Analysing Effect of t-SNE and 1-D CNN on Performance of Hyperspectral Image Classification

Hariharan S
Department of Electronics and Communication Engineering,
Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham,
India
hariharan.india98@gmail.com

Dakshin T K
Department of Electronics and Communication Engineering,
Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham,
India
dakshintk@gmail.com

Vijayaraghav K
Department of Electronics and Communication Engineering,
Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham,
India
gkvraghav@gmail.com

Rajesh C B
Department of Electronics and Communication Engineering,
Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham,
India
cb_rajesh@cb.amrita.edu

Vignesh M
Department of Electronics and Communication Engineering,
Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham,
India
vickeymoorthy1999@gmail.com

Article History: Received: 10 November 2020; Revised 12 January 2021 Accepted: 27 January 2021; Published online: 5 April 2021

***Abstract*—** Feature extraction is a crucial step in Hyperspectral Image classification that aids in processing data effectively without losing relevant information. This step is essential when dealing with images with high dimensions because they suffer from Hughes phenomenon or the curse of high dimensionality. This phenomenon occurs in high dimensional datasets where the number of training samples is limited. In this paper, we have studied the influence of feature extraction techniques in HSI classification. We have compared the efficiency of three widely used techniques, namely Principal Component Analysis, t-Stochastic Neighbor Embedding and Convolutional Neural Network. Overall classification accuracy for PCA when used with KNN, a commonly used classification algorithm was found to be 69.79% while t-SNE with KNN was 71.04%. When CNN was used for feature extraction, its outperformed t-SNE and PCA with a wide margin with classification accuracy reaching as high as 95.06%.

Keywords—— *feature extraction, convolutional neural network, t-SNE, principal component analysis, hyperspectral image classification*

I. Introduction

Hyperspectral image classification is an emerging technology applied in geology, mining, ecology and surveillance. Each pixel in the image contains the entire spectrum of the scene which aids in accurate mining of all available information from the scene.

In Machine Learning problems that have limited quantity of data samples of high dimensions, an enormous quantity of training data is necessary. The predictive accuracy of the algorithm increases as the number of features increases but then decreases, which is known as Hughes phenomenon [1].

Since there is a huge amount of information present in a scene and small number of samples available Hyperspectral Image Classification becomes a daunting task. This issue is addressed by reducing the complexity of the data set using feature extraction [2].

In feature extraction, the number of features in the dataset is reduced by conceiving new features from the original ones [3]. The new set of reduced features comprises information present in the original features.

Principal Component Analysis (PCA) is an extensively used linear feature extraction technique where a mixture of input features that encompasses all the available information is obtained from the input [4]. PCA does this by preserving the crucial parts in the data that exhibit maximum variance.

t-distributed Stochastic Neighbor (t-SNE) Embedding is a manifold learning feature extraction technique utilized particularly for high dimensional datasets [5]. Unlike PCA which is mathematical, t-SNE is probabilistic. It is a variation from stochastic neighbor embedding as important visualizations are obtained by reducing the bias of crowding points in the middle of the plot [6]. The utilization of deep learning has significantly increased in recent years due to its exceptional performance in terms of classification accuracy. For image recognition and classification challenges, Convolutional Neural Networks are used extensively. The visual system of humans has influenced the construction of CNN architecture [7]. They prove to be an excellent combination of feature extractors and classifiers. In our paper, we assess the influence of feature extraction in HIS classification by comparing the performance of three different classes of FE techniques. The major contributions of this paper are: 1) Identify if PCA, t-SNE or CNN provides better accuracy for classification. 2) Verify results obtained using other datasets.

II. METHODOLODY

Three feature extraction techniques belonging to different categories are considered.

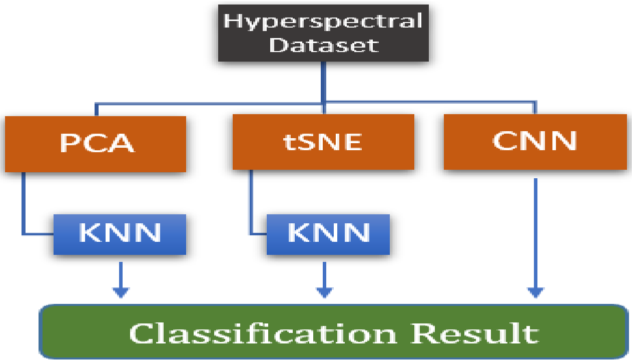

Fig. 1. Methodology flow diagram

A. Principal Component Analysis

The adjoining bands of a hyperspectral image are extremely correlated and contain redundant information. PCA finds the optimum linear combination of the bands of the image which expresses the variation of image pixel values [8].

To perform PCA, the data has to be standardized. first. This is done in order to obtain a gaussian form with standard deviation 1 and mean 0. The average of pixel values is subtracted from each pixel and divided by deviation. This is followed by calculating the covariance matrix of the input image.

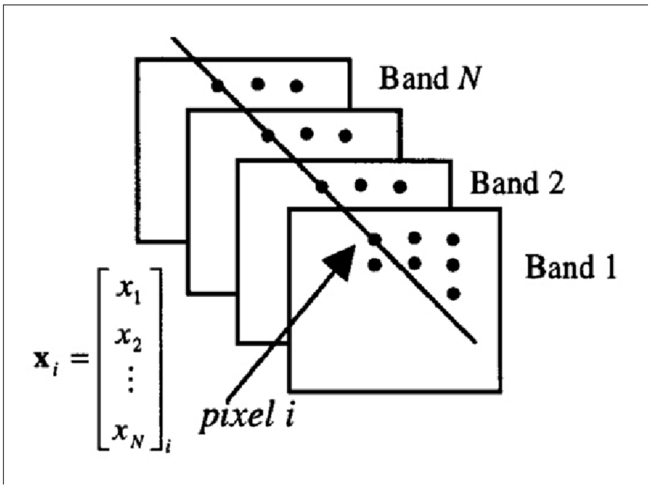
Band N
Band 2
Band 1
$\mathbf{x}_i = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}_i$
pixel i

Fig. 2. Pixel vector in PCA. (*Gonzales and Woods (1993)*)

Covariance is obtained using the formula

$$C_x = \frac{1}{N}\sum_{j=1}^{N} (x_i - a)(x_i - a)^T$$

where:
x_i is image pixel vector
N = a*b
a is the total quantity of rows and b is the total quantity of columns.

The Eigen decomposition of the covariance matrix is obtained. The eigenvectors and eigenvalues are then ranked in descending order based on the maximum variance. The top k eigenvectors obtained from the result of the scree plot represents the new bands which are an orthogonal transformation of the original image vectors. The original image can then be transformed via the new k-dimensional feature subspace.

B. t- Distributed Stochastic Neighbour Embedding

t-SNE is a non-linear feature extraction technique used chiefly for high dimensional datasets [9]. The algorithm works as follows. The probability of similitude of data points in low dimensional and high dimensional space is calculated. This similarity is determined as the conditional probability that one point would choose another as a neighbour if they were chosen with respect to the probability density under normal distribution centred at first point. This difference between conditional probabilities (which represents similarity between two points) is minimized to the fullest extent for the ideal representation of points in the lower dimensional space. The sum of Kullback Leibler divergence of all data points is curtailed by the gradient descent method to calculate the minimization of the sum of the difference of conditional probabilities [10,11].

In t-SNE Student t-distribution is utilized. The joint probability q_{ij} for this distribution is defined as

$$q_{ij} = \frac{(1+\|y_i - y_j\|^2)^{-1}}{\sum \quad k \neq 1(1+\|y_k - y_l\|^2)^{-1}}$$

The cost function in this case is defined as:

$$C_{t-SNE} = KL(P||Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}}$$

In low dimensional space, pairwise similarities are given by:

$$q_{ij} = \frac{exp\ (-\|y_i - y_j\|^2)}{\sum_{k\neq l} \quad exp\ (-\|y_k - y_l\|^2)}$$

For high dimensional space, pairwise similarities is defined by:

$$p_{ij} = \frac{exp\ (-\frac{\|x_i - x_j\|^2}{2\delta_i{}^2})}{\sum_{k\neq l} \quad exp\ (-\frac{\|x_i - x_j\|^2}{2\delta_i{}^2})}$$

Thus t-SNE matches high dimensional data to low dimensional space and tries to find patterns in the data by analysing and classifying based on the clusters obtained based on the data points similarity with numerous features.

C. 1-D CNN

Convolutional Neural Networks are extensively used in image processing and have proved to exhibit excellent performance for Hyperspectral Image Classification. To implement CNN for feature extraction, an architecture [12] with five layers is used. The network consists of input layer, a Convolutional layer, a Max Pooling layer, Fully Connected layer and an Output layer. Conventional CNNs utilize spatial and spectral data for classification. To exhibit efficiency of CNN, the spectral signature data of each pixel is considered.

1. *Training:*

We initialize the trainable parameters between -0.05 and 0.05. The process of training includes two crucial steps: Forward propagation and Backward propagation. Forward propagation computes the classification result with current parameters. Backward propagation updates the parameters after each iteration to limit the cost function to the minimum.

2. *Forward propagation*:

Hyperbolic tangent function is implemented as the activation function for Convolutional layer and fully connected layer. The maximum function is utilized in the Max pooling layer. Owing to the fact that the CNN output is a multiclass classifier, the result of the FC layer is given to Softmax layer that results in a distribution over the number of classes that needs to be identified. The batch size is fixed as 32.

3. *Backward propagation*:

Parameters that need to be trained are updated by utilizing gradient descent algorithm in back propagation. The cost is reduced once the first iteration is over by passing the resultant weights through each layer. The mathematical intuition for this is to determine partial derivative for weights in each layer[13]. In the architecture, C1 and M2 act ad trainable feature extractors

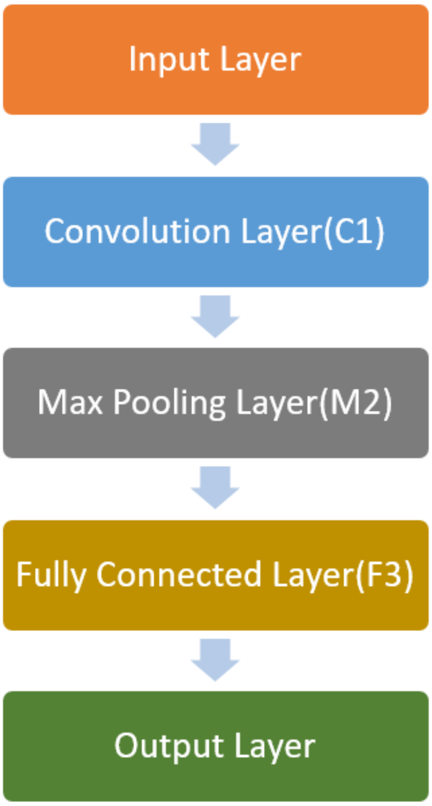

Fig. 3. Architecture of CNN

TABLE I. TRAINABLE PARAMETER IN EACH LAYER

Layer	**Parameters**	**Values**
Input	n1*1	n1= number of bands

Convolutional layer(C1)	Kernel size : k1*1
	Nodes: 20*n2*1
	Trainable parameters : 20*(k1+1)
		n1= number of bands
		k1 = n1/9
		n2=n1-k1+1

Max Pooling layer(M2)	Kernel size : k2*1
	Nodes: 20*n3*1
	Trainable parameters: 20*(k1 + 1)
		k2=n2/n3=5
		n3=n2/k2=40

Fully Connected layer(F3)	Nodes: n4 Trainable parameters : 20*(k1+1)	n4=100 (Arbitrary)
Output layer	Nodes: n5 Trainable parameters : (n4+1)*n5	n5=number of output classes

The loss function is given by

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n} \sum_{j=1}^{n_5} 1\{j = Y^{(i)}\} log\ (y_j^i)$$

where n is the number of samples used for training. Y is the output required.

As the number of iterations increase the difference between the actual output and desired output decreases until this discrepancy reaches minimum.

III. EXPERIMENTS AND RESULTS

A. Datasets

The Indian Pines data set was obtained using the AVIRIS sensor. The region covered is north-western Indiana. The dataset consists of 220 spectral channels in the visible and infrared spectrum. This covers the range 0.4 to 2.45 um. The image scene has a spatial resolution of 20m.

The data for development is obtained by dividing the data into training and testing samples which can be utilized for parameter tuning in the case of CNN. Each pixel is scaled uniformly between -1 and 1.

The other dataset, Salinas was obtained by Aviris sensor as well. It captures the Salinas valley scene and constitutes 3.7 m of spatial resolution. This scene consists of 220 spectral bands with 16 different classes.

B. PCA with KNN

The first k principal components from the result of the scree plot are selected from the 200 original bands available in the image. The hyperspectral image pixel values are stored as a vector whose length is the total number of pixels.

The result of PCA is then utilized by the KNN classification algorithm. KNN stands for K nearest neighbors. Here, K stands for the number of nearest neighbor pixels that each pixel uses to assess and vote the label of the chosen pixel. The measure used to find the similarity of closest point is Euclidean distance. The algorithm is run over different k values in order to find the optimum value exhibiting maximum accuracy.

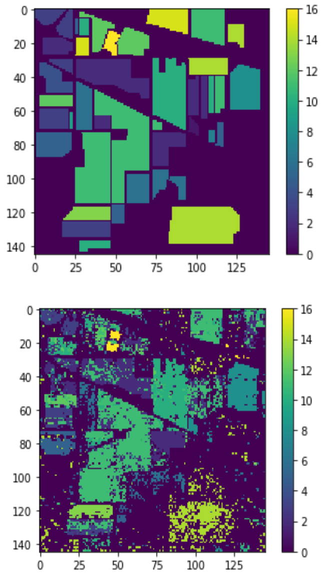

Fig. 4. Ground truth of Indian Pines dataset (left) Classified image output using PCA with KNN (right)

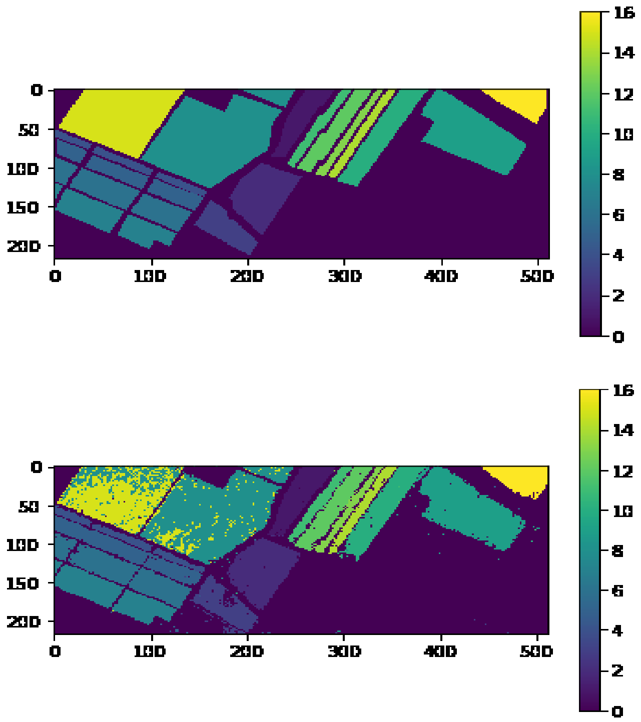

Fig. 5. Ground truth of Salinas dataset(left) and Classified image output using PCA with KNN (right)

C. t-SNE with KNN

The data is standardized before applying t-SNE. Perplexity is a tunable parameter that plays an imperative role in the performance of t-SNE algorithm. This value lies somewhere between 5 and 50. Multiple plots were analyzed and 30 was chosen for the experiment.

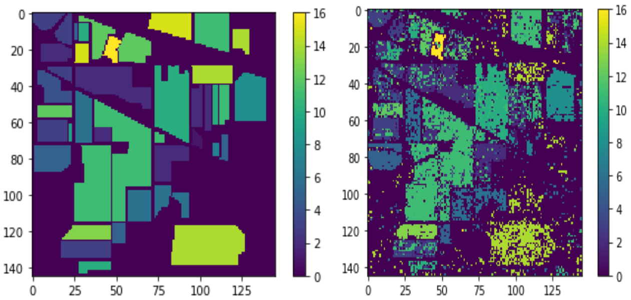

Fig. 6. Indian Pines ground truth image (left) vs Classified image output using t-SNE with KNN (right)

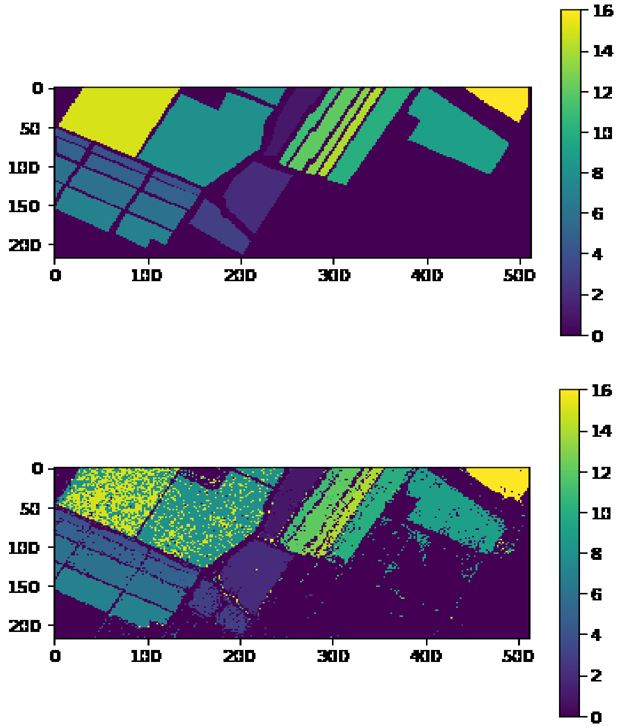

Fig. 7. Indian Pines ground truth image(sbove) vs Classified image output using t-SNE with KNN (below)

D. 1-D CNN

The dataset is split into training and testing data. The training dataset constitutes 50% whereas testing constitutes the remaining 50%. The data is standardized such that each data point lies in a particular range. The learning model parameters are then generated and transformed parameters are obtained before feeding into the neural network. Since convolutional neural networks require the categorical data to be converted into numbers, one hot encoding is done.

The batch size is taken as 32 for forward propagation. Since our objective is to extract crucial features from the plethora of data available, valid padding is done. This ensures that after each layer, the number of features reduces drastically to the most important ones. Dropout regularization is done by neglecting nodes in a random manner. This will cut down the cost of storage, time and interdependencies arising in the nodes.

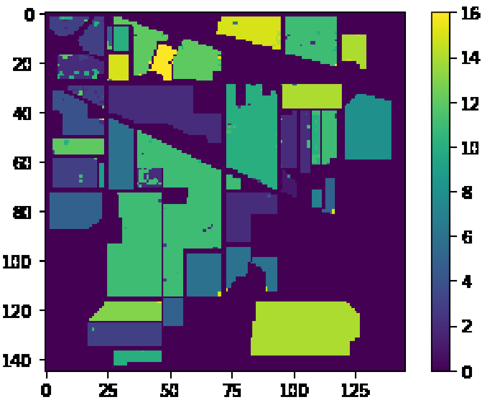

Fig. 8. Classified image output of 1-d CNN

E. Analysis

1) Indian Pines Dataset

TABLE II. COMPARISON OF ACCURACIES OF THE THREE FEATURE EXTRACTION TECHNIQUES UNDER REVIEW FOR INDIAN PINES DATASET

Method	**PCA with KNN**	**t-SNE with KNN**	**1D CNN**
Overall Accuracy (%)	69.79	71.04	95.06
Average Accuracy (%)	65.39	67.07	86.91
Kappa coefficient (%)	55.54	57.93	84.1
Class			
Undefined	85.27	84.22	93.79
Alfalfa	0	14.29	84.78
Corn-notil	56.67	57.79	93.84
Corn-mintill	48.35	56	91.81
Corn	3.45	27.14	87.76
Grass Pasture	55.92	67.95	88.82
Grass-trees	33.33	59.92	91.51
Grass-Pasture mowed	5	66.67	89.29
Hay-Windrowed	93.9	87.23	96.03
Oats	0	0	80
Soybean-notill	66.57	64.31	95.58
Soybean-mintill	73.56	67.95	69.94
Soybean-clean	25.12	37.14	95.62
Wheat	79.56	80.65	97.56

Woods	40.16	44.38	72.25
Buildings-Grass-Drives	0	2.5	57.51
Stone-Steel-Tower	67.74	68	91.4

2) *Salinas Dataset*

TABLE III. COMPARISON OF ACCURACIES OF THE THREE FEATURE EXTRACTION TECHNIQUES UNDER REVIEW FOR SALINAS DATASET

Method	**PCA with KNN**	**t-SNE with KNN**	**1-D CNN**
Overall Accuracy (%)	85.76	87.27	89.96
Average Accuracy (%)	89.2	86.55	91.33
Kappa coefficient (%)	82.4	80.29	86.85
Class			
Undefined	90.24	91.89	90.24
Brocoli_green_weeds_1	94.82	87.98	99.15
Brocoli_green_weeds_2	97.78	93.82	98.15
Fallow	83.12	82.14	68.17
Fallow_rough_plow	87.61	87.39	88.38
Fallow_smooth	93.11	88.63	97.72
Stubble	97.9	91.41	98.53
Celery	98.79	94.49	99.16
Grapes_untrained	81.56	75.18	93.58
Soil_vinyard_develop	92.75	90.61	97.58
Corn_green_weeds	89.65	88.92	94.51
Lettuce_romaine_4wk	85.99	86.62	73.03
Lettuce_romaine_5wk	95.59	90.62	98.5
Lettuce_romaine_6wk	83.45	86.69	99.34
Lettuce_romaine_7wk	84.43	76.95	94.77
Vinyard_untrained	61.68	65.9	64.94
Vineyard_vertical_trells	97.96	92.22	96.9

IV. CONCLUSION

In our paper, we have done a detailed analysis of three feature extraction techniques that are prevalently used for image classification problems. The datasets used were Indian Pines and Salinas datasets which are standard available datasets used for hyperspectral image processing. Experimental analysis of Indian Pines dataset indicates that CNN outperforms PCA and t-SNE by a huge margin with an overall accuracy of 91.41%. But this comes at the cost of increased time and computational complexity. When we compare t-SNE with PCA, there is only a slight improvement in the performance of t-SNE with 71.04% compared to 69.79% of PCA. This increase in accuracy is not justified by the enormous computational complexity that t-SNE has. This deviance in performance of t-SNE despite it being cited as a novel technique might be the fact that t-SNE is better suited for visualization of high dimensional datasets in lower dimensional space rather than in classification tasks. In the case of Salinas dataset, the results agree with results of Indian Pines dataset but since the scene consists of similar classes of vegetation, all three feature extraction techniques exhibit similar performance. The average accuracy also tends to be higher owing to the similarity of the classes. In conclusion, CNN can be used if high accuracy is required despite computational cost, PCA can be used if computational resources are not available at the cost of accuracy and t-SNE can be used for visualization tasks.

REFERENCES

[1]. G. Hughes, “On the mean accuracy of statistical pattern recognizers,” IEEE Trans. Inf. Theory, vol. IT-14, no. 1, pp. 55–63, Jan. 1968

[2]. S. Reshma, S. Veni and J. E. George, "Hyperspectral crop classification using fusion of spectral, spatial features and vegetation indices: Approach to the big data challenge," *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI),* Udupi, 2017, pp. 380-386.

[3]. J. B. Dias et al., “Hyperspectral remote sensing data analysis and future challenges,” IEEE Geosci. Remote Sens. Mag., vol. 1, no. 2, pp. 6–36, Feb. 2013.

[4]. M. Farrell and R. Mersereau, “On the Impact of PCA Dimension Reduction for Hyperspectral Detection of Difficult Targets,” *IEEE Geoscience and Remote Sensing Letters*, vol. 2, no. 2, pp. 192–195, 2005.

[5]. N. Rogovschi, J. Kitazono, N. Grozavu, T. Omori, and S. Ozawa, “t-Distributed stochastic neighbor embedding spectral clustering,” *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017.

[6]. Miao, A., Zhuang, J., Tang, Y., He, Y., Chu, X. and Luo, S. (2018). Hyperspectral Image-Based Variety Classification of Waxy Maize Seeds by the t-SNE Model and Procrustes Analysis. *Sensors*, 18(12), p.4391.

[7]. N. Kruger et al., “Deep hierarchies in primate visual cortex. What can we learn for computer vision?” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1847–1871, Aug. 2013.

[8]. G. Licciardi, P. R. Marpu, J. Chanussot, and J. A. Benediktsson, “Linear versus nonlinear PCA for the classification of hyperspectral data based on the extended morphological profiles,” IEEE Geosci. Remote Sens. Lett. vol. 9, no. 3, pp. 447–451, May 2011.

[9]. L. van der Maaten and G. Hinton, “Visualizing high-dimensional data using t-sne” 2008.

[10]. Maaten, L.V.D.; Hinton, G. Visualizing data using t-SNE. J. Mach. Learn. Res. 2008, 9, 2579–2605.

[11]. Maaten, L.V.D. Learning a parametric embedding by preserving local structure. In Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AI-STATS), Clearwater, FL, USA, 16–19 April 2009; pp. 384–391.

[12]. Hu, W., Huang, Y., Wei, L., Zhang, F. and Li, H. (2015). Deep Convolutional Neural Networks for Hyperspectral Image Classification. *Journal of Sensors*, 2015, pp.1-12.

[13]. Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. M¨uller, “Efficient backprop,” in Neural Networks: Tricks of the Trade, pp. 9–48, Springer, Berlin, Germany, 2012.