**Prior Knowledge Based 1st Layer in Deep Learning**

**(Only draft, 80% )**

Dinh-Thuan Dang1, Jing-Wein Wang2

1Department of Electronic Engineering, National Kaohsiung University of Scienceand Technology, Kaohsiung 80778, Taiwan

[ddthuan@pdu.edu.vn](mailto:firstname.secondname@springernature.com)

2Institute of Photonic Engineering, National Kaohsiung University of Science andTechnology, Kaohsiung 80778, Taiwan

jwwang@nkust.edu.tw

**Abstract.** In this paper, we evaluate 2D Scattering transform in image classification. We design some neural layers is equivalent to 2D Scattering transform in neural network. We compare this two methods to classify defects in the NEU surface defect database.

**Keywords:** Scattering transform, neural network, deep learning, design neural network, classification, NEU surface.

# Introduction

Convolutional Neural Networks (CNN) play a significant role in driving a variety of technologies including computer vision, natural language processing, and speech recognition.

In an effort to improve classification performance, CNN sizes have grown dramatically trainable parameters. Complexity has become an important consideration as CNN have come into use in practical systems. In particular, the storage and computational complexities, as well as energy consumption, are important considerations in both training and inference modes.

The methods proposed to address the issue of large models can be categorized into three general areas. One general approach is pre-defined constrained filter design, where in the standard CNN filter kernels are constrained in some fashion to reduce the computational and storage complexity. Primary examples in the category are MobileNet and ShuffleNet which rely on separable and grouped filter kernels, respectively. A second group of techniques is pruning during training, wherein parameters are removed from the model as training is performed to produce a low-complexity trained model for inference. A third category for complexity reduction is weight quantization, wherein the trained weights are grouped and quantized to save storage, possibly including an iterative training procedure.

Scattering transform can belong to the first method. A scattering transform builds invariant, stable and informative signal representations for classification. It is computed by scattering the signal information along multiple paths, with a cascade of wavelet modulus operators implemented in a deep convolutional network. It is stable to deformations, which makes it particularly effective for image, audio and texture discrimination.

We do:

* Intergrate scattering transform to extracts features in neural network.
* ***Build neural layers equivalente*** scattering transform.
* Classify NEU surface database.

# Related work

## Scattering transform

The Scattering Transform has been first defined and studied mathematically in [70, 69]. It has been successfully used in classification task of stationary textures and small digits [13, 15]. As a result, state-of-the-art classification results are obtained on hand-written digit recognition and texture classification.

**Scattering transform** build invariant, stable and informative representations through a non-linear, unitary transform, which delocalizes signal information into scattering decomposition paths. They are computed with a cascade of wavelet modulus operators, and correspond to a convolutional network where filter coefficients are given by a wavelet operator.

Their invariance and stability properties, scattering operators linearize deformations. This linearization property can be exploited to build linear generative classifiers in the scattering domain, which are computed with simple class-conditional PCA. When applied to stationary textures, scattering transforms provide new texture descriptors, incorporating high order moments which can discriminate non-Gaussian properties.

**Scattering transform** is defined as a complex-valued convolutional neural network whose filters are fixed to be wavelets and the non-linearity is a complex modulus. Each layer is a wavelet transform, which separates the scales of the incoming signal. The wavelet transform is contractive, and so is the complex modulus, so the whole network is contractive. The result is a reduction of variance and a stability to additive noise. The separation of scales by wavelets also enables stability to deformation of the original signal. These properties make the scattering transform well-suited for representing structured signals such as natural images, textures, audio recordings, biomedical signals, or molecular density functions.

Let us consider a set of wavelets , such that there exists some satisfying:

Given a signal , we define its scattering coefficient of order k corresponding to the sequence of frequencies to be:

The fig shows the 2nd order scattering transform tree. Each modulus uses 1 low-pass filter and 2 band-pass filter.

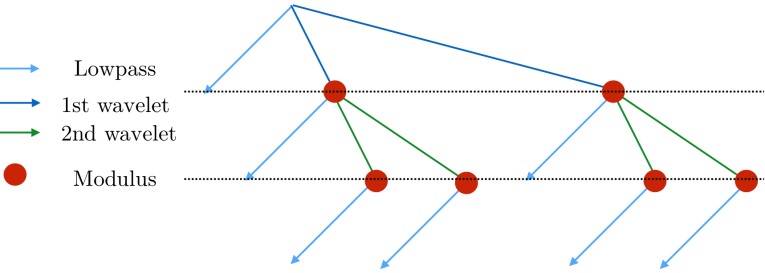


Fig.

## 2nd Order scattering transform

Consider a signal with finite energy, i.e. , with u the spatial position index and an integer , which is the spatial scale of our scattering transform. Let be a local averaging filter with a spatial window of scale . This filter is chosen to be dilated from a low-pass filter that builds invariance up to 2:

can be set as:

A local averaging operator as:

Applying the latter on the signal we obtain the zeroth order scattering coefficient:

A solution to avoid the loss of high frequency information is provided by the use of wavelets. A family of wavelets is obtained by dilating and rotating the complex mother wavelet such that:

, where is the rotation by , and is the scale of the wavelet.

A given wavelet has thus its energy concentrated at a scale j, in the angular sector . Let be an integer parametrizing a discretization of . A wavelet transform is the convolution of a signal with the family of wavelets introduced above, with an appropriate downsampling:

Observe that and have been discretized: the wavelet is chosen to be selective in angle and localized in Fourier. With appropriate iscretization of is approximatively an isometry on the set of signals with limited bandwidth, and this implies the energy of the signal is preserved: there exists , for such signals ,

To achieve invariance, we apply a non-linear point-wise complex modulus to , followed by an averaging , which builds a non trivial invariant. Here, the mother wavelet is analytic, thus is regular which implies that the energy in Fourier of is more likely to be contained in a lower frequency domain than . Thus, preserves more energy of . It is possible to define , which can also be written as:

Again, the use of the averaging builds an invariant to translation up to . Once more, we apply a second wavelet transform , with the same filters as , on each channel. This permits the recovery of the high-frequency loss due to the averaging applied to the first order, leading to:

which can also be written as:

Filter

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

Each row: 1 low-pass filter 1st column , 8 Band-pass filter

# Design equivalente neural model

## 2D scattering transform intuition



|  |  |  |
| --- | --- | --- |
| Fig a | Fig b | Fig c |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| a | b | c | d |
|  |  |  |  |

## Equivalente neural layers

**Scattering transform as a neural network**

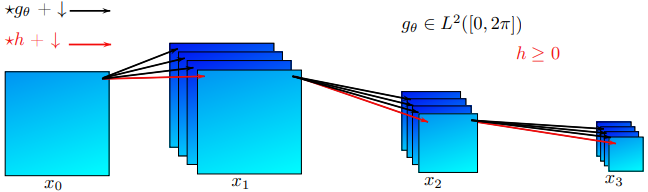


Fig.

and are convolution operator, is down-sample (pool) operator.

**Neural design:** the *first convolution* layer has 3 input color channels. We use 9 filters, equivalente to 1 low-pass filter and 8 bank-pass filters. So we have 27 output channels, kernel size is 5. The *second convolution* layer has 81 output channels, kernel size is 5. The *third convolution* layer has 243 output channels, kernel size is 5. Right behind convolution 1 and convolution 2 layer is pool layer. Through each pool layer, image size reduced by half. If image size is [64,64], output of scattering layer is [16, 16]. Scattering layer code is descibled in Pytorch below.

self.scattering = nn.Sequential(OrderedDict([

('**conv1**', nn.Conv2d(3, 27, kernel\_size=5, padding=2)),

('pool1', nn.MaxPool2d(kernel\_size=(2, 2))),

('relu1', nn.ReLU()),

('**conv2**', nn.Conv2d(27, 81, kernel\_size=5, padding=2)),

('pool2', nn.MaxPool2d(kernel\_size=(2, 2))),

('relu2', nn.ReLU()),

('**conv3**', nn.Conv2d(81, 243, kernel\_size=5, padding=2)),

('relu3', nn.ReLU())

]))

This output is equivalente to 2D scattering transform with J=2.

## Deep network architecture for classification task

## 

## Fig.

# Results

## Dataset, metric

**NEU-CLS surface defect database**

Fig. shows the sample images of some kinds of typical surface defects. we can clearly observe that the intra-class defects existing large differences in appearance, for instance, the scratches (the last column) may be horizontal scratch, vertical scratch, and slanting scratch, etc.

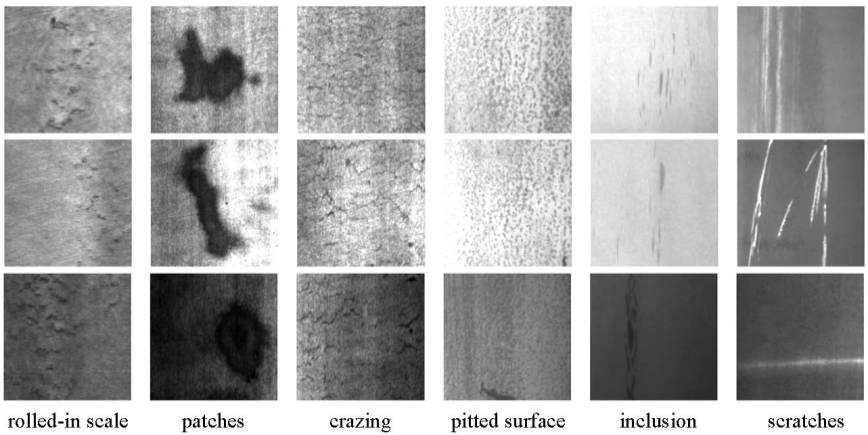
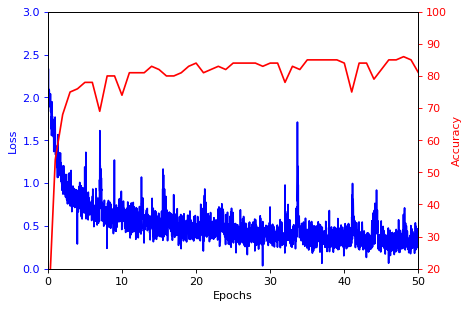
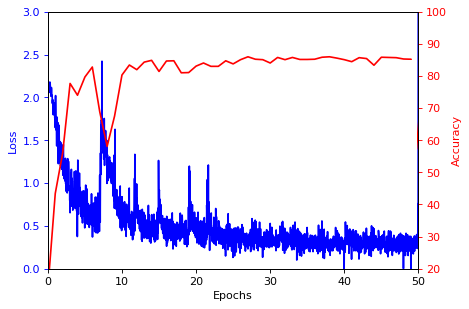


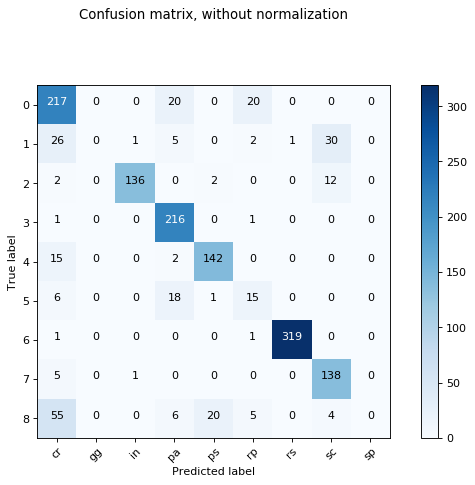
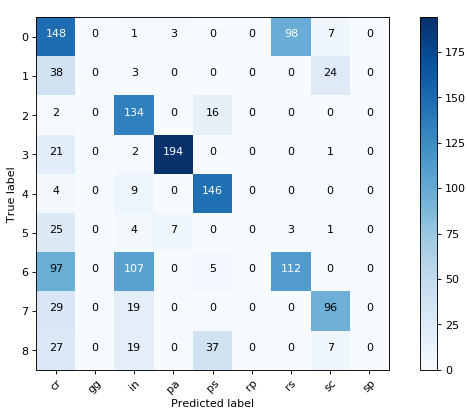
Fig.

**Accurance results.**



Deep learning Scattering transform

**Confusion Matrix results.**



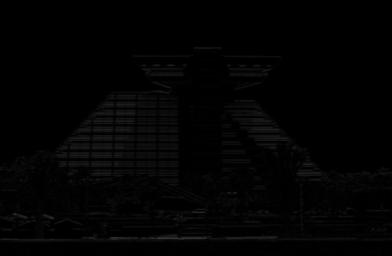
# Conclusions

# Appendix

















# References (in MathPhySci)

# References (in Basic)