Comparative study of feature extraction approaches for maritime vessel identification in CBIR

Bryan L. G. dos Santos[1,2], Ana C. Lorena[1], Juliano E. C. Cruz[2]

[1]Instituto Tecnológico de Aeronáutica (ITA)
São José dos Campos – SP – Brazil

[2]Empresa Brasileira de Aeronáutica (EMBRAER)
São José dos Campos – SP – Brazil

Abstract. *Maritime surveillance and monitoring systems are crucial in coastal security and resource management. Vessel recognition and identification are key tasks. However, visual inspection is a costly and labour-intensive process. This study compares methods for an automated approach for vessel identification using digital image processing. The performance of classical and Machine Learning-based feature extraction methods is evaluated and compared using a maritime vessel dataset to verify their ability to identify different vessels. The results show that BEiT-v2 achieves the highest identification performance with a mean Average Precision (mAP) of 95.05%. VGG-19 offers the best balance between accuracy (second-highest mAP) and computational cost. These findings suggest that Machine Learning methods are valuable for vessel identification, with the optimal choice depending on the specific needs of the application.*

Resumo. *Os sistemas de vigilância e monitorização marítima são cruciais na segurança costeira e na gestão de recursos. O reconhecimento e identificação de navios são tarefas fundamentais. No entanto, a inspeção visual é um processo caro e trabalhoso. Este estudo compara métodos para uma abordagem automatizada para identificação de navios utilizando processamento digital de imagens. O desempenho dos métodos de extração de características clássicos e baseados em aprendizado de máquina é avaliado e comparado usando um conjunto de dados de embarcações marítimas para verificar sua capacidade de identificar diferentes embarcações. Os resultados mostram que o BEiT-v2 atinge o mais alto desempenho de identificação com uma precisão média média (mAP) de 95,05%. O VGG-19 oferece o melhor equilíbrio entre precisão (segundo maior mAP) e custo computacional. Estas descobertas sugerem que os métodos de aprendizagem automática são valiosos para a identificação de embarcações, com a escolha ideal dependendo das necessidades específicas da aplicação.*

1. Introduction

Ship recognition and identification are critical tasks for various maritime applications, including surveillance, national defence, port traffic management, and pollution control [Guo et al. 2021]. Effective identification of inshore and offshore vessels is crucial for monitoring maritime traffic, preventing illegal activities like smuggling and pollution dumping in fisheries, and ensuring safe navigation.

Within the realm of visual analysis, object detection, recognition, and identification represent a hierarchical sequence of tasks [Biberman 1973, Driggers et al. 1997].

Detection establishes the presence of an object without classifying it. Recognition elevates understanding by assigning the object to a broader category. Recognizing the visual blob as a vessel or other object helps move forward in the visual understanding process, but this step does not differentiate between a cargo ship, fishing boat, or other types. Finally, identification achieves the most granular level of detail. Identifying the specific vessel class or even a specific ship provides the most detailed and substantial information.

Traditionally, video ship detection, recognition, or identification has relied significantly on staff monitoring screens, which is inefficient, costly, and can lead to misjudgments [Shao et al. 2018]. Computer vision and image processing technology have been recently widely used in maritime surveillance to solve these problems. The main sources of images available for this purpose are: satellite imagery [Bo et al. 2021], Synthetic Aperture Radar (SAR) imagery [Chang et al. 2019], and infrared and visible imagery from on-ground cameras [Liu et al. 2020]. Satellite imagery usually covers a large area, but the revisit period is relatively long and can be affected by clouds and fog [Chen et al. 2019]. Methods based on airborne SAR are relatively mature, but cannot detect ships in scan-blind zones, which are mainly affected by rain and waves [Li et al. 2018]. Detection in infrared images from on-ground cameras has good penetrability on clouds and fog. However, there are no visual features such as color and texture, and the image is affected by sea thermal radiation and atmospheric effects [Xie et al. 2017a]. Compared to other image detection methods, visible spectrum imagery contains rich color, texture, and spectral information, but its application is limited to daytime.

One way to identify a maritime vessel with digital image processing is to extract a set of descriptors or features from the image, compare them to a database of ship features, and measure the similarity between those sets of features. In a match, the resulting similar vessel is returned to the user; in a mismatch, the new set of features can be added to the database as a new ship. This method of searching for a similar image with another image as a query is known as Content-Based Image Retrieval (CBIR). Feature extraction is one of the first steps in CBIR, which converts visual data into numerical descriptions that machines can manipulate. The accuracy in similarity of the retrieved images is greatly affected by the quality of the extracted features [Piras and Giacinto 2017]. A recent trend in image retrieval research focuses on using deep learning to improve accuracy at the expense of a higher computational cost, particularly at the training phase [Markowska-Kaczmar and Kwaśnicka 2018].

Visual features from classical approaches are generally designed following heuristics and can be categorized into local and global features. These local and global features are usually grouped into low-level features and are often hand-crafted. Machine Learning-based feature extraction has gained increased attention in the last decade, being able to automatically extract features from the images by training an ML model on sets of images [Hameed et al. 2021]. This study compares low-level and ML-based descriptors used in CBIR for maritime vessel identification. State-of-the-art (SOTA) deep learning methods have shown remarkable retrieval accuracy compared to classical low-level methods but at a higher computational cost.

This paper is structured as follows: Section 2 presents the overall CBIR architecture and steps; Section 3 presents the materials and methods employed in this study; Section 4 presents the experimental results achieved; and finally, Section 5 concludes this

paper with the most important findings and future work.

2. CBIR

A typical CBIR framework consists of stages that can usually be divided into offline and online, as shown in Fig. 1. In the offline stage, a database of features of known images is built, which will be used later when the user performs queries to the system. The online stage is when the user actively uses the system, inputting new images to be recognized by the system, given its stored database.

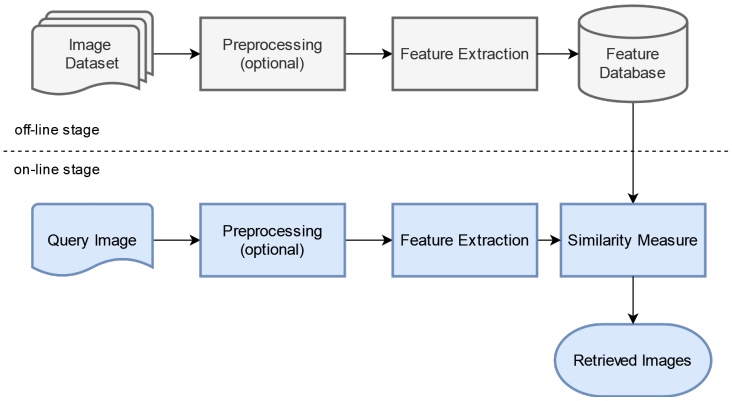

Figure 1. A typical CBIR framework.

An optional image preprocessing stage can be included in the framework's architecture. This includes resizing, segmentation, denoising, scaling, etc. This optional stage is followed by the feature extraction stage, in which a visual concept is converted to a numerical form. The extracted features can be low-level features, i.e., color, shape, texture, spatial information, or local descriptors [Hameed et al. 2021]. The final stage measures the similarity between the extracted features from the query image and all other images in the database to retrieve the most relevant images and return them to the user.

The main problem of CBIR is measuring the similarity between images correctly. Images cannot be directly compared at the pixel level because objects in images can undergo various changes and transformations, such as background changes, brightness, resolution, point of view, etc. Therefore, to efficiently compute similarities, we must define a way to represent these images, which is done by descriptors or feature extractors.

2.1. Image descriptors

Classically, an image can be represented by global and local features. In the global feature representation, images are represented by one multi-dimensional feature vector that describes information about the entire image. Global features can be interpreted as specific properties of the image that affect all pixels, such as color, texture, shape, histogram, edges, or even a specific descriptor extracted from some filters applied to the image [Oliva and Torralba 2001]. In contrast, the main purpose of local feature representation is to be invariant to viewpoint and illumination changes yet still be able to distinguish images based on some salient regions. Based on its local structure, an image is represented by a set of local feature descriptors extracted from a set of image regions or key points [Awad and Hassaballah 2016].

Global features are much faster and more compact while also being easier to compute and generally using less memory. Nevertheless, the global representation is not invariant to significant transformations and is sensitive to clutter and occlusion. Local features are expected to be more useful for image matching and object identification because local structures are more pronounced and stable [Bianco et al. 2015].

Advancements in the last decade in ML have revealed the remarkable ability of deep learning models to capture transferable image representations [Markowska-Kaczmar and Kwaśnicka 2018]. These models, trained on vast image datasets, utilize the initial layers to extract generic image features, which can be later transferred to other domains. This approach facilitates the automatic generation of image descriptors, albeit with limited direct interpretability of the learned features.

2.2. Measuring Similarity

The retrieval performance of a CBIR system is affected by how the features are extracted and how those features are compared to other feature sets in the database. Therefore, the similarity measurement directly impacts the accuracy of the system. One widely used distance measure is the Minkowski family distance due to its simple computation and implementation [Hameed et al. 2021]. It is a bin-by-bin distance function, so for each element $x_i \in X$ and $y_i \in Y$ this distance function only compares x_i with y_i and it can be calculated as:

$$L_p(X, Y) = \left(\sum_{i=1}^{N} |x_i - y_i|^p \right)^{\frac{1}{p}}, \tag{1}$$

where X and Y are two feature vectors and N is the total dimensionality of the feature sets. The Minkowski distance is also called L_p-Norm. For $p = 1$, the L_1 distance, also known as the Manhattan distance, is computed, which is variant to coordinate system rotation but robust to reflections and translations. If $p = 2$, L_2 distance is calculated. It is also known as Euclidean distance and is invariant to orthogonal transformation. If $p = \infty$, then it is the L_∞ distance, also known as Chebyshev distance, which is computationally efficient and robust to outliers, but with the cost of loss of information.

Another common distance measure used is the cosine distance. Given two n-dimensional feature vectors X and Y, this distance can be calculated by the angle θ between these two vectors as:

$$d(X, Y) = 1 - cos\theta = 1 - \frac{X \cdot Y}{\|X\| \cdot \|Y\|}. \tag{2}$$

2.3. Performance Evaluation

The main metrics for retrieval evaluation are precision, recall and mean average precision [Alzu'bi et al. 2015]. Precision (P_k) is the ratio of the relevant images (true positives) retrieved within the first k results to the total number of images retrieved:

$$P_k = \frac{\text{No. of relevant images retrieved}}{\text{No. of retrieved images}} \tag{3}$$

Recall (R_k) is the ratio of the number of relevant images retrieved within the first k results to the total number of relevant images and is defined as:

$$R_k = \frac{\text{No. of relevant images retrieved}}{\text{No. of relevant images}}. \tag{4}$$

For a single user query q, the average precision (AP) is the mean value of the precision values at every relevant image and is defined as:

$$AP_q = \frac{1}{N_{RI}} \sum_{k=1}^{N_{RI}} P_k(R_k) \tag{5}$$

where N_{RI} is the number of relevant images in the database for the current query image.

For a set of queries N_Q, the mean average precision (mAP) is the mean of the average precision scores over all queries, defined as:

$$mAP_Q = \frac{1}{N_Q} \sum_{q=1}^{N_Q} AP_q. \tag{6}$$

3. Methodology

3.1. Dataset

Some well-known image datasets have vessel images, such as ImageNet [Deng et al. 2009], PASCAL Visual Object Classes (VOC) [Everingham et al. 2015], and Microsoft Common Objects in Context (MS COCO) [Lin et al. 2015]. However, the total number of vessels is limited and multiple images of the same vessel are uncommon. One way to distinguish between vessels is by its IMO (International Maritime Organization), a ship identification number scheme which assigns a unique and permanent number to each ship. The MARVEL [Gundogdu et al. 2016] dataset is a large-scale image dataset for maritime vessels, consisting of 2 million user-uploaded images and their attributes, including vessel identity, type, photograph category, and year of construction, collected from a community website. It has over 35,000 unique vessels with more than 10 examples per vessel. The dataset used in this paper (Fig. 2) is a subset of MARVEL. It has 100 unique ships with 42 images of each, where they appear in multiple environments, weather and lighting conditions, resolutions, and different viewpoints (Fig. 2).

Figure 2. Example of the dataset samples in MARVEL.

3.2. Feature extraction methods

Classical and SOTA feature extraction methods were chosen for this study. The classical ones are SIFT, ORB, and FREAK due to their importance in the CBIR field. The SOTA methods selected are all deep learning-based methods, mainly pre-trained CNNs and Transformers, chosen given their outstanding performance on image classification (top-1 accuracy). The SOTA methods were first trained on ImageNet-1K dataset and the specialized to this paper's dataset with transfer learning. The feature descriptor is obtained in the last layer before the fully connected network in charge of classification.

Scale-invariant feature transform (SIFT) [Lowe 2004] is one of the most commonly used local descriptors. It includes both a key point detector and a descriptor. SIFT is robust against image rotation and scaling, but it performs poorly in matching at high dimensions and needs a fixed-size vector for encoding to check image similarity.

ORB [Rublee et al. 2011] aims to enhance image-matching applications in low-power devices without GPU acceleration while maintaining accuracy. ORB is built on the FAST (Features from Accelerated Segment Test) key point detector [Rosten and Drummond 2005] and the BRIEF (Binary Robust Independent Elementary Features) descriptor [Calonder et al. 2010], hence the name ORB (Oriented FAST and Rotated BRIEF).

FREAK (Fast Retina Keypoint) detects the key points using an approach inspired by the human retina [Alahi et al. 2012]. FREAK is a binary descriptor and to sample the pairs of pixels to compare intensity, it uses the retinal sampling grid, which is a circular grid with a higher density of points near the center.

The Inception architecture [Szegedy et al. 2014] uses a parallel filter strategy within a single module to capture multi-scale features from the input image. It employs different convolutional filters applied concurrently to extract both local and global information. The resulting outputs from these parallel branches are then concatenated, forming a richer feature representation while reducing the number of parameters. The version V4 presents a more uniform and simplified architecture [Szegedy et al. 2017].

The Visual Geometry Group (VGG) network architecture [Simonyan and Zisserman 2014] is characterized by its simplicity and efficacy. It leverages repeated stacks of 3x3 convolutional layers, capturing essential spatial information like vertical and horizontal edges. The number after VGG refers to the number of layers with VGG-16 or VGG-19, respectively, consisting of 16 and 19 convolutional layers.

Residual Networks (ResNets) introduced the concept of residual learning [He et al. 2016]. This framework addresses the vanishing gradient problem that hinders the training of very deep networks. ResNets address this by introducing residual blocks, which bypass the main layers with a direct identity connection. ResNet-200D has two hundred layers and also employs average pooling for downsampling. ResNeXt101 [Xie et al. 2017b] is based on the ResNet architecture and it is constructed by repeating blocks that aggregate a set of transformations with the same topology.

MobileNet [Howard et al. 2017] was introduced as a new class of efficient models for mobile and embedded vision applications. This architecture introduced the depthwise separable convolutions as a replacement for traditional convolution layers. MobileNetV3 implements some improvements which maintain the accuracy of previous versions but

with a considerably lower inference time.

DenseNet [Huang et al. 2017] was designed to ensure maximum information flow between layers in the network. This was achieved by connecting all layers, with matching feature-map sizes, directly with each other.

The Vision Transformer (ViT) [Dosovitskiy et al. 2020] divides an input image into fixed-size, non-overlapping patches. These patches are then linearly projected into lower-dimensional embeddings, and positional encodings are added to each patch embedding to incorporate spatial information. The resulting sequence of vectors is then fed into a standard Transformer encoder.

BEiT [Bao et al. 2021], Bidirectional Encoder representation from Image Transformers, is a self-supervised vision representation model that uses a pre-training phase called Masked Image Modeling (MIM). MIM generates image patches from each image, as in ViT, and visual tokens. During pre-training, some of the image patches are randomly masked. The model learns to recover the visual tokens of the original image even though some of the image patches are masked. The key enhancement introduced by BEiT-v2 [Peng et al. 2022] was to employ a semantic-rich visual tokenizer for the token generation during a pre-training phase.

4. Results

The dataset is usually split into training and test using an 80/20 ratio in CBIR systems employing neural network [Qiao et al. 2020]. Out of the 80% used for training, 70% was used for training the networks and 30% for validation. The 20% for test is used as the query dataset and the remaining 80% is used as gallery dataset, simulating the feature database. Classical methods do not require training, therefore only the split for query and gallery dataset was used.

4.1. Classical Methods

Fig. 3 shows the mAP for the classical methods for increasing numbers of retrieved images. As expected, given the changes in the images in the dataset, none of the classic methods performs satisfactorily for this kind of application, with very low mAP values.

Tab. 1 summarizes the characteristics and performance of the classical methods. Calculation time is the time it takes for each method to detect and compute the features of a given image, presented as an average over the entire dataset. ORB was the best method concerning mAP, however the performance is still very low (5.65%).

Table 1. Summary of classic methods.

Model	Feature Vector Dimension	mAP [%] @N=30	Calc. Time [s]
ORB	512×32	**5.65**	0.062
FREAK	512×64	1.73	0.293
SIFT	512×128	0.85	0.731

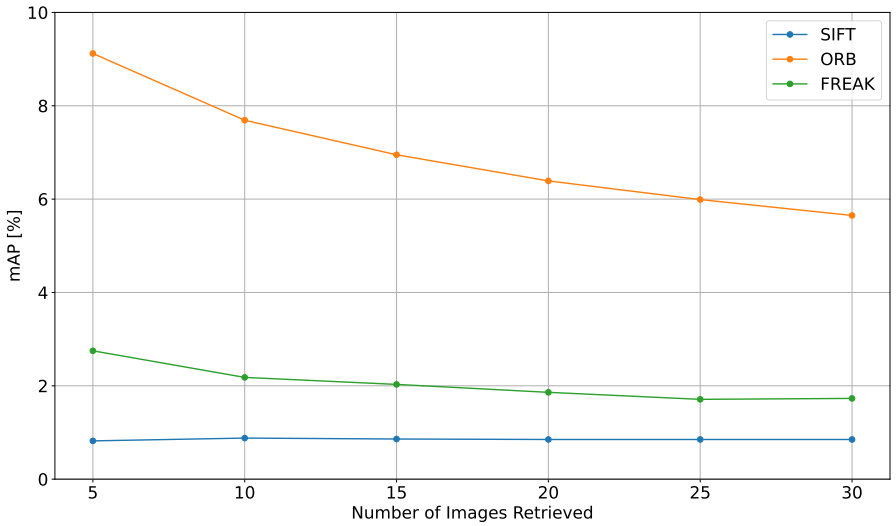

Figure 3. SIFT, ORB and FREAK mAP.

4.2. SOTA Methods

Tab. 2 presents the hyperparameters used among all models for training. All models were trained in an RTX 3070 with 8GB of memory. Tab. 3 summarizes the results of SOTA methods, and Fig. 4 summarizes the best methods analyzed over the number of retrieved images. Cosine and Euclidean distance were the prevailing dissimilarity metrics, Manhattan came close for some models, and Chebyshev always underperformed the others. When comparing the different models with 30 images retrieved ($N = 30$), VGG-19 presents a high mAP of 94.99%, with the more recent models around the same accuracy rate. BEiT-v2 was the best with 95.05% of mAP despite having the smallest feature descriptor, with only 768 features. Regarding inference time, VGG-19 was almost an order of magnitude faster than the other models. All models were implemented using the same library, the inference was run in the same computer, and no optimization was made.

Table 2. Training hyperparameters.

Parameter	Value
Pre-Trained on:	ImageNet-1K
Epochs	300
Batch size	8
Optimizer	SGD
Initial LR	3.0e-3
LR Scheduler	Cosine
Decay Rate	0.1
Weight Decay	2.0e-5
Warm-up Epochs	5
RandAugment (M / N / MSTD)	9 / 2 / 0.5
EMA Decay	0.99998

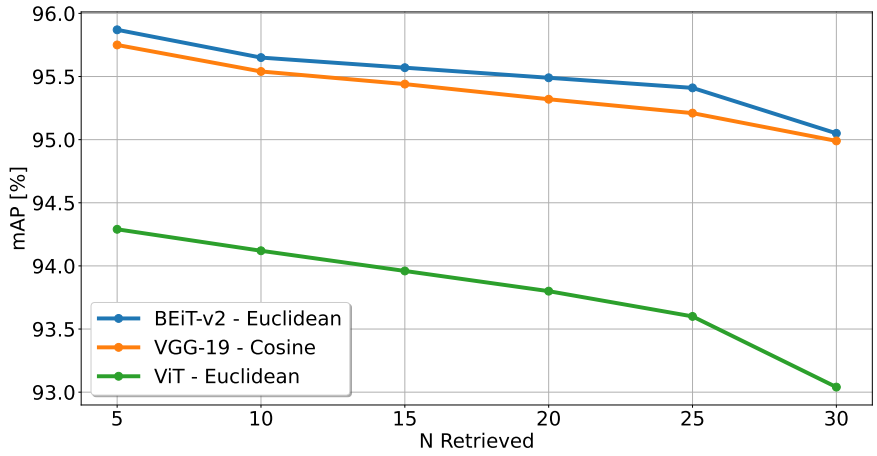

Figure 4. Summary of the best performing state-of-the-art methods.

Table 3. Summary of results of SOTA methods.

Model	Metric	mAP [%] @N=30	Number of Params [M]	Number of Features	Inference Time [μs]
BEiT-v2	Euclidean	**95.05**	85.8	**768**	8.16
VGG-19	Cosine	**94.99**	140.0	4096	**0.84**
ViT	Euclidean	**93.04**	87.6	**768**	7.59
ResNeXt-101	Cosine	79.85	81.6	2048	9.65
ResNet-200d	Cosine	77.30	62.8	2048	14.48
DenseNet-201	Euclidean	54.98	18.3	1920	15.22
MobileNetV3	Euclidean	31.17	4.3	1280	5.03
Inception-v4	Cosine	11.12	41.3	1536	11.20

Fig. 5 shows the Precision-Recall curve as the number of images retrieved increases. BEiT-v2 and VGG-19 have very similar behavior, maintaining a high precision as the recall increases. Conversely, the Inception-v4 and MobileNetV3 models showed non-satisfactory results with low precision values. Overall, the area under the curves of the VGG-19 and BEiT-v2 models are the highest, evidencing their ability to cope with the precision-recall trade-off in this application.

5. Conclusions

This study comparatively evaluated CBIR approaches for automated vessel identification using various distance metrics. Results shown that BEiT V2 presented outstanding results, with 95.05% mAP and so does VGG-19, with 94.99% mAP, which also had the fastest inference time (0.84μs) They also presented better precision-recall curves, managing to keep an overall trade-off of both metrics high. Neural networks with Transformer architecture performed better in average than the convolutional architectures, where in the top 3 best mAP, two of them use Transformers. Other advantages are their short feature vector (768) and low inference time (between 7.5 and 8.2 μs). Classical descriptor approaches had the worst retrieval results. They are not good or able to generalize according

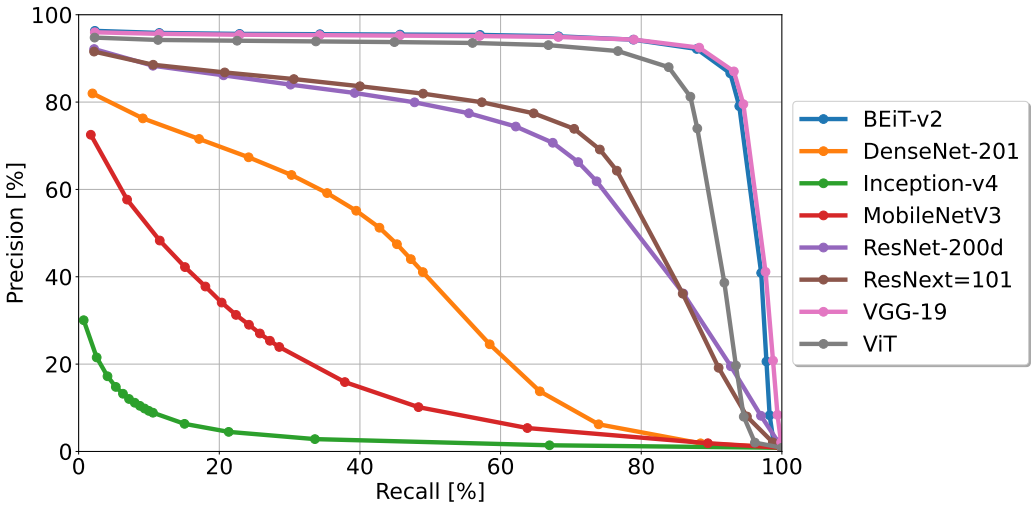

Figure 5. Precision and Recall curve for different deep-learning methods.

to the object itself, but only to images captured that are very near in the spatiotemporal dimension.

Further investigations may include evaluating network generalization with larger datasets, exploring image pre-processing techniques, and optimizing inference speed through multi-layer output fusion or hyperparameter tuning.

Acknowledgments

To the research agencies FAPESP (grant 2021/06870-3) and CNPq and to EMBRAER.

References

Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). Freak: Fast retina keypoint. In *Conference on Computer Vision and Pattern Recognition*, pages 510–517. IEEE.

Alzu'bi, A., Amira, A., and Ramzan, N. (2015). Semantic content-based image retrieval: A comprehensive study. *Journal of Visual Communication and Image Representation*, 32:20–54.

Awad, A. I. and Hassaballah, M. (2016). Image feature detectors and descriptors. *Studies in Computational Intelligence. Springer International Publishing, Cham.*

Bao, H., Dong, L., Piao, S., and Wei, F. (2021). Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*.

Bianco, S., Mazzini, D., Pau, D. P., and Schettini, R. (2015). Local detectors and compact descriptors for visual search: a quantitative comparison. *Digital Signal Processing*, 44:1–13.

Biberman, L. (1973). *Perception of displayed information*. Plenum Press.

Bo, L., Xiaoyang, X., Xingxing, W., and Wenting, T. (2021). Ship detection and classification from optical remote sensing images: A survey. *Chinese Journal of Aeronautics*, 34(3):145–163.

Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). Brief: Binary robust independent elementary features. In *European Conference on Computer Vision*, pages 778–792. Springer.

Chang, Y.-L., Anagaw, A., Chang, L., Wang, Y. C., Hsiao, C.-Y., and Lee, W.-H. (2019). Ship detection based on YOLOv2 for SAR imagery. *Remote Sensing*, 11(7):786.

Chen, X., Qi, L., Yang, Y., Postolache, O., Yu, Z., and Xu, X. (2019). Port ship detection in complex environments. In *International Conference on Sensing and Instrumentation in IoT Era*, pages 1–6. IEEE.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929.

Driggers, R. G., Cox, P. G., and Kelley, M. (1997). National imagery interpretation rating system and the probabilities of detection, recognition, and identification. *Optical Engineering*, 36(7):1952–1959.

Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136.

Gundogdu, E., Solmaz, B., Yücesoy, V., and Koc, A. (2016). Marvel: A large-scale image dataset for maritime vessels. In *Asian conference on computer vision*, pages 165–180.

Guo, H., Yang, X., Wang, N., and Gao, X. (2021). A CenterNet++ model for ship detection in SAR images. *Pattern Recog.*, 112:107787.

Hameed, I. M., Abdulhussain, S. H., and Mahmmod, B. M. (2021). Content-based image retrieval: A review of recent trends. *Cogent Engineering*, 8(1):1927469.

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

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Conference on Computer Vision and Pattern Recognition*, pages 4700–4708. IEEE.

Li, R., Liu, W., Yang, L., Sun, S., Hu, W., Zhang, F., and Li, W. (2018). DeepUNet: A deep fully convolutional network for pixel-level sea-land segmentation. *Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(11):3954–3962.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (2015). Microsoft COCO: Common objects in context.

Liu, T., Pang, B., Ai, S., and Sun, X. (2020). Study on visual detection algorithm of sea surface targets based on improved YOLOv3. *Sensors*, 20(24):7263.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

Markowska-Kaczmar, U. and Kwaśnicka, H. (2018). Deep learning—a new era in bridging the semantic gap. In *Bridging the Semantic Gap in Image and Video Analysis*, pages 123–159. Springer.

Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175.

Peng, Z., Dong, L., Bao, H., Ye, Q., and Wei, F. (2022). BEiT v2: Masked image modeling with vector-quantized visual tokenizers.

Piras, L. and Giacinto, G. (2017). Information fusion in content based image retrieval: A comprehensive overview. *Information Fusion*, 37:50–60.

Qiao, D., Liu, G., Dong, F., Jiang, S.-X., and Dai, L. (2020). Marine vessel re-identification: A large-scale dataset and global-and-local fusion-based discriminative feature learning. *IEEE Access*, 8:27744–27756.

Rosten, E. and Drummond, T. (2005). Fusing points and lines for high performance tracking. In *International Conference on Computer Vision*, volume 2, pages 1508–1515. IEEE.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *2011 International conference on computer vision*, pages 2564–2571.

Shao, Z., Wu, W., Wang, Z., Du, W., and Li, C. (2018). Seaships: A large-scale precisely annotated dataset for ship detection. *IEEE transactions on multimedia*, 20(10):2593–2604.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Conference on Artificial Intelligence*, volume 31. AAAI.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going deeper with convolutions. *CoRR*, abs/1409.4842.

Xie, B., Hu, L., and Mu, W. (2017a). Background suppression based on improved top-hat and saliency map filtering for infrared ship detection. In *International Conference on Computing Intelligence and Information System*, pages 298–301. IEEE.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017b). Aggregated residual transformations for deep neural networks. In *Conference on Computer Vision and Pattern Recognition*, pages 1492–1500. IEEE.