Feature-Space Oversampling for Addressing Class Imbalance in SAR Ship Classification

Ch Muhammad Awais
ISTI-CNR & University of Pisa
Pisa, Italy
chmuhammad.awais@phd.unipi.it

Marco Reggiannini
ISTI-CNR & NBFC
Pisa, Palermo, Italy
marco.reggiannini@isti.cnr.it

Davide Moroni
ISTI-CNR
Pisa, Italy
davide.moroni@isti.cnr.it

Oktay Karakuş
Cardiff University
Cardiff, UK.
karakuso@cardiff.ac.uk

***Abstract*—SAR ship classification faces the challenge of long-tailed datasets, which complicates the classification of underrepresented classes. Oversampling methods have proven effective in addressing class imbalance in optical data. In this paper, we evaluated the effect of oversampling in the feature space for SAR ship classification. We propose two novel algorithms inspired by the Major-to-minor (M2m) method M2m$_f$, M2m$_u$. The algorithms are tested on two public datasets, OpenSARShip (6 classes) and FuSARShip (9 classes), using three state-of-the-art models as feature extractors: ViT, VGG16, and ResNet50. Additionally, we also analyzed the impact of oversampling methods on different class sizes. The results demonstrated the effectiveness of our novel methods over the original M2m and baselines, with an average F1-score increase of 8.82% for FuSARShip and 4.44% for OpenSARShip.**

***Index Terms*—SAR Ship classification, Over Sampling, Imbalanced Datasets, Feature Space Exploitation.**

I. Introduction

Synthetic aperture radar (SAR), in conjunction with the Automatic Identification System (AIS), plays a crucial role in maritime traffic monitoring applications, including deep learning-based SAR ship classification [1]. However, SAR ship classification datasets often face two significant challenges: insufficient data and high class imbalance [2]. Deep learning models require large datasets for effective training, and when data is limited, transfer learning techniques are commonly employed [3]. In traditional transfer learning, the feature extraction layers of a pretrained model are frozen, and a classifier is trained on the extracted features. To address data imbalance, data augmentation techniques are typically used to enhance the dataset [4]. However, these methods frequently fail for SAR data due to the presence of significant noise and the inherently low resolution of SAR images.

The feature space provides an alternative approach to address these challenges by generating synthetic features using techniques like oversampling in feature space [5]. Oversampling leverages features extracted from a pretrained network to create synthetic features, thereby mitigating class imbalance. Previous works have explored feature-space methods for SAR ship classification. For example, Li et al.[6] compared oversampling techniques for three-class classification on OpenSARShip and FuSARShip, Zhang et al. [7] applied oversampling on six classes of OpenSARShip using accuracy as a performance metric, and Xie et al. [8] proposed a novel loss function evaluated on six classes of FuSARShip.

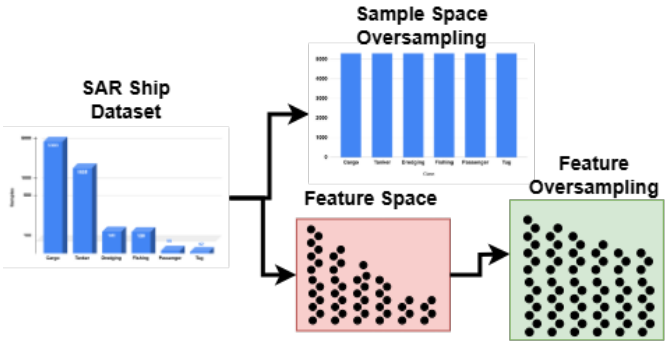


Fig. 1: Sampling Overview

The oversampling technique Major-to-minor (M2m) [9] is particularly effective in addressing data imbalance in sample space. M2m generates synthetic samples for minority classes by utilizing samples from majority classes, thereby increasing the representation of underrepresented classes without using the samples from minority classes. M2m has not been explored for SAR ship classification, in this study, we compare the original M2m with two modified versions of M2m that perform oversampling in feature space (Figure 1). We adopted the F1-score [10], a metric well-suited for imbalanced data, to evaluate the effectiveness of the proposed framework comprehensively. Our main contributions are as follows:

- We propose two effective oversampling frameworks inspired by the Major-to-minor (M2m) method, specifically designed to address the challenges of class imbalance in SAR ship classification.
- We improve the classification performance across datasets, highlighting the generalizability of our approach.

The rest of the paper is arranged as follows: Section II

This work was supported by National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 1.4 - Call for tender No. 3138 of 16 December 2021, rectified by Decree n.3175 of 18 December 2021 of Italian Ministry of University and Research funded by the European Union – NextGenerationEU. Award Number: Project code CN_00000033, Concession Decree No. 1034 of 17 June 2022 adopted by the Italian Ministry of University and Research, CUP D33C22000960007, Project title "National Biodiversity Future Center - NBFC".
This work was conducted during an exchange period at Cardiff University.

outlines our proposed methodology. Section III presents the results of our experiments, while Section IV discusses these findings. Finally, Section V concludes the paper.

II. Methodology

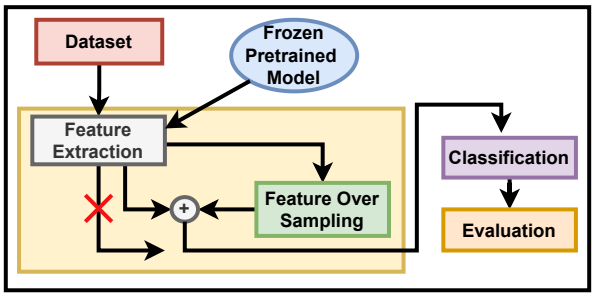


Fig. 2: (1) Dataset: SAR data; (2) Feature extraction: pre-trained model with frozen layers; (3) Feature oversampling: method to generate synthetic features; (4) Classification: assigns labels to the features; (5) Evaluation: compares the predicted labels with the original labels to assess performance.

The proposed approach consists of two oversampling methods in feature space to mitigate the imbalance issue (see Figure 2). Traditionally, the training pipeline can be divided into three parts: (1) *feature extractor*: usually a pre-trained network which takes an input and provides features. Feature extractor is often frozen, meaning that the weights are not updated while training; (2) *classifier*: a model that takes the features extracted by the feature extractor and predicts the class of the extracted features; (3) *loss function*: takes the predicted output of the classifier and compares it with the actual label, then updates the weights of the classifier (since the extractor is frozen). In our methodology, the features extracted by the feature extractor are exploited to help the classifier learn better; in Fig. 2, it is presented as the "Feature Over Sampling" block.

The two proposed oversampling methods, M2m$_f$ and M2m$_u$, are compared with the baseline and the original M2m (M2m$_{orig}$). M2m$_f$ (described in Algorithm 1), is based on the implementation provided in the original M2m paper but operates in the feature space instead of the sample space. It takes extracted features and increases the number of features representative of the minority classes by using features from the majority classes.

M2m$_f$ first identifies the minority classes with samples fewer than a defined minority threshold M_v, which is set by analyzing the samples of all classes. Second, centroids (ctr_m) are calculated from the features of minority classes (average feature values of each class). These centroids are used to generate synthetic features by introducing minor changes to the majority class features. A synthetic feature is retained only if it differs by a defined minimum distance ($d_t = 0.5$) from existing synthetic features (to keep samples unique). Lastly, only (M_v) synthetic features are retained, and these features

Algorithm 1: M2m$_f$

Data: Set of extracted Features D, Minority Value M_v, Distance Threshold d_t, lambda $\lambda = 0.1$.
c_m = class indexes c such that $len(D[c]) < M_v$
$ctr_m = \{$class centroids $ctr[c]$, $c \in c_m\}$
$D_M = \{D \setminus D_m\}$ where $D_m = \bigcup_{c \in c_m} D[c]$
Initialize synthetic features S: $S[c] = \emptyset$, $c \in c_m$
for $c \in c_m$ **do**
  for $\mathbf{f} \in D_M$ **do**
    $\mathbf{s} = \mathbf{f} + \lambda(\text{ctr}[c] - \mathbf{f})$
    d_m = min dist of $\mathbf{s}$ from any feature in $S[c]$
    if $d_m > d_t$ **then**
      Add $\mathbf{s}$ to $S[c]$
    if $len(S[c]) > M_v$ **then**
      break inner loop: Go to next minority class
Output: S

are appended to the original dataset (D). The updated features are then used to train the models.

The second variation of the feature-based algorithm, M2m$_u$, follows a similar pattern to its predecessor but introduces a different eligibility criterion for retaining synthetic samples. Instead of evaluating the distance from existing synthetic features, this variation retains a new synthetic sample only if it is deemed "*similar*" to features in the original dataset. This adjustment ensures that synthetic features remain closely aligned with existing ones. To achieve this, *Cosine similarity* is calculated between the new synthetic feature, $\mathbf{s}$, and the features of the same class $D[c]$ (as in Equation (1)). Synthetic features are retained only if their similarity exceeds a predefined threshold ($sim_t = 0.8$).

$$sim = cosine(s, D[c]) \tag{1}$$

The baseline comprised models that combined frozen feature extraction algorithms with a non-frozen classifier in a pipeline. The M2m$_{orig}$ implementation is taken from the code provided by the original paper [9], by simply increasing the number of samples based on the updated ratios using "*WeightedRandomSampler*" from "*pytorch*".

TABLE I: Datasets and Minority Values M_v. (**bold** represents the minority classes c_m which were oversampled with the M_v features)

Dataset	M_v	Classes
FuSARShip	800	**Fishing**, Cargo
	500	Fishing, Cargo, **Bulk**, **Tanker**
	500	Fishing, Cargo, **Bulk**, **Tanker**, **Tug**, **Container**, **Passenger**, **GeneralCargo**, Dredging
OpenSARShip	3000	**Tanker**, Cargo
	250	Tanker, Cargo, **Fishing**, **Bulk**
	250	Tanker, Cargo, **Fishing**, **Bulk**, **Tug**, **Passenger**

The dataset and their corresponding minority values (M_v) are presented in Table I. The M_v values were determined

by analyzing the sample distribution within the dataset. For instance, in the FuSARShip 2-class dataset, the "Cargo" class contained approximately 1,700 samples, while the "Fishing" class had only around 800. To address the aforementioned imbalance, an additional 800 samples were generated for the "Fishing" class. Similarly, in the 4-class dataset, other classes contained fewer than 500 samples, so 500 additional samples were generated for each minority class, and so forth.

Three pretrained networks (VGG16 [11], ResNet50 [12], and ViT-16_base224 [13]) are utilized for feature extraction. For VGG and ResNet, the images were resized down to 64×64, whereas for ViT, the image size was set to 224×224. The features from feature extractors undergo average pooling and flattening before being processed by a fully connected block. This block consists of a linear layer, followed by a ReLU activation function, a dropout layer for regularization, and a final linear layer to generate the class predictions.

The models were trained for 30 epochs using a batch size of 32, with a learning rate of 0.0001. The training process utilized the *Adam* optimizer and *CrossEntropy* as the loss function. Python libraries such as *PyTorch* and *timm* were employed for implementing and training the models. The code is available at https://github.com/cm-awais/SAR_sampling.

III. Results

The results section is structured to provide a comprehensive understanding of the findings. It is divided into two main parts: the first focuses on the impact of oversampling across different datasets, analyzing how this technique influences performance metrics. The second explores the effects of oversampling on class imbalances, highlighting its role in adjusting class sizes and improving model performance in scenarios with uneven data distribution.

A. Oversampling and datasets

The results for FuSARShip (Table II) suggested that the M2m_u enhanced F1-scores, achieving an 8.82% increase compared to the baseline and even a 0.63% improvement over the proposed M2m_f. While models with ViT and VGG as feature extractors demonstrated improved performance, the model using ResNet as the feature extractor predominantly exhibited signs of underfitting. Notably, the M2m_u method consistently outperformed all other approaches on average, whereas the original M2m_{orig} method yielded the lowest performance among the methods evaluated.

In contrast, the results for OpenSARShip (Table III) provided a different trend compared to FuSARShip results. Particularly, M2m_f mostly performed better, achieving an average improvement of F1 of 5.4% over baseline and 0.98% compared to the proposed M2m_u. Similar to FuSARShip, the models using ViT and VGG as feature extractors improved classification performance, whereas models with ResNet as the feature extractor again showed signs of under-fitting. On average, M2m_f achieved the best performance, while M2m_{orig} emerged as the poorest-performing method.

TABLE II: Performance analysis for FUSARShip.

CLS	Models	Baseline	M2m_{orig}	M2m_f	M2m_u
2	ViT	55.92	68.08	69.37	**69.95**
	VGG	57.23	68.7	69.37	**70.81**
	ResNet	55.4	**60.23**	55.4	55.4
4	ViT	42.61	53.23	**67.88**	61.79
	VGG	60.91	62.72	63.3	**69.54**
	ResNet	48.39	12.3	42.61	42.61
9	ViT	36.95	0.92	58.52	**59.92**
	VGG	53.56	45.05	60.26	**62.38**
	ResNet	38.27	5.69	36.3	36.3
AVG		49.92	41.88	58.11	**58.74**

TABLE III: Performance analysis for OpenSARShip.

CLS	Models	Baseline	M2m_{orig}	M2m_f	M2m_u
2	ViT	63.51	63.51	**74.1**	74.06
	VGG	67.04	**73.16**	73.07	69.57
	ResNet	63.51	61.56	63.51	63.51
4	ViT	59.64	15.23	**69.36**	68.46
	VGG	63.23	64.44	65.78	**68.13**
	ResNet	59.63	58.5	59.64	59.64
9	ViT	58.13	40.84	**70.39**	68.38
	VGG	59.94	60.23	**67.55**	62.88
	ResNet	57.96	38.22	57.96	57.96
AVG		61.4	52.85	**66.82**	65.84

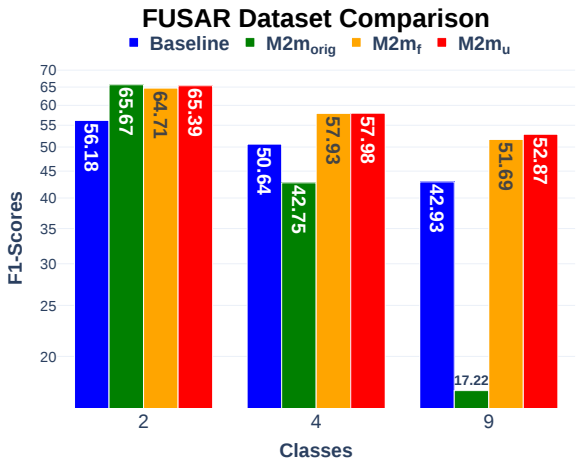
B. Oversampling and Number of Classes

Another aspect of our methodology examined the impact of oversampling on different class configurations. For FuSARShip (Figure 3a), M2m_u demonstrated superior performance across 4- and 9-class datasets, while M2m_{orig} excelled in 2-class classification. Similarly, for OpenSARShip (Figure 3b), M2m_f outperformed other methods in 2- and 6-class configurations, whereas M2m_u achieved the best results for 4-class classification. Notably, M2m_{orig} consistently underperformed compared to baselines when applied to datasets with 4 or more classes in both cases.

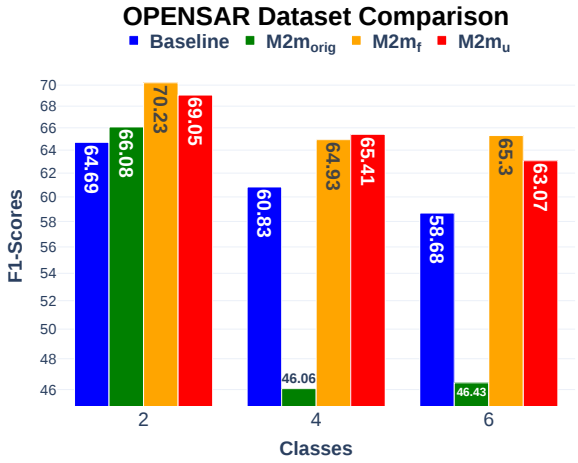
IV. Discussion

A. Dataset Performance Analysis

The results for FuSARShip demonstrated the superiority of M2m_u, with feature-level methods outperforming both the baseline and the M2m_{orig}. This highlights the importance of feature-based oversampling for the FuSARShip dataset. In contrast, the OpenSARShip dataset presented a different trend, with M2m_f outperforming other training strategies. We believe this discrepancy in performance is due to differences in data quality between the two datasets. FuSARShip, being a high-resolution dataset compared to OpenSARShip, exhibited notable differences in the number of synthetic samples added during augmentation. Specifically, 500 synthetic samples were generated for 4- and 9-class configurations in FuSARShip, whereas only 250 synthetic samples were added for 4- and 6-class configurations in OpenSARShip. These disparities in resolution and augmentation scale may have constrained the performance improvements observed in OpenSARShip.



(a) Class-based performance for FuSARShip



(b) Class-based performance for OpenSARShip

Fig. 3: Class-based F1-score performance comparison for FuSARShip and OpenSARShip datasets, with scores presented on a logarithmic scale for improved visualization.

Despite these differences, both methodologies successfully improved performance, achieving an average F1-score increase of 8% for FuSARShip and 5% for OpenSARShip, on the other hand, M2m$_{orig}$ at the sample level decreased performance, highlighting the effectiveness of the feature-level oversampling methods.

B. Model Performance Analysis

No single model emerged as conclusively superior across all scenarios. VGG showed promising results in FuSARShip with an average F1-score of 65.94%, while ViT achieved 64.57%. Conversely, in OpenSARShip, ViT outperformed VGG with a 3% higher F1-score. It is noteworthy that ViT performed worse than VGG in baseline models but achieved competitive results with oversampling interventions, indicating the reliability of ViT when paired with feature-level oversampling methods.

In contrast, ResNet underperformed in all scenarios. We hypothesize that its deeper architecture may have led to underfitting or challenges in handling the limited dataset, while the shallower architectures of ViT and VGG demonstrated greater effectiveness.

C. Class-Based Performance

We evaluated three different versions of both datasets to analyze the general effects of feature oversampling. For the 4-class versions of both datasets, M2m$_u$ outperformed all other methods. For the 9-class version of FuSARShip, M2m$_u$ also yielded the best results, whereas for the 2-class version of FuSARShip, M2m$_{orig}$ gained the best F1-score. However, for classes 2 and 6 in OpenSARShip, M2m$_f$ significantly outperformed M2m$_u$, achieving up to 2% better F1-scores. Except for 2 classes of both datasets, M2m$_{orig}$ performed poorer even compared to baselines, indicating the inability of sample-based oversamples for SAR Ship classification datasets.

D. Cost-Effectiveness and Future Recommendations

Feature oversampling techniques are cost-effective because they eliminate the need to reiterate the feature extraction process across the entire architecture. Once synthetic features are generated, only the classifier requires training. This approach reduces both training time and computational resource usage.

For future work, it is important to note that the parameter values used in the proposed algorithms (M2m$_u$, M2m$_f$) were derived from the original paper [9] and the authors' experience. These values should be tailored to the specific research problem being addressed. This study underscores the effectiveness of feature-based oversampling in SAR ship classification from three perspectives:

- Improved performance on imbalanced datasets.
- Cost-effective training and resource utilization.
- Simplicity and adaptability to diverse imbalanced datasets.

The results not only highlight the effectiveness of feature space oversampling for SAR ship classification datasets, but also leave room for future research on sample space oversampling techniques dedicated to SAR data, or the development of more robust feature space oversampling techniques.

V. Conclusion

In this study, we proposed two feature-level oversampling algorithms that significantly enhanced SAR ship classification. The robustness of our proposed strategies proved to be applicable to SAR datasets, and have the potential to be extended to other fields of research. The results also indicated the lack of performance for sample space oversampling methods for SAR datasets. The proposed method demonstrated a notable increase in classification performance across different imbalanced classes and datasets, highlighting the importance of feature-level approaches in SAR ship classification.

In the future, our goal is to develop SAR-dedicated sample space oversampling methods, and refine feature space oversampling further and extend its application to additional classes and datasets, paving the way for more robust solutions in SAR ship classification and beyond.

REFERENCES

[1] Z. Yan, X. Song, L. Yang, and Y. Wang, "Ship classification in synthetic aperture radar images based on multiple classifiers ensemble learning and automatic identification system data transfer learning," *Remote Sensing*, vol. 14, no. 21, 2022. [Online]. Available: https://www.mdpi.com/2072-4292/14/21/5288

[2] Y. Zhang, Z. Lei, H. Yu, and L. Zhuang, "Imbalanced high-resolution sar ship recognition method based on a lightweight cnn," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[3] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, and M. A. Azim, "Transfer learning: a friendly introduction," *Journal of Big Data*, vol. 9, no. 1, p. 102, 2022.

[4] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[5] A. Gosain and S. Sardana, "Handling class imbalance problem using oversampling techniques: A review," in *2017 international conference on advances in computing, communications and informatics (ICACCI)*. IEEE, 2017, pp. 79–85.

[6] Y. Li, X. Lai, M. Wang, and X. Zhang, "C-saso: A clustering-based size-adaptive safer oversampling technique for imbalanced sar ship classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2022.

[7] Y. Zhang, Z. Lei, L. Zhuang, and H. Yu, "A cnn based method to solve class imbalance problem in sar image ship target recognition," in *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 5, 2021, pp. 229–233.

[8] N. Xie, M. Xiong, F. Wei, T. Zhang, Z. Yang, and W. Yu, "Ca-loss: A cosine affinity loss for imbalanced sar ship classification," in *IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium*, 2024, pp. 9070–9074.

[9] J. Kim, J. Jeong, and J. Shin, "M2m: Imbalanced classification via major-to-minor translation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 13 896–13 905.

[10] C. Awais and M. Reggiannini, "Deep learning for sar ship classification: Focus on unbalanced datasets and inter-dataset generalization," in *2024 International Conference on Electromagnetics in Advanced Applications (ICEAA)*, 2024, pp. 1–1.

[11] K. Simonyan, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[13] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.