ANNSVM: A Novel Method for Graph-Type Classification by Utilization of Fourier Transformation, Wavelet Transformation, and Hough Transformation

Sarunya Kanjanawattana
Graduate School, Shibaura Institute of Technology, Tokyo, Japan
nb14503@shibaura-it.ac.jp

Masaomi Kimura
Information Science and Engineering, Shibaura Institute of Technology, Tokyo, Japan
masaomi@sic.shibaura-it.ac.jp

Abstract

Image classification plays a vital role in many areas of study, such as data mining and image processing; however, serious problems collectively referred to as the course of dimensionality have been encountered in previous studies as factors that reduce system performance. Furthermore, we also confront the problem of different graph characteristics even if graphs belong to same types. In this study, we propose a novel method of graph-type classification. Using our approach, we open up a new solution of high-dimensional images and address problems of different characteristics by converting graph images to one dimension with a discrete Fourier transformation and creating numeric datasets using wavelet and Hough transformations. Moreover, we introduce a new classifier, which is a combination between artificial neuron networks (ANNs) and support vector machines (SVMs), which we call ANNSVM, to enhance accuracy. The objectives of our study are to propose an effective graph-type classification method that includes finding a new data representative used for classification instead of two-dimensional images and to investigate what features make our data separable. To evaluate the method of our study, we conducted five experiments with different methods and datasets. The input dataset we focused on was a numeric dataset containing wavelet coefficients and outputs of a Hough transformation. From our experimental results, we observed that the highest accuracy was provided using our method with Coiflet 1, which achieved a 0.91 accuracy.

Keywords: Graph-type classification, wavelet transformation, dimensionality reduction, artificial neural networks, support vector machines, discrete Fourier transformation,

1. Introduction

In past decades, there has been a growing interest in image classification. In much of literature, researchers have encountered problems with image classification and have attempted to find solutions by utilizing a variety of classification techniques, such as support vector machines (SVMs) (Barla, Odone & Verri, 2003) (Chapelle, Haffner & Vapnik, 1999) and artificial neural networks (ANNs) (Frate, Pacifici, Schiavon & Solimini, 2007) (Veluchamy, Perumal & Ponuchamy, 2012), as well as including image analysis techniques, such as wavelet transformation. Typically, a large input dataset size is a serious problem in the study of image classification because images generally contain many features and attributes, in particular, two-dimensional images. In fact, the performance of a classification system is inverse to the size of the input dataset. Many existing studies have developed image classification methods based on large-scale datasets, attempting to somehow mitigate the problem of high-dimensional data, as two-dimensional images (Sanchez & Perronnin, 2011). If we use the two-dimensional images in an inappropriate way, this may significantly decrease the classification performance. Therefore, we need to propose a plausible solution to effectively solve this problem.

An initial input used in our study involves a collection of graph images collected from the academic literature. A graph is a graphical representation of a set of objects, and there are a variety of graph types, such as line graphs, plots and pie charts. Such graphs map dependent or independent quantitative variables and represent the essential content that summarizes the given data. An initial

form of our data collection is also two-dimensional image; however, we do not use it directly to our method but convert it to more suitable and usable form, which should enhance the quality of our classification.

Low-level features (e.g., color, texture, etc.) are extensively used to classify and analyze generic images (Arivazhagan, Ganesan & Padam Priyal, 2006) (Park, Lee & Kim, 2004). Notwithstanding, these are not essential properties for categorizing graph types, yet other objects, such as lines, plots, and primitive shapes. Furthermore, a crucial problem we focus on in this study is differences in graph characteristics. Naturally, the same type of graph may include several different objects and characteristics. For example, there are many points in a scatter plot (e.g., see Figure 1a), but the positions of these points are certainly different from one scatter plot to another (e.g., compare to Figure 1b) depending on the real data. Moreover, there is also a line in the scatter plot shown in Figure 1b. Unfortunately, these uncertain characteristics of graphs cause difficulties for traditional classification systems (Kanjanawattana & Kimura, 2015) (Anthimopoulos, Marios M and Gianola, Lauro and Scarnato, Luca and Diem, Peter and Mougiakakou & Stavroula, 2014). Assuming that, we use convolution neural network (CNNs) to classify the example images showing in Figure 1. Based on a basic process on CNN, it convolves the images using multiple filters and feature maps. A typical random kernel is similar to an edge detector; hence, the edge is an important feature for CNN. Moreover, a result of convolution process provides an output matrix containing edge characteristics. However, it is the inessential property for graph-type classification, yet dominant objects, such as lines and circles. We realize that CNN may be unsuitable for graph images. Regarding our method, we also convolve the graph images from two-dimension to one-dimension, still include significant characteristics, such as a profile of pixel appearance and the focused objects in the graphs. We emphasize to classify the images based on their essential characteristics rather than image features.

This system classifies graph types based on the presence of dominant objects in images. For example, it checks a presence of plots in data space to classify plot graph but does not focus on their plots' positions or direction of correlation because we do not currently consider data interpretation, but only classify the types. Concisely, we address the problem of classification with images containing particular characteristics, even if they are grouped in the same category. Further, we realize a difficulty of the course of dimensionality that always occurs when working with image data. Thus, to minimize the problems, we must establish a productive classification method to create a new data representative that replaces the two-dimensional image that can handle problems of dimensionality and different characteristics.

The main focus of our study is therefore devoted to a graph classification system that classifies graphs into their types based on dominant different characteristics that the graph contains.

Our method involves the image convolution that transforms a two-dimensional image into a one-dimensional image while retaining necessary information, including constructed numeric datasets that represent results of wavelet and Hough transformations. We also propose a new classifier called ANNSVM which is a combination of ANNs and SVMs. This system aims to provide benefits to researchers who give interests to other's studies and need to prove validation of their studies by comparing their results with other related studies, demonstrating that they acquire a bar graph illustrating the accuracy of each software from their experiments; thus, to investigate other software and their accuracies in other studies, they need to specify a graph type as a bar graph which is the same as theirs for effective comparison. A finding in our previous study (Kanjanawattana & Kimura, 2015) already proved that significant knowledge can be discovered in graph images, such as a relationship between axis titles. In fact, different types of the graphs also offer particular knowledge based on their dominant graphical characteristics. Therefore, a process to distinguish graph type can enhance a capability of image information extraction system (Kanjanawattana &

Kimura, 2016) and acquire wide and specific information. For example, as the simulated case showing above, its results should be displayed in a bar graph because the bar graph can implicitly express categorized data (e.g., their accuracies corresponding to each software) much better than a line graph. On the other hand, a line graph should be a better choice when dealing with continuous data, such as vehicle speed. The graph-type classification possibly applies to a range of applications, such as search engine and image interpretation systems.

The major objectives of our study are (1) to propose a novel method for classifying graph types with greater accuracy, (2) to extract dominant characteristics of graphs to be a new data representative suitable for identifying graph types, and (3) to indicate what features make our data separable, which improves the quality of a classification system.

To evaluate our proposed method, we conducted several experiments to compare our approach with classical methods, for example, ANNs, CNNs, and SVMs. Both ANNs and SVMs are extensively recognized as being high-potential tools for classification. The important part of ANNs is their ability in learning iterations, which defines how weights of each neuron should be periodically adjusted. The SVM uses a boundary to isolate data that is different depending on a kernel. Here, the kernels tested in this study are radial basis function (RBF) and linear kernels. The CNN is a powerful tool to categorize images by analyzing image edges. The CNN is based on the ANN with one more additional step called a convolution step. Typically, the CNN provides great results as shown in previous studies (Lawrence, Giles, Tsoi & Back, 1997), particularly when working with two-dimensional images containing low-level image features, e.g., image edges, as dominant characteristics.

In addition to this introductory section, the remainder of this paper is organized as follows. In Section 2, we present previous work related to our present study. In Section 3, we describe details regarding the methodology used in this study. In Section 4, we describe our experiments and results, then discuss our findings. Finally, we summarize the key content of our study in Section 5.

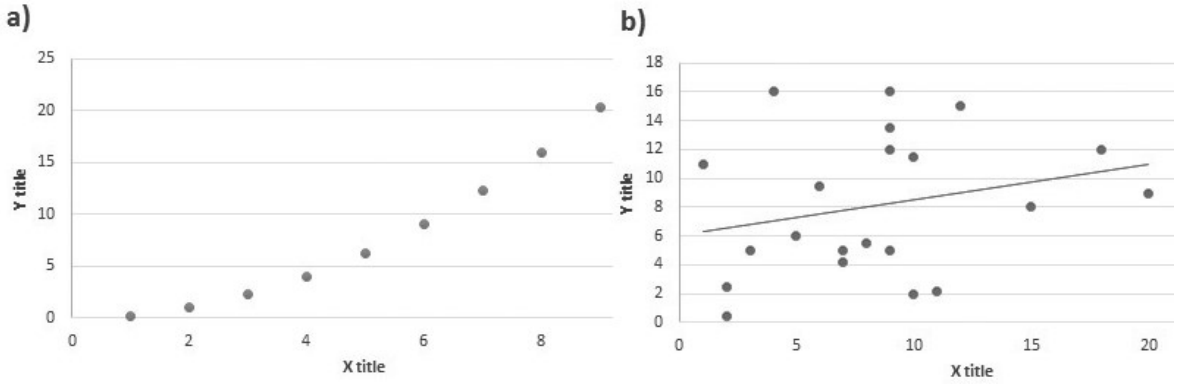


Figure 1. Example displaying two scatter plots with different characteristics and patterns: (a) scatter plot containing only points and (b) scatter plot containing points in different positions than (a) and a line

2. Related works

An image is a useful resource that presents meaningful information using the graphical content. Humans can easily grasp the semantical information implied in images without necessarily reading any context appearing in related documents. Image classification has made great progress in many different areas, including medical analysis (Ibrahim, Osman & Mohamed, 2013) (Klöppel, Stonnington, Chu, Draganski, Scahill, Rohrer, & Frackowiak, 2008) and academia (Cai, Wang, Sun & Chen, 2003) (Xia, Du, He & Chanussot, 2014).

A major problem of image classification is the course of high-dimensional images. A number of previous studies have developed several potential methods for overcoming this difficulty (Aggarwal & Yu, 2001) (Song & Wang, 2013). Sanchez et al. (Sanchez & Perronnin, 2011) proposed an image classification on large-scale images including multiple classes. They address the compression of such high-dimensional signatures with two lossy compression schemes: dimensionality reduction based on hash kernels and data encoding with product quantizers. We realized that the image analysis process was an important procedure for the classification because images consist of raw data in an inappropriate input form for the existing classification systems. Thus, a preprocessing step used to prepare and construct input data in a ready form became an important procedure in extracting dominant image features. Fan et al. (Fan, Shen & Davatzikos, 2005) presented a method for classifying medical images based on their regions by using a nonlinear SVM. They built adaptive regional feature extraction and feature selection procedures that consolidate robust features from high-dimensional morphological measurements obtained from brain magnetic resonance (MR) images. Their robust feature selection removed irrelevant and redundant features to improve classification.

Considerable attention has been paid to extending the ideas of extracting image features. A popular method was to extract low-level features instead of using their actual pixel values. Low-level features, such as color, texture and shape have been reported in much of the literature during the past few decades (Cheng & Chen, 2003) (Vailaya, Figueiredo, Jain & Zhang, 1999) (Veluchamy, Perumal & Ponuchamy, 2012). The visual features were extracted and used as inputs of classification. Sergyan (Sergyan, 2008) proposed a color histogram based classification approach and contributed to an image search system; however, the traditional methods involved the low-level features, but are insufficient for our target images, even if the idea of using low-level features was effective in the previous studies, because to classify the graph types, we did not focus on the low-level features, but on the presence of objects in the graphs, such as rectangles and scatter plots. They were adequate to classify the graph types.

The Hough transformation (Duda & Hart, 1972) was a famous feature extraction technique used to detect objects, such as circles, rectangles, and etc., from images. Kwon et al. (Kwon, 1994) presented a method for classifying human ages based on facial images. They performed a Hough transform to find a parabolic curve in the human chin. Wavelet transformation was also often employed for image features analysis because it can potentially handle images with different scales and that contain noises (Sarlashkar, Bodruzzaman & Malkani, 1998). In studies of image classification, wavelet analysis has been applied to images and the distribution of wavelet coefficients is considered to characterize images. Arivazhagan et al. (Arivazhagan & Ganesan, 2003) proposed a texture classification system using wavelet analysis on a set of texture images that extracts statistical values, such as mean and standard deviation.

To advance image classification, data mining and machine learning algorithms have been gaining importance in recent years. As such, there are some well-known algorithms usually applied to image classification given their simplicity and performance characteristics; these algorithms include ANNs, SVMs, and CNNs, each of which is described below.

The ANN is an information-processing paradigm inspired by human neural networks. Veluchamy et al. (Veluchamy, Perumal & Ponuchamy, 2012) proposed blood cell classification prediction for normal and abnormal cell classes. They extracted images of blood cells, sorting gray level statistics and algebraic moment invariants, then classified the target images using an ANN. Frate et al. (Frate, Pacifici, Schiavon & Solimini, 2007) used an ANN to remotely sense to distinguish among areas made of artificial coverage including asphalt or buildings, and open spaces, such as bare soil or vegetation. They attempted to apply their classification system to high-

resolution images from satellites. These previous studies showed that ANNs could handle problems involving high-dimensional images and worked perfectly with nonlinear separable data.

The main concept underlying SVMs is to find a proper hyper-plane, defined by types of kernels, to separate data belonging to multiple categories. Both linear and radial basis function (RBF) kernels have been frequently applied to classification problem via SVMs. Cusano et al. (Cusano, Ciocca & Schettini, 2003) introduced an innovative image annotation tool based on the SVM for classifying image regions into one of seven classes, i.e., sky, skin, vegetation, snow, water, ground, and buildings, or as an unknown. They used image histograms as feature vectors. Another study (Fan, Shen & Davatzikos, 2005), Fan et al., proposed a method for classification of medical images, i.e., brain images, using SVM and deformation-based morphometry. Their study involved three steps: feature extraction, feature selection, and nonlinear classification. These previous studies showed that SVMs were effective when being applied to low-level features; however, it remained a question about SVMs efficiency when dealing with our data because the extracted features of this system were different from existing studies.

In recent years, CNNs have been developed by extending the classical ANN. Here, CNNs have great potential for image classification and are also faster if a computer conditionally supports a graphic processing unit (GPU) (Strigl, Kofler & Podlipnig, 2010). CNNs can be proficiently applied to large image databases (Krizhevsky, Sutskever & Hinton, 2012). For example, Kang et al. (Kang, Kumar & Doermann, 2014) presented a novel classification based on CNNs to classify document images presented in different layouts. They also employed a technique called dropout to reduce over-fitting in the fully connected layers (Srivastava, Hinton, Krizhevsky, Sutskever & Salakhutdinov, 2014); however, CNNs here had an obvious drawback. Inputs of CNNs were images comprising edges, such as building and human images because the CNN can give fully effort to analyzing the image by using Gabor filters (Mehrotra, Namuduri & Ranganathan, 1992). In this study we convolved the input image into one-dimensional images without edges or any objects.

Therefore, we needed to conduct experiments to evaluate the performance of CNNs applied to our constructed data.

Further, from our observations during the existing study, we acknowledged that traditional classification methods, such as SVM and ANN, suffered from a problem of parametric, i.e., parameters were very sensitive to change, the value of a parameter changed the results either for the better or for the worse (Lorena & De Carvalho, 2008). In this study, we solved this obstacle and obtained the best results that could be produced by our current system.

3. Methodology

3.1. Definition of our datasets

The target data in this study consisted of a collection of graphs or diagrams. Graphs are classified into three distinguishable types, i.e., bar graphs, pie charts, and two-dimensional charts (2Dchart), the latter including line graphs, plot graphs, and area graphs. We merge and set them to the two-dimensional chart because the regular graph structure from those graph types was similar.

For example, the line graph and plot graph's structure contains titles that appear on both axes, and there are lines and plots presented in a data section. Note that lines in line graph can be recognized as continuous data plots; thus, it can be likely identified as another kind of plot graph. The three focused graph types contain apparently own characteristics. For example, for a pie chart, at least one circle should be apparent and reside in the graph; further, axis titles do not appear. For a bar graph, its main structure is comprised of a title along the y-axis and categories along the x-axis. For a 2D chart, its structure contains titles that appear on both axes. Further, this system is a part of graph information extraction (Kanjanawattana & Kimura, 2016); thus, to precisely extract information, we must use particular methods to extract knowledge depending on the types that contain their own general structures, such as a presence of axis titles and dominant objects. We

realize that we can use the same method to extract information from both the line and plot graphs; whilst, another particular method is also specifically applicable to the type of bar graph. Therefore, in other words, we choose the graph types and limit them to three different types because not only they contain separable graph type characteristics, but it is also convenient for our graph information extraction (Kanjanawattana & Kimura, 2016) to accurately extract their information.

In this study, we convert the two-dimensional image dataset, which is squared to a resolution of 64 x 64, to one-dimensional images. The one-dimensional image is a constructed image with a size of approximately 1 x 64 that is applied to our method. Moreover, we create numeric datasets assembling wavelet coefficients and outputs from the Hough transformation. The main motivation for using these techniques is to reduce data dimensionality and gain only necessary information used for classifying. Our input data is two-dimension images that contain a huge volume of information; thus, we need to reduce the data size because only partial information is necessary for classification.

For example, the dominant objects in the graphs can be detected by using Hough transformation which is important information to classify the types. In contrast, image background, such as texts in the graphs, is an inessential part of classification; therefore, it should be ignored.
In our previous study (Kanjanawattana & Kimura, 2015), we performed experiments with these image sizes, including other bigger image sizes. As reasonable results, we found that these sizes (i.e., 64 x 64 and 1 x 64) provided the most accurate results as compared to other sizes because they contain adequate information for classifying. Image sizes that were too large were inappropriate for our experiments because of the course of high-dimensional data and the problem of sparsity. Similarly, we avoided using image sizes that were too small due to the difficulty of unclear image expression.

To create numeric datasets, we use a wavelet transformation to analyze the one-dimensional images and acquire the sequences of wavelet coefficients based on several wavelet families applied in this study, i.e., Coiflet1, Coiflet3, Coiflet5, Daubechies2, Daubechies10, Daubechies20, Haar, Symlet2, Symlet10, and Symlet20. We selected these for two reasons. First, the coefficients can provide significant characteristics for the classification. Each wavelet family has a different oscillation. If the level of the wavelet increases, the wavelet provides much better compression results (Islam, Md Rafiqul and Bulbul, Farhad and Shanta & Shewli, 2012). Second, they are used in several previous studies (Avci, 2008). Overall, they have proven to work well for image classification. All datasets used in this study are summarized below.

- 1Dimg dataset: the original one-dimensional images
- 2Dimg dataset: the converted two-dimensional images
- WL dataset: the numeric datasets that contain only results from the wavelet transformation
- HT dataset: the numeric datasets that contain only results from the Hough transformation
- WLHT dataset: the numeric datasets that contain both results from both the wavelet and Hough transformations

The main dataset we currently emphasize in this study is WLHT, i.e., the numeric datasets that include results from both the wavelet and Hough transformations. To classify images, conventional methods directly use 2Dimg, i.e., the two-dimensional images, as inputs to the systems. Hence, for evaluation, we decided to apply our method to images in 1Dimg and 2Dimg, then compare to our main dataset, WLHT. Finally, we used WL and HT to evaluate which is better for yielding separable data.

3.2. Our method

In this subsection, we propose a new method for graph-type classification using a combination of the SVM and the ANN that include several techniques, such as the discrete Fourier

transform (DFT), Hough transformation, and wavelet transformation. We divide our system into two major steps, i.e., a preprocessing step and an application of classification.

3.2.1. Preprocessing step

The preprocessing step is a crucial part of our approach. It is used to construct one-dimensional images (i.e., 1Dimg) and numeric datasets containing wavelet coefficients and Hough transformations (i.e., WLHT). To generate the one-dimensional images, the essential procedures are shown in Figure 2. Initial inputs of this process are two-dimensional graph images that have already been cleaned and converted to a gray-scale. Our method consists of four steps, each of which is described below.

First, we collect graph images as raw data, which contain different scales and sizes, and therefore need to be normalized. We clean the images by omitting irrelevant areas. For example, we omit unnecessary text that has nothing to do with our classification procedure. Moreover, to standardize the sizes and shapes of the images, we resize and reshape them to be 64 x 64 squares.

Second, we examine each image pixel, each of which contains one color value. After each pixel is projected along the x- and y-axes, we count the number of projected pixels with a color value greater than zero to reduce image dimensionality. We, therefore, obtain two one-dimensional images from the x- and y-axes.

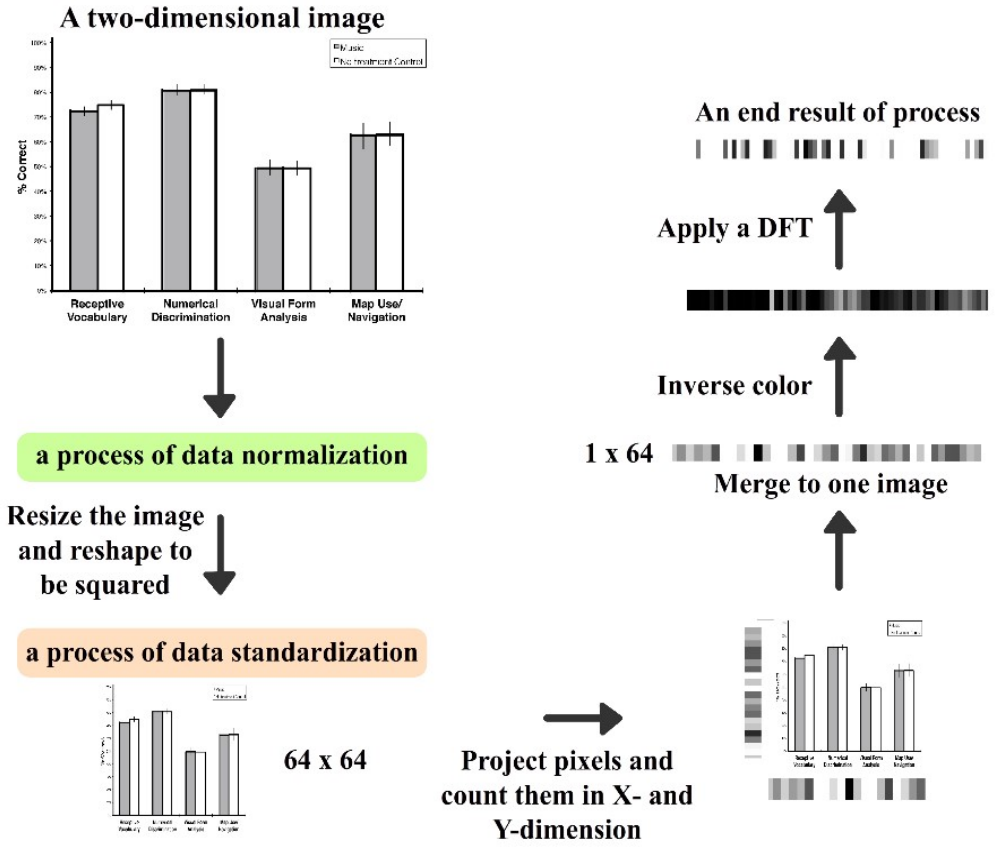


Figure 2. Illustrating the core process of one-dimensional image construction by applying a DFT

Third, the one-dimensional images acquired from the previous step are combined into one piece by concatenating the one-dimensional image of the x-axis to the one-dimensional image of the y-axis.

Finally, we apply a DFT to the obtained one-dimensional images with inverted colors. We invert the colors here because following the previous step, we gather some high values along axes, which are represented as white or light colors, that obviously present existing information. Ordinarily, white color indicates no information, while in contrast, a black or dark color instinctively represents meaningful information. Therefore, the process of inverting colors helps to clarify the data expression and prevent confusion in the interpretation.

We decided to use a DFT here for two reasons. First, the DFT can uncover dominant information from images but does not need to concern itself with the position of how the objects changed. For example, an important part of pie charts is located in the low-frequency domain because the level of pixel distribution is low. On the contrary, the high-frequency domain presents an important part of plot graphs. Second, a regular factor for classifying graph types is a similarity of characteristics in each type of graph, but our target inputs probably offer different characteristics, even though they are of the same type, because some objects represented in the graphs depend on real data, as shown in the example cases of Figure 1.

After completing the process of one-dimensional image construction, we generate numeric datasets including wavelet coefficients and results from the Hough transformation (i.e., WLHT).

The one-dimensional DFT images are presented in the form of frequencies by the procedure illustrated in Figure 2; therefore, we regard them as signals that can be analyzed via wavelet analysis, which analyzes and decomposes signals into elementary forms at different scales and positions. The prospective results of wavelet analysis here are sequences of wavelet coefficients that represent the similarity extent comparing the examined section of signals to the scaled and shifted wavelets. Note that the wavelet coefficients are calculated at every possible scale and along every position of time. Further, we use a variety of wavelet families to obtain the corresponding coefficients. We apply the wavelet transformation in our study because we obtain correlated information in signals by using wavelet families. Moreover, a sequence of wavelet coefficients is substantially divergent in different images and is considered to characterize images. Thus, we obtain the dominant patterns from various types of images based on different wavelet families.

Hough transformation is a basic technique in image processing that is used to detect features of particular shapes within target images, such as circles, lines and single plots. The basic Hough transformation identifies lines in an image, but the later Hough transformation has been extended to be able to identify arbitrary shapes, most commonly circles and rectangles. Moreover, it can deal with the scatter plots in plot graph. Detected objects are counted and collated into the object detection attributes of the given dataset. To reduce variations, we categorize the number of detectable objects into five categories, i.e., C_0, C_1, C_2, C_3, and C_4. If the number of detectable objects is between zero and five, we assign its number as C_1. If it is between six and 10, C_2 is assigned. If it is between 11 and 15, C_3 is assigned. If it is greater than 16, we assigned C_4.

Otherwise, it is set to C_0. Further, to determine the graph's containing areas, we recognize that the filled areas are often located from middle to bottom with horizontal alignment. Thus, we allocate a specific region inside the image and calculate area density, which is measured by counting the number of color pixels divided by the total number of pixels in the given region. If the density exceeds a predefined threshold the value of the area attribute is set to C_1. Conversely, if the density is lower than the threshold, the value of the area attribute is set to C_0.

Comparing 2Dimg and WLHT, the characteristics of WLHT should be more reasonable for our classification system, as opposed to ordinary images, for two key reasons. First, the sizes of images from WLHT are smaller because we construct our numeric datasets based on the one-dimensional images, thus the number of dimensions certainly decreases. Indeed, the image classification system often provides better results if working with smaller image-scaling sizes (Sanchez & Perronnin, 2011). Further, processing time should be substantially less since the number

of pixels correlates to the amount of information to be processed. Second, our data can handle problems of different graph characteristics better than ordinary images due to the benefits of DFT.

3.2.2. Application of classification

In this study, we propose a new classification algorithm that is a combination of ANNs and SVMs called ANNSVM. As shown in Figure 3, WLHT, i.e., with the one-dimensional images generated in the preprocessing step, serves as input to our classification system. The application of classification consists of two steps, each of which is described below.

First, we apply the ANN to WLHT. To obtain reasonable results from the ANN, we configured and tuned the following five ANN parameters: number of hidden layers, the number of nodes in each hidden layer, the number of nodes in the output layer, learning rate, and momentum.

Essentially, if the number of nodes in the hidden layers increases, processing time increases, and the resultant ANN will suffer from over-fitting. Conversely, too small of a number of hidden layers will cause under-fitting for the ANN. In our setting, the number of hidden layers and the number of nodes in each hidden layer were fixed at five. Concerning the learning rate and momentum settings, these impact sensitive training performances are set to optimal values obtained via a grid search technique. The number of nodes in the output layer was three because there are three different class labels (i.e., 2Dchart, bar, and pie) in our datasets.

We used the ANN here because our datasets have nonlinear separation, and the ANN is also highly applicable to nonlinear modeling. Thus the ANN with multiple hidden layers was an optimal candidate; however, since the ANN is a black box learning approach, it is difficult to interpret implicit relationships between inputs and outputs.

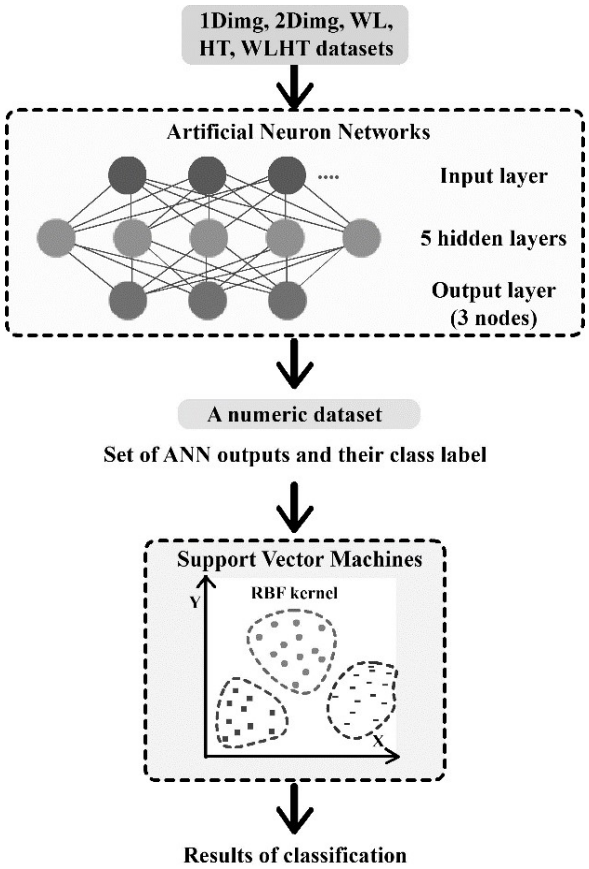


Figure 3. Demonstrating the process of classification by applying the ANN, then the SVM

After this first stage, we used three numeric outputs from the ANN as new temporary datasets. Note that the three numeric outputs here are the results from three output nodes of the output layer. Typically, these values represent a classification result by justifying a predicted class.

For our case, the system must proceed forward to the second classifier, i.e., SVM; since the system did not justify a predicted class yet but kept the numeric results for being an input of SVM. Via a training process, we combined them with a class label to obtain training datasets.

Second, we applied the SVM to the new temporary numeric dataset. The SVM uses a technique called kernels to find an optimal boundary between the training data. The tested kernel was RBF kernel. We used this nonlinear kernel because it can capture more complex relationships among data; in contrast, the training time is slightly longer.

Fortunately, since our new temporary datasets contain a small number of attributes, the speed of kernel processing was reasonably fast; however, the SVM using the RBF kernel practically encounters the hyper-parameter problem.

Significant parameters required for the RBF kernel are cost parameter and gamma. Cost parameter C determines the influence of the misclassification of each training example. If C is large, a boundary correctly classified the training example, but a margin of a boundary is smaller and not smooth. Conversely, a small C provides a smooth boundary but incorrectly classifies more examples. The gamma parameter defines how far the influence of a single training example reaches.

A larger gamma represents a close distance from the boundary to support vectors and vice versa. Further, gamma affects the shape of the boundary separating the training examples. We set these parameters by using optimal results of a grid search (Salatino & Antonio, 2014). Moreover, from a practical viewpoint, the SVM has a high algorithmic complexity that causes the testing phase to be longer.

We used the SVM classifier for three reasons. First, the SVM guarantees a global optimum solution, i.e., it can capture the lowest values from a given domain. Second, the dimensionality of the input space does not explicitly affect to computational complexity, but the smaller data size is surely outperformed. We preferred to use a smaller size of data because this system was a combination of two classifiers. Even though the problem of high dimensionality slightly affects SVM, it might cause bad results to the entire system, in both cases of speed and generation performances. Third, there is a sparse density of pixels in an image, i.e., a low density of pixels that describes information in our one-dimensional image. The SVM automatically gives a sparse solution, because the Lagrange multipliers are equal to zero for the non-support vector; therefore, the corresponding input vector can be omitted in the summation.

Primarily, no particular classifier exists for all data distributions; furthermore, we think that, if there are numerous data, only one classifier may not be discriminative well enough.

In this study, we combined ANN and SVM together because we acknowledged that individually using either SVM or ANN alone has limitations. Fusion of these two algorithms helps to enhance their abilities of classification and mitigate their drawbacks.

Based on the data used in this study, the features in input data had been assembled from various sources, such as results of wavelet transformation and results of a presence of objects provided by Hough transformation; thus, training a single classifier may provide inappropriate results. Moreover, we realized that integrating outputs from the multiple classifiers reduces a risk of classification error because the first classifier (ANN) analyzed the data and provided the results with an empirical data pattern, including some small errors.

After that, SVM took a place to operate the output of ANN, which already uncovered the data pattern, and mitigated the errors.

4. Experiments and results

4.1. Comprehensive tests

In this study, we conducted several experiments to address the various questions of this study, which are described below.

- Which method is the best solution for classifying graph types?
- Which features of the data improve the performance of our method and cause the data to be separable?
- What is the most appropriate dataset to serve as a new representative for classification?
- What significant differences of results exist in our experiments?

We divided our experiments into five major tests that include several minor tests. The CNN_1Dimg and CNN_2Dimg (i.e., Figure 4a) was designed to utilize the CNN with sets of one- and two-dimensional images (i.e., 1Dimg and 2Dimg) to compare performance between it and our main method for 1Dimg and 2Dimg. In SVM_WLHT (i.e., Figure 4b) and ANN_WLHT (i.e., Figure 4c), we applied the SVM and the ANN respectively to WLHT because both of them are popular algorithms used for image classification. We implemented a method that combined these two algorithms, i.e., the SVM and ANN approaches, in SVMANN and ANNSVM. The difference between these two experiments was the ordering of the algorithms. The SVMANN (i.e., Figure 4d) consisted of the same two steps as our proposed method, but the order of algorithms differed. The first step of SVMANN was to get the raw decision values of the SVM that presented the actual outputs from the SVM, which were used to decide which class an instance should belong to. Since we had three different classes in this study, outputs of the SVM contained three numeric values, which were inputs of the ANN. Note that we used only the RBF kernel for this method because it often provides good results when performing with complicated information, and our datasets are nonlinear data. After we combined the outputs of the SVM with a class label, we applied the ANN to the outputs and obtained classification results. The last experiment was ANNSVM_ALL (i.e., Figure 4e) in which we used our proposed method, i.e., the ANNSVM. Note that ANNSVM_ALL represents the experiments conducted by ANNSVM with all datasets used in this study. In ANNSVM_1Dimg and ANNSVM_2Dimg, the ANNSVM was applied to 1Dimg and 2Dimg to compare results with those of CNN_1Dimg and CNN_2Dimg. In ANNSVM_WL and ANNSVM_HT, we also applied the ANNSVM to WL and HT because we needed to evaluate how data affected the system. Further, the most significant experiment was ANNSVM_WLHT, in which we presented the performance of our main method applied to WLHT to indicate the effectiveness of our proposed idea. To evaluate our approach, we compared results to other experiments that also used WLHT. Regarding SVMANN_WLHT and ANNSVM_WLHT, we conducted experiments to show the difference of performance in a case that we switched their orders. A motivation for rearranging the order of algorithms was to examine whether the results had been influenced by algorithms switched.

In this study, accuracy values of each dataset showed the performance of each method. These values represent are the proportion of the total number of predictions that were correctly classified.

Initially, we classified training instances into three classes, with approximately 300 images per class. The graphs had been selectively gathered from the Web. We manually normalized the collected images by eliminating unused areas, such as unnecessary text. Moreover, we evaluated the experiments with 10 folds cross-validation because such an approach can mitigate the problem of over-fitting.

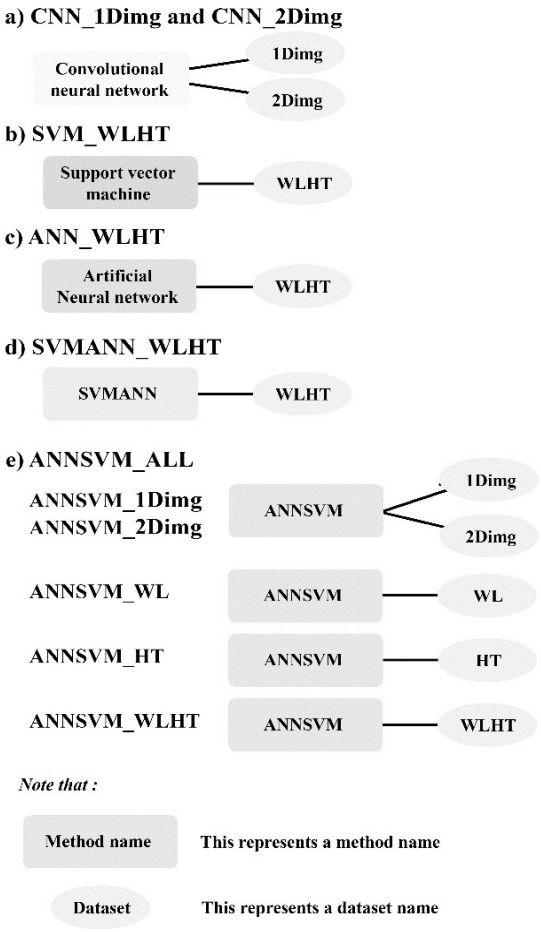


Figure 4. Processes of all experiments: (a) applying the CNN to 1Dimg and 2Dimg in CNN_1Dimg and CNN_2Dimg respectively, (b) applying the SVM to WLHT in SVM_WLHT, (c) applying the ANN to WLHT in ANN_WLHT, (d) applying the SVMANN to WLHT in SVMANN_WLHT, and (e) applying the ANNSVM to all datasets in ANNSVM_1Dimg, ANNSVM_2Dimg, ANNSVM_WL, ANNSVM_HT, and ANNSVM_WLHT

4.2. Results

We compared the results of CNN_1Dimg, CNN_2Dimg, ANNSVM_1Dimg, and ANNSVM_2Dimg to confirm the validity of ANNSVM when applied to images. The 1Dimg represented the dataset of one-dimensional images, while 2Dimg represented the dataset of two-dimensional images. Results are shown in Figure 5. The CNN_1Dimg and CNN_2Dimg provided similar accuracies, approximately 0.33, which were close to the results of ANNSVM_1Dimg and ANNSVM_2Dimg with the linear kernel, i.e., approximately 0.35; however, the experiment of our proposed method (i.e., ANNSVM with the RBF kernel) presented largely different results. In 1Dimg, the accuracy increased to 0.79. Comparing this results to those of 2Dimg applied to our proposed method, the accuracy was approximately 0.56. Thus, compared to two-dimensional images, the one-dimensional images were a better candidate for graph-type classification using ANNSVM with the RBF kernel.

a)

Dataset	CNN	ANNSVM	
	TP rate	Linear kernel	RBF kernel
		TP rate	TP rate
1Dimg	0.33	0.349	0.791
2Dimg	0.34	0.349	0.564

b)

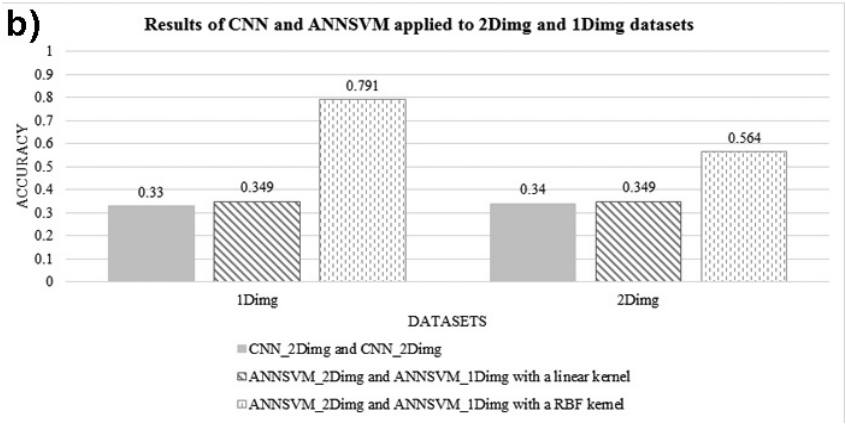


Figure 5. Results from CNN and ANNSVM that used 1Dimg and 2Dimg: (a) table statistically showing summarized results and (b) bar graph graphically illustrating results from these experiments

a)

Dataset	ANNSVM	
	Linear kernel	RBF kernel
	TP rate	TP rate
WL	0.349	0.8185
HT	0.349	0.762

b)

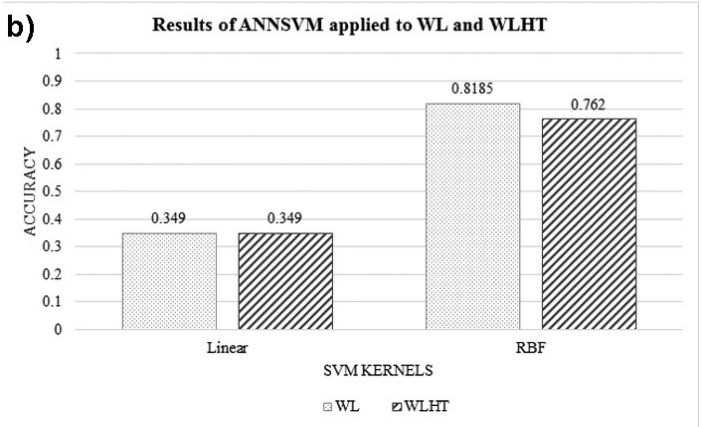


Figure 6. Results from ANNSVN that used WL and HT a) table statistically showing summarized results and (b) bar graph graphically illustrating results from these experiments

To identify which features of data influentially impacted data separability, we conducted experiments for ANNSVM with WL and HT (i.e., Figure 6). The WL contained only wavelet coefficients, whereas HT included only results of the Hough transformation. We found that, again, results obtained via the linear kernel were not significant; however, using the RBF kernel, accuracy

for WL was higher than that of HT, indicating that wavelet coefficients provide influential features that make data separable.

We conducted SVM, ANN, SVMANN, and ANNSVM with WLHT constructed from our preprocessing method. Results are shown in Figure 7.

We found that the results of SVM_WLHT showed accuracy for the RBF kernel as slightly better than that of a linear kernel as indicated in Figure 7b. They were on average 0.85 for the linear kernel and 0.86 for the RBF kernel. As for the outcome from ANN_WLHT, it was moderately 0.83.

Apparently, accuracy in SVM_WLHT which used the SVM was slightly more appropriate.

In SVMANN_WLHT, we found accuracies for WLHT were stably high, with an average of 0.9, independent of wavelet families. The highest accuracy in SVMANN_WLHT was 0.905 in the case of Symlet 2. As for ANNSVM_WLHT, the average accuracy was 0.87 for the RBF kernel (i.e., our proposed method) and 0.35 for the linear kernel. Though the average accuracy for our proposed method was slightly lower than that of the SVMANN in SVMANN_WLHT, the highest accuracy among all experiments was 0.91 in the case of ANNSVM_WLHT in which ANNSVM was applied to data obtained by Coiflet 1 (i.e., Figure 7c).

5. Discussion

Reviewing our results for the CNN applied to 1Dimg and 2Dimg, we obtained only low accuracies for both datasets. This fact disagrees with other studies but agrees with our assumption.

This situation occurred for two reasons. First, our target images were the graphs that contained different characteristics, even though they belonged to the same class. It was difficult for the traditional classification system (i.e., the CNN here) to reliably classify these data. Second, after it was converted to a one-dimensional image by our preprocessing method, it did not contain any visual image features, only the frequency domain of images obtained via DFT.

Since CNN filters commonly work to detect image edges, it is not suitable for the CNN to handle our data; however, after observing results of CNN_1Dimg and CNN_2Dimg, we discovered that they provided similar accuracies, i.e., 0.33 and 0.34, for 1Dimg and 2Dimg, respectively. Results of CNN_1Dimg and CNN_2Dimg suggested that both one-dimensional and two-dimensional images contained the same information.

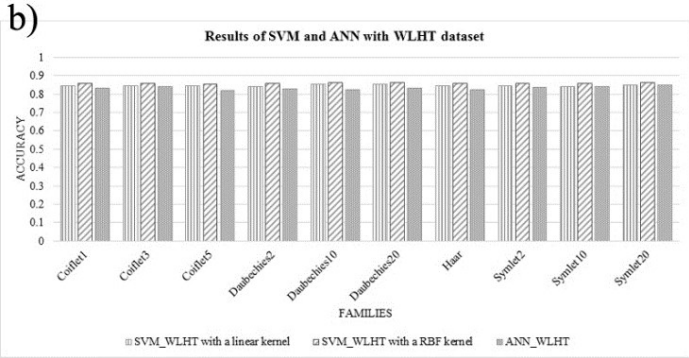
During a convolution process, we possibly obtained the similar convolved images to use in a CNN classification because the values of pixels in the one-dimensional images roughly substitute for the location of objects in the two-dimensional images. For example, if black pixels, which stand for a larger number of counted pixels, are continuously connected to a portion of a one-dimensional image, the DFT decomposes them as low frequencies.

As results of ANNSVM_1Dimg and ANNSVM_2Dimg showed, we obtained highly accurate results from our proposed method, particularly with the RBF kernel. Our datasets were not linearly separable, thus a linear kernel did not work well. Our proposed method offered substantially better results when it was applied to 1Dimg, whose dimensionality was reduced but the important information was preserved.

a)

Wavelet family	SVM_WLHT		ANN_WLHT	SVMANN_WLHT	ANNSVM_WLHT	
	Linear	RBF			Linear	RBF
	TP rate	TP rate	TP rate	TP rate	TP rate	TP rate
Coiflet1	0.847	0.86	0.833	0.902	0.349	0.914
Coiflet3	0.847	0.858	0.842	0.896	0.349	0.913
Coiflet5	0.846	0.856	0.821	0.9	0.349	0.735
Daubechies2	0.842	0.858	0.829	0.893	0.349	0.892
Daubechies10	0.852	0.862	0.823	0.903	0.349	0.906
Daubechies20	0.856	0.862	0.833	0.879	0.349	0.891
Haar	0.847	0.858	0.823	0.89	0.349	0.728
Symlet2	0.847	0.858	0.838	0.905	0.349	0.903
Symlet10	0.842	0.857	0.84	0.887	0.349	0.877
Symlet20	0.848	0.863	0.848	0.902	0.349	0.893

b)



c)

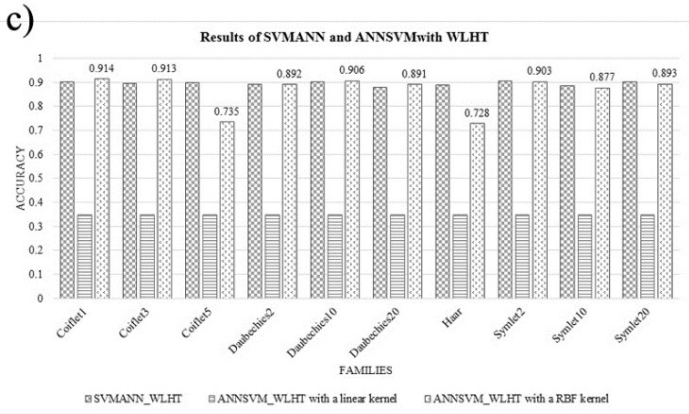


Figure 7. Results from SVM, ANN, SVMANN, and ANNSVM that used WLHT: (a) table statistically presenting summarized results, (b) bar graph graphically illustrating results from SVM_WLHT and ANN_WLHT, and (c) bar graph graphically showing results from SVMANN_WLHT and ANNSVM_WLHT

From the results of ANNSVM_WL and ANNSVM_HT, we found that wavelet coefficients had a larger impact on classification than the Hough transformation data because the results from our proposed method applied to WL were more accurate than those of HT. The wavelet coefficients can capture the dominant characteristics from the graphs better than the Hough transformation. The one-dimensional image represented in the frequency domain had oscillations with different amplitudes depending on the graph types. For example, a dominant part of a pie chart should be in the low-frequency domain, because there is a large island of concatenated pixels in a one-dimensional image, and it has only a few changes. Conversely, since the scatter plot contains many widely spread points, its dominant part should be located in the high-frequency domain. Performing the wavelet transformation, if a mother wavelet and a part of the wavelet function have a close match, the wavelet coefficient will be large. Assuming we use a suitable wavelet family with the example pie chart case, the wavelet coefficients in the low-frequency domain should be large as compared to other parts of the domain. These coefficients represented the location of objects in the

graph, including their frequencies. The Hough transformation cannot detect the position of objects or frequencies, only the shape of objects. Considering the problem of noise, the wavelet transformation can handle noise better than the Hough transformation, because the Hough transformation is sensitive to noise if the image has low quality.

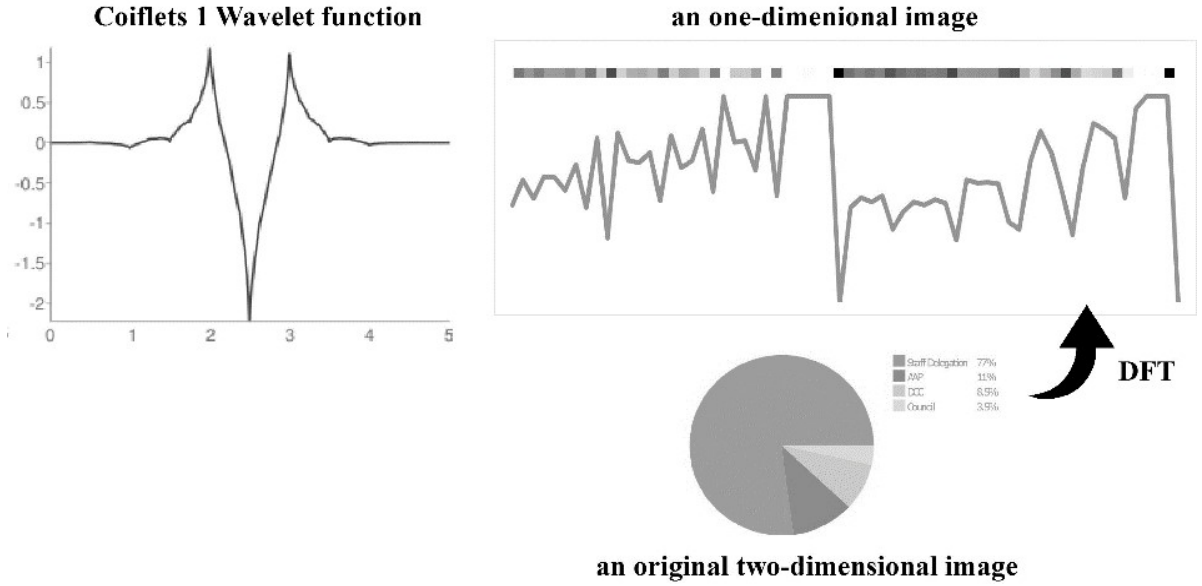


Figure 8. Simulation of Coiflet 1 (PyWavelets discussion group, 2008), analyzing as one-dimensional images

Using only the wavelet coefficients was inadequate for classification. For example, for the pie chart, we obtained large wavelet coefficients located in the low-frequency domain; however, if we changed a circle in the pie chart to other shapes, such as a radar chart, the wavelet transformation gave results that were similar to those of the original pie chart. The Hough transformation can solve this problem since it detects the shapes of objects.

We, therefore, assembled these two features in order to make our data more separable.

Comparing results from SVM and ANN applied to WLHT, we found that the SVM with the RBF kernel clearly outperformed the ANN. The difference here comes from the fact that the ANN can get stuck in local minima, while the SVM is guaranteed to find a global optimal value.

Moreover, we observed that the results of each experiment were rather similar. To confirm significant differences between them, we statistically analyzed the results using ANOVA and the T-test. We primarily performed the ANOVA to check the popularity equality, then tested via the T-test for each pair. Finally, we rejected the null hypothesis in all cases, which showed that the results of SVM_WLHT and ANN_WLHT certainly differed.

Further, we interpreted results from SVMANN_WLHT and ANNSVM_WLHT, which were the combination of the SVM and the ANN. In SVMANN_WLHT, we consistently received good accuracy values, but the highest accuracy was provided by our proposed method. To analyze results from ANNSVM_WLHT, we compared the linear and RBF kernels. Results showed that the linear kernel was not appropriate for our method because our datasets are not linearly separable, whereas the RBF kernel provided higher accuracy values. The RBF kernel generally outperforms the linear kernel because the linear kernel is suitable if the number of features is larger than the number of instances or a dataset is a very large-scale dataset; however, in general, to obtain a good model, many instances should be employed for training. In this study, WLHT contained 198 attributes and

917 instances, and the temporary datasets produced by our proposed method contained four features and 917 instances. Because of these, the linear kernel was not appropriate. Results of ANNSVM_WLHT suggested that the most suitable wavelet family was Coiflet 1 because the wavelet functions resemble the distribution of frequency in the one-dimensional images, as illustrated in Figure 8. Statistical analyses via ANOVA and the T-test showed that there was no significant difference between the results of SVMANN_WLHT and ANNSVM_WLHT with the RBF kernel. Therefore, based on this statistical evidence, we do not need to be concerned about the order of these methods. In other words, both ANNSVM and SVMANN can effectively classify graph images.

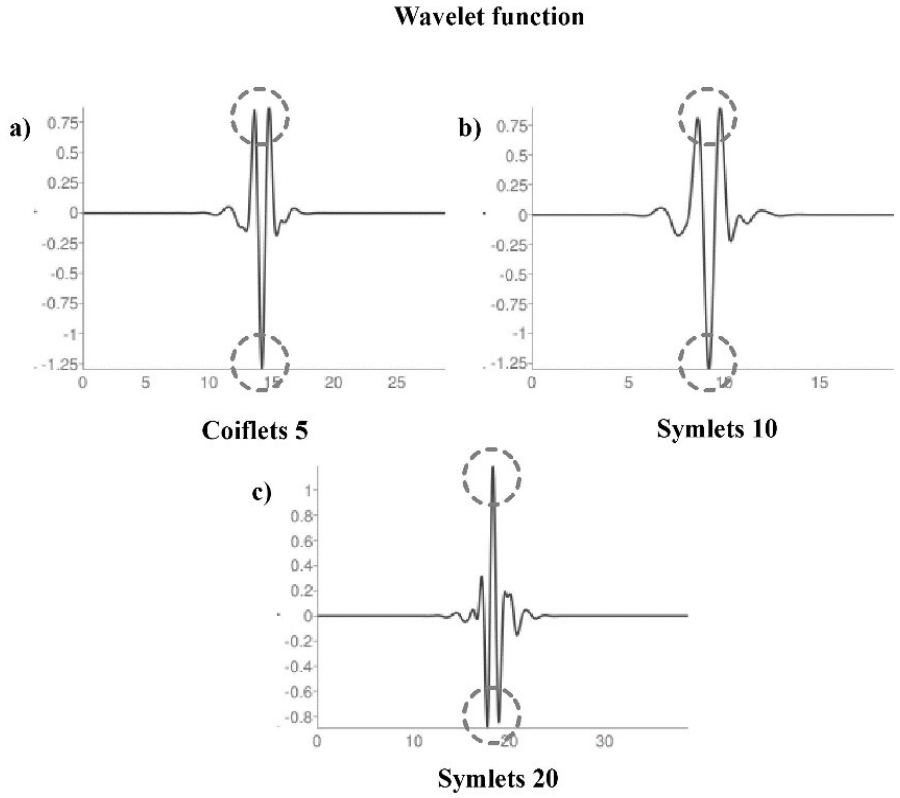


Figure 9. Illustration of three different wavelets (PyWavelets discussion group, 2008) with three waves that have high amplitude values, as indicated by the dashed red circles:
(a) mother wavelet of Coiflet 5, (b) mother wavelet of Symlet 10, and (c) mother wavelet of Symlet 20

From the results of ANNSVM_WLHT, shown in Figure 7a, the results of Coiflet 5 and Haar were considerably lower than others in the same experiment, whereas all accuracy values in SVMANN_WLHT were consistently stable. Analyzing these results, we found two possible reasons here. First, the ANN, which is the first stage of the ANNSVM, is not suitable to provide a temporary dataset that is separable by the SVM if we input data generated by these two wavelets. Second, the unsuitable mother wavelets were generated from our datasets. The mother wavelet of Coiflet 5 contained triple-high oscillation amplitude (i.e., Figure 9a). We considered that this mother wavelet was inappropriate for our data because overall our data possibly contained only a few matches with the mother wavelet of Coiflet 5. Moreover, the Symlet 10 (i.e., Figure 9b) and 20 (i.e., Figure 9c) also provided supportive results that were lower than others in ANNSVM_WLHT (i.e., Figure 7a) because their mother wavelets also had a similar shape as that of Coiflet 5. For similar reasons, the Haar wavelet was not proper because it is a step function.

After considering the unconventional results from Haar, Coiflet 5, Symlet 10, and Symlet 20 as described above, we again examined the significant differences between SVMANN_WLHT and ANNSVM_WLHT after omitting these wavelet families from our experiments; we did so in order to verify their effects. We performed the T-test on the results without the omitted wavelet families. Statistical results showed that the results of SVMANN_WLHT and ANNSVM_WLHT are equal, even if those wavelets are properly omitted; however, during the T-test, we observed that the average true positive (TP) rate of our proposed method remarkably improved to 0.90 which is greater than the mean of SVMANN_WLHT, i.e., 0.89. From these results, for graph-type classification, our proposed method is clearly more suitable because we obtained the highest accuracy and an acceptable average value, both outperforming results of SVMANN_WLHT.

=== Detailed Accuracy By Class ===

	Accuracy	TP Rate	FP Rate	Precision	Recall	F-Measure	Class
	0.875	0.875	0.05	0.903	0.875	0.889	2Dchart
	0.95	0.95	0.05	0.902	0.95	0.925	bar
	0.919	0.919	0.029	0.938	0.919	0.929	pie
Avg.	0.914	0.914	0.043	0.914	0.914	0.914	

=== Confusion Matrix ===

a	b	c	<-- classified as
280	25	15	a = 2Dchart
12	285	3	b = bar
18	6	273	c = pie

Figure 10. Detailed accuracy separated by classes and a confusion matrix which belongs to the dataset of Coiflet 1 applied by our main method (ANNSVM)

With regard to accuracy values of each class as presented in Figure 10, we observed that the accuracy of the two-dimensional chart class was the lowest (i.e., 0.875), while others were over 0.9. Results here suggested that both the bar and pie classes have their own unique characteristics, as opposed to the 2Dchart class. For example, the graph images that contained some rectangles were individually categorized in the bar graph class. A similar phenomenon occurred for circles in the pie chart class. In contrast, the 2Dchart class contained mixed types of graphs; hence, the graph characteristics belonging to the 2Dchart class varied.

Our proposed method is a combination of the ANN and the SVM, thus called the ANNSVM. We used the ANN to construct a temporary dataset, then applied the SVM for graph-type classification. Results from our proposed method showed that our approach outperformed the traditional methods because when we concurrently use two effective algorithms the strengths of both are encouraged and the weaknesses mitigated. For example, the ANN suffered from a problem of local minima, but the SVM strategy solves this problem; hence, the results from our proposed method guarantee the global optimization.

6. Conclusions

In this paper, we proposed a new method of graph-type classification by introducing a new preprocessing step to establish novel representative datasets instead of generic two-dimensional images and a novel classifier based on a combination of traditional algorithms. We conducted several experiments to verify our proposed method and the constructed datasets. Moreover, we extended the scope of our experiments to test data separability. To our knowledge, this is the first study to classify graph images belonging to the same type with different characteristics. We

investigated a reliable solution that can be applied to real-world data. Moreover, most results obtained from our experiments showed good agreement with our assumption.

As noted above, we conducted several experiments to evaluate our proposed method. We applied the CNN to two-dimensional graph images; however, we obtained very low accuracy values from these CNN experiments. We, therefore, state that the CNN was not a powerful algorithm for classifying graph types. To compare our proposed method to the CNN, we applied it to our constructed data, which combines the wavelet coefficients and outputs from the Hough transformation. The dataset consisted of three classes: bar, pie, and 2Dchart, with about 300 images per class. From our experimental results, we obtained the highest accuracy values in our experiments based on our proposed method, up to 0.91. Further, the most proper wavelet family applied to our data was Coiflet 1. As shown in the SVMANN_WLHT and ANNSVM_WLHT, the order of algorithms had not affected the results. It denotes that, regardless of using ANNSVM or SVMANN, the results were acceptable. Obviously, our method has been successful in classifying graph images. Moreover, the difficulty of different image characteristics has been overcome via our approach. The findings of our study suggest that our proposed method greatly contributes to graph-type classification.

In our future research, we will continuously develop our graph-type classification methodology. We will extract significant information from the graphs, such as axis titles and data point labels. The method for extracting such information will be different based on different graph types because of the dissimilarity of graph structures. Moreover, we will extend our study to be a semantic system using ontology. The extractable information will be an essential part of creating the ontology. We expect that our future system will be able to extract explicit and implicit information that represent intended relationships hidden in the graphs.

References

Aggarwal, C. C., & Yu, P. S. (2001, May). Outlier detection for high dimensional data. In *ACM Sigmod Record* (Vol. 30, No. 2, pp. 37-46). ACM.

Anthimopoulos, M. M., Gianola, L., Scarnato, L., Diem, P., & Mougiakakou, S. G. (2014). A food recognition system for diabetic patients based on an optimized bag-of-features model. *IEEE journal of biomedical and health informatics*, *18*(4), 1261-1271.

Arivazhagan, S., & Ganesan, L. (2003). Texture classification using wavelet transform. *Pattern recognition letters*, *24*(9), 1513-1521.

Arivazhagan, S., Ganesan, L., & Priyal, S. P. (2006). Texture classification using Gabor wavelets based rotation invariant features. *Pattern recognition letters*, *27*(16), 1976-1982.

Avci, E. (2008). Comparison of wavelet families for texture classification by using wavelet packet entropy adaptive network based fuzzy inference system. *Applied Soft Computing*, *8*(1), 225-231.

Barla, A., Odone, F., & Verri, A. (2003, September). Histogram intersection kernel for image classification. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on* (Vol. 3, pp. III-513). IEEE.

Cai, C. Z., Wang, W. L., Sun, L. Z., & Chen, Y. Z. (2003). Protein function classification via support vector machine approach. *Mathematical biosciences*, *185*(2), 111-122.

Chapelle, O., Haffner, P., & Vapnik, V. N. (1999). Support vector machines for histogram-based image classification. *IEEE transactions on Neural Networks*, *10*(5), 1055-1064.

Cheng, Y. C., & Chen, S. Y. (2003). Image classification using color, texture and regions. *Image and Vision Computing*, *21*(9), 759-776.

Cusano, C., Ciocca, G., & Schettini, R. (2003, December). Image annotation using SVM. In *Electronic Imaging 2004* (pp. 330-338). International Society for Optics and Photonics.

Del Frate, F., Pacifici, F., Schiavon, G., & Solimini, C. (2007). Use of neural networks for automatic classification from high-resolution images. *IEEE Transactions on Geoscience and Remote Sensing*, *45*(4), 800-809.

Duda, R. O., & Hart, P. E. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, *15*(1), 11-15.

Fan, Y., Shen, D., & Davatzikos, C. (2005). Classification of structural images via high-dimensional image warping, robust feature extraction, and SVM. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2005*, 1-8.

Fan, Y., Shen, D., & Davatzikos, C. (2005). Classification of structural images via high-dimensional image warping, robust feature extraction, and SVM. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2005*, 1-8.

Ibrahim, W. H., Osman, A. A. A., & Mohamed, Y. I. (2013, August). MRI brain image classification using neural networks. In *Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on* (pp. 253-258). IEEE.

Islam, M. R., Bulbul, F., & Shanta, S. S. (2012). Performance analysis of Coiflet-type wavelets for a fingerprint image compression by using wavelet and wavelet packet transform. *International Journal of Computer Science and Engineering Survey*, *3*(2), 79.

Kang, L., Kumar, J., Ye, P., Li, Y., & Doermann, D. (2014, August). Convolutional neural networks for document image classification. In *Pattern Recognition (ICPR), 2014 22nd International Conference on* (pp. 3168-3172). IEEE.

Kanjanawattana, S., & Kimura, M. (2015, December). Graph-type classification based on artificial neural networks and wavelet coefficients. In *Proceedings of Second International Conference on Digital Information Processing, Data Mining, and Wireless Communications (DIPDMWC2015)*.

Kanjanawattana, S., & Kimura, M. (2015, November). A Proposal for a Method of Graph Ontology by Automatically Extracting Relationships between Captions and X-and Y-axis Titles. In *KEOD* (pp. 231-238).

Kanjanawattana, S., & Kimura, M. (2016). Extraction of Graph Information Based on Image Contents and the Use of Ontology. *International Association for Development of the Information Society*.

Klöppel, S., Stonnington, C. M., Chu, C., Draganski, B., Scahill, R. I., Rohrer, J. D., Frackowiak, R. S. (2008). Automatic classification of MR scans in Alzheimer's disease. *Brain*, *131*(3), 681-689.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

Kwon, Y. H. (1994, June). Age classification from facial images. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on* (pp. 762-767). IEEE.

Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, *8*(1), 98-113.

Lorena, A. C., & De Carvalho, A. C. (2008). Evolutionary tuning of SVM parameter values in multiclass problems. *Neurocomputing*, *71*(16), 3326-3334.

Mehrotra, R., Namuduri, K. R., & Ranganathan, N. (1992). Gabor filter-based edge detection. *Pattern recognition*, *25*(12), 1479-1494.

Park, S. B., Lee, J. W., & Kim, S. K. (2004). Content-based image classification using a neural network. *Pattern Recognition Letters*, *25*(3), 287-300.

PyWavelets discussion group. (2008). Retrieved from http://wavelets.pybytes.com.

Salatino, A. A. (2014). Retrieved from http://infernusweb.altervista.org/wp/?p=786.

Sánchez, J., & Perronnin, F. (2011, June). High-dimensional signature compression for large-scale image classification. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (pp. 1665-1672). IEEE.

Sarlashkar, A. N., Bodruzzaman, M., & Malkani, M. J. (1998, March). Feature extraction using wavelet transform for neural network based image classification. In *System Theory, 1998. Proceedings of the Thirtieth Southeastern Symposium on* (pp. 412-416). IEEE.

Sergyan, S. (2008, January). Color histogram features based image classification in content-based image retrieval systems. In *Applied Machine Intelligence and Informatics, 2008. SAMI 2008. 6th International Symposium on* (pp. 221-224). IEEE.

Song, Q., Ni, J., & Wang, G. (2013). A fast clustering-based feature subset selection algorithm for high-dimensional data. *IEEE transactions on knowledge and data engineering*, *25*(1), 1-14.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929-1958.

Strigl, D., Kofler, K., & Podlipnig, S. (2010, February). Performance and scalability of GPU-based convolutional neural networks. In *Parallel, Distributed and Network-Based Processing (PDP), 2010 18th Euromicro International Conference on* (pp. 317-324). IEEE.

Vailaya, A., Figueiredo, M., Jain, A., & Zhang, H. J. (1999, July). Content-based hierarchical classification of vacation images. In *Multimedia Computing and Systems, 1999. IEEE International Conference on* (Vol. 1, pp. 518-523). IEEE.

Veluchamy, M., Perumal, K., & Ponuchamy, T. (2012). Feature extraction and classification of blood cells using artificial neural network. *American journal of applied sciences*, *9*(5), 615.

Veluchamy, M., Perumal, K., & Ponuchamy, T. (2012). Feature extraction and classification of blood cells using artificial neural network. *American journal of applied sciences*, *9*(5), 615.

Veluchamy, M., Perumal, K., & Ponuchamy, T. (2012). Feature extraction and classification of blood cells using artificial neural network. *American journal of applied sciences*, *9*(5), 615.

Xia, J., Du, P., He, X., & Chanussot, J. (2014). Hyperspectral remote sensing image classification based on rotation forest. *IEEE Geoscience and Remote Sensing Letters*, *11*(1), 239-243.

Sarunya Kanjanawattana received the B.E. degree in Computer Engineering from Suranaree University of Technology, Nakhonratchasima, Thailand, in 2008, and M. Eng from Asian Institute of Technology, Pathum Thani, Thailand in 2011. Currently, she is a Ph.D. student of Shibaura Institute of Technology, Tokyo, Japan. In the present, she works at Suranaree University of Technology as a lecturer in the department of computer engineering. Her research interests included data mining, machine learning, natural language processing, ontology and image processing.

Masaomi Kimura received a B.E. (1994) degree in precision engineering, and M.S. (1996), and D.S. (1999) degrees in Physics from University of Tokyo. He is a professor at the Department of Information Science and Engineering from Shibaura Institute of Technology. His current interests include data engineering, complex systems, and medical safety.