Histogram Layer for Texture Classification

Taewook (Tyler) Kim[1], Joshua Peeples[2], Dr. Alina Zare[2]

[1]Kent School, Kent, CT 06757;
[2]Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32608

Introduction and Motivation

Texture Classification

Images from KTH-TIPS 2b Dataset

- **Definition**: an area of texture analysis that focuses on assigning images into a texture classes.
- **Applications**: medical imaging, defense, agriculture.

Traditional Approach

Input → Hand-crafted features → Classifier → Output

- **Definition**: Researchers used different hand-crafted features such as Local Binary Patterns (LBP), Gray-Level Co-Occurrence Matrices (GLCM), and Edge Histogram Descriptors (EHD).
- **Problem**: Very laborious; often needed to determine features empirically.

Deep Learning Approach

Input → Deep Learning (Automated Feature Learning) → Output

- **Definition**: With the recent advances on Convolutional Neural Networks (CNN), researchers have applied deep neural networks to improve performance and avoid the laborious process of developing hand-crafted features.
- **Problem**: Training deep learning models requires a copious amount of labeled data along with immense amounts of computational power.

Method

In this work, we propose a novel model that incorporates a localized histogram layer for convolutional neural networks (CNNs). Our studies will be the first attempt use a radial basis function instead of standard histogram operation which creates several advantages

$$y_k = \begin{cases} 1, B_k - w \le x_k < B_k + w \\ 0, otherwise \end{cases}$$

Standard Operation

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

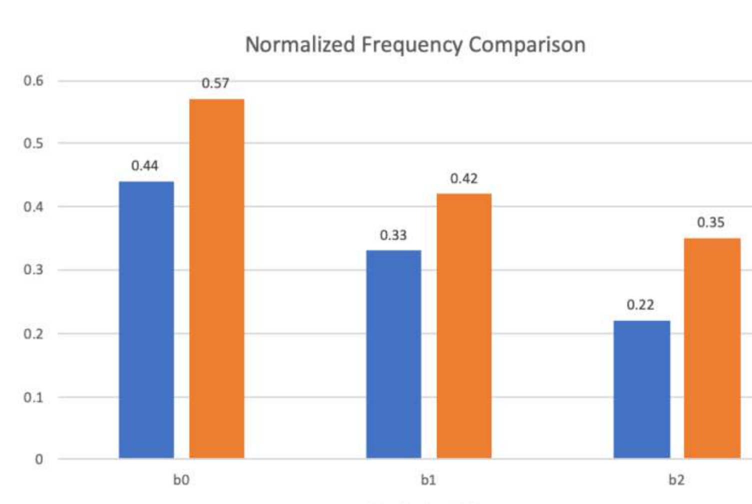
Radial Basis Function

1. Spatial information will be retained as opposed to previous global methods.
2. The histogram layer will be less sensitive to various outliers and ambiguity in the data because of "soft" binning assignments (see example below)
3. Using RBFs (Radial Basis Functions) will also provide a differentiable histogram operation allowing the model to learn via back-propagation.

$$M \left\{ \begin{array}{|c|c|c|} \hline 1 & 3 & 2 \\ \hline 3 & 1 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \right. \quad \underbrace{\hphantom{1\;3\;2}}_{N}$$

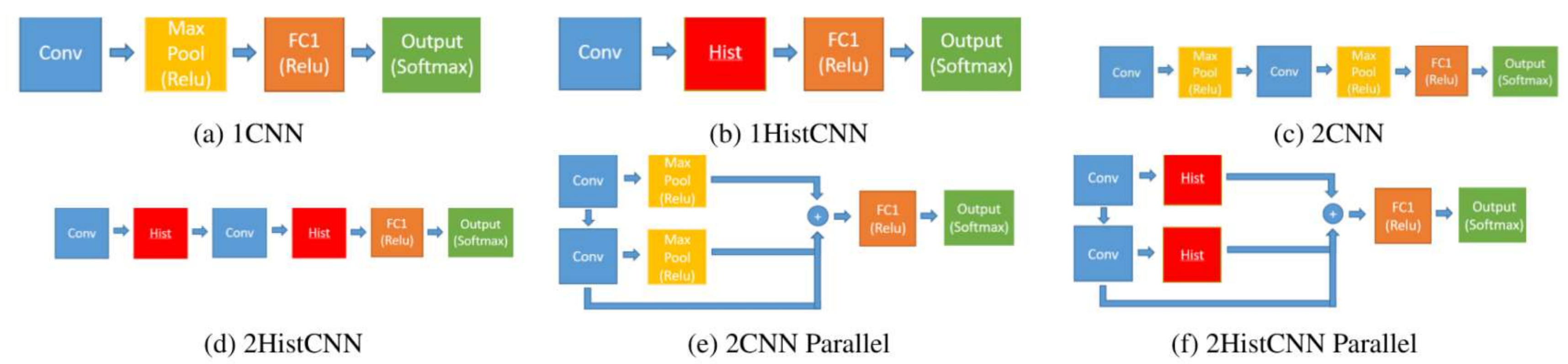
M x N window

$\mu_k = \{1, 2, 3\}$
$\sigma_k = \{1, 1, 1\}$



Binning of window

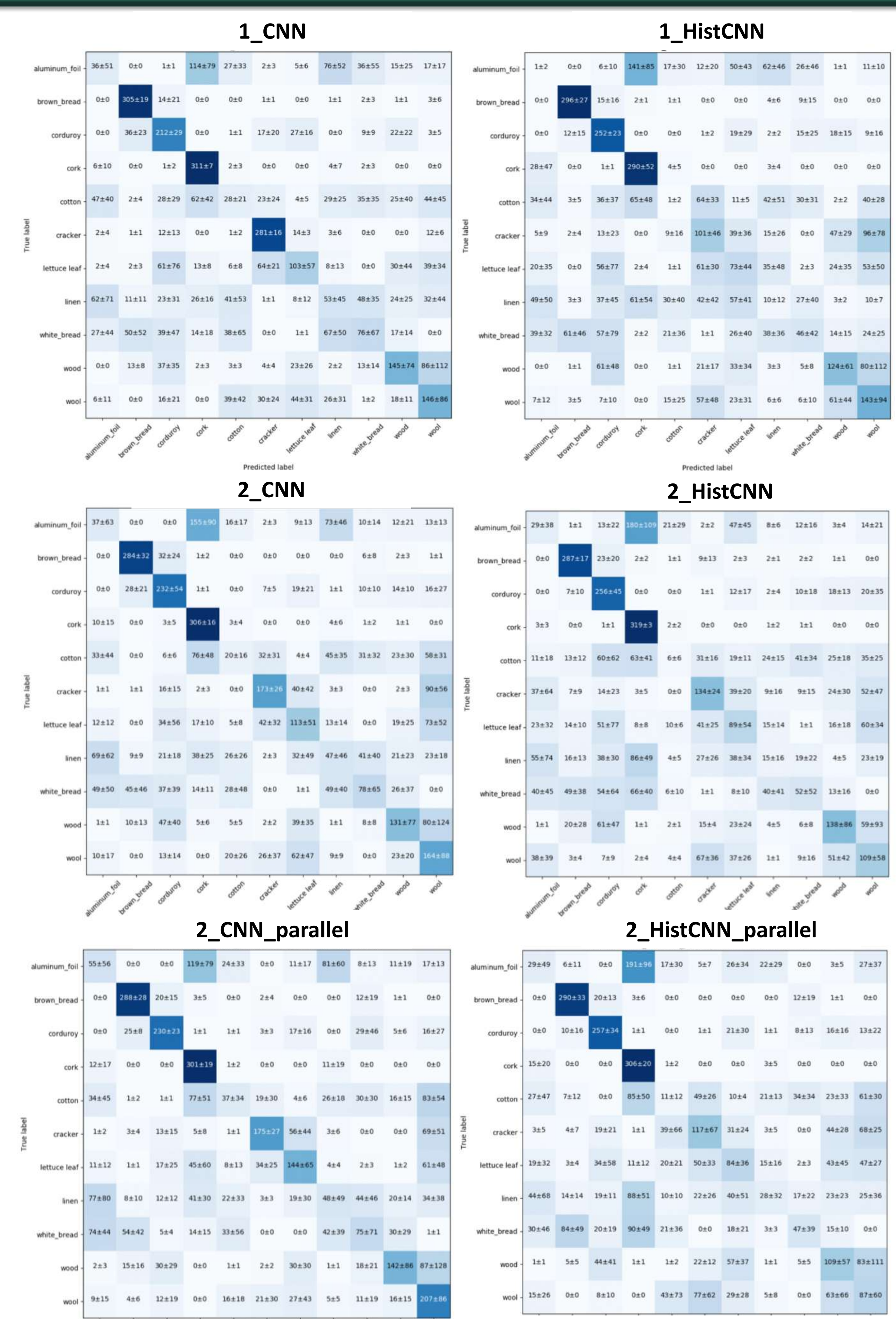
Experimental Setup



Architectures used for each texture dataset.

- To evaluate the performance of the proposed model, we use six different artificial neural networks (ANN) with the following constraints: 1) Similar architectures and 2) Number of parameters.

Results & Discussion



Confusion matrices reflect how each network performed in specific classes.

Average Overall Accuracy

Network	Average Overall Accuracy	
	Non-Standardized	Standardized
1_CNN	38.87 ± 4.05	47.48 ± 0.97
1_HistCNN	34.90 ± 4.09	37.43 ± 3.33
2_CNN	29.51 ± 5.14	44.35 ± 1.07
2_HistCNN	36.75 ± 2.58	40.14 ± 1.49
2_CNN_parallel	40.31 ± 3.79	47.66 ± 1.22
2_HistCNN_parallel	35.91 ± 4.15	38.20 ± 2.85

Best & Worst Classes

Best	Worst
- Brown bread	- Cotton
- Corduroy	- Linen
- Cork	- Aluminum Foil

- Every network improved performance significantly with standardization, averaging about 6.50% increase in average overall accuracy.
- Most of the histogram networks showed poorer accuracy than the corresponding CNNs. However, since this is a multi-class problem, average overall accuracy alone cannot determine networks' performance. Therefore, we looked at average f1-score, precision, and recall for further evaluation.

1_CNN & 1_HistCNN

	1_CNN Avg. F1-score	**1_HistCNN Avg. F1-score**	**1_CNN Avg. Precision**	**1_HistCNN Avg. Precision**	**1_CNN Avg. Recall**	**1_HistCNN Avg. Recall**
Sample A	0.45 ± 0.24	0.31 ± 0.28	0.47 ± 0.22	0.30 ± 0.27	0.48 ± 0.29	0.36 ± 0.32
Sample B	0.43 ± 0.22	0.35 ± 0.29	0.43 ± 0.21	0.33 ± 0.30	0.47 ± 0.27	0.40 ± 0.32
Sample C	0.44 ± 0.25	0.28 ± 0.28	0.43 ± 0.24	0.29 ± 0.31	0.49 ± 0.28	0.33 ± 0.33
Sample D	0.40 ± 0.23	0.35 ± 0.30	0.42 ± 0.22	0.35 ± 0.29	0.46 ± 0.29	0.41 ± 0.32

2_CNN & 2_HistCNN

	2_CNN Avg. F1-score	**2_HistCNN Avg. F1-score**	**2_CNN Avg. Precision**	**2_HistCNN Avg. Precision**	**2_CNN Avg. Recall**	**2_HistCNN Avg. Recall**
Sample A	0.40 ± 0.23	0.33 ± 0.25	0.41 ± 0.22	0.31 ± 0.21	0.44 ± 0.28	0.41 ± 0.33
Sample B	0.40 ± 0.23	0.37 ± 0.24	0.40 ± 0.21	0.37 ± 0.24	0.44 ± 0.26	0.42 ± 0.31
Sample C	0.42 ± 0.22	0.34 ± 0.25	0.44 ± 0.22	0.34 ± 0.22	0.46 ± 0.29	0.40 ± 0.34
Sample D	0.38 ± 0.24	0.32 ± 0.24	0.40 ± 0.24	0.31 ± 0.22	0.43 ± 0.27	0.38 ± 0.32

2_CNN_parallel & 2_HistCNN_parallel

	2_CNN_par Avg. F1-score	**2_HistCNN_par Avg. F1-score**	**2_CNN_par Avg. Precision**	**2_HistCNN_par Avg. Precision**	**2_CNN_par Avg. Recall**	**2_HistCNN_par Avg. Recall**
Sample A	0.44 ± 0.26	0.31 ± 0.26	0.45 ± 0.24	0.31 ± 0.31	0.48 ± 0.31	0.36 ± 0.31
Sample B	0.44 ± 0.24	0.31 ± 0.27	0.48 ± 0.23	0.31 ± 0.30	0.46 ± 0.30	0.36 ± 0.30
Sample C	0.45 ± 0.24	0.38 ± 0.25	0.46 ± 0.24	0.39 ± 0.30	0.49 ± 0.32	0.43 ± 0.30
Sample D	0.42 ± 0.28	0.31 ± 0.26	0.43 ± 0.24	0.35 ± 0.31	0.48 ± 0.33	0.38 ± 0.30

- From these comparisons, we learned that the CNNs implemented with histogram layer did not necessarily perform better on texture classification.

Conclusion

Contributions
- Learned standardization improves performance significantly.
- Histogram layer did not necessarily perform better

Future Works
- Different initialization techniques for histogram layer can be used
- Tuning of parameters (i.e. window size, number of bins, kernel size)

Acknowledgement

- **Joshua Peeples**
- **Dr. Alina Zare**
- **UF Student Science Training Program (SSTP)**
- **University of Florida Research Computing - HiPerGator**