Modeling Extent-of-Texture Information for Ground Terrain Recognition

Shuvozit Ghose
Institute of Engineering and Management
Kolkata, India 700091
Email: shuvozit.ghose@gmail.com

Pinaki Nath Chowdhury
Indian Statistical Institute
Kolkata, India 700108
Email: contact@pinakinathc.me
www.pinakinathc.me

Partha Pratim Roy
Indian Institute of Technology
Roorkee, India 247667
Email: proy.fcs@iitr.ac.in
parimal.iitr.ac.in

Umapada Pal
Indian Statistical Institute
Kolkata, India 700108
Email: umapada@isical.ac.in
www.isical.ac.in/∼umapada

***Abstract*—Ground Terrain Recognition is a difficult task as the context information varies significantly over the regions of a ground terrain image. In this paper, we propose a novel approach towards ground-terrain recognition via modeling the Extent-of-Texture information to establish a balance between the order-less texture component and ordered-spatial information locally. At first, the proposed method uses a CNN backbone feature extractor network to capture meaningful information of a ground terrain image, and model the extent of texture and shape information locally. Then, the order-less texture information and ordered shape information are encoded in a patch-wise manner, which is utilized by intra-domain message passing module to make every patch aware of each other for rich feature learning. Next, the Extent-of-Texture (EoT) Guided Inter-domain Message Passing module combines the extent of texture and shape information with the encoded texture and shape information in a patch-wise fashion for sharing knowledge to balance out the order-less texture information with ordered shape information. Further, Bilinear model generates a pairwise correlation between the order-less texture information and ordered shape information. Finally, the ground-terrain image classification is performed by a fully connected layer. The experimental results indicate superior performance of the proposed model[1] over existing state-of-the-art techniques on publicly available datasets like DTD, MINC and GTOS-mobile.**

I. Introduction

Ground terrain recognition is a popular area of research in the context of computer vision because of its widespread applications in robotics and automatic vehicular control [1]–[4]. In the field of autonomous driving [3], ground terrain classification is very important because certain types of terrain may negatively affect the movement of a robot. Similarly, the knowledge of surrounded terrain information may help a robot to modify the course of its action during autonomous navigation [2], [5]. The goal of ground terrain recognition is quite similar to that of object recognition, but various factors have made the ground terrain recognition quite a challenging task. Firstly, real-world ground terrain images usually have highly complicated terrain surfaces and may not have any obvious feature or edge points. Moreover, the terrain surface may be as highly complex as the surface of Earth. Secondly, many class boundaries of the ground terrain images are ambiguous. For example, the class *"leaves"* is similar to *"grass"*, whereas the grass images contain a few leaves. Similarly, *"asphalt"* class is similar to *"stone-asphalt"* which is an aggregate mixture of stone and asphalt. Finally, the context information varies significantly over the regions of a ground terrain image, like some local regions possess significant texture information, while shape information is more dominant at some other parts.

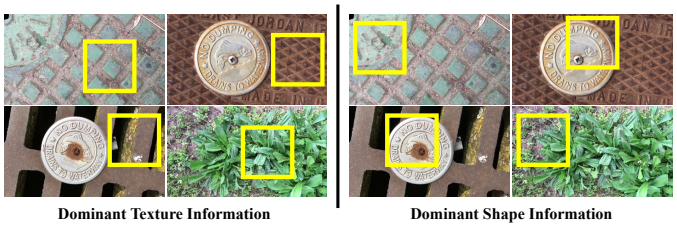

Fig. 1. Examples of real-world ground terrain image. Left image shows the texture dominant local regions (yellow box), while the right shows shape dominant local regions.

Traditionally, ground terrain images are filtered with a set of handcrafted filter banks [6]–[9], followed by grouping outputs into bag-of-words or texton histograms for the purpose of ground terrain recognition. Later, Cimpoi *et al.* [10] developed deep filter banks based on the convolution layers of a deep model for texture recognition, description, and segmentation. A notable contribution of their work was the introduction of FV-CNN that replaced the handcrafted filter banks with pre-trained convolutional layers for the feature extraction. Next, Zhang *et al.* [11] introduced a Deep Texture Encoding Network with an encoding layer integrated on top of the convolution layers that could encode the texture information of the ground terrain and material surface images into order-less texture representation. This order-less representation was later used by the classifier for ground terrain and material recognition. The main drawback of their work was that they ignored shape information of the ground terrain and material surface images. To overcome this issue, Xue *et al.* [12] further presented Deep Encoding Pooling Network which integrated order-less texture details and global spatial information for the task of ground terrain recognition. But, from the figure 1, we can see that most real-world ground terrain images show

[1]The source code of the proposed system is publicly available at https://github.com/ShuvozitGhose/Ground-Terrain-EoT

wide variations in texture and shape information at different local regions in an image. While bounding boxes of the left image of figure 1 show the texture dominant local regions, the bounding boxes of the right image of figure 1 show the shape dominant local region. Thus, the classification of such realistic ground terrain images requires a more local level modeling of texture and shape information. For this reason, we propose a novel approach towards ground-terrain recognition via modeling the extent of texture information to establish a balance between the order-less texture and ordered-spatial information locally. We first use a CNN backbone feature extractor network to capture the meaningful information of the ground terrain image. Then, we model the extent of texture and shape information locally. Unlike [12], [13], we encode the order-less texture information and ordered-spatial information patch-wise. Next, we utilize Intra-domain Message passing mechanism to make every patch aware of each other for rich feature learning. Subsequently, we combine extent of texture information with encoded texture information, and the extent of shape information with the encoded shape information patch-wise to establish a more local level balance of the texture and shape information. Furthermore, we exploit Extent-of-Texture Guided Inter-domain Message passing module, for sharing knowledge about the other domain to balance out the order-less texture information with ordered shape information patch-wise. Next, we aggregate all the patches to get the global order-less texture and ordered shape information of the ground terrain image. Furthermore, a Bilinear model captures pairwise correlation between the order-less texture and ordered shape information. Finally, a classifier is used to classify the ground terrain image. The contribution of this paper is as follows:

- We propose a novel approach towards ground-terrain recognition by modeling the extent of texture information to establish a balance between the order-less texture and ordered-spatial information locally.
- We introduce Intra-domain Message passing mechanism in the context of ground terrain recognition to make every local region aware of each other for rich feature learning.
- We present Extent of texture Guided (EoT) Inter-domain Message passing module in the context of ground terrain for sharing knowledge about the other domain to balance out the order-less texture information with ordered shape information locally.
- Our approach shows a superior classification accuracy on DTD, MINC and GTOS-mobile datasets as compared to the previous state-of-the-arts methods.

The remaining of this paper is organized as follows. In section II, we discuss some relevant works in the field of ground terrain recognition and graph convolutional neural networks. The proposed framework is detailed in Section III. Section IV describes the datasets, implementation details, and experimental results. Section V concludes the paper.

II. Related Works

Terrain Recognition is a well-known problem for decades in pattern recognition and computer vision community due to its crucial applications in the field of robotics. Earlier, the classical models were developed on geometric features, and Curvature-Based Approaches were exploited extensively in the context of terrain recognition. Goldgof *et al.* [14] used a Gaussian and mean curvature profile for extracting special points on the terrain, and compared these specials points with the points of maximum and minimum curvature to recognize the particular regions of the terrain. Instead of using single geometrical feature, Yu and Yuan [15] exploited multi-features including geometrical feature and color feature to classify terrain from ladar data for autonomous navigation. Based on the lighting direction of texture features, Chantler *et al.* [16] developed a probabilistic model which was robust to lighting direction and could classify the texture samples by comparing the likelihoods of each candidate with their estimated lighting. Further, Andreas *et al.* [17] exploited Lissajouss ellipses to develop a classifier that could classify surface textures images under unknown illumination tilt angles. Manduchi *et al.* [2] introduced an obstacle detection technique based on stereo range measurements and proposed a color-based classification system to label the detected obstacles according to a set of terrain classes. One of the key features of this method was that it did not rely on typical structural assumption on the scene such as the presence of a visible ground plane for terrain classification. Cula and Dana [7] constructed a compact representation with the help of bidirectional texture function which captures the underlying statistical distribution of features in the image texture as well as the variations in this distribution with viewing and illumination direction. Thereafter, this compact representation was used by texture classifier, to classify texture images of unknown viewing and illumination direction efficiently. Leung and Malik [8] developed 3D textons on the basis of the textural appearance of material surfaces. Based on local geometric and photometric properties of the tiny surface patches of the material, the main idea was to construct a 3D texture vocabulary. Finally, a unified model was proposed to represent and recognize visual appearance of materials from the 3D textons vocabulary. Based on the Bidirectional Feature Histograms, Cula and Dana [18] designed a 3D texture recognition method which employed the Bidirectional Feature Histograms as the surface model. The Bidirectional Feature Histograms captured the variation of the underlying statistical distribution of local structural image features, as the viewing and illumination conditions were changed; and classified surfaces based on a single texture image of unknown imaging parameter. Varma and Zisserman [19] presented a statistical approach for texture classification from single images under unknown viewpoint and illumination. In this method, texture was modelled by the joint probability distribution of filter responses and was represented by the frequency histogram of filter response cluster centers. Bhunia *et al.* [20] used a generative approach towards texture retrieval.

Later, Cimpoi *et al.* [10] developed deep filter banks based on the convolution layers of a deep model for texture recognition, description, and segmentation. Instead of focusing on

texture instance and material category, they proposed a human-interpretable vocabulary of texture attributes to describe common texture patterns. One of the key contributions of this work was the application of deep features to image regions and transferred features from one domain to another. Zhang *et al.* [21] proposed deep encoder-decoder model with near-to-far learning strategy for the purpose of terrain segmentation. Next, Zhang *et al.* [11] introduced Deep Texture Encoding Network with an encoding layer integrated on top of the convolution layers that could encode the texture image into order-less texture representation. This order-less representation was later used by a classifier for texture and material recognition. Xue *et al.* [12] presented Deep Encoding Pooling Network which integrated order-less texture details with local spatial information for the task of ground terrain recognition. The framework learned a parametric distribution in feature space in a fully supervised manner and gave the distance relationship among classes to implicitly represent ambiguous class boundaries.

On the other hand, Graph based networks have gained popularity in many fields like machine translation [22], text recognition [23] and learning sentence representation [24]. Though convolution Neural Networks (CNN) have an amazing capability of feature extraction, their applications are limited to fixed grid-like structure. On the contrary, graph convolutional networks provide a simple alternative of feature extraction for the arbitrarily structures. The graph based feature extraction was first presented by Frasconi *et al.* [25] and Sperduti *et al.* [26]. They used recursive neural network on directed acylic graphs for the purpose of feature extraction. Later, Gori *et al.* [27] and Scarcelli *et al.* [28] proposed Neural Networks which extended the idea of recursive networks to both cyclic and acyclic types of graph structure. Recently, Velickovic *et al.* [29] proposed Graph Attention network which introduced the concept of attention mechanism in the context of Graph neural networks. Graph neural networks outputs a hidden representation for each node by attending to its neighbourhood nodes in coherence with a self-attention strategy. In the section III-D, we have exploited a graph attention network to facilitate every patch to attend to every other patch for rich feature learning.

III. Proposed Framework

A. Overview

Contrary to the existing approaches [12], [13] that neglect modeling of Extent-of-Texture (EoT) information, we propose a novel approach towards ground-terrain recognition by modeling EoT information to establish a balance between order-less texture component and ordered-spatial information. Our key observation is that the extent of texture information varies significantly over the regions of an image, such as some local regions possess significant texture information; while shape information is more dominant at some other parts. Therefore, we follow a patch based feature extraction approach in order to balance between the Texture ($\mathcal{T}$) and Shape ($\mathcal{S}$) domain locally. Overall, our framework could be grouped into five steps: (i) We introduce a novel way of modeling the EoT information. (ii) Off-the-shelf texture and shape feature extractors are employed to obtain patch-wise feature representation in each domain - $\mathcal{T}$ and $\mathcal{S}$. (iii) Intra-domain message passing mechanism is used to make every patch feature aware of the each other for rich feature learning. (iv) Thereafter, the patch feature from both $\mathcal{T}$ and $\mathcal{S}$ domains are combined guided by the EoT information. (v) Finally, we aggregate the patch features, followed by a bilinear operation to fuse two domains followed by two fully-connected layers for final classification.

B. Modeling Extent-of-Texture (EoT) Information

Extent-of-Texture (EoT) modeling is the primary step in our architecture. Let the ground terrain image be $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, where H and W are the height and width of the ground terrain image. To capture the order-less texture information and ordered-spatial information effectively, we have used a backbone CNN feature extractor network $G(\cdot)$ that takes $\mathbf{I}$ as an input and outputs latent feature representation $\mathbf{Z}$. Thus,

$$\mathbf{Z} = G(\mathbf{I}; \theta_G) \tag{1}$$

where $\mathbf{Z} \in \mathbb{R}^{8 \times 8 \times 512}$ and θ_G is the parameter of the network. In general, $\mathbf{Z}$ is the latent feature representation of the local order-less texture information and ordered-spatial information. We adapt modified ResNet-18 architecture as CNN feature extractor network with rectified non-linearity (ReLU) activation after each layer. To model Extent-of-Texture (EoT) information locally, we perform patch-extraction on $\mathbf{Z}$ using a sliding window mechanism where the window size and stride is chosen as (3×3) and 1 respectively. The patch-extraction operation generates $\psi = \{\psi_1, \psi_2, \psi_3 ..... \psi_k\}$ patches, where $\psi_{\mathbf{i}} \in \mathbb{R}^{3 \times 3 \times 512}$ and k is the number of patches. In our model, the value of k is 36. Next, an average pooling operation with kernel size (3×3) is performed on ψ which results in $\psi^* = \{\psi_1^*, \psi_2^*, \psi_3^* ..... \psi_k^*\}$ patches, where $\psi_{\mathbf{i}}^* \in \mathbb{R}^{1 \times 1 \times 512}$. Let $\mathbf{X} = \{x_1, x_2, x_3 ..... x_k, \}$, where x_i denotes the central region of the $\psi_{\mathbf{i}}$ patch e.g. $x_i = \psi_i[2; 2; :]$ and $x_i \in \mathbb{R}^{1 \times 1 \times 512}$. The cosine similarity between ψ^* and $\mathbf{X}$ describes the order-less texture information $\mathcal{T}$, where $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3 ..... \mathcal{T}_k\}$ and $\mathcal{T}_i$ denotes the order-less texture information of the i^{th} patch. Therefore,

$$\psi_i^* = AvgPool(\psi_i, 3) \tag{2}$$

$$\mathcal{T}_i = \frac{\psi_i^* \cdot x_i}{||\psi_i^*||_2 \ ||x_i||_2} \tag{3}$$

$$\mathcal{T}_i = \frac{\mathcal{T}_i - \mathcal{T}_{min}}{\mathcal{T}_{max} - \mathcal{T}_{min}} \tag{4}$$

Here, $||\cdot||_2$ represents L2-norm, $\mathcal{T}_{min}$ and $\mathcal{T}_{max}$ are the fixed value of 0.5 and 0.9 respectively. In our experiment, we have observed that the cosine distance between ψ^* and x_i varies between 0.5 and 0.9. We perform a normalization operation on $\mathcal{T}$ using equation 4 for readjusting the range from 0 to 1. In this context, a high value of $\mathcal{T}$ indicates the presence of greater extent of the order-less texture information , whereas a small value of $\mathcal{T}$ represents higher shape information. Let the

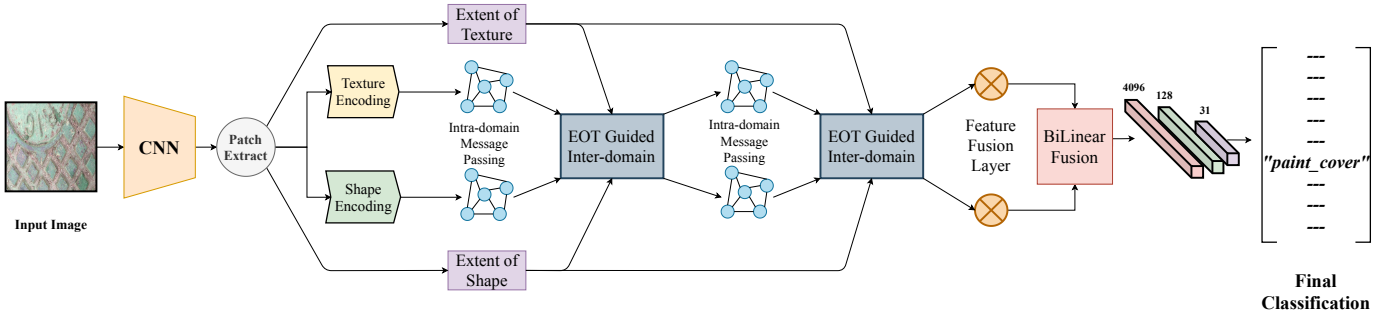

Fig. 2. Our proposed Method. The CNN backbone feature extractor network reduces input to $Z \in \mathbb{R}^{8\times8\times512}$. The patch extraction generates 36 patches from Z. These patches are later used in Extent of texture modeling, texture encoding, shape encoding and Extent of shape modeling as shown in the diagram. The texture and shape encoding are passed through Intra-domain Message Passing and merged with Extent of texture and shape in the EoT Guided Inter-domain Message Passing module. The output texture and shape patches are combined in the feature fusion layers and then passed through Bilinear fusion layer. Finally, classifier is used.

ordered shape information $\mathcal{S}$, where $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3 \ldots\ldots \mathcal{S}_k\}$ and $\mathcal{S}_i$ denotes the ordered-spatial information of the i^{th} patch. Then,

$$\mathcal{S}_i = 1 - \mathcal{T}_i \tag{5}$$

C. Texture and Shape Encoding Module

In our architecture, we have used an off-the-shelf texture encoding layer proposed by Zhang *et al.* [13] for texture encoding. The texture encoding layer integrates the entire dictionary learning and visual encoding pipeline to provide an order-less representation for texture modeling. For each $\psi_i \in \psi$, let $\delta = \{\delta_1, \delta_2, \ldots, \delta_m\}$ be M visual descriptors and $\mathbf{\Lambda} = \{\lambda_1, \lambda_2, \ldots \lambda_n\}$ be N learned codewords. We calculate the residual vectors $r_{i,j} = \delta_i - \lambda_j$ where, $i = 1, 2, \ldots m$ and $j = 1, 2, \ldots n$. The residual encoding corresponding to δ_j is calculated as:

$$t_j = \sum_{i=1}^{M} w_{i,j} r_{i,j} \tag{6}$$

$$\text{where}, w_{i,j} = \frac{exp(-s_j||r_{i,j}||^2)}{\sum_{l=1}^{N} exp(-s_l||r_{i,l}||^2)} \tag{7}$$

Here $s_1, s_2, \ldots s_N$ are learnable smoothing factors for each cluster center $\lambda_j \in \mathbf{\Lambda}$. Let $\mathbf{E} = \{t_1, t_2, \ldots t_n\}$ be the encoded order-less texture information, t_i where having a dimension of $\mathbb{R}^{512\times N}$ is fed to a fully connected layer $f_c : \mathbb{R}^{4096} \rightarrow \mathbb{R}^F$ to give $\mathbf{E_t} = \{e_{t1}, e_{t2}, \ldots e_{tn}\}$, where $e_{ti} \in \mathbb{R}^F$ and e_{ti} represents the final encoded order-less texture information of the i^{th} patch. Here, N is the number of learned codewords and F is the size of output of the fully connected layer. In our architecture, we have used $N = 8$ and $F = 64$.

For shape encoding, first we have performed average pooling on each $\psi_i \in \psi$ that converts $\mathbb{R}^{3\times3\times512} \rightarrow \mathbb{R}^{1\times1\times512}$. Next, a fully connected layer maps $f_c : \mathbb{R}^{512} \rightarrow \mathbb{R}^F$ to give $\mathbf{E_s} = \{e_{s1}, e_{s2}, \ldots e_{sn}\}$, where $e_{si} \in \mathbb{R}^F$ and e_{si} represents the final encoded ordered shape information of the i^{th} patch.

D. Intra-domain Message Passing

We develop an Intra-domain Message Passing module to make every patch feature aware of each other for rich feature learning. A graph attention layer (GAT) [29] is employed to allow every patch to attend to every other patches. For this reason, we have designed two separate complete graphs, each having k nodes, where the degree of each node is k. The i^{th} node of the E_t representational graph represents the e_{ti} patch of E_t, and the i^{th} node of the E_s representational graph represents the e_{si} patch of E_s. So, graph $\mathbf{E_t} = \{e_{t1}, e_{t2}, \ldots e_{tk}\}$, where $e_{ti} \in \mathbb{R}^{\mathbb{F}}$ and F is the number of features in each node (in our case, $F = 64$). We compute the hidden representation for each e_{ti} by attending to all e_{tj} where $j = 1, 2, \ldots k$ using self-attention mechanism. To transform e_{ti} to higher-level features, a shared linear transformation matrix $\mathbf{W} \in \mathbb{R}^{F\times F}$ is applied to each e_{ti}. This is followed by encompassing self attention over all nodes using a shared attention mechanism orchestrated by a vector $\tilde{\mathbf{a}} \in \mathbb{R}^{2F}$. The final representation e'_{ti} for each e_{ti} is computed as follows:

$$e_{i,j} = LeakyReLU(\vec{\mathbf{a}}^T[\mathbf{W}e_{ti}||\mathbf{W}e_{tj}]) \tag{8}$$

$$\alpha_{i,j} = softmax_j(e_{i,j}) = \frac{exp(e_{i,j})}{\sum_{l=1}^{k} exp(e_{i,l})} \tag{9}$$

$$e'_{ti} = \sigma(\sum_{j=1}^{k} \alpha_{i,j} \mathbf{W} e_{tj}) \tag{10}$$

where $||$ represents concatenation of two vectors, σ represents a Relu non-linear activation function and k is the number of nodes connected to i. We use multi- head attention to stabilise the self-attention mechanism in GAT layers. The GAT layer outputs $\mathbf{E'_t} = \{e'_{t1}, e'_{t2}, \ldots e'_{tk}\}$ for $\mathbf{E_t}$ representational graph. Similarly, we obtain $\mathbf{E'_s} = \{e'_{s1}, e'_{s2}, \ldots e'_{sk}\}$ for $\mathbf{E_s}$ representational graph.

E. EoT Guided Inter-domain Message Passing

In this section, first we have combined the extent of texture information $\mathcal{T}$ with E'_t and the extent of shape information $\mathcal{S}$ with E'_s to establish a local level balance of the order-less texture and ordered shape information. Later, the EoT Guided Inter-domain Message Passing module is used for sharing

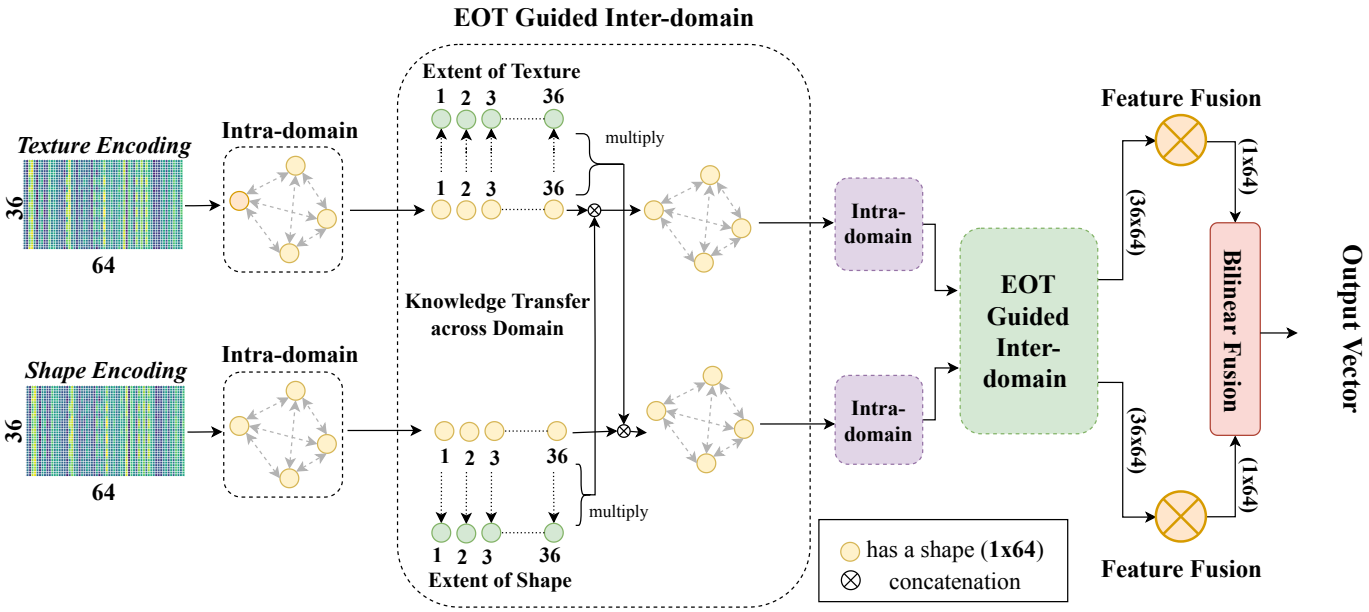

Fig. 3. Architecture of the EoT Guided Inter-domain Message Passing. Feature map of the 36 patches from Texture encoding and shape encoding layer, having shape $\mathbb{R}^{36\times64}$ is given to Intra-domain Message Passing module for information exchange among the patches. From the resulting 36 features for texture and shape, a fused vector is calculated using weighted average where extent of texture/shape serves as the weights. This fused vector is concatenated with each of the 36 feature vectors from the other domain followed by a fully connected layer to reduce dimensionality of each vector back to 64. We further enrich the representation using a second Intra-domain and EoT Guided Inter-domain Message Passing module before fusing the 36 vector of each domain in the Fusion layer followed by Bilinear fusion.

knowledge about the other domain to balance out the order-less texture information with ordered-spatial information as depicted in the figure 3. Mathematically,

$$r_t^{'} = \sum_{j=1}^{k} \mathcal{T}_j e_{tj}^{'} \tag{11}$$

$$r_s^{'} = \sum_{j=1}^{k} \mathcal{S}_j e_{sj}^{'} \tag{12}$$

where k is the number of patches, $r_i^{'t}$ and $r_i^{'s}$ are the representative vectors of texture and shape domains respectively. $\mathcal{T}_j \in \mathcal{T}$ and $\mathcal{S}_j \in \mathcal{S}$ are described in section III-B, whereas $e_{tj}^{'} \in E_t'$ and $e_{sj}^{'} \in E_s'$ are described in section III-D. Let $\mathbf{W}_t$ and $\mathbf{W}_s$ are a pair of shared learnable weights, the EoT Guided Inter-domain Message Passing module generates

$$\mathcal{T}_i^{'} = \mathbf{W}_t[e_{ti}^{'}||r_s^{'}] \tag{13}$$

$$\mathcal{S}_i^{'} = \mathbf{W}_s[e_{si}^{'}||r_t^{'}] \tag{14}$$

Where, $\mathcal{T}_i^{'}$ and $\mathcal{S}_i^{'}$ are the balanced texture and shape information of the i^{th} patch respectively. We further enrich this representation using a succession of Intra-domain Message Passing module and EoT Guided Inter-domain Message Passing module to derive $\mathcal{T}^{''} = \{\mathcal{T}_1^{''}, \mathcal{T}_2^{''}, \ldots \mathcal{T}_k^{''}\}$ and $\mathcal{S}^{''} = \{\mathcal{S}_1^{''}, \mathcal{S}_2^{''}, \ldots \mathcal{S}_k^{''}\}$, where each i^{th} patch having a dimension of $\mathbb{R}^{K\times F}$.

F. Fusion Layer and Bilinear Model

In the feature fusion layer, we have aggregated the all $\mathcal{T}_i^{''}$ and $\mathcal{S}_i^{''}$ patches for $i = 1, 2, \ldots k$ to get global order-less texture representation $\mathcal{T}_{fuse}$ and ordered shape representation $\mathcal{S}_{fuse}$ respectively. Let $\mathcal{T}^{''}$ and $\mathcal{S}^{''}$ be two matrices of size $\mathbb{R}^{k\times F}$, where each column of $\mathcal{T}^{''}$ and $\mathcal{S}^{''}$ represents $\mathcal{T}_i^{''}$ and $\mathcal{S}_i^{''}$ patches respectively. Then,

$$\mathcal{T}^{fuse} = \mathcal{V}_t^T \mathcal{T}^{''} \tag{15}$$

$$\mathcal{S}^{fuse} = \mathcal{V}_s^T \mathcal{S}^{''} \tag{16}$$

Where, $\mathcal{V}_t$ and $\mathcal{V}_s$ are learnable weight vectors, each having a size $\mathbb{R}^k$. Next, $\mathcal{T}_{fuse}$ and $\mathcal{S}_{fuse}$ are passed through the Bilinear fusion model [12] to balance the texture and shape information. Let f_{out} be the final output of the Bilinear fusion layer. Then,

$$f_{out} = \sum_{i=1}^{F}\sum_{j=1}^{F} \omega_{i,j} \mathcal{T}_i^{fuse} \mathcal{S}_j^{fuse} \tag{17}$$

where, $f_{out} \in \mathbb{R}^{4096}$ and $\omega_{i,j}$ is a learnable weight to balance the interaction between texture and shape information. Here, the f_{out} captures a pairwise correlation between the order-less texture and ordered shape information. Finally, f_{out} is passed through a sequence of two fully-connected layers to generate the classification result of the ground-terrain image as depicted in the figure 2.

IV. Experiment

A. Datasets

GTOS-mobile [12] dataset, which is a modified version of the original GTOS database [13], is used for evaluation of our approach. GTOS-mobile dataset consists of 31 classes captured using mobile phones consisting of 81 videos from which 6066 frames were used as a test set. We follow the train-test split as described in Xue *et al.* [12]. GTOS-mobile differs from the original GTOS dataset where the classes: dry grass, ice mud, mud-puddle, black ice and snow were removed. Further similar classes such as asphalt and metal are merged. Hence, unlike the original GTOS dataset consisting of 40 classes, the GTOS-mobile dataset has 31 classes as defined in Xue *et al.* [12]. The resolution of videos was maintained at 1920×1080 followed by resizing the shorter edge to 256. Hence each image has a resolution of 455×256.

Describable Texture Dataset(DTD) [30] consists of 47 classes with each class consisting of 120 instances. The images are collected from Google and Flickr using key attributes for each class and a list of joint attributes. Annotation of images were carried out using Amazon Mechanical Turk in several iterations.

Materials in Context Database - 2500(or MINC-2500) [31] is a subset of the original MINC dataset with 2500 instances in every class/category where each image has been resized to 362×362.

B. Implementation Details

We have implemented the entire model in PyTorch [32] and executed the code on a server having Nvidia Titan X GPU with 12 GB of memory. L2 loss was employed as the objective function during experimentation. We have adapted a modified ResNet-18 architecture as CNN feature extractor network with rectified non-linearity (ReLU) activation after each layer. The full resolution input image is resized into different scales followed by cropping the center 256×256 regions because such a pre-processing step simulates observing the GTOS dataset at different distances. During training, we have initialized our CNN backbone feature extractor network using pre-trained ImageNet [33] weights. Following previous works in texture recognition [12], we follow single-scale training and multi-scale training with identical data augmentation and training procedures. For single-scale training mechanism, an input image is resized into 286×286 and crop a region of size 256×256 from the center. Multi-scale training incorporates randomly resizing an input image into 256×256, 384×384, and 512×512 followed by cropping a region of 256×256 from the image center. The training data additionally undergoes a 50% chance of horizontal flip with random changes in brightness, contrast and saturation. Images in both training and testing sets, finally undergo per-channel normalisation before being fed to the model. Training is performed for 30 epochs with a batch size of 128. We have used stochastic gradient descent (SGD) optimizer with learning rate 0.01, momentum 0.9, decay rate of 0.0001 while training our model in an end-to-end fashion.

C. Baselines Methods

To justify each design choice and their contribution, we design four alternative baselines.

Baseline-1 (B1): Following the existing DEP [12] method, we feed the features from convolutional layers to the texture and shape encoding layers. Output from the encoding layers are then merged together using Bilinear model [34], [35].

Baseline-2 (B2): We construct a baseline where features from convolutional layers are segmented into patches using patch extraction and fed to both texture encoding and global average pooling. After passing through texture and shape encoding layers, the feature vectors of patches corresponding to both texture and shape are merged using the aforementioned Fusion layer to get a global representative vector for texture and shape respectively. These vectors are then combined using Bilinear models for eventual classification.

Baseline-3 (B3): Here we apply Intra-domain Message Passing module using Graph Attention networks [29] to derive features from the texture and shape encoding layer. This is followed by Fusion and bilinear layer, similar to B2.

Baseline (B4): We replace the Intra-domain Message Passing module in baseline B3 with the EoT Guided Inter-domain Message Passing module to enable a better balance between order-less texture component with ordered shape information.

We compare the aforementioned baselines and Deep-TEN [13] with our proposed methodology to examine the contribution of each design choice. We maintain an identical training and evaluation procedure where ResNet-18 generates a feature map of dimension $8 \times 8 \times 512$. Experiments on GTOS-mobile dataset are shown in Table I which demonstrates that the proposed methodology improves upon Deep-TEN [13] by nearly 6% and DEP [12] by nearly 4%.

D. Performance Analysis

From Table I, we observe that B1 has achieved a significant improvement of $1.85\%(6.06\%)$ from Deep-TEN [13] for Single-scale (Multi-scale) setup by incorporating the spatial information instead of replying solely on texture encoding layer. Although B1 incorporated both texture and spatial information, it did not account for the local level variations of the order-less texture and ordered-spatial information. Using patch extraction in B2, we mitigate this limitation and further improve recognition performance over B1 by $1.74\%(1.6\%)$ for Single-scale (Multi-scale). While B2 was able to enhance recognition accuracy by taking a more local level approach through patch extraction, it did not correlate the patch information with each other to enrich features and hence, we observe an inferior performance as compared to B3. Using Intra-domain Message Passing to make every patch aware of each other via Graph-Attention layer resulted in an improvement of $0.74\%(0.53\%)$ over B2. In B4, replacing graph attention layer in B3 by the EoT Guided Inter-domain Message Passing mechanism resulted in an improvement of $1.12\%(0.58\%)$ over B2, indicating the significance of information exchange across different domains by the EoT Guided Inter-domain Message Passing module. While both B3 and B4 show considerable

TABLE I

COMPARISON OF **DEEP-TEN**, BASELINE **B1, B2, B3** AND **B4** WITH THE PROPOSED METHODOLOGY FOR SINGLE SCALE AND MULTI SCALE TRAINING ON GTOS-MOBILE [12] DATASET USING A PRE-TRAINED RESNET-18 MODULE AS THE CONVOLUTIONAL LAYER. BASELINE B1 IS SIMILAR TO DEEP ENCODING POOLING NETWORK (DEP) BY XUE *et al.* [12].

	Deep-TEN [13]	**B1 [12]**	**B2**	**B3**	**B4**	**Proposed Method**
Single Scale	74.22	76.07	77.81	78.55	78.93	**80.39**
Multi Scale	76.12	82.18	83.78	84.31	84.36	**85.71**

TABLE II

COMPARING OUR METHOD WITH SEVERAL STATE-OF-THE-ART METHODS ON DESCRIBABLE TEXTURES DATASET (DTD) AND MATERIALS IN CONTEXT DATABASE (MINC)

Method	DTD [30]	MINC-2500 [31]
FV-CNN [36]	72.3	63.1
Deep-TEN [13]	69.6	80.4
DEP [12]	73.2	82.0
Proposed Method	**75.7**	**85.3**

improvements over B2, it can be observed from Table I that B4 performs slightly better than B3 by $0.38\%(0.05\%)$. This implies that sharing knowledge across texture and shape domains before fusing the local texture and shape features in the Fusion layer is slightly more beneficial as compared to exchanging knowledge among similar entities followed by merging texture and spatial information in the Fusion layer.

E. Comparison on DTD and MINC datasets

To ensure an equal comparison, we replace ResNet-18 with ResNet-50 and include a (1×1) convolutional layer to convert the number of output channels from 2048 to 512. Evaluation on Describable Textures Database (DTD) [30] and Materials in Context Database (MINC) [31] expresses the generalisability of the proposed method. From Table II, we can observe that for DTD (MINC-2500) dataset, the proposed method shows $2.5\%(3.3\%)$ improvement as compared to state-of-the-art methods. Additionally, a multi-scale training mechanism is likely to improve fine-grained visual recognition results for all methods as demonstrated by Lin *et al.* [35]. Although, we do not include a multi-scale training setup in our experimental section, one can expect enhancement of performance for both the proposed method as well as existing baselines by using multi-scale training.

V. CONCLUSION

In this paper, we have proposed a novel approach towards ground-terrain recognition via modeling the extent of texture information to establish a balance between the order-less texture component and ordered-spatial information locally. The driving idea of our architecture is the modeling of context information locally. The proposed framework is simple and easy to implement. It is capable of detecting ground terrain in the real-world scenario. We demonstrate the effectiveness of our system by conducting experiments on publicly available ground terrain datasets.

REFERENCES

[1] A. Angelova, L. Matthies, D. Helmick, and P. Perona, "Fast terrain classification using variable-length representation for autonomous navigation," in *CVPR*, 2007, pp. 1–8.

[2] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," *Autonomous robots*, vol. 18, no. 1, pp. 81–102, 2005.

[3] D. F. Wolf, G. S. Sukhatme, D. Fox, and W. Burgard, "Autonomous terrain mapping and classification using hidden markov models," in *ICRA*, 2005, pp. 2026–2031.

[4] M. Hebert and N. Vandapel, "Terrain classification techniques from ladar data for autonomous navigation," 2003.

[5] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. R. Bradski, "Self-supervised monocular road detection in desert terrain," in *Robotics: science and systems*, vol. 38, 2006.

[6] J. S. De Bonet, "Multiresolution sampling procedure for analysis and synthesis of texture images," in *ACM Annual Conference on Computer graphics and interactive techniques*, 1997, pp. 361–368.

[7] O. G. Cula and K. J. Dana, "Compact representation of bidirectional texture functions," in *CVPR*, vol. 1, 2001, pp. I–I.

[8] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International journal of computer vision*, vol. 43, no. 1, pp. 29–44, 2001.

[9] S. Konishi and A. L. Yuille, "Statistical cues for domain specific image segmentation with performance analysis," in *CVPR*, vol. 1, 2000, pp. 125–132.

[10] M. Cimpoi, S. Maji, I. Kokkinos, and A. Vedaldi, "Deep filter banks for texture recognition, description, and segmentation," *International Journal of Computer Vision*, vol. 118, no. 1, pp. 65–94, 2016.

[11] H. Zhang, J. Xue, and K. Dana, "Deep ten: Texture encoding network," in *CVPR*, 2017, pp. 708–717.

[12] J. Xue, H. Zhang, and K. Dana, "Deep texture manifold for ground terrain recognition," in *CVPR*, 2018, pp. 558–567.

[13] H. Zhang, J. Xue, and K. Dana, "Deep ten: Texture encoding network," *arXiv preprint arXiv:1612.02844*, 2016.

[14] D. B. Goldgof, T. S. Huang, and H. Lee, "A curvature-based approach to terrain recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 11, no. 11, pp. 1213–1217, 1989.

[15] C.-p. Yu and X. Yuan, "Terrain classification for autonomous navigation using ladar sensing," in *International Conference on Information Science and Engineering*, 2009, pp. 1467–1470.

[16] M. J. Chantler, G. McGunnigle, A. Penirschke, and M. Petrou, "Estimating lighting direction and classifying textures." in *BMVC*, 2002, pp. 1–10.

[17] A. Penirschke, M. J. Chantler, and M. Petrou, "Illuminant rotation invariant classification of 3d surface textures using lissajouss ellipses," in *International Workshop on Texture Analysis and Synthesis*, 2002, pp. 103–108.

[18] O. G. Cula and K. J. Dana, "3d texture recognition using bidirectional feature histograms," *International Journal of Computer Vision*, vol. 59, no. 1, pp. 33–60, 2004.

[19] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *International journal of computer vision*, vol. 62, no. 1-2, pp. 61–81, 2005.

[20] A. K. Bhunia, S. R. K. Perla, P. Mukherjee, A. Das, and P. P. Roy, "Texture synthesis guided deep hashing for texture image retrieval," in *WACV*, 2019, pp. 609–618.

[21] W. Zhang, Q. Chen, W. Zhang, and X. He, "Long-range terrain perception using convolutional neural networks," *Neurocomputing*, vol. 275, pp. 781–787, 2018.

[22] L. D. Jianpeng Cheng and M. Lapata, "Long short-term memory-networks for machine reading," *arXiv preprint arXiv:1601.06733*, 2016.

[23] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2298–2304, 2015.
[24] C. N. d. S. M. Y. B. X. B. Z. Zhouhan Lin, Minwei Feng and Y. Bengio, "A structured self-attentive sentence embedding," *arXiv preprint arXiv:1703.03130*, 2017.
[25] M. G. Paolo Frasconi and A. Sperduti, "A general framework for adaptive processing of data structures," *IEEE transactions on Neural Networks*, vol. 9, no. 5, pp. 768–786, 1998.
[26] A. Sperduti and A. Starita, "Supervised neural networks for the classification of structures," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 714–735, 1997.
[27] G. M. Marco Gori and F. Scarselli, "A new model for learning in graph domains," in *IJCNN*, 2005, pp. 729–734.
[28] A. C. T. M. H. Franco Scarselli, Marco Gori and G. Monfardini, "The graph neural network," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.
[29] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," *ArXiv*, vol. abs/1710.10903, 2017.
[30] I. K. S. M. M. Cimpoi, S. Maji and A. Vedaldi, "Describing textures in the wild," in *CVPR*, 2014, pp. 3606–3613.
[31] N. S. S. Bell, P. Upchurch and K. Bala, "Material recognition in the wild with the materials in context database," in *CVPR*, 2015, pp. 3479–3487.
[32] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NeurIPS Autodiff Workshop*, 2017.
[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.
[34] J. B. Tenenbaum and W. T. Freeman, "Separating style and content," in *Advances in neural information processing systems*, 1997, pp. 662–668.
[35] A. R. T.-Y. Lin and S. Maji, "Bilinear cnn models for fine-grained visual recognition," in *ICCV*, 2015, pp. 1449–1457.
[36] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," in *CVPR*, 2015, pp. 3828–3836.