An auto-encoder based classifier for crop mapping from multitemporal multispectral imagery

Masoumeh Hamidi[a]*, Abdolreza Safari[a], Saeid Homayouni[b]

[a] *School of Surveying and Geospatial Engineering, College of Engineering, U. of Tehran, Tehran, Iran*

[b] *Centre Eau Terre Environnement, Institut Nationale de la Recherche Scientifique, Québec, Canada*

* Corresponding author: m.hamidi@ut.ac.ir

An auto-encoder based classifier for crop mapping from multitemporal multispectral imagery

Crop mapping is a challenging task due to the spatial, spectral, and temporal variations within the cropland. These variations cause high intra-class and low inter-class variability problems. In this study, inspired by Deep Learning (DL) techniques, two Auto-Encoder (AE)-based learning schemes are proposed to exploit the spatio-temporal features in order to increase the stability of remotely sensed data classification for crop mapping. The first strategy is based on stacking the spatio-spectral features of different imaging dates to provide spatio-temporal spectral features and then feeding them as input to the Stacked AEs (SAEs). The spatio-spectral features are achieved by stacking the spectral features of all pixels in a Neighbourhood Window (NW) in each imaging date. The second strategy is an ensemble learning scheme, in which the base classifiers are SAEs trained considering different NW sizes. This method has an advantage such that while using the neighbourhood information, it maintains the classification accuracy in the boundary areas. To evaluate the performance of the proposed strategies, they were compared to the conventional classifiers, namely Linear Support Vector Machines (SVMs), Gaussian SVM, and Random Forest (RF). In addition, the effect of different train data sampling strategies and different proportions of train data on the classification performance was examined. The proposed ensemble (EN) strategy with a configuration of combining three NW sizes of 1, 3, and 5 pixels (i.e., 'AE, EN (1, 3, 5)') reached the highest accuracy in all experiments. Notably, in the experiment with a set of four imaging dates and 5.0% train data, it achieved an Overall Accuracy (OA) of 95.26%, the Kappa coefficient (*K*) of 0.94, and the average class accuracy of 92.16%.

Keywords: crop mapping, stacked auto-encoders, deep learning, ensemble learning, spatio-temporal classification

1. Introduction

Crop maps are essential for various agro-environmental applications including detailed quantification of agricultural products (Massey et al. 2017), accurate estimation of

agricultural statistics (Kussul et al. 2015), recovery of biophysical variables of plants using radiative transfer models (Immitzer, Vuolo, and Atzberger 2016), crop growth monitoring (Niazmardi et al. 2018), crop inventories (Peña and Brenning 2015; Khosravi, Safari, and Homayouni 2018), crop insurance (Whelen and Siqueira 2017), crop production forecasting (Arvor et al. 2011; Kussul et al. 2015; Immitzer, Vuolo, and Atzberger 2016; Inglada et al. 2016; Massey et al. 2017; Skakun et al. 2017; Sonobe et al. 2017; Whelen and Siqueira 2017; Niazmardi et al. 2018), and implementation of quota limits (Foody, McCulloch, and Yates 1994).

Crop mapping is a unique challenge due to high spatial, spectral, and temporal dynamics in the growing season (Bargiel 2017; McNairn et al. 2002; Kenduiywo, Bargiel, and Soergel 2018). Inherent spectral variability of crop types is repeatedly influenced by local climate or farming practices (Peña-Barragán et al. 2011; Song et al. 2017). Specifically, different crops in the same region may have similar spectral signatures, while crops of the same type may demonstrate different spectral behaviour in different location and situation (Song et al. 2017). The spectral variety of crops is typically influenced by the phenological stage of the crops at the time of imaging, and also by the spatial and spectral characterisations of the sensor. (Peña and Brenning 2015).

In recent years, new satellite multispectral sensors such as RapidEye and Sentinel-2 have provided high spatial resolutions and relatively high temporal resolutions with global coverage. Land-cover classification using these data allows for a significant distinction between different classes (Bargiel and Herrmann 2011). Temporal observations can provide valuable information on crop phenology and help to distinguish different crop types (Bargiel 2017). Each crop type has a specific crop calendar with well-defined cultivation times and unprecedented growth and

development patterns. As a result, spectral reflections can be distinguished during the growth period of different crops. Using images of optical sensors with high temporal resolutions also improves the accessibility of high-quality images in key phenological phases for classification by providing a large number of observation opportunities. Hence, multi-temporal optical data will be the preferred data source for classification (McNairn et al. 2009) because of a better understanding of the relationship between observations and plant phenology (Inglada et al. 2016), valuable information on crop phenology, and also improved accessibility and quality of data.

Another solution to increase the stability of classification algorithms against the high intra-class variability problem is to involve their group information in the classification procedure instead of assuming pixels isolated; this is very useful, especially in images with high spatial resolution. One way to extract information from larger pixel neighbourhoods is to classify images at the pixel level while including a softening process before the classification framework (Ozdarici-Ok, Ok, and Schindler 2015). The Softening hypothesis is based on this base idea that as long as the spatial sampling is sufficiently dense, the neighbouring pixels will tend to have the same class labels (Ozdarici-Ok, Ok, and Schindler 2015). On the other hand, factors such as growth stage, crown structure, cultivation pattern, and soil context not only affect the return signal received by the sensor but also define the structural and contextual features (texture patterns) for each crop (Peña-Barragán et al. 2011), which can help in identifying crop types. To this end, neighbourhoods with spatial dimensions proportionate with the spatial resolution of the sensor are considered, and the structural and contextual features are extracted, which are then fed as input to the classifier. The problem with using this method is that the softening in the boundary areas will cause loss of spatial information and reduce the classification accuracy. To overcome this

problem, a solution based on the ensemble learning of trained classifiers on different sizes of Neighbourhood Windows (NWs) is proposed in this paper. This solution, while using the neighbourhood information, maintains the classification accuracy in the boundary areas.

The present study proposes two learning schemes based on a successful type of Deep Learning (DL) methods, i.e., Stacked Auto-Encoders (SAEs) for crop mapping from multi-temporal multispectral satellite earth observations. In order to increase the stability of the classification scheme against the high intra-class and low inter-class variability problems, both temporal and spatial data enrichments were utilised. The potential of the proposed methods was evaluated in comparison to traditional machine learning methods, namely Linear Support Vector Machine (LSVM) (Scholkopf and Smola 2001), Gaussian SVM (GSVM) (Scholkopf and Smola 2001), and Random Forest (RF) (Liaw and Wiener 2002). In addition, the effects of different train data sampling strategies and different proportions of train data on the classification accuracy were examined. The novelty and specific objectives of the paper could be summarised as follows:

1) Evaluating the potential of SAEs for accurate crop mapping in comparison to conventional classifiers
2) Evaluating the effect of spatial and spatio-temporal data enrichments on the classification accuracy of different methods
3) Proposing an neighbourhood-based method for improving the classification accuracy
4) Proposing an ensemble-based method for improving the classification accuracy
5) Evaluating the effect of different train data sampling strategies on the classification accuracy of different methods

6) Evaluating the effect of train data amount on the classification accuracy of different methods.

2. Related work

Several methods have so far been proposed for the classification of crops using Remote Sensing (RS) imagery. Maximum Likelihood (ML) (Arvor et al. 2011; Ozdarici-Ok, Ok, and Schindler 2015; Salehi, Daneshfar, and Davidson 2017; Waldhoff, Lussem, and Bareth 2017), Decision Trees (DT) (Inglada et al. 2015; Marais Sicre et al. 2016; Massey et al. 2017), RF (Immitzer, Vuolo, and Atzberger 2016; Inglada et al. 2016; Richard et al. 2017; Song et al. 2017; Sonobe et al. 2017; Zhang and Roy 2017; Khosravi, Safari, and Homayouni 2018), and SVMs (Peña and Brenning 2015; Waldhoff, Lussem, and Bareth 2017; Xiong et al. 2017; Niazmardi et al. 2018) are among the most frequently used classification methods. These methods are supervised classifiers based on manually extracted features (appearance descriptors), which are able to describe the information content of the images locally (Volpi and Tuia 2017).

Traditional machine learning methods are highly dependent on the choice of input features (Goodfellow, Bengio, and Courville 2016). Feature extraction and selection requires the user's knowledge and experience, as well as a robust method (Goodfellow, Bengio, and Courville 2016). In particular, most of the algorithms have free parameters that must be optimised along with the selection of input features. However, in RF, this problem is alleviated somewhat because of a feature selection scheme that is included in the algorithm. Nevertheless, none of these conventional methods is trained in an end-to-end manner so that the final system can train the feature extractors and the classifier simultaneously.

DL, as a machine learning-based method, tries to develop a system that could be trained in an end-to-end manner. A set of simultaneously trained feature extractors are

capable of extracting features from the raw data that are optimised for the task. Moreover, DL methods are a specific type of Artificial Neural Networks (ANNs) that provide a high level of potency and flexibility through the representation learning of data in the form of a hierarchy of features. In these methods, the more complicated features are obtained from ones that are more straightforward. Therefore, more abstract representations, in the higher levels, are provided from broader concepts in the lower layers. DL has recently been used in various fields of machine learning and proved its efficiency (Goodfellow, Bengio, and Courville 2016).

Recent progress of DL methods and their increasing application, especially in image processing, have caused them to be taken into consideration in RS research. Recently, extensive studies have investigated the potential of deep networks in resolving different remote sensing problems like image pre-processing (Masi et al. 2016; Yuan, Zheng, and Lu 2017), dimensionality reduction (Zabalza et al. 2016), image classification (Marmanis et al. 2016; Sherrah 2016; Volpi and Tuia 2017; Cheng et al. 2018; Zhou et al. 2019), change detection (El Amin, Liu, and Wang 2016; Gong et al. 2016), image matching (Altwaijry et al. 2016; Mou et al. 2017), target detection and location (Vakalopoulou et al. 2015; Diao et al. 2016; Cheng, Zhou, and Han 2016), navigation (Hudjakov and Tamre 2011; Maturana and Scherer 2015), and 3D analysis (Armagan, Hirzer, and Lepetit 2017; Audebert et al. 2017). Here, we review only the image classification studies.

Marmanis et al. (2016) utilised a large pre-trained Convolutional Neural Network (CNN) on the ImageNet dataset (Deng et al. 2009) for urban scene classification purpose (UC Merced Land Use benchmark). They initially extracted representations using the pre-trained network and then fed them as the inputs to a

smaller supervised CNN classifier. This method led to transfer representation learning obtained from computer vision data to RS imagery.

Sherrah (2016) applied CNNs for pixel-level classification of high-resolution remote sensing data in urban areas (International Society for Photogrammetry and Remote Sensing (ISPRS) Vaihingen and Postdam benchmark data sets). They used a pre-trained CNN and also a Fully Convolutional Neural Network (FCN) which were trained from scratch and obtained state-of-the-art accuracies in both cases.

Volpi and Tuia (2017) proposed an FCN structure like SegNet (Badrinarayanan, Kendall, and Cipolla 2017) which uses sequential convolution and deconvolution layers to provide pixel-level predictions in high-resolution RS imagery. The network was tested on ISPRS Vaihingen and Postdam benchmark data sets and achieved reliable results.

Cheng et al. (2018) proposed optimising discriminative objective function in convolutional networks and named the resulted networks as discriminative CNNs (D-CNNs). The new discriminative objective function was achieved by imposing a metric learning regularisation term on the CNN features. They claimed that D-CNN models are more discriminative in confronting with between-class similarity and within-class diversity problems. The D-CNN models were evaluated on three publicly available benchmark datasets for remote sensing scene classification purpose using three widely used pre-trained CNN models.

Zhou et al. (2019) proposed a two-step framework named Compact and Discriminative Stacked AutoEncoder (CDSAE) for Hyperspectral Image (HSI) classification. The first step was learning feature mappings by training discriminative SAE with a local Fisher discriminant regularisation on each hidden layer. The second step was to learn a classifier and update DSAE with a diversity regularisation over the

hidden neurons and a local Fisher discriminant regularisation on the last feature layer. Fisher's Local Discriminatory Regularisation and Diversity Regularisation were used to learn the maps of discriminating characteristics and to balance the dimensionality of the characteristics and the representational capacity of the characteristics, respectively. With experiments on three HSI public datasets, they demonstrated the effectiveness of their proposed method.

In this paper, we propose two learning schemes based on SAEs for crop mapping from multi-temporal multispectral satellite imagery. The first scheme is to extract stacked spatio-temporal spectral features from all imaging dates and then feeding them to the SAE. The spectral, spatial, and temporal features are utilised together in order to increase the discrimination capability of the algorithms. The second scheme is an ensemble learning method, in which the base classifiers are SAEs trained on different spatial neighbourhood sizes. Performance of the proposed methods was evaluated on an agricultural area in Canada and compared to conventional classifiers, namely SVMs and RF. In addition, the effect of train data sampling strategy and the amount of train data on the classification performance were examined.

3. Materials and methods

3.1. Study area and data

The study area is located in the southwest of the Manitoba state in Canada between latitudes 47° 32′ 16″ N and 48° 12′ 56″ N, and longitudes 97° 5′ 2″ W and 97° 45′ 13″ W. The area is an agricultural region with various annual and perennial crop types, and its spatial extent is about 5689 ha. For the analysis, we used a time series of RapidEye data with original 5 m spatial resolution images. The RapidEye sensors' data consist of five spectral bands (blue, green, red, red-edge, and near-infrared). These images were

initially atmospherically corrected and orthorectified. However, to have a high level of co-registration accuracy, we applied a precise geo-referencing to the time series using a first-order polynomial function. We used four different dates from the atmospherically corrected and orthorectified RapidEye imagery during the 2012 growing season (Figure 1).

Figure 1. about here.

Figure 2. about here.

The ground reference map and the amount of data in each crop class are represented in Figure 2 (*a*) and (*c*), respectively. One example of the training data set is also displayed in this figure (Figure 2 (*b*)). In the classification scheme, seven different crop classes, including corn, pea, canola, soybean, oat, wheat, and broadleaf, were considered. These data were collected to support the Soil Moisture Active-Passive Mission Validation Experiment (SMAPVEX) 2012 campaign of National Aeronautics and Space Administration (NASA) (McNairn et al. 2014).

3.2. Experimental design

The structure of the study is represented in Figure 3. The first strategy is represented in a smaller box as a 'Single classifier,' and the second ensemble-based strategy is represented in the outer box as 'Ensemble strategy.' First, an image data cube is constructed by stacking all the spectral bands of multi-temporal imagery (see Figure 1). This data cube is then pre-processed in order to create a normalised dataset. The normalisation was performed using the maximum and minimum values of each band to create values between -1 and 1.

The ground reference map (Figure 2 (*a*)) is used to generate a train set. Two other inputs to train the data sampling process are patch size (s_p) and train data

percentage. The process identifies positions and class indices of the train set. All other pixels are then considered as a test set, and their positions and class indices are saved.

The next step in the process chain is spatio-temporal feature extraction. This step appends all spectral, spatial, and temporal input features to the train and test data sets for a specified NW and selected dates. The train and test data sets are then used to train and test the classifier. The output of the testing process would be class indices and class probabilities for the test set, as well as the parameters of accuracy analysis (i.e., Overall Accuracy (OA), Kappa coefficient (*K*), and average per-class accuracies).

For the ensemble strategy, other NW sizes (e.g. NW = 1, 3, 5) are then selected to train new classifiers. After training n_{EN} different classifiers with different NWs, we fuse their results in a decision level fusion process in order to create the final fused class probabilities and indices.

Figure 3. about here.

To evaluate the efficiency of the proposed methods, we compared our AE-based classifiers with three traditional classifiers (i.e., LSVM, GSVM, and RF). The process of training and testing these classifiers is the same as mentioned for our first proposed method (single classifier) in the second paragraph of this section, except that the training process is one-step (not pre-training and then fine-tuning) and the class probabilities are not the output of the classifier.

3.2.1. Train data sampling strategies

In this paper, we tried two different stratified sample selection methods based on random points and random patches, respectively. Random points are groups of single pixels, and random patches are groups of rectangular patches having a maximum number of s_p pixels in each direction and all having the same class label. All of the points and patches were selected randomly throughout the entire area. The number of

groups was considered equal to the number of classes. The sampling method based on random points will yield the best results, but in real-world scenarios, it is impractical. Consequently, we will employ both strategies and compare the results in one of the experiments. For the other experiments, we will use the patch-based method as the sampling strategy that is more similar to the real-world conditions.

The reason for considering randomness in the sampling process is to be able to provide different train data and see the effect of such a strategy on the classification performances. Therefore, we evaluated each classifier using ten different random train sets and then compared the results.

The automatic random patch selection method was employed as follows: For each new patch, two numbers were selected randomly, as the coordinates of the first pixel with the position in the upper left corner. Then new pixels were added to the current patch in the row direction until the size of the patch in this direction reached the patch size (s_p), or the class label of newly added pixel changed. Similarly, this process was repeated for the column direction. We then saved the upper left and lower right coordinates of the current patch. The steps, mentioned above, were repeated for each new patch and each class until the total number of samples, considered for each class, was reached. Examples of the selected train samples for the random points and patches with the patch sizes of 10, 20, and 30 pixels are shown in Figure 4.

Figure 4. about here.

3.2.2. Spatio-temporal feature extraction and classification

In order to increase the stability of classification algorithms against the high intra-class variability problem, we involved spatial and temporal information in the classification process. For spatial information, we used two strategies.

Strategy 1. Single classifier: The first strategy was to apply the neighbourhood information of pixels considering an NW size. All pixel values inside an NW around a target pixel *i* are included as its input features in the classification framework.

Strategy 2. Ensemble learning: For the second strategy, we combined the results of classifiers trained with different NWs using an ensemble method. This method is a decision level fusion strategy and is shown in Figure 5, where a case with three NWs of 1, 3, and 5 are displayed (1, 9, and 25 neighbourhoods), and pixel *i* is the one to be classified. As illustrated, in this case, three different AEs will be trained to use three different numbers of input features. For example, if we use all the five spectral bands of single-date images, the number of input features in these three networks would be 5, 45, and 125, respectively (NW × NW × 5). Class probabilities are obtained from the classifiers trained with different NWs. They are then averaged to produce the final ensemble probability (Figure 5). In a case with n_{EN} trained classifiers, the probability of pixel *i* being in class *k* can be computed as Equation (1):

$$\mathrm{P}\left(y^{(i)} = k \mid \boldsymbol{x}^{(i)}\right) = \frac{1}{n_{\mathrm{EN}}} \sum_{j=1}^{n_{\mathrm{EN}}} \mathrm{P}_j\left(y^{(i)} = k \mid \boldsymbol{x}^{(i)}\right), k = 1, 2, \ldots, n_{\mathrm{C}} \quad (1)$$

where, P_j is the obtained probability from the j^{th} classifier, $\boldsymbol{x}^{(i)}$ is the input feature vector for pixel *i*, $y^{(i)}$ is its class label, n_{C} is the total number of classes (here 7, see Figure 2 (*c*)). The final class identification number of pixel *i* ($c^{(i)}$) would be computed using argmax function (Equation (2)):

$$c^{(i)} = \mathrm{argmax}\left(\mathrm{P}\left(y^{(i)} = k \mid \boldsymbol{x}^{(i)}\right)\right) \quad (2)$$

Figure 5. about here.

In Figure 6, the numbers show the contribution time of each pixel in the final classification result for the case displayed in Figure 5. As we see, after applying the ensemble method, pixels near the pixel to be classified will have more effect than far pixels on the classification result. Therefore, we can say that this ensemble strategy would be a weighted scheme for involving neighbourhood information in the classification result, which applies more weights on the near pixels. This is in agreement with the real-world conditions and will reduce the effect of blurriness, especially in the boundary areas.

Figure 6. about here.

To include temporal information, we merely stacked the input features of different dates and fed them as overall input features to the network (a feature level fusion strategy). Consequently, the overall amount of input features will be the number of features in a single-date multiplied by the number of dates. For example, if we use all the five spectral bands of RapidEye imagery as input features in an 'NW = 1' configuration and also use two dates, the total amount of input features will be ten (5 × 2). As a result, the number of input units of the network will be equal to this number.

3.2.3. SAE

Autoencoders: Stacked AEs (SAEs), obtained from the sequences of AEs, is an unsupervised learning algorithm designed in a way that reconstructs its input data in the output. In other words, labels assigned to the input values are set to the input values themselves ($\boldsymbol{y}^{(i)} = \boldsymbol{x}^{(i)}$, for the i^{th} input value). Each AE has two parts: an encoder, which codes the input data as a hidden representation (also called *latent representation*) in a hidden layer; and, a decoder, which decodes the hidden representation to reconstruct the input data in the output layer. An example of AE having four input and output units and three hidden units is represented in Figure 7.

Figure 7. about here.

The forward propagation of an AE in the encoder part (hidden layer) and the decoder part (output layer) is calculated from Equations (3) and (4), respectively.

$$\begin{aligned} \boldsymbol{a}^{(1)} &= \boldsymbol{x} \\ \boldsymbol{z}^{(2)} &= \mathbf{w}^{(1)}\boldsymbol{a}^{(1)} + \boldsymbol{b}^{(1)} \end{aligned} \tag{3}$$

$$\begin{aligned} \boldsymbol{a}^{(2)} &= f\left(\boldsymbol{z}^{(2)}\right) \\ \boldsymbol{z}^{(3)} &= \mathbf{w}^{(2)}\boldsymbol{a}^{(2)} + \boldsymbol{b}^{(2)} \\ h(\boldsymbol{x}; \mathbf{w}, \boldsymbol{b}) &= \boldsymbol{a}^{(3)} = f\left(\boldsymbol{z}^{(3)}\right) \end{aligned} \tag{4}$$

where, $\mathbf{w}^{(1)}$, $\boldsymbol{b}^{(1)}$, $\mathbf{w}^{(2)}$, and $\boldsymbol{b}^{(2)}$ are the weight and bias parameters for the encoder and decoder parts, respectively, *f* is the activation function, *h* is the function that AE learns, and $\boldsymbol{x}$ and $\boldsymbol{a}$ represent inputs and activations, respectively. The superscripts in the parenthesis are layer numbers.

An AE tries to learn a 'h ($\boldsymbol{x}$; $\mathbf{w}$, $\boldsymbol{b}$) is approximately equal to $\boldsymbol{x}$' function, which is, indeed, an approximation of an identity function, such that output $\hat{\boldsymbol{x}}$ be similar to $\boldsymbol{x}$. Although the identity function seems to be a simple function to be learned, by imposing some constraints on the network, interesting structures about the data can be discovered (Ng et al. 2010). Generally, constraints such as limiting the number of hidden units are applied. For example, considering a network with s_1 = 100 input units, which accepts gray values of a 10 × 10 image (100 pixels) in the input, and s_2 = 50 units in the hidden layer (Layer 2 in Figure 8), the network must learn a compressed representation of the input data; in other words, given only the activations of hidden units $\boldsymbol{a}^{(2)} \in \mathbb{R}^{50}$ it must try to reconstruct 100 input pixels (Ng et al. 2010). Even if the number of hidden units is larger than the number of input pixels, it could be possible to recover interesting

structures of data by imposing other constraints on the network. Especially, with imposing sparsity constraints on the hidden units, even with a large number of hidden units, AE could extract structures from the input data (Ng et al. 2010). The sparsity constraint could be applied by imposing a constraint on the average activation value of each hidden unit in the network (Equation (5)):

$$\hat{\rho}_j = \rho \tag{5}$$

where, ρ is the sparsity parameter that usually is selected as a small value near zero, and $\hat{\rho}_j$ is the average activation value of hidden unit j for n_S train samples, and calculated as Equation (6):

$$\widehat{\boldsymbol{\rho}}_j = \frac{1}{n_S}\sum_{i=1}^{n_S}\left[\boldsymbol{a}_j^{(2)}\left(\boldsymbol{x}^{(i)}\right)\right] \tag{6}$$

To establish this constraint, most of the unit activations must be near zero.

Generally, a penalty term is added to the optimisation function to enforce the sparsity constraint, which penalises the sharp difference between the values of $\hat{\rho}_j$ and ρ. The Kullback-Leibler (KL) divergence can be used to this end. In this case, the cost (objective) function will be obtained with Equation (7):

$$J_{\text{sparse}}(\mathbf{w}, \boldsymbol{b}) = J(\mathbf{w}, \boldsymbol{b}) + \beta\sum_{j=1}^{s_2}\text{KL}\left(\rho||\hat{\rho}_j\right), \tag{7}$$

where, $J(\mathbf{w}, \boldsymbol{b})$ is the cost function without sparsity constraint, that for example, could be squared-error or cross-entropy cost function. In this paper, we used a cross-entropy cost function (Equation (8)):

$$J(\mathbf{w},\boldsymbol{b}) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\ln h\left(\boldsymbol{x}^{(i)};\mathbf{w},\boldsymbol{b}\right) + \left(1-\boldsymbol{y}^{(i)}\right)\ln\left(1-h\left(\boldsymbol{x}^{(i)};\mathbf{w},\boldsymbol{b}\right)\right)\right] \quad (8)$$

The Kullback-Leibler (KL) divergence is defined as Equation (9):

$$\mathrm{KL}\left(\rho||\hat{\rho}_j\right) = \rho\log\frac{\rho}{\hat{\rho}_j} + \ (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j} \quad (9)$$

Assuming $\rho = \hat{\rho}_j$, KL divergence (Equation (9)) would be zero, and with distancing of $\hat{\rho}_j$ from ρ, KL will increase.

SAE is a neural network composed of several AEs, which is obtained by sequentially stacking their encoder and decoder parts (Figure 8). Training of SAEs is accomplished in two unsupervised steps: In the first step, all the constructor elements (AEs) are trained separately. This step is, generally, referred to as layer-wise pre-training. In the second step, parameters of the entire network are improved through the back-propagation algorithm. Both training steps are accomplished using unlabelled data.

Figure 8. about here.

In classification applications, the decoder part of the SAE network is eliminated, and a classification layer is added at the end (Figure 9). The last layer of the network is a *Softmax layer* used to generate class probabilities based on the features of the hidden layer of the last AE. Training of a classification SAE also includes two steps of pre-training and fine-tuning, with the difference that in the fine-tuning stage, the network parameters are tuned using labelled train samples. Therefore, in a classification SAE, the pre-training step is a training process that uses unlabelled data (unsupervised), and the fine-tuning step is a training process with labelled data (supervised).

Figure 9. about here.

A Softmax layer gets real values as its inputs (the outputs of the last hidden layer) and generates class probabilities that lie between 0 and 1 and sum to 1 (See Equation (10)):

$$\mathrm{P}\left(y = j \middle| \boldsymbol{x}^{(i)}\right) = \frac{\exp\left(h_j^{(i)}\right)}{\sum_{k=1}^{n_\mathrm{C}} \exp\left(h_k^{(i)}\right)} \tag{10}$$

where, $\mathrm{P}\left(y = j \middle| \boldsymbol{x}^{(i)}\right)$ is the probability of input $\boldsymbol{x}^{(i)}$ being in class *j*, exp is the exponential function, $h_j^{(i)}$ is the output of the last hidden layer (input of Softmax layer) for input $\boldsymbol{x}^{(i)}$ and class *j*, and n_C is the total number of classes.

Generalisation of AEs: The main specification of a machine learning algorithm is its level of generalisation or its performance on new unseen data (Goodfellow, Bengio, and Courville 2016). Many generalisation strategies have been proposed to reduce the test error, possibly at the expense of increased training errors (Goodfellow, Bengio, and Courville 2016). Here, two common forms of generalisation methods, namely L2 *regularisation* and *early stopping*, were used. L2 regularisation strategy, also known as *weight decay*, drives the weights closer to the origin by adding a regularisation term ($t_\mathrm{r} = \frac{1}{2}\|\omega\|_2^2$; where $\|.\|_2$ is the Euclidean norm) to the objective function (Goodfellow, Bengio, and Courville 2016).

The early stopping strategy is one of the most commonly used methods in DL to increase generalisation. One behaviour often observed when training models with sufficient representational capacity is that the training error decreases steadily over time, while the validation set error begins to rise after some iterations (Goodfellow, Bengio, and Courville 2016). Using the parameter set at the iteration with the lowest validation set error, we will have a model with better generalisation capability (lower

test error). However, it is possible that after some iterations of no improvement, the validation set error will decrease again. Instead of running the training algorithm until reaching a (local) minimum of the validation error, we run it until we do not have any improvement in the validation set error for some successive iterations like, for example, 50. Whenever the validation set error improves, a copy of the parameters is stored. After the training algorithm terminates, these parameters are returned rather than the last parameters (Goodfellow, Bengio, and Courville 2016). Early stopping strategy can also be considered as a very efficient hyper-parameter selection algorithm as it can determine the number of epochs required to train the model.

3.2.4. Experiments

We tried different structures for SAE, and finally, an optimum network structure with three layers (input layer, hidden layer, and Softmax layer) and 75 units in the hidden layer was selected. The number of input units differs in each case according to the number of its input features (for example, five in the case of five input spectral bands of single-date image). However, the number of output units was selected as seven equal to the total number of classes. The other hyper-parameters of SAE and traditional classifiers were selected based on 5-fold cross-validation technique. For the SAE, they are the regularisation parameter (ω) and the sparsity parameter (ρ), which were set to 0.004 and 0.15, respectively.

For traditional classifiers, parameter setting was as follows: For LSVM, the kernel function was linear; the kernel scale was selected using a heuristic procedure in each experiment. For GSVM, the kernel function was Gaussian (as its name), and the kernel scale was 2.2. In both LSVM and GSVM, the box constraint was considered as 1; the multi-class classification was performed based on ‘one-vs.-one’ coding and a

standardisation procedure were considered that centres and scales each column of the predictor data (all pixel values for a specified feature) by the column mean and standard deviation. In RF, the maximum number of decision splits was considered as 16314, the number of decision trees and the minimal leaf size were considered as 30 and 1, respectively, and the number of predictors selected at random for every decision split was considered as the square root of the number of predictors (input features) for classification.

The experiments will cover evaluating the effects of different sampling strategies, spatial accuracy improvement, spatio-temporal accuracy improvement, and amount of train data on the classification performance. For all experiments, all five spectral bands of RapidEye imagery were used as input features. All experiments except the last one were done with ten different sets of train and test data samples created randomly. In the last experiment, only one sample data set was used. We also examined what classification accuracy could be obtained if the red-edge channel is not available considering the setting of the last experiment. To have a fair comparison, all the experiments were implemented using MATLAB v2017.b on a Desktop PC with Intel Corei7 CPU 3.06 GHz, 12 GB RAM. Explanation and setting of the experiments performed in this study are as follows:

Experiment 1: Different sampling strategies: In the first experiment, we examined the effect of each sampling method (random points and random patches) on the performance of different classifiers. To this end, we considered only 1 pixel neighbourhoods (i.e. NW = 1) in the feature extraction step. In the patch-based strategy, three values of 10, 20, and 30 pixels were tested as patch size (s_p). Each sampling case was accomplished ten different times using 5.0% of the ground reference data as the

train data. Patterns of the selected samples in each case are displayed in Figure 4 for the first run. The input features were five spectral bands of the image data of date 14 July.

Experiment 2: Different spatial feature extraction and classification strategies: This experiment evaluated the effect of the two proposed spatial feature extraction strategies (single classifier and the ensemble method) on the classification performance. For the first strategy, we compared results obtained by three different NW sizes of 1, 3 and 5 pixels. For the second one, we tested three possibilities of ensemble learning using these three NW values, i.e. EN (1, 3), EN (3, 5), and EN (1, 3, 5). Other NWs, like 7 and 9 pixels, were also examined, but the results were less accurate than what mentioned above. Hence we did not include them in the manuscript. Here, the data sampling method was a patch-based strategy with a patch size (s_p) of 10 pixels, which ran ten times. Like the previous experiment, 5.0% of the ground truth data was considered as the train data and the rest as the test data. In this experiment, we used the five spectral bands of image data acquired on 5 July as input features, since it has been collected in the middle of the growing season.

Experiment 3: Different spatio-temporal feature extraction and classification settings: In this experiment, the effect of spatio-temporal accuracy improvement was analysed. To this end, we used the time series of four image data represented in Figure 1 as input data. The other settings were the same as the second experiment.

Experiment 4. Different train data proportions: To evaluate how the amount of train data will affect different classifiers' performances, we considered eight different proportions of train data from 0.5% to 70.0%. We tested both of the proposed spatio-temporal feature extraction strategies in this experiment. For the first single classifier

strategy, three NWs of 1, 3, and 5 pixels were assumed. For the second ensemble-based method, three cases of EN (1, 3), EN (3, 5), and EN (1, 3, 5) were tested. All other settings were also like the same as the third experiment except that this experiment is based on just one train data set for each proportion (shown in Figure 10). The reason is that this experiment was very time consuming and could not be completed in a reasonable time if repeated ten different times. Moreover, we performed a test to see what would be obtained if the red-edge channel is removed from the input data.

Figure 10. about here.

4. Results and Discussion

4.1. Different sampling strategies

The classification accuracy decreases by increasing the patch sizes in all classifiers (Table 1). The GSVM and AE methods reached the highest accuracies in all the four cases with a slight difference from each other. The Kappa coefficients showed similar patterns. With increasing the patch sizes (s_p), the standard deviations increase too (Table 1). This is represented with more details in Figure 11, which shows OAs in ten runs of four train sample selection cases (s_p = 1 for the random points, s_p = 10, 20, and 30 for the random patches) and their mean values. It shows that the accsuracy will be more dependent on the selected train samples if we use larger patch sizes. Therefore, we choose the patch-based strategy with s_p = 10 for train sample selection in the following experiments, which is more similar to the real-world conditions (In the real conditions we cannot easily provide random train samples with the desired number) and has minimum changes in different runs.

Table 1. about here.

Figure 11. about here.

Different spatial feature extraction and classification strategies

As shown in Table 2, the spatial data enrichment strategy (NW > 1) improves the classification accuracies considerably in all classifiers when using single-date data. The amount of OA improvement in the case of 'NW = 3' is 7.82, 1.87, 9.17, and 7.27% for LSVM, GSVM, RF, and AE, respectively. In the experiment with 'NW = 5' configuration, also improvements to the 'NW = 1' case are observed in all classifiers except GSVM, but the accuracy is less than what was achieved with 'NW = 3' configuration. To this point, RF reached the maximum performance, which was obtained in the 'NW = 3' configuration. For this classifier, the OA and *K* were 94.05% and 0.93, respectively. For the AE classifier, we applied ensemble strategy considering three different configurations: 'EN (1, 3)', 'EN (3, 5)', and 'EN (1, 3, 5)'. The third configuration (i.e. 'EN (1, 3, 5)') reached the maximum accuracy among all the other methods and configurations (OA = 94.75%, and *K* = 0.94).

Table 2. about here.

The spatial enrichment strategy has improved the per-class accuracies and the average per-class accuracies in all classifiers but GSVM (Table 3). The improvements in the average per-class accuracy for the first strategy are 14.89, 15.50, and 12.75% for LSVM, RF, and AE, respectively, which were obtained in the 'NW = 3' configuration. For the ensemble strategy, the amount of this improvement is 13.95 for AE in the 'EN (1, 3, 5)' configuration, which is higher than its equivalent value in the first strategy (i.e., 12.75). Therefore, we conclude that the two proposed spatial enrichment methods have improved the classification performance in terms of average per-class accuracy.

The highest average per-class accuracy is for the proposed method of 'AE, EN (1, 3, 5)', which is 90.26%. Among the non-ensemble methods, RF classifier with the spatial configuration of 'NW = 3' reached the maximum average per-class accuracy of

90.22%. Comparing the per-class accuracies, we see that for oat and broadleaf, the performance of the proposed AE-based classifier was higher than that of RF for classifying these classes, with a considerable margin (83.66% vs 79.19% and 73.84% vs 72.10%, respectively). For the class pea, the condition is reversed (i.e., the per-class accuracy obtained from RF for this class (93.47%) is much higher than its equivalent value obtained from the AE-based classifier (87.00%)).

Table 3. about here.

Different spatio-temporal feature extraction and classification settings

By comparing Table 4 with the results represented in Table 2, one can observe the effect of temporal data enrichment on the classification performance. As can be seen, the accuracies obtained from the configurations of all classifiers (except 'GSVM, NW = 3') have been improved using multi-temporal data.

Table 4 also shows the effect of spatial data enrichment on the classification performance in the case of multi-temporal data. As we see, like the single-date experiment (i.e., the previous experiment), the first spatial data enrichment strategy has improved the classification performance in all classifiers except GSVM. Moreover, the improvements from the second strategy (i.e., the ensemble method) are even more significant. This also could be concluded by comparing the accuracies presented in Table 5, which shows that per-class accuracies and average per-class accuracies have also been improved using the proposed spatial enrichment strategies.

In the previous section, we saw that the experiment with the 'NW = 3' configuration obtained better results than the 'NW = 5' setting. Consequently, we considered 'NW = 5' configuration just for the AE classifier, which then was used in the ensemble configurations. Using the first spatial data enrichment strategy, RF and AE reached the maximum accuracies in the 'NW = 3' configuration among all

classifiers and settings. In this case, the OAs of 94.92 and 94.26% and the *K*s of 0.94 and 0.93 were obtained for these two classifiers, respectively. For the AE classifier, we applied the ensemble strategy considering three different configurations: 'EN (1, 3)', 'EN (3, 5)', and 'EN (1, 3, 5)'. As shown in Table 4, the third configuration reached the maximum accuracies compared to all other methods and configurations (OA = 95.26%, and $K = 0.94$). Therefore, the proposed ensemble-based SAE reached maximum performance in the case of multi-temporal data, too.

Table 4. about here.

Table 5 reveals that the highest average per-class accuracy is for the proposed method of 'AE, EN (1, 3, 5)', which is 92.16%. This method also obtained the highest per-class accuracy for the classes of oat and broadleaf again with a considerable margin from their equivalent values obtained from RF (85.38% vs 79.98% and 82.81% vs 74.54%, respectively). Among the non-ensemble methods, the RF classifier with the configuration of 'NW = 3' reached a maximum average per-class accuracy of 91.35%. This classifier obtained the highest per-class accuracy for the class of pea again with a considerable difference from the proposed AE (94.68% vs 87.97%), which is in agreement with what we found in the case of single-date data (previous experiment).

The reason for this fixed behaviour could be explained by comparing our findings with the ground reference data (Figure 2). As shown, the geometric shapes of the broadleaf and oat fields are more complicated than that of other fields (See Figure 2 (*a*)). Our proposed ensemble-based method ('AE, EN (1, 3, 5)') has worked much better in these geometrically complex fields. Because this is an advantage of this method that while using the neighbourhood information, it maintains the classification accuracy in the boundary areas, on the other hand, class pea has very small samples to create train data (only 1.10% of samples, Figure 2 (*c*)). This is why our AE-based classifier reached

a low accuracy for this class comparing to the RF classifier because the AE-based classifiers cannot reach high accuracies when a minimal number of train data are available.

Table 5. about here.

We selected one dataset from these ten train datasets in order to show the classification maps obtained from different configurations in the experiment. To this end, we selected a dataset that its results had a maximum likelihood of the mean values of all data. Here, the selection was based on the comparison of the trends of average class accuracy changes between different configurations in each dataset with those of mean values. For this comparison, we considered just three configurations with the highest values (i.e., 'AE, EN (1, 3, 5)', 'RF, NW = 3', and 'LSVM, NW = 3'). The dataset that has maximum similarity in the trend of accuracy change between different classifiers to that of mean values was the best choice. According to the comparison results, the train dataset No.3 was selected since it has the highest likelihood of the mean results obtained among the ten datasets.

Classification maps obtained using the third data set (displayed in Figure 2 (*b*)) are represented in Figure 12. We also compared the OA and *K* values of different classifiers based on this dataset. The highest OA and *K* values are for the proposed 'AE, EN (1, 3, 5)' method (95.55% and 0.95, respectively). Per-class accuracies (%) and average per-class accuracies (%) are also displayed in Table 6 for this dataset. As shown, the maximum average per-class accuracy has also been obtained with the proposed 'AE, EN (1, 3, 5)' method with a value of 93.09%. Similar to what was obtained with comparing the mean values represented in Table 5, the result of using the third train dataset also shows that the RF classifier with the 'NW = 3' configuration reached the best average per-class accuracy among the non-ensemble methods. It is to

be noted that again the oat and broadleaf classes reached higher accuracies in the proposed AE-based classifier than RF. Moreover, the reverse situation here is true for the pea class again. This is in agreement with what we concluded earlier on the complexity of the geometric forms of the crop fields and the number of training data.

Table 6. about here.

Figure 12. about here.

Different proportions of train data

Classification accuracies obtained from different train data proportions are shown for the five best classifiers in Figure 13 to a better comparison. As can be seen, the accuracies of all classifiers have improved significantly by adding train data from 0.5% to 5.0%. However, adding more data after that has improved accuracy in all classifiers but with a slow trend. This motivates us to decide on how much train data should be gathered to get proper classification results. As we saw in the first experiment on train data sampling strategies (section 3.2.1), the best sampling was a patch-based strategy using 10 pixel patches. Using a 10 pixel patch size and 5.0% train data proportion as the base, we can count the number of patches sampled for each crop type and get an indicator for train data requirement in the form of the number of patches per X ha. By using the sample shown in Figure 4 (*b*) and the maximum number of patches between all the crop types (42 for canola), this indicator was calculated as 7 patches per 1000 ha.

The accuracy comparison between different classifiers shows that AE classifier with the ensemble configuration of 'EN (1, 3, 5)' reaches the best performance among all classifiers when the training data is equal to or more than 5.0%. The AE and RF classifiers with the NW size of 3 pixels and the GSVM classifier with the NW size of 1 pixel are in the next positions.

Figure 13. about here.

When there is a little amount of train data (0.5%), the LSVM classifier with no spatial data improvement strategy ('LSVM, NW = 1') reaches the maximum performance (Table 7). In the case of 1.0% train data, similar conditions are true for the RF classifier. When we have 5.0% or more train data, the AE-based classifiers gain better results than other classifiers; the ensemble configuration of 'AE, EN (1, 3, 5)' shows the best results, and among the non-ensemble methods, AE with NW size of 3 pixels has the best performance. It is worth noting that the RF with 'NW = 3' configuration achieved results close to the AE with 'NW = 3'.

Table 8 shows that with a meagre amount of train data (e.g., 0.5% and 1.0%), the LSVM classifier obtains the best results. After increasing the amount of train data equal to or more than 5.0%, the AE classifier with ensemble strategies reaches to the best performances. Regarding the non-ensemble strategies, AE and RF classifiers with the NW size of 3 pixels gain better results than other methods. These results are in agreement with what was concluded by comparing the OA and *K* values in the previous paragraph.

Table 7. about here.

Table 8. about here.

Figure 14 indicates the total calculation time for the five best-selected classifiers for a better comparison. The first column in Figure 14 shows the calculation times for all the train data proportions (0.5, 1.0, 5.0, 10.0, 20.0, 30.0, 50.0, and 70.0%), while the second column shows the calculation times for the train data proportions equal to or less than 10.0 (0.5, 1.0, 5.0, and 10.0%) in order to get a closer view. Table 9 shows that with the train data proportions less than or equal to 1.0%, the LSVM was the fastest method. While when using train data proportions more than or equal to 5.0%, the RF was the fastest. According to Table 9, the classification methods can be sorted based on

their cumulative calculation time from 0.5% to 70.0% as: ‘RF, NW = 1’, ‘RF, NW = 3’, ‘GSVM, NW = 1’, ‘LSVM, NW = 3’, ‘LSVM, NW = 1’, ‘AE, NW = 3’, ‘AE, NW = 1’, ‘AE, NW = 5’, ‘AE, EN (1, 3)’, ‘AE, EN (3, 5)’, ‘AE, EN (1, 3, 5)’, and ‘GSVM, NW = 3’.

Figure 14. about here.

Table 9. about here.

In order to see what would be happening if the imaging sensor does not have red-edge band, we also examined a case by removing this band from the input data using the settings of this section, i.e., using all four-date imageries and the patch size of 10 pixels. Train data proportion and NW size were considered as 5.0% and NW = 1, respectively. The OA (%) and *K* pairs of (92.58, 0.91), (94.10, 0.93), (94.56, 0.93), and (94.44, 0.93) were obtained for the methods of LSVM, GSVM, RF, and AE, respectively. These values are comparable with the case that used all five spectral bands of RapidEye data displayed in the first four rows of the 6^{th} and 7^{th} columns of Table 7. Accordingly, the methods and results presented in this study could also be applied in a broader spectrum of image data that may not have a red-edge band.

Conclusions

In this study, two SAE-based learning schemes were proposed for spatio-temporal crop mapping and applied on the multi-temporal multispectral RapidEye imagery. The experiments showed that involving spatial information in the classification scheme, i.e., NW > 1, improves the classification performance considerably when using a single-date image. The best performance observed was with ‘AE, EN (1, 3, 5)’ with the OA of 94.75%, the Kappa coefficient of 0.94, and the average class accuracy of 90.26%.

In the experiments on multi-temporal data, involving spatial information in the classification scheme improved the final results in all classifiers but GSVM. However,

these improvements were less than those in the single-date setting because using multi-temporal data has improved the performance by itself. The best performance was with 'AE, EN (1, 3, 5)' with the OA of 95.26%, the *K* of 0.94 and the average class accuracy of 92.16%.

Experiments on different train data sampling strategies showed that the classification accuracy would be more dependent on the sampled data if we use larger patch sizes. The best choice was a 10 pixel patch-based strategy, which we used in our next experiments for train data sampling. Experiments on different train data proportions showed that the accuracy of classification significantly improves by adding train data from 0.5% to 5.0%. However, beyond this proportion, the accuracy improvements in all classifiers follow smoother trends. Based on our experiments, gathering 7 patches per 1000 ha (a 10 pixel patch size almost equivalent to 0.5% train data proportion) is an excellent indication for field-work required to train data collection in homogenous agricultural areas. The best performance was with the 'AE, EN (1, 3, 5)' when the train data was equal to or more than 5.0%. With a meagre amount of train data (less than 5.0%), the LSVM classifier obtained the best results. Comparing the computation times showed that among the classifiers with the best performance, RF with 'NW = 3' was the fastest, and the 'AE, EN (1, 3, 5)' was the slowest method.

Briefly, according to the present research findings, AEs and RFs are two reliable classifiers in crop mapping, and their performance could be improved by involving spatial and temporal information in the classification. We strongly recommend always using NW > 1 to do crop mapping especifically when high spatial resolution image data are available. We could also improve the performance of AEs using an ensemble strategy to use spatial neighbourhood information in a weighted scheme. Though the ensemble-based AE obtained slightly better performance, the RFs reached a much

higher speed. This proves that not everything needs deep learning to perform well. Moreover, geometrical complexity of fields and number of train data in each class are two crucial factors, which influence the classification results. In fields with high geometrical complexity, ensemble-based AE shows better performance, and in classes with a minimal number of train data, RF would generate better results. This motivates us to use our ensemble-based AE in areas with more geometrical complexity like humanmade urban areas to see its performance as our future work, network structure (including the number of layers and units in each layer), and NW size would be different. In addition, we will examine the potential of other types of DL methods such as CNNs and Recursive Neural Networks (RNNs) in crop mapping and ways to improve the performance in terms of calculation time and final classification accuracy.

Acknowledgement:

We would like to thank Dr. Heather McNairn, Ottawa Research and Development Centre of the Agriculture and Agri-food Canada, for providing both remote sensing and reference data used in this research.

References

Altwaijry, Hani, Eduard Trulls, James Hays, Pascal Fua, and Serge Belongie. 2016. Learning to match aerial images with deep attentive architectures. Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Armagan, Anil, Martin Hirzer, and Vincent Lepetit. 2017. Semantic segmentation for 3D localization in urban environments. Paper presented at the Urban Remote Sensing Event (JURSE), 2017 Joint.

Arvor, Damien, Milton Jonathan, Margareth Simões Penello Meirelles, Vincent Dubreuil, and Laurent Durieux. 2011. "Classification of MODIS EVI time series for crop mapping in the state of Mato Grosso, Brazil." *International Journal of Remote Sensing* 32 (22):7847-71.

Audebert, Nicolas, Alexandre Boulch, Hicham Randrianarivo, Bertrand Le Saux, Marin Ferecatu, Sébastien Lefevre, and Renaud Marlet. 2017. Deep learning for urban remote sensing. Paper presented at the Urban Remote Sensing Event (JURSE), 2017 Joint.

Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. 2017. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." *IEEE transactions on pattern analysis and machine intelligence* 39 (12):2481-95.

Bargiel, Damian. 2017. "A new method for crop classification combining time series of radar images and crop phenology information." *Remote sensing of Environment* 198:369-83.
Bargiel, Damian, and Sylvia Herrmann. 2011. "Multi-temporal land-cover classification of agricultural areas in two European regions with high resolution spotlight TerraSAR-X data." *Remote Sensing* 3 (5):859-77.
Cheng, Gong, Ceyuan Yang, Xiwen Yao, Lei Guo, and Junwei Han. 2018. "When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs." *IEEE Transactions on Geoscience and Remote Sensing* 56 (5):2811-21.
Cheng, Gong, Peicheng Zhou, and Junwei Han. 2016. "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images." *IEEE Transactions on Geoscience and Remote Sensing* 54 (12):7405-15.
Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. Paper presented at the 2009 IEEE conference on computer vision and pattern recognition.
Diao, Wenhui, Xian Sun, Xinwei Zheng, Fangzheng Dou, Hongqi Wang, and Kun Fu. 2016. "Efficient saliency-based object detection in remote sensing images using deep belief networks." *IEEE Geoscience and Remote Sensing Letters* 13 (2):137-41.
El Amin, Arabi Mohammed, Qingjie Liu, and Yunhong Wang. 2016. Convolutional neural network features based change detection in satellite images. Paper presented at the First International Workshop on Pattern Recognition.
Foody, GM, MB McCulloch, and WB Yates. 1994. "Crop classification from C-band polarimetric radar data." *International Journal of Remote Sensing* 15 (14):2871-85.
Gong, Maoguo, Jiaojiao Zhao, Jia Liu, Qiguang Miao, and Licheng Jiao. 2016. "Change detection in synthetic aperture radar images based on deep neural networks." *IEEE transactions on neural networks and learning systems* 27 (1):125-38.
Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*: MIT Press. http://www.deeplearningbook.org.
Hudjakov, Robert, and Mart Tamre. 2011. "Ortophoto analysis for UGV long-range autonomous navigation." *Estonian Journal of Engineering* 17 (1):17-27.
Immitzer, Markus, Francesco Vuolo, and Clement Atzberger. 2016. "First experience with Sentinel-2 data for crop and tree species classifications in central Europe." *Remote Sensing* 8 (3):166.
Inglada, Jordi, Marcela Arias, Benjamin Tardy, Olivier Hagolle, Silvia Valero, David Morin, Gérard Dedieu, Guadalupe Sepulcre, Sophie Bontemps, and Pierre Defourny. 2015. "Assessment of an operational system for crop type map production using high temporal and spatial resolution satellite optical imagery." *Remote Sensing* 7 (9):12356-79.
Inglada, Jordi, Arthur Vincent, Marcela Arias, and Claire Marais-Sicre. 2016. "Improved early crop type identification by joint use of high temporal resolution SAR and optical image time series." *Remote Sensing* 8 (5):362.
Kenduiywo, Benson Kipkemboi, Damian Bargiel, and Uwe Soergel. 2018. "Crop-type mapping from a sequence of Sentinel 1 images." *International Journal of Remote Sensing*:1-22.
Khosravi, Iman, Abdolreza Safari, and Saeid Homayouni. 2018. "MSMD: maximum separability and minimum dependency feature selection for cropland classification from optical and radar data." *International Journal of Remote Sensing* 39 (8):2159-76.
Kussul, Nataliia, Sergii Skakun, Andrii Shelestov, Mykola Lavreniuk, B Yailymov, and Olga Kussul. 2015. "Regional scale crop mapping using multi-temporal satellite imagery." *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 40 (7):45.
Liaw, Andy, and Matthew Wiener. 2002. "Classification and regression by randomForest." *R news* 2 (3):18-22.
Marais Sicre, Claire, Jordi Inglada, Rémy Fieuzal, Frédéric Baup, Silvia Valero, Jérôme Cros, Mireille Huc, and Valérie Demarez. 2016. "Early detection of summer crops using high spatial resolution optical image time series." *Remote Sensing* 8 (7):591.

Marmanis, Dimitrios, Mihai Datcu, Thomas Esch, and Uwe Stilla. 2016. "Deep learning earth observation classification using ImageNet pretrained networks." *IEEE Geoscience and Remote Sensing Letters* 13 (1):105-9.

Masi, Giuseppe, Davide Cozzolino, Luisa Verdoliva, and Giuseppe Scarpa. 2016. "Pansharpening by convolutional neural networks." *Remote Sensing* 8 (7):594.

Massey, Richard, Temuulen T Sankey, Russell G Congalton, Kamini Yadav, Prasad S Thenkabail, Mutlu Ozdogan, and Andrew J Sánchez Meador. 2017. "MODIS phenology-derived, multi-year distribution of conterminous US crop types." In *Remote sensing of Environment*, 490-503.

Maturana, Daniel, and Sebastian Scherer. 2015. 3d convolutional neural networks for landing zone detection from lidar. Paper presented at the Robotics and Automation (ICRA), 2015 IEEE International Conference on.

McNairn, H, J Ellis, JJ Van Der Sanden, T Hirose, and RJ Brown. 2002. "Providing crop information using RADARSAT-1 and satellite optical imagery." *International Journal of Remote Sensing* 23 (5):851-70.

McNairn, Heather, Catherine Champagne, Jiali Shang, Delmar Holmstrom, and Gordon Reichert. 2009. "Integration of optical and Synthetic Aperture Radar (SAR) imagery for delivering operational annual crop inventories." *ISPRS Journal of Photogrammetry and Remote Sensing* 64 (5):434-49.

McNairn, Heather, Thomas J Jackson, Grant Wiseman, Stephane Belair, Aaron Berg, Paul Bullock, Andreas Colliander, Michael H Cosh, Seung-Bum Kim, and Ramata Magagi. 2014. "The soil moisture active passive validation experiment 2012 (SMAPVEX12): Prelaunch calibration and validation of the SMAP soil moisture algorithms." *IEEE Transactions on Geoscience and Remote Sensing* 53 (5):2784-801.

Mou, Lichao, Michael Schmitt, Yuanyuan Wang, and Xiao Xiang Zhu. 2017. A CNN for the identification of corresponding patches in SAR and optical imagery of urban scenes. Paper presented at the Urban Remote Sensing Event (JURSE), 2017 Joint.

Ng, Andrew, Jiquan Ngiam, Chuan Yu Foo, Yifan Mai, and Caroline Suen. 2010. *UFLDL tutorial*: Computer Science Department, Stanford University. http://deeplearning.stanford.edu/tutorial/.

Niazmardi, Saeid, Saeid Homayouni, Abdolreza Safari, Jiali Shang, and Heather McNairn. 2018. "Multiple kernel representation and classification of multivariate satellite-image time-series for crop mapping." *International Journal of Remote Sensing* 39 (1):149-68.

Ozdarici-Ok, Asli, Ali Ozgun Ok, and Konrad Schindler. 2015. "Mapping of agricultural crops from single high-resolution multispectral images—Data-driven smoothing vs. parcel-based smoothing." *Remote Sensing* 7 (5):5611-38.

Peña-Barragán, José M, Moffatt K Ngugi, Richard E Plant, and Johan Six. 2011. "Object-based crop identification using multiple vegetation indices, textural features and crop phenology." *Remote sensing of Environment* 115 (6):1301-16.

Peña, MA, and A Brenning. 2015. "Assessing fruit-tree crop classification from Landsat-8 time series for the Maipo Valley, Chile." *Remote sensing of Environment* 171:234-44.

Richard, Kyalo, Elfatih M Abdel-Rahman, Sevgan Subramanian, Johnson O Nyasani, Michael Thiel, Hosein Jozani, Christian Borgemeister, and Tobias Landmann. 2017. "Maize Cropping Systems Mapping Using RapidEye Observations in Agro-Ecological Landscapes in Kenya." *Sensors* 17 (11):2537.

Salehi, Bahram, Bahram Daneshfar, and Andrew M Davidson. 2017. "Accurate crop-type classification using multi-temporal optical and multi-polarization SAR data in an object-based image analysis framework." *International Journal of Remote Sensing* 38 (14):4130-55.

Scholkopf, Bernhard, and Alexander J Smola. 2001. *Learning with kernels: support vector machines, regularization, optimization, and beyond*: MIT press.

Sherrah, Jamie. 2016. "Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery." *arXiv preprint arXiv:1606.02585*.

Skakun, Sergii, Belen Franch, Eric Vermote, Jean-Claude Roger, Inbal Becker-Reshef, Christopher Justice, and Nataliia Kussul. 2017. "Early season large-area winter crop

mapping using MODIS NDVI data, growing degree days information and a Gaussian mixture model." *Remote sensing of Environment* 195:244-58.

Song, Qian, Qiong Hu, Qingbo Zhou, Ciara Hovis, Mingtao Xiang, Huajun Tang, and Wenbin Wu. 2017. "In-Season Crop Mapping with GF-1/WFV Data by Combining Object-Based Image Analysis and Random Forest." *Remote Sensing* 9 (11):1184.

Sonobe, Rei, Yuki Yamaya, Hiroshi Tani, Xiufeng Wang, Nobuyuki Kobayashi, and Kan-ichiro Mochizuki. 2017. "Mapping crop cover using multi-temporal Landsat 8 OLI imagery." *International Journal of Remote Sensing* 38 (15):4348-61.

Vakalopoulou, Maria, Konstantinos Karantzalos, Nikos Komodakis, and Nikos Paragios. 2015. Building detection in very high resolution multispectral data with deep learning features. Paper presented at the Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International.

Volpi, Michele, and Devis Tuia. 2017. "Dense semantic labeling of subdecimeter resolution images with convolutional neural networks." *IEEE Transactions on Geoscience and Remote Sensing* 55 (2):881-93.

Waldhoff, Guido, Ulrike Lussem, and Georg Bareth. 2017. "Multi-data approach for remote sensing-based regional crop rotation mapping: a case study for the Rur catchment, Germany." *International Journal of Applied Earth Observation and Geoinformation* 61:55-69.

Whelen, Tracy, and Paul Siqueira. 2017. "Use of time-series L-band UAVSAR data for the classification of agricultural fields in the San Joaquin Valley." *Remote sensing of Environment* 193:216-24.

Xiong, Jun, Prasad S Thenkabail, James C Tilton, Murali K Gumma, Pardhasaradhi Teluguntla, Adam Oliphant, Russell G Congalton, Kamini Yadav, and Noel Gorelick. 2017. "Nominal 30-m cropland extent map of continental Africa by integrating pixel-based and object-based algorithms using sentinel-2 and Landsat-8 data on Google earth engine." *Remote Sensing* 9 (10):1065.

Yuan, Yuan, Xiangtao Zheng, and Xiaoqiang Lu. 2017. "Hyperspectral Image Superresolution by Transfer Learning." *IEEE Journal of selected topics in applied earth observations and remote sensing* 10 (5):1963-74.

Zabalza, Jaime, Jinchang Ren, Jiangbin Zheng, Huimin Zhao, Chunmei Qing, Zhijing Yang, Peijun Du, and Stephen Marshall. 2016. "Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging." *Neurocomputing* 185:1-10.

Zhang, Hankui K, and David P Roy. 2017. "Using the 500 m MODIS land cover product to derive a consistent continental scale 30 m Landsat land cover classification." *Remote sensing of Environment* 197:15-34.

Zhou, Peicheng, Junwei Han, Gong Cheng, and Baochang Zhang. 2019. "Learning compact and discriminative stacked autoencoder for hyperspectral image classification." *IEEE Transactions on Geoscience and Remote Sensing* 57 (7):4823-33.

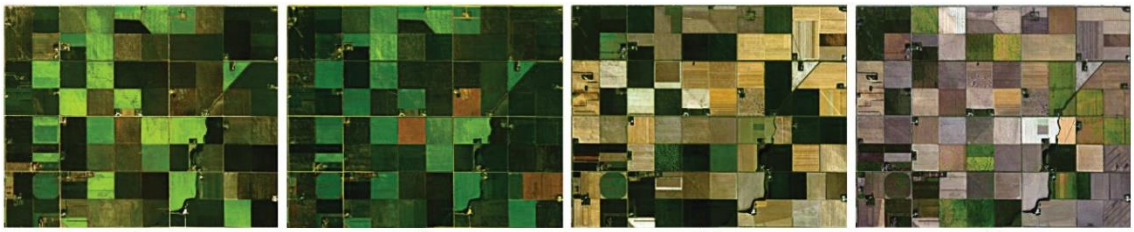
List of Figures' captions

Figure 13. Classification accuracy obtained from different classifiers using different train data percentages; (*a*) Best OA (%), (*b*) Best Kappa coefficient, and (*c*) Best average class accuracy (%)

Figure 14. Training and test time (in seconds) of different classifiers using different train data proportions; (*a*) train data proportion = 0.5, 1.0, 5.0, 10.0, 20.0, 30.0, 50.0, 70.0%; (*b*) train data proportion = 0.5, 1.0, 5.0, 10.0%)

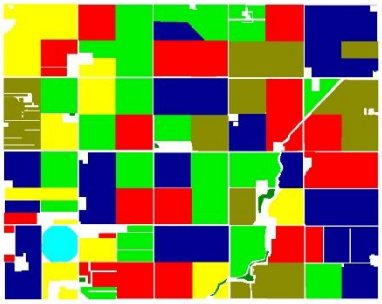
List of Tables' captions

Figure 1.

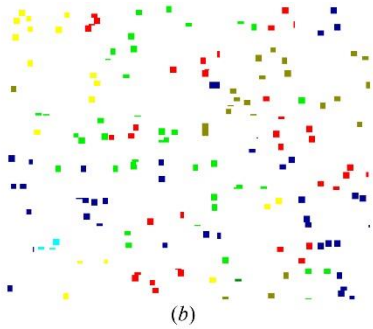


(*a*) (*b*) (*c*) (*d*)

Figure 2



(*a*)



(*b*)

ID	Colour	Class	No. of samples	Proportion of samples (%)
1		Corn	39157	12.02
2		Pea	3598	1.10
3		Canola	75671	23.23
4		Soybean	74042	22.73
5		Oats	47117	14.46
6		Wheat	85074	26.11
7		Broadleaf	1143	0.35

(*c*)

Figure 3.

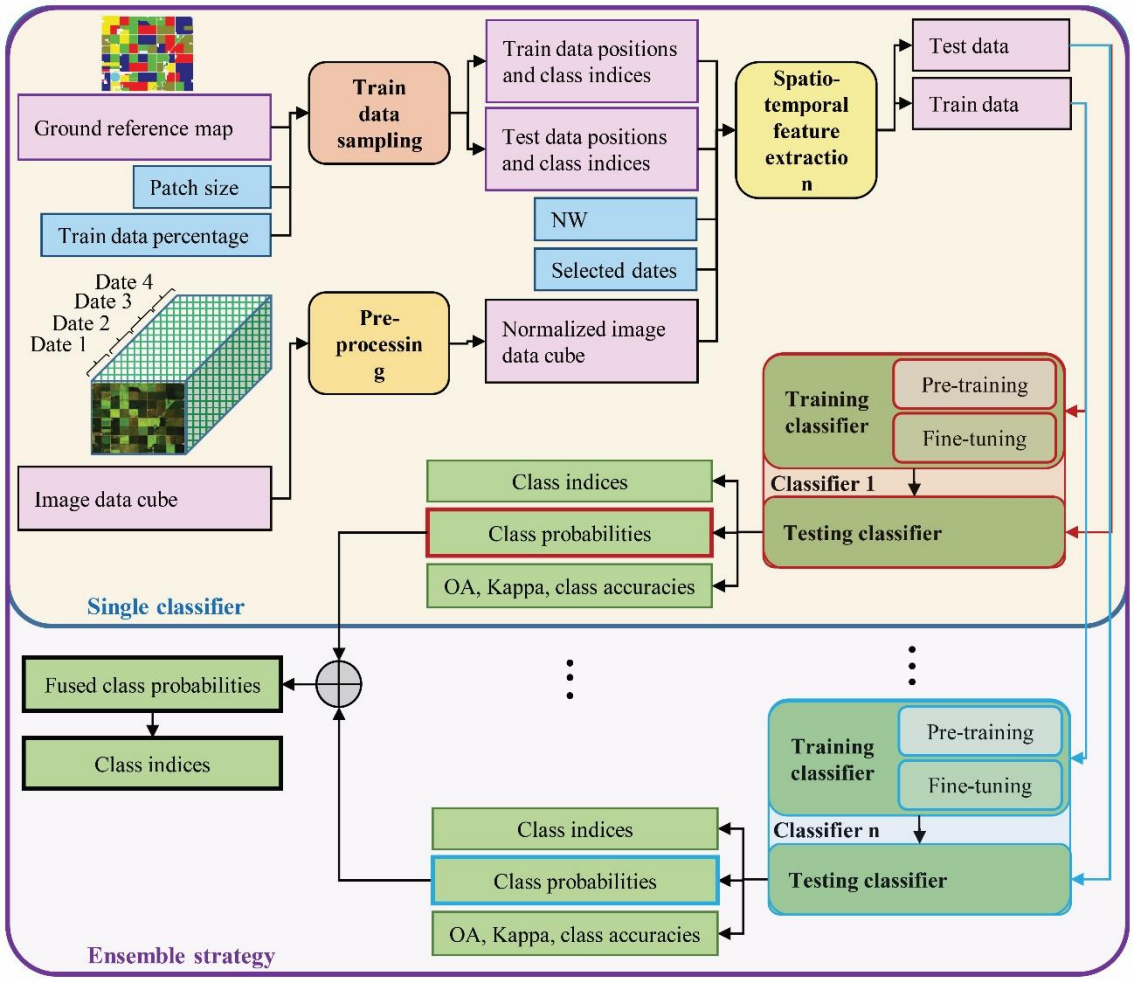

Ground reference map
Patch size
Train data percentage
Train data sampling
Train data positions and class indices
Test data positions and class indices
NW
Selected dates
Spatio-temporal feature extraction
Test data
Train data
Date 4
Date 3
Date 2
Date 1
Image data cube
Pre-processing
Normalized image data cube
Training classifier
Pre-training
Fine-tuning
Classifier 1
Testing classifier
Class indices
Class probabilities
OA, Kappa, class accuracies
Single classifier
Fused class probabilities
Class indices
Training classifier
Pre-training
Fine-tuning
Classifier n
Testing classifier
Class indices
Class probabilities
OA, Kappa, class accuracies
Ensemble strategy

Figure 4.

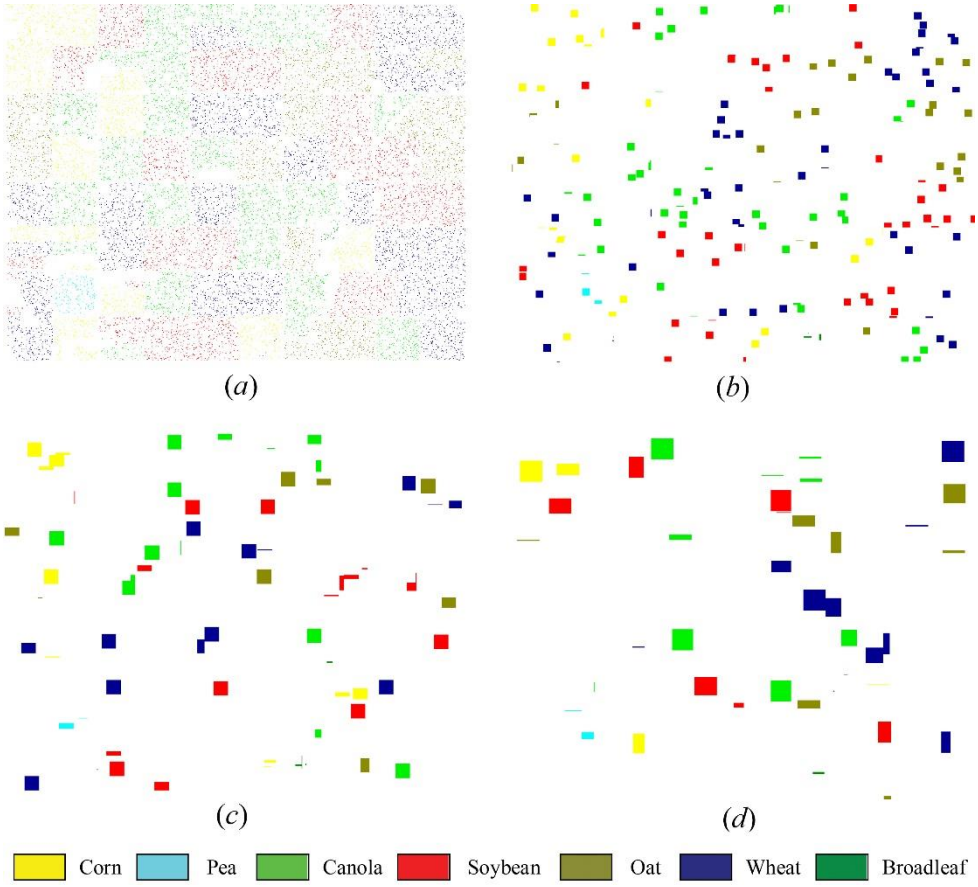


Figure 5.

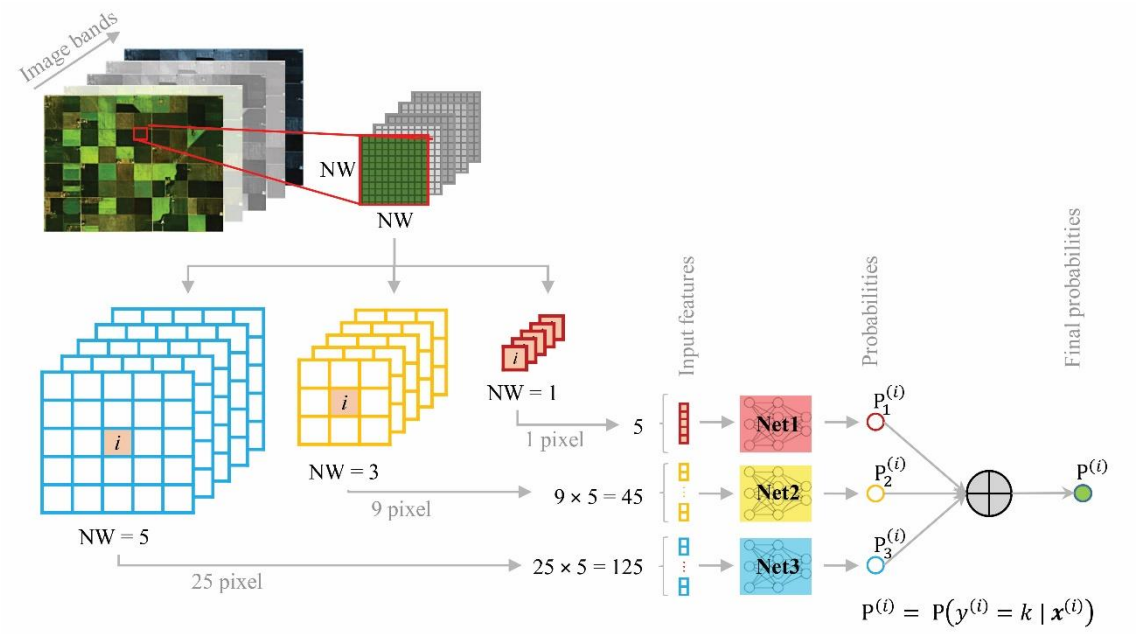


Image bands
NW
NW
i
i
i
NW = 5
NW = 3
NW = 1
25 pixel
9 pixel
1 pixel
25 × 5 = 125
9 × 5 = 45
5
Input features
Net1
Net2
Net3
Probabilities
Final probabilities
$P^{(i)}$
$P_1^{(i)}$
$P_2^{(i)}$
$P_3^{(i)}$
$P^{(i)} = P(y^{(i)} = k \mid \boldsymbol{x}^{(i)})$

Figure 6.

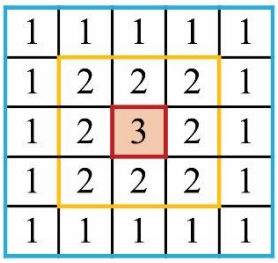


Figure 7.

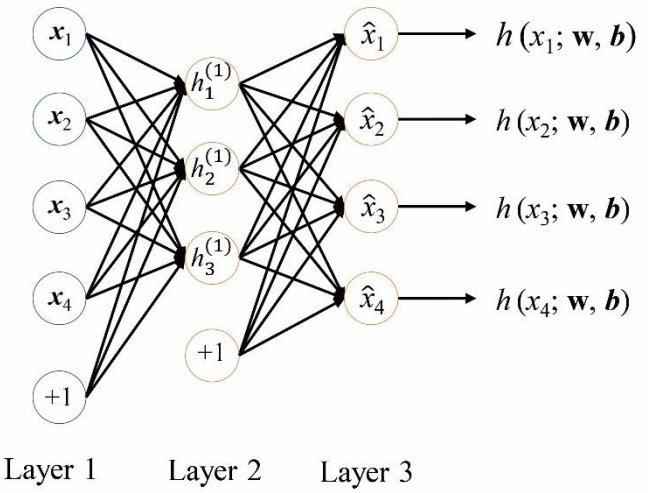

x_1
x_2
x_3
x_4
+1
$h_1^{(1)}$
$h_2^{(1)}$
$h_3^{(1)}$
+1
$\hat{x}_1$
$\hat{x}_2$
$\hat{x}_3$
$\hat{x}_4$
$h(x_1; \mathbf{w}, \boldsymbol{b})$
$h(x_2; \mathbf{w}, \boldsymbol{b})$
$h(x_3; \mathbf{w}, \boldsymbol{b})$
$h(x_4; \mathbf{w}, \boldsymbol{b})$
Layer 1
Layer 2
Layer 3

Figure 8.

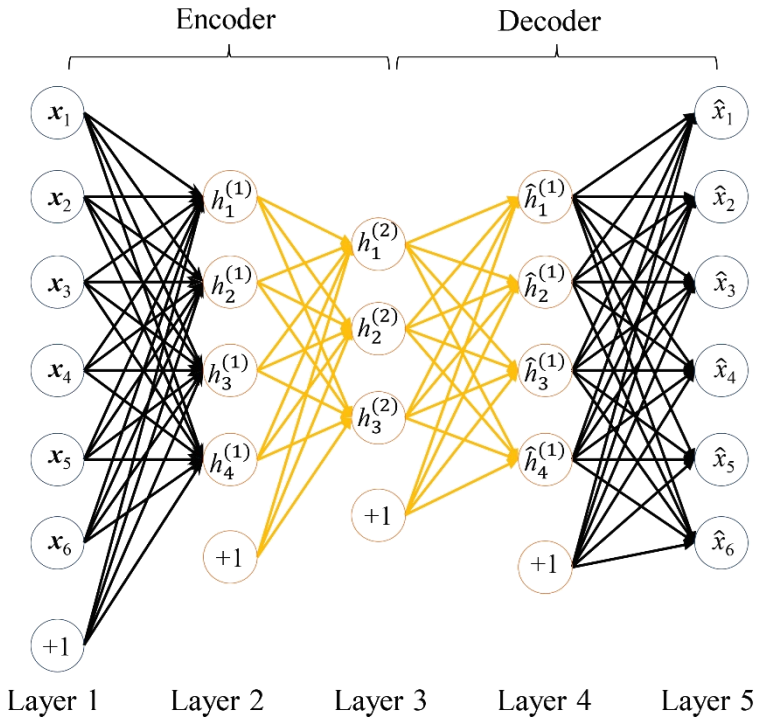

Encoder
Decoder
x_1
x_2
x_3
x_4
x_5
x_6
+1
$h_1^{(1)}$
$h_2^{(1)}$
$h_3^{(1)}$
$h_4^{(1)}$
+1
$h_1^{(2)}$
$h_2^{(2)}$
$h_3^{(2)}$
+1
$\hat{h}_1^{(1)}$
$\hat{h}_2^{(1)}$
$\hat{h}_3^{(1)}$
$\hat{h}_4^{(1)}$
+1
$\hat{x}_1$
$\hat{x}_2$
$\hat{x}_3$
$\hat{x}_4$
$\hat{x}_5$
$\hat{x}_6$
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5

Figure 9.

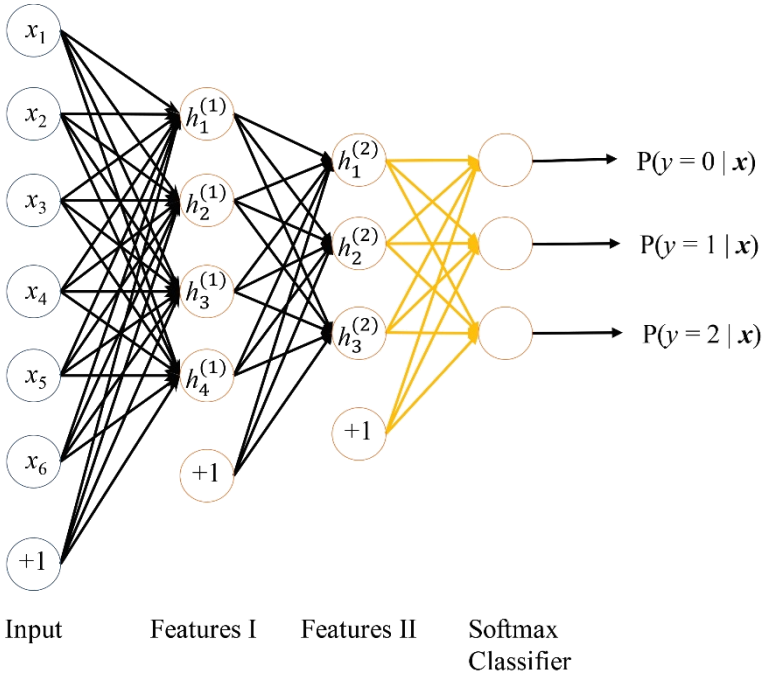

x_1
x_2
x_3
x_4
x_5
x_6
+1
$h_1^{(1)}$
$h_2^{(1)}$
$h_3^{(1)}$
$h_4^{(1)}$
+1
$h_1^{(2)}$
$h_2^{(2)}$
$h_3^{(2)}$
+1
P(y = 0 | $\boldsymbol{x}$)
P(y = 1 | $\boldsymbol{x}$)
P(y = 2 | $\boldsymbol{x}$)
Input
Features I
Features II
Softmax Classifier

Figure 10.

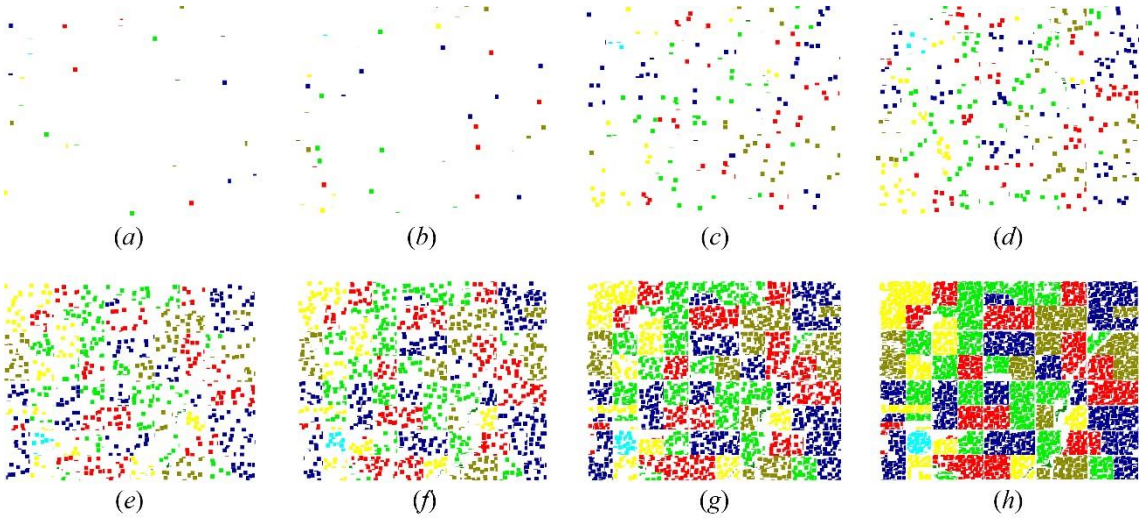


Figure 11.

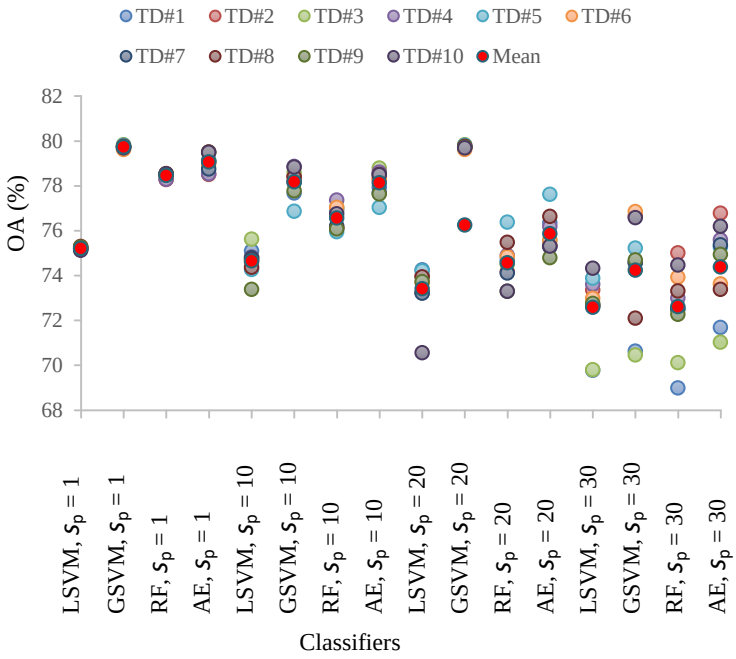

TD#1
TD#2
TD#3
TD#4
TD#5
TD#6
TD#7
TD#8
TD#9
TD#10
Mean
OA (%)
82
80
78
76
74
72
70
68
LSVM, s_p = 1
GSVM, s_p = 1
RF, s_p = 1
AE, s_p = 1
LSVM, s_p = 10
GSVM, s_p = 10
RF, s_p = 10
AE, s_p = 10
LSVM, s_p = 20
GSVM, s_p = 20
RF, s_p = 20
AE, s_p = 20
LSVM, s_p = 30
GSVM, s_p = 30
RF, s_p = 30
AE, s_p = 30
Classifiers

Figure 12.

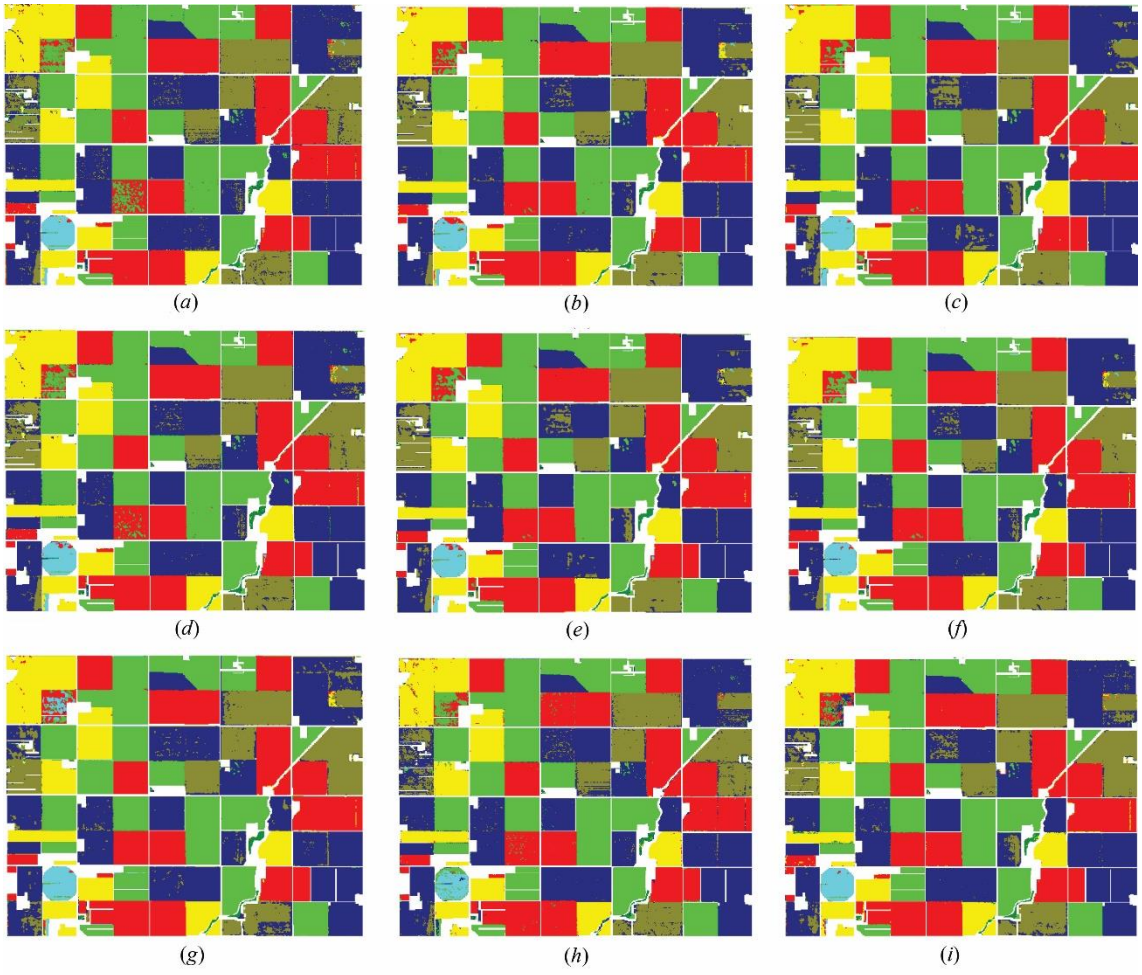
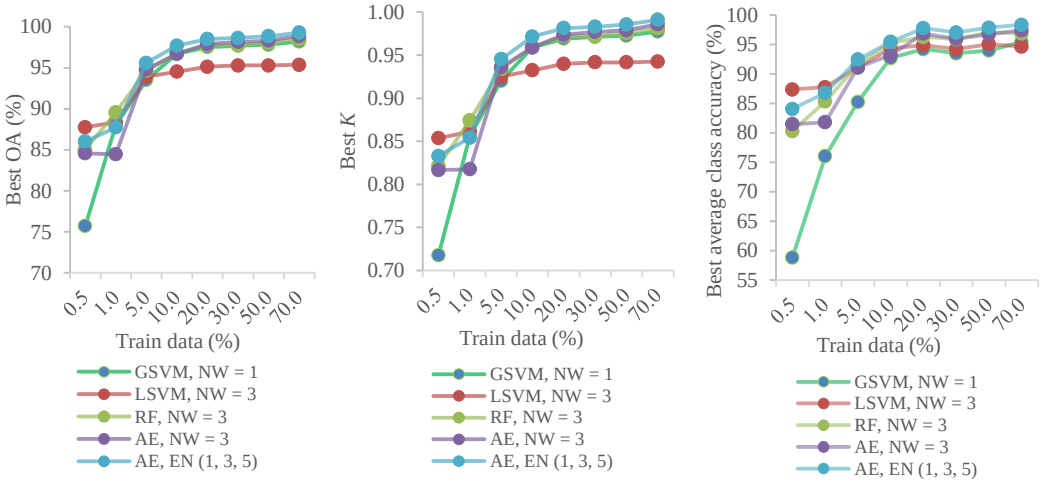
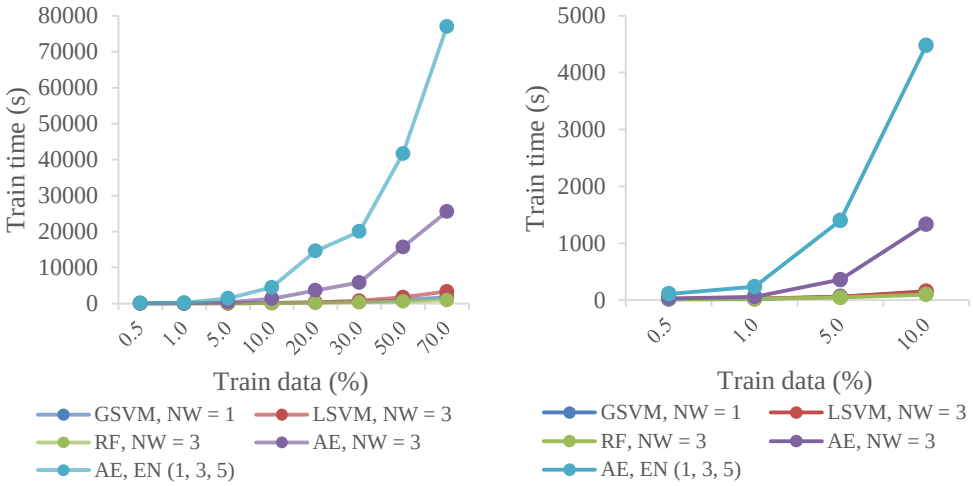


Figure 13.



(*a*) (*b*) (*c*)

Figure 14.


80000
70000
60000
50000
40000
30000
20000
10000
0
Train time (s)
0.5
1.0
5.0
10.0
20.0
30.0
50.0
70.0
Train data (%)
GSVM, NW = 1
LSVM, NW = 3
RF, NW = 3
AE, NW = 3
AE, EN (1, 3, 5)
5000
4000
3000
2000
1000
0
Train time (s)
0.5
1.0
5.0
10.0
Train data (%)
GSVM, NW = 1
LSVM, NW = 3
RF, NW = 3
AE, NW = 3
AE, EN (1, 3, 5)

(*a*) (*b*)

Table 1.

Classifier		*w* = 1 pixel		*w* = 10 pixel		*w* = 20 pixel		*w* = 30 pixel	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD
LSVM	OA (%)	75.21	0.05	74.64	0.58	73.40	1.07	72.58	1.57
	K (%)	71.09	0.06	70.50	0.71	69.24	1.19	68.36	1.67
GSVM	OA (%)	**79.73**	0.06	**78.18**	0.60	**76.25**	0.89	74.23	2.39
	K (%)	**76.30**	0.07	**74.58**	0.70	**72.48**	0.96	70.31	2.59
RF	OA (%)	78.46	0.10	76.56	0.52	74.56	0.94	72.61	1.86
	K (%)	74.91	0.11	72.81	0.58	70.70	1.00	68.6	1.99
AE	OA (%)	79.05	0.38	78.13	0.55	75.86	0.83	**74.38**	1.91
	K (%)	75.49	0.44	74.51	0.64	72.07	0.89	**70.46**	2.07

Table 2.

Classifier		NW = 1		NW = 3		NW = 5		EN (1, 3)		EN (3, 5)		EN (1, 3, 5)	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
LSVM	OA (%)	84.91	0.30	**92.73**	0.36	92.20	0.45						
	K (%)	81.94	0.37	**91.08**	0.43	90.47	0.53						
GSVM	OA (%)	86.20	0.46	**88.07**	1.27	67.37	1.87						
	K (%)	83.42	0.56	**85.57**	1.47	63.21	1.84						
RF	OA (%)	84.88	0.53	**94.05**	0.57	93.09	0.55						
	K (%)	81.93	0.64	**92.66**	0.68	91.52	0.67						
AE	OA (%)	86.04	0.46	93.31	0.74	92.64	0.73	94.22	0.80	93.99	0.61	**94.75**	0.54
	K (%)	83.26	0.55	91.78	0.89	90.98	0.86	92.87	0.96	92.59	0.74	**93.51**	0.65

Table 3.

Class name	Classifier											
	LSVM, NW = 1	LSVM, NW = 3	GSVM, NW = 1	GSVM, NW = 3	RF, NW = 1	RF, NW = 3	AE, NW = 1	AE, NW = 3	AE, NW = 5	AE, EN (1, 3)	AE, EN (3, 5)	AE, EN (1, 3, 5)
Corn	76.96	**96.62**	76.49	**87.11**	74.92	**96.31**	78.91	96.39	95.07	96.22	96.40	**96.47**
Pea	73.51	**90.65**	**72.45**	30.25	77.11	**93.47**	70.49	84.04	83.87	84.46	86.00	**87.00**
Canola	97.24	**98.38**	97.50	**97.66**	97.58	**98.38**	97.51	97.18	96.06	**98.47**	97.49	98.30
Soybean	94.00	**97.30**	**95.89**	94.92	93.87	**96.89**	94.27	95.94	95.82	97.24	96.47	**97.49**
Oat	59.90	**77.53**	**66.57**	65.07	63.73	**79.19**	66.50	81.92	82.29	81.87	83.27	**83.66**
Wheat	84.69	**90.69**	84.40	**90.17**	83.10	**95.22**	84.03	93.09	92.56	94.42	94.39	**95.17**
Broadleaf	33.48	**72.89**	**28.51**	16.23	32.69	**72.10**	42.47	**74.68**	36.54	74.54	55.94	73.84
Average per-class accuracy	74.26	**89.15**	**74.54**	68.77	74.72	**90.22**	76.31	89.04	83.17	89.60	87.14	**90.26**

Table 4.

Classifier		NW = 1		NW = 3		NW = 5		EN (1, 3)		EN (3, 5)		EN (1, 3, 5)	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
LSVM	OA (%)	92.87	0.38	**94.08**	0.41								
	K (%)	91.24	0.45	**92.70**	0.49								
GSVM	OA (%)	**93.96**	0.66	56.46	2.68								
	K (%)	**92.53**	0.80	52.23	2.63								
RF	OA (%)	94.51	0.44	**94.92**	0.60								
	K (%)	93.21	0.54	**93.70**	0.73								
AE	OA (%)	93.92	0.72	94.26	0.61	93.92	0.8	95.02	0.6	94.66	0.68	**95.26**	0.68
	K (%)	92.50	0.87	92.92	0.74	92.52	0.95	93.83	0.73	93.41	0.83	**94.13**	0.83

Table 5.

Class name	Classifier											
	LSVM, NW = 1	LSVM, NW = 3	GSVM, NW = 1	GSVM, NW = 3	RF, NW = 1	RF, NW = 3	AE, NW = 1	AE, NW = 3	AE, NW = 5	AE, EN (1, 3)	AE, EN (3, 5)	AE, EN (1, 3, 5)
Corn	**97.66**	97.31	**95.55**	40.25	**97.30**	97.04	97.48	97.30	96.34	**97.86**	97.17	97.59
Pea	**92.67**	91.76	**70.63**	0.00	94.48	**94.68**	86.37	84.94	82.48	87.32	85.16	**87.97**
Canola	**99.41**	99.06	**99.51**	91.10	99.17	**99.19**	98.53	97.77	97.04	**98.78**	97.81	98.64
Soybean	**98.34**	98.21	**98.25**	60.34	98.10	**98.11**	96.98	97.37	97.62	97.79	97.73	**97.99**
Oat	74.64	**80.88**	**75.90**	24.79	78.58	**79.98**	81.02	83.74	84.41	83.99	85.04	**85.38**
Wheat	90.35	**92.16**	**95.97**	50.39	95.07	**95.91**	93.18	93.67	92.78	94.75	94.17	**94.75**
Broadleaf	**79.84**	79.27	**62.22**	0.00	72.99	**74.54**	77.67	63.32	77.48	67.25	66.01	**82.81**
Average per-class accuracy	90.42	**91.24**	**85.43**	38.13	90.81	**91.35**	90.18	88.30	89.74	89.68	89.01	**92.16**

Table 6.

Class name	Classifier											
	LSVM, NW = 1	LSVM, NW = 3	GSVM, NW = 1	GSVM, NW = 3	RF, NW = 1	RF, NW = 3	AE, NW = 1	AE, NW = 3	AE, NW = 5	AE, EN (1, 3)	AE, EN (3, 5)	AE, EN (1, 3, 5)
Corn	**97.55**	97.22	**95.64**	42.70	**97.38**	97.02	97.44	96.81	95.78	**97.64**	96.60	97.27
Pea	**94.34**	93.17	**71.36**	0.00	**94.46**	95.87	93.17	83.05	92.55	90.68	87.83	**93.93**
Canola	**99.75**	99.27	99.78	**100.00**	**99.77**	99.49	99.69	99.23	99.38	**99.76**	99.37	99.71
Soybean	**95.84**	95.63	**94.57**	44.10	95.37	**95.55**	92.35	94.97	95.29	94.50	95.47	**95.48**
Oat	77.38	**80.41**	**77.08**	31.29	82.65	**85.46**	82.76	86.04	**90.37**	85.28	89.23	88.28
Wheat	90.93	**93.88**	**96.59**	59.79	94.02	**94.49**	94.88	95.27	90.01	**95.94**	93.92	95.39
Broadleaf	77.81	**78.45**	**61.79**	0.00	**75.05**	74.40	75.51	78.64	76.70	79.19	79.19	**81.58**
Average per-class accuracy	90.52	**91.15**	**85.26**	39.70	91.24	**91.75**	90.83	90.57	91.44	91.86	91.66	**93.09**

Table 7.

Classifier	Training percentage (%)															
	0.5		1.0		5.0		10.0		20.0		30.0		50.0		70.0	
	OA (%)	*K*	OA (%)	*K*	OA (%)	*K*	OA (%)	*K*	OA (%)	*K*	OA (%)	*K*	OA (%)	*K*	OA (%)	*K*
LSVM, NW = 1	**89.37**	**0.87**	**88.72**	**0.86**	92.92	0.91	93.24	0.92	93.91	0.92	93.92	0.92	93.57	0.92	93.48	0.92
LSVM, NW = 3	87.76	0.85	88.37	0.86	**93.92**	**0.92**	**94.52**	**0.93**	**95.13**	**0.94**	**95.28**	**0.94**	**95.28**	**0.94**	**95.35**	**0.94**
GSVM, NW = 1	**75.74**	**0.72**	**88.04**	**0.86**	**93.54**	**0.92**	**96.69**	**0.96**	**97.52**	**0.97**	**97.69**	**0.97**	**97.81**	**0.97**	**98.18**	**0.98**
GSVM, NW = 3	28.78	0.25	39.08	0.34	58.52	0.54	64.38	0.60	76.77	0.73	78.21	0.75	80.32	0.77	81.85	0.78
RF, NW = 1	**86.06**	**0.83**	**90.20**	**0.88**	94.27	0.93	96.63	0.96	97.34	0.97	97.30	0.97	97.34	0.97	97.59	0.97
RF, NW = 3	85.02	0.82	89.56	0.87	**94.65**	**0.93**	**96.75**	**0.96**	**97.77**	**0.97**	**97.76**	**0.97**	**98.05**	**0.98**	**98.48**	**0.98**
AE, NW = 1	83.43	0.80	87.98	0.86	94.47	0.93	96.74	0.96	97.47	0.97	97.59	0.97	97.52	0.97	97.89	0.97
AE, NW = 3	84.58	0.82	84.47	0.82	94.82	0.94	96.70	0.96	97.90	0.97	98.16	0.98	98.31	0.98	98.87	0.99
AE, NW = 5	82.60	0.79	83.88	0.81	94.44	0.93	96.75	0.96	97.95	0.97	98.05	0.98	98.52	0.98	98.93	0.99
AE, EN (1, 3)	85.97	0.83	**88.91**	**0.87**	95.37	0.94	97.41	0.97	98.36	0.98	98.48	0.98	98.65	0.98	99.09	0.99
AE, EN (3, 5)	85.15	0.82	85.32	0.83	95.38	0.94	97.20	0.96	98.28	0.98	98.44	0.98	98.74	0.98	99.15	0.99
AE, EN (1, 3, 5)	**86.02**	**0.83**	87.74	0.85	**95.59**	**0.94**	**97.71**	**0.97**	**98.51**	**0.98**	**98.63**	**0.98**	**98.86**	**0.99**	**99.28**	**0.99**

Table 8.

Classifier	Training percentage (%)							
	0.5	1.0	5.0	10.0	20.0	30.0	50.0	70.0
LSVM, NW = 1	86.85	**88.00**	90.47	92.57	93.65	92.84	92.70	92.73
LSVM, NW = 3	**87.37**	87.76	**91.13**	**94.34**	**94.88**	**94.23**	**95.07**	**94.65**
GSVM, NW = 1	**58.81**	**76.05**	**85.26**	**92.71**	**94.21**	**93.51**	**93.96**	**95.46**
GSVM, NW = 3	18.04	23.89	39.70	44.16	53.44	54.54	56.04	57.37
RF, NW = 1	**81.88**	**85.73**	91.41	94.86	96.20	95.32	95.84	95.64
RF, NW = 3	80.30	85.32	**91.61**	**95.24**	**96.29**	**95.89**	**97.06**	**96.88**
AE, NW = 1	80.62	86.90	91.61	94.38	95.81	95.34	95.59	96.15
AE, NW = 3	81.48	81.80	91.11	93.18	96.85	95.98	96.75	97.42
AE, NW = 5	79.79	83.52	91.10	95.62	96.95	96.14	97.48	98.10
AE, EN (1, 3)	84.06	**87.41**	92.31	95.82	97.39	96.49	97.18	97.62
AE, EN (3, 5)	82.10	83.62	91.71	**96.78**	97.42	96.66	97.60	97.81
AE, EN (1, 3, 5)	**84.06**	86.80	**92.46**	95.48	**97.78**	**97.04**	**97.84**	**98.34**

Table 9.

Classifier	Training percentage (%)							
	0.5	1.0	5.0	10.0	20.0	30.0	50.0	70.0
LSVM, NW = 1	**1.00**	**1.23**	5.29	21.65	57.40	124.63	336.02	709.23
LSVM, NW = 3	2.96	3.91	12.28	26.72	57.59	87.00	176.26	325.51
GSVM, NW = 1	1.84	2.02	**3.81**	**6.75**	**13.53**	**21.24**	**39.60**	**66.57**
GSVM, NW = 3	7.18	18.24	86.23	305.36	1233.57	1180.65	2006.68	4949.08
RF, NW = 1	4.73	5.06	10.28	28.90	64.74	137.39	320.06	617.13
RF, NW = 3	36.61	61.47	320.39	996.59	4182.72	8589.53	20627.70	36491.90
AE, NW = 1	3.97	4.46	8.66	18.26	34.11	53.71	101.65	167.98
AE, NW = 3	5.66	10.81	66.61	245.65	669.04	1073.77	2895.97	4702.30
AE, NW = 5	7.56	14.41	105.52	273.81	783.88	1430.38	2775.86	4524.62
AE, EN (1, 3)	12.84	29.06	152.84	551.01	1902.61	2254.42	4902.65	9651.38
AE, EN (3, 5)	13.23	25.22	172.13	519.45	1452.92	2504.15	5671.83	9226.91
AE, EN (1, 3, 5)	20.41	43.47	258.36	824.82	2686.48	3684.80	7678.51	14175.99