DEEP TEXTURE MANIFOLD FOR GROUND TERRAIN RECOGNITION

Nidhi H R
Dept of CSE ,JSSSTU
Mysuru, India
nidhi12032004@outlook.com

Neethushree G N
Dept of CSE ,JSSSTU
Mysuru, India
neethushreegn@gmail.com

Navyashree R
Dept of CSE ,JSSSTU
Mysuru, India
navyaravilatha@gmail.com

I Prashamsa Prabhu
Dept of CSE ,JSSSTU
Mysuru, India
iprashamsaprabhu@gmail.com

Sneha N G
Dept of CSE ,JSSSTU
Mysuru, India
snehang2021@outlook.com

Eshwari
Dept of CSE ,JSSSTU
Mysuru, India
eshwari5451@gmail.com

***Abstract*—We propose a texture network called Deep Encoding Pooling (DEP) for recognition of underground activities. Ground recognition is an important task for building and deploying a robot or unmanned vehicle in the outdoor environment. DEP's architecture integrates unordered textured details and local spatial information, and DEP's performance outperforms state-of-the-art operating methods. Guaranteed maintenance is provided by the GTOS library, which contains more than 30,000 images of 40 outdoor soil types. In order to measure realistically, we use images measured from mobile phones held in similar soils, not from existing GTOS data. This new test file, called GTOS-mobile, contains 81 videos from 31 ground types (such as grass, concrete, asphalt and sand). The results show that the performance is good not only for GTOS-mobile but also for many general documents (MINC and DTD). Using the discriminative features learned from this network, we create a new texture called the DEP manifold. We comprehensively study parameter distributions in feature space that provide relationships between classes and provide a way to implicitly represent fuzzy class boundaries.**

I. Introduction

Terrain identification Terrain is an important area for restoration. Searching computer vision for potential applications in autonomous driving and robot navigation. Detection by CNN has been successful in object recognition. CNN architecture balances relative space conservation. Information and aggregation (using convolutional layers). Spatial information (integration layer). This structure is designed for object recognition, scene perception and face recognition. It can be used in applications where spatial order is important for detection and classification. However, in tissue recognition. Use low-order components to create invariance in the spatial arrangement outside.

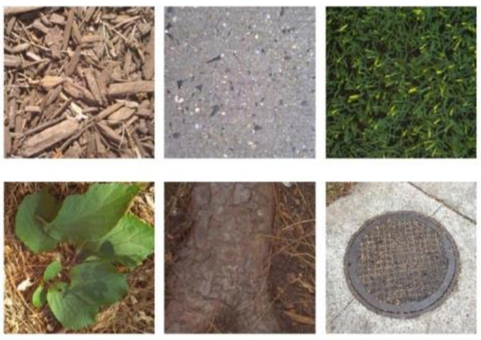


Figure 1: Homogeneous textures (upper row) compared to more common real-world instances with local spatial structure that provides an important cue for recognition .

In the classical approach to texture modelling, the image is after filtering with a set of hand-crafted filter banks, grouping output to text in histogram , or bag of words. Subsequently, Cimpoi et AI. Introduce FV-CNN already replaced a hand-crafted filter bank. Convolution layer trained for feature extraction, and achieving advanced results recently, Zhang et al. Introducing deep texture encryption networks (Deep-TEN) This tutorial brings dictionary and feature fusion approaches to the CNN pipeline of end-to-end material/texture recognition networks. Detection algorithms that focus on texture detail are suitable for images that only contain one substance. However, in "real-world images", homogeneous surfaces rarely fill the entire field of view, and many materials exhibit regular structures.

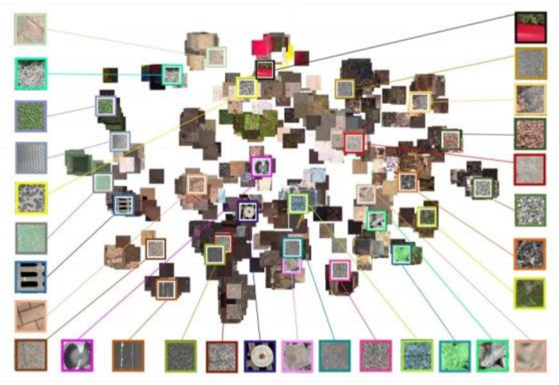


Figure 2: The result of texture manifold by DEP-manifold. Images with colour frames are images in the test set. The material classes are (from upper left to counter-clockwise): plastic cover, painted turf, turf, steel, stone-cement, painted cover, metal cover, brick, stone-brick, glass, sandpaper, asphalt, stone-asphalt, aluminium, paper, soil, mulch, painted asphalt, leaves, limestone, sand, moss, dry leaves, pebbles, cement, shale, roots, gravel and plastic. Not all classes are shown here for space limitations.

For texture recognition, the local spatial order is an important feature, because the level order is not completely low. This diagnosis is shown in Figure 1. As semantic segmentation balances local details with the context of the global scene. Design pixel detection . A network that balances both low-order components and ordered spatial information.

As shown in Figure 3, we introduce a deep encryption fusion (DEP) network that takes advantage of fewer orders. Representation and local spatial information for identification. The output from the convolution layer is fed to two features combine representation layers. Encryption layer and the global average of the integration layer. The encoding layer is used to capture visual details and the overall texture pattern. The intermediate fusion layer collects spatial information. The features of the coding layer and the global average pooling layer are processed with a bilinear model. Apply DEP to ground detection problems. GTOS extended dataset . The resulting network is as follows: It works great not only for GTOS but also for other applications. Common databases (MINC and DTD).

To distinguish the terrain, many class boundaries it is vague. For example, the "asphalt" class looks like this: "Stone asphalt" is an aggregate that is a mixture of stone and asphalt. The "leaf" class is similar to "grass". An example of a "leaf" sample image in the GTOS database there is grass in the background. Similarly, the image is grass it contains several leaves, so it is interesting to know that it is not this includes not only the class label, but also the position in the nearest equivalent class or manifold. To find the relationships, we introduce a new texture manifold method, DEP-manifold. It is presented between the newly captured images and the existing images in the dataset. Distributed random neighbourhood embedding t (t-SNE) provides two-dimensional embedding and Barnes-Hut t-SNE accelerates the original t-SNE from O(n^2) is O (n log n). t-SNE and Barnes-Hut are both t-SNE as a nonparametric embedding algorithm, there is no natural way to perform out-of-sample expansion. Parametric t-SNE and supervised t-SNE are deep Embedding neural networks in data to achieve non-linearity Parametric embedding inspired by this work, I introduce it to you. The embedding distribution is from the nonparametric embedding algorithm output generation and prediction of human behavior using deep neural networks. If you want to get the old coordinates of the texture image directly. This texture the manifold uses DEP network function and is called DEP-manifold.

The training set is the Ground Terrain Database (GTOS) . We use 31 classes of Earth images (over 30,000 images in the dataset). Instead of using GTOS images as a test dataset, we collect 81 ground videos of similar terrain classes in GTOS Mobile. Hold your phone in any light/scenery. Our motivations are: The training set (GTOS) is: It is obtained comprehensively (with known distances and Views (high resolution calibrated cameras) to gain scene knowledge. The test suite is obtained below. Very different and more realistic conditions (mobile cameras, handheld video, unedited recording). Education experiments with GTOS and GTOS-mobile enable evaluation of knowledge transfer in the network.

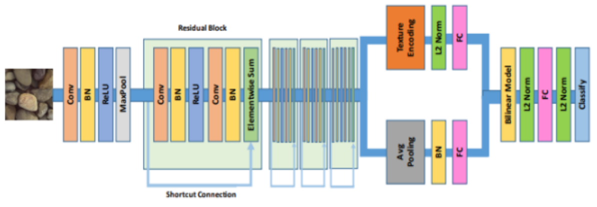


Figure 3: A Deep Encoding Pooling Network (DEP) for material recognition. Outputs from convolutional layers are fed into the encoding layer and global average pooling layer jointly and their outputs are processed with a bilinear model.

RELATED WORK:

The bilinear models introduced by Tenenbaum and Freeman processes two factors mainly.linetal. introduced the Bilinear CNN models that use outer product of feature maps from convolutional layers of two CNNs and reach state-of-the-art for fine grained visual recognition. However this method has two drawbacks. First, pairs of features maps have compatible feature dimensions, i.e. the same height and width are required by the bilinear models. The second disadvantage is the computational complexity this technique computes the outer product at each feature map location. We use the bilinear models to get the maximum benefits and overcome the disadvantages. Then output from fully connected layers can be treated as vectors, and there is no dimensionality restriction for the outer product of two vectors.

Deep Encoding Pooling Network

Encoding Layer: The texture encoding layer blends the entire dictionary learning and visual encoding pathways into a single CNN layer, The encoding layer acts as a global feature pooling above the yielding in an order less representation for texture modelling. The encoding layer acts as a global feature pooling above the convolutional layers. Here we briefly describe our work. Let X = {x1, ...xm} be M visual descriptors = {c1, ...cn} is the code book with N learned code words. .We calculate the residual vector rij by the formula rij = xi−cj „ where i = 1...m and j = 1...n.The residual encoding for code word cj can be represented as,

$$e_j = \sum_{i=1}^{N} w_{ij} r_{ij}, \qquad (1)$$

where wij is the assigning weight for residual vector rij and is given by,

$$w_{ij} = \frac{\exp(-s_j \|r_{ij}\|^2)}{\sum_{k=1}^{m} \exp(-s_k \|r_{ik}\|^2)}, \qquad (2)$$

where,s1, ...sm are learnable smoothing factors.the visual descriptors X are pooled into a set of N residual encoding vectors E = {e1, ...en}.The encoding layer can capture more texture details by increasing the number of learnable code.

Bilinear Models: Bilinear models are two-factor models where the outputs are linear in one factor if the other factor is constant.Bilinear models use parameters to balance the roles of the two components.Let at and bs represent the material texture information and spatial information with vectors of parameters and with dimensionality I and J. The bilinear function Y ts is given by

$$Y^{ts} = \sum_{i=1}^{I} \sum_{j=1}^{J} w_{ij} a_i^t b_j^s, \qquad (3)$$

where wij is a learnable weight to balance the interaction between material texture and spatial information.The outer product representation reflects a pairwise relationship between material texture encodings and spatial observation structure.

Deep Encoding Pooling (DEP) Network : The Deep Encoding Pooling (DEP) Network is shown in figure 3.we use convolutional layers with non-linear layers from

ImageNet [9] pre-trained CNNs as feature extractors.Convolutional layer outputs are together fed into the texture encoding layer and the global average pooling layer.Outputs from the texture encoding layer preserve texture details, while outputs from the global average pooling layer preserve local spatial information.The dimension of outputs from the texture encoding layer is determined by the codeword N the feature maps channel C (N×C).We reduce the size of feature maps with fully linked layers for both branches to improve computational efficiency and to robustly integrate feature maps with bilinear models. A bilinear model is used to process feature maps from the texture encoding and global average pooling layers, which are then followed by a fully connected layer and a classification layer with non-linearities.

Table 1 is an instantiation of DEP based on 18-layer ResNet . We set 8 codewords for the texture encoding layer. The size of input images are 224×224. Outputs from CNNs are fed into the texture encoding layer and the global average pooling layer jointly. The dimension of outputs from the texture encoding layer is 8×512 = 4096 and the dimension of outputs from global average pooling layer is 512. We reduce the dimension of feature maps from the deep encoding layer and the global average pooling layer to 64 via fully connected layers. The dimension of outputs from bilinear model is 64 × 64 = 4096. Following prior works , resulting vectors from the texture encoding layer and bilinear model are normalized with L2 normalization.

Both the texture encoding layer and the bilinear models are distinct from each otherThe overall architecture is a directed acyclic graph and all the parameters can be trained by back propagation.As a result, the Deep Encoding Pooling Network is trained from beginning to end utilizing stochastic gradient descent and back-propagation.

II. EXPERIMENT

In This Section we evaluate the performance of DAIN,DEP and TEAN frameworks, we consider recognition performance with different DAIN variations for recognition and compare three other state-of-the-art approaches on our GTOS dataset, concluding that multiview DAIN works best.we compare the recognition performance of DEP with fine-tuning MobileNet, bilinear CNN and Deep-TEN.We compare the DEP network with the following three methods based on ImageNet.: (1) CNN with ResNet, (2) CNN with DeepTen and(3) CNN with bilinear models. All three methods support end-to-end training. For equal comparison, we use an identical training and evaluation procedure for each experiment.

CNN with global average pooling (ResNet) :Resnet is deep convolutional neural network architecture commonly used for image classification tasks.The model is usually trained on a large dataset like Imagenet,which has 1000 classes,but here we have used only 13 classes to train the model.we have removed fully connected network to monitor the model only on 13 classes and for some images we have involved dimension reduction and also instead of applying fully connected network we have used Global Average Pooling methodologies.

CNN with texture encoding (Deep-TEN) :modified architecture integrates texture encoding into the 18-layer ResNet, following the principles of Deep-TEN. The texture encoding layer is inserted in place of the global average pooling layer, and additional adjustments are made to align the model for the specific task, including reducing the number of output channels and introducing a bilinear mapping. The overall architecture is designed for image classification on datasets like GTOS with 31 classes of ground terrain images.

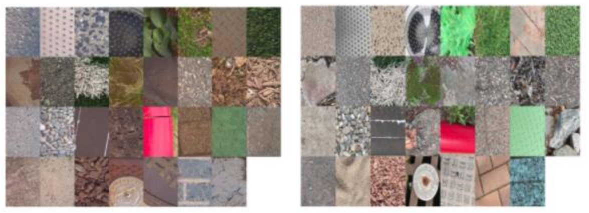
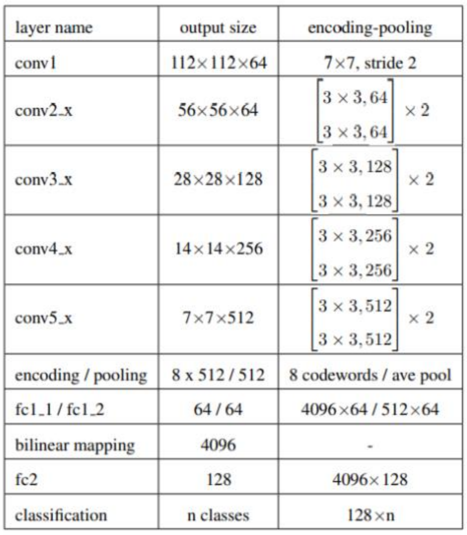


Figure 4: Comparison of images from the GTOS dataset (left) and GTOS-mobile (right) video frames. The training set is the ground terrain database (GTOS) with 31 classes of ground terrain images (over 30,000 images in the dataset). GTOS is collected with calibrated viewpoints. GTOS-mobile, consists of 81 videos of similar terrain classes captured with a handheld mobile phone and with arbitrary lighting/viewpoint. A total of 6066 frames are extracted from the videos with a temporal sampling of approximately 1/10th seconds. The figure shows individual frames of 31 ground terrain classes.



layer name	output size	encoding-pooling
conv1	112×112×64	7×7, stride 2
conv2_x	56×56×64	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$
conv3_x	28×28×128	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$
conv4_x	14×14×256	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$
conv5_x	7×7×512	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$
encoding / pooling	8 x 512 / 512	8 codewords / ave pool
fc1_1 / fc1_2	64 / 64	4096×64 / 512×64
bilinear mapping	4096	-
fc2	128	4096×128
classification	n classes	128×n

Table 1: The architecture of Deep Encoding Pooling Network based on 18-layer ResNet [12]. The input image size is 224 × 224.

CNN with bilinear models (Bilinear-CNN) : BilinearCNN] employs bilinear models with feature maps from convolutional layers. Outputs from convolutional layers of two CNN streams are multiplied using outer product at each location and pooled for recognition. To make an equal comparison, we employ the 18-layer pre-trained ResNet as CNN streams for feature extractor. Feature maps from the last convolutional layer are pooled with bilinear models. The dimension of feature maps for bilinear models is 7×7×512 and the pooled bilinear feature is of size 512×512. The pooled bilinear feature is fed into the classification layer for classification.

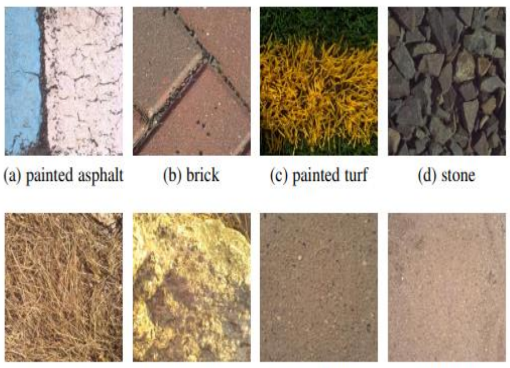
Dataset and Evaluation

Dataset Extending the GTOS database [36], we collect GTOS-mobile consisting of 81 videos obtained with a mobile phone (Iphone SE) and extract 6066 frames as a test set. To simulate real world ground terrain collection, we walk through similar ground terrain regions in random order to collect the videos. Scale is changed arbitrarily by moving far or close and changes in viewing direction are obtained by motions in a small arc. The resolution of the videos is 1920x1080, and we resize the short edge to 256 while keeping the aspect ratio for experiments. As a result, the resolution of the resized images are 455×256. Some materials in GTOS were not accessible due to weather, therefore we removed the following classes: dry grass, ice mud, mud-puddle, black ice and snow from the GTOS dataset. Additionally, we merged very similar classes of asphalt and metal. The original GTOS set is 40 classes;

there are 31 material classes in the modified dataset. The class names are (in the order of top-left to bottomright): asphalt, steel, stone-cement, glass, leaves, grass,plastic cover, turf, paper, gravel, painted turf, moss, cloth, stone-asphalt, dry leaves, mulch, cement, pebbles, sandpaper, roots, plastic, stone-brick, painted cover, limestone, soil, sand, shale, aluminum, metal cover, brick, painted asphalt.

	ResNet [12]	Bilinear CNN [19]	Deep-TEN [38]	DEP (ours)
Single scale	70.82	72.03	74.22	**76.07**
Multi scale	73.16	75.43	76.12	**82.18**

Multi-scale Training Images in the GTOS dataset were captured from a fixed distance between the camera and ground terrain, however the distance between the camera and ground terrain can be arbitrary in real world applications. We infer that extracting different resolution patches with different aspect ratio from images in GTOS to simulate observing materials at different distances and viewing angles will be helpful for recognition.



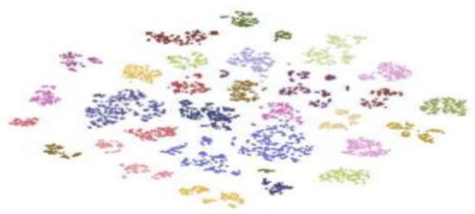
Multi-scale Training Images in the GTOS dataset were captured from a fixed distance between the camera and ground terrain, however the distance between the camera and ground terrain can be arbitrary in real world applications. We infer that extracting different resolution patches with different aspect ratio from images in GTOS to simulate observing materials at different distances and viewing angles will be helpful for recognition. resize the full resolution images into different scales, and extract 256×256 center patches for experiment. Through empirical validation, we find that resizing the full resolution images into 256×256, 384×384 and 512×512 works best. Training procedure We employ an identical

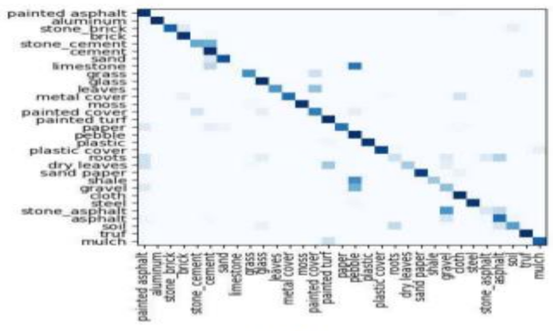
Training procedure We employ an identical data augmentation and training procedure for experiments. For single scale training experiment, we resize the full resolution images into 384×384 and extract 256×256 center patches as training sets. For multi scale training experiment, we resize the full resolution images into 256×256, 384×384 and 512×512, and extract 256×256 center patches as training set.we crop a random size (0.8 to 1.0) of the original size and a random aspect ratio (3/4 to 4/3) of the original aspect ratio, resize the cropped patches to 224×224 for experiment. All images are pre-processed by subtracting a per color channel mean value and normalized to unit variance with a 50% chance horizontal flip. The learning rate of newly added layers is 10 times of the pre-trained layers. The experiment starts with learning rate at 0.01, momentum 0.9, batch size 128; the learning rate decays by a factor of 0.1 for every 10 epochs, and is finished after 30 epochs.

Recognition Results

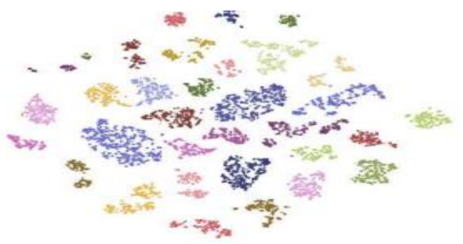
When comparing the performance of single-scale and multi-scale training, multi-scale training outperforms single-scale training for all approaches. It proves our inference that extracting different resolution patches with different aspect ratio from images in GTOS to simulate observing materials at different distances and viewing angles will be helpful for recognition. The multi-scale training accuracy for combined spatial information and texture details (DEP) is 82.18%. That's 9.02% better than only focusing on spatial information (ResNet) and 6% better than only focusing on texture details (Deep-TEN). To gain insight into why DEP outperforms ResNet and Deep-TEN for material recognition, we visualize the features before classification layers of ResNet, Deep-TEN and DEP with BarnesHut t-SNE.

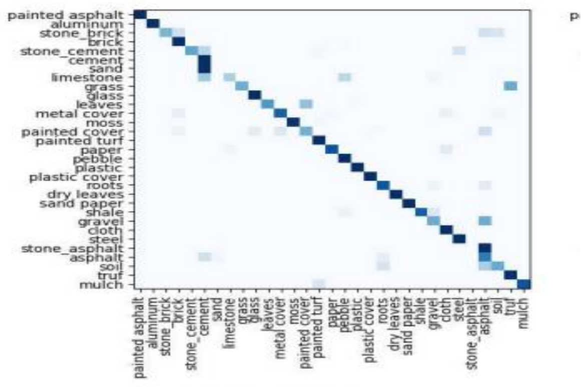
Evaluation on MINC and DTD Dataset: To show the generality of DEP for material recognition, we experiment on two other material/texture recognition datasets: Describable Textures Database (DTD) , and Materials in Context Database (MINC).For an equal comparison, we build DEP based on a 50-layer ResNet , the feature maps channels from CNN streams are reduced from 2048 to 512 with a 1×1 convolutional layer.DEP outperforms the state-of-the-art on both datasets. Note that we only experiment with single scale training.





(a) ResNet





(b) Deep-TEN

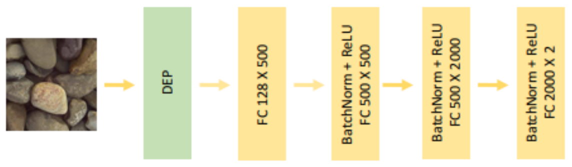
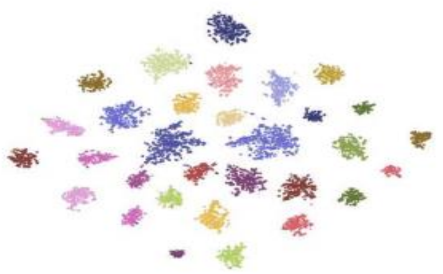
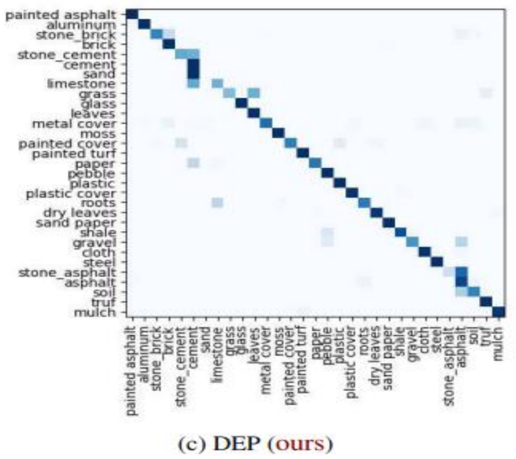


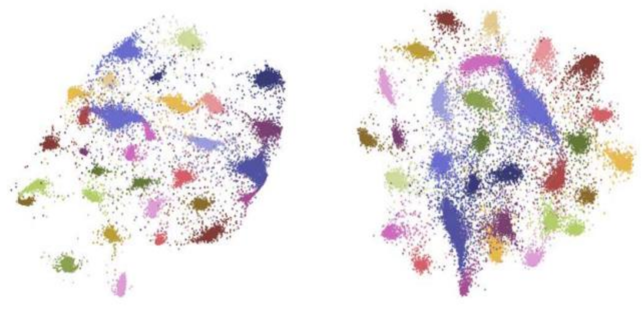
Figure 6: The deep network for texture manifold, we employ DEP as feature extractor, outputs from the last fully connected layer of DEP works as input for texture embedding.

Texture Manifold: For the embedded distribution of DEPParametric t-SNE, the classes are distributed unevenly with crowding in some areas and sparseness in others. The DEPmanifold has a better distribution of classes within the 2D embedding. We illustrate the texture manifold embedding by randomly choosing 2000 images from the training set to get the embedded distribution; then we embed images from the test set into the DEP-manifold.By observing the texture manifold, we find that for some classes, although the recognition accuracy is not perfect, the projected image is within the margin of the correct class, such as cement and stone cement. Based on the similarity of material classes on the texture manifold, we build the confusion matrix for material recognition algorithms.





(c) DEP (ours)



(a) DEP-parametric t-SNE (b) DEP-manifold

Confusion Matrix and Feature Visualization :

We compute the confusion matrix of MobileNet, DEP, DAIN and TEAN and visualize features before classification layers with Barnes-Hut tSNE.For features visualization, we employ images from validation set and extract features before classification layers of four models for experiment.

ResNet (left), Deep-TEN (mid) and DEP (right). For Barnes-Hut t-SNE, we randomly choose 10000 images from the training set and extract features before classification layers of three models for experiment. We see that DEP separates and clusters the classes better. Some classes are misclassified, however, they are typically recognized as a nearby class. (Dark blue represents higher values and light blue represents lower values in the confusion matrix.)

Conclusion:

In conclusion, this journal paper has delved into the exciting realm of deep texture modeling for ground terrain manifold representation.Through a comprehensive exploration of state-of-the art techniques, we have demonstrated the efficacy of leveraging deep learning methodologies for capturing intricate details and complex patterns inherent details and complex patterns inherent in ground terrains.Our study has highlighted the significance of incorporating deep texture representations to enhance the fidelity of terrain manifold models.The integration of neural networks has proven to be instrumental in overcoming the limitations of traditional models,enabling a more nuanced and realistic depiction of diverse terrains.The adaptability of deep texture models to various environmental conditions and terrains further underscore their potential in applications such as virtual reality,gaming,and simulation.Looking ahead,the insights gained from this study open avenues for future research in the realm of deep texture for terrain modelling.As technology continues to advance,we anticipate that further innovations in deep learning architectures,data augmentation techniques , and transfer learning methodologies will continue to refine and extend the capabilities of terrain representation. In conclusion,our findings underscore the transformative potential of deep texture modeling for ground terrain manifolds.

Acknowledgement:

We extend our sincere gratitude to all those who have contributed to the successful completion of this research endeavors on deep texture modeling for ground terrain manifolds.This work represents the collective efforts of numerous individuals and institutions whose support and expertise have been invaluable. We express our appreciation to our research collaborators for their dedication and insightful contributions throughout the various stages of this project.Their collaborations and constructive feedback have significantly enriched the quality of our work.Our heartfelt thanks go to the reviewers and editors for their meticulous evaluation,constructive criticism,and valuable suggestions that have greatly enhanced the rigor and clarity of this manuscript.We would also like to acknowledge the broader scientific community for sharing knowledge, resources,and fostering an environment collaboration.The vibrant exchange of ideas has played a crucial role in shaping the trajectory of our research.

References:

1.Aittala, M., & Durand, F. (2015). Gaussian Material Synthesis. ACM Transactions on Graphics (TOG), 34(4), 1-12.

2.Zhang, Z., He, T., Zhang, Z., Zhang, C., & Zheng, N. (2018). Manifold Learning for Texture Coordinates. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 773-781.

3.Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image Style Transfer Using Convolutional Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2414-2423.

4.Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5967-5976.

5. Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), 234-241.

6.Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. Science, 313(5786), 504-507.

7.Long, J., Shelhamer, E., & Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3431-3440.

8.Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., & Vedaldi, A. (2014). Describing Textures in the Wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3606-3613.