Histogram Layers for Synthetic Aperture Sonar Imagery

Joshua Peeples[1], Alina Zare[2], Jeffrey Dale[3,4], James Keller[3]

[1]Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA

[2]Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA

[3]Department of Computer Science and Electrical Engineering, University of Missouri, Columbia, MO, USA

[4]Naval Surface Warfare Center Panama City Division, 110 Vernon Ave, Panama City, FL 32407, USA

jpeeples@tamu.edu, azare@eng.ufl.edu, jjdale@mail.missouri.edu, KellerJ@missouri.edu

Abstract—Synthetic aperture sonar (SAS) imagery is crucial for several applications, including target recognition and environmental segmentation. Deep learning models have led to much success in SAS analysis; however, the features extracted by these approaches may not be suitable for capturing certain textural information. To address this problem, we present a novel application of histogram layers on SAS imagery. The addition of histogram layer(s) within the deep learning models improved performance by incorporating statistical texture information on both synthetic and real-world datasets.

Index Terms—deep learning, histograms, texture analysis, SAS imagery

I. INTRODUCTION

Synthetic aperture sonar (SAS) produces high resolution images of the seafloor [1]. The imagery generated through SAS can be used to perform important tasks such as automatic target recognition [2], environmental segmentation [3], and automated scene understanding [4]. Machine learning methods have been applied to SAS imagery to improve the accuracy and efficiency for these different applications. A critical aspect of the machine learning methods is the representation (*i.e.*, features) of the SAS imagery. As a result, several works investigate feature extraction approaches using a) handcrafted [5] and/or b) deep learning [6] features. Deep learning models, particularly convolutional neural networks (CNNs), have led to great strides in research for processing SAS imagery [2]–[4], [6]. A main reason for this success is the ability of deep learning frameworks to automate the feature extraction process and perform follow-on tasks (*e.g.*, classification, segmentation). Literature has suggested that CNNs are biased towards texture features rather than shape [7], [8]. For SAS-related efforts, texture information is crucial to represent the data [5]. CNNs are biased towards a particular type of texture (*i.e.*, *structural* texture) [9], [10]. To improve the information extracted within these deep learning models, other types of texture features can be incorporated into the network.
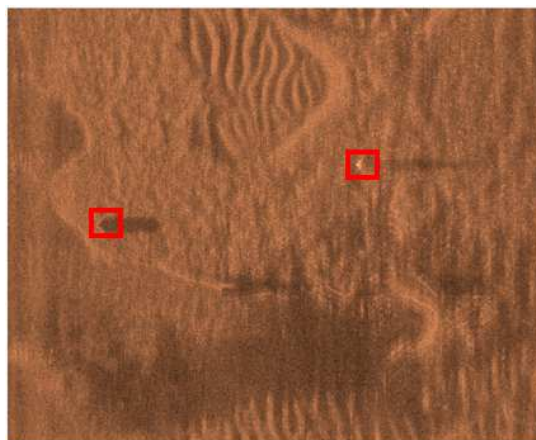
Fig. 1: SAS image example for understanding the environmental context that can be used in different tasks (*e.g.*, object recognition). The seafloor changes between different textures types and texture features can help capture this information. The objects of interest in the image are marked by red rectangles.

Texture information is a common type of image feature that has led to much success [11]. For example, texture can be used to provide environmental context in sonar imagery [12]–[14] that can be used in various tasks such as identifying potential hazards. In Figure 1, there are several difficult aspects of the environment that texture features can capture. For example, the seafloor types blend (*e.g.*, sand ripple patterns change, overlap between different seafloor types) and texture approaches can model these gradients and regions of transition that occur within the SAS imagery. Texture is always present within images [15]; therefore, the representation of texture content within an image is an important component of texture analysis. Additionally, the need for informative, domain-specific (*e.g.*, SAS) texture information has led to several methods for texture feature extraction, particularly histogram-based approaches [12], [16], [17].

As a result of the success of histogram-based, handcrafted features, histogram layer(s) have been proposed [9], [18]–[21]. These methods bridge the gap between handcrafted and deep
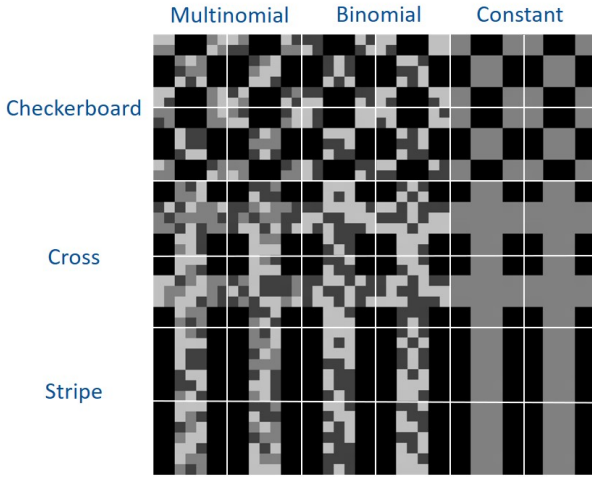
Fig. 2: Examples of different types of structural and statistical textures. The structural and statistical textures vary along the vertical and horizontal axis respectively.

learning approaches to improve the *statistical* texture context within the model. Structural textures consist of defining a set of texture examples and an order of spatial positions for each exemplar [22]. On the other hand, statistical textures represent the data through parameters that characterize the distributions and correlation between the intensity and/or feature values in an image, as opposed to understanding the structure of each texture [23]. Methods that focused on extracting statistical texture, particularly those computing measures between pairs of pixels (*i.e.*, second order), have outperformed the other texture analysis approaches [24]–[26]. To illustrate the difference between structural and statistical textures, we show an example in Figure 2. The structural textures are a checkboard, cross, and stripe displayed along the vertical axis of the figure. The statistical textures are shown along the horizontal axis where the foreground pixels values were sampled from a multinomial, binomial, and constant distributions. CNNs would have difficulty learning weights that could account for both the spatial ordering of the pixels (*i.e.*, structural changes) and individual pixel changes (*i.e.*, statistical changes). However, histogram layer(s) could be used to learn the statistical distributions to further enhance the features learned by the model to correctly classify the textures [9].

Inspired by the utility of statistical texture features, histogram layer(s) can be used to provide a powerful representation of the data more suitable for SAS imagery. We hypothesize that incorporating histogram layer(s) within deep learning models will improve performance on SAS-related tasks. The contributions of this work are the following:

- First application of histogram layer(s) for SAS imagery
- Investigation into model configuration for histogram layer(s)
- Thorough analysis of statistical and structural textures for SAS imagery

II. METHOD

A. Histogram Layer Review

We briefly summarize the local, radial basis function (RBF) histogram layer that was previously introduced [9]. The histogram layer can be applied to several data modalities with spatial information (*e.g.*, grayscale, RGB, and convolutional feature maps). Histograms generally require three hyperparameters: bin centers (γ_{bd}), bin widths (μ_{bd}), and the number of bins (B). With the histogram layer, the bin centers and widths are learned by the network instead of setting these values *a priori*. Given input data, $\mathbf{X} \in \mathbb{R}^{M \times N \times D}$, where M and N are the spatial dimensions while D is the feature dimensionality, the output tensor of the local histogram layer, $\mathbf{Y} \in \mathbb{R}^{R \times C \times B \times D}$ with spatial dimensions R and C after applying a histogram layer with kernel size $S \times T$ is shown in Equation 1:

$$Y_{rcbd} = \frac{1}{ST} \sum_{s=1}^{S} \sum_{t=1}^{T} e^{-\gamma_{bd}^2 (x_{r+s,c+t,d} - \mu_{bd})^2}. \quad (1)$$

Each input feature dimension is treated independently; therefore, BD histogram features maps are extracted (*e.g.*, three input channels and three bins would result in nine histogram feature maps). The histogram layer is constructed using pre-existing layers that add flexibility and ease of implementation.

B. Network Architectures

We implemented a) shallow and b) deep networks to evaluate the statistical and structural features of the SAS imagery. Following [9], the shallow models were composed of a feature extractor (histogram or convolutional layer), global average pooling layer (GAP), and output classification layer. The backbone architecture for the deep networks was Resnet18 (Figure 3a). In this work, we investigated series and parallel configuration (previously used in [9], [19]) as shown in Figures 3b and 3c respectively. The series configuration would encode only statistical texture information before the output classification layer. The parallel configuration would consist of both statistical and structural texture features. As a result, we are able to observe the impact of structural, statistical, and combination of these texture features through the performance of the baseline (*i.e.*, ResNet18), series histogram, and parallel histogram models respectively for SAS data.

III. EXPERIMENTAL PROCEDURE

A. Dataset Description

a) Statistical PISAS Dataset: The Pseudo Image SAS (PISAS) dataset [4] contains ten classes of sand ripple and rocky textures. Instead of the simple structures of cross, checkerboard, and stripe as shown in Figure 2, one class of sand ripple and rocky textures were selected to represent more SAS-related structures. Examples of sand ripple and rocky textures in the three different statistical classes are shown in Figure 4. The statistical PISAS dataset consisted of 14,208 images.

(a) Resnet18 Backbone
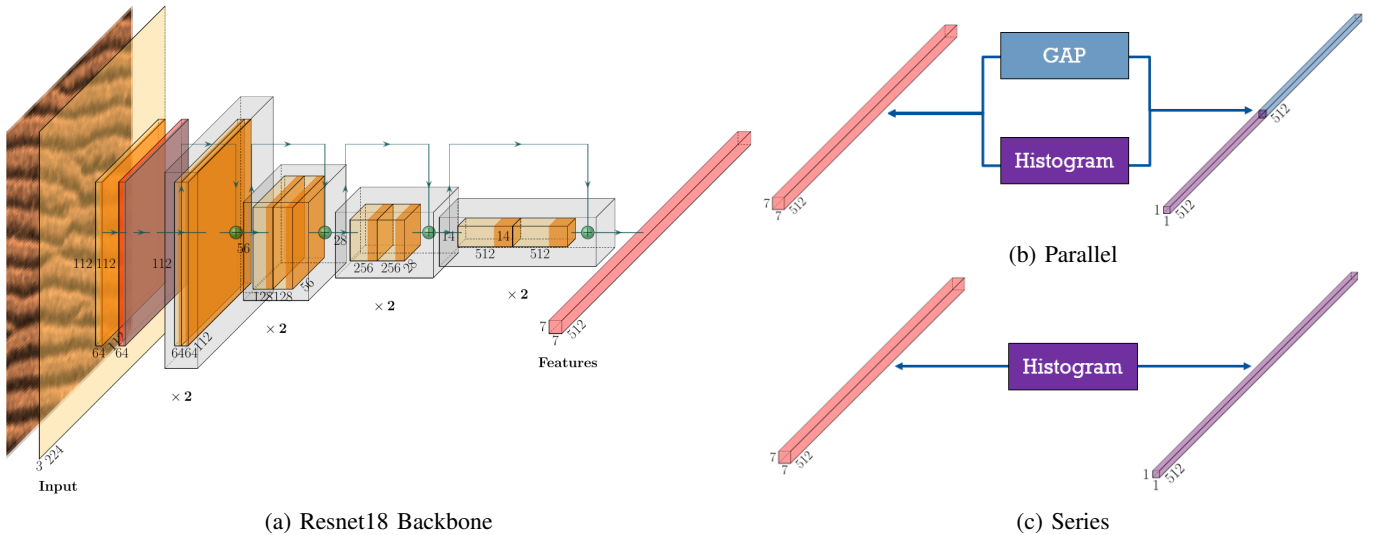

(b) Parallel

(c) Series

Fig. 3: Convolutional backbone and histogram model configurations used for multi-site dataset experiments. Resnet18 was used as the backbone architecture to extract convolutional features. In Figure 3a, the convolutional, ReLU, and max pooling layers are represented with the light yellow, orange, and purple colors respectively. The local features maps produced by the histogram layer are $2 \times 2 \times 128$. The histogram features are reshaped to be the same size (*i.e.*, $1 \times 1 \times 512$) as the feature vector produced by the global average pooling layer (GAP).
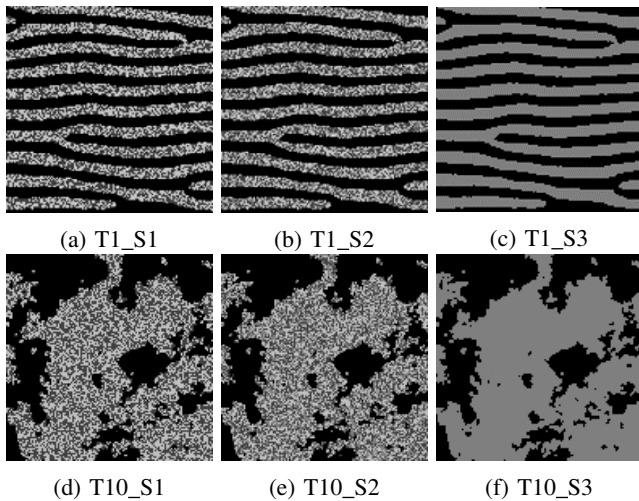

(a) T1_S1 (b) T1_S2 (c) T1_S3

(d) T10_S1 (e) T10_S2 (f) T10_S3

Fig. 4: Example of sand ripple ("T1") and rocky ("T10") images from the PISAS dataset with different statistical distributions. The same statistical classes (multinomial, binomial, and constant) and two SAS-related structures (*i.e.*, sand ripple and rocky) were used to generate a total of six distinct classes. The binomial, multinomial, and constant textures are referenced as "S1", "S2", and "S3" respectively. The subset of PISAS consisted of a total of 14, 208 images.

b) Multi-site Dataset: Example images for the multi-site dataset [27] are shown in Figure 5. The dataset consisted of 94 training, 12 validation, and 12 test images (118 images total) of pure textures from multiple locations along the seafloor. Random patches were sampled from the validation and test splits five times to generate 60 images for each set. A total

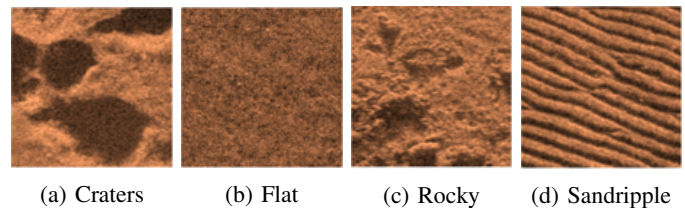(a) Craters (b) Flat (c) Rocky (d) Sandripple

Fig. 5: Example of images from multi-site dataset [27]. The dataset consisted of real-world SAS imagery pure texture regions from multiple locations on the seafloor. Four seafloor textures types were identified to train, validate and test the model: craters, flat, rocky, and sand ripple. The multi-site dataset contained 118 total images.

of four classes comprised the dataset: flat, rocky, sand ripple, and craters.

B. Experimental Design

a) Shallow Networks: We followed a similar training procedure as in [9] for the synthetic dataset experiments where each model was evaluated as a *local* feature extractor. The network architecture consisted of a feature extractor (convolutional or histogram layer), global average pooling layer, and fully connected output layer. The "CNN" model has a convolutional layer with ReLU as the feature extractor while the "Hist" model used the histogram layer as the feature extractor. The experimental settings for each model was the following: 70/10/20 training, validation, and testing splits, 300 epochs, and early stopping by checking if the validation loss did not decrease after 10 consecutive epochs. The input image
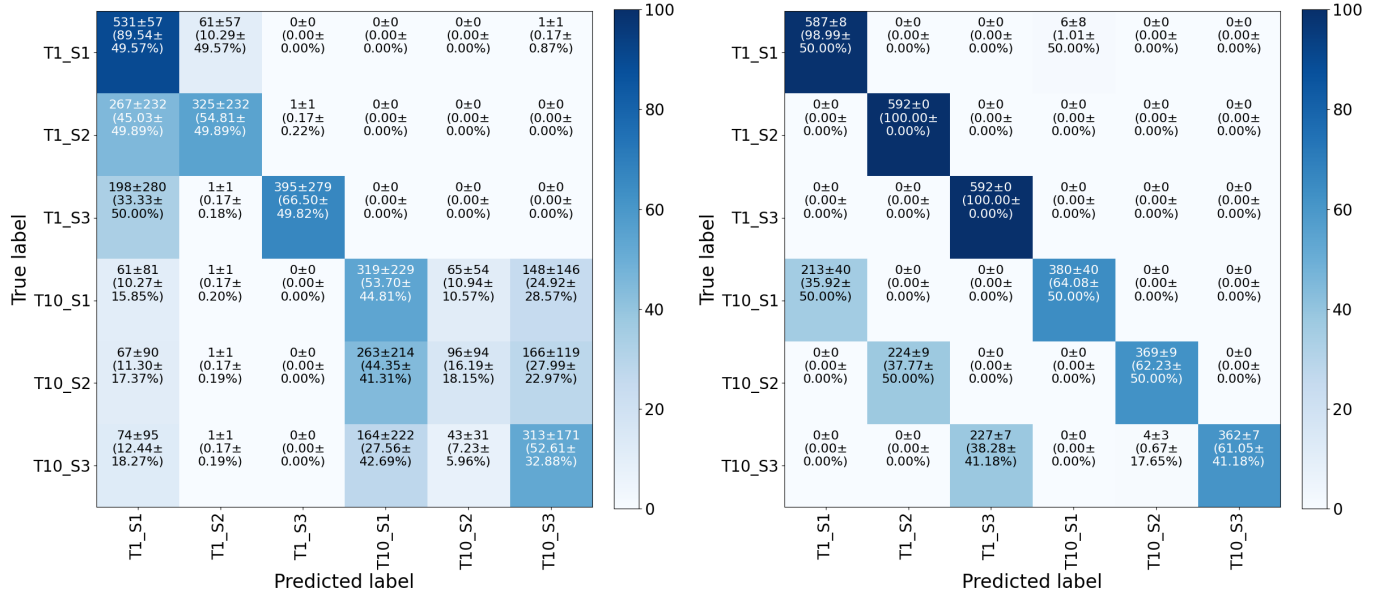
(a) CNN (55.65 ± 21.0) (b) Hist (81.11 ± 0.76)

Fig. 6: Average confusion matrices for the convolutional and histogram layer models on the PISAS dataset. "T1" and "T10" are sand ripple and rocky textures respectively. "S1", "S2", and "S3" are the binomial, multinomial, and constant distributions respectively.

patches were 128×128. The model parameters were adjusted so that the kernel size was 7×7 with a 3×3 stride, and a batch size of 128 was used to train each model. The number of bins for the RBF histogram layer model (*i.e.*, Hist) and convolutional kernels for the CNN model were set to three. The histogram layer bins were initialized to be equally spaced between the range of the input data (*i.e.*, 0 and 1). A total of three experimental runs were performed.

b) Deep Networks: The HistRes_B model [9] was applied to the multi-site dataset, and two configurations were investigated: parallel and series. The training details were the following: 100 epochs, learning rate of 0.001, Adam optimization [28], mini batch size of 16, 224×224 input image patches, and random crop data augmentation. A total of five experimental runs of random initialization were completed to evaluate the stability of each method. The number of bins was varied from 4, 8, and 16.

C. Results and Discussion

a) CNN vs. Histogram Layer Models: The results of each shallow network on the PISAS test data are shown in Table I. Similar trends in the results follow for the SAS structures as observed by the simple structures in [9]. The convolutional model can easily distinguish the sand ripple and rocky textures, but struggles with the statistical differences in comparison to the histogram layer. The histogram layer can be important in SAS application when the statistics of the data may change due to environmental conditions such as speckle noise [30]. The
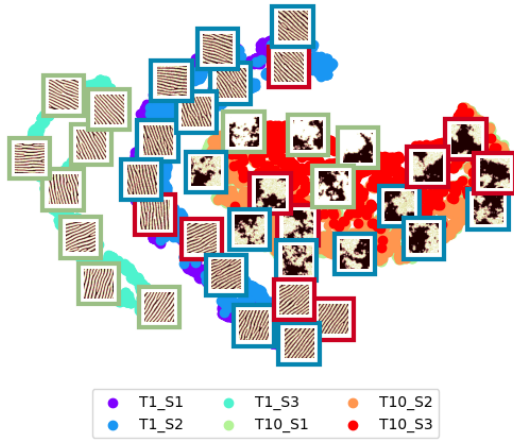
TABLE I: The average test accuracy for each model per class type is shown. The best average performance is bolded. The statistical labels relay the classification performance for predicting the distribution of the foreground pixels, while the structural labels show each model's prediction for sand ripple or rocky textures. Three experimental runs were conducted, and error values are reported with ± 1 standard deviation.

Label	CNN	Hist
Both	55.65±21.0	**81.11±0.76**
Statistical	57.37±18.8	**99.90±0.07**
Structural	**94.28±7.51**	81.21±0.73

confusion matrices also highlight the differences between the two models, as shown in Figure 6. The histogram layer model has the most difficulty with the structural changes between the sand ripple and rocky images, since there is equal weighting on spatial contribution of neighbors due to the average pooling operation. However, the CNN has more degradation in performance when observing statistical changes, particularly for the rocky texture ("T10").
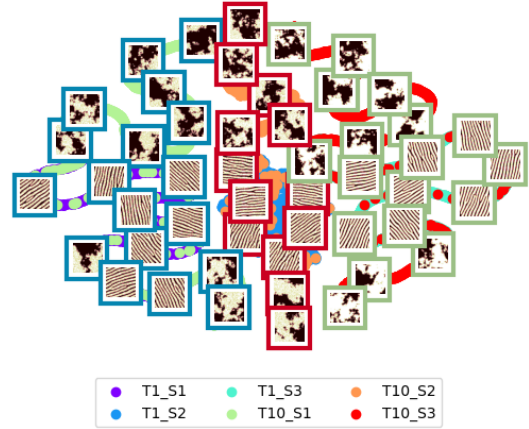
Interesting observations can also be seen qualitatively in the t-SNE projection of the training data, shown in Figure 7. The t-SNE projections used the same initialization and random seed. The only difference is the features that were projected into the lower dimensional space (convolutional or histogram features). For the CNN, the sand ripple and rocky images are projected near each other. However, the statistical textures

● T1_S1	● T1_S3	● T10_S2
● T1_S2	● T10_S1	● T10_S3

(a) CNN (77.70)

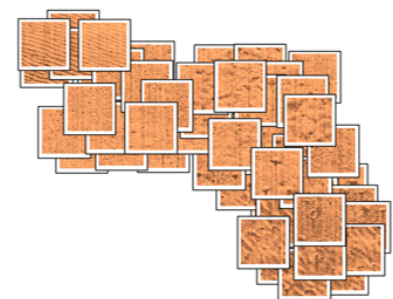|		
● T1_S1	● T1_S3	● T10_S2
● T1_S2	● T10_S1	● T10_S3

(b) Hist (82.18)

Fig. 7: t-SNE projections for the convolutional and histogram layer models on the PISAS dataset. The test classification accuracy is shown in parenthesis. Example images from each class are shown. The frames around each image represent the statistical class of the PISAS dataset. Red, green, and blue frames represent the multinomial, constant, and binomial distributions respectively. The CNN model appears to map structural textures near one another in the projected space (*i.e.*, sand ripple images are grouped to the left region of the projection and the rocky textures are grouped in the right of the projection). On the other hand, the histogram layer model maps statistical textures close to one another. The binomial statistical textures (blue frames) are grouped in the left of the projection. The multinomial (red frames) and constant (green frames) statistical textures are clustered in the center and right of the projection respectively.
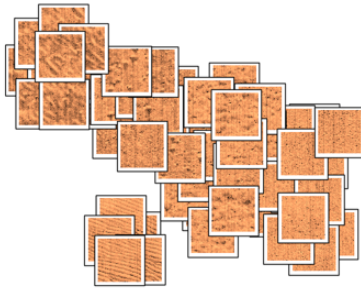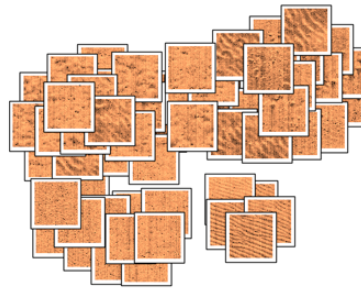
(a) ResNet18 (95.00)   (b) Series (86.67)   (c) Parallel (100.00)

Fig. 8: t-SNE projections for the best baseline ResNet18, HistRes_16 Series, and HistRes_16 Parallel. The test classification accuracy for the model is shown in parenthesis. The baseline ResNet18 did not have the histogram layer; as a result, the model will focus on structural textures. The projection of the ResNet18 features on the test data highlight that images with similar structures are in the mapped near one another in the lower dimensional space. Sand ripple images are to the left of the projection while there is a transition to rocky and crater textures in the middle with flat textures projected to the far right. The contrary is shown for the Series histogram model. Textures with similar statistics (*i.e.*, pixel intensities) are projected near one another. The parallel model consist of both statistical and structural textures information; therefore, we observe a mix of the different textures in the projected space.

are mapped to similar regions of the feature space after the t-SNE projection. The clusters are also appeared tighter for the histogram layer model, indicating that the histogram layer may be able to handle small intra-class variations in the data through the soft binning operations to characterize images in the same class.

Additional analysis was needed to verify the plausible conclusions discussed above to ensure the difference is a result of the higher dimensional structure and not an artifact of the t-SNE mapping. Therefore, we used a cluster validity metric to assess the quality of the features learned by each method with regard to the labels. The Calinski-Harabsz index [29] considers both intra- and inter-cluster distances. Larger values indicate more compact and separated classes in the feature
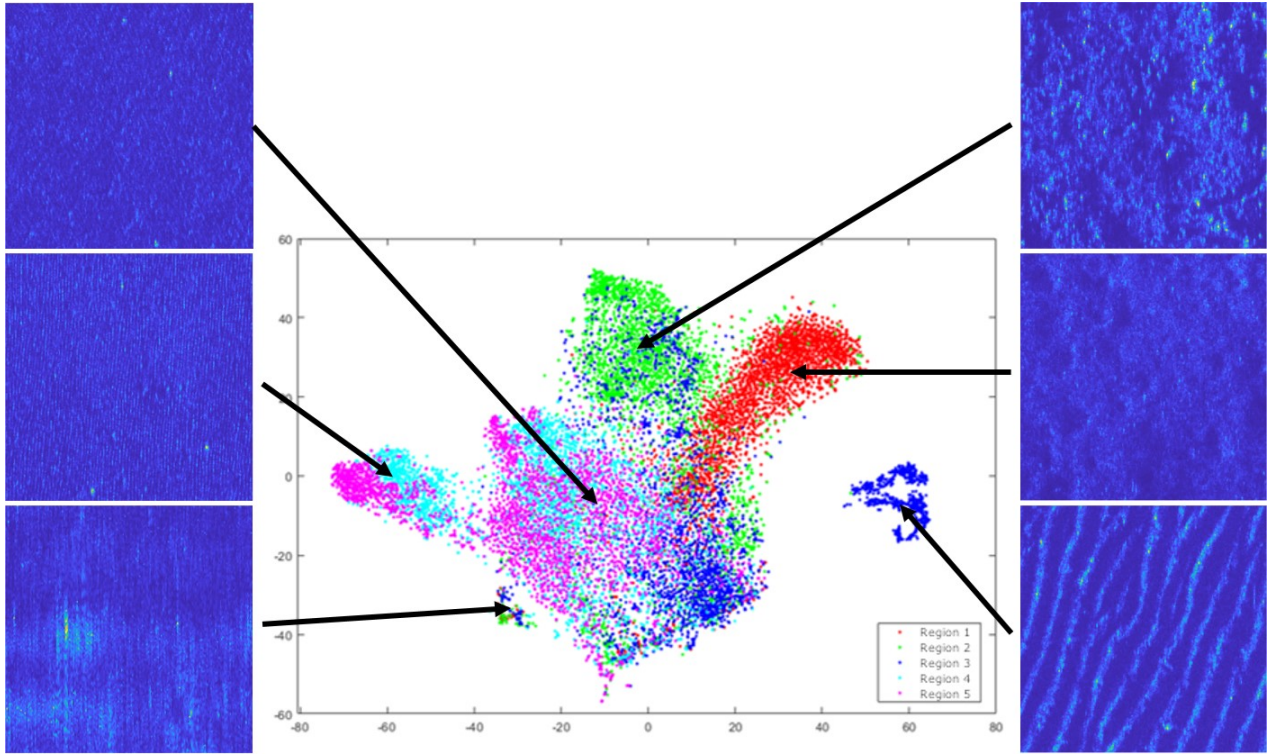
Fig. 9: t-SNE projections of SAS images from various locations of the seafloor passed through the best parallel HistRes_16 model. The colors indicate images collected from different regions. The best parallel model was trained on a small subset of labelled SAS imagery (*i.e.*, multi-site dataset). The model was able to learn features that generalized to new images from different regions. In the projected space, we see meaningful groupings such as sand ripple textures in the far right of the projection. The model was also able to extract features that grouped images containing cross talk, as shown in the bottom left-hand corner of the projection.

TABLE II: The average Calinski-Harabasz index [29] for each model per class type is shown. The best average performance is bolded (larger values indicate better separability and compactness). The statistical labels demonstrate the feature quality for predicting the distribution of the foreground pixels, while the structural labels show the feature quality for distinguishing between the sand ripple or rocky textures. Three experimental runs were conducted, and error values are reported with ± 1 standard deviation.

Label	CNN	Hist
Both	3,333.88$\pm$2,163.06	**26,129.02$\pm$236.77**
Statistical	40.55$\pm$32.50	**50,516.00$\pm$460.19**
Structural	**15,399.12$\pm$9951.00**	134.28$\pm$3.92

space. We observed that the metrics matched our analysis of the lower dimensional space, as shown in Table II. The histogram layer model outperformed the CNN for statistical textures, while the CNN was superior for structural textures. Overall, we noted that the histogram features had the best features across all classes. Experiments on additional SAS images in the *Parallel vs. Series Configuration* section further demonstrated the power of statistical features for deep learning models.

TABLE III: The average test accuracy for each model on the SAS dataset is shown. The best average performance is bolded. The baseline ResNet18 model achieved a test classification accuracy of 84.66$\pm$11.42. Five experimental runs were conducted, and error values are reported with ± 1 standard deviation.

Configuration	HistRes_4	HistRes_8	HistRes_16
Parallel	**90.33$\pm$4.00**	**82.67$\pm$8.27**	**88.33$\pm$9.60**
Series	77.33$\pm$5.01	73.33$\pm$5.06	81.00$\pm$4.03

b) Parallel vs. Series Configuration: The test performance on the multi-site dataset for the model configuration experiments is shown in Table III. The parallel and series models were robust to the number of bins, as there was not a significant difference in performance (*i.e.*, overlapping error bars). The parallel model outperformed both the baseline and series models. This further validated the utility of using both statistical and structural information to maximize texture classification. In addition to the overall test performance, a qualitative analysis was performed using t-SNE (Figure 8). The convolutional model mapped textures with similar structures near each other, while the histogram models mapped points with similar statistics near one another. For the parallel

configuration, there was a mix of similar statistical and structural SAS images near one another in the projected space.

Once training and evaluation was completed on the subset of labeled SAS images, the model was applied to hold-out SAS images from multiple locations on the seafloor. The t-SNE projection is shown in Figure 9. The model successfully extracted features from the hold-out images that showed meaningful groupings in the lower dimensional projection space. The parallel HistRes_16 model can also potentially be used to identify images with cross-talk (bottom left-hand corner of Figure 9). The results in this section demonstrated the use of the histogram layer for various real-world applications for SAS texture classification.

IV. CONCLUSION

We presented a novel application of histogram layer(s) for SAS image classification. The results presented in this work show the impact of statistical texture features within deep learning frameworks. For SAS imagery, histogram layer(s) provide a powerful feature representation that can jointly be tuned with CNNs to optimize performance. Future work includes exploring other convolutional backbones, investigating other binning functions, performing semantic segmentation, and extensive comparisons with other SAS image analysis approaches.

REFERENCES

[1] M. P. Hayes and P. T. Gough, "Synthetic aperture sonar: a review of current status," *IEEE journal of oceanic engineering*, vol. 34, no. 3, pp. 207–224, 2009.

[2] G. Richard, J.-M. Le Caillec, J. Habonneau, and D. Gueriot, "A deep sas atr explainability framework assessment," in *OCEANS 2021: San Diego–Porto*. IEEE, 2021, pp. 1–5.

[3] Y.-C. Sun, I. D. Gerg, and V. Monga, "Iterative, deep synthetic aperture sonar image segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.

[4] D. Stewart, S. Johnson, and A. Zare, "Image-to-height domain translation for synthetic aperture sonar," *arXiv preprint arXiv:2112.06307*, 2021.

[5] D. P. Williams, "Unsupervised seabed segmentation of synthetic aperture sonar imagery via wavelet features and spectral clustering," in *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2009, pp. 557–560.

[6] D. Kohntopp, B. Lehmann, D. Kraus, and A. Birk, "Seafloor classification for mine countermeasures operations using synthetic aperture sonar images," in *OCEANS 2017-Aberdeen*. IEEE, 2017, pp. 1–5.

[7] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," in *International Conference on Learning Representations*, 2018.

[8] K. Hermann, T. Chen, and S. Kornblith, "The origins and prevalence of texture bias in convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19000–19015, 2020.

[9] J. Peeples, W. Xu, and A. Zare, "Histogram layers for texture analysis," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 4, pp. 541–552, 2022.

[10] L. Zhu, D. Ji, S. Zhu, W. Gan, W. Wu, and J. Yan, "Learning statistical texture for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12537–12546.

[11] L. Liu, J. Chen, P. Fieguth, G. Zhao, R. Chellappa, and M. Pietikäinen, "From bow to cnn: Two decades of texture representation for texture classification," *International Journal of Computer Vision*, vol. 127, no. 1, pp. 74–109, 2019.

[12] J. Peeples, M. Cook, D. Suen, A. Zare, and J. Keller, "Comparison of possibilistic fuzzy local information c-means and possibilistic k-nearest neighbors for synthetic aperture sonar image segmentation," in *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXIV*, vol. 11012. International Society for Optics and Photonics, 2019, p. 110120T.

[13] A. Zare, N. Young, D. Suen, T. Nabelek, A. Galusha, and J. Keller, "Possibilistic fuzzy local information c-means for sonar image segmentation," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2017, pp. 1–8.

[14] D. P. Williams, "Fast unsupervised seafloor characterization in sonar imagery using lacunarity," *IEEE transactions on Geoscience and Remote Sensing*, vol. 53, no. 11, pp. 6022–6034, 2015.

[15] R. M. Haralick, K. Shanmugam *et al.*, "Textural features for image classification," *IEEE Transactions on systems, man, and cybernetics*, no. 6, pp. 610–621, 1973.

[16] Z. Zhu, X. Xu, L. Yang, H. Yan, S. Peng, and J. Xu, "A model-based sonar image atr method based on sift features," in *OCEANS 2014-TAIPEI*. IEEE, 2014, pp. 1–4.

[17] J. T. Cobb, K. C. Slatton, and G. J. Dobeck, "A parametric model for characterizing seabed textures in synthetic aperture sonar images," *IEEE Journal of Oceanic Engineering*, vol. 35, no. 2, pp. 250–266, 2010.

[18] V. Sedighi and J. Fridrich, "Histogram layer, moving convolutional neural networks towards feature-based steganalysis," *Electronic Imaging*, vol. 2017, no. 7, pp. 50–55, 2017.

[19] Z. Wang, H. Li, W. Ouyang, and X. Wang, "Learnable histogram: Statistical context features for deep neural networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 246–262.

[20] I. Yusuf, G. Igwegbe, and O. Azeez, "Differentiable histogram with hard-binning," *arXiv preprint arXiv:2012.06311*, 2020.

[21] H. Sadeghi and A.-A. Raie, "Histnet: Histogram-based convolutional neural network with chi-squared deep metric learning for facial expression recognition," *Information Sciences*, vol. 608, pp. 472–488, 2022.

[22] A. Materka, M. Strzelecki *et al.*, "Texture analysis methods–a review," *Technical university of lodz, institute of electronics, COST B11 report, Brussels*, pp. 9–11, 1998.

[23] A. Humeau-Heurtier, "Texture feature extraction methods: A survey," *IEEE Access*, vol. 7, pp. 8975–9000, 2019.

[24] J. S. Weszka, C. R. Dyer, and A. Rosenfeld, "A comparative study of texture measures for terrain classification," *IEEE transactions on Systems, Man, and Cybernetics*, no. 4, pp. 269–285, 1976.

[25] R. M. Haralick *et al.*, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.

[26] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[27] S. Walker, J. Peeples, J. Dale, J. Keller, and A. Zare, "Explainable systematic analysis for synthetic aperture sonar imagery," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2021, pp. 2835–2838.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[29] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[30] A. Abu and R. Diamant, "Robust image denoising for sonar imagery," in *2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO)*. IEEE, 2018, pp. 1–5.