Using a Convolution Neural Network in the Detection and Enumeration of Stellar Sea Lions from Aerial Photographs in the Western Aleutian Islands

Drew Kristensen
University of Puget Sound
dkristensen@pugetsound.edu

Patrick Ryan
University of Puget Sound
pryan@pugetsound.edu

Abstract

We trained a convolution neural network on examples of both sea lions of all types as well as on sections of the images that didn't pertain to sea lions. On our test data, we achieved a 10% error rate whereas on the raw images themselves, we had around 40% error rate. Our network consisted of three convolution layers, each paired with a pooling layer, and two hidden layers with 256 and 128 neurons respectively.

Introduction

Currently, the NOAA has to employ technicians to enumerate and categorize the sea lions that appear in the images. From this, the NOAA is able to track the status of the sea lion populations in the areas they are focused on. However, given the time sensitivity of these measures - as sea lions can move from location to location - we seek to tackle the problem of enumerating the sea lions in an image to assist the NOAA with their problem.

Related Works

Similar projects have been undertaken to solve more general image recognition problems, such as competing in the ILSVRC competition in 2012 to achieve first place in image recognition across 1000 classes. This project, which was undertaken in Krizhevsky, Sutskever, and Hinton, of the University of Toronto in 2012, inspired the basic sturcture of our intial structure of our network, as we hoped to use the success in classifying the wider problem in our paired down problem.

Background

A convolutional neural network is a neural network that uses convolution layers to alter the inputs in order to shift away from the original input and towards some useful translations that the networks can use. This limits the effects of random orientations of objects in images and increases the effectiveness of object classification. The training images were given as two sets of identical images except one set was marked with dots corresponding to each sea lion and its respective category. From this, we could extract examples of both the sea lions and the background from our dotted training directory.

System Description

To create our CNN, we utilized the keras framework with a theano back end. This was the quickest, easiest way we found that could get us started early. The network is composed of three convolution layers which are followed by pooling layers. This section provides our network with image transformations that have been proven to work well in generalizing images regardless of their orientation - something that this project needed to be able to handle well, given that sea lions don't all align themselves to true north at all times. The network works by utilizing the convolution layers to transform the images in some way so that by the time the neurons in the network recieve input, the convolution layers have abstracted away from the original input and hopefully are left with patterns that the neurons can be trained to recognize. To feed the entire images into our network, we utilized the sliding window approach, wehre we itterate over the entire image selecting 64x64 pixel selections of the image and testing for sea lions within that small image. Once we get the result, we move the window 48 pixels over so we look at part of the previous image to ensure we don't miss any sea lions. This continues for the entire image and what returns back is an array of the locations where the network suspects a sea lion is. To extract examples of sea lions from the marked images, we set up a program that would crop 64x64 pixel images centered on the location of the marked dots from the dotted training images, but took the actual images from the clean images to keep the dots out of our examples. To extract examples of the background, we simply generated random x,y pairs that we compared to our list of sea lions that we found, and excluded any pairs that included a sea lion in the cropped image. We trained our network on both the small and a portion of the large dataset that the NOAA provided. From the small network, we found that it performed with 91% accuracy on the test set.

Results

Our network achieved 91% accuracy on our test set, which was the accuracy we had been hoping for. For this measure, we checked how many accurately the network could separate the two classes after training, however, the images that composed the test set were also 64x64 images that were either centered on a sea lion or completely lacked a sea lion. This caused our network to report a much higher accuracy than it could perform on real world example, as we weren't searching through each pixel to evaluate whether or not it had a sea lion. While we lacked the time and forethought to write a numerical function that evaluated our accuracy on a whole image, by observation, we found that we were typically around 80% precision on the sea lions, but only 50% accuracy. From an entire image, we would usually get about half of our signaled positives on the terrain, marking false positives. Given that our network was only trained on the smaller of the two datasets, the network struggled with slides where the terrain was sea lion colored (brownish) and had some aspect of shadow running across the image. We had hoped to remedy this problem by using the larger dataset of the two, but given our internet speeds and hard drive space on our machines, it became unfeasible, as the larger file was 98 gigabytes compared to the smaller 98 megabytes. Furthermore, since we had not written a function that would evaluate the image as a whole, working a function that does this into the evaluation phase my have proved beneficial, as it could give us a more accurate measure of our performance on the actual examples we would be working on.

Conclusion

Overall, we believe this was a successful start to a larger project. With what we know now, we believe we could more successful with both our structure and our resources. What this project is a step towards is improving general image classification problems, as the difficulty comes from identifying objects that blend in extremely well with their surroundings. We hope to continue this project to both improve our experience with machine learning approaches and improve the abilities of our project.