

# Feature Operators for PolSAR Images

Jun Xiang  
404895

MSc. Geoinformation & Geodesy

Anupama Rajkumar  
415252

MSc. Autonomous Systems

## Abstract

*In this paper, we implement and evaluate various feature operators for polarimetric synthetic aperture radar (PolSAR) image classification. First, we compute features on polarimetric, color, texture, morphological profiles, and modern feature operators based on deep learning. Next, we classify the extracted features using the k-nearest neighbor (KNN). Finally, we evaluate and compare the classification performance of these features on the Oberpfaffenhofen E-SAR sensor (DLR, L-band) data. The experimental evaluations reveal that statistics of polarimetric parameters perform well on all the classes whereas other features perform well on Field and Forest classes but have a scope of improvement in City, Grassland, and Street classes.*

**Keywords** - Feature extraction (FE), K-Nearest Neighbors (KNN), PolSAR, Autoencoder (AE), PolSAR Image Classification.

## 1. Introduction

Polarimetric synthetic aperture radar (PolSAR) is an advanced imaging radar system which provides rich information of the Earth's surface under all-weather and day-and-night conditions, PolSAR data records the response of a physical scene to polarized electromagnetic waves, by analyzing the complex backscattering coefficients from PolSAR data, terrain and land use of the earth's surface can be identified and classified.

The satellite data volume is ever growing and remote sensing is facing big data challenges [34]. Remote sensing data is high dimensional and conventional techniques are not efficient for processing this data. In order to deal with the so-called curse of dimensionality feature extraction (FE) becomes imperative [2]. FE techniques lead to improvement in accuracy and the speed of classification. Many of the existing FE methods do not work well with data that is nonlinear like the PolSAR data. Therefore, various feature operators that work with nonlinear data needs to be employed.

In this paper we implement some popular feature operators based on polarimetric parameters, target decomposition, texture, color, morphological operations and a modern feature extractor like autoencoder (AE) and provide a quantitative comparison of classification capability among these for full PolSAR data.

## 2. Related Work

Accurate terrain classification of PolSAR images is an interesting and challenging task, the accuracy of classification is highly influenced by the feature descriptors of PolSAR data, there are many proposed PolSAR feature extractors, which can be generally divided into handcrafted features and modern features the employ more sophisticated techniques. The distinguishing factor for our work is that while most of the literature only computes handcrafted features and evaluates their performance on a specific dataset, we compute and evaluate the performance of both handcrafted and the modern features.

### 2.1. Handcrafted Features

Handcrafted features include polarimetric features which are derived directly or indirectly from backscattering coefficients and their simple transform, as well as traditional image-processing features based on color, texture and morphological profiles.

Polarimetric features have been evaluated by many studies for terrain classification and crop mapping [14, 26, 17]. The trend was to introduce traditional image-processing features to PolSAR data. In 2005 Benediktsson [1] illustrated that morphological profiles are effective to extract spatial-contextual features under different scales and shapes from hyper-spectral data. In 2014 Uhlmann [29] extracted color features based on pseudo color image and showed that color features provided noteworthy improvement on terrain classification in addition to polarimetric and texture features. Uhlmann and Jingliang also [29, 12] showed that texture features generally performed poorly than polarimetric, color, and morphological features but texture features still provided valuable discrimination on some classes like wa-

ter, forest, and buildings whose pixels are distributed in a concentrated manner [22].

## 2.2. Modern Feature Extractors

With the increase in the volume of data, the field of remote sensing is facing the challenge where the algorithms developed should be fast enough to process the size of data. Deep learning is a rapidly growing trend and has been applied to numerous fields. In the last few years, it has become popular in remote sensing with a large number of papers being published on this topic. Neural networks (NNs) - shallow and deep can exploit feature representation learned from data unlike handcrafted features [34].

There is a vast literature on the use of modern and more sophisticated architectures for extracting features from remote sensing data. These feature extractors range from unsupervised NNs like Autoencoder [28] of various depths to shallow supervised NNs like Random Forests (RF) [13] and deep supervised NNs like Deep Belief Networks (DBN), Convolutional NNs (CNNs) [20]. Apart from training new networks [11, 34] discuss about using CNNs that are pre-trained on natural image dataset like GoogleLeNet [27] in order to achieve higher terrain classification performance.

The data chosen in this work is not as big as some datasets like SEN12MS [24]. It is due to this reason that we proceeded with a simpler network like AE for feature extraction instead of DBN or CNN.

Different variants of autoencoders like stacked autoencoder (SAE), denoising autoencoder (DAE), and deep convolutional autoencoder (DCAE) have been previously used for feature extraction [3, 10, 8]. In our work, we use vanilla autoencoder as explained in 8 to extract features from the PolSAR data. Concurrent with our method, the encoded information which is a representation of the input data is classified using classification algorithms like KNN.

## 3. Feature Extraction

### 3.1. Polarimetric

SAR polarimetry can provide rich information of the shape, orientation, and dielectric property of backscatters to distinguish objects of interest. The scattering properties are described by  $2 \times 2$  Sinclair scattering matrix  $S$ :

$$S = \begin{bmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{bmatrix} \quad (1)$$

where the first and second subscripts represent the transmitted and received signals, respectively (H: horizontal; V: vertical)

In monostatic back scattering case,  $S_{HV} = S_{VH}$ , and scattering matrix  $S$  can be vectorized to the lexicographic scattering vector  $k_L$  and Pauli scattering vector  $k_p$ . From

these two vectorizations we can generate covariance matrix  $C$  and coherency matrix  $T$  respectively:

$$k_L = \begin{bmatrix} S_{HH} \\ \sqrt{2}S_{HV} \\ S_{VV} \end{bmatrix}, [C] = \langle k_L k_L^{*T} \rangle \quad (2)$$

$$k_p = \frac{1}{\sqrt{2}} \begin{bmatrix} S_{HH} + S_{VV} \\ S_{HH} - S_{VV} \\ 2S_{HV} \end{bmatrix}, [T] = \langle k_p k_p^{*T} \rangle$$

where  $k_L^{*T}$  and  $k_p^{*T}$  represent the matrix transpose of the complex conjugate of  $k_L$  and  $k_p$  respectively, and  $\langle \dots \rangle$  indicates sub window averaging (default window size 3).

The polarimetric features can be divided into two categories:

(1) Based on polarimetric parameters which mainly comes from Sinclair scattering matrix  $S$  and its simple transform such as covariance matrix  $C$  and coherency matrix  $T$ .

(2) Based on different target decomposition theorems.

Target decomposition provides significant information about the geophysical and geometrical structure of the targets by decomposing the averaged scattering to independent components. All the decomposition can be divided into coherent (based on scattering matrix  $S$ ) and non-coherent (based on covariance matrix  $C$  or coherent matrix  $T$ ), in this work we will focus on the following four target decomposition theorems.

(1) *Pauli Decomposition*: This decomposition expresses the scattering matrix  $S$  in so-called Pauli basis, which indicates elementary scattering mechanisms of the target.

$$S = \alpha \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \beta \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} + \gamma \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (3)$$

where  $\alpha$ ,  $\beta$  and  $\gamma$  are complex numbers and are given by:

$$\alpha = \frac{(S_{HH} + S_{VV})}{\sqrt{2}}, \beta = \frac{(S_{HH} - S_{VV})}{\sqrt{2}}, \gamma = \sqrt{2}S_{HV} \quad (4)$$

(2) *Krogager Decomposition* [16]: In the Krogager Decomposition, a symmetric scattering  $S$  matrix is decomposed into three coherent components which have physical interpretation in terms of sphere  $K_S$ , diplane  $K_D$ , and helix  $K_H$  targets, after converting scattering matrix  $S$  to the right-left (R, L) circular basis, these three parameters are presented in Figure 1.

$$S_{RR} = \frac{(S_{HH} - S_{VV} + i2S_{HV})}{2}$$

$$S_{LL} = \frac{(S_{VV} - S_{HH} + i2S_{HV})}{2} \quad (5)$$

$$S_{RL} = S_{LR} = \frac{i(S_{HH} + S_{VV})}{2}$$

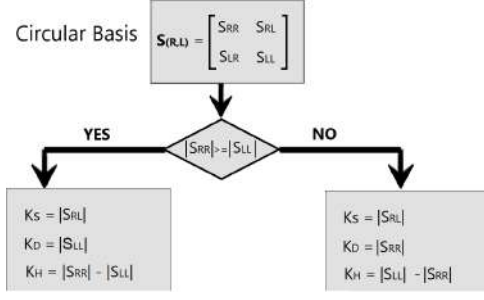


Figure 1: Krogager Decomposition

(3) *Freeman-Durden Decomposition* [7]: The Freeman-Durden decomposition models the covariance matrix  $C$  in terms of the sum of the volume  $P_v$ , double-bounce  $P_d$  and surface scattering components  $P_s$  as showed in Figure 2.

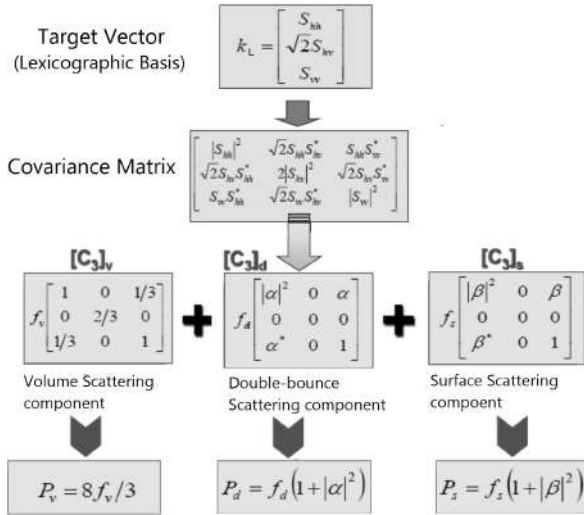


Figure 2: Freeman-Durden Decomposition

(4) *Cloude and Pottier Decomposition* [4]: The Cloude and Pottier decomposition extracts averaged parameters from coherency matrix  $T$  based on eigenvector-eigenvalue decomposition.  $H$  stands for the entropy which indicates the degree of statistical disorder of the scattering phenomenon,  $A$  stands for the anisotropy which presents the relative importance of secondary mechanisms, and  $\alpha$  indicates types of scattering mechanisms as showed in Figure 3.

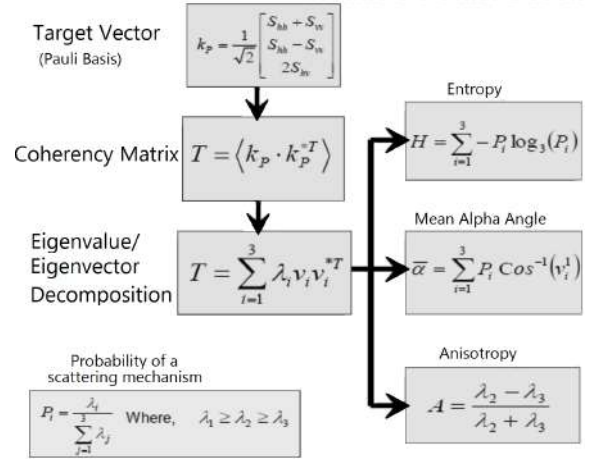


Figure 3: Cloude and Pottier Decomposition

It has been shown that the statistic vectors derived from polarimetric parameters outperformed the original polarimetric parameters in global terrain classification [12], we compute maximum, minimum, mean, standard deviation, and median as the local statistic for each patch from all the ten polarimetric parameters [30] in Table 1, we name them as PolStat-Feature.

On the whole, the investigated polarimetric features include:

(1) *PolStat-Feature*: maximum, minimum, mean, standard deviation, and median of polarimetric parameters in Table 1, total length 50.

(2) *TD-Feature*: target decomposition components as listed in Table 2 with total length 12.

(3) *CT-Feature*: amplitude of upper triangle matrix elements of covariance matrix  $C$  and coherency matrix  $T$  as showed in following function, total length 12.

$$\begin{bmatrix} C_{00} & C_{01} & C_{02} & |C_{11}| & |C_{12}| & |C_{22}| \\ T_{00} & T_{01} & T_{02} & |T_{11}| & |T_{12}| & |T_{22}| \end{bmatrix} \quad (6)$$

### 3.2. Texture

So far the most commonly employed texture features in SAR classification are GLCM, local binary pattern (LBP), edge histogram descriptor (EHD) and Gabor wavelets [25, 5, 6, 15]. In this work we select the GLCM [9] and LBP [23] as the texture descriptors, apply them to the intensity of HH, VV and HV channels.

GLCM uses co-concurrence matrix to describe the distribution of pixels in a given image area, our GLCM features include energy, contrast, homogeneity and entropy extracted from co-concurrence matrix. To compute the GLCM, the intensity of each channel in each patch is reduced to 16 bins, the orientations  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  and  $135^\circ$

Table 1: Polarimetric parameters considered in this work

Polarimetric parameters [30]	Expression
Intensity of HH channel(dB)	$10 \cdot \log  S_{HH} ^2$
Intensity of HV channel(dB)	$10 \cdot \log  S_{VV} ^2$
Intensity of VV channel(dB)	$10 \cdot \log  S_{HV} ^2$
Phase difference HH-VV	$\text{actan}(\langle S_{HH} S_{VV}^* \rangle)$
Co-polarize ratio(dB)	$10 \cdot \log \left( \frac{ S_{VV} ^2}{ S_{HH} ^2} \right)$
Cross-polarized ratio(dB)	$10 \cdot \log \left( \frac{ S_{HV} ^2}{ S_{HH} ^2} \right)$
HV/VV ratio(dB)	$10 \cdot \log \left( \frac{ S_{HV} ^2}{ S_{VV} ^2} \right)$
Copolarization ratio	$\frac{\langle S_{VV} S_{VV}^* \rangle}{\langle S_{HH} S_{HH}^* \rangle}$
Depolarization ratio	$\frac{\langle S_{HV} S_{HV}^* \rangle}{\langle S_{HH} S_{HH}^* \rangle + \langle S_{VV} S_{VV}^* \rangle}$
Amplitude of HH-VV correlation	$\left  \frac{\langle S_{HH} S_{VV}^* \rangle}{\sqrt{ S_{HH} ^2  S_{VV} ^2}} \right $

Table 2: Polarimetric target decomposition considered in this work

Decomposition	Feature details
Pauli	$ \alpha ,  \beta ,  \gamma $
Krogager [16]	sphere, diplane, helix
Freeman-Durden [7]	surface, double-bounce, volume
Cloude and Pottier [4]	entropy, anisotropy, alpha angle

were chosen. A window size of 7 pixels was chosen.

LBP labels the central pixel by thresholding its neighborhood pixels and results a binary number, to make it more rotation invariant, the rotated LBP is generated by circularly shifting the binary pattern until its minimum decimal is obtained as showed in Figure 4. We compute the LBP features at each pixel with radius 1, neighborhood pixels 8.

GLCM energy, GLCM contrast, GLCM homogeneity, GLCM entropy, and LBP are computed on each patch of pixel size  $10 \times 10$ . Then all four GLCM features are normalized to four 8-bin histograms, also LBP features are normalized to 32-bin histogram, by concatenating the resulted histograms together, we get a vector named Texture-Feature, length 64 for one channel.

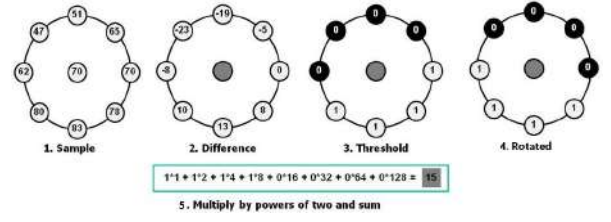


Figure 4: An example of LBP computation

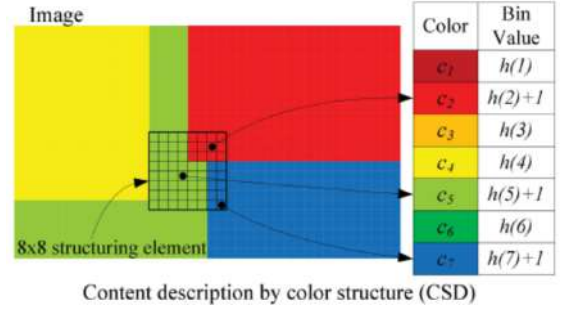


Figure 5: Accumulation of the color structure histogram

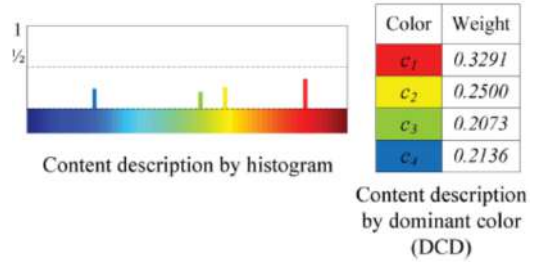


Figure 6: Accumulation of the color histogram and DCD

### 3.3. Color

In this work, the color features are extracted from pseudo color images presented in Pauli color coding ( $|HH + VV|$  as red,  $|HV|$  as green and  $|HH - VV|$  as blue), and we use MPEG-7 dominant color descriptor (DCD) [21] and MPEG-7 color structure descriptor (CSD) [21] to extract color features.

The MPEG-7 DCD can represent the color distributions of the underlying image region. The colors of an image in given area are dynamically clustered, and only predominant colors are preserved. Figure 6 [29] shows the content description of MPEG-7 DCD, four dominant colors and their weights are calculated from the color histogram.

The MPEG-7 CSD reveals the relationships among colors. To extract CSD, the image area is first transformed and quantized into the hue-min-max-difference color space

[21] to reduce the computation complexity, then a sliding  $8 \times 8$  structure element is applied, the resulting CSD vector is a histogram where each bin is the number of occurrence of certain pixel contained in the sliding window, as shown in Figure 5 [29], there are three colors (c2, c3 and c7) in the current sliding window, then the color histogram bins which corresponding to these three colors are updated accordingly.

On the whole, the investigated color feature is composed of 32 bin CSD histogram, and the weight and color values of 1 most dominant color from DCD, and it is named as Color-Feature and has a total length of 36.

### 3.4. Morphological Profiles

Morphological Profiles are effective in extracting spatial-contextual information from SAR data, the champions of IEEE GRSS Data Fusion Contest 2017 and 2018 [33, 32] chose morphological profiles of hyper-spectral data as one of the most important handcrafted features.

The morphological operators applied in this work are opening, opening by reconstruction, closing and closing by reconstruction. We set 4 morphological operators on intensity of HH, VH and VV channels with circular structuring elements of radius 1, 3 and 5, and apply them to each patch of pixel size  $10 \times 10$ , the resulted morphological feature at each patch of one channel is a stacked vector of 12 morphological profiles, we name it as MP-Feature.

Opening and closing are two commonly used morphological operators. Opening dilates an eroded image in order to recover the eroded parts as much as possible while closing erodes a dilated image in order to recover the dilated parts as much as possible. An advanced version of opening and closing are called opening by reconstruction and closing by reconstruction. They can further improve the ability of traditional opening and closing operations by completely keep or remove the objects in the image which fit or unfit the structuring element. To compute opening by reconstruction, first, erode the image with the structuring element, then set the output image as the marker and the original image as the mask, then dilate the objects in the marker image until they won't expand beyond the mask image. Figure 7 shows one part of gray-scaled intensity image of HH channel of Oberpfaffenhofen along with its corresponding opening, opening by reconstruction, closing and closing by reconstruction.

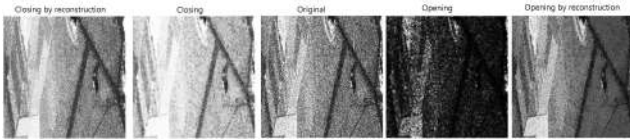


Figure 7: Morphological operations on intensity of HH channel with the structuring element radius size 3

### 3.5. Autoencoder

An autoencoder is a neural network (NN) that consists of an input layer, hidden layer(s), and an output layer. A vanilla AE as shown in figure 8 consists of a single hidden layer which is also referred to as an encoding layer whereas the output layer is referred to as a decoding layer. An AE constrains the output values to be equal to the input values [3]. The reconstruction error between the input and output is used to adjust the weight and bias vectors for each layer. In this way, the hidden layer learns features by encoding the input data. Although an AE is an unsupervised neural network, it does not require label information. The encoding of AE is obtained using the following formula:

$$y = f_e(Wx + b1) \quad (7)$$

The reconstruction of the input signal is obtained using:

$$z = f_d(W'y + b2) \quad (8)$$

The reconstruction error or cost function is calculated as:

$$E(W, b) = \frac{1}{2} |x^i - z^i|^2 \quad (9)$$

where:

$x$  : input data

$y$  : output of hidden layer or encoded values

$z$  : output of AE

$b1$  : input bias

$b2$  : hidden layer bias

$W$  : input-to-hidden layer weights

$W' : W^T$

$f_e()$  : activation function of encoder

$f_d()$  : activation function of decoder

For our implementation, we chose Sigmoid function as  $f_e()$  which is formulated as:

$$f_e(x) = \frac{1}{1 + e^{-x}} \quad (10)$$

The decoding function  $f_d()$  was chosen as a linear function.

#### 3.5.1 Autoencoder Architecture

Figure 8 depicts the AE architecture used for feature extraction. The AE model has an input and output dimensions of nine and the encoder or hidden layer dimension of five. For each pixel,  $i$  of the PolSAR image, coherency matrix as shown in (2) is used to define the input feature vector of nine dimensions [10] which is represented as:

$$x = \begin{bmatrix} T_{00} & T_{11} & T_{22} & re(T_{01}) & im(T_{01}) \\ re(T_{02}) & im(T_{02}) & re(T_{12}) & im(T_{12}) \end{bmatrix} \quad (11)$$



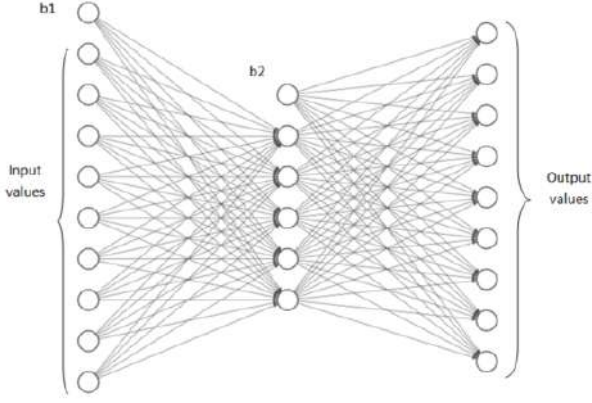


Figure 8: Proposed vanilla AE architecture. The input of size nine is the feature vector obtained from (11) for each pixel. The input value is encoded in the hidden layer of size five. This vector is used for feature extraction. The output of an AE has same dimension as that of the input vector.

The procedure for training the AE is summarized as follows:

- (1) *Step 1*: Initialise the trainable hyperparameters  $W$  and  $b$  with random values.
- (2) *Step 2*: Minimise the cost function ,  $E(W, b)$  as a function of  $W$  and  $b$ .
- (3) *Step 3*: Apply stochastic gradient descent algorithm (SGD) [18] to optimise  $W$  and  $b$ .
- (4) *Step 4*: The encoded value in the hidden layer is used as feature representation of the input layer for feature extraction.

## 4. Experimental Results and Analysis

Figure 9 represents the flowchart of our workflow. The features extracted by various feature operators described earlier are classified. K- nearest neighbor (KNN) classifier is used for this. Finally, we determine the performance by calculating the overall accuracy (OA) of classification.

### 4.1. Data

For our experiments we use a fully polarimetric image of 6640 x 1390 pixels acquired over Oberpfaffenhofen, Wessling in Germany by E-SAR sensor (DLR, L-band). The resolution of the image is 1.5 m per pixel [13]. The ground truth is annotated with five classes: City (red), Field (yellow), Forest (dark green), Grassland (light green), Streets (blue). The false color image and the ground truth map are shown in Figure 10a and 10b respectively.

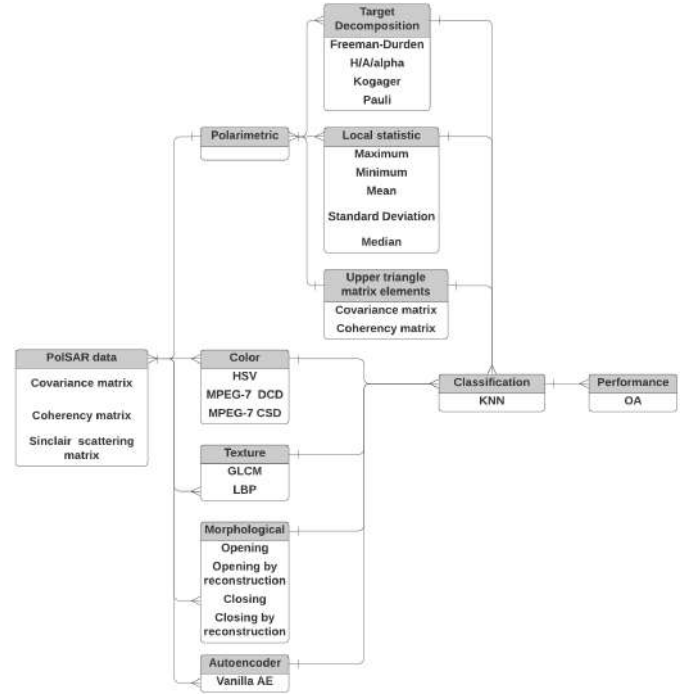


Figure 9: Representation of the implemented workflow. The PolSAR data is used as input for different feature operators. The feature representation of the input obtained is classified using KNN classifier and OA is used as the performance measure.



(a) False color image



(b) Ground truth

Figure 10: Oberpfaffenhofen E-SAR sensor (DLR, L-band) (a) False color image; (b) Ground truth : City (red), Field (yellow), Forest (dark green), Grassland (light green), Streets (blue), Black (unlabelled pixels)

### 4.2. Experiments

The experiments are carried out in the following way:

- (1) *Step 1*: We tried to despeckle the image using refined Lee filter [19]. However for most of the hand-crafted features especially PolStat-Feature, Texture-Feature

Table 3: Data size (pixels) of each class

Class name	train size	test size	total
City	603296	150824	754120
Field	2586318	646580	3232898
Forest	1226954	306738	1533692
Glassland	323515	80879	404394
Street	250348	62587	323515

Table 4: Class-wise OA of Oberpfaffenhofen dataset for different feature operators

Feature name	City	Field	Forest	Grass	Streets	All
PolStat-Feature	0.87	0.99	0.98	0.84	0.85	0.96
TD-Feature	0.44	0.99	0.92	0.60	0.15	0.84
CT-Feature	0.28	0.97	0.88	0.40	0.52	0.80
Texture-Feature	0.68	0.99	0.93	0.57	0.49	0.88
Color-Feature	0.61	0.97	0.94	0.35	0.39	0.85
MP-Feature	0.42	0.96	0.89	0.63	0.66	0.84
Autoencoder	0.37	0.87	0.58	0.31	0.41	0.70

and Color-Feature, the final overall accuracies were poor than those without despeckling, and [12] also showed that despeckling filter can lead to a great loss of texture information. Thus in this paper all the features are computed for the entire image without despeckling.

(2) *Step 2*: For classification, the computed features are divided into balanced training and testing data sets with a ratio of 4: 1 for each class as shown in Table 3.

(3) *Step 3*: All the features are classified by KNN. For handcrafted features,  $k$  was set to 20 whereas it was set to 10 for AE.

(4) *Setp 4*: 5 folds of cross-validation are performed to compute the overall accuracy (OA) for each class. The average of the cross-validation accuracies computed across all the folds provides an estimate of the performance of feature extraction. The results are presented in the following section 4.3.

The trainable hyperparameters of AE, namely the hidden layer dimension, the learning rate  $\eta$ , the momentum  $m$  were set to 5, 0.1, and 0.9 respectively. The model was trained for 100 epochs.

### 4.3. Results

(1) *Feature maps*: The spatial representation of calculated features are plotted and shown in figure 12.

(2) *Classification results*: The classification results of calculated features are plotted and shown in figure 13.

(3) *Classification Performance*: The class-wise and complete image OA scores of calculated features are shown in table 4

### 4.4. Analysis

From class-wise OA performance as shown in Table 4 and classification results as shown in Fig. 13, we can see that PolStat-Feature performs best class-wisely, while other handcrafted features only work well on Field (above 97%) and Forest (above 88%), and work poorly on City, Grass, and Streets (less than 70%), especially TD-Feature only gets 15% on Street and CT-Feature only gets 28% on City, this can be explained by Fig. 11 which is a plot of feature values per class. It can be seen that the target decomposition (TD-Feature) values of Street (blue) are fully mixed with that of Field (yellow), and the coherency and covariance element (CT-Feature) values of City (red) is fully mixed with that of Forest(dark green). Because back-scattering mechanisms between streets and fields and between city and forest are quite similar, Streets and fields have more surface scattering which can't be distinguished by target decomposition while city and forest have more volume scattering which can't be differentiated by simple coherency and covariance matrix elements.

From Table 4, we can also find that morphological features (MP-Feature) perform better in Grassland (63%) and Streets (66%) than most other features. However, MP-Feature does not work well on City (42%). The reason could be that the shape and size of structuring elements used in this paper are not right for City. In this paper, we use circular SE with radius 1, 3, and 5 which corresponds to circular ground objects with radius around 1.5 m, 4.5 m, and 7.5 m. For land class City, the shape of structuring elements (SE) should be rectangle and the size should be bigger. By adjusting the shape and size of SE, the accuracy of MP-Feature for City can be further improved. As for Texture-Feature and Color-Feature, neither could perform well on Grassland and Streets (less than 60%), and neither could separate Grassland (cyan) from Forest (dark green) and Street (blue) from Field (yellow) efficiently. The reason could be that these targets have quite similar texture or color values.

Vanilla AE implemented as a modern feature extractor achieves an accuracy of 70%. This is lower than the accuracy achieved in the work presented in [3], [10] and [8]. This is because all these work employ multi-layer autoencoders and it has been demonstrated in [3] that multi-layer AEs achieve better accuracy as compared to single layer network like the one used in this work. It was observed in [8] that first layer accuracy of pixel feature is 73.21% which close to the result achieved in the experiments performed in this work. Looking at this, we can say that we can expect to see an improvement in OA with the use of multi-layer network. The class-wise performance trend of AE is similar to handcrafted features with better OA for field and forest points as compared to city, grassland and streets. The possible explanation for

this could be that since input feature vector is derived from coherence matrix, the classification behavior is similar to that observed by CT-feature as explained earlier.

## 5. Conclusions

This paper discusses and evaluates several handcrafted and modern feature operators that can be used to extract the non-linear relationship between elements from a feature vector generated from PolSAR data. The extracted features are classified using the KNN algorithm and comparison of class-wise OA as well as that of the entire dataset has been presented.

We can see that the features extracted by the statistics of polarimetric parameters are capable of achieving higher accuracies in all five land-use types, and morphological features can also work well on a certain type of land type when the proper shape and size of structuring elements are applied for morphological extractors, other features like target decomposition, covariance and coherency elements, texture and color can't work well on all the land types in our data set, but they still provide good classification accuracy on certain land types (Field and Forest). In contrast, single-layer features like vanilla AE do not produce better classification results, and a multi-layer strategy is desired. It should be noted that increased network complexity leads to an increase in training and prediction time.

The future work includes evaluating the combination of handcrafted features and improving the modern feature extractor by selecting better hyperparameters, making the network more complex than the one used currently, and implementing more feature operators. It would be interesting to observe the performance when different input feature vectors like texture [8] is used instead of covariance matrix  $C$  or coherence matrix  $T$ .

In addition to this, future extension of our work would involve applying the implemented feature operators to bigger datasets with various land types, for instance, SEN12MS [24] and using different classification methods like RF and CNN.

## References

- [1] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson. Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):480–491, 2005. 1
- [2] Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10):6232–6251, 2016. 1

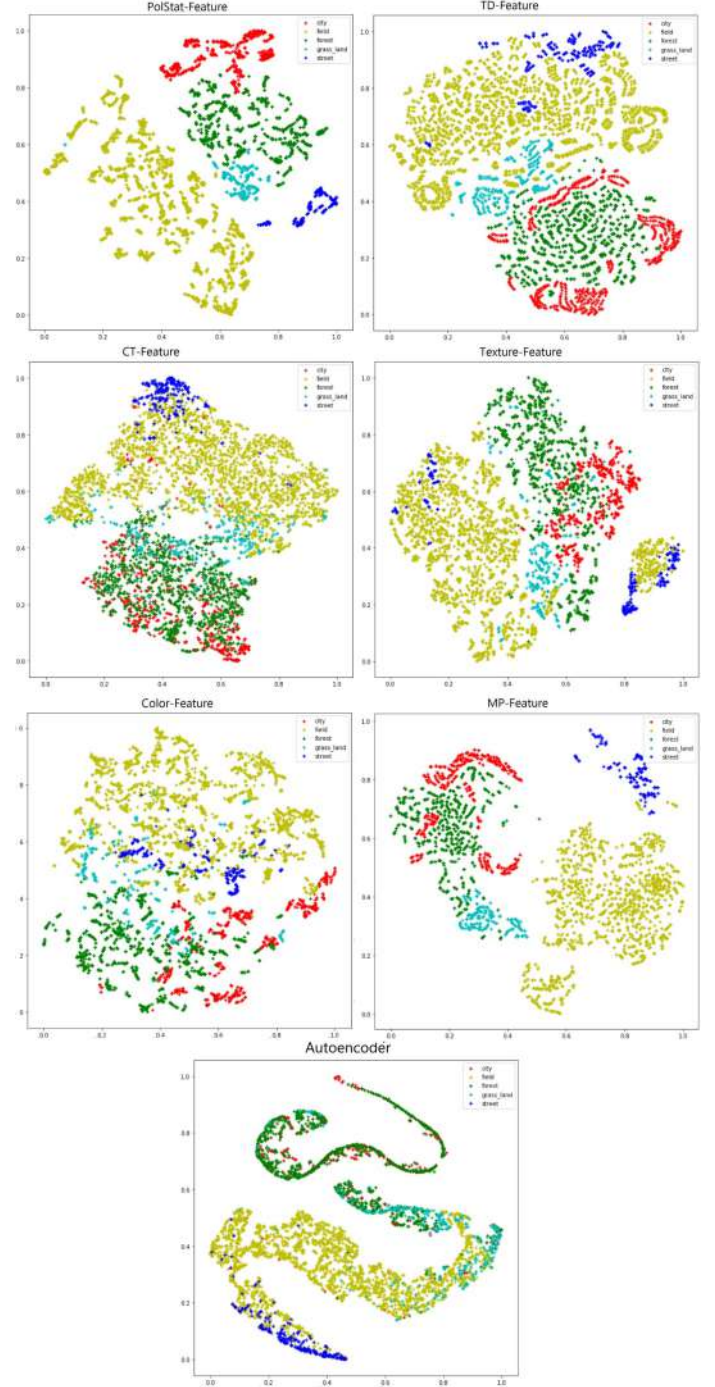
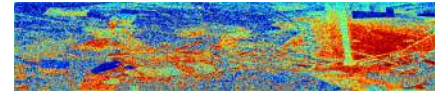


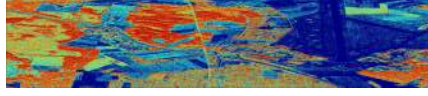
Figure 11: Features plotted by class (dimension reduced by t-nse [31])

- [3] Xing Chen, Li Ma, and Xiaoquan Yang. Stacked denoise autoencoder based feature extraction and classification for hyperspectral images. *J. Sensors*, 2016:3632943:1–3632943:10, 2016. 2, 5, 7

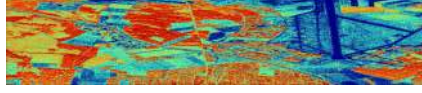




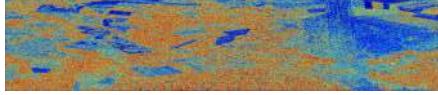
(a) Co-polarize ratio



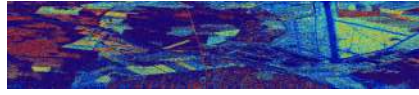
(b) C01



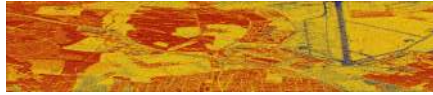
(c) T11



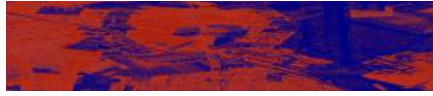
(d) cloude alpha



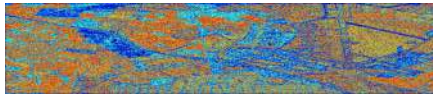
(e) freeman surface



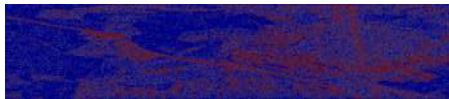
(f) pauli alpha



(g) krogager helix



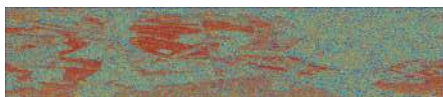
(h) DCD value 1



(i) CSD bin 16



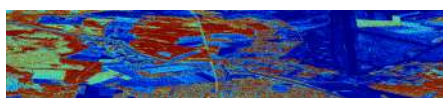
(j) GLCM Contrast of HH channel bin 1



(k) LBP bin 2



(l) opening by reconstruction



(m) Autoencoder feature 1



(a) PolStat-Feature



(b) TD-Feature



(c) CT-Feature



(d) Texture-Feature



(e) Color-Feature



(f) MP-Feature



(g) Autoencoder

Figure 13: Classification results for Oberpfaffenhofen dataset, (a) PolStat-Feature, (b) TD-Feature, (c) CT-Feature, (d) Texture-Feature, (e) Color-Feature, (f) MP-Feature, (g) Autoencoder

[4] S. R. Cloude and E. Pottier. An entropy based classification scheme for land applications of polarimetric sar. *IEEE Transactions on Geoscience and Remote Sensing*, 35(1):68–78, 1997. 3, 4

[5] F. Dell’Acqua and P. Gamba. Texture-based characterization

Figure 12: Spatial representation of calculated features

- of urban environments on satellite sar images. *IEEE Transactions on Geoscience and Remote Sensing*, 41(1):153–159, 2003. 3
- [6] K. Ersahin, B. Scheuchl, and I. Cumming. Incorporating texture information into polarimetric radar classification using neural networks. In *IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium*, volume 1, page 563, 2004. 3
- [7] A. Freeman and S. L. Durden. A three-component scattering model for polarimetric sar data. *IEEE Transactions on Geoscience and Remote Sensing*, 36(3):963–973, 1998. 3, 4
- [8] J. Geng, J. Fan, H. Wang, X. Ma, B. Li, and F. Chen. High-resolution sar image classification via deep convolutional autoencoders. *IEEE Geoscience and Remote Sensing Letters*, 12(11):2351–2355, 2015. 2, 7, 8
- [9] Robert Haralick, K. Shanmugam, and Ih Dinstein. Textural features for image classification. *IEEE Trans Syst Man Cybern*, SMC-3:610–621, 01 1973. 3
- [10] B. Hou, H. Kou, and L. Jiao. Classification of polarimetric sar images using multilayer autoencoders and superpixels. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(7):3072–3081, 2016. 2, 5, 7
- [11] Fan Hu, Gui-Song Xia, Jingwen Hu, and Liangpei Zhang. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sensing*, 7(11):14680–14707, Nov 2015. 2
- [12] Jingliang Hu, Pedram Ghamisi, and Xiao Zhu. Feature extraction and selection of sentinel-1 dual-pol data for global-scale local climate zone classification. *ISPRS International Journal of Geo-Information*, 7:379, 09 2018. 1, 3, 7
- [13] Ronny Hänsch and Olaf Hellwich. Classification of polarimetric sar images by stacked random forests. *ISPRS International Journal of Geo-Information*, 7(2):74, 2018. Available Open Access publishedVersion at <https://depositonce.tu-berlin.de/handle/11303/9981>. 2, 6
- [14] Xianfeng Jiao, John M. Kovacs, Jiali Shang, Heather McNairn, Dan Walters, Baoluo Ma, and Xiaoyuan Geng. Object-oriented crop mapping and monitoring using multi-temporal polarimetric radarsat-2 data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 96:38 – 46, 2014. 1
- [15] U. Kandaswamy, D. A. Adjeroh, and M. C. Lee. Efficient texture analysis of sar imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 43(9):2075–2083, 2005. 3
- [16] E. Krogager. New decomposition of the radar target scattering matrix. *Electronics Letters*, 26(18):1525–1527, 1990. 2, 4
- [17] J. Larrañaga, A.; Álvarez-Mozos. On the added value of quad-pol data in a multi-temporal crop classification framework based on radarsat-2 imagery. *Remote Sense*, 8:335, 2016. 1
- [18] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998. 6
- [19] Jong-Sen Lee. Refined filtering of image noise using local statistics. 1981. 6
- [20] X. Liu, L. Jiao, X. Tang, Q. Sun, and D. Zhang. Polarimetric convolutional network for polsar image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(5):3040–3054, 2019. 2
- [21] B. Manjunath, J.R. Ohm, Vinod Vasudevan, and Akio Yamada. Color and texture descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11:703 – 715, 07 2001. 4, 5
- [22] A. Masjedi, M. J. Valadan Zoej, and Y. Maghsoudi. Classification of polarimetric sar images based on modeling contextual information and using texture features. *IEEE Transactions on Geoscience and Remote Sensing*, 54(2):932–943, 2016. 2
- [23] Matti Pietikäinen, Timo Ojala, and Zelin Xu. Rotation-invariant texture classification using feature distributions. *Pattern Recognit.*, 33:43–52, 2000. 3
- [24] Michael Schmitt, Lloyd Hughes, Chunping Qiu, and Xiao Zhu. Sen12ms – a curated dataset of georeferenced multi-spectral sentinel-1/2 imagery for deep learning and data fusion. 06 2019. 2, 8
- [25] K. S. Shanmugan, V. Narayanan, V. S. Frost, J. A. Stiles, and J. C. Holtzman. Textural features for radar image analysis. *IEEE Transactions on Geoscience and Remote Sensing*, GE-19(3):153–156, 1981. 3
- [26] H. Skriver. Crop classification by multitemporal c- and l-band single- and dual-polarization and fully polarimetric sar. *IEEE Transactions on Geoscience and Remote Sensing*, 50(6):2138–2149, 2012. 1
- [27] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. 2
- [28] R. Tanase, M. Datcu, and D. Raducanu. A convolutional deep belief network for polarimetric sar data feature extraction. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 7545–7548, 2016. 2
- [29] S. Uhlmann and S. Kiranyaz. Integrating color features in polarimetric sar image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 52(4):2197–2216, 2014. 1, 4, 5
- [30] Fawwaz T. Ulaby and Charles Elachi. *Radar polarimetry for geoscience applications*, volume 1. Norwood, MA, USA: Artech House, 1990. 3, 4
- [31] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 8
- [32] Y. Xu, B. Du, and L. Zhang. Multi-source remote sensing data classification via fully convolutional networks and post-classification processing. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 3852–3855, 2018. 5
- [33] Naoto Yokoya, Pedram Ghamisi, and Junshi Xia. Multi-modal, multitemporal, and multisource global data fusion for local climate zones classification based on ensemble learning. pages 1197–1200, 07 2017. 5
- [34] Zhu, D. Tuia, L. Mou, G. Xia, L. Zhang, F. Xu, and F. Fraundorfer. Deep learning in remote sensing: A comprehensive

review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36, 2017. [1](#), [2](#)