The Journal of Open Source Software

UWMM-Baseline: A Python Package for Supervised Classification of Underwater Military Munitions Using Multi-Modal Data

Author 1[1¶] **and Author 2**[2]

1 University of Girona, Spain **2** University of Miami, USA ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

The presence of underwater military munitions (UWMM) in coastal and marine environments poses significant environmental and safety risks. Accurate detection and classification of UWMM are critical for remediation efforts, requiring robust methods to process multi-modal data, such as optical imagery and 3D reconstructions. UWMM detection leverages computer vision, machine learning, and 3D modeling to identify munitions against complex seabed backgrounds. Existing approaches, such as those using 2D image features, often suffer from high false positive rates, necessitating the integration of geometric (3D) data to improve accuracy.

`UWMM-Baseline` is an open-source Python package designed for the supervised classification of UWMMs using multi-modal data, including optical imagery (2D) and digital elevation models (DEMs) derived from 3D reconstructions of the scene (3D). Building on the methodology of Gleason et al. (2015) (Gleason et al., 2015), the package implements a modular pipeline for dataset creation, feature extraction, classification, and evaluation. Key features include:

- **Dataset Creation:** Automated tile extraction and labeling from underwater imagery and DEMs.
- **Feature Extraction:** Extraction of 2D (color, texture) and 3D (elevation, curvature, rugosity) features, which can be combined into a 2.5D feature set.
- **Classification:** Support for binary (UWMM vs. background) and multi-class (UWMM type) classification using Support Vector Machines (SVM).
- **Inference and Evaluation:** A complete inference pipeline to generate prediction masks on new data and quantitative evaluation using mean Intersection over Union (mIoU).
- **Modularity:** A flexible, configuration-driven framework allowing integration of new features, models, and data modalities.

`UWMM-Baseline` is designed for researchers, oceanographers, and environmental engineers working on UWMM detection, as well as educators teaching computer vision or marine science. It integrates with popular scientific Python libraries and supports workflows for both research and practical applications, such as seabed surveys and environmental monitoring.

Statement of Need

UWMM detection is a pressing challenge in marine environments, where legacy munitions from military activities contaminate coastlines and pose risks to ecosystems and human safety. Traditional detection methods, such as acoustic sonar or traditional 2D optical imagery, are often limited by resolution or seabed complexity. Optical imagery, combined with SfM-derived digital elevation models (DEMs), offers high-resolution data for precise UWMM identification, as demonstrated by Gleason et al. (2015) (Gleason et al., 2015). However, existing software

The Journal of Open Source Software

tools for UWMM detection are either proprietary, domain-specific, or lack the flexibility to handle multi-modal data in a unified framework.

`UWMM-Baseline` addresses this gap by providing a free, open-source, and modular Python package that implements established UWMM classification techniques using modern libraries. Unlike general-purpose computer vision libraries, `UWMM-Baseline` is tailored for underwater environments, incorporating domain-specific preprocessing and feature extraction (e.g., rugosity, curvature from DEMs).

The package's modularity enables researchers to experiment with new features, classifiers, or data sources (e.g., sonar, stereo vision), while its accessibility supports educational use in courses on machine learning, oceanography, or environmental science. By providing a full pipeline from data ingestion to evaluation, `UWMM-Baseline` lowers barriers to entry for UWMM detection research, fostering innovation in environmental monitoring and remediation.

Background

Figure 1: Inference output of the UWMM baseline model. The leftmost image represents the output of the model trained exclusively on 2D features, the central image shows the output of the model trained exclusively on 3D features, and the rightmost image illustrates the output of the model trained on 2.5D extracted features.

The detection of UWMM in underwater environments involves processing optical imagery to identify munitions against varied seabed backgrounds (e.g., coral reefs, seagrass). The core challenge lies in distinguishing human-made objects from similarly shaped or textured natural features. Gleason et al. (2015) demonstrated that while 2D image features (color, texture) achieve moderate accuracy (>80% for binary classification), they suffer from high false positives due to background complexity (Gleason et al., 2015). Incorporating 3D features (e.g., elevation, curvature) from SfM-derived DEMs significantly improves accuracy (89-95%) and reduces false positives, as these features capture the distinct geometric properties of munitions. This combined 2D and 3D approach is often referred to as 2.5D.

The methodology is broken down into a multi-stage pipeline. First, large survey images are processed into smaller, manageable tiles or tiles. This approach allows the model to learn local, high-resolution features and creates a large, diverse dataset suitable for training supervised machine learning models. The general workflow is as follows:

1. **Dataset Creation:** Optical images, depth maps, and ground-truth masks are used to generate a labeled dataset of image tiles.
2. **Feature Extraction:** A comprehensive set of 2D and 3D features is extracted from each tile.
3. **Model Training:** A classifier, such as an SVM, is trained on the extracted features.
4. **Inference:** The trained model is used to predict the locations of UWMM in new, unseen images.

5. **Evaluation:** The model's predictions are compared against ground-truth data to assess performance quantitatively.

This project implements this entire workflow in a configurable and automated pipeline, building on the foundational work of Gleason et al. to provide an accessible software tool.

Methodology

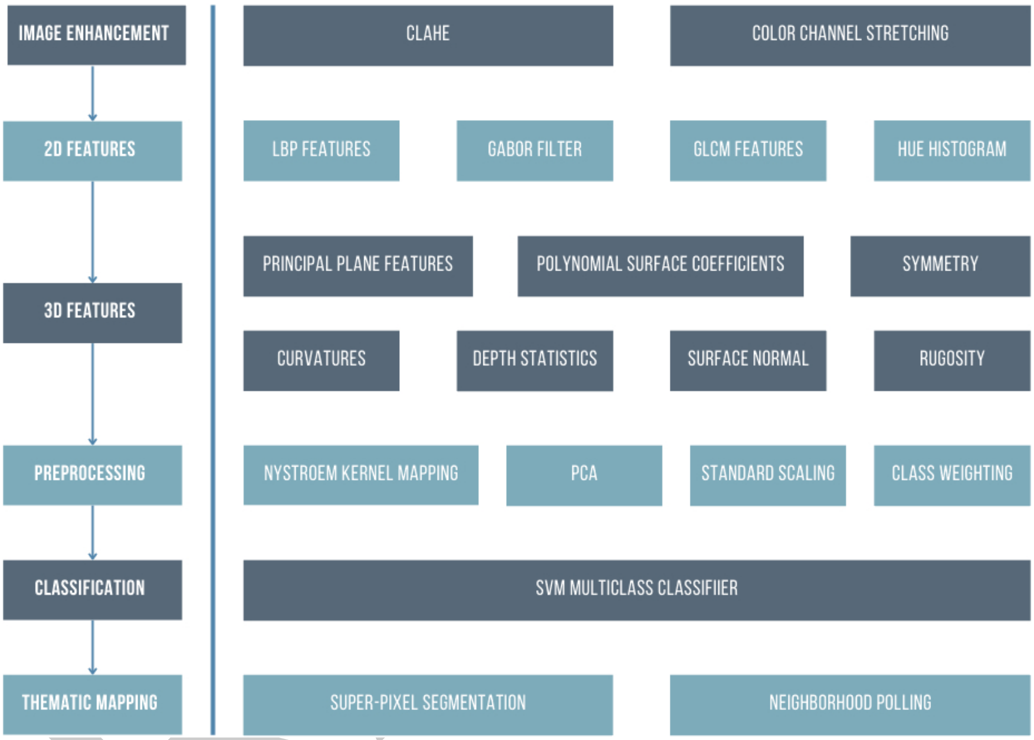

Figure 2: The implemented classification framework is depicted in the above flowchart. The left section outlines the primary stages of the pipeline, while the right section details the specific components and features utilized in each stage.

The `UWMM-Baseline` pipeline processes underwater images, depth maps, and masks to generate a labeled dataset of uniformly sized tiles for training and testing. It identifies potential underwater munitions (UWMM) and background locations using masks, extracting corresponding square tiles from image and depth data. To address class imbalance, a fixed number of background tiles are sampled per image, and only a subset of available UWMM pixels are used as tile centers. Data augmentation is applied by rotating UWMM tiles at multiple angles.

From the processed tiles, 2D and 3D features are extracted, extending the feature set proposed by Gleason et al. (2015) (Gleason et al., 2015). The 2D features, derived from optical images, include:

- **2D Features (from optical images):**
  - **Color Histograms:** HSV color distributions to capture seabed and UWMM appearance.
  - **Local Binary Patterns (LBP):** Texture descriptors robust to illumination changes (Ojala et al., 1994).
  - **Gray Level Co-occurrence Matrix (GLCM):** Texture metrics (e.g., contrast, dissimilarity) for seabed characterization (Haralick et al., 1973).

The Journal of Open Source Software

    - **Gabor Filters:** Edge and texture detection across multiple scales and orientations (Granlund, 1978).
- **3D Features (from depth maps):**
    - **Principal Plane Features:** Statistics of depth values, polynomial coefficients from a fitted surface, and rugosity (ratio of surface area to planar area).
    - **Curvatures and Normals:** Mean and Gaussian curvatures, shape index, curvedness, and surface normal vector statistics.
    - **Symmetry Features:** Gabor filters applied to depth maps to capture structural symmetry.

The classifier is implemented using a Support Vector Machine (SVM) with a radial basis function (RBF) kernel, approximated via the Nystroem method Williams & Seeger (2000). The model supports training on 2D, 3D, or combined 2.5D feature sets. Prior to inference, superpixel segmentation identifies homogeneous regions in the input image (Achanta et al., 2012). These regions are subdivided into overlapping tiles, each processed by the trained model to predict class labels. A neighborhood poll determines UWMM presence by evaluating whether the number of positive tile predictions within a region exceeds a predefined threshold. The final output is a segmentation mask indicating detected UWMM locations (see Figure 1).

This implementation leverages standard Python libraries, including NumPy (Harris et al., 2020), OpenCV (Bradski, 2000), scikit-learn (Pedregosa et al., 2011), scikit-image (Walt et al., 2014), and SciPy (Virtanen et al., 2020) for feature extraction, model training, and data processing.

Model Training and 3D Feature Extraction

The classification pipeline in `UWMM-Baseline` relies on feature extraction and Support Vector Machine (SVM) classification. For a tile $\mathbf{x} \in \mathbb{R}^{m \times n}$ (image) and corresponding DEM $\mathbf{d} \in \mathbb{R}^{m \times n}$, the feature vector $\mathbf{f}_{2.5D}$ combines 2D and 3D features:

$$\mathbf{f}_{2.5D} = [\mathbf{f}_{2D}, \mathbf{f}_{3D}]$$

The 3D features, $\mathbf{f}_{3D}$, are extracted from the elevation map of each tile. This process involves several calculations to describe the geometry of the surface.

A 2D polynomial surface of the third degree is fitted to the elevation map of a tile to model its shape (Shihavuddin et al., 2014). The equation for this surface is:

$$f(x, y) = p_1 + p_2 x + p_3 y + p_4 x^2 + p_5 xy + p_6 y^2 + p_7 x^2 y + p_8 x y^2 + p_9 y^3$$

The nine coefficients of this polynomial $(p_1, \ldots, p_9)$ are extracted through least-squares fitting and used as features (Shihavuddin et al., 2014).

Additional statistical features are calculated from the elevation values (z_i) within a tile, including the standard deviation, skewness, and kurtosis (Shihavuddin et al., 2014). With z_m as the mean elevation, S as the standard deviation, and N as the number of data points, the skewness and kurtosis are calculated as:

$$\text{Skewness} = \frac{\sum (z_i - z_m)^3}{(N-1)S^3}$$

$$\text{Kurtosis} = \frac{\sum_{i=1}^{N} (z_i - z_m)^4}{(N-1)S^4}$$

With regards to surface curvature, the Gaussian (G) and mean (M) surface curvatures are first calculated from the partial derivatives of the elevation map where the two principal curvatures, k_1 and k_2, can then be derived (Shihavuddin et al., 2014):

$$k_1 = M + \sqrt{M^2 - G}$$

$$k_2 = M - \sqrt{M^2 - G}$$

These principal curvatures are then used to calculate a shape index (S) and a curvedness index (C):

$$S = \frac{2}{\pi} \arctan\left(\frac{k_2 + k_1}{k_2 - k_1}\right), \quad C = \sqrt{\frac{k_1^2 + k_2^2}{2}}$$

Finally, rugosity (r) is computed to characterize the roughness of the seafloor habitat (Shihavuddin et al., 2014). It is calculated by dividing the contoured surface area of the tile (A_s) by the area of its orthogonal projection onto the principal plane (A_p):

$$r = \frac{A_s}{A_p}$$

Once the full feature vector is assembled, the SVM classifier solves the following optimization problem:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i$$

subject to:

$$y_i(\mathbf{w}^T\phi(\mathbf{f}_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, N$$

where $\mathbf{w}$ is the weight vector, b is the bias, ξ_i are slack variables, C is the regularization parameter, ϕ is the kernel mapping (RBF), and y_i are the class labels (Cortes & Vapnik, 1995).

Acknowledgements

We thank [Advisor/Collaborator Names] for guidance on underwater imaging and SfM techniques, and [Institution/Organization] for providing access to underwater datasets. This work was supported by [Funding Agency, Grant Number].

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(11), 2274–2282. https://doi.org/10.1109/TPAMI.2012.120

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Gleason, A. C. R., Shihavuddin, A., Gracias, N., Schultz, G., & Gintert, B. E. (2015). Improved supervised classification of underwater military munitions using height features derived from optical imagery. *OCEANS 2015 - MTS/IEEE Washington*, 1–5. https://doi.org/10.23919/OCEANS.2015.7404580

Granlund, G. H. (1978). In search of a general picture processing operator. *Computer Graphics and Image Processing*, *8*(2), 155–173. https://doi.org/10.1016/0146-664X(78)90047-3

Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-3*(6), 610–621. https://doi.org/10.1109/TSMC.1973.4309314

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Ojala, T., Pietikainen, M., & Harwood, D. (1994). Performance evaluation of texture measures with classification based on kullback discrimination of distributions. *Proceedings of 12th International Conference on Pattern Recognition*, *1*, 582–585 vol.1. https://doi.org/10.1109/ICPR.1994.576366

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Shihavuddin, A. S. M., Gracias, N., Garcia, R., Campos, R., Gleason, A. C. R., & Gintert, B. (2014). Automated detection of underwater military munitions using fusion of 2D and 2.5D features from optical imagery. *Marine Technology Society Journal*, *48*(4), 61–71. https://doi.org/doi:10.4031/MTSJ.48.4.7

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Walt, S. van der, Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., & contributors, the scikit-image. (2014). Scikit-image: Image processing in Python. *PeerJ*, *2*, e453. https://doi.org/10.7717/peerj.453

Williams, C. K. I., & Seeger, M. (2000). Using the nyström method to speed up kernel machines. *Proceedings of the 14th International Conference on Neural Information Processing Systems*, 661–667.