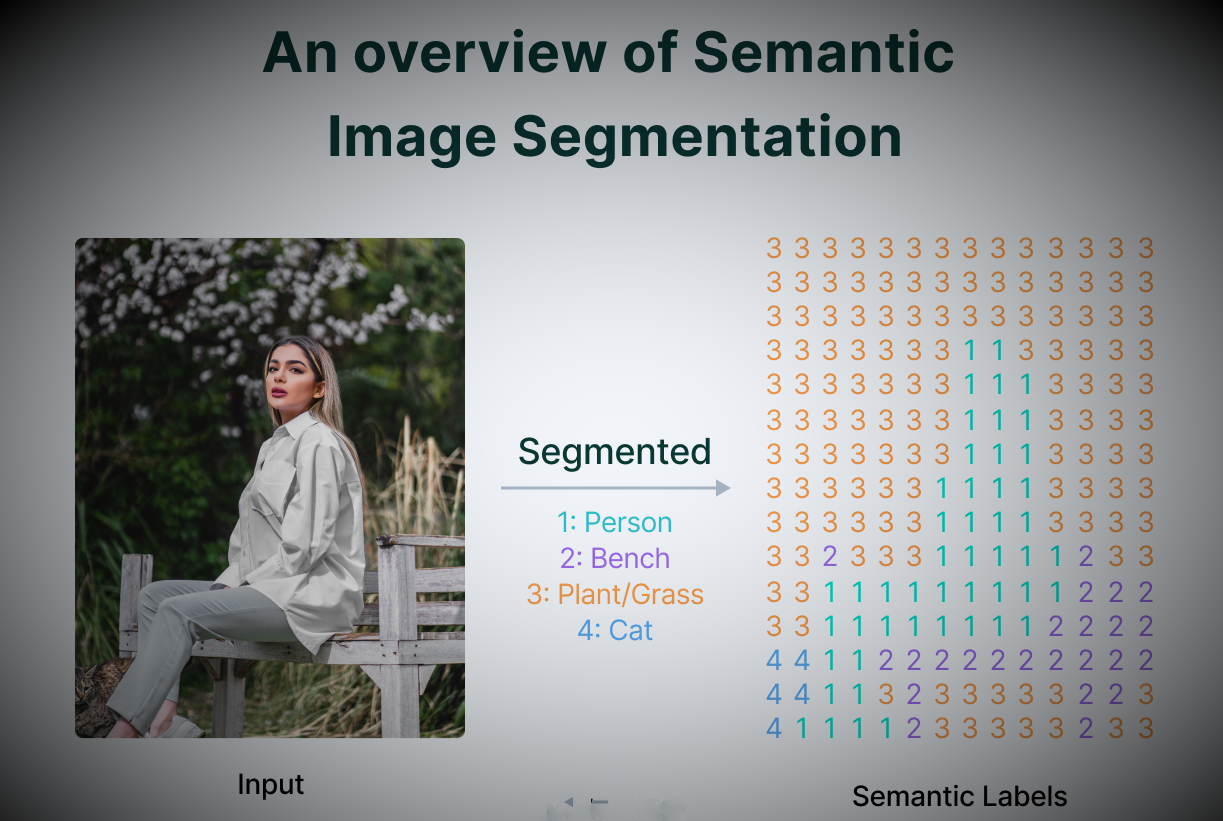
## Deep Learning-Based Methods

Semantic segmentation models provide segment maps as outputs corresponding to the inputs they are fed.

These segment maps are often n-channeled with n being the number of classes the model is supposed to segment. Each of these n-channels is binary in nature with object locations being “filled” with ones and empty regions consisting of zeros. The ground truth map is a single channel integer array the same size as the input and has a range of “n”, with each segment “filled” with the index value of the corresponding classes (classes are indexed from 0 to n-1).

The model output in an “n-channel” binary format is also known as a two-dimensional one-hot encoded representation of the predictions.

Neural networks that perform segmentation typically use an encoder-decoder structure where the encoder is followed by a bottleneck and a decoder or up sampling layers directly from the bottleneck (like in the FCN).



### Convolutional Encoder-Decoder Architecture

Encoder decoder architectures for semantic segmentation became popular with the onset of works like SegNet (by Badrinarayanan *et. a.*) in 2015.

SegNet proposes the use of a combination of convolutional and downsampling blocks to squeeze information into a bottleneck and form a representation of the input. The decoder then reconstructs input information to form a segment map highlighting regions on the input and grouping them under their classes.

Finally, the decoder has a sigmoid activation at the end that squeezes the output in the range (0,1).

SegNet was accompanied by the release of another independent segmentation work at the same time, U-Net ( by Ronnerberger *et. al.*), which first introduced skip connections in Deep Learning as a solution for the loss of information observed in downsampling layers of typical encoder-decoder networks.

Skip connections are connections that go from the encoder directly to the decoder without passing through the bottleneck.

In other words, feature maps at various levels of encoded representations are captured and concatenated to feature maps in the decoder. This helps to reduce data loss by aggressive pooling and downsampling as done in the encoder blocks of an encoder-decoder architecture.

Skip Connections were a big hit, specifically in the domain of medical imaging, with U-Net providing state-of-the-art results in cell segmentation for the diagnosis of diseases.

Following UNet, DeepLab by Facebook served as a milestone, providing state-of-the-art results on semantic segmentation.

DeepLab made use of atrous convolutions replacing simple pooling operations and preventing significant information loss while downsampling. They further introduced multi-scale feature extraction with the help of Atrous Spatial Pyramid Pooling to help the network segment objects regardless of their sizes. To recover boundary information, one of the most important parts of semantic as well as instance segmentation, they made use of fully connected Conditional Random Fields (CRFs). Coupling the fine-grained localization accuracy of CRFs, the recognition capacity of [CNNs](https://www.v7labs.com/blog/convolutional-neural-networks-guide) led DeepLab to provide highly accurate segment maps, beating methods like FCNs and SegNet by a wide margin.

Papers like SegNet, U-Net, and DeepLab laid the groundwork for future work like Mask-RCNN, the DeepLab series by Facebook, and works like PspNet and GSCNN.



## Applications of Image Segmentation

Image segmentation is an important step in artificial vision. Machines need to divide [visual data](https://www.v7labs.com/blog/computer-vision-datasets) into segments for segment-specific processing to take place.

Image segmentation thus finds its way in prominent fields like Robotics, Medical Imaging, Autonomous Vehicles, and Intelligent Video Analytics.

Apart from these applications, Image segmentation is also used by satellites on aerial imagery for segmenting out roads, buildings, and trees.

Here are a few of the most popular real-world use cases of image segmentation.

### Robotics (Machine Vision)

Image segmentation enhances machine perception and locomotion by pointing out things in their line of motion, allowing them to successfully shift courses and understand the context of their surroundings. Apart from locomotion, segmentation of images helps machines segregate the objects they are working with and enables them to interact with real-world objects using only vision as a reference. This allows the machine to be useful almost anywhere without much constraint.

* Instance segmentation for robotic grasping
* Recycling object picking
* Autonomous navigation and SLAM

### Medical imaging

Medical Imaging is a branch of computer vision that focuses on detecting diseases from visual data, including both simple visual data and biomedical scans. Segmentation forms an important role in medical imaging as it helps doctors identify possible malignant features in images in a fast and accurate manner.

Using image segmentation, diagnosis of diseases can not only be speeded up but can also be made cheaper, thereby benefiting thousands across the globe.

* X-Ray segmentation
* CT scan organ segmentation
* Dental instance segmentation
* Digital pathology cell segmentation
* Surgical video annotation

### Smart Cities

CCTV cameras are commonly used in smart cities to monitor pedestrians, traffic, and crime in real time. With the use of picture segmentation, this monitoring may be simply automated. With AI-based monitoring, crimes can be reported faster, road accidents can be followed up with immediate ambulances, and speeding cars can be easily caught and penalized.

The use of image segmentation and AI-based monitoring can thus improve the lifestyle of people.

* Pedestrian detection
* Traffic analytics
* License plate detection
* Video Surveillance

### Self-Driving Cars

Self-driving automobiles are one of the most common uses of image segmentation, with route planning and mobility strongly reliant on it. Semantic and instance segmentation helps these vehicles to identify road patterns and other vehicles, thereby enabling a hassle-free and smooth ride.

* Drivable surface semantic segmentation
* Car and pedestrian instance segmentation
* In-vehicle object detection (stuff left behind by passengers)
* Pothole detection and segmentation.

# 3. Methodology

In this paper I have worked in Instance Segmentation by using mask\_rcnn\_resnet50\_fpn model. I implemented a code in my google colab research notebook by python language. First I imported some libraries like torch, torchvision, matplotlib, random, cv2, numpy etc. Then I have trained my model with a pre-trained model mask\_rcnn\_resnet50\_fpn. Which is pre-trained by coco datasets.

Here is the Coco classes [ '\_\_background\_\_', 'person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus', 'train', 'truck', 'boat', 'traffic light', 'fire hydrant', 'N/A', 'stop sign', 'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe', 'N/A', 'backpack', 'umbrella', 'N/A', 'N/A', 'handbag', 'tie', 'suitcase', 'frisbee', 'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat', 'baseball glove', 'skateboard', 'surfboard', 'tennis racket', 'bottle', 'N/A', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl', 'banana', 'apple', 'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza', 'donut', 'cake', 'chair', 'couch', 'potted plant', 'bed', 'N/A', 'dining table', 'N/A', 'N/A', 'toilet', 'N/A', 'tv', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone', 'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'N/A', 'book', 'clock', 'vase', 'scissors', 'teddy bear', 'hair drier', 'toothbrush']

After training my datasets I took some picture for segmentation.

## Mask-RCNN

Mask RCNN, is a Convolutional Neural Network (CNN) and state-of-the-art in terms of image segmentation and instance segmentation. Mask R-CNN was developed on top of Faster R-CNN, a Region-Based Convolutional Neural Network.

There are two stages of 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. Backbone is a [FPN](https://arxiv.org/abs/1612.03144) style deep neural network. It consists of a bottom-up pathway, a top-bottom pathway and lateral connections. Bottom-up pathway can be any ConvNet, usually ResNet or VGG, which extracts features from raw images. Top-bottom pathway generates feature pyramid map which is similar in size to bottom-up pathway. Lateral connections are convolution and adding operations between two corresponding levels of the two pathways. FPN outperforms other single ConvNets mainly for the reason that it maintains strong semantically features at various resolution scales.

Mask R-CNN was built using Faster R-CNN. While Faster R-CNN has 2 outputs for each candidate object, a class label and a bounding-box offset, Mask R-CNN is the addition of a third branch that outputs the object mask. The additional mask output is distinct from the class and box outputs, requiring the extraction of a much finer spatial layout of an object.

Mask R-CNN is an extension of Faster R-CNN and works by adding a branch for predicting an object mask (Region of Interest) in parallel with the existing branch for bounding box recognition.

### Advantages of Mask R-CNN

* **Simplicity:** Mask R-CNN is simple to train.
* **Performance:** Mask R-CNN outperforms all existing, single-model entries on every task.
* **Efficiency:** The method is very efficient and adds only a small overhead to Faster R-CNN.
* **Flexibility:** Mask R-CNN is easy to generalize to other tasks. For example, it is possible to use Mask R-CNN for [human pose estimation](https://viso.ai/deep-learning/pose-estimation-ultimate-overview/) in the same framework.

The key element of Mask R-CNN is the pixel-to-pixel alignment, which is the main missing piece of Fast/Faster R-CNN. Mask R-CNN adopts the same two-stage procedure with an identical first stage (which is RPN). In the second stage, in parallel to predicting the class and box offset, Mask R-CNN also outputs a binary mask for each RoI. This is in contrast to most recent systems, where classification depends on mask predictions.

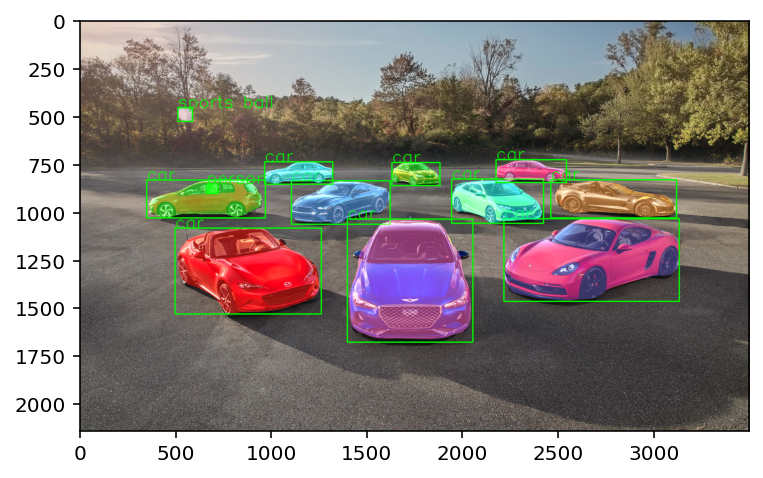
Furthermore, Mask R-CNN is simple to implement and train given the Faster R-CNN framework, which facilitates a wide range of flexible architecture designs. Additionally, the mask branch only adds a small computational overhead, enabling a fast system and rapid experimentation.

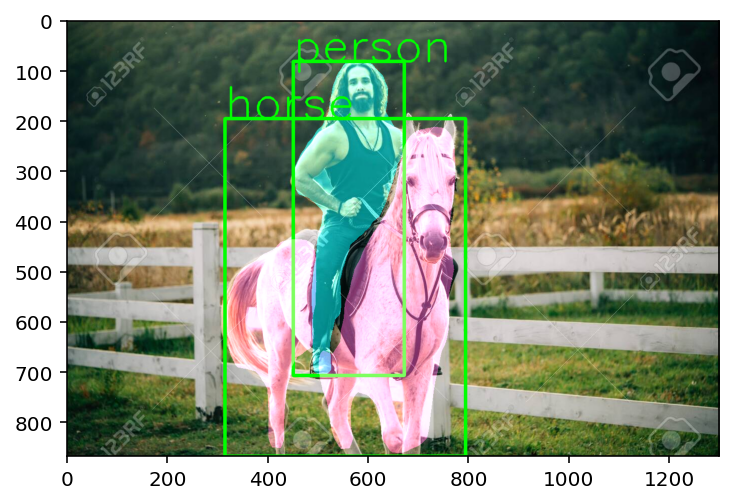
# 4. Result and Discussion

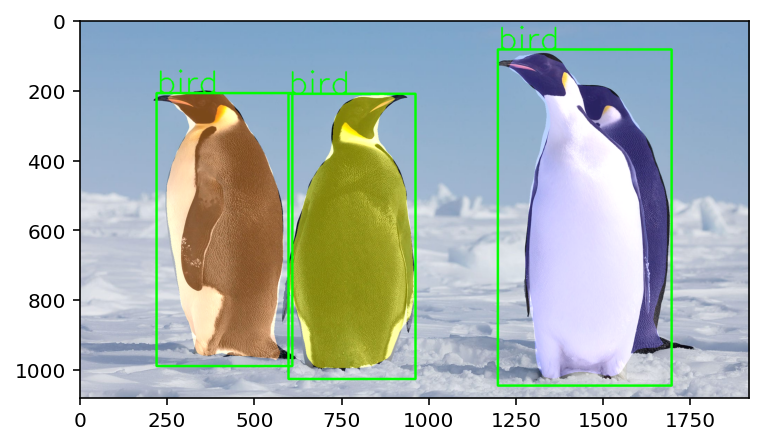
By using my image segmentation cod I got some excellent result. Here are some result with original picture.

We can see that in this image our algorithm detect the car perfectly and segmented all the car with different color. This is pretty good result as our expectation.

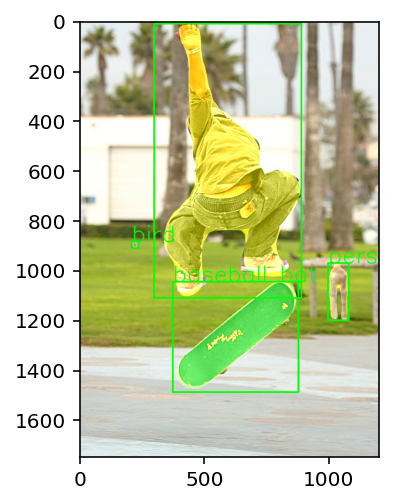






For this second picture result is perfect. But below for third and fourth picture result is not the same as our expectation. In the third picture it detected penguin as a bird because in our datasets there is no instance penguin. And in the fourth image the person is detected as a bird because this person’s image is like a bird and skateboard detected as a baseball bat which is wrong though we have skateboard instance in our datasets. But segmented correctly.





# 5. Conclusion

In this paper I have reviewed some image segmentation technique and I worked with a segmentation called instance segmentation. I tried to do instance segmentation with mask\_rcnn\_resnet50\_fpn model with pytorch. My purpose was to detect instance and segment them. My algorithm can detect some instance perfectly and do segment nicely but for some instance it is not able to detect correctly. In the next paper I will try to improve my algorithm and I will try to do segmentation with different model and different method. I have learned so many interesting things in this research. Image segmentation is very important in this age of artificial intelligence. In the future, image segmentation will use many fields like medical imaging, automation car, robotics etc. I’m very interested to do research in this topic.

# 6. References

1. He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969).

2. Johnson, J.W., 2018. Adapting mask-rcnn for automatic nucleus segmentation. *arXiv preprint arXiv:1805.00500*.

3. Haralick, R.M. and Shapiro, L.G., 1985. Image segmentation techniques. *Computer vision, graphics, and image processing*, *29*(1), pp.100-132.

4. Yanowitz, S.D. and Bruckstein, A.M., 1989. A new method for image segmentation. *Computer Vision, Graphics, and Image Processing*, *46*(1), pp.82-95.

5. Pham, D.L., Xu, C. and Prince, J.L., 2000. Current methods in medical image segmentation. *Annual review of biomedical engineering*, *2*(1), pp.315-337.

6. Pal, N.R. and Pal, S.K., 1993. A review on image segmentation techniques. *Pattern recognition*, *26*(9), pp.1277-1294.

7. Milletari, F., Navab, N. and Ahmadi, S.A., 2016, October. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)* (pp. 565-571). IEEE.

8. Norouzi, A., Rahim, M.S.M., Altameem, A., Saba, T., Rad, A.E., Rehman, A. and Uddin, M., 2014. Medical image segmentation methods, algorithms, and applications. *IETE Technical Review*, *31*(3), pp.199-213.

9. Patil, D.D. and Deore, S.G., 2013. Medical image segmentation: a review. *International Journal of Computer Science and Mobile Computing*, *2*(1), pp.22-27.

10. Shu, J.H., Nian, F.D., Yu, M.H. and Li, X., 2020. An improved mask R-CNN model for multiorgan segmentation. *Mathematical Problems in Engineering*, *2020*.

11. Vuola, A.O., Akram, S.U. and Kannala, J., 2019, April. Mask-RCNN and U-net ensembled for nuclei segmentation. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)* (pp. 208-212). IEEE.

12. Bolya, D., Zhou, C., Xiao, F. and Lee, Y.J., 2019. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 9157-9166).

13. Yang, L., Fan, Y. and Xu, N., 2019. Video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 5188-5197).

14. De Brabandere, B., Neven, D. and Van Gool, L., 2017. Semantic instance segmentation for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 7-9).