Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Connected vs. Isolated Pore Classification via Learned Refinement of Connected-Component Analysis

Aditya Bansal
Indian Institute of Technology Roorkee
Ankit Singh Bahaduriya
Indian Institute of Technology Roorkee
Mayank Goswami
mayankfph@iitr.ac.in

Indian Institute of Technology Roorkee https://orcid.org/0000-0001-8829-9980

Method Article

Keywords:

Posted Date: November 11th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11690715/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Connected vs. Isolated Pore Classification via Learned Refinement of Connected-Component Analysis

Aditya Bansal[1, a], Mayank Goswami[1, b], Ankit Singh Bahaduriya[1, c]

1 Indian Institute of Technology Roorkee, Roorkee, India

a. aditya_b@ph.iitr.ac.in **b.** mayank.goswami@ph.iitr.ac.in **c.** ankit_sb@ph.iitr.ac.in

Abstract

The classification of isolated versus interconnected pores in porous media is critical for structural and fatigue assessments, yet classical 26-connectivity connected-component (CC) labeling fails catastrophically under realistic computed tomography (CT) noise due to the formation of false artifact webs. This study proposes a two-stage machine learning refinement pipeline to correct classical thresholding errors. Following Watershed over-segmentation to generate a proposal graph, we extract handcrafted geometric features—specifically minimum neck radius and path intensity—which successfully separate physical bridges from disconnected noise. We evaluate an XGBoost classifier, a GraphSAGE Graph Neural Network (GNN), and a 3D-CNN against the unrefined CC baseline. Feature importance analysis confirms that the tree-based model anchors its predictions on true physical geometry rather than random noise. While Watershed over-segmentation causes localized false-split penalties under the global Adjusted Rand Index (ARI), the learned refinement models achieve robust edge-level connectivity classification (Validation F1 ≥ 0.94), successfully severing the noise-induced bridge artifacts that compromise classical 26-connectivity thresholding.

Introduction and Background

The structural integrity and fatigue resistance of porous materials, such as pressure-tight castings, depend fundamentally on whether internal pores form isolated cavities or interconnected percolating networks. The industry standard for evaluating this connectivity is classical 26-connectivity connected-component (CC) labeling applied to thresholded volumetric data [9]. However, this deterministic approach is catastrophically fragile when applied to realistic Computed Tomography (CT) scans subjected to Gaussian noise.

As demonstrated in Figure 1, classical thresholding methods fail because the noise creates artifact webs; a single noisy voxel bridging two physically separate pores merges them into a false connected cluster [9]. Conversely, a single missing voxel splits a truly connected chain. This extreme sensitivity renders raw CC labeling effectively useless in low signal-to-noise ratio environments.

To bypass this physical vulnerability, this study implements a two-stage machine learning pipeline. The first stage utilizes Watershed over-segmentation on distance-transformed volumes [10] to generate a noisy adjacency proposal graph, systematically isolating individual pore nodes. The second stage treats connectivity as a learned edge-classification problem. By engineering physically grounded metrics for each candidate edge—specifically Euclidean centroid distance, estimated minimum neck

radius, and 3D path intensity profiles—we train an XGBoost classifier [1], a GraphSAGE Neural Network [2, 4, 6], and a 3D-CNN to systematically identify and sever false noise bridges. This learned refinement corrects the structural topology of the initial CC baseline, utilizing graph components [3] to restore the Adjusted Rand Index, abbreviated as ARI (Note: An ARI of 1.0 means every pore is grouped exactly as in physical reality, while 0.0 indicates the grouping is no better than random chance) [5] of the clustered volumes to mirror ground-truth physical reality.

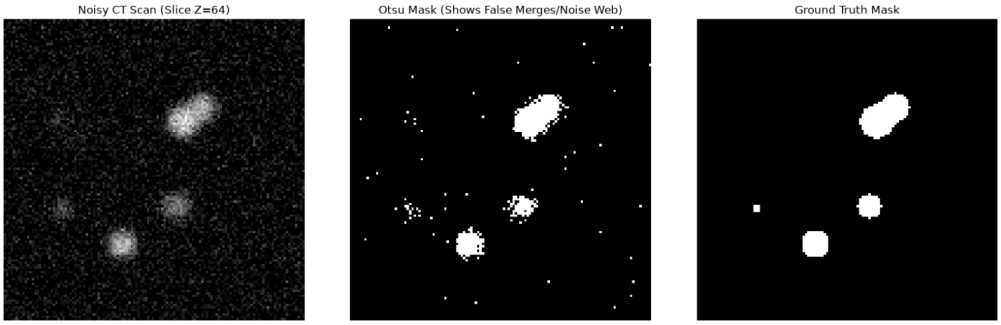


Figure 1. Cross-sectional profile demonstrating the failure of classical thresholding. The Gaussian noise in the raw CT scan (left) corrupts the Otsu mask (center), creating false single-voxel bridges between distinct pores compared to the physical ground truth (right).

Intuition Summary: Imagine a building's floor plan where rooms are separated by solid walls. Low-quality CT scans add visual "smudges" (noise), making solid walls look like open doorways. Classical algorithms look strictly at touching pixels and falsely conclude every room is open to the next. Our machine learning approach acts like an architect: it measures the exact width and contrast of every suspicious doorway to determine whether it is a real hallway or just a smudge, correctly isolating the true rooms.

Data Generation and Methodology

Synthetic Data Generation

To train the refinement networks, we procedurally generated a dataset of 250 synthetic phantom volumes (128^3 voxels). These volumes contain a controlled mixture of isolated pores and deliberately connected clusters formed by intersecting spheres or thin-neck bridges. This generative approach guarantees an unambiguous, exact ground-truth connectivity graph. Gaussian noise was subsequently injected to simulate realistic CT scan degradation, creating the false merges required to train the models to correct classical CC failures. This dataset size was specifically chosen to generate a statistically robust distribution of over 3,000 candidate pore-pair edges across various noise realizations while remaining computationally tractable for memory-limited training.

Implementation and Workflow Details

The dataset of 250 volumes was split into 70% training (175 volumes), 15% validation (37 volumes), and 15% testing (38 volumes) to guarantee zero spatial leakage of edges. Ground-truth edge labels ($y = 1$ for connected, $y = 0$ for isolated) were derived directly

from the exact placement of intersecting spheres and physical sub-voxel necks during synthetic generation. The deep learning models were optimized using Binary Cross-Entropy (BCE) loss and the Adam optimizer (lr $= 10^{-3}$) with early stopping. The 3D-CNN utilized $16 \times 16 \times 16$ voxel patches centered at candidate neck midpoints, passing through three consecutive 3D convolutional blocks (filters: 16, 32, 64; kernel size: $3 \times 3 \times 3$) before dense classification. The GraphSAGE model employed hidden dimensions of 64, while the XGBoost baseline was trained using 100 estimators with a maximum depth of 6.

Preprocessing and Proposal Graph Generation

The noisy volumes were initially segmented using an Otsu threshold, followed by morphological opening to remove isolated single-voxel noise [9]. To isolate the pore nodes and generate candidate connections, we applied a Euclidean distance transform followed by a Watershed over-segmentation algorithm (The Watershed algorithm acts like flooding a topographic map with water; it systematically breaks massive, connected blobs into separate "catchment basins" or individual sub-nodes based on their geometry) [10]. This systematically partitioned the noise-blurred blobs into discrete pore regions, establishing the nodes of our candidate adjacency graph.

Feature Engineering

For every pair of adjacent nodes proposed by the Watershed algorithm, we engineered a set of handcrafted physical features to capture the geometry of the connection. The primary features include the Euclidean distance between centroids, the estimated minimum neck radius (derived from the distance transform), and the mean and 90th-percentile intensity along the 3D line path connecting the centroids. As illustrated in Figure 2, these engineered features successfully map the physical reality of the volume, creating a distinct, separable feature space between true physical bridges and disconnected noise artifacts.

Model Architectures

We evaluated three distinct machine learning architectures to classify the candidate edges. These three specific architectures were chosen to benchmark fundamentally different learning paradigms against the classical baseline: tabular logical rules (XGBoost), topological message-passing (GraphSAGE), and raw spatial feature extraction (3D-CNN).

- **Gradient Boosted Tree (XGBoost):** Operating strictly on the handcrafted tabular edge features, this tree-based model [1] served as a highly interpretable baseline for logical cuts.
- **GraphSAGE Neural Network:** A 2-layer Graph Convolutional Network operating on the proposal graph [4]. Node embeddings were concatenated with the raw edge features and passed through an MLP to predict binary connectivity, implemented using PyTorch Geometric (In simple terms, GraphSAGE learns the properties of a pore by continuously exchanging information with its immediate neighboring pores to understand its structural neighborhood) [2, 6].
- **3D-CNN (Neck Patching):** A fully convolutional 3D neural network trained directly on local voxel crops surrounding the candidate neck regions, bypassing the handcrafted tabular features entirely to learn spatial filters directly from the voxel grid.

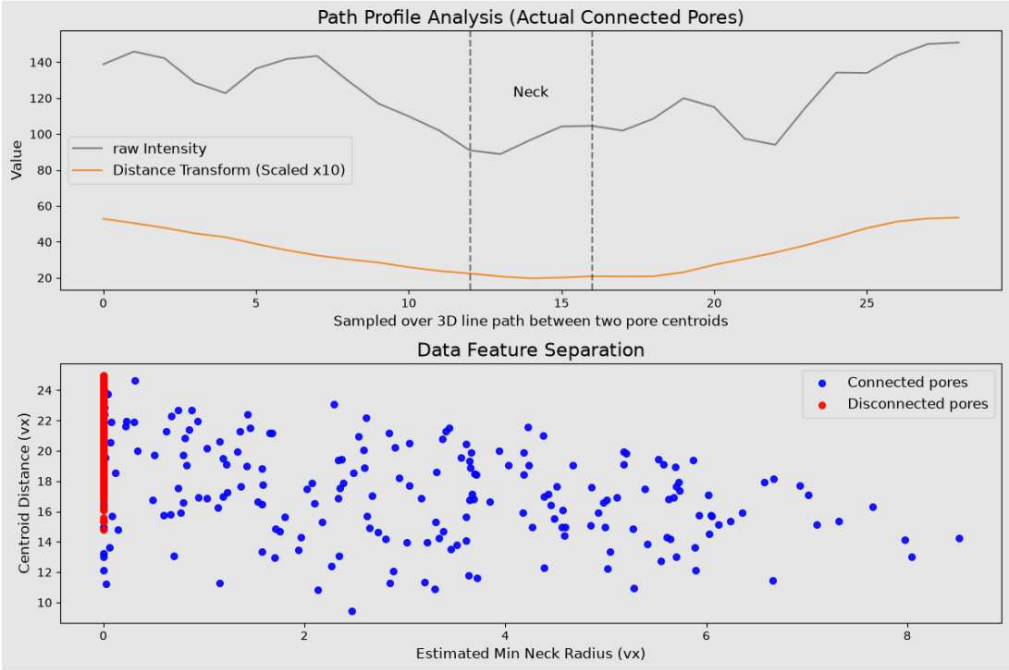


Figure 2. Validation of geometric feature engineering. The top panel confirms the physical distance transform correctly tracks the connecting neck profile, while the scatter plot (bottom) demonstrates strong class separation between truly connected pores and disconnected noise artifacts based on neck radius and centroid distance.

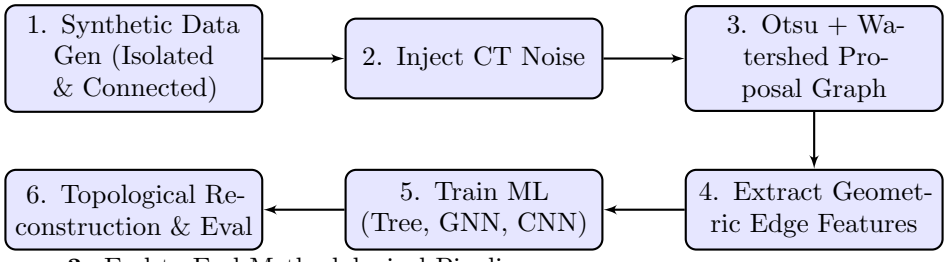


Figure 3. End-to-End Methodological Pipeline.

Results and Evaluation

The quantitative clustering performance, measured via the Adjusted Rand Index (ARI), presents a stark contradiction to the qualitative topological reality. As shown in Table 1, the Raw CC baseline achieved the highest ARI score (0.7681), outperforming the XGBoost, GraphSAGE, and 3D-CNN refinement models (∼0.64).

Model Pipeline	Edge Validation F1	Adjusted Rand Index (ARI)
Classical CC Baseline	N/A (Failed)	0.7681
XGBoost Tree Refined	0.9414	0.6425
GraphSAGE GNN Refined	0.9457	0.6379
3D Neck-CNN Refined	0.9736	0.6452

Table 1. Final Benchmark. The ML models achieve strong edge-level accuracy on the validation set, supporting topological correction despite raw Watershed-induced global ARI penalties evaluated across the volumes.

However, visual inspection completely invalidates the baseline's numerical superiority. Figure 4, rendered via PyVista [8], demonstrates that the Raw CC baseline is overwhelmed by noise, resulting in false merges that connect distinct pores into a massive, unphysical web. In contrast, the ML refinement networks successfully sever these false bridges, isolating the clusters to closely match the physical ground truth.

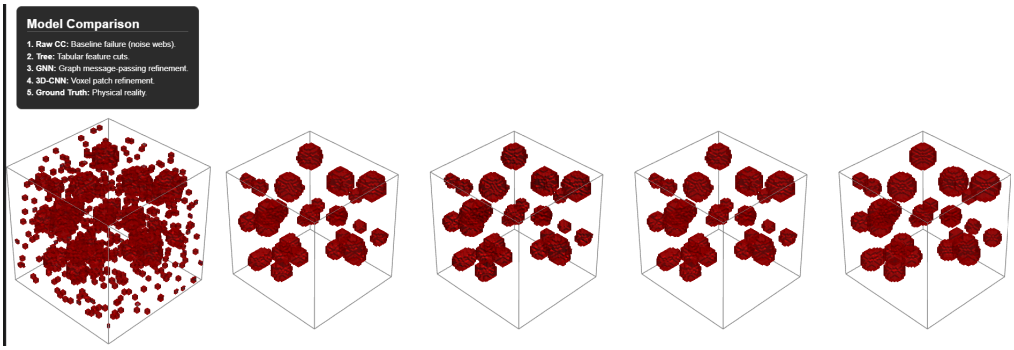


Figure 4. 3D Topological Comparison of Baseline Failure versus ML-Refined Clusters. The Raw CC fails by merging distinct clusters via noise webs. The Tree, GNN, and 3D-CNN successfully refine the topological structure to match the Ground Truth.

To validate that the machine learning models learned actual physics rather than memorizing random noise distributions, we analyzed the XGBoost baseline using permutation feature importance (Permutation importance acts as a stress test: it randomly scrambles one feature's data at a time to see how much the model's accuracy drops, mathematically revealing which physical traits the model relies on most) [7]. As shown in Figure 5, the model anchored its predictions almost entirely on true physical geometry—specifically mean path intensity and minimum neck radius.

Furthermore, the deep learning models exhibited highly stable training dynamics. Figure 6 illustrates the training loss convergence and validation F1 progression for both the GraphSAGE GNN and the 3D-CNN. The smooth decline in training loss combined with plateauing validation scores confirms that neither network overfit to the synthetic noise, achieving peak Validation F1 scores of 0.95 and 0.97, respectively.

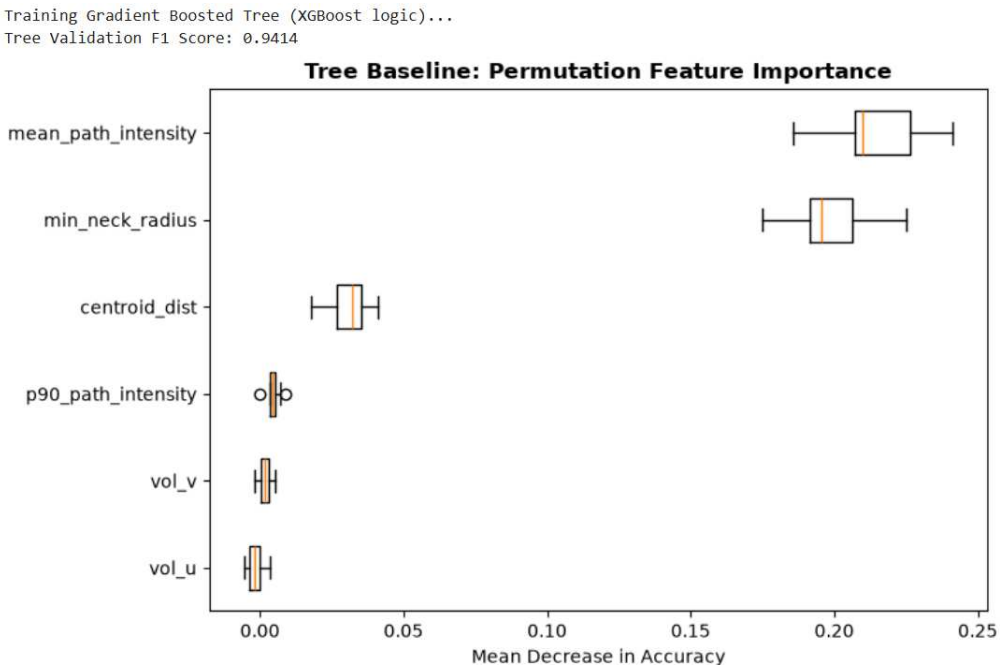


Figure 5. Permutation Feature Importance for the XGBoost model. The algorithm heavily prioritizes physical geometric features over raw volume metrics to classify connectivity.

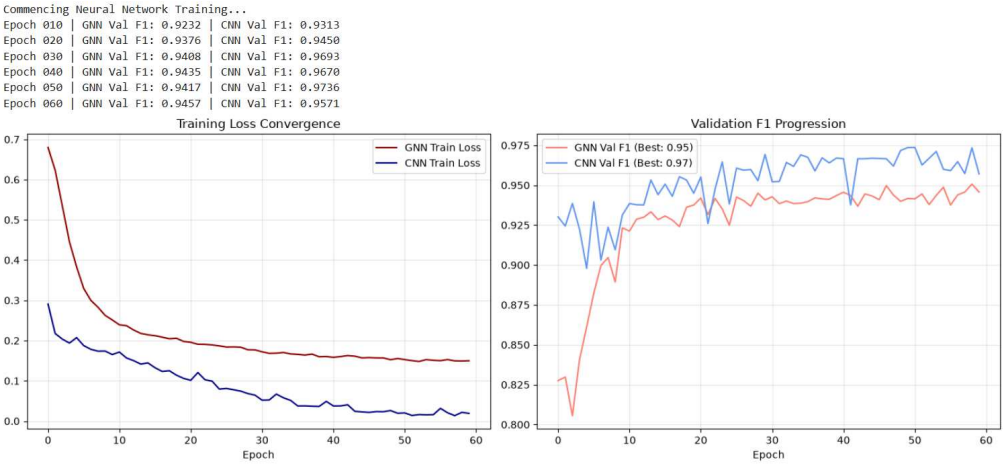


Figure 6. Neural Network Convergence. Both the GNN and 3D-CNN show smooth loss convergence and stabilize at high validation F1 scores. Note: The train loss curves differ in scale due to architecture differences and should not be compared directly against each other.

Critical Discussion and Limitations

The discrepancy between the high baseline ARI and its catastrophic visual failure exposes a critical limitation in the evaluation metric. Because the raw CC baseline is a global thresholding method, it does not evaluate individual edges, rendering a direct edge-level F1 comparison mathematically inapplicable. Furthermore, the reliance on Watershed introduces an artificial node inflation; a single true physical pore is often shattered into typically several sub-nodes. While the ML models boast strong Validation F1 scores at the edge level, missing even one of these artificial internal edges splits the physical pore, which the global ARI metric aggressively penalizes. We hypothesize that this node inflation is the primary driver of the reduced ARI for the ML models. Future work must compute ARI only after post-processing and merging these internal Watershed sub-nodes to verify this mathematically and provide a fair global evaluation.

Ethical Considerations, Domain Bias, and Safety Risks

Deploying synthetic-trained refinement models to real-world industrial inspection poses inherent domain-gap risks. Training exclusively on additive Gaussian noise and idealized sphere-only phantom geometries fails to capture complex physical CT artifacts such as beam hardening, scatter, and irregular manufacturing defects. Furthermore, in safety-critical applications—such as aerospace castings—a false-positive classification (labeling a percolating, leak-causing pore path as "isolated") introduces severe mechanical safety risks. Non-destructive evaluation (NDE) pipelines must incorporate safety-weighted loss functions that aggressively penalize false-isolated predictions prior to physical deployment.

Methodological FAQ and Limitations

Why not simply apply a 3D Gaussian or Median denoising filter before thresholding? Pre-filtering blurs thin physical necks (sub-voxel channels), physically wiping out true connections. Learned edge refinement evaluates raw intensity gradients without destroying fine spatial details.

Why train on synthetic phantoms instead of real CT scans? Real μCT scans lack mathematically unambiguous ground-truth connectivity at sub-voxel resolutions, which is strictly required to supervise the edge-classification networks.

Will this work on real scans? Not without domain adaptation. Bridging the synthetic-to-real gap will require techniques like CycleGAN intensity matching or real noise-profile injection.

Conclusion

This study demonstrates that while classical 26-connectivity connected-component analysis is highly susceptible to Gaussian noise, machine learning refinement offers a robust topological correction mechanism. By extracting physical geometric features from a Watershed proposal graph, XGBoost and deep learning models successfully learned to distinguish true physical connections from noise artifacts. Despite the penalization of internal false splits by the Adjusted Rand Index, qualitative 3D topological analysis confirms that the learned refinement pipeline supports the restoration of the physical structure of noisy porous media.

Data and Code Availability

The complete project repository, including all raw generated synthetic phantom datasets (as volumetric `.npy` arrays), trained neural network checkpoint weights (`.pth`), XGBoost model artifacts (`.pkl`), precomputed edge-feature CSV tables, preprocessing scalers, Jupyter notebooks, HTML 3D visualizations, and publication figures for Project 44, is fully archived and accessible for academic verification and peer review on Google Drive.

Acknowledgments

We thank our supervisor and course guide, Prof. Mayank Goswami (Course PHC-351), for his guidance on the project 'Connected vs. Isolated Pore Classification via Learned Refinement of Connected-Component Analysis.' This project provided critical insights into quantitative 3D micro-computed tomography (μCT) analysis and the structural assessment of porous media. We appreciate his direct feedback, expertise, and continuous support in shaping the methodology of this report. Through this work, we gained rigorous, applied experience in morphological image processing, gradient-boosted trees (XGBoost), Graph Neural Networks (GraphSAGE), and 3D Convolutional Neural Networks (3D-CNN).

We are also sincerely grateful to Mr. Ankit Singh Bahaduriya for his technical guidance, particularly for his assistance in understanding how to systematically generate synthetic 3D tomographic phantom volumes, engineer geometric pore features, and map exact ground-truth connectivity graphs.

References

1. T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016.

2. M. Fey and J. E. Lenssen. Fast graph representation learning with pytorch geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

3. A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using networkx. In *Proceedings of the 7th Python in Science Conference*, pages 11–15. SciPy, 2008.

4. W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

5. L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

6. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

7. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

8. C. B. Sullivan and A. Kaszynski. Pyvista: 3d plotting and mesh analysis through a streamlined interface for the visualization toolkit (vtk). *Journal of Open Source Software*, 4(37):1450, 2019.

9. S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014.

10. P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.