Evaluating Computer Vision Architectures for Ship Classification: A Comparative Study

Mateus R. Cruz, Samuel B. Mafra, Eduardo H. Teixeira, Danilo M. Oliveira and Felipe A. P. Figueiredo.

***Abstract*— The complexity and vastness of the ocean make maritime monitoring a critical aspect of both national security and international trade. This paper investigates the potential of computer vision (CV) technology for monitoring maritime activity, with a particular emphasis on ship classification. The study compares various CV architectures and techniques, highlighting their relative strengths and weaknesses. The main objective of using CV in this context is to identify and track vessels and other objects of interest, which could assist in enforcing maritime regulations, increasing trade efficiency, and detecting security risks. The paper presents numerical results obtained during the training and validation of these architectures, providing valuable insights into how CV technology can be employed to improve maritime monitoring.**

***Keywords*— Maritime monitoring, Computer Vision, Image Classification, Artificial Intelligence.**

I. Introduction

Maritime monitoring is a critical aspect of national security and international trade [1]. This is because the ocean is a vast and complex environment, and monitoring it can be challenging. Improving maritime monitoring can involve using computer vision (CV) technology, which allows computers to interpret and analyze visual data [2]. CV can aid in maritime monitoring through different images and videos from vessels. The primary goal of using is to identify and track vessels and other objects of interest. This can include everything from fishing boats to cargo ships to military vessels. By monitoring vessel traffic in real-time, authorities can identify potential security threats, enforce maritime regulations, and improve the efficiency of international trade.

There are many different CV techniques that can be used for maritime monitoring, including object detection [3], object recognition [4], and image classification [5]. Object detection involves identifying the presence of specific objects in an image or video, while object recognition involves identifying the type of object. Image classification is a more general technique that involves classifying an entire image based on its content. In the marine monitoring scenario, image classification can be used to distinguish between the hundreds of variations of existing markings or just the type of each one. But, one of the key challenges in using vision capabilities for maritime monitoring is selecting the right architecture for the task. This is because there are many different architectures to choose from, each with its strengths and weaknesses for particular tasks.

Convolutional Neural Networks (CNNs) are particularly well-suited for image classification tasks [6], as they can automatically learn to recognize complex patterns in images. They work by applying a series of filters to an input image, each of which detects a specific type of feature [7]. By combining the outputs of these filters, the network can identify higher-level features and classify the image accordingly. In addition to the general CV architectures, there are also specific pre-trained neural networks that have been used for image classification tasks. Some popular examples include ResNet, SqueezeNet, DenseNet, VGG, and AlexNet. These networks differ in terms of their architecture, computational requirements, and performance metrics [7].

In addition to general CV architectures, there are pre-trained neural networks specifically designed for image classification tasks. These networks, such as ResNet, SqueezeNet, DenseNet, VGG, and AlexNet, have been trained on large-scale image datasets and achieved impressive performance across various domains. Pre-trained networks offer the advantage of transfer learning, where the knowledge gained from training on one dataset can be transferred and applied to a different but related dataset. By fine-tuning pre-trained networks on maritime-specific data, authorities can leverage their learned features and accelerate the development of accurate and efficient maritime monitoring systems. Also, transfer learning offers several advantages, including reduced training time, improved performance, generalization to new tasks, effective utilization of limited labeled data, extraction of high-level features, and domain adaptation.

The choice of the CV technique and architecture depends on several factors, including the specific objectives of maritime monitoring, the available data, computational resources, and performance requirements. For example, if the goal is to detect and track vessels in real-time, object detection techniques may be more suitable. On the other hand, if the focus is on

Mateus Raimundo da Cruz, Instituto Nacional de Telecomunicações (Inatel), Santa Rita do Sapucaí (MG), e-mail: mateusrc@dtel.inatel.br; Samuel Baraldi de Mafra, Departamento de Engenharia de Telecomunicações, Instituto Nacional de Telecomunicações, Santa Rita do Sapucaí-MG, e-mail: samuelbmafra@inatel.br; Felipe Augusto Pereira de Figueiredo, Departamento de Engenharia de Telecomunicações, Instituto Nacional de Telecomunicações, Santa Rita do Sapucaí-MG, e-mail: felipe.figueiredo@inatel.br; Eduardo Henrique Teixeira,Instituto Nacional de Telecomunicações, Santa Rita do Sapucaí-MG, e-mail: eduardoteixeira@dtel.inatel.br; Danilo Machado de Oliveira, Instituto Nacional de Telecomunicações, Santa Rita do Sapucaí-MG, e-mail: danilomachado@mtel.inatel.br; This work is partially supported RNP, with resources from MCTIC, Grant No. 01245.020548/2021-07, under the Brazil 6G project of the Radiocommunication Reference Center (Centro de Referência em Radiocomunicações - CRR) of the National Institute of Telecommunications (Instituto Nacional de Telecomunicações - Inatel), Brazil, Huawei, under the project Advanced Academic Education in Telecommunications Networks and Systems, contract No PPA6001BRA23032110257684, Brazil, the National Council for Scientific and Technological Development-CNPq (403827/2021-3), FAPESP (2021/06946-0),and by Minas Gerais State Agency for Research and Development (FAPEMIG) via Grant No. TEC - APQ-03283-17

categorizing vessels into different types, image classification techniques can be applied. It is essential to consider the strengths and limitations of each technique and architecture to ensure their compatibility with the requirements of maritime monitoring applications.

This article explores the use of CV for maritime monitoring in more detail and presents some architectural results. In addition, different CV architectures and techniques are compared, highlighting their strengths and weaknesses for ship classification. Brief descriptions of each architecture are given, and some numerical results obtained during the training and validation are presented.

II. Related works

When applying CV techniques for maritime monitoring, selecting the appropriate architecture for the task is essential. CNNs have proven to be highly effective for image classification tasks. CNNs excel at automatically learning and recognizing complex patterns and features in images. Also, the ability of CNNs to capture intricate visual information makes them well-suited for maritime monitoring applications.

The choice of the CV technique and architecture depends on several factors, including the specific objectives of maritime monitoring, the available data, computational resources, and performance requirements. For example, if the goal is to detect and track vessels in real-time, object detection techniques may be more suitable. On the other hand, if the focus is on categorizing vessels into different types, image classification techniques can be applied. It is essential to consider the strengths and limitations of each technique and architecture to ensure their compatibility with the requirements of maritime monitoring applications.

III. Models Comparison

Image classification task counts with a large number of deep learning architectures available in the literature, where each architecture has its own characteristics. The architectures chosen to be trained and compare in this study are as follows: (I) resnet18, (II) resnet34, (III) resnet50, (IV) resnet101, (V) resnet152, (VI) squeezenet1_0, (VII) squeezenet1_1, (VIII) densenet121, (IX) densenet169, (X) densenet201, (XI) densenet161, (XII) vgg16, (XIII) vgg19, and (IXX) alexnet.

The ResNet architectures (ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152) are known for their deep structure, which allows them to capture complex features and patterns in images. They are suitable for tasks that require high accuracy but may be slower and require more computational resources than other architectures. SqueezeNet1.0 and SqueezeNet1.1 are lightweight architectures designed to have a small number of parameters and be fast to train and execute. They are suitable for tasks where speed and memory efficiency are important but may sacrifice some accuracy compared to other architectures.

DenseNet (DenseNet121, DenseNet169, DenseNet201, and DenseNet161) are similar to ResNet in their deep structure but have a unique feature of densely connected blocks, which allows for better feature reuse and efficient parameter usage. They are suitable for tasks that require high accuracy and are computationally efficient. VGG16 and VGG19 are classic deep learning architectures that have been widely used in CV tasks. They have a relatively simple structure compared to ResNet and DenseNet but have a large number of parameters, which can be both an advantage and a disadvantage depending on the task. Finally, AlexNet is another classic deep learning architecture that was one of the first to achieve high accuracy in the ImageNet challenge. It has a relatively shallow structure compared to ResNet and DenseNet but is still suitable for tasks that require high accuracy and speed.

The choice of the deep learning architecture for a specific task depends on several factors, such as the size and complexity of the dataset, the computational resources available, and the desired accuracy and speed of the model. In the case of maritime monitoring, the classification of ships from images requires a model that can handle a large number of classes and detect subtle differences in the appearance of ships. Table I brings more details about the presented architectures.

TABLE I
Models characteristics, Parameters, Parameters Size and Model Estimated Size.

Model	Parameters	Param. size	Est. Size
resnet18	11,689,512	44.59	107.96
resnet34	21,797,672	83.15	180.01
resnet50	25,557,032	97.49	384.62
resnet101	44,549,160	169.94	600.25
resnet152	60,192,808	229.62	836.78
squeezenet1.0	1,248,424	4.76	97.14
squeezenet1.1	1,235,496	4.71	59.05
densenet121	7,978,856	30.44	203.19
densenet161	28,681,000	109.41	418.81
densenet169	14,149,480	53.98	255.39
densenet201	20,013,928	76.35	325.33
vgg16	138,357,544	527.79	747.15
vgg19	143,667,240	548.05	787.31
alexnet	61,100,840	233.08	242.03

The development of a model is a complex process that involves several steps, with model training being one of the most crucial steps. CV models are trained on a large dataset to identify patterns and learn from the data. The training process involves selecting an appropriate algorithm, setting the hyperparameters, and feeding the model with a proper dataset to learn from. The first step in model training is to prepare the data. This includes preprocessing the data, cleaning it, and splitting it into training, validation, and test sets. The dataset used is available from the Kaggle community and is ready to use, requiring no pre-processing or annotation steps. In this way, several hours of work can be saved and used in other stages of model development. The dataset consists of five ship classes grouped in 5626 images. Figure 1 shows how the images are divided among the classes:

The dataset was split into training and validation in the proportions of 80% and 20%, respectively. Splitting a dataset into training and validation sets is a common practice in CV training. The purpose of this is to use the training set to train a model, and then use the validation set to evaluate the performance of the model on data that it has not seen

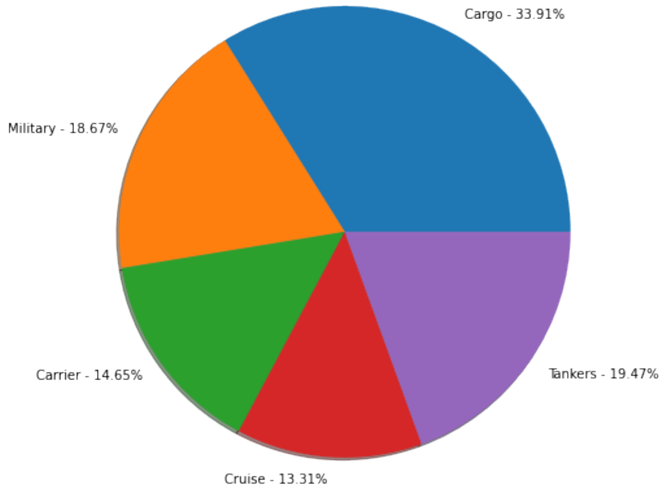

Fig. 1. Amount of observation in each classe in the dataset.

Fig. 2. Images from the dataset.

before. The next step is to feed the model with the prepared dataset and start training. During the training process, the model learns from the input data and tries to identify patterns that can be used to accurately classify new images. The article aims to compare different architectures to the same dataset and training parameters (Optimization Function, Adam, Learning Rate: 0.001, Epochs: 25), helping to identify the weight and performance of each architecture to properly classify ships. It is important to monitor the training process closely, as the model can overfit or underfit the data, leading to poor performance on new data. To prevent overfitting. Overfitting occurs when a model is too complex and performs very well on the training data but poorly on new data. Underfitting occurs when a model is too simple and performs poorly on both the training data and new data.

Table II, outlines the performance of various deep learning architectures on a ship classification task. The models were evaluated using three metrics (expressed as a percentage): model Loss, Error Rate (ER), and Accuracy. The ResNet architectures (ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152) have the lowest ER and high accuracies, with ResNet101 having the lowest ER of 0.27%. ResNet101 also has the lowest model loss of 0.001%, indicating that it is a very robust and accurate model for the ship classification task. SqueezeNet1.1 has a much higher ER and lower accuracy compared to other models, indicating that it may not be the best choice for this task. The other models, including DenseNet architectures (DenseNet121, DenseNet161, DenseNet169, and DenseNet201), VGG16, VGG19, and AlexNet, have varying levels of performance, with some models performing better than others.

Loss (1): Commonly used in machine learning called the categorical cross-entropy loss. This loss function is used when dealing with classification problems where the output variable (y) and the predicted output variable (ŷ) are both categorical.

$$\text{Loss}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{i,j} log(\hat{y}_{i,j}). \tag{1}$$

Error Rate (2): Provides a measure of the models performance in terms of misclassification, indicating the proportion of samples that were classified incorrectly.

$$\text{Error Rate} = \frac{\text{Number of misclassified images}}{\text{Total number of images}}. \tag{2}$$

Accuracy (3): Is a commonly used metric to evaluate the performance of a classification model. It measures how well the model predicts the correct classes for a given set of data.

$$\text{Accuracy} = \frac{\text{Number of correctly classified images}}{\text{Total number of images}}. \tag{3}$$

Overall, the ResNet architectures appear to be the best choice for the ship classification task, with ResNet101 being the most accurate and robust model. However, the choice of model ultimately depends on the specific requirements of the task, and a careful analysis of each model's strengths and weaknesses is necessary to make an informed decision.

Although the sum of the two metrics is equal to 1 due to the complementary nature of the two, each metric brings valuable information about different aspects of the model, which allows for a more complete and informed analysis of its performance. For example, the ER metric represents the ratio of incorrect classifications to total samples. This metric provides a direct idea of the model's ER, which can be crucial in scenarios where the costs associated with classification errors are asymmetric. In addition, the ER can help identify specific classes or situations where the model is having difficulties. The Accuracy metric, on the other hand, measures the proportion of correct classifications out of the total samples. It is a widely used metric to assess the overall performance of the model, providing an overview of how well it is doing in terms of correct predictions. Accuracy is especially useful when all

TABLE II
MODEL PERFORMANCE METRICS.

Model	Loss %	Error Rate %	Accuracy %
resnet18	1,85	0,45	99,54
resnet34	1,50	0,30	99,69
resnet50	1,13	0,36	99,63
resnet101	0,001	0,27	99,72
resnet152	1,70	0,30	99,69
squeezenet1.0	1,94	0,57	99,42
squeezenet1.1	23,94	6,01	93,98
densenet121	0,95	0,27	99,72
densenet161	0,91	0,24	99,75
densenet169	2,02	0,42	99,57
densenet201	1,11	0,30	99,69
vgg16	2,02	0,54	99,45
vgg19	0,82	0,27	99,72
alexnet	2,55	0,84	99,15

classes are of relatively equal importance and when there is no significant imbalance between classes.

Figure 3 illustrates several performance metrics of the model during training. It is noticeable the homogeneity in the behavior of all models, although some excel in the first few training epochs. The model that presents the worst performance during this stage is Squeezenet in version 1.1. However, the first version of the architecture performed better in the same dataset and training parameters. Figure f.a illustrates ER Curve, where it is visible the homogeneity in the behavior of all models, although some excel in the first few training epochs. However, the first version of the architecture performed better in the same dataset and training parameters, requiring most studies about this performance disparity. However, it is important to note that transfer learning applied from pre-trained models on the COCO dataset seems to have increased the speed of convergence of the models. For, it is notable a good performance of all models at the beginning of the training, presenting a ER of less than 10%. However, a comparison between both training methods is necessary in order to guarantee this hypothesis.

Continuing with the analysis, it is evident from Figure f.b that all models exhibit a consistent behavior during the training epochs, with the exception of Squeezenet 1.1. Initially, all models start with an initial loss of less than 30% in the validation set, indicating some level of understanding of the ships dataset. As the training progresses, the models gradually improve their accuracy and approach values close to 0% loss, indicating convergence. However, Squeezenet 1.1 stands out as the model with the worst performance during this stage (23.94%). Despite the other models showing consistent improvement, Squeezenet 1.1 seems to struggle to converge and achieve a low loss value. This performance disparity raises the need for further investigation and analysis. Interestingly, when considering the performance of the first version of the model, it outperformed the version 1.1 under the same dataset and training parameters. This observation highlights the importance of understanding the architectural changes and modifications made between the two versions that may have affected the model's performance

Further, Figure f.c presents the models loss on the training set. Similarly, it is possible to see homogeneity in the behavior of all models during the epochs. All models showed an accuracy above 92% in the very first epochs and showed a convergence to the steady state in less than 20 epochs. Again, the Squeezenet 1.1 model showed turbulent training relative to the others and does not seem to have converged by the end of the 25 epochs. The Squeezenet 1.1 model have the less complex and intricate architecture compared to the other models. This lack of complexity could lead to challenges in optimizing the model during training, resulting in a turbulent training process. The hyper-parameters used during training, such as learning rate, batch size, or optimization algorithm, can significantly impact the training process. If the hyper-parameters were not well-tuned for the Squeezenet 1.1 model, it could contribute to the observed turbulent behavior. Inappropriate hyper-parameter settings might cause unstable updates, hinder convergence, or result in oscillating loss values.

Finally, the Figure f.d shows and compare the Loss Curve in the training set. It is evident that all models started with a relatively low loss of less than 6%. This initial performance indicates that the models had a basic understanding of the training data right from the beginning. Notably, the behavior of Squeezenet 1.1 stands out in this comparison. Initially, it lagged behind the other models, exhibiting a slower convergence rate. However, as the training progressed, Squeezenet 1.1 gradually caught up with the other models and approached similar loss values. This observation indicates that although Squeezenet 1.1 had a slower start, it eventually managed to learn and converge effectively. The homogeneity of the Loss Curve among the models further highlights their similar learning patterns and abilities to adapt to the training data. Despite their architectural differences and performance disparities during the initial stages, the models demonstrate a shared characteristic of gradually converging to lower loss values in the training set.

IV. CONCLUSIONS

In conclusion, this study has demonstrated the potential of CV technology for maritime monitoring, particularly in ship classification. The comparison of different CV architectures and techniques has highlighted their strengths and weaknesses, providing valuable insights for future research. The numerical results obtained during the training and validation of these architectures have shown promising performance in identifying and tracking vessels. By improving maritime monitoring through CV technology, authorities can enforce regulations, improve trade efficiency, and enhance national security. Overall, this study contributes to the growing body of research on how a CV can be used to enhance various applications in the maritime industry.

REFERENCES

[1] G. Melillos et al., "The use of remote sensing for maritime surveillance for security and safety in Cyprus," Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXV. SPIE, Apr. 24, 2020. doi: 10.1117/12.2567102.

[2] M. Chua, D. W. Aha, B. Auslander, K. Gupta, and B. Morris, "Comparison of Object Detection Algorithms on Maritime Vessels," Defense Technical Information Center, Jan. 2014. doi: 10.21236/ada619022.

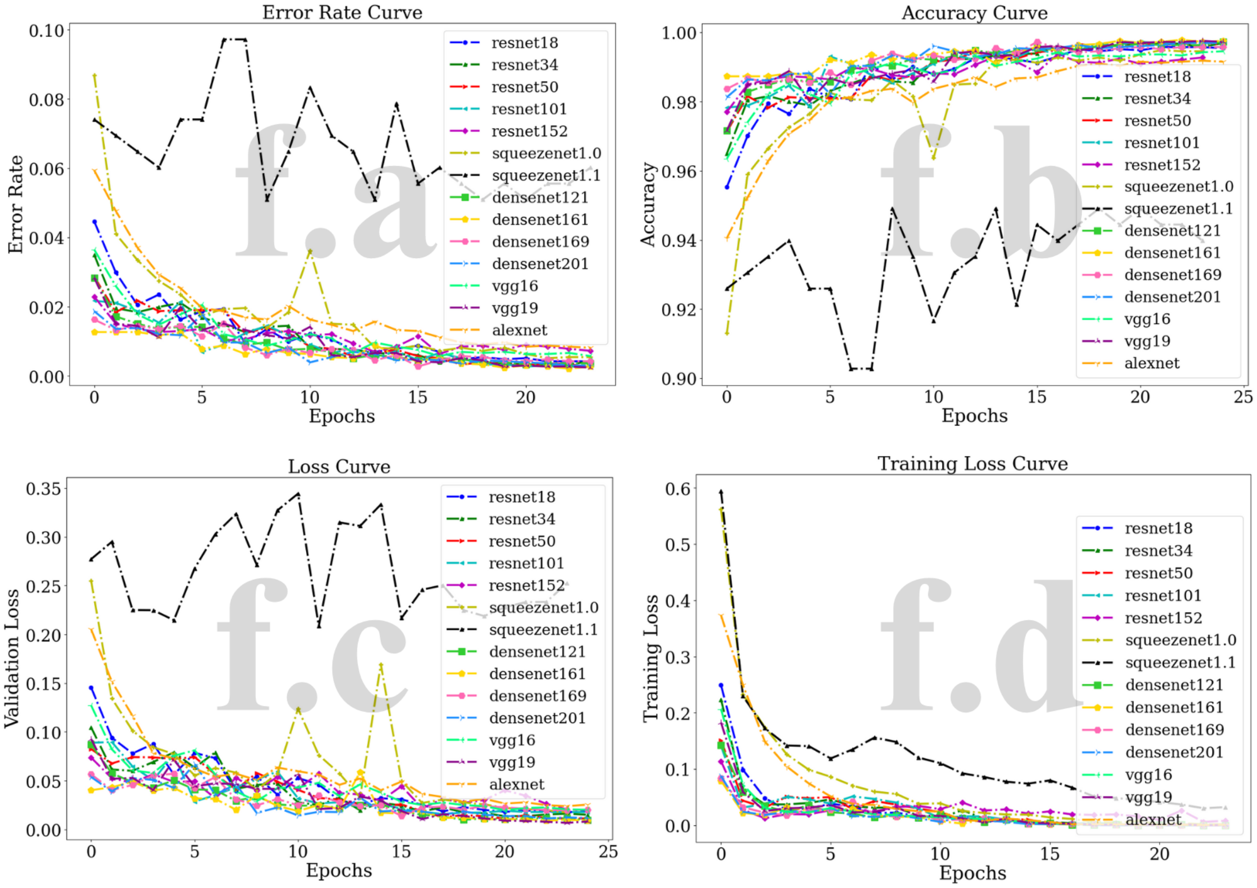

Fig. 3. Curves for Error Rate, Accuracy, and Loss presented by all devices during during the FL training.

[3] S. Moosbauer, D. Konig, J. Jakel, and M. Teutsch, “A Benchmark for Deep Learning Based Object Detection in Maritime Environments,” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, Jun. 2019. doi: 10.1109/cvprw.2019.00121.

[4] “Object recognition by computer: the role of geometric constraints,” Choice Reviews Online, vol. 29, no. 05. American Library Association, pp. 29-2747-29–2747, Jan. 01, 1992. doi: 10.5860/choice.29-2747.

[5] H. Lang, J. Zhang, X. Zhang, and J. Meng, “Ship Classification in SAR Image by Joint Feature and Classifier Selection,” IEEE Geoscience and Remote Sensing Letters, vol. 13, no. 2. Institute of Electrical and Electronics Engineers (IEEE), pp. 212–216, Feb. 2016. doi: 10.1109/lgrs.2015.2506570.

[6] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, “Review of Image Classification Algorithms Based on Convolutional Neural Networks,” Remote Sensing, vol. 13, no. 22. MDPI AG, p. 4712, Nov. 21, 2021. doi: 10.3390/rs13224712.

[7] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” Insights into Imaging, vol. 9, no. 4. Springer Science and Business Media LLC, pp. 611–629, Jun. 22, 2018. doi: 10.1007/s13244-018-0639-9.

[8] Amrita Biswas, M K Ghose and Moumee Pandit. Article: Comparison of Different Neural Network Architectures for Classification of Feature Transformed Data for Face Recognition. International Journal of Computer Applications 96(12):25-31, June 2014. Full text available.

[9] S. Sharma, K. Senzaki and H. Aoki, "Comparative Study of Feature Extraction Approaches for Ship Classification in Moderate-Resolution SAR Imagery," IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 2018, pp. 6995-6998, doi: 10.1109/IGARSS.2018.8518966.

[10] T. -H. Tran and T. -L. Le, "Vision based boat detection for maritime surveillance," 2016 International Conference on Electronics, Information, and Communications (ICEIC), Danang, Vietnam, 2016, pp. 1-4, doi: 10.1109/ELINFOCOM.2016.7563033.

[11] D. K. Prasad, H. Dong, D. Rajan and C. Quek, "Are Object Detection Assessment Criteria Ready for Maritime Computer Vision?," in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 12, pp. 5295-5304, Dec. 2020, doi: 10.1109/TITS.2019.2954464.

[12] Y. Yang, R. Yan, and S. Wang, “Integrating Shipping Domain Knowledge into Computer Vision Models for Maritime Transportation,” Journal of Marine Science and Engineering, vol. 10, no. 12. MDPI AG, p. 1885, Dec. 04, 2022. doi: 10.3390/jmse10121885.

[13] L. Su, Y. Chen, H. Song, and W. Li, “A survey of maritime vision datasets,” Multimedia Tools and Applications. Springer Science and Business Media LLC, Mar. 01, 2023. doi: 10.1007/s11042-023-14756-9.

[14] A. Mahajan and S. Chaudhary, “Categorical Image Classification Based On Representational Deep Network (RESNET),” 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA). IEEE, Jun. 2019. doi: 10.1109/iceca.2019.8822133.