Ship Classification by ResNet Transfer Learning

Chu Zheng
Nov 2022

Table of content

Introduction

Marine industry is an important industry sector in Singapore. Singapore's story has historically been a maritime story, one that stretches as far back as the 2nd century. Archaeologists have uncovered evidence that suggests Singapore was already a flourishing trading port more than 500 years ago. Singapore's port was and remains an important driver of Singapore's connectivity and competitiveness. The Singapore Government continues to make significant investments in our port infrastructure as part of the Next Generation Tuas Port, to ensure that Singapore retains its leading position as a global hub port.

With the advent of Industry 4.0, ships are becoming smarter while business processes get increasingly digitized. Skill sets and knowledge in areas like systems engineering, data analytics, cyber security and green shipping will thus become more important as the sector continues to transform. One of these areas where machine learning can contribute to is to monitor ship traffic by identifying ship classes from real time satellite imageries around Singapore and within the South East Asia region. In the past, ship detection models have been explored in CRISP. In this proposal a ship classification model is built from transfer learning of ResNet model with hyperparameter tuning, which improves validation accuracy from 75% to 96%, and a real time classification module is developed to aid future ship classification tasks in CRISP's GIS system and has been tested on Google Earth ship imageries with satisfactory result. An original dataset of 870 detailed ship images is used for training and validation.

About the data:

The majority of data is taken from MASATI v2 (MAritime SATellite Imagery dataset) and ShipRImageNet dataset where detailed optical satellite ship images of several common ship types are provided. However, the images are not classified nor taken without the ships' surroundings. Until the report is completed the ready ship classification datasets on the web have not emerged except for these two datasets. Other than that, open source optical satellite ship datasets are primarily collected for detection tasks, which do not require a high resolution satellite for capturing details of those ships. Hence ship detection datasets do not serve for classification purposes. To complement the two ship classification datasets above, more ship images are collected from Google Earth, where ground resolution is high.

MASATI v2 contains 1789 PNG format RGB band images of single ship with spatial resolution a few meters/pixel and image resolution 512 x 512 pixels sourced from Microsoft Bing Map. It was introduced in this paper *Automatic Ship Classification from Optical Aerial Images with Convolutional Neural Networks* in *Remote Sensing* Journal in 2018.

ShipRImageNet comprises 3433 BMP format RGB band images of scenes containing ships with spatial resolution ranging from 0.12 to 6m/pixel and image resolution around 900 x 900 pixels taken from multiple sources. The dataset was published in paper *ShipRSImageNet: A Large-Scale Fine-Grained Dataset for Ship Detection in High-Resolution Optical Remote Sensing Images* from *IEEE* Journal in 2021.

The final dataset used for the model comprises seven balanced classes, i.e. boat, bulk cargo, container, cruiser, oil tank, submarine and yacht with an average of 120 ship image samples each, as shown in the bar chart below. Other categories of ships are not included such as fishing boats (as opposed to general white ships), service ships for construction, cruise ships for holidays, etc, because the number of these

ship types is not aligned with the popular ship types. On top of that, unpopular ships might bring the problem of unbalanced dataset, which is adding bias in classification meaning the unpopular ships are likely to be misidentified as the popular ones.

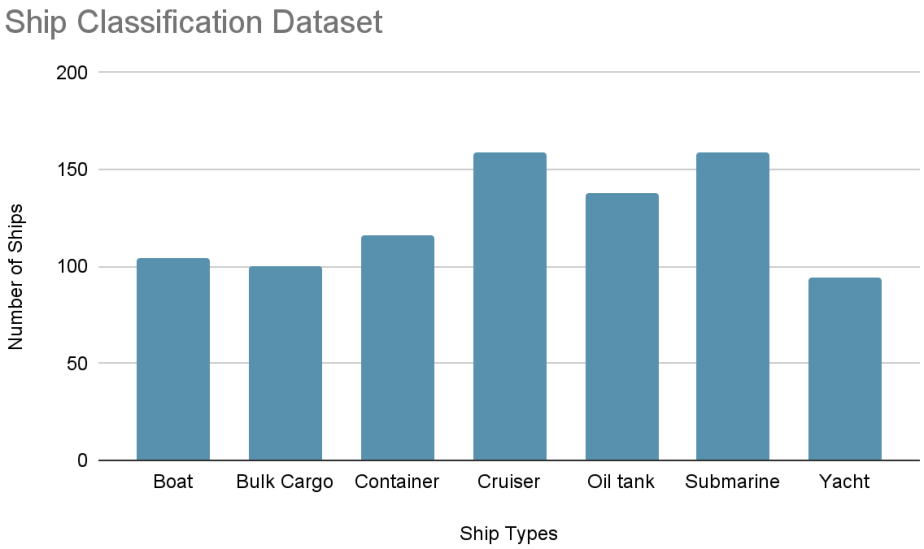


Figure 1. Number of ships in each category

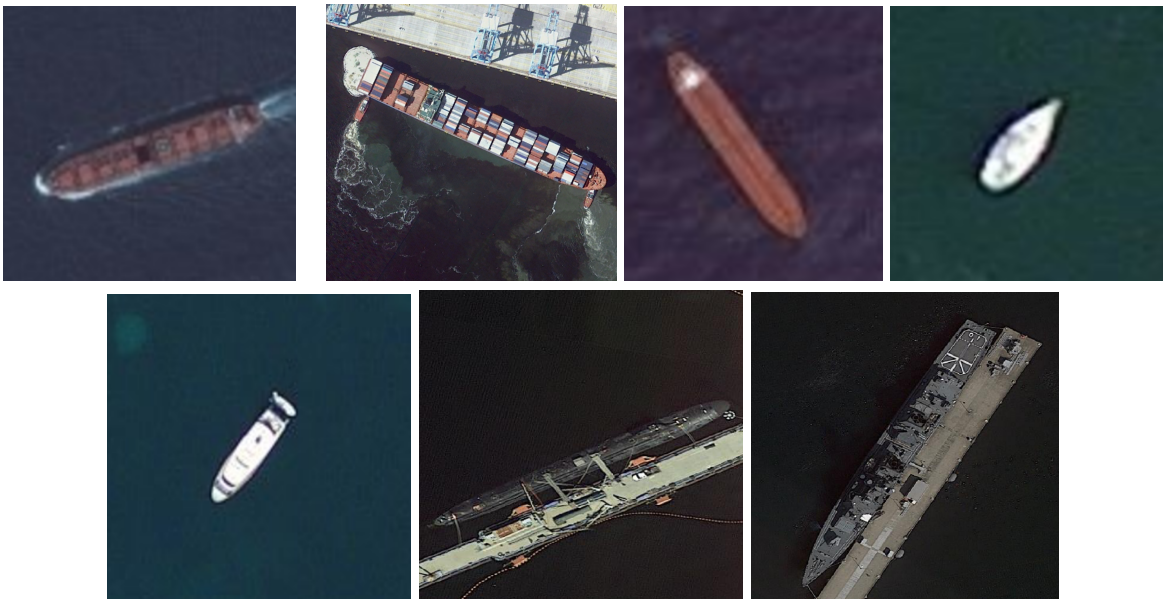


Figure 2. Examples of each ship type, left to right: bulk cargo, container, oil tank, boat, yacht, submarine, cruiser

About the model:

There are several popular models built on CNN to accomplish image classification tasks, such as VGG-16, Inception-v1 and AlexNet etc. Among them, ResNet is a common architecture used in CRISP's machine learning projects.

One of the problems ResNets solve is the famous known vanishing gradient. This is because when the network is too deep, the gradients from where the loss function is calculated easily shrink to zero after several applications of the chain rule. This results in the weights never updating its values and therefore, no learning is being performed. With ResNets, the gradients can flow directly through the skip connections backwards from later layers to initial filters.

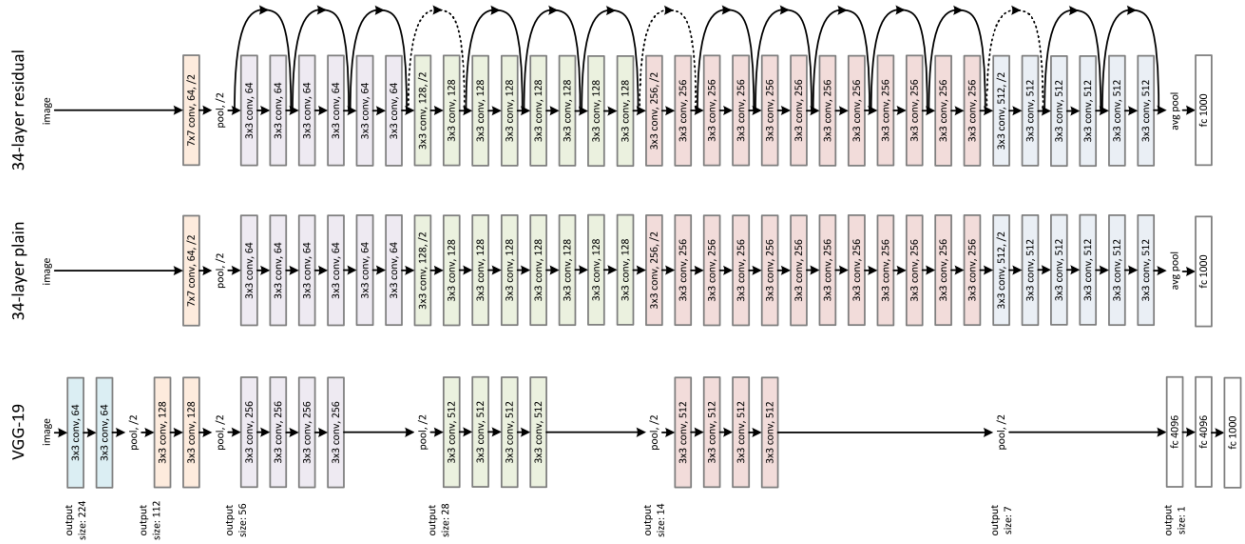


Figure 3. ResNet 34 architecture

The ResNet 34 architecture is depicted above in Figure 3. From left to right, ResNet consists of one convolution and pooling step (orange) followed by 4 layers of similar behavior. Then, each of the layers follow the same pattern. They perform 3x3 convolution with a fixed feature map dimension (F) [64, 128, 256, 512] respectively, bypassing the input every 2 convolutions. Furthermore, the width (W) and height (H) dimensions remain constant during the entire layer. Lastly, the dotted line is there, precisely because there has been a change in the dimension of the input volume (of course a reduction because of the convolution). Note that this reduction between layers is achieved by an increase on the stride, from 1 to 2, at the first convolution of each layer; instead of by a pooling operation, which we are used to seeing as down samplers.

About tuning:

Result from pure ResNet

Training accuracy is around 30 percent. Training time is about a day for this un-optimised model. With the long training time and low training accuracy, it is not advisable to continue with this model.

Initial result from transfer learning without tuning

Training accuracy is 97% but validation accuracy is 75%. Training time is around 5 hours. The training accuracy has increased compared to the previous model, i.e. without transfer learning, but validation accuracy is to be improved further.

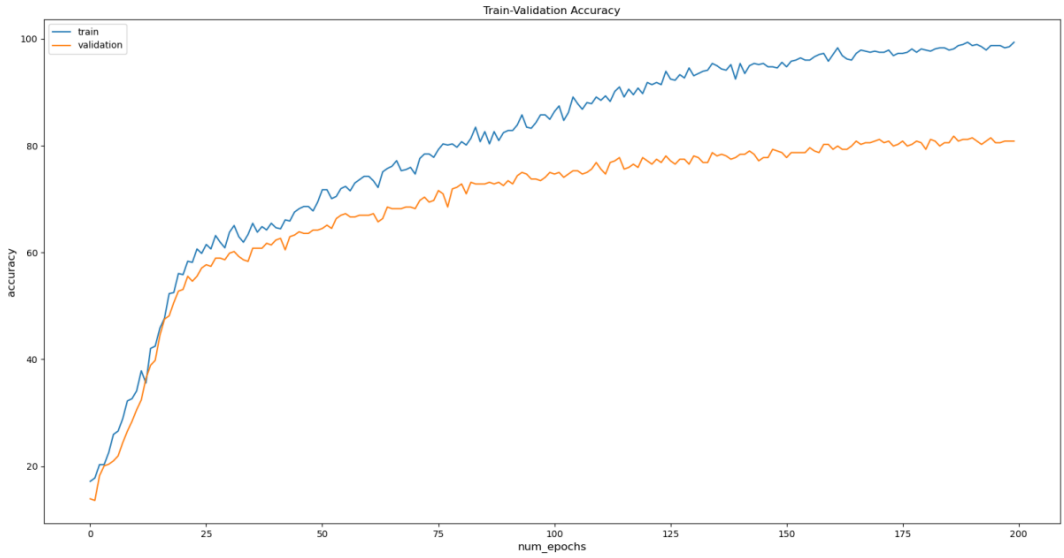


Figure 4. Transfer Learning's training and validation accuracies before tuning

The problem of overfitting

When training accuracy is high and validation accuracy is low, the problem of overfitting occurs. It means that with new ship images, the trained model cannot correctly classify them although it does well within the training set. Another intuitive way to interpret overfitting is by a statistical example called logistic regression.

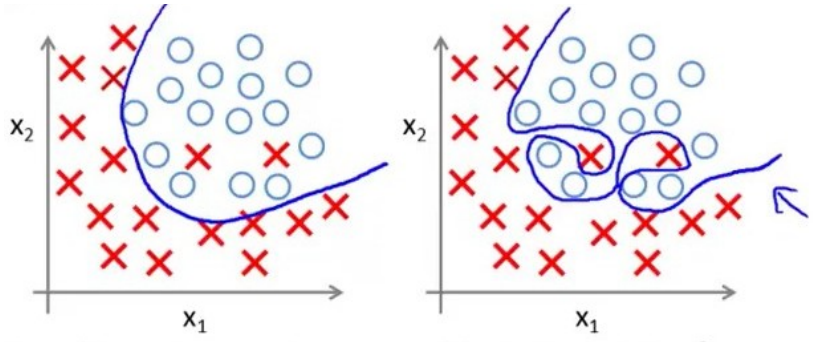


Figure 5. Metaphor of overfitting in logistic regression

From a statistical point of view, suppose a binary classification and prediction problem that is solved by logistic regression, as shown in figure 5. On the right is an overfitting example, in contrast with a correct example on the left. With overfitting the prediction model (shown as the blue line) is so saturated with the influence of outliers that prediction of average data is affected. Similarly, with image classification, outliers in the training set bend the training model too much in order to achieve the accuracy target, but in fact the model is not optimal yet.

Ways to solve overfitting:

Common methods to solve the overfitting problem is to:

A) get more data

B) add data augmentation

C) reduce model complexity

D) tune hyperparameters

The first two points attempt to reduce the effect of outliers, if outliers confuses the model which happens when there is not enough data. The third point reduces the number of model parameters to prevent overfitting. The last point is to smoothen the “prediction curve” in the metaphor above, by adjusting the model itself. The three most important hyperparameters in tuning practice are learning rate, batch size and number of layers. Note that model complexity is discussed as a hyperparameter in this report.

Data augmentation

As compared to the last figure, the training process is faster as shown by the early appearance of the plateau. However, validation accuracy is still around 80%.

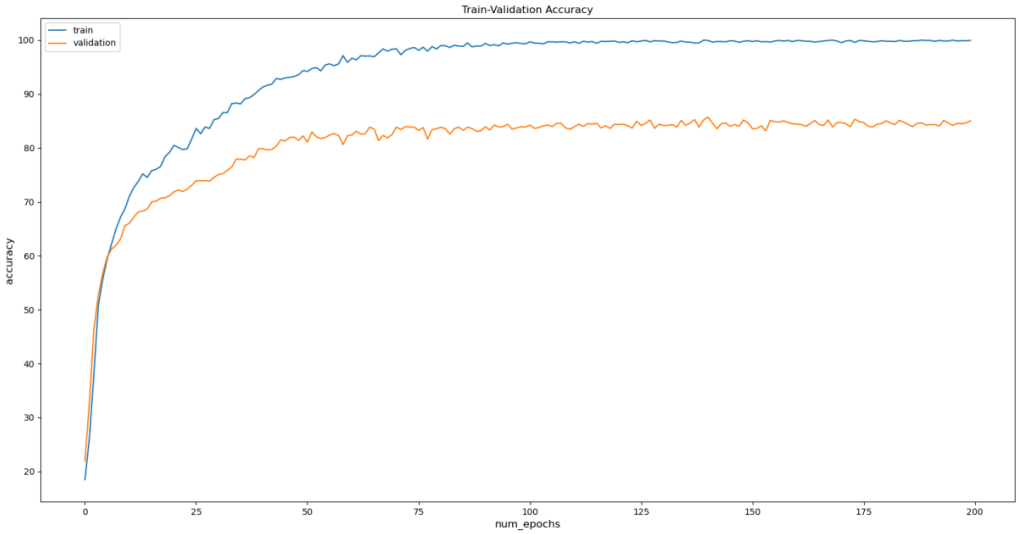


Figure 6. Training and validation accuracies after image augmentation

Tuning procedure

Learning rate is the most important parameter to tune. So this number is addressed first.

1) Learning rate

The comparison of the first six plots in Appendix, i.e the tuning process on learning rate is shown below. From the line graph below, the optimal learning rate is around 0.01 or 0.02. Hence the next two experiments use learning rate 0.01 and 0.02 for best results.

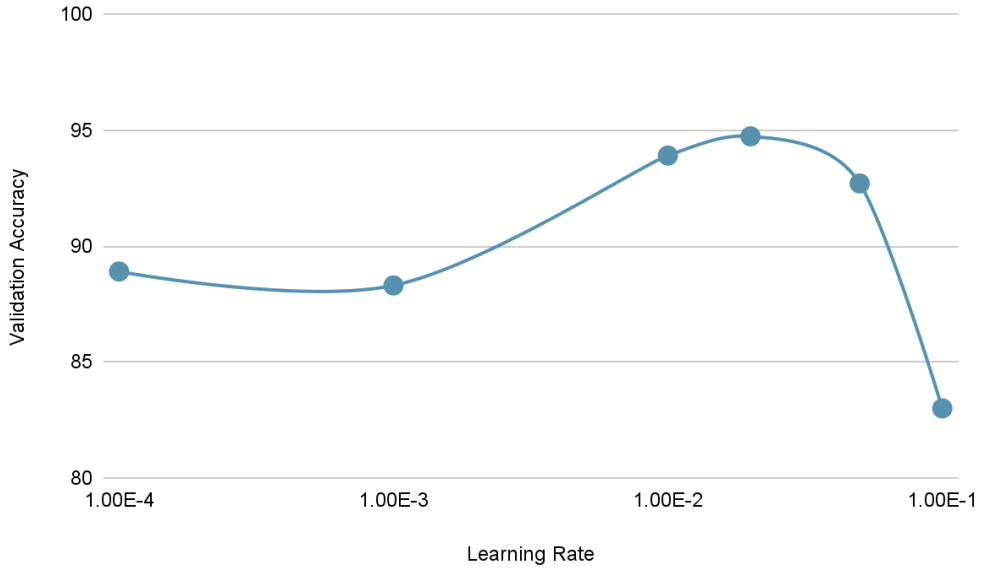


Figure 7. Validation accuracy with the effect of learning rate

2) Batchsize

After tuning the learning rate, batch size is increased from 32 to 64. Batch size is compared twice with the same learning rate 0.01 or 0.02 and layer 18. The results of different batch sizes are similar. Corresponding source plot of training and validation accuracy of each test is the 7th and 8th plot in Apppendix.

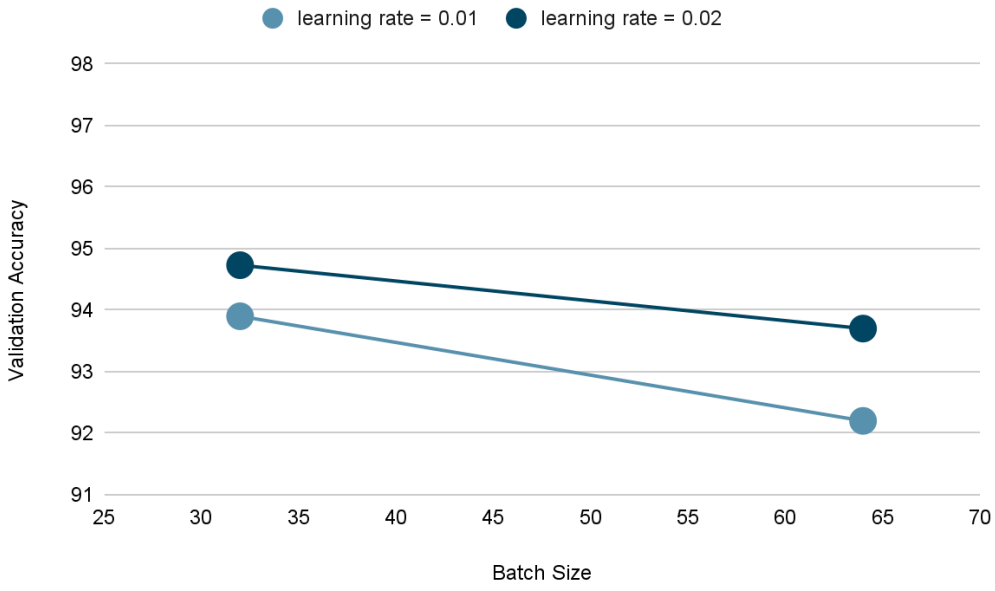


Figure 8. Validation accuracy with the effect of batch size

3) Number of layers

Number of layers is compared twice with the same learning rate 0.01 or 0.02 and batch size 32. The results of different numbers of layers are similar. Corresponding figures in Appendix are the 9th and 10th plot.

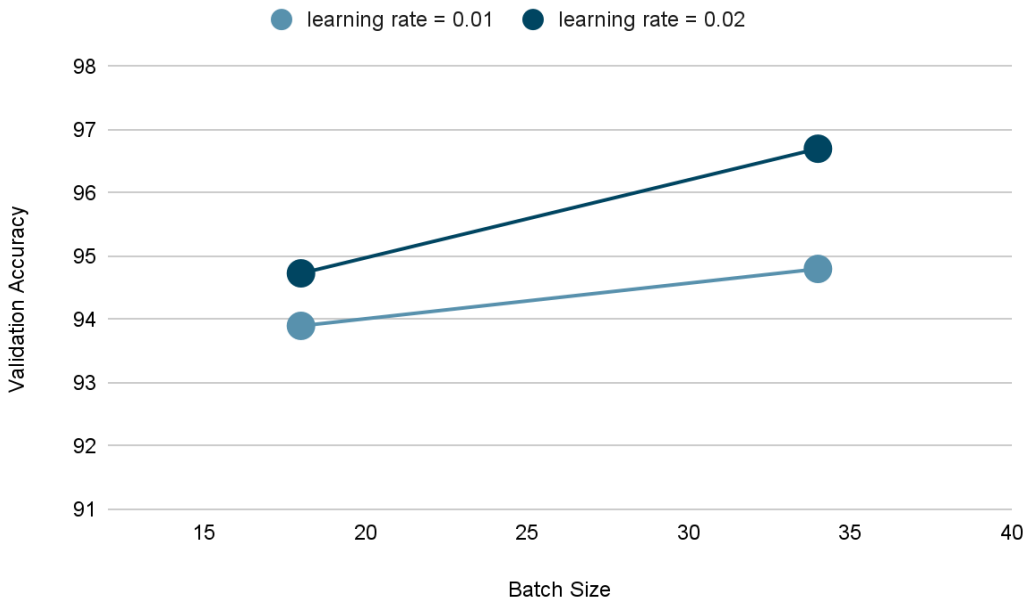


Figure 9. Validation accuracy with the effect of number of layers

Hence the best parameters are: learning rate 0.02, batchsize 32, number of layers 34, which gives a validation of 96.7%.

About automation on google earth:

In the figure below, open Google Earth engine on the left side of the screen and the jupyter notebook on the right. Position the designated ship in the same location and as large. Classification result is computed approximately after two seconds of running with an average CPU. It will refresh after the result is shown.

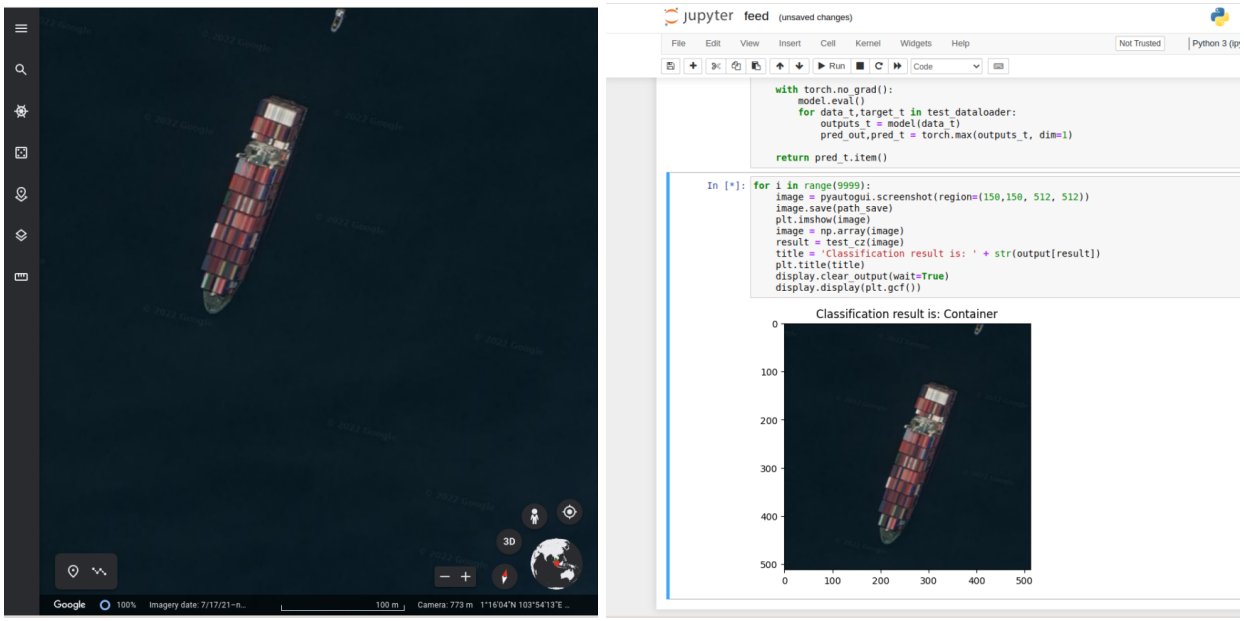


Figure 10. Demo of automated real time classification from web browser

Limitations:

Some drawbacks are inevitable, but the next thing to work on is to expand the types of ships. Below are some limitations the team has discovered.

Boat and yacht can be mixed. This happens because a yacht's body is pure white when the satellite is right above, and with its similar shape to a boat, they can be misidentified into each other.

Figure 11. Images of yacht and boat

Bulk cargos can be misidentified as containers when the test window is zoomed out. This is because some particular subtypes of bulk cargos with a small number have similar box patterns as containers, and containers can look mono-coloured on cloudy days. Combined with both groups' outliers, some ships can be misidentified.

Figure 12. Images of bulk cargo and container

All ships can be misidentified as boats when zoomed out. This is because the shape of all ships are similar due to hydrodynamics and so when big ships appear to be small they look exactly like a boat.

Figure 13. Image of a zoomed out container ship

Open sea can be misidentified as a submarine.

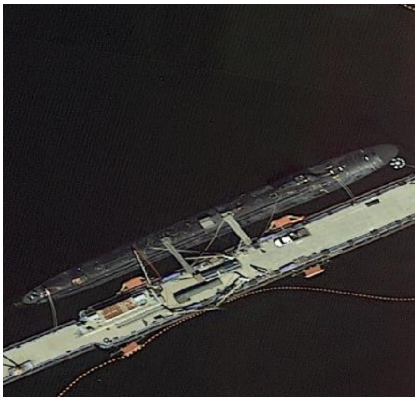


Figure 14. Image of submarine

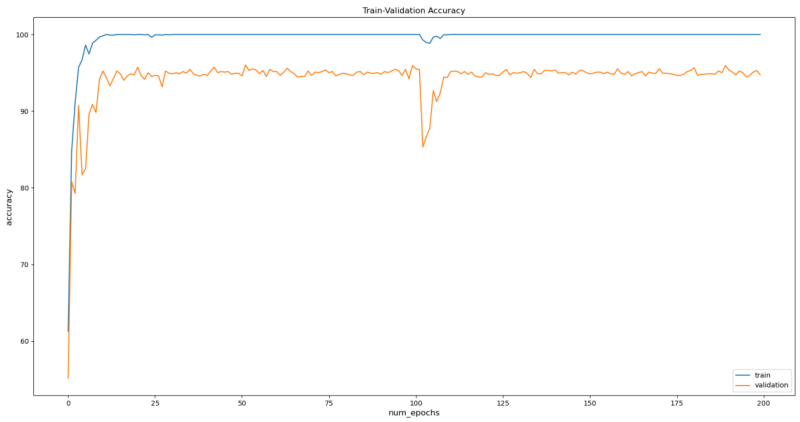
A different design of submarine is not recognised by the model.

Insert pic

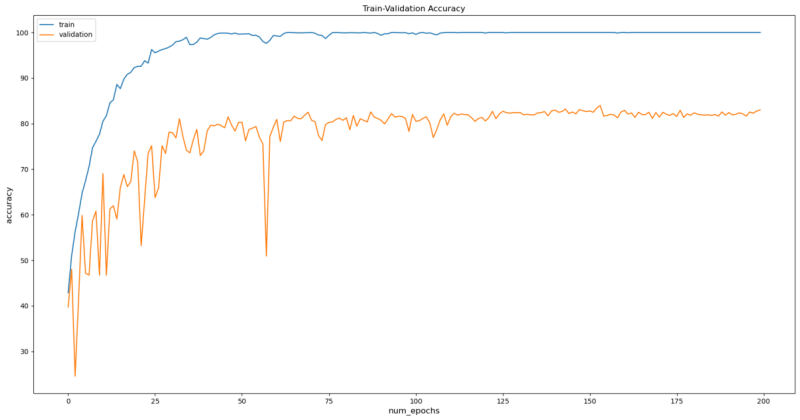
pending to find again

Appendix

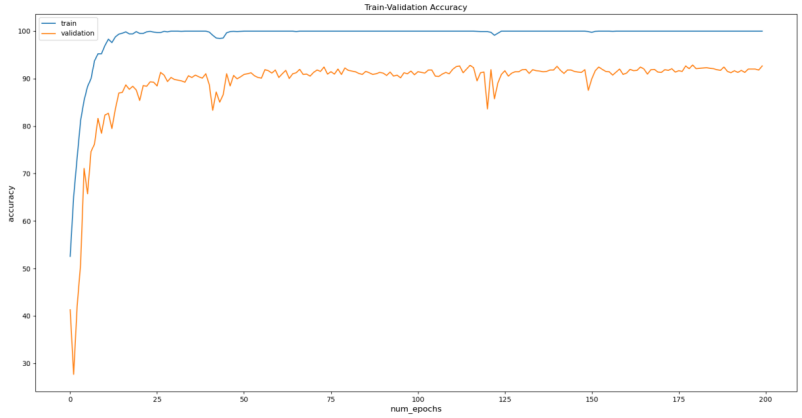
1. lr 0.02 batchsize 32 layer 18: 94.73



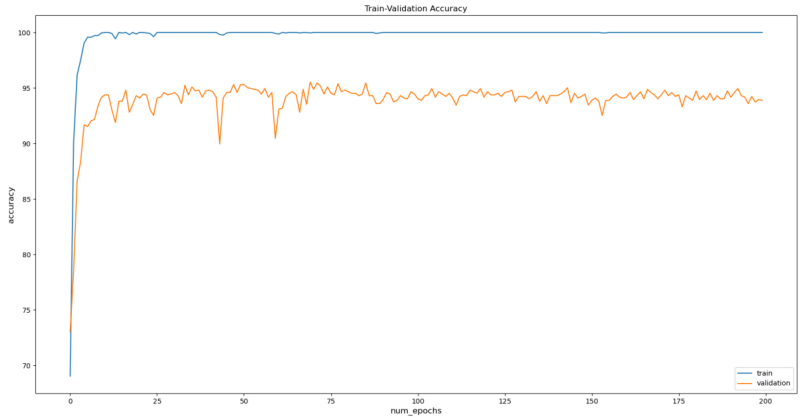
2. lr 0.1 batchsize 32 layer 18: 83.0



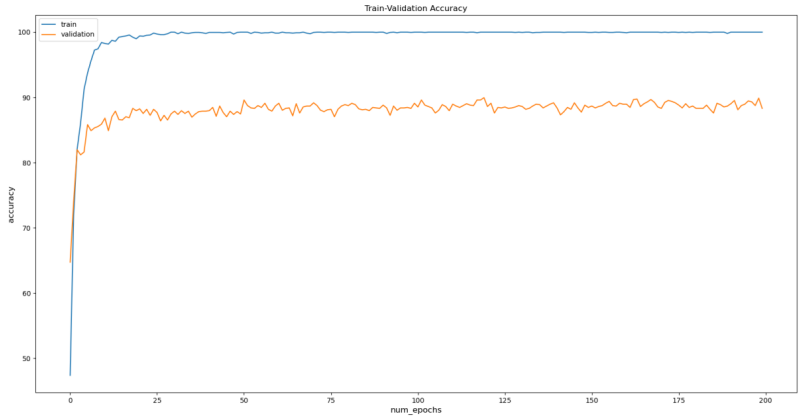
3. lr 0.05 batch size 32 layer 18: 92.7



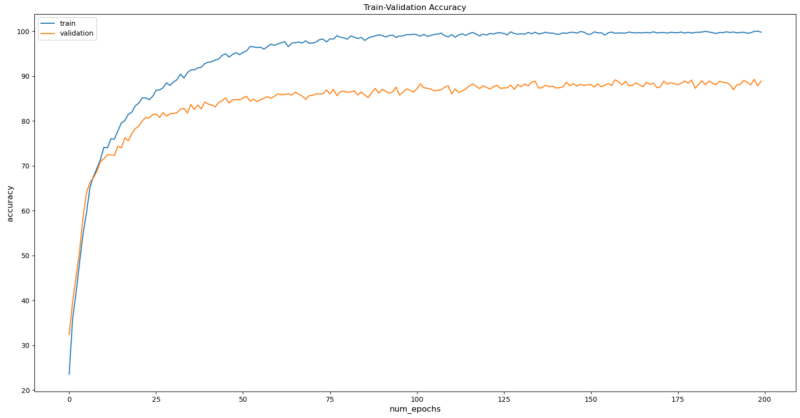
4. lr 0.01 batchsize 32 layer 18: 93.9



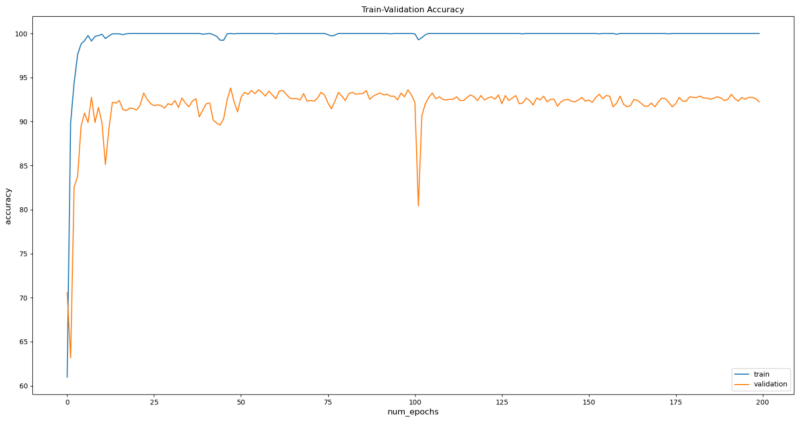
5. lr0.001 batchsize 32 layer 18: 88.3 stable



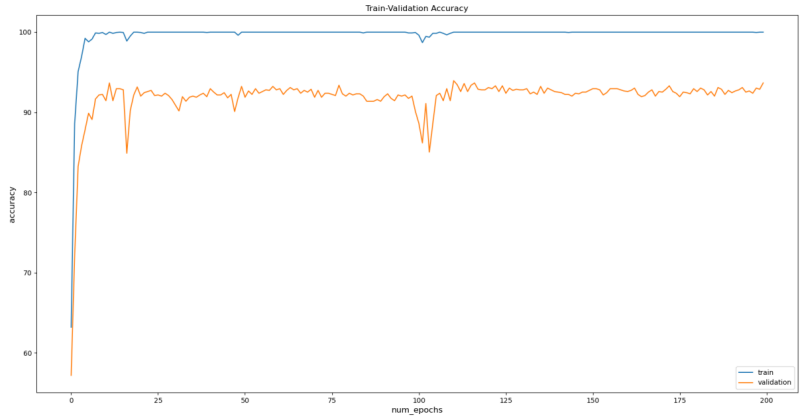
6. Lr0.0001 batchsize 32 layer 18: 88.9 stable



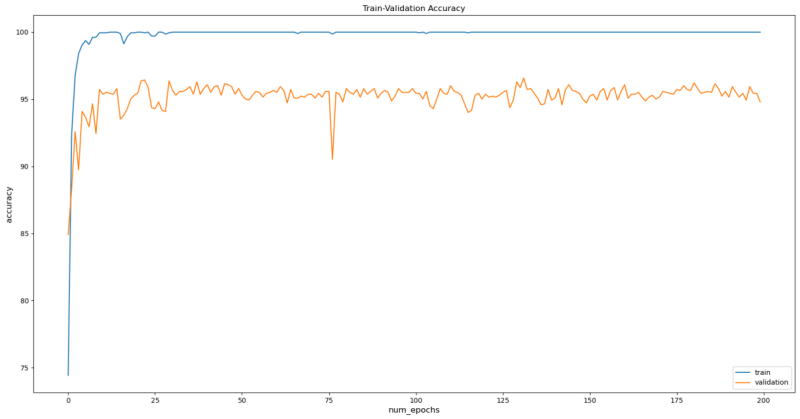
7. lr 0.01 batchsize 64 layer 18: 92.2



8. lr 0.02 batchsize 64 layer 18: 93.7



9. lr 0.01 batchsize 32 layer 34: 94.8



10. lr 0.02 batchsize 32 layer 34: 96.7 stable