LEt-SNE: A HYBRID APPROACH TO DATA EMBEDDING AND VISUALIZATION OF HYPERSPECTRAL IMAGERY

Megh Shukla[⋆†] *Biplab Banerjee*[†‡] *Krishna Mohan Buddhiraju*[†]

[⋆]Mercedes-Benz Research and Development India Pvt. Ltd.
[†]Centre of Studies in Resources Engineering, Indian Institute of Technology Bombay

ABSTRACT

Hyperspectral Imagery (and Remote Sensing in general) captured from UAVs or satellites are highly voluminous in nature due to the large spatial extent and wavelengths captured by them. Since analyzing these images requires a huge amount of computational time and power, various dimensionality reduction techniques have been used for feature reduction. Some popular techniques among these falter when applied to Hyperspectral Imagery due to the famed curse of dimensionality. In this paper, we propose a novel approach, **LEt-SNE**, which combines graph based algorithms like t-SNE and Laplacian Eigenmaps into a model parameterized by a shallow feed forward network. We introduce a new term, *Compression Factor,* that enables our method to combat the curse of dimensionality. The proposed algorithm is suitable for manifold visualization and sample clustering with labelled or unlabelled data. We demonstrate that our method is competitive with current state-of-the-art methods on hyperspectral remote sensing datasets in public domain.

***Index Terms*—** LEt-SNE, Dimensionality Reduction, Manifold Visualization, Hyperspectral, Clustering

1. INTRODUCTION

With the increasing availability of hyperspectral imagery, researchers face a challenging task of analyzing this data. Storing and processing this vast amount of data is cumbersome and expensive which leads us to an extensively studied topic, Dimensionality Reduction. The principle behind dimensionality reduction is the utilization of statistical information within the data to come up with a condensed representation for the same. A subset of dimensionality reduction, *Manifold Learning* [1], deals with the non-linear embedding of data in lower dimensions. When dealing with a high dimensional dataset such as hyperspectral imaging, we often encounter a phenomenon commonly known as the *curse of dimensionality*; which entails that as the dimensionality d of the dataset increases, the concept of *neighbourhood* is lost. The distance between the farthest and the nearest samples is negligible when compared to the distance from a fixed sample to its nearest neighbor. This phenomenon is rigorously studied in [2, 3, 4, 5], with algorithms based on the euclidean distance suffering from the same. The focus of this paper is to create an algorithm that solves a three-fold problem: Manifold visualization, supervised clustering, and present a proof of concept for unsupervised clustering using image segmentation techniques. The core algorithm fuses a modification of t-SNE with Laplacian Eigenmaps into a model parameterized with a shallow fully connected neural network yielding quick encodings on unseen samples. To circumvent the curse of dimensionality, we introduce *Compression factor*, which creates an illusion of modifying inter-sample distance. We compare our approach with state-of-the-art techniques on three open source remote sensing datasets: *Indian Pines, Pavia University,* and *Salinas* and present the results.

Related Work: Dimensionality reduction can be subdivided into broadly two categories. Feature selection algorithms such as Genetic Algorithms [6] and Ant Colony Optimization [7] have been widely used, but do not provide information about the underlying manifold. Feature extraction algorithms such as PCA [8] and LDA do not model non-linearities in the data, whereas non-linear methods such as kernel-PCA suffer from prohibitive time complexity [9]. They also capture the global structure of data at the cost of local variations in the manifold. Another limitation of LDA is that the dimensionality of the embeddings is bounded by the number of classes present in the dataset. Graph based algorithms such as *Laplacian Eigenmaps* [10] and *Locally Linear Embedding (LLE)* [11] do not scale well with addition of new samples as they need to recompute the eigendecomposition to obtain new embeddings. *t-SNE* [12], though otherwise a beautiful visualization technique, fails to effectively deal with the curse of dimensionality. A less common variant of t-SNE is the parameteric t-SNE [13], which uses Restricted Boltzmann Machines and pretraining which leads to a complicated training procedure. Recent approaches include *UMAP* [14] which relies on projecting points along a Reimannian manifold, and Autoencoders [15, 16, 17]. *Spherical Stochastic Neighbor Encoding* [18], constrains samples onto the surface of a $\mathbb{R}^m$ unit hypersphere in a $\mathbb{R}^{m+1}$ space resulting in an ineffective use of the hyperspace. It remains to be seen if the sSNE can scale to large remote sensing datasets.

[‡]B. Banerjee was partially supported by SERB, DST (ECR/2017/000365)

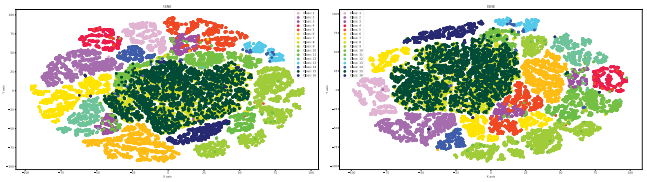


Fig. 1: t-SNE visualization of Salinas: Notice the inconsistencies in the spatial relation between classes across multiple runs.

2. METHODOLOGY

The key points considered when designing LEt-SNE include parameterization, computational time, the curse of dimensionality and the quality of embeddings produced.

Approaches using eigendecomposition as their solution need to recompute embeddings afresh when exposed to unseen samples. Parameterized models therefore have an advantage when mapping unseen samples since they model a function $Y = f(X, w)$, which takes as input X with parameters w to quickly compute the embeddings Y. A natural candidate for implementing this is a fully connected neural network architecture. The advantages presented by a neural network are multifold: they learn the task of feature extraction; stochasticity in the training process with the use of mini-batch weight updates acts as a regularizing mechanism. Mini-batches lead to faster convergence as the weights are periodically updated without waiting for an epoch to complete.

2.1. Revisiting Laplacian Eigenmaps and t-SNE

Laplacian Eigenmaps is a graph based method with an emphasis on preserving the local structure of the manifold and encouraging the discovery of natural clusters in the dataset. Let y_i be the i^{th} sample from embedding Y, then the minimization of objective $J(y)$ (Eq: 1) ensures that neighbouring samples ($\mathcal{A}_{ij} = 1$) in the graph have encodings similar to each other. The choice of neighbours for a given sample are done by picking the top *k* samples having the lowest euclidean distance to the fixed sample. The traditional solution involves constraint optimization which results in the eigendecomposition of the graph laplacian ($\mathcal{L}$).

$$J(y) = \sum_{i,j}(y_i - y_j)^2 \mathcal{A}_{ij} = 2Y^T \mathcal{L} Y \tag{1}$$

t-SNE is another popular algorithm for manifold visualization,which gives euclidean distances between samples a probabilistic interpretation. Let x_i be the i^{th} sample of X in $\mathbb{R}^n$, then the probability of x_j being a neighbour of x_i is given in Eq: 2. The variance σ_i^2 is a loose interpretation of the density of samples around x_i. A similar *t-distribution* $q_{j|i}$ is computed for $y \in \mathbb{R}^m (m < n)$, followed by minimizing the KL divergence between p_{ij} and q_{ij}. The resultant y obtained are representative of the graph structure in x.

$$p_{j|i} = \frac{exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-||x_i - x_k||^2/2\sigma_i^2)} \tag{2}$$

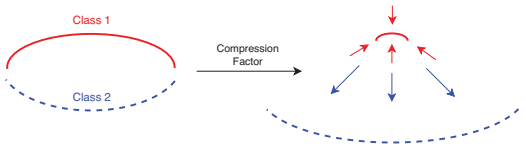


Fig. 2: Compression acting on Class 1 samples. In practice, compression dominates as outward expansion from all the samples cancel out.

2.2. LEt-SNE

Our preliminary attempts at a parameterized algorithm began with Laplacian Eigenmaps. As before, Let $Y = f(X, w)$ then the objective (Eq: 1) reduces to minimizing $\nabla_w Y^T L Y$ using gradient descent. In its current form, reducing $||w||$ will minimize the objective without the network learning anything meaningful. We corroborate this with experiments, which confirm that the network 'cheats' by either suppressing the L2-norm of the weights or collapsing all samples into the same region. Optimizing the network with respect to the constraints $Y^T \mathcal{D} Y = I$ [10] ($\mathcal{D}$ is the Degree matrix) and $||w||_2 = k$ failed as it prevented the loss from converging to lower values. A key takeaway from this experiment was Laplacian Eigenmaps' ability to form tight cluster of embeddings for neighbouring samples in X. The question arises, can we devise a method on how to keep dissimilar points apart?

We turn our attention to SNE and see how it alleviates this problem. If all samples X are collapsed into very similar encodings Y, then the euclidean distance between an encoding Y and all other points will not vary significantly. Thus, using Eq: 2, $q_{j|i} \approx 1/|Y| \ \forall j$ i.e, the neighbourhood distribution for encodings Y will approximate a uniform distribution. The objective of SNE hence imposes a large penalty on the collapse of embeddings into a small region. However, by introducing t-SNE in our solution, we also need to address the curse of dimensionality.

Recall the curse of dimensionality; as the dimensionality of our data increases, the variation in the inter-sample distance decreases. A similar scenario as the one discussed previously arises, this time among the samples in X. Therefore, $p_{j|i} \approx 1/|X| \ \forall j$, which leads to t-SNE giving inaccurate visualizations[2]. The effect of the curse of dimensionality leads to ambiguity in the relation between classes as shown in Fig: 1. This leads us to the next question, could we stretch inter-sample distances to beat the curse of dimensionality?

Our solution lies in defining a new term, *Compression Factor*. The Compression Factor (CF) uses the Adjacency matrix ($\mathcal{A}$) generated by Laplacian Eigenmaps to give an illusion of manipulating the distance between samples in $\mathcal{X}$. We scale up the values of $p_{j|i}$ if x_i and x_j are connected in $\mathcal{A}$. This compression is mathematically defined in Eq: 3. As

[2]This holds true in a simplified scenario so as to provide intuition as to why t-SNE does not perform well in high-dimensional applications

a consequence we find that if $CF > 1$; $\forall j \in neighbour\,(i)$, $\tilde{p}_{j|i} \uparrow$, whereas $\forall k \notin neighbour\,(i)$, $\tilde{p}_{k|i} \downarrow$. This approach helps in limiting the effect of curse of dimensionality, by creating the illusion that the difference in sample distances is larger than they appear, as shown in Fig: 2.

$$\tilde{p}_{j|i} = \frac{p_{j|i} * \{(CF-1) * \mathcal{A}_{ij} + 1\}}{\sum_j p_{j|i} * \{(CF-1) * \mathcal{A}_{ij} + 1\}} \tag{3}$$

We also modify t-SNE to retain the conditional probabilities as proposed in SNE instead of the joint probabilities due to the strong gradients obtained in comparison to the latter[12]. Although we do not fix any particular network architecture, we recommend the use of Batch Normalization[19] for rapid convergence as well as adapting to the scale specified. In LEt-SNE, perplexity plays the role of determining the scale of our embeddings instead of translating to the number of neighbors as in t-SNE. In the next subsections, we describe the three modes of operation of the algorithm.

2.2.1. LEt-SNE for Manifold Visualization

The algorithm for manifold visualization is straightforward and is shown in Eq: 4, where $p_{i|j}$, $q_{i|j}$ are computed using Eq:{ 2, 3}. The Adjacency matrix A is computed using the *top-k* nearest neighbours. For this task, it is more suitable to keep a low value for the number of neighbours hyperparameter, as well as a low value of compression factor (~5), to prevent the probability values from saturating and retain some structure from the original encodings.

$$w^* = arg\min_w \mathbb{E}_x \left(\mathcal{Y}^T \mathcal{L} \mathcal{Y} + \lambda \sum_{i,j} \tilde{p}_{i|j} log \frac{\tilde{p}_{i|j}}{q_{i|j}} \right) \tag{4}$$

2.2.2. LEt-SNE for Labelled Clustering

Instead of computing the adjacency for X using the *top-k* neighbours approach, we directly use class labels. The new adjacency matrix is defined as:

$$A_{ij} = \begin{cases} 1 : \text{class}_\text{i} == \text{class}_\text{j} \\ 0 : otherwise \end{cases}$$

A high compression factor (> 50) acts upon the new adjacency matrix, saturating the probabilities and creating an illusion of tight clustering between intra-class samples and large separation between inter-class samples. In case the samples of a class come from a multimodal distribution, we can divide the samples into subclasses each of which captures a single mode of the distribution. The multimodality of a class can be observed in the manifold visualization technique described earlier. To ensure that the embeddings adequately represent this illusion, we compute $KL(q||p)$ instead of $KL(p||q)$. With this change, if $p_{j|i}$ is small (as is the case with inter-class separation), the corresponding $q_{j|i}$ too has to be a small value to prevent incurring a large loss. An explanation can be found in [20]. The objective for minimization is:

$$w^* = arg\min_w \mathbb{E}_x \left(Y^T \mathcal{L} Y + \lambda \sum_{(i,j)} q_{i|j} log \frac{q_{i|j}}{\tilde{p}_{i|j}} \right) \tag{5}$$

Note that we do not use classification gradients to allow the network to explore spatial relations within the dataset.

2.2.3. LEt-SNE for Unlabelled Clustering

So far, we have seen two methods to compute the Adjacency matrix: *top-k* and class labels. Are there any alternative approaches to design the Adjacency matrix such that Eq: 5 can be used for unlabelled data?

Let us assume a pixel in an image (I) to belong to a particular class, then it is highly likely that its 8-neighbours belong to the same class too. We then partition the image into disjoint regions, with pixels (samples) within a region considered connected components when computing the adjacency matrix. Formally, let I be the Image composed of our samples X, segmented into regions R such that $I = R_0 \cup R_1 \ldots R_n$ and $R_i \cap R_n = \varnothing\ \forall i, j \in n; i \neq j$. The adjacency matrix is defined as:

$$A_{ij} = \begin{cases} 1 : \text{R}_{x_\text{i}} == \text{R}_{x_\text{j}} \\ 0 : otherwise \end{cases}$$

As a proof of concept, we use two segmentation algorithms: Watershed [21] and SLIC [22]. We prevent oversegmentation in SLIC by employing Region Adjacency Graph and Graph Cut algorithm.

3. EXPERIMENTATION AND RESULTS

To keep the paper concise, we select a few experiments from each of the three modes of operation and produce them here. The code and the supporting material, including all experiments and class confusion maps[23] are available at GitHub: meghshukla/LEt-SNE.

We use three datasets popular among the remote sensing community to verify our results: Indian Pines, Salinas and Pavia University. Indian Pines and Salinas features 16 classes each, with the former having considerably more overlap in classes than the latter. The Pavia University dataset contains 103 hyperspectral bands with 9 classes present. Further details on the datasets can be found in [23]. The data preprocessing step is limited to standardization with zero mean and unit variance. Our implementation is primarily based on TensorFlow [24]. We use monte-carlo approximations of Eq:{ 4, 5} over mini-batches m for optimizing the weights w.

Manifold Visualization: The two dimensional embeddings of Indian Pines are shown in Fig: 4. We use visual inspection to analyze the quality of embedding as done in [12]. We note that all approaches show similar characteristics, such as elongated strips of *Classes: 10-12* on one side and *Class 14*. UMAP visualization clusters the classes together, but lacks the fine structure as shown in LEt-SNE. On the other hand, LEt-SNE captures the local structure as well as global structure to a large extent.

Clustering with labels: We evaluate the separation of classes and quality of clustering by training and evaluating the embeddings using a SVM classifier. The evaluation results for Pavia University and Salinas dataset are shown in Fig: 3.

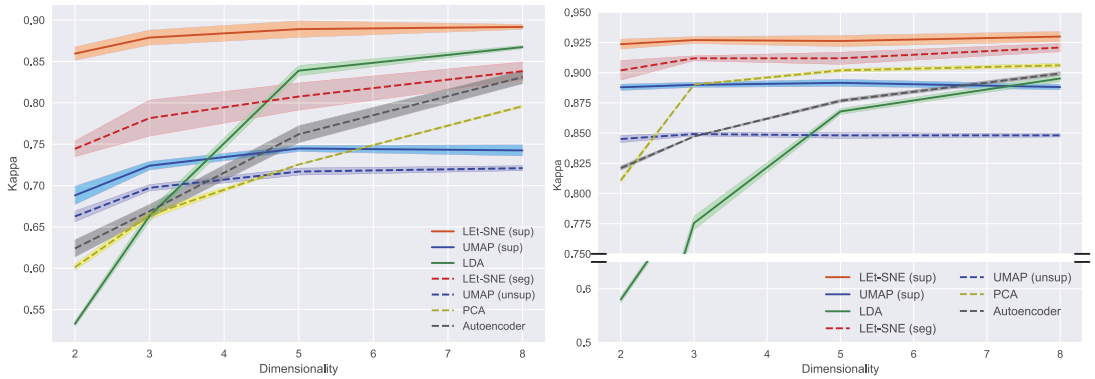


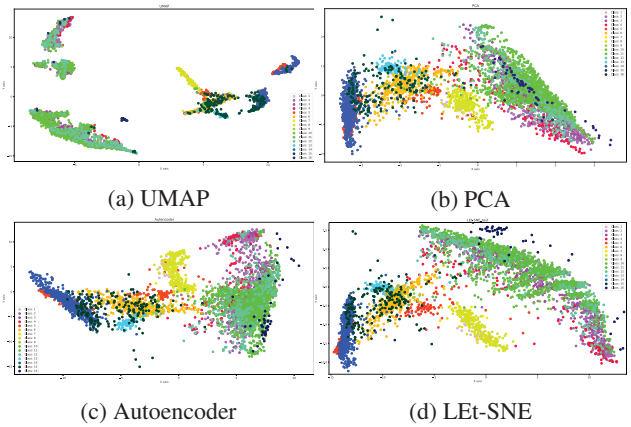
Fig. 3: Pavia (left) and Salinas (right): Comparing various supervised and unsupervised approaches

We note that LEt-SNE (sup) outperforms all approaches, indicating better separation of samples using class labels. An intuition behind the results can be obtained by visualizing the embeddings shown in Fig: 5. We see that LEt-SNE provides better clustering and discriminative power between classes in comparison to UMAP, which is also verified in the confusion map. The importance of *Compression Factor* is apparent from Table: 1, where we note a significant improvement in performance of the algorithm.

Clustering without labels: Depending on the choice of algorithm, a 3-channel (for SLIC) or 1-channel (for Watershed) image is provided as input for segmentation. The first principal component obtained from transforming the original channels using PCA, or the grayscale of the False Color Composite (FCC) image could be used as the 1-channel input. Similarly, the first three principal components or the FCC could be used as a 3-channel input based on which segmentation is performed. For Salinas, we use the grayscale image with Watershed algorithm, whereas for Pavia and Indian Pines we use the Princpal Components and SLIC for segmentation. The region segmentation and embeddings for Salinas dataset in shown in Fig: 6. Refer to Fig: 3, where we notice that even in the absence of labels, the segmentation based adjacency matrix used by LEt-SNE (seg) provides vital information which is used by compression factor to provide meaningful embeddings.

4. CONCLUSION

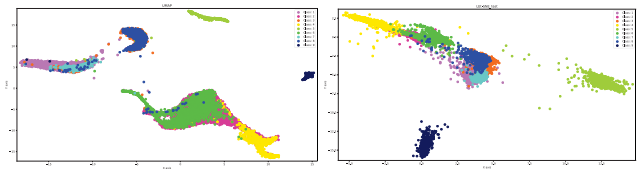
In this work, we have attempted to solve the problem of dimensionality reduction by proposing a new method, LEt-SNE. We have focused on parameterization, computational time, curse of dimensionality and producing intuitive embeddings when designing the algorithm. We have successfully demonstrated the use of Compression Factor to help alleviate the curse of dimensionality. With LEt-SNE, we solve a threefold problem: Manifold Visualization, Clustering with Labels and Clustering without labels; thereby extending on the use cases of t-SNE. Our results show that LEt-SNE is competitive with popular state-of-the-art algorithms on common remote sensing datasets.



(a) UMAP (b) PCA

(c) Autoencoder (d) LEt-SNE

Fig. 4: Indian Pines: Manifold Visualization



(a) UMAP (b) LEt-SNE

Fig. 5: Pavia: Clustering with labels

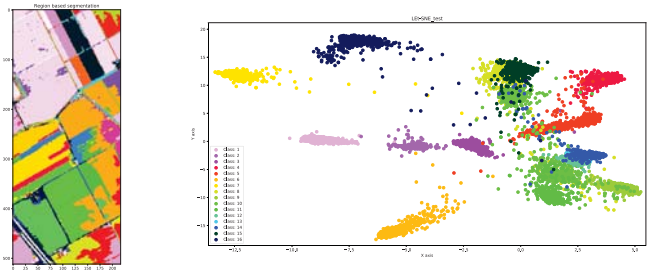


Fig. 6: Salinas: (Left) Color coded disjoint regions (Right) LEt-SNE embeddings

Table 1: Accuracy and Compression Factor: LEt-SNE (sup) with Dimensions = 2

Compression	Indian Pines	Salinas	Pavia
NA	0.4936	0.7877	0.7534
200	**0.6207**	**0.9236**	**0.8594**

5. REFERENCES

[1] Lawrence Cayton, "Algorithms for manifold learning," *Univ. of California at San Diego Tech. Rep*, vol. 12, no. 1-17, pp. 1, 2005.

[2] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim, "On the surprising behavior of distance metrics in high dimensional space," in *International conference on database theory*. Springer, 2001, pp. 420–434.

[3] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft, "When is "nearest neighbor" meaningful?," in *Database Theory — ICDT'99*. 1999, pp. 217–235, Springer Berlin Heidelberg.

[4] Damien François, Vincent Wertz, and Michel Verleysen, "The concentration of fractional distances," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 19, pp. 873–886, 08 2007.

[5] Alexander Hinneburg, Charu C. Aggarwal, and Daniel A. Keim, "What is the nearest neighbor in high dimensional spaces?," in *Proceedings of the 26th International Conference on Very Large Data Bases*, 2000, pp. 506–515.

[6] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain, "Dimensionality reduction using genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 2, pp. 164–171, July 2000.

[7] S. Sharma, K. M. Buddhiraju, and B. Banerjee, "An ant colony optimization based inter domain cluster mapping for domain adaptation in remote sensing," in *2014 IEEE Geoscience and Remote Sensing Symposium*, July 2014, pp. 2158–2161.

[8] L.J. Cao, K.S. Chua, W.K. Chong, H.P. Lee, and Q.M. Gu, "A comparison of pca, kpca and ica for dimensionality reduction in support vector machine," *Neurocomputing*, vol. 55, no. 1, pp. 321 – 336, 2003, Support Vector Machines.

[9] Lijun Zhang, Tianbao Yang, Jinfeng Yi, Rong Jin, and Zhi-Hua Zhou, "Stochastic optimization for kernel pca," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. 2016, AAAI'16, pp. 2316–2322, AAAI Press.

[10] Mikhail Belkin and Partha Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[11] Sam T. Roweis and Lawrence K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[12] Laurens van der Maaten and Geoffrey Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[13] Laurens van der Maaten, "Learning a parametric embedding by preserving local structure," in *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, 16–18 Apr 2009, vol. 5 of *Proceedings of Machine Learning Research*, pp. 384–391.

[14] Leland McInnes and John Healy, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv*, 2018.

[15] S. P. Luttrell, "Hierarchical self-organising networks," in *1989 First IEE International Conference on Artificial Neural Networks, (Conf. Publ. No. 313)*, Oct 1989, pp. 2–6.

[16] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[17] Vincent et. al Pascal, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[18] D. Lunga and O. Ersoy, "Spherical stochastic neighbor embedding of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 2, pp. 857–871, Feb 2013.

[19] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[20] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016, http://www.deeplearningbook.org.

[21] Serge Beucher, "Watershed, hierarchical segmentation and waterfall algorithm," in *Mathematical morphology and its applications to image processing*, pp. 69–76. Springer, 1994.

[22] Radhakrishna et al. Achanta, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[23] Megh Shukla, "LEt-SNE: A hybrid approach to data embedding and visualization of hyperspectral bands in satellite imagery," M.Tech. Thesis, CSRE, IIT Bombay, 2019.

[24] Martín Abadi et.al, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.