Unsupervised Dual Transformer Learning for 3-D Textured Surface Segmentation

Iyyakutti Iyappan Ganapathi, *Member, IEEE*, Fayaz Ali Dharejo, *Member, IEEE*, Sajid Javed, Syed Sadaf Ali, and Naoufel Werghi, *Senior Member, IEEE*

***Abstract*— Analysis of the 3-D texture is indispensable for various tasks, such as retrieval, segmentation, classification, and inspection of sculptures, knit fabrics, and biological tissues. A *3-D texture* represents a locally repeated surface variation (SV) that is independent of the overall shape of the surface and can be determined using the local neighborhood and its characteristics. Existing methods mostly employ computer vision techniques that analyze a 3-D mesh globally, derive features, and then utilize them for classification or retrieval tasks. While several traditional and learning-based methods have been proposed in the literature, only a few have addressed 3-D texture analysis, and none have considered unsupervised schemes so far. This article proposes an original framework for the unsupervised segmentation of 3-D texture on the mesh manifold. The problem is approached as a binary surface segmentation task, where the mesh surface is partitioned into textured and nontextured regions without prior annotation. The proposed method comprises a mutual transformer-based system consisting of a label generator (LG) and a label cleaner (LC). Both models take geometric image representations of the surface mesh facets and label them as texture or nontexture using an iterative mutual learning scheme. Extensive experiments on three publicly available datasets with diverse texture patterns demonstrate that the proposed framework outperforms standard and state-of-the-art unsupervised techniques and performs reasonably well compared to supervised methods.**

***Index Terms*— 3-D surface, segmentation, texture, transformers, unsupervised.**

I. Introduction

With the widespread use of 3-D cameras and scanning devices to capture the rich geometrical properties of object surfaces, many computer vision-based interdisciplinary applications have emerged in recent years. A large volume of work has been addressing the problem of segmenting, classifying, and retrieving 3-D shapes based on their similarities using triangle mesh and point clouds as input [1], [2], [3], [4], [5], [6], [7]. However, the segmentation and classification of 3-D geometric textures (or, simply, 3-D textures) are less explored. Unlike shape, 3-D texture is a surface feature characterized by repetitive geometric, regular, or random patterns on the surface. These patterns can be considered geometric corrugations of the surface that alter its local appearance without affecting its global shape. There is a diverse range of surfaces exhibiting 3-D texture, including knit fabrics, artwork patterns, artist styles, and natural structures such as tree barks [8], [9]. Texture-based applications can benefit various industries, including remote sensing, 3-D content creation, and animation [10]. One of the most important uses is in cultural preservation, where researchers have studied and developed methods to retrieve and categorize cultural objects based on texture [11], [12], [13]. Recent progress in this field has shown remarkable performance in transforming historical buildings into semantically structured 3-D models, enabling enhanced detection and comprehension of heritage structures [14].

All the 3-D texture classification and segmentation methods developed so far have relied on supervised schemes that require demanding manual annotation of a large amount of data. Manual annotation of textured regions on 3-D surfaces is even more tedious than its counterpart in 2-D images, as it requires repeating the procedure over multiple views. Also, the manual annotation is susceptible to error because the annotator operates on a 2-D projection of the surface.

In this article, we present an original framework for the unsupervised segmentation of the 3-D texture segmentation on the mesh manifold. The problem is approached as a fully unsupervised binary surface segmentation where the mesh surface is partitioned into textured and nontextured regions (see examples in Fig. 1). This novel scheme eliminates labor-intensive labeling while achieving comparable segmentation performance to supervised methods. The behavior of autoencoder models during our attempts to reconstruct surface patches served as inspiration for our strategy. We found that the reconstruction error for textured patches (which are heterogeneous) was typically greater than for their nontextured counterparts (which are homogeneous or smooth patches). In Fig. 2, we present the distribution of reconstruction loss for texture and nontexture patches collected from different surfaces. The heterogeneity of textured surfaces, which exhibit a greater degree of entropy compared to the homogeneous nontextured patches. Based on these observations, we hypothesize that this difference in behavior could be amplified and utilized more effectively through a cleaner learning mechanism in an adversarial scheme for fully unsupervised classification of surface patches.

Manuscript received 19 April 2023; revised 23 November 2023; accepted 28 January 2024. The work of Naoufel Werghi was supported in part by the research grant from ASPIRE Award for Research Excellence under Grant AARE20-279 and in part by the research grant from Khalifa University under Grant CIRA-2021-052. *(Corresponding author: Iyyakutti Iyappan Ganapathi.)*

Iyyakutti Iyappan Ganapathi, Syed Sadaf Ali, and Naoufel Werghi are with the C2PS and the Department of Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates (e-mail: iyyakutti.ganapathi@ku.ac.ae; syed.ali@ku.ac.ae; naoufel.werghi@ku.ac.ae).

Fayaz Ali Dharejo is with the Department of Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates, and also with the Computer Vision Laboratory, CAIDAS, University of Würzburg, 97074 Würzburg, Germany (e-mail: fayaz.ali@ku.ac.ae).

Sajid Javed is with the Department of Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates (e-mail: sajid.javed@ku.ac.ae).

Digital Object Identifier 10.1109/TNNLS.2024.3365515

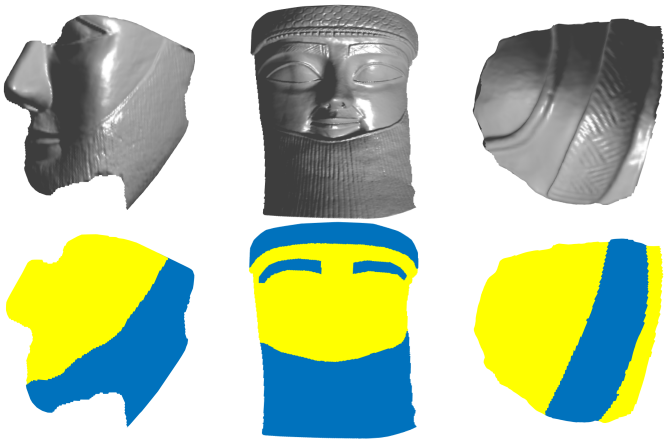


Fig. 1. Samples of 3-D surfaces exhibiting diverse texture patterns are shown. The top row displays cultural heritage artifacts, and the bottom row distinguishes nontextured areas in yellow and textured areas in blue.

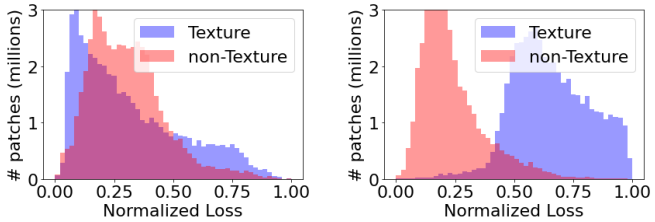


Fig. 2. Distributions of reconstruction loss for texture and nontexture patches. (a) Losses obtained early in the process show that there is a significant overlap between the distributions of texture and nontexture patches, resulting in a high misclassification error, whereas (b) losses obtained near the end of the process show that there is a noticeable separation between the distributions of texture and nontexture patches, resulting in a lower misclassification error.

The proposed model consists of two main components: a label generator (LG) and a label cleaner (LC). The generator is trained to reconstruct surface patch features, and the reconstruction loss function is utilized to assign labels to the patches. Patches with low loss are labeled as texture, while those with high loss are labeled as nontexture. This set of pseudo-labels is expected to contain several misclassified patches, and thus there is a need for further segregation. To address this, we introduce a discriminative learning mechanism that involves training a binary classifier with the pseudo-labeled patches. The classifier is then used to reclassify the patches in the second stage, correcting the initial assignments. For instance, a patch labeled as textured initially could be reclassified as nontextured, and vice versa, as the classifier training is not expected to be entirely accurate. The modified set of pseudo-labeled patches is then utilized in the next iteration to enhance the generator further. By repeating this procedure iteratively, the pseudo-LG and pseudo-LC modules learn from each other and enhance the overall surface patch classification performance.

The proposed framework outperforms the classical unsupervised approaches and baseline methods on three datasets: *KU 3DTexture* [15], *SHREC'18* [16], and *SHREC'17* [17]. In summary, our original contributions are summarized as follows.

1) We propose leveraging the surface patch reconstruction error as an underlying concept for classifying textured and nontextured patches.
2) We present a fully unsupervised mutual transformer learning approach for 3-D texture segmentation on mesh surfaces. To the best of our knowledge, this is the first attempt at facet-level texture segmentation.
3) We validated the proposed framework for texture segmentation on three datasets with complex texture patterns and varying resolutions, achieving significantly better results than conventional clustering and baseline approaches.

II. Related Work

As a recent topic, there is not yet a large volume of work on 3-D texture analysis. Nevertheless, the existing research can be classified into three main categories: 3-D texture classification [18], [19], [20], 3-D texture retrieval [17], [21], [22], and 3-D texture segmentation [23], [24], [25].

In the 2-D image domain, local descriptors like Gabor or local binary pattern (LBP) [26] are often used to define texture patterns based on repeatability, randomness, and orientation [27], [28]. However, in the 3-D domain, texture analysis is still in its early stages. In the realm of classification, Werghi et al. [19], [29] introduced the concept of 3-D texture by proposing mesh-LBP, an extension of LBPs to the mesh manifold, which uses a structure of local ordered rings to classify textured patterns on mesh surfaces. They later extended it to other applications, such as 3-D face recognition, in subsequent work [18], [20], [30]. The shape retrieval community found Werghi et al.'s 3-D texture concept intriguing, which prompted them to publish a number of 3-D relief pattern datasets in the SHREC contests [17]. Moscoso et al. [21] further contributed to this field by introducing the edge-LBP descriptor, which uses contours defined based on a sphere-mesh intersection. They applied this representation to match archeological fragments using the Battacharya distance as a metric [22]. Thompson et al. [21] later presented various techniques, all focused on identifying the best representation for characterizing 3-D texture patterns and related similarity metrics.

Liu et al. [23] introduced a supervised snake-based segmentation method, requiring manual selection of snake contours that evolve to distinguish smooth surfaces and relief patterns. Zatzarinni et al. [24] approached similar issues analytically using a height function over the surface, tailored for relief patterns. However, these methods are specific to identifying protrusions on the main surface and cannot be extended to the broader context of 3-D texture. Tortorici et al. [25] proposed a recent approach using convolution tools on the mesh for weakly supervised texture feature extraction, employing random forest. Additionally, a mesh convolution with spherical harmonics as orthonormal bases for 3-D meshes is presented, but identifying small variations on 3-D surfaces remains challenging [31]. Further, spectral descriptors, a popular texture analysis category, offer resilience to 3-D object transformations by leveraging the Laplace–Beltrami operator [32], [33]. Despite its real-time robustness, the complexity of this approach scales with the number of vertices in the input mesh. Simplifying complexity by ensuring a consistent vertex count in 3-D input samples may lead to information loss, impacting texture recognition performance.

Choi et al. [34] introduced a semantic segmentation method, utilizing FC-DenseNet to extract 3-D scripts from rough surfaces, with training based on feature images from local shape features. Similarly, generalized 3-D semantic segmentation and classification networks are found in the literature, with GNN-based approaches being well-suited for holistic 3-D surface tasks [5], [6], [13]. However, GNN struggles to incorporate minor surface variations (SVs) due to the insensitivity of node proximity to deformation. Further, recent advancements in point cloud understanding include the development of novel approaches such as the next iteration of PointNets, referred to as PointNeXt [35], learning point-level representations through various aggregations [36], and the introduction of a universal point set operator for point clouds known as PointMixer [37]. These networks effectively address challenges associated with the inherent sparsity, unordered nature, and irregularity of point clouds, showcasing high efficiency in part-based segmentation and classification tasks. However, there remains a need to address the tracking of SVs, particularly when traversing the 3-D surface, to examine intricate surface patterns.

III. Problem Definition

The following outlines the objectives of a proposed learning model:

1) *Input:* A 3-D surface texture $\mathcal{M}$.
2) *Output:* $\mathcal{L} = f(\mathcal{M})$, a proposed learning model that maps $\mathcal{M}$ to $\mathcal{L}$, where $\mathcal{M}$ is the input surface texture and $\mathcal{L}$ is the facet labels.
3) *Objective:* To minimize the texture and nontexture facet-level classification errors.

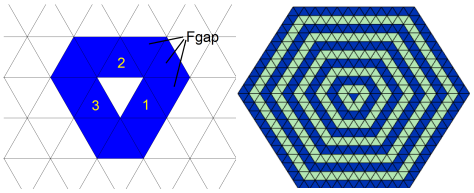
IV. Preliminaries

A 3-D texture pattern is a manifold embedded in 3-D Euclidean space with 3-D points $P(u, v) = x(u, v), y(u, v), z(u, v)$ where x, y, z are the coordinates in 3-D space and u, v are the independent variables that correspond to the manifold dimension. The following preliminaries provide an overview of the terminologies used in this proposed research work.

Definition 4.1: A mesh $\mathcal{M} = \{\mathcal{V}, \mathcal{E}, \mathcal{F}\}$ is a polygonal representation of a surface where $\mathcal{V}$ is a set of vertices $\{x, y, z\} \in \mathcal{R}^3$, $\mathcal{E}$ is a set of edges connecting neighboring vertices pair and $\mathcal{F}$ is the set of faces, that is, polygons connected with edges and vertices from $(\mathcal{E}, \mathcal{F})$.

Definition 4.2: Curvature (Cur) is a geometric property that is frequently utilized in 3-D surface analysis. It is defined by the intersection of curves with normal planes in different orientations t. For example, k_t is the curvature defined by (P_0, t, n), where P_0 is the plane, t is the direction, and n is the normal to the surface. The curvature at any point can be computed as a combination of two curvatures, k_{t_1} and k_{t_2}, maximum and minimum, in two principal directions, t_1 and t_2.

Definition 4.3: Ordered ring facets (ORF) is computed at each facet by employing adjacent faces {1, 2, 3} and other faces, Fgap, as shown in Fig. 3(a). To normalize the starting position of a facet in a ring, we reorder it so that the first facet in each ring is closest to the centroid of the rings. A regular mesh with $R = r_1, r_2, \ldots, r_n$ ordered rings, where r_1 represents the first ring with 12 facets, r_2 represents the second ring with 24 facets, and r_n represents the nth ring with $n \cdot 12$ facets. The facets in each ring are described using the proposed features, which aid in describing the texture of a 3-D mesh. Fig. 3(a) and (b) depicts an illustration of one ring and ten rings generated on a 3-D mesh surface.



(a) (b)

Fig. 3. ORF. (a) Ring constructed at a facet using the adjacent facets {1, 2, 3} and the Fgap facets. (b) Ten rings constructed similarly using the adjacent and Fgap facets.

Algorithm 1 Pseudocode for LD and SV

Input: vertices, facets, RingList (R)
$R = r_1,\ r_2,\ \cdots, r_n$
$v_1,\ v_2,\ \cdots v_n \leftarrow f_1,\ f_2,\ \cdots f_n \leftarrow R$
$C \leftarrow GetCenter(v_1, v_2,\ \cdots v_n)$
$\hat{C} \leftarrow mean(C)$
$H \leftarrow \hat{C}' * \hat{C}$
$[\Lambda, V] \leftarrow \text{eig}(H)$
$\lambda_1 \leftarrow \Lambda_1, \lambda_2 \leftarrow \Lambda_2, \lambda_3 \leftarrow \Lambda_3$
$v_1 \leftarrow V_1, v_2 \leftarrow V_2, v_3 \leftarrow V_3$
normal $\leftarrow v_1$
if $sign([0\ 0\ 1] * \text{normal}) < 0$ **then**
  normal = -normal
end if
Construct a plane Ax + By + Cz + D=0 using C and normal
LocalDepth $\leftarrow d = |Ax_0 + By_0 + Cz_0 + D|/\sqrt{(A^2 + B^2 + C^2)}$
SurfaceVariation $\leftarrow \frac{\lambda_1}{\lambda_1+\lambda_2+\lambda_3}$

Definition 4.4: Local depth (LD) is computed using the ORF where the neighbors' vertices C are extracted from the facets, and then a covariance matrix $H = \hat{C}^T * \hat{C}$ is computed, where $\hat{C} = C - \bar{C}$ and $\bar{C}$ is the mean of vertices C. Further, eigenvalues and eigenvectors are obtained by decomposing H, where the eigenvector of the smallest eigenvalue is chosen as a normal. A plane is then constructed using the obtained normal to find the LD of any point by computing the distance between the point and the plane. Algorithm 1 provides pseudocode for implementation.

Definition 4.5: SV is computed using the eigenvalues $\lambda_1 < \lambda_2 < \lambda_3$, obtained from the decomposition of H.

Definition 4.6: Shape Index (SI) used to quantify the curvature at a point P is given by

$$\text{SI}(P) = \frac{1}{2} - \frac{1}{\pi} \arctan\left(\frac{k_1(P) + k_2(P)}{k_1(P) - k_2(P)}\right)$$

where k_1 and k_2 represent the maximum and minimum principal curvatures, respectively, with the condition that $k_1 > k_2$ is satisfied for all points P. The expressions for k_1 and k_2 are defined as follows:

$$k_1(P) = H(P) + \sqrt{H^2(P) - K(P)}$$
$$k_2(P) = H(P) - \sqrt{H^2(P) - K(P)}.$$

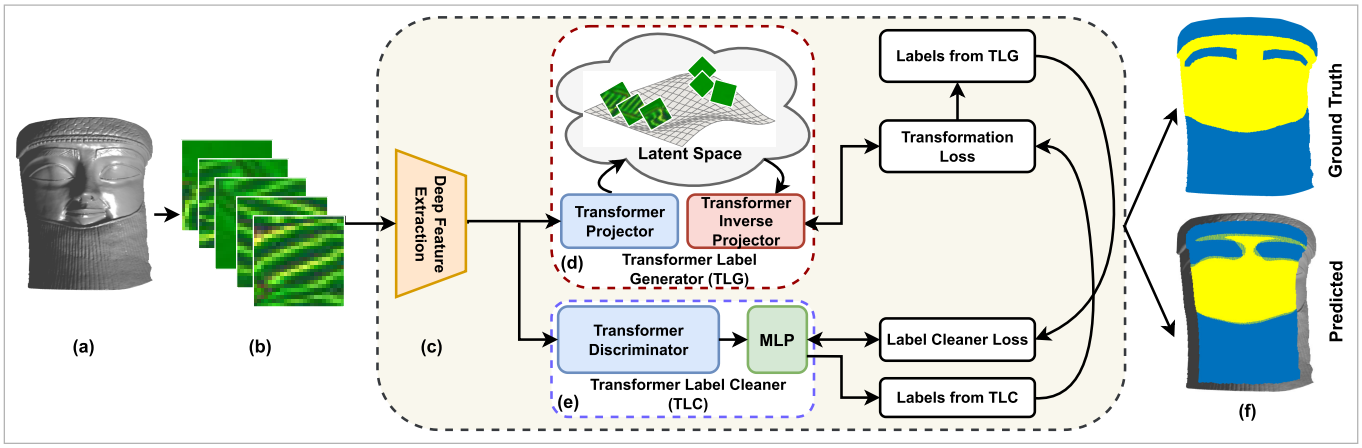


Fig. 4. Outline of the proposed surface patch classification for texture segmentation. (a) 3-D surface and (b) 2-D surface patch images computed across the mesh triangle facets using geometric features (see Fig. 5). (c) Deep feature extraction from the surface patch images. (d) LG inputs deep features and assigns a pseudo-label (texture or nontexture) to each surface patch. Noticeably, this assignment produces misclassified surface patches (i.e., noisy labels). (e) LC cleans the noisy pseudo-labels generated in (d) repeated over several iterations. (f) Ground truth and the predicted results, where yellow and blue represent the nontexture and texture regions, respectively.

Algorithm 2 Calculate AZ and EL

Input: vertex, face, RingList (R)

$R \leftarrow \{r_1, r_2, \ldots, r_n\}$ {(R contains n rings, each with multiple facets)}

$v_1, v_2, \ldots, v_n \leftarrow f_1, f_2, \ldots, f_n \leftarrow R$ {(Obtain facets and vertices from R)}

normal(x, y, z) $\leftarrow$ computeNormal$(vertex, face)$ {(Compute normal)}

Azimuth $\leftarrow$ atan2(y, x) {(Calculate Azimuth)}

Elevation $\leftarrow$ atan2$(z, \sqrt{x^2 + y^2})$ {(Calculate Elevation)}

Here, $H(P)$ and $K(P)$ denote the mean and Gaussian curvatures at point P.

Definition 4.7: Azimuth (AZ) and *Elevation* (EL) are defined as follows.

1) The AZ angle represents the horizontal angle in a polar coordinate system, measured in degrees or radians. In our study, it is used to describe the orientation or direction of a surface feature.
2) The EL angle represents the vertical angle in a polar coordinate system, measured in degrees or radians. It describes the inclination or tilt of a surface feature with respect to the horizontal plane.

Psuedo code to implement AZ and EL given in Algorithm 2.

V. Proposed Methodology

The schematic illustration of our proposed method is depicted in Fig. 4. The method encompasses three main steps: patch image extraction, deep feature extraction, and unsupervised patch classification using dual transformers.

A. Surface Patch Image Extraction

Our segmentation technique uses local classification, in which the mesh surface is browsed and a neighborhood around each triangle facet is constructed as shown in Fig. 3(b); each neighborhood creates a multichannel geometric image with each channel representing a geometric feature.

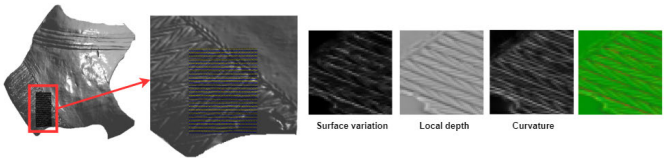


Fig. 5. Example of a surface patch image extraction. A facet grid is constructed around a central facet (here, a 24 × 24), and three different geometric descriptors are computed at each facet of the grid: SV, LD, and curvature, producing a three-channel image.

The multichannel image is constructed using the ORF structure developed in [20]. We extract an ORF from each facet and utilize it to generate a grid to encode facets as a 2-D matrix. Further, at each facet, three geometric descriptors are computed: SV, LD, and mean curvature, and the resulting geometric maps are stacked to generate a three-channel geometric image, which we refer to as the *surface patch image*, as shown in Fig. 5.

B. Deep Feature Extraction

The geometric image, while reflecting the local geometry of a surface patch, does not possess sufficient discrimination capacity. For improved discrimination, a pre-trained ResNet model is employed to create a deep feature representation f, from geometric images. The model has not been tuned or exposed to texture or nontexture data in an effort to stick to the concept of a fully unsupervised framework.

C. Initial Patch Clustering

Unsupervised learning techniques are more effective when the classes are homogeneous and compact (e.g., a k-mean clustering works fine when the feature space's class distributions are compact and reasonably separated). While such an ideal scenario is unlikely in our data, we can reduce the heterogeneity of the classes' samples (here, patch instances in the texture and the nontexture classes). Assuming our deep features have adequate discrimination capacity, one method is to do mean-shift clustering on the deep feature samples,

select the two most predominant clusters, and discard the rest. The two dominating clusters are anticipated to display reasonable compactness as a density-based approach, whereas the excluded clusters are most likely to contain hard samples. Another simpler and computationally less demanding approach, which we found working reasonably, is to run the *K-means* clustering with many clusters above 2. In the experimentation, we empirically found $K = 5$, a suitable value.

D. Unsupervised Patch Classification

As mentioned before, our unsupervised patch classification employs a model composed of two modules, the LG, and the LC. The two models encompass an autoencoder-like model and a binary classifier, respectively. For both models, we adopted a transformer backbone architecture. While transformers demonstrated remarkable performance in several image analysis tasks [38], [39], [40], our primary motivation stems from their capacity to model both short-range and long-range dependencies. This aspect is quite present in the textured surface patches because of the repetitive patterns all along their surface. We dubbed the LG and the LC the transformer LG (TLG) and the transformer LC (TLC).

1) Transformer Label Generator: Our transformer projector comprises a multi-head self-attention (MSA) layer and a multilayer perceptron (MLP) containing two fully connected layers. The filtered patch instances obtained from the initial patch clustering (IPC) are passed to TLG. Here, their deep feature representations are projected into a latent space using a transformer-based projector and then inverse-transformed to the original space using a transformer-based inverse projector. The transformation loss is then used to assign pseudo-labels to each patch instance.

We employed a similar transformer architecture proposed by Vaswani et al. [41]. Let N be the number of patches in the mesh surface, and let f_i be the deep feature representation of the ith patch, then we re-arrange f_i as a sequence of position-aware word representations $g_i = [g_{i,1}, g_{i,2}, \ldots, g_{i,n_k}]$, n_k is the length of the sequence. The projector converts g_i to a latent representation p_L via. the following sequence of transformations:

$$\begin{aligned} p_0 &= g_i \\ q_x &= k_x = v_x = \mathbf{LN}(p_{x-1}) \\ \hat{p}_x &= \mathbf{MSA}(q_x, k_x, v_x) + p_{x-1} \\ p_L &= [q_{i,1}, q_{i,2}, \ldots, q_{i,n_k}] \end{aligned} \tag{1}$$

where $x = 1, \ldots, L$ denotes the number of layers and LN represents layer normalization. In the TLG architecture, the latent space retains the same size as the input sequence.

The architecture of the inverse projector is similar to that of the transformer projector. It consists of two MSA layers followed by MLP. There is also a latent learned bias vector b utilized in reconstructing features $z_L = [\hat{g}_{i,1}, \hat{g}_{i,2}, \ldots, \hat{g}_{i,n_k}]$ via the sequence of transformations

$$\begin{aligned} z_0 &= p_L \\ q_x &= k_x = \mathbf{LN}(z_{x-1}) + b, \quad v_x = \mathbf{LN}(z_{x-1}) \\ \hat{z}_x &= \mathbf{MSA}(q_x, k_x, v_x) + z_{x-1}, \quad \hat{q}_x = \mathbf{LN}(\hat{z}_x) + b \\ \hat{k}_x &= \hat{v}_x = \mathbf{LN}(z_0), \quad \tilde{z}_x = \mathbf{MSA}(\hat{q}_x, \hat{k}_x \hat{v}_x) + \hat{z}_x \\ z_x &= \mathbf{MLP}(\mathbf{LN}(\tilde{z}_x)) + \tilde{z}_x. \end{aligned}$$

We optimize the TLG by minimizing the following loss function:

$$\mathcal{L}_{\text{TLG}} = \sum_{i=1}^{n} ||g_i - \hat{g}_i||_1 \tag{2}$$

where n is the total number of surface patches in a batch. Once optimized, the reconstruction error is computed for the ith patch instance as

$$e^i_{\text{TLG}} = ||g_i - \hat{g}_i||_1. \tag{3}$$

Afterward, we generate its pseudo-label in the first iteration by thresholding

$$l_i = \begin{cases} 1, & \text{if } e^i_{\text{TLG}} - \text{average}_{\text{batch}}\left(e^i_{\text{TLG}}\right) \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where the label 1 and 0 correspond to the texture and nontexture, respectively.

In the subsequence iterations, the pseudo-label assignment is modified. For a patch labeled nontexture in the previous iteration, the reconstruction error of (3) is used. The reconstruction error of the following equation is used for a patch-labeled texture:

$$e^i_{\text{TLG}} = ||\tilde{g}_f - \hat{g}_i||_1 \tag{5}$$

where $\tilde{g}_f$ is a random Gaussian vector having normal distribution. Empirically, we found that switching to the above formula enhances the capacity of the TLG to detect the textured patches and improves the overall segmentation. To train a generator to produce desired images in a generative framework, a negative correlation between the discriminator and generator losses must be achieved [42]. In our network, a similar approach is employed to get desired labels by increasing the loss of the discriminator for texture by providing a fixed Gaussian as input.

2) Transformer Label Cleaner: We also employ transformer architecture similar to the transformer projector for the TLC, where the last layer is connected to a dense neuron. Further, the TLC, a binary classifier, is trained with the patches used in the previous step and their pseudo-labels generated in (4), using a simple binary cross-entropy loss

$$\mathcal{L}_{\text{TLC}} = \frac{1}{n} \sum_{i=1}^{n} -(l_i \log(\phi_i) + (1 - l_i) \log(1 - \phi_i)) \tag{6}$$

where ϕ_i is the output of the binary classifier represents the probability of a textured region, and $1 - \phi_i$ represents the probability of a nontextured region. Once trained, each patch instance is passed to the binary classifier, and its label is adjusted as follows:

$$l_i = \begin{cases} 1, & \text{if } \phi_i \geq \text{average}_{\text{batch}}(\phi_i) \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

These adjusted labels l_i are used to train the LG in the next iteration.

TLG and TLC alternate over the batch of surface patches until the mesh surface is completely covered. The algorithm goes into the next epoch till a maximum number of epochs is reached. Fig. 6 depicts an exemplar of the evolution of the patch classification across the epochs. It is evident that the

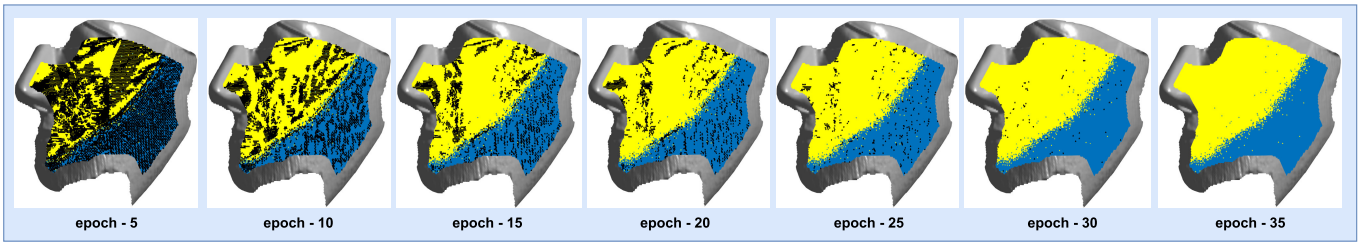


Fig. 6. Illustration of the segmentation improvement across the iterations. Correctly classified texture facets are colored in blue, nontexture in yellow, and misclassified in black.

Algorithm 3 Proposed Algorithm

Input: N deep feature vectors representing all the surface patches
for each epoch **do**
Take a batch of n samples
Minimize TLG's loss function (2)
Compute construction errors e^i_{TLG}, i=*1:n*, using equation (3)
set pseudo-label l_i using equation (4)
———Label Cleaner———
Minimize TLC's loss function (6) using the labels l_i
Compute $\phi_i \leftarrow TLC(g_i, l_i),\ i = 1:n$
if $\phi_i > \beta_c$ **then**
$l_i \leftarrow$ textr
end if
while iter **do**
Take a batch of n samples
Minimize TLG's loss function (2)
if $l_i ==$ textr **then**
$g_i \leftarrow Gaussian\ noise(g_f)$
Compute e^i_{TLG} as per equation (5)
else
Compute e^i_{TLG} as per equation (3)
end if
———Label Cleaner———
Minimize TLC's loss function (6) using the labels l_i
Compute $\phi_i \leftarrow TLC(g_i, l_i),\ i = 1:n$
Compute l_i using equation (7)
end while
end for
Return cleaned label l_i

segmentation improves as the number of iterations increases, resulting in well-separated textured and nontextured regions. The pseudo code for implementation of the proposed approach is given in Algorithm 3.

E. Weakly Supervised Algorithm

To verify performance, the majority of existing unsupervised algorithms in the literature are trained in a weakly supervised manner. Weakly supervised experimental settings refer to situations where the amount or quality of labeled data is insufficient for training a deep learning model. We therefore include weak supervision in our proposed framework. In the first setting, the proposed algorithm is trained with weak supervision, only a subset of the data is labeled, and the rest of the data is unlabeled. The model learns from both labeled and unlabeled data to improve its performance to distinguish between textures and nontextures.

VI. Experimental Results

We evaluate our frameworks using three datasets: SHREC'17 [17], SHREC'18 [16], and KU 3DTexture [15]. SHREC'17 contains 15 distinct textures with 720 meshes, and each texture class contains 48 samples with varying mesh resolutions. The dataset SHREC'18 has 12 distinct surfaces with distinct texture patterns, each with a unique resolution. The KU 3DTexture [15] contains 89 real-world data samples with dense texture regions. Since the problem involves classifying each facet, the number of facets exposed to the network is essential. The data have a minimum of 10–785 K facets per sample. Even though the number of surfaces used is smaller, we found that the overall number of available facets is sufficient to train the network. Despite this, we have utilized augmented data and subjected our model to various surface variances to generalize to previously unseen patches.

The performance of the proposed method is compared to ten existing techniques, including seven based on supervised and three based on unsupervised techniques. All unsupervised methods employed in the performance evaluation incorporated ORF. Conversely, in supervised approaches, the baseline techniques employed for comparison accept either 3-D point clouds or 3-D meshes as inputs and therefore do not incorporate ORF. The proposed method is evaluated and compared using F1-Score, Precision, and Recall, with an Intersection over Union (IoU) threshold of 0.5. Additionally provided is the mean IoU (mIoU) score. The objective is to categorize each facet of a given surface as belonging to a texture or nontexture region.

A. Quantitative Analysis

We compared the proposed method to unsupervised and fully supervised approaches. Since there is no current unsupervised approach for texture segmentation on 3-D surfaces, we initially implemented three traditional methods to compare the proposed method with: K-Means [43], density-based spatial clustering of applications with noise (DBSCAN) [44], and Gaussian Mixture Model (GMM) Clustering [45]. In addition, we compare the performance of the proposed method with popular 3-D shape classification and segmentation networks [35], [46], [47], [48], [49], [50], [51]. The objective of these networks is to segment distinctive and consistent structures. Distinctive shapes are utilized to discern the unique structures within each class. Our goal is to classify individual

TABLE I

QUANTITATIVE RESULTS OF OUR METHOD AND BASELINES ON THE SHREC'17 [17] DATASET. BOLD FONT INDICATES THE TOP-PERFORMING RESULTS, WHILE THE BLUE FONT HIGHLIGHTS THE SECOND-BEST PERFORMANCE

		Pre ↑	Rec ↑	F1 ↑	mIoU ↑	Parameters	Inference Time (Sec)
Supervised Approaches	PointNet [CVPR'17] [46]	-	-	-	51.8	3.5M	0.2
	PointNet++ [NeurIPS'17] [47]	-	-	-	48.3	1.5M	0.1
	MeshSegNet [TMI'20] [48]	-	-	-	62.4	1.8M	0.1
	BAAFNet [CVPR'21] [49]	-	-	-	56.2	5.6M	0.1
	PointMLP [ICLR'22] [50]	-	-	-	67.3	13.2M	0.3
	PointNeXt-S [NeurIPS'22] [35]	-	-	-	69.1	1.5M	0.1
	CurveNet [ICCV'21] [51]	-	-	-	66.4	5.5M	0.1
	Proposed$_{sup}$				**79.0**	5.4M	0.6
Unsupervised Approaches	K-Means [43]	23.6	21.4	22.6	30.5	-	10.3
	DBSCAN [44]	27.1	26.4	26.7	36.5	-	5.3
	GMM Clustering [45]	12.1	8.3	10.2	16.4	-	5.4
	Proposed	**68.2**	**69.1**	**69.0**	**70.1**	5.4M	21.0

TABLE II

QUANTITATIVE RESULTS OF OUR METHOD AND BASELINES ON THE SHREC'18 DATASET [16]. BOLD FONT INDICATES THE TOP-PERFORMING RESULTS, WHILE THE BLUE FONT HIGHLIGHTS THE SECOND-BEST PERFORMANCE

		Pre ↑	Rec ↑	F1 ↑	mIoU ↑	Parameters	Inference Time (Sec)
Supervised Approaches	PointNet [CVPR'17] [46]	-	-	-	54.2	3.5M	0.6
	PointNet++ [NeurIPS'17] [47]	-	-	-	58.1	1.5M	0.6
	MeshSegNet [TMI'20] [48]	-	-	-	60.3	1.79M	0.5
	BAAFNet [CVPR'21] [49]	-	-	-	58.7	5.3M	0.6
	PointMLP [ICLR'22] [50]	-	-	-	66.7	13.2M	1.2
	PointNeXt-S [NeurIPS'22] [35]	-	-	-	68.1	1.5M	0.1
	CurveNet [ICCV'21] [51]	-	-	-	70.4	5.5M	0.6
	Proposed	-	-	-	**82.0**	5.4M	3.8
Unsupervised Approaches	K-Means [43]	33.6	25.4	29.5	38.2	-	20.6
	DBSCAN [44]	28.7	30.6	29.4	35.0	-	9.4
	GMM Clustering [45]	10.3	8.2	9.1	12.1	-	9.5
	Proposed	**68.1**	**69.6**	**70.0**	**73.4**	5.4M	52.0

points or facets as textured or nontextured based on local SVs, rather than segmenting the overall shape. Moreover, there exists an imbalance between the proportions of texture and nontexture regions. Therefore, we refined the loss functions, introducing a balanced focal loss to prioritize challenging classes, thereby adapting the models for texture and nontexture classification. The input comprises labeled point clouds or 3-D meshes, with each point and facet annotated.

1) Evaluation on SHREC'17 Dataset: This dataset presents a significant challenge due to the wide variety of mesh resolutions and texture patterns. The proposed technique has yielded promising results and demonstrates its robustness against varying mesh resolutions. Table I clearly shows that the proposed method under supervised and unsupervised conditions performed better than the classical and deep learning-based methods. Moreover, it is worth mentioning that the proposed unsupervised approach performs better than all the supervised approaches [35], [46], [47], [48], [49], [50], [51] with a better margin. The proposed method using supervised and unsupervised is the best performer, and PointNeXt-S [35] and DBSCAN [44] are the second-best performer.

2) Evaluation on SHREC'18 Dataset: We additionally evaluate our method on SHREC'18, which has 3-D surfaces with multiple texture patterns on each surface with complex boundaries between the patterns. Also, the surfaces with varying mesh resolution which is further challenging. As shown in Table II, the proposed method under supervised and unsupervised is the best performer, and curveNet [51], and K-Means [43] is the second-best performer. Also, the scores obtained by our unsupervised approach are close to our fully supervised counterpart, and also it is superior to all supervised approaches [35], [46], [47], [48], [49], [50], [51].

TABLE III

QUANTITATIVE RESULTS OF OUR METHOD AND BASELINES ON THE KU 3DTEXTURE [15] DATASET. BOLD FONT INDICATES THE TOP-PERFORMING RESULTS, WHILE THE BLUE FONT HIGHLIGHTS THE SECOND-BEST PERFORMANCE

		Pre ↑	Rec ↑	F1 ↑	mIoU ↑	Parameters	Inference Time (Sec)
Supervised Approaches	PointNet [CVPR'17] [46]	-	-	-	48.2	3.5M	1.0
	PointNet++ [NeurIPS'17] [47]	-	-	-	49.7	1.5M	0.8
	MeshSegNet [TMI'20] [48]	-	-	-	58.0	1.8M	0.8
	BAAFNet [CVPR'21] [49]	-	-	-	51.1	5.5M	0.7
	PointMLP [ICLR'22] [50]	-	-	-	67.0	13.2M	1.6
	PointNeXt-S [NeurIPS'22] [35]	-	-	-	68.9	1.5M	0.1
	CurveNet [ICCV'21] [51]	-	-	-	64.0	5.5M	0.9
	Proposed$_{sup}$	-	-	-	**80.3**	5.4M	5.6
Unsupervised Approaches	K-Means [43]	12.6	18.4	16.1	22.5	-	26.1
	DBSCAN [44]	17.1	26.8	22.5	30.6	-	18.9
	GMM Clustering [45]	6.9	10.1	23.5	15.3	-	18.5
	Proposed	**65.2**	**66.4**	**65.0**	**66.2**	5.4M	58.0

3) Evaluation on KU 3DTexture Dataset: The results of our method, together with other sets of supervised and unsupervised approaches, are reported in Table III. Our method surpasses the classical clustering-based unsupervised methods [43], [44], [45] by large margins on all metrics showing the advantage of our method in fully leveraging both transformer modules, LG, and LC. The proposed unsupervised approach has superior performance than five techniques [46], [47], [48], [49], [51] out of seven. KU 3DTexture has diverse patterns and complex surfaces, so the results obtained are slightly less compared to the other two datasets. Also, in the case of the supervised approach, the proposed approach has shown superior performance compared to [46], [47], [48], [49], and [51]. Though these approaches are designed for 3-D shape analysis and demonstrated remarkable performance on semantic segmentation of 3-D shapes, in our case, they are not successful in capturing the textures on 3-D surfaces.

B. Qualitative Analysis

A few samples in Fig. 7 show the effectiveness of the proposed technique. The 3-D surfaces presented have multiple texture patterns; however, we are interested in binary classification; we consider all texture patterns as one class and all nontexture patterns as another. A few facet misclassifications on the segmented surfaces using qualitative analysis are discovered, particularly at the texture and nontexture boundaries. This is because using ordered rings around a facet at boundaries covers neighboring facets from texture to nontexture regions. We use a wide range of facets, from texture and nontexture, to handle these challenges to some extent. However, the issues are inescapable because the surfaces come in various patterns and resolutions. The top two rows in Fig. 7 show the 3-D surfaces and respective ground truths, and the remaining rows show the predicted results, where blue represents the texture region and yellow represents the nontextured region. Since ORF does not cover the entire surface due to boundary restrictions, only the central portion of surfaces is utilized for training and testing.

VII. ABLATION STUDIES

We conducted five ablative tests on the SHREC'18 dataset to evaluate different configurations of the proposed model. The performance of each variant has been analyzed by introducing components to the base model, and the results are presented in Table IV, showcasing the impact of each component on the overall system.

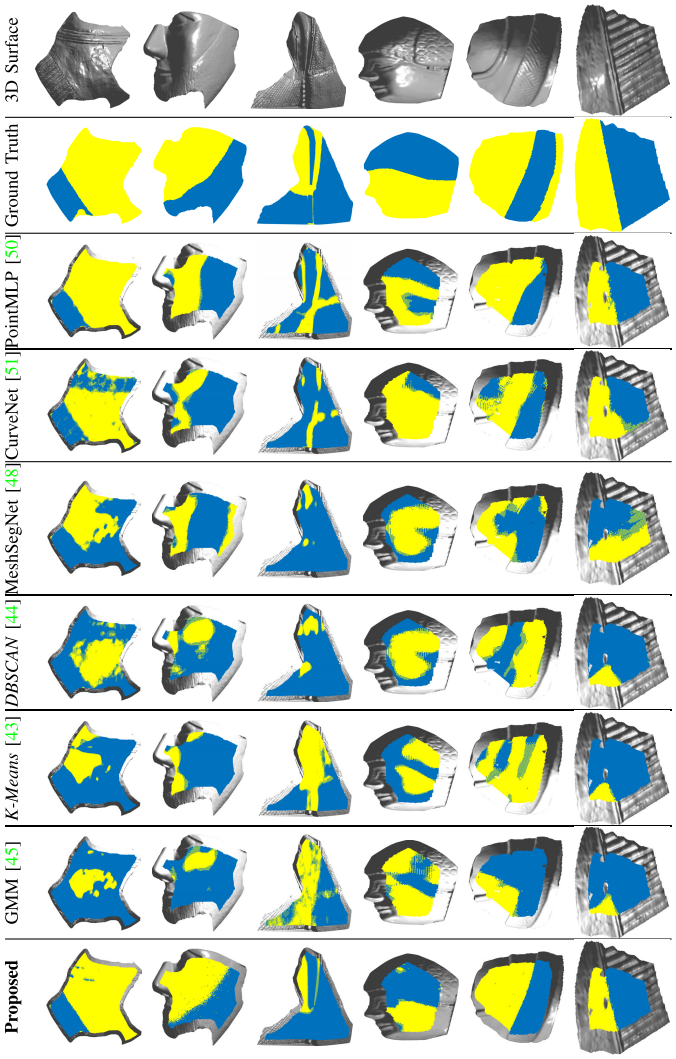


Fig. 7. Segmentation outcomes of the proposed approach are depicted using a selection of samples. The initial row exhibits the original 3-D surfaces, while the second row exhibits the associated ground truth. In the ground truth, the nontexture region is highlighted in yellow, and the texture region is represented in blue. The segmentation from both baseline methods and the proposed approach are shown in rows from the third to the last.

TABLE IV

ABLATION STUDY FOR PROPOSED MODULES ON SHREC'18. ALG STANDS FOR AUTO-ENCODER-BASED LABEL GENERATOR, MLC STANDS FOR MLP-BASED LABEL CLEANER, TLG STANDS FOR TRANSFORMER LABEL GENERATOR AND TLC STANDS FOR TRANSFORMER LABEL CLEANER

Module	Pre ↑	Rec ↑	F1 ↑	mIoU ↑
ALG	60.2	59.4	60.1	62.8
ALG + MLC	62.5	63.2	63.6	64.2
TLG	64.0	63.1	63.4	66.0
TLG + Initial Clustering	65.8	66.0	66.3	67.5
TLG + TLC	68.3	67.1	67.8	70.5
TLG + TLC + Initial Clustering	**69.6**	**68.1**	**70.0**	**73.4**

1) Initially, the experiment involved an autoencoder-based label generator (ALG), achieving an overall F1-score of 60.1%.
2) The addition of an LC based on MLP led to a 3.5% increase in the overall F1-score.
3) Replacing the autoencoder with a transformer-based architecture (TLG) resulted in a 3.3% improvement compared to ALG.
4) Including initial clustering along with TLG boosted the overall F1-score by 6%.
5) Integrating a transformer-based discriminator to fine-tune the pseudo-labels generated by TLG yielded a 7.7% increase in overall performance.
6) The complete network TLG + TLC + IC demonstrated the most significant improvement, with a 10% increase compared to the base model.

Despite initial clustering not precisely labeling patches, the addition of this module to the framework, feeding the two largest clusters to the LG and LC, significantly enhanced performance. Instance clustering effectively eliminates uncertain patches, enabling the proposed network to establish a decision boundary with high confidence. Similarly, the discriminator module played a crucial role, improving efficiency by approximately 7.7% through the removal of misclassified labels. The architectures of simple ALG includes seven fully connected layers [1024, 512, 256, 128, 256, 512, 1024] and an MLP-based LC (MLC). We set the maximum number of epochs to 200. However, we can reduce the number of epochs using a proper convergence criterion (e.g., when the number of texture and nontexture labels stabilizes).

TABLE V

MEAN AND VARIATIONS OF THE TWO DISTRIBUTIONS ACROSS THE EPOCHS. AS THE EPOCH INCREASES, THE MEAN OF THE TEXTURE AND THE NONTEXTURE SHIFT TOWARD HIGH AND LOW VALUES, RESPECTIVELY

Epoch	*non-Texture (mean)*	*Texture (mean)*
5	0.65	1.05
10	0.50	1.10
15	0.40	1.15
20	0.37	1.17
25	0.35	1.25
30	0.33	1.30

A. Loss Distributions

We labeled 3-D surfaces using the reconstruction losses of texture and nontexture patches. We hypothesize that these losses would be significant for textured patches because of the surface's local shape heterogeneity and low for nontexture patches. We reported the loss distributions for several increasing epochs in Fig. 8 to support our idea. We notice that the loss distributions of the textured and the nontextured patches move toward the right-end (high value) and the left-end (low value) as the epochs progress. We can also observe that the shape of the two distributions evolves from a multimodal with a high variation to a unimodal with a low variation. This behavior is also confirmed by the evolution of the means of two distributions across the same epochs, which we reported in Table V. This evolution is reflected in the segmentation, which starts with heterogeneous and mixed regions and converges toward two compact and separated regions corresponding to the texture and the nontexture classes.

Note that the segmentation can be improved further using morphological post-processing operations, e.g., hole filling.

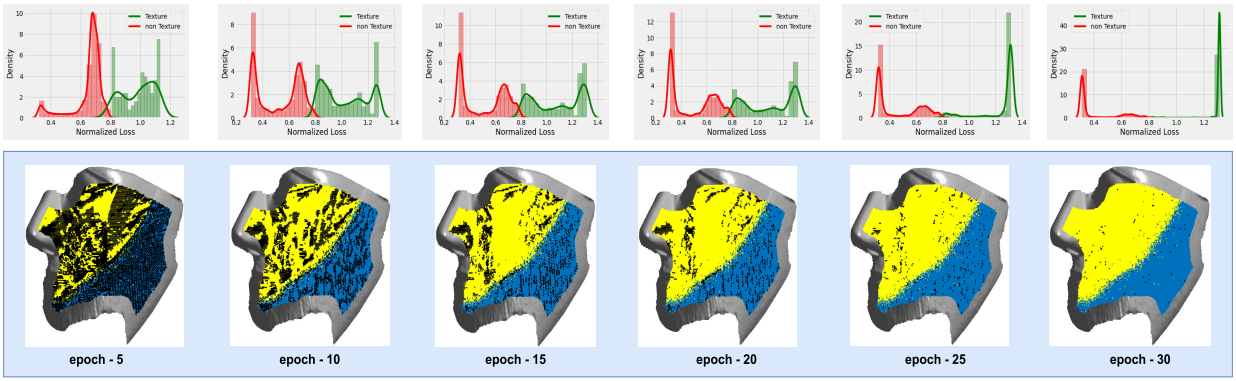


Fig. 8. Loss distributions and facet classification sampled across several epochs. The top row shows the loss distributions for texture and nontexture patches, while the bottom row shows the facets classifications (yellow: nontexture, blue: texture, and black: mis-classified).

TABLE VI
ABLATION STUDY ON GRID SIZE OF FEATURE PATCH GENERATION ON SHREC'18. BOLD REPRESENT THE BEST PERFORMANCE AND BLUE HIGHLIGHT REPRESENT THE SECOND-BEST PERFORMANCE

Grid size	Pre ↑	Rec ↑	F1 ↓	mIoU ↓
8 x 8	63.6	60.5	62.4	68.6
16 x 16	65.0	68.2	67.1	70.6
24 x 24	**68.1**	**69.6**	70.0	73.4
32 x 32	68.0	69.1	**70.2**	**74.0**
20 x 20	66.2	67.4	66.7	70.5

For example, a single facet classified as texture surrounded by nontexture facets should be converted to a nontexture, and vice versa. Many facets meeting this description can be spotted by zooming in on the last segmentation figure at epoch 30 in Fig. 8.

B. Parameters Selection

We tested extensively the parameters that influence performance in the proposed approach.

Grid size is an important parameter since it determines the feature image size at each facet. It is essential to choose a grid size that covers a facet with sufficient surface area to determine whether the facet belongs to texture or not. After experimenting with various grid sizes, we found that a range of 24–32 worked best with our proposed method, yielding superior results for both low- and high-resolution meshes. A small grid size does not adequately cover the surface area and performs poorly, while increasing the grid size has a border effect that reduces the segmented surface's area. As shown in Table VI, the performance of various grid sizes has been evaluated, revealing that grid sizes 24 and 32 provide superior performance compared to other grid sizes. Although there is a slight performance difference between grid sizes 24 and 32, we chose a grid size of 24 for our experiment.

Feature selection is another important parameter that affects performance. We have tested multiple feature combinations to extract patches and checked the performance of the proposed approach. Since the texture pattern is a local variation on the surface, as expected, a combination of SVs, LD, and curvatures has shown better results. Table VII summarizes the performance for different combinations. However, we observed that LD plays an important role, and its combination with other geometric features has consistently shown better results. The other combinations related to SVs, such as SI, also show better results; however, they produce false positives in edge-like structures detected as texture.

C. Comparison With Weakly Supervised Approach

The primary objective of our work is to perform entirely unsupervised classification of texture and nontexture at the facet level, which has not yet been described in the literature. To achieve this goal, we implemented weakly supervised training, in which a subset of labeled data serves as a representative for the task. We conducted experiments on three datasets, varying the percentage of labeled data from 0% (representing unsupervised learning) to 100% (representing supervised learning). For weakly supervised learning, we incrementally employed 10% of labels to train our proposed model and evaluated its performance. The results, as shown in Table VIII, demonstrate that the performance is close to unsupervised learning when fewer labels are used and close to supervised learning when the entire labeled dataset is used. Additionally, we found that each of the three datasets with 50% labeled data yielded promising results comparable to those of supervised approaches.

D. Comparison With Unsupervised Learning-Based Methods

We compared our unsupervised segmentation method with conventional techniques. Additionally, we evaluated several contemporary approaches, including self-supervised [52], [53], weakly supervised [54], [55], [56], and unsupervised learning-based methods [57], [58], [59], [60], [61]. However, these methods are designed for 2-D images and cannot be applied directly to 3-D meshes. To use these methods on 3-D models, we need to first convert them to a 2-D domain. However, this conversion is challenging because 3-D models contain complex manifolds and mapping points from a point cloud to a regular 2-D grid image is difficult. Additionally, focusing on the surfaces of 3-D models is critical for texture identification, and projecting to a 2-D domain often results in the loss of information for small details, which are essential for texture recognition.

TABLE VII

ABLATION STUDIES ON GEOMETRIC FEATURE SELECTION FOR PATCH GENERATION. CUR—CURVATURE, AZ—AZIMUTH ANGLE, SV—SURFACE VARIATION, EL—ELEVATION ANGLE, SI—SHAPE INDEX, AND LD—LOCAL DEPTH

	[Cur, AZ, EL]	*[LD, AZ, EL]*	*[SI, LD, Cur]*	*[SV, AZ, EL]*	*[SV, LD, AZ]*	*[SV, LD, Cur]*	*[SV, SI, AZ]*	*[SV, SI, Cur]*	*[SV, SI, LD]*
Pre	56.8	54.3	67.3	54.2	50.1	**68.1**	61.0	62.6	63.2
Rec	56.0	55.1	**70.5**	52.7	52.6	69.6	60.7	63.0	64.5
F1	57.1	54.7	68.5	53.4	51.3	**70.0**	60.9	62.8	64.0

TABLE VIII

USING THREE DATASETS, THE PERFORMANCE OF THE PROPOSED METHOD FOR TEXTURE VERSUS NONTEXTURE CLASSIFICATION IN TWO DISTINCT SCENARIOS, INCLUDING A COMPLETELY UNSUPERVISED ALGORITHM AND WEAKLY SUPERVISED ALGORITHM, HAS BEEN EVALUATED. FOR UNSUPERVISED SETTINGS, 0% LABELS ARE USED, BUT FOR WEAKLY SUPERVISED SETTINGS, VARIABLE PERCENTAGES OF LABELS ARE USED, AND MIOU IS REPORTED FOR EACH DATASET

Datasets	**0%**	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
KU 3D Texture	66.2	68.7	69.2	70.8	73.1	74.8	76.7	78.2	80.1	80.2	80.3
SHREC'17	70.1	70.9	71.8	73.2	74.5	75.6	76.4	77.6	78.4	79.0	79.0
SHREC'18	73.4	74.0	74.6	74.9	75.6	76.2	79.6	80.2	81.9	82.0	82.0

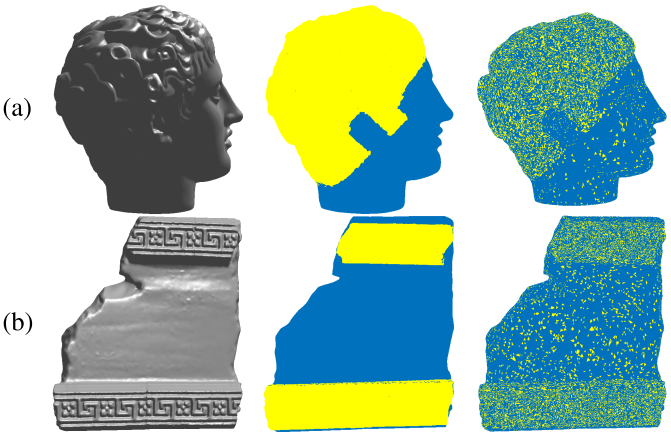


Fig. 9. Proposed method exhibits some instances of failure, as shown above, highlighting two specific scenarios. (a) In the first scenario, a 3-D head example is chosen, where the hair texture is globally visible but not locally evident. (b) In the second scenario, a surface has highly dense texture regions and less dense nontexture regions. The ground truth of (b) reveals that the texture (yellow) regions occupy a relatively smaller percentage of space on the 3-D surface compared to the nontexture (blue) regions.

VIII. FAILURE CASES

The proposed approach exhibited lower performance in specific cases. One notable scenario involved the selection of a 3-D surface characterized by a complex manifold, where an almost equal number of facets represented texture and nontexture regions. Upon visual inspection, it became apparent that, despite the equal representation, the texture regions occupied a significantly smaller area compared to the nontexture regions. In essence, the texture regions were highly dense relative to the nontexture regions. The use of a fixed grid size for creating 2-D images at each facet made this imbalance worse, which contributed to the observed decline in performance. We are actively addressing this issue in our future work by implementing adaptive grid sizes with a multiscale approach. This adaptive strategy aims to better handle surfaces with imbalanced texture and nontexture regions, thereby improving the robustness and overall performance of our technique. In addition to the previously mentioned scenario, we also observed another case where the surface texture apparent on a global scale while not visible at the local level. Specifically, we chose a 3-D surface representing hair, where the SVs are evident when observed holistically. However, the challenge arose when attempting to capture these variations at the local level within the neighborhood. Due to the spread of the texture region over a large manifold, local representations struggled to distinctly capture the intricate texture details. The challenges in discerning the texture variations locally impacted the overall performance of the proposed technique in this specific case. Qualitative analysis of both samples is shown in Fig. 9.

IX. CONCLUSION AND FUTURE WORK

In this article, a novel method for segmenting surfaces into textured and nontextured regions is presented. The proposed method is entirely unsupervised, unlike previous techniques, which are limited to classification and retrieval and rely on human annotation for training networks. The proposed fully unsupervised framework consists of an LG and LC in which samples are projected onto a latent space and then inverse-projected to the original space using a projector and an inverse projector. Instances are assigned a texture or nontexture pseudo-label based on the transformation error. The LC then cleans these pseudo-labels using a label-cleaning mechanism. Both modules learn from each other in an iterative manner to produce improved labels. The generator module used a discriminative learning mechanism based on the estimated cleansed labels. This makes the transformation error go up for positive examples and down for negative examples. We conducted experiments on three distinct datasets in both fully unsupervised and weakly supervised settings and achieved segmentation results comparable to those of supervised methods. We intend to develop a multiclass segmentation method for textured surfaces with adaptive grid size as part of our future work.

REFERENCES

[1] D. Krawczyk and R. Sitnik, "Segmentation of 3D point cloud data representing full human body geometry: A review," *Pattern Recognit.*, vol. 139, Jul. 2023, Art. no. 109444.

[2] H. Du, X. Yu, F. Hussain, M. A. Armin, L. Petersson, and W. Li, "Weakly-supervised point cloud instance segmentation with geometric priors," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 4260–4269.

[3] C. Wu, X. Bi, J. Pfrommer, A. Cebulla, S. Mangold, and J. Beyerer, "Sim2real transfer learning for point cloud segmentation: An industrial application case on autonomous disassembly," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 4520–4529.

[4] M. S. Lee, S. W. Yang, and S. W. Han, "GaIA: Graphical information gain based attention network for weakly supervised point cloud semantic segmentation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 582–591.

[5] Z. Du, H. Ye, and F. Cao, "A novel local–global graph convolutional method for point cloud semantic segmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 14, 2022, doi: 10.1109/TNNLS.2022.3155282.

[6] C.-Q. Huang, F. Jiang, Q.-H. Huang, X.-Z. Wang, Z.-M. Han, and W.-Y. Huang, "Dual-graph attention convolution network for 3-D point cloud classification," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 6, 2022, doi: 10.1109/TNNLS.2022.3162301.

[7] S. Li, Y. Liu, and J. Gall, "Rethinking 3-D LiDAR point cloud segmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 16, 2021, doi: 10.1109/TNNLS.2021.3132836.

[8] A. Othmani, L. F. C. L. Y. Voon, C. Stolz, and A. Piboule, "Single tree species classification from terrestrial laser scanning data for forest inventory," *Pattern Recognit. Lett.*, vol. 34, no. 16, pp. 2144–2150, Dec. 2013.

[9] M. Zeppelzauer et al., "Interactive 3D segmentation of rock-art by enhanced depth maps and gradient preserving regularization," *J. Comput. Cultural Heritage*, vol. 9, no. 4, pp. 1–30, Sep. 2016.

[10] P. Akiva, M. Purri, and M. Leotta, "Self-supervised material and texture representation learning for remote sensing tasks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8203–8215.

[11] C. Morbidoni, R. Pierdicca, R. Quattrini, and E. Frontoni, "Graph CNN with radius distance for semantic segmentation of historical buildings tls point clouds," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XLIV-4/W1-2020, pp. 95–102, Sep. 2020.

[12] E. Grilli, E. M. Farella, A. Torresani, and F. Remondino, "Geometric features analysis for the classification of cultural heritage point clouds," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. XLII-2/W15, pp. 541–548, Aug. 2019.

[13] I. I. Ganapathi, S. Javed, R. B. Fisher, and N. Werghi, "Graph based texture pattern classification," in *Proc. 8th Int. Conf. Virtual Reality (ICVR)*, May 2022, pp. 363–369.

[14] F. Matrone et al., "A benchmark for large-scale heritage point cloud semantic segmentation," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. B2, pp. 1419–1426, Jan. 2020.

[15] I. I. Ganapathi and N. Werghi, "Labeled facets: New surface texture dataset," in *Proc. Eurographics Workshop 3D Object Retr.*, S. Berretti, T. Thehoaris, M. Daoudi, C. Ferrari, and R. C. Veltkamp, Eds., 2022, pp. 1–6.

[16] S. Biasotti et al., "SHREC'18 track: Recognition of geometric patterns over 3D models," in *Proc. Eurographics Workshop 3D Object Retr.*, vol. 2, 2018, pp. 71–77.

[17] S. Biasotti et al., "SHREC'17 track: Retrieval of surfaces with similar relief patterns," in *Proc. 10th Eurographics Workshop 3D Object Retrieval*, 2017, pp. 1–9.

[18] N. Werghi, C. Tortorici, S. Berretti, and A. Del Bimbo, "Boosting 3D LBP-based face recognition by fusing shape and texture descriptors on the mesh," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 5, pp. 964–979, May 2016.

[19] N. Werghi, S. Berretti, and A. del Bimbo, "The mesh-LBP: A framework for extracting local binary patterns from discrete manifolds," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 220–235, Jan. 2015.

[20] N. Werghi, C. Tortorici, S. Berretti, and A. Del Bimbo, "Representing 3D texture on mesh manifolds for retrieval and recognition applications," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2521–2530.

[21] E. Moscoso Thompson et al., "SHREC 2020: Retrieval of digital surfaces with similar geometric reliefs," *Comput. Graph.*, vol. 91, pp. 199–218, Oct. 2020.

[22] E. M. Thompson and S. Biasotti, "Description and retrieval of geometric patterns on surface meshes using an edge-based LBP approach," *Pattern Recognit.*, vol. 82, pp. 1–15, Oct. 2018.

[23] S. Liu, R. R. Martin, F. C. Langbein, and P. L. Rosin, "Segmenting geometric reliefs from textured background surfaces," *Comput.-Aided Design Appl.*, vol. 4, no. 5, pp. 565–583, Jan. 2007.

[24] R. Zatzarinni, A. Tal, and A. Shamir, "Relief analysis and extraction," in *Proc. ACM SIGGRAPH Asia papers*, Dec. 2009, pp. 1–9.

[25] C. Tortorici, S. Berretti, A. Obeid, and N. Werghi, "Convolution operations for relief-pattern retrieval, segmentation and classification on mesh manifolds," *Pattern Recognit. Lett.*, vol. 142, pp. 32–38, Feb. 2021.

[26] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," in *Proc. Int. Conf. Pattern Recognit.*, vol. 1, 1994, pp. 582–585.

[27] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Diego, CA, USA, Jun. 2005, pp. 886–893.

[28] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[29] N. Werghi, C. Tortorici, S. Berretti, and A. D. Bimbo, "Local binary patterns on triangular meshes: Concept and applications," *Comput. Vis. Image Understand.*, vol. 139, pp. 161–177, Oct. 2015.

[30] N. Werghi, M. Rahayem, and J. Kjellander, "An ordered topological representation of 3D triangular mesh facial surface: Concept and applications," *EURASIP J. Adv. Signal Process.*, vol. 2012, no. 1, pp. 1–20, Dec. 2012.

[31] H. Lei, N. Akhtar, M. Shah, and A. Mian, "Mesh convolution with continuous filters for 3-D surface parsing," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 13, 2023, doi: 10.1109/TNNLS.2023.3281871.

[32] F. A. Limberger and R. C. Wilson, "Feature encoding of spectral signatures for 3D non-rigid shape retrieval," in *Proc. Brit. Mach. Vis. Conf.*, 2015, pp. 1–56.

[33] I. Sipiran, J. Lokoc, B. Bustos, and T. Skopal, "Scalable 3D shape retrieval using local features and the signature quadratic form distance," *Vis. Comput.*, vol. 33, no. 12, pp. 1571–1585, Dec. 2017.

[34] Y.-C. Choi, S. Murtala, B.-C. Jeong, and K.-S. Choi, "Deep learning-based engraving segmentation of 3-D inscriptions extracted from the rough surface of ancient stelae," *IEEE Access*, vol. 9, pp. 153199–153212, 2021.

[35] G. Qian et al., "PointNext: Revisiting PointNet++ with improved training and scaling strategies," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2022, pp. 23192–23204.

[36] T. Yao, Y. Li, Y. Pan, and T. Mei, "HGNet: Learning hierarchical geometry from points, edges, and surfaces," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 21846–21855.

[37] J. Choe, C. Park, F. Rameau, J. Park, and I. S. Kweon, "Pointmixer: Mlp-mixer for point cloud understanding," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 620–640.

[38] A. Kulkarni and S. Murala, "Aerial image dehazing with attentive deformable transformers," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 6305–6314.

[39] J. Jain, Y. Zhou, N. Yu, and H. Shi, "Keys to better image inpainting: Structure and texture go hand in hand," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 208–217.

[40] P. Zhang, L. Yang, J. Lai, and X. Xie, "Exploring dual-task correlation for pose guided person image generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7703–7712.

[41] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.

[42] I. Goodfellow, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[43] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.

[44] M. Ester, H. P. Kriegel, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, vol. 6, 1996, pp. 226–231.

[45] G. J. McLachlan and K. E. Basford, *Mixture Models: Inference and Applications to Clustering*, vol. 38. New York, NY, USA: M. Dekker, 1988.

[46] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[47] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5099–5108.

[48] C. Lian et al., "Deep multi-scale mesh feature learning for automated labeling of raw dental surfaces from 3D intraoral scanners," *IEEE Trans. Med. Imag.*, vol. 39, no. 7, pp. 2440–2450, Jul. 2020.

[49] S. Qiu, S. Anwar, and N. Barnes, "Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1757–1767.

[50] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual MLP framework," 2022, *arXiv:2202.07123*.

[51] T. Xiang, C. Zhang, Y. Song, J. Yu, and W. Cai, "Walk in the cloud: Learning curves for point clouds shape analysis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 915–924.

[52] A. Ziegler and Y. M. Asano, "Self-supervised learning of object parts for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14502–14511.

[53] L. Yang, W. Zhuo, L. Qi, Y. Shi, and Y. Gao, "ST++: Make self-trainingWork better for semi-supervised semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4258–4267.

[54] L. Xu, W. Ouyang, M. Bennamoun, F. Boussaid, and D. Xu, "Multi-class token transformer for weakly supervised semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 4310–4319.

[55] Z. Chen, T. Wang, X. Wu, X.-S. Hua, H. Zhang, and Q. Sun, "Class re-activation maps for weakly-supervised semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 969–978.

[56] L. Ru, Y. Zhan, B. Yu, and B. Du, "Learning affinity from attention: End-to-end weakly-supervised semantic segmentation with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16846–16855.

[57] X. Xia and B. Kulis, "W-Net: A deep model for fully unsupervised image segmentation," 2017, *arXiv:1711.08506*.

[58] W. Kim, A. Kanezaki, and M. Tanaka, "Unsupervised learning of image segmentation based on differentiable feature clustering," *IEEE Trans. Image Process.*, vol. 29, pp. 8055–8068, 2020.

[59] M. Engelcke, A. R. Kosiorek, O. P. Jones, and I. Posner, "GENESIS: Generative scene inference and sampling with object-centric latent representations," 2019, *arXiv:1907.13052*.

[60] Y. Ouali, C. Hudelot, and M. Tami, "Autoregressive unsupervised image segmentation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Nov. 2020, pp. 142–158.

[61] P. Savarese, S. S. Y. Kim, M. Maire, G. Shakhnarovich, and D. McAllester, "Information-theoretic segmentation by inpainting error maximization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4028–4038.

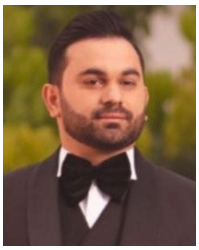
Iyyakutti Iyappan Ganapathi (Member, IEEE) received the Ph.D. degree from IIT Indore, Indore, India.

He was an Assistant Professor at Woosong University, Daejeon, South Korea. He is currently a Post-Doctoral Fellow at the Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates. His research interests include 3-D image processing, biometrics, computer vision, and machine learning.

Fayaz Ali Dharejo (Member, IEEE) received the B.E. degree in electronic engineering from QUEST, Nawabshah, Pakistan, in 2016, the M.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2018, and the Ph.D. degree in computer applied engineering from the Institute of Computer Network Information Center, Chinese Academy of Sciences, University of Chinese Academy of Sciences, Beijing, China, in 2022.

He has been a Post-Doctoral Fellow at Khalifa University, Abu Dhabi, United Arab Emirates, since September 2022. He has authored more than 30 articles within in reputed ISI impact factor journals, including IEEE TRANSACTIONS ON FUZZY SYSTEMS (TFS), *ACM Transactions on Intelligent Systems and Technology* (TIST), *International Journal of Intelligent Systems*, IEEE TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS (TCBB), and IEEE GEOSCIENCE AND REMOTE SENSING LETTERS (GRSL). His research interests include image enhancement, lightweight models, and computer vision.

Dr. Dharejo is a member of professional bodies, such as PEC, ACM, IEEE, and IEEE Societies, such as IEEE Geoscience and Remote Sensing Society, IEEE Computer Society, and IEEE Signal Processing Society.



Sajid Javed received the B.Sc. degree in computer science from the University of Hertfordshire, Hatfield, U.K, in 2010, and the combined master's and Ph.D. degree in computer science from Kyungpook National University, Daegu, South Korea, in 2017.

He was a Research Fellow at the University of Warwick, Coventry, U.K., from 2017 to 2018, and at Khalifa University (KU), Abu Dhabi, United Arab Emirates, from 2019 to 2021. He is currently a Faculty Member at KU. His research interests include visual object tracking in the wild, multiobject tracking, background–foreground modeling from video sequences, moving object detection from complex scenes, and cancer image analytics, including tissue phenotyping, nucleus detection, and nucleus classification problems. His research themes involve developing deep neural networks, subspace learning models, and graph neural networks.

Syed Sadaf Ali received the B.Tech. and Ph.D. degrees from IIT Indore, Indore, India.

He was a Post-Doctoral Researcher at the CNRS Laboratory, ENSEA, Cergy, France. He is currently a Post-Doctoral Researcher at Khalifa University, Abu Dhabi, United Arab Emirates. His research interests include image processing, computer vision, pattern recognition, and biometric template security.

Naoufel Werghi (Senior Member, IEEE) received the Habilitation and Ph.D. degrees in computer vision from the University of Strasbourg, Strasbourg, France.

He has been a Research Fellow at the Division of Informatics, The University of Edinburgh, Edinburgh, U.K., and a Lecturer at the Department of Computer Sciences, University of Glasgow, Glasgow, U.K. He has also been a Visiting Professor at the University of Louisville, Louisville, KY, USA; the University of Florence, Florence, Italy; the University of Lille, Lille, France; and the Korea Advanced Institute of Science and Technology, Daejeon, South Korea. He is currently a Professor at the Electrical Engineering and Computer Science Department, Khalifa University for Science and Technology, Abu Dhabi, United Arab Emirates. His main research area is 2-D/3-D image analysis and interpretation, where he has been leading several funded projects related to biometrics, medical imaging, remote sensing, and intelligent systems.

Dr. Werghi is a member of the IEEE Signal Processing Society, and IEEE Pattern Analysis and Machine Intelligence. He is an Associate Editor of the *EURASIP Journal for Image and Video processing*.