

Meta-analysis of deep neural networks in remote sensing: A comparative study of mono-temporal classification to support vector machines

Shahriar S. Heydari, Giorgos Mountrakis*

Department of Environmental Resources Engineering, State University of New York, College of Environmental Science and Forestry, 1 Forestry Drive, Syracuse, NY 13210, United States

ARTICLE INFO

Keywords:

Deep learning
Classification
Convolutional neural network
Deep belief network
Stacked auto encoder
Support vector machine

ABSTRACT

Deep learning methods have recently found widespread adoption for remote sensing tasks, particularly in image or pixel classification. Their flexibility and versatility has enabled researchers to propose many different designs to process remote sensing data in all spectral, spatial, and temporal dimensions. In most of the reported cases they surpass their non-deep rivals in overall classification accuracy. However, there is considerable diversity in implementation details in each case and a systematic quantitative comparison to non-deep classifiers does not exist. In this paper, we look at the major research papers that have studied deep learning image classifiers in recent years and undertake a meta-analysis on their performance compared to the most used non-deep rival, Support Vector Machine (SVM) classifiers. We focus on mono-temporal classification as the time-series image classification did not offer sufficient samples. Our work covered 103 manuscripts and included 92 cases that supported direct accuracy comparisons between deep learners and SVMs.

Our general findings are the following: (i) Deep networks have better performance than non-deep spectral SVM implementations, with Convolutional Neural Networks (CNNs) performing better than other deep learners. This advantage, however, diminishes when feeding SVM with richer features extracted from data (e.g. spatial filters). (ii) Transfer learning and fine-tuning on pre-trained CNNs are offering promising results over spectral or enhanced SVM, however these pre-trained networks are currently limited to RGB input data, therefore currently lack applicability in multi/hyperspectral data. (iii) There is no strong relationship between network complexity and accuracy gains over SVM; small to medium networks perform similarly to more complex networks. (iv) Contrary to the popular belief, there are numerous cases of high deep networks performance with training proportions of 10% or less.

Our study also indicates that the new generation of classifiers is often overperforming existing benchmark datasets, with accuracies surpassing 99%. There is a clear need for new benchmark dataset collections with diverse spectral, spatial and temporal resolutions and coverage that will enable us to study the design generalizations, challenge these new classifiers, and further advance remote sensing science. Our community could also benefit from a coordinated effort to create a large pre-trained network specifically designed for remote sensing images that users could later fine-tune and adjust to their study specifics.

1. Introduction

Artificial neural networks (ANNs) first started with cybernetics in the 1940–1960s and led to the invention of the first single neuron model named perceptron (Rosenblatt, 1958). Being a data-driven model with the ability to simulate arbitrary computing functions through optimization, ANNs found a wide range of applications. The next major breakthrough happened in late 80s with the invention of back-propagation and a gradient-based optimization algorithm to train a neural network with one or two hidden layers with any desired number of

nodes (Rumelhart et al., 1986). The back-propagation method has worked well for non-deep structures (1–2 hidden layers) but gradient-based training of deep neural networks (DNNs) could get stuck in local minima or plateaus due to the dramatic increase in number of model parameters and vanishing of gradients during backpropagation (Bengio, 2009). There is no standard definition to label a neural network as deep, but it mostly refers to network of two hidden layers or more, used to automatically extract a hierarchical set of features from data. Compared to 1–2 layer structures, DNNs promise to provide more compact models for the same modeling capabilities (Bengio, 2009). However,

* Corresponding author.

E-mail addresses: sshahhey@sy.edu (S.S. Heydari), gmountrakis@esf.edu (G. Mountrakis).

<https://doi.org/10.1016/j.isprsjprs.2019.04.016>

Received 1 February 2018; Received in revised form 17 April 2019; Accepted 24 April 2019

Available online 01 May 2019

0924-2716/ © 2019 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS). Published by Elsevier B.V. All rights reserved.

the high node number of DNNs made it difficult to train and optimize in a practical manner.

The seminal work of Hinton et al. (2006) showed that unsupervised pre-training of each layer, one after another, could considerably improve results. This layer-wise training approach, named greedy algorithm, was the key that opened new avenues to deep neural networks. The greedy algorithm could also be followed by a fine-tuning process, in which the entire network is tuned together using backpropagation, but this time from a much better starting point. Deep network theories and practices have expanded considerably during the last decade. It has resulted in establishment of some major network types (with continuous enhancements) and numerous applications in different domains. In close relationship with image processing and computer vision, remote sensing (RS) is one of many areas that deep learning is targeting.

Generally and following discussion in L. Zhang et al. (2016), we can categorize remote sensing applications of deep learning into four groups: (1) RS image pre-processing, (2) scene classification, (3) pixel-based classification and image segmentation, and (4) target detection. For image pre-processing tasks, we can name pan-sharpening, denoising, and resolution enhancement as major applications. Scene classification is done based on some extracted features from a scene, which the deep networks are assumed to be good at. The non-deep approaches normally use some handcrafted features extracted from the scene to feed the classifier (SVM, KNN, etc.) and predict the scene type. Deep networks have opened the door to direct use of spectral and spatial information together to generate a richer set of features automatically. This automatic extraction increases the potential for good generalization and scalability of this method compared to handcrafted features. Handcrafted features tend to be tailored closely to a specific case and application and possibly perform better than any automatic system, but because of this specificity they cannot be easily or successfully generalized to another cases/studies. This type of work is closely related to image recognition task but for categorization of remote sensing scenes (such as agricultural field, residential area, airport, parking lot, etc.), therefore sharing network configurations between computer vision and remote sensing applications is common here. Pixel classification and segmentation (or semantic labeling) are similar to scene classification but operate at the pixel rather than scene level, and produce a thematic map instead of a single category index. This is perhaps the most studied RS application and deep networks have shown performance benefits due to their ability to co-process spatial and spectral data easily, especially for hyperspectral images. Our main target in this paper is to focus on image or scene classification – we do not address other applications. In addition, we focus on mono-temporal classification as the time-series image classification is still in its infancy. Target or object detection is generally an extension to the three aforementioned groups, where specific objects defined by their shape or boundary are extracted from an image. This field has found many useful but challenging applications in high resolution and real time image/video processing.

Following the explosive growth of new algorithmic developments and case studies in deep learning RS applications in the past 3–4 years, several review manuscripts have been published (Ghamisi et al., 2017, Xia et al., 2017, L. Zhang et al., 2016, or P. Liu et al., 2017) and a Special Issue in this journal has summarized several recent advancements (Mountrakis et al., 2018). The majority of these reviews are descriptive and do not offer a quantitative assessment of deep learning benefits building on the extensive available comparisons in the literature. The overall goal of this work is to bridge this knowledge gap by undertaking a meta-analysis comparing deep and non-deep classification algorithms through a meta-analysis of published research.

Other meta-analysis works exist but they do not examine explicitly

deep learning benefits. For example, Khatami et al. (2016) grouped all neural network types under one category and did not distinguish deep networks from other implementations. Ma et al. (2017) conducted similar meta-analysis focusing on object-based classification (thus excluding pixel-based ones) without separating deep learning methods. There are some other papers that review deep learning architectures in general such as Deng (2014) or W. Liu et al. (2017), or for specific type of data, such as Camps-Valls et al. (2014) on hyperspectral data classification. These works also lack quantitative comparisons using a meta-analysis approach.

The overarching goal is to provide readers with the “big picture” of current research and build on the collective knowledge of published works to assess deep learning benefits in remote sensing. To undertake the proposed meta-analysis task, we reviewed major research papers and built a database of case studies of deep network applications in the remote sensing field while extracting main network and data characteristics. This database was analyzed to identify deep learning classification performance and its distribution across these network (e.g. network complexity) and data characteristics (e.g. spatial resolution). We expect this analysis to provide a knowledge baseline as the remote sensing community further incorporates deep learning in related activities.

The structure of the manuscript is as follows. A brief overlook of deep network types is presented in Section 2 along with key introductory references. A summary table is also provided to describe extracted parameters for each research paper. In Section 3 after introducing a descriptive statistics and summarizing design ideas encountered in the selected research papers and used datasets, we provide our main comparative analysis and discuss important research questions about parameters effect on network performance. The last section provides concluding remarks.

2. Methods

In this section we first describe three DNN methods that have been popular in RS tasks. Section 2.2 contains an explanation on the paper database and associated characteristics and metrics used in the comparative accuracy analysis between DNNs and non-deep methods.

2.1. Summary of popular deep neural networks in remote sensing

The deep learning paradigm is concentrated on automated hierarchical feature extraction. Numerous methods and their modifications have been devised along the past years. Here we briefly introduce the three most widely used structures which were used in our identified studies. More detailed descriptions of each structure can be found in many machine learning textbooks, for example Bengio (2009) and Goodfellow et al. (2016), or tutorials such as Le (2015) and Deng (2014). Zhu et al. (2017) and L. Zhang et al. (2016) also provide tutorials for deep learning for remote sensing (RS) applications.

Deep networks have been developed to enhance and enrich data representations in an automated and intelligent manner. A good representation is, of course, dependent on the specific application and should be learned from training data. One important deep network category in this class is based on Autoencoders (AEs). The idea behind an autoencoder is basically an encoder-decoder network to regenerate the input as accurately as possible in its output. Under specific conditions, the encoder part works as a good feature extractor and can be stacked to build deep networks named Stacked Auto Encoders or SAEs (the decoder part is not used). The imposed condition on objective function is typically a form of sparsity, but other variants are also studied. To put it simply, AE learns a deterministic representation of the

Table 1
Parameters collected on each case study.

Reference	Citation code for the referenced research paper
Network type	One of below categories: – Convolutional Neural Network (CNN), – Deep Belief Network (DBN), – Stacked AutoEncoder (SAE)
Learning strategy	One of below categories: – Unsupervised – Unsupervised & fine-tuning – Semisupervised – (fully) Supervised – Transfer learning – Transfer learning & fine-tuning
Number of parameters	Number of trainable network parameters, i.e. weights and biases of network neurons and connections. We manually created this number to approximate network complexity
Dataset	Name of dataset used for the research, including: – Brazilian coffee, NWPU-RESISC45, RSSCN7, UC Merced, and WHU-RS19: 3-band images used in scene classification, – Indian Pines, Houston, Kennedy Space Center, Pavia University, Pavia City Center, and Salinas: hyperspectral images used in pixel classification, – ISPRS Potsdam and ISPRS Vaihingen: very high resolution images used in image segmentation, – Others: Remaining datasets.
Spatial resolution	Dataset spatial resolution expressed through pixel size
# of channels	Number of spectral and auxiliary channels
Training proportion	Proportion of training data size in reference dataset
Metric type	Metric used for reporting performance in research case, including Overall Accuracy, Average Accuracy, Average Precision, F1, Kappa, etc
Deep network result	Best reported value of network classification performance
SVM results	Best achieved performance of SVM implementation

input by minimizing a cost function based on the difference between input and the regenerated one at the decoder output. This learning takes place using gradient-based optimization and standard back-propagation techniques. AEs are well suited to unsupervised learning and can be trained layer-wise, possibly followed by a supervised fine-tuning phase of the entire network. For a good overview of autoencoders with some work examples and executable codes see Andrew Ng's Deep Learning tutorial at http://deeplearning.stanford.edu/wiki/index.php/UFLDL_Tutorial. Vincent et al. (2010) also provide more details on autoencoders and unsupervised learning.

Another way of thinking about data representation is to learn the statistical distribution of input, i.e. a probabilistic approach. This approach has led to Generative models or Structured Probabilistic Models. Deep belief network (DBN) based on stacking layers of Restricted Boltzmann Machine (RBM) is the most popular variant for RS applications. Here the aim is to minimize the Boltzmann cost function, to maximize “the similarity (in a probabilistic sense) between the representation and projection of the input” (Singhal et al., 2016). This optimization does not use an assumed output, so a different algorithm (contrastive divergence) is required to train the neurons. However, similarly to the autoencoder, training is unsupervised and, more important, it can be done in a greedy layer-wise approach for a stack of layers. This layer-wise approach was devised by the seminal work of Hinton et al. (2006) and later implemented by both SAEs and DBNs. Therefore SAEs and DBNs are often discussed together in the literature (e.g. Vincent et al. (2010)). When trained, the network can provide extracted features for the new data to be classified. Tutorials on RBM and DBN are available through the internet, for example see <https://deeplearning4j.org/restrictedboltzmannmachine>, which includes executable codes.

The third type, which is the most used structure in recent years, is the convolutional neural network (CNN). Inspired by the human visual system and designed to process images, it has limited connection to only adjacent neurons in each layer, with the same connection weights for each neuron within each layer. It may include down-sampling in each layer, which reduces the processing resolution but adds translation

invariance property to the network. Each layer's output is typically named a map and it is generally desired to have multiple maps generated at each layer. Here the filter weights are tuned typically by supervised training, as the limited number of shared parameters in each layer (compared to a fully connected network) allows it. There are also some pre-trained large network structures publicly available for use and fine-tuning them for specific applications is another common approach. For a university course on convolutional neural networks readers are referred to <http://cs231n.stanford.edu/>. Zeiler and Fergus (2014) also provide a discussion on visualization and understanding of the internal CNN workings.

Working with sequence data is another important type of remote sensing works, particularly on three bases: studying hyperspectral signal variations and analyzing their dependencies; adding the time dimension as another data element to explore land use feature patterns (profiles) and use them in classification; and pursuing detection of changes in land cover or land use by processing time-series data. Neural networks – and specifically Recurrent Neural Networks – are gaining momentum for these applications but the number of published papers is still low. These networks are promising with new modifications such as adding more powerful and deep memory cells (see for example Lyu et al. (2016), Mou et al. (2017), Rußwurm and Körner (2017), Rußwurm and Körner (2018), Ndikumana et al. (2018), Niculescu et al. (2018), or Sharma et al. (2018)). However, we did not consider sequence data applications in our paper due to lack of enough data and our focus was only on feed-forward networks and its three main variants: SAE, DBN, and CNN.

2.2. Comparative performance database creation

Our overarching goal is to look at the analyzed DNN case studies and compare them together and to a well-known non-deep classifier, Support Vector Machine (SVM). SVM will serve both as a representative for non-deep classifier to compare with deep networks, and as a baseline to compare different DNN architectures. SVMs were selected as the benchmarking algorithm because: (i) they were found to be the best

non-deep performing classifiers in an extensive comparison of published work (Khatami et al., 2016), and (ii) the majority of DNN papers found in this review chose to include SVM as the main benchmark, thus validating our decision. We also examine accuracy trends across data and method characteristics. Direct comparisons of published works are not feasible due to variances in data types, sampling design, algorithmic details, and test metrics. Therefore, we concentrated on aggregating results from manuscripts where accuracy metrics are reported mutually under common conditions for deep and non-deep implementations. This database was then used to do comparative meta-analysis and other quantitative statistical analyses.

The result was 103 research papers from 2014 until Nov. 2018 covering 183 case studies that include deep learning-based classification, 92 cases of which supported direct comparisons of accuracy to SVM. The main characteristics of these case studies are summarized in Table A1, Appendix A, with each column of the appendix table defined in Table 1 below. These parameters reflect the most important aspects of the research design and we use them to present the discussion of our research questions in the subsequent sections. We treat each data set in a research paper as a separate case, because the output result and possibly the network structure may vary per case in any single paper.

One of the most important parameters in network specification is the number of network parameters which reflects network complexity. This is typically a surrogate of network depth and width. It is expected that a bigger network would be more powerful, but the network architecture and way of processing (reflected in other columns of the table) greatly impacts this performance. Therefore, it is not unexpected that a smaller but more elegant network outperforms a larger one in obtained accuracy. For example, in classifying the ISPRS Potsdam and Vaihingen datasets, Maggiori et al. (2016) achieved > 1% better accuracy than Volpi and Tuia (2017) by a network having around 1/10th of their network size. This number is mostly calculated from network parameters given in the cited paper but in some cases it is given in the cited paper as well. In cases that given information was not sufficient or ambiguity was not cleared by correspondence, the entry was left blank. This number includes parameters in as many network branches as implemented, but it does not include parameters associated with additional stages of combination or fusion with other data or algorithms. It also counts the network layers parameters up to the last layer before the final classifier, which is typically a Softmax layer but SVM is also used. In around 70% of our cases the deep network is followed by a Softmax classifier, therefore we drop the final classifier type from our list of parameters.

The learning strategy column is another important network parameter. It does not point to the final classifier training as it is always supervised, but shows the methodology for determining network parameters. The supervised learning is the most common approach in deep networks. It can also have different variations in the form of cost function or optimization procedure, or being enhanced by data-driven techniques such as active learning. Those advanced cases are designated as supervised + in our database. The fine-tuning options show the cases when network parameters are fine-tuned after an initial unsupervised learning or transferred from a pre-trained network in transfer learning. Transfer learning is available to CNN only. DBNs are usually limited to unsupervised & fine-tuning type, while SAEs are used with both unsupervised learning techniques. Semi-supervised learning is also used in some cases, which is a strategy for using both labeled and unlabeled data in optimizing the network cost function.

Spatial resolution in our collected research cases varies from 5 cm for VHR imagery to 30 m for Landsat, left blank if not provided. The number of channels shows the ones that have been actually used in the experiment (some channels have been set aside for their low quality in

some studies but not in the others). Note that in some cases the input channels are processed and dimensionality was reduced (mostly employing PCA) and the result is applied to the network, but we do not mention this dimensionality reduction in the table, although we take it into consideration when calculating the number of parameters and consider the network in its actual tested configuration. There are two cases of using Landsat and one case of MODIS imagery that has been indicated in table separately due to importance of these data sources. Although from one hand they are of less attention today because of their inferior spatial resolution, but from the other hand they are of interest for their rich temporal dimension in time-series analysis. Data fusion from different sources is also experiencing growing attention, especially adding height data through digital elevation models (DEM). We discuss this further in the design options (Section 3.3) but an in-depth analysis of this issue was outside the scope of this work.

Another important factor in network prediction performance is the data training size. More training data typically leads to better network generalization, but in many cases the labeled training data is very limited. The corresponding column shows the rounded proportion of (labeled) training data samples to the entire reference data set, varying from as low as 0.1% to 90%. We refer to it as “training proportion” hereafter, and consider the proportion in one single run of the network, therefore a cross-validation scheme does not change the value in the table from a similar hold-out.

The reported classification accuracy (overall or average) value is the best performance reported for the reference dataset in each case. It is reported as a number between 0 and 100 except for the metric Average Normalized Modified Retrieval Rank (ANMRR). Although overall accuracy is an aggregate metric and cannot show important class-dependent performance values, but it is still the most widely used metric due to its simplicity and general applicability. Even though in some cases more detailed evaluations are provided along with overall accuracy, due to different experimental designs and data structures in our meta-analysis, these detailed metrics were not widely comparable and therefore class-specific measures were not included.

In some cases, an additional pre- or post-processing step complements the deep network to enhance the performance, for example merging the resulting map with an auxiliary segmentation result, adding a conditional random field (CRF) layer for edge enhancement, or object-based processing. These methods differ largely in implementation details and experiment setup so cannot be directly compared to assess the processing gains; we provide more details on them in Section 3.3.

Although the chosen non-deep methods vary greatly in type and options from paper to paper, there are still numerous cases where DNNs are compared to an SVM-based implementation, with Random Forest and KNN being the next classifier types used by much less frequency in our observed cases. Therefore we chose those papers reporting on SVM results as the candidates for doing our quantitative analysis (in the next section). SVM is a good choice for benchmarking because it is a well-established and proven classification tool with generally superior performance (Mountrakis et al., 2011; Khatami et al., 2016; Heydari and Mountrakis, 2018). Note that in remote sensing image or scene classification tasks, we are generally interested in both feature generation and classification. Neural networks can do both automatically – and deep networks put more stress on the feature extraction task – but SVM classifiers should be fed with features already generated by another algorithm. The SVM implementation itself may vary between processing the raw pixels data or some secondary handcrafted spectral/spatial features derived from data. To ensure a more fair comparison we separated these two cases due to the potential important impact of working with features instead of raw data. Clearly, there are many

Table 2
Basic statistics of collected case studies.

Year	2014	2015	2016	2017	2018
Number of cases	4	27	21	22	33
Network type	CNN		DBN		SAE
Number of cases	150		9		25
Dataset spatial resolution	< 30 cm		30 cm – 3 m		> 3 m
Number of cases	23		88		48
Spectral and auxiliary bands	1–3		4–10		11–99
Number of cases	59		48		70

variations and methods for handcrafting features and each paper may include a different set of methods for comparison, so we could not go into their implementations detail and a detailed comparison. Furthermore, SVM optimization methods varied (e.g. hyperparameters and kernel choice), however we assumed (and it was also stressed in some papers) that after tuning the authors reported their best SVM performance.

We should mention here that although our meta-analysis covers many different cases, each case has almost a unique setting of the above parameters and therefore our analysis is naturally limited in depth and statistical richness. Our objective was to study general trends and for the first time in the literature offer a quantitative meta-analysis of DNNs in remote sensing applications. Our quantitative analysis did not go into a detailed analysis of the effect of every design option due to lack of data.

3. Results and discussion

3.1. Descriptive statistics

Table 2 provides information on case studies distribution by year, network type, spatial resolution and input dimensionality. Note that some manuscripts may contain more than one study, and spatial or input dimensionality information was not always available.

There is an increase in research papers on deep networks for remote sensing classification applications after 2014, continuing to date. CNN is the most commonly used network type, then SAE followed by DBN. Most of the datasets are either hyperspectral (> 100 spectral channels) or less than 10 channels. Just one case study had spectral channels between 10 and 100. Hyperspectral dataset are of high spatial resolution (around 1 m) so sit in the middle group of spatial resolution category. Very high resolution ones (< 30 cm) are mostly available in RGB with possibly adding Near-Infrared band and/or DSM data to it, with just one very recent case incorporating a drone-based six band experiment at spatial resolution of 4.7 cm. More information on datasets will be given in the next section.

3.2. Datasets in the selected case studies

A wide variety of approximately 60 different datasets were used throughout the selected case studies. They included frequently used datasets along with datasets selected from public sources such as Google Earth, QuickBird, WorldView, Landsat archives and proprietary data sources. Cases that have been used more than twice in our review are listed in Table 3. The table includes both scene and pixel

classification applications as indicated in the last column, and the “labelled elements” column should be interpreted accordingly. Among them, the Brazilian Coffee, NWPU-RESISC45, RSSCN7, UC Merced, and WHU-RS19 have been used for scene classification while the others concentrated on pixel classification/image segmentation. It is important to note a significant limitation. While still being extensively used even in papers from 2018, some of the commonly used datasets are old and outdated: the major issue is their small size compared to datasets with millions of elements typically used in computer vision and other artificial intelligence studies. This issue has been partly addressed by some very high resolution datasets such as ISPRS Vaihingen and Potsdam datasets, which became a standard test bench for newly arrived (mostly CNN-based) networks. Furthermore, hyperspectral cases are limited to a single scene and some datasets cover a very small geographic area, which limits the generalization ability of the obtained results. Again, there is a new dataset presented through IEEE GRSS contest in 2018 which consists of a relatively large area of 1.4 km² covered by both very high resolution (5 cm) RGB and high resolution (1 m) multispectral data. However, none of our reviewed articles was based on this new dataset (Le Saux et al., 2018).

There is still a need to create more large and rich datasets for remote sensing applications in different spatial and spectral resolutions. Preparing datasets for tackling temporal applications is another important issue, which is even more restricted than other applications. However, the decision to pick specific labels and the procedure for creating ground truth maps is very application-specific. Provision of auxiliary data (commonly DSM based on LiDAR) is also an important enhancement that is available in few datasets and should be encouraged.

3.3. Network design options

In terms of network optimization for deep networks the simplest way is to change the network depth (number of layers) and width (neurons per layer). Additional modifications include changes in the activation function, the type of classifier or the training strategy (supervised/unsupervised). Looking beyond these fairly common adjustments, we present in Table 3 a descriptive summary of the most important design innovations we encountered. The table is organized to titles summarizing the main design point, followed by specific design ideas in each section. The number of papers using each option is provided to suggest popularity. Some design options are not exclusive to a specific network type (e.g. network mixing options), while others are only applicable to specific network types (e.g. fully convolutional network). The classification task type may also require special provisions.

Table 3
Specifications for most frequently used datasets.

Dataset name	Sensor platform	Dataset size	Image size (pixels)	Labelled elements	Spatial res.	# of spectral and aux. channels	# of classes/classification task
Brazilian coffee	SPOT	50,000	64 × 64	50,000 scenes		RG + NI	3 class, but highly imbalanced/scene
Houston (2013 GRSS)	ITRES-CASI	1	1905 × 349	15,029	2.5 m	144 + LiDAR	15 class/pixel
Indian pines	AVIRIS	1	145 × 145	10,249	20 m	220	16 class/pixel
ISPRS Potsdam	Aerial photo	38	6000 × 6000	24 full images (of 38)	5 cm	RG + NI + DSM	6 class/pixel
ISPRS Vaihingen	Aerial photo	33	circa 2500 × 2000	16 full images (of 33)	9 cm	RG + NI + DSM	6 class/pixel
KSC (Kennedy Space Center)	AVIRIS	1	512 × 614	5211	18 m	224	13 class/pixel
NWPU-RESISC45	Google Earth images	31,500	256 × 256	31,500 scenes	0.2–30 m	RGB	45 class, 700 samples per class/scene
Pavia Center	ROSIS	1	512 × 614	148,152	1.3 m	103	9 class/pixel
Pavia University	ROSIS	1	610 × 340	42,776	1.3 m	103	9 class/pixel
RSCN7	Google Earth images	2800	400 × 400	2800 scenes		RGB	7 class, 400 samples per class/scene
Salinas	AVIRIS	1	512 × 217	5348	3.7 m	224	16 class/pixel
UC Merced	USGS satellite imagery	2100	256 × 256	2100 scenes	1 ft	RGB	21 class, 100 samples per class/scene
WHU-RS19	Google Earth images	950	600 × 600	950 scenes	0.5 m	RGB	19 class, 50 samples per class/scene

Table 4

Network options and design innovations in collected papers.

Option	Frequency
<i>Making dense (full resolution) output options (for CNN):</i>	
Fully Convolutional Network (convolution and deconvolution)	13
No down-sampling	1
<i>Multiscale capability options:</i>	
Getting multiscale input	10
Using multiscale kernels (filters)	7
Skip links (forwarding features/scores from one layer to another non-adjacent layer)	8
<i>Network mixing options (fusion/aggregation method varies by case):</i>	
Parallel handcrafted features	3
Parallel 1-D and 2-D convolutional networks	3
Parallel networks on different band combinations or sensors	9
Cascaded networks	3
Parallel (Ensemble) of different deep networks	10
<i>Training options:</i>	
Salient patches selection to train/test network	2
Active learning or iterative feature selection (removing inferior features)	4
Data augmentation or adding virtual samples to the input data	15
Other specialized methods	9
<i>Multimodal processing:</i>	
3-D processing modules	7
Spatial averaging/filtering over a neighborhood for spectral + spatial input generation	2
PCA dimensionality reduction and spectral + spatial input generation	9
Sequence data/temporal processing	4
Other specialized methods	2
<i>Other features:</i>	
Feeding network with handcrafted features (not raw data)	4
Optimizing input band selection with genetic algorithms	1
MRF/CRF processing or boundary detection	7
Denoising SAE implementation	3
Initial and/or final data/feature filtering or segmentation to enhance object discrimination	12
Sparse or other type of coding to create codebook after feature generation and classify the code	4
Emerging network modules (e.g. residual module, inception module, LSTM)	10

For example, in image segmentation the objects' boundary alignment is of primary concern, while in scene classification this is not important. This makes edge enhancement techniques more relevant to the former application than the latter. As each design idea is presented and tested in a unique setting with single or multiple choices of listed options on different datasets and compared with different non-deep rivals, comparison between different design ideas and quantitative analysis of their merit is not possible. However, we discuss general findings on design options below and our intent is that this table will act as a preliminary catalog and guide future research, either through gap analysis or through frequently-implemented method identification.

3.3.1. Dense (fully connected) networks

This CNN-type network is the de-facto network of choice for very high resolution classification and almost all of the image segmentation works – particularly experiments with ISPRS Potsdam and Vaihingen datasets. The competition in this field is extensive, and some of the most popular networks have been implemented in this category to win the ISPRS competition. It is always possible to run the entire network and classify the image pixel by pixel, but it means a huge redundancy in calculations and therefore a direct map-to-map conversion (which typically contains chain of downsampling and then upsampling) is preferred. Upsampling design is a hot topic and each paper tries to find a better way to do it. Edge enhancement and additional segmentation techniques have also been examined by different approaches to enhance the result (we will refer to it in another paragraph in this section).

3.3.2. Multiscale capability options

This issue is of a particular interest in CNN networks due to the limited connectivity of their neurons to the previous layer, but other network types may also use it when they use a sliding window mechanism in their input layer. In custom CNN networks the multiscale filters with or without skip links (forwarding features/scores from one layer to another non-adjacent layer) is a promising choice, but this option is not typically available for pre-trained networks.

3.3.3. Network mixing options

There is a variety of practices for this option as listed by categories in Table 4. The most frequent option is ensemble of different networks or using a parallel network on different bands (especially when the additional input in form of DSM or LiDAR is provided). Parallel 1-D (spectral) and 2-D (spatial) network is also found in some cases, but other forms of spectral/spatial input combination are more frequent (we discuss it in a later paragraph). As the use of pretrained networks becomes more common, parallel networks are the natural way of overcoming the imported network input limitation to RGB channels.

3.3.4. Training options

Engineering the input data is the most frequent form of enhancing training operations in deep networks, which is implemented in a variety of methods. The simplest case is to crop, rotate and flip the input patches (basic data augmentation) or adding virtual samples to the input data (particularly used for making input set more balanced). Recently, active learning and interactive sample selection approaches are gaining more attention. There are other specially designed algorithms used to enhance data quality, such as salient patch selection; or specialized methods for calculating network parameters, such as calculation of neurons weights by clustering or PCA decomposition instead of training.

3.3.5. Multimodal processing

Deep networks in remote sensing classification started with processing spectral components but quickly evolved to process other dimensions of data as well. Data processing in spatial context is now typical, especially with CNN, and joint spectral-spatial processing in 3-D convolutional filters is popular. Before that, other techniques such as averaging over spatial dimension or PCA compression of spectral dimension were common, but newer 3-D architectures has shown slightly better performance in our case studies. The newest trend in multimode processing is to incorporate sequence/temporal processing, for example by treating spectral component of hyperspectral imagery as a (correlated) sequence, or working on time-series of spectral-spatial data cubes.

3.3.6. Other features

In addition to the above, we identified numerous special algorithms and techniques throughout our survey that are organized in this section. In earlier studies we saw some cases of performance improvement by feeding network with handcrafted features, but it seems to be an obsolete idea now. Object-based classification, image segmentation, and additional MRF/CRF processing have been attractive research areas from the early days and still draw a lot of attention. Parallel to that, developing and applying newer and more complicated network modules (for example residual modules in CNN or LSTM in RNN) in RS applications are increasing trends. In the reviewed cases, newly emerging modules seem to have the upper hand at the expense of much larger and more complicated networks. The other options found are:

- CRF postprocessing of deep network predictions to delineate and enhance object edges.

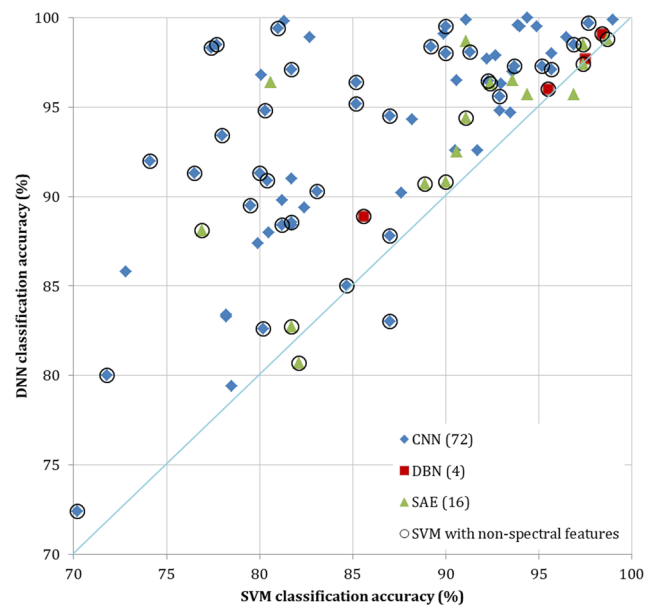


Fig. 1. Comparative performance distribution of DNN vs SVM.

- Initial segmentation and creation of superpixels to feed deep network.
- Merging of deep network predicted map and segmented or CRF/MRF generated map based on network prediction confidence.
- Other pre/post processing methods (e.g. GLCM/gabor filters).

There is no dominant method among the aforementioned techniques and new methods are continuously emerging.

3.4. DNN vs. SVM classification accuracy comparisons

This section focuses on classification accuracy comparisons between DNN and SVM methods. We focused on SVM comparisons since the majority of the manuscripts we reviewed selected SVM as their benchmark. The SVM choice over other methods (e.g. RF) is further supported by a previously conducted meta-analysis, where SVM was found to outperform other (non-deep) methods (Khatami et al., 2016). A detailed table summarizing each study is available in Appendix A. For a case study to be considered both methods were tested on the same dataset and results were reported in the form of average or overall accuracy.

Deep networks are usually designed to employ high volumes of available spectral and spatial data. However, in many of the selected cases of pixel classification, the authors compare DNNs to simple spectral processing by SVM or other non-deep rivals, thus providing an unfair advantage to DNNs as they also incorporate spatial information. Knowledge of feature generation details may not be a primary concern in deep networks as it is optimized automatically by the network, but finding the best method for feature generation to feed an SVM is not a straightforward task that often requires trial and error for each dataset. On the other hand, designing the best deep networks out of standard basic schemes is not a trivial issue and we see new designs continuously arising. To further inform readers in all figures in this section, we marked the cases with enhanced feature generation for non-deep classifier (SVM) with a black circle to separate them from cases using exclusively spectral information in their SVM implementation. Initial summary results are depicted in Fig. 1. Also as there were just two cases with SVM accuracy below 70%, we set our scale to start from that and

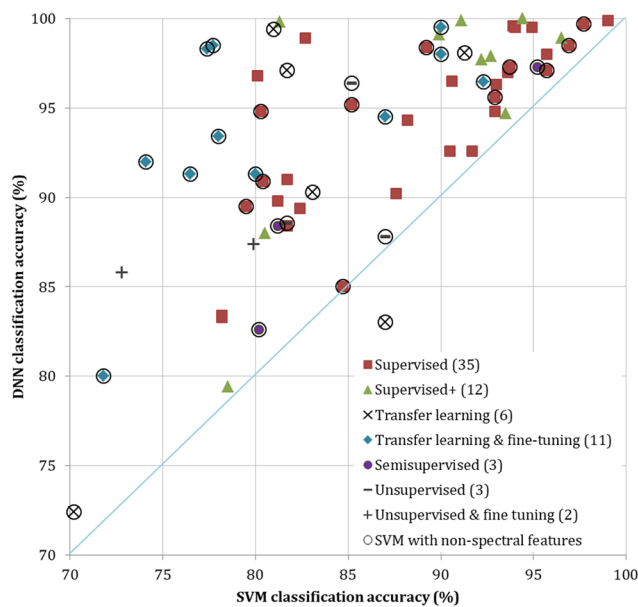


Fig. 2. Comparative performance distribution across learning methods for CNN.

omitted those two in display to reduce the congestion on the upper accuracy values in the provided figures.

In general, deep learning approaches offer consistently better results than SVM methods, even when only the cases with enhanced (non-spectral) features are considered. The reported improvement (difference in accuracy value) can be as high as 30% for CNN, 16% for SAE and 3% for DBN. The DBN values may not be very representative due to the scarcity of this network type application in remote sensing, but this lower application rate itself can be a sign of its lack of merit and/or underlying complexity. CNN accuracy benefits are often attributed to the integrated processing of spatial and spectral information, while for SAE or DBN benefits involve specific experimental design. As an

example, one author used the average values of a neighborhood around each pixel (to be classified) for each band and added it to the central pixel's own data, then fed the SAE or DBN with this composite data vector. It is also common in hyperspectral image classification to process input image with dimensionality reduction techniques such as PCA first, and then build the spatial information to be added to the original central pixel for classification.

CNN also has the ability to preserve the spatial relationships while processing through different layers, as spatial filtering takes place in each layer without flattening data to a row vector. In SAE or DBN implementations, the spatial information is flattened and concatenated to the spectral data at the network input, and although the spatial information is implicitly included, the spatial relations between vector values are lost (Y. Li et al., 2017; Yue et al., 2015; Basu et al., 2015). However, the CNN spatial coverage is limited to a neighborhood of fixed size at the input, increasing step by step while resolution is reduced accordingly in pooling layers. This issue is restrictive to scale-dependent information, although it can be remedied to some extent by a multiscale structure (for example see X. Chen et al., 2014, Zhao et al., 2015, or Zhao and Du, 2016). Other networks are not inherently limited by these rules, though the strength of spatial relationships is generally reduced with the increasing distance from the central point according to Tobler's law in geography (Tobler, 1970). It is also important to consider that the SAE and DBN methods are trained in an unsupervised fashion while the CNN method follows a supervised approach. Therefore, the CNN implementation may be advantageous due to the incorporation of labeling information (Y. Li et al., 2017; Shi and Pun, 2018). SAE and DBN are also trained in a greedy layer-wise fashion that may limit potential learning opportunities; each layer's parameters are fixed when tuning the next layer. Joint training of layers for SAE and DBN has been proposed in Zhou et al. (2014) and reported to perform better than typical greedy layer-wise approach, but it is not of common use.

To investigate further DNN accuracy gains, we examined their distribution across five contributing factors, namely the DNN learning method, the network complexity, spatial resolution, input dimensionality and training dataset proportion. Due to the low number of case studies and variation of design parameters and datasets employed in different studies we do not report a multivariable regression model. Instead, we limit our analysis to single factor distribution plots.

3.4.1. Distribution across learning methods

Figs. 2 and 3 present accuracy comparisons for different learning methods for CNN and SAE, respectively. DBNs have a single learning option therefore they are omitted from this analysis. Starting with Fig. 2 and CNN methods, the majority of cases have used supervised training or its enhanced version shown as supervised+ (cases with different cost function or optimization procedure, or being enhanced by data-driven techniques such as active learning). CNNs using supervised learning is mostly compared to spectral SVM and tend to offer higher relative gains in more complex classifications, where the corresponding SVM accuracy is lower. This result is expected due to integration of spatial data in CNN and lack of it in spectral SVM. As mentioned in C. Zhao et al. (2017), W. Zhao et al. (2017), there are similarities between low-level features in different classes that cannot be resolved solely by the spectral components and integration of spatial data is required (an example is the road and building roof pixels in an aerial image). As seen in the upper right corner of the graph, in cases where the SVM was fed with enhanced features, the performance is fairly close to the supervised learning DNN cases. One benefit of deep networks is the flexibility to build the features automatically and match them to the specific dataset under study, contrary to handcrafted features that

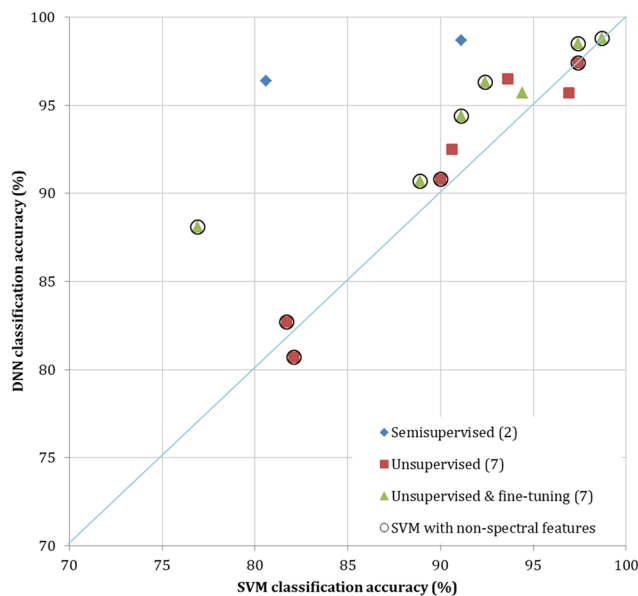


Fig. 3. Comparative performance distribution across learning methods for SAE.

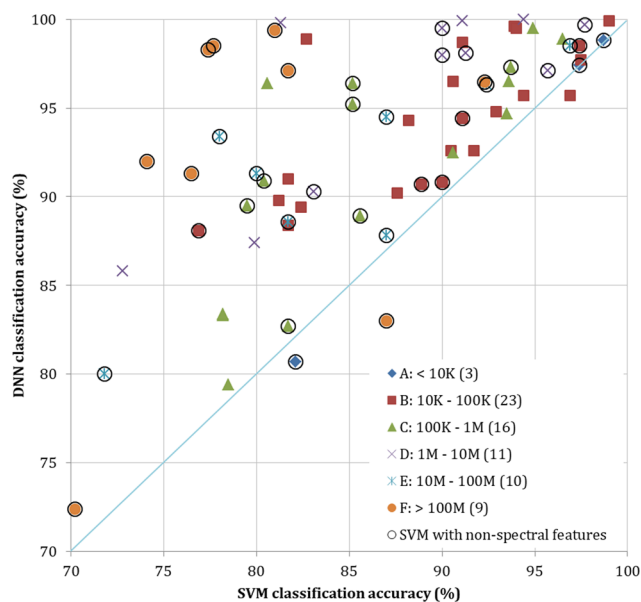


Fig. 4. Comparative performance distribution across network complexity.

should be selected among many variants for SVM or other non-deep classifiers.

Transfer learning in CNN also offers some improvements visible in Fig. 2 – and especially when combined with fine-tuning – over enhanced SVM. Base networks are typically taken from models developed and trained in computer vision industry and will not be introduced here. The most widely used model in our reviewed cases as AlexNet (9 cases), followed by VGG-16 (8 cases), VGG-M (5 cases), and GoogLeNet (4 cases). Other networks have also been applied with lower frequency such as ResNet, other VGG-series networks, SegNet, Overfeat, and CaffeNet. Their improved performance over supervised learning CNN cases could be attributed to the fact that supervised CNN cases are usually custom designed and small in size compared to CNN networks used for transfer learning, therefore they may not be much more powerful than a SVM fed with enhanced features. A large fully supervised network may achieve considerable improvement over an enhanced SVM, but the computational budget might be prohibitive and the risk of overfitting high. Transfer learning, however, uses a proven network architecture that is pre-set using an extensive collection of labelled image data, and reduces user involvement into network design issues. Successful application of this technique suggests that the features generated by those large image collections have a good generalization capability and can be matched to arbitrary datasets assuming a fine-tuning step. Comparisons of different possibilities for feature extraction, supervised training and transfer learning (with or without fine tuning) for selected CNN architectures are described in detail in Nogueira et al. (2017) and tested for three well-known scene classification datasets. They suggest to use transfer learning and fine-tuning instead of fully supervised training because the pre-trained networks start from a better initialization state in the search space. A significant limitation though is that pre-trained networks are not currently applicable in multispectral/hyperspectral classification tasks because existing pre-trained networks come from computer vision – trained on ImageNet – using RGB images. However, as studied in Huang et al. (2018), we can mix a big pre-trained network fed by the RGB portion of spectrum with a smaller deep network capable of mining the entire

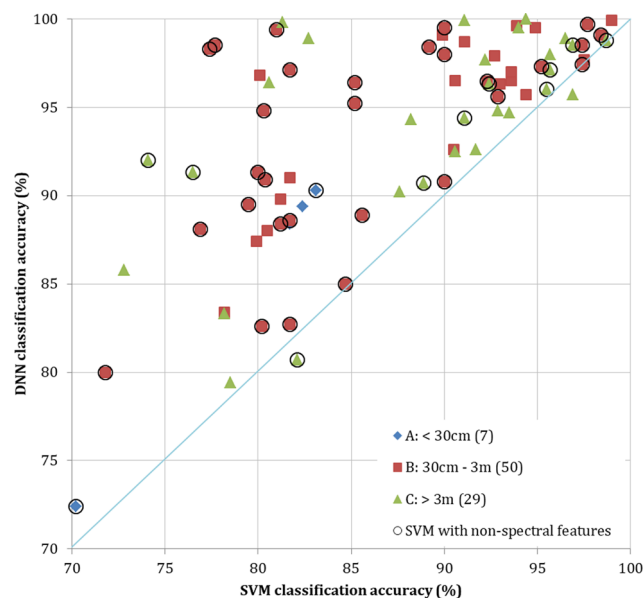


Fig. 5. Comparative performance distribution across spatial resolution.

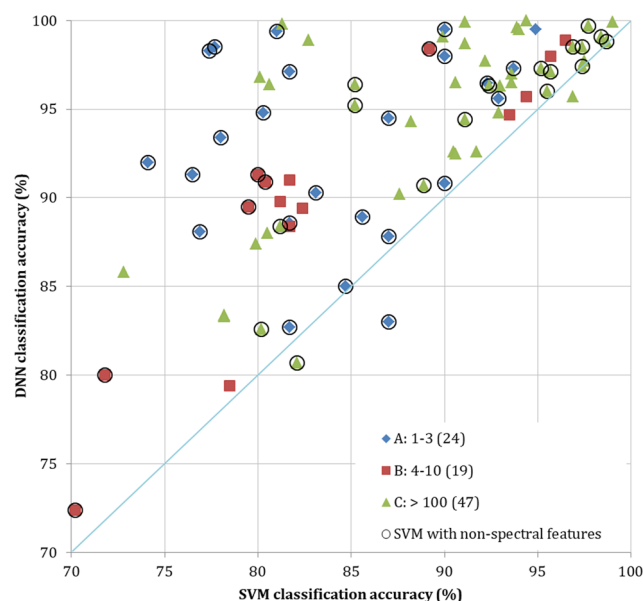


Fig. 6. Comparative performance distribution across input data dimensionality.

spectrum and obtain good results. Such a combination can also be run on limited number of input samples as the large network is pretrained. For the other options of semisupervised and unsupervised learning we can see limited improvement but there are not enough samples for conclusive results.

Looking at Fig. 3 and the SAE learning methods, fine-tuning of unsupervised methods tends to offer some gains over enhanced SVM, while there is no gain without fine-tuning. An explanation could be that unsupervised learning receives its strength from using much more data (labeled or unlabeled), so the features may be more representative of

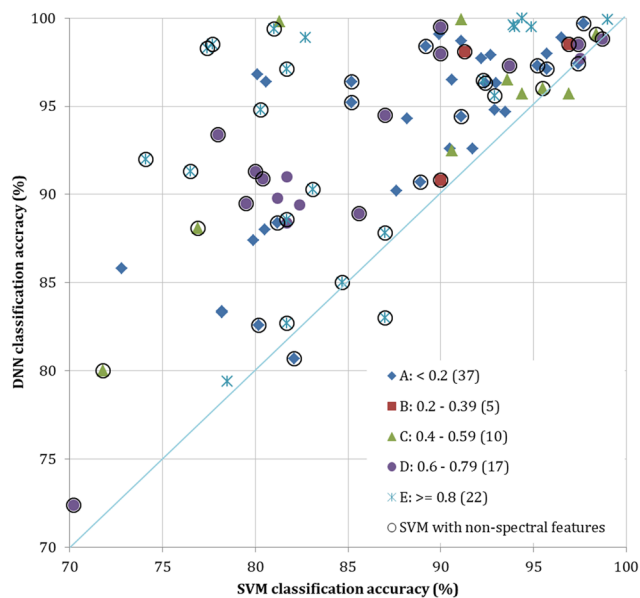


Fig. 7. Comparative performance distribution across training proportion.

the data. However, matching them to classes requires an extra step of supervised learning. Therefore, unsupervised learning alone is comparable to enhanced SVM, and fine-tuning further improves results. Semi-supervised learning was used in two cases with better results, but its application detail is case-dependent. There are different methods and also underlying assumptions about actual class distribution for doing semi-supervised learning (for example see [Zhu and Goldberg \(2009\)](#) and [Camps-Valls et al. \(2014\)](#)). Each method and assumption is embedding a specific additional regularization term for unlabeled data in the optimization cost function but there is no standardized way of doing that. This lack of standardization may be a cause for its limited use.

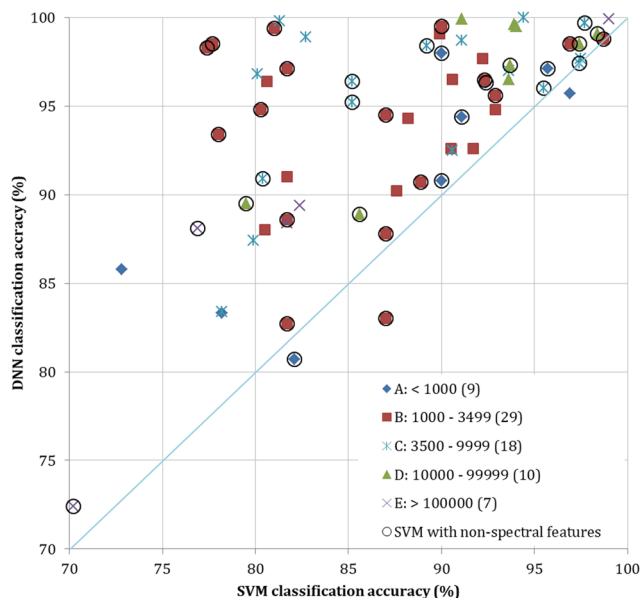


Fig. 8. Comparative performance distribution across training data size.

3.4.2. Distribution across network complexity

To examine this, we discretized the number of parameters to six bins from less than 10 K (class A) to greater than 100M (class F); the result is shown in [Fig. 4](#). Extremely low end (class A) cases are rare and do not seem to offer considerable improvements. Class B has the highest frequency (23 cases), then class C, D, E, and F with 16, 11, 10, and 9 cases correspondingly. It can be seen that class B, which is a still relatively small network, has been compared to spectral SVM only and is mostly present in the upper right part of the graph, where the performance of spectral SVM is already high. These cases are those mostly associated with supervised learning method mentioned before. But larger networks (especially classes D and F) show considerable improvements over enhanced SVM. Based on this, we may advise to use larger networks (with fine-tuning) as mentioned before. However, this graph also demonstrates that all network complexity classes have the potential to achieve accuracy of 95% or more, which may be sufficient for many cases, especially considering other data limitations (e.g. registration errors). Note that class F cases are all ImageNet pre-trained networks, which are naturally the largest networks in our study cases.

3.4.3. Distribution across spatial resolution

The corresponding graph is shown in [Fig. 5](#). It is difficult to discern a specific pattern with respect to the spatial resolution, therefore no conclusive remarks could be made.

3.4.4. Distribution across input data dimensionality

[Fig. 6](#) organizes the results in three general categories, mostly separating RGB (group A) and hyperspectral images (group C), with group B being cases employing additional multispectral components such as NI and/or auxiliary data such as DSM/LiDAR.

Although it seems that multispectral group (B) generally achieves a bit less improvement compared to other two groups, there is no strong evidence and supporting theory for that.

3.4.5. Distribution across training size/proportion

In the examined manuscripts the sampling is either a single pixel (for pixel classification or image segmentation applications) or an image patch (for scene classification or target detection applications). Labeled data size is mostly in the order of a few thousands, with additional cases with considerably more labelled data. Sampling is done within the labelled dataset, with the proportion varying substantially in different implementations from as low as 0.1% to as high as 90%. We consider two different ways of training data size affecting network simulations. The first issue is the training data size, which should be considered in accordance to the network size and number of parameters. A large network with few training data may experience overfitting and lack of generalization, while a small network may not be powerful enough to model a complex set of training data. The other issue is the training data proportion, which imposes the same underfitting/overfitting scenario. We compared the absolute number of training data units (pixels or scenes) to the number of network parameters in our database and found that in almost 90% of cases we have less data units than networks parameters to be tuned. The overfitting control mechanisms such as regularization are always included in the network design and it will alleviate overfitting problem, but there is still a substantial difference between the remote sensing and computer vision fields, as we have very large reference datasets in the latter. Looking at [Table 4](#), the only case with millions of samples in remote sensing are ISPRS datasets, but the winners are all CNNs and there is no comparison reported with SVMs (competition is just between different CNN architectures) so we couldn't include them in our SVM-based charts.

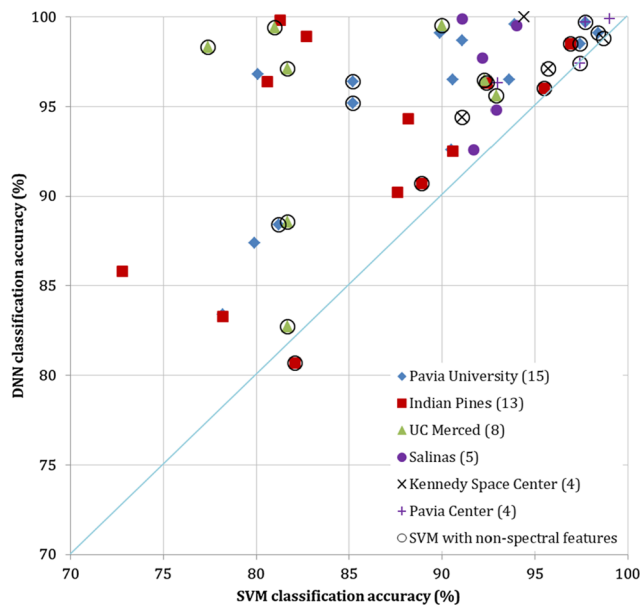


Fig. 9. Comparative performance distribution across widely used datasets.

In order to examine how DNN gains are influenced by training data absolute size and relative proportion with respect to the testing data two figures were produced. Fig. 7 shows the comparative performance categorized in training proportions from A (less than 20%) to E (greater than 80%), and Fig. 8 groups cases by absolute training dataset size from A (below 1000) to E (over one hundred thousand). The observed variability in the graphs and the lack of a consistent pattern suggest that high training size or proportion are not a general requirement for deep learning algorithms because there are various cases of high (> 95%) overall accuracy from very low to very high sampling ratio or size. A closer examination took place to further investigate training size and proportion with respect to network and learning method type. In cases of DBN, the training proportion was always high (> 50%) but there was no explanation or justification for it in the reviewed articles. In general, the CNN methods with supervised learning have been used in all training proportions, while CNNs with transfer learning with fine-tuning were run with higher training proportion. This may be attributed to overfitting concerns in transfer learning cases, as the base network is usually large with millions of parameters. Therefore, it is an open question how the transfer learning works in remote sensing cases where low training ratios are predominant.

There is also another concern that has not been discussed in the reviewed papers. About 64% of the cases in our entire database (and about 73% of cases used in the figures) belong to pixel classification category with the rest focusing on scene classification. In scene classification cases we have completely separate train and test scenes, therefore adding spatial data in training phase (which naturally happens in any CNN network) will not affect the testing performance. In pixel classification application the train and test pixels are chosen independently, but if the spatial processing is part of algorithm (that is typical), the training and testing pixels' neighborhoods may overlap and this may violate the basic assumption of independent training/testing samples. The real impact of this issue is not discussed in any of reviewed literature and it seems that the authors didn't consider it critical. It can be also argued that with multiple pooling layers in a CNN

network and enlarging scale of pixel influence, there is always some trace of even far pixels on training phase. Therefore, strictly enforcing independency rule to the neighboring pixels may invalidate all of the CNN networks, which is no desire for anybody.

3.4.6. Review of widely-used data sets

In previous sections the objective was to reveal patterns (or lack of any pattern) in different networks comparative performance along important parameters. The main limitation of this analysis is that the comparisons could not be done by varying just one parameter and fixing the others while we could not have such a control in our data collection (hence we used the term 'distribution' instead of 'effect' in our section titles). In this section we go one step further and look at different cases as applied on the same dataset to extract more information on the competency of different network types. There are some datasets that are heavily used in various papers and therefore can serve as a benchmark for algorithmic comparison. Except for ISPRS Potsdam and ISPRS Vaihingen (where comparison to SVM was not available), Fig. 9 shows the result graphically.

Here are some observations:

- Indian Pines (a hyperspectral dataset): The highest accuracy here is obtained by CNN at 99.8% overall accuracy, but SAE and CNN have generally the same level of performance. Their improvement over spectral SVM can be as large as 16% for both network types. However, this high gain is reduced to just about 2–4% in comparison to enhanced SVM cases.
- Kennedy Space Center (a hyperspectral dataset): Recently CNN achieved an accuracy of 100% on this dataset (Haut et al., 2018) but with a high training proportion of 85% and 5.6% gain improvement over spectral SVM. Other comparisons were made with CNN and SAE but with enhanced SVM. The best accuracy of SAE was 98.8%, which was almost the same as a very sophisticated SVM implementation.
- Pavia Center (a hyperspectral dataset): CNN implementations show a little improvement up to 3.3% with training proportion of 10%. The peak achieved accuracy was 99.95% but with a training proportion of 80% with very minor gains over SVM, and in all cases it was compared to the spectral SVM. We have only one SAE implementation for this dataset in our list, which does not improve over the enhanced SVM.
- Pavia University (a hyperspectral dataset): Here CNN here works better than SAE with a maximum overall accuracy of 99.7% and improvements up to 16.7% over spectral SVM, while for SAE it is at most 7.6% (both with training proportion of 10%). We have about 2% improvement over a very sophisticated SVM implementation for this dataset, but for SAE the gain over enhanced SVM is minor.
- Salinas (a hyperspectral dataset): This dataset was only applied to CNN and the best achieved accuracy was 99.9% with a training proportion of 50%. This was 8% gain in accuracy compared to spectral SVM, and other results showed some other gains. But there was no case of comparison with enhanced SVM.
- UC Merced (an RGB dataset used for scene classification): Here CNN works well with maximum overall accuracy of 99.5% and improvement up to 21% over SVM, while SAE was tested once with improvement of just 1%. In all cases, training proportion was high (60–80%) and it was compared to enhanced SVM, but SVM was fed with very different features in different cases.

For the ISPRS Potsdam and Vaihingen datasets, the CNNs has been the winner over all of the recent contests, so the race is only between

them and there is no research that compare them to a SVM based classification. Therefore, we could not include them in Fig. 7. In both ISPRS datasets the best results are achieved by transfer learning & fine-tuning in recent years. The best case was based on ResNet-101 with overall accuracy of 91.1% for both cases, followed by VGG-16 and SegNet-based cases with overall accuracy of 90.3%. Training proportion is standardized at 30% for Vaihingen and 45% for Potsdam (except ResNet-101 case, where the training proportion was 47% and 63%, respectively). These implementations are large networks, but a recent paper (Zhang et al., 2018b) has also achieved accuracies of 89.4% for Potsdam and 88.4% for Vaihingen with a small supervised network with number of parameters much less than above transferred networks (but with increasing training proportion to 70–75%). In almost all cases additional enhancement techniques such as joint segmentation, CRF processing or multiscale blocks has been implemented to boost the performance a bit higher.

The above datasets, while used extensively for classification assessment, should be avoided in the future. They are relatively small to match the generalization capabilities of deep networks and in most cases there are already algorithms that reach 100% accuracy, therefore offering limited opportunities for improvement. It is necessary to develop new, large and multi-sensory datasets for remote sensing image classification, especially for hyperspectral data, to help better investigate the potential of deep networks.

4. Concluding remarks

While the number of case studies precluded detailed statistical analysis on the effect of each contributing factors generally we can see that:

- Deep networks have generally better performance than spectral SVM implementations, with CNNs performing better than other deep learners. This advantage, however, diminishes when using SVM over more rich features extracted from data.
- Transfer learning and fine-tuning on pre-trained CNNs offer promising results even when compared to enhanced SVM implementations, and they provide for flexibility and scalability because there is no need to manually engineer the features or use a very large training dataset. However, these pre-trained networks are currently limited to RGB input data, therefore currently lack applicability in multi/hyperspectral data. They have also not been tested in low training proportion scenarios.
- There is no strong relationship between network complexity and accuracy gains over SVM; small to medium networks perform similarly to more complex networks.
- Contrary to popular belief, there are numerous cases of good deep network performance with training proportions of 10% or lower.

As previously noted, deep networks are important due to their ability to extract useful rich features automatically from large data sets without the need for manual feature extraction. For example, automatic feature extraction has been used in Rußwurm and Körner (2018) to automatically detect cloud occlusion in temporal remote sensing data. This automation of feature extraction also has limitations, most notably the difficulty to extract and evaluate these features. The visualizations in deep networks rarely go further than the first two layers, which focus on very basic features like edges and gradients. There have been limited trials to describe and visualize the extracted features and even

developing methods for it (for example see Zeiler and Fergus (2013) or Yosinski et al. (2015)), but currently research is lacking in remote sensing tasks.

We compare different studies and reflect on their findings in a collective manner. The possible reasons for deep network strengths in each individual aspect (network type, learning strategy, sampling proportion, etc.) was discussed in previous sections without going into mathematical formulas, due to the nature of meta-analysis. The majority of manuscripts reported that the SVM (or other rivals) parameters have been tuned and optimized for best performance, but there is a lack of consistency in reporting and protocol (e.g. grid search density). Establishing best optimization practices would benefit our community by limiting inconsistencies that could lead to result bias.

Another important conclusion is that algorithms are now outpacing benchmark datasets. We already see accuracy estimations exceeding 99% for some well-known datasets such as Indian Pine, Pavia Center and University, Salinas, and UC Merced. To allow deep learners to reach their full potential, it is paramount that more elaborate benchmark datasets should become available with diverse spectral/spatial/temporal resolution and geographic coverage.

We could not analyze further the processing time because either it was not available in many cases, or it was not specified if it contains the entire time for optimizing meta-parameters or not. It is generally true that deep networks need considerably more processing time for training (though the testing/simulation process is generally quick) but with continuous increases in processing power, deep networks are readily usable particularly by incorporating both CPUs and GPUs together. It would be interesting to evaluate the time saved by using pre-trained networks and just fine-tuning them, but currently there were no statistics reported to extract conclusive guidance.

There are numerous design options currently offered (see Table 3). Multiscale input is particularly useful to capture geographic relationships in earth observations. Furthermore, fully convolutional networks are promising for dense semantic labeling (classification of all image pixels at once and producing the same output dense map as the input image size). Other researches have added various segmentation techniques, boundary detection and correction methods and CRF/MRF post-processing and showed their benefit to enhance classification of edge pixels. While existing comparisons suggest the potential of CNN, they do not concretely identify a winning design among different options. For example, at the ISPRS Vaihingen image segmentation contest three CNN methods were within 1.2% of overall accuracy (Sherrah, 2016; Audebert et al., 2016; and Marmanis et al., 2016b). Looking into the future, remote sensing experts will favor 3-D CNN structures from pre-processing, dimensionality reduction methods like PCA or shallow 1-D and 2-D networks. The current state of the art 3-D CNN structures has already offered significant improvements and the training process is becoming easier (see Chen et al., 2016; Y. Li et al., 2017). Furthermore, our community would significantly benefit from a coordinated investment from large funding institutions to create a pre-trained DNN for remote sensing data (similar to the ImageNet for RGB images). This pre-trained network would harness the power of large data volumes while allowing fine-tuning to specific applications.

Acknowledgments

This work was supported by the USDA McIntire Stennis program, a SUNY ESF Graduate Assistantship and NASA's Land Cover Land Use Change Program (grant # NNX15AD42G).

Appendix A

Table A1
Database of collected deep network application in remote sensing.

Reference	Other network parameters			Dataset specification		Best reported performances				
	Network Type	# of parameters	Learning type	Dataset	Spatial resolution	# of channels	Training proportion	Metric type	Deep network	SVM (Non deep)
(Penatti et al., 2015)	CNN	289 M	Transfer learning	Brazilian coffee		3	0.8	Average accuracy	83	87
(Yu et al., 2017)	CNN	24.6 M	Unsupervised	Brazilian coffee		3	0.8	Overall accuracy	87.8	87
(Castelluccio et al., 2015)	CNN	5 M	Transfer learning	Brazilian coffee		3		Overall accuracy	91.8	
(Nogueira et al., 2017)	CNN	60 M	Transfer learning & fine-tuning	Brazilian coffee		3	0.6	Overall accuracy	94.5	87
(Wu and Prasad, 2018)	CNN + RNN		Semisupervised	Houston	2.5 m	144		Overall accuracy	82.6	80.2
(Xu et al., 2018)	CNN		Supervised +	Houston	2.5 m	144 + 1	0.19	Overall accuracy	88	80.5
(Pan et al., 2018)	DBN			Houston	2.5 m	144		Overall accuracy	90.8	
(Li et al., 2014)	DBN	14.7 K	Unsupervised & fine-tuning	Houston	2.5 m	144		Overall accuracy	97.7	97.5
(Zabalza et al., 2016)	SAE	4.2 K	Unsupervised	Indian Pines	20 m	200	0.05	Overall accuracy	80.7	82.1
(Ghamisi et al., 2016)	CNN	188 K	Supervised	Indian Pines	20 m	200	0.05	Overall accuracy	83.3	78.2
(Shi and Pun, 2018)	CNN	2.5 M	Supervised	Indian Pines	20 m	200	0.01	Overall accuracy	85.2	
(Mou et al., 2018b)	CNN	1.44 M	Unsupervised & fine-tuning	Indian Pines	20 m	200	0.05	Overall accuracy	85.8	72.8
(C. Zhao et al., 2017)	SAE	30.2 K	Unsupervised & fine-tuning	Indian Pines	20 m	200	0.1	Overall accuracy	89.8	88.9
(W. Hu et al., 2015)	CNN	80.6 K	Supervised	Indian Pines	20 m	220	0.2	Overall accuracy	90.2	87.6
(Pan et al., 2018)	CNN			Indian Pines	20 m	200		Overall accuracy	90.7	
(Xing et al., 2016)	SAE	241 K	Unsupervised	Indian Pines	20 m	200	0.5	Overall accuracy	92.1	90.6
(W. Li et al., 2017)	CNN	57.9 K	Supervised	Indian Pines	20 m	220	0.2	Overall accuracy	94.3	88.2
(Chen et al., 2015)	DBN		Unsupervised & fine-tuning	Indian Pines	20 m	200	0.5	Overall accuracy	96	95.5
(Li et al., 2015)	SAE	21.7 M	Unsupervised & fine-tuning	Indian Pines	20 m	200	0.05	Overall accuracy	96.3	92.4
(Sun et al., 2017)	SAE	107 K	Semisupervised	Indian Pines	20 m	200	0.1	Overall accuracy	96.4	80.6
(Ding et al., 2017)	CNN	380 K	Unsupervised	Indian Pines	20 m	200	0.5	Overall accuracy	97.8	
(Ma et al., 2015)	SAE	14.2 K	Unsupervised & fine-tuning	Indian Pines	20 m	200	0.1	Overall accuracy	98.2	
(Paoletti et al., 2018)	CNN	96 M	Supervised	Indian Pines	20 m	200	0.24	Overall accuracy	98.4	
(Chen et al., 2016)	CNN	44.9 M	Supervised	Indian Pines	20 m	200	0.2	Overall accuracy	98.5	96.9
(H. Zhang et al., 2017)	CNN		Supervised	Indian Pines	20 m	200	0.1	Overall accuracy	98.8	
(Makantasis et al., 2015)	CNN	97.6 K	Supervised	Indian Pines	20 m	224	0.8	Overall accuracy	98.9	82.7
(Y. Li et al., 2017)	CNN	197 K	Supervised	Indian Pines	20 m	200	0.5	Overall accuracy	99.1	
(Haut et al., 2018)	CNN	8.9 M	Supervised +	Indian Pines	20 m	200	0.5	Overall accuracy	99.8	81.3
(Sherrah, 2016)	CNN	3.26 M	Supervised	ISPRS Potsdam	5 cm	5	0.45	Overall accuracy	84.1	
(Volpi and Tuia, 2017)	CNN	6.38 M	Supervised	ISPRS Potsdam	5 cm	5	0.45	Overall accuracy	85.8	
(Maggiori et al., 2016)	CNN	530 K	Supervised	ISPRS Potsdam	5 cm	4	0.45	Overall accuracy	87	
(Zhang et al., 2018a)	CNN	17 K	Supervised	ISPRS Potsdam	5 cm	4	0.75	Overall accuracy	89.4	82.4
(Sherrah, 2016)	CNN	22.7 M	Transfer learning & fine-tuning	ISPRS Potsdam	5 cm	4	0.45	Overall accuracy	90.3	
(Yongcheng Liu et al., 2018)	CNN	481 M	Transfer learning & fine-tuning	ISPRS Potsdam	5 cm	4 (DSMs not used)	0.63	Overall accuracy	91.1	
(Tschannen et al., 2016)	CNN	30 K	Supervised	ISPRS Vaihingen	9 cm	5	0.3	Overall accuracy	85.5	
(Paisitriangkrai et al., 2015)	CNN		Supervised	ISPRS Vaihingen	9 cm	5	0.3	Overall accuracy	86.9	
(W. Zhao et al., 2017)	CNN		Supervised	ISPRS Vaihingen	9 cm	4	0.1	Overall accuracy	87.1	66.6
(Volpi and Tuia, 2017)	CNN	6.38 M	Supervised	ISPRS Vaihingen	9 cm	4	0.3	Overall accuracy	87.3	
(Marcos et al., 2018)	CNN	100 K	Supervised	ISPRS Vaihingen	9 cm	4	0.45	Overall accuracy	87.6	
(Zhang et al., 2018b)	CNN	17 K	Supervised	ISPRS Vaihingen	9 cm	4	0.7	Overall accuracy	88.4	81.7
(Maggiori et al., 2016)	CNN	727 K	Supervised	ISPRS Vaihingen	9 cm	4	0.3	Overall accuracy	88.9	
(Sherrah, 2016)	CNN	3.26 M	Supervised	ISPRS Vaihingen	9 cm	4	0.3	Overall accuracy	89.1	
(Audebert et al., 2016)	CNN	32 M	Transfer learning & fine-tuning	ISPRS Vaihingen	9 cm	4	0.3	Overall accuracy	89.8	
(Marmaris et al., 2016b)	CNN	806 M	Transfer learning & fine-tuning	ISPRS Vaihingen	9 cm	4	0.3	Overall accuracy	90.3	

(continued on next page)

Table A1 (continued)

Reference	Other network parameters			Dataset specification			Best reported performances		
	Network Type	# of parameters	Learning type	Dataset	Spatial resolution	# of channels	Training proportion	Metric type	Deep network
(Yongcheng Liu et al., 2018)	CNN	481 M	Transfer learning & fine-tuning	ISPRS Vaihingen	9 cm	3 (DSMs not used)	0.47	Overall accuracy	91.1
(C. Zhao et al., 2017)	SAE	20.8 K	Unsupervised & fine-tuning	Kennedy Space Center	18 m	224	0.1	Overall accuracy	93.5
(Chen et al., 2016)	CNN	5.85 M	Supervised	Kennedy Space Center	18 m	224	0.1	Overall accuracy	97.1
(Y. Chen et al., 2014)	SAE	8.72 K	Unsupervised & fine-tuning	Kennedy Space Center	18 m	176	0.6	Overall accuracy	98.8
(Haut et al., 2018)	CNN	8.8 M	Supervised +	Kennedy Space Center	18 m	224	0.85	Overall accuracy	100
(Ishii et al., 2015)	CNN	60 M	Supervised	Landsat 8	30 m	3	0.35	F1	71
(Wou et al., 2018a)	CNN + RNN	155 K	Supervised	Landsat ETM	30 m	6	0.25	Overall accuracy	98
(Karalas et al., 2015)	SAE	155 K	Unsupervised & fine-tuning	MODIS	500 sq.m	7	0.25	Average precision	62.8
(Zhou et al., 2017)	CNN	126 M	Transfer learning & fine-tuning	Other	0.5 m	3	0.25	ANMRR	0.04
(Kemker et al., 2018)	CNN	11.9 M	Supervised +	Other	4.7 cm	6	0.25	Average accuracy	57.3
(Kemker et al., 2018)	CNN	69 M	Supervised +	Other	4.7 cm	6	0.25	Average accuracy	59.8
(Bittner et al., 2017)	CNN	134 M	Transfer learning & fine-tuning	Other	0.5 m	1	0.25	F1	70
(Iagrange et al., 2015)	CNN	141 M	Transfer learning	Other	5 cm	4	0.6	Overall accuracy	72.4
(Cao et al., 2016)	CNN	60 M	Transfer learning & fine-tuning	Other	5 cm	3	0.6	F1	72.4
(Ji et al., 2018)	CNN	102 K	Supervised +	Other	15 m	4	0.85	Overall accuracy	79.4
(Fu et al., 2017)	CNN	102 K	Supervised	Other	1 m	3	0.9	F1	79.5
(Tang et al., 2017)	CNN	102 K	Transfer learning & fine-tuning	Other	1 m	3	0.9	Average precision	79.5
(Huang et al., 2018)	CNN	39 M	Transfer learning & fine-tuning	Other	0.5 m	4	0.57	Overall accuracy	80
(Chen et al., 2018)	CNN	39 M	Transfer learning & fine-tuning	Other	0.5 m	4	0.57	Overall accuracy	80
(Chen et al., 2018)	CNN	39 M	Transfer learning & fine-tuning	Other	8, 16 m	3	0.57	Average precision	80
(Chen et al., 2013)	DBN	4.2 M	Unsupervised & fine-tuning	Other	5 cm	3	0.2	F1	81.7
(Marcos et al., 2018)	CNN	430 K	Supervised	Other	5 cm	4	0.7	Overall accuracy	82.6
(Cheng et al., 2017b)	CNN	14.7 M	Transfer learning	Other	30 m	3	0.2	Overall accuracy	84.3
(Yanfei Liu et al., 2018)	CNN	14.7 M	Supervised	Other	4 m (MSI), 1 m (Pan)	3	0.8	Overall accuracy	85
(Yongcheng Liu et al., 2018)	CNN	481 M	Transfer learning & fine-tuning	Other	1 m	3	0.93	F1	85.6
(Geng et al., 2015)	SAE	28.4 K	Unsupervised & fine-tuning	Other	0.38 m	1	0.5	Overall accuracy	88.1
(Iguensat et al., 2017)	CNN	177 K	Supervised	Other	0.38 m	1	0.18	Overall accuracy	88.6
(Han et al., 2018)	CNN	286 M	Semisupervised	Other	30 m	1	0.18	Overall accuracy	88.6
(Zhao et al., 2015)	DBN	379 K	Unsupervised & fine-tuning	Other	0.6 m	1	0.7	Overall accuracy	88.9
(Zhang et al., 2018c)	CNN	226 K	Supervised	Other	50 cm	4	0.6	Overall accuracy	89.5
(Zhang et al., 2018a)	CNN	17 K	Supervised	Other	50 cm	4	0.5	Overall accuracy	89.6
(Zhang et al., 2018b)	CNN	17 K	Supervised	Other	50 cm	4	0.7	Overall accuracy	89.8
(Vakalopoulou et al., 2015)	CNN	60 M	Transfer learning	Other	0.6 m	4	0.4	Average precision	90
(Qayyum et al., 2017)	CNN	6.61 M	Transfer learning	Other	15 cm	3	0.8	Overall accuracy	90.3
(Cheng et al., 2017a)	CNN	134 M	Transfer learning & fine-tuning	Other	30 m	3	0.8	Overall accuracy	90.3
(F. Zhang et al., 2015)	SAE	90.3 K	Unsupervised	Other	1 m	3	0.25	Overall accuracy	90.8

(continued on next page)

Table A1 (continued)

Reference	Other network parameters			Dataset specification			Best reported performances		
	Network Type	# of parameters	Learning type	Dataset	Spatial resolution	# of channels	Training proportion	Metric type	Deep network
(Zhang et al., 2018a)	CNN	17 K	Supervised	Other	50 cm	4	0.5	Overall accuracy	90.9
(Zhang et al., 2018c)	CNN	226 K	Supervised	Other	50 cm	4	0.6	Overall accuracy	90.9
(Zhang et al., 2018b)	CNN	17 K	Supervised	Other	50 cm	4	0.7	Overall accuracy	91
(Zhao and Du, 2016)	CNN	39 M	Supervised	Other	1.8 m	8	0.15	Overall accuracy	91.1
(Huang et al., 2018)	CNN		Transfer learning & fine-tuning	Other	1.24 m	8	0.62	Overall accuracy	91.3
(Khan et al., 2017)	CNN	151 M	Transfer learning & fine-tuning	Other	25 m	3	0.9	Overall accuracy	91.3
(Han et al., 2018)	CNN	286 M	Semisupervised	Other				Overall accuracy	91.4
(X. Chen et al., 2014)	CNN	395 K	Supervised	Other		1		F1	91.6
(L. Zhang et al., 2015)	CNN	44 M	Transfer learning	Other		3	0.75	F1	91.8
(Khan et al., 2017)	CNN	151 M	Transfer learning & fine-tuning	Other	25 m	3	0.9	Overall accuracy	92
(F. Zhang et al., 2017)	CNN	266 K	Supervised	Other	1.2 m	3	0.8	Overall accuracy	92.4
(Pan et al., 2018)	CNN		Transfer learning & fine-tuning	Other	1 m	84		Overall accuracy	93.2
(S. Liu et al., 2018)	CNN	28.4 M	Transfer learning & fine-tuning	Other		3	0.67	Overall accuracy	93.4
(Basu et al., 2015)	DBN	3.6 K	Unsupervised & fine-tuning	Other		4	0.8	Overall accuracy	93.9
(Långkvist et al., 2016)	CNN	1.91 M	Unsupervised & fine-tuning	Other	0.5 m	6	0.7	Overall accuracy	94.5
(Ji et al., 2018)	CNN	102 K	Supervised +	Other	4 m	4	0.17	Overall accuracy	94.7
(Rezaee et al., 2018)	CNN	53.9 M	Transfer learning & fine-tuning	Other	5 m	5 (3 used for CNN)	0.46	Overall accuracy	94.8
(Cui et al., 2018)	CNN	9.7 K	Supervised	Other	2 m (MSI), 0.5 m (Pan)	8 (MSI) + Pan	0.8	Overall accuracy	94.8
(Yanfei Liu et al., 2018)	CNN		Supervised	Other	2 m	3	0.8	Overall accuracy	94.8
(Xing et al., 2016)	SAE	52.8 K	Unsupervised	Other	30 m	224	0.5	Overall accuracy	95.5
(Gong et al., 2017)	SAE	81 K	Unsupervised & fine-tuning	Other	2 m	4	0.5	Overall accuracy	95.7
(Ma et al., 2016)	CNN		Supervised	Other		4	0.8	Overall accuracy	96
(W. Zhao et al., 2017)	CNN	286 M	Supervised	Other	0.5 m	8	0.1	Overall accuracy	96.3
(Han et al., 2018)	CNN		Semisupervised	Other		3		Overall accuracy	96.8
(Hu et al., 2017)	CNN		Supervised	Other	1 m	161		Overall accuracy	97
(Yu et al., 2016)	DBN	2.43 M	Unsupervised & fine-tuning	Other	0.27 m	3		F1	97
(Wu and Prasad, 2018)	CNN + RNN		Semisupervised	Other	1 m	360		Overall accuracy	97.3
(Wang et al., 2018)	CNN	252 K	Supervised	Other		3	0.74	Overall accuracy	97.3
(Basu et al., 2015)	DBN	3.6 K	Unsupervised & fine-tuning	Other		4	0.8	Overall accuracy	97.9
(Xu et al., 2018)	CNN		Supervised +	Other	1 m	63 + 1	0.03	Overall accuracy	97.9
(Nogueira et al., 2017)	CNN	5 M	Transfer learning & fine-tuning	Other	0.5 m	3	0.6	Overall accuracy	98
(Tao et al., 2018)	CNN		Supervised	Other	0.5–4 m	4	0.008	Overall accuracy	98.4
(Ma et al., 2016)	CNN		Supervised	Other		4	0.8	Overall accuracy	98.4
(Gong et al., 2018)	CNN	139 M	Transfer learning & fine-tuning	Other	2 m	3	0.8	Overall accuracy	98.5
(Wang et al., 2015)	CNN	438 K	Supervised	Other		3	0.6	Overall accuracy	98.7
(Fan Zhang et al., 2016)	CNN		Supervised	Other	1 m	3	0.2	Overall accuracy	98.8
(Weng et al., 2018)	CNN	3.4 M	Transfer learning	Other		3	0.25	Overall accuracy	98.8
(Gong et al., 2018)	CNN	139 M	Transfer learning & fine-tuning	Other		3	0.8	Overall accuracy	98.8
(Ji et al., 2018)	CNN	107 K	Supervised +	Other	4 m	4	0.03	Overall accuracy	98.9
(Maggiore et al., 2017)	CNN	459 K	Supervised	Other	1 m	3	0.9	Overall accuracy	99.5
(Y. Li et al., 2017)	CNN	128 K	Supervised	Other	30 m	242	0.5	Overall accuracy	99.6
(Basaed et al., 2016)	CNN	56.4 K	Supervised	Other	30 m	10	0.75	Overall accuracy	99.7
(Weng et al., 2018)	CNN	3.4 M	Transfer learning	Other			0.5	Overall accuracy	99.7

(continued on next page)

Table A1 (continued)

Reference	Other network parameters			Dataset specification			Best reported performances			
	Network Type	# of parameters	Learning type	Dataset	Spatial resolution	# of channels	Training proportion	Metric type	Deep network	SVM (Non deep)
(W. Zhao et al., 2017)	CNN		Supervised	Pavia Center	1.3 m	103	0.1	Overall accuracy	96.3	92.98
(Shi and Pun, 2018)	CNN	673 K	Supervised	Pavia Center	1.3 m	103	0.001	Overall accuracy	97	
(Aptoula et al., 2016)	CNN	1.31 M	Supervised	Pavia Center	1.3 m	103	0.05	Kappa	97.4	
(Zabalza et al., 2016)	SAE	2.4 K	Unsupervised	Pavia Center	1.3 m	103	0.05	Overall accuracy	97.4	97.4
(Ben Hamida et al., 2018)	CNN	3681	Supervised	Pavia Center	1.3 m	103	0.05	Overall accuracy	98.9	
(Tao et al., 2015)	SAE		Unsupervised	Pavia Center	1.3 m	103	0.05	Overall accuracy	99.6	
(Zhao and Du, 2016)	CNN		Supervised	Pavia Center	1.3 m	103	0.05	Overall accuracy	99.7	97.7
(Makantasis et al., 2015)	CNN	10.9 K	Supervised	Pavia Center	1.3 m	103	0.8	Overall accuracy	99.9	99
(Ghamisi et al., 2016)	CNN	188 K	Supervised	Pavia University	1.3 m	103	0.1	Overall accuracy	83.4	78.2
(Mou et al., 2018b)	CNN	1.39 M	Unsupervised & fine-tuning	Pavia University	1.3 m	103	0.1	Overall accuracy	87.4	79.9
(Wu and Prasad, 2018)	CNN + RNN		Semisupervised	Pavia University	1.3 m	103		Overall accuracy	88.4	81.2
(Ding et al., 2017)	CNN	226 K	Unsupervised	Pavia University	1.3 m	100	0.5	Overall accuracy	90.6	
(W. Hu et al., 2015)	CNN	80.6 K	Supervised	Pavia University	1.3 m	103	0.05	Overall accuracy	92.6	90.5
(Yue et al., 2015)	CNN	182 K	Supervised	Pavia University	1.3 m	103		Overall accuracy	95.2	85.2
(Xing et al., 2016)	SAE	212 K	Unsupervised	Pavia University	1.3 m	103	0.5	Overall accuracy	96	93.6
(Zhao et al., 2015)	CNN	239 K	Unsupervised	Pavia University	1.3 m	103	0.1	Overall accuracy	96.4	85.2
(W. Li et al., 2017)	CNN	57.9 K	Supervised	Pavia University	1.3 m	103	0.05	Overall accuracy	96.5	90.6
(Zhao and Du, 2016)	CNN		Supervised	Pavia University	1.3 m	103	0.1	Overall accuracy	96.8	80.1
(Ben Hamida et al., 2018)	CNN	6862	Supervised	Pavia University	1.3 m	103	0.05	Overall accuracy	97.2	
(Paoletti et al., 2018)	CNN	173 M	Supervised	Pavia University	1.3 m	103	0.04	Overall accuracy	97.8	
(Aptoula et al., 2016)	CNN	1.31 M	Supervised	Pavia University	1.3 m	103	0.1	Kappa	97.9	
(Shi and Pun, 2018)	CNN	673 K	Supervised	Pavia University	1.3 m	103	0.01	Overall accuracy	98.5	
(Y. Chen et al., 2014)	SAE	29 K	Unsupervised & fine-tuning	Pavia University	1.3 m	103	0.6	Overall accuracy	98.5	97.4
(Ma et al., 2015)	SAE		Unsupervised	Pavia University	1.3 m	103	0.1	Overall accuracy	98.6	
(Ma et al., 2015)	SAE	10 K	Unsupervised & fine-tuning	Pavia University	1.3 m	103	0.1	Overall accuracy	98.7	
(Sun et al., 2017)	SAE	30.2 K	Semisupervised	Pavia University	1.3 m	103	0.1	Overall accuracy	98.7	
(Xu et al., 2018)	CNN		Supervised +	Pavia University	1.3 m	103	0.04	Overall accuracy	99.1	91.1
(Chen et al., 2015)	DBN		Unsupervised & fine-tuning	Pavia University	1.3 m	103	0.5	Overall accuracy	99.1	89.9
(Y. Li et al., 2017)	CNN	110 K	Supervised	Pavia University	1.3 m	103	0.5	Overall accuracy	99.4	98.4
(Makantasis et al., 2015)	CNN	10.9 K	Supervised	Pavia University	1.3 m	103	0.8	Overall accuracy	99.6	93.9
(H. Zhang et al., 2017)	CNN		Supervised	Pavia University	1.3 m	103	0.05	Overall accuracy	99.7	
(Chen et al., 2016)	CNN	5.85 M	Supervised	Pavia University	1.3 m	103	0.1	Overall accuracy	99.7	97.7
(Zhou et al., 2017)	CNN	126 M	Transfer learning & fine-tuning	RSSCN7	1.3 m	3		ANMRR	0.3	
(Zou et al., 2015)	DBN	3.1 M	Unsupervised & fine-tuning	RSSCN7		3	0.5	Average accuracy	77	
(Wu et al., 2016)	SAE	2.53 M	Unsupervised	RSSCN7		3	0.5	Overall accuracy	90.4	
(W. Hu et al., 2015)	CNN	80.6 K	Supervised	Salinas	3.7 m	220	0.05	Overall accuracy	92.6	91.7
(W. Li et al., 2017)	CNN	57.9 K	Supervised	Salinas	3.7 m	204	0.05	Overall accuracy	94.8	92.9
(Xu et al., 2018)	CNN		Supervised +	Salinas	3.7 m	204	0.06	Overall accuracy	97.7	92.2
(Ma et al., 2015)	SAE	37.7 K	Unsupervised & fine-tuning	Salinas	3.7 m	204	0.01	Overall accuracy	98.3	
(Makantasis et al., 2015)	CNN	10.9 K	Supervised	Salinas	3.7 m	224	0.8	Overall accuracy	99.5	94
(Haut et al., 2018)	CNN	8.9 M	Supervised +	Salinas	3.7 m	204	0.5	Overall accuracy	99.9	91.1
(Zhou et al., 2017)	CNN	126 M	Transfer learning & fine-tuning	UC Merced	1 ft	3		ANMRR	0.33	
(Zhou et al., 2015)	SAE	51.6 K	Unsupervised	UC Merced	1 ft	3		Average precision	64.5	
(F. Zhang et al., 2015)	SAE	301 K	Unsupervised	UC Merced	1 ft	3	0.8	Overall accuracy	82.7	81.7
(Romero et al., 2016)	CNN	49.1 M	Unsupervised	UC Merced	1 ft	3	0.8	Overall accuracy	84.5	
(Yu et al., 2017)	CNN	24.6 M	Unsupervised	UC Merced	1 ft	3	0.8	Overall accuracy	88.57	81.7
(Marmaris et al., 2016a)	CNN	155 M	Transfer learning & fine-tuning	UC Merced	1 ft	3	0.7	Overall accuracy	92.4	

(continued on next page)

(continued on next page)

Table A1 (continued)

Reference	Other network parameters				Dataset specification		Best reported performances			
	Network Type	# of parameters	Learning type	Dataset	Spatial resolution	# of channels	Training proportion	Metric type	Deep network	SYM (Non deep)
(Wu et al., 2016)	SAE	2.53 M	Unsupervised	UC Merced	1 ft	3	0.5	Overall accuracy	92.7	
(Weng et al., 2017)	CNN	60 M	Transfer learning	UC Merced	1 ft	3	0.7	Overall accuracy	93.4	
(Luus et al., 2015)	CNN	920 K	Supervised	UC Merced	1 ft	3	0.8	Overall accuracy	93.5	
(Fan Zhang et al., 2016)	CNN		Supervised	UC Merced	1 ft	3	0.8	Overall accuracy	94.5	
(Han et al., 2018)	CNN	286 M	Semisupervised	UC Merced	1 ft	3		Overall accuracy	94.5	
(Yanfei Liu et al., 2018)	CNN		Supervised	UC Merced	1 ft	3	0.8	Overall accuracy	95.6	92.9
(Zhou et al., 2016)	CNN	126 M	Transfer learning & fine-tuning	UC Merced	1 ft	3	0.8	Overall accuracy	96.48	92.3
(F. Hu et al., 2015)	CNN	19.6 M	Transfer learning	UC Merced	1 ft	3	0.8	Overall accuracy	96.9	
(Castelluccio et al., 2015)	CNN	5 M	Transfer learning	UC Merced	1 ft	3		Overall accuracy	97.1	
(Gu et al., 2018)	CNN	117 M	Transfer learning	UC Merced	1 ft	3	0.8	Overall accuracy	97.1	81.7
(Gong et al., 2018)	CNN	139 M	Transfer learning & fine-tuning	UC Merced	1 ft	3	0.8	Overall accuracy	98.3	77.4
(Penatti et al., 2015)	CNN	204 M	Transfer learning	UC Merced	1 ft	3	0.8	Average accuracy	99.4	81
(Nogueira et al., 2017)	CNN	5 M	Transfer learning & fine-tuning	UC Merced	1 ft	3	0.6	Overall accuracy	99.5	90
(F. Hu et al., 2015)	CNN	44.1 M	Transfer learning	WHU-RS19		3	0.6	Overall accuracy	98.6	

References

- Aptoula, E., Ozdemir, M.C., Yanikoglu, B., 2016. Deep learning with attribute profiles for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* 13, 1970–1974. <https://doi.org/10.1109/LGRS.2016.2619354>.
- Audebert, N., Le Saux, B., Lefèvre, S., 2016. Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. In: *Asian Conference on Computer Vision*. Springer, Cham, pp. 180–196.
- Basaeed, E., Bhaskar, H., Al-Mualla, M., 2016. Supervised remote sensing image segmentation using boosted convolutional neural networks. *Knowl.-Based Syst.* 99, 19–27. <https://doi.org/10.1016/j.knosys.2016.01.028>.
- Basu, S., Ganguly, S., Mukhopadhyay, S., DiBiano, R., Karki, M., Nemani, R., 2015. DeepSAT: a learning framework for satellite imagery. In: *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 37.
- Ben Hamida, A., Benoit, A., Lambert, P., Ben Amar, C., 2018. 3-D deep learning approach for remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.* 56, 4420–4434. <https://doi.org/10.1109/TGRS.2018.2818945>.
- Bengio, Y., 2009. Learning deep architectures for AI. *Found. Trends Mach. Learn.* 2, 1–127. <https://doi.org/10.1561/22000000006>.
- Bittner, K., Cui, S., Reinartz, P., 2017. Building extraction from remote sensing data using fully convolutional networks. *ISPRS – Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* XLII-1/W1, 481–486. <https://doi.org/10.5194/isprs-archives-XLII-1-W1-481-2017>.
- Camps-Valls, G., Tuia, D., Bruzzone, L., Benediktsson, J.A., 2014. Advances in hyperspectral image classification: earth monitoring with statistical learning methods. *IEEE Signal Process. Mag.* 31, 45–54. <https://doi.org/10.1109/MSP.2013.2279179>.
- Cao, Y., Niu, X., Dou, Y., 2016. Region-based convolutional neural networks for object detection in very high resolution remote sensing images. *IEEE* 548–554. <https://doi.org/10.1109/FSKD.2016.7603232>.
- Castelluccio, M., Poggi, G., Sansone, C., Verdoliva, L., 2015. Land use classification in remote sensing images by convolutional. *Neural Networks ArXiv150800092 Cs*.
- Chen, F., Ren, R., Van de Voorde, T., Xu, W., Zhou, G., Zhou, Y., 2018. Fast automatic airport detection in remote sensing images using convolutional neural networks. *Remote Sens.* 10, 443. <https://doi.org/10.3390/rs10030443>.
- Chen, X., Xiang, S., Liu, C.-L., Pan, C.-H., 2014. Vehicle detection in satellite images by hybrid deep convolutional neural networks. *IEEE Geosci. Remote Sens. Lett.* 11, 1797–1801. <https://doi.org/10.1109/LGRS.2014.2309695>.
- Chen, X., Xiang, S., Liu, C.-L., Pan, C.-H., 2013. Aircraft detection by deep belief nets. *IEEE* 54–58. <https://doi.org/10.1109/ACPR.2013.5>.
- Chen, Y., Jiang, H., Li, C., Jia, X., Ghamisi, P., 2016. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* 54, 6232–6251. <https://doi.org/10.1109/TGRS.2016.2584107>.
- Chen, Y., Lin, Z., Zhao, X., Wang, G., Gu, Y., 2014. Deep learning-based classification of hyperspectral data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 7, 2094–2107. <https://doi.org/10.1109/JSTARS.2014.2329330>.
- Chen, Y., Zhao, X., Jia, X., 2015. Spectral-spatial classification of hyperspectral data based on deep belief network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 8, 2381–2392. <https://doi.org/10.1109/JSTARS.2015.2388577>.
- Cheng, G., Han, J., Lu, X., 2017a. Remote sensing image scene classification: benchmark and state of the art. *Proc. IEEE* 105, 1865–1883. <https://doi.org/10.1109/JPROC.2017.2675998>.
- Cheng, G., Li, Z., Yao, X., Guo, L., Wei, Z., 2017b. Remote sensing image scene classification using bag of convolutional features. *IEEE Geosci. Remote Sens. Lett.* 14, 1735–1739. <https://doi.org/10.1109/LGRS.2017.2731997>.
- Cui, W., Zheng, Z., Zhou, Q., Huang, J., Yuan, Y., 2018. Application of a parallel spectral-spatial convolution neural network in object-oriented remote sensing land use classification. *Remote Sens. Lett.* 9, 334–342. <https://doi.org/10.1080/2150704X.2017.1420265>.
- Deng, L., 2014. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Trans. Signal Inf. Process.* 3. <https://doi.org/10.1017/atsip.2013.9>.
- Ding, C., Li, Y., Xia, Y., Wei, W., Zhang, L., Zhang, Y., 2017. Convolutional neural networks based hyperspectral image classification method with adaptive kernels. *Remote Sens.* 9, 618. <https://doi.org/10.3390/rs9060618>.
- Fu, G., Liu, C., Zhou, R., Sun, T., Zhang, Q., 2017. Classification for high resolution remote sensing imagery using a fully convolutional network. *Remote Sens.* 9, 498. <https://doi.org/10.3390/rs9050498>.
- Geng, J., Fan, J., Wang, H., Ma, X., Li, B., Chen, F., 2015. High-resolution SAR image classification via deep convolutional autoencoders. *IEEE Geosci. Remote Sens. Lett.* 12, 2351–2355. <https://doi.org/10.1109/LGRS.2015.2478256>.
- Ghamisi, P., Chen, Y., Zhu, X.X., 2016. A self-improving convolution neural network for the classification of hyperspectral data. *IEEE Geosci. Remote Sens. Lett.* 13, 1537–1541. <https://doi.org/10.1109/LGRS.2016.2595108>.
- Ghamisi, P., Plaza, J., Chen, Y., Li, J., Plaza, A.J., 2017. Advanced spectral classifiers for hyperspectral images: a review. *IEEE Geosci. Remote Sens. Mag.* 5, 8–32. <https://doi.org/10.1109/MGRS.2016.2616418>.
- Gong, M., Zhan, T., Zhang, P., Miao, Q., 2017. Superpixel-based difference representation learning for change detection in multispectral remote sensing images. *IEEE Trans. Geosci. Remote Sens.* 55, 2658–2673. <https://doi.org/10.1109/TGRS.2017.2650198>.
- Gong, X., Xie, Z., Liu, Y., Shi, X., Zheng, Z., 2018. Deep salient feature based anti-noise transfer network for scene classification of remote sensing imagery. *Remote Sens.* 10, 410. <https://doi.org/10.3390/rs10030410>.
- Goodfellow, I., Bengio, Y., Courville, A., 2016. *Deep Learning*. The MIT Press, Cambridge, Massachusetts.

- Gu, X., Angelov, P.P., Zhang, C., Atkinson, P.M., 2018. A massively parallel deep rule-based ensemble classifier for remote sensing scenes. *IEEE Geosci. Remote Sens. Lett.* 15, 345–349. <https://doi.org/10.1109/LGRS.2017.2787421>.
- Han, W., Feng, R., Wang, L., Cheng, Y., 2018. A semi-supervised generative framework with deep learning features for high-resolution remote sensing image scene classification. *ISPRS J. Photogramm. Remote Sens.* 145, 23–43. <https://doi.org/10.1016/j.isprsjprs.2017.11.004>.
- Haut, J.M., Paoletti, M.E., Plaza, J., Li, J., Plaza, A., 2018. Active learning with convolutional neural networks for hyperspectral image classification using a new bayesian approach. *IEEE Trans. Geosci. Remote Sens.* 56, 6440–6461. <https://doi.org/10.1109/TGRS.2018.2838665>.
- Heydari, S.S., Mountrakis, G., 2018. Effect of classifier selection, reference sample size, reference class distribution and scene heterogeneity in per-pixel classification accuracy using 26 Landsat sites. *Remote Sens. Environ.* 204, 648–658. <https://doi.org/10.1016/j.rse.2017.09.035>.
- Hinton, G.E., Osindero, S., Teh, Y.-W., 2006. A fast learning algorithm for deep belief nets. *Neural Comput.* 18, 1527–1554. <https://doi.org/10.1162/neco.2006.18.7.1527>.
- Hu, F., Xia, G.-S., Hu, J., Zhang, L., 2015. Transferring deep convolutional neural networks for the scene classification of high-resolution multispectral remote sensing imagery. *Remote Sens.* 7, 14680–14707. <https://doi.org/10.3390/rs71114680>.
- Hu, J., Mou, L., Schmitt, A., Zhu, X.X., 2017. FusioNet: a two-stream convolutional neural network for urban scene classification using PolSAR and hyperspectral data. *IEEE* 1–4. <https://doi.org/10.1109/JURSE.2017.7924565>.
- Hu, W., Huang, Y., Wei, L., Zhang, F., Li, H., 2015. Deep convolutional neural networks for hyperspectral image classification. *J. Sens.* 2015, 1–12. <https://doi.org/10.1155/2015/258619>.
- Huang, B., Zhao, B., Song, Y., 2018. Urban land-use mapping using a deep convolutional neural network with high spatial resolution multispectral remote sensing imagery. *Remote Sens. Environ.* 214, 73–86. <https://doi.org/10.1016/j.rse.2018.04.050>.
- Ishii, T., Nakamura, R., Nakada, H., Mochizuki, Y., Ishikawa, H., 2015. Surface object recognition with CNN and SVM in Landsat 8 images. *IEEE* 341–344. <https://doi.org/10.1109/MVA.2015.7153200>.
- Ji, S., Zhang, C., Xu, A., Shi, Y., Duan, Y., 2018. 3D convolutional neural networks for crop classification with multi-temporal remote sensing images. *Remote Sens.* 10, 75. <https://doi.org/10.3390/rs10010075>.
- Karalas, K., Tsagkatakis, G., Zervakis, M., Tsakalides, P., 2015. Deep learning for multi-label land cover classification. In: Bruzzone, L. (Ed.), p. 96430Q. <https://doi.org/10.1117/12.2195082>.
- Kemker, R., Salvaggio, C., Kanan, C., 2018. Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning. *ISPRS J. Photogramm. Remote Sens.* 145, 60–77. <https://doi.org/10.1016/j.isprsjprs.2018.04.014>.
- Khan, S.H., He, X., Porikli, F., Bennamoun, M., 2017. Forest change detection in incomplete satellite images with deep neural networks. *IEEE Trans. Geosci. Remote Sens.* 1–17. <https://doi.org/10.1109/TGRS.2017.2707528>.
- Khatami, R., Mountrakis, G., Stehman, S.V., 2016. A meta-analysis of remote sensing research on supervised pixel-based land-cover image classification processes: general guidelines for practitioners and future research. *Remote Sens. Environ.* 177, 89–100. <https://doi.org/10.1016/j.rse.2016.02.028>.
- Lagrange, A., Le Saux, B., Beaupere, A., Boulch, A., Chan-Hon-Tong, A., Herbin, S., Randrianarivo, H., Ferecatu, M., 2015. Benchmarking classification of Earth-observation data: from learning explicit features to convolutional networks, in: IGARSS 2015.
- Långkvist, M., Kiselev, A., Alirezaie, M., Loutfi, A., 2016. Classification and segmentation of satellite orthoimagery using convolutional neural networks. *Remote Sens.* 8, 329. <https://doi.org/10.3390/rs8040329>.
- Le, Q.V., 2015. A Tutorial on Deep Learning Part 2: Autoencoders, Convolutional Neural Networks and Recurrent Neural Networks.
- Le Saux, B., Yokoya, N., Hansch, R., Prasad, S., 2018. Advanced multisource optical remote sensing for urban land use and land cover classification [Technical Committees]. *IEEE Geosci. Remote Sens. Mag.* 6, 85–89. <https://doi.org/10.1109/MGRS.2018.2874328>.
- Lguensat, R., Sun, M., Fablet, R., Mason, E., Tandeo, P., Chen, G., 2017. EddyNet: a deep neural network for pixel-wise classification of oceanic eddies. *ArXiv171103954 Phys.*
- Li, J., Bruzzone, L., Liu, S., 2015. Deep feature representation for hyperspectral image classification. *IEEE* 4951–4954. <https://doi.org/10.1109/IGARSS.2015.7326943>.
- Li, T., Zhang, J., Zhang, Y., 2014. Classification of hyperspectral image based on deep belief networks. In: 2014 IEEE International Conference on Image Processing (ICIP). Presented at the 2014 IEEE International Conference on Image Processing (ICIP), pp. 5132–5136. 10.1109/ICIP.2014.7026039.
- Li, W., Wu, G., Zhang, F., Du, Q., 2017. Hyperspectral image classification using deep pixel-pair features. *IEEE Trans. Geosci. Remote Sens.* 55, 844–853. <https://doi.org/10.1109/TGRS.2016.2616355>.
- Li, Y., Zhang, H., Shen, Q., 2017. Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sens.* 9, 67. <https://doi.org/10.3390/rs9010067>.
- Liu, P., Choo, K.-K.R., Wang, L., Huang, F., 2017. SVM or deep learning? A comparative study on remote sensing image classification. *Soft Comput.* 21, 7053–7065. <https://doi.org/10.1007/s00500-016-2247-2>.
- Liu, S., Li, M., Zhang, Z., Xiao, B., Cao, X., 2018. Multimodal ground-based cloud classification using joint fusion convolutional neural network. *Remote Sens.* 10, 822. <https://doi.org/10.3390/rs10060822>.
- Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F.E., 2017. A survey of deep neural network architectures and their applications. *Neurocomputing* 234, 11–26. <https://doi.org/10.1016/j.neucom.2016.12.038>.
- Liu, Yongcheng, Fan, B., Wang, L., Bai, J., Xiang, S., Pan, C., 2018. Semantic labeling in very high resolution images via a self-cascaded convolutional neural network. *ISPRS J. Photogramm. Remote Sens.* 145, 78–95. <https://doi.org/10.1016/j.isprsjprs.2017.12.007>.
- Liu, Yanfei, Zhong, Y., Fei, F., Zhu, Q., Qin, Q., 2018. Scene classification based on a deep random-scale stretched convolutional neural network. *Remote Sens.* 10, 444. <https://doi.org/10.3390/rs10030444>.
- Luus, F.P.S., Salmon, B.P., van den Bergh, F., Maharaj, B.T.J., 2015. Multiview deep learning for land-use classification. *IEEE Geosci. Remote Sens. Lett.* 12, 2448–2452. <https://doi.org/10.1109/LGRS.2015.2483680>.
- Lyu, H., Lu, H., Mou, L., 2016. Learning a transferable change rule from a recurrent neural network for land cover change detection. *Remote Sens.* 8, 506. <https://doi.org/10.3390/rs8060506>.
- Ma, L., Li, M., Ma, X., Cheng, L., Du, P., Liu, Y., 2017. A review of supervised object-based land-cover image classification. *ISPRS J. Photogramm. Remote Sens.* 130, 277–293. <https://doi.org/10.1016/j.isprsjprs.2017.06.001>.
- Ma, X., Geng, J., Wang, H., 2015. Hyperspectral image classification via contextual deep learning. *EURASIP J. Image Video Process* 2015. <https://doi.org/10.1186/s13640-015-0071-8>.
- Ma, Z., Wang, Z., Liu, C., Liu, X., 2016. Satellite Imagery Classification Based on Deep Convolution Network. *World Acad. Sci. Eng. Technol.* 10.
- Maggiori, E., Tarabalka, Y., Charpiat, G., Alliez, P., 2017. Convolutional neural networks for large-scale remote-sensing image classification. *IEEE Trans. Geosci. Remote Sens.* 55, 645–657. <https://doi.org/10.1109/TGRS.2016.2612821>.
- Maggiori, E., Tarabalka, Y., Charpiat, G., Alliez, P., 2016. High-Resolution Semantic Labeling with Convolutional Neural Networks. *ArXiv Prepr. ArXiv161101962*.
- Makantasis, K., Karantzalos, K., Doulamis, A., Doulamis, N., 2015. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In: *Geoscience and Remote Sensing Symposium (IGARSS)*, 2015 IEEE International. IEEE, pp. 4959–4962.
- Marcos, D., Volpi, M., Kellenberger, B., Tuia, D., 2018. Land cover mapping at very high resolution with rotation equivariant CNNs: towards small yet accurate models. *ISPRS J. Photogramm. Remote Sens.* 145, 96–107. <https://doi.org/10.1016/j.isprsjprs.2018.01.021>.
- Marmanis, D., Datcu, M., Esch, T., Stilla, U., 2016a. Deep learning earth observation classification using imagenet pretrained networks. *IEEE Geosci. Remote Sens. Lett.* 13, 105–109. <https://doi.org/10.1109/LGRS.2015.2499239>.
- Marmanis, D., Schindler, K., Wegner, J.D., Galliani, S., Datcu, M., Stilla, U., 2016b. Classification with an edge: improving semantic image segmentation with boundary detection. *ArXiv Prepr. ArXiv161201337*.
- Mou, L., Bruzzone, L., Zhu, X.X., 2018a. Learning Spectral-Spatial-Temporal Features via a Recurrent Convolutional Neural Network for Change Detection in Multispectral Imagery. *ArXiv180302642 Cs*.
- Mou, L., Ghamisi, P., Zhu, X.X., 2018b. Unsupervised spectral-spatial feature learning via deep residual conv-deconv network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 56, 391–406. <https://doi.org/10.1109/TGRS.2017.2748160>.
- Mou, L., Ghamisi, P., Zhu, X.X., 2017. Deep recurrent neural networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* 55, 3639–3655. <https://doi.org/10.1109/TGRS.2016.2636241>.
- Mountrakis, G., Im, J., Ogole, C., 2011. Support vector machines in remote sensing: a review. *ISPRS J. Photogramm. Remote Sens.* 66, 247–259. <https://doi.org/10.1016/j.isprsjprs.2010.11.001>.
- Ndikumana, E., Ho Tong Minh, D., Baghdadi, N., Courault, D., Hossard, L., 2018. Deep recurrent neural network for agricultural classification using multitemporal SAR sentinel-1 for camargue, France. *Remote Sens.* 10, 1217. <https://doi.org/10.3390/rs10081217>.
- Niculescu, S., Ienco, D., Hanganu, J., 2018. Application of deep learning of multi-temporal sentinel-1 images for the classification of coastal vegetation zone of the danube delta. *ISPRS – Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLII-3*, 1311–1318. <https://doi.org/10.5194/isprs-archives-XLII-3-1311-2018>.
- Nogueira, K., Penatti, O.A.B., dos Santos, J.A., 2017. Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognit.* 61, 539–556. <https://doi.org/10.1016/j.patcog.2016.07.001>.
- Paisitkriangkrai, S., Sherrah, J., Janney, P., Hengel, A.V.-D., 2015. Effective semantic pixel labelling with convolutional networks and Conditional Random Fields. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Presented at the 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 36–43. 10.1109/CVPRW.2015.7301381.
- Pan, B., Shi, Z., Xu, X., 2018. MugNet: deep learning for hyperspectral image classification using limited samples. *ISPRS J. Photogramm. Remote Sens.* 145, 108–119. <https://doi.org/10.1016/j.isprsjprs.2017.11.003>.
- Paoletti, M.E., Haut, J.M., Plaza, J., Plaza, A., 2018. A new deep convolutional neural network for fast hyperspectral image classification. *ISPRS J. Photogramm. Remote Sens.* 145, 120–147. <https://doi.org/10.1016/j.isprsjprs.2017.11.021>.
- Penatti, O.A.B., Nogueira, K., Santos, J.A. dos, 2015. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Presented at the 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 44–51. 10.1109/CVPRW.2015.7301382.
- Qayyum, A., Malik, A.S., Saad, N.M., Iqbal, M., Faris Abdullah, M., Rasheed, W., Rashid Abdullah, T.A., Bin Jafaar, M.Y., 2017. Scene classification for aerial images based on CNN using sparse coding technique. *Int. J. Remote Sens.* 38, 2662–2685. <https://doi.org/10.1080/01431161.2017.1296206>.
- Rezaee, M., Mahdianpari, M., Zhang, Y., Salehi, B., 2018. Deep convolutional neural network for complex wetland classification using optical remote sensing imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 11, 3030–3039. <https://doi.org/10.1109/JSTARS.2018.2838665>.

- 1109/JSTARS.2018.2846178.
- Romero, A., Gatta, C., Camps-Valls, G., 2016. Unsupervised deep feature extraction for remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.* 54, 1349–1362. <https://doi.org/10.1109/TGRS.2015.2478379>.
- Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.* 65, 386–408. <https://doi.org/10.1037/h0042519>.
- Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. *Nature* 323, 533–536. <https://doi.org/10.1038/323533a0>.
- Rußwurm, M., Körner, M., 2018. Multi-Temporal Land Cover Classification with Sequential Recurrent Encoders. *ArXiv1802.02080 Cs*.
- Rußwurm, M., Körner, M., 2017. Multi-temporal land cover classification with long short-term memory neural networks. *ISPRS – Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLII-1/W1*, 551–558. <https://doi.org/10.5194/isprs-archives-XLII-1-W1-551-2017>.
- Sharma, A., Liu, X., Yang, X., 2018. Land cover classification from multi-temporal, multi-spectral remotely sensed imagery using patch-based recurrent neural networks. *Neural Netw.* 105, 346–355. <https://doi.org/10.1016/j.neunet.2018.05.019>.
- Sherrah, J., 2016. Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. *ArXiv Prepr. ArXiv1606.02585*.
- Shi, C., Pun, C.-M., 2018. Superpixel-based 3D deep neural networks for hyperspectral image classification. *Pattern Recognit.* 74, 600–616. <https://doi.org/10.1016/j.patcog.2017.09.007>.
- Singhal, V., Gogna, A., Majumdar, A., 2016. Deep dictionary learning vs deep belief network vs stacked autoencoder: an empirical analysis. In: Hirose, A., Ozawa, S., Doya, K., Ikeda, K., Lee, M., Liu, D. (Eds.), *Neural Information Processing*. Springer International Publishing, Cham, pp. 337–344. https://doi.org/10.1007/978-3-319-46681-1_41.
- Sun, X., Zhou, F., Dong, J., Gao, F., Mu, Q., Wang, X., 2017. Encoding spectral and spatial context information for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* 14, 2250–2254. <https://doi.org/10.1109/LGRS.2017.2759168>.
- Tang, T., Zhou, S., Deng, Z., Zou, H., Lei, L., 2017. Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining. *Sensors* 17, 336. <https://doi.org/10.3390/s17020336>.
- Tao, C., Pan, H., Li, Y., Zou, Z., 2015. Unsupervised spectral-spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification. *IEEE Geosci. Remote Sens. Lett.* 12, 2438–2442. <https://doi.org/10.1109/LGRS.2015.2482520>.
- Tao, Y., Xu, M., Lu, Z., Zhong, Y., 2018. DenseNet-based depth-width double reinforced deep learning neural network for high-resolution remote sensing image per-pixel classification. *Remote Sens.* 10, 779. <https://doi.org/10.3390/rs10050779>.
- Tobler, W.R., 1970. A computer movie simulating urban growth in the detroit region. *Econ. Geogr.* 46, 234. <https://doi.org/10.2307/143141>.
- Tschannen, M., Cavigelli, L., Mentzer, F., Wiatowski, T., Benini, L., 2016. Deep Structured Features for Semantic Segmentation. *ArXiv1609.07916 Cs*.
- Vakalopoulou, M., Karantzalos, K., Komodakis, N., Paragios, N., 2015. Building detection in very high resolution multispectral data with deep learning features. *IEEE* 1873–1876. <https://doi.org/10.1109/IGARSS.2015.7326158>.
- Vincent, P., Larochelle, H., Larochelle, H., Bengio, Y., Manzagol, P.-A., 2010. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* 11, 3371–3408.
- Volpi, M., Tuia, D., 2017. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* 55, 881–893.
- Wang, J., Song, J., Chen, M., Yang, Z., 2015. Road network extraction: a neural-dynamic framework based on deep learning and a finite state machine. *Int. J. Remote Sens.* 36, 3144–3169. <https://doi.org/10.1080/01431161.2015.1054049>.
- Wang, S.-H., Sun, J., Phillips, P., Zhao, G., Zhang, Y.-D., 2018. Polarimetric synthetic aperture radar image segmentation by convolutional neural network using graphical processing units. *J. Real-Time Image Process.* 15, 631–642. <https://doi.org/10.1007/s11554-017-0717-0>.
- Weng, Q., Mao, Z., Lin, J., Guo, W., 2017. Land-use classification via extreme learning classifier based on deep convolutional features. *IEEE Geosci. Remote Sens. Lett.* 14, 704–708. <https://doi.org/10.1109/LGRS.2017.2672643>.
- Weng, Q., Mao, Z., Lin, J., Liao, X., 2018. Land-use scene classification based on a CNN using a constrained extreme learning machine. *Int. J. Remote Sens.* 39, 6281–6299. <https://doi.org/10.1080/01431161.2018.1458346>.
- Wu, H., Liu, B., Su, W., Zhang, W., Sun, J., 2016. Deep filter banks for land-use scene classification. *IEEE Geosci. Remote Sens. Lett.* 13, 1895–1899. <https://doi.org/10.1109/LGRS.2016.2616440>.
- Wu, H., Prasad, S., 2018. Semi-supervised deep learning using pseudo labels for hyperspectral image classification. *IEEE Trans. Image Process.* 27, 1259–1270. <https://doi.org/10.1109/TIP.2017.2772836>.
- Xia, G.-S., Hu, J., Hu, F., Shi, B., Bai, X., Zhong, Y., Zhang, L., Lu, X., 2017. AID: a benchmark data set for performance evaluation of aerial scene classification. *IEEE Trans. Geosci. Remote Sens.* 55, 3965–3981. <https://doi.org/10.1109/TGRS.2017.2685945>.
- Xing, C., Ma, L., Yang, X., 2016. Stacked denoise autoencoder based feature extraction and classification for hyperspectral images. *J. Sens.* 2016, 1–10. <https://doi.org/10.1155/2016/3632943>.
- Xu, X., Li, W., Ran, Q., Du, Q., Gao, L., Zhang, B., 2018. Multisource remote sensing data classification based on convolutional neural network. *IEEE Trans. Geosci. Remote Sens.* 56, 937–949. <https://doi.org/10.1109/TGRS.2017.2756851>.
- Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., Lipson, H., 2015. Understanding Neural Networks Through Deep Visualization. *ArXiv1506.06579 Cs*.
- Yu, Y., Gong, Z., Wang, C., Zhong, P., 2017. An unsupervised convolutional feature fusion network for deep representation of remote sensing images. *IEEE Geosci. Remote Sens. Lett.* 1–5. <https://doi.org/10.1109/LGRS.2017.2767626>.
- Yu, Y., Guan, H., Zai, D., Ji, Z., 2016. Rotation-and-scale-invariant airplane detection in high-resolution satellite images based on deep-Hough-forests. *ISPRS J. Photogramm. Remote Sens.* 112, 50–64. <https://doi.org/10.1016/j.isprsjprs.2015.04.014>.
- Yue, J., Zhao, W., Mao, S., Liu, H., 2015. Spectral-spatial classification of hyperspectral images using deep convolutional neural networks. *Remote Sens. Lett.* 6, 468–477. <https://doi.org/10.1080/2150704X.2015.1047045>.
- Zabalza, J., Ren, J., Zheng, J., Zhao, H., Qing, C., Yang, Z., Du, P., Marshall, S., 2016. Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging. *Neurocomputing* 185, 1–10. <https://doi.org/10.1016/j.neucom.2015.11.044>.
- Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (Eds.), *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I*. Springer International Publishing, Cham, pp. 818–833. https://doi.org/10.1007/978-3-319-10590-1_53.
- Zeiler, M.D., Fergus, R., 2013. Visualizing and Understanding Convolutional Networks. *ArXiv1311.2901 Cs*.
- Zhang, C., Pan, X., Li, H., Gardiner, A., Sargent, I., Hare, J., Atkinson, P.M., 2018a. A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification. *ISPRS J. Photogramm. Remote Sens.* 140, 133–144. <https://doi.org/10.1016/j.isprsjprs.2017.07.014>.
- Zhang, C., Sargent, I., Pan, X., Gardiner, A., Hare, J., Atkinson, P.M., 2018b. VPRS-based regional decision fusion of CNN and MRF classifications for very fine resolution remotely sensed images. *IEEE Trans. Geosci. Remote Sens.* 56, 4507–4521. <https://doi.org/10.1109/TGRS.2018.2822783>.
- Zhang, C., Sargent, I., Pan, X., Li, H., Gardiner, A., Hare, J., Atkinson, P.M., 2018c. An object-based convolutional neural network (OCNN) for urban land use classification. *Remote Sens. Environ.* 216, 57–70. <https://doi.org/10.1016/j.rse.2018.06.034>.
- Zhang, F., Du, B., Zhang, L., 2017. A multi-task convolutional neural network for mega-city analysis using very high resolution satellite imagery and geospatial data. *ArXiv1702.07985 Cs*.
- Zhang, Fan, Du, B., Zhang, L., 2016. Scene classification via a gradient boosting random convolutional network framework. *IEEE Trans. Geosci. Remote Sens.* 54, 1793–1802. <https://doi.org/10.1109/TGRS.2015.2488681>.
- Zhang, F., Du, B., Zhang, L., 2015. Saliency-guided unsupervised feature learning for scene classification. *IEEE Trans. Geosci. Remote Sens.* 53, 2175–2184. <https://doi.org/10.1109/TGRS.2014.2357078>.
- Zhang, H., Li, Y., Zhang, Y., Shen, Q., 2017. Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network. *Remote Sens. Lett.* 8, 438–447. <https://doi.org/10.1080/2150704X.2017.1280200>.
- Zhang, L., Shi, Z., Wu, J., 2015. A hierarchical oil tank detector with deep surrounding features for high-resolution optical satellite imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 8, 4895–4909. <https://doi.org/10.1109/JSTARS.2015.2467377>.
- Zhang, L., Zhang, L., Du, B., 2016. Deep learning for remote sensing data: a technical tutorial on the state of the art. *IEEE Geosci. Remote Sens. Mag.* 4, 22–40. <https://doi.org/10.1109/MGRS.2016.2540798>.
- Zhao, C., Wan, X., Zhao, G., Cui, B., Liu, W., Qi, B., 2017. Spectral-spatial classification of hyperspectral imagery based on stacked sparse autoencoder and random forest. *Eur. J. Remote Sens.* 50, 47–63. <https://doi.org/10.1080/22797254.2017.1274566>.
- Zhao, W., Du, S., 2016. Learning multiscale and deep representations for classifying remotely sensed imagery. *ISPRS J. Photogramm. Remote Sens.* 113, 155–165. <https://doi.org/10.1016/j.isprsjprs.2016.01.004>.
- Zhao, W., Du, S., Emery, W.J., 2017. Object-based convolutional neural network for high-resolution imagery classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 10, 3386–3396. <https://doi.org/10.1109/JSTARS.2017.2680324>.
- Zhao, W., Guo, Z., Yue, J., Zhang, X., Luo, L., 2015. On combining multiscale deep learning features for the classification of hyperspectral remote sensing imagery. *Int. J. Remote Sens.* 36, 3368–3379. <https://doi.org/10.1080/2150704X.2015.1062157>.
- Zhou, W., Newsam, S., Li, C., Shao, Z., 2017. Learning low dimensional convolutional neural networks for high-resolution remote sensing image retrieval. *Remote Sens.* 9, 489. <https://doi.org/10.3390/rs9050489>.
- Zhou, W., Shao, Z., Cheng, Q., 2016. Deep feature representations for high-resolution remote sensing scene classification, in: 2016 4th International Workshop on Earth Observation and Remote Sensing Applications (EORSA). Presented at the 2016 4th International Workshop on Earth Observation and Remote Sensing Applications (EORSA), pp. 338–342. <https://doi.org/10.1109/EORSA.2016.7552825>.
- Zhou, W., Shao, Z., Diao, C., Cheng, Q., 2015. High-resolution remote-sensing imagery retrieval using sparse features by auto-encoder. *Remote Sens. Lett.* 6, 775–783. <https://doi.org/10.1080/2150704X.2015.1074756>.
- Zhou, Y., Arpit, D., Nwogu, I., Govindaraju, V., 2014. Is Joint Training Better for Deep Auto-Encoders? *ArXiv14051380 Cs Stat*.
- Zhu, X., Goldberg, A.B., 2009. Introduction to semi-supervised learning. *Synth. Lect. Artif. Intell. Mach. Learn.* 3, 1–130. <https://doi.org/10.2200/S00196ED1V01Y200906AIM006>.
- Zhu, X.X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., Fraundorfer, F., 2017. Deep learning in remote sensing: a comprehensive review and list of resources. *IEEE Geosci. Remote Sens. Mag.* 5, 8–36. <https://doi.org/10.1109/MGRS.2017.2762307>.
- Zou, Q., Ni, L., Zhang, T., Wang, Q., 2015. Deep learning based feature selection for remote sensing scene classification. *IEEE Geosci. Remote Sens. Lett.* 12, 2321–2325. <https://doi.org/10.1109/LGRS.2015.2475299>.