# Homework 3- Segmentation

Algorithms and Applications in Computer Vision – 046746 – Spring 2021

Students: Joshua Melamed 332708007 | Zahi Cohen 204525612

## Part 1 – Classic Vs. Deep Learning-based Semantic Segmentation

### 1.1

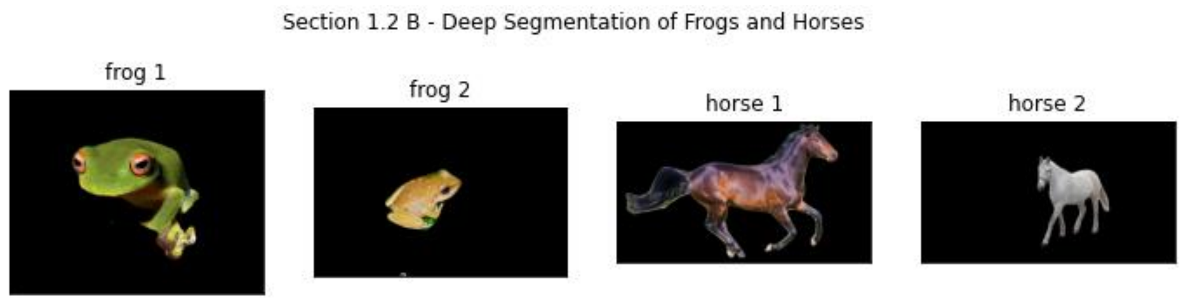
In the figure below, one can see the images from the ./data/frogs and ./data/horses folders:

Section 1.1 - Frogs and Horses


### 1.2

We chose to use **GrabCut** as our classic segmentation method, and **DeepLabV3 RESNET 101** as our deep learning-based method. We applied the two segmentation methods on the supplied images, and the results after GrabCut segmentation and deep learning-based segmentation can be seen in the first and second figures below, respectively.

Section 1.2 A - Classical Segmentation of Frogs and Horses




As mentioned above, the classic method that we chose was GrabCut. This algorithm requires the user to first select a rectangular bounding box which approximates the desired foreground region. The algorithm defines a graph from a supplied image whose nodes contain the pixel label of each pixel and whose edges which connect other pixel nodes contain the similarity cost. The graph's edges which connect nodes to two other nodes (representing label 0 and label 1) contain the cost to assign each pixel to the opposite label. The graph cuts algorithm is then applied to the graph, and then the unary potentials are recalculated. These steps are repeated until convergence is achieved.

The advantage of this classical segmentation method is that it is very quick, and still manages to provide decent segmentation. However, this method has two main disadvantages. First, it requires user intervention and thus is difficult to use when large numbers of images must be segmented. Second, as seen in the results above, the quality of the segmentation is poor when compared with other more advanced methods.

The DeepLabV3 RESNET 101 model is a deep learning model which utilizes cascaded and parallel atrous convolution modules to segment images. The atrous convolution modules have the effect of increasing the field of view of the filters, without increasing the number of weights. The model is an improvement on the DeepLabV2 model, and it adds 1x1 convolutions and batch normalization to further improve performance.

The main advantage of this segmentation method it that is achieves extraordinary performance on a wide variety of input images, as can be seen in the cases of horse 1 and horse 2 in the figure above. The primary disadvantage of this method is its higher time and space complexity. We discovered that segmentation with this model takes much more time to complete than segmentation using the classic approach. Furthermore, it has higher memory usage, which may require powerful hardware (i.e. GPU), if many images are to be segmented at a time.

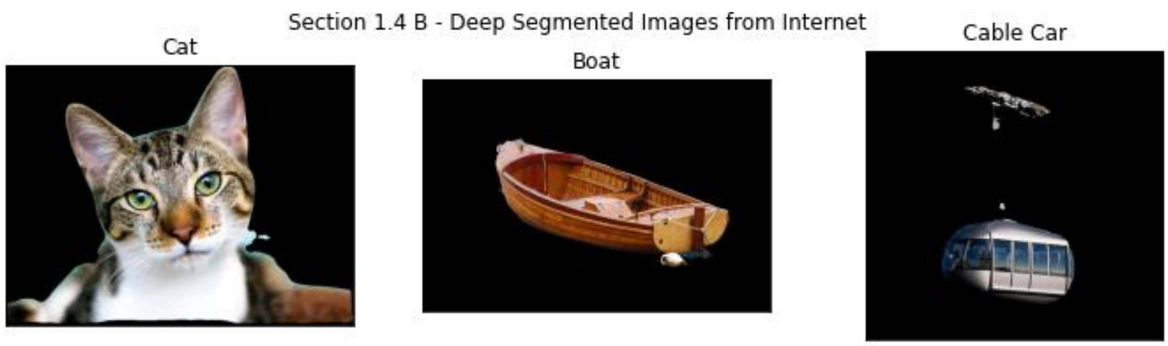
## 1.3

In the figure below, one can see the three images that we chose. For a living being, we chose an image of a cute cat. For a commonly used object, we chose a boat. For a not-so-commonly used object, we chose a cable car (like those which have just been installed at the Technion).

Section 1.3 - Images from the Internet

## 1.4

In the figures below, one can see the results of applying both the classic and deep learning-based segmentation methods on our chosen images.


We can see that the segmentation results of the deep learning-based method on the images of the cat (living being) and the boat (commonly used object) are better than those of the classic method on the same images. However, the classic method performs better on the image of the cable car (not commonly used object). We believe that this is because the deep-learning model relies on information learned during the training phase. We suspect that the model performs poorly on the image of the cable car because it likely was not exposed to many – if any – similar images when it was trained. We believe that if the deep-learning model would be trained on many different images of cable cars, it would achieve better segmentation results that even the best performing classic method.

## 1.5

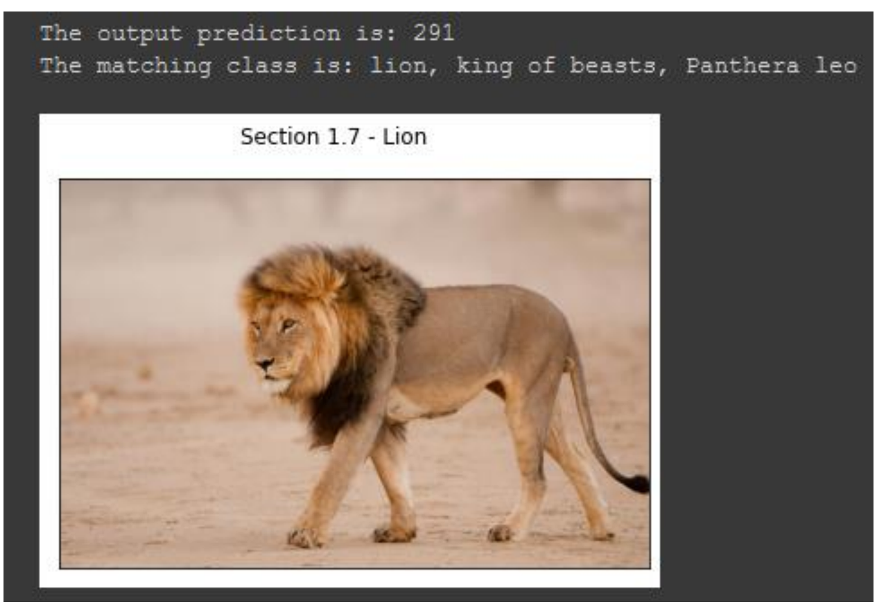
We believe that the reason why segmentation can be rough around the edges is that sometimes the edges are not clearly defined and therefore the segmentation algorithm cannot correctly determine the real outline of the object. We think that this problem can be alleviated by applying a form of preprocessing: image sharpening. This will exaggerate the edges that define the outline between objects and may lead to more accurate segmentation. We propose to implement this preprocessing by applying Laplacian filtering to the image of the form: $f[m,n] - \alpha\nabla^2 f[m,n]$, where $f[m,n]$ is the original image, and $\nabla^2 f[m,n]$ is the Laplacian filtered image, and $\alpha$ is a positive scalar.

## 1.6

We chose to load the same pre-trained VGG16 classifier that we used in the previous homework assignment.

## 1.7

We picked an image of a lion in the African wilderness, as seen below. We fed the image to the pre-trained network and the model correctly predicted that the image depicted a lion. This is not surprising since the image clearly depicts a lion in its natural environment, and the model has likely seen many similar images during training.



## 1.8

We chose to feed the image of the lion to the deep learning-based segmentation model. The output can be seen in the figure below. We can see that the segmentation of the model was sufficient. However, in the tail area, the algorithm had difficulty distinguishing between the lion's tail, and the background. We assume that this is because the background color is very similar to the color of the tail.

Section 1.8 - Deep Segmented Lion

## 1.9

We decided to make the lion swim under the ocean. To do so, we first chose an image containing a scene from under the ocean. We created a mask by marking all pixels of the background of the segmented lion image as white, and then taking the negative of that image. We then resized the new background image to be the same size as the segmented lion image. Finally, we used to mask to superimpose the new background on the segmented lion image. The resulting lion can be seen in the figure below.

Section 1.9 - Lion Under Water

## 1.10

We fed the image of the lion under the ocean to the VGG16 pre-trained network. Surprisingly, the model successfully classified the image as a lion, despite the unfamiliar background. We believe that in this case, the model was able to distinguish between the animal in the image and the image background, and therefore made the decision based on the (correctly) identified animal alone. We suspect that the fact that the ocean is a completely alien environment for a lion emphasizes the lion in the image and does not hinder the classification ability of the model.

```
The output prediction is: 291
The matching class is: lion, king of beasts, Panthera leo
```

## Part 2 – Jurassic Fishbach

### 2.1

We filmed a short video of Zahi, in which the camera is held still, but Zahi waves his hands. We then converted the video to individual frames and resized the images to be in $960 \times 540$ resolution. We modified the supplied function to not save each frame to disk, but rather to variables in memory only, which drastically reduced the function's running time. In the figure below, one can see two sample frames from the video.

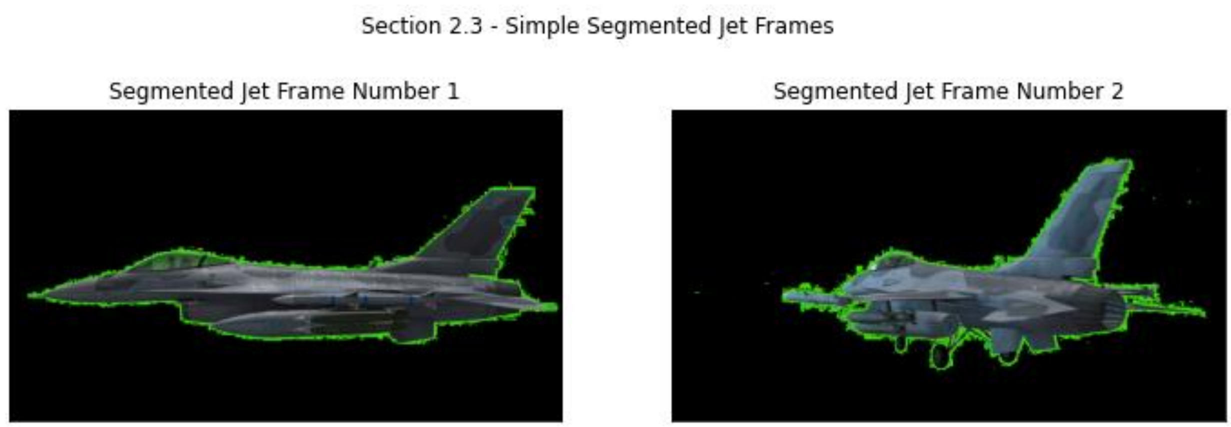

### 2.2

We chose to use the same deep learning-based segmentation method that we used in Part 1 due to its overall superior segmentation abilities. Two sample images can be seen in the figure below.

We noticed that the hand was segmented poorly – we suspect that the reason is that the hand was moved too quickly for the relatively low frame rate at which the video was sampled (15 FPS). This resulted in blurriness in the hand area, which – as discussed in Part 1.5 – negatively impacts segmentation performance.

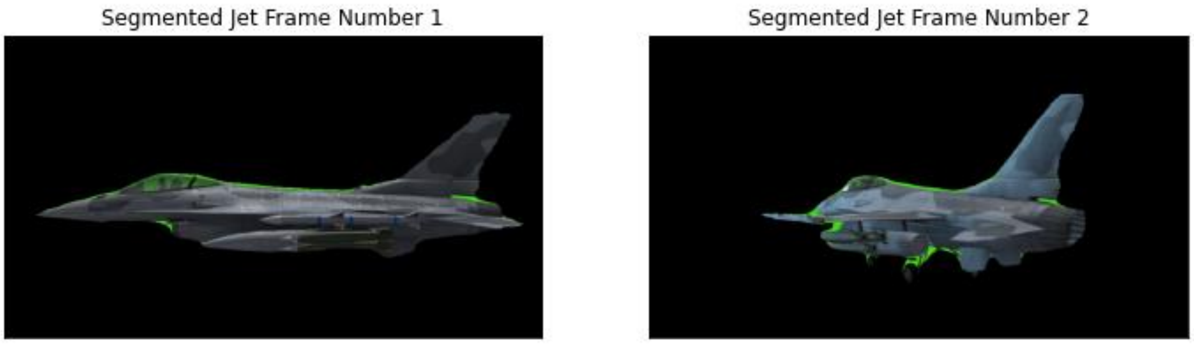
## 2.3

We picked the video of the F16 fighter jet and converted it to images. Regarding segmentation, we tried two different approaches. First, we tried to identify the color of the solid green background, and then use that color to create a mask. This method provided decent results but performed poorly around the edges of the airplane. We think this is because the outline separating the jet and the background is not sharp. The results of this segmentation method can be seen in the figure below.




Next, we attempted to segment the images using the same deep learning-based method that we used in the previous sections. This method achieved better results on images of the plane in which most of the profile of the plane was visible from the side, but poorer results on frames in which the airplane was rotated. In the end, we chose the second approach because we felt that overall, the segmentation was of a higher quality.

Section 2.3 - Deep Segmented Jet Frames




## 2.4

We chose a background image of a beautiful galaxy. Then we stitched all the frames together. We achieved this by first resizing all the frames of Zahi, the jet, and the background to resolution $960 \times 540$. Next, we created masks of all the black pixels in each of the frames of Zahi. Then, we used the masks to superimpose pixels from the frames of the jet onto the matching frames of Zahi - only in locations where no pixels of Zahi were present. This ensured that the jet would appear *behind* Zahi. Next, we created new masks in a similar fashion from these partially finished frames. We then used these masks to superimpose pixels of the background onto empty locations in the frames. These finalized frames were stitched together at 15 FPS to create a 10-second-long video. Two sample frames from this video can be seen in the figure below.

Section 2.4 - Final Video Example Frames