Designing deep CNN models based on sparse coding for aerial imagery: a deep-features reduction approach

Abdul Qayyum, Aamir Malik, Naufal M Saad and Moona Mazher

Universiti Teknologi PETRONAS, Seri Iskandar, Malaysia

ABSTRACT

Traditional methods focus on low-level handcrafted features representations and it is difficult to design a comprehensive classification algorithm for remote sensing scene classification problems. Recently, convolutional neural networks (CNNs) have obtained remarkable performance outcomes, setting several remote sensing benchmarks. Furthermore, direct applications of UAV remote sensing images that use deep convolutional networks are extremely challenging given high input data dimensionality with relatively small amounts of available labelled data. We, therefore, propose a CNN approach to scene classification that architecturally incorporates sparse coding (SC) technique for dimension reduction to minimize overfitting. Outcomes were compared with principal component analysis (PCA) and global average pooling (GAP) alternatives that use fully connected layer(s) in pre-trained CNN architecture(s) to minimize overfitting. SC was used to encode deep features extracted from the last convolutional layer of pre-trained CNN models by using different features maps in which deep features had been converted into low-dimensional SC features. These same sparse-coded features were concatenated by means of different pooling techniques to obtain global image features for scene classification. The proposed algorithm outperformed current state-of-the-art algorithms based on handcrafted features. When using our own UAV-based dataset and existing datasets, it was also exceptionally efficient computationally when learning data representations, producing a 93.64% accuracy rate.

Introduction

UAV imagery is used for environment considerations and the monitoring of various resources. These remotely acquired data hold spatial and spectral properties that can be analyzed and applied for modeling (Jaakkola et al., 2010). UAV-based systems are cost-efficient reliable solutions for the monitoring of land-based objects. Limitations include non-uniform terrain such as hills, mountains and forests. Deployment is sometimes difficult when attempting to access resources and legal constraints can also hinder UAV monitoring.

UAV sensors have very high-resolution capability and have received much attention from researchers and manufacturers (Cheriyadat, 2014; Laliberte & Rango, 2009). Numerous UAV applications are used for resource management (C. Zhang & Kovacs, 2012) of vegetation, rivers (Gómez-Candón, De Castro, & López-Granados, 2014), disasters, seismography (C. Zhang & Kovacs, 2012) and precision agriculture (Gómez-Candón et al., 2014). UAV vectors operate at low altitudes to obtain high-resolution images that cover relatively small areas (Gómez-Candón et al., 2014). Due to challenges in data preprocessing such as triangulation, orthorectification and mosaicking (Rigas, Economou, & Fotopoulos, 2013), it is useful to classify UAV images in smaller parcels rather than larger mosaics.

Very high-resolution (VHR) acquired data are used for classification but holds wide-ranging variation regarding objects of interest. Increasing levels of detail increases computational complexity when using pixel-based classification approaches and can pose uncertainties in spectral signatures of urban objects. Acquired images also contain numerous statistical properties due to pixel dimensionality; hence, they are sometimes difficult to classify. Several papers on remote sensing images have considered improved classifiers and features representations. For scene classification, constructing a comprehensive (holistic) representation is a feasible solution. The Bags-of-Visual-Words (BOW) model (Sun, Sun, Wang, Yu, & Xiangjuan, 2012) for scene classification is effectively the most attractive solution currently in use by the remote sensing community. In last few years, extensive research has considered more suitable image descriptors for particular classification tasks. Local image descriptors such as local binary patterns (LBP) (Otávio A.B. Penatti, Valle, & Torres, 2012), histograms of oriented gradients (HOG) (T. Ojala, Pietikäinen, & Maenpaa, 2002) and scale-invariant feature transform (SIFT) (Ojala, Pietikäinen, & Topi, 2000) have been used to handcraft features descriptors,
especially for scene classification and object recognition tasks. Another features descriptor is based on SIFT-BOVW, and has been successfully applied in remote sensing applications for image scene classification tasks. Although these studies proved effective for scene classification of VHR images, there have been no improvements in BoVW-type techniques due to discrete constraints presented by the BoVW model.

Currently, deep learning methods (Maggiori, Tarabalka, Charpiat, & Alliez, 2016, Zhang, Zhang, & Bo, 2016, Zeiler & Fergus, 2014) have been successfully applied to classical image problems, including object recognition and detection, natural language processing, speech recognition and scene classification. Deep learning techniques have achieved much success with numerous improvements compared to state-of-the-art classical features extraction methods. As such, they are of significant interest in academic and industrial communities (Bengio, 2009).

Deep convolutional neural networks (CNNs) (Robinson & Yun, 2016) are recognized as the most worthwhile and are used in leading deep learning approaches for object recognition and detection tasks (Scherer, Müller, & Behnke, 2010). Their success is due to the efficient use of graphics-processing units (GPUs), rectified linear units, new dropout regularization and effective data augmentation (Chen & Lin, 2014). CNNs are popular in the computer vision community and in various applications based on natural language processing, hyperspectral image processing and medical image analytics. Their major advantage lies in deep architecture(s) (Zhang et al., 2016), which allows for the extraction of a set of discriminating features at multiple levels of abstraction. However, training a deep CNN from scratch (full training) presents limitations. First, they require large amounts of labelled scratch data. Second, deep training necessitates extensive memory and computational resources, without which the training process is extremely time-consuming. Third, deep CNN training is enormously complicated due to overfitting and convergence issues, which resolutions often call for reiterative adjustments in the network’s learning parameters. Consequently, deep learning from scratch is not only tiresome and time-consuming but also requires diligence, expertise and colossal amounts of patience.

An alternative to training from scratch is to fine-tune a CNN network based on large datasets derived from different extant applications. Pre-trained CNN models have been successfully applied in various computer vision tasks as baselines to transfer learning or as features generators (Robinson & Yun, 2016). Features extraction from pre-trained CNNs models that have been previously used as fine-tuned initial or later network components provide discriminating deep features useful for classification tasks. However, features extracted from the final layer of a pre-trained CNN have particularly high-dimensional features space based on small UAV datasets. High-dimensional features space can produce overfitting. Overfitting problems are solved by using (i) features reduction approaches based on a standard principal component analysis (PCA); (ii) sparse coding (SC) with pre-specified dictionaries; or (iii) a global average pooling (GAP) layer. In addition to introducing a GAP layer, these features-dimension reduction techniques can be applied to pre-trained CNN architecture(s) to further classify aerial images using standard classification algorithms.

The main contributions in this paper are as follows:

- Extraction of deep features from the final convolution layer (C5) of a pre-trained CNN model.
  - Pre-trained CNN models limit input image size due to the fully connected (FC) layer. To use any image size, the fully connected layer can be removed and the final convolutional layer can be used for deep features extraction.
  - Our dataset uses small training images. Due to size, CNN pre-trained models can produce overfitting.
    - Different features reduction techniques are used to reduce overfitting: these include a dropout layer, regularization, PCA and GAP. We therefore propose a SC technique using a hybrid layer to compress features in lower subspace to minimize overfitting based on CNN deep features. Performance of the proposed technique is compared with PCA, GAP, dropout and pre-trained CNN architecture(s) with fully connected layers (FC-CNNs) (see Section 5).
- The hybrid dictionary is based on the Ricker wavelet function and used to generate ridgelet based elements. It is introduced here specifically to extract sparse-coded features.
- Different spatial pooling techniques were applied to sparse-coded features to obtain global image features for scene classification.
- Different classification algorithms (SVM, RF and ELM) were tested to measure classification accuracy.

A CNN extracts features from high-level image data. Deep features provide detailed spatial information created by hierarchical structures. We investigated deep features’ characteristics using a classification framework with sparse representations. The objective was to introduce a features reduction technique utilizing the last layer of a pre-trained CNN architecture to minimize overfitting in small UAV datasets. Furthermore, instead
of using a fully connected layer, the GAP layer is introduced to minimize overfitting problems in pre-trained CNN architecture. Comparisons were made with outcomes from extant deep features techniques using standard pre-trained CNN-based architecture(s) for small UAV datasets and state-of-the-art datasets. We also compared our proposed algorithm with existing hand-crafted features-based classification techniques and are pleased to report results that showed greater accuracy and less overfitting. The GAP layer provided less computational costs and also minimized overfitting with a small UAV dataset.

Related works

Features extraction techniques are used to reduce data dimensionality in high-dimensional remote sensing data applications. The traditional data reduction method, per literature review, is PCA (Ham, Lee, Mika, & Bernhard, 2004). Different nonlinear dimensionality reduction methods, based on dictionary learning and manifold learning, have also been used for features reduction. The manifold learning method (Ham et al., 2004) is a state-of-the-art tool used for local descriptors in remote sensing image manifolds (Bachmann, Ainsworth, & Fusina, 2006). Auto-encoders are also used (Del, Licciardi, & Duca, 2009) to extract shallow structures in remote sensing but is computationally expensive and requires tuning for many parameters (regularization, learning rate), which heuristically limits network structure. The literature holds few reports on deep architecture use in remote sensing image classification by means of low-dimensional high-resolution images. The strength of deep networks that enhance noisy aerial image classification has been explored (Goodfellow, Lee, Le, Saxe, & Ng, 2009). Another author proposed a hybrid deep neural network to explore scale-variant features to detect vehicles in satellite images (Xueyun Chen et al., 2014). The framework based on stacked auto-encoders has been used to classify high-resolution data (Hoo-Chang Shin, Orton, Collins, & Leach, 2013).

Deep learning features extraction methods can handle nonlinear spatial–spectral image analysis and efficiently train algorithms that go unnoticed by state-of-the-art frameworks. A dictionary based on sparse representation modeling can efficiently represent features in low-dimensional space. Different dictionaries can be used for image classification and spatial–spectral sparse representation. Sparse representation by means of learned dictionaries is used to pan-sharpen images (Liping & Juncheng, 2014). A sparse Bags-of-Words tool can code automatic target detection (B. Zhao, Zhong, & Zhang, 2016). A saliency-based algorithm can code for segmentation (Zhang, Bo, & Zhang, 2015) and unsupervised learning using sparse features based on aerial images (Ranzato, Huang, Boureau, & LeCun, 2007). Thus, SC facilitates unsupervised classification instead of unsupervised features extraction techniques. The present work employs SC for features reduction via its incorporation with well-known CNN architecture(s) to minimize overfitting.

Dataset

The system is based on UAV composed of state-of-the-art Global Navigation Satellite System (GNSS) receivers on system board as well as the ground base station. The battery timing is almost 2 h (Michini et al., 2011). The images acquired have a resolution almost half of the centimeter. The software controlling the UAV as shown in Figure 1(a,b). The trajectory and all paths are automatically given to UAV for flying and landing using the joystick and controlling software associated with this UAV machine.

Our UAV can fly 72 km per hour and the battery timing is usually 3 h to maintain the flight. Our area of interest to monitor the vegetation or trees near power transmission poles is a village of Tambunan, Sabah, East Malaysia (Abdul Qayyum et al., 2017). The area is very challenging due to bad weather conditions, changing weather within minutes, cloudy environment and most of the time, rainy season. The vendor experienced that the only best window time is between 6.30 and 7.30 in the morning. The ground sample distance (GSD) varies and it depends on the height of the UAV. The 5 cm GSD is at the height of 150 m and 10 cm GSD for 300-meter height. The maximum height is 2000 m and the normal height is

![Figure 1](image-url). (a) UAV used for data collection of area of interest, (b) software for controlling UAV.
700 m. The 15 cm GSD is for the height of 700 m. This is the normal range to collect the data. In our design experiment, we used height of 700 m for 15 km span area in a square kilometer. The image sample used for classification based on four different classes (building, power poles, roads, trees) is used for classification.

### Methodology

The first step is aerial data acquisition using a UAV sensor. Deep features are extracted by pre-trained CNN models that employ small training images. The last convolutional layer, before the fully connected layer, is used for deep features. Features reduction techniques are applied to convert deep features into low-dimensional space. For our purposes, different features reduction techniques were used with the proposed SC tool, which is based on a hybrid dictionary, to express features in low-dimensional space. Finally, we employed a different approach to classification. Each step of the proposed method is now described.

#### Introduction of CNN and pre-trained models

The CNN has a variety of applications in the area of image and pattern recognition, natural language processing, video analysis, object recognition, speech recognition (Abdel-Hamid et al., 2014), object detection and classification (Scherer et al., 2010), scene parsing and scene classification (Abdul Qayyum et al., 2017). The deep CNN models capture the data in hierarchical way and these models are based on sequential modules, where output of previous module is the input of next module. These modules are called layers. Each layer is parameterized by a set of random weights and biases units. The weights in CNNs models are shared locally; the same weights are applied at every location of the input at every layer. The filter is formed by connecting the weights with output unit. The CNN consists of convolutional layer which contains input and set of learnable filters, a nonlinear function and pooling layer, which sums the statistics of the features to reduce the computation cost. The CNN architectures have numerous hyper-parameters (learning rate, momentum and regularization) for training the layers using input data samples. The operations accomplished in a convolutional layer can be summarized as

\[
A^l = \text{pool}_i(\phi(A^{l-1} * f^i))
\]

Where \(A^{l-1}\) is the input feature map to the \(l^{th}\) layer, \(f^i = (w^i, b^i)\) is the set of learnable parameters based on weights and biases of the layer, \(\phi(\cdot)\) is the pointwise nonlinearity, pool is a subsampling operation, \(t\) is the size of the pooling region and \(*\) the symbol denotes linear convolution. The input data for first layer is based on UAV colored image. i.e., \(A^0 = 1\), where \(I \in R^{W_1 \times H_1 \times C_0}\) is the input image, \(V^0\) and \(H^0\) are its width and height, respectively; and \(C_0\) is the number of input spectral channels. The input to a subsequent layer \(l\) is a feature map \(A^{l-1} \in R^{V^{l-1} \times H^{l-1} \times C^{l-1}}\), where \(V^{l-1}\) and \(H^{l-1}\) are the width and height of the \(l^{th}\) layer's input feature map and \(C^{l-1}\) is the number of outputs of the \((l-1)^{th}\) layer. The most important steps to design a CNNs models are the number of layers, the size of receptive field and type of spatial pooling and another important feature is how to train such architectures. The deep CNNs can be trained using standard back propagation (Kwon, Kim, Jinoh Kim, Suh, & Kim, 2017) algorithms.

For classification problems and especially scene classification, the main problem is how to generalize the algorithm by utilizing deep learning algorithms, for example, a CNN (Kwon et al., 2017). We discussed the brief review of different CNN successful modern architectures as shown in our work.

#### AlexNet

It is an innovative deep CNN architecture and as compared to older networks, AlexNet is the combination of five convolutional layers and three fully connected layers. The main advantage and successful point is its network used data augmentation, dropout and rectified linear units (ReLU) non-linearity. The ReLU function contains only a positive number of the training phase; the data augmentation is used to reduce overfitting with larger CNNs and uses small patches of images with horizontally and vertically flipped patches of original images. The dropout method also reduces co-adaptations of neurons and substantially reduces overfitting in fully connected layers. The first convolutional layer has 96 channels and 55 width and height. The second convolutional layer consists of 256 numbers of planes and 27 by 27 width and height. The third layer has 384 numbers of channels with 13 × 13 width and height and similarly the size of the fourth convolutional layer is same as with third convolutional layer. The fifth convolutional layer has same width and height as compared to fourth, but a number of channels are different which is equal to 256. The layer 6, 7 and 8 are fully connected layers with 4096 dimensions in FC6 and FC7. The FC8 has a different size which is equal to 1000 dimension.

#### CaffeNet

It is used for general purpose CNNs and also used in other deep models with changeable and perhaps the fastest existing implementations for effective training. It has a similar architecture as AlexNet with two
modifications. The first is that it is used without data augmentation and the second is that it uses a different order of pooling and normalization layers. The performance of this network model is similar to AlexNet.

**VGGNet**
The VGG-F is very fast and simple to use and similar to AlexNet. The fundamental difference between VGG-F and AlexNet is that it uses a smaller number of filters and strides in convolutional layers. Furthermore, there are two other types of VGGNet, VGG-M and VGG-S. The VGG-M is the medium architecture and has a change in the pooling and smaller stride in the first convolutional layer. VGG-S has very slow and simplified architecture and uses a smaller number of filters in fifth layers. Particularly, The VGG-F has been used in this experiment due to simple and fast response.

**GoogLeNet**
The GoogLeNet (Hu, Xia, Jingwen, & Zhang, 2015) proposed in ILSVRC-2014 competition for classification and detection. Its main distinctiveness is the use of inception modules that decrease the complexity of the lavish filters of traditional architectures. The inception modules have multiple filters used in parallel with different resolutions. GoogLeNet utilizes the filters of various sizes at the same layer that maintains more spatial information. It uses small number of parameters in the network which may provide lesser overfitting problem. The GoogLeNet has 22 layers and more than 50 convolutional layers dispersed inside the inception modules.

**CNN deep features based on sparse coding**
Features generated by the CNN model using spatial data are high dimensional and not effective for classification purposes. We therefore propose a SC technique using subspace-based representations of deep features that improve classification performance by reducing features space. Images hold high-dimensional objects that present a difficult task for machines classification. Nonetheless, UAV images contain high-dimensional spatial and spectral features that can be represented in low-dimensional subspace. Low-dimensional features are encompassed by adapting fixed as well as adaptive dictionaries atoms that employ training pixels of the same class. Test pixels of an unknown class are represented as a linear combination of dictionary atoms. SC is used at the last convolutional layer to reduce features space acquired from pre-trained CNN architecture. Such reduced dimensions improve overfitting for small datasets. Figure 2 outlines the proposed method used for features reduction.

**Sparse representation and the proposed hybrid dictionary**
Sparse representation is widely used in numerous signal and image processing applications, in machine learning, in neuroscience to study brain signals, and in biomedical imaging (Fang & Shutao, 2010). Its main purpose is to represent signals as linear combinations with typical patterns called atoms that are then extracted from an over-complete dictionary. For instance, if $y$ is a signal or 2D image that belongs to $N$ dimensional space; then $y \in \mathbb{R}^n$ and $D \in \mathbb{R}^{n \times M}$ can be used to form a dictionary matrix where $M$ atoms are defined as column vectors, $d_j \in \mathbb{R}^M$ and $j = 1, \ldots, M$. We were interested to find a sparse vector such that $x \in \mathbb{R}^n$ and $y \simeq Dx$. The problem then becomes one of the optimization (Mailhe, Gribonval, Bimbot, & Vandergheynst, 2009):

$$\min \|x\|_0 \text{ subject to } \min \|y - Dx\|_2 \leq \varepsilon$$  \hspace{1cm} (2)

where $\varepsilon$ is the reconstruction error of signal $y$ using dictionary $D$; and $x$ is the sparse code.

Alternatively:
\[
\min \|y - Dx\|_2 \text{ subject to } \|x\|_0 \leq \rho 
\]  
(3)
where \(\rho\) is specific sparsity level.

The vector, \(x \in \mathbb{R}^k\), represents coefficients of a given signal \(y\) with respect to dictionary \(D\). Compared to PCA, SC computes a sparse vector with a minimum of non-zero coefficients.

Sparse coefficient construction necessitates \(l_0\)-norm, which is used to count non-zero entries of a specified vector. This construction type becomes a NP hard problem (A. Qayyum et al., 2016) and can only be solved by optimization greedy algorithms (OGA). The simplest and most efficient of these algorithms are matching pursuit (MP) (Chen, Donoho, & Saunders, 1998) and orthogonal matching pursuit (OMP) (Mailhe et al., 2009). Their essential gist is to acquire the sparsest of coefficients by competently employing dictionary elements.

A dictionary is “over-complete” when the number of pixels in the image patch are fewer than the number of elements in the dictionary used for sparse representation of the image patch. This caused us to improve various over-complete dictionaries that have more atoms than dimensions in the signal, which then guarantees signification within a wider range of signal phenomena. Our attentive effort to minimize loss by an over-complete dictionary showed promising properties provided by orthogonal transforms. The most important steps are the “how” of construction and the “how” of choosing an efficient dictionary that offers an optimized solution with sparse coefficients. Figure 3 illustrates a block diagram, based on SC, of the proposed approach using features maps from the last convolutional layer to build features vectors for use as training images.

Selecting an over-complete dictionary is fundamentally important to the process of generating sparse recovery atoms. Tasks used in image processing applications include fusion, super-resolution, denoising and others, depending on the type of dictionary castoffs used for sparse representation.

In this paper, sparse representation is used to estimate basis elements responsible for the efficient generation of sparse coefficients. Therefore, a ridgelet-based over-complete dictionary was the better choice for sparse-based image classification. We propose the Ricker wavelet function, a negative normalized derivative of the Gaussian function to produce ridgelets as basis elements for the hybrid dictionary, and for which experimental results demonstrated superior performance. Defined 2D ridgelets were based on a wavelet type function Equation (4):

\[
\psi_{x,y,t}(t_a, t_b) = x^{-1/2}((t_a \cos \theta + t_b \sin \theta - x)/y) 
\]  
(4)

where \(\psi(\bullet)\) is the wavelet function based on the Ricker wavelet, shown in Equation (5).

\[
\psi(t) = 2/3 \pi^{1/4}(1 - t^2/\sigma^2)e^{-t^2/\sigma^2} 
\]  
(5)

Ridgelet bases were obtained by choosing different values for \(x, y\) and \(\theta\) in Equation (4). These bases were then used as vectors for enclosure in the hybrid dictionary using ridgelet functions.

Ridgelet analysis of object representation is very effective with singularities along lines derived by a concatenation of 1D wavelet transforms (Do & Vetterli, 2003). Singularities are frequently joined together along edges in an image, which in fact inspired our use of ridgelet transforms. Thus, the proposed hybrid dictionary (based on ridgelet bases functions) promised to be the better choice for the construction of an over-complete dictionary, which,

![Figure 3](image-url). Proposed model using features maps from the last convolutional layer to produce output features vectors using sparse coding.
in turn, would provide better approximations for sparse representation.

After dictionary generation, sparse coefficients were extracted based on the proposed dictionary and then pooling was applied to sparse-coded features which were converted into global features for image classification. Pooling is used by numerous visual recognition methods to combine nearby spatial features detectors into local and global features that can be used to remove irrelevant data from input images. Pooling offers many advantages to image transformation and provides a more compact representation that obtains robust noise management. The pooling operation is typically a sum, an average, or some other commutative combination.

The max pooling function is defined by the absolute value of sparse-coded coefficients Equation (6).

\[ Z_i = \max\{|x_{i1}|, |x_{i2}|, ..., |x_{iM}|\} \]

where \( Z_i \) is the \( i^{th} \) element of the pooling function; \( x_{ij} \) represents the matrix element at the \( i^{th} \) row and \( j^{th} \) column of \( X \), which is the sparse-coded feature matrix; and \( M \) shows the number of feature maps.

Table 1 shows the algorithm used to compute a features matrix based on deep features extracted from last convolutional layer which is composed of features maps. These features maps are passed to SC to encode features as sparse coefficients, which are then combined, using pooling, and then further concatenated as the features matrix. After the pooling of sparse coefficients, the features matrix is obtained by using all training images and then fed into the classifier for image classification. Classification algorithms are discussed in the next section. Figure 3 shows the proposed method for extracting coded features based on pooling and SC.

**CNN based on GAP layer technique**

Traditional convolutional networks use lower layer convolutions. Features maps in the last layer are converted into vector forms and sent to the fully connected layer for the classification task, which employs standard classification algorithms such as softmax logistic regression (Farabet, Couprie, Najman, & Yann, 2013). This structuring thus uses convolutional layers for features extraction, which features are then applied to traditional classifiers.

Traditional CNNs can produce overfitting due to fully connected layer parameters.

Different methods are used to avoid overfitting and produce the generalization ability of the CNN model (Jarrett, Kavukcuoglu, Ranzato, & LeCun, 2009). In the present work, we use GAP to minimize effects from overfitting. The idea standing behind GAP is to generate a features map for each corresponding category of the classification task. Instead of using the fully connected layer, an averaged vector from each features map is fed into the classifier according to the number of classes (see Figure 4a). The advantage of GAP over the use of a fully connected layer is that it uses less parameters yet produces better correspondence between features maps and categories. Another advantage is that there is no need to optimize parameters because it avoids overfitting and improves accuracy rates for both training and testing. Furthermore, GAP sums spatial information; thus, it is more robust for spatial translations of input. Figure 4(b) outlines GAP and fully connected layers.

**Classification algorithms**

**Support vector machine**

The support vector machine (SVM) has been used widely in image processing and pattern recognition applications. It uses hyperplane for separating training data using multidimensional training values of fixed different number of classes. The SVM classification detail is found in (Palaniappan, Sundaraj, & Sundaraj, 2014). The SVM classifier is introduced as binary classifier. The binary SVM classifier is converted into multiclass classifier using two strategies. These strategies are called one against one and the one against all (Manikandan & Venkataramani, 2009). The OAO classifier technique classifies every pair of class and using the most common label for each pixel. The OAA technique classifies each class against the rest and it chooses the label for each pixel with largest confidence. This strategy performs better when a number of classes are small usually less than

**Table 1. Proposed algorithm based on sparse coding using deep CNN features.**

| FOR each \( i = 1 \): training data do |
| --- |
| FOR each \( k = 1 \): \( M \) do |
| Extract deep features using CNN last convolutional layer \( A^l = \text{pool}_d(A^{l-1} \ast f^l) \) |
| Extract features maps from \( c_l Y = [y_1, y_2, ... , y_n] \) where each \( y_i \in R^{n \times k} \) is a feature vector and \( k \) is the total number of features maps. |
| Design dictionary matrix based on proposed dictionary, where \( M \) is the dictionary size \( D = [d_1, d_2, ..., d_M] \) \( D \in R^{n \times M} \) |
| Design sparse coding based on dictionary and input features maps \( \min \| x_0 \| \) subject to \( \min \| y - DX_0 \|_2 \leq \epsilon \) |
| Extract sparse codes for features coefficient matrix: \( X = [x_1, x_2, ..., x_M] \) \( X \in R^{n \times M} \) |
| \( U = \max \| x_0 \| \) \( \forall k = 1, ..., M \) |
| \( F_M = [U_1, U_2, ..., U_M] \) \( F_M \in R^{n \times M} \) |
| ENDFOR |
| ENDFOR |
In this paper, we used OAA due to usage of a small number of classes (less than five).

**Random forest algorithm**

The random forest (RF) can be used for image classification in remote sensing application due to its superiority and robustness to noise as compared with other classifiers (C. Zhao et al., 2017). Feng, Liu, and Gong (2015) proposed RF and it was based on ensemble learning technique. It required less number of parameters while running as compared to other machine learning classifiers (SVM, ANN). The popularity of RF increased gradually due to an achievement of equal or higher accuracy in comparison with SVM for image classification in the field of remote sensing (Feng et al., 2015). RF is based on an ensemble of much independent individual classification and dependent as regression tree (CART), it can be defined as

$$g(u, t, \theta), \quad t = 1, 2, \ldots, j$$

(7)

Where $g$ denoted as RF classifier, $u$ is the input feature vector and $\theta$ is the predictor variable which is used as independently identically distributed (i.i.d.) random process for producing each CART tree. The RF had the final response for calculating all decision trees output. There is less issue of overfitting due to individual decision tree in RF. Due to abundant advantages, the RF has great potential for classifying the UAV images in the remote sensing field.

**Extreme learning machine (ELM)**

The extreme learning machine (ELM) is a basic and new learning algorithm and has been developed using the basis of single hidden layer feed-forward neural networks (SLFN) (G. Huang, Huang, Song, & You, 2015). It is a very time-consuming process to adjust the input weights and hidden layer basis for all feed-forward neural networks. To minimize or overcome these problems using traditional gradient-based learning algorithms, G. B. Huang, Zhu, and Siew (2006) proposed an SLFN by randomly choosing input weights and hidden layer biases for infinite activation function in the hidden layer. The SLFN can be observed as a linear system and determination of output weights updated analytically. The deep details of ELM are found in Huang et al. (G. Huang et al., 2015) and Song et al. (G. Huang et al., 2015).

**Simulation and results**

A CNN-based framework achieves higher overall accuracy when compared to handcrafted-based features extraction techniques. Most remote sensing-based scenes use identical features belonging to different groups depending on spatial features and spectral intensity. A CNN framework extracts indispensable features from high spatial resolution remote sensing images in which generalized fundamental features are gradually changed into higher-level data. High-resolution remote sensing applications require high-level data in more robust forms in addition to extremely effective low-level features representations of significance in datasets for small scenes. The CNN model thus provides a more robustly persuasive performance with a combination of high- and low-level features data.

We observed superior performances in urban imaging of buildings, roads, trees and power transmission poles, but reduced results for vegetation and trees. Even a fusion of multiple handcrafted features cannot generate comparable results with CNN-based UAV high-resolution datasets. Average accuracy rates of the proposed technique when using machine learning algorithms were 80% for training and 20% for testing with a small UAV dataset. This dataset contained four classes (buildings, trees, power transmission poles and roads). Each class comprised 625 samples. Different numbers of training class samples were used to evaluate deep features using various...
classification algorithms. Two training samples per class were used for features extraction and classification (Figure 5).

We computed accuracy using ground truth versus estimations per Equation (8).

\[
\text{Accuracy} = \frac{TP + TN}{(FP + TN + FN + TP)} \tag{8}
\]

\(TP = \text{true pos}, \; FP = \text{false pos}, \; TN = \text{true neg}, \; FN = \text{false neg}\)

Precision and recall are the most commonly used evaluation metrics in pattern recognition and information retrieval assessments. Both depend on the relevance of classification criteria and can be defined as the ratio of the number of retrieved elements to the total number of relevant elements in an instance per Equation (9). This ratio depends on true positive and false positive values and is calculated per ground truth sample labels versus estimated sample labels.

\[
\text{Precision} = \frac{TP}{(TP + FP)} \tag{9}
\]

\[
\text{Recall} = \frac{TN}{(TN + FP)} \tag{10}
\]

Recall is defined as the ratio between the number of relevant elements in an instance to the number of retrieved elements per Equation (10). It is calculated as ground truth labels versus estimates (computed) per the proposed algorithm and then compared with results from handcrafted feature algorithms. Figure 5(a) shows performance accuracy rates. SC-based features reduction method produced superior results using the RF algorithm compared to other methods. Figure 5(b) shows that precision value results for the proposed CNN-based method were nearly comparable with accuracy outcomes. Features extraction from the last fully connected layer, without reduction (CNN-FC), produced better results compared to PCA and GAP features reduction techniques and also showed less accuracy compared to SC-based features reduction. The dropout layer was used to reduce overfitting in pre-trained networks and then applied to features reduction of CNN features. Dropout can improve CNNs by reducing overfitting (Yu, Xiaomin, Luo, & Ren, 2017) and is widely used in many deep learning applications. The chief idea is to reduce the co-adaptation of hidden units. The dropout operation sets the output of each neuron to zero with probability. By using this strategy, the network is forced to learn more robust features and reduce noise. The drop rate was set at 0.6 or greater, which is higher than the most commonly used value of 0.5. The dropout layer provided less accuracy than the proposed method (see Figure 6(a)).

Precision values increased when lowering the number of layers and with increasing numbers of samples. Our SC-based approach also produced high precision value with the RF algorithm. The performance recall matrix showed similar results and were nearly equal with marginally increased values using the PCA design and RF classifier (see Figure 6(c)). Precision and recall values using the dropout layer also produced improved results when using SVM and RF algorithms. Accuracy, precision and recall performance metrics were evaluated using pretrained CNN-VGG16 features.

Using pre-trained CNN models with our proposed features reduction techniques, we calculated computation time (Figure 7). The VGG16 pre-trained network consumed less time compared to other CNN networks. Furthermore, the GAP layer removed the fully connected layer and required less computing time due to the use of fewer parameters. Due to network complexity, SC and PCA techniques used comparatively more time, especially when using the ELM algorithm. Thus, the VGG16 network was more effective for real time application. The proposed SC approach also yielded higher accuracy with pre-trained VGG16 networks (highest results, Figure 8).

Pre-trained CNN models achieved similar classification accuracy rates with GoogLeNet and VGG-16, which were comparatively better than AlexNet and
CaffeNet. GoogLeNet yielded better accuracy than did AlexNet and CaffeNet, but results were unimpressive compared to VGG-16. The likely reason for this outcome being high-level semantic scene input data from extracted features of the intelligent CNN features learning mode. When comparing the number of hidden layers, GoogLeNet and VGG-16 produced better results and showed more profound structures than other pre-trained CNN models, which held eight layers. By increasing the number of layers, extracted features became deeper and more closely related to high-level semantic scene data, yielding a more highly descriptive features vector. Hence, GoogLeNet and VGG-16 demonstrated better results for image scene classification of VHR imagery than did AlexNet and CaffeNet.

Comparisons were also made between all pre-trained networks using (a) the last fully connected layer (CNN-FC) and (b) the dropout layer. Features extracted from

Figure 6. Comparison of accuracy rates based on a VGG16 pre-trained network using SC, FC, PCA, GAP and dropout layer: (a) accuracy, (b) precision, (c) recall.

Figure 7. Computational time comparison for pre-trained CNNs using SC, PCA and GAP.
the last fully connected layer had dimensional lengths of \(1 \times 4096\) (GoogLeNet) and \(1 \times 1024\) (VGG16), respectively, for AlexNet and CaffeNet. These deep features were used by various classification algorithms to classify aerial images. VGG16 provided the highest classification accuracy (94.64\%) using the SVM classifier based on proposed UAV dataset. GoogLeNet, using the ELM algorithm, had the second highest classification accuracy rate (Figure 9(a)). Accuracy rates produced by the proposed SC features reduction technique were superior to using the fully connected layer of pre-trained networks. The dropout layer was also used to reduce overfitting and enhance features reduction for all pre-trained networks, but it provided less accuracy for all classifiers (SVM, ELM and RF) (Figure 9(c)). This was likely due to low accuracy from pre-trained networks from the UAV based dataset. Some classes from the ImageNet dataset of pre-trained CNN networks were slightly different from the UAV-based dataset.

**Figure 8** shows different learning rate outcomes for proposed CNN architecture(s) (SC vs. PCA). Average accuracy increased with an increasing learning rate, with the highest achieved at a learning rate of 0.9 using either SC or PCA. Training and testing errors were also evaluated using proposed and existing features reduction techniques and pre-trained CNN architecture(s) with FC (St-CNN). The proposed SC-based CNN architecture minimized overfitting for a small dataset by using a reduced set of features, as did the GAP layer method. Both CNN-based architecture(s) were intended to reduce overfitting problems. Training and testing losses using the standard CNN model with a fully connected layer on a small dataset showed high training and testing values (Figure 10). SC- and PCA-based training and testing values are seen in Figure 10(a,c). Overall, combined comparisons of SC, PCA and GAP training and testing losses with standard CNN architecture(s) are shown in Figure 10(d). These results demonstrate that SC-CNN

**Figure 9**. Comparison of accuracy rates using pre-trained networks (AlexNet, CaffeNet, GoogLeNet and VGG16): (a) proposed CNN-SC; (b) pre-trained networks using last fully connected layer (CNN-FC); (c) pre-trained networks using the dropout layer.
produced smaller training and testing gaps, which indicated no overfitting for a small UAV dataset. Figure 10 shows a comparison of training versus testing losses with respect to proposed and standard CNN based algorithms. The largest gap between training and testing losses is seen in Figure 10(a) for the proposed St-based features using CNN. Blue and black lines show training and testing gaps compared to GAP and PCA based features gaps in Figure 10(b,c). Complete training and testing losses for proposed versus fully connected features (St.CNN features) are shown in Figure 10(d).

**Visualization of CNN filters and the last convolutional layer**

CNN filters used in the experiment for each convolutional layer are shown in Figure 11. The size of each filter was $5 \times 5$. These filters were used to extract features maps. A CNN’s power lies in its weights and starting point. Weights were selected randomly and set to a standard deviation of 0.001. Weights for the UAV image kernel of the first convolutional layer are shown in Figure 11. The first column represents six features maps of the convolutional layer. The second column represents six features maps of the ReLU layer. The third column represents the normalization layer of features maps. The last column comprises six features for the first pooling operation. Convolutional, ReLU, normalization and pooling layers of the CNN model are shown in Figure 12. All results are taken from the Tower Class dataset. Visualization results showed that deep spatial properties of the CNN features maps were able to extract prominent features. An instinctive

---

**Figure 10.** Training and testing losses: (a) St.CNN versus SC; (b) St.CNN versus GAP; (c) St.CNN versus PCA; (d) St.CNN versus proposed CNN architecture(s).

**Figure 11.** Filter weights used in first convolutional layer (size $32 \times 32$ due to 32 features maps). Each image (size $5 \times 5$) held a total of 32 kernels used in first convolutional layer.
understanding of CNN activation allows that each layer’s visualization was achieved by image inversion and reconstruction by the proposed model. Features extracted from convolutional layers were reconstructed as images similar to the original. Deeper layers produced blurred images while features taken from the last fully connected layer could not be inverted to any recognizable image. Reconstructed images contained several similar but meaningful yet randomly distributed components. Results demonstrated that data from low-level layers were reorganized by FC layers to recreate further abstracted representations. Reconstruction results of local region’s features maps taken from Conv1–Conv5 and fully connected layers (Fc6–Fc8) for various classes (Tower, Building, Vegetation, Road) are shown in Figure 13.

The receptive field was of a larger size on features maps with respect to deeper layers for the tower-class (Figure 14). By increasing the number of hidden layers, features vectors grew more descriptive and presented higher degrees of semantic data. Results showed that deep CNN models have an important role to play in the representation of deep extracted features.

Comparison with existing methods

According to the prior reports, several descriptors have been used (van de Sande, Gevers, & Snoek, 2010) for remote sensing image classification. These include texture, color-image retrieval/classification, and web image retrieval. Our comparisons were based on the BOW model, Color Histograms and variations as the most effective methods for performance valuations of low- and mid-level features. Table 2 shows comparative outcomes of the proposed method versus existing handcrafted features methods.

Low-level descriptors

Numerous descriptors used to represent visual elements are reported in the literature (Otavio A. B. Penatti, Nogueira, & Dos Santos, 2015). Different descriptors of distinct data components related to remote sensing images, including color, texture and gradient, were evaluated for deep features extraction and used for outcome comparisons in this study. Global low-level descriptors were: (1) border/interior pixel classification (BIC) (Stehling, Nascimento, & Falcão, 2002); (2) local color histogram (LCH) (Swain & Ballard, 1991); (3)
histogram of oriented gradients (HOG) (Swain & Ballard, 1991); and (4) auto-correlogram color (ACC). Implementation details are found in Otávio A.B. Penatti et al. (2012). Implementation of the HOG descriptor is found in the VLFeat framework (Vedaldi & Fulkerson, 2010).

Figure 13. Reconstructed images: (a) five conv1 and three fully connected layers for class 1 (tower); (b) same layers for class 2 (building); (c) same layers for class 3 (vegetation); (d) same layers for class 4 (road).

Figure 14. Reconstructed images based on receptive fields for the tower-class sample based on the pre-trained CNN model.
### Table 2. Comparison of accuracy rates: proposed method versus handcrafted-based features extraction.

| Methods                | Training samples (100) | Training samples (200) | Training samples (300) | Training samples (400) | Training samples (500) |
|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|
| SIFT                   | 77.23                  | 76.32                  | 82.11                  | 82.61                  | 82.97                  |
| BoCF                   | 78.11                  | 79.31                  | 81.41                  | 81.49                  | 81.97                  |
| LCH                    | 74.45                  | 73.51                  | 80.13                  | 80.51                  | 80.91                  |
| HOG                    | 63.67                  | 68.51                  | 73.45                  | 79.95                  | 78.11                  |
| ACC                    | 77.32                  | 78.41                  | 82.14                  | 82.71                  | 83.56                  |
| Proposed SC            | 81.78                  | 82.67                  | 85.12                  | 87.23                  | 93.34                  |
| using VGG16            |                        |                        |                        |                        |                        |
| Proposed PCA           | 82.24                  | 83.45                  | 86.67                  | 88.23                  | 91.16                  |
| using VGG16            |                        |                        |                        |                        |                        |
| Proposed GAP           | 84.89                  | 85.78                  | 87.34                  | 91.49                  | 92.13                  |
| using VGG16            |                        |                        |                        |                        |                        |

### Mid-level descriptors

BoVW and variations (Avila, Thome, Cord, Valle, & Araújo, 2013) are used in remote sensing images as mid-level features descriptors. BoVW uses a codebook to discriminate patches then computes statistics for each visual word occurrence in a test dataset. This has been state-of-the-art for the computer vision community for years and remains a good work horse for many image tasks. For BoVW in our original dataset, we used the scale-invariant feature transform (SIFT) (Lowe, 2004) as the features descriptor.

We evaluated the proposed CNN-SC’s performance with the NWPU-RESISC45 dataset (Cheng, Han, & Xiaoqiang, 2017). This benchmark contains 31,500 images divided into 45 classes. Each class holds 700 (256 × 256) images for red–green–blue (RGB) spectral ranges. Spatial resolution changes from 30 to 0.2 m per pixel for most scene classes. This dataset has the largest number of scene classes and total images. Rich image variations, large within-class diversity, and high between-class similarities make this dataset robust and challenging.

Table 3 shows comparative experimental outcomes using the NWPU-RESISC45 dataset. The proposed algorithm with SC representation and pre-trained VGG16 proved excellent. Neither GAP nor PCA achieved better results than sparse representation. Both methods used the SVM classifier for classification. BoCF’s outcome was also slightly less than CNN-SC (Cheng, Zhenpeng, Yao, Guo, & Wei, 2017).

Used by the US Geological Survey Department, UC Merced data are an open-source publicly available aerial image dataset of approximately 30-cm spatial resolution (Yi Yang & Newsam, 2011). It has a total of 2100 (256 × 256) image patches (RGB bands) of various American cities with 21 semantic categories, each with 100 samples per class.

Table 4 shows comparative performance outcomes for the proposed CNN-SC algorithm versus extant scene classification algorithms using the UCM dataset. The CNN-SC algorithm yielded results superior to all others with only the gradient boosting random convolutional network (GBRCN) approaching its accuracy. All pre-trained networks using fully connected layer features provided less accuracy.

We performed various cross-validation procedures on proposed techniques to enhance results. Optimal results were achieved for 5- and 10-fold (k = 5,10) cross validation as shown in Table 5, showing little room for improvement. Cross-validation with different k values based on our own and on existing datasets showed better performance at 10-k fold for the
proposed approach versus existing techniques for all datasets. Experimentally, and due to limited space, only the best results are shown.

Discussion

The major contribution of this work is its CNN-based design and use of a small remote sensing image dataset based on features reduction techniques. This is in addition to our comparison of results to those obtained by GAP and PCA techniques that replaced fully connected layers with pre-trained CNN architecture(s). Results were also compared with approaches using pre-trained CNN architecture(s) with fully connected layers. Using a small training dataset, proposed techniques using existing pre-trained architecture(s) can be exploited by remote sensing applications in place of fully trained networks for image scratching. Our objective was not to emphasize high performance but rather to explore the use of pre-trained architecture(s) for fine-tuning, which showed excellent capability compared to those based on fully trained networks from scratch.

We evaluated CNN-based architecture(s) based on our own UAV dataset and compared performances with state-of-the-art datasets used by the remote sensing community. The proposed CNN-SC method uses deep features acquired from the last convolutional layer of pre-trained CNN architecture(s) for scene classification of aerial images. Our hybrid dictionary uses SC ridgelet-basis elements and yielded excellent results with an exceptional ability to express discriminative features for scene classification. The proposed method also reduces features space from high-dimensional to low-dimensional. In addition, SC features are concatenated to generate global features to further enhance scene classification. By using pre-trained CNN architecture(s) to reduced features dimensions, and by employing machine learning algorithms (SVM, ELM, RF) for classification of reduced features-dimension sets, the proposed system avoided overfitting with a small UAV dataset.

Existing features reduction techniques (GAP and PCA) were used to minimize overfitting and provide opportunity for pre-trained CNN architecture(s) use in small datasets. Four different pre-trained CNN architecture(s) (AlexNet, CaffeNet, GoolgLeNet and VGG16) were fine-tuned by incorporating them with SC, PCA and GAP features reduction techniques. Instead of fine-tuning all layers in all networks, only the last layers were fine-tuned by random weights and hyper-parameters.

Training and testing losses were calculated using pre-trained CNN architecture(s) with and without proposed CNN architecture(s). Deep SC-based CNN architecture(s) outperformed all others by minimizing training and testing error gaps. Gaps between training and testing errors were reduced by changing hyper-parameter regularization approaches such as “weight decay” or “dropout”, or by using fewer parameters in fully connected layers. Optimized performance of deep SC-based CNN models can be achieved by using a larger network and by appropriately setting regularization hyper-parameters in the last layer of the network. However, computational complexity is increased with the use of a larger network. The main goal of this paper is to show that the proposed SC–CNN-based networks produces better performance with pre-trained networks and increases efficiency due to the small gap between the testing and training errors.

Features reduction techniques can reduce overfitting in CNN-based models for small datasets. Our SC approach to features reduction minimized overfitting compared to PCA- and GAP-based algorithms. Pre-trained CNN models without features reduction obtained larger gaps between training and testing losses, which likely indicates overfitting in the CNN model for small datasets. However, it is difficult to recognize whether a system is underfitting or suffers other defects when both testing and training errors are high. If a training-testing gap in a fine-tuned network is not within an acceptable range, even when improving regularization hyper-parameters more data would be required to achieve optimized performance. Thus, we can conclude that training and testing losses are helpful when determining a system’s over/under-fitting status when gaging performance and fitting it for an optimized parameters.

When developing machine learning models, researchers use different sets of datasets daily with different training sample sizes (increased or decreased), and adjust regularization parameters to affect a deep model’s capacity in terms of error rate and computational resources (memory, runtime). They change hyper-parameters of training optimization algorithms to improve optimization techniques related to deep CNN architecture(s). As evidenced in the literature, CNN-based automatic features extraction methods outperform handcrafted methods in aerial remote sensing imagery. Hence, it is more than reasonable to apply deep pre-trained CNN networks for use as deep features extractors such as our proposed and existing features reduction techniques. The proposed SC–CNN-based deep CNN features obtained high performances compared to existing features reduction methods (GAP, PCA and dropout) using UAV-based and existing state-of-the-art remote sensing datasets. Results were also compared with low-level and mid-level handcrafted state-of-the-art techniques using UAV and existing datasets. Comparisons between SC-CNN were also made
Conclusion

We based this method on pre-trained CNN architecture(s) that incorporated SC for features reduction, specifically to minimize overfitting. Performance outcomes were compared with PCA- and GAP-based techniques, which also used pre-trained CNN architecture(s) to minimize overfitting. The proposed method used a pooling technique to concatenate SC features extracted from the proposed hybrid dictionary for scene classification. The proposed algorithm was extremely computationally efficient at learning data representations and outperformed all current state-of-the-art algorithms in aerial scene classification for both extant and our own UAV dataset. Future work will consider a larger number of hyper-parameters to further minimize overfitting and improve accuracy for use in remote sensing scene classification with different number of samples.

Acknowledgments

This work is supported by Ministry of Green, Water and technology Malaysia collaborated with Tanaga Nasional Berhad and Universiti Teknologi PETRONAS (UTP).

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Abdel-Hamid, O., Mohamed, A.-R., Jiang, H., Deng, L., Penn, G., & Dong, Y. (2014). Convolutional neural networks for speech recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22(10), 1533–1545. doi:10.1109/TASLP.2014.2399736
Avila, S., Thome, N., Cord, M., Valle, E., & Araújo, A.D.A. (2013). Pooling in image representation: the visual codelword point of view. Computer Vision and Image Understanding, 117(5), 453–465. doi:10.1016/j.cviu.2012.09.007
Bachmann, C.M., Ainsworth, T.L., & Fusina, R.A. (2006). Improved manifold coordinate representations of large-scale hyperspectral scenes. IEEE Transactions on Geoscience and Remote Sensing, 44(10), 2786–2803. doi:10.1109/TGRS.2006.881801
Bengio, Y. (2009). Learning deep architectures for AI. Foundations and Trends® in Machine Learning, 2(1), 1–127. doi:10.1561/2200000006
Chen, S.S., Donoho, D.L., & Saunders, M.A. (1998). Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing, 43(1). Society for Industrial and Applied Mathematics: 129–159. doi:10.1137/S003614450037906X
Chen, X.-W., & Lin, X. (2014). Big data deep learning: Challenges and perspectives. IEEE Access, 2, 514–525. doi:10.1109/ACCESS.2014.2325029
Chen, X., Xiang, S., Liu, C.-L., Pan, C.-H., Chen, X., Xiang, S., ... Pan, C.H. (2014). Vehicle detection in satellite images by hybrid deep convolutional neural networks. IEEE Geoscience and Remote Sensing Letters, 11(10), 1797–1801. doi:10.1109/LGRS.2014.2309695
Cheng, G., Han, J., & Xiaoqiang, L. (2017). Remote sensing image scene classification: Benchmark and state of the art. Proceedings of the IEEE, 105(10), 1865–1883. doi:10.1109/JPROC.2017.2675998
Cheng, G., Zhenpeng, L., Yao, X., Guo, L., & Wei, Z. (2017). Remote sensing image scene classification using bag of convolutional features. IEEE Geoscience and Remote Sensing Letters, 14(10), 1735–1739. doi:10.1109/LGRS.2017.2731997
Cheriyadat, A.M. (2014). Unsupervised feature learning for aerial scene classification. IEEE Transactions on Geoscience and Remote Sensing, 52(1), 439–451. doi:10.1109/TGRS.2013.2241444
Del, F.F., Liciardi, G., & Duca, R. (2009). Autoassociative neural networks for features reduction of hyperspectral data. In 2009 first workshop on hyperspectral image and signal processing: Evolution in remote sensing (pp. 1–4). Grenoble. IEEE. doi:10.1109/WISPERS.2009.5288997
Do, M.N., & Vetterli, M. (2003). The finite ridgelet transform for image representation. IEEE Transactions on Image Processing, 12(1), 16–28. doi:10.1109/TIP.2002.806252
Fang, L., & Shutao, L. (2010). An efficient dictionary learning algorithm for sparse representation. 2010 Chinese Conference on Pattern Recognition (CCPR), Chongqing, China, 1–5. doi:10.1109/CCPR.2010.5659325
Farabet, C., Couprie, C., Najman, L., & Yann, L. (2013). Learning hierarchical features for scene labeling. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), 1915–1929. doi:10.1109/TPAMI.2012.231
Feng, Q., Liu, J., & Gong, J. (2015). UAV remote sensing for urban vegetation mapping using random forest and texture analysis. Remote Sensing, 7(1), 1074–1094. doi:10.3390/rs70101074
Gómez-Candón, D., De Castro, A.I., & López-Granados, F. (2014). Assessing the accuracy of mosaics from Unmanned Aerial Vehicle (UAV) imagery for precision agriculture purposes in wheat. Precision Agriculture, 15 (1), 44–56. doi:10.1007/s11119-013-9335-4
Goodfellow, I., Lee, H., Le, Q.V., Saxe, A., & Ng, A.Y. (2009). Measuring invariances in deep networks. In Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K. I. Williams, & A. Culotta (Eds.), Proceeding NIPS’09, Proceedings of the 22nd International Conference on Neural Information Processing Systems, British Columbia, Canada, (pp. 646–654). Curran Associates, Inc.
Ham, J., Lee, D.D., Mika, S., & Bernhard, S. (2004). A kernel view of the dimensionality reduction of manifolds. In Twenty-First International Conference on Machine Learning – ICML ’04, Banff, Alberta, Canada. New York, New York, USA: ACM Press. doi:10.1145/1015330.1015417
Hoo-Chang Shin, M.R., Orton, D.J., Collins, S.J.D., & Leach, M.O. (2013). Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data. IEEE Transactions on...
