IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 27, Issue 4, Ser. 2 (July. – August. 2025), PP 36-46
www.iosrjournals.org

The Coherent Neural Network-Enabled Concurrent Execution Methods On CPU And GPU Architectures Using CUDA Technology

Dr. Yass K. Salal[1], Rawa Ibrahim[2]
[1]General Directorate of Education in Al- Qadisiya, Ministry of Education, Iraq
[2]Department of Computer Science, College of Computer Science and Information Technology, University of Al-Qadisiyah, Iraq

Abstract:
The development of efficient training algorithms for Neural Networks is a focus in recent years due to their high computational cost and lengthy training time. This led academics to focus on boosting Neural Network training through hardware architecture breakthroughs and parallel programming techniques. We reviewed the concepts and mechanisms of common Neural Network training algorithms, including the Coherent Algorithm. Matrix multiplication is a significant portion of the work-load during training due to its frequent occurrence. Parallel matrix multiplication is now possible with many-core GPU technologies, making Neural Network training much faster than in the past. CUDA is a high-performance parallel programming model that utilises contemporary GPU computers with multiple cores. In this paper, we suggest tweaking the coherent neural network algorithm using CUDA parallel and testing on a core GPU machine.Finally, we find that the planned tactics train Coherent Neural Networks faster than traditional methods, we found that the test sample vectors are linked to the correct cluster with a probability of over 90% for handwritten digits. The precision was measured using F-measures for each digit and clothing item category, where the Fashion MNIST dataset's "ankle boot" category has the greatest F-measure value of 0.97, while the "coat" category has the lowest at 0.4. "8" has the lowest F-measure value at 0.872, while "0" and "3" have the highest at 0.983.

Keywords*: Clustering, MNIST, Fashion MNIST, Neural network, CUDA.*

Date of Submission: 25-07-2025 Date of Acceptance: 05-08-2025

I. Introduction

In recent years, Deep Neural Networks (DNN) have gained significant popularity for a variety of applications, including the recognition of handwriting, the identification of images and objects, the classification of data, the identification of patterns, and the processing of speech and natural language.

The vectors from the test sample are identified as belonging to the correct cluster with a probability of over 90% in the case of handwritten numerals for both sequential and parallel learning.

The two categories of human activities are separated by neurone information technology [1].

1. Activities necessitating a solution that is plain, precise, and unequivocal, and that is based on a defined process and criteria.
2. Activities in which it is feasible to estimate the most critical factors but impossible to assess all reactions.

The initial activity can be resolved using conventional computer programs, as the constrained criterion set enables the development of a solution method and program to resolve the issue, irrespective of its complexity. Recognition issues in the second category are frequently "fuzzy," as evidenced by the outputs of neural networks [2, 3], and they frequently have multiple solutions. Neural networks are employed in computer diagnostics to facilitate classification. The following stages should be taken to construct the optimal neural network:

1) Selecting the activation function of the hidden layer neurone;
2) Selecting the network topology;
3) Selecting the training methods;
4) Training the network.

Consequently, the automatic identification of precise data. Handwriting recognition enhances human-computer interaction (HCI). Data classification based on image recognition that is not reliable may result in significant complications. Data clustering is the solution. Pattern recognition of the structure and vector value of each collection member is necessary for grouping.

A GPU-paralleled Kohonen neural network with CUDA is employed in the fashion MNIST datasets. In the F-measure metric, MNIST's clustering performed optimally with 0 and 3, and it performed poorly with 8. The "coat" had the lowest F-measure, while the "ankle boot" had the highest.

Additionally, we investigated the training periods of the core and GPU. The average acceleration coefficient was determined by CUDA in order to train Kohonen neural networks. The test sample was used to calculate F-measures for each digit and clustering accuracy. Fashion MNIST is classified as MNIST and contains 10,000 testing samples and 60,000 training samples. Each sample is a greyscale image that belongs to one of ten categories. Acceleration and clustering accuracy were assessed for both datasets.

Following this introduction, the rest of the article follows the same format. Section 2 provides an overview of the pertinent connected literature. Section 3 provides a mathematical illustration of the problem description. The Materials and Methods section details our approach. The findings are detailed in Section 5. Sections 6 and 7 conclude the article by examining the findings and making recommendations for future improvements.

II. Literature Review

A major obstacle to neural network algorithm development is their high processing costs [4]. In [5], resilient continuous clustering is examined. [6] Shows how to enhance MNIST handwritten digit classification. Grouping handwritten digits with k- and c-means is common [7–8]. Clustering methods include partition-based, density-based, grid-based, and hierarchical [9]. Handwritten messages were sorted using the Kohonen neural network [10]. Grouping with K-means is popular. Various datasets yield 90% accuracy with this strategy. Subspace clustering works on multidimensional data [11-12]. CUDA technology is increasingly utilised to speed up neural network training on massive datasets. Because training operations are same, this is the fundamental reason. Text recognition with neural networks was 15 times faster with CUDA [13]. Based on [14], CUDA technology enabled MapReduce distributed computing architecture self-organising maps. Kernels are optimised to use shared memory efficiently and access it well under execution. Tenfold speed gain. In [15], Batch-SOM clustering and CUDA parallel training sample level processing were applied. Speed increased thirteen-fold. Using handwritten digits from the MNIST dataset and Fashion MNIST apparel pieces, this paper provides a complete review of clustering approaches. We clustered using the Kohonen neural network. The suggested approach used the Euclidean norm to determine the distance between digit images, but other metrics could be used [16-17]. The number of clusters for each object was varied and did not exceed sixty. CUDA trained the Kohonen network on central and graphics processors.

III. Description Of The Issue From A Mathematical Perspective

Consider the issue of each category as a pattern recognition challenge [18]. Let X represents a collection of item descriptions, and Y denotes a set of class IDs, which may consist of either numerical values or names. Unidentified target dependence exists, with its values discernible solely for the objects in the final training sample. The goal is to develop an algorithm capable of classifying ($x \in X$). The basic data parameters are represented as a matrix:

$$X_{N\times P} = \left\| x_{ij} \right\|$$

Here i=1,2,…, n designate the element index; j=1,2,…, p represent the indicator index; xij indicates the value of the j-th indicator for the i-th element; n indicates the sample size; and p refers to the number of indicators. Furthermore, there is a vector comprising the values of distinct integral indicators for each item:

$$X_{N\times 1} = \{y_i\}$$

Let us formulate a representation of the function:

$$F(X,W)=Y(output)$$

Whx1 is defined as the vector of synaptic weight coefficients; Y_outputnx1 is designated as the vector of output values. Envision a neural network that performs the transformation F: X→Y, which converts the matrix X from the feature space of inputs X to the matrix - column Y of the output space Y. The neural network functions within the state W of the state space W. Additionally, imagine a training sample (Xα,Yα), where α ranges from 1 to h. We will evaluate the overall error "E" generated by the network in state W.

Where Whx1 is defined as the vector of synapse weight coefficients; Y_outputnx1 is identified as the vector of output values. Imagine a neural network that carries out the transformation F: X→Y, which changes the matrix X from the feature space of inputs X to the matrix - column Y of the output space Y. The neural network operates in the state W from the state space W. Furthermore, consider a training sample (Xα,Yα), with α taking values from 1 to p. We will analyze the total error E produced by the network in the state W.

$$E=E(W)=\sum_{\alpha} \left\| F(X^{\alpha}, W - Y^{\alpha} \right\| = \sum_{\alpha} \sum_{i} [F_i(X^{\alpha}, W) - Y^{\alpha}]^2,$$

The determination of a set of adaptive weights for the neural network, which is denoted as W*, is of utmost importance. These weights must guarantee a minimum for a particular quadratic functional, also known as min E(W)→0. The following is a description of the clustering problem that will be presented after that, for the purpose of this discussion, let us define m is the dimension of the space that is being input. A random selection is made from this space to obtain the input vector, which is then labelled as follows:

$$x = [x_1, x_2, \ldots\ldots, x_m]^T$$

The dimensions of the input space are aligned with the synaptic weight vector of each neuron during the process. To indicate the synaptic weight of a neuron, we shall use the following notation:

$$w_j = \left[w_{j1}, w_{j2}, \ldots\ldots, w_{jm}\right]^T, j = 1,2, \ldots., l$$

Where *l* represents the total count of neurons within the network, to identify the optimal vector w that corresponds to the input vector x, it is necessary to evaluate the scalar products $w_j^T x$ for j = 1,2,…,l and choose the highest value. This process will establish the position that should serve as the center of the topological neighborhood for the activated neuron. The criterion for matching, which focuses on maximizing the scalar product, is mathematically equivalent to minimizing the Euclidean distance between the vectors x and w. By utilizing the index i(x) to denote a neuron, the fundamental nature of clustering will be illustrated in this expression: The value of l indicates the total number of neurons that are contained within the network. For the purpose of determining the best vector wj that corresponds to the input vector x, it is required to evaluate the scalar products $w_j^T x$ for j = 1, 2,..., l and select the highest value. Through this approach, the position that ought to function as the center of the topological neighborhood for the activated neuron will be determined. In terms of mathematics, the criterion for matching, which is centered on the maximization of the scalar product, is similar to the minimization of the Euclidean distance between the vectors x and w. Using the index i(x) to represent a neuron, the fundamental principle of clustering will be demonstrated by the use of the following expression:

$$i(x) = \arg min_j \left\|x - w_j\right\|, j = 1,2, \ldots., l$$

The training of an artificial neural network must be optimised for graphics processors. Situational specifics and complexity determine neural network architecture [18]. There are optimal setups for tackling some problems. Developers use intuition and knowledge to choose neural network architecture. There are fundamental criteria for setting up a new configuration:

- Network performance improves with more nodes, interconnections, and layers;
- Feedback integration and increased capabilities raise concerns about dynamic stability.

Neurocomputer science has a distinct arena for studying a network's necessary and sufficient properties to solve a task. The condition greatly affects neural network building, making it difficult to provide extensive and precise guidance. Essential brain network functions depend on synaptic connections. The developer must determine the optimal variable weighting coefficients while creating a neural network architecture for a task.

The number of neurones in the input layer should equal the number of signals, according to neural network theory [18]. Similar to the output layer, the number of neurones should match the output signals. Because it's ideal, the number of synaptic weights in a neural network with one hidden layer can be estimated using the formula:

$$\frac{n_y\, n_p}{1 + log_2 n_p} \leq n_w \leq n_y \left(\frac{n_p}{n_y} + 1\right)\left(n_x + n_y + 1\right) + n_y$$

Where the value of n_w denotes the required quantity of synaptic weights that are produced by the artificial neural network; n_x and n_y represent the dimensions of the input and output signals; and n_h represents the number of elements that are contained inside the training sample. Following this, once we have established n_w, we are able to compute the total number of neurons that are present in the internal layer, also known as the hidden layer:

$$n = \frac{n_w}{n_x + n_y}$$

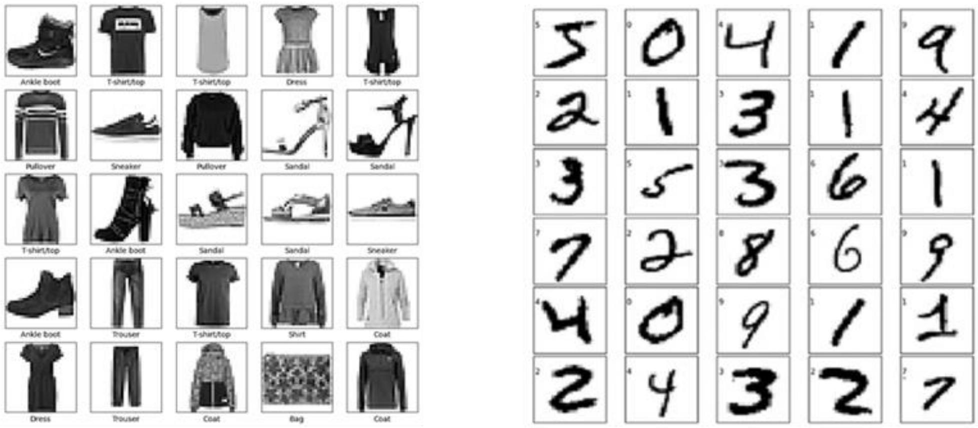
Where n_w denotes the estimated number of neurons in the hidden inner layer, research [19] shows that the precision of pattern recognition processes in selected datasets is contingent upon the neuron quantity in the hidden layer. A relatively small quantity of neurons can attain improved precision.

IV. Materials And Methods

The technique of clustering organizes a collection of items into small groupings known as clusters. The objects are grouped in such a way that within each cluster, they have a greater degree of similarity with one another than they do with entities that belong to other clusters. Consider, for example, two different kinds of data: a collection of handwritten numerals and a collection of photographs displaying 10 different categories of clothing items.

Datasets exploration & algorithm

As is widely recognized in the discipline of machine learning, the MNIST is a collection of handwritten digits that includes 60,000 training images and 10,000 test images. Each gray scale image in this database is associated with a label from one of ten different classes, and it is offered in two different formats: as a vector of pixel values and as a label, each gray scale image measures 28 pixels by 28 pixels respectively. Figure 1 depicts the dataset that was used for this investigation.



(a) **(b)**

Figure 1: Illustrates the databases utilized in the study: (a) Fashion MNIST, (b) MNIST

The MNIST and Fashion MNIST datasets are frequently employed to evaluate various methodologies in pattern recognition [20, 21], clustering [22], and other algorithmic techniques [23–25]. A diverse array of algorithms is evaluated on this database. K-nearest neighbor approaches used on MNIST yield an error rate of 5%, whereas multilayer perceptrons, contingent upon their training techniques and layer count, attain an error rate of approximately 2–5%. Convolutional neural networks (CNN) provide marginally superior performance with an error rate of under 1%, whereas hierarchical neural networks (HNN) can enhance accuracy further [26].

It is imperative to emphasize the adaptability of neural networks in general, indicating that it is vital to demonstrate that the suggested algorithm can categorize not only handwritten numbers but also many types of images. The Fashion MNIST dataset was employed for this purpose.

The Fashion-MNIST dataset contains data with dimensions of 28 by 28 pixels and consists of ten categories: T-shirt, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot. The training dataset comprises 60,000 images, whereas the test dataset consists of 10,000 images.

Kohonen neural network is an unsupervised learning network including a single layer with modifiable weights. The neural network weights are adjusted to ensure that vectors inside the same cluster activate the identical output neuron [27].

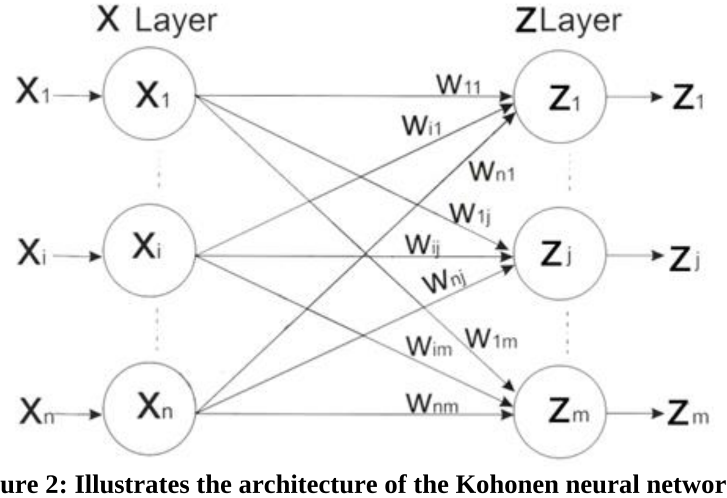


Figure 2: Illustrates the architecture of the Kohonen neural network

The outputs of the neural network, denoted as z_j, are calculated using the formula $z_j = \Sigma iwijx_i$, , in this context, xi denotes the input values utilised by the Kohonen neural network. The dimensions of the images in the MNIST dataset are 28 pixels by 28 pixels.

The Kohonen neural network comprises 790 neurons. The weights w_{ij} are the centres of the clusters and help train the network. The training procedure for the Kohonen neural network can be encapsulated by the subsequent algorithm.

Step 1: Standardize the input vectors x_p.
Step 2: Randomly select the values of the weights w_{ij} from the training sample vectors. This is significant because, in cases of non-uniform distribution (where weights are allocated randomly), the weights may be positioned distant from the input vectors and so fail to participate in the training process, this does not engage in the training.
Step 3: The distances between the input vectors x_p, characterized by coordinates x_{pi} and weights w_{ij}, are calculated using the formula:

$$\text{Dj}= \left(\sum_{i=1}^{784} \left(x_i^p - w_{ij}\right)^2\right)^{1/2}$$

The winning vector is determined as the vector wj that has the minimum distance Dj to the input vector xp.

Step 4: The coordinates of the selected winning vector from the previous step are adjusted using the formula: $w_l= w_l+ \theta(x^p-w_l)$, where q denotes the learning rate.

Step 5: Steps 3 and 4 are repeated for all vectors x_p. The value of θ is determined using the formula $\theta = \alpha\theta$, where $\alpha < 1$. If $q > e$, the process proceeds to step 3, where the weights are modified after evaluating all input vectors once again.

Compute Unified Device Architecture (CUDA) technology

CUDA memory operations outperform clusters [28]. From the seventh series onwards, NVIDIA graphics accelerators use CUDA parallel computing architecture with a non-graphical processing software interface. An individual Kohonen neural network is needed for each digit, making training time-consuming. Parallelisation will employ CUDA [29].

CUDA-capable GPUs use multi-core CPUs. In these video devices, multiprocessors share thousands of memory registers, texture, and constant caches. Removing vertex and pixel processing data types and computational concepts during frame construction is a big benefit of CUDA. Mathematics and physics can be accelerated by several dozen times on normal computers using CUDA technology [30–31]. On standard CPUs, CUDA technology enhances mathematical and physical problem performance by orders of magnitude. A neural network using CUDA, OpenMP, and central processor parallelism increases hardware-software interface and programming language performance.

The newest software library versions and tips on using graphics processors for general computations are available on the NVIDIA developer portal. Although the NVIDIA webpage is extensive, research [28][32] gives direct performance assessments and numerical integration with graphics processors. Applying these strategies is valid and can boost growth.

V. Results

Clusters of Handwritten Digits

The variation in writing styles for the same digit makes artificial intelligence recognition of handwritten digits difficult. Using the MNIST database, many digit representations are evident. For example, the digit 1 might be written straight or slightly slanted. The number of these properties varies greatly by digit. Because of this, each digit may have different clusters.

The number of clusters needed to segment input images is unknown. Find the right number for each digit. The following automatic clustering method will be used [33]:
step 1: using the Euclidean metric, we calculate the squares of all vector distances in the training sample. A matrix is made from the values.

$$d^2(x^i, x^j) = \left\|x^i, x^j\right\|^2 = \sum_{l=1}^{n} \left(x_l^i, x_l^j\right)^2$$

step 2. Determine the greatest value of the matrix created in the preceding step. This element denotes the maximum distance among all vectors (max $d^2(x^i, x^j)$).
step 3. Establish the permissible distance between vectors inside the same cluster as a specified percentage of the maximum distance among the vectors.
step 4. Select an arbitrary i-th column from the matrix (vector x^i). Classify all entries in this column with values below the permissible threshold as part of the same cluster (with rows designated as j). Omit the i-th column along with all j-th columns.

Step 5. If the matrix remains non-empty, return to step 4.

Table 1 shows the optimal number of clusters for the Fashion MNIST dataset, which is the outcome of using artificial clustering. This number is then applied to each digit.

The categories "Shirt" and especially "Bag" exhibit a restricted number of clusters. This can be ascribed to the unambiguous nature of the visuals; all images categorized as "Bag" are regarded as fundamental rectangles.

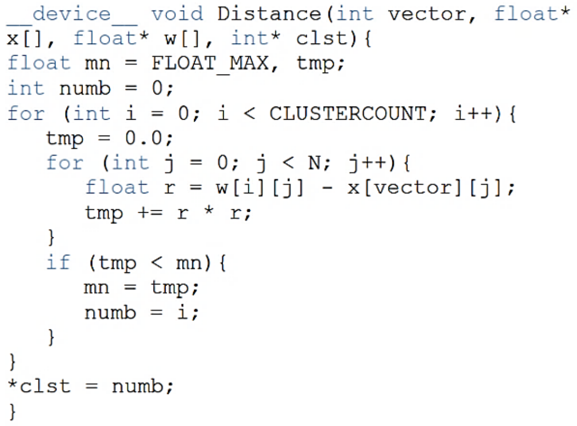
Table 1: The optimal number of clusters for clothes items and digits from the Fashion MNIST and MNIST datasets respectively.

Fashion MNIST		MNIST	
Type of clothes item	NO. clusters	Digital	NO. clusters
T-shirt/Top	27	0	38
Trouser	36	1	47
Pullover	30	2	29
Dress	36	3	45
Coat	42	4	27
Sandal	44	5	31
Shirt	11	6	48
Sneaker	47	7	28
Bag	1	8	36
ankle boot	39	9	29

So, after determining the ideal number of clusters for each digit, we then utilise the Kohonen neural network to construct the clusters themselves. This will be done once we have determined the optimal number of clusters.

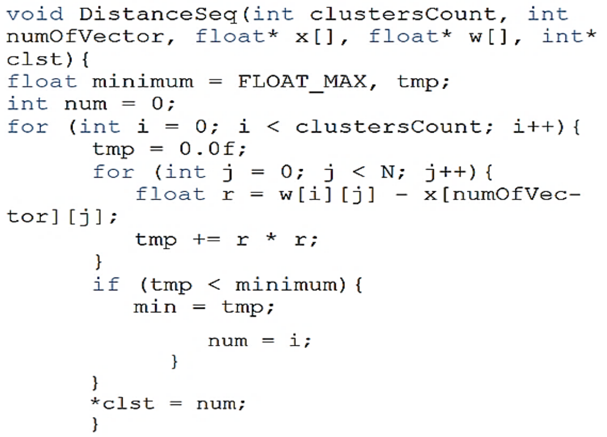
Implementation of the Kohonen neural network

The ideal parameters for training the Kohonen neural network were determined experimentally: α = 0.96, ε = 0.02, θ = 0.6. The following are the principal phases of training this network, which require the most time. The initial step influences the computation of the distance between the weights and the vector derived from the training sample. We will introduce a function to compute the distance and determine the minimal distance on the processor (m = 39):



```
__device__ void Distance(int vector, float*
x[], float* w[], int* clst){
float mn = FLOAT_MAX, tmp;
int numb = 0;
for (int i = 0; i < CLUSTERCOUNT; i++){
   tmp = 0.0;
   for (int j = 0; j < N; j++){
      float r = w[i][j] - x[vector][j];
      tmp += r * r;
   }
   if (tmp < mn){
      mn = tmp;
      numb = i;
   }
}
*clst = numb;
}
```

This section delineates the distance from the vector in the training set x[numOfVector] to each cluster w[i]. The total number of clusters, referred to as clustersCount, ranges from 28 to 49 depending on the digit being trained. The complexity of this function is 0 (number of clusters * N), where N equals 784, as all pixels of the image are iterated over each cluster. The subsequent step influences the navigation of all vectors and modifies the weights of the neurons. The second part of the training segment involves transmitting all vectors and adjusting the weights of the neurons. The procedure for training the Kohonen neural network is sequentially outlined as follows; all of the vectors that were included in the training sample are put through a pass during the training section. The variable vectorsCount is used to represent the total number of vectors that are present in the sample, and the variable distMin is used to specify the cluster number that is linked with the current vector image.

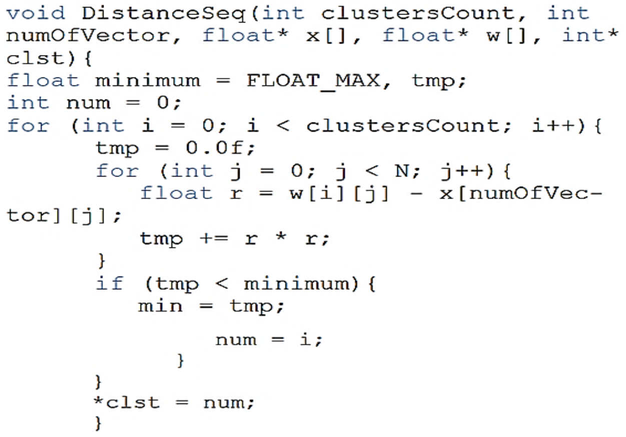


```
void DistanceSeq(int clustersCount, int
numOfVector, float* x[], float* w[], int*
clst){
float minimum = FLOAT_MAX, tmp;
int num = 0;
for (int i = 0; i < clustersCount; i++){
      tmp = 0.0f;
      for (int j = 0; j < N; j++){
         float r = w[i][j] - x[numOfVec-
tor][j];
         tmp += r * r;
      }
      if (tmp < minimum){
         min = tmp;
              num = i;
           }
      }
      *clst = num;
      }
```

Kohonen neural network training algorithm with CUDA technology

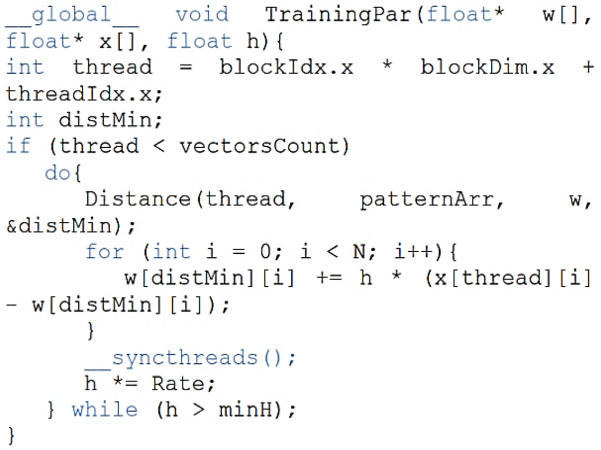
As mentioned, the algorithm comprises two lengthy steps. Around 6,000 images with 784 input vector dimensions and 28 to 49 clusters per digit make up the training dataset. Network training usually takes a few dozen cycles. The scenario with the parameters in the preceding section requires 84 iterations of training. Further parallelisation layers include training phase, sample, layer, neurone, and weights. The amount of neurones, computing nodes, and artificial neural network computational design determine the parallelisation level. This study uses parallelisation at the training sample level, requiring several input vectors. We will determine the number of parallelisation blocks, ensuring each block contains 256 threads for input vectors. One vector from the training sample will be processed by each thread. The algorithm's speed improves by eliminating the loop that scans every image. Graphics processor distance function calculates vector distance.

This function is implemented as follows:



```
void DistanceSeq(int clustersCount, int
numOfVector, float* x[], float* w[], int*
clst){
float minimum = FLOAT_MAX, tmp;
int num = 0;
for (int i = 0; i < clustersCount; i++){
      tmp = 0.0f;
      for (int j = 0; j < N; j++){
         float r = w[i][j] - x[numOfVec-
tor][j];
         tmp += r * r;
      }
      if (tmp < minimum){
         min = tmp;
              num = i;
           }
      }
      *clst = num;
      }
```

Here is the code for the GPU function that training the Kohonen network:



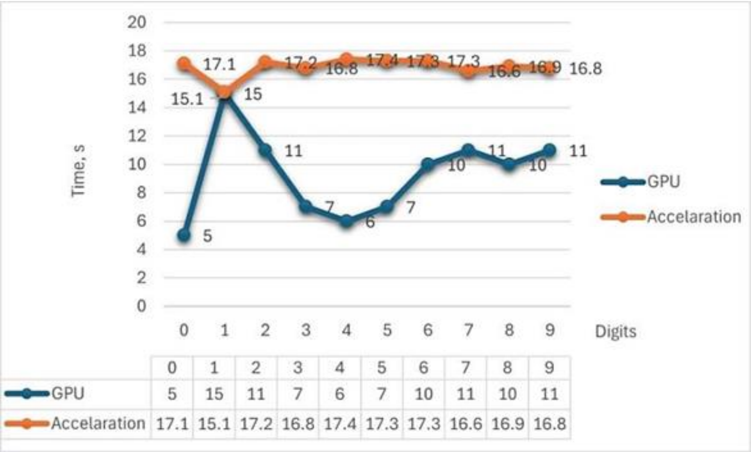
```
__global__  void  TrainingPar(float*  w[],
float* x[], float h){
int  thread  =  blockIdx.x  *  blockDim.x  +
threadIdx.x;
int distMin;
if (thread < vectorsCount)
   do{
      Distance(thread,    patternArr,    w,
&distMin);
      for (int i = 0; i < N; i++){
         w[distMin][i] += h * (x[thread][i]
- w[distMin][i]);
      }
      __syncthreads();
      h *= Rate;
   } while (h > minH);
}
```

In this context, Rate represents the parameter θ. It is crucial to acknowledge that during the for loop, the weight w might be simultaneously modified by many threads. However, due to the substantial sample size and the frequent modifications to the weight, this 'conflict' of inputting new values into the array is not significant and has negligible effect on the ultimate outcome.

The computer utilized for these computations within the Visual Studio 2022 environment possesses the following specifications:

- Operating system – Windows 10, 64,
- GPU model – NVidia GeForce GTX GeForce RTX 4080,
- GDDR 6x size – 8 GB,
- CPU model – Intel Core i7-5600HQ 2.60 GHz,
- RAM size – 16 GB.

The execution time data for the algorithm associated with each digit allows for the calculation of the speedup achieved through CUDA technology. The speedup factor is defined as the ratio of the execution time of the algorithm on the processor compared to that on the graphics processor. The results are presented in Figure 3.



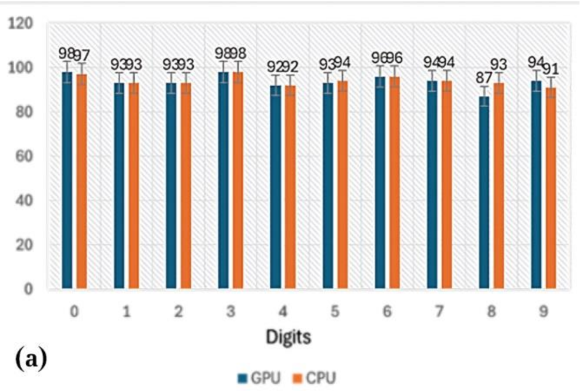
	0	1	2	3	4	5	6	7	8	9
GPU	5	15	11	7	6	7	10	11	10	11
Accelaration	17.1	15.1	17.2	16.8	17.4	17.3	17.3	16.6	16.9	16.8

Figure 3: Training time of Kohonen neural network utilizing GPU and GPU acceleration in clustering.

The results show that Kohonen neural network training improves by 16.9. Similar acceleration is seen for Fashion MNIST. Use the test sample to verify cluster formation accuracy. We will measure the proportion of vectors representing MNIST handwritten digits database images in the cluster. The distance from the digit vector to all digit clusters must be calculated. We consider an image vector to be in the target cluster if it is spatially close to the digit cluster set. Figure 3 shows the proportions of test sample images in their digit clusters. Whether CPU and GPU clusters are formed independently is important.

Figure 4. a, b shows how central processing units and graphic processor units identify garments differently. Thus, sequential and parallel algorithms share characteristics. The illustration shows that CPU and GPU clusters give nearly comparable results. These results cannot be substituted. To assess Kohonen neural network grouping accuracy for MNIST data, we calculate the F-measure. We calculate the Fashion MNIST dataset F-measure (Table 2).

The "coat" class is underrepresented, but the "ankle boot" and "trouser" classes are distinguished. Kohonen neural network recognition averages 74%. Fashion MNIST performs worse than MNIST in object recognition. The dataset's less defined cluster centres may cause clustering errors, explaining the difference.



(a)

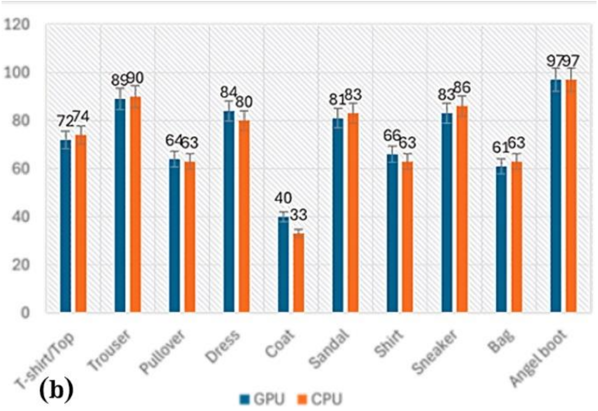


Figure 4: The distribution of GPU and CPU utilization in the test sample for clustering is as follows: a) MNIST dataset; b) Fashion MNIST dataset.

VI. Discussion

Pattern recognition is complicated by different numerical representations. Cluster number or size may increase significantly. Uniform numerical representation is also difficulty. If the cluster count is low, clusters representing different digits may overlap [34].

Because most of the test sample's digit images are in the right group, Figure 4 and Table 2 show that the Kohonen neural network can recognise handwritten digit patterns. Recognition is worst for Digit 8. Kohonen neural networks identify 1 almost always, with nearly 100% accuracy. Figure 4 histograms show that the CUDA-implemented technique clusters effectively even with many clusters. Thus, for 0, 1, 3, 6, and 8, the number of clusters varies between 45 and 50, while the test sample fraction inside each cluster remains steady. The central processor's clustering accuracy is better for the numbers 4, 5, 7, and 9, whose cluster count approaches 30. Adding clusters improves GPU clustering. As part of an ensemble or hierarchical neural network, the clusters may recognise images. This can modify cluster numbers. Hopfield neural networks can store up to 50 objects for MNIST images [35].

Table 2: F-measure for clusters generated on GPU and CPU using the Fashion MNIST and MNIST datasets

Digits	GPU	CPU	Clothes items	GPU	CPU
0	**0,98**	**0,97**	**T-shirt/Top**	**0,72**	**0,74**
1	**0,93**	**0,93**	**Trouser**	**0,89**	**0,9**
2	**0,93**	**0,93**	**Pullover**	**0,64**	**0,63**
3	**0,98**	**0,98**	**Dress**	**0,84**	**0,80**
4	**0,92**	**0,93**	**Coat**	**0,40**	**0,33**
5	**0,93**	**0,94**	**Sandal**	**0,81**	**0,83**
6	**0,96**	**0,96**	**Shirt**	**0,66**	**0,63**
7	**0,94**	**0,94**	**Sneaker**	**0,83**	**0,86**
8	**0,87**	**0,93**	**Bag**	**0,61**	**0,63**
9	**0,94**	**0,91**	**ankle boot**	**0,97**	**0,97**

VII. Conclusions

The Kohonen neural network was used to cluster MNIST handwritten digits and Fashion MNIST apparel items. For each digit, the ideal cluster count was found. The study showed that test sample photographs are associated to the right cluster with a relatively high probability for each digit. F-measure digits 0 and 3 were 0.98, indicating the best clustering results. The F-measure for 8 was 0.87, indicating the worst clustering. At 0.97, "ankle boot" received the highest Fashion MNIST F-measure rating. On the other hand, the category "coat" had the lowest value, 0.4, and CUDA technology parallelises Kohonen neural network training on a GPU. Parallelisation is done at the training sample level. Between datasets, the approach is projected to be 18 times faster. F-metric values for the two databases analysed range widely, but clustering findings are similar. MNIST's maximum F-measure difference is 0.02, while Fashion MNIST's is 0.05. The presented method is appropriate for managing large amounts of data without losing clustering accuracy.

For the future work, we will once again examine the current algorithm to determine whether it can be utilised in Cryptography, Image Processing, video Frames, Bio-Informatics, and Weather Forecasting to enhance performance by leveraging the capabilities of contemporary multi-core CPU-based and manycore GPU-based systems, utilising CUDA and other parallel programming models and paradigms.

References

[1]. Cheng, Chen, And Guang-Tao Zhang. "Deep Learning Method Based On Physics Informed Neural Network With Resnet Block For Solving Fluid Flow Problems." Water, Vol. 13, No. 4, 2021, P. 423, Doi:10.3390/W13040423.

[2]. Sorano, Ruslan, Et Al. "Evolutionary Optimization Of Artificial Neural Networks And Tree-Based Ensemble Models For Diagnosing Deep Vein Thrombosis." Linköping Electronic Conference Proceedings, Vol. 208, 2024, Pp. 178–87, Doi:10.3384/Ecp208020.

[3]. E Modhej, Nazal, Et Al. "Pattern Separation Network Based On The Hippocampus Activity For Handwritten Recognition." IEEE Access, Vol. 8, 2020, Pp. 212803–17, Doi:10.1109/Access.2020.3040298.

[4]. Park, Sang-Min, And Young-Gab Kim. "A Metaverse: Taxonomy, Components, Applications, And Open Challenges." IEEE Access, Vol. 10, Jan. 2022, Pp. 4209–51, Doi:10.1109/Access.2021.3140175.

[5]. M. Parseh, M. Rahmanimanesh, And P. Keshavarzi, "Persian Handwritten Digit Recognition Using Combination Of Convolutional Neural Network And Support Vector Machine Methods," The International Arab Journal Of Information Technology, Vol. 17, No. 4, Pp. 572–578, Jun. 2020.

[6]. Y. Pei And L. Ye, "Cluster Analysis Of MNIST Data Set." Journal Of Physics Conference Series, Vol. 2181, No. 1, 2022, P. 012035, Doi:10.1088/1742-6596/2181/1/012035.

[7]. Obaidullah, Sk. Md., Et Al. "Handwritten Indic Script Identification In Multi-Script Document Images: A Survey." International Journal Of Pattern Recognition And Artificial Intelligence, Vol. 32, No. 10, Apr. 2018, P. 1856012, Doi:10.1142/S0218001418560128.

[8]. Haidar, Imane M., Et Al. "High Performance And Lightweight Single Semi-Lattice Algebraic Machine Learning." IEEE Access, Vol. 12, 2024, Pp. 50517–36, Doi:10.1109/Access.2024.3376525.

[9]. A. B. Ayed, M. B. Halima, And A. M. Alimi, "Survey On Reddy, Akkala Yugandhara, And Tarakeswara Rao Balaga. "Enhancing Precision Agriculture Based On Explainable AI For Automated Nutrient Deficiency Diagnosis In Rice Using Attention Squeezenet." Ingénierie Des Systèmes D Information, Vol. 30, No. 1, 2025, Pp. 181–90, Doi:10.18280/Isi.300115.

[10]. Rajpal, Danveer, And Akhil Ranjan Garg. "Deep Learning Model For Recognition Of Handwritten Devanagari Numerals With Low Computational Complexity And Space Requirements." IEEE Access, Vol. 11, 2023, Pp. 49530–39, Doi:10.1109/Access.2023.3277392.

[11]. Guo, Wengang, Et Al. "Deep Embedded K-Means Clustering." 2021 International Conference On Data Mining Workshops (ICDMW), Dec. 2021, Pp. 686–94, Doi:10.1109/Icdmw53433.2021.00090.

[12]. J. Cai, J. Fan, W. Guo, S. Wang, Y. Zhang, And Z. Zhang, "Efficient Deep Embedded Subspace Clustering." 2022 IEEE/CVF Conference On Computer Vision And Pattern Recognition (CVPR), 2022, Pp. 21–30, Doi:10.1109/Cvpr52688.2022.00012.

[13]. K. S. Mohamed, "Parallel Computing: Openmp, MPI, And CUDA," In Springer Ebooks, 2020, Pp. 63–93.

[14]. B. Huang, R. Cheng, Z. Li, Y. Jin, And K. C. Tan, " IEEE Transactions On Evolutionary Computation, 2024, P. 1, Doi:10.1109/Tevc.2024.3388550.

[15]. Shafiq, Muhammad, And Zhaoquan Gu. "Deep Residual Learning For Image Recognition: A Survey." Applied Sciences, Vol. 12, No. 18, 2022, P. 8972, Doi:10.3390/App12188972.

[16]. B. Gope, S. Pande, N. Karale, S. Dharmale, And P. Umekar, "Handwritten Digits Identification Using Mnist Database Via Machine Learning Models." IOP Conference Series Materials Science And Engineering, Vol. 1022, No. 1, 2021, P. 012108, Doi:10.1088/1757-899x/1022/1/012108.

[17]. Henshaw, Christopher, Et Al. "Number Recognition Through Color Distortion Using Convolutional Neural Networks." Computers, Vol. 14, No. 2, 2025, P. 34, Doi:10.3390/Computers14020034.

[18]. Krivalcevich, E. A., And M. I. Vashkevich. "Investigation Of Hardware Implementation Of A Feedforward Neural Network For Handwritten Digit Recognition Based On FPGA." Doklady BGUIR, Vol. 23, No. 2, 2025, Pp. 101–08, Doi:10.35596/1729-7648-2025-23-2-101-108..

[19]. Saxena, Aarush. "An Introduction To Convolutional Neural Networks." International Journal For Research In Applied Science And Engineering Technology, Vol. 10, No. 12, Dec. 2022, Pp. 943–47, Doi:10.22214/Ijraset.2022.47789.

[20]. Paniccia, D., Et Al. "A Supervised Machine-Learning Approach For Turboshaft Engine Dynamic Modeling Under Real Flight Conditions." The Aeronautical Journal, July 2025, Pp. 1–25, Doi:10.1017/Aer.2025.10024.

[21]. Gope, Birjit, Et Al. "Handwritten Digits Identification Using MNIST Database Via Machine Learning Models." IOP Conference Series Materials Science And Engineering, Vol. 1022, No. 1, 2021, P. 012108, Doi:10.1088/1757-899x/1022/1/012108.

[22]. Chen, Di, Et Al. "Automating Crystal-Structure Phase Mapping By Combining Deep Learning With Constraint Reasoning." Nature Machine Intelligence, Vol. 3, No. 9, Sept. 2021, Pp. 812–22, Doi:10.1038/S42256-021-00384-1.

[23]. Fard, Maziar Moradi, Et Al. "Deep K-Means: Jointly Clustering With K-Means And Learning Representations." Pattern Recognition Letters, Vol. 138, July 2020, Pp. 185–92, Doi:10.1016/J.Patrec.2020.07.028.

[24]. Dwivedi, Yogesh K., Et Al. "Setting The Future Of Digital And Social Media Marketing Research: Perspectives And Research Propositions." International Journal Of Information Management, Vol. 59, July 2020, P. 102168, Doi:10.1016/J.Ijinfomgt.2020.102168.

[25]. A. M. Roy, J. Bhaduri, T. Kumar, And K. Raj, "Wildect-YOLO: An Efficient And Robust Computer Vision-Based Accurate Object Localization Model For Automated Endangered Wildlife Detection." Ecological Informatics, Vol. 75, Nov. 2022, P. 101919, Doi:10.1016/J.Ecoinf.2022.101919.

[26]. Z. Kayumov, D. Tumakov, And S. Mosin, "Hierarchical Convolutional Neural Network For Handwritten Digits Recognition," Procedia Computer Science, Vol. 171, 2020, Pp. 1927–1934.

[27]. Fonseca, Gustavo, And Danilo Candido Vieira. "Overcoming The Challenges Of Data Integration In Ecosystem Studies With Machine Learning Workflows: An Example From The Santos Project." Ocean And Coastal Research, Vol. 71, No. Suppl 3, Jan. 2023, Doi:10.1590/2675-2824071.22044gf.

[28]. Y. Nagasaka, A. Nukada, And S. Matsuoka, "Cache-Aware Sparse Matrix Formats For Kepler GPU," In 2014 20th IEEE International Conference On Parallel And Distributed Systems, IEEE., Pp. 281–288, Dec. 2014.

[29]. Raschka, S., Patterson, J., & Nolet, C. Machine Learning In Python: Main Developments And Technology Trends In Data Science, Machine Learning, And Artificial Intelligence. Information, 11(4), 2020, 193. Https://Doi.Org/10.3390/Info11040193.

[30]. Shirokanev, A., Andriyanov, N., & Ilyasova, N. Development Of Vector Algorithm Using CUDA Technology For Three-Dimensional Retinal Laser Coagulation Process Modeling. Computer Optics, 45(3), 2021. Https://Doi.Org/10.18287/2412-6179-Co-828.

[31]. D. P. Tabakov, S. V. Morozov, And D. S. Klyuev, "Application Of The Thin-Wire Integral Representation Of The Electromagnetic Field To Solving The Problem Of Diffraction Of Electromagnetic Waves On Conducting Bodies." Physics Of Wave Processes And Radio Systems, Vol. 25, No. 2, 2022, Pp. 7–14, Doi:10.18469/1810-3189.2022.25.2.7-14.

[32]. J. A. Stuart And J. D. Owens, "Multi-GPU Mapreduce On GPU Clusters," In 2011 IEEE International Parallel & Distributed Processing Symposium, IEEE., Pp. 1068–1079, May 2011.

[33]. Taha, Kamal. "Observational And Experimental Insights Into Machine Learning-Based Defect Classification In Wafers." Journal Of Intelligent Manufacturing, Jan. 2025, Doi:10.1007/S10845-024-02521-0.

[34]. J. Sun, J. Li, R. Li, L. Wu, L. Cao, And M. Sun, "Addressing Unfamiliar Ship Type Recognition In Real-Scenario Vessel Monitoring: A Multi-Angle Metric Networks Framework," Frontiers In Marine Science, Vol. 11. 2025, Doi: 10.3389/Fmars.2024.1516586.

[35]. Liu, Ge, Et Al. "A Quantum Hopfield Neural Network Model And Image Recognition." Laser Physics Letters, Vol. 17, No. 4, Feb. 2020, P. 045201, Doi:10.1088/1612-202x/Ab7347.