Using Artificial Intelligence for Image Classification of Ships

Tiemen English
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
t.f.english@student.utwente.nl

ABSTRACT

Ships sail all across the world to the most far-flung locations to transport people, cargo, and firepower across vast distances. In most of those cases, the information about the ship's location, size, and destination are provided with an automatic system that sends such information all around, helping to avoid collisions or other such complications. In some cases, however, the information provided by the automatic system provided to it by humans can be influenced by accidental human error or malicious intent, and be incorrect.

This paper looks at the building of a suitable dataset of images of ships from imagery available online, namely from the site Shipspotting.com and then using this data to create a classifier. The experimental results show the effectiveness of the use of a pre-trained 18 layer ResNet model and showcase the benefits of upgrading to a 34 layer ResNet model when trying to achieving good performance for ship image classification.

Keywords

artificial-intelligence, image classification, transfer learning, ships

1. INTRODUCTION

We live in a world where a large number of ships pass through our oceans with widely varying purposes. These ships go all across the world, show up at many ports, harbors and canals with locks. Ports make use of the Automatic Identification System (AIS)[16] for identification, a system that combines the radio-frequency range with GPS location and some identifying information about the ship in question. This is efficient and allows for proper identification of ships, however, this is a system that uses only hardware on the ship in question, and so leaves the opportunity for mistakes[20], or in the worst case masquerade the entire ship as a different vessel. To overcome this, an effective method would be to automatically have a system identify if the ship's AIS data corresponds to the ship that arrives physically. To this end, visual identification software is a good method of identifying such discrepancies, were they ever to occur. Visual identification of ships also has uses as systems that are not connected to the AIS as well as maritime management, pollution, and security[6]. This research examines image classification via a pre-trained neural network to attain a high degree of accuracy in figuring out what kind of ship is sailing in front of the camera, as long as the ship is the primary object in view. A lot of research has been done in the field of maritime discerning, most of it in the identification of ship wakes from satellite imaging[12]. A wake is the shape of the waves left after the ship and requires top-down imagery. This same data can be used to estimate the trajectory of a ship, which also helps to identify faults in AIS data[22]. This research, however, focuses on imagery from a horizontal point of view. This is useful when a ship is spotted from other ships, the shore, or harbors, therefore being able to aid in automatic ship recognition in cases of high congestion or when coming across ships out at sea, as long as the ships can be individually isolated from the bigger picture.

The rest of the paper is organized as follows. Section 2 presents the background and related work and talks about basic information needed to understand the steps taken in section 3. Section 3 gives a brief introduction to the methodology used in the presented approach. Dataset characteristics and experimental results are discussed in Section 4, while Sections 5 and **??** concludes and presents future work.

1.1 Problem statement and research question

In machine learning, a problem is usually defined as either supervised or unsupervised. A supervised learning problem involves an output that is caused by the observations that are the input. Unsupervised learning assumes that all output observations are caused by variables that are not directly observable. A classification problem in machine learning is about the way a neural network groups data by class based on predetermined characteristics, which makes it a supervised learning problem[14]. The class in which an image belongs is determined exclusively by the features of the image. In the maritime field, ships are sorted primarily by purpose, and for the most part the external features of a ship are significantly different when comparing the different purposes a craft was built for.

One such a supervised learning problem is a classification problem, one with a discrete solution, namely that the input images can all be classified as a specific output class. This means that there is a functional connection between the input and output. The goal of this paper is to generate a classification algorithm that takes training data as input and gives a classifier that can, as accurately as possible, describe this relationship between an image and the class it belongs to. This paper specifically looks at mul-

28th Twente Student Conference on IT Febr. 2nd, 2021, Enschede, The Netherlands.

ticlass classification, where an assumption is not made on the mechanism creating the training points, instead, the goal is to find a probability distribution between the training images input and class output that can be generally applied to the greater problem space of maritime vessel image classification. The goal of this research is thus to attain a high accuracy with a classifier made with the use of a dataset that, for this paper, will be curtailed to only five classes.

Research question:
What is the highest average accuracy attainable for a classifier when trained on images of ships in five categories?

1.2 Main contribution

This research shows that even at a small amount of calculating power in a consumer-grade computer, a comparatively high accuracy can be attained when training a neural net to classify a small number of ship classes. This research specifically looks specifically at classifying ships in a small number of epochs with a relatively small amount of data and small amounts of layers. It shows the ease of use and accuracy gained when pre-trained neural net model libraries are used, and shows that adding multiple layers in a residual neural network has little benefit in this exact case.

2. RELATED WORK AND BACKGROUND

There has been a sizeable amount of work done in the field of ship identification based on many different types of data, but the largest part of the research is focused on satellite imagery. This, because ships are dispersed over an extremely wide area during their transits, without vantage points from which this area can be surveilled. Other research focuses on the extrapolation of global positioning data, be that for collision prevention[13, 17], the abovementioned trajectory prediction for the management of waterways[22] or tracking of air quality and environmental impact[21]. These last few papers acquire their data through AIS data. AIS data contains, among other things, GPS and data on the purpose of the vessel. The image recognition techniques discussed in this report can be used to back up and cross-reference this data. If a discrepancy is found, a ship that is not where it is supposed to be still needs to be identified. The general use of AI has, of course, had an incredible amount of research done on it, from the recognition of street signs[7] to content of children's books[10]. The goal of this paper is to identify the capacity for this technology to be applied to maritime vessels, with a focus on reaching a high degree of accuracy.

It would be illustrative to understand how computer vision works, and what a convolutional neural network is. Image files are an array of pixels, that each have a brightness value. This brightness is usually defined as two bytes of data, a number from 0 to 255, multiplied by the number of pixels that make up the image. For color images, the amount of data is tripled, as each pixel needs a brightness for the RGB values, the red, green, and blue. A computer then has an array of three times the array of bytes, which are put together for humans to see a natural image. For computer-based image recognition, it necessary to recognize the parts of a picture that make it discernible as a specific class. These 'parts' are called features. If enough features of a class are found, chances are that the image that is being looked at contains this class of image. The question then becomes: How does one identify these features.

Manual feature selection is possible, and one can write an algorithm to identify these features. The angles of the straight lines that make up the prow of a ship can be generalized and, with a lot of effort, identified in a natural image. This is, however, not a robust method of feature detection, as images have a wide range of angles of observation, brightnesses and distance from which they are taken. To remedy this problem, various image and signal processing techniques have been proposed, until Schmidhuber's research group published a set of papers that proposed neural networks[9] as a viable and efficient method to solve the image recognition problems. In more advanced even later methods, image recognition was done by connecting subsections of the array to neurons in deeper layers, so that each neuron has access to only the information in a small patch of pixels. For a higher efficiency, the information a neuron has access to is partially "seen" by other neurons. This method is called convolution.

In these convolutional neural networks, feature extraction is then done by comparing smaller sections to all other references during training. The benefit of the convolution is that the subsections of the image are still held in context of each other. Features are convoluted into a map of features, that can be pooled, which lowers the size of the feature maps and makes each calculation more manageable[1]. This type of technique leads to lower-level features being rectified and convoluted in higher layers to find combinations of features that allow identification of higher level features. For our example, a ship must consist of a hull with a bow, a waterline and the top edge. A container ship will have a stack of colorful containers or a mast for those containers to be fastened to, an aircraft carrier has a ski-jump for aircraft to get some height on takeoff. It is not uncommon for pictures to have one or more of these features obscured. However, if a large amount of these features is present on an image in the right location comparative to each other, a high probability arises for the algorithm to classify the object in the image. Models can be trained to look for features that are specific to the training data[2].

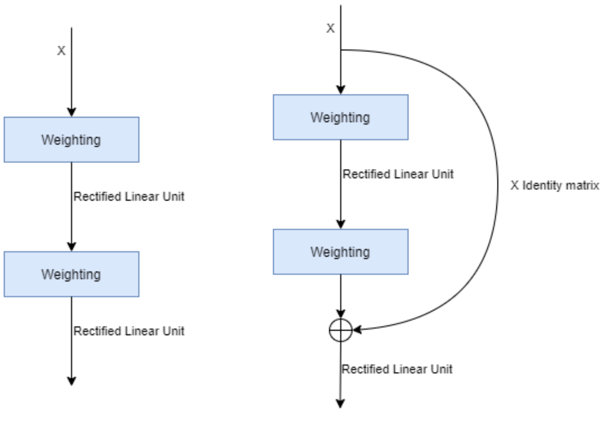


Figure 1. Images without and with residual connections respectively.

The model used for this project is a residual neural network based on the study by Kaiming in 2015[8]. It works like a conventional convolutional neural network, where every few layers, two, in this case, it changes weights of a function X, before being rectified. Rectified linear units essentially take the function and return negative values to zero. This improves performance by keeping information on relative intensities[15]. A residual neural network retains 'residual' connections from prior layers to hold at least the previous level of efficiency, theoretically increasing performance when adding more layers and thus combating overfitting. These residual connections are visible in Figure 1. After two layers of the network, the matrix is added to the existing function from prior layers. This allows for additional learned features to impact the function more significantly than with a conventional convolutional net. This leads to a more robust neural net that is more capable than common convolutional neural networks. The image demonstrates a step with two layers, the ResNet18 used in this paper has 18 of these layers.

3. METHODOLOGY

This research consists of three large parts. The first part is retrieving the data, as the database of ship images was not prepackaged for use. The second part is image preprocessing. Neural networks reach higher levels of efficiency with normalized datasets, as normalizing a dataset combats overfit. The third part involves the creation of the classifier itself, which is done using a residual neural network.

The first step for this research was retrieving the dataset from the Shipspotting.com website. This was done by retrieving, for each class of ship, the full HTML webpage for the list of images with a length of 9999 items maximum. The images are only the thumbnails and represent higher resolution images available on the site. Thumbnails have limited the size at a maximum horizontal resolution of 210 by 158 pixels maximum, making them significantly more representative of the resolution attainable when the image is taken from a small section of a video camera view that would be taking in a larger scene. Each image is downloaded to a folder on disk specific to its class. For the image classification, three sets are created, a training dataset, a validation set and a test set. Each set contains a subset for each class. The class-specific training sets contain 300 images each, the validation and test sets contain 100 images. A representation of the dataset is displayed in Figure 2. Only 500 images were used per class as opposed to the total amount of images collected. To get a reasonably representative image of the most common traffic on the seas, but still include a wide variety of different ship classes For that reason, the classes for the classification became two classes of cargo ship, two classes of warship, and one class of passenger ship. The two cargo classes are liquid cargo tankers and container ships, the warships aircraft carriers and corvettes, the passenger ships cruise ships. The navy vessels were chosen as those are expected to be more problematic for automatic recognition. All of these ships are common ship types for their respective class[5].

The second step, the data pre-processing. As libraries for neural net models are widely available, one was used for the code. PyTorch was used. PyTorch was chosen because it supports convolutional neural nets, and though this library runs primarily on CUDA, it has very good OpenMP support. CUDA and OpenMP are API's that allow access to processor cores for efficient parallel processing[3, 4]. These libraries have transformation functions that allow for efficient normalization of images. For the training of the neural net, the images are first altered and normalized to be a more robust range of images. Each image is resized to a square of 210 by 210 in the training, validation and test sets. The training and validation sets are then randomly cropped to a square of 110 pixels squared and flipped horizontally. These operations take a random part of the image and apply a flip it randomly to get the highest possible normalization of the training data. A randomized horizontal flip is not performed on the test set. A visual representation is given in Figure 3 All images in the datasets are then mapped to a tensor datatype and normalized to stop the brightness from having adverse effects.

The images are then fed to an 18-layer, and then a 34-layer pre-trained ResNet model provided by the PyTorch library. The model was pre-trained on the ImageNet dataset. Since the pre-trained model has 512 features, that is the number of features that will be continued with. The loss function is cross-entropy loss, which is a common loss function for machine learning. The optimizer used is Adam, an optimizer that is also common[11]. This method was inspired by a short study from the blogsite NaadiSpeaks[19], and proves very capable. This implementation of ResNet follows the original proposal paper for residual neural networks[8] very closely. It uses 18 layers of weighting, with rectifiers between each layer, and an addition of the previous identity matrix every two layers. This is done because the weight layers tend towards zero, while the addition of the identity matrix ensures that only the deviation from that previous identity matrix is learned.

The code for this classifier can be found at this link:
https://github.com/TFEnglish/ShipAI/

4. RESULTS

The pre-trained model starts out, after one epoch, with an acceptable accuracy of around 50%, visible in Figure 4, which is better than pure random chance with an untrained neural network. An untrained ResNet18 with the exact same method after one epoch only shows an accuracy of around 40%. Thus, using the pre-trained Residual Neural Models is gives higher quality results. The results are displayed on the confusion matrix. In the following calculations, the following things will be referred to as such: All positive cases P, all negative cases N, true positive values TP, true negative values TN, false-positive values FP and false-negative values FN. From the confusion matrix in Figures 6 and 9 a lot of information can be pulled.The solutions to the following calculations for individual classes are shown in Table **??**. Firstly, the total average accuracies. The average for each class is calculated with the formula:

$$ACC = \frac{TP + TN}{P + N}$$

The average accuracies for ResNet18 and ResNet34 are 0.965 and 0.968 respectively. The average training accuracies in each epoch are shown in the graphs in Figure 4 and Figure 7. These seem to level out quite quickly, from epoch 8 onward. Similarly, the losses shown in Figure 5 and Figure 8 show that it also levels out at roughly the same point.

The precision or positive predictive value of each class is calculated with:

$$PPV = \frac{TP}{TP + FP}$$

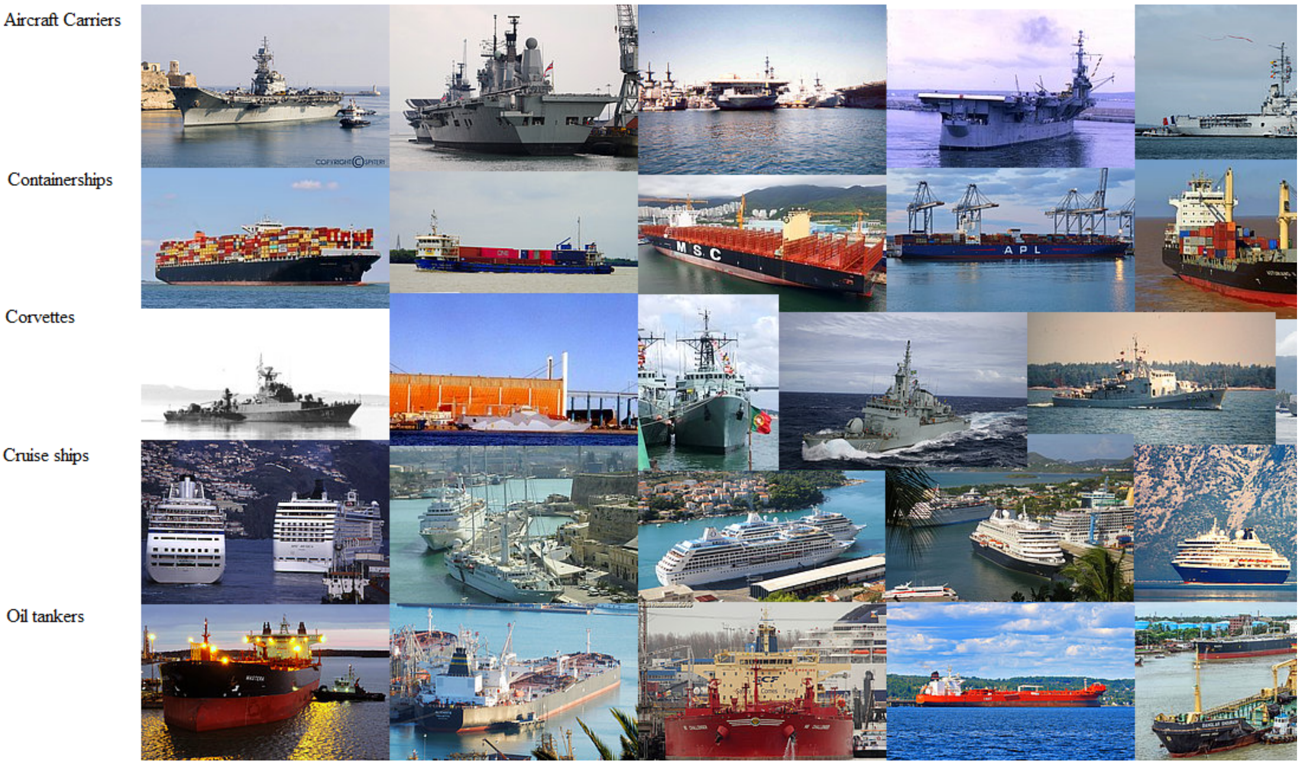


Figure 2. A representative sample of the images collected from Shipspotting.com

Table 1. A table with the Accuracies, Sensitivities, Specificities, F1-scores, False-positive and False Omission Rates of each class after training for 35 epochs with the pre-trained ResNet18 and ResNet 34

ResNet18: Class	**acc**	**Precision**	**Sensitivity**	**F1-score**	**Specificity**	**FPR**	**FOR**
Aircraft Carrier	0.946	0.970	0.808	0.881	0.991	0.008	0.191
Container-ship	0.984	0.960	0.969	0.964	0.989	0.010	0.030
Corvette	0.946	0.760	0.974	0.853	0.941	0.058	0.025
Cruise Ship	0.976	0.920	0.968	0.943	0.978	0.021	0.031
Oil Tanker	0.974	0.980	0.907	0.942	0.994	0.005	0.092

ResNet34: Class	**acc**	**Precision**	**Sensitivity**	**F1-score**	**Specificity**	**FPR**	**FOR**
Aircraft Carrier	0.954	0.900	0.882	0.891	0.973	0.026	0.117
Container-ship	0.980	0.940	0.969	0.954	0.983	0.016	0.030
Corvette	0.944	0.870	0.861	0.865	0.966	0.033	0.138
Cruise Ship	0.980	0.960	0.950	0.955	0.989	0.010	0.049
Oil Tanker	0.980	0.950	0.959	0.954	0.986	0.013	0.040

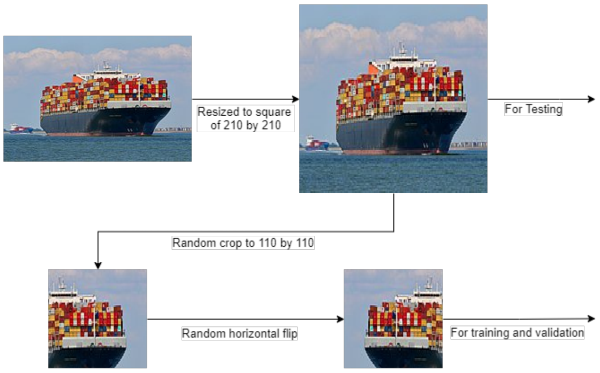


Figure 3. The image transformations as performed.

Notable is that the precision for the corvette class is notably lower than the other classes. This is because a lot of the corvettes were classified as aircraft carriers, as visible in the confusion matrices. The 34 layer net has this problem less significantly.

sensitivity, recall, or true positive rate formula is this one:

$$TPR = \frac{TP}{P}$$

Here it is notable that the opposite of the precision is going on, where the true positive rate for the aircraft carrier class is lowered. Here the sensitivity of the Aircraft carrier-class too is significantly lower with the ResNet34.

The F1-score is supposed to be a one-number score with which to rate the accuracy of a classifier. This is calculated as such for each class, and the average of the whole can be taken afterwards:

$$F1 = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR}$$

The average F1-scores for ResNet18 and ResNet34 are 0.917 and 0.924 respectively, indicating a slightly better score for the higher-layer network.

specificity or true negative rate is calculated with this for-

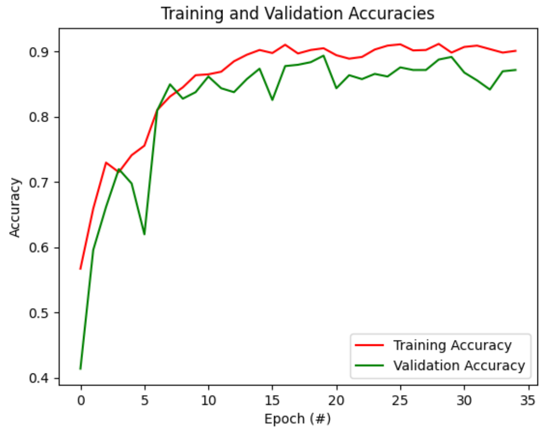


Figure 4. The average accuracies at each epoch for 5 classes with 18 layers.

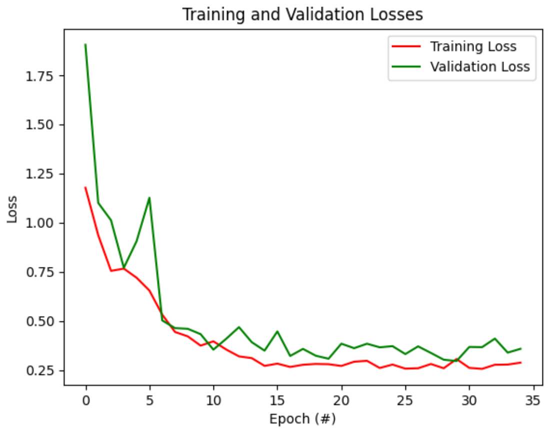
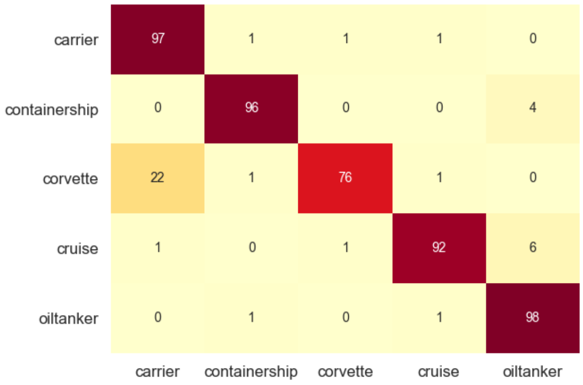


Figure 5. The average losses at each epoch for 5 classes with 18 layers.



	carrier	containership	corvette	cruise	oiltanker
carrier	97	1	1	1	0
containership	0	96	0	0	4
corvette	22	1	76	1	0
cruise	1	0	1	92	6
oiltanker	0	1	0	1	98

Figure 6. The confusion matrix for 5 classes after 35 epochs with 18 layers.

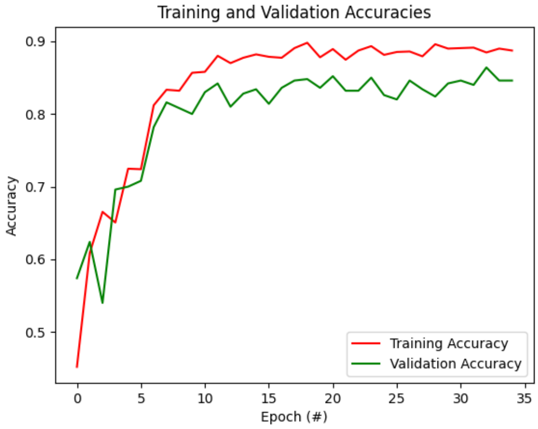


Figure 7. The average accuracies at each epoch for 5 classes with 34 layers.

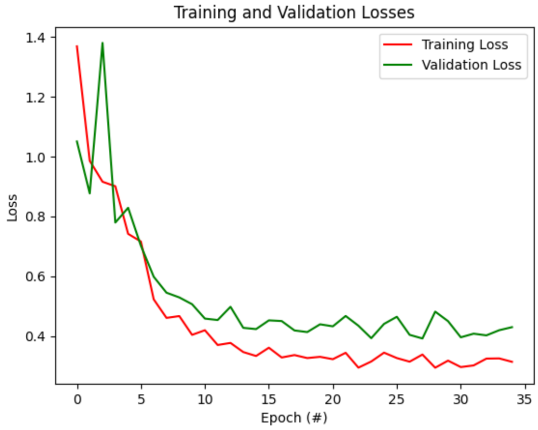


Figure 8. The average losses at each epoch for 5 classes with 34 layers.

mula:

$$TNR = \frac{TN}{N}$$

The values in this column show a very tight spread with no notably high or low numbers.
The False-positive rate and False Omission Rate are calculated with the following two formulas:

$$FPR = \frac{FP}{N}$$

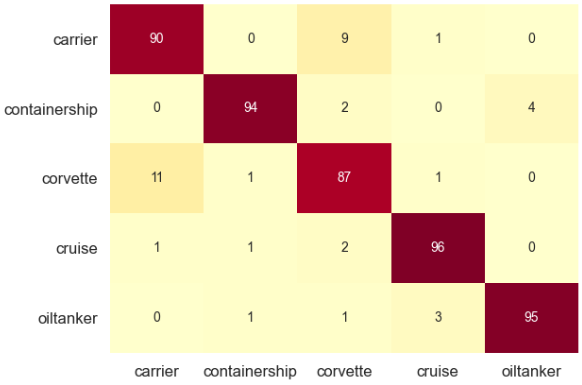
and

$$FOR = \frac{FN}{FN + TP}$$

These values were calculated for their indicative values for the specific use cases that this paper could be involved in. The False-positive rate for both networks is notably higher in the corvette class, and for the higher layer count network, also higher in the aircraft carrier class.

5. CONCLUSION

In Table 1, most classes do very well. The accuracy for all classes is above 94%, with a slight edge given to the higher layer count ResNet. This is only two-tenths of a percent, however, thus the conclusion is that the accuracy doesn't change much with a small dataset when more layers are added. The precision tells a different story, as there is a significant dip in the precision of the corvette class. The confusion matrices in Figures 6 and 9 show that this is because a significant number of corvettes, 22% for the



	carrier	containership	corvette	cruise	oiltanker
carrier	90	0	9	1	0
containership	0	94	2	0	4
corvette	11	1	87	1	0
cruise	1	1	2	96	0
oiltanker	0	1	1	3	95

Figure 9. The confusion matrix for 5 classes after 35 epochs with 34 layers.

18 layer, and 11% for the 34 layer network, are wrongfully labeled as carriers. This is presumably because military ships often have a high similarity in features such as the color gradient, the smaller equipment such as radar domes on the superstructure. This can presumably be remedied with a larger dataset that better exemplifies the differences between the two types of ship. The same problem can be seen in the sensitivity, where the aircraft carrier and corvette classes have lower values than the other classes. This would mean that, in its current state, these two classes have a lower chance of being detected correctly than the other three.

The overall F1-score is a quick metric that gives an idea of the quality of the classifier. Here both score very similarly at 91.7% and 92.4% for the 18 and 34 layer networks specifically. This difference is very minor with the current training set size, however should be more significant once larger sets are employed to train this classifier. With the current training set size, the difference is minor.

The specificity is the True Negative Rate, the rate at which elements are that are marked as not the class we're looking at are truly not that class. Here, this algorithm scores extremely well. This means that, if a ship is marked as not being the class it is expected to be, it is most likely to be correct in that assessment. This means that if indeed, a ship was masqueraded as a different class of ship from what it is supposed to be, it would identify it as a different ship in most cases, with a 99% rate for aircraft carriers and oil tankers for the 18 layer net. The 34 layer net does not reach this level of specificity, but this is still quite good.

The False-positive Rate is also interesting in this regard, as it explains the probability of falsely identifying ships as the class it is expected to be. These rates are quite low, but again are very significantly higher for the corvette class in both classifiers, this time even more so for the 18 layer ResNet.

The False Omission Rate shows the livelihood of a negative being a true negative, so how many ships are omitted when looking at that class that should be included. this is markedly high for the aircraft carrier class in both networks. the corvette is also significantly higher in this class for the 34 layer network, showing that when looking for corvettes and aircraft carriers with this technique, it is more likely to miss them than when looking at container ships, cruise ships and oil tankers.

These results show that these five classes are differentiated quite effectively after 35 epochs of training. The assumption is that if more training data is used, even higher accuracies and F1-scores when differentiating between the 5 classes of ships should be achievable for a classifier. This method, even with the low amount of data it was able to be trained on, is useful in confirming whether a large ship sailing up to a harbor is indeed of the class it claims it is via AIS data, but significantly more so when it's not a warship. With more training on a larger dataset, this method will be able to reach a significantly lower false-positive rates and higher sensitivities for the warship classes as well. It is not significantly beneficial to add larger numbers of layers with a small dataset and a small number of classes such as the classifier in this paper was trained to do. With the results above, it is not yet capable of reaching proficiencies at its job that make sense to implement as a primary feature. At most, I can see these levels of accuracy be implemented as a predictive substitute for filling in forms at arrival. It can be implemented to identify the class of common merchant ships in large harbors in the world, slightly easing the workload of the port authorities and pilots. This research was performed locally on a consumer-grade computer, showing that on the fly, training does not require a large supercomputer or even a large computer cluster to achieve usable results. If the calculations were performed optimally with the right hardware, it would have taken no time at all to complete. As it stands, however, it is not useful as more than that, as a stand-alone system to automatically identify incoming traffic.

6. CONSIDERATIONS

The training for this paper was performed on a very small subsection of the available dataset. This was done because the calculations were run locally, and the use of the whole dataset would have lead to significantly longer processing times. Using the full dataset for training would lead to a more accurate and representative result[18], and would otherwise have been preferable. Some of the classes on the Shipspotting.com website had more images to use than the 9999 images the scraper was set up to collect, and this does not include the myriad of other databases of ship images that exist out on the internet. From this research, not much can be said about the value of classification on larger datasets or larger amounts of classes. This was originally planned to be a part of this research but left for future research because of time constraints. Similarly, not much can be said about the calculating power required for training at that scale, but that too would be interesting to do in future research.

The PyTorch library that was used has the option to run on OpenCL as well when using an externally maintained package. This was important as at the time of writing of the code, as these technologies make use of processor chips designed to speed up matrix operations. There was no local access to CUDA hardware, as this a technology created by and exclusively in use with Nvidia GPUs. OpenCL is an equivalent technology originally designed by Apple to be open-source and is available for graphics cards from the other primary GPU manufacturer, AMD. Since a GPU from AMD was the only one available, and the plan was to run the simulations locally, this was a primary consideration. Since Neural network training is a task that is largely matrix operations, the compute cores in graphics hardware are quite capable of performing these calculations with a significantly higher efficiency. In the end, this OpenCL functionality was not used due to time constraints during the implementation. All calculations were eventually done on an AMD CPU rather than GPU, and used OpenMP instead. With more processing time available, the expectation is that higher accuracies can be attained. This would still show results viable for consumer hardware, as in the current generation of computers, GPUs with these CUDA cores or Stream processors are still very common, and are capable of doing large matrix operations.

7. ACKNOWLEDGMENTS

My supervisor Elena Mocanu for being a great supervisor, if only I'd allow myself to be better supervised.
My dad, for taking some time away from mom to help.
My friend Chris Coerdes, for assisting in data collection and general help.

8. REFERENCES

[1] S. Albawi, T. A. Mohammed, and S. Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6, 2017.

[2] A. Amini. Mit 6.s191: Convolutional neural networks, Feb 2021.

[3] P. Christensson. Cuda. *TechTerms*.

[4] P. Christensson. Opencl. *TechTerms*.

[5] V. Eyring, H. W. Köhler, J. van Aardenne, and A. Lauer. Emissions from international shipping: 1. the last 50 years. *Journal of Geophysical Research: Atmospheres*, 110(D17), 2005.

[6] J. Goodman. Tanker's impossible voyage signals new sanction evasion ploy. *ABCNews*.

[7] L. Han, L. Ding, and L. Qi. Real-time recognition of road traffic sign in moving scene image using genetic algorithm. In *Proceedings of the 4th World Congress on Intelligent Control and Automation (Cat. No.02EX527)*, volume 2, pages 1027–1030 vol.2, 2002.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[9] S. Hochreiter. Untersuchungen zu dynamischen neuronalen Netzen. Diploma thesis, Institut für Informatik, Lehrstuhl Prof. Brauer, Technische Universität München, 1991.

[10] C. Huang and H. Jiang. Image indexing and content analysis in children's picture books using a large-scale database. *Multimedia Tools and Applications*, 78, 2019.

[11] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] Y. Liu, J. Zhao, and Y. Qin. A novel technique for ship wake detection from optical images. *Remote Sensing of Environment*, 258:112375, 2021.

[13] T. M. P. K. Mingyang Zhang, Jakub Montewka and S. Hirdaris. A big data analytics method for the evaluation of ship - ship collision risk reflecting hydrometeorological conditions. *Reliability Engineering System Safety*, 213:107674, 2021.

[14] E. Mocanu, P. Nguyen, M. Gibescu, and W. Kling. Optimized parameter selection for assessing building energy efficiency. 04 2014.

[15] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress.

[16] I. M. Organisation. A.1106(29) guidelines for the onboard operational use of shipborne automatic identification systems (ais). 2015.

[17] H. Rong, A. Teixeira, and C. Guedes Soares. Spatial correlation analysis of near ship collision hotspots with local maritime traffic characteristics. *Reliability Engineering System Safety*, 209:107463, 2021.

[18] D. Soekhoe, P. Van Der Putten, and A. Plaat. On the impact of data set size in transfer learning using deep neural networks. In *International Symposium on Intelligent Data Analysis*, pages 50–60. Springer, 2016.

[19] H. Thilakarathne. Transfer learning in convnets.

[20] D. Yang, L. Wu, and S. Wang. Can we trust the ais destination port information for bulk ships?–implications for shipping policy and practice. *Transportation Research Part E: Logistics and Transportation Review*, 149:102308, 2021.

[21] L. Yang, Q. Zhang, Y. Zhang, Z. Lv, Y. Wang, L. Wu, X. Feng, and H. Mao. An ais-based emission inventory and the impact on air quality in tianjin port based on localized emission factors. *The Science of the total environment*, 783:146869, April 2021.

[22] L. Zhao and X. Fu. A visual method for ship close encounter pattern recognition based on fuzzy theory and big data intelligence. In *2020 the 4th International Conference on Big Data Research (ICBDR'20)*, ICBDR 2020, page 94–100, New York, NY, USA, 2020. Association for Computing Machinery.