On the use of appearance features for multiple object tracking in a maritime scenario

Luca Ballan, Richard J.M. den Hollander, Jan Baan, Friso G. Heslinga, and Wyke Huizinga

TNO Intelligent Imaging, Oude Waalsdorperweg 63, 2597AK The Hague (NL)

ABSTRACT

Multiple object tracking (MOT) interest has grown in recent years, both in civil and military contexts, enhancing situational awareness for better decision-making. Typically, state-of-the-art methods integrate motion and appearance features to preserve the trajectory of each object over time, using new detection information when available. Visual features are fundamental when it comes to solving temporary occlusion or complex trajectories, i.e. non-linear motion associated with high object speeds or low framerate. Currently, these features are extracted by powerful deep learning-based models trained on the re-identification (ReID) task. However, research focuses mostly on scenarios involving pedestrians or vehicles, limiting the adaptability and transferability of such methods to other use cases. In this paper we investigate the added value of a variety of appearance features for comparing vessel appearance. We also include recent advances in foundation models that show their out-of-the-box applicability to unseen circumstances. Finally, we discuss how the robust visual features could improve multiple object tracking performances in the specialized domain of maritime surveillance.

Keywords: deep learning, multiple object tracking, re-identification, appearance features, foundation models, decision support

1. INTRODUCTION

Situational awareness is critical for ensuring the safety and security of maritime activities, ranging from commercial shipping to coastal monitoring and search and rescue operations.[1,2] In this context, multiple object tracking (MOT) enables continuous monitoring of vessels and other maritime entities across video streams provided by electro-optical (EO) sensors. MOT can be particularly challenging due to the complexity of the scene and irregular object trajectories.[3]

Traditional tracking methods often rely on motion and geometric features,[4] but these are insufficient when trajectories are irregular or interrupted e.g. due to occlusion.[5] In contrast, features based on appearance do not require a regular object trajectory and allow for recognition of visual characteristics across sequential frames. However, dynamic maritime conditions, varying object appearances and the presence of similar-looking objects complicates appearance based tracking. Hybrid methods, that use both motion and appearance features, could combine the advantages of both methods.[6]

Xiao *et al.* proposed a motion-based tracker that solely relies on object trajectory information, as they find that appearance features are not discriminative enough.[7] Wang *et al.*, on the other hand, proposed SMILETrack, a method that is solely based on appearance features.[8] They use a siamese neural network-architecture to compute the similarity between targets. Luo *et al.* propose a diffusion-based tracker,[9] where bounding boxes are associated based on appearance features only. A tracker that uses both motion and appearance features was proposed by Zhang *et al.*[10] This tracker uses graph neural networks to associate detections based on motion features, appearance features and track history features.

Re-identification (ReID) methods, based on (short-term) visual consistency of individual objects, have emerged as a powerful tool to support the consistency of MOT, providing features that are robust against variable camera specifications, object orientation and time shifts.[11] The advent of deep learning (DL) has significantly advanced the extraction of appearance features, with ReID models and the more recent general-purpose foundation models

Corresponding author: Luca Ballan, E-mail: luca.ballan@tno.nl

at the forefront of this progress. ReID models are specifically designed to distinguish between different instances of objects, even in the presence of significant intra-class variations and inter-class similarities.[12]

Meanwhile, foundation models, such as Vision Transformers (ViTs)[13] pretrained on extensive datasets, offer robust feature representations that can be adapted to various downstream tasks.

Given the variety of appearance feature extractors, it is important to explore which appearance feature extractor associates the detections correctly for specific applications. In our work, we investigate the applicability of appearance features in a maritime scenario, where multiple object tracking is fundamental for continuous situational awareness. In such a maritime application, many ships can be in field-of-view, and the image quality in such a scenario can be low. This is in contrast to most MOT challenges, where it is often about person or vehicle tracking using high quality video data.

We research the efficacy of features from different feature extraction methods, and evaluate their performance on real-world maritime data. Our contributions include an analysis of the strengths and limitations of these features, relative to the variation of parameters related to the MOT task. We apply the feature extractors and analyse the features for association on a maritime dataset. This dataset includes vessels of different type, size, trajectory behavior, and challenging environmental conditions. We show that the best features for association in this dataset can be obtained using a model that was finetuned for ReID of ships.

Our work is organized as follows. Section 2 develops on the dataset and data processing methodologies, used to extract image features. The experimental setup, including adopted metrics for evaluation, is described in Section 3. Quantitative results follow in Section 4, for all the analysed methodologies. Finally, in Section 5 we elaborate on a number of discussion points, and conclude with possibilities for future work.

2. DATA & METHODOLOGY

In this section, we outline the framework used for our experiments on appearance features, in the context of MOT. A typical MOT framework consists of three primary components: object detection, feature extraction and data association. Object detection localizes objects of interest in each frame, feature extraction captures distinctive characteristics of the objects, and data association links these across frames to maintain consistent identities. These steps are therefore considered when building our data processing and evaluation pipelines. First, the dataset built for our experiments is described, and examples are given. Then, we show how appearance features are extracted. These will be used to compute different association metrics on the MOT task, in Section 3.

2.1 WHD TRACK DATASET

To evaluate our methodology, we use a newly build in-house video dataset of annotated tracks. The dataset consists of a set of vessel tracks selected from videos recorded during the 2016 World Harbor Days (WHD) in Rotterdam. The videos have a high resolution of 5120×3840 pixels (px) and a frame rate of 12 fps. A YOLOv5x[14] detection model, pretrained on the MS COCO dataset,[15] is used to detect all objects belonging to the class "boat". An in-house developed motion-based tracker[11] has been used to build tracks from raw detections, where interpolation has been applied to fill in missing bounding boxes. For a subset of the experiments we have smoothed the detection positions and dimensions in order to create a more stable track. Smoothed tracks can have more precise detections in case of temporary occlusion or single frame detection errors.

After tracking and interpolation, we have selected a subset of 480 tracks as our dataset. All tracks having a duration < 10 seconds, which are too similar to each other, or are static, are removed from the full dataset. The selected tracks were manually verified for containing the same vessel identity during their lifetime. We've also taken variation in vessel type, orientation, scale, background and complexity of trajectory into account for the selection. We will refer to the 480-tracks dataset as the WHD track dataset.

Figure 1 shows a schematic representation of the above mentioned steps, up to the evaluation setup described below. It also includes some detection examples. Figure 2 shows a distribution of the average bounding box size (area A in px) over the tracks, and an illustration of vessel appearance at different resolutions.

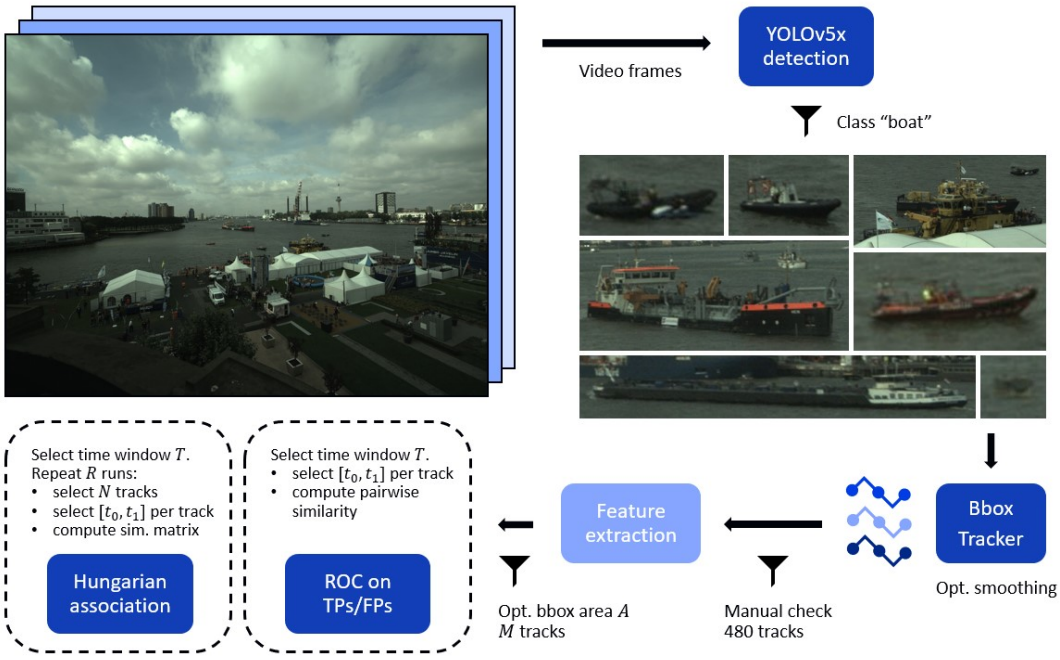

Figure 1. Visualization of the framework used for creating the WHD track dataset, including feature extraction and the evaluation of these features. A YOLOv5x detection model is run on the video frames, and all detections having class “boat” are subsequently tracked. Next, a subset of the tracks is selected manually for the evaluation experiments. In the evaluation, features are extracted at detection level and used to compute matches between vessel images. The quality of the matches is evaluated using several metrics.

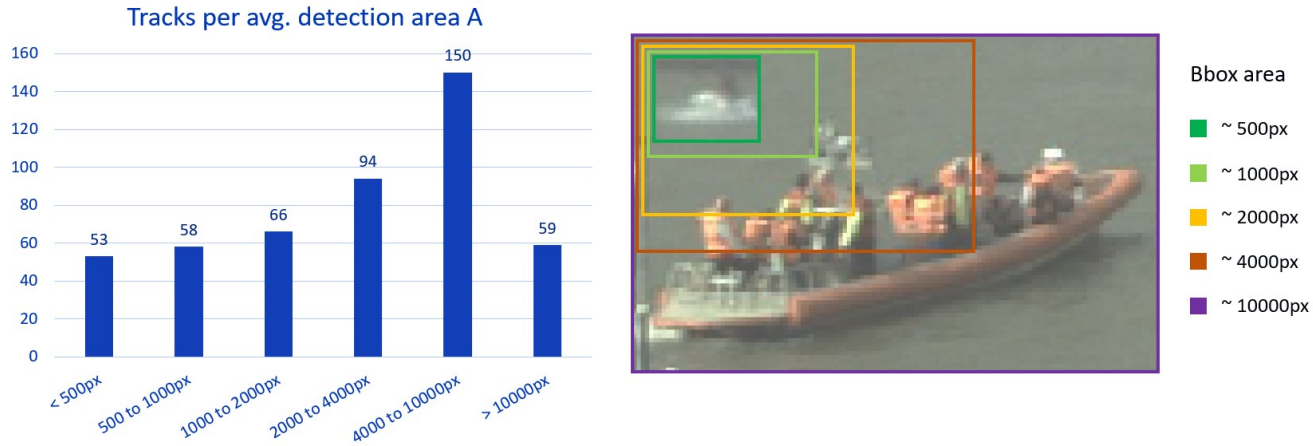

Figure 2. Distribution of the tracks in terms of average detection area A. A vessel sample is shown in the foreground at original resolution. In the background, a smaller Jet Ski is a good example of the smaller range of resolution ($A < 500$px).

2.2 APPEARANCE FEATURE EXTRACTION

For appearance features extraction from the WHD dataset, a deep neural network (DNN) is used. We used two types of DNN architectures: 1) a CNN model, ResNet50,[16] and 2) a transformer-based model, DINOv2.[17] We used two different models to quantify the difference in appearance features, i.e. to further analyse how different DL-based architectures behave as extraction backbones, for the MOT association task.

The feature vectors of each detection in each track are stored. Both models have a default input size of

Figure 3. Example of single Hungarian association experiment. N vessel tracks are selected from the WHD track dataset. For each of them, two timestamps $[t_0, t_1]$ are selected such that $t_1 - t_0 \approx T$. The bounding boxes (in red) are non-smoothed detections. In some cases only a portion of the vessel is included due to imprecision of the detector. On the right, the similarity matrix is computed using appearance features for the selected timestamps. In the shown example we used $N = 5$, $T = 10$, pretrained DINOv2 as feature extractor.

224 pixels. The output size of ResNet50's last convolutional layer is 2048, while DINOv2's output size depends on the size of its Vision Transformer backbone. We used the ViT-B (base) as backbone, which has an output size of 768. The models are comparable in size, having the same order of magnitude in terms of parameters. However, their pretraining strategies differ: ResNet50 was pretrained in a supervised manner on ImageNet1k,[18] whereas DINOv2 is pretrained in a self-supervised manner on a large dataset called LVD-142M, consisting of ImageNet-22k, the train split of ImageNet-1k, Google Landmarks and several fine-grained datasets. For a full overview see Table 15 in the appendices of Oquab *et al.*[17]

The detection bounding boxes are enlarged 20% in both dimensions before cropping the vessels from the original video frames, and computing the ResNet50 and DINOv2 feature vectors. As vessels have an elongated shape, we augment the cropped vessel images to 224×224px: we resize the largest dimension to 224px and consecutively pad the smallest dimension with zeros.

We also use a dedicated feature extraction method using DINOv2 characteristic patches: here we extract average patch tokens from the ViT-B backbone, to compare them to the more common layer called "class token". A 224×224px bounding box is cropped around a detection at original resolution (without resizing). We extract the local tokens corresponding to the 14×14px patches on the vessel - each being a 768 long vector - and average these into a single feature vector. For instance, a small speed boat of 10×25px will be described by two patch tokens. Background information and clutter is therefore discarded with this method. Important to note is that basically all vessels are below 224×224px in original size, therefore the cropping strategy previous to the patch tokens extraction fits all track samples in WHD; moreover, given the nature of the dataset, we assume that generally background information is not relevant to distinguish the identity of a vessel.

Besides DNN-feature extractors, we also used a classical feature extractor method as comparison. We adopt a simple strategy based on RGB histograms for a given pair of images. The similarity is computed as the cosine similarity between vectors of concatenated R,G, and B channel histograms for each of the images.

A summary of the feature extraction methods described above, and their input and output specifications, is visualized in Figure 4.

3. EXPERIMENTS

A number of factors contribute to the complexity of the task of MOT. Here we explain these factors in more detail.

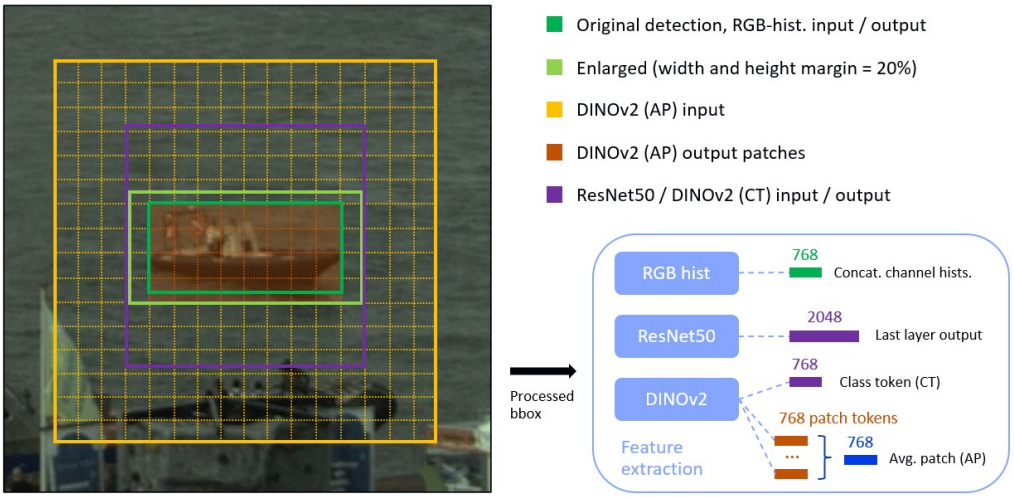

Figure 4. I/O specifications for the feature extraction methods. Color-coded are the processed portions of the vessel image (left), and corresponding feature vectors outputted by each adopted method (right).

First, there is a variable and potentially large number of entities to keep track of at a given time. Objects can move in and out of the scene, or enter a state of partial or complete occlusion. The number of vessels, N, contributes to the complexity of the scene, as chances are higher that with more vessels, the objects to be tracked can be confused. Second, let T be the time window in seconds between a measurement and the next one (typically corresponding to the frame rate or detection rate). With increasing T values, changes in appearance are likely to increase as well. A vessel can change its orientation, or move to larger distance from the sensor or become occluded. Third, the resolution impacts the quality of appearance features, as smaller and more distant vessels provide less information for distinguishing them. Finally, the last step of state-of-the-art MOT relies on an association mechanism, where detections of the same vessel identity are to be connected in a single track. Typically, the Hungarian algorithm,[19] or an extension of it, is used here to compute which combination of detections over time has the overall lowest cost. In terms of appearance features, a wrong association of two vessels with increasing T ideally has a higher cost, given by the lower similarity between their appearance.

Given the considerations above for the multiple object tracking scenario, we study the robustness and versatility of the various appearance features by defining two types of experiments. These are based on a sampling strategy using a predefined time window: given time T, a random pair of detections at timestamps t_0 and t_1 of a track (i.e. same vessel identity) is fetched, such that $t_1 - t_0 \approx T$. Following is the description of the two experiment types:

1. **Pairwise comparison**. We build a set of $2M$ pairs, where M is the number of tracks selected at any given experiment (when all tracks are selected, $M = 480$). Of these pairs, half are true associations between detections at timestamps t_0 and t_1 of each vessel, and the other half are pairs of wrongly associated detections. We select $[t_0, t_1]$ and the wrongly associated pairs randomly. We compute the features of these detections using the feature extractors, and with these features we compute the similarity between the pairs. The sampling is repeated R times. For example, if all tracks are used, and $R = 10$, there will be a total of $480 \times 2 \times 10 = 9600$ pairs, of which 4800 are positives (extracted by a random $[t_0, t_1]$ pair from each track 10 times), and the other half are negatives. The result are visualized in a ROC curve with TP and FP rates at different matching thresholds.

2. **Hungarian association**. We select N tracks randomly out of the WHD dataset. Again, we sample pairs of detections at $[t_0, t_1]$ according to a defined time window T. We then compute the $N \times N$ Hungarian association matrix between the N detections at timestamp t_0 and the N detections at timestamp t_1. The random selection is repeated R times. Finally, we count the number of correctly solved (H_c) and the

number of incorrectly solved (H_i) association matrices by the Hungarian algorithm. Accuracy is defined as ratio of correct association matrices

$$accuracy = \frac{H_c}{H_c + H_i} \tag{1}$$

The ROC will quantify the relation between correct and incorrect matches, and provides insight into how well the absolute matching score can prevent tracks to be switching between different vessel identities, and whether a matching threshold exists that will limit this number of false matches while keeping sufficient correct matches intact. The Hungarian association experiment will show whether there is a lot of confusion, in terms of a relative ordering of matching scores, between a given set of multiple vessel appearances, see Figure 3.

The DNN architectures generally have to be trained on the scenario data in question. However, the intended scenario data may not always be available in sufficient quantities or annotations may be costly. DINOv2 has shown remarkable capabilities as a model when applied out of the box, i.e. as a model trained on a large public dataset, and therefore we will use a.o. a pretrained version of the models in our experiments. We will refer to these models as *pretrained ResNet50* and *pretrained DINOv2*. In a real-world maritime scenario, however, the data quality may be lower than when compared to public train datasets, for instance the vessels can have low pixel resolution. Therefore, we also finetune our models on the MARVEL[20] dataset, using a lower resolution as input size, comparable to the detection size in the WHD track dataset. For ResNet50 we train on 128×128px image crops, and for DINOv2 the inputs are 112×112px. We refer to these models as *finetuned ResNet50* and *finetuned DINOv2*. The MARVEL dataset consists of almost 400k images of 26 vessel types. It is a very diverse dataset comprising many viewpoints, vessel characteristics and environmental conditions. In this work, we use the verification set of MARVEL: pairs of recordings of ships with the same identity/IMO number under different viewpoints or backgrounds. In total, the dataset contains 4k unique vessel identities. Examples of the MARVEL dataset and performances of the DL-based models are shown in Figure 5.

Model	Rank-1	Rank-5	Rank-10	mAP	mINP
Pretrained ResNet50	6.63	15.31	22.96	1.63	1.06
Pretrained DINOv2 ViT-B	4.59	18.37	26.53	1.92	1.06
Finetuned ResNet50	90.82	96.94	97.96	62.71	7.88
Finetuned DINOv2 ViT-B	89.80	94.39	94.90	64.99	11.42

Figure 5. Left: examples of three different vessel identities and their recording variations in the MARVEL dataset. Right: performances (%) on the MARVEL ReID test dataset for pretrained and finetuned models.

4. RESULTS

In this section we describe the experiments performed on the WHD track dataset. To evaluate the effectiveness of appearance features for MOT in maritime scenarios, the experiments are done varying the number of considered tracks N, time window T between detections, and bounding box area A. We compute metrics for all feature extraction models and strategies previously described in Section 2.2. The obtained results provide insight in how a MOT framework could be tuned in order to optimize its performances in a real-world setting. Note that results for the ROC curves and Hungarian association metrics are computed on the WHD dataset, while the MARVEL dataset is used exclusively to finetune the DNN feature extraction models.

Pretrained vs. finetuned

The two DNN models can be used as a pretrained variant (Section 2) or preemptively finetuned on MARVEL (Section 3). This yields 4 different models which allow feature extraction for the WHD dataset, as described in Section 2.2. The results in Figure 6 indicate that a pretrained ResNet50 model gives relatively poor results, even for low values of T. The CNN lacks features that can distinguish vessels that have different identities. The classification capability of the model seems limited to a coarse-grained differentiation between vessel types, e.g. cargo vs. taxi boat, or rough appearance differences, e.g. color. The finetuned ResNet50 produces in general the best results, followed by pretrained DINOv2. Interesting to note here is that a generic foundation model can be successfully applied to an unseen scenario, producing results that are competitive to a model that is specifically tuned on the vessel ReID task. However, the finetuned DINOv2 has decreased performance, suggesting that tuning the foundation model to fit the MARVEL re-identification task does not work well on WHD, where small appearance changes are more frequent. Nevertheless, compared to the transformer-based model, the CNN definitely benefits from further training on vessel data.

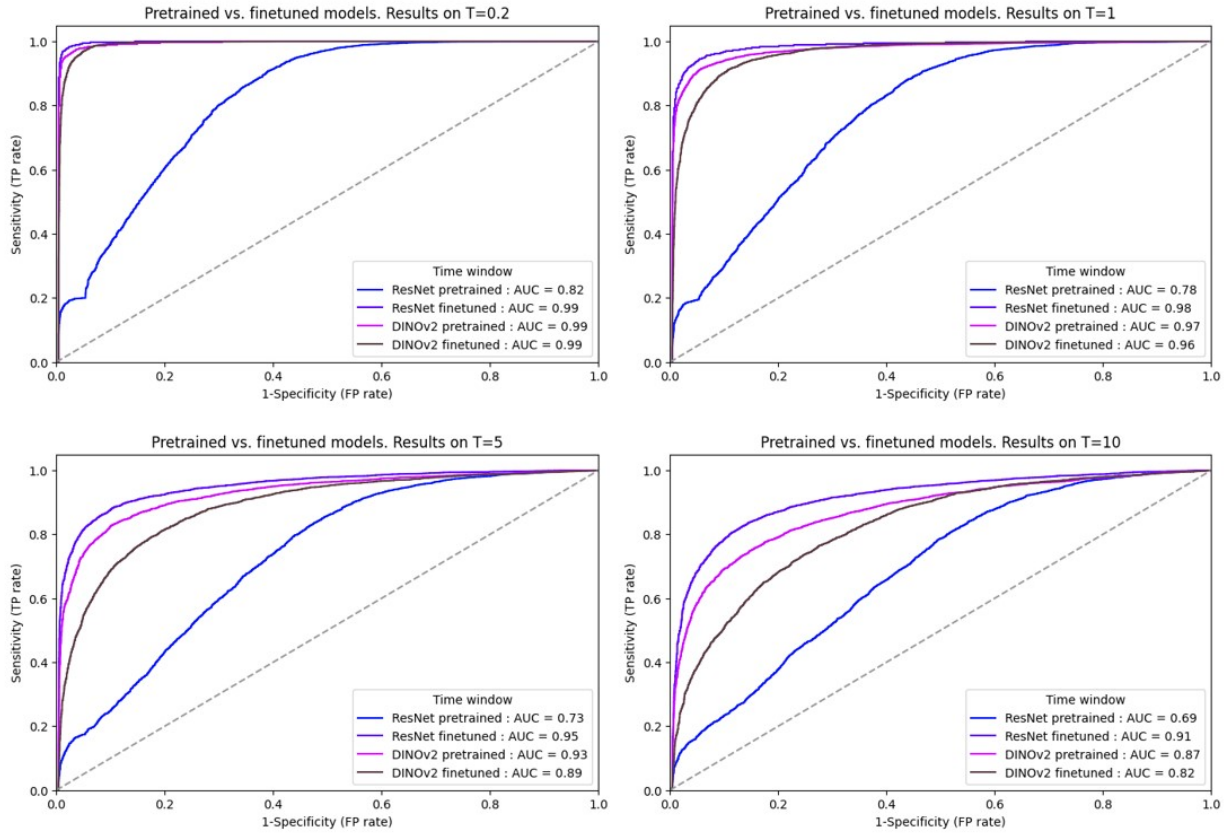

Figure 6. ROC curves for pretrained and finetuned ResNet50 and DINOv2 models. Each graph shows performances for a chosen time window T in seconds.

Smoothed vs. non-smoothed detections

The use of raw vs. smoothed detections on track (see Section 2.2) can affect performances of a tracking system at different levels, and in particular at the appearance-based association step. We show the quantitative difference in using non-smoothed and smoothed bounding boxes in Figure 7, for the best performing models - finetuned ResNet50 and pretrained DINOv2. For both models, the improvement is consistent on all tested time windows between detections. Using smoothed detections means that sudden and non-realistic changes in position and

size of the bounding boxes are averaged out. This brings consistently a 0.1 to 0.4 increase in AUC. The relative increase in performance is higher for ResNet50 than DINOv2 (average increase of 0.3 and 0.17 in AUC, respectively), suggesting that DINOv2 features are more robust to short spatial perturbations. Unless further specified, we considered smoothed detections in the following experiments.

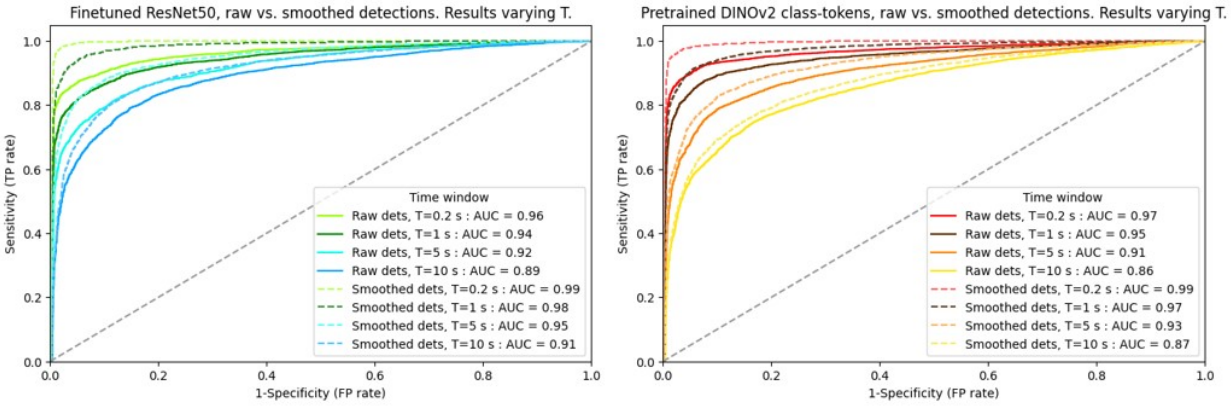

Figure 7. ROC curves for ResNet50 (left) and DINOv2 (right) feature matching, for non-smoothed and smoothed detections before feature extraction. Results are over $R = 10$ runs on all 480 tracks.

Class tokens vs. average patch tokens

We previously described an alternative feature extraction method using DINOv2's intermediate patch tokens, instead of the final class token. Applying this strategy led to improved results. As shown in Figure 8, for all T values the AUC score is higher, and now almost perfectly overlapping with the finetuned ResNet50 scores. ResNet50 still shows a slightly better performance when $T = 10$ seconds, suggesting higher robustness to bigger appearance shifts.

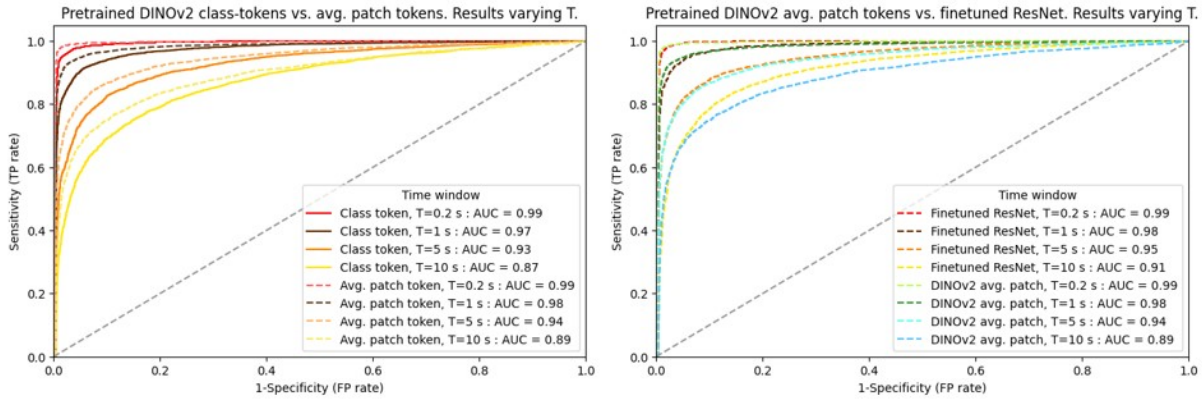

Figure 8. ROC curves for pretrained DINOv2 class tokens and avg. patch tokens (left). For reference, curves for DINOv2 avg. patch tokens are also shown together with the ROC curves for the finetuned ResNet50 (right).

Vessel resolution

In Figure 9 it is shown how the better performing finetuned ResNet50 and pretrained DINOv2 using avg. patch tokens, perform at different resolutions. Here the number of tracks M is a subset of the original 480 WHD tracks, which were used in full in the other ROC experiments (see Figure 2 for the subsets according to area A). For both

ResNet50 and DINOv2, a heavier drop in accuracy happens associating vessels with area $A < 2000$px. This is due to the fact that for lower A, less information is provided to the feature extractor. ROC curves for pretrained DINOv2 average patch tokens look more spread, highlighting a bigger difference according to the bounding box sizes considered. This may be caused by the fact that, for example, most smaller vessels can go down to less than 15×30px, which corresponds roughly to the information contained in only two DINOv2 input patches. The visual information here contained is therefore heavily reduced. DINOv2 performs better than ResNet50 on high resolutions, e.g. $A > 10000$px. ResNet50, instead, is more robust to resolution variations and retains better scores at lower resolution.

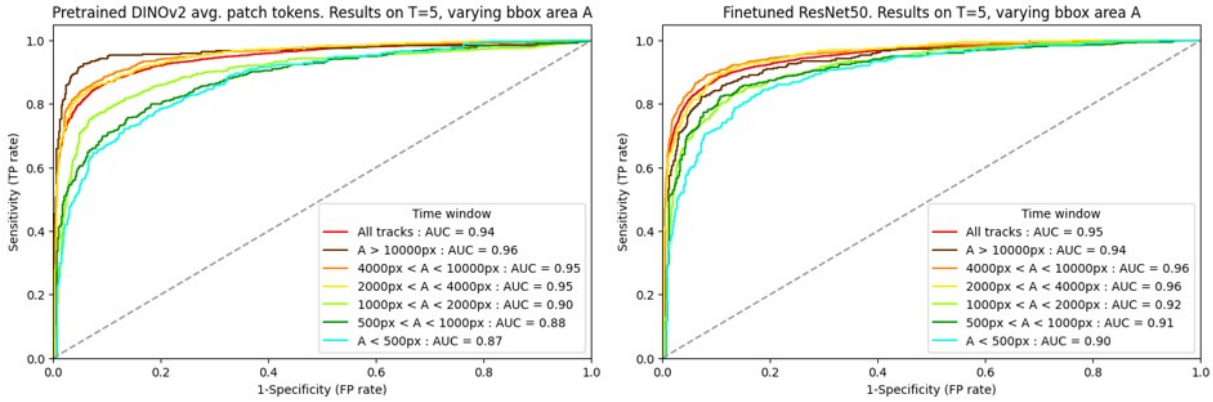

Figure 9. ROC curves for ResNet50 (left) and DINOv2 (right), by varying bounding box area A. Typically, less pixels on target (green/blue) yield worse results.

Matching assignment

The Hungarian association metric has been computed for a varying number of corresponding tracks N in the scene. Furthermore, we show results derived for multiple time window T values, for the best two models: finetuned ResNet50 and pretrained DINOv2. Figure 10 shows ROC-AUC results. For a low number of tracks N, the performance is hardly affected by an increasing T. For larger N the accuracy drops more quickly over time, and the number of individual association errors is higher. Results for $T > 5$ seconds indicate a higher chance of appearance changes, e.g. by change of orientation, pixels on target or occlusion. The DINOv2 result confirms that the average patch token strategy is superior over the standard class token as feature vector. The accuracy of the Hungarian association is comparable between the finetuned ResNet50 (left) and DINOv2 (right) when $T < 5$ seconds. With increasing T and larger appearance feature changes the CNN turns out to be more robust.

Non-DNN matching

For completeness, we have also compared our best DL-based models to a traditional feature extraction method, see Figure 11. The traditional method is based on color histograms (see Section 2.2), and has not been optimized with respect to the data. The method serves as a lower limit for what is achievable when reducing the computational complexity of the feature extraction. It turns out that the results are nearly equal to those achieved by DL-based models, when considering $N = 5$ tracks, and using consecutive timestamps (as state-of-the-art trackers currently do, processing all available detections). As soon as larger appearance changes occur, though, the benefits of using ResNet50 or DINOv2 as feature extractors become more evident. Nonetheless, it shows that traditional feature extraction methods, when properly optimized, can possibly beat DL-based models when it comes to computational complexity, explainability of results and ease of implementation.

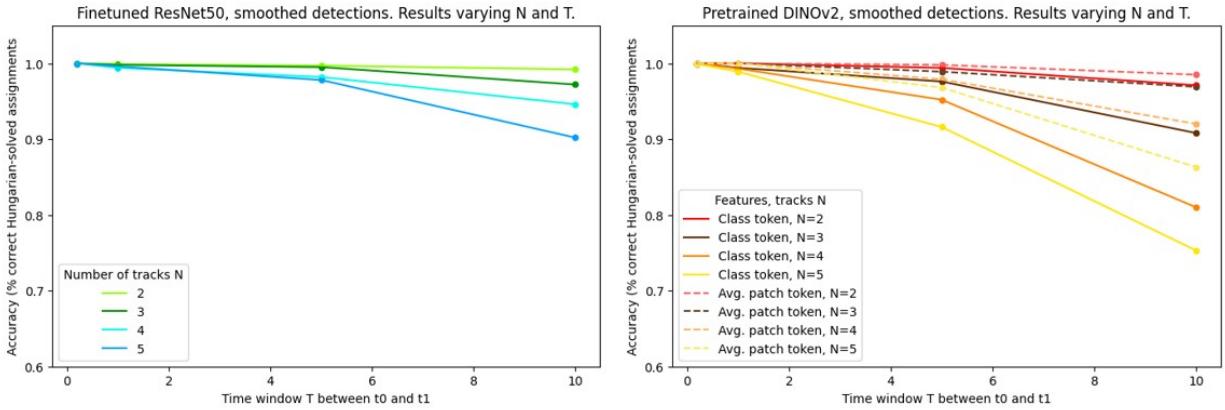

Figure 10. Accuracy of the highest performing DNN models on the Hungarian association step for varying N and T values. Left: finetuned ResNet50 model. Right: DINOv2 class token vs. average patch token.

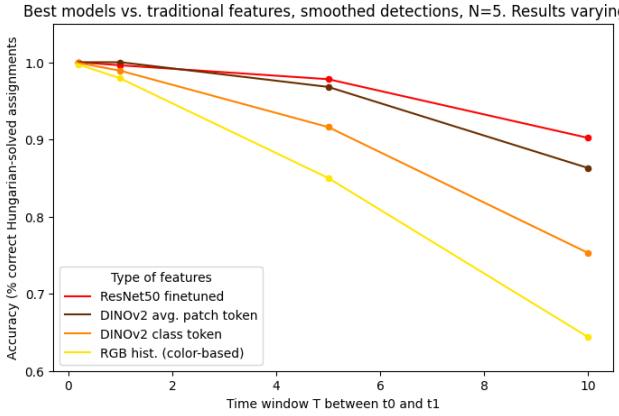

Figure 11. Accuracy of the highest performing DNN models (pretrained DINOv2 versions and finetuned ResNet50) and a traditional feature extraction method, on the Hungarian association step. Results are shown for $N = 5$ tracks, and varying T.

5. DISCUSSION & CONCLUSIONS

In this work we have analysed the impact of a number of methodological choices for appearance feature matching, typically present in a modern multiple object tracking (MOT) framework. We have experimented with two DNN models on a realistic maritime scenario using an in-house vessel tracking dataset.

Our findings indicated that best performance on matching vessel pairs was achieved by finetuning a ResNet50 model on an external ReID dataset. Interestingly, a non-tuned DINOv2 foundation model based on average patch tokens performed nearly as well as the ResNet50 model, except in case of low resolution vessels or larger appearance differences. This offers the potential of having the pretrained foundation model as a baseline in new scenarios, without the need for finetuning it on a representative dataset, which may not be available in every scenario. The out-of-the-box performance depended on the choice of token strategy in DINOv2, preferring average patch tokens over class tokens in the network. Tuning the DINOv2 model on an external ReID dataset turned out to decrease matching performance, and this was unexpected when compared to the ResNet50 model's finetuning results. One can argue that the external ReID dataset contains mostly large appearance changes, and that small changes as seen between short term detection correspondences are underrepresented. However, this does not explain in full the improved ResNet50 performance using finetuning, so we expect the training of the foundation model to just have more complexities.

When solving the assignment problem for multiple vessels at a time, based on the feature matching scores, the

finetuned ResNet50 model performed best, where the pretrained DINOv2 struggled more with large appearance changes. Interestingly, the performance of the pretrained foundation model is approximated by a traditional histogram-based feature matching method. Although DINOv2 has higher performance than the histogram-based method, the latter wasn't optimized in any way on the data; straightforward choices have been made for the histogram's extraction, representation and matching algorithms.

We expect that in many MOT situations, the assignment problem will be limited to linking few vessels within a short time window. At the same time, occlusions will happen during MOT and they make that larger appearance variations will occur, so one should be prepared for a worst-case matching situation. It will depend on the scenario whether this indeed requires more complex feature extraction, or it is acceptable to not assign the detections involved and allow more track breaks as a result.

In future work, we would like to extend the assignment evaluation with different numbers of vessels, simulating the (dis)appearance of vessels in the scene. This would not only require the feature matches to have a good relative ordering, but also that the absolute scores can be used. Although the ROC results illustrate the effect of a chosen threshold for matching, they do not provide the full picture of multiple vessel assignment using a threshold. Another line of work is the finetuning of DL-models on limited appearance changes, especially verifying whether a foundation model can benefit from this approach. Finally, an investigation of non-DL methods for feature extraction as opposed to DL-models could be useful, in particular when efficient and explainable methods are desirable.

ACKNOWLEDGMENTS

We gratefully acknowledge the Rotterdam Port Authority for their support in the data acquisition trial.

REFERENCES

[1] Qiao, D., Liu, G., Lv, T., Li, W., and Zhang, J., "Marine vision-based situational awareness using discriminative deep learning: A survey," *Journal of Marine Science and Engineering* **9**(4) (2021).

[2] Ballan, L., Melo, J. G. O., van den Broek, S. P., Baan, J., Heslinga, F. G., Huizinga, W., Dijk, J., and Dilo, A., "EO and radar fusion for fine-grained target classification with a strong few-shot learning baseline," in [*Signal Processing, Sensor/Information Fusion, and Target Recognition XXXIII*], **13057**, 130570K, International Society for Optics and Photonics, SPIE (2024).

[3] Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., and Kim, T.-K., "Multiple object tracking: A literature review," *Artificial Intelligence* **293**, 103448 (2021).

[4] Kim, I. S., Choi, H. S., Yi, K. M., Choi, J. Y., and Kong, S. G., "Intelligent visual surveillance — a survey," *International Journal of Control, Automation and Systems* **8**(5), 926–939 (2010).

[5] Chen, X., Wang, S., Shi, C., Wu, H., Zhao, J., and Fu, J., "Robust ship tracking via multi-view learning and sparse representation," *Journal of Navigation* **72**(1), 176–192 (2019).

[6] Scarrica, V. M., Panariello, C., Ferone, A., and Staiano, A., "A hybrid approach to real-time multi-object tracking," (2023).

[7] Xiao, C., Cao, Q., Zhong, Y., Lan, L., Zhang, X., Luo, Z., and Tao, D., "Motiontrack: Learning motion predictor for multiple object tracking," *Neural Networks* **179**, 106539 (2024).

[8] Wang, Y.-H., Hsieh, J.-W., Chen, P.-Y., Chang, M.-C., So, H.-H., and Li, X., "Smiletrack: Similarity learning for occlusion-aware multiple object tracking," *Proceedings of the AAAI Conference on Artificial Intelligence* **38**(6), 5740–5748 (2024).

[9] Luo, R., Song, Z., Ma, L., Wei, J., Yang, W., and Yang, M., "Diffusiontrack: Diffusion model for multi-object tracking," *Proceedings of the AAAI Conference on Artificial Intelligence* **38**(5), 3991–3999 (2024).

[10] Zhang, Y., Liang, Y., Leng, J., and Wang, Z., "Scgtracker: Spatio-temporal correlation and graph neural networks for multiple object tracking," *Pattern Recognition* **149**, 110249 (2024).

[11] Bouma, H., Baan, J., Landsmeer, S., Kruszynski, C., van Antwerpen, G., and Dijk, J., "Real-time tracking and fast retrieval of persons in multiple surveillance cameras of a shopping mall," in [*Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2013*], **8756**, 96–108, SPIE (2013).

[12] Wei, W., Yang, W., Zuo, E., Qian, Y., and Wang, L., "Person re-identification based on deep learning — an overview," *Journal of Visual Communication and Image Representation* **82**, 103418 (2022).

[13] 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., "An image is worth 16x16 words: Transformers for image recognition at scale," *CoRR* **abs/2010.11929** (2020).

[14] Jocher, G., "Ultralytics yolov5," (2020).

[15] Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L., "Microsoft COCO: common objects in context," *CoRR* **abs/1405.0312** (2014).

[16] HeK, M., RenS, Q., et al., "Deep residual learning for image recognition," in [*2016 IEEE Conference on Computer Vision and Pattern Recognition*], 770–778 (2016).

[17] Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al., "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193* **1** (2023).

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

[19] Kuhn, H., "The hungarian method for the assignment problem," *Naval Research Logistic Quarterly* **2** (05 2012).

[20] Gundogdu, E., Solmaz, B., Yücesoy, V., and Koç, A., "Marvel: A large-scale image dataset for maritime vessels," in [*Computer Vision – ACCV 2016. ACCV 2016. Lecture Notes in Computer Science(), vol 10115. Springer, Cham.*], (2017).