


Article

Target Classification in Synthetic Aperture Radar Images Using Quantized Wavelet Scattering Networks [†]

Raghu G. Raj ¹, Maxine R. Fox ^{1,2} and Ram M. Narayanan ^{2,*} 

¹ U.S. Naval Research Laboratory, Radar Division, Washington, DC 20375, USA; raghu.raj@nrl.navy.mil (R.G.R.); maxine.fox@nrl.navy.mil (M.R.F.)

² Department of Electrical Engineering, The Pennsylvania State University, University Park, State College, PA 16802, USA

* Correspondence: ram@engr.psu.edu; Tel.: +1-814-863-2602

[†] This paper is an extended version of our conference paper: Fox, M.R.; Raj, R.G.; Narayanan, R.M. Quantized wavelet scattering networks for signal classification. In Proceedings of the SPIE Conference on Radar Sensor Technology XXIII, Baltimore, MD, USA, 15–17 April 2019; pp. 110030V-1–110030V-10.

Abstract: The need to classify targets and features in high-resolution imagery is of interest in applications such as detection of landmines in ground penetrating radar and tumors in medical ultrasound images. Convolutional neural networks (CNNs) trained using extensive datasets are being investigated recently. However, large CNNs and wavelet scattering networks (WSNs), which share similar properties, have extensive memory requirements and are not readily extendable to other datasets and architectures—and especially in the context of adaptive and online learning. In this paper, we quantitatively study several quantization schemes on WSNs designed for target classification using X-band synthetic aperture radar (SAR) data and investigate their robustness to low signal-to-noise ratio (SNR) levels. A detailed study was conducted on the tradeoffs involved between the various quantization schemes and the means of maximizing classification performance for each case. Thus, the WSN-based quantization studies performed in this investigation provide a good benchmark and important guidance for the design of quantized neural networks architectures for target classification.

Keywords: adaptive wavelet scattering network; backpropagation; classification; convolutional neural networks; MSTAR; synthetic aperture radar; wavelet scattering network



Citation: Raj, R.G.; Fox, M.R.; Narayanan, R.M. Target Classification in Synthetic Aperture Radar Images Using Quantized Wavelet Scattering Networks. *Sensors* **2021**, *21*, 4981. <https://doi.org/10.3390/s21154981>

Academic Editors: Mengdao Xing and Jiayi Ma

Received: 19 April 2021

Accepted: 20 July 2021

Published: 22 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Feature extraction and classification are essential ingredients in imagery analysis in myriad applications: remote sensing, military, nondestructive testing, ultrasound, medical, cell analysis, etc. In particular, image classification is the process of organizing images into different classes based on the output of feature extraction operators applied to images. There are innumerable approaches to feature extraction, a necessary precursor to classification, including decision-theoretic approaches using quantitative descriptors such as length, area, and texture [1,2]; structural approaches using qualitative descriptors, such as relational descriptors [3]; projection of data into fixed basis sets, such as wavelets [4] and Zernike polynomial moments [5], or adaptive basis sets [6]. Other examples include robust edges and corners that are popular in computer vision, blind synthesis of template classes by using singular value decomposition, Karhunen–Loeve Transform [7,8] and estimation theoretic templates [9], motion-based covariance matrix-based features for multi-sensor architectures [10], and finally micro-Doppler- [11] and vibrometry-based [12] features that have applications in radar-based sensing systems. The advent of deep neural networks, a variant of which is the focus of our work, has systematized to a large extent the process of feature extraction and classification.

Prior to feature extraction, several pre-processing steps are required to recognize targets/objects embedded within an image. For example, a common first step for image analysis is image segmentation [13]. Depending on the available independent knowledge about an image, specialized approaches to segmentation to isolate regions containing features of interest may be applied [14]. Another technique, image clustering, can be used to divide images, subregions, or even features into similar groups or “clusters”. For multiband images and multi-dimensional data, cluster analysis is usually encountered to find the clusters and fit boundaries between them that can be used for classification.

Neural networks, which are used to analyze and process images, have been shown to largely circumvent excessive dependence on pre-processing issues such as segmentation, etc., as described above. In particular, neural networks can be based on task or abstraction levels. Tasks include spatial or spectral filtering, feature extraction, object detection and recognition, and image understanding. Abstraction levels include pixels, features, measurement values, object relationships, and scene types.

Overfitting is a common problem in image classification because there are usually too few training samples, resulting in a model with poor generalization performance. One solution to overfitting is to use data augmentation, which is a technique to generate more training data from the current training set. It is an artificial way to boost the size of the training set, reducing overfitting. Data augmentation is typically done by data transformations and processing, such as rotation, shifting, resizing, adding noise, contrast change, etc. Data augmentation is only performed on the training data, not on the validation or test set.

Recently, convolutional neural networks (CNNs) have achieved state-of-the-art performance on several benchmark tasks in the computer vision literature. This suggests the possibility of applying CNNs to high resolution images for detecting and classifying features and objects. CNNs are a supervised classification model with many more free model parameters compared to other supervised methods. As a result, CNNs need large quantities of training data [15]. This is a major challenge for many applications, wherein the collection of training data is time consuming and expensive.

CNNs have found application to synthetic aperture radar (SAR) in recent works using datasets such as the publicly available MSTAR dataset [16–21]. These studies typically used predefined network architectures that can contain many learnable parameters; other available architectures can contain upwards of a million parameters [22–24], which require significant amounts of memory. To reduce needed memory, the quantization of generalized or otherwise large CNNs has been studied for the classification of optical datasets [25,26]. Despite promising results, extensions to other datasets and network architectures are not proven easily due to their variability. The success of a CNN design is dependent upon many factors, including the arrangement of layers the number and initialization of learnable parameters in each layer, the learning rate and update methods implemented during training, and the size and complexity of the available training data. Given such variability, the development of a network for benchmarking these designs would prove useful.

Several authors have implemented wavelet scattering networks (WSNs) to demonstrate the paramount properties of CNNs, particularly convolution, nonlinearity, and the layered architecture [27–29]. With far fewer design parameters and lower memory requirements, the similar functionality of a WSN may be utilized as a benchmark for comparing quantization schemes among different CNN architectures in future work.

The foundation of a WSN, the wavelet scattering transform, is itself an effective instrument in feature extraction due to its provision of translation invariance, stability, and the ability to linearize small diffeomorphisms that result from its layered architecture of scattering wavelets. It is even used as a preprocessing measure wherein a WSN performs preliminary feature extraction prior to the training of a deep neural network (DNN) for localization [30]. The freedom of choosing an appropriate kernel of a linear transform has been exploited fully, which is generally known as the adaptive wavelet transform [31]. Several levels of adaptivity were investigated in detail. To overcome the problem that the

noise cannot generally be predicted in a noisy signal, an adaptive filter based on the wavelet transform method was implemented [32]. The results show that the Hopfield neural network adaptive filter model based on wavelet transform achieved the best denoising effect. The adaptive wavelet invariant moments (AWIM) formulation was proposed and developed to ensure that the discrete wavelet transform (DWT) coefficients were both translation and scale invariant [33]. This technique was successfully implemented for scale-invariant texture identification.

The WSN in [27] computed a translation-invariant image representation which was stable to deformations and preserved high-frequency information for classification. It was implemented by cascading wavelet transform convolutions with nonlinear modulus and averaging operators. A detailed mathematical analysis of WSNs explained important properties of deep convolution networks for classification. A windowed scattering transform was defined by a local integration, and, as the window size increased, it converged to a wavelet scattering transform that was translation invariant [28].

An architecture employing a deep WSN to extract translation- and rotation-invariant features was used by a conditional random field to perform scene segmentation on aerial images recorded from an unmanned aerial vehicle [34]. A parametric log transformation with dual-tree complex wavelets was proposed and implemented to extract translation-invariant representations from a multi-resolution image [35]. The parametric transformation improved the pruning algorithm, while the dual-tree wavelets improved the computational efficiency of the network.

A two-layer WSN was presented for object classification [29]. This scattering transform computed a spatial wavelet transform on the first layer and a new joint wavelet transform along spatial, angular, and scale variables in the second layer. By applying a WSN in various color spaces, color texture classification was achieved, with the opponent RGB-based WSN outperforming other color spaces [36].

A framework was implemented to classify cell images based on WSNs and random forests [37]. The WSN computed rotation-invariant wavelet coefficients as representations of cells images, while a random forest classifier was trained to predict the pattern label of a cell image from six classes. The feasibility of deep WSN-based features for classification of ultrasound images acquired in a muscle computer interface was investigated [38]. Deep learning-based feature extractors were seen to be invariant to translation, rotation, and slight deformations, thereby preserving motion classification accuracy.

A wavelet scattering transform was used to extract reliable features that are stable to small deformation and are rotation-invariant when applying an artificial neural network (ANN) to indoor fingerprinting-based localization, where the signal is inherently unstable [30]. The extracted features were used by a DNN model to predict the location. An automatic target recognition method for SAR images was developed based on a super-resolution generative adversarial network (SRGAN) and deep convolutional neural network (DCNN) [39]. The approach was able to suppress background clutter, enhance target feature characterization ability, and achieve automatic target classification and recognition.

The roto-translation properties of the WSN were incorporated in a convolutional architecture to construct a rotation invariant CNN for image classification [40]. Another scale- and rotation-invariant feature extraction method, the speeded-up robust features (SURF) is a local feature detector and descriptor that utilizes multi-scale representation based on box filters [41]. The binary robust independent elementary features (BRIEF) descriptor improves upon SURF by reducing computation [42].

Complex-valued CNNs have been proposed which yield improved performance over their real-valued counterparts, especially in those with bigger kernel sizes [43]. These can be likened to nonlinear multiwavelet packets, thus making the mathematical analysis from the signal processing perspective available for a rigorous formulation of the properties of complex-valued convolutional networks. Moreover, these are more relevant because many images, especially SAR images, are in complex form.

Expanding upon our preliminary work in [44], this paper systematically explores the application of quantized WSNs to target classification of SAR imagery for a large range of SNR conditions; in particular, we used the MSTAR SAR dataset to validate the techniques presented in this paper. After presenting a quantitative description of the implementation of the important quantization schemes, a detailed study was conducted on the tradeoffs involved between the various quantization schemes and the means of maximizing classification performance for each case. Finally, due to its structural similarity with CNNs, the WSN-based quantization studies performed in this paper may provide a good benchmark and important guidance for the design of quantized CNN-based neural networks.

The remainder of this paper is organized as follows. Section 2 presents a brief introduction of the basics of WSNs and the windowed scattering transform. Section 3 explores the unique architecture of the WSN and the development of the quantization method and scales. Section 4 describes the methodology for the application of WSNs for classification of the MSTAR dataset. Section 5 presents the results and discussion thereof. Conclusions and recommendations for future work on this topic are presented in Section 6.

2. Wavelet Scattering Networks Fundamentals

The WSN shares the key properties of a CNN, primarily convolution, nonlinearity, and layer-wise architecture. A WSN is a windowed scattering transform that extracts features at multiple resolutions using scattering wavelets, a subset of wavelet filters that aid in achieving translation invariance, stability, and the linearization of small diffeomorphisms. Daughter wavelets are constructed from a mother wavelet, ψ , at various scales and orientations. For a scaling factor of 2^j and rotation angle of θ , a 2D daughter wavelet is

$$\psi_{j,\theta} = 2^{2j}\psi\left(2^j r_\theta^{-1} \vec{u}\right), \quad (1)$$

where r_θ is the rotation matrix and the position vector $\vec{u} = [x, y]^T$. For convenience, this paper uses the notation ψ_λ , where $\lambda \equiv j, \theta$ indicates a combination of scale and orientation for the remainder of this section, which provides relevant background on the scattering transform, reproduced in part from [26] for better understanding of the WSN architecture during the discussion of the quantization scheme below.

A WSN is a windowed scattering transform, consisting of multiple windowed scattering propagators, typically of multiple scattering orders, like that shown in Figure 1 for a maximal scattering order of $M = 3$. The scattering propagators are formed using a set of daughter wavelets constructed via (1), and the windowing is performed by a son wavelet.

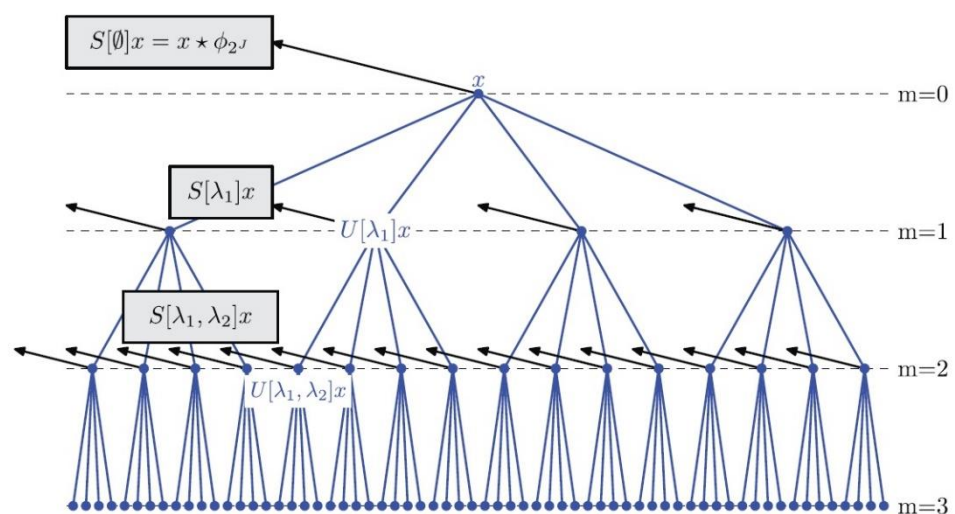


Figure 1. Structure of a wavelet scattering network which computes the windowed scattering transform. Reprinted with permission from ref. [27]. Copyright © 2013 IEEE.

The collection of daughter wavelets, ψ_λ for all $\lambda \in \Lambda$, defined for J scales and L orientations at Q wavelets per octave, act as high-pass filters. The scale of each is $2^{j/Q}$ for unique $j = 0, 1, \dots, J-1$, where Q acts as a quality factor for the employed filter bank; the quality factor is set to unity in this work, as in the example in the ScatNet documentation [45]. In addition, the orientations, $\theta \in \Theta = \{l\pi/L\}_{l=0}^{L-1}$.

The Littlewood–Paley wavelet transform [46,47] of an input image, X , with a daughter wavelet, ψ_λ (wherein each element of X is convolved) is subsampled according to its frequency bandwidth

$$W[\lambda]X = \downarrow_{2^d}(\psi_\lambda * X) \quad (2)$$

where $\downarrow_{2^d}(Z)$ denotes the subsampling operation on Z . The \log_2 subsampling rate d is determined as a function of the \log_2 filter resolution, r_{ψ_λ} , and the resolution of the input at the m th scattering order, $r_X^{(m)}$, such that

$$d = \max\{0, r_{\psi_\lambda} - r_X^{(m)} - \zeta\} \quad (3)$$

where $r_{\psi_\lambda} = j/Q$ and ζ is the oversampling factor. The \log_2 resolution of the output is

$$r_X^{(m+1)} = r_X^{(m)} + d \quad (4)$$

Equation (2) is not translation invariant; therefore, similar to the non-linear activation functions in the CNN, it is passed through a non-linear function for demodulation. The modulus of the output of Equation (2) may be used for this purpose. The complete process of the wavelet transform and its demodulation is described by the scattering operator $U[\lambda]$, given by

$$U[\lambda]X = |W[\lambda]X|. \quad (5)$$

A scattering propagator applies Equation (5) at each λ along a path $p = (\lambda_1, \lambda_2, \dots)$ where each λ_f in p is unique. For an m th-order path, i.e., a path of length m , the scattering propagator is defined as

$$U[p]X = U[\lambda_m] \dots U[\lambda_2]U[\lambda_1]X, \quad (6)$$

where $U[O] = I$ and I is the identity matrix. In a WSN, the paths in the set of paths, P , that define a scattering propagator are unique and vary in length between 0 and the maximal scattering order, M . The number of paths in P is restricted so that only filters of increasing scale can be used, i.e., $j_m = j_{m-1} + Q$.

To form a windowed scattering propagator, a son wavelet, ϕ_J , is constructed from a father wavelet, ϕ , such that

$$\phi_J = 2^{-2(J-1)}\phi\left(2^{-(J-1)}\vec{u}\right), \quad (7)$$

which acts as a low-pass filter. (Note that, if $Q \neq 1$, then the scaling of ϕ_J is $2^{(J-1)/Q}$.) The windowed scattering propagator subsamples the convolution of Equation (7) with the scattering propagator in Equation (6), i.e.,

$$S_J[p]X = \downarrow_{2^d}(U[p]X * \phi_J). \quad (8)$$

Note that the path length of the windowed scattering propagator is the same as the scattering propagator.

Figure 2 provides an example of the paths found in P for $J = 4$, $L = 2$, $Q = 1$, and $M = 2$. Note that these paths overlap: For example, the path $p = ((0, \theta_1))$ is part of the path $p = ((0, \theta_1), (1, \theta_2))$. The output from each $U[p]$ is passed to ϕ_4 , as well as to all orientations of the high-pass filters with scales 2^j , such that $j_{m-1} + 1 \leq j < J$, where $2^{j_{m-1}}$ is the scale of the most recent filter along the path.

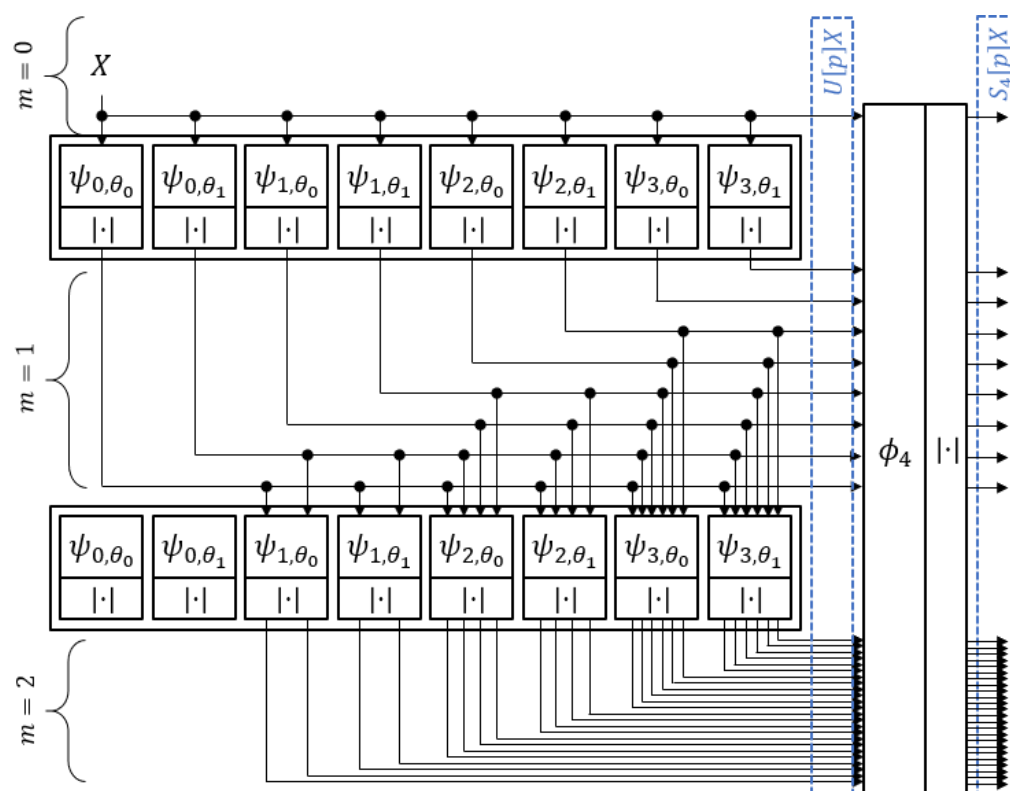


Figure 2. WSN paths with $J = 4$, $L = 2$, and $M = 2$.

Each $S_j[p]X$ is a feature map akin to those output from the convolutional layers of a CNN and may be input to a classifier, such as a support vector machine (SVM) or the stable SoftMax function, following additional processing.

The combinations of M , J , and L implemented in this section are provided in Table 1. Three and five scales were used to understand the effect of both the scale, as well as the network complexity. Both scales were used for $M = 0$ networks to better understand the effect of the scale in ϕ_j on the output, as they should behave as CONV layers with one filter. For $M > 0$, differing J and L were implemented to explore the effect of network complexity, particularly with the application of the updates during backpropagation. Note that, in the ScatNet framework, at $M = 1$, all daughter wavelets ψ_λ are utilized once at $r = 0$ and the windowing function ϕ_j at multiple resolutions depending upon the value of J ; at $M = 2$, all ψ_λ with $j > 2$ are utilized at more than one resolution.

Table 1. System parameters for WSN.

M	J	L	Number of Paths
0	-	-	1
1	3	1	4
1	3	2	7
1	3	2	25
1	5	1	6
1	5	2	11
1	5	8	41
2	3	1	7
2	3	2	19
2	5	1	16
2	5	2	51

3. Quantization of a Wavelet Scattering Network

The WSN requires less memory than the large CNNs used for classification applications. The parameters of consequence in a WSN include the following: number of wavelet scales (J), number of wavelets per octave (Q), number of wavelet orientations (L), maximal scattering order (M), and oversampling factor (ζ). Due to its functional similarity to CNNs, a WSN can provide a benchmark for the comparison of quantization schemes. This section explores the quantization of a wavelet scattering network using a set encoding method.

For an input to a network, X , each windowed scattering propagator, $S_J[p]X$, for all paths $p \in P$, as well as the intermediate operations of the propagator, $U[\lambda_1]X$, $U[\lambda_1, \lambda_2]X, \dots, U[p]X$, are quantized; thus, the output along a path of length T undergoes $T + 1$ quantizations. The calculation and application of quantization levels is based on the ScatNet [45] implementation, wherein the m th order scattering operations, $U[p^{(m)}]X = U[\lambda]U[p^{(m-1)}]X$, and the windowing operations for the previous scattering order, $S_J[p^{(m-1)}]X$, are applied in one iteration of the scattering code.

The outputs, $y = (U[p^{(m)}]X \forall p^{(m)} \in P, S_J[p^{(m-1)}]X \forall p^{(m-1)} \in P)$ from this scattering layer, or s-layer, are fed to a quantization layer, or q-layer, wherein each input, $Y \in y$, is quantized to K unique levels. These quantization levels are generated using the values in Y , such that each windowed scattering propagator with q-layers is defined using a quantization operator Q :

$$S_J[p]X = QS_JQU[\lambda_m]QU[\lambda_{m-1}] \dots QU[\lambda_2]QU[\lambda_1]QX. \tag{9}$$

Figure 3 provides a depiction of the s-layers and their corresponding q-layers for an $M = 2$ network with $L = 1$. First, the input X is quantized, then passed to the first s-layer ($m = 1$). The output from the operation with $U[\lambda_1]$ is then quantized; the output from the operation with $U[\lambda_2]$ is then quantized; etc. Lastly, the output from the operation with $S_J[\phi]$ is quantized. This process is repeated for each scattering order of the network. Note that the quantization processes within all q-layers in a network are isolated events.

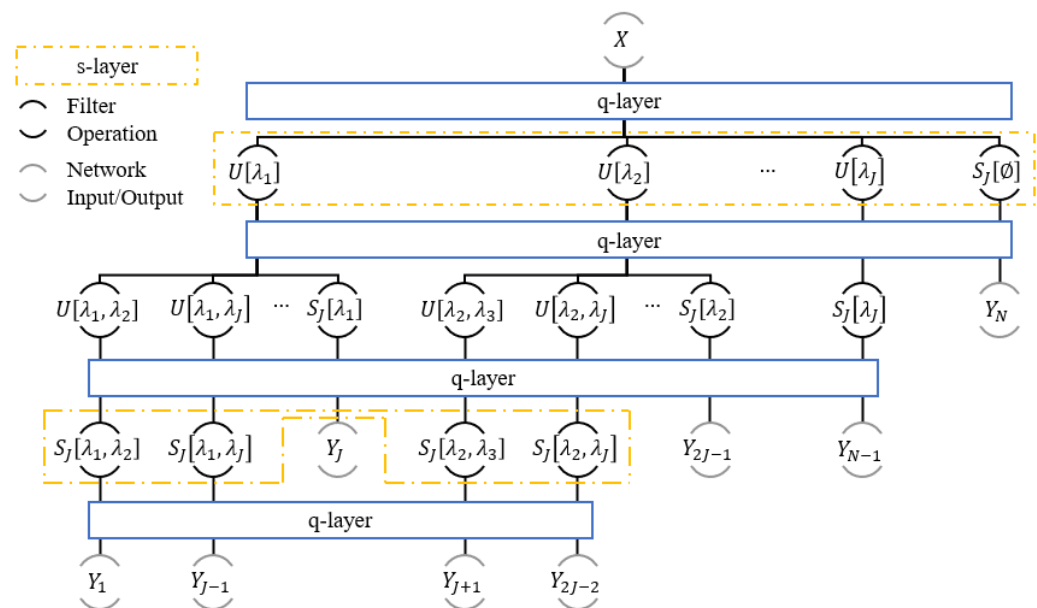


Figure 3. Location of the q-layers in an $M = 2$ WSN where $L = 1$.

The purpose of this quantization scheme is to provide an initial comparison for quantization by limiting the number of unique values permitted. The effectiveness of such a quantization scheme is dependent upon the size of each $Y \in y$. Suppose that $|Y| = N$ for an s-layer. If each value in Y requires b bits for representation, then the total number of bits

required to represent all V values in Y is Vb . However, the sizes of the N outputs contained in y vary with the scales of the filters used in their computation, with the maximum number of values \hat{V} in Y given by $\hat{V} \leq D_X$, where D_X is the number of values in the input to the network, X . In addition, the number of outputs in y varies with the depth of the s -layer in the network.

Figure 4 provides an example for a WSN for $J = 5, L = 1, Q = 1, \zeta = 1$, and $M = 5$. In Figure 4a, the number of calls to each scattering wavelet filter, ψ_λ , with $j > 1$, increases and then decreases with each s -layer; ψ_λ with $j = 1$ are called once only in the first and second s -layers; ψ_λ with $j = 0$ is called only once in the first s -layer. The windowing filter, ϕ_J , is called during each s -layer, once for each call to a ψ_λ in the previous s -layer and once in the first s -layer. The \log_2 subsampling rate of each output, d_f , provided in Figure 4b, is used to determine the size of the output from each filter, $D_x/2^{d_f}$, where D_x is the number of elements in the input. The size of the output along any path is quantified as the sum of the \log_2 down-sampling rates or the \log_2 resolution of the output, which is shown in Figure 4c.

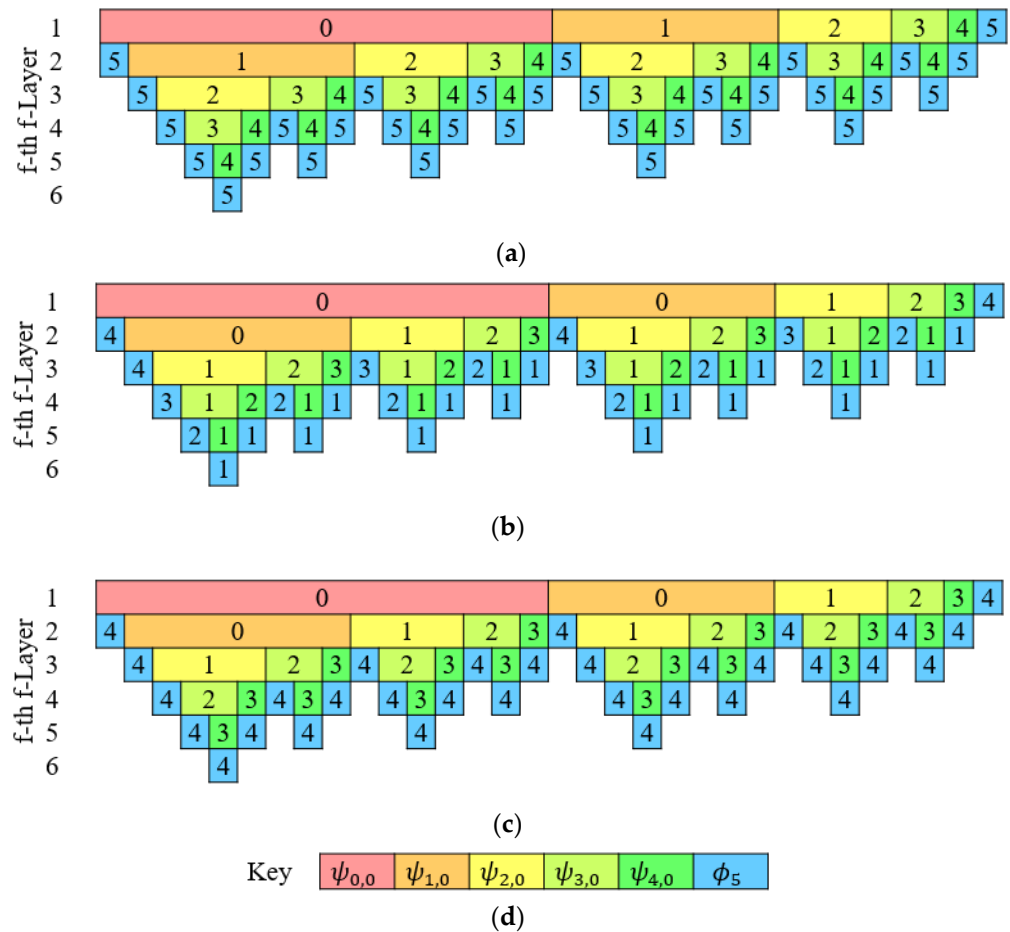


Figure 4. Example of a quantization scheme for a WSN with $J = 5$ and $L = 1$ at the input of each filter for all possible s -layers in the network: (a) filter scales ψ_λ and ϕ_J ; (b) \log_2 subsampling rate d_f ; (c) cumulative \log_2 subsampling rate $r_j = \sum d_f$; (d) color key for filter scales.

For a general network with $Q = 1$, the size of the output images and the number of outputs from an s -layer are derived in the following two subsections.

3.1. Sizes of Filter Outputs

The input along a path of length $T, p^{[T]}$, undergoes $T + 1$ operations; that is, the input passes through T bandpass filters associated with the operators $U[\lambda_i], \psi_{\lambda_1}, \psi_{\lambda_2}, \dots, \psi_{\lambda_i}, \dots, \psi_{\lambda_T}$, with scales 2^{j_i} ($j_i = 0, 1, \dots, J - 1$) and one windowing filter, ϕ_J , with scale 2^J . For sim-

plicity, the f th intermediate output along a scattering propagator $U[p^{(T)}]X$ is denoted as $U[p_f^{(T)}]X$, such that $U[p_f^{(T)}]X = U[p^{(f)}]X$, where $f \leq T$.

Setting $Q = 1$, the resolution, $r_{\psi_\lambda}^{(f)}$, of a filter used in the f th s-layer reduces to the scale of the filter, j_f , thereby reducing the \log_2 subsampling rate, d_f , to [48]

$$d_f = \begin{cases} 0, & j_f < r_X^{(f)} + \zeta \\ j_f - r_X^{(f)} - \zeta, & j_f \geq r_X^{(f)} + \zeta \end{cases}, \quad (10)$$

where $r_X^{(f)}$ is the resolution of the input to the f th filter.

Equation (4) can be reduced by examining the two cases of Equation (10). In the first case, where $j_f < r_X^{(f)} - \zeta$, both d_f and $r_X^{(f)}$ are always 0; this allows the simplification of the case statement to $j_f < \zeta$. For the second case, where $j_f \geq r_X^{(f)} - \zeta$, substituting Equation (10) into Equation (4) yields

$$r_X^{(f+1)} = j_f - \zeta. \quad (11)$$

This result is independent of $r_X^{(f)}$, and therefore independent of the scales of the previous filters along the path. This simplifies the case statement to $j_f \geq \zeta$.

Similarly, the resolution of the output from the f th filter in a path is given by

$$r_X^{(f+1)} = \begin{cases} 0, & j_f < \zeta \\ j_f - \zeta, & j_f \geq \zeta \end{cases}. \quad (12)$$

Since $r_X^{(f+1)}$ depends only upon the most recent filter scale, the notation $r_j = r_X^{(f+1)}$ will denote the resolution of the output from a filter with scale 2^j for any f .

As described in Equation (3), the resolution of the output is the summation of the \log_2 -down-sampling rate, d_n , for $1 \leq n \leq f$. The final resolution of the output, $r_X^{(T+2)}$, i.e., a feature map, is the summation of d_f for $1 \leq f \leq T + 1$. Therefore, for a network input, $X \in \mathbb{R}^{D_1 \times D_2}$, the number of values in the output, Y , from the f th s-layer is

$$V_j^{(f)} = \frac{D_1 D_2}{2^{2(r_j+1)}}, \quad (13)$$

where the +1 is attributed to the down-sampling by the ϕ_j . The size of an output is independent of the s-layer in which it is produced, again depending upon only the resolution of the output. To reduce the number of unique values in Y , the number of levels $K < V_j^{(f)}$.

The total number of values, $V^{(f)}$, submitted from the f th s-layer to the following q-layer is given by

$$V^{(f)} = A^{(f)}R^{(f)} + \sum_J V_j^{(f)}N_j^{(f)}, \quad (14)$$

where $N_j^{(f)}$ is the number of outputs from each bandpass filter with a scale 2^j , and $A^{(f)}$ is the number of values in each of the $R^{(f)}$ lowpass filter outputs.

3.2. Number of Filter Outputs per s-Layer

The total number of s-layers in the network is equal to $M + 1$. The number of outputs from a filter in each s-layer varies with the maximum scale, J , the number of orientations per scattering wavelet, L , and the scattering order, m , associated with the operations in an s-layer. In the f th s-layer, layer inputs undergo the operations $U[p^{(m)}]\forall p^{(m)} \in P$ and $S_J[p^{(m-1)}]\forall p^{(m-1)} \in P$.

If $M = 0$, then the total number of operations in the first s-layer, and the entire network, is one. However, if $M > 0$, the total number of outputs equal the number of filters. Therefore,

$$N^{(1)} = \begin{cases} J \cdot L + 1, & M > 0 \\ 0, & M = 0 \end{cases} \quad (15)$$

The second s-layer accepts $N^{(1)} - 1$ outputs from the first layer $U[p^{(1)}]X \forall p^{(1)} \in P$ as inputs to the filters ($S_J[p]X$ constitutes a terminated path for any p). The windowing filter operates on all $J \cdot L$ inputs to the s-layer, while the bandpass filters only operate on those inputs with paths ending with a smaller scale: An input with path $p^{(1)} = ((j_2, \theta \in \Theta))$ is operated upon by each ψ_λ , where $\lambda = (j, \theta \in \Theta)$ with $j_2 + Q \leq j < J$. Using $a_j = \psi_\lambda$ and $L = 1$, the total number of outputs from the ψ_λ filters can be stated as a function of j as follows,

$$\begin{aligned} \sum_{k=0}^{J-2} \sum_{j=k+1}^{J-1} a_k &= (a_1 + a_2 + \dots + a_{J-2} + a_{J-1}) + (a_2 + \dots + a_{J-2} + a_{J-1}) + \dots + (a_{J-2} + a_{J-1}) + (a_{J-1}) \\ &= a_1 + 2a_2 + \dots + (J-2)a_{J-2} + (J-1)a_{J-1} \\ &= \sum_{m=0}^{J-1} k a_k \\ &= \sum_{m=0}^{J-1} N_k^{(2)} a_k \end{aligned} \quad (16)$$

For $L > 1$, the number of outputs from each ψ_λ is $N_j^{(2)} = jL$. Therefore, the total number of outputs from the second s-layer is

$$N^{(2)} = (N^{(1)} - 1) + L \cdot \sum_{k=0}^{J-1} k. \quad (17)$$

If $M > 2$, then this pattern continues, with each of the $N^{(f-1)} - N^{(f-2)}$ inputs to an s-layer. The number of outputs from any bandpass filter in the s-layer is

$$N_j^{(f)} = \begin{cases} L \cdot \sum_{k=0}^{j-1} N_k^{(f-1)}, & f > 1 \\ 1, & f = 1 \end{cases} \quad (18)$$

The summation may also be expressed in terms of $N_j^{(f)}|_{L=1}$, the number outputs from a filter in the f th layer for $L = 1$ as

$$\begin{aligned} L \cdot \sum_{k=0}^{j-1} N_k^{(f-1)} &= L \cdot \sum_{k_f=0}^{j-1} \left(L \cdot \sum_{k_{f-1}=0}^{k_f-1} \dots \left(L \cdot \sum_{k_1=0}^{k_2-1} N_{k_1}^{(1)} \right) \right) \\ &= L^{(f-1)} \cdot \sum_{k=0}^{j-1} N_k^{(f-1)}|_{L=1} \end{aligned} \quad (19)$$

Furthermore, because $j_f > j_{f-1}$, we have $N_j^{(f)} = 0$ for $j < f - 1$, which modifies Equation (20) to

$$N_j^{(f)} = \begin{cases} L^{(f-1)} \cdot \sum_{k=f}^{j-1} N_k^{(f-1)}|_{L=1}, & j \geq f - 1 > 0 \\ 0, & j < f - 1 \\ 1, & f = 1 \end{cases} \quad (20)$$

The total number of outputs for an s-layer is therefore [48]

$$N^{(f)} = \begin{cases} N^{(f-1)} - N^{(f-2)}, & f = M + 1 \\ N^{(f-1)} - N^{(f-2)} + \sum_{k=f}^{j-1} N_k^{(f)}, & 1 < f \leq M \\ 1, & f = 1 \end{cases} \quad (21)$$

3.3. Quantization Scales

We explore several quantization scales to create K quantization levels. The quantization levels for $Y \in y$ are denoted as v_Q .

3.3.1. Uniform Scale

The uniform, or linear, quantization scale provides a good performance benchmark. The uniform scale was constructed by uniformly spacing values, such that the K levels are

$$v_Q \in (\min(Y), \min(Y) + dv_Q, \min(Y) + 2dv_Q, \dots, \max(Y)), \quad (22)$$

where $dv_Q = (\max(Y) - \min(Y) + 1)/2^K$.

3.3.2. Log Scale

Log-scale quantization provides another simple benchmark; however, to prevent values $v_Q \in Y$ that exist outside the domain of the log function, a value dz must be added to all values in Y . This results in the transformation $Y \rightarrow Z: z = \log(v + dz) \forall z \in Z$. The log-scale is constructed as

$$Z_Q \in (\min(Z), \min(Z) + dv_Q, \min(Z) + 2dv_Q, \dots, \max(Z)), \quad (23)$$

$$v_Q \in \left(e^{\min(Z_Q) + dv_Q}, e^{\min(Z_Q) + 2dv_Q}, \dots, e^{\max(Z_Q)} \right) \quad (24)$$

where $dv_Q = (\max(Y) - \min(Y) + 1)/2^K$. To prevent dealing with unreasonably large numbers (which would require more memory for representation), the shifting value $dz = 1$.

3.3.3. K-Means Scale

Quantization via k-means clustering is a common method. In this paper, the k-means scaling was implemented using Lloyd's algorithm with random initializations for K centroids, or quantization levels. Following the convergence of the clustering algorithm, the nearest neighbor method maps the values in Y to the K quantization levels. This method assumes convergence for the success of this quantization scale. Therefore, convergence to a global minimum is improved with successful clustering by proper selection of centroid initializations using the k-means++ algorithm [49], which uses a random number generator (RNG) for randomized seeding. The k-means++ algorithm is $O(\log k)$.

3.3.4. Probability Distribution Scale

To explore the impact of the data on the quantization levels, the probability distribution function (PDF) and the output from each s -layer are used to generate quantization levels. Best-fit PDFs for the data were selected using maximum likelihood estimation (MLE). The global best-fit PDF was determined by assessing the individual fits to the s -layer outputs for each class.

The inverse Gaussian and the gamma distributions were selected to generate quantization scales. At each s -layer, the data in Y are fitted to one of these distributions. Quantization levels are then determined using an RNG, until K unique levels are found. Values were generated using the algorithms presented in [50,51] for the inverse Gaussian and the gamma scales, respectively. Lastly, each $y \in Y$ is mapped to a level using nearest neighbor. Note that the computational time required to generate the levels for these scales is dependent on the number of unique values required, which may decline with increasing scattering order.

As with the log-scale, domain restrictions must be handled. The inverse Gaussian distribution has support on $[0, \infty)$, therefore shifting the data such that $\min(y \in Y) \geq 0$ aids in obtaining the fitted PDF. The support of the gamma distribution, $(0, \infty)$ requires shifting the values such that $\min(y \in Y) > 0$; the value of dz should be as small as possible. Following the calculation and application of the quantization levels, the data are shifted by $-dz$.

3.3.5. Quantile Scale

Another PDF-based quantization scale is considered that requires no RNG. The distribution of the values in Y are divided into K quantiles. The midpoint of each quantile is then used as a quantization level, v_Q . This provides a more static fit of a PDF to the data; moreover, there are no support considerations required for its implementation.

4. Quantized Wavelet Scattering Network Results

The performances of the quantization method and scales were tested using a second-order WSN with Morlet wavelets at five scales ($J = 5$) and eight orientations ($L = 8$) and the Gaussian windowing function.

The Morlet wavelet is defined as

$$\psi(\vec{u}) = (e^{j\zeta x} - K) \exp\left(-\frac{x^2 + s^2 y^2}{2\sigma_\psi^2}\right), \quad (25)$$

where s is the slant or eccentricity of the elliptical Gaussian envelope, σ_ψ is the standard deviation of the elliptical Gaussian envelope, ζ is a parameter which permits a trade-off between the x - and y -resolutions, and K is a constant to ensure that the average value of $\psi(\vec{u})$ is zero.

The Gaussian windowing function is represented as

$$\phi(\vec{u}) = \exp\left(-\frac{x^2 + y^2}{2\sigma_\phi^2}\right), \quad (26)$$

where σ_ϕ is its standard deviation.

The parameter values of the mother Morlet wavelet and the Gaussian windowing function, as defined in Equations (25) and (26), are provided in Table 2. The size of the filters at $r = 0$ was 144×144 pixels.

Table 2. Parameter values of mother Morlet wavelet and Gaussian windowing function.

Parameter	Value
σ_ϕ	0.8
σ_ψ	0.8
S	0.5
ζ	2.356

The output of the WSN is modified to form a feature vector $\vec{a} \in \mathbb{R}^{\Sigma N^{(f)}}$, such that each element is equal to $\sum_{y \in Y} y$ for a unique windowed scattering propagator. These 681 features were input to a linear kernel SVM for classification. A subset of the MSTAR dataset was used for performance analysis in MATLAB. Each quantization scale was evaluated for with 2, 4, 16, and 256 quantization levels. In addition to the input and each $S_J[p]$ and $U[p]$, each feature vector was quantized; the same quantization scale was implemented at each location. For the RNG-based quantization schemes, k-means, gamma, and inverse Gaussian, the effect of random number generation was evaluated by seeding the Mersenne Twister RNG (the default in MATLAB) with 10 different seeds. The effect of noise addition was assessed for four signal-to-noise ratios (SNRs) of 2, 10, 20, and 50 dB.

After assessing the implementations of the quantization schemes, the underlying performance of the WSN-SVM was compared to that of a linear kernel SVM and ResNet18 [23] for each SNR. The ResNet18 architecture was trained from scratch using stochastic gradient descent with momentum 0.9 for a maximum of 50 epochs with batch sizes of 256; the initial learn rate 1×10^{-4} was scaled by a factor of 0.9 every four epochs. The described training options were loosely tailored for the infinite SNR case, then used across the remaining SNR cases.

Note that compared to the WSN-SVM architecture, both the SVM and ResNet18 are more complex classifiers. Input to the SVM is the full SAR image (64^2 features), requiring more memory in comparison to the 681 features used in the WSN-SVM. ResNet18 is a 72-layer CNN with 11.7 million learnable parameters.

4.1. Description of the MSTAR Dataset and Augmentations

Eight of the available classes from the mixed target subset of the MSTAR database were used for classification, as shown in Figure 5. In existing works that apply CNNs to SAR data, the 15° and 17° depression angle data are used, typically separated into training and test data. While the difference between 15° and 17° may be negligible, only the 15° data were used in order to remove any possible confusion during analysis.

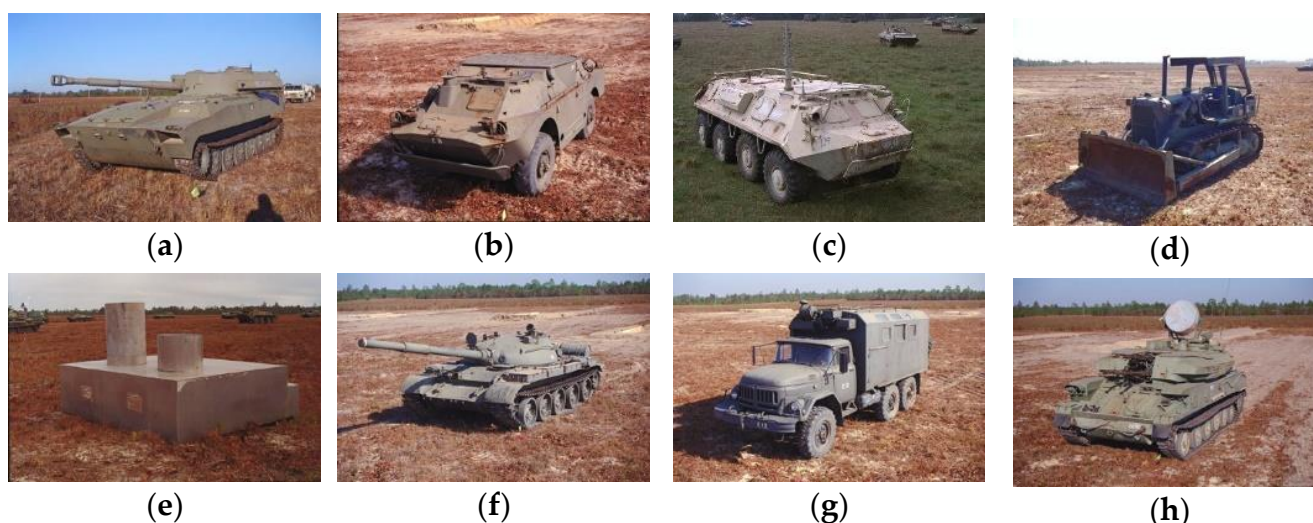


Figure 5. Classes in the MSTAR dataset: (a) 2S1; (b) BRDM-2; (c) BTR-60; (d) D7; (e) SLICY; (f) T62; (g) ZIL-131; (h) ZSU-23-4.

The number of samples from each class in the 15° subset of the MSTAR database was 274, except for the BTR-60, which was 195, totaling 2112 samples. Note that the number of samples for the BTR-60 was approximately 70% of the other classes, making this an unbalanced dataset. Each of the samples were cropped to 64×64 images, centered on the target, to reduce the effect of the target's surroundings on the classification results, while retaining the entire target. The SLICY samples were uncropped, as they were only 54×54 ; to address this discrepancy, these images were symmetrically padded to the uniform size for the WSN-SVM, SVM, and ResNet18 classifiers.

Figure 6 shows the histograms and the fitting of an inverse Gaussian PDF for two example cases within the MSTAR database using the methodology described in Section 3.3.4.

To assess the effectiveness of classification in the presence of noise, white Gaussian noise, η , was added to the dataset prior to partitioning the samples into training and test data, such that the new noisy sample is given by $\tilde{X} = |X + \eta|$. Although this method of noise addition does not reflect the actual presence of noise in SAR imagery, it provides a measure to assess the robustness of the network in classifying more complex data.

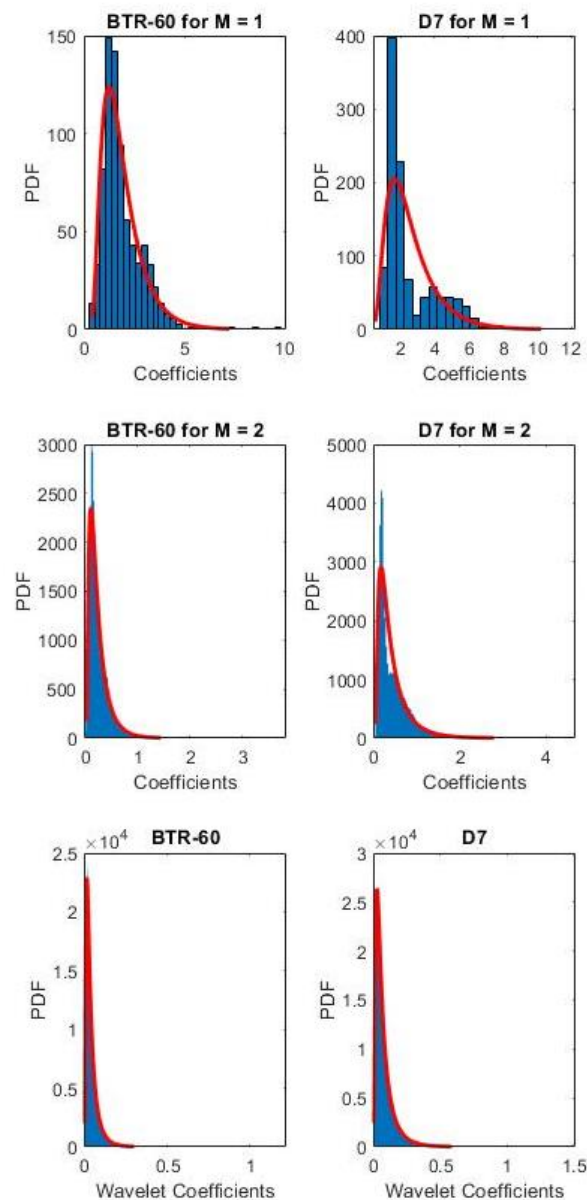


Figure 6. Histogram of MSTAR dataset wavelet coefficients separated by class for each s -layer in a WSN for $Q = 1$, $J = 5$, and $L = 8$: (**Top**) $M = 1$; (**Middle**) $M = 2$; (**Bottom**) fitted inverse Gaussian PDF.

4.2. Evaluation Metrics

For the WSN-SVM and SVM-only architectures, classification was performed using a one-vs.-all SVM with a linear kernel function. No consensus for assessing the success of a multi-class classification algorithm exists. Because the MSTAR dataset is unbalanced due to the fewer samples present for the BTR-60, the balanced accuracy was calculated rather than traditional binary calculation of accuracies. We employed the balanced accuracy, ACC, given by [52]

$$\text{ACC} = \frac{1}{2}(\overline{\text{TPR}} + \overline{\text{TNR}}), \quad (27)$$

where $\overline{\text{TPR}}$ and $\overline{\text{TNR}}$ are the true positive and negative rates, respectively. This was expanded to a multi-class classifier implementation by calculating the balanced accuracy for each class, with the true positive and negative rates calculated as macro-averages. To accommodate the small size of the dataset, 20-fold validation was used.

5. Results and Discussion

5.1. Effects of RNG Seeding

Figure 7 shows the results of the RNG-based quantization scales under each SNR condition for 10 different seeds. Regardless of the number of quantization levels, the variability in network performance was negligible in all cases. Of the three quantization scales, k-means outperformed the PDF-based methods for unique quantization levels 2, 4, and 16 for $\text{SNR} > 10$ dB; however, for $\text{SNR} \leq 10$ dB, the k-means scale only outperformed the others for two and four levels.

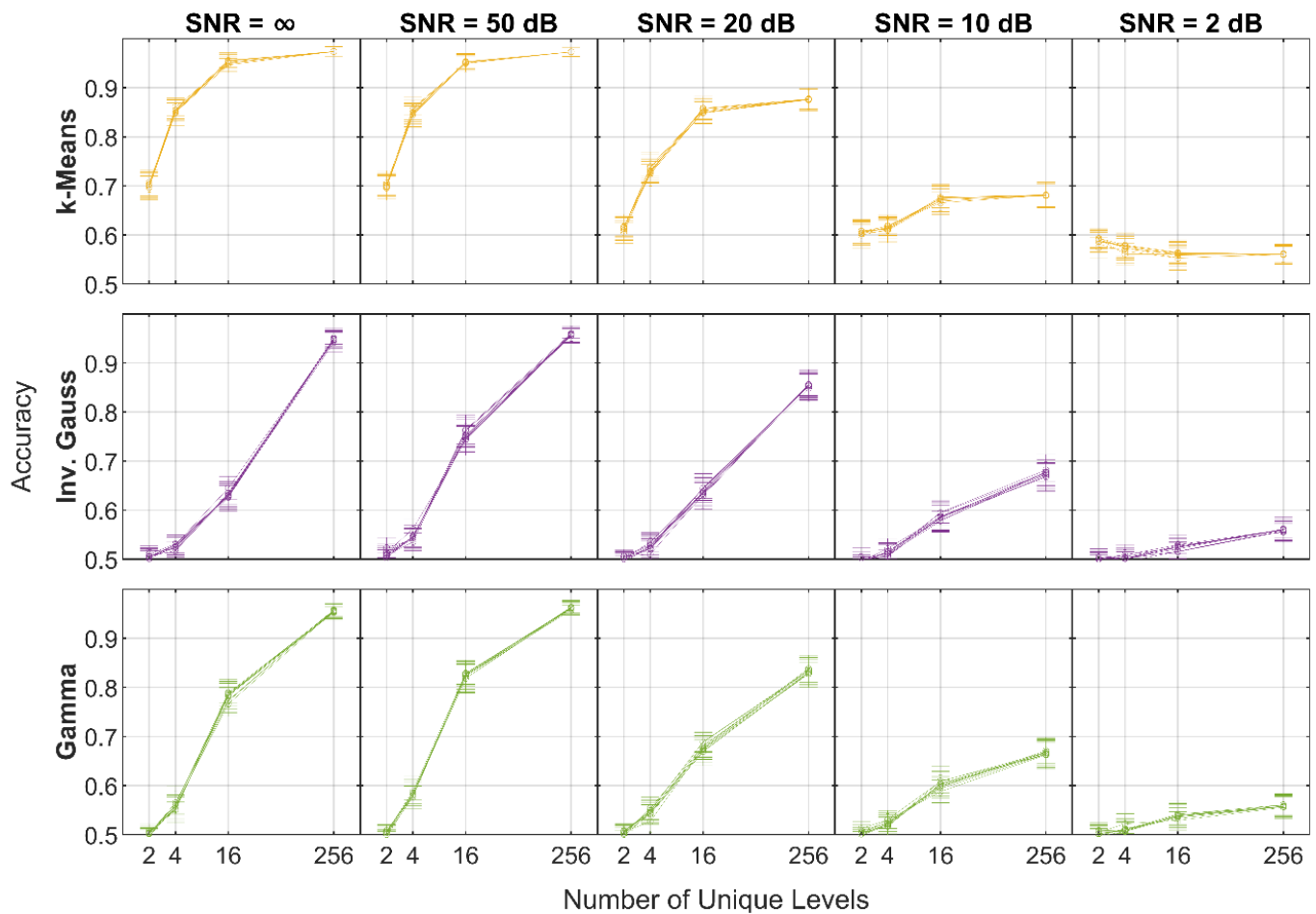


Figure 7. Accuracy of the networks implementing RNG-based quantization scales for 10 different initial seeds.

To simplify discussion in the remainder of this paper, the performance of the RNG-based methods uses the average performance and error.

5.2. Noiseless and Noisy Datasets

The effect of quantization on the noiseless dataset for all values of K assessed is shown in Figure 8. As expected, a smaller K yielded poorer accuracy. The k-means and quantile scales performed best for two levels on average, with only the quantile scale performing significantly better than the uniform, log, and PDF-based distributions. In addition, the quantile scale had the smallest standard deviation (0.0199) for two levels, excluding the gamma and inverse Gaussian scales. However, as the number of levels increases, the quantile scale falls behind the k-means and log scales until 256 levels, where performances of these three are similar to that of the non-quantized dataset.

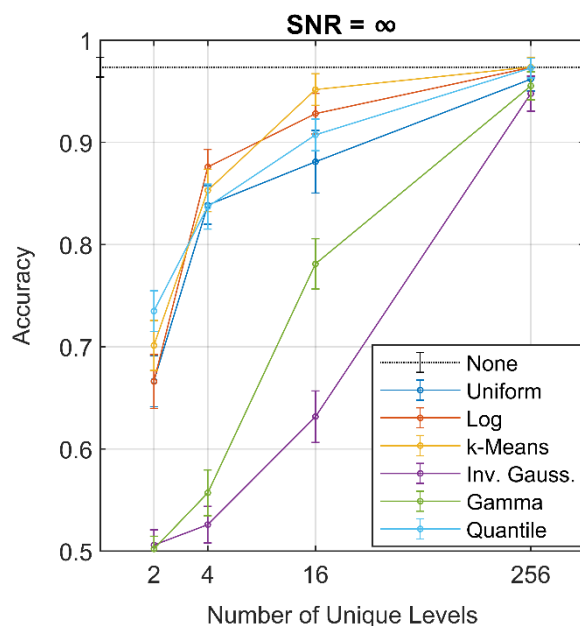


Figure 8. Accuracy of the quantizer scales for $\text{SNR} = \infty$.

The steepest decrease in accuracy occurred in the gamma and inverse Gaussian scale between $K = 256$ and $K = 16$. This was likely caused by poor choice of the quantization values at the input and the first s-layer. The input to the network contained $64^2 = 4096$ values, which were quantized to K unique values. For $K = 256$, the maximum number of unique quantization levels is 256; if all 4096 values of the input are unique and all 256 levels exist in the input, then only $256/4096 = 0.0625$ of the input is accurately represented, with the remaining 0.9375 of the input quantized. For $K = 16$, the fraction of unique quantization levels to unique values in the input reduces to 0.0039. Poor selection of initial quantization levels would, therefore, have a significant impact throughout the network. In addition, by quantizing to fewer levels, the output from the following layer likely loses the distribution shape found in the non-quantized data, such that the pre-selected PDF is no longer the best-fit within the WSN.

Due to the comparatively high accuracy of the quantile scale (nearing 0.75 even at two levels), one of the underlying causes of the poor performance of the gamma and inverse Gaussian quantization scales was likely the RNG employed to generate the quantization levels. As there is no guarantee that the output of each q-layer retains its shape, as shown in Figure 6, the estimated PDF is not necessarily a best-fit function. While an RNG is also core to the k-means scale, the initial centroids were heuristically selected to improve performance per the k-means++ algorithm, whereas the other RNG-based scales presuppose the underlying PDF.

As the SNR decreased, the quantized WSN-SVMs generally experienced the same decrease in accuracy as the non-quantized WSN-SVMs at $K = 256$, as shown in Figure 9. The results indicate that only 256 quantization levels are required for a quantized WSN-SVM to achieve similar performance to the non-quantized network, regardless of SNR, due in part to the decreasing size of the outputs of each filter: the majority of filters in the network produce outputs containing 256 or fewer values. (This can be quickly verified from the \log_2 subsampling rates chart in Figure 4b.) The number of unique values is only reduced at the scattering propagator $U[p]$ for $p = O, ((0, \theta)), ((0, \theta), (1, \theta)),$ and $((1, \theta))$ for all $\theta \in \Theta$. Therefore, the performance of each quantized WSN-SVM is likely more affected by appropriate selection at the majority of quantization locations. At $K = 256$, the average accuracy of the k-means, log, and quantile scales typically coincided with the non-quantized network, although the error increased with decreasing SNR.

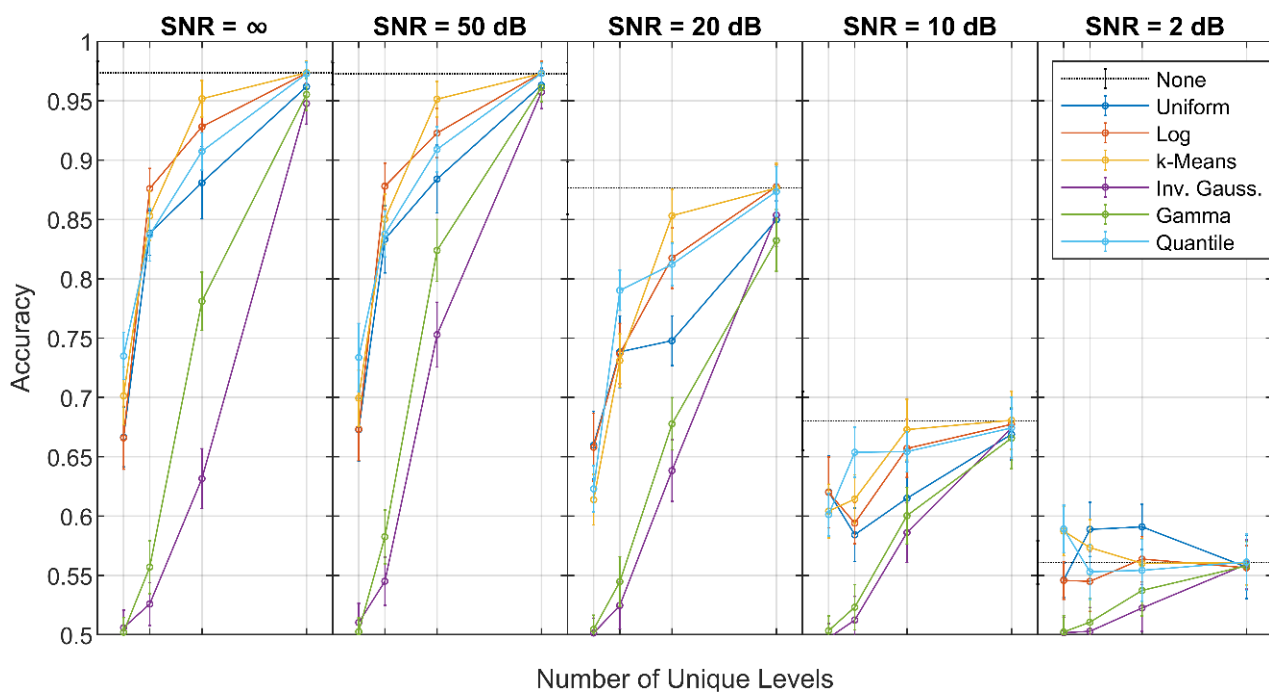


Figure 9. Accuracy of the quantizer scales for infinite, 50, 20, 10, and 2 dB SNR.

For $K < 256$, the quantile scale appeared to better retain its performance relative to that of the non-quantized WSN-SVM at each SNR; that is, the accuracy of the quantile scale at these levels decreased less with SNR than other scales, although performance typically remained comparable with the k-means and log scales. For the 2 dB SNR case, the quantile, k-means, and uniform scales outperformed the non-quantized data, which might indicate some inherent trend in the data, despite the noise addition.

5.3. Comparison with the SVM and ResNet18

The results of the non-quantized WSN-SVM, SVM, and ResNet18 for all five SNR cases is provided in Table 3. The WSN-SVM architecture performed comparably to the standard SVM, with the WSN-SVM constituting a less complex training process with only 681 features input to its SVM compared to the 4096 features of the standard SVM. Apart from outperforming traditional SVM, Table 3 shows that WSN-SVM substantially outperformed ResNet18 for the infinite, 50, and 20 dB SNR cases. Since the considered SNR regime (around 20 dB) is operationally significant in many applications, WSN-SVM can thus be a potentially powerful alternative to both CNNs and traditional ML approaches such as SVM in such scenarios. Furthermore, even though WSNs do not perform as well as CNNs in the low-SNR regime, the computational complexity to train WSNs is substantially less as well.

Table 3. Performance accuracies of non-quantized WSNs, SVMs, and ResNET18.

SNR (dB)	WSN-SVM	SVM	ResNet18
∞	0.974 ± 0.0071	0.973 ± 0.011	0.818 ± 0.0098
50	0.973 ± 0.0053	0.972 ± 0.011	0.819 ± 0.011
20	0.871 ± 0.013	0.835 ± 0.021	0.799 ± 0.016
10	0.681 ± 0.020	0.659 ± 0.024	0.774 ± 0.027
2	0.557 ± 0.015	0.571 ± 0.023	0.754 ± 0.024

The poorer performance of ResNet18 in the mid- to higher-SNR regimes could be attributed to the implemented training method and associated parameters. However, for the 10 and 2 dB SNR cases, the accuracy of ResNet18 dropped by approximately 0.04

and 0.06, respectively, from that of the infinite SNR, whereas the WSN-SVM dropped by approximately 0.29 and 0.43. The robustness of ResNet18 to decreasing SNR may be attributed to both the adaptability of the network, as well as its depth and the number of filters per layer.

Note that though it does not contain learnable parameters, the wavelet parameters of the WSN presented in Table 2 may also be adjusted to improve performance.

6. Conclusions

Due to its structural similarity with CNNs, the WSN-based quantization studies performed in this study may provide a good benchmark for future work in the quantization of CNN-based neural networks. We explored the classification accuracy of quantized WSNs with a multi-class SVM. We overcame the limited performance of RNG-based PDF quantization schemes by incorporating data-driven methods for selecting the quantization levels as indicated by the performance of k-means and quantile schemes. The deleterious effects of compounding of error resulting from poorly selected quantization levels and the limitations of RNG-based PDF quantization scaling were overcome by devising RNG-based k-means scaling and statically generated levels of the PDF-based quantile scales. The performance of various quantization methods was quantitatively studied for different levels of noise, which can provide guidance for the design of quantized CNNs under practical operating scenarios. Based on the comparison of the WSN-SVM with ResNet18, future work should investigate the application of the quantization scheme in ResNet18 and similar CNN architectures.

Author Contributions: R.M.N. and R.G.R. proposed the concept. M.R.F. performed the initial study and development of algorithms. R.G.R. helped refine the algorithms, which were implemented by M.R.F. All authors participated equally in the data analysis, interpretation, and presentation of results. R.M.N. wrote the first draft of the paper and the other authors contributed to its final form. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by an NRL 6.1 Base Program and by the U.S. Office of Naval Research, grant number N00014-16-1-2354.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Musman, S.; Kerr, D.; Bachmann, C. Automatic recognition of ISAR ship images. *IEEE Trans. Aerosp. Electron. Syst.* **1996**, *32*, 1392–1404. [[CrossRef](#)]
2. Aydogan, D.B.; Hannula, M.; Arola, T.; Dastidar, P.; Hyttinen, J. 2D texture based classification, segmentation and 3D orientation estimation of tissues using DT-CWT feature extraction methods. *Data Knowl. Eng.* **2009**, *68*, 1383–1397. [[CrossRef](#)]
3. Jawahir, W.N.; Yussof, H.W.; Burkhardt, H. Relational features for texture classification. In Proceedings of the International Conference on Signal Processing, Image Processing and Pattern Recognition (SIP), Jeju Island, Korea, 28 November–2 December 2012; pp. 438–447.
4. Srinivas, U.; Monga, V.; Raj, R.G. SAR ATR using discriminative graphical models. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *50*, 591–606. [[CrossRef](#)]
5. Amoon, M.; Rezai-Rad, G.A. Automatic target recognition of synthetic aperture radar (SAR) images based on optimal selection of Zernike moments features. *IET Comput. Vis.* **2013**, *8*, 77–85. [[CrossRef](#)]
6. McKay, J.; Monga, V.; Raj, R.G. Robust sonar ATR through Bayesian pose-corrected sparse classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 5563–5576. [[CrossRef](#)]
7. Olson, C.F.; Huttenlocher, D.P. Automatic target recognition by matching oriented edge pixels. *IEEE Trans. Image Process.* **1997**, *6*, 103–113. [[CrossRef](#)]

8. Bhatnagar, V.; Shaw, A.; Williams, R.W. Improved automatic target recognition using singular value decomposition. In Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seattle, WA, USA, 12–15 May 1998; pp. 2717–2720.
9. Suvorova, S.; Schroeder, J. Automated target recognition using the Karhunen-Loeve transform with invariance. *Digit. Signal. Process.* **2002**, *12*, 295–306. [[CrossRef](#)]
10. Jansen, R.W.; Sletten, M.A.; Ainsworth, T.L.; Raj, R.G. Multi-channel synthetic aperture radar based classification of maritime scenes. *IEEE Access* **2020**, *8*, 127440–127449. [[CrossRef](#)]
11. Hall, D.L.; Ridder, T.D.; Narayanan, R.M. Abnormal gait detection and classification using micro-Doppler radar signatures. In Proceedings of the SPIE Conference on Radar Sensor Technology XXIII, Baltimore, MD, USA, 15–17 April 2019. [[CrossRef](#)]
12. Rodenbeck, C.T.; Beun, J.; Raj, R.G.; Lipps, R.D. Vibrometry and sound reproduction of acoustic sources on moving platforms using millimeter wave pulse-Doppler radar. *IEEE Access* **2020**, *8*, 27676–27686. [[CrossRef](#)]
13. Coleman, G.B.; Andrews, H.C. Image segmentation by clustering. *Proc. IEEE* **1979**, *67*, 773–785. [[CrossRef](#)]
14. Haralick, R.M.; Shapiro, L.G. Image segmentation techniques. *CVGIP* **1985**, *29*, 100–132.
15. Tajbakhsh, N.; Shin, J.Y.; Gurudu, S.R.; Hurst, R.T.; Kendall, C.B.; Gotway, M.B.; Liang, J. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Trans. Med. Imaging* **2016**, *35*, 1299–1312. [[CrossRef](#)] [[PubMed](#)]
16. Profeta, A.; Rodriguez, A.; Clouse, H.S. Convolutional neural networks for synthetic aperture radar classification. In Proceedings of the SPIE Conference on Algorithms for Synthetic Aperture Radar Imagery XXIII, Baltimore, MD, USA, 21 April 2016. [[CrossRef](#)]
17. Soldin, R.J.; MacDonald, D.N.; Reisman, M.; Konz, L.R.; Rouse, R.; Overman, T.L. HySARNet: A Hybrid machine learning approach to Synthetic Aperture Radar automatic target recognition. In Proceedings of the SPIE Conference on Automatic Target Recognition XXIX, Baltimore, MD, USA, 15–18 April 2019. [[CrossRef](#)]
18. Shao, J.; Qu, C.; Li, J. A performance analysis of convolutional neural network models in SAR target recognition. In Proceedings of the 2017 IEEE Conference on SAR in Big Data Era: Models, Methods and Applications (BIGSAR DATA), Beijing, China, 13–14 November 2017. [[CrossRef](#)]
19. Morgan, D.A.E. Deep convolutional neural networks for ATR from SAR imagery. In Proceedings of the SPIE Conference on Algorithms for Synthetic Aperture Radar Imagery XXII, Baltimore, MD, USA, 23 April 2015. [[CrossRef](#)]
20. Cha, M.; Majumdar, A.; Kung, H.T.; Barber, J. Improving SAR automatic target recognition using simulated images under deep residual refinements. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 2606–2610.
21. Fox, M.R.; Narayanan, R.M. Application and performance of convolutional neural networks to SAR. In Proceedings of the SPIE Conference on Radar Sensor Technology XXII, Orlando, FL, USA, 16–18 April 2018. [[CrossRef](#)]
22. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015. [[CrossRef](#)]
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
24. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
25. Lin, D.D.; Talathi, S.S.; Annapureddy, V.S. Fixed point quantization of deep convolutional networks. In Proceedings of the 33rd International Conference on Machine Learning (ICML'16), New York, NY, USA, 20–22 June 2016; pp. 2849–2858.
26. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2704–2713.
27. Bruna, J.; Mallat, S. Invariant scattering convolution networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1872–1886. [[CrossRef](#)] [[PubMed](#)]
28. Mallat, S. Group invariant scattering. *Commun. Pure Appl. Math.* **2012**, *65*, 1331–1398. [[CrossRef](#)]
29. Oyallon, E.; Mallat, S.; Sifre, L. Generic deep networks with wavelet scattering. *arXiv* **2014**, arXiv:1312.5940.
30. Soro, B.; Lee, C. A wavelet scattering feature extraction approach for deep neural network based indoor fingerprinting localization. *Sensors* **2019**, *19*, 1790. [[CrossRef](#)]
31. Szu, H.H. Why adaptive wavelet transform? In Proceedings of the SPIE Conference on Visual Information Processing II, Orlando, FL, USA, 14–16 April 1993; pp. 280–292.
32. Xiao, Q.; Ge, G.; Wang, J. The neural network adaptive filter model based on wavelet transform. In Proceedings of the IEEE 2009 Ninth International Conference on Hybrid Intelligent Systems, Shenyang, China, 12–14 August 2009; pp. 529–534.
33. Xiong, H.; Zhang, T.; Moon, Y.S. A translation- and scale-invariant adaptive wavelet transform. *IEEE Trans. Image Process.* **2000**, *9*, 2100–2108.
34. Nadella, S.; Singh, A.; Omkar, S.N. Aerial scene understanding using deep wavelet scattering network and conditional random field. In Proceedings of the 4th European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 205–214.

35. Singh, A.; Kingsbury, N. Dual-tree wavelet scattering network with parametric log transformation for object classification. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 2622–2626.
36. Wu, J.; Jiang, L.; Han, X.; Senhadji, L.; Shu, H. Performance evaluation of wavelet scattering network in image texture classification in various color spaces. *arXiv* **2014**, arXiv:1407.6423.
37. Li, B.H.; Zhang, J.; Zheng, W.S. HEp-2 cells staining patterns classification via wavelet scattering network and random forest. In Proceedings of the 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), Kuala Lumpur, Malaysia, 3–6 November 2015; pp. 406–410.
38. Khan, A.A.; Dhawan, A.; Akhlaghi, N.; Majdi, J.A.; Sikdar, S. Application of wavelet scattering networks in classification of ultrasound image sequences. In Proceedings of the 2017 IEEE International Ultrasonics Symposium (IUS), Washington, DC, USA, 6–9 September 2017. [[CrossRef](#)]
39. Shi, X.; Zhou, F.; Yang, S.; Zhang, Z.; Su, T. Automatic target recognition for synthetic aperture radar images based on super-resolution generative adversarial network and deep convolutional neural network. *Remote Sens.* **2019**, *11*, 135. [[CrossRef](#)]
40. Rodriguez, R.; Dokladalova, E.; Dokladal, P. Rotation invariant CNN using scattering transform for image classification. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 654–658.
41. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–356. [[CrossRef](#)]
42. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. BRIEF: Binary robust independent elementary features. *Lect. Notes Comput. Sci.* **2010**, *6314*, 778–792.
43. Popa, C.A. Complex-valued convolutional neural networks for real-valued image classification. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 816–822.
44. Fox, M.R.; Raj, R.G.; Narayanan, R.M. Quantized wavelet scattering networks for signal classification. In Proceedings of the SPIE Conference on Radar Sensor Technology XXIII, Baltimore, MD, USA, 15–18 April 2019. [[CrossRef](#)]
45. Andén, J.; Sifre, L.; Mallat, S.; Kapoko, M.; Lostanlen, V.; Oyallon, E. Scatnet. Available online: <http://www.di.ens.fr/data/software/scatnet> (accessed on 28 April 2020).
46. Frazier, M.; Jawerth, B.; Weiss, G. *Littlewood-Paley Theory and the Study of Function Spaces*, 1st ed.; American Mathematical Society: Providence, RI, USA, 1991; pp. 42–49.
47. Myers, Y. *Wavelet and Operators: Volume 1*, 1st ed.; Cambridge University Press: Cambridge, UK, 1995; pp. 18–65.
48. Fox, M.R. Quantization and Adaptivity of Wavelet Scattering Networks for Classification Purposes. Master’s Thesis, The Pennsylvania State University, University Park, PA, USA, May 2020.
49. Arthur, D.; Vassilvitskii, S. K-means++: The advantages of careful seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA’07), New Orleans, LA, USA, 7–9 January 2007; pp. 1027–1035.
50. Michael, J.R.; Schucany, W.R.; Haas, R.W. Generating random variates using transformations with multiple roots. *Am. Stat.* **1976**, *30*, 88–90.
51. Marsaglia, G.; Tsang, W.W. A simple method for generating gamma variables. *ACM Trans. Math. Softw.* **2000**, *26*, 363–372. [[CrossRef](#)]
52. Brodersen, K.H.; Ong, C.S.; Stephan, K.E.; Buhmann, J.M. The balanced accuracy and its posterior distribution. In Proceedings of the IEEE 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 3121–3124.