International Journal of Multimedia and Ubiquitous Engineering
Vol. 10, No. 1 (2015), pp. 341-352
http://dx.doi.org/10.14257/ijmue.2015.10.1.32

Accelerating 2D Fault Diagnosis of an Induction Motor using a Graphics Processing Unit

Jia Uddin[1], Dinh Nguyen Van[2], and Jong-Myon Kim[1*]

[1]School of Electrical Engineering, University of Ulsan, South Korea
[2]School of Electrical Engineering, Hanoi University of Science and Technology, Vietnam
E-mail: {engrjiauddin, nguyendinh1987, jongmyon.kim}@gmail.com}

Abstract

This paper presents a computationally efficient graphics processing unit (GPU) implementation of a reliable fault diagnosis method using two-dimensional (2D) representation of vibration signals. The fault diagnosis method first converts time-domain vibration signals into 2D gray-level images to exploit texture information from the converted images. Then, the global dominant neighborhood structure (GNS) map is utilized to extract texture features by averaging local neighborhood structure (LNS) maps of central pixels. In addition, the principle component analysis (PCA) algorithm is employed to select only the most dominant features. Finally, the selected features are used as inputs to a one-against-all multi-class support vector machine (OAA-MCSVM) to identify each fault of the induction motor. Despite the fact that the 2D fault diagnosis methodology shows satisfactory classification accuracy, its computational complexity limits its use in real-time applications. To accelerate the 2D fault diagnosis method, this paper utilizes an NVIDIA GeForce GTX 580 GPU, where all tasks are executed in parallel. The experimental results indicate that the proposed GPU-based approach achieves about 118.5× faster operation than the equivalent sequential CPU implementation while maintaining 100% classification accuracy.

Keywords: *Fault diagnosis, induction motor, global dominant neighborhood structure (GNS), local neighborhood structure (LNS), graphics processing unit*

1. Introduction

Undoubtedly, an induction motor is a vital component in much industrial equipment, such as pumps, air compressors, and conveyor belts [1]. However, various parts of the induction motor, including the bearings, stator, and rotor, are involved in several faults that can occur after long-term operation of the motor, which leads to deterioration of the industrial processes' productivity. Therefore, reliable condition monitoring is an important issue in current industrial processes.

In general, fault diagnosis of an induction motor is performed by analyzing current, voltage, vibration, acoustic emission (AE), and thermal signals [2,3]. Especially, current and voltage signals have been widely used to classify abnormalities of the induction motor due to their easy signal measurement and their low cost to monitor. However, these signals contain unnecessary components such as harmonic characteristics that usually cause signal distortion. In contrast, AE and thermal signals operate in higher frequency bands with lower energy levels compared to current and voltage signals. The AE signal can be used to identify various faults earlier than the commonly used vibration signal; however, it is

[*] Corresponding author

ISSN: 1975-0080 IJMUE

highly affected by various types of noise [4]. Therefore, the use of vibration signals in state monitoring of an induction motor is a preferred alternative due to their higher energy level.

In conventional fault diagnosis methods, signals are processed in one of the following domains: time, frequency, or time–frequency. Statistical time-domain features are robust with respect to the changing-load issue, while frequency features are very close to the rotation properties of a machine. However, both approaches may lose significant amounts of signal information during operations, especially for non-stationary and nonlinear signals. As an alternative, time–frequency analysis is widely used for non-stationary and nonlinear vibration signals because it presents detailed frequency coefficients. Examples of time–frequency signal analysis methods include short-time Fourier transform (STFT), wavelet transform, Wigner-Ville distribution, Hilbert Huang transform (HHT), and Teager-Huang transform (THT) [5, 6]. In the STFT, the limitation of the trade-off between time and frequency resolutions is a major constraint. Although the wavelet transform (WT) has improved performance over the STFT, it has a non-adaptive nature due to its uniform resolution for all scales. The WVD of discrete time signals suffers from the aliasing problem. More usefully, the combined HHT/THT is adaptive to input signals and can select intrinsic mode functions (IMFs) based on the input signals. However, selection of the number of IMFs based on a threshold value is a limiting factor in both the HHT and the THT [7].

In conventional 1D signal processing approaches, significant signal information may be lost due to the deficiency in relative information between time and frequency coefficients. In addition, the features collected from the raw or processed signals have large dimensions that tend to increase the computational burden of the subsequent classifier and degrade its generalization capability. Further, the process of collecting feature vectors directly from the raw signal is not an efficient approach in a practical scenario, where load and rotational speeds are varied at any instant.

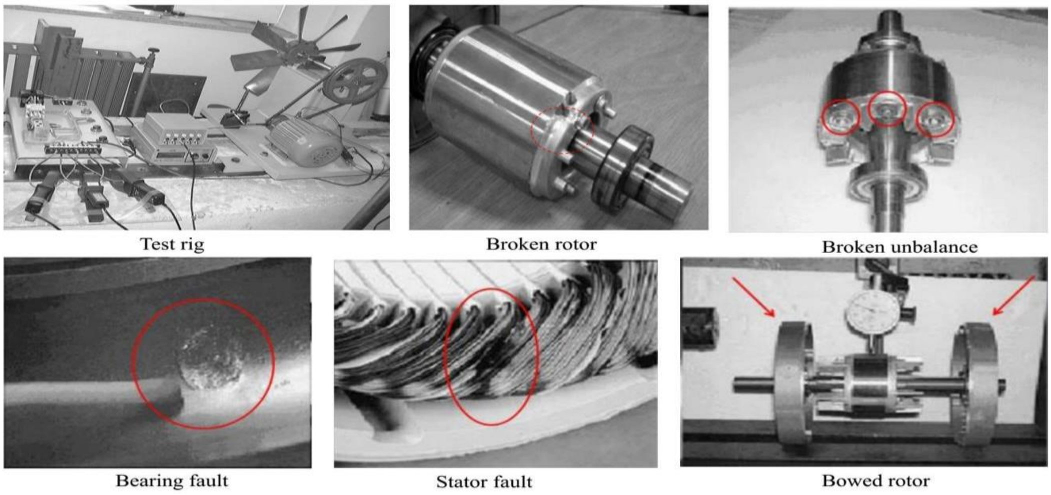
To address this issue, some researchers have utilized texture information in their 2D fault diagnosis algorithms [8-10], where the texture image converted from the 1D fault signal gives additional feature information. However, compared to conventional 1D approaches, 2D fault-diagnosis algorithms require additional steps, including conversion from 1D signals to 2D gray images and texture feature extraction, which increases the computational complexity. This drawback has motivated research on parallel processing of 2D fault diagnosis algorithms using a graphics processing unit (GPU).

This paper proposes an efficient parallel implementation of the GNS/LNS map-based texture fault-diagnosis method [8] on a GPU. The selected fault diagnosis method utilizes a 2D gray image converted from a 1D vibration signal to extract texture features from the converted images by using local neighborhood structure (LNS) maps, where a global dominant neighborhood structure (GNS) map is constructed by averaging the LNS maps of all central pixels in the image. Principle component analysis (PCA) is then utilized to select the most distinctive features for multi-class support vector machines (MCSVMs). Additionally, this paper compares the performance of this GPU-based approach with that provided by an equivalent sequential CPU implementation.

The rest of this paper is organized as follows. Section 2 presents detailed information about the vibration signals of an induction motor. Section 3 describes the GNS map-based texture fault diagnosis model, and Section 4 describes the proposed GPU-based implementation of the fault diagnosis method and compares the performance of the GPU-based approach with the equivalent CPU implementation. Finally, Section 5 provides conclusions for the paper.

2. Vibration Signals of an Induction Motor

To collect vibration signals of induction motors, we set up an experiment using pulleys, a belt, a shaft, a fan with changeable blade pitch angle, and two pole induction motors, as depicted in Figure 1, with the following characteristics: 0.5 kW, 60 Hz, 220 V, and 3560 rpm. In the experiment, seven different faults (see Table 1) and one healthy operation were acquired under full-load and steady-state conditions: broken rotor bars (BRBF), bowed rotor shaft (BSF), bearing outer race fault (BF), rotor imbalance fault (RIF), parallel misalignment fault (PMF), angular misalignment fault (AMF), phase imbalance fault (PIF), and healthy (NO). The sampling frequency of each fault was 8 KHz. Each fault had 105 signals of one-second duration. To evaluate the classification accuracy of the 2D fault diagnosis model, a set of 53 one-second signals was used for training and the remaining 52 signals were used for testing.



Test rig

Broken rotor

Broken unbalance

Bearing fault

Stator fault

Bowed rotor

Figure 1. A Test Rig and Various Induction Motor Faults

Table 1. Faults of an Induction Motor Used in this Study

Type of faults	Fault description
AMF	AM is the effective angle between two shaft centerlines, and the angle between shaft centerlines is 0.48^0
BRBF	Among the 34 rotor bars, 12 are involved in plastic deformation of a grinding furrow: 5 mm in diameter and 15 mm in depth
PMF	The offset between two centerlines of the motor and load has been changed 0.1 mm
RIF	Unbalanced mass of 15.64 g·cm is added at the right end of the rotor
BF	A spalling on the outer race of the bearing is replicated
BSF	The shaft is slack in the middle (0.075 mm), which causes dynamic air-gap eccentricity
PIF	4.3 Ω resistance is connected to one of the three-phase wires of the induction motor

3. GNS Map-Based 2D Fault-Diagnosis Method

This section presents our recently published fault diagnosis method, called GNS map-based 2D fault diagnosis [8], which consists of the following four steps: (a) conversion from a 1D vibration signal into a 2D image, (b) construction of feature vectors, (c) extraction of the most dominant features, and (d) fault classification. Figure 2 shows a block diagram of the GNS map-based fault diagnosis method. To convert a 1D time-domain signal into a 2D gray image, we divide sample values of the vibration signal into a number of same-length subparts and arrange each subpart as a column in a 2D matrix. The length of the subpart decides the height of the matrix and the number of subparts defines the width of the matrix.

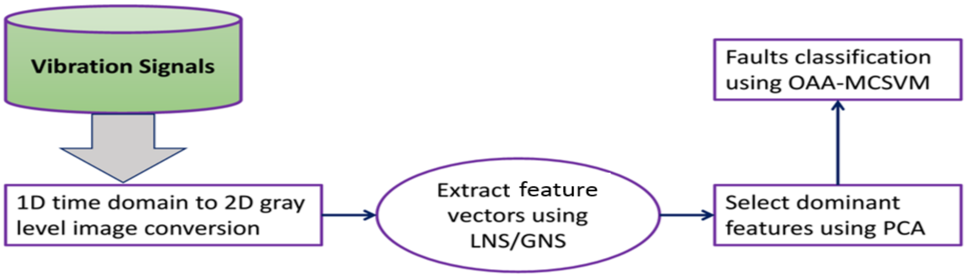

Vibration Signals
1D time domain to 2D gray level image conversion
Extract feature vectors using LNS/GNS
Select dominant features using PCA
Faults classification using OAA-MCSVM

Figure 2. A Block Diagram of the GNS Map-based 2D Faults-diagnosis Method [8]

Figure 3 illustrates a detailed conversion process from a 1D time-domain signal into a 2D texture gray image. Sample values are normalized in the range of 0–255 to convert the matrix into an image, where normalized amplitudes represent intensities of the corresponding pixels in the image. The coordinates of the pixel corresponding to the i^{th} sample of a signal is pixel (k,j), where j = floor (i/M), k = modulo (i/M), and M and N are the column and row lengths, respectively, of an $M \times N$ image.

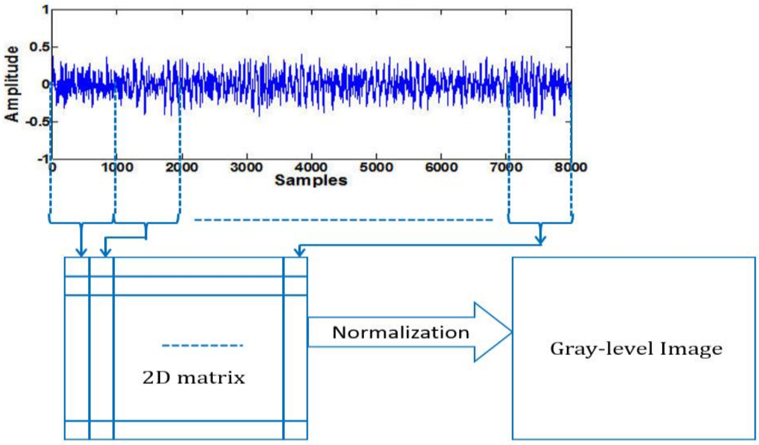


Figure 3. Conversion from a 1D Time-domain Signal into a 2D Gray-level Image

Figure 4 illustrates examples of resultant texture images of different fault signals.

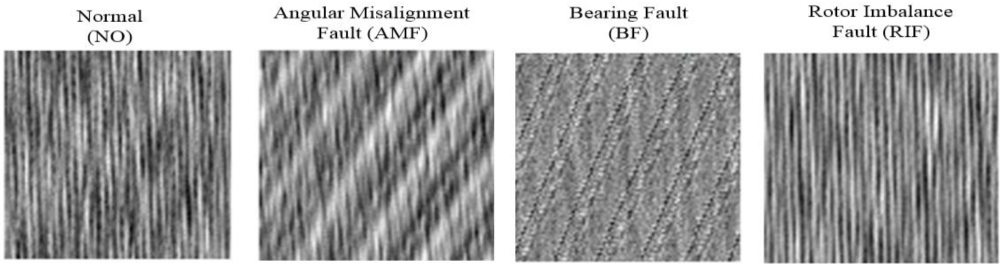


Figure 4. Converted Texture Images of Different Fault Signals

To construct feature vectors, we calculate LNS maps of the central pixels of a gray image. In the LNS map, three factors are considered: searching window, neighborhood window, and the gap between central pixels [8, 11]. The size of the searching and neighborhood windows affects the accuracy of the construction of a neighbor structure map in describing the global homogeneity information. Table 2 presents parameters related to the GNS map used in this study.

Table 2. Statistical Parameters Used in this Study

Parameters	Values
Searching window	21×21
Neighborhood window	13×13
Image size	89×89
number of central pixels	144
Size of DNS maps	21×21
Size of GNS map	21×21
Number of concentric circles	10
Dimensions of the feature vector	216 (=8+16+8×24)

Detailed construction procedures of an LNS map and a GNS map are shown in Figures 5 and 6, respectively. After generating the LNS maps of all central pixels in an image, a GNS map is constructed by averaging the LNS maps. Figure 7 shows examples of the LNS maps of different fault signals.

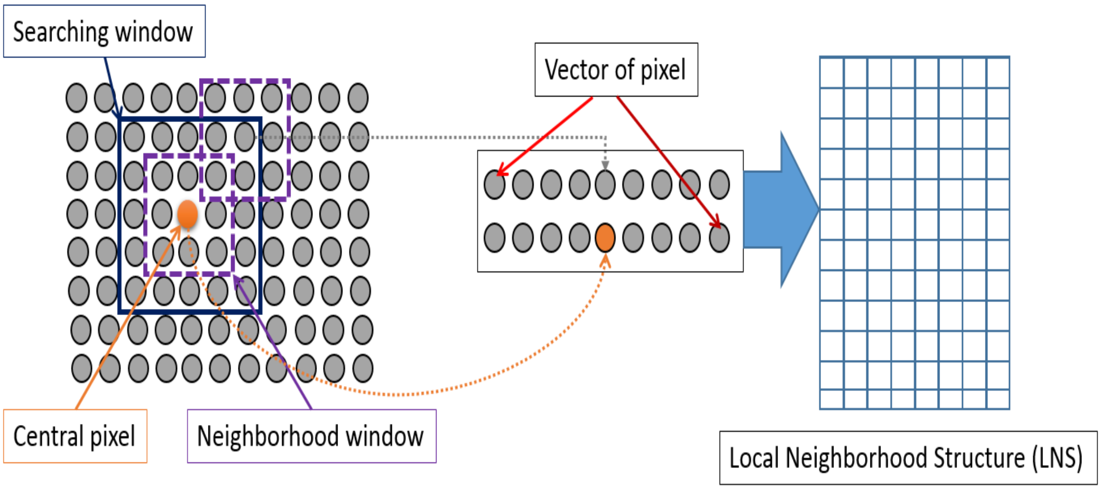


Figure 5. Construction Procedure of an LNS Map

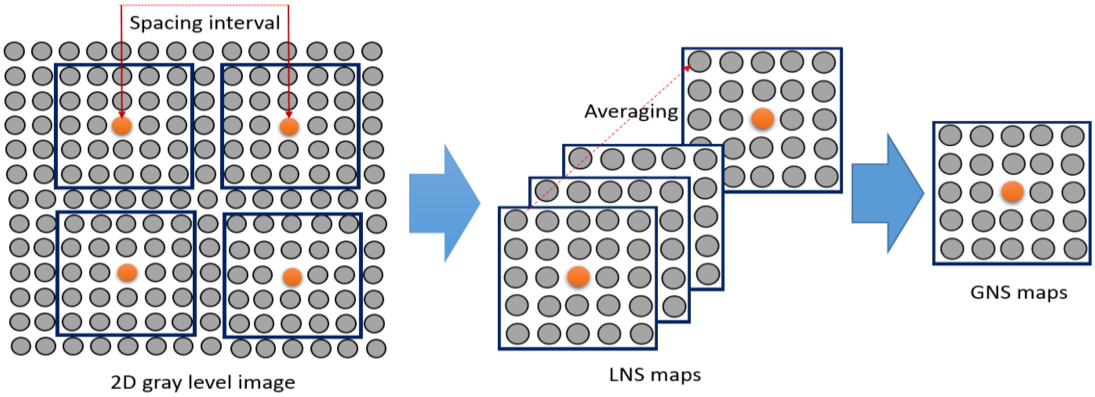


Figure 6. Construction Procedure of a GNS Map

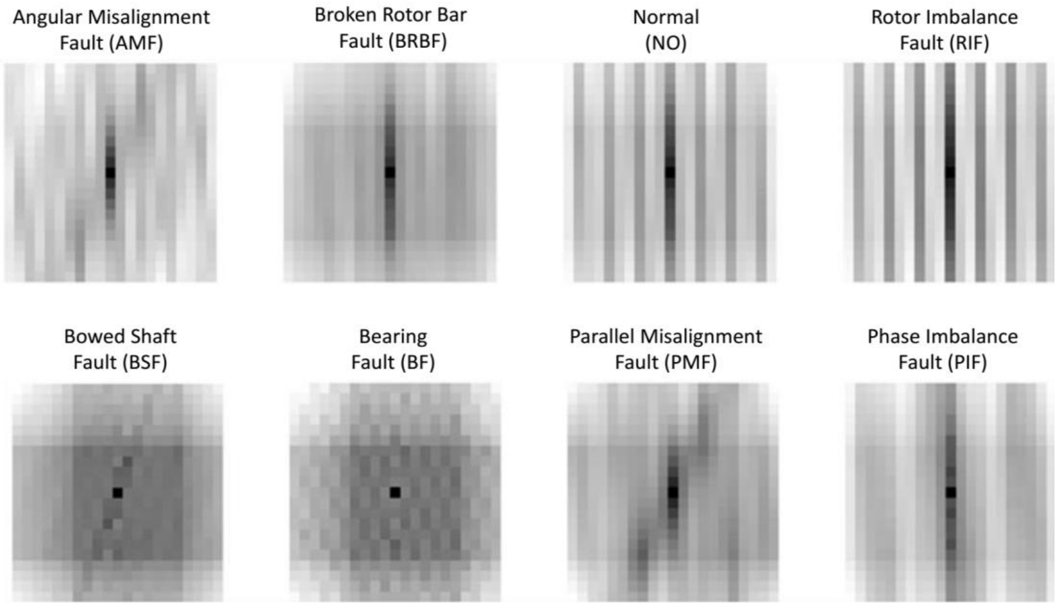

Angular Misalignment Fault (AMF)
Broken Rotor Bar Fault (BRBF)
Normal (NO)
Rotor Imbalance Fault (RIF)
Bowed Shaft Fault (BSF)
Bearing Fault (BF)
Parallel Misalignment Fault (PMF)
Phase Imbalance Fault (PIF)

Figure 7. LNS Maps of Different Types of Faults [8]

A well-known multivariate statistical PCA algorithm [12] is then used to extract the most distinctive features by eliminating the redundant features, resulting in improved classification accuracy. Figure 8 shows an example of dimension reduction of feature vectors using the PCA algorithm.

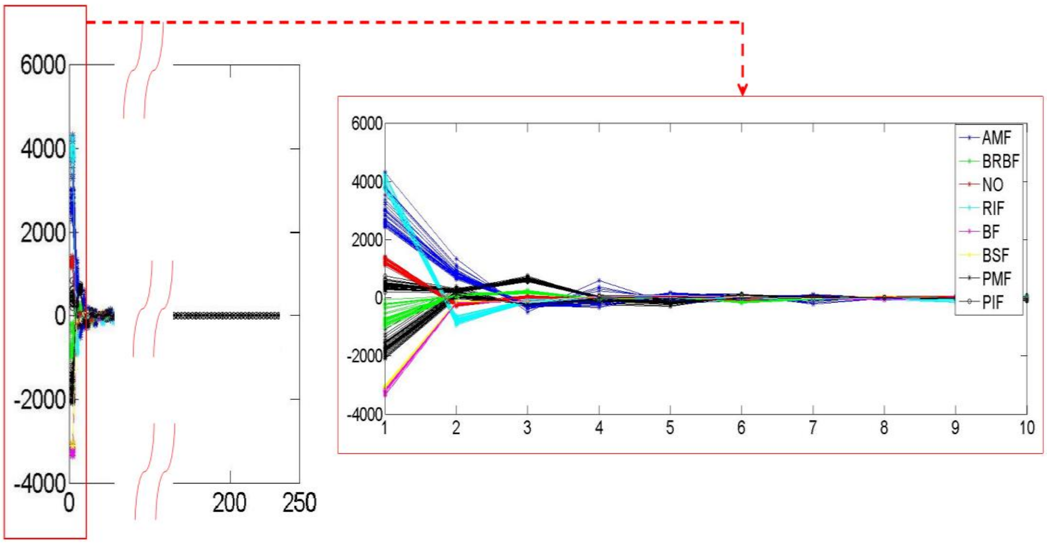


Figure 8. Dimension Reduction of Feature Vectors Using the PCA Algorithm

Finally, a one-against-all multi-class support vector machine (OAA-MCSVM) classifier is used to identify each fault by using the most significant features as inputs of the OAA-MCSVM, as shown in Figure 9. In the OAA-MCSVM approach, each SVM structure discriminates one class from the others, and the final decision is made by selecting the SVM structure that yields the highest output value. To support the nonlinear characteristics of the vibration signals, we utilize a Gaussian radial basis function (RBF) kernel in each SVM. The selection of the sigma value of the Gaussian RBF is a significant parameter because it greatly affects the classification accuracy of the SVMs.

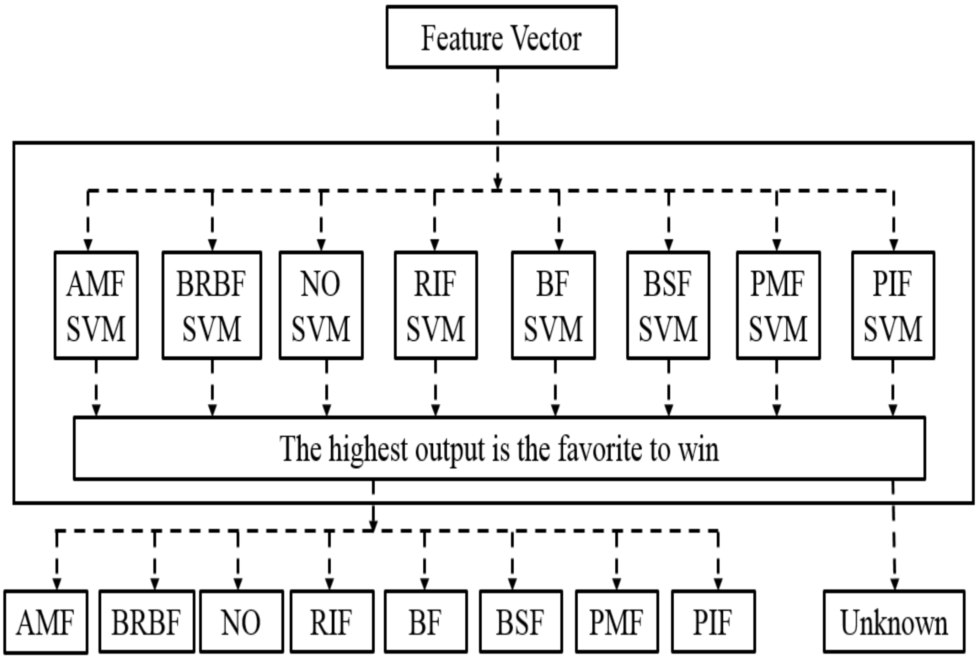


Figure 9. A Multi-class SVM Structure

In this study, we obtain the maximum classification accuracy using numbers of PCA dimensions that vary from 3 to 6 and select the three most significant dimensions to form a feature vector. The experimental results demonstrate that the GNS-based 2D fault diagnosis method achieves 100% classification accuracy by exhibiting 100% true positive (TP) and 0% fault positive (FP) when using the three most significant dimensions, where TP is the number of faults in class *i* that are correctly classified into class *i*, while FP is the number of faults in the other classes that are incorrectly classified into class *i* [13]. Table 3 presents an optimal range of σ values for each SVM to classify each fault of an induction motor. The value of σ varies from 0.1 to 2 with an interval of 0.1.

Table 3. Optimal Sigma Values of Each SVM structure

Faults	Optimal range	Selected values
AMF (SVM1: class 1)	$0.2 < \sigma < 2.0$	1.0
BRBF (SVM2: class 2)	$0.1 < \sigma < 2.0$	0.9
NO (SVM3: class 3)	$0.1 < \sigma < 1.4$	0.6
RIF (SVM4: class 4)	$0.1 < \sigma < 1.9$	1.0
BF (SVM5: class 5)	$0.1 < \sigma < 0.3$	0.2
BSF (SVM6: class 6)	$0.1 < \sigma < 0.4$	0.2
PMF (SVM7: class 7)	$0.1 < \sigma < 1.2$	0.6
PIF (SVM8: class 8)	$0.1 < \sigma < 1.1$	0.5

4. GPU Map-Based Implementation of the 2D Fault Diagnosis Method

In this paper, we implement the GNS map-based 2D fault-diagnosis method on an NVIDIA GeForce GTX 580 GPU [14-16] to accelerate the 2D fault diagnosis method, where the GPU executes all tasks in parallel. The detailed hardware configurations of the GTX GPU are presented in Table 4.

Table 4. Hardware Specifications of the GPU

Parameter	Value
Device	GeForce GTX 580
Total amount of global memory	3072 Mbytes
Total number of CUDA Cores	512
GPU clock speed	1.54 GHz
Memory clock rate	2004.00 MHz
Memory bus width	384 bit
Warp size	32
Maximum number of threads per block	1024
Maximum sizes of the dimensions of a block	1024×1024×1024
Maximum sizes of the dimensions of a grid	65535×65535×65535

Figure 10 shows a detailed GPU-based implementation of the 2D fault-diagnosis method. GPU executes the major operations of the 2D fault-diagnosis algorithm, including LNS and GNS maps generation, extraction of dominant dimensions using PCA, and fault classification using MCSVM. A main function of the CUDA C code for the GPU-based implementation is illustrated in Figure 11.

Initially, a CPU transfers the converted texture images along with the central pixels to an NVIDIA GTX GPU by allocating memory. Then, the GPU performs the 2D fault diagnosis kernels in parallel using a number of threads in the GPU [17].

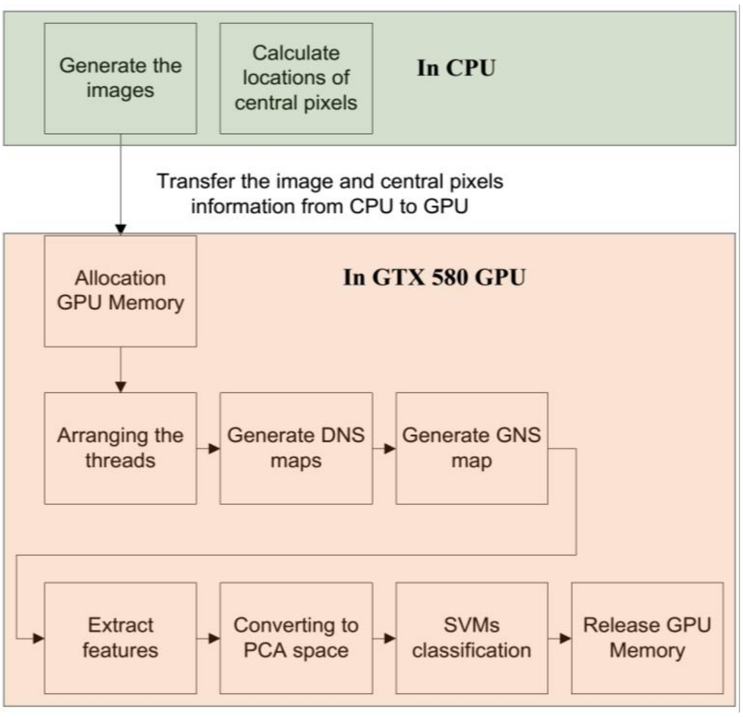


Figure 10. GPU Implementation of the 2D Fault-diagnosis Method

```
main (){
//generating an image
//calculate the central pixels
//allocate memory in GPU
cuErr = cudaMalloc(&D_Msignal.width, size);
cuErr = cudaMalloc(&D_Img.width, size);
cuErr = cudaMalloc(&D_CenPixList.length, size);
//transfer the image and central pixels information to GPU
ccuErr = cudaMemcpy (D_Msignal.width, Msignal.width, size,
cudaMemcpyHostToDevice);
uErr = cudaMemcpy(D_CenPixList.length,CenPix
List.length,size,cudaMemcpyHostToDevice);
//preparing memory in GPU for DNS & GNS
cuErr = cudaMalloc(&D_DNSData.width, size);
cuErr = cudaMemcpy (D_DNSData.width, DNSDa ta.width, size,
cudaMemcpyHostToDevice);
GDNS_Generation<<<numBlock,Blocksize>>>(D_DNSData, D_ParList, d_test,
d_testInt);
//extracting feature vectors
F_Extraction<<<numBlock,Blocksize>>>(D_FeatureVector, D_DNSData, D_ParList,
d_test, d_testInt);
//converting to PCA
PCATransform<<<numBlock,Blocksize>>>(D_FVinPCA, D_TransFormMat,
D_FeatureVector, d_test, d_testInt);
//applying SVMs
//transfer the result from GPU to CPU
cuErr = cudaMemcpy (FVinPCA.length, D_FVinPCA.length, (3+1)*sizeof(float),
cudaMemcpyDeviceToHost);
//CUDA memory free
}
```

Figure 11. A Main Function of the CUDA C Code for the GPU-based Implementation

In the GPU-based implementation, thread organization is a vital issue to achieve high performance. Table 5 provides the thread organization of the block and grid sizes for each kernel of the 2D fault-diagnosis method, and Figure 12 shows an example of the thread organization for the LNS map with 144 central pixels, where 144 threads concurrently run in different blocks.

Table 5. Thread Organizations

Kernels	Block Size	Number of blocks

LNS map
(12×12=144 central pixels)
	Blocksize.x=2
	Blocksize.y=2
		numBlock.x=6
		numBlock.y=6

GNS map
(21×21 size)
	Blocksize.x=2
	Blocksize.y=2
		numBlock.x=11
		numBlock.y=11

Feature vector extraction
(10 centric circle)
	Blocksize.x=2
	Blocksize.y=2
		numBlock.x=3
		numBlock.y=1

PCA
(3 dimensions)
	Blocksize.x=2
	Blocksize.y=2
		numBlock.x=1
		numBlock.y=1

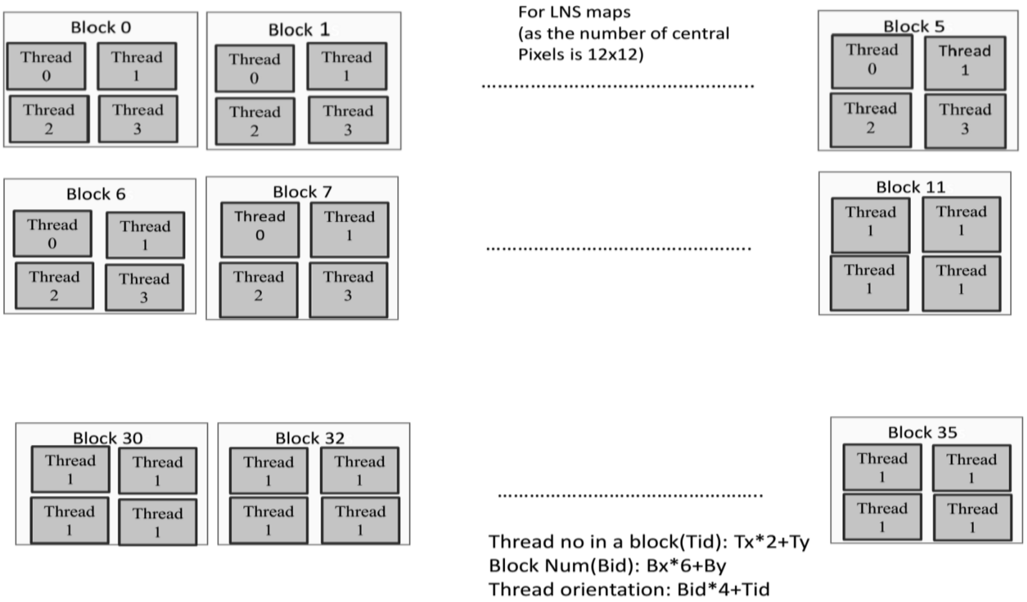


Figure 12. An Example of Thread Organization for the LNS Map Kernel

Table 6 provides the execution times of the CPU- and GPU-based approaches for three major kernels: GNS map generation, feature extraction, and fault classification. The difference between the CPU- and GPU-based approaches is that the several threads in the GPU execute these kernels in parallel, whereas the execution pattern of the CPU implementation is serial. Therefore, the CPU implementation requires much more execution time than the GPU-based approach. The experimental results show that the most time-consuming kernel is the GNS map process. As expected, the greatest speedup of the GPU is obtained in this kernel due to its tremendous reduction of data processing operations, as shown in Table 6. Specifically, the GPU achieves a 119.7× speed increase over the CPU for the GNS map process. In addition, since the fault classification kernel using the OAA-MCSVM requires sequential operations, the GPU-based approach achieves less speedup over the CPU implementation. Overall, the GPU-based approach achieves about 118.5× faster operation than the equivalent sequential CPU implementation while maintaining 100% classification accuracy.

Table 6. Execution Time (ms) of the GPU- and CPU-based Implementations

Parameters	CPU	GPU	Speedup
GNS map generation	7180	60	119.7
Feature extraction	6.7	0.067	100
Fault classification	16.70	0.7	23.9
Total	7203.4	60.767	118.5

5. Conclusions

This paper presented a computationally efficient GPU-based implementation of the GNS-based 2D fault-diagnosis method of an induction motor. In the 2D fault-diagnosis method, a 2D representation of vibration signals was used to extract texture features from the converted image. Then, a GNS map was used to extract feature vectors by averaging the LNS maps of all central pixels. In addition, the PCA algorithm was employed to eliminate insignificant features. Finally, an MCSVM was used to identify each fault of the induction motor. The previous 2D fault-diagnosis methodology showed 100% classification accuracy, but required high computational time. Thus, this paper implemented the 2D fault diagnosis method on an NVIDIA GeForce GTX 580 GPU to accelerate the method, where the tasks were executed in parallel using a number of threads. The experimental results show that the GPU-based approach achieves 118.5× speedup over the equivalent CPU implementation.

Acknowledgements

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2012R1A1A2043644), and was supported by a National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2013R1A2A2A05004566).

References

[1] K. B. Aravindh, G. Saranya, S. R. Selvakumar, M. Saranya and E. P. Sumesh, "Fault detection in induction motor using WPT and multiple SVM", International Journal of Control and Automation, vol. 3, no. 2, **(2010)**, pp. 9-20.

[2] P. E. William and M. W. Hoffman, "Identification of Bearing Faults Using Time Domain Zero-Crossing", Mechanical Systems and Signal Processing, vol. 25, no. 8, **(2011)**, pp. 3078-3088.

[3] W. Li and C. K. Mechefske, "Detection of induction motor faults: A comparison of stator current, vibration and acoustic method", Journal of Vibration and Control, vol. 12, no. 2, **(2006)**, pp. 165-188.

[4] X. Liu, X. Wu and C. Liu, "A comparison of acoustic emission and vibration on bearing fault detection", Proceeding of the International Conference on Transportation, Mechanical, and Electrical Engineering, **(2011)**; Changchun, China.

[5] J. Treetrong, "Fault detection of electric motors based on frequency and time frequency analysis using extended DFT", International Journal of Control and Automation, vol. 4, no. 1, **(2011)**, pp. 49-58.

[6] G. F. Bin, J. J. Gao, X. J. Li and B. S. Dhillon, "Early fault diagnosis of rotating machinery based on wavelet packets-Empirical mode decomposition feature extraction and neural network", Journal of Mechanical Systems and Signal Processing, vol. 27, **(2012)**, pp. 696-711.

[7] H. Li, H. Zheng and L. Tang, "Gear Fault Detection Based on Teager-Huang Transform", International Journal of Rotating Machinery, vol. 502064, **(2010)**, pp. 1-9.

[8] J. Uddin, M. Kang, D. V. Nguyen and J. M Kim, "Reliable Fault Classification of Induction Motors Using Texture Feature Extraction and a Multiclass Support Vector Machine", Mathematical Problems in Engineering, Hindawi Corp., vol. 2014, **(2014)**, pp. 1-9.

[9] D. Nguyen, M. Kang, C. Kim and J. Kim, "Highly Reliable State Monitoring System for Induction Motors Using Dominant Features in a 2-Dimension Vibration Signal", New Review of Hypermedia and Multimedia, vol. 13, no. 3-4, **(2013)**, pp. 245-258.

[10] J. Uddin, R. Islam and J. Kim, "Texture Feature Extraction Techniques for Fault Diagnosis of Induction Motors", Journal of Convergence, vol. 5, no. 2, **(2014)**, pp. 15-20.

[11] F. M. Khellah, "Texture Classification Using Dominant Neighborhood Structure", IEEE Transaction on Image Processing, vol. 20, no. 11, **(2011)**, pp. 3270-3279.

[12] A. Herve and J. W. Lynne "Principal component analysis", Wiley Interdisciplinary Review: Computational Statistics, vol. 2, no. 4, **(2010)**, pp. 433-459.

[13] T. Dech, D. Dursun, M. Phayung and K. Nihat, "A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition", Journal of Expert Systems with Applications, vol. 41, **(2014)**, pp. 321-330.

[14] J. Uddin, E. Oyekanly, C. H. Kim and J. M. Kim, "High Performance Computing for Large Graphs of Internet Applications using GPU", International Journal of Multimedia and Ubiquitous Engineering, vol. 9, no. 3, **(2014)**, pp. 269-280.

[15] I. K. Jung, J. Uddin, M. Kang, C. H. Kim and J. M. Kim, "Accelerating a Bellman-Ford Routing Algorithm Using GPU", Lecture Notes in Electrical Engineering (Springer), vol. 301, **(2014)**, pp. 153-160.

[16] J. Uddin, I. K. Jung, M. Kang, C. H. Kim and J. M Kim, "Accelerating IP routing algorithm using graphics processing unit for high speed multimedia communication", Multimedia Tools and Applications, **(2014)**, pp. 1-15.

[17] K. David and H. Wen-mei, "Programming Massively Parallel Processors-A hands on Approach", Editor, Elsevier, 2nd Edition, **(2012)**.

Authors

Jia Uddin received a BS degree in computer and communication engineering from International Islamic University, Chittagong (IIUC), Bangladesh, in 2005, and an MS degree in electrical engineering with an emphasis on telecommunications from Blekinge Institute of Technology (BTH), Sweden, in 2010. Currently, he is pursuing a PhD in computer engineering at the University of Ulsan (UoU), South Korea. He is an assistant professor (currently on study leave) in the Science and Engineering Department at IIUC, Bangladesh. His research interests include fault diagnosis, parallel computing, and wireless networks. He is a member of the IEB and the IACSIT.

Dinh V. Nguyen received a BS degree in electrical engineering from Hanoi University of Science and Technology, Vietnam, in 2010 and an MS degree in electrical, electronics, and computer engineering from the University of Ulsan, South Korea, in 2013. He is a lecturer of electrical engineering at Hanoi University of Science and Technology, Vietnam. His current research interests include signal processing, machine learning, and pattern recognition.

Jong-Myon Kim received a BS in electrical engineering from Myongji University, Yongin, Korea, in 1995, an MS in electrical and computer engineering from the University of Florida, Gainesville, in 2000, and a PhD in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, in 2005. He is an associate professor of Electrical Engineering at the University of Ulsan, Korea. His research interests include multimedia processing, multimedia-specific processor architecture, parallel processing, and embedded systems. He is a member of IEEE and the IEEE Computer Society.