# Deep Self-Organizing Maps for Visual Data Mining

Chathurika S. Wickramasinghe, Kasun Amarasinghe, Daniel Marino, Milos Manic
Virginia Commonwealth University, Richmond, Virginia
brahmanacsw@vcu.edu, amarasinghek@vcu.edu, marinodl@vcu.edu, misko@ieee.org

*Abstract*— **Visual data mining facilitates the involvement of domain experts in the data mining processes. The effectiveness of visual data mining is especially dominant when paired with unsupervised methods due to the abundance of unlabeled data. Deep Self-Organizing Maps (DSOMs) are unsupervised learning architectures capable of high level feature abstraction. In this paper, we analyze the effectiveness of using DSOMs for visual data mining. DSOM's visual data mining capability was evaluated using the following visual data explorations methodologies: 1) U-Matrix, 2) hit maps and 3) data histograms. In comparison with traditional single layered SOM architectures, experimental results showed that DSOMs produced more accurate visual representations of the underlying data distributions. Therefore, DSOM is a viable method for generating easily understandable visual representations of high-dimensional complex datasets. These visual representations can be powerful tools in the real world, leading to better understanding of systems and thus enabling the design of better algorithms for control and monitoring.**

***Keywords— Deep Learning, Self-Organizing Map (SOM); Deep Self Organizing Map (DSOM); MNIST; Unsupervised classification; Visual Data Mining***

## I. INTRODUCTION

Data mining methodologies have become almost indispensable with the increase of amount and complexity in data in almost every domain [1], [2]. Data mining is an interactive process which requires intuition and human knowledge coupled with modern machine learning techniques [3]. Visual data mining (VDM) is the process of exploration, interaction, and reasoning with abstract data in human perceivable way [2]. Thus, it allows humans to incorporate human intelligence in the data mining process, and it has been shown that human involvement increase the effectiveness of data mining processes [2].

Pattern recognition is an important aspect of data mining in a multitude of areas [4]–[11]. Supervised pattern recognition (SPR) approaches albeit high performance, require labeled data[9], [12]. Unlabeled data is abundant in real world applications and obtaining sufficient amount of labeled data can be costly [3],[12],[13]. Therefore, SPR poses challenges in cases where there is a lack of available labeled data. Due to that unsupervised and semi-supervised learning based pattern recognition methods are becoming common [10], [11], [13]. In these methods, they use expertise of domain experts in order to improve the effectiveness of data mining and to validate the learning processes.

Self-Organizing Maps (SOMs) are unsupervised learning algorithms which have the capability of mapping a high-dimensional data distribution onto a low-dimensional grid while preserving the most important topological and metric relationships of the input data [14], [15],[16]. These mappings can be used to visualize these high dimensional data while preserving their topological structure. Since SOMs have the capability of adjusting the network for representing the topological properties of input data, it has been widely used in visual data mining as a dimensionality reduction and feature extraction tool [2], [17], [18]. SOM has the ability of visualization of multi-dimensional data in a human perceivable way [14], [15],[16]. SOMs have better capability of revealing the overlapping structure in clusters compared to other traditional cluster analysis techniques [14] [15]. SOMs have successfully used in many areas including speech recognition, robotics, process control and telecommunication [6]–[9], [5].

Deep Self-Organizing Maps were proposed in literature to add the high level feature abstraction capability to single layered SOMs. [19] [20]. In order to overcome this limitation, Liu et al [20] proposed the Deep SOM (DSOM), an architecture composed of multiple layers, similar to a Deep Neural Network (DNN), . Their work suggested adding multiple sequential layers of SOMs and "sampling" layers. The sampling layer combined multiple SOMs form the preceding layer into a single map. Even though DSOM solves the high-level feature abstraction limitation, DSOM architecture proposed by Liu et al. is computationally expensive [21]. In our previous work, we proposed a parallelizable DSOM architecture to alleviate the performance bottle neck while preserving the high level feature abstraction capabilities [21]. In [21], we showed that growing the network in width by adding parallel SOM layers feeding into the same sampling layer, as opposed to growing in depth improved computational time while retaining classification accuracy and feature abstraction capability. In this paper, by the acronym "DSOM" we refer to the parallelizable DSOM architecture we presented in [21].

In this paper, we evaluate the effectiveness of DSOM for visual data mining capabilities. The analysis is carried out using widely used visual data exploration methods such as U-matrix, hit maps and data histograms. These visualizations are qualitatively compared to the same visualizations generated by single layer SOM (referred to as SOM hereafter).

The rest of the paper is organized as follows. Section II provides brief overview of the SOM learning algorithm; Section III overviews the DSOM architecture we proposed in [21] and DSOM based visual data mining; Section IV presents and discusses the experimental results, and finally, section V concludes the paper.

## II. UNSUPERVISED LEARNING ALGORITHM OF SELF ORGANIZING MAPS

This section briefly reviews the traditional SOM architecture by T. Kohenen [22].The Self Organizing Map

978-1-5386-5024-0/18/$31.00 ©2018 IEEE

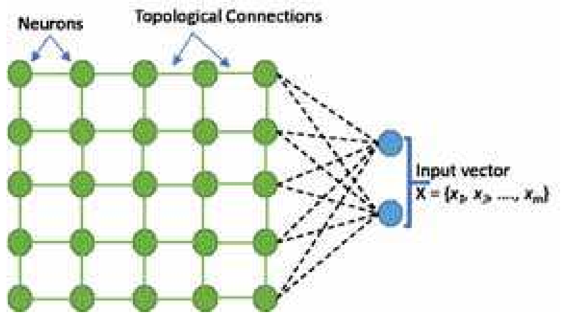


Figure 1: 2D Self Organizing Map Architecture

algorithm was introduced in 1981 by T. Kohenen [22]. It uses winner-take-all competitive leaning method.

A typical SOM network consists of a topological grid of neurons which are arranged in 1D or 2D lattice. Each neuron maintains a weight vector $W = \{w_1, w_2, \ldots., w_m\}$ of $m$ dimension, where $m$ is the dimension of input feature vector. Input pattern can be represented as $X = \{x_1, x_2, \ldots., x_m\}$. Figure 1 illustrates the structure of 2D SOM architecture. The learning process of the SOM network can be described as follows,

**Step 1:** Randomly initialize all the weight vectors of the SOM network.

**Step 2:** Randomly select an input pattern X from the training data set.

**Step 3:** Find the Best Matching Unit (BMU) for the selected input X by calculating the Euclidian distance between X and weight vectors of the neurons in the neuron lattice. BMU can be calculated as,

$$BMU = \arg\min_j \| x - w \|^2, \; j = 1,2,\ldots,N \tag{1}$$

Here, N is the number of neurons in the SOM.

**Step 4:** The weights in the neighborhood neurons ($j$) of BMU neuron (winning neuron $j^*$) can be calculated as follows,

$$w_j(t+1) = w_j(t) + \eta(t)\alpha(t, j, j^*)\left(x - w_j(t)\right) \tag{2}$$

$$j \in N_{j^*}$$

where, $N_{j^*}$ defines the neighbourhood region of $j^*$, and $\eta(t)$ is the learning rate at epoch $t$. The learning rate is decayed through the epochs as follows:

$$\eta(\mathrm{t}) = 0.49\left(1 - \frac{\mathrm{t}}{\mathrm{epoch}}\right) + 0.01 \tag{3}$$

where, 0.49 is a constant which found experimentally [20]. Neighbourhood calculation performs using the following equation,

$$\alpha(\mathrm{t}, \mathrm{j}, \mathrm{j}^*) = \exp\left(-\frac{\| x - w_j \|}{2\delta_\mathrm{t}^2}\right) \tag{4}$$

Where $\delta_t$ corresponds to the radius of the neighbourhood. It has to be noted that the neighborhood is decayed along with the epochs.

**Step 5:** Repeat step 2-4 until a specified convergence criteria has been reached.

Classification on the last SOM layer can be performed using the sample hits of each neuron. Proposed unsupervised classifier implementation is discussed later in the paper.

## III. Deep Self Organizing Map Based Visual Data Mining

This section discusses DSOM architecture and the DSOM based visual data exploration techniques for visual data mining.

As mentioned, DSOMs have proven to be capable of high level feature abstraction. Therefore, we hypothesize that the output from DSOMs will provide better visualizations that reflect the relationships that exist in input data when compared to the SOM. These visualizations can help domain experts/users extract knowledge from the data such as available patterns, input correlations and pattern distributions. This can lead to better understanding of complex systems and thus result in better data-driven and expert knowledge driven prediction, monitoring and control systems.

### *A. Deep Self Organizing Maps*

The main idea behind the DSOM merges the concepts of unsupervised learning from SOMs and high level feature abstraction Convolution Neural Networks (CNNs). In CNNs, each unit in a layer receives inputs from a set of units located in a small neighborhood of its preceding layer [23], [24]. CNNs possess the idea of local receptive fields which are capable of extracting basic features such as edges, endpoints and corners. These extracted features are combined in the subsequent layers to obtain higher level features. This layers structure is incorporated in the DSOMs such that SOMs in higher-level layers are able to learn more abstract information than the SOM layer in its preceding layer. DSOMs are primarily used for image data.

In DSOMs, input patterns (images) are divided into small patches and each patch is processed using a separate SOM, i.e. in a *SOM layer* each patch is processed using separate SOMs at parallel. The wining neuron indexes of all SOMs that processed the patches are then organized into a single 2D grid in the next layer (sampling layer). In the parallelizable DSOM presented in [21], more than one SOM layer in used parallel (see Figure II). These parallel layers can have different map sizes accomplishing two major goals: 1) improve the computational efficiency of DSOMs and 2) improve the generalization capability. Computational efficiency is increased by increasing the number of operations that can be carried out in parallel to improve computational efficiency. Generalization capability is increased by supporting learning features of different resolution though different map sizes. This process of SOM and sampling layers are repeated until the last layer. The final layer is a single SOM which takes the preceding sampling layer as the input.

Use of multiple maps in parallel results in a lesser number of serial SOMs in PD-SOM architecture. Computations of

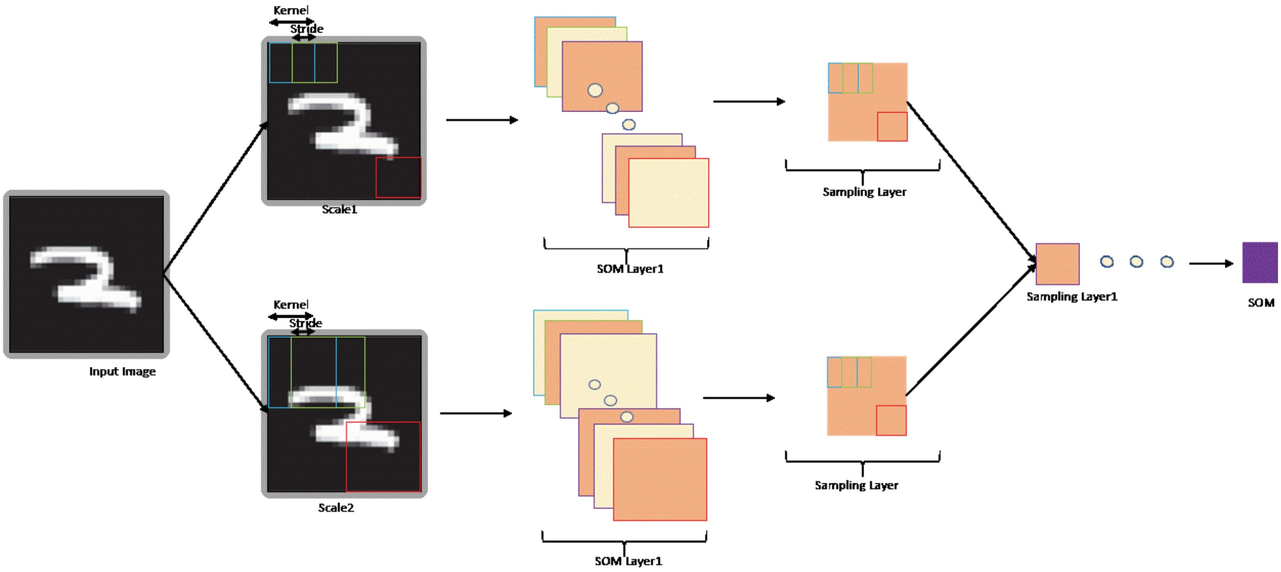


Figure 2: Deep Self Organizing Map

parallel SOM layers are performed in parallel, and thus results in less computation time compared to the initial DSOM in [20]. Further, experimental results showed that the parallelizable DSOM architecture showed a considerable improvement in robustness to noisy data. The parallelizable DSOM architecture can be extended by adding more parallel layers to make it wider and by adding more SOM layers and sampling layers to make it deeper. In the interest of brevity, the complete training algorithm of DSOM is not presented in this paper. If interested, readers are directed to [21] for the detailed training process.

DSOM was implemented as an unsupervised classifier. Unsupervised classifiers (clustering based classifiers) don't require labeled data initially [11], [25].They require a set of training datapoints which can be clustered by a unsupervised learning based model. Once the clusters are generated, the domain expert can interpret these clusters using different knowledge discovery and data visualization methods and can assign a class for each cluster.

### *B. DSOM based Visual Data Mining*

As mentioned before, data visualization and visual data exploration play important roles in knowledge discovery [18]. Domain experts and analysts need tools for generation of hypotheses about models and datasets. Therefore, VDM plays a major role in knowledge discover by providing interactive data presentations and various visual displays for domain experts [18].

SOMs are widely used in visual data mining as dimensionality reduction and feature extraction tools due to their capability of mapping high dimentional data into a low dimention space, i.e. They have the data projecton capability allowing visual inspection [2], [26]. Once data is mapped to low dimensional spaces, human experts can explore and interect with data which allows to incooparate human knowledge into data mining process.

There are number of methods which have been proposed in the literature to explore information based on SOMs. Most commonly used methods are U-matrix, P-matrix, clustering of model vectors, projecting model vectors into low dimensional spaces, hitmaps and data histograms [27] [26]. As an initial step, this paper focuses on three methods: U-matrix, hit map and data histograms. The visualization techniques are presented below.

**U-matrix**: (Unified Distance Matrix) is one of the most widely used methods for visualizing the cluster structure of SOMs [28], [29]. It shows the distance between weight vectors of neighboring neurons (immediate neighbors) using color codes [30]. If distances between neighboring units are small, then they represents a cluster pattern with similar characteristics. If neighboring units are far apart, then these units are located on low dense input space with few patterns. They can be considered as separation between clusters.

**HitMaps:** This shows how often a neuron is chosen as the BMU. Hit map information can be utilized in clustering the SOM by using zero-hit units to indicate cluster boarders [28] .

**Data Histograms**: These represent how many data items are represented by a specific unit. This is also a slightly different representation of hitmap representation.

## IV. Experimental Results and Discussion

This section discusses the experiments performed in the paper. First, the unsupervised classifier implementation is discussed. Then the experimental setup used for different test cases is presented. After that, the experiments carried out to test different architectures are presented. Finally, comparison of two model based on different visual data mining methods are presented and their importance for future data mining is discussed.

TABLE I: RESULTS OBTAINED FOR SOM ALGORITHM

| Model | Layer 1 MapSize | Train Accuracy | Test Accuracy | Test Accuracy for Different Noise Levels [%] | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 2 | 5 | 10 | 20 | 40 | 50 | 60 |
| 1 | 8*8 | 61.98 | 61.864 | 61.908 | 62.03 | 62.02 | 62.066 | 58.348 | 52.596 | 25.742 |
| 2 | 12*12 | 63.16 | 62.764 | 62.866 | 62.806 | 62.89 | 62.648 | 58.888 | 53.136 | 27.236 |
| 3 | 16*16 | 65.59 | 64.638 | 64.782 | 64.738 | 64.654 | 64.398 | 59.548 | 52.64 | 26.416 |
| 4 | 20*20 | 66.76 | 66.338 | 66.354 | 66.314 | 66.414 | 65.866 | 61.098 | 54.762 | **28.072** |
| 5 | 24*24 | **69.36** | **68.57** | **68.638** | **68.576** | **68.712** | **68.130** | **63.136** | **55.574** | 26.072 |

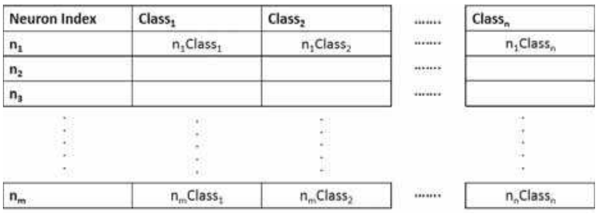


Figure 3: Generated hit map representation

### A. Implementation of Unsupervised Classifier

In this paper, we implemented an unsupervised classifier using SOMs. In order to analyse the clustering capability we used a labeled dataset in this study. However, it has to be noted that the labels were not used in the training. After training was completed, a special type of neuron hitmap was generated using the idea of conventional neuron hitmap. This map was generated based on the number of times a specific neuron become BMU for a specific class. Figure 3 shows the special hit map that we generated. Each cell ($n_m$,$Class_n$) represent number of times the neuron 'm' became BMU for the class 'n'. This type of a hitmap was generated as opposed to conventional hitmap because it provides the user with a better understanding of the class distributions in the data.

By looking at the special hit map and the U-matrix, a domain expert can assign a class for each neuron/cluster. For this experiment, since we had the labels available, we assigned a class for each neuron based on the maximum number of times a specific unit became the BMU for a specific class. When labels are available and when there are ties between classes, a domain expert will get involved to resolve the rites. In such cases, domain expert will assign classes to units based on the classes of neighboring neurons, by looking at the U-matrix. This method reduces the human workload for designing the classifier and at the same time, allows integrating human expertise in the classification process.

After each neuron is assigned a class label, the train accuracy and test accuracy are calculated as performance measures. The robustness of the models were measured by introducing different salt and pepper noise levels to the MNIST dataset. Salt and pepper noise removal is one of the major topics in the field of image enhancement [31]. It is a type of impulse noise that occur in images. Further, it has been shown that such noise has a substantial effect on the performance of image processing applications in various industrial tasks such as face recognition [32].

TABLE II: THE ARCHITECTURE OF THE DSOM [20]

| Layer | Nmap | MapSize | Patch(k) | Stride |
|---|---|---|---|---|
| Layer 1 | 100 | - | - | 2 |
| Layer 2 | 1 | 8*8 | 5*5 | 1 |

### B. Experimental Setup

MNIST benchmark handwritten character recognition data set is used for the training and testing of two models. It contains images of hand written characters (0 to 9 digits) of 28X28 pixels in size. The ratio of training to test data set was used as 3:10. For this experiment, a significantly smaller training set of 3000 images were used to reduce the classifier training time. The complete test set of 10000 images were used to test the accuracy of algorithms. Building an efficient classifier using a small training data will be advantageous in cases where there is only limited amount of training data to traing a supervised classifier and to for implementing classifiers which are time and cost efficient. The training data set was selected randomly while maintaining the balance class lables.

### C. Classification Accuracy: SOM

As mentioned before, the map size was selected within 8-24 range which is the maximum and minimum map size used for DSOM models. So, it allowed to perform a fair comparison between DSOM and SOM models. Table 1 presents the results obtained for SOM architecture.

It was noticed that the increase the map size result in an increase of train accuracies. For the noise range 2-20, there were no much differences in test accuracy. There was a decrement in accuracy with the increase of noise for all the models. For higher noise levels (40-60) the decrement was drastic compared to lower noise levels. The highlighted result in the table was the maximum result obtained for SOM, out of all models under each accuracy criteria.

### D. Classification Accuracy: DSOM

As discussed previously the parameters for the DSOM was selected based on our previously presented DSOM [21]. Table II presents the architecture details of DSOM. Map sizes were selected experimentally and the range was limited to 18x18 to 24x24. It has to be noted that the map sizes were changed only in the first layer of the architecture. Last layer map size kept as constant (8*8) where unsupervised classification is performed. Table III presents the results obtained for DSOM architecture. Only the result of best for models within selected range was presented with relevant map sizes in parallel SOM layers.

TABLE III: RESULTS OBTAINED FOR DSOM ALGORITHM

| Model | Layer1 | | | | Train Accuracy | Test Accuracy | Test Accuracy for Different Noise Levels [%] | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Patch Scale1 | Patch Scale2 | Map Size 1 | Map Size 2 | | | 2 | 5 | 10 | 20 | 40 | 50 | 60 |
| 1 | 10 | 10 | 22*22 | 18*18 | 85.12 | 83.744 | 83.804 | 83.818 | 83.632 | 83.196 | 75.504 | 64.05 | **21.718** |
| 2 | 10 | 10 | 22*22 | 16*16 | 85.68 | 83.82 | 83.68 | 83.772 | 83.656 | 82.92 | 75.028 | 61.712 | 20.724 |
| 3 | 10 | 10 | 22*22 | 14*14 | **86.64** | **84.868** | **84.788** | **84.838** | **84.608** | **84.044** | **75.846** | **63.408** | 21.706 |
| 4 | 10 | 10 | 22*22 | 24*24 | 85.72 | 83.802 | 83.872 | 83.79 | 83.624 | 83.336 | 75.616 | 63.224 | 20.612 |

It can be observed that all DSOM models showed better train and test accuracies compared to SOM architecture. The different between train to test accuracy was around 2%. Similar to SOM architecture, there were no significant difference in test accuracies for the lower noise levels (between 2 to 20). Further, for higher noise levels, there was a huge accuracy decrement.

The test accuracy difference between the best SOM model and the best DSOM model was more than 18%, which was a significant difference. This difference accounted for the better performance of DSOM architecture compared SOM.

*E. Visual Data Mining*

As discussed earlier, one major characteristics of SOMs is that they can be easily interpreted via various visualization techniques. This means that SOMs are very suitable for visual data mining. As mentioned, this paper uses the U-Matrix, hitmaps and data histograms for evaluating the VDM capability of the DSOMs. Since the classifications of all the DSOM models were performed on a 2D neuron map of size 8X8 (last SOM layer), the SOM model with 8X8 was selected for the comparison.

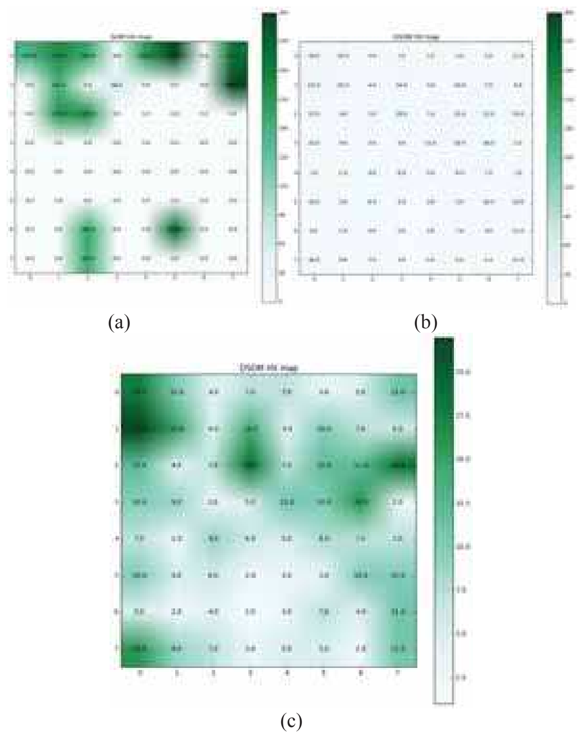

Figure 4: (a) Hit map Representations for SOM, (b) DSOM – scale representation (c) DSOM- Unscaled Representation

**Hit map:** Hit map representation which shows how often a unit is chosen as a BMU. Figure 4 (a) and (b) represents the hit map observed for SOM and DSOM, respectively. Hit maps SOM and DSOM were mapped to the same scale for comparison purposes. It was observed that only a few units of the SOM were activated and most units showed 0 hit value, whereas in DSOM, all units showed a hit value greater than 0 (all units were active). When comparing the neuron hits, active SOM units showed very high neuron hits compared to DSOM. In Fig 4(b), it appears as if the DSOM neurons don not show a difference in operation. However, In Figure 4 (c), which represents an unscaled hit map of DSOM, it can be seen that some units show higher activity compared to the other units.

As mentioned, Fig 3 presents the special hit map which we used for implementation of the unsupervised classifier. Using that we generated a new hit map representation where each unit represents the class label which it activated as BMU at the highest frequency (See Figure 5). It was observed that (8*8) SOM model doesn't show proper clusters whereas DSOM hit map shows better clusters, where neighboring units act as one

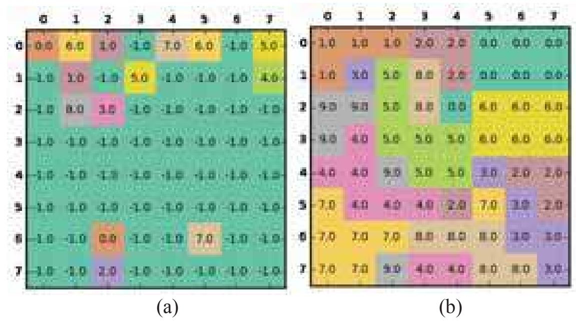

Figure 5: Class Label distribution (a) SOM, (b) DSOM

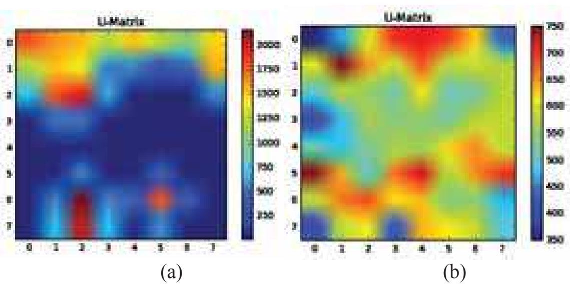


Figure 6: U-matrix representation (a) SOM, (b) DSOM

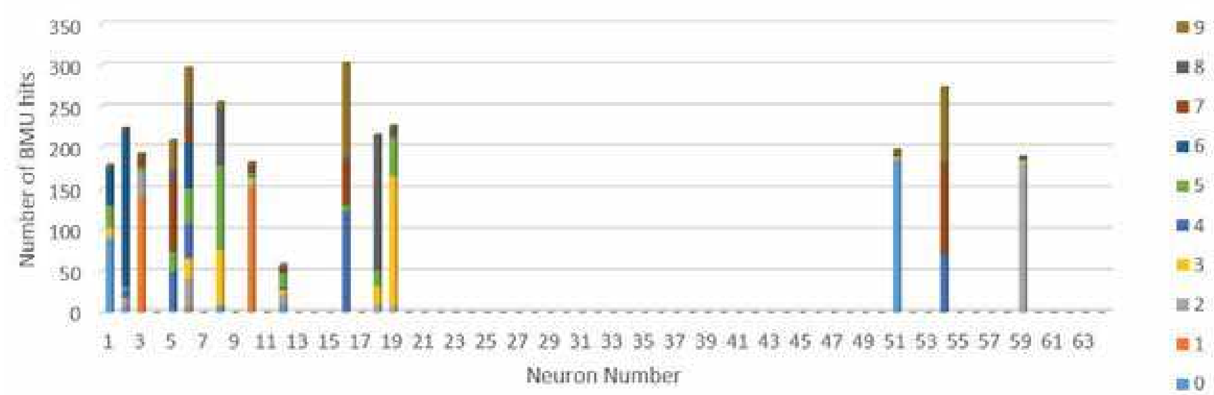


Figure 7: Data Histogram for SOM

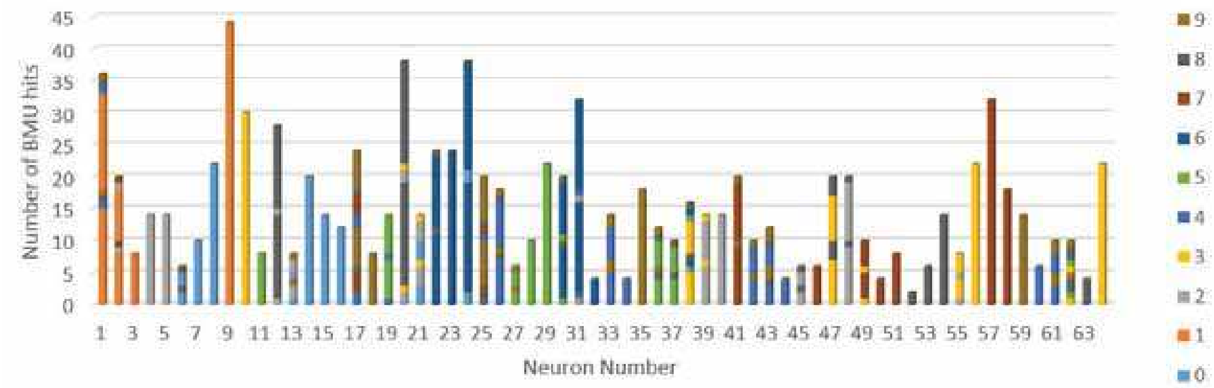


Figure 8: Data Histogram for DSOM

cluster to represent one class. The '-1' value is assigned for neurons with 0 BMU hits.

**U-Matrix:** Figure 6 (a) represents the U-matrix obtained for SOM architecture whereas Figure 6 (b) represents the U-matrix observed for DSOM architecture. It was observed that SOM U-matrix doesn't show any large clusters or cluster separations for both SOM and DSOM but for DSOM it showed some cluster separations. Large blue color area in the SOM U-matrix corresponds to the area with inactive units. Further analysis of the U-Matrix is needed to improve the visualization on this front. There are several methodologies proposed in literature on ways of calculating the U-Matrix. These methodologies will have to be explored. Furthermore, as alternatives of the u-matrix, other weight vector visualization techniques such as t-distributed Stochastic Neighbor Embedding (t-SNE) can be evaluated [33].

**Data histograms:** Figures 6 and 7 represent the data histograms obtained for SOM and DSOM architectures respectively. Data histograms visualize which units are activated and how often it became BMU compared to other units in the map. It also represents the amount of each unit acted as the BMU for each class label. According to the data histogram of SOM, it was observed that all the activated units were activated for more than one class label. There were very few units which were activated for only a single class (See neuron number 51, for class label 0). Further, number of hits per unit was significantly higher for SOM units compared to DSOM units. According to the data histogram for DSOM, it can be seen that all the units have been activated to some degree. However, it can be seen that most of the neuron has been activated only for a specific class label. In the ones that has many class labels, one class label has dominated the others in terms of frequency.

The better cluster separations observed in hit map and data histograms, make it easier for the domain expert to label the neuron and group the neurons based on classes. Hence, hit map and data histograms improve the visual data mining process.

## V. CONCLUSIONS

This paper analyzed the effectiveness of using DSOMs for Visual Data Mining. The VDM capability was evaluated using visual data exploration methodologies implemented on the DSOM. Specifically, U-Matrix, neuron hit map and data histograms were used in this study. These visualizations generated by the DSOM were compared to the single layered SOM. The algorithms were tested on the MNIST dataset. In terms of classification accuracy, DSOM significantly out-performed the SOM. In terms of visualizations for VDM, experimental results showed that DSOM based hit map and data histograms provided a better representation of the underlying data distribution than the SOM. Due to its high-level feature abstraction capabilities, DSOM is able produce visualizations which accurately reflect the input data distributions. This enables a user to examine these visualizations and extract patterns, relationships and behavior that exist in data and glean a better understanding of systems. This understanding can lead to better predictive systems, better monitoring systems and improved control schemes. Therefore, based on experimental results, it can be concluded that DSOM is a viable method for visual data mining.

## REFERENCES


[1] M.-S. Chen, J. Han, and P. S. Yu, "Data mining: an overview from a database perspective," *IEEE Trans. Knowl. Data Eng.*, vol. 8, no. 6, pp. 866–883, Dec. 1996.

[2] D. Wijayasekara, O. Linda, and M. Manic, "CAVE-SOM: Immersive visual data mining using 3D Self-Organizing Maps," in *The 2011 International Joint Conference on Neural Networks*, 2011, pp. 2471–2478.

[3] G. Liao, T. Shi, S. Liu, and J. Xuan, "A Novel Technique for Data Visualization Based on SOM," in *Artificial Neural Networks: Biological Inspirations – ICANN 2005*, 2005, pp. 421–426.

[4] M. Mehta, R. Agrawal, and J. Rissanen, "SLIQ: A fast scalable classifier for data mining," in *Advances in Database Technology — EDBT '96*, 1996, pp. 18–32.

[5] M. G. Schultz, E. Eskin, F. Zadok, and S. J. Stolfo, "Data mining methods for detection of new malicious executables," in *Proceedings 2001 IEEE Symposium on Security and Privacy. S P 2001*, 2001, pp. 38–49.

[6] D. Wu, X. Luo, G. Wang, M. Shang, Y. Yuan, and H. Yan, "A Highly-Accurate Framework for Self-Labeled Semi-Supervised Classification in Industrial Applications," *IEEE Trans. Ind. Inform.*, vol. PP, no. 99, pp. 1–1, 2017.

[7] L. Wen, X. Li, L. Gao, and Y. Zhang, "A New Convolutional Neural Network Based Data-Driven Fault Diagnosis Method," *IEEE Trans. Ind. Electron.*, vol. PP, no. 99, pp. 1–1, 2017.

[8] F. Luo *et al.*, "Advanced Pattern Discovery-based Fuzzy Classification Method for Power System Dynamic Security Assessment," *IEEE Trans. Ind. Inform.*, vol. 11, no. 2, pp. 416–426, Apr. 2015.

[9] J. R. Stack, T. G. Habetler, and R. G. Harley, "Fault classification and fault signature production for rolling element bearings in electric machines," *IEEE Trans. Ind. Appl.*, vol. 40, no. 3, pp. 735–739, May 2004.

[10] G. Littlewort *et al.*, *Towards social robots: Automatic evaluation of human-robot interaction by face detection and expression classification*. 2003.

[11] N. Grira, M. Crucianu, and N. Boujemaa, "Unsupervised and Semi-supervised Clustering: a Brief Survey," in *in 'A Review of Machine Learning Techniques for Processing Multimedia Content', Report of the MUSCLE European Network of Excellence (FP6*, 2004.

[12] I. Cohen, F. G. Cozman, N. Sebe, M. C. Cirelo, and T. S. Huang, "Semisupervised learning of classifiers: theory, algorithms, and their application to human-computer interaction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 12, pp. 1553–1566, Dec. 2004.

[13] J. Ruiz-del-Solar and P. Navarrete, "FACERET: An Interactive Face Retrieval System Based on Self-Organizing Maps," in *Image and Video Retrieval*, 2002, pp. 157–164.

[14] C. Budayan, I. Dikmen, and M. T. Birgonul, "Comparing the performance of traditional cluster analysis, self-organizing maps and fuzzy C-means method for strategic grouping," *Expert Syst. Appl.*, vol. 36, no. 9, pp. 11772–11781, Nov. 2009.

[15] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, "Engineering applications of the self-organizing map," *Proc. IEEE*, vol. 84, no. 10, pp. 1358–1384, Oct. 1996.

[16] S. Yu, H. Li, Q. Xu, and X. Wu, "Fuzzy self-organizing maps for data mining with incomplete data sets," in *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, 2010, vol. 14, pp. V14-336-V14-340.

[17] J. Vesanto, "SOM-based data visualization methods," *Intell. Data Anal.*, vol. 3, no. 2, pp. 111–126, Jan. 1999.

[18] M. C. F. de Oliveira and H. Levkowitz, "From visual data exploration to visual data mining: a survey," *IEEE Trans. Vis. Comput. Graph.*, vol. 9, no. 3, pp. 378–394, Jul. 2003.

[19] A. Rauber, D. Merkl, and M. Dittenbach, "The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data," *IEEE Trans. Neural Netw.*, vol. 13, no. 6, pp. 1331–1341, Nov. 2002.

[20] N. Liu, J. Wang, and Y. Gong, "Deep Self-Organizing Map for visual classification," in *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1–6.

[21] C. S. Wickramasinghe, K. Amarasinghe, and M. Manic, "Parallalizable deep self-organizing maps for image classification," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017, pp. 1–7.

[22] T. Kohonen, "The self-organizing map," *Neurocomputing*, vol. 21, no. 1, pp. 1–6, Nov. 1998.

[23] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[24] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *Computer Vision – ECCV 2014*, 2014, pp. 818–833.

[25] K. Perumal, "Semi-supervised and Unsupervised Methods for Categorizing Posts in Web Discussion Forums," *ArXiv160400119 Cs*, Mar. 2016.

[26] T. Fincke, V. Lobo, and O. Bação, *Visualizing self-organizing maps with GIS*. .

[27] E. Pampalk, A. Rauber, and D. Merkl, "Using Smoothed Data Histograms for Cluster Visualization in Self-Organizing Maps," in *Artificial Neural Networks — ICANN 2002*, 2002, pp. 871–876.

[28] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 586–600, May 2000.

[29] A. Ultsch and H. Siemon, "Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis," presented at the Proceedings of International Neural Networks Conference (INNC), 1990, pp. 305–308.

[30] F. Bação, V. Lobo, and M. Painho, "The self-organizing map, the Geo-SOM, and relevant variants for geosciences," *Comput. Geosci.*, vol. 31, no. 2, pp. 155–163, Mar. 2005.

[31] K. R. Babu and K. V. N. Sunitha, "Image De-noising and Enhancement for Salt and Pepper Noise Using Improved Median Filter-Morphological Operations," in *Advances in Communication, Network, and Computing*, 2012, pp. 7–14.

[32] M.-H. Hsieh, F.-C. Cheng, M.-C. Shie, and S.-J. Ruan, "Fast and efficient median filter for removing 1–99% levels of salt-and-pepper noise in images," *Eng. Appl. Artif. Intell.*, vol. 26, no. 4, pp. 1333–1338, Apr. 2013.

[33] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. Nov, pp. 2579–2605, 2008.