Statistics Project 1:
Statistical Analysis of Ship Data for Object Detection

Edwin Sanchez

September 24th, 2023

Abstract

Nowadays, Deep Learning research is the main focus of many labs across the nation and the world. Researchers everywhere are trying to find new insights in this exciting technology and make an impact in the emerging field. However, newer researchers may be ”putting the cart before the horse” when it comes to breaking into this field. It is important to understand that Deep Neural Networks are first and foremost *statistical models*, and heavily rely on statistical theory as a foundation for all which it does. To help ground myself in the statistical practices necessary to do Deep Learning research in Object Detection, I take a ship dataset used in the Machine Intelligence Computer Vision in 3D (MICV3D) lab at IUPUI and perform statistical analysis on object count and and object location to get a better understanding of the data used to train the models we use in the lab.

1 Introduction

Deep Learning has become all the rage in the research world as of late. Most labs are either developing Deep Learning models or testing their usage in some capacity. For career researchers and mathematicians who have been in the field for some time, this is a simple step forward. However, for researchers not familiar with the core underlying principles and students looking to get acquainted with the field, there are prior steps that need to be taken before one can begin to study Deep Learning.

One of these foundations is Statistics. Statistics is arguably the most important foundational piece of mathematics that all of Deep Learning builds upon and is reliant upon in order to work. In order for a Deep Learning model to train, you need to give it data. What kind of data? Statistics is what gives you that answer.

You can't just throw anything into a Deep Learning model and expect it to perform on your problem. For instance, you need to have a reasonable assumption that the key to solving the problem lies in the data you have. One way Statistics can be used is to get a picture or develop an understanding of your data before it is used. In this project, I will use some basic Statistics to decide whether or not my training, validation, and test data is ready for a project in the MICV3D lab.

2 Background

In the MICV3D lab, we are using ship data, or images of different types of boats on the open ocean, to see if we can train an object detector or object tracker to find objects in an image as well as track their movements. Our focus is less on the models themselves and more on the performance of the model on our data.

Figure 1: *Three (3) sample images of ships from the training dataset.*

In the lab, we will be using object detection models that are pre-built by other researchers, whose focus was on developing object detection and object tracking models. Our focus will be doing something called *transfer learning*. This is when you take a model and its weights that were trained for one task and then apply it to another task. Usually the task you are transferring to is somewhat similar to the original task the model was trained for.

Transfer Learning is also usually done when you have only a limited set of data. In our case, this makes sense as we only have a total of 843 images (usually you would want images in the hundreds of thousands or millions to get a much more robust model). But before we do transfer learning we need to understand what data we have from a Statistical perspective. We already

have the images sectioned off into 3 subsets: train, validation, and test sets. One question is *How many images do we have in each set? How many objects?*. By answering this question, we can find out the ratio of images and objects across all 3 sets. It is also important to understand what we can about the objects themselves: *What can do we know about ships we have so far?* For instance, looking at the average size of our objects in our images will give us a better picture of what it means to be a 'ship' in the context of our problem.

Answering these questions will help us to gain a better understanding of the data before we work with it, as well as help us to draw conclusions on how an object detector may act with respect to our data. Additionally, we can get an idea of the representational power of our data before we give it to the model.

3 Methods

This section details how information was extracted from the data on hand. First, I discuss what tools were used in the project. Second, I discuss what kind of data was used to for the analysis.

For this project I followed the following procedure: **1:** I preprocessed the data to be read into `R`. **2:** I reviewed the data I had and tried to come to an understanding of what the data might tell me or what might be useful in the data. **3:** I skimmed the data with `R` to get a quick idea of what data I had. **4:** Based on my understanding of the data (2) and what I saw when I skimmed over the data, I began reviewing the data in earnest, trying to answer my questions. **5:** Finally, after reviewing the data, I drew conclusions about the data.

3.1 Tools

I used the `R` programming language to perform the statistical analysis on the data. I also used the `Python` programming language to format the data in a useful way to be imported into `R`. All code used for this project can be found in the project's GitHub repository[1]. Additionally, instructions for using the data and code in the repository are detailed in the README instructions on the repository.

3.2 Data

I collected the following data to perform analysis on. This data was collected for each of the 3 dataset splits: train, validation, and test. The data was originally in MSCOCO JSON format. The `python` script was used to extract the following information into `.csv` files that could be easily loaded into `R`.

4 Results

As mentioned previously, my data was broken into three sets: a train set, a validation set, and a test set. These sets are necessary for Deep Learning training. Therefore, it may be useful to review the spread of the data between these three sets.

To do this, I begin by looking at the image and object count in each set. Based on the data, there are 146, 193, and 154 images in the train, validation, and test sets respectively. There are 2115, 193, and 154 objects in the train, validation, and test sets respectively. However, simply looking at the number of objects and images in each set is not very helpful. Since the test and validation

[1] `https://github.com/Edwin-Sanchez2003/R-Project-1-Statistics`

sets are meant to be much smaller than the train set, we need to look at things differently to get a meaningful understanding of the data.

From the left graph below, we can see that the train data has a large number of images with only a single object in them, followed by several images with 2 objects in them. The distribution of objects in much less pervasive with the validation and train sets. The box and whisker plot on the right goes farther, showing us the density of images with object counts across the datasets. Here, we can see that our datasets aren't following similar distributions, which may lead to adverse affects in training. Specifically, one thing that may happen is that our validation data will tell us to stop training early because we stop performing well on images with object counts between 1 and 3, when our train and test data is distributed more around 1 and 6 objects in each image. We should try to rebalance the dataset to have more or an equivalent spread of object counts across all three sets.

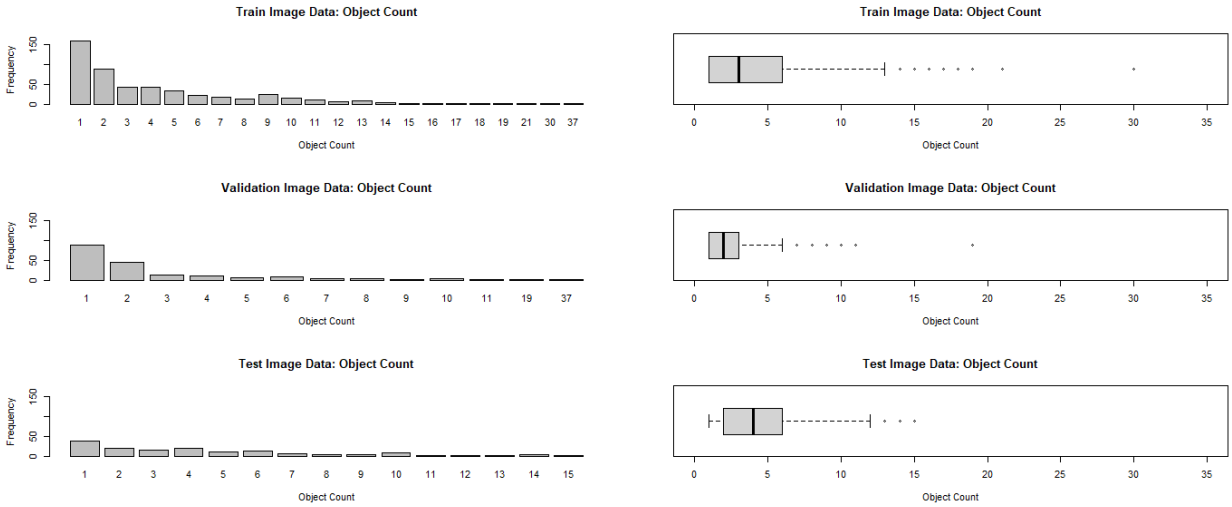

Figure 2: *Graphs visualizing the distribution of objects across images in each of the datasets. (Left) The number of images with a certain object count in each set. (Right) The distribution of the object count in images as a box and whisker plot.*

Another interesting thing to look at is where objects are in the images for each dataset. To do this, I took the center point of each object and normalized the coordinates based off of the the width and height of the image they are in. The result is the graph below. From the graphs, we can deduce that our objects tend to be more toward the top-center of the images. At the same time, there's usually a threshold where objects are much less likely to show up on the other side of, closer to the top of the image. This results in an interesting dynamic, where objects are skewed toward the top of the image, but only to a distance from the top of the image. This makes sense because most of our images are on horizon of the ocean (near the 0.4 mark on the y-axis), meaning that the ships usually rest a bit above the center of the image, but not so high that they aren't fully visible (the noticeable gap from 0.0 to 0.2 on the y-axis). The histograms in the appendix further support these understandings, from the perspective of the x and y axes independently.

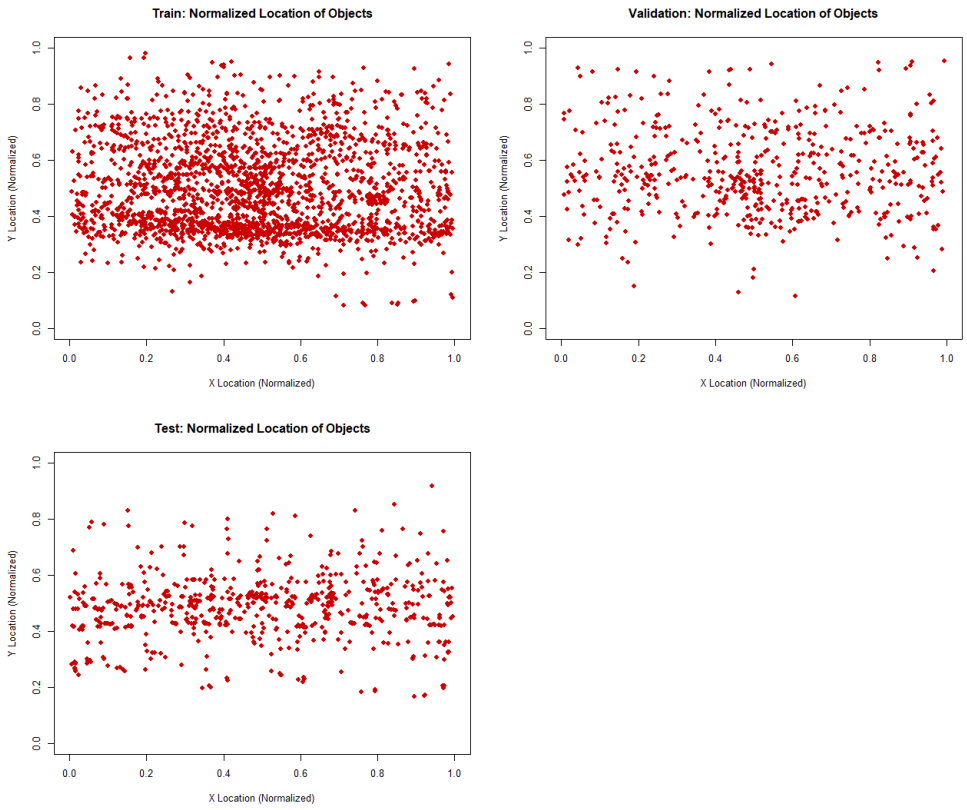

Figure 3: *Graphs showing where the centers of objects are in the images for the 3 datsets. The locations are normalized to between 0 and 1.*

5 Conclusions and Future Work

From our first look at the data, we have gotten a better picture of what kind of data we have before we use it for training. Firstly, we found out that our validation data is not evenly distributed when it comes to object count in images compared to the train and test sets, which could skew our results. We also found and interesting way to show that most ships rest just above the horizon line in our images. It may be worthwhile to augment our dataset later with more images where ships aren't along this horizon line, as our model will become more robust to objects being in other locations.

While this is a great start, there's a lot more questions to answer. For instance, what about the width and height of the objects in each set? What are the most common aspect ratios, and are they evenly distributed across our sets? This would tell us what "kind" of ships we have in our dataset - sailboats vs cargo ships, and so on as they would be expected to have different average aspect ratios for each class. Knowing what aspect ratios are most common and whether they're spread well across the sets will further tell us if our dataset splits are distributed well enough to be a good benchmark for our work.

All in all, statistical analysis is a must-use tool for all who are in the business of understanding data and using it to make worthwhile decisions.

A Data Collected

This appendix section describes the data that was collected for statistical analysis. This is split into two categories: data on each *image* in a dataset, and data on each *object* in a dataset.

A.0.1 Image Data

For each image in the dataset the following was extracted:

- **id:** The id of the image in the dataset.
- **file name:** The name of the image file.
- **object count:** The number of objects (ships) in the image.
- **width:** The width of the image.
- **height:** The height of the image.
- **area:** The area of the image.
- **aspect ratio:** The aspect ratio of the image, in text form (ex: 16:9).
- **aspect ratio float:** The aspect ratio of the image as a float (width/height).

A.0.2 Object Data

For each object in the dataset the following was extracted:

- **image id:** The id of the image in the dataset that the annotation belongs to.
- **x:** The x coordinate for the center of the object in the image.
- **y:** The y coordinate for the center of the object in the image.
- **x norm:** The x coordinate for the center of the object in the image, normalized based on the width of the image it is in.
- **y norm:** The y coordinate for the center of the object in the image, normalized based on the height of the image it is in.
- **width:** The width of the object.
- **height:** The height of the object.
- **area:** The area of the object.
- **aspect ratio:** The aspect ratio of the object, in text form (ex: 16:9).
- **aspect ratio float:** The aspect ratio of the object as a float (width/height).

B Other Fun Figures

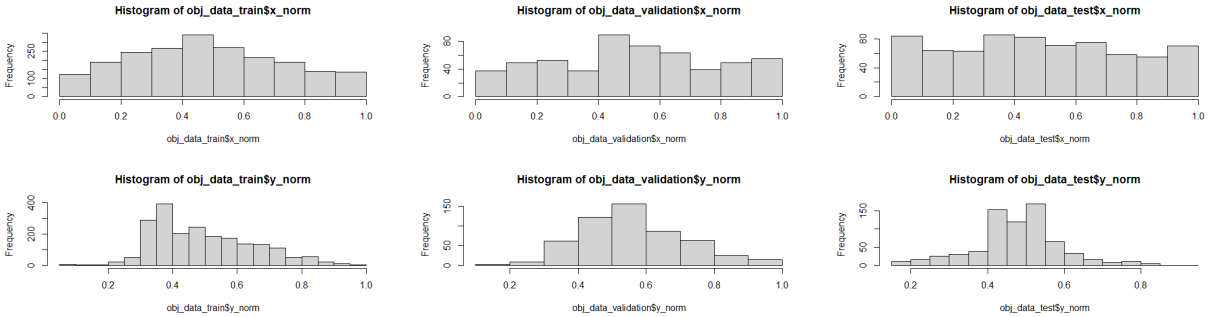

Figure 4: *Graphs visualizing the distribution of object location in the images, normalized. (Top) The distribution of the x coordinates of the objects in the images. (Bottom) The distribution of the y coordinates of objects in the images, which is a bit more skewed toward the top of center-top of the images. Note that image coordinates read the top left corner as (0,0), which means that the skewedness of the distribution is toward the top of the images.*