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.

Learning to Estimate Surface Area from Voxelised 3D Objects: A Comparative Study of Classical Geometry, 3D Convolutional Networks and PointNet

Tanmay Kumar
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: Surface-area estimation, voxelisation, 3D morphology, Marching Cubes, 3D CNN, PointNet, geometric regression.

Posted Date: November 13th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11715919/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.

Learning to Estimate Surface Area from Voxelised 3D Objects: A Comparative Study of Classical Geometry, 3D Convolutional Networks and PointNet

Tanmay Kumar[1] and Mayank Goswami [1, 2, #]
[1] Department of Physics, IIT Roorkee
[2]*Mehta Family School of Data Science and Artificial Intelligence*, IIT Roorkee,
Roorkee, Uttarakhand-247667, India.
[#]mayank.goswami@ph.iitr.ac.in

Abstract

Surface area is a fundamental descriptor in three-dimensional morphology, with relevance to porous materials, biomedical imaging, particulate systems and computational geometry. In voxelised data, however, surface area is not directly represented: it must be inferred from a discrete occupancy field, and conventional estimators exhibit resolution- and noise-dependent bias. This work investigates whether learned regressors can estimate the surface area of a 3D object more accurately than direct geometric estimators when only its binary voxel representation is available. A synthetic dataset of 15,000 geometric instances was generated across five shape families—spheres, ellipsoids, cylinders, superellipsoids and Gaussian-random-field (GRF) blobs—with three voxel resolutions (32^3, 64^3, 128^3) and three levels of synthetic boundary perturbation (0, 0.02 and 0.05), giving 135,000 volume variants. Ground-truth areas were obtained analytically for spheres and cylinders and from a higher-resolution Marching Cubes reference mesh for the remaining families. A 3D convolutional neural network (CNN) was trained independently at each resolution, while PointNet was trained on 2,048 surface points sampled from Marching Cubes meshes at 32^3. On the held-out test set, PointNet achieved an MAEof0.1953andMAPEof2.73%, outperforming the matched CNN (MAE 0.2813, MAPE 5.25%), Marching Cubes (MAE 0.7889, MAPE 13.95%) and voxel-face counting (MAE 3.8983, MAPE 58.07%). The CNN remained substantially more accurate than the classical baselines at all three resolutions, although its aggregate error was not monotonic with resolution. The results demonstrate the promise of learned geometric priors for surface-area estimation while exposing important limitations arising from shape imbalance, reference-mesh uncertainty, architecture scaling and the dependence of PointNet preprocessing on Marching Cubes. **Index Terms—** Surface-area estimation, voxelisation, 3D morphology, Marching Cubes, 3D CNN, PointNet, geometric regression.

1 Introduction

Three-dimensional imaging and simulation commonly represent objects on a regular grid. Computed tomography, micro-CT, materials microscopy, medical segmentation and numerical simulation all produce data in which a continuous object is sampled into finite voxels. A voxel stores occupancy or an intensity value over a finite volume; it does not explicitly store the continuous boundary. Consequently, geometric quantities such as surface area must be estimated from a discretised representation.

Surface area is not merely a visual descriptor. In porous media it controls the area available for interfacial transport and reaction; in biomedical morphology it characterises tissue and trabecular interfaces; in particulate materials it affects adsorption, dissolution and contact phenomena. Errors in area estimation can therefore propagate into downstream physical models. The problem is difficult because a voxel grid approximates a surface through a staircase-like boundary. Increasing the grid resolution can reduce some discretisation errors, but also increases memory and computation, and may expose the estimator to more boundary perturbations.

Two common geometric approaches are exposed voxel-face counting and isosurface reconstruction using MarchingCubes. Facecounting is inexpensive but treats the boundary as axis-aligned square facets. Marching Cubes reconstructs a triangle mesh by interpolating an isosurface through each grid cell [1]. It

generally provides a smoother geometric approximation, but the reconstructed area remains sensitive to grid resolution, thresholding and local noise. Both methods compute the surface from the observed voxel pattern and can therefore respond strongly to small boundary irregularities.

A learned estimator offers a different strategy. Rather than reconstructing every local surface element and summing its area, a model can learn statistical relationships between the three-dimensional occupancy pattern and its known area. A CNN can extract local volumetric features and combine them into a global regression.

PointNet instead operates on unordered point sets and uses shared pointwise transformations with a symmetric aggregation operation to obtain permutation-invariant representations [2]. These methods may learn to discount voxel-scale irregularities that do not systematically change the underlying geometric surface.

The present study compares these approaches under controlled synthetic conditions. It evaluates a 3D CNN at three resolutions, compares it with two classical estimators at matched test instances, and evaluates PointNet against the CNN and classical methods at 32^3. The study is designed as a controlled benchmark rather than a claim of immediate clinical or industrial deployment. Its central questions are: (i) how much error is introduced by direct voxel-based area estimation; (ii) whether learned regressors reduce that error; (iii) how performance changes with resolution, shape family and synthetic boundary noise; and (iv) what is gained or lost by representing the volume as a point cloud.

2 Problem Objectives and Motivation

2.1 Objectives

The work has five objectives:

1. Construct a reproducible synthetic dataset with known geometric instances and multiple voxel/noise variants.
2. Establish analytic or numerical reference surface areas and implement two classical baselines.
3. Train a 3D CNN independently at 32^3, 64^3 and 128^3, using instance-level train/validation/test separation.
4. Train a PointNet regressor at 32^3 using a fixed size surface point cloud and compare it on exactly matched test variants.
5. Quantify performance using absolute, relative and signed-error metrics, including stratification by noise and shape family.

2.2 Why the problem matters

The motivation is not that classical surface reconstruction is obsolete. Rather, explicit reconstruction may be unnecessarily sensitive or computationally expensive when the desired output is a single global descriptor. A learned estimator could act as a fast surrogate for repeated measurements, particularly where the geometry distribution is known or can be represented in training. However, this advantage is conditional: a model trained on synthetic shapes may fail on real segmented objects whose topology, anisotropy, partial-volume effects and intensity statistics differ from the training distribution. The benchmark therefore examines both performance and the limits of the conclusions that can be drawn.

3 Theory and Classical Estimators

3.1 Voxel representation and physical scale

Let a binary volume be $V[i, j, k] \in \{0, 1\}$, where 1 denotes an occupied voxel and 0 denotes background. The sampled volume represents an object in a cubic domain $[-1, 1]^3$. For an $N \times N \times N$ grid, the isotropic voxel spacing S

$$h = \frac{2}{N}, \qquad N \in \{32, 64, 128\}. \tag{1}$$

A face of one voxel therefore has area h^2. The same dimensionless domain is used across resolutions, so all reported surface areas are in squared domain units. A change in voxel count does not imply a change in the underlying physical object; it changes the sampling of that object.

3.2 Exposed-voxel-face counting

Each occupied voxel has six axis-aligned faces. A face contributes to the external surface if its neighbouring voxel is empty or lies outside the volume. If n_f is the total number of exposed faces, the estimator is

$$A_{\text{voxel}} = n_f h^2. \tag{2}$$

In implementation, the volume is padded with a layer of background voxels. Differences along each of the three axes are counted; each occupancy transition corresponds to one exposed face. Padding ensures that faces on the outer boundary are included. The method is computationally simple, but the resulting surface is a union of axis-aligned squares. Oblique and curved surfaces are represented by a staircase, producing orientation dependent and often substantial positive bias.

3.3 Marching Cubes

Marching Cubes reconstructs an isosurface from scalar values at the corners of each grid cell [1]. For a binary occupancy field, the isovalue is set to 0.5. Each cube is classified according to which of its eight vertices lie above the threshold. The resulting edge intersections are linearly interpolated, and a lookup table determines the triangles that connect them. The resulting mesh approximates the boundary between foreground and background.

If the mesh contains triangular faces T_m, its area is the sum of triangle areas:

$$A_{\mathrm{MC}} = \sum_m \frac{1}{2} \left\| (\mathbf{v}_{m,2} - \mathbf{v}_{m,1}) \times (\mathbf{v}_{m,3} - \mathbf{v}_{m,1}) \right\|. \qquad (3)$$

The input is padded with zeros so that a closed surface is formed around objects touching the grid boundary. Vertex coordinates are scaled using the isotropic spacing h.

Marching Cubes reduces the axis-aligned staircase effect, but it does not eliminate discretisation bias. Noise can introduce small-scale surface undulations, and triangle area summation can count these undulations as additional area.

3.4 Learning as geometric regression

The learning task is to approximate a mapping

$$f_\theta : V \mapsto\!\!\rightarrow A, \qquad (4)$$

where the input is a binary 3D volume and the scalar target is the reference surface area. Unlike a mesh estimator, the model is not required to output an explicit surface. It may learn shape-scale cues and typical boundary statistics from the training distribution. This can reduce sensitivity to local voxel perturbations, but it also creates a risk of learning dataset-specific priors rather than a general geometric law.

4 Previous Work

The Marching Cubes algorithm introduced by Lorensen and Cline remains a standard method for extracting polygonal isosurfaces from volumetric data [1]. Its widespread use in medical and scientific visualisation reflects its local, grid-based construction and straightforward triangle-mesh output. Its geometric accuracy depends on sampling, interpolation and the topology of the sampled field.

Point-based deep learning became practical with PointNet, which directly consumes unordered point sets and uses a symmetric max-pooling operation to obtain a representation invariant to point ordering [2]. PointNet demonstrated that point clouds can be processed without first converting them to regular 3D grids. Subsequent work such as PointNet++ introduced hierarchical local neighbourhood processing to capture multi-scale geometric structure [3]. In this project, the original PointNet formulation was selected as a compact baseline for scalar area regression.

Three-dimensional convolutional networks have also been used for volumetric recognition and segmentation. The 3D CNN approach considered here follows the general principle of applying spatial convolutions over all three volume axes, allowing local volumetric patterns to be progressively encoded into feature maps. The model is deliberately compact and is not intended as an exhaustive architecture search. The focus is a matched comparison across the available data resolutions and a transparent evaluation of the observed behaviour.

A key distinction from much of the preceding work is the target variable: this study predicts a single global geometric quantity rather than a class label, voxel mask Table 1: Dataset composition by shape family.

Family	Instances	Variants
Sphere	1,250	11,250
Ellipsoid	1,250	11,250
Cylinder	1,250	11,250
Superellipsoid	1,250	11,250
GRF blob	10,000	90,000
Total	15,000	135,000

or pointwise correspondence. It also uses instance-level grouping across resolution and noise variants to avoid placing near-identical variants of one underlying shape in different data splits.

5 Dataset Generation and Reference Areas

5.1 Shape families

The dataset contains 15,000 underlying geometric instances distributed across five families. Sphere, ellipsoid, cylinder and super ellipsoid families each contain 1,250 instances. GRF blobs contain 10,000 instances. The larger GRF subset was chosen to provide a broad set of irregular smooth shapes; however, this creates a family imbalance that must be considered when interpreting aggregate metrics.

Spheres and cylinders provide geometries with analytic surface-area formulae. Ellipsoids and superellipsoids provide smooth but more varied curvature and aspect ratios. GRF blobs provide irregular, spatially correlated shapes that are less readily described by a simple closed-form area formula. These families collectively span regular and irregular boundary structures, but they do not represent the full diversity of real segmented objects.

5.2 Resolution and noise variants

Each underlying instance is rendered at three resolutions and three boundary-noise levels, yielding nine variants per instance and 135,000 volume variants in total. The resolutions are 32^3, 64^3 and 128^3. The perturbation levels are $\sigma \in \{0, 0.02, 0.05\}$. Noise is applied to the implicit geometric field before thresholding to create controlled boundary uncertainty. It is not a complete model of CT acquisition noise, scanner blur, partial volume intensity distributions or segmentation error.

5.3 Reference area construction

For spheres, the reference area is computed from $A = 4\pi r^2$. For cylinders, the corresponding closed-surface area includes the curved lateral area and both end caps, $A = 2\pi r h_c + 2\pi r^2$, where r is radius and h_c is cylinder height. For ellipsoids, superellipsoids and GRF blobs, the reference area is computed from a high-resolution MarchingCubesmeshatresolution256. Thus, the latter labels are numerical mesh references, not exact analytic areas of the underlying continuous implicit surfaces.

Table 2: Unique underlying instances in each split.

Family	Train	Val.	Test
Sphere	875	187	188
Ellipsoid	875	187	188
Cylinder	875	187	188
Superellipsoid	875	187	188
GRF blob	7,000	1,500	1,500
Total	10,500	2,248	2,252

A convergence check compared mesh areas at resolutions 128, 192 and 256 for generated instances. This provides evidence about the stability of the reference calculation but does not establish exact convergence for every possible geometry. The use of the same broad class of mesh extraction for some labels and for the classical test baseline also means that the reference is not wholly independent of the reconstruction method. This limitation is discussed in Section VIII.

5.4 Splitting protocol

The 15,000 underlying instances are divided into train, validation and test partitions in a stratified, instance grouped manner. The split proportions are approximately 70%, 15% and 15%. All nine variants of a given instance remain in the same partition. This prevents leakage in which a model sees one noise/resolution variant of a test geometry during training. The resulting variant counts are 94,500 training, 20,232 validation and 20,268 test samples. Each resolution-specific test evaluation contains 6,756 samples: 2,252 at each noise level.

6 Method

6.1 Classical baseline implementation

Both classical methods are evaluated on the same test volumes used for the learned models. The voxel estimator uses Eq. (2). Marching Cubes uses the 0.5 isovalue, zero padding, physical spacing (h, h, h), and triangle-mesh area summation using Eq. (3). Matching is performed on the tuple (instance identifier, resolution, noise level), ensuring that all methods are scored on identical samples.

6.2 3D CNN architecture

The CNN accepts a single-channel binary volume. Its feature extractor contains four 3D convolutional blocks with channel widths 16, 32, 64 and 128. Each convolution uses a 3 × 3 × 3 kernel with unit padding, followed by batch normalisation and ReLU activation. A 2 × 2 × 2 max-pooling operation follows each of the

first three blocks. Adaptive global average pooling then maps the final feature tensor to a 128-dimensional vector, independent of the spatial dimensions remaining after convolution. The regression head is a fully connected layer from 128 to 64 units, ReLU, dropout with probability 0.2, and a final scalar output. The model contains approximately 299,777 trainable parameters.

The model is trained independently for each input resolution. The use of a common architecture makes the experiments straightforward to compare, but its receptive field occupies a different fraction of the input domain at each resolution. Therefore, the three models do not have identical effective spatial context in physical coordinates. This is a methodological limitation rather than evidence that one resolution is intrinsically harder.

6.3 CNN training

The CNN is trained using Smooth L1 loss and the AdamW optimiser with learning rate 10^{-3} and weight decay 10^{-4}. A ReduceLROnPlateau scheduler halves the learning rate after three validation plateaus, with a minimum learning rate of 10^{-6}. Training is capped at 40 epochs and uses early stopping with patience 7 based on validation MAE and a minimum improvement threshold of 10^{-4}. The random seed is 42. Batch sizes are 16, 4 and 2 for resolutions 32, 64 and 128 respectively, reflecting the increasing memory requirement of volumetric convolution. The best validation checkpoint is retained for test evaluation.

6.4 PointNet input construction

For the PointNet branch, each binary volume is converted into a triangle mesh using Marching Cubes. A point cloud of 2,048 points is sampled from the mesh. Triangles are selected with probability proportional to their surface area, and points are sampled uniformly within the selected triangles using barycentric coordinates. The point cloud is centred, but its absolute scale is retained. Unit-sphere normalisation is not applied because it would remove the size information needed to recover area: under a uniform scale change by factor s, area changes as $A' = s^2 A$.

6.5 PointNet architecture and training

PointNet applies shared pointwise multilayer transformations to the 3D coordinates and aggregates point features with symmetric max pooling, making the representation invariant to input ordering [2]. A 3D input transform network(T-Net) predicts a 3×3 alignment matrix. Its regularisation term penalises deviations from orthogonality:

$$L_{\text{reg}} = \| TT^{\top} - I \|_F^2. \quad (5)$$

The total training loss is Smooth L1 regression loss plus $0.001 L_{\text{reg}}$. PointNet is trained at 32^3 with Adam, learning rate 10^{-3}, weight decay 10^{-5}, batch size 32, a maximum of 50 epochs and patience 8. The best validation checkpoint is used for the test set.

An important qualification is that PointNet's points are sampled from a Marching Cubes mesh. It is therefore not independent of Marching Cubes preprocessing. The comparison measures whether PointNet can estimate area more accurately from a fixed-size representation of that mesh than the direct mesh-area sum; it does not compare two fully independent input pipelines.

6.6 Evaluation metrics

For predictions $\hat{A}_i$ and reference areas A_i, the error is $e_i = \hat{A}_i - A_i$. We report mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MAPE), mean bias and coefficient of determination:

MAE $(1/n)\,\Sigma\,\lvert e_i \rvert$ — Typical error, in area units	**RMSE** $\sqrt{((1/n)\,\Sigma\, e_i^{\,2})}$ — Penalises large errors	**MAPE** $(100/n)\,\Sigma\,\lvert e_i / A_i \rvert$ — Relative error in %
Bias $(1/n)\,\Sigma\, e_i$ — Over (+) or under (−) estimation	**Ratio** $(1/n)\,\Sigma\,\hat{A}_i / A_i$ — >1 over, <1 under	**R^2** $1 - SSres / SStot$ — vs. predicting the mean

MAE and RMSE are in squared domain units. MAPE is useful for comparing relative errors across area magnitudes, while bias indicates systematic over- or underestimation. A negative R^2 means that the estimator performs worse, in squared-error terms, than predicting the test-set mean for every sample.

Table 3: Matched test performance at 32^3 (6,756 samples).

Method	MAE	RMSE	MAPE	Bias	R^2
PointNet	0.1953	0.2682	2.73%	0.0034	0.9868
3D CNN	0.2813	0.3488	5.25%	0.0174	0.9777
Marching Cubes	0.7889	1.1435	13.95%	0.7867	0.7603
Voxel faces	3.8983	4.1751	58.07%	3.8983	−2.1959

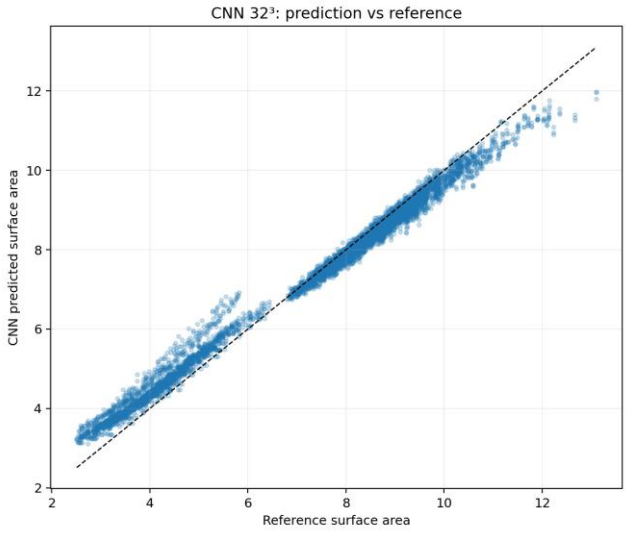


Figure 1: CNN predicted versus reference surface area at 32^3. The concentration around the diagonal indicates good aggregate agreement, with residual scatter and family-dependent structure.

7 Results

7.1 Overall comparison at matched resolution

Table 3 reports the primary matched comparison at 32^3. All four methods are evaluated on the same 6,756 test variants. PointNet achieves the lowest error, followed by the CNN, Marching Cubes and voxel-face counting. Relative to Marching Cubes, PointNet reduces MAE by approximately 75.2%; relative to voxel counting, the reduction is approximately 95.0%. The CNN reduces MAE by approximately 64.3% relative to Marching Cubes.

The signed bias is particularly informative. PointNet and the CNN are close to unbiased in aggregate, whereas Marching Cubes and voxel counting overestimate area. Voxel counting has a negative coefficient of determination, reflecting a large systematic error relative to the spread of reference areas. The point-cloud estimator's mean prediction ratio is 1.0025, compared with 1.0232 for the CNN, 1.1389 for Marching Cubes and 1.5807 for voxel counting.

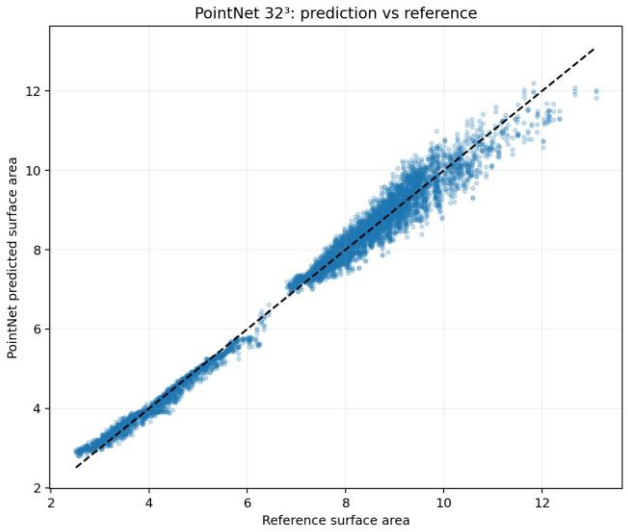


Figure 2: PointNet predicted versus reference surface area at 32^3. The tighter alignment is consistent with its lower matched MAE and RMSE.

Table 4: Aggregate test errors by resolution (6,756 samples per resolution).

N	Method	MAE	RMSE	MAPE
32	CNN	0.2813	0.3488	5.25%
	MC	0.7889	1.1435	13.95%
	Voxel	3.8983	4.1751	58.07%
64	CNN	0.4366	0.5495	8.73%
	MC	1.6503	2.8349	30.84%
	Voxel	5.2007	6.2153	83.67%
128	CNN	0.3784	0.5531	8.69%
	MC	3.7658	6.7164	71.56%
	Voxel	8.4251	11.6918	145.07%

7.2 CNN performance across resolutions

Table 4 summarises the CNN and classical methods across all three resolutions. The CNN obtains its lowest aggregate MAE at 32^3 (0.2813). Its MAE rises to 0.4366 at 64^3, then falls to 0.3784 at 128^3. The classical baselines worsen markedly as resolution increases under the selected synthetic boundary perturbation model. At 128^3, the Marching Cubes MAE is 3.7658, and the voxel-face MAE is 8.4251.

The CNN outperforms the two classical baselines at every tested resolution, but the CNN's own error does not decrease monotonically with finer sampling. The 64^3 model has the largest CNN MAE. The 128^3 model improves on 64^3 in MAE but not RMSE, indicating that its typical absolute error is lower while its larger errors remain consequential. A finer input grid does not automatically improve a learned model: it changes the input dimensionality, feature statistics, effective receptive field, batch size and optimisation conditions. Here, each

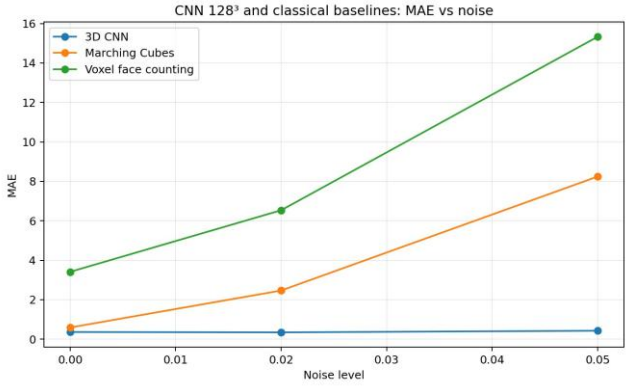


Figure 3: CNN MAE by synthetic boundary-noise level at 128^3. The model is least accurate at the highest perturbation level.

resolution was trained separately with the same broad architecture and a resolution-dependent batch size; no architecture scaling or multi-seed study was performed.

7.3 Noise dependence

At 32^3, CNN MAE increases moderately from 0.2718 at zero noise to 0.2957 at noise level 0.05. At 64^3, aggregate CNN MAE is almost unchanged across the three noise groups (approximately 0.435–0.438), while at 128^3 it increases from 0.3626 and 0.3451 to 0.4275 at the highest noise level. The response is therefore resolution-dependent rather than a single monotonic trend.

For the classical methods, the increase is much stronger, especially at higher resolutions. At 128^3, Marching Cubes MAE rises from 0.5905 at zero noise to 8.2411 at noise 0.05: voxel-face counting rises from 3.4128 to 15.3274. This is consistent with the construction of the synthetic perturbation: noise perturbs the implicit field before thresholding, creating additional small-scale boundary structure. The mesh and exposed-face estimators measure that perturbed boundary directly. This result should be interpreted as sensitivity to this particular boundary-noise model, not as a universal statement that Marching Cubes becomes less accurate at higher resolution.

7.4 Shape-family dependence

At 32^3, PointNet's family-wise MAE ranges from 0.0799 for ellipsoids to 0.2378 for GRF blobs. Its MAPE ranges from 1.84% for superellipsoids to 4.16% for cylinders. The CNN's family-wise MAE ranges from 0.2041 for GRF blobs to 0.5107 for spheres. The CNN's aggregate metric is strongly influenced by the GRF family, which constitutes two-thirds of all instances and two-thirds of the test variants. Accordingly, a single pooled score does not give equal weight to each Table 5: PointNet family-wise test performance at 32^3.

Family	Samples	MAE	MAPE
Cylinder	564	0.1651	4.16%
Ellipsoid	564	0.0799	2.32%
GRF blob	4,500	0.2378	2.65%
Sphere	564	0.1085	3.18%
Superellipsoid	564	0.0894	1.84%

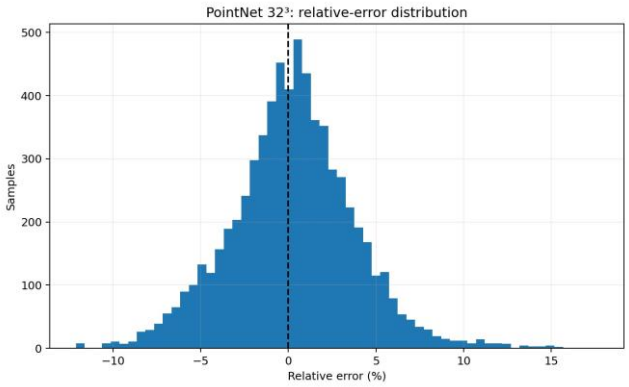


Figure 4: PointNet relative-error distribution at 32^3. The distribution is centred close to zero with a comparatively narrow spread.

geometric family.

The high GRF sample count does not imply that GRF is intrinsically the most difficult family. Its MAPE remains low, while some smaller families have larger relative errors. This may reflect the distribution of scales and shapes within each family, the amount of training data, and the learned representation. A balanced-family evaluation or reweighted training objective would be required to separate these effects more cleanly.

7.5 PointNet robustness to noise

PointNet's aggregate MAE at 32^3 is 0.1887, 0.1877 and 0.2096 for noise levels 0, 0.02 and 0.05 respectively. Its MAPE remains between 2.62% and 2.92%. The mean prediction ratio stays close to unity (0.9973, 0.9994 and 1.0106). These results suggest that the sampled point representation and PointNet regression are comparatively robust to the selected perturbation levels. However, the point clouds are generated from the perturbed volumes via Marching Cubes, and the fixed sample count may smooth out some local mesh complexity. It is not possible to attribute the performance solely to PointNet's architecture without further preprocessing ablations.

7.6 Training behaviour

The CNN training histories show that training MAE generally decreases, while validation MAE can fluctuate substantially between epochs. The best checkpoint is therefore selected by validation MAE rather than by the final epoch. For the 32^3 CNN, the best validation MAE was 0.2854 at epoch 3 and early stopping occurred at epoch 10. At 64^3, the best validation MAE was 0.4332 at epoch 2 and early stopping occurred at epoch 9. The 128^3 model was trained with batch size 2 and required substantially longer per epoch. These fluctuations are consistent with small, correlated minibatches and batch-normalisation statistics, although the present experiments do not isolate the cause.

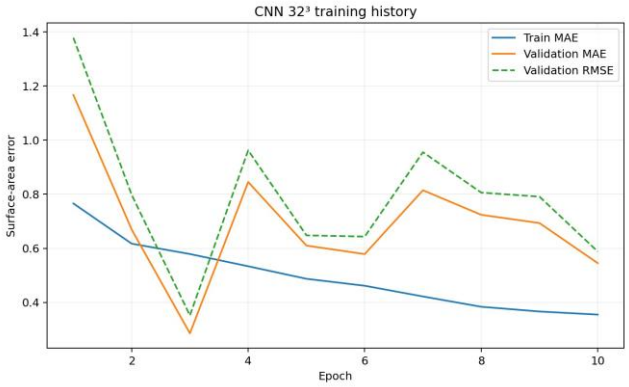


Figure 5: Training and validation behaviour of the CNN at 32^3. Validation performance is non-monotonic, motivating checkpoint selection and early stopping.

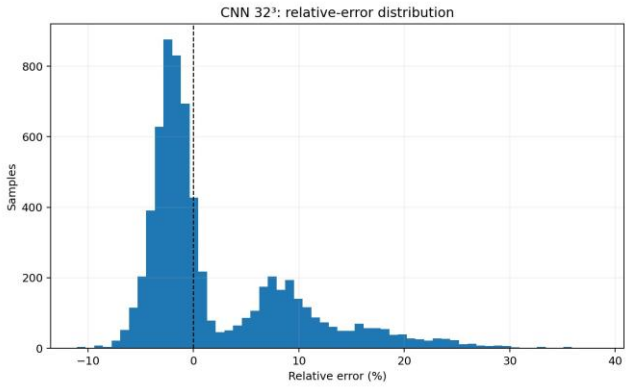


Figure 6: CNN relative-error distribution at 32^3.

7.7 Resolution-specific error distributions

The aggregate metrics conceal differences in the tails of the error distributions. Figure 6–8 show the relative error distributions for the CNN at each resolution. The 32^3 distribution is comparatively concentrated, consistent with its lowest MAE and RMSE. The 64^3 and 128^3 distributions exhibit wider tails. This is important because RMSE penalises occasional large deviations more strongly than MAE. In particular, the 128^3 CNN has a lower MAE than the 64^3 CNN but a slightly higher RMSE (0.5531 versus 0.5495), suggesting that a subset of samples still produces comparatively large errors.

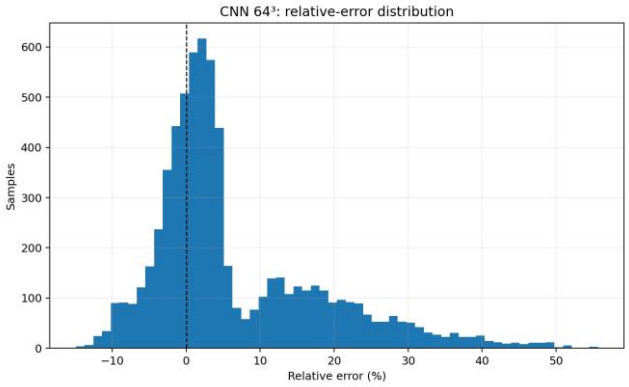


Figure 7: CNN relative-error distribution at 64^3.

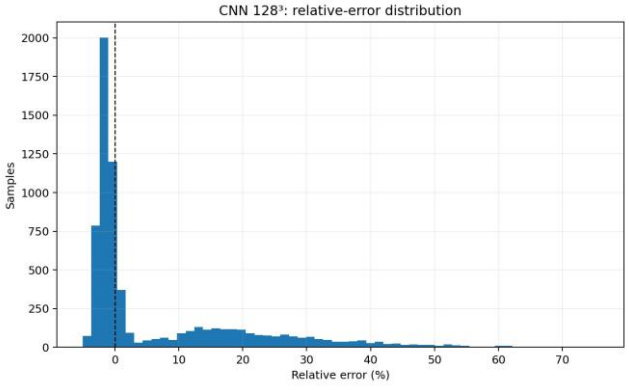


Figure 8: CNN relative-error distribution at 128^3.

7.8 Family-by-resolution observations

The CNN's family-level errors vary substantially. At 32^3, GRF blobs have MAE 0.2041 and MAPE 2.27%, while sphere MAE is 0.5107 and MAPE 13.44%. At 64^3, GRF MAE is 0.2726 (MAPE 3.02%), while ellipsoid MAE reaches 0.8674 (MAPE 23.59%). At 128^3, GRF MAE falls to 0.1354 (MAPE 1.57%), whereas cylinder MAE rises to 1.0379 (MAPE 28.14%). The ranking of shape families is therefore not stable across resolutions. This behaviour argues against a single simple explanation such as "irregular shapes are harder" or "regular shapes are easier." The observed error is affected by family-specific geometry, sample count, parameter ranges, noise response and the interaction between those factors and the model's effective receptive field.

7.9 Noise-stratified classical performance

The response of the classical estimators is summarised in Table 6. For each resolution, the clean-input Marching Cubes error is considerably lower than its error at noise 0.05. At 32^3, for example, MC MAE rises from 0.4426to1.3203. At128^3itrisesfrom0.5905to8.2411. Voxel-face counting shows a similar but larger effect.

The trend is consistent with the geometry of the perturbation process: finer grids contain more boundary samples at which a fixed field perturbation can alter the threshold occupancy. This mechanism is specific to the noise definition used here and should not be generalised to all imaging noise processes.

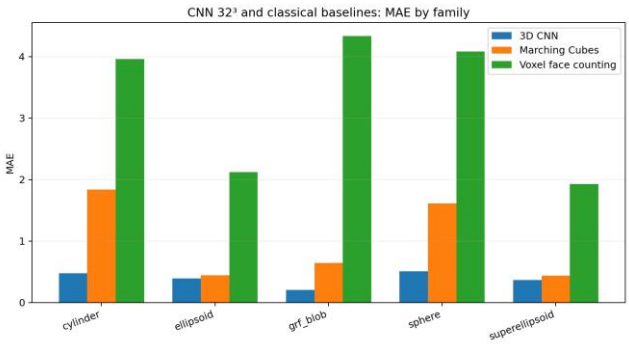


Figure 9: CNN MAE by shape family at 32^3.

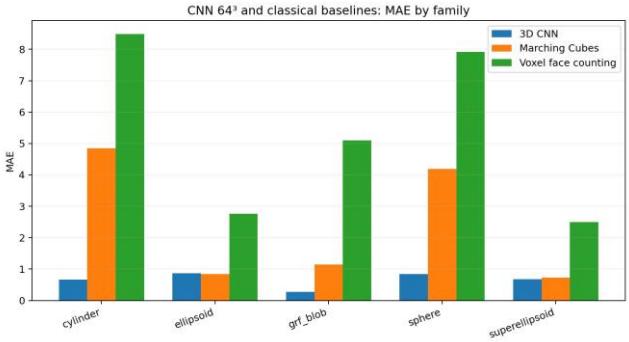


Figure 10: CNN MAE by shape family at 64^3.

7.10 PointNet versus CNN at the family level

PointNet outperforms the CNN in pooled MAE and MAPE at 32^3, but family-level results show that the gain is not uniform in absolute terms. PointNet's MAE is lower than the CNN's for every family in this matched test set. For GRF blobs, the difference is modest in absolute units (0.2378 versus 0.2041 is a CNN advantage), while PointNet is substantially better for ellipsoids, spheres, cylinders and superellipsoids. In relative terms, the GRF family has many more samples and contributes strongly to the pooled score. The comparison therefore benefits from reporting both samples' weighted overall metrics and family-specific metrics. It would be misleading to say PointNet wins every family: the CNN's GRF MAE is lower (0.2041 versus 0.2378), although PointNet's GRF MAPE is only slightly higher (2.65% versus 2.27%).

7.11 Interpretation of the model representations

A voxel CNN sees the full volume, including interior voxels that are not themselves part of the boundary. Con-

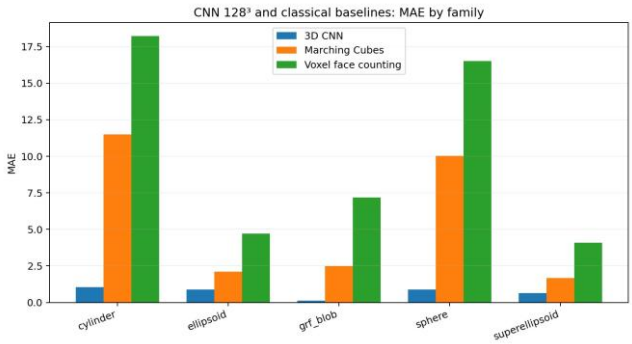


Figure 11: CNN MAE by shape family at 128^3.

Table 6: Classical baseline MAE by resolution and noise.

Resolution	Noise	MC MAE	Voxel MAE
32	0.00	0.4426	3.2749
32	0.02	0.6039	3.6254
32	0.05	1.3203	4.7946
64	0.00	0.5555	3.3827

64	0.02	1.1410	4.4405
64	0.05	3.2546	7.7790
128	0.00	0.5905	3.4128
128	0.02	2.4658	6.5350
128	0.05	8.2411	15.3274

volution and pooling can nevertheless learn volumetric cues correlated with surface geometry, such as object extent, local curvature patterns and boundary-to-volume relationships. Global average pooling compresses the final feature tensor into a fixed-length descriptor. This choice supports a common fully connected head across input sizes, but it also discards explicit spatial location and may dilute sparse boundary activations. Because surface area is an extensive quantity, an architecture with global average pooling is not guaranteed to preserve scale information optimally; the present experiment evaluates this design empirically but does not compare it with global max pooling, sum pooling or explicit coordinate/scale features.

PointNet sees only the sampled boundary points. Area-weighted triangle selection makes the sampling density proportional to mesh area, while uniform sampling within each selected triangle avoids concentrating points at mesh vertices. Centring removes translation without destroying absolute scale. The point cloud thus contains direct boundary geometry but not the original voxel occupancy field. Its symmetric aggregation makes the output invariant to point order, an appropriate property for an unordered point set. A limitation is that fixed-size sampling discards some mesh detail and introduces sampling variability. Repeated point-cloud sampling and point-count ablations would be required Table 7: CNN family-wise MAE at each resolution.

Family	32^3	64^3	128^3
Cylinder	0.4753	0.6617	1.0379
Ellipsoid	0.3910	0.8674	0.8968
GRF blob	0.2041	0.2726	0.1354
Sphere	0.5107	0.8435	0.8837
Superellipsoid	0.3647	0.6818	0.6336

to quantify this source of variance.

7.12 What the experiment establishes

The evidence supports three bounded conclusions. First, direct exposed-face counting is a weak estimator for the curved surfaces and perturbation levels represented in this dataset. Second, the learned estimators substantially reduce test error relative to the two classical methods under the same synthetic data distribution. Third, PointNet is the strongest tested pipeline at 32^3, but its advantage cannot be attributed to architecture alone because it has more parameters and uses a Marching Cubes preprocessing stage. The study does not establish that PointNet will outperform a carefully tuned CNN on real volumes, nor that the learned estimators recover the exact continuous surface area.

7.13 Cross-resolution family summary

Table 7 gives the CNN MAE by family and resolution. The values demonstrate that the aggregate resolution trend is not shared uniformly by all families. For GRF blobs, MAE improves from 0.2726 at 64^3 to 0.1354 at 128^3. For cylinders, it increases from 0.6617 to 1.0379. The ellipsoid and sphere errors remain high at both finer resolutions. This family-dependent response is consistent with a model whose learned features interact with shape geometry and scale, rather than a simple estimator whose error is controlled only by voxel spacing.

7.14 PointNet noise-stratified metrics

The PointNet error is relatively stable across the three noise groups. Table 8 shows that MAE changes by only about 0.022 area units between clean and highest noise samples. MAPE remains below 3% in all three groups. Bias changes sign: the model slightly underestimates clean and moderately perturbed examples, then slightly overestimates at noise 0.05. This indicates that the near-zero overall bias is not simply a consequence of a constant correction, but results from small, signed errors across the test distribution.

Table 8: PointNet performance by noise at 32^3.

Noise	MAE	RMSE	MAPE	Bias
0.00	0.1887	0.2637	2.64%	−0.0367
0.02	0.1877	0.2610	2.62%	−0.0187
0.05	0.2096	0.2795	2.92%	0.0657

7.15 Why aggregate scores need qualification

The test split preserves the family proportions of the complete dataset. As a result, the 4,500 GRF test variants at each resolution outweigh the 564 variants contributed by each of the other four families. The overall MAE is a sample-weighted mean of absolute

errors and therefore gives GRF blobs a larger influence. A macro-average of the five family MAEs would answer a different question: expected performance when each shape family is given equal importance. Both should be reported in future benchmark releases. The same consideration applies to the training objective, which minimises per-sample regression error and therefore allocates more gradient updates to the majority family.

The area distribution is also not independent of family. For example, a family may occupy a narrower range of surface areas than another. A trend between error and area magnitude may therefore partly reflect which shape families occupy that range. A more rigorous analysis of scale dependence would generate overlapping area distributions across all shape families or stratify jointly by family and area quantile. No independent causal claim about area scale is made in this study.

8 Discussion

The strongest result is the improvement achieved by learned regression over direct geometric baselines on this synthetic benchmark. At 32^3, PointNet achieves an MAE approximately four times lower than Marching Cubes and twenty times lower than voxel-face counting. The CNN also substantially outperforms the classical methods. The low aggregate bias of both learned estimators indicates that they are not merely applying a fixed global correction to the classical area estimate.

One plausible explanation is that the learned models capture statistical regularities of the generated shape families. Direct estimators respond to every geometric irregularity in the observed volume. A regressor trained against the reference area may learn to treat some voxel scale boundary fluctuations as perturbations rather than as meaningful new surface features. The present study does not prove that the network explicitly performs geometric smoothing; it establishes only the observed reduction in error under the defined data distribution.

PointNet outperforms the CNN in the matched 32^3 comparison. This is notable because the point cloud has a fixed number of 2,048 samples, whereas the CNN processes the complete voxel volume. Thepointrepresentationmayfocuscomputationonther econstructedboundary and discard redundant interior/background voxels. Nevertheless, PointNet has approximately 1.55 million parameters compared with approximately 0.30 million for the CNN, and it receives a mesh-derived point cloud. The comparison is therefore not a parameter-matched architecture ablation. It is a comparison of two complete representation-and-regression pipelines.

The resolution experiment reveals an important trade-off. Classical estimators deteriorate as the grid is refined under the applied perturbation model, while the CNN remains substantially more stable. The CNN's own best aggregate performance occurs at the coarsest resolution, and the 64^3 model is less accurate than both the 32^3 and 128^3 models. The current architecture, batch size schedule and training protocol were not optimised separately for each resolution. Consequently, the results should not be interpreted as evidence that coarse voxelisation is always preferable. Instead, they show that increased input resolution alone does not guarantee lower prediction error.

The family-wise analysis is also essential. Because GRF blobs account for 10,000 of the 15,000 instances, pooled test metrics are dominated by that family. The low GRF MAPE can make the overall metric appear stronger than performance on the smaller regular-shape families. Reporting per-family errors and equal-family macro-averages would be preferable in future work. Moreover, area bins and shape families are not independent in this generated dataset, so apparent trends with area magnitude may be confounded by family composition.

9 Conclusion

This work developed and evaluated a synthetic benchmark for estimating the surface area of voxelised 3D objects. The dataset contains 15,000 underlying instances and 135,000 variants across five shape families, three resolutions and three levels of synthetic boundary perturbation. Two classical estimators—exposed-voxel face counting and Marching Cubes—were compared with a 3D CNN at all resolutions, and PointNet was evaluated at 32^3.

The principal objectives were met. The dataset was generated and audited; reference areas were assigned analytically or by a high-resolution mesh procedure; all variants of each underlying instance were kept within a single split; and the learned models were evaluated against classical baselines on matched test samples. At 32^3, PointNet achieved MAE 0.1953 and MAPE 2.73%, while the CNN achieved MAE 0.2813 and MAPE 5.25%. Both were substantially more accurate than Marching

Cubes and voxel-face counting. The CNN also outperformed the classical baselines at 64^3 and 128^3.

The project therefore demonstrates that data-driven regression can provide a more accurate estimate of reference surface area than direct voxel geometry for the synthetic shape distribution studied. It does not yet establish generalisation to real CT volumes or arbitrary unseen shape classes. The result is best regarded as a proof of concept and a controlled comparison of geometric representations.

10 Limitations and Future Work

1. **Synthetic domain only.** The dataset contains procedurally generated shapes, not real CT or microCT segmentations. Scanner point-spread functions, anisotropic voxels, segmentation errors, partial volume effects and complex topology are not represented.
2. **Reference-area uncertainty.** Ellipsoid, superellipsoid and GRF labels use a resolution-256 Marching Cubes reference. This is a numerical reference, not an exact continuous area. Convergence was checked at selected resolutions, but a formal error bound for every instance is unavailable.
3. **Noise model.** The perturbation is applied to the implicit field before thresholding. It should not be equated with realistic image-intensity noise or segmentation uncertainty.
4. **Family imbalance.** GRF blobs comprise two-thirds of the dataset. Pooled scores are therefore sample weighted towards irregular blobs. Equal-family sampling and macro-averaged metrics are needed.
5. **Architecture and optimisation.** One CNN architecture was used at all resolutions, with different batch sizes. The effective receptive field, batch normalisation statistics and optimisation dynamics vary with resolution. Multiple random seeds and resolution-specific architecture tuning are required.
6. **PointNet preprocessing.** Point clouds are sampled from Marching Cubes meshes. PointNet therefore depends on mesh extraction and is not a fully independent alternative to the mesh pipeline. The 2,048point budget was not selected through an ablation study.
7. **Single-seed results.** The reported experiments use seed 42. No multi-seed variance estimates are available, so small differences should not be treated as statistically conclusive.

Future work should include balanced family sampling, multi-seed training, resolution-specific or multiresolution CNN architectures, point-count and sampling ablations, anisotropic voxel spacing, more realistic imaging/noise models, and evaluation on real segmented volumes. A particularly useful extension would compare against mesh references generated at substantially higher resolutions with a documented convergence tolerance, while independently validating a subset of shapes using analytic or alternative numerical methods. Runtime and memory comparisons should also be performed under a common hardware and preprocessing protocol.

11 Reproducibility and Computational Notes

The dataset-generation and experiment workflows were implemented in Python notebooks. Dataset generation used NumPy, SciPy, scikit-image and pandas; the learning pipelines used PyTorch, with Matplotlib for plotting and pandas for metric export. The random seed was set to 42 for the principal generation/training workflow. Volumes were stored in compressed shards, while metadata were stored separately and linked by an instance identifier. Each metadata record identifies the family, split, seed, resolution, noise level, reference area, reference method and voxel spacing. The complete metadata table contains 135,000 rows and 12 columns.

The data audit verified 15,000 unique underlying instances, nine variants per instance, no cross-split instance duplication, 1,878 volume shards and 1,878 metadata shards. All 1,878 volume shards were successfully read during validation. The packed arrays were stored as unsigned 8-bit values and unpacked to binary volumes for training. This organisation reduces storage overhead while preserving the ability to load a selected resolution and variant. It also allows the split to be checked independently of the model code.

The test set for each resolution contains 6,756 samples. The PointNet test comparison uses the same set of instance/resolution/noise keys as the CNN and classical baselines. This matched design is important: comparing metrics computed on different test subsets could otherwise confound model performance with sample difficulty. The CNN models were trained separately at each resolution; PointNet was trained

only at 32^3. The training durations differ considerably because 3D convolutional cost and activation memory grow rapidly with spatial resolution. The 128^3 experiment required a small batch and substantially longer per-epoch time than the 32^3 experiment.

For reproducibility, the model checkpoints, training histories, raw test predictions, matched comparison tables and plots were exported. The training code and data-generation notebook form part of the project deliverables. The exact software versions and accelerator details should be added to the final camera-ready version from the saved notebook environment metadata; the current experiment used a Kaggle GPU runtime. Hardware-dependent wall-clock timings are not reported as a central result because the PointNet pipeline includes mesh extraction and point-cloud caching, and a fair runtime study would need to measure preprocessing, inference and memory under a common protocol.

12 Additional Results: Error Metrics and Reporting Choices

MAE is used as the principal scalar metric because it is directly interpretable in the area units and is less dominated by outliers than RMSE. RMSE is retained because large errors matter in downstream geometric calculations and because the relative behaviour of MAE and RMSE can reveal heavy-tailed residuals. MAPE is reported to compare errors across shapes of different areas, but it can become unstable for very small reference areas; the generated dataset has positive nonzero areas, and a small numerical floor is used in the implementation to avoid division by zero. Mean bias distinguishes overestimation from underestimation, which MAE alone cannot show. The coefficient of determination provides a variance-normalised comparison but can be negative when an estimator is substantially worse than the constant-mean predictor.

The evaluation is performed on the held-out test partition only. Validation MAE is used for checkpoint selection and early stopping, not for the final performance table. The test partition is not used to select hyperparameters. However, the model architecture, point count, noise levels and resolution choices were specified as part of the project design rather than optimised through a broad search. The absence of a large hyperparameter search reduces the risk of overfitting the test set, but it also means that the reported models should be interpreted as representative implementations rather than best possible models.

A further distinction concerns the reference labels. For analytic families, the target is computed from the known generating parameters. For non-analytic families, the target is a high-resolution mesh estimate. Consequently, the model is learning to reproduce a mixture of analytic and numerical reference conventions. This is acceptable for a controlled benchmark provided it is stated explicitly, but it means that absolute error is measured against the selected reference procedure rather than against an experimentally measured continuous surface.

Acknowledgment

The authors acknowledge the computational resources provided by Kaggle notebooks and the open-source scientific Python and PyTorch ecosystems used in this work.

Digital Resource: https://youtu.be/lkUFB1lQ9W8

References

[1] W. E. Lorensen and H. E. Cline, “Marching Cubes: A high-resolution 3D surface construction algorithm,” *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[2] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep learning on pointsetsfor3Dclassification and segmentation,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 652–660.

[3] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep hierarchical feature learning on point sets in a metric space,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[4] D. Maturana and S. Scherer, “VoxelNet: A 3D convolutional neural network for real-time object recognition,” in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 922–928.

[5] T. S. Newman and H. Yi, “A survey of the marching cubes algorithm,” *Computers & Graphics*, vol. 30, no. 5, pp. 854–879, 2006.