Computer Vision HW3

Anaelle Yasdi 203642129

Michael Novitsky 311773915

Part 1: Classic Vs. Deep Learning-based Semantic Segmentation

1.1

In this section we will display the horse and frog images supplied in this assignment:

frog1.jpg

frog2.jpg

horse1.png

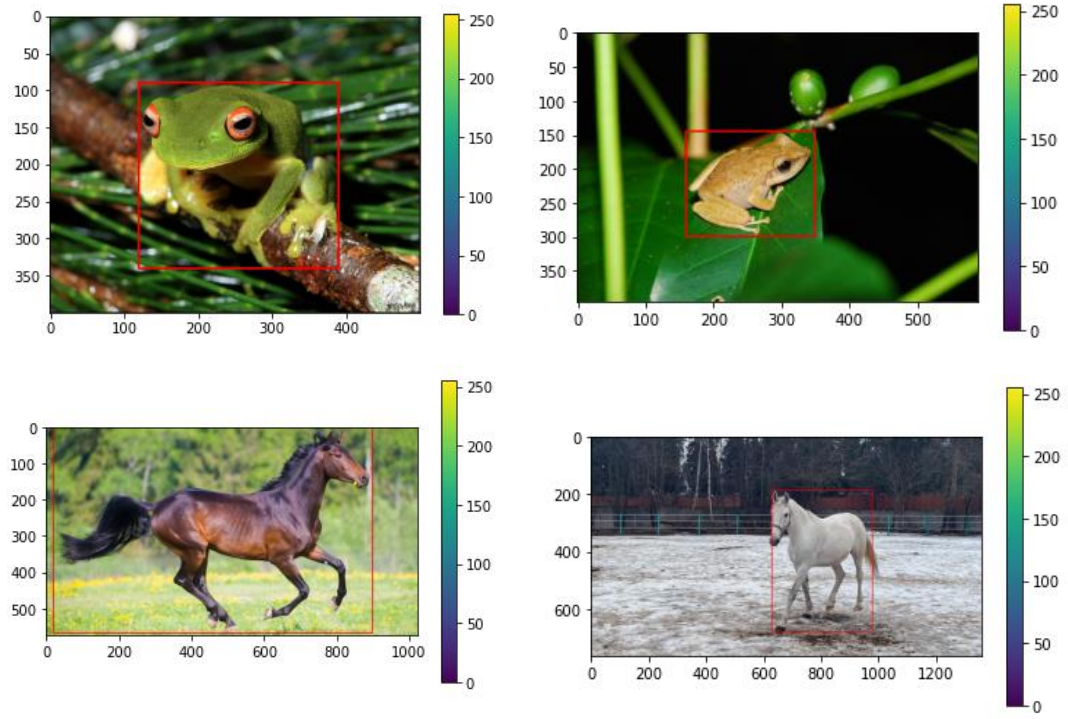
horse2.jpg

1.2

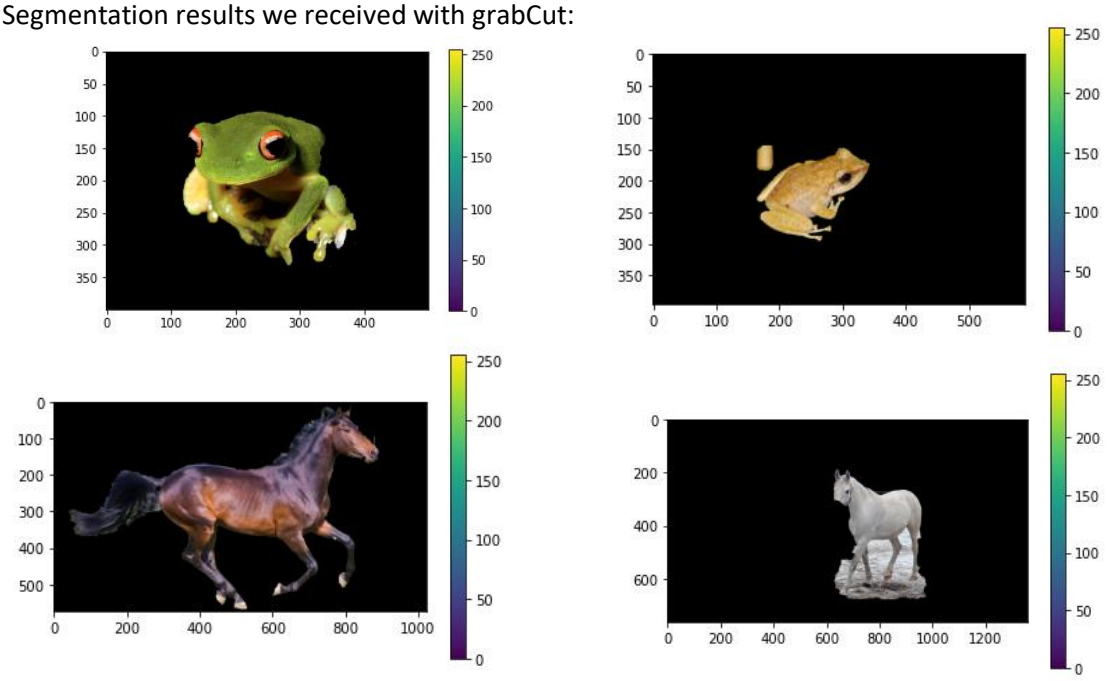
In this section we picked a classis and deep segmentation algorithm to segment the images from previous section.

For the classic algorithm we chose to use grabCut algorithm, which is implemented in OpenCV. This algorithm builds a Markov Random Fields graph using the affinity between 2 neighboring pixels in the image. The affinity between 2 pixels is defined using their difference in intensity, distances, and texture. The affinity between the pixels defines the edge connecting the 2 pixels nodes. Foreground and background nodes are added to the graph and are connected to all the pixels with an edge from the prior supplied by the user – in our case we used a bounding box rectangle. Finally, the algorithm uses min-cut max-flow to partition the graph with minimal cost.

As we explained above, we used a bounding box that we manually placed to contain the object, as the prior for the segmentation as can be seen below:



Segmentation results we received with grabCut:



We can notice that when the object and background have similar texture and RGB intensity, the grabCut algorithm struggles to segment properly around the object edges since those pixels has high affinity – this is one if the downsides of the grabCut algorithm. We also noticed that the algorithm is very sensitive to the initial bounding box. On the other hand, the algorithm converges rapidly, and gives good results on most of the images we tested.

The Deep algorithm we used was the pretrained network DeepLabV3. This network presented Atrous spatial pyramid pooling (ASPP) that uses Atrous convolution, a convolution technique that dilates the filter with zero padding to improve the filter receptive field without adding more weights. This Atrous convolution is applied on the same input with different dilation rate to detect spatial patterns. Since this network was not trained on frogs or horses, we used the background label to create the mask – anything that was not labeled as background was detected as an object.

Segmentation results we received with DeepLabV3:

The network successfully segmented the objects and outperformed the classic method, but it was heavier computationally both in memory and runtime. We can see that for horses the results are much better since the DeepLabV3 model that we used was trained on 20 classes that include horses and didn't include frogs, but still we can see that this network learned pretty well how to detect the background in frogs' pictures so segmentation results for frog images is pretty good.

1.3

In this section we picked 3 different objects: Fly, car, and spacecraft.

car.jpeg

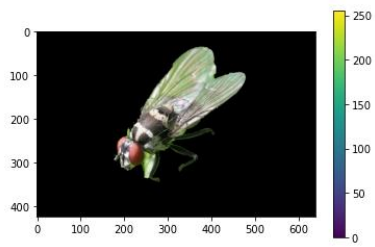
fly.jpeg

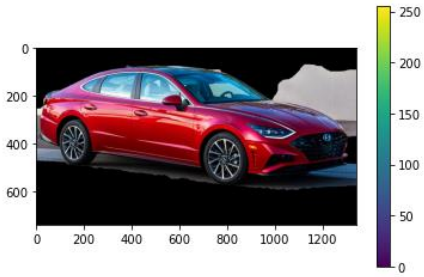
spacecraft.jpeg

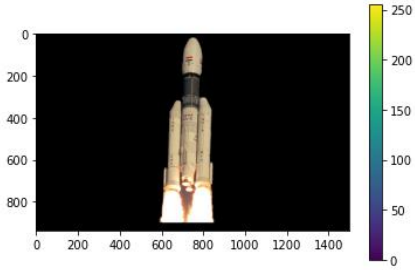
1.4

We applied our segmentation method on the images from above.

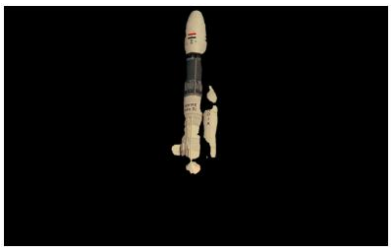
Segmentation results we received with grabCut:







The deep network results:



We can notice again that the deep network performed very well on image of an object that it was trained on like a car, but performed poorly on the fly and spacecraft.

GrabCut method performed very well on the fly and spacecraft images, but in the car image it added some background on the top right.

The grabCut method received additional information about the bounding box surrounding each object, which was not supplied to the DNN and this fact explains the good results (without specifying the bounding box we got much worse results), but it also required from us to manually annotate each image before running grabCut.

GrabCut doesn’t require information about what is the class that we try to segment and it does not have training time like the DNN, which can be very long. Training the DNN required a

dataset that was manually annotated and this is very expensive, but if we want to segment many images and we have enough compute, it is easier with the DNN since it does not require manual annotation at test time like grabCut.

1.5

We noticed that around the edges, segmentation result contains some background in the mask. We suggest using a post processing morphological operation. In the second part of this assignment we used erosion on the output masks to smooth the edges and eliminate the background around them. We noticed that when using erosion, the segmented object is less rough around the edges and it has a more “natural” look.

1.6

We used the VGG16 network like we did in the previous assignment.

1.7

Result after inference:

The model successfully classified this bird as lorikeet as we expected.

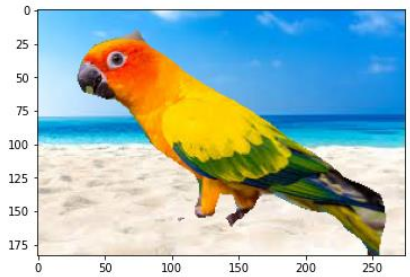
1.8

We segmented the bird using the deep network DeepLabV3:

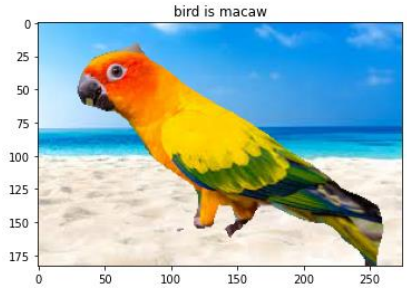
The bird was successfully segmented as expected since one of the classes this network was trained on is “bird” which is a superclass of Lorikeet.

1.9 – 1.10

We composed the segmented bird on an exotic beach image:



Result after inference:



We can notice that the network still classified the image as a parrot, but a different one – Macaw. We think that the reason of misclassification is because some of the Macaw images in Imagenet probably contained a background with a beach and Lorikeet images had different background. We tried to look for images of Lorikeet and Macaw in Imagenet, but the site was under maintenance, so we looked for images of Lorikeet and Macaw on google and saw that Macaw had few images with beach in their background and Lorikeet didn't. If most of the images of lorikeet do not contain a beach in the background, the classifier will learn that the presence of a beach in the image indicates that the object is not Lorikeet.

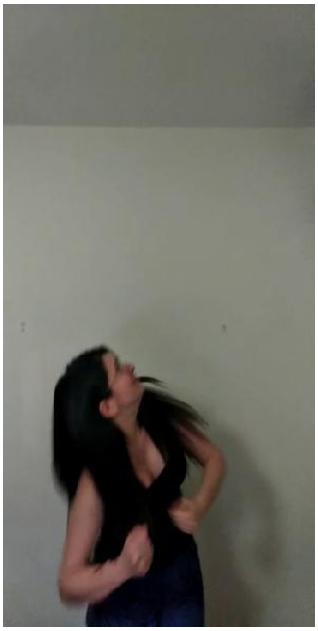
Every dataset (including Imagenet) has finite number of samples that represent each class, and thus it has an internal bias which causes models that train on this data to be also biased.

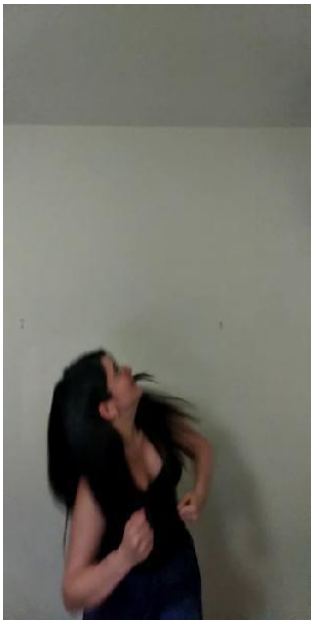
Part 2: Jurassic Fishbach

2.1

In this section we filmed 2 short movies – one of Michael, and another one of Anaelle. In this video we appear running away from something scary thing that is chasing us. We converted the images to frames and resized it to different sizes to make us look the same size in the final video. Michael's frames were resized to 400x200 and Anaelle's frames to 600x300

Two Frames from Anaelle's video:



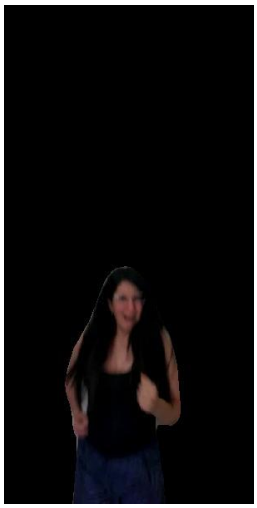


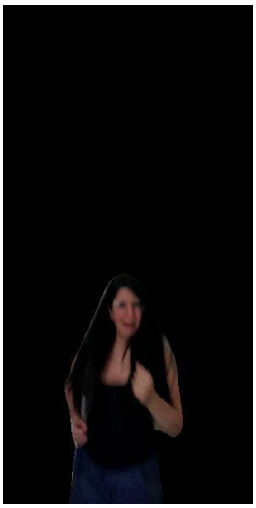
Two Frames from Michael's video:

1.2

We used DeepLabV3 DNN from previous sections to segment ourselves from the background. Since this network was trained with “person” images, we expected the results to be good. As we explained in section 1.5, we pre-processed the segmentation mask using erosion with a kernel size (7,7) to erase the background around the edges.

Two segmented frames of Anaelle (after mask with erode):





Two segmented frames of Michael (after mask with erode):

1.3

In this section we chose to work with the dinosaur video. We picked the interesting part for our short video (seconds 9-16) and saved the relevant frames. Since the dinosaur background has a uniform color, we segmented it using threshold operations. We noticed that the background RGB is (69 ,255, 8), so everything in the range [(50, 200, 0), (80, 255, 20)] was set to zero as the background. This segmentation method was very fast and showed good results. In order to improve those results (some green background was still around the dinosaur edges) we used again erosion with kernel size (7, 7).

Two segmented frames of dinosaur (after mask with erode):

1.4

In the final part, we picked the background image and for every frame in the video we padded the masked object with zeros to match the background image size, and moved the segmented object to the desired location in the final image. Finally, we stitched the object to the background image. We first stitched the dinosaur and then Anaelle's and Michael's photos for every frame in the video. The final result is a video of us running away from the dinosaur in Jurassic park.

Frames from final video: