Minimal learning machine in hyperspectral imaging classification

Anna-Maria Hakola[a] and Ilkka Pölönen[a]

[a]Faculty of Information Technology, University of Jyväskylä, 40100, Jyväskylä, Finland

ABSTRACT

A hyperspectral (HS) image is typically a stack of frames, where each frame represents the intensity of a different wavelength of light. Each spatial pixel has a spectrum. In the classification of the HS image, each spectrum is classified pixel-by-pixel. In some of the real-time applications, the amount of the HS image data causes performance challenges. Those issues relate to the platforms (e.g. drones) payload restrictions, the issues of the available energy and to the complexity of the machine learning models.

In this study, we introduce the minimal learning machine (MLM) as a computationally cheap training and classification machine learning method for the hyperspectral imaging classification. MLM is a distance-based method that utilizes mapping between input and and output distances. Input distance is a distance between the training set and its subset R. Output distance is corresponding distances between the label values of the training set and the subset R. We propose a training point selection framework, which reduces the number of data points in the R by selecting the points class-by-class, in the direction of the principal components of each class.

We test MLM's performance against four other classification machine learning methods: Random Forest, Artificial Neural Network, Support Vector Machine and Nearest Neighbours classifier with three known hyperspectral data sets. As the main outcomes, we will show how the performance is affected by the size of the subset R. We compare our subset selection method MLM's performance to the random selection MLM's performance. Results show that MLM is an computationally efficient way to train large training sets. MLM reduces the complexity of the analysis and provides computational benefits against other models. Proposed framework offers tools that can improve the MLM's classification time and the accuracy rate compared to the MLM with randomly picked training points.

Keywords: Hyperspectral Imaging, Minimal Learning Machine, Classification, Principal Component Analysis, Distance Learning

1. INTRODUCTION

Hyperspectral (HS) images contains information that allows the characterization, identification and classification of the targets, such as land-covers with improved accuracy and robustness.[1] Since the technical evolution of optical sensors[2] has been improving the imagers, there has been several new application domains, for example in the medical diagnosis,[3] skin cancer research,[4] forest industry[5] and agricultural applications[3] .

Hyperspectral image classification is a process, where single pixels are assigned into a set of classes[2] . Classification approaches can be split into categories of *supervised, unsupervised and semi-supervised classifiers*[6] . Supervised methods classifies based on model, which is created with training samples and their labels. Unsupervised classifiers are using clustering methods without labelled training samples and semi-supervised classifiers are using both, labelled and unlabelled training samples[2,6] .

HS images can be classified pixel-wise with *spectral classifiers* or spectral-spatial with *spectral-spatial classifiers*[7] . The spectral classifier considers the HS images as a list of spectral information, while the spectral-spatial-classifier uses both, the spectral and the spatial information[6] .

Further author information: (Send correspondence to Anna-Maria Hakola)
Anna-Maria Hakola.: E-mail: anna.m.hakola@jyu.fi, Telephone: +358 (0)40 54 92 12 0
Ilkka Pölönen : E-mail: ilkka.polonen@jyu.fi, Telephone: +358 (0)400 248 140

The HS image classification can be a challenge[2] . For example the accuracy results that can be reached with standard classifiers and multispectral images are typically compromised with HS images[1] . The typical challenges of the spectral classifiers are the relatively small size of the training set from the high-dimensional data,[7] the high number of spectral channels,[1] the spatial variability of the spectral signature[1] and the quality of the spectral data. As an example, the challenges are related to the Hughes phenomenon,[8] the conditions, such as incident illumination or instrument noises[7] or to the high cost of the true sample labelling[1] .

Despite the challenges, there are many classic classification methods that can perform well with HS images[1,3,6] . Some of the popular machine learning classification approaches utilises the neural networks, support vector machines, random forests or deep learning classification methods[6] . Because those methods can be complex and time-consuming, it is an interesting idea to introduce HS images to the relatively new classifier, which is an easy to implement and has had a promising results on performance and accuracy[9] .

Minimal learning machine[9] (MLM) is a supervised, distance-based machine learning classification method that utilizes mapping between input and and output distances. The input distance is a distance between the training set and its subset R. The output distance is corresponding distances between the label values of the training set X and subset R. The MLM classification model is a generalization of a nearest neighbour classifier[9] . This approach requires argument sorting for the distances between the input and output values and assigning of the closest label value from the ground truth labels.

Previous studies confirms, that one of the main advantages of the basic MLM is that there is only one parameter that requires tuning[9] . The parameter is the number of the training samples (R). The closer the amount of the selected training points (R) and the size of the entire training data X goes, the more accurate are the results. That might encourage to increase the amount of the selected training points, but it will also increase the computational efficiency and the used time.

When we are presenting the HS images to a supervised spectral MLM classifier, we need to pay attention to a few features of the HS images. HS image consists of a large amount of spectral bands of which are the dimensions of the spectral data. The high number of dimensions and the amount of the data makes the processing computationally and memory costly. Other challenges are the curse of the dimensionality[8,10] and the redundancy among the samples[11] . Those challenges might have an impact to the classifiers performance and accuracy.

Dimensionality reduction methods offers one solution for those challenges. One of the most widely used method[11] is the Principal Component Analysis (PCA) with its different extensions. With PCA, the data is projected with the orthogonal projections of which maximises the variance of the data. The data is yielded to a new uncorrelated coordinate system[12] . With PCA, we can reduce the dimensions and select the training points intentionally for the classifier.

As a main results in this study, we propose a new approach which can increase the accuracy rate and training time, and reduce the classification time of the MLM classifier. The proposed framework (subsection 2.3) focuses on the selection of the training output samples, and it consists from the data point selection algorithm, the MLM training and the MLM classification algorithms.

The data point selection algorithm 1 utilises the PCA and selects only 3 data points from each of the classes in the direction of each used principal components. The aim is to have a collection of data points which represents the geometry of each class. This approach differs from the original MLM approach of picking the training points randomly[9] . The objective is to minimize the size of R, as it is the most influential factor[9] in the computational complexity of this method, without reducing the accuracy with reduced training points. The new framework is called the PC-MLM.

The proposed framework uses three hyper-parameters. The amount of principal components (PC), the number of neighbours and optionally the distance metrics (Euclidean, Manhattan or Cosine). Those hyper-parameters controls the size of the subset R. The focus of the new framework is to improve the performance and still maintain the accuracy rate in an acceptable level.

We will test the PC-MLM framework against four other supervised classification machine learning methods: Random Forest, Artificial Neural Network, Support Vector Machine and k-Nearest Neighbours (kNN) classifier with three known hyperspectral data sets (Indian Pines, Salinas and Pavia City). As the main outcomes, we

show how the performance is affected by the size of the subset R. Our hypothesis is that the MLM is the fastest model to train when the size of the training set is large. MLM reduces the complexity of the analysis and provides computational benefits against other models. PC-MLM framework offers tools that can improve the accuracy rate and classification time compared to the MLM with randomly picked training points with large data sets.

The content of the paper is organized as follows. The section 2 describes methods, PC-MLM frameworks algorithms and the demonstration materials. The results are introduced in the section 3. The analysis of the results are discussed into the section (4), and the final section 5 concludes the study.

2. MATERIAL AND METHODS

On this section, we will present the MLM extension to the hyperspectral imaging. We explain the idea of the intentionally selected training points (PC-MLM) and we will introduce the framework with step-by-step algorithms. Finally there are a short overview to the comparison methods and the introduction of the selected HS images as an demonstration material.

2.1 Minimal learning machine

MLM is an supervised machine learning algorithm that can be can be extended to classification tasks[9] . MLM's learning process has three phases[13] . It consists of building a linear mapping between the matrices of input and output distances. Input distance is a distance between the training set and its subset R. Output distance is corresponding distances between the label values of the training set and subset R. The generalization phase utilises the learned mapping and provides an estimation of the distances between output values and the target output value. The third phase is an optimization problem, based on the predicted output distances and the ground truth points. The third phase is the computationally most complex[9] .

In our framework, the basic MLM model is introduced to hyperspectral image classification. The aim is to reduce the complexity of the optimization by selecting three training points from the directions of principal components of the data set. Attempt is to mimics data sets geometry.

In the case of HS images, the training set of spectra with d wavebands is $\mathbf{x}_i \in X \subset \mathbb{R}^d$ and $\mathbf{m}_k \in R$ is the intentionally selected sampled subset of the X. Correspondingly $\mathbf{y}_i \in Y \subset \mathbb{R}$ are the labels of the training set and $\mathbf{t}_k \in T$ are the subset of the Y. The training set X consist of N samples, and subset R has K samples. Now $d(\mathbf{x}_i, \mathbf{m}_k)$ and $\delta(\mathbf{y}_i, \mathbf{t}_k)$ are the linear mapping distances.

After selecting the samples, we can define two matrices based on these distances $\mathbf{\Delta}_y \in \mathbb{R}^{N\times K}$ and $\mathbf{D}_x \in \mathbb{R}^{N\times K}$. By assuming the linear mapping between these two distance matrices, we have a linear model

$$\mathbf{\Delta}_y = \mathbf{D}_x\mathbf{B} + \mathbf{E}, \tag{1}$$

where $\mathbf{B}$ is coefficients and $\mathbf{E}$ is the residual. Coefficients $\mathbf{B}$ can be approximated using the ordinary least squares estimator[13]

$$\widehat{\mathbf{B}} = (\mathbf{D}_x^T\mathbf{D}_x)^{-1}\mathbf{D}_x^T\mathbf{\Delta}_y. \tag{2}$$

Now $\hat{\mathbf{B}}$ is a linear model between distances $\delta(\mathbf{y}_i, \mathbf{t}_k)$ and $d(\mathbf{x}_i, \mathbf{m}_k)$. Now, for the new spectrum $\mathbf{x}_n$ the distance between its label y_n and set T is

$$\delta(y_n, T) = d(\mathbf{x}_n, R)\widehat{\mathbf{B}}. \tag{3}$$

Label y_n can be estimated by solving an quadratic optimisation problem

$$\min_{y_n} \sum_{k=1}^{K} \left(y_n - \mathbf{t}_k\right)^T \left(y_n - \mathbf{t}_k\right) - \delta^2(y_n, T)\Big)^2. \tag{4}$$

The equation 4 is the bottle neck of the MLM[13] . If we solve it with optimization methods, our classification is time-consuming and computationally expensive. We can consider that the model $\widehat{\mathbf{B}}$ is a of generalization

of the nearest neighbour classifier. Equation 3 gives us distances to the nearest label values of $\mathbf{x}_n$. For the classification of the $\mathbf{x}_n$, we need to perform argument sorting for $\delta(\mathbf{y}_n, T)$ and assign the closest label value from T. By selecting k closest values and use majority voting of labels; we have similar results than with kNN. The detailed implementation is shown in algorithms 2 and 3.

In MLM, the subset R is selected picking data points randomly[9] . When the size of R approaches the size of X, the accuracy of the results improves[9] . The objective is to minimize the size of R, as it is the most influential factor in the computational complexity of this method. In our framework, we utilise previous MLM steps with the idea of using principal components to select the data points intentionally to represent the shape of the selected component.

2.2 The selection of the reference points

The Principal Component Analysis (PCA), finds the directions of the data variance and projects the data according to the variance orthogonally, in as many directions as we have dimensions in the data[12] . The PCA reveals the geometry of the data; the first PC component has the highest variation of the data, the second component reveals the highest variance on orthogonal direction of the first component and so on.

On this study, we utilise the properties of the PCA, by using the component directions to select intentionally the data points from training set X to the reference set R. As a result, the R represents the geometry of each class of the the HS image classification targets. We performed this separately for each class of the training set X. We selected the minimum, maximum and median positions from each of the selected principal components (PC) directions. Because of the noise in the data sets, the minimum and maximum positions were tuned by moving both of the extreme values 5% towards the median position. By using this strategy, we can significantly reduce the size of the reference set R.

On figures 1, 2 and 3 the training data from the each of the HS images is visualised class-by-class after the PCA. The figures shows the first and the second principal components on the X and Y axis. The selected reference points are marked to the figures with colors. Black represents the first direction, red is the second and yellow is the direction of the third principal component. The details of the ground truth classes can be seen on the subsection 2.4, tables 2 and 1.

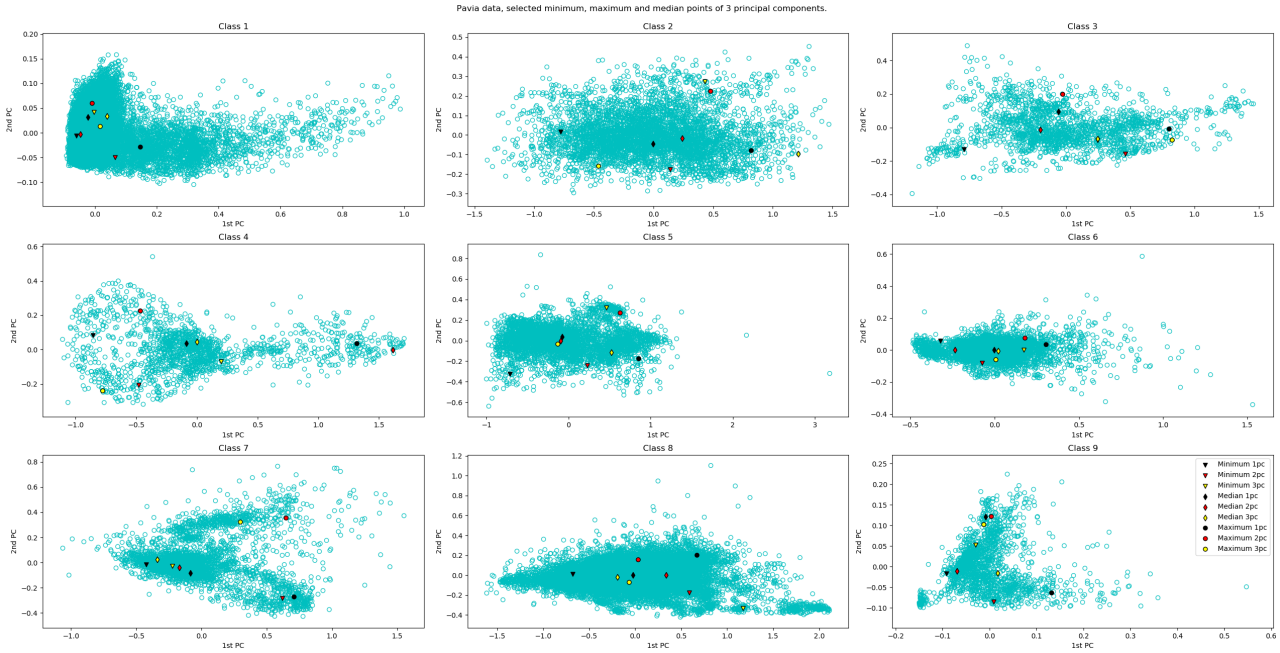

Figure 1. Pavia Centre HS image. The class-by-class visualisation of the first and second principal components. The selected data points from the first three principal components of the class, of which represents the geometry of the each class are marked with colors. The total amount of selected data points with three principal components was 81.

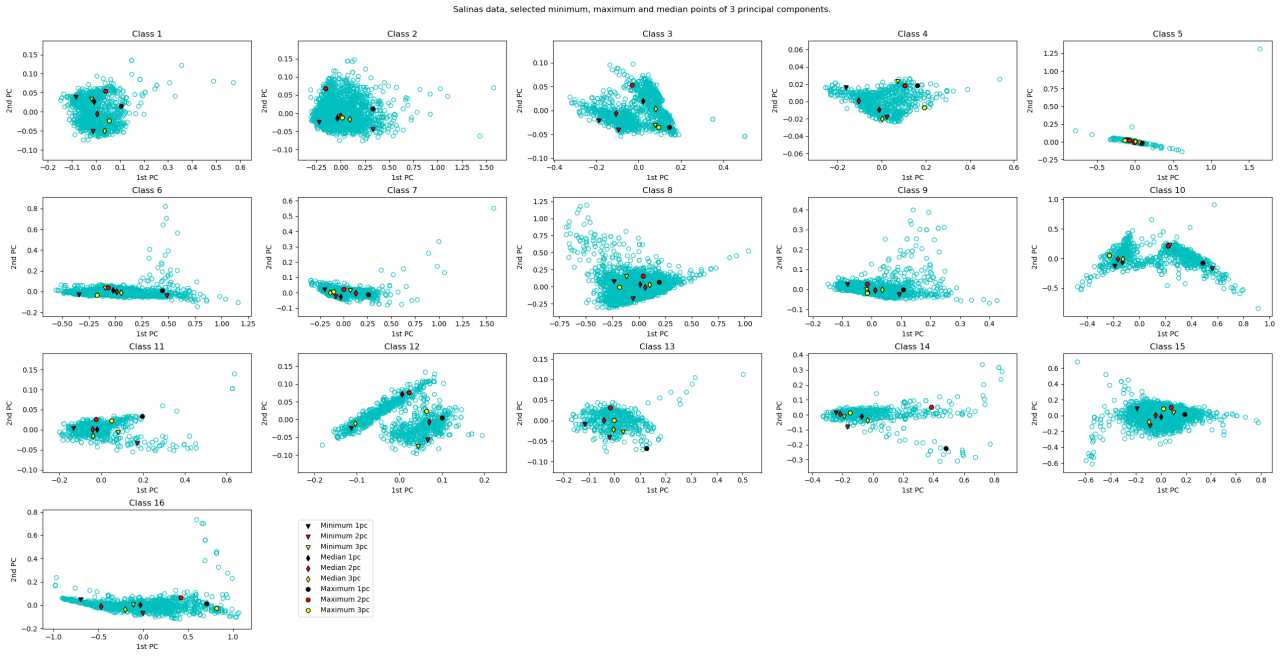

Figure 2. Salinas HS image. The class-by-class visualisation of the first and second principal components. The selected data points from the first three principal components of the class, of which represents the geometry of the each class are marked with colors. The total amount of selected data points with three principal components was 144.

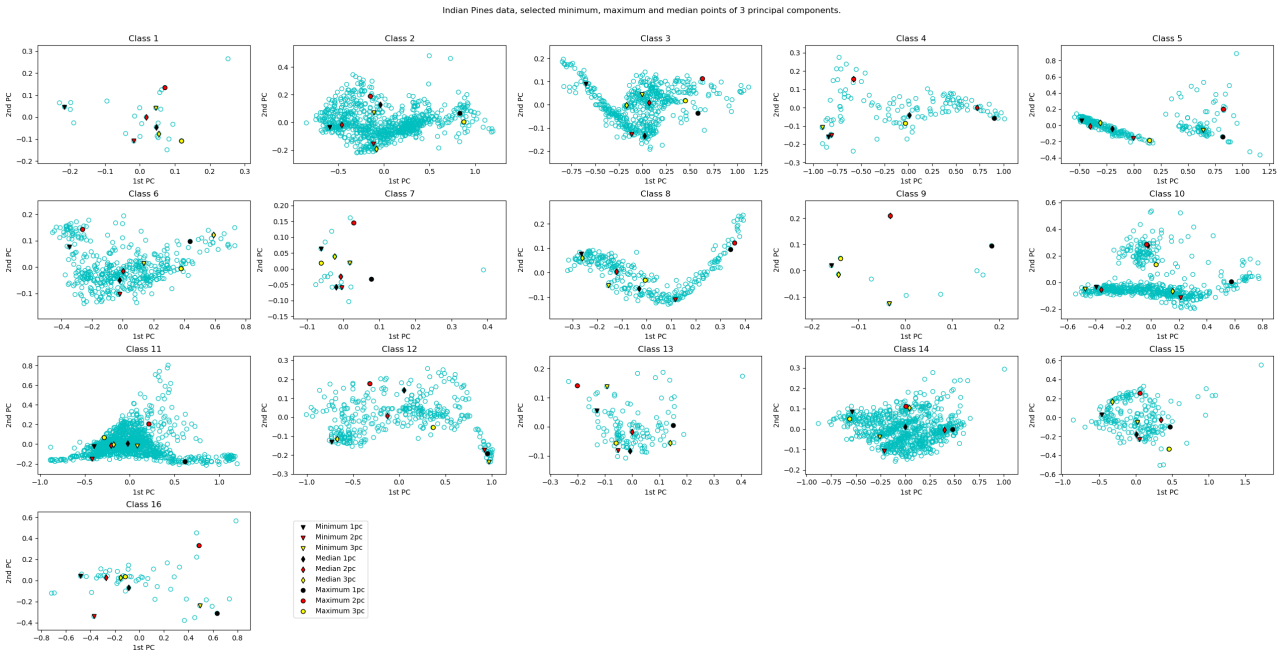

Figure 3. Indian Pines HS image. The class-by-class visualisation of the first and second principal components. The selected data points from the first three principal components of the class, of which represents the geometry of the each class are marked with colors. The total amount of selected data points with three principal components was 144.

2.3 The PC-MLM framework

The proposed framework consists of three algorithms. At first, the randomised training arrays are introduced to the PC algorithm 1. The training phase starts from the reference points selection algorithm 1 and ends to the first MLL algorithm 2. Prediction is the last step of the framework, the implementation can be seen on algorithm 3. The training time of this study is calculated from performing the algorithms 1 and 2, and the classification time is the time used on performing the algorithm 3. The framework was implemented in Python. The PCA, accuracy rate, distance metrics were implemented with Scikit-learn[14] metrics and decomposition methods.

Algorithm 1: The selection of the reference points

Input: Training data, the number of components
Step 1: Perform PCA class-by-class
For each class
Compute number of components with PCA from the training data
Step 2: Select training points from each component
For each component
Argument sort component
Select median point from sorted score
Select subtracted minimum from sorted score
Select subtracted maximum from sorted score
Step 3: store selected points
Append selected points to set R and their labels to set T
Result: The R and T are now sorted arrays that represents the data sets shape in the direction of the selected principal components. The R and T are ready for the MLM classifier implementation

Algorithm 2: The MLM training phase

Input: X (Training data), Y (Labels of the X), R (Selected reference points), T (Labels of the R), distance metric

Step 1: Calculate output distances

Use Euclidean method and calculate the distances between Y and T.

Step 2 : Calculate input distances

Use distance metric and calculate the distances between X and R

Step 3: Approximate the coefficients

Use the ordinary least squares estimator (equation 2), and mirror the input distances to output distances

Output: Trained model $\widehat{\mathbf{B}}$

Algorithm 3: The MLM prediction phase

Input: New data, T, R, $\widehat{\mathbf{B}}$, distance metric

Step 1: Calculate new distances

Calculate new_distances, use selected metrics and calculate distances between New data and R

Step 2: Solve the equation 3

$\delta(y_n, T)$ = new_distances.dot($\widehat{\mathbf{B}}$)

perform an argument sort to $\delta(y_n, T)$

Step 3: Select labels

if *number of neighbours != 1* **then**

select neighbours (n shortest distances from sorted $\delta(y_n, T)$)

The result label is the mode of the neighbours

else

select the nearest neighbour from the T, use the indexes of the sorted $\delta(y_n, T)$

end

Result: The data is classified with PC-MLM framework.

2.4 Spectral images and preprocessing

The experiments were done with three known hyperspectral images, downloaded from the Grupo De Intelligencia Computational (GIC)[15] . The Pavia Centre HS image is the largest one of these three data sets. It has been acquired by the ROSIS sensor with 102 spectral bands with the geometric resolution of 1.3 meters and with a spectrum coverage ranging from 430 to 860 nm[16] . The image size is 1096 x 1096 pixels. The image ground truth (table 1) is divided into 9 classes[15] . The geometry of Pavia HS image's each class can be seen on figure 1.

The Salinas scene is collected with AVIRIS sensor. It has 224 spectral bands and the labels are divided in to 16 classes (table 2). The size of the data set is 512 x 217 pixels with 3.7m spatial resolution over the range of 400–2500 nm[17] . Salinas data set is preprocessed, 20 water absorption bands has been removed from the data[15] . The Salinas HS image's class-by-class geometry can be seen on figure 2.

Third HS image is the smallest one, Indian Pines. It consists of 145 x 145 pixels and 224 spectral reflectance bands in the wavelength range of 400 - 2500 nm. It has been captured with AVIRIS sensor over the Indian Pines test site. The ground truth (table 2) has been divided in to 16 classes that represents agriculture, forest and other natural perennial vegetation. 20 Water absorption bands has been removed, leaving the total amount of bands to 200. Indian Pines data set differs from the other data sets with its ground truth coverage. Some of the ground truth classes has only 5% coverage.[15] The geometry of Indian Pines classes can be seen on figure 3.

Table 1. Pavia Centre HS image ground truth classes and number of the samples

#	Class	Samples
1	Water	824
2	Trees	820
3	Asphalt	816
4	Self-Blocking Bricks	808
5	Bitumen	808
6	Tiles	1260
7	Shadows	476
8	Meadows	824
9	Bare Soil	820

Table 2. Salinas and Indian Pines HS image, ground truth classes and number of the samples

	Salinas			Indian Pines	
#	Class	Samples	#	Class	Samples
1	Brocoli green weeds 1	2009	1	Alfalfa	46
2	Brocoli green weeds 2	3726	2	Corn-notill	1428
3	Fallow	1976	3	Corn-mintill	830
4	Fallow rough plow	1394	4	Corn	237
5	Fallow smooth	2678	5	Grass-pasture	483
6	Stubble	3959	6	Grass-trees	730
7	Celery	3579	7	Grass-pasture-mowed	28
8	Grapes untrained	11271	8	Hay-windrowed	478
9	Soil vinyard develop	6203	9	Oats	20
10	Corn senesced green weeds	3278	10	Soybean-notill	972
11	Lettuce romaine 4wk	1068	11	Soybean-mintill	2455
12	Lettuce romaine 5wk	1927	12	Soybean-clean	593
13	Lettuce romaine 6wk	916	13	Wheat	205
14	Lettuce romaine 7wk	1070	14	Woods	1265
15	Vinyard untrained	7268	15	Buildings-Grass-Trees-Drives	386
16	Vinyard vertical trellis	1807	16	Stone-Steel-Towers	93

All of the three HS images were preprocessed as follows. At first, the spectral data was scaled between 0 and 1. The spectral data and the ground truth data were converted and reshaped to two-dimensional arrays. On the array, each spatial pixel has its values, which is the pixel spectrum. On second, the spectral data and the ground truth data were randomised with random permutation method, and all non-classified values were removed from the both of the data sets. The randomised spectral data and its ground truth were divided to training (60%), validating (20%) and testing (20%) portions. The framework's model was trained with the training data, performance and parameters were validated with validation data and the results were achieved with the test data.

2.5 Reference methods

We tested PC-MLM against four other classification machine learning methods: Random Forest (RF), Artificial Neural Network (ANN), Support Vector Machine (SVM) and k-Nearest Neighbours (kNN).

SVM is a supervised machine learning model which is based on the theory of statistical learning[18] . The basic idea is to separate the classes with hyperplane in a high or infinite dimensional space[14] . Hyperplane is the decision surface of which the classifier uses to make the decisions on the classification phase. The hyperplane is optimal, when the margin between the nearest positive and nearest negative training sample is maximized.[18]

ANN methods are widely used in image analysis. Basic model of ANN consists of the input, hidden and output layers[14] . The variables in the input layer are called nodes, in our case the nodes are the training set samples. The nodes on the output layer represents the values of the output classes. There are weighted links between the layers, which controls the flow from input layer thru the hidden layers finally to the output layer.[18]

RF classifier is an ensemble learning algorithm of which are more accurate and robust towards noise than single classifiers[19,20] . The original RF classifier consists of a group of a single voting classifiers. Each classifier votes for the assignation of the most frequent class for the input vector.[20] On this study, we implemented the RF classifier from the Scikit-learn Python library, of which combines the classifiers by averaging their probabilistic prediction, which differs from the Breimans original voting strategy[14,20,21] . The fourth reference method, the Nearest Neighbours classifier (kNN) was used with reference points R. We used the Scikit-learn Nearest neighbour classifier implementation[14] to predict the classes.

The implementations of all the classifiers were from the Scikit-learn Python library[14] . SVM, ANN and RF were optimized with two rounds of hyper-parameter search with the largest data set Pavia Centre. At the first phase, we narrowed down the selection of possible hyper-parameter values with Scikit-learn randomized Search CV method[14] . On the second phase, based on the results of the first search, we narrowed down the values and run the Scikit-learn's Grid Search CV method[14] . As a result, we found the hyper-parameters of the best accuracy rate for each classifier. The kNN classifier was tested with the same number of neighbours than in our PC-MLM classifier.

All computations were done using Dell laptop with Intel Core i5-7300U CPU 2,6 GHz processor and 8 GB memory.

3. RESULTS

The first part of the section presents the results of a comparison of the distance metrics of the MLM and PC-MLM classifiers. Second, we present a classifier comparison with figures and tables, using the distance metrics selected at the beginning of the chapter. Finally, we will see how the PC-MLM framework can improve the MLM classifier performance on large data sets.

3.1 MLM with different distance metrics

Since the MLM is a distance based method, it is important to evaluate different distance calculation metrics and use the most suitable metric in the PC-MLM implementation. Fig. 4 shows that the Euclidean distance metric reached the overall best accuracy rate with all of the HS images. The Cosine distance metrics accuracy rate was significantly lower than with the other methods.

Based on these results, we selected the Euclidean distance method to our framework and all of the results shown this study are produced with it.

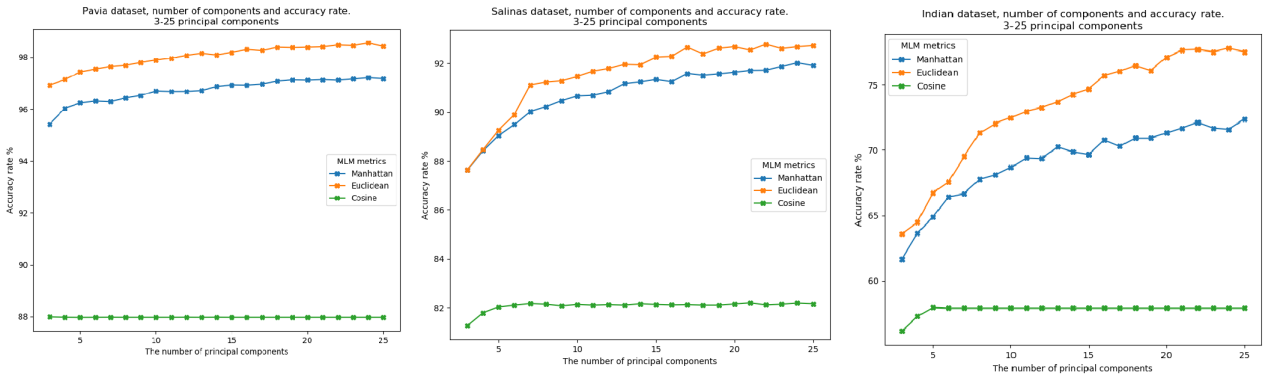

Figure 4. The comparison of the distance methods. The accuracy rate with different principal components (sizes of R).

3.2 The classifier comparison

Tables 3, 5 and 4 and figures 5, 7 and 6 shows that the MLM is one of the fastest classifiers in the comparison, and it can produce comparable accuracy results against other spectral classifiers. MLM can train the largest Pavia data set faster than the other classifiers. With smaller data sets, the training time of MLM is close to the fastest model, the kNN classifier. PC-MLM has reached the third place on the training time comparison with all of the data sets.

The classification maps (Figs. 5, 6 and 7) visualises the prediction results of all of the classifiers on the comparison. Maps were produced by training the models with each HS images training data, and by predicting with the whole data of the HS images.

Table 3. Pavia Centre method comparison, 25 principal components, 30 nearest neighbours

	MLM	25 PC-MLM	25 PC kNN	SVM	ANN	RF
Number of training samples	X: 88891, R: 675	X: 88891, R: 675	675	88891	88891	88891
Training time	**5.92**	12.39	6.93	24.21	552.54	36.86
Classification time	2.06	1.82	0.10	10.75	1.61	**0.33**
Accuracy rate	94.48	98.49	95.17	**99.34**	99.00	98.87

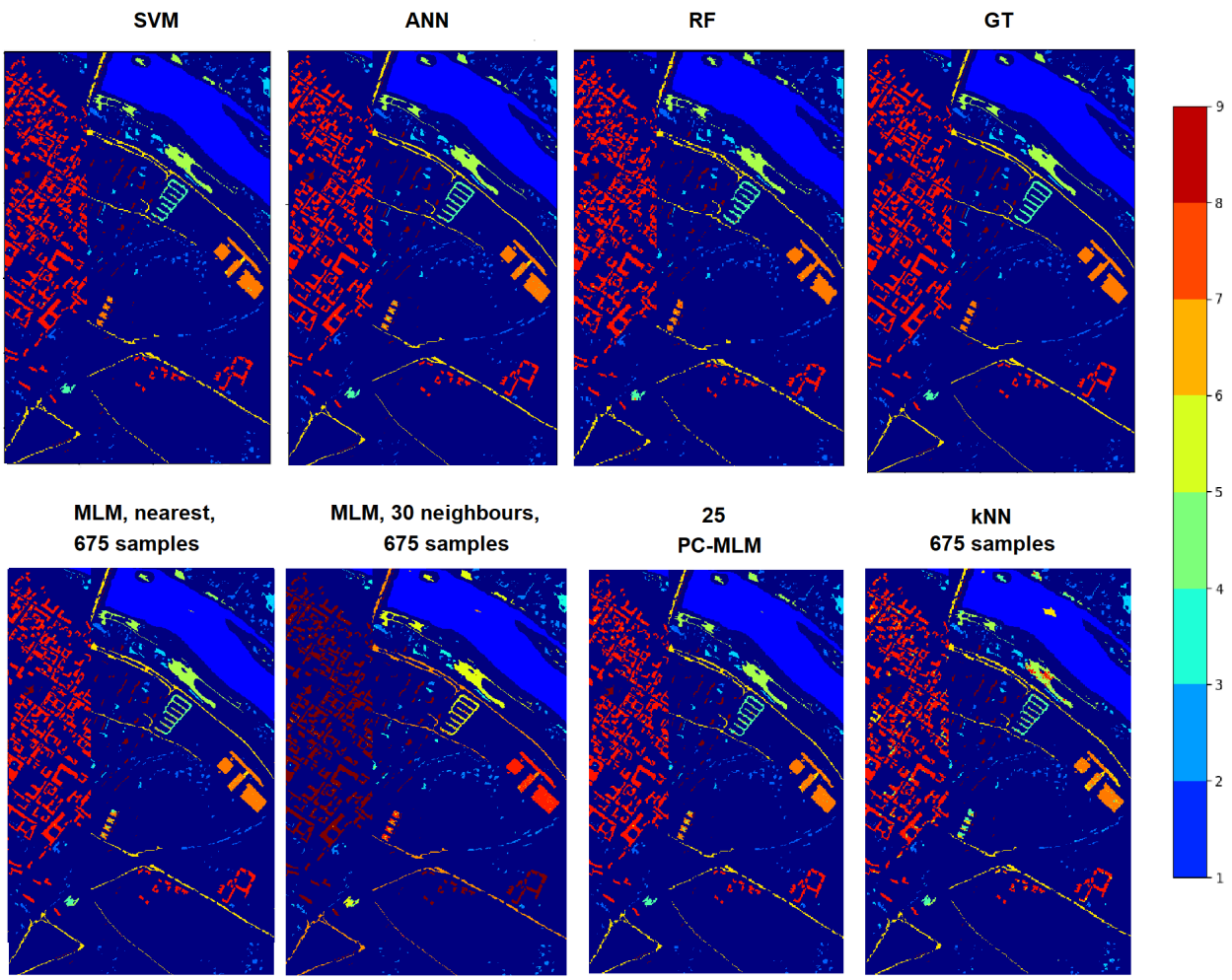

Figure 5. The classification maps of the Pavia Centre HS image. Maps are produced with the comparison classifiers. Numerical results of these maps are shown in the table 3.

The classification time comparison shows that the PC-MLM classifies all of the data sets faster than the MLM classifier. With Pavia data, the fastest classifier was kNN. RF was the fastest with Salinas and the Indian Pines comparison winner was ANN classifier.

The Accuracy rate comparison shows that the best results with Pavia data were achieved with SVM classifier. Smaller data sets reached best accuracy scores with RF classifier. Table 3 shows that the PC-MLM with 25 PC components reached promising results against the other classification methods in all of the categories. The Pavia data's results shows that with 25 PCA components and R size of 675, the PC-MLM is overall faster than SVM, ANN and RF and it reaches 98.49% accuracy.

Table 4. Indian Pines method comparison, 25 principal components, 30 nearest neighbours

	MLM	25 PC-MLM	25 PC kNN	SVM	ANN	RF
Number of training samples	X: 6149, R: 1200	X: 6149, R: 1200	1200	6149	6149	6149
Training time	1.51	2.12	**0.88**	5.14	6.82	5.39
Classification time	0.26	0.20	0.28	1.99	**0.01**	0.04
Accuracy rate	78.31	77.89	60.18	**88.78**	77.17	86.68

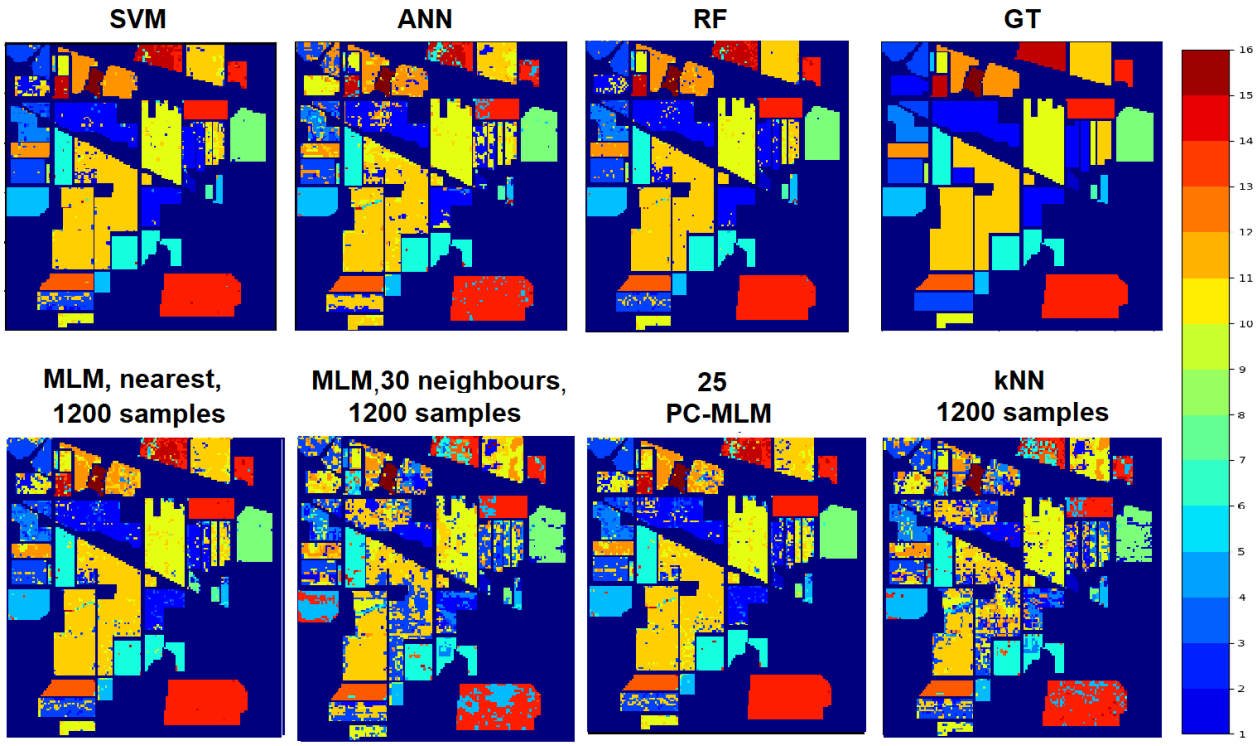

Figure 6. The classification maps of the Indian Pines HS image. Maps are produced with the comparison classifiers. Numerical results of these maps are shown in the table 4.

Table 5. Salinas method comparison, 25 principal components, 30 nearest neighbours

	MLM	25 PC-MLM	25 PC kNN	SVM	ANN	RF
Number of training samples	X: 35477, R:1200	X: 35477, R:1200	1200	32477	32477	32477
Training time	7.21	11.34	**3.85**	34.03	184.40	20.33
Classification time	1.62	1.40	0.06	22.57	0.93	**0.22**
Accuracy rate	93.01	92.82	85.31	93.82	92.57	**94.71**

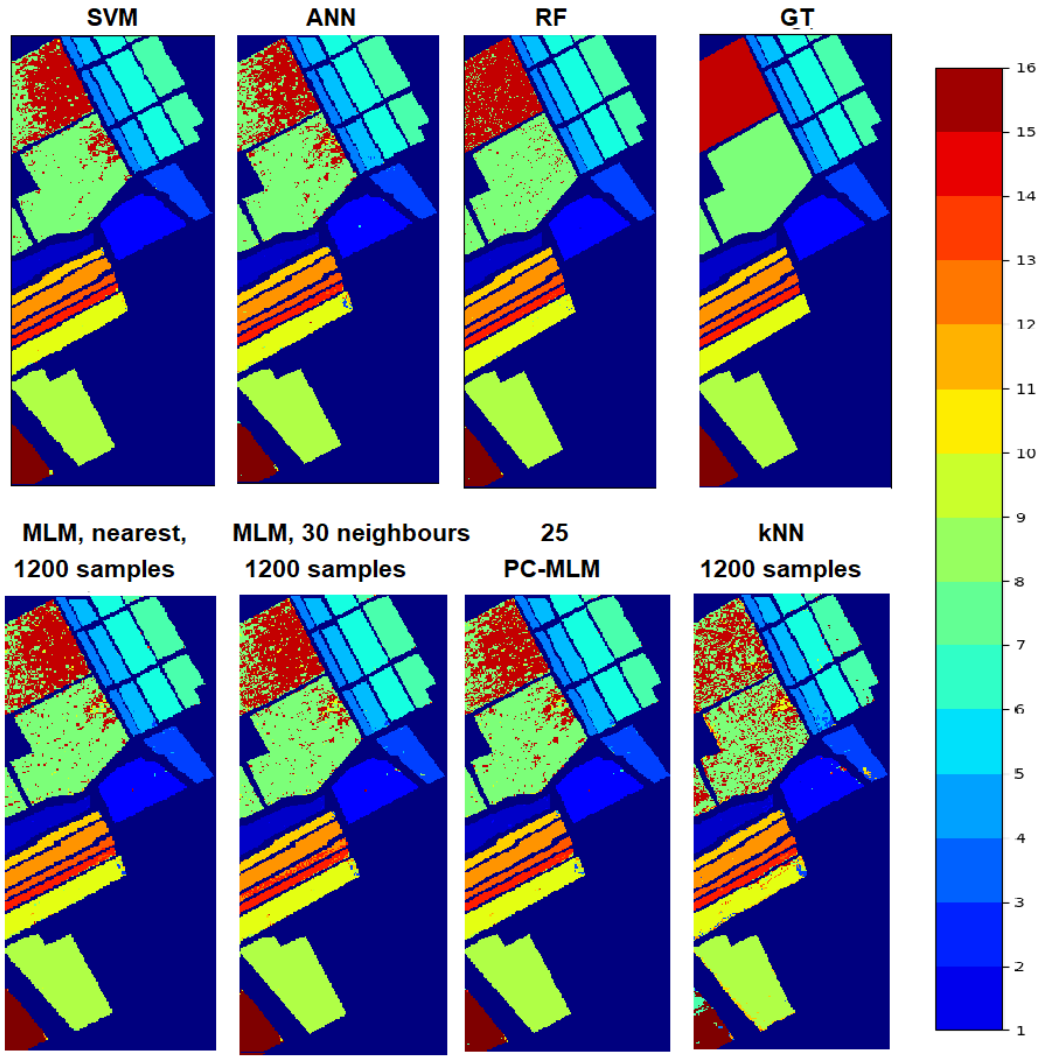

Figure 7. The classification maps of the Salinas HS image. Maps are produced with the comparison classifiers. Numerical results of these maps are shown in the table 5.

The previous results shows that the MLM and PC-MLM classifiers can reach promising accuracy rates against the other classifiers and decrease the time of training and classification. Those advances can be seen on results of all of the data sets, regardless of the size of the data, but the best results can be accomplished with larger ones.

3.3 The performance of the MLM and PC-MLM models

Figure 8 shows the comparison between the PC-MLM classifier and MLM classifier with randomly picked training points (randomised MLM). The reference set R was formed with 6-30 principal components. The amount of the randomly picked training points were equal with the size of the R.

The results of each data set is presented separately on rows in figure 8. The left side represents the MLM and PC-MLM nearest neighbour classifiers. On the right side, there are the results of the MLM and PC-MLM 30 neighbour classifiers. The markers on the lines on the figures represents the amount of principal components, x-axis shows the number of selected training points and y-axis shows the accuracy rate percent or time in seconds.

Results were measured 10 times and the final results are the average of 10 results. The training data X was randomised in every round for the randomised MLM.

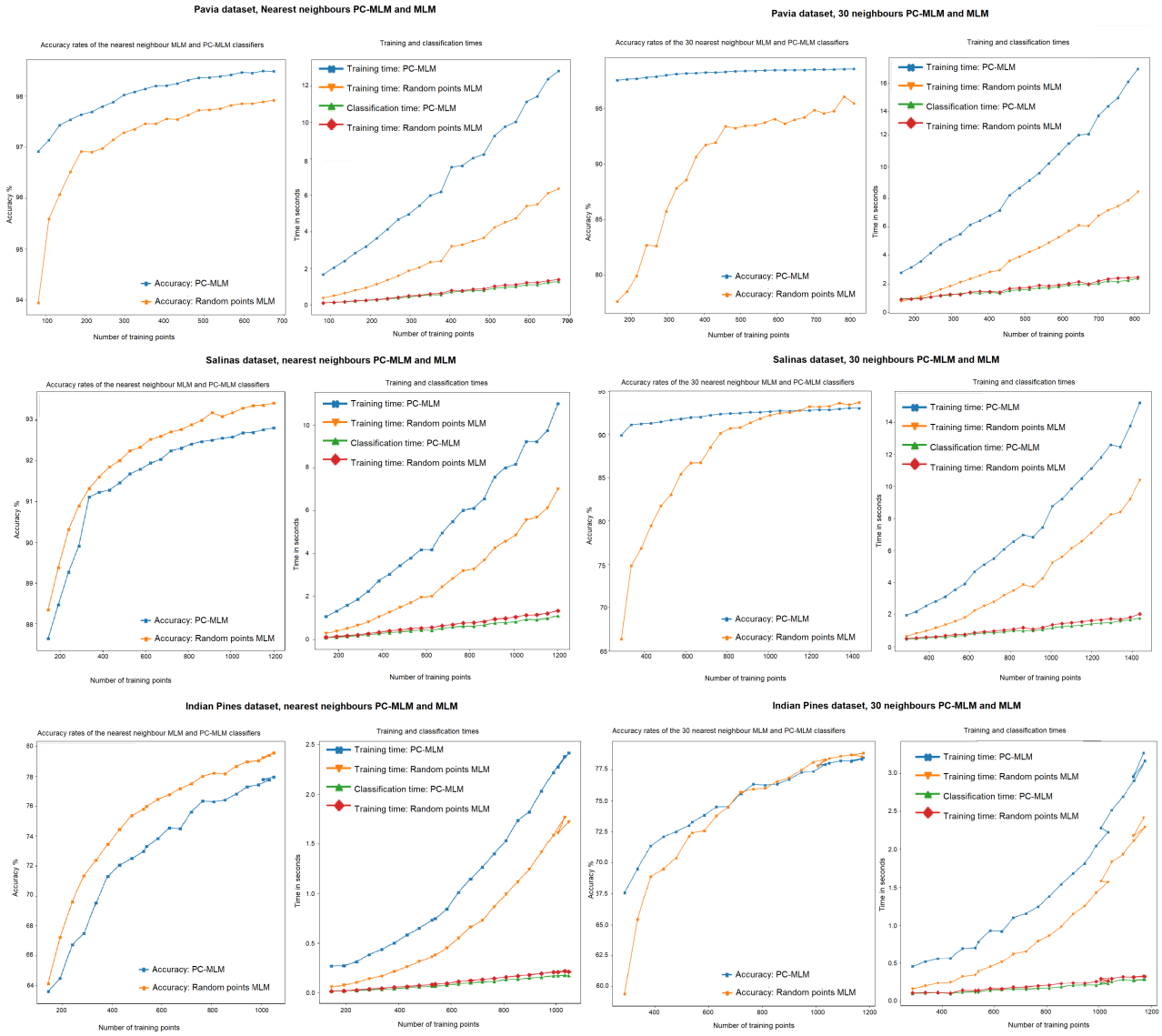

Figure 8. Pavia, Salinas and Indian Pines HS images, the comparison of the PC-MLM and randomised MLM with nearest and 30 nearest neighbour classifiers. The accuracy rate and the number of training points of different amount of principal components.

Figure 8 confirms an observation that the accuracy rate increases when the size of the training points increases. The structure of the data affects on the results. If the data set is large and it has less ground truth classes, it will most likely perform better than a smaller data set with high number of classes.

Table 6 shows that with Pavia data, the PC-MLM reached 97.54% accuracy rate with only 6 principal components, which means 162 selected training points from the whole training data (X) of 88891 samples. With the same amount of training points, the accuracy rate of randomised MLM was 96.47%. The Pavia data set's accuracy rate was better with the PC-MLM in all of the evaluation rounds with different amounts of selected training points. The best comparable accuracy rate of this comparison was nearest neighbour PC-MLM's 98.48%

with 675 training points. The result of the randomised MLM with same amount of training points was 97.92%. This PC-MLM's result can be improved by increasing the number of neighbours; the results in the tables 6, 8 and 7 shows, that the increased number of neighbours decreases the accuracy of the MLM classifier, but it may have a positive impact on the accuracy of the PC-MLM model.

Table 6. Pavia Centre, the number of the principal components, size of the R and the accuracy rates: PC-MLM nearest neighbour ans 30 neighbours classifiers against MLM nearest neighbour and 30 neighbours classifiers. The size of the training set X: 88 891 samples.

PC	The R size	PC-MLM nearest	MLM nearest	PC-MLM 30 neighbours	MLM 30 neighbours
3	81	96.91	95.20	46.50	73.16
4	108	97.13	95.24	70.35	74.72
5	135	97.43	96.11	70.56	75.92
6	162	97.54	96.47	97.54	76.85
10	270	97.88	97.14	97.88	84.97
15	405	98.02	97.55	98.23	90.78
20	540	98.39	97.75	98.37	93.41
25	675	98.48	97.92	98.49	94.48
30	810	-	-	98.55	95.33

We can see from the tables 6,7, 8 and figure 8, that the results of the Pavia data differs from the results of the other data sets. The PC-MLM has better accuracy rates than randomized MLM in Pavia results. The trend of the increasing accuracy rate with the increments of the size of R is similar with all of the data sets. Salinas and Indian Pines results (figure 8, tables 7 and 8) shows, that the accuracy rate of the randomised MLM exceeds over the accuracy rate of the PC-MLM's accuracy.

With Salinas data, the MLM with randomised training points performed better on the accuracy rate comparison. For example with 25 principal components, the results were 92.82% (PC-MLM, 30 neighbours) and 93.41% (randomised MLM, nearest neighbour). Indian Pines data followed the similar trend with the Salinas data, but the difference of the accuracy rates was bigger. With 25 components, the MLM with nearest neighbours classifier reached 79.52% accuracy, where the PC-MLM could reach only 77.94% accuracy.

Table 7. Salinas, the number of the principal components, size of the R and the accuracy rates: PC-MLM nearest neighbour ans 30 neighbours classifiers against MLM nearest neighbour and 30 neighbours classifiers. The size of the training set X: 32 477 samples.

PC	The R size	PC-MLM nearest	MLM nearest	PC-MLM 30 neighbours	MLM 30 neighbours
3	144	87.63	88.34	14.15	48.21
4	192	88.46	89.38	49.06	52.52
5	240	89.27	90.31	49.07	61.50
6	288	89.90	90.89	89.90	68.66
10	480	91.45	92.00	91.46	81.41
15	720	92.25	92.70	92.23	88.82
20	960	92.55	93.08	92.61	92.12
25	1200	92.80	93.41	92.82	93.01

The training and classification time (fig. 8, table 3) follows the same pattern with all of the data sets. The randomised MLM was faster on the training, but more time-consuming on the classification than the PC-MLM. For example the average classification time of with Pavia data and 675 training points was 2.06 seconds with randomized MLM, PC-MLM performed the same classification tasks on average of 1.82 seconds. The according average training times were 12.39 seconds (randomised MLM), and 5.92 seconds (PC-MLM).

Table 8. Indian Pines, the number of the principal components, size of the R and the accuracy rates: PC-MLM nearest neighbour ans 30 neighbours classifiers against MLM nearest neighbour and 30 neighbours classifiers. The size of the training set X: 6149 samples.

PC	The R size	PC-MLM nearest	MLM nearest	PC-MLM 30 neighbours	MLM 30 neighbours
3	144	63.61	65.15	12.49	48.29
4	192	64.49	67.23	34.78	53.49
5	240	66.73	69.60	36.93	56.53
6	288	67.49	71.37	67.46	61.41
10	480	72.49	75.53	72.35	71.18
15	675	74.49	77.15	74.47	74.68
20	900	76.79	78.65	76.63	77.58
25	1050	77.94	79.52	77.89	78.31
30	1170	-	-	78.31	78.54

4. DISCUSSION

The MLM and PC-MLM classifiers seems to perform well with hyperspectral images. The accuracy results are promising against the other classifiers. MLM and PC-MLM reaches comparable training and classification times towards most of the other methods. Those advances can be seen on all of the data sets, regardless of the size of the data set. However, the classifiers were optimized with the largest Pavia data set, which may have an affect on the performance with smaller data sets, like Salinas and Indian Pines.

Compared to the original MLM classifier, the new PC-MLM framework slightly increases the number of hyper-parameters with two obligatory and one optional parameter (distance method, the number of principal components and the number of neighbours). One of the benefits of the original MLM is that it has only one hyper-parameter to tune, which is the size of the randomly selected training points (R)[9] . Our model extends the amount of parameters, but it seems to still maintain MLM's overall advances of the easy implementation, fast classification time and good accuracy rate compared to the other models (see tables 3, 5, 4 and classification maps 5, 7 and 6). With the reference classifiers, the hyper-parameter tuning was time consuming, and especially the ANN classifier was hard and time-consuming to optimize, which confirms previous observations of using ANN classifier[22] . The RF and SVM classifier were easier and faster to optimise.

When we compare the PC-MLM's performance against the MLM with randomly selected points, we can see from the figure 8, that the PC-MLM is more time-consuming on the training, but it is faster on the classification phase. The reason, why the PC-MLM is fast in classification, is inside of the implementation of the reference point selection (algorithm 1). We present in the MLM chapter 2.1, that the optimization problem can be solved with the nearest neighbours method. After performing the reference point selection algorithm 1, our R and T are sorted and organised class-by-class. When we are picking randomly samples from the X, the R and T will remain unorganised. On the classification phase, the PC-MLM's $\delta(y_n, T)$ can be sorted faster than the MLM's $\delta(y_n, T)$.

The performance of all of the classifiers followed similar trend with each of the HS images. For example, the Indian Pines had the lowest performance (table 4), while the Pavia HS image reached the highest accuracy scores (table 3). One explanation for this trend can be that all of the classifiers were optimised only for the largest Pavia data set.

The MLM and the PC-MLM can classify data with significantly smaller amount of reference points (R) than SVM, ANN and RF classifiers. The MLM and PC-MLM uses the information from the whole training set X on the model $\hat{\mathbf{B}}$ (algorithm 2), but in the prediction phase, the classifier calculates only the distances between the new data and the reference set R. Model utilises the $\hat{\mathbf{B}}$ with the new distances and selects the labels for the new data with nearest neighbour method (algorithm 3). Therefore it is relatively simple and computationally less complex machine learning model compared to the reference models.

The study confirms previous findings that the RF classifier can perform well on the land-cover classifications and remote sensing[23] . We trained SVM, ANN and FR models with training sets that had 60% amount of the HS image data points. With RF classifier, it is a good size for avoiding the over fitting[23] . Compared to the MLM and PC-MLM classifiers training points, the size difference is remarkable. MLM and PC-MLM had only 675 samples and RF 88 891 samples on the Pavia classification task. The accuracy results were close to each other, but the MLM and PC-MLM could perform results faster than RF classifier.

Since there are similarities with the kNN classifier, it was an interesting to see, how the kNN classifier could perform with the same intentionally selected training points than the MLM and PC-MLM classifiers. The results shows, that the MLM's and PC-MLM's model performed more accurately than the kNN classifier.

As an limitation for the PC-MLM framework, these tests were done only with three HS images and only relatively well optimized classifiers. The PC-MLM framework is now ready for the future improvements. For example, it is an interesting question to solve, how the structure and the size of the data set, and the geometry of each class affects to the accuracy.

The HS images class-by-class principal component visualisations (fig. 1, 2 and 2) reveals the geometry of each class and the selected training points. The labels and amounts of the samples can be seen on the ground truth tables 1, 2. The prediction results are shown in the classification maps shows (figs. 5, 7 and 6).

With Pavia data, the amount of the samples in the ground truth labels had even distribution (table 1), the selected training points and their position can be seen on figure 1. Figure 1 shows that Pavia's training points had less scatter than the classes in the figures 2 (Salinas) and 3 (Indian pines). The reference point selection algorithm 1 worked well with Pavia data. Salinas and Indian Pines were more difficult to predict. Salinas had more classes and samples than Pavia, but the image size was significantly smaller and the amount of spectral bands were twice as big as Pavia's.

Indian Pines was the smallest data set with the biggest variance on the amount of samples per class. The principal component visualisation (fig. 3) shows that the training points were scattered. Figure also shows, how the selected training points are representing the geometry of each class. The classification map 6 reveals, that the mistakes in the classification of the data set covers all of the classes, which differs from, for example, the classification map of Salinas 2, where the mistakes are more clearly focused on few categories.

Based on these observations, it would be interesting to research more the relationship between the geometry of the classes, the structure and size of the data and the intentional reference point selection method (algorithm 1). For example, when the minimum and maximum positions were moved closer to the median position, the optimization of the new locations were done with Pavia data. This movement increased the accuracy rate, and it was necessary because of the noise in the data. It would have been interesting to see, how the accuracy rate results would have behaved, if this optimisation has been done individually to each data set.

We tested MLM and PC-MLM with a large HS image and small number of classes (Pavia Centre), a large HS image with relatively large number of classes (Salinas) and with a small HS-image and relatively large number of classes with wide range on the amount of samples per classes (Indian Pines). It might have been interesting to see how it would have affected on all of the classifiers if we would have performed these tests with a small data set and small number of classes. Some of the previous classification studies (eg.[24]) has cleaned the Indian Pines data, by limiting the classes to the largest ones, since there are classes that has only limited amount of samples (see from table 2). It would have been interesting to try this experiment again with Indian Pines data and limited classes.

Another idea to improve the PC-MLM is to study the relation between the number of components and number of neighbours. The results indicates that with a low number of principal components it might be good

to use the nearest neighbour in the predicting phase, but when we are optimising the classifier, there is a point with the number of principal components, where the number of neighbours should be increased to achieve better accuracy rate. The way we select the positions of the minimum and maximum values seems also have an affect on the number of neighbours, which means that there might be another place to improve the PC-MLM method. Finding out these optimisation rules might improve and simplify the optimisation process on the future.

The feature extraction (FE) methods are important in hyperspectral data classification. Different FE methods can be used for extraction of geometric structures, shape and texture from the HSI[25] .The aim of the FE is to get an excellent representation from the original data. On this study, we used the PCA, which is a popular method. Our findings confirms the previous research of the FE methods. With those methods, we can reduce the number of the training samples (R size) and reduce the computing time, without decreasing the accuracy rate. The FE is an important step before using the HS image classification methods[10,26] .

5. CONCLUSIONS

The MLM is an effective tool for HS image classification. Comparison against four other spectral classifiers reveals, that MLM is an easy to implement, it provides good accuracy rates and while classifying, it consumes less time than the reference models SVM, ANN, RF and kNN.

MLM's accuracy rate and classification time can be improved by selecting the reference points class-by-class intentionally with principal component analysis to respect the geometry of each class. The proposed framework for improving the performance is called the PC-MLM. However, the improvement is dependent on the size of the data. The results indicates, that MLM's performance can be improved with large HS images, but with smaller HS images, the basic version of the MLM might perform better than the PC-MLM.

ACKNOWLEDGMENTS

This study is partly funded by Jane and Aatos Erkko Foundation (Grant No. 170015) and Academy of Finland (Grant No. 327862).

REFERENCES

[1] Camps-Valls, G. and Bruzzone, L., "Kernel-based methods for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing* **43**(6), 1351–1362 (2005).

[2] Camps-valls, G., Tuia, D., and Bruzzone, L., "Advances in hyperspectral image classification," *IEEE Signal Processing Magazine* (January), 45–54 (2013).

[3] Khan, M. J., Khan, H. S., Yousaf, A., Khurshid, K., and Abbas, A., "Modern Trends in Hyperspectral Image Analysis: A Review," *IEEE Access* **6**, 14118–14129 (2018).

[4] Pölönen, I., Rahkonen, S., Annala, L., and Neittaanmäki, N., "Convolutional neural networks in skin cancer detection using spatial and spectral domain," **10851**, 10 (2019).

[5] Tuominen, S., Näsi, R., Honkavaara, E., Balazs, A., Hakala, T., Viljanen, N., Pölönen, I., Saari, H., and Ojanen, H., "Assessment of classifiers and remote sensing features of hyperspectral imagery and stereo-photogrammetric point clouds for recognition of tree species in a forest area of high species diversity," *Remote Sensing* **10**(5), 1–28 (2018).

[6] Ghamisi, P., Plaza, J., Chen, Y., Li, J., and Plaza, A. J., "Advanced Spectral Classifiers for Hyperspectral Images: A review," *IEEE Geoscience and Remote Sensing Magazine* **5**(1), 8–32 (2017).

[7] He, L., Li, J., Liu, C., and Li, S., "Recent Advances on Spectral-Spatial Hyperspectral Image Classification: An Overview and New Guidelines," *IEEE Transactions on Geoscience and Remote Sensing* **56**(3), 1579–1597 (2018).

[8] Hughes, G. F., "Comments "on the Mean Accuracy of Statistical Pattern Recognizers"," *IEEE Transactions on Information Theory* **IT-15**(3), 420–423 (1969).

[9] de Souza, A. H., Corona, F., Barreto, G. A., Miche, Y., and Lendasse, A., "Minimal Learning Machine: A novel supervised distance-based approach for regression and classification," *Neurocomputing* **164**, 34–44 (2015).

[10] Zhao, B., Ulfarsson, M. O., Sveinsson, J. R., and Chanussot, J., "Unsupervised and Supervised Feature Extraction Methods for Hyperspectral Images Based on Mixtures of Factor Analyzers," *Remote Sensing* **12**(7), 1179 (2020).

[11] Yanni, Dong (IEEE), Du, Bo, (IEEE), and Zhang, L. I., "Dimensionality Reduction and Classification of Hyperspectral Images Using Ensemble Discriminative Local Metric Learning," *Indonesian Journal of Electrical Engineering and Computer Science* **3**(3), 503–511 (2016).

[12] Plaza, A., Martínez, P., Plaza, J., and Pérez, R., "Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations," *IEEE Transactions on Geoscience and Remote Sensing* **43**(3), 466–479 (2005).

[13] Mesquita, D. P., Gomes, J. P., and Souza Junior, A. H., "Ensemble of Efficient Minimal Learning Machines for Classification and Regression," *Neural Processing Letters* **46**(3), 751–766 (2017).

[14] Varoquaux, G., Buitinck, L., Louppe, G., Grisel, O., Pedregosa, F., and Mueller, A., "Scikit-learn," *GetMobile: Mobile Computing and Communications* **19**(1), 29–33 (2015).

[15] Graña, M., Veganzons, M., and Ayerdi, B., "Grupo de iIntelligencia Computational (CIC)." Hyperspectral Remote Sensing Scenes http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes. (Accessed 9 May 2020).

[16] Xie, F., Li, F., Lei, C., Yang, J., and Zhang, Y., "Unsupervised band selection based on artificial bee colony algorithm for hyperspectral image classification," *Applied Soft Computing Journal* **75**, 428–440 (2019).

[17] Sawant, S. S. and Manoharan, P., "Unsupervised band selection based on weighted information entropy and 3D discrete cosine transform for hyperspectral image classification," *International Journal of Remote Sensing* **41**(10), 3948–3969 (2020).

[18] Petropoulos, G. P., Kontoes, C. C., and Keramitsoglou, I., "Land cover mapping with emphasis to burnt area delineation using co-orbital ALI and Landsat TM imagery," *International Journal of Applied Earth Observation and Geoinformation* **18**(1), 344–355 (2012).

[19] Breiman, L., "Bagging predictors," *Machine Learning* **24**(2), 123–140 (1996).

[20] Breiman, L., "Random forests," *Machine learning* (45), 5–32 (2001).

[21] Breiman, L., "Arcing Classifiers," *Annals of Statistics* (26), 801–824 (1996).

[22] Raczko, E. and Zagajewski, B., "Comparison of support vector machine, random forest and neural network classifiers for tree species classification on airborne hyperspectral APEX images," *European Journal of Remote Sensing* **50**(1), 144–154 (2017).

[23] Rodriguez-Galiano, V. F., Ghimire, B., Rogan, J., Chica-Olmo, M., and Rigol-Sanchez, J. P., "An assessment of the effectiveness of a random forest classifier for land-cover classification," *ISPRS Journal of Photogrammetry and Remote Sensing* **67**(1), 93–104 (2012).

[24] Zhao, B., Ulfarsson, M. O., Sveinsson, J. R., and Chanussot, J., "Unsupervised and Supervised Feature Extraction Methods for Hyperspectral Images Based on Mixtures of Factor Analyzers," *Remote Sensing* **12**(7), 1179 (2020).

[25] Imani, M. and Ghassemian, H., "An overview on spectral and spatial information fusion for hyperspectral image classification: Current trends and challenges," *Information Fusion* **59**(October 2019), 59–83 (2020).

[26] Casasent, D. P. and Chen, X.-w., "Waveband selection for hyperspectral data: optimal feature selection," in [*Optical Pattern Recognition XIV*], **5106**, 259 (2003).