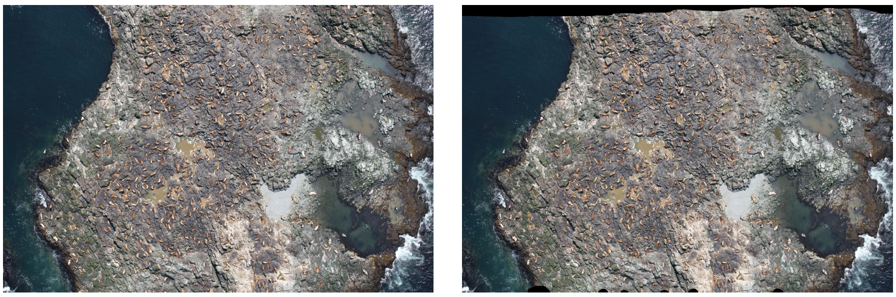
0.1 Steller Sea Lion Dataset

The goal of Steller Sea Lion Count competition is to estimate each type of sea lion in each one of test images. The different types of sea lion are: adult males, subadult males, adult females, and pups. There are totally 947 training images and 18639 testing images. For each training image, we have two versions: the original one and the one with colored dots in the center of each sea lion. Different images may have different sizes but all the sizes are around 4500x3000x3, thus the image is quite large occupying around 5MB space. The high resolution images bring us two major problems. First, during training we need to deal with memory consumption and we can not have a large number of batch size, otherwise they won't fit into GPU memory. Second, in the testing phase, we need to have short inference time due to huge amount testing images. In figure 4.1, a sampled training image pair is provided. Different color indicates different sea lion types:



(a) Original image (b) Labeled image

Figure 1: Training image pair

- red: adult males
- magenta: subadult males
- brown: adult females
- blue: juveniles
- green: pups

As we can see in the figure, there are some black regions in the labeled images. The black regions are added by the data provider in order to filter out controversial sea lions. Another thing to notice is the number of sea lions in each image. In the image pair example above, we have around 900 sea lions but the number of sea lions varies a lot in different images. More specifically, we can have only one or two sea lions in some images. Also it is not an uniform distribution for sea lion types. Here is a summary of sea lion type distribution in the training dataset:

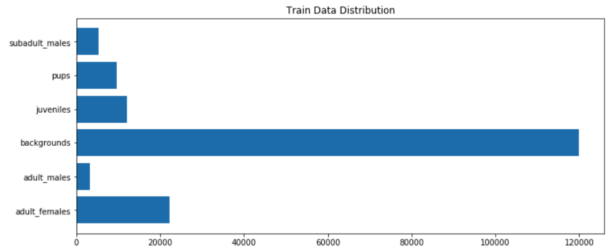


Figure 2: Sea lion types distribution

0.2 Data Preprocessing

In order to construct the training dataset, we need to do some data preprocessing. First of all, we use blob detection to find the color of each dot and its coordinates. Then we can use these coordinates to construct target maps which are used in our Count-ception architecture. Data augmentation is used to balance sea lion types and improve classification performance.

0.2.1 Blob Detection

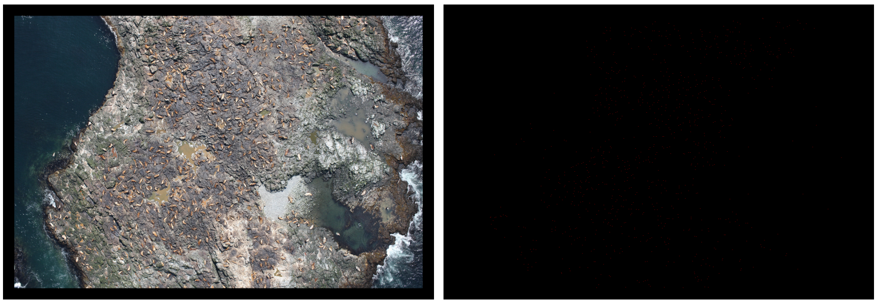
In computer vision, blob detection methods are aimed at detecting regions in a digital image that differ in properties, such as brightness or color, compared to surrounding regions. Informally, a blob is a region of an image in which some properties are constant or approximately constant; all the points in a blob can be considered in some sense to be similar to each other. The most common method for blob detection is convolution.

A dot in labeled images is a blob which contains the same pixel values, thus we can use blob detection to find the center coordinates of the dot. After we get the center coordinates, we can use R, G, B values to decide the color and get the corresponding sea lion type. Luckily we don't need to implement blob detection algorithm from scratch, there are many open source implementations for this algorithm and in this thesis we use the version provided by opencv.

0.2.2 Target Map Construction

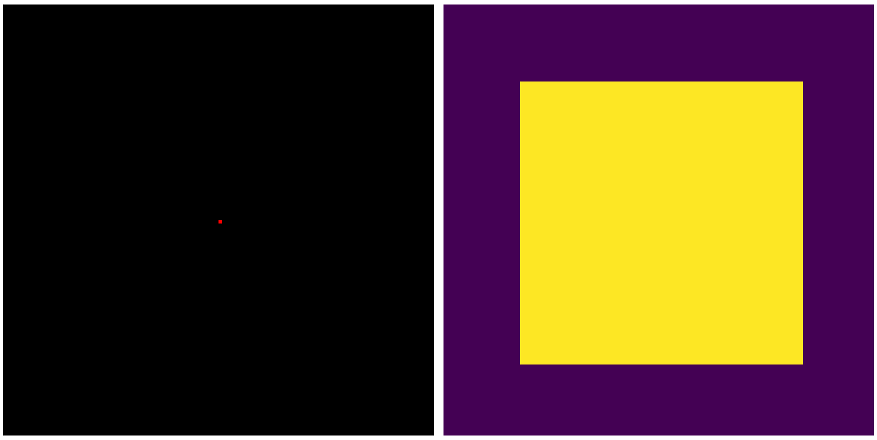
In order to make Count-ception architecture work, we need to construct the target map manually. After we get the coordinate for each sea lion, we can construct a virtual image indicating positions of each sea lion. The virtual image has the same dimension as the original image and has 255 pixel value at each sea lion coordinate, all the other pixel values are zero.

As mentioned before, target map network is a simple CNN architecture with one convolution layer. The target construction network is used on the



(a) Original image (b) Virtual image

Figure 3: Preprocessed image



(a) Virtual image (b) Activation map

Figure 4: Receptive field

virtual image in order to get the target map which is a heat map indicating positive object area. The convolution operation is implemented by pad zero and stride one and the filter is filled with ones. When we convolve the filter with the virtual image, we will get a positive value if the object dot is inside the filter, otherwise we will get a zero. Let us see an example, we have a virtual image of size 120x120 and there is an object dot at center of the image. After convolution, the virtual image turns into target map.

Recall that in the labeled images, we have a colored dot at the center of each sea lion and when we design the target map network, we need to pay attention to the filter size. We call the filter is activated when we get a positive output. In the example above, the filter size is 48x48 and it is activated when the red dot hits the right bottom corner of the filter. It is easy to see that the activation area in the virtual image is twice the size of the filter minus one, as illustrated in figure 4.7. When we try to design

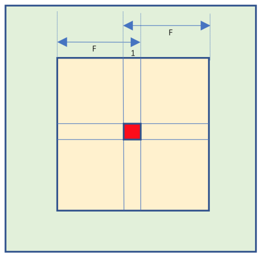


Figure 5: Activation area

the filter size, we need to keep in mind that the object should lie inside the activation area. Note that the original image and virtual image has the same size. In Steller sea lion dataset, the largest type of sea lion is the adult male which is around 96x96 pixels, so we can make the filter size equals to 48.

0.2.3 Data Augmentation

Data augmentation is a common technique used to improve the performance of neural networks. Overfitting is a challenge for deep learning models due to the lack of data and complex structures. We can not simplify the deep learning architecture when we have a non trivial problem but we can increase the training data by data augmentation. Given an input image, we can rotate, flip and zoom it to enrich our dataset. In Steller sea lion dataset, we use data augmentation to balance the sea lion types and improve classification performance.