

1 Thematic2.5D: A Toolkit for Evaluating 2D and 3D 2 Feature Effects in Supervised Classification

3 Taqi Hamoda  ¹, Hayat Rajani  ¹, Ahmed Alketbi², Arthur Gleason³, and
4 Nuno Gracias  ¹

5 1 University of Girona, Spain  2 Technology Innovation Institute, UAE 3 University of Miami, USA

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Open Journals](#) 

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).¹⁸¹⁹

6 Summary

7 The accurate detection and classification of objects in complex environments is a critical
8 task across various domains, requiring robust methods to process multi-modal data, such as
9 optical imagery and three-dimensional (3D) reconstructions. Object detection often leverages
10 computer vision, machine learning, and 3D modeling to distinguish targets from complex
11 backgrounds. Existing approaches relying solely on two-dimensional (2D) image features often
12 suffer from high false positive rates due to background variability, necessitating the integration
13 of 3D geometric data to enhance accuracy.

14 Thematic2.5D is an open-source Python package designed for supervised classification using
15 multi-modal data, including optical imagery and depth-maps derived from 3D reconstructions.
16 Building on the methodology of Gleason et al. (2015), the package provides a modular pipeline
for dataset creation, feature extraction, classification, and evaluation. Its key features include:

- **Dataset Creation:** Automated tile extraction and labeling from imagery and depth-maps for diverse applications.
- **Feature Extraction:** Extraction of 2D-derived (color, texture) and 3D-derived (curvature, rugosity) features, which can be combined into a hybrid 2.5D feature set.
- **Classification:** Support for binary (object vs. background) and multi-class 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.

28 Thematic2.5D serves as a versatile toolkit for researchers, data scientists, and educators working
29 in computer vision, machine learning, or thematic mapping. It integrates with popular scientific
30 Python libraries and supports workflows for both research and practical applications, such
31 as environmental surveys, urban mapping, or industrial inspections. The package provides a
32 classical framework to evaluate the relative contributions of 2D and 3D feature modalities,
33 enabling users to quantify the impact of integrating geometric data with traditional image
34 features for improved classification performance.

35 Statement of Need

36 Object detection in complex environments is a widespread challenge across fields like environmental
37 monitoring, urban planning, and industrial inspection, where distinguishing objects from
38 varied backgrounds is critical. Traditional detection methods, such as those using acoustic or
39 optical imagery, are often limited by resolution or background complexity. Optical imagery
40 combined with depth-maps offers high-resolution data for precise object identification, as

⁴¹ demonstrated by Gleason et al. (2015). However, existing software tools are often proprietary,
⁴² domain-specific, or lack the flexibility to handle multi-modal data in a unified framework.

⁴³ Thematic2.5D addresses this gap by providing a free, open-source Python toolkit that imple-
⁴⁴ ments a classical supervised classification pipeline, inspired by Shihavuddin et al. (2014), for
⁴⁵ semantic segmentation and thematic mapping. The package serves as a robust framework
⁴⁶ for evaluating the effectiveness of 2D-derived (e.g., color, texture) versus 3D-derived (e.g.,
⁴⁷ curvature, rugosity) features, allowing researchers to assess the relative contributions of these
⁴⁸ modalities to classification accuracy.

⁴⁹ By offering a comprehensive pipeline from data ingestion to evaluation, Thematic2.5D provides
⁵⁰ an accessible entry point for multi-modal classification research. The package's modularity
⁵¹ enables experimentation with new features, classifiers, or data sources (e.g., sonar, stereo
⁵² vision), while its accessibility supports educational use in courses on machine learning, computer
⁵³ vision, or data science.

⁵⁴ Background



Figure 1: The image shows the Thematic2.5D model's output for UWMM detection, comparing results trained on 2D, 3D, and 2.5D features (left, center, and right, respectively). The 2.5D result (right) was superior in this case. The model allows the user to select the optimal feature combination for their application.

⁵⁵ Object detection in complex scenes involves processing optical imagery to identify targets
⁵⁶ against varied backgrounds (e.g., natural landscapes, urban settings). The core challenge
⁵⁷ lies in distinguishing objects from similarly shaped or textured background elements. It was
⁵⁸ demonstrated that while 2D-derived features (color, texture) achieve moderate accuracy (>80%
⁵⁹ for binary classification), they suffer from high false positives due to background complexity
⁶⁰ (Shihavuddin et al., 2013). Incorporating 3D-derived features (e.g., curvature, rugosity) from
⁶¹ depth-maps significantly improves accuracy (89-95%) and reduces false positives by capturing
⁶² distinct geometric properties. This combined, hybrid approach, referred to as the 2.5D approach,
⁶³ enhances classification performance by leveraging both visual and geometric information.

⁶⁴ The Thematic2.5D pipeline provides a classical framework to systematically evaluate the
⁶⁵ contributions of 2D and 3D features. The methodology is broken down into a multi-stage
⁶⁶ pipeline. First, large survey images are processed into smaller, manageable tiles. This approach
⁶⁷ allows the model to learn local, high-resolution features and creates a large, diverse dataset
⁶⁸ suitable for training supervised machine learning models. The general workflow is as follows:

- ⁶⁹ 1. **Dataset Creation:** Optical images, depth maps, and ground-truth masks are used to
⁷⁰ generate a labeled dataset of image tiles.
- ⁷¹ 2. **Feature Extraction:** A comprehensive set of 2D-derived and 3D-derived features is
⁷² extracted from each tile.
- ⁷³ 3. **Model Training:** A classifier, such as an SVM, is trained on the extracted features.

- 74 4. **Inference:** The trained model is used to predict the locations of objects in new, unseen
 75 images.
 76 5. **Evaluation:** The model's predictions are compared against ground-truth data to assess
 77 performance quantitatively.
- 78 This project implements this entire workflow in a configurable and automated pipeline, building
 79 on the foundational work of Shihavuddin et al. (2014), to provide an accessible software tool
 80 for evaluating feature modalities in supervised classification.

81 Methodology

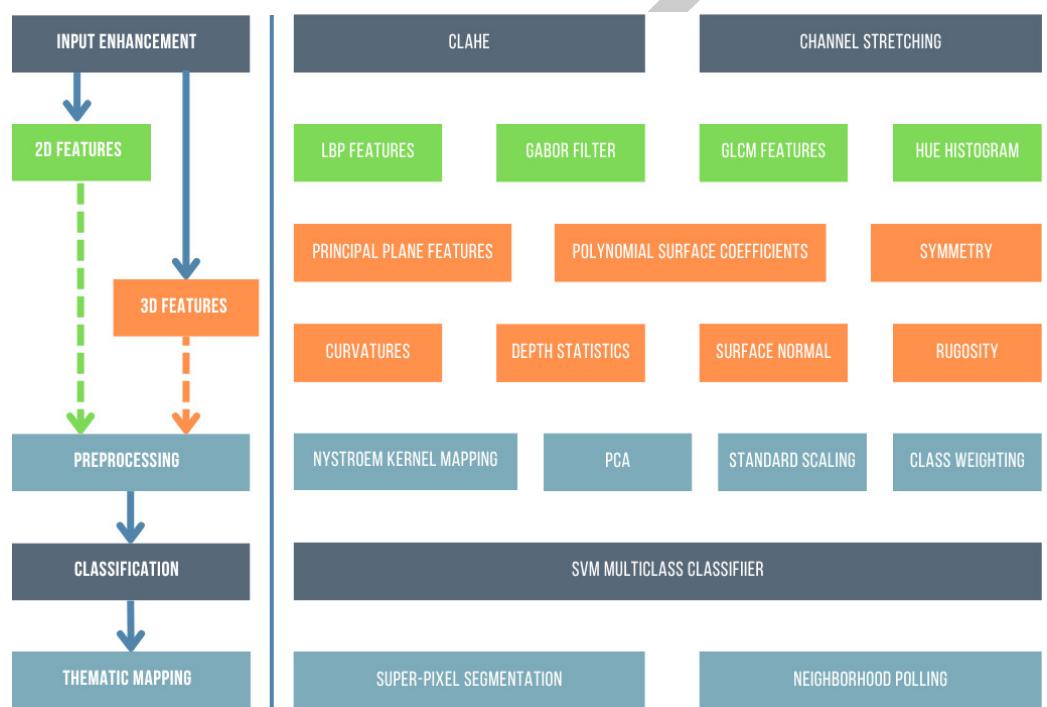


Figure 2: The implemented classification framework is depicted in the above flowchart. The left section outlines the primary stages of the pipeline, while the right section details the specific components and features utilized in each stage. The dashed arrows represent the path being optional.

82 The Thematic2.5D pipeline processes images, depth-maps, and masks to generate a labeled
 83 dataset of uniformly sized tiles for training and testing. It identifies potential object and
 84 background locations using masks, extracting corresponding square tiles from image and depth
 85 data. To address class imbalance, a fixed number of background tiles are sampled per image,
 86 and only a subset of available object pixels are used as tile centers. Data augmentation is
 87 applied by rotating object tiles at multiple angles.

88 From the processed tiles, 2D-derived and 3D-derived features are extracted, extending the
 89 feature set proposed by Shihavuddin et al. (2014) and Gleason et al. (2015):

- 90 ■ **2D-derived Features (from optical images):**
 - 91 – **Color Histograms:** HSV color distributions to capture background and object
 appearance (Shihavuddin et al., 2013).
 - 92 – **Local Binary Patterns (LBP):** Texture descriptors robust to illumination changes
 (Ojala et al., 1994).
 - 93 – **Gray Level Co-occurrence Matrix (GLCM):** Texture metrics (e.g., contrast, dissim-
 ilarity) for background characterization (Haralick et al., 1973).

- 97 – **Gabor Filters:** Edge and texture detection across multiple scales and orientations
 98 ([Granlund, 1978](#)).
- 99 ▪ **3D-derived Features (from depth maps):**
- 100 – **Principal Plane Features:** Statistics of depth values, polynomial coefficients from a
 fitted surface, and rugosity (ratio of surface area to planar area).
- 101 – **Curvatures and Normals:** Mean and Gaussian curvatures, shape index, curvedness,
 and surface normal vector statistics.
- 102 – **Symmetry Features:** Gabor filters applied to depth maps to capture structural
 symmetry.
- 103 The classifier is implemented using a SVM ([Cortes & Vapnik, 1995](#)) with a radial basis function
 (RBF) kernel, approximated via the Nystroem method ([Williams & Seeger, 2000](#)). The model
 supports training on 2D-derived, 3D-derived, or combined 2.5D feature sets (green, orange, or
 combined arrows respectively as seen in [Figure 2](#)). Prior to inference, superpixel segmentation
 ([Achanta et al., 2012](#)) identifies homogeneous regions in the input image. For each identified
 superpixel, a sliding window centered on its centroid is used to extract a series of overlapping
 tiles. These tiles are individually processed through the trained model to predict class labels. A
 neighborhood poll determines object 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 objects locations (see [Figure 1](#)).
- 104 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.

119 Model Training and 3D-derived Feature Extraction

120 The classification pipeline in Thematic2.5D relies on feature extraction and SVM-based classi-
 121 fication. For a tile $x \in \mathbb{R}^{m \times n}$ (image) and corresponding depth-map $d \in \mathbb{R}^{m \times n}$, the feature
 122 vector $f_{2.5D}$ combines 2D-derived and 3D-derived features:

$$f_{2.5D} = [f_{2D}, f_{3D}]$$

123 The 3D-derived features, f_{3D} , are extracted from the depth-map of each tile. This process
 124 involves several calculations to describe the geometry of the surface.

125 A 2D-derived polynomial surface of the third degree is fitted to the depth-map of a tile to
 126 model its shape ([Shihavuddin et al., 2014](#)). The equation for this surface is:

$$f(x, y) = p_1 + p_2x + p_3y + p_4x^2 + p_5xy + p_6y^2 + p_7x^2y + p_8xy^2 + p_9y^3$$

127 The nine coefficients of this polynomial (p_1, \dots, p_9) are extracted through least-squares fitting
 128 and used as features ([Shihavuddin et al., 2014](#)).

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

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

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

¹³³ With regards to surface curvature, the Gaussian (G) and mean (M) surface curvatures are first
¹³⁴ calculated from the partial derivatives of the depth map where the two principal curvatures,
¹³⁵ k_1 and k_2 , can then be derived (Shihavuddin et al., 2014):

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

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

¹³⁶ These principal curvatures are then used to calculate a shape index (S) and a curvedness index
¹³⁷ (C):

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

¹³⁸ 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)
¹³⁹ by the area of its orthogonal projection onto the principal plane (A_p):
¹⁴⁰

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

¹⁴¹ Once the full feature vector is assembled, the SVM classifier solves the following optimization
¹⁴² problem:

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

¹⁴³ subject to:

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

¹⁴⁴ where \mathbf{w} is the weight vector, b is the bias, ξ_i are slack variables, C is the regularization
¹⁴⁵ parameter, ϕ is the kernel mapping (RBF), and y_i are the class labels (Cortes & Vapnik, 1995).

¹⁴⁶ Acknowledgements

¹⁴⁷ This work was supported by the SERDP project MR23-3821, titled “Optically Derived 3D
¹⁴⁸ Models for Munitions Location and Identification”, funded by the DoD, DOE and EPA.

¹⁴⁹ References

- ¹⁵⁰ Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süstrunk, 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>
- ¹⁵³ Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- ¹⁵⁴ Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. <https://doi.org/10.1007/BF00994018>

- 156 Gleason, A. C. R., Shihavuddin, A., Gracias, N., Schultz, G., & Gintert, B. E. (2015).
157 Improved supervised classification of underwater military munitions using height features
158 derived from optical imagery. *OCEANS 2015 - MTS/IEEE Washington*, 1–5. <https://doi.org/10.23919/OCEANS.2015.7404580>
- 160 Granlund, G. H. (1978). In search of a general picture processing operator. *Computer Graphics
161 and Image Processing*, 8(2), 155–173. [https://doi.org/10.1016/0146-664X\(78\)90047-3](https://doi.org/10.1016/0146-664X(78)90047-3)
- 162 Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural features for image clas-
163 sification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6), 610–621.
164 <https://doi.org/10.1109/TSMC.1973.4309314>
- 165 Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D.,
166 Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk,
167 M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., ... Oliphant,
168 T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- 170 Ojala, T., Pietikainen, M., & Harwood, D. (1994). Performance evaluation of texture measures
171 with classification based on kullback discrimination of distributions. *Proceedings of 12th
172 International Conference on Pattern Recognition*, 1, 582–585 vol.1. <https://doi.org/10.1109/ICPR.1994.576366>
- 174 Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M.,
175 Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D.,
176 Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python.
177 *Journal of Machine Learning Research*, 12, 2825–2830.
- 178 Shihavuddin, A. S. M., Gracias, N., Garcia, R., Campos, R., Gleason, A. C. R., & Gintert,
179 B. (2014). Automated detection of underwater military munitions using fusion of 2D and
180 2.5D features from optical imagery. *Marine Technology Society Journal*, 48(4), 61–71.
181 <https://doi.org/doi:10.4031/MTSJ.48.4.7>
- 182 Shihavuddin, A. S. M., Gracias, N., Garcia, R., Gleason, A. C. R., & Gintert, B. (2013). Image-
183 based coral reef classification and thematic mapping. *Remote Sensing*, 5(4), 1809–1841.
184 <https://doi.org/10.3390/rs5041809>
- 185 Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D.,
186 Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson,
187 J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... SciPy
188 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in
189 Python. *Nature Methods*, 17, 261–272. <https://doi.org/10.1038/s41592-019-0686-2>
- 190 Walt, S. van der, Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager,
191 N., Gouillart, E., Yu, T., & contributors, the scikit-image. (2014). Scikit-image: Image
192 processing in Python. *PeerJ*, 2, e453. <https://doi.org/10.7717/peerj.453>
- 193 Williams, C. K. I., & Seeger, M. (2000). Using the nyström method to speed up kernel machines.
194 *Proceedings of the 14th International Conference on Neural Information Processing Systems*,
195 661–667.