Histogram Approach: Deep Learning Texture Analysis

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

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

Abstract

The ABSTRACT is to be in fully-justified italicized text, at the top of the left-hand column, below the author and affiliation information. Use the word "Abstract" as the title, in 12-point Times, boldface type, centered relative to the column, initially capitalized. The abstract is to be in 10-point, single-spaced type. Leave two blank lines after the Abstract, then begin the main text. Look at previous CVPR abstracts to get a feel for style and length.

1. Introduction

Texture analysis is a process of characterizing texture in an image by spatial variation in pixel intensity values or gray levels. Image texture is a complex visual pattern that is one of the crucial sources of visual information. Some of the texture properties include but not limited to perceived lightness, uniformity, density, roughness, and granulation [5]. This field has been one of the popular research topics in computer vision and machine learning due to its broad impact on numerous fields [2]. In medical imaging, texture analysis can quantify the tumor heterogeneity based on MRI, CT, or PET images. [6] In [10], researchers implement texture analysis based on leaf images to classify various crop diseases.

Approaches to texture analysis are categorised into four approaches: structural, statistical, model-based, and transform. **Structural analysis** represent texture by microtexture and macrotexture based on pre-defined primitives and the placement rules. **Statistical approaches** represent texture indirectly with methods such as analyzing statistics given by pairs of pixels. **Model-based texture analysis** interprets an image texture using fractal and stochastic models. Lastly, the **transform methods** of texture analysis use methods such as Fourier, Gabor, and Wavelet transforms to interpret texture characteristics such as frequency or size. [5]

1.1. Related Works

Texture analysis is similar to object recognition but it differs because spatial relationships of patterns in the image are important for most texture analysis approaches, as opposed to points of interests for object recognitions [2]. Texture-based datasets also have higher dimensionality compared to simple color and shape-based datasets used in object recognition, which makes it a harder task [1]. In order to confront the complexity of texture datasets, several different methods to perform texture analysis have been proposed throughout the literature.

One of the previous approaches is a traditional feature approach where researchers used different texture descriptors to observe the region homogeneity and the histograms of these region borders. In [1], the authors present a review of various hand-crafted features including traditional texture descriptors such as Local Binary Patterns (LBP)[9] and Gray-Level Co-Occurrence Matrices (GLCM)[4] as well as a patch- and multiscale-based approaches. These methods have been used for the past decades and have proved their effectiveness in various applications.

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. Rather than manually designing features for the machine learning model, deep learning approaches are capable of automatically extracting features from labeled datasets and yield higher performance. However, deep neural networks such as Convolutional Neural Network (CNN) require a copious amount of labeled data along with immense amounts of processing power.

In recent years, there have been several studies where researchers combined the deep learning approach and traditional hand-crafted features to create a hybrid model.

Nguyen et al. [7] proposed a new presentation attack detection (PAD) method that uses (CNN) and the multi-level local binary pattern (MLBP) method together to form a hybrid feature with a higher discrimination ability. In [8], the authors proposed a hybrid model which can learn from both deep learning features derived from spectral data as well has hand-crafted features derived from synthetic aperture radar (SAR) imagery and elevation. Some of these hybrid models do not require an immense amount of labeled data but combine automated feature learning and traditional hand-crafted feature to improve performance [8]. Zhang et al., proposed a Deep Texture Encoding Network where they added an encoding layer on top of convolutional layers [13]. This model was able to learn convolutional features and encoding representation simultaneously. A theoretical analysis was on implementing deep neural networks specifically for texture classification purposes was conducted to demonstrate the need for the integration of traditional and neural features [1]. These pioneering attempts showed improved performance compared to previous methods with deep learning algorithms or hand-crafted features alone.

There were also papers which focused on implementing histogram layers for various applications. In [11], the authors added a histogram layer to the CNN to mimic existing features such as projection spatial rich model (PSRM) and proposed the model for steganalysis applications. By implementing a new histogram layer, they were able to capture the statisctics from the feature maps with the histogram layer. Wang et al [12] proposed a learnable histogram layer that can backpropagate errors as well as learn the optimal bin centers and widths. Two architectures were developed for object detection and semantic segmentation, and both models were able to achieve high performance. However, previous studies were limited to using global histograms which caused loss in spatial information. They were also limited to using a single histogram layer.

1.2. Goal of Research

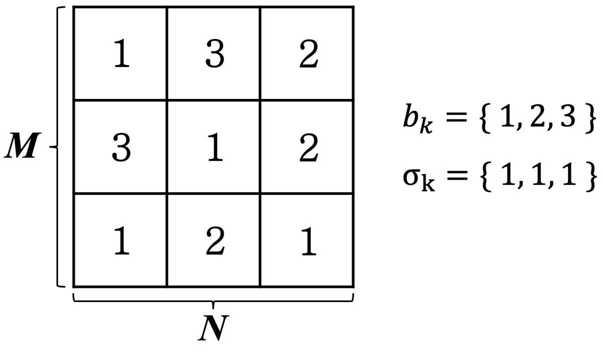

Figure 1. M×N Window

In this paper, we propose a novel model that incorporates a localized histogram layer for convolutional neural networks (CNNs). Our studies will be the first attempt to implement such a hybrid model for texture analysis applications and implementing localized histograms will provide several advantages. 1) Spatial information, which is important for texture, will be retained as opposed to previous global methods. 2) Our approach will use radial basis functions (RBFs) which will relax the binning constraint because each feature value's contribution will be based on their proximity to each bin center and associated bin width. Additionally, this will enable the network to be less sensitive to various outliers and ambiguity in the data. 3) This architecture will allow a differentiable histogram operation for deep learning algorithms as well. 4) Lastly, our model will have a stackable histogram layers which is capable of capturing more higher-level features.

2. Methodology

2.1. Binning Operation

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

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

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:

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

The equation 2 returns the count y_k, for kth bin center μ_k and width σ_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.

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

	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

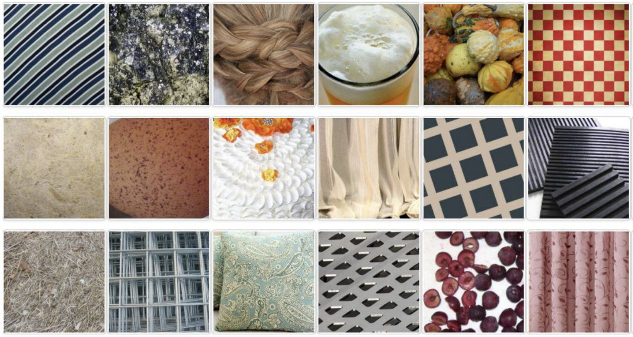

Figure 2. Example of DTD Images [3]

$w = 0$, 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, σ_k is set to be 1, which makes the difference between the two values larger. σ_k is a crucial factor that determines this difference. As σ_k value becomes smaller, the model will only consider the values that are very close to the bin center. If model learns the optimum σ_k value, we will be able to account for ambiguity in the data as opposed to crisp histograms. As σ_k value becomes larger, every element will fall into the same bin.

2.2. Experiment Design

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.

2.3. 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, μ is the mean of that feature value, σ 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 (μ) and standard deviation (σ) 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 [**?**], all the samples were first resized to 256 × 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 224×224 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[**?**].

References

[1] Saikat Basu, Supratik Mukhopadhyay, Manohar Karki, Robert DiBiano, Sangram Ganguly, Ramakrishna Nemani, and Shreekant Gayaka. Deep neural networks for texture classification—a theoretical analysis. *Neural Networks*, 97:173 – 182, 2018.

[2] P. Cavalin and L. S. Oliveira. A review of texture classification methods and databases. In *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*, pages 1–8, Oct 2017.

[3] 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.

[4] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, Nov 1973.

[5] Andrzej Materka and Michal Strzelecki. Texture analysis methods – a review. Technical report, INSTITUTE OF ELECTRONICS, TECHNICAL UNIVERSITY OF LODZ, 1998.

[6] Jakub Nalepa, Janusz Szymanek, Michael P. Hayball, Stephen J. Brown, Balaji Ganeshan, and Kenneth Miles. Texture analysis for identifying heterogeneity in medical images. In Leszek J. Chmielewski, Ryszard Kozera, Bok-Suk Shin, and Konrad Wojciechowski, editors, *Computer Vision and Graphics*, pages 446–453, Cham, 2014. Springer International Publishing.

[7] Dat Nguyen, Tuyen Pham, Na Rae Baek, and Kang Ryoung Park. Combining deep and handcrafted image features for presentation attack detection in face recognition systems using visible-light camera sensors. *Sensors*, 18:699, 02 2018.

[8] Rahul Nijhawan, Josodhir Das, and Balasubramanian Raman. A hybrid of deep learning and hand-crafted features based approach for snow cover mapping. *International Journal of Remote Sensing*, 40(2):759–773, 2019.

[9] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, July 2002.

[10] L. S. Pinto, A. Ray, M. U. Reddy, P. Perumal, and P. Aishwarya. Crop disease classification using texture analysis. In *2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pages 825–828, May 2016.

[11] Vahid Sedighi and Jessica Fridrich. Histogram layer, moving convolutional neural networks towards feature-based steganalysis. *Electronic Imaging*, 2017:50–55, 01 2017.

[12] Zhe Wang, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang. Learnable histogram: Statistical context features for deep neural networks. *CoRR*, abs/1804.09398, 2018.

[13] Hang Zhang, Jia Xue, and Kristin J. Dana. Deep TEN: texture encoding network. *CoRR*, abs/1612.02844, 2016.