

# Using TensorFlow to Detect Objects in an Image

\*Note: Sub-titles are not captured in Xplore and should not be used

Jerome Mitchell  
Informatics and Computer  
Science  
Indiana University  
Bloomington, United States  
jeromitic@umail.iu.edu

Edsel Norwood  
Math and Computer Science  
Elizabeth City State University  
Elizabeth City, United States  
edselnorwood83@gmail.com

Thomas Johnson  
Math and Computer Science  
Elizabeth City State University  
Elizabeth City, United States  
thomash.johnson261@gmail.com

Reginald Kelley  
Math and Computer Science line  
Elizabeth City State University  
Elizabeth City, United States  
reggiekelley43@gmail.com

Taeyonn Reynolds  
Math and Computer Science  
Elizabeth City State University  
Elizabeth City, United States  
taeyonnreynolds@gmail.com

*The abundance of various types of satellite imagery has created a challenge with processing and analyzing data into useful information. In this project, there was exploration in development, implementation, and evaluation of a machine learning algorithm, specifically a neural network, to automate the detection of ships to track traffic in a desired port or region. A graphical approach to computation using TensorFlow, which offers easy massive parallelization and deployment to cloud was also utilized. The final result was an algorithm capable of receiving images from various sources of imagery at various resolutions and able to detect all detected ships.*

**Keywords**—*machine-learning, neural network, identification, ships, deep learning, image processing, TensorFlow*

## I. INTRODUCTION

Satellite imagery has become a valuable tool in remote sensing technology. Satellites have a vantage point of overlooking the Earth from space and extending the awareness of humanity beyond the confines of the Earth. Satellites have thus become one of the essential tools for remote sensing, offering a topographical view Earth and its surface from a considerable distance. Not only does satellite imagery provide a topographical visual of an area, satellite imagery can observe regions at various scales. This allows data to be collected and examined at various levels to identify specific phenomena at a small scale then a larger scale. Such capabilities have caused satellite imagery to become a vital component to research when examining areas that have considerable dimensions.

Machine-learning algorithms are programs that accept input of data to adapt for performing a task. The general rule with machine-learning algorithms is that the more data that is inputted, the better the performance of the machine learning algorithm. The machine-learning algorithm goes through two phases: a training phase and a test phase. The training phase acts to develop the algorithm through processing examples in the data. The machine-learning algorithm's success is

significantly affected by the conditions set for training, such as how long it will be trained, if the training will be supervised or unsupervised. Supervised training is when the dataset used for training includes the correct calculations or classifications for the algorithm to check over to adapt for higher accuracy. Unsupervised training allows the algorithm to do the classifications or calculations on its own with little human guidance afforded in the correct calculations or classifications being available. Supervised machine-learning algorithms will generally outperform unsupervised machine learning algorithms.

Two general rules of thumb can be followed to typically improve the overall performance of a machine-learning algorithm. The first is to train the algorithm through multiple iterations, with the larger number the number of iterations often leading to higher rate of accuracy for the trained model that is produced. The more iterations that are allowed for training, the more assessments of mistakes and successes that can be made to allow the algorithm to adapt a more effective model to deploy during the testing phase.

The second general rule of thumb is to utilize large amounts of data during the training. Using small amounts of data will provide a rather limited span of possibilities for the algorithm to observe during training, which will reflect in the poor accuracy of the resulting model. Using large amounts of data will offer a broader variation of the target for classification or computation, which will therefore lead to a more accurate model. The data the algorithm is exposed to during the training is highly significant as that is what defines the accuracy of the final model, which is used in the testing phase. Making sure the algorithm is exposed to accurate data is just as important as exposing the algorithm to a large quantity of data.

After the training phase when the machine-learning algorithm has adapted for better performance, the algorithm enters the testing phase in which the algorithm is set to performing the task on its own.

As the machine-learning algorithm is being tested, the model that was developed during the training is evaluated in its efficiency and accuracy. Depending on the data that was utilized in the training as well as the depth of the training, the model developed by the machine-learning algorithm for the testing phase is evaluated as the model is used in full application. The machine-learning algorithm must not fall within the two extremes. The first is having an unusually low accuracy such as 10%, which can hint to the algorithm underperforming. The second is having an unreasonably high accuracy such as 100%, which suggests the algorithm is overperforming. The machine-learning algorithm being utilized for this project is a neural network, which is modeled after the structure of the human brain. A neural network is constructed of an input layer, hidden layer, and output layer. Data enters the neural network through the input layer where each node in the input layer accepts data that will be processed through the rest of the neural network. After being stored in the input layer, the data enters the hidden layer after being multiplied by weights in the neural network to adjust for bias in the neural network's processing. The data is then stored within nodes in the hidden layer which will transfer the data to the succeeding arrangement of nodes within the hidden layer or to the output layer, each exchange the data is multiplied by weights again. Once the data reached the output layer, the results are processed and printed out for the user to see.

The purpose of this research was to employ a neural network in examining the presence of shipping vessels in provided satellite images. The satellite image has boats in various direction and locations, which could provide an issue for a simple algorithm to classify the boats from the remainder of content in the image. The neural network is tasked with identifying all the boats that are present in the image. The accuracy of the neural network will in turn display if the neural network can identify such images as boats. This will in turn provide reason as to whether the neural network will can be a useful instrument to identifying boats and possibly other targeted content.

## II. MEHTODOLOGY

### A. Datasets

The dataset being used is ShipsNet, which is constructed from the image chips that have been retrieved from satellite images that are stored at Planet. There are 100s of images that are labelled "no-ship" or "ship." Each image is 80x80 pixel RGB properties. The file type allowing the images to be analyzed by the Python code is JSON. The dataset was retrieved from <https://www.kaggle.com/rhammell/ships-in-satellite-imagery/data>, a website that hosts datasets of ships captured in satellite pictures in within the expanse of the San Francisco Bay [3].

### B. Neural Network

The neural network utilized for this research is a convolutional neural network. A convolutional neural network is a variation of the basic neural network that is created with the expectation that the neural network's inputted data will be pictures [2]. Due to the expectation of pictures being used, the

structure of the neural network can the be altered to accommodate the specificity of the input data [2]. The neural network can be streamlined in the conditions and functions that drive it [2].

The reason for using a convolutional neural network rather than a standard neural network is both for efficiency and accuracy [2]. The standard neural network can handle pictures that are 32 pixels by 32 pixels with RGB colors, but anything larger will cause the neural network to be overburdened [2]. The weights of the neural network will cause it to expand its calculations to unnecessary levels through the weights expanding in number to account for the picture dimensions and color characteristics, such as if the picture is in RGB or Grayscale colors hues [2]. The convolutional neural network avoids this issue by using a structure that limits the weights through keeping the nodes tethered to specific portions throughout each layer [2]. The entirety of each layer is not connected to every node that is available within the neural network, which makes the convolutional neural network more effective at working with pictures [2].

A convolutional neural network has four basic layers: the input layer, convolutional layer, pooling layer, and the fully connected layer [2]. The layers are established in the order listed previously, with each layer simplifying the original picture data to quantifiable data that may be processed through each successive layer until classifiable data reaches the fully-connected layer [2]. The oddity of the fully-connected layer is that this layer is the only one not restricted as the preceding layers were as its nodes will receive the data from the preceding layer with restriction to what portion of nodes get which selection of data [2]. The convolutional neural network is defined in python file model.py.

### C. Software

The TensorFlow library is a critical part of tis research project. TensorFlow is an opensource library that is used for high performance computing with data that is quantifiable [1]. TensorFlow is adaptable for usage on multiple computing platforms for using computational resources over across multiple pieces of equipment for completing high performance computing tasks [1]. TensorFlow also has a strong foundation for those wishing to utilize it for deep learning or machine learning projects [1]. TensorFlow version 1.2.0 is being used for this research.

The programming language being used to construct, run, and maintain the neural network is Python. The version required for the code being implemented for this research is 3.5 or greater. Python is an open source programming language, which grants the advantage of a multitude of accessible libraries and documentation for developing the convolutional neural network that is being used in this research project.

Python Imaging Library is a Python library dedicated to providing software-structured capabilities for handling and altering image data via the use of Python code. This library was developed in Python and the programming language C. The Pillow library is a fork of the Python Imaging Library that

is available. The version of Pillow being utilized for this research is 4.1.1.

NumPy is a math-based Python library that focuses on matrices and the mathematics that can be performed on matrices. Matrices are necessary for restructuring the original image data from the input into data that the model will be able to analyze. The version of NumPy that is being employed in this research is 1.13.0+mkl.

SciPy is a Python library designed for implementing mathematical and computational algorithms for STEM based purposes. This library allows for access to a considerable amount of mathematical functions to be utilized, many of which work in conjunction with NumPy. The version of SciPy that is being utilized in this research project is version 0.19.0.

TFLearn is the deep-learning library that is used in conjunction with TensorFlow to employ deep learning techniques within Python code. The version of TFLearn that is being utilized for this research project is version 0.3.1.

#### D. Process

The convolutional neural network algorithm must first be trained to construct a model which ship may achieve identification. In examining an image, the algorithm will use a window of 80 pixels by 80 pixels in RGB to observe sections of the image classify that section of the image as having a “ship” present or not. When multiple sections that are adjacent to one another have been classified as “ship”, those instances are treated as one, which is visually represented in the computer-generated square that appears in the image itself. After multiple iterations of the training have been completed, a model is developed because of the convolutional neural network analyzing mistakes and successes that have occurred in the training to ultimately devise a model that retains the most accuracy in identifying ships in the image data. Each time the convolutional neural network is trained, the previous model is deleted to make way for a new model. The convolutional neural network is then tested against a new image with the finished product being the ships in said image being enclosed in colored squares to be identified once the image is opened.

### III. RESULTS

The model managed to identify all but one ship in the image and misidentified two portions of land in the San Francisco Bay area as ships. The model had a high level of accuracy in classifying ships from non-ships in the image with a total of three mistakes being made.



Figure 1: The results of testing a trained model on a picture of various ships in the San Francisco Bay.

The three mistakes that have been made by the convolutional neural network display that the current models that are being generated are not up to par for the algorithm to be taken into real-world applications. Instead, the code will need to be improved upon and the model further trained before more acceptable outcomes can be acquired. Thus, research will have to continue to make the convolutional neural network algorithm far more efficient and accurate for better models to be produced.

### IV. RELATED LITERATURE

The Google Brain team’s presentation at the 12<sup>th</sup> USENIX Symposium on Operating Systems Design and Implementation was focused on the utilities that TensorFlow could provide as a foundation for immersive machine learning projects [4]. This conference submission displays the concepts that made TensorFlow such a reliable buttress for this research project considering that the machine learning algorithm is a convolutional neural network [4].

### V. FUTURE WORKS

The convolutional neural network model can be altered to allow for more layers to exist. More layers will affect the way that the data is handled from the input to the output which can in turn affect the accuracy and efficiency of the neural network. Such changes will have to be observed carefully to ensure that significant changes in accuracy or efficiency are not overlooked. The convolutional network used for this research would then be used as a baseline to evaluate whether the convolutional neural network with the alteration in the number of layers in the convolutional neural network is performing better or worse. The results of such comparisons will provide an understanding as to whether the convolutional neural networks with an increased number of layers are a viable option for research. Multiple runs of the algorithm of this research and the augmented convolutional neural network would have to take place to be able to fully measure the average performance of both algorithms.

The number of nodes in each layer can be altered to examine whether accuracy and efficiency will shift in any crucial way to suggest that altering the number of nodes can be a viable option for upgrading the performance of the

algorithm overall. Altering the number of nodes could present the similar possibilities of changing the overall performance of the convolutional neural network as altering the number of layers would. Changing the number of nodes and changing the number of layers would have to take place separately as altering both the number of nodes and the number of layers in the neural network can hide the impact that changing either the number of nodes or the number of layers could have. The two must be examined separately so that the effects of changing the number of nodes or the number of layers can be observed independent of the other, then a convolutional neural network with an altered number of nodes and layers can be developed to observe the influence that an altered number of nodes and altered number of layers will have on the neural network.

This research was focused on the usage of one machine learning algorithm which was a neural network. The research further specified the neural network that would be used to be the convolutional neural network. This is one example of the wide amount of machine learning algorithms that can be utilized. There are other variations of neural networks that can be used, k-cluster algorithms, linear regression algorithms, and a plethora of others that can be compared to the convolutional neural network that is observed here. Comparing other machine learning algorithms will be important in verifying if the convolutional neural network was the best option to employ for the dataset in this research project. Comparing machine-learning algorithms could also provide more information as to what machine learning algorithms can provide an effective baseline for future comparisons. If an algorithm outperforms the convolutional neural network that is used in this research, then further investigation can be made as to what caused this superior performance. If not, then further research can be established as to why the convolutional neural network outperformed the alternative algorithm. Comparing machine learning algorithms can provide a better understanding as to which algorithms work best under specific conditions and why whether those conditions are broad or idiosyncratic in nature.

A more nuanced approach to improving the accuracy of the models developed by the convolutional neural network would be increasing the data that the developing models are exposed to over training as well as increasing the number of iterations that the algorithm is exposed to in developing a

model. This would be the simplest to implement and is appealing to the two basic advantages that can be implemented with machine learning algorithms in general. There is a risk of giving too much data to the machine learning algorithm to the point that the algorithm's effectiveness is lowered due to being trained on to large a quantity of data. Another drawback of more intense training for the models is that to see the measure of improvement that altering the training process could have on the models that the convolutional neural network algorithm is generating will take time in future research endeavors.

## VI. CONCLUSION

In conclusion the developed algorithm was not 100% adequate in the fact that the model had three mistakes. It missed one ship and classified two non-ship objects as ships. Although the algorithm had errors the Center of Excellence in Remote Sensing Education and Research (CERSER) program at Elizabeth City State University can benefit from the use of the algorithm. The algorithm can be modified and re-trained to detect any object the user wishes with the correct dataset.

## ACKNOWLEDGMENT

Acknowledgement to Dr. Linda B. Hayden for funding this research and providing support in the resources that allowed for the research to continue throughout the semester.

Acknowledgement given to Jerome Mitchell whose guidance in this research project was invaluable to its completion.

## REFERENCES

- [1] "TensorFlow", *TensorFlow*, 2018. [Online]. Available: <https://www.tensorflow.org/>. [Accessed: 22- Apr- 2018].
- [2] A. Karpathy, "CS231n Convolutional Neural Networks for Visual Recognition", *Cs231n.github.io*, 2018. [Online]. Available: <http://cs231n.github.io/convolutional-networks>. [Accessed: 22- Apr- 2018].
- [3] Burachonok, V. and Galkissa, M. (2018). *Ships in Satellite Imagery / Kaggle*. [online] Kaggle.com. Available at: <https://www.kaggle.com/rhammell/ships-in-satellite-imagery/data> [Accessed 23 Apr. 2018].