# Histogram Approach: Deep Learning Texture Analysis
## Methodology

SSTP Participant
Tyler (Taewook) Kim
Kent School
1 Macedonia Rd., Kent, CT 06757
kimt20@kent-school.edu

Grduate Research Assistant
Joshua Peeples
University of Florida
1064 Center Dr, Gainesville, FL 32611
jpeeples@ufl.edu

## 1. Methodology

### 1.1. Binning Operation

Binning operation in standard histogram can be expressed with the following counting fuction:

$$y_k = \begin{cases} 1, B_k - w \leq x_k < B_k + w \\ 0, otherwise \end{cases} \quad (1)$$

This indicator function returns "1" if the element falls into a bin and returns "0" else. The condition when an element falls into a bin is defined by $B_k - w \leq x_k < B_k + w$, $B_k$ being the bin center and $w$ being the bin width. However, the standard histogram operation is not differentiable and cannot be used for the backpropagation.

Therefore, our approach is to perform a localized binning operation with a sliding window. RBFs will be used instead of standard histogram operation to approximate the count values:

$$y_k = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} e^{-\frac{(x_{ij} - \mu_k)^2}{\sigma_k^2}}. \quad (2)$$

The function above returns the count $y_k$, for kth bin center $\mu_k$ and width $\sigma_k$ from the feature map value $x_{ij}$ .

An example with M×N window in an image is shown in Figure 1. Each square represents a pixel with a single intensity value of 1, 2, or 3. Every pixel will contribute to every bin but not equally due to the RBF we are using.

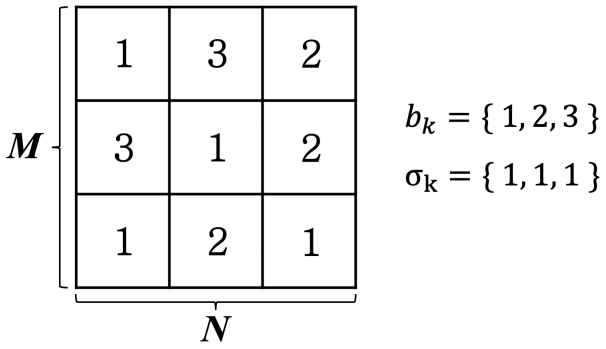


Figure 1. M×N Window

| | $b_0$ **Value: 1** | $b_1$ **Value: 2** | $b_2$ **Value: 3** |
|---|---|---|---|
| Standard | 0.44 | 0.33 | 0.22 |
| RBF | 0.57 | 0.42 | 0.35 |

Table 1. Normalized Frequency Comparison

Table 1 compares the resulting count value from the two equations introduced. Each column represents the bins, $b_0$, $b_1$ and $b_2$ each indicating bin center of 1, 2, and 3 respectively. The first row shows the value calculated by (1) with $w = 0$, and the second row is based on (2). RBF is used to approximate the counting function because it allows differentiation thereby allowing the model to learn via backpropagation. As shown in Figure 1, $\sigma_k$ is set to be 1, which makes the difference between the two values larger. $\sigma_k$ is a crucial factor that determines this difference. As $\sigma_k$ value becomes smaller, the model will only consider the values that are very close to the bin center. If model learns the optimum $\sigma_k$ value, we will be able to account for ambiguity in the data as opposed to crisp histograms. As $\sigma_k$ value becomes larger, every element will fall into the same bin.

### 1.2. Experiment Design

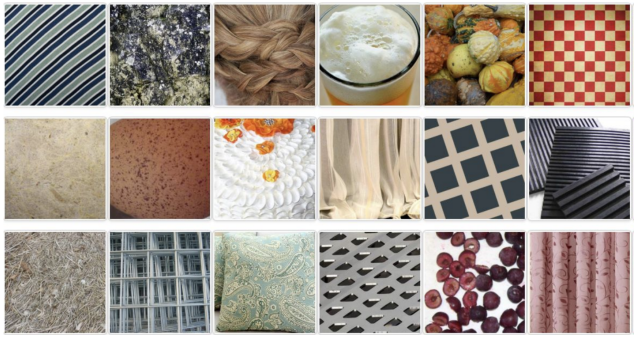

Figure 2. Example of DTD Images [1]

To evaluate the performance of the proposed model, we used Describable Textures Dataset (DTD), a texture database with 5640 images classified into 47 categories. We will use five different artificial neural networks (ANN) that are comprised of convolutional and histogram layers to classify the images. We will compare the results both quantitatively and qualitatively. In qualitative analysis, we will focus on displaying images that are classified correctly or incorrectly by each network. In quantitative analysis, we record accuracy(class and overall), class precision, class recall, class F1 score, learning curve, and confusion matrices.

### 1.3. Training Procedure

For our training procedure, our data is first scaled to between 0 and 1. Next, we wanted to look at the effects of preprocessing our data through standardization

$$Z = \frac{x - \mu}{\sigma}. \tag{3}$$

The equation 3 represents the standardization operation, where $x$ is the original feature value, $\mu$ is the mean of that feature value, $\sigma$ is its standard deviation, and $Z$ is the resulting standardized data. For our dataset, $x$ is the RGB value for each pixel and the corresponding mean ($\mu$) and standard deviation ($\sigma$) for each channel will be used to normalize the data through equation 3 . Data preprocessing is considered to be one of the most important components of training the model, and we want to compare the results from standardizing the data and relating this to the performance of each network.

For the training process, we use the same data augmentation and optimization technique for the six different neural networks. Similar to [3], all the samples were first resized to 256 $\times$ 256 then a crop of 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 is extracted. This crop is finally resized to 224$\times$224 and horizontal flips ($p = 0.5$) were used. The experiment starts with learning rate of 0.01 and batch size of 128. Throughout 100 total epochs, the learning rate decays by factor of 0.9 for every 10 epochs. During the learning rate decay, Adam, an adaptive learning rate optimization algorithm, is used. Adam allows for efficient stochastic optimization that only requires first-order gradients, allowing for little memory requirement[2].

## References


[1] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[2] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[3] Jia Xue, Hang Zhang, and Kristin J. Dana. Deep texture manifold for ground terrain recognition. *CoRR*, abs/1803.10896, 2018.