applied sciences

MDPI

Article

Line-segment Feature Analysis Algorithm Using Input Dimensionality Reduction for Handwritten Text Recognition

Chang-Min Kim [1], Ellen J. Hong [2], Kyungyong Chung [3] and Roy C. Park [4,*]

1 Division of Computer Information Engineering, Sangji University, Wonju 26339, Korea; changingstart@gmail.com
2 Department of Computer and Telecommunication Engineering, Yonsei University, Wonju 26493, Korea; ellenhong@yonsei.ac.kr
3 Division of Computer Science and Engineering, Kyonggi University, Suwon 16227, Korea; kychung@kyonggi.ac.kr
4 Department of Information Communication Software Engineering, Sangji University, Wonju 26339, Korea
* Correspondence: roypark@sangji.ac.kr

Received: 31 August 2020; Accepted: 30 September 2020; Published: 1 October 2020

Abstract: Recently, demand for handwriting recognition, such as automation of mail sorting, license plate recognition, and electronic memo pads, has exponentially increased in various industrial fields. In addition, in the image recognition field, methods using artificial convolutional neural networks, which show outstanding performance, have been applied to handwriting recognition. However, owing to the diversity of recognition application fields, the number of dimensions in the learning and reasoning processes is increasing. To solve this problem, a principal component analysis (PCA) technique is used for dimensionality reduction. However, PCA is likely to increase the accuracy loss due to data compression. Therefore, in this paper, we propose a line-segment feature analysis (LFA) algorithm for input dimensionality reduction in handwritten text recognition. This proposed algorithm extracts the line segment information, constituting the image of input data, and assigns a unique value to each segment using 3×3 and 5×5 filters. Using the unique values to identify the number of line segments and adding them up, a 1-D vector with a size of 512 is created. This vector is used as input to machine-learning. For the performance evaluation of the method, the Extending Modified National Institute of Standards and Technology (EMNIST) database was used. In the evaluation, PCA showed 96.6% and 93.86% accuracy with k-nearest neighbors (KNN) and support vector machine (SVM), respectively, while LFA showed 97.5% and 98.9% accuracy with KNN and SVM, respectively.

Keywords: feature extraction; dimensionality reduction; line-segment features; *k*-nearest neighbor; support vector machine

1. Introduction

In the past, the artificial intelligence field has accomplished a quantum leap. Currently, studies utilizing artificial intelligence are conducted and applied to various fields [1–3]. Especially, owing to the improvement in computing power, the areas of utilization constantly become wider in the field of computer vision. Furthermore, previous studies performed research related to image-text recognition [4–7]. Similarly, there are studies on optical character recognition (OCR), which is considered the representative technique of today for detecting texts utilizing image data [8–11]. Text and handwriting recognition started with the recognition of numbers in Latin text in the late-1950s and is expanding to include various languages, such as Chinese, Japanese, Arabic, and Persian. Currently,

the methods for recognizing new languages are becoming more accurate [12,13]. The recognition accuracy has increased through different recognition models because of the differences in the form of each language and in note-taking methods. Recently, the demand for handwriting recognition, such as the automation of mail sorting and electronic memo pads, has exponentially increased in various industrial fields. In addition, in the image recognition field, methods using convolutional neural networks (CNN), which show outstanding performance, have been applied to handwriting recognition [14,15].

Ashiquzzaman et al. [16] proposed an efficient CNN-based text recognition for Arabic handwriting. Their study used data augmentation to enhance the accuracy of the model. In addition, the method used dropout layers to resolve the problem of data overfitting and appropriately changed the activation function to overcome the vanishing gradient problem. Sampath and Gomati [17] proposed a hybrid neural network training algorithm for English handwriting OCR. It removed the noise of the input image and adjusted the size of the image using a median filter. In addition, feature sets and positional and structural descriptors were extracted from the input image. After the feature sets were extracted, the proposed FLM-based neural network recognized the handwritten text. FLM is a method that combines the firefly algorithm with the Levenberg–Marquardt (LM) algorithm for neural network training [18]. The proposed FLM-based neural network was integrated into a feedforward neural network, and, based on the size of the training data, the number of hidden neurons, and the number of hidden layers, 95% accuracy was achieved.

Shivakumara et al. [19] proposed a CNN-recurrent-neural-network (RNN)-based license plate recognition method. Their study investigated the combination of CNN and RNN, that is, bi-directional long short-term memory (BLSTM). Since CNN has high recognition power, it has been used in feature extraction, while BLSTM has been used in the extracting function of the context information based on past information. In addition, for the classification of license plates, dense cluster-based voting was proposed to differentiate the foreground from the background. The methods of these studies can achieve fast speed and high accuracy regarding the input data, can objectively and accurately judge complicated images, and can provide various services. However, the recognition rate is low, and there is a limitation to the recognition of various styles of writing if they fall outside a certain standard, or there is a modification of the image. In addition, there is a disadvantage that a large volume of labeled data is needed for the classification work [20,21]. Moreover, it is difficult to maintain spatial and structural consistency for the results of image segmentation. These technical drawbacks cause problems, such as unclear image outlines and incomprehensible small-area segmentation. In addition, to accomplish a high recognition rate in the handwriting recognition field, a deep structure CNN has been used. However, in the handwriting recognition field, terminals with limited resources, such as smartphones or tablet PCs, are often used. Therefore, memory occupied by the model and the calculation speed must be considered [22,23].

Thus, in this paper, we propose a line-segment feature analysis algorithm using input dimensionality reduction for handwritten text recognition. The proposed method is a dimensional reduction algorithm that compensates for the previously mentioned issues. It extracts points, lines, and faces from the original data. To extract such information, it uses a filter obtained by modifying the parameters of a 3×3 Laplacian mask. The information extracted through this filter is called a line segment map (LS-map), and it has a value of 0 or 1. LS-map performs a synthesis multiplication using a 3×3 filter and a 5×5 filter with unique numbers 0, 1, 2, 4, 8, 16, 32, 64, and 128 as parameters. Through this process, area 1 in the LS-map becomes replaced with a unique number, and because of the replaced information, the information about all line segments contains a series of numerical patterns according to each feature (visual information, such as vertical, horizontal, or curved). By summing up these patterns, we assign a unique value to each feature, and, by accumulating these patterns, we generate a 1-D vector. This vector has a maximum size of 256 owing to the filter parameters. This process is executed once for a 3×3 filter and 5×5 filter and, thus, generates two vectors. By merging these two vectors, a 1-D vector with the size of 512 elements is created. This vector is

then used as input data for the learning model. This newly proposed algorithm is called line-segment feature analysis (LFA) because it generates new data using the features of line segments.

This paper is organized as follows. Section 2 describes the handwritten text recognition service using artificial intelligence (AI) and the feature selection algorithm for data dimensionality reduction. Section 3 describes the line-segment feature analysis algorithm for the input dimensionality reduction in machine learning. For that purpose, we use feature extraction of the line-segment features (LS-features) analysis algorithm, unique number matching with LS-features, and eigenvalue cumulative aggregation. Section 4 describes the experiment with the proposed algorithm. Section 5 concludes the paper.

2. Related Research

2.1. Technology Trends for Handwritten Text Recognition

Text analysis is done through the inference of the characters of this document from the word, phrases, sentences, and paragraphs in the document [24,25]. Research on character has a considerably long history in the pattern recognition field. At the beginning, it had encountered recession due to the difficulty of character recognition. Research subjects to recognize were changed to different ones, such as voices and images. With the remarkable development of hardware technology and pattern recognition technology, there has been more necessary character recognition. With the emergence of the technologies that were undeveloped before, new methodologies were introduced [26–29]. Character recognition, in which character data is used as an input pattern, is generally dived into the learning stage and generalization stage. Figure 1 shows the process of character recognition system based on pattern recognition. In the learning stage, common attributes (features) are extracted from machine-printed or hand-written data input. A variety of machine learning algorithms are applied to generate a discrimination function (model) that helps to divide the extracted features into classes. If it is possible to discriminate between a pattern in an equal class and a pattern in an unequal class, it is simpler to solve a classification issue. However, it is impossible or extremely difficult to extract distinguishable features clearly from down-to-earth data. Accordingly, in the feature extraction step, it is required to extract features to minimize information loss. In addition, the machine learning process is of importance. The process makes it possible to learn a discrimination function that helps clear classification with the use of extracted features. In terms of inference, it is unreasonable to discriminate simply with the use of the information obtained from one pattern input.

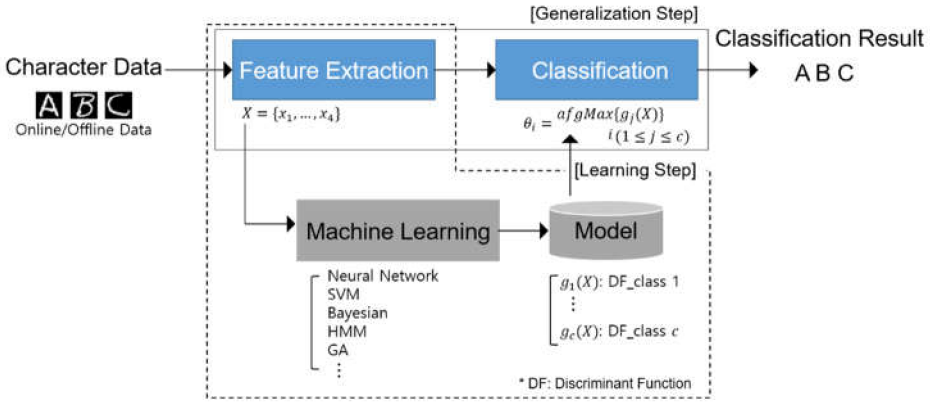


Figure 1. Process of character recognition system based on pattern recognition.

Generally, to model an optimal discrimination function for accurate classification, it is essential to optimize learning or training multiple data repeatedly. Therefore, relevant research has been actively conducted. For character recognition, Arica et al. [30] estimates such parameters as an inclination angle, a baseline, a stroke width and height, and conducts character segmentation in combination with gray scale and binary information. With the use of the Hidden Markov Model (HMM), labeling is applied to

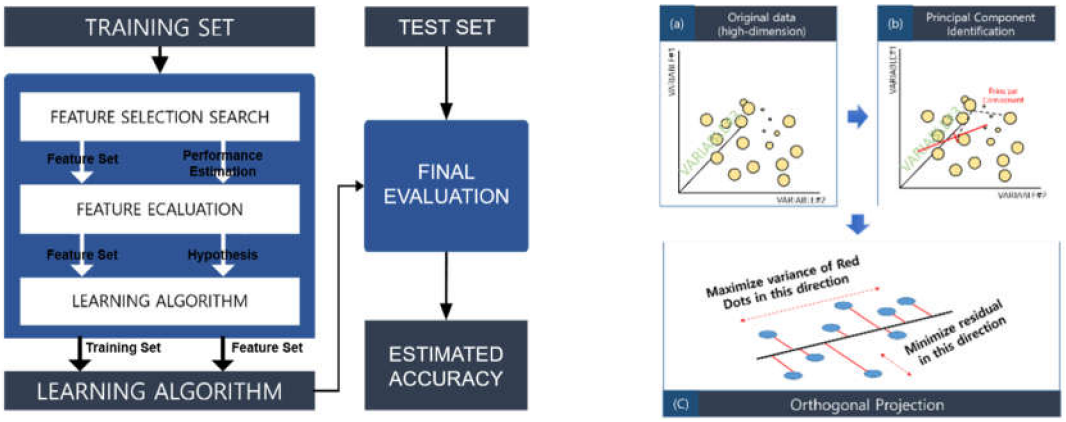
candidate characters, and then ranking is processed. The features presenting character candidates are extracted from segments, and character recognition is performed through learning. Hazra et al. [31] proposed the KNN (k-Nearest Neighbor) classifier for recognizing hand-written and printed English characters based on feature extraction technology. The KNN is a supervised learning algorithm to address a regression and classification issue. In the training step, the KNN algorithm makes a multi-dimensional feature vector assigned to a class label in order to improve recognition performance. Zanchettin et al. [32] proposed the hybrid KNN-SVM method for recognizing hand-written characters. For the improvement in the performance of KNN, Specialized SVM was developed. The developed method shows higher accuracy of character recognition than MLP, KNN, and hybrid MLP-SVM. In machine learning based character recognition, input data are character images. Therefore, with a high number of image data and a large size of the image, it takes long to learn. This method requires high dimensionality of input data.

Therefore, it is hard to express them as they are and apply them to an analysis. Therefore, dimensionality reduction is required in order to reduce a very large vector matrix to the dimension that can be processed.

2.2. Feature Selection Algorithm for Data Dimensionality Reduction

A dimensionality reduction technique is applied to reduce input data from a high-dimensional system to a low-dimensional system. The reduced data are divided into existing features and new features created by combining existing features. The technique of selecting only some of the features is called feature selection, and the technique of creating new features from the combination of existing features is called feature extraction. Feature selection is used to select some features for optimal classification, to create a subset, and then to classify target data using this subset. In this way, it is possible to reduce the operational volume and shorten the learning time. Nevertheless, feature selection is a complex process of selecting a subset and requires considerable time [33–36].

Figure 2 shows a dimensionality reduction technique. Figure 2a shows the feature selection procedure. The first process in feature selection is to extract all features of the learning data, and the next process is to create a subset of selected features. The learning algorithm is executed with the subset, and its performance is analyzed. This procedure is repeated until certain requirements are met. However, when target data are classified with a subject, it is highly likely to lower the performance of classification due to factors such as rotation and size. In addition, when classification is made only with some features, the accuracy is reduced. If the number of features is large, it takes too long to select the best subject.



(**a**) (**b**)

Figure 2. Dimensionality reduction technique. (**a**) Feature selection procedure. (**b**) Principal component analysis process

Feature extraction is a technique used to convert high-dimensional data to low-dimensional data and to generate new features. There are various methods of feature extraction, including principal component analysis (PCA), linear dimensionality reduction (LDA), independent component analysis (ICA), and non-negative matrix factorization. Among them, PCA is the representative technique [37–40]. PCA selects the main axis in the direction that has the largest data variance, and it then generates all the data. With data X, the correlation matrix R is calculated. With the use of R, the eigenvector γ and the eigenvalue Λ are calculated. γ, known as the largest value of the calculated Λ, is the direction with the largest variance in X. The new feature data drawn in this process have reduced dimensions with a minimized error. The components of γ have a direction that has no correlation between the factors.

Figure 2b shows the process of projecting data orthographically with the main components. As shown, the data are generally high-dimensional (e.g., three dimensions). The factors constituting an image are located in these dimensions. With the use of the variance value of these factors, the main axis is identified. At this time, the main axis selected has the direction with the largest variance value. Through this process, high-dimensional data are converted to low-dimensional data. Since PCA represents all the inputs in a linear combination, it becomes difficult to explain the process of converting all input variables into a linear combination, and the information about the output data becomes unknown [41–43]. For example, if there is image A (28×28), PCA will obtain 784 ($28 \times 28 = 784$) variables as input. Currently, PCA performs a covariance matrix on A to extract a series of eigenvectors that are principal vectors. In this case, it is difficult to determine the number of elements constituting the main components of the input data and on which features (characteristics) they are covariant. In addition, PCA can derive different output results (data size) even at the same compression rate, depending on the shape of the data (internal features of data). To collect output data of the same size, the parameters (number of principal components) must be changed to allow PCA to process all data at once or have all data of the same size. Converting all data at once is somewhat unreasonable, and, if the output size of PCA becomes fixed by setting parameters, the accuracy may be degraded depending on the selected parameters. At the time of writing, various techniques have been suggested and used to address the shortcomings of PCA. However, in this study, an LFA algorithm is suggested to solve the previously mentioned issues of PCA with a new approach. This newly suggested algorithm reduces the size of data by extracting and adding the line segment information of the given data.

3. Line-Segment Feature Analysis Algorithm for Input Dimensionality Reduction

This algorithm utilizes line-segment features to reduce the data dimensions. This means that the features of a point, line, and face are considered the fundamental factors constituting an image. In this study, line-segment information of an image is applied to a series of operation processes, and, then, one-dimensional line-segment cumulative total data with a size of 512 elements are created. For this reason, the algorithm is called a line-segment feature algorithm. Figure 3 shows the flow of the line-segment feature analysis algorithm for input dimensionality reduction. This algorithm identifies and totals line-segment information, and then creates a series of features through three processes: (a) feature extraction from the handwritten text image in which features such as points, lines, and faces are extracted through a filter, (b) unique number matching with LS-features (by applying a series of unique numbers to the extracted LS-maps, each line segment is assigned a series of numeric patterns), (c) eigenvalue cumulative aggregation (unique values are generated by summing the given numeric patterns, and this value is used to generate cumulative aggregate data).

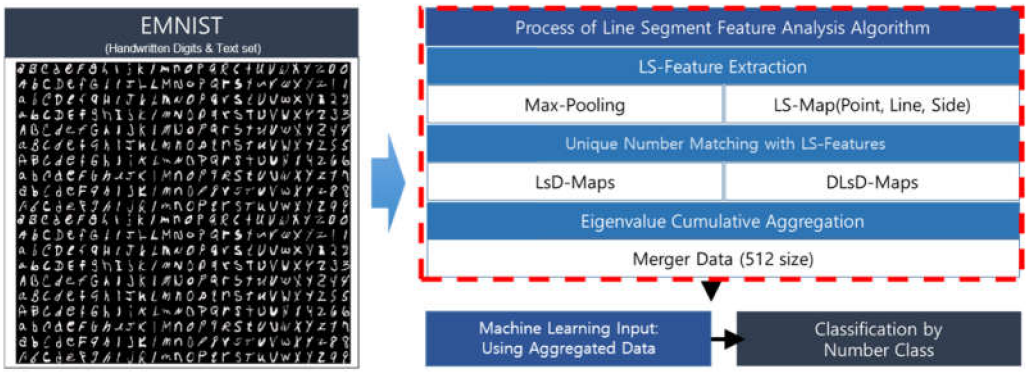


Figure 3. Flow of the line-segment feature analysis algorithm for input dimensionality reduction.

3.1. Feature Extraction Process for LFA

Algorithm 1 shows the feature detection algorithm, and Figure 4 shows the feature extraction process of the LFA algorithm, which is the first step of the algorithm. In this process, the features of a point, line, and side of input data are extracted. The original data are classified into points, lines, and faces to detect various characteristics. A line refers to the basic segment information, while a side refers to the original information. A point is information that distorts a segment, and the characteristics are extracted by distorting segments in the subsequent process.

Algorithm 1: Feature Detection Algorithm

Input: $[x_1, x_2, \ldots, x_n]$, K_w, K_h
def *Extraction of Feature*
  LF = [[0, 1, 0][1, −4, 1][0, 1, 0]]
  PF = [[1, −1, 1],[−1, 1, −1],[1, −1, 1]]
  for i ***from*** 0 ***to*** n ***do***
    for w ***from*** 0 ***to*** W-K_w ***do***
      for h ***from*** 0 ***to*** H-K_h ***do***
        for m ***from*** 0 ***to*** K_w ***do***
          for n ***from*** 0 ***to*** K_h ***do***
            L(w,h) += $x_i(w + m, h + n) \times$ LF
            P(w,h) += $x_i(w + m, h + n) \times$ PF
    if L > threshold ***then*** L[L > threshold] = 1 ***else*** L[L < threshold] = 0
    if P > threshold ***then*** P[P > threshold] = 1 ***else*** P[P < threshold] = 0
    $LS_{Map}(i) = [L, P, x_i]$
Output: LS_{Map}

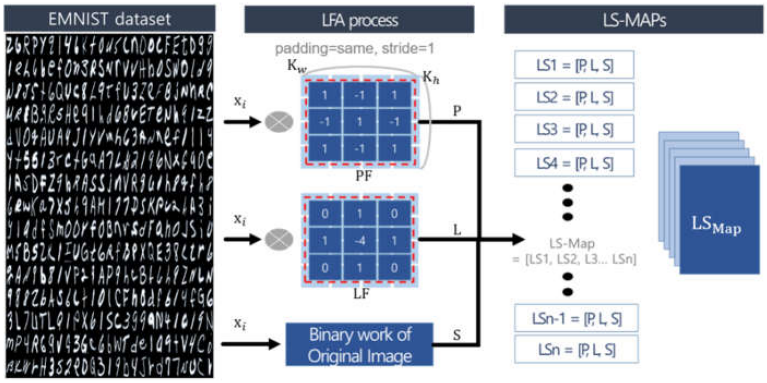


Figure 4. Feature extraction process of the line-segment feature algorithm (LFA).

K_w and K_h in Algorithm 1 are the size of the filter, and x is the handwritten text image. W and H are the dimensions of x and $\bigotimes$ is a convolution operation. n is the number of images, and the data calculated through the filter is LS_{Map}.

The process of extracting the characteristics of point, line, and faces in the LFA is described in Equations (1)–(5). This algorithm uses PF and LF filters in Equation (1) in order to extract the point, line, and faces features of line segments. Each filter is able to extract the contour, point segment, and face areas of the original data. The convolution ($\bigotimes$) of each filter by x, each point, line, and side map is generated (as shown in Equations (2) and (3)). The parameters of LF and PF filter in Equation (1) used and changed the parameters of the Laplacian filter. The LF filter is the Laplacian filter {{0, −1, 0}, {−1, 4, −1}, {0, −1, 0}}, and the PF is a filter designed by changing some parameters of the Laplacian filter in order to generate distortion, such as points around the contour of an object.

$$PF = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}, LF = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (1)$$

$$POINT_{map} = x \bigotimes PF \quad (2)$$

$$LINE_{map} = x \bigotimes LF \quad (3)$$

$$SIDE_{map} = x \quad (4)$$

$$LS_{Map} = \left[POINT_{map}, LINE_{map}, SIDE_{map}\right] \quad (5)$$

The features extracted by using the image of the upper case 'L' are shown in Figure 5. Figure 5a shows the original data. When PF is applied to (a), the contour of an area has some distortion, and the point segment as presented in Figure 5b is generated. LF extracts the contour of x. The extracted contour is shown in Figure 5c, and $SIDE_{map}$ is the application of the original data as is, which is shown in Figure 5d (as shown in Equation (4)). The italic-type number database used in this study is from the Extending Modified National Institute of Standards and Technology (EMNIST) [44], which has binary data as presented in Figure 5a. Therefore, it is possible to obtain side data without any additional work. The features of each segment extracted in this way are bound to generate an LS_{Map}, which is shown in Equation (5). When $\bigotimes$ in Equations (2) and (3) is executed, stride and padding are 1 value. Therefore, each feature has the same size as the original data, and the converted data are internal data only, which are either 0 or 1.

Figure 5. Output data of each feature. (**a**) Original image. (**b**) Point image. (**c**) Line image, (**d**) Face image.

3.2. Unique Number Matching with LS-Features

In the second process of the LFA, a unique number matches the extracted line-segment information for the LS_{Map}, as shown in Figure 6. Figure 6a shows the characteristics map extracted from Section 3.1, and Figure 6b,c are filters with a series of parameters. The reason this unique number is provided is to express the information about all the segments (e.g., curves, horizontal lines, and vertical lines) constituting the visual data as numerical information in a series, which makes it easy to combine. Algorithm 2 shows the unique number matching algorithm. K is the filter (the filter of size 3×3 or 5×5), and K_w and K_h are the sizes of K. N is the number of LS_{Map} (the number of images). LS is the

data with unique numbers matched. The unique numbers used are 0, 1, 2, 4, 8, 16, 32, 64, and 128 values. Each unique number is not the same, and a different total unique number is drawn. They are similar to a series of values used to convert a binary number to a decimal number in the engineering field. The LFA algorithm utilizes a 3 × 3 filter (Figure 6b) with unique numbers for matching with normal line-segment data, and a 5 × 5 filter (Figure 6c) with 0 in between unique numbers for extracting data of the distorted line segment. The LS_{Map} is 0 or 1. If the filter with unique numbers matches the LS_{Map}, only the area where the unique number is 1 reacts and is displayed. This, again, emulates the process of converting a binary number to a decimal number in the engineering field. The line-segment data are presented with 1, and a unique number is assigned to each line segment.

Algorithm 2: Unique Number Matching Algorithm

Input: LS_{Map}, K, K_w, K_h
def *Unique Number Matching*
  for i ***from*** 0 ***to*** N ***do***
    for w ***from*** 0 ***to*** W-K_w ***do***
      for h ***from*** 0 ***to*** H-K_h ***do***
        for m ***in*** 0 ***to*** K_w ***do***
          for n ***in*** 0 ***to*** K_h ***do***
            L(w, h) += LS_{Map}(i,0)(w + m, h + n) × K(K_w, K_h))
            P(w, h) += LS_{Map}(i,1)(w + m, h + n) × K(K_w, K_h))
            S(w, h] += LS_{Map}(i,2)(w + m, h + n) × K(K_w, K_h))
    LS(i) = [L, P, S]
Output: LS

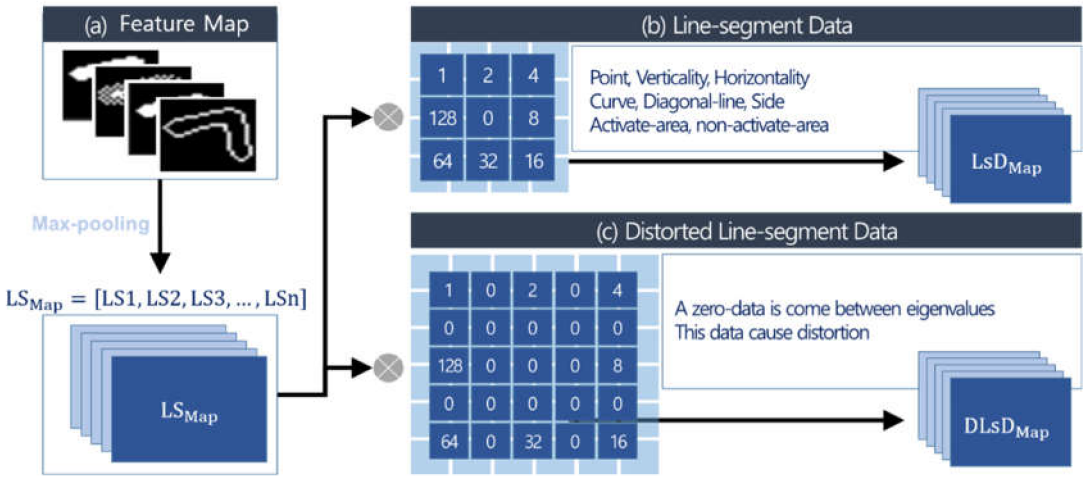


Figure 6. Unique number matching process of line-segment feature analysis.

Table 1 shows some of the types of normal line-segment data that can be extracted by a 3 × 3 filter. Table 1 presents some of the values that can be matched and extracted with the LS_{Map} using a 3 × 3 filter. As shown in Table 1, there are basic line types (e.g., point, vertical, horizontal, curved, and diagonal) and additional types, which include an activate area and a non–activate area. In an activate area, all areas in a filter are filled with 1, and all unique numbers 0, 1, 2, 4, 8, 16, 32, 64, and 128 values are found. In a non–activate area, all areas are filled with 0, and no unique numbers are detected. The median value of the 3 × 3 filter is filled with zeros. In other words, if the median value of the 3 × 3 filter is activated, it is judged to be a non–activate area.

Table 1. Types and values of line segments for the 3 × 3 filter.

LINE TYPE	Quantity
Point	1, 2, 4, 8, 16, 32, 64, 128
Verticality	{1, 2, 4}, {32, 64}, {8, 0, 16}, {8, 0}, ⋯
Horizontality	{1, 8}, {1, 8, 32}, {4, 16, 128}, ⋯
Curve	{1, 2, 8}, {8, 32, 64}, {16, 32, 64, 128}, ⋯
Diagonal	{1, 16}, {2, 8}, {1, 0, 128}, {4, 0, 32}, ⋯
Side	{0, 16, 32, 64}, {1, 2, 4, 8, 0, 16}, ⋯
Activate-area	0
Non–activate area	{0, 1, 2, 4, 8, 16, 32, 64, 128}

The 5 × 5 filter has 0 added in between unique numbers of the 3 × 3 filter. Since unique numbers are placed in the outermost positions, it is possible to obtain information on distorted line segments. A distorted line-segment technique is used to analyze the activate areas presented in the outer position, and to extract line segments without considering the center value of the 5 × 5 filter. Figure 7 shows the process of activating the area extraction using the 5 × 5 filter. In Figure 7, (a) is the input data, (b) shows the 5 × 5 filter, and (c) is the matched result.

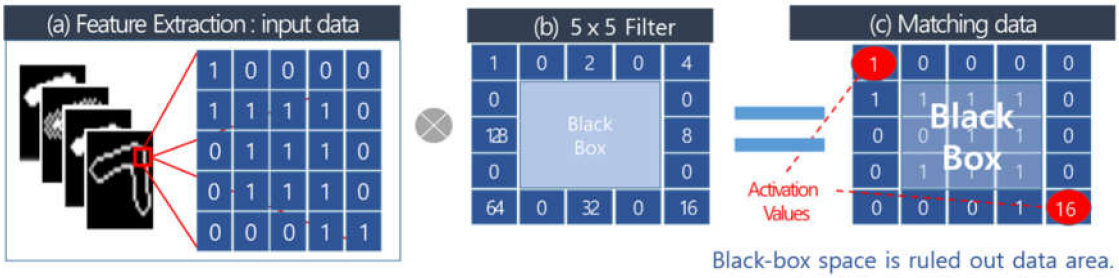


Figure 7. The process of activating the area extraction using the 5 × 5 filter.

As shown in Figure 7, the center of the 5 × 5 filter is a black box (consisting of zeros). If input data come in, the 1 located at the center is ignored. In other words, no matter what value is located in the area of the black box, the value is excluded. As presented in Figure 7c, a unique number can match the activated value presented around the area of the black box, and, thereby, the line-segment {1, 16} can be obtained. When the data in Table 1 are compared, it is possible to find that they are diagonal. From Figure 7a, it is possible to obtain partial diagonal and side information. However, if the 5 × 5 filter is applied, only diagonal information is detected.

Through this process, the line-segment information on the input data is distorted and extracted. As described in this section, the LFA extracts the LS_{Map} and enables a series of unique numbers to match line segments of the LS_{Map} through the 3 × 3 and 5 × 5 filters. Through this process, it is possible to give an eigenvalue to each line segment in terms of the matched unique numbers. An eigenvalue is the total amount of the matched unique numbers. In the engineering field, the value is calculated by matching a series of values to digits of a binary value and then finally adding them up. The LFA algorithm adds up the parameter values of the filter converted from an active area (area expressed as 1) by multiplying the convolution (stride, padding is 1) of 3 × 3 filters having 0, 1, 2, 4, 8, 16, 32, 64, and 128 parameters to the LS_{Map}. In this way, each line segment can obtain a particular value. If different line segments have the same value, they can be judged to be of the same type of line. In other words, by totaling the matched unique numbers, it is possible to find the types of line segments placed in a particular image.

3.3. Eigenvalue Cumulative Aggregation

The eigenvalues of the line segments extracted from the processes described above are different depending on the type of the line segment. In this study, the eigenvalue of a line segment is called

the eigenvalues map (EV_{Map}). An EV_{Map} includes information on all line segments distributed in the input data. In this section, the process of aggregating information on all line segments of an EV_{Map} is described (Figure 8). Algorithm 3 shows the cumulative aggregation algorithm. N is the number of LS, and $ARRAY_{1D}$ [256] is one-dimension arrangement of 256 elements in size. {0, ... } is the expression that initializes the arrangement to 0. W and H are the sizes of LS.

Algorithm 3: Cumulative Aggregation Algorithm

Input: LS
def *Cumulative Aggregation*
  for n ***in*** N ***do***
    $ARRAY_{1D}[256] = \{0, \dots\}$
    for w ***from*** 0 ***to*** W ***do***
      for h ***from*** 0 ***to*** H ***do***
        $LS_{1D}(w \times W + h) = LS(n)(w,h)$
    for i ***from*** 0 ***to*** 256 ***do***
      $ARRAY_{1D}(i) = ARRAY_{1D}(i)+1$
    $EV_{Map}(n) = ARRAY_{1D}$
Output: EV_{Map}
//Depending on the size of the filter, it is aggregated into LsD_{Map} or $DLsD_{Map}$.

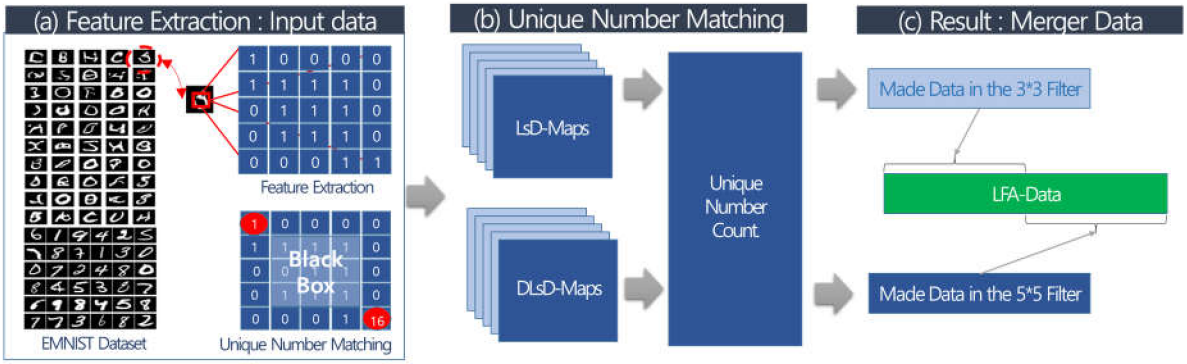


Figure 8. Eigenvalue aggregation of the line-segment feature analysis.

It is possible to measure an eigenvalue of up to 256 types. This is because the total number of matched unique numbers cannot exceed 256 values. In other words, it is only possible to calculate a vector with a maximum size of 256 elements. The aggregation process is presented in Equation (6).

$$\begin{aligned} \Delta &= array([\gamma, 1]) \\ \Delta\left[EV_{Map}(w,h)\right] &= \sum_{h}^{H}\sum_{w}^{W}\Delta\left[EV_{Map}(w,h)\right] + 1 \end{aligned} \tag{6}$$

In Equation (6), Δ represents aggregation data, γ represents the size of the aggregation data, or 256 value. H and W indicate the size of the EV_{Map}. As shown in Equation (6), a one-dimensional one-vector Δ is generated. During this time, the size was 256 values. The maximum presentation range is fixed at 256 types owing to the size of the unique number, as described in Section 3.2. Therefore, the size of Δ is set to 256 value. The initial value of the generated Δ is set to 0, and each index value of Δ increases with the use of the internal value of the EV_{Map}. In other words, Δ cumulatively aggregates with the use of an eigenvalue as an index value. The calculated Δ is used to obtain the number of line segments according to the types of line segments in the data. The LFA extracts information on accurate line segments with the use of the 3×3 filter, extracts information on distorted line segments with the use of the 5×5 filter, and then aggregates them. By summing up the cumulative aggregate data generated through the two filters, a 1-D vector with a size of 512 elements is obtained.

4. Experiment and Results

The experiment was performed on 64bit Windows, Intel® Core CPU i7-6700, 16GB RAM, and VGA NVIDIA GeForce GTX 1060 6GB. The data used in the experiments were the handwritten text data of the Extended MNIST (EMNIST) dataset with a size of 28 × 28. The learning data of this dataset is 124,800 data points, and the test dataset is composed of 20,800 data points. The PCA used in these experiments compressed 99% of the original data and had data with a size of 331 elements. This environment was designed to identify the features in a more accurate manner, using the highest compression rate of the PCA. The test data were generated by adjusting the parameters of PCA to be equal to the size of the learning data. In this study, the data processed using LFA and PCA were compared and analyzed, and LFA and PCA were analyzed using k-nearest neighbors (KNN) and support vector machine (SVM), which are representative machine learning techniques. In addition, to raise the reliability of the accuracy measurement, other dimensionality reduction techniques, linear discriminant analysis (LDA), independent component analysis (ICA), and t-distributed stochastic neighbor embedding (TSNE), were also performed. This experiment compared and considered the data processed through the suggested LFA and PCA. Figure 9 shows the size of the resulting data according to the compression of the data. The division was made by combining the training and testing data of the EMNIST database and then randomly mixing them and dividing them into the predetermined training and test set sizes.

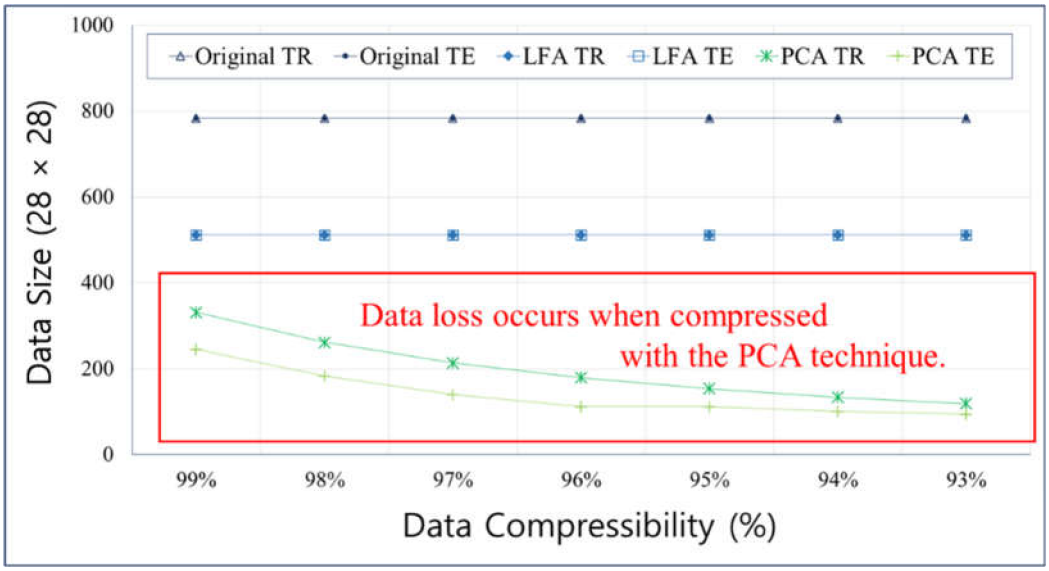


Figure 9. The size of the resulting data according to the compression of the data (TR: training, TE: test).

The results in Figure 9 show that the training and testing sizes were different for each compression rate. This is attributed to the linear combination, which is one of the shortcomings of PCA. PCA calculates the principal components by taking a covariance matrix for all inputs. In this process, PCA selects data with a large variance as the principal components without considering the features of the selected principal components. Therefore, PCA can yield different-sized outputs depending on the number and class distribution of the data. One technique to solve this issue is to fix the parameters of the PCA. As such, PCA can generate data of the same size regardless of the inputs. In this case, the selection of parameters for PCA is very critical. In the case of Figure 9, the output data shows a large difference of approximately 90 in the size between 99% and 98% compression. On the other hand, LFA maintains a constant size by identifying the number of linear segments and generating cumulative aggregate data. In addition, because its internal elements are of an integer type, they have relatively light capacities. Figure 10 shows the capacities of the processed data.

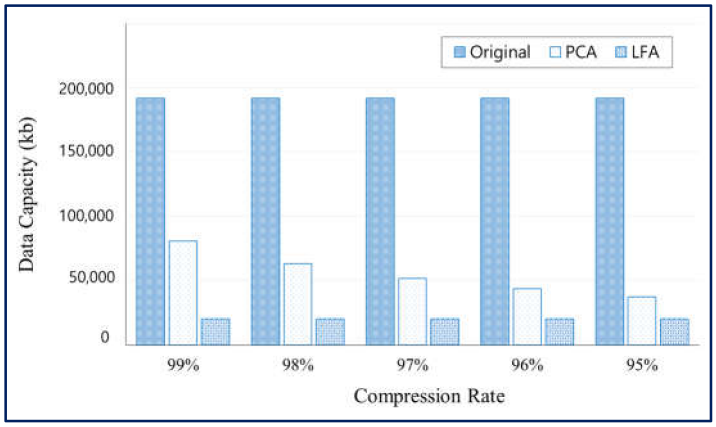
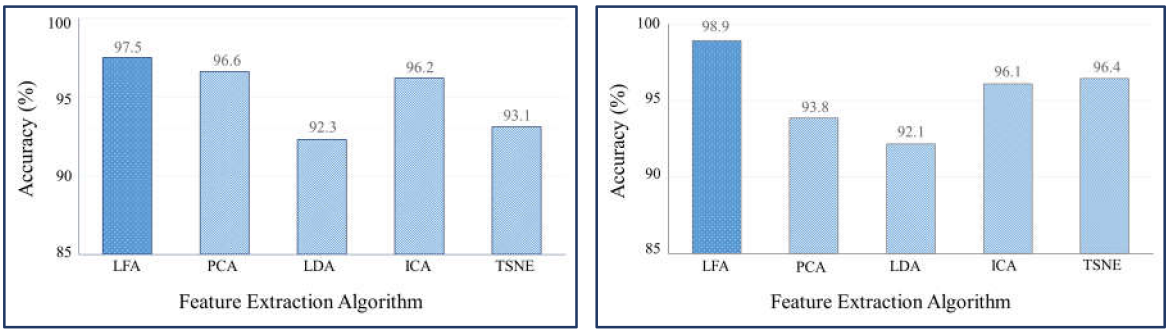


Figure 10. Capacities of the processed data (kb).

Since LFA derives an outcome by cumulatively summing up the unique values of all line segments, it only takes an integer value, and its aggregate data is simple count data. Therefore, it requires low memory capacity to store the data. In this experiment, the accuracy of KNN and SVM on LFA, PCA, LDA, ICA, and TSNE was measured. Figure 11 shows the accuracy measurement data using machine learning. The KNN and SVM were used with all parameters to set to their default values. We evaluated the performance of each technique in the same environment, which allowed us to understand the accuracy of each dimensionality reduction algorithms in the default environment. In the analysis in Figure 11, it is shown that the accuracy of each dimensionality reduction algorithms was 90% or above. In terms of accuracy with KNN, LFA showed the highest accuracy (97.5%), which was followed by PCA (96.6%), as shown in Figure 11a. This was followed by ICA (96.2%), TSNE (93.1%), and LDA (92.3%). On the other hand, in terms of SVM accuracy, LFA showed the highest accuracy (98.9%), which was followed by TSNE (96.4%), ICA (96.1%), PCA (93.8%), and LDA (92.1%), as shown in Figure 11b. In conclusion, LFA, showed relatively higher accuracy than the other dimensionality reduction algorithms. Especially using SVM, LFA showed the highest accuracy at 98.9% among the measured performances.



(**a**) (**b**)

Figure 11. Accuracy measurement data using machine learning. (**a**) Accuracy measurement data using k-nearest neighbors. (**b**) Accuracy measurement data using a support vector machine.

We suggest the LFA algorithm that reduces the size of image through the analysis of line types. The suggested algorithm showed the reduction of computation by the size reduction of the image with the use of EMNIST in KNN and SVM environments, along with higher performance than the existing model in accuracy. We consider the reason for these experimental results as the result of strong classification of characteristics. In the LFA algorithm, the lines for the shapes of objects existing in the input images are converted to aggregated data. This property maintains the characteristics of shapes while classifying strong characteristics effectively in reducing data sensitive to shapes. Letters are one

of data sensitive to shapes. In recognizing letters, colors are classified as insignificant characteristics, and it classifies based only on the shapes of letters. If letter "O" is converted through the LFA algorithm, the aggregated values on the forms of the curve and diagonal line are greatly enhanced, and other forms show a low aggregate. This factor functions as a big advantage in classifying shapes. The distribution of aggregated values includes different shape characteristics. For these reasons, in the reduction of sizes of cursive letter images, the linear characteristics of the relevant letters are highlighted, and the sizes are effectively reduced without loss of characteristics. Thanks to this process, the high computation of the existing learning model is decreased.

In addition, this study measured precision, recall, accuracy, and the receiver operating characteristic (ROC) curve to evaluate the classification performance [45,46]. Figures 12 and 13 show precision and recall, and ROC is shown in Figure 14. After examining each performance evaluation, it is noted that LFA shows high accuracy, precision, and recall, and it is found that its ROC curve indicates a good performance. Figures 12 and 13 show low performance at some classes (6, 8, and 16). However, the other algorithms also show low classification performance of those classes. When reviewing ROC curves, LFA shows higher performance than the comparison algorithms with KNN. In SVM, on the other hand, LFA, LDA, and TSNE show similar performances with high performance in some sections. When these results are combined, the LFA shows a similar and partially higher performance than the comparison algorithms across all results calculated by the two models. Although this algorithm cannot be evaluated as showing a better performance than the existing dimensionality reduction algorithms, it can be evaluated as effective in reducing image data dimensionality.

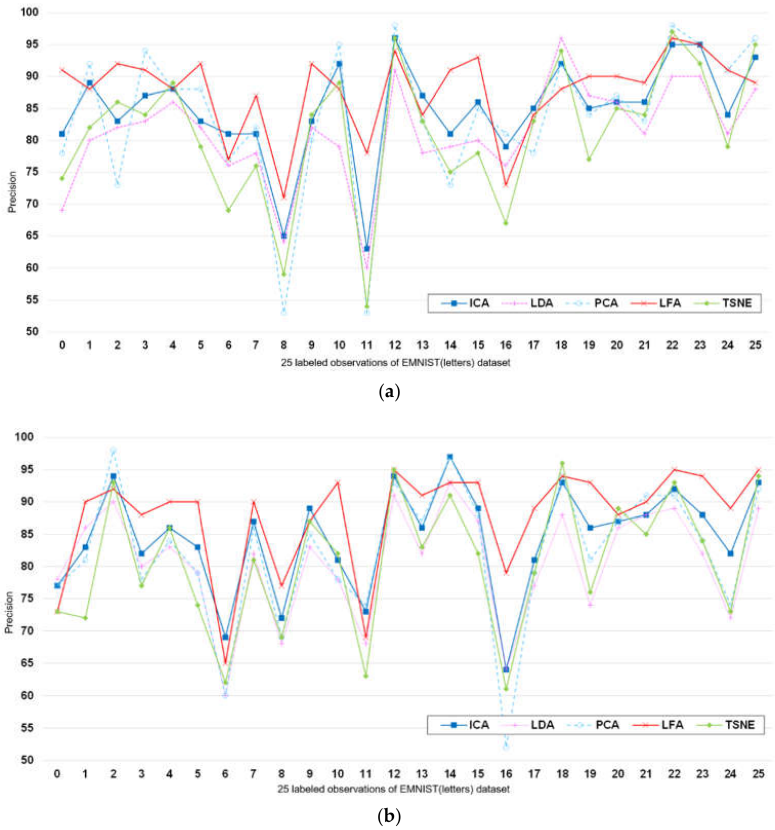
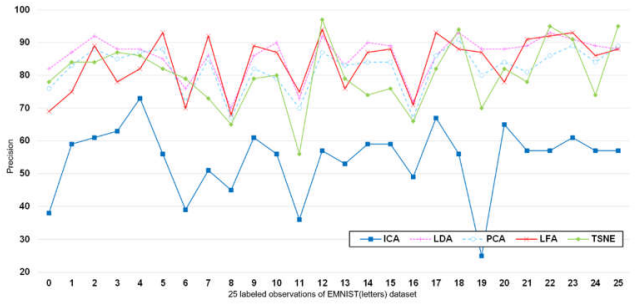
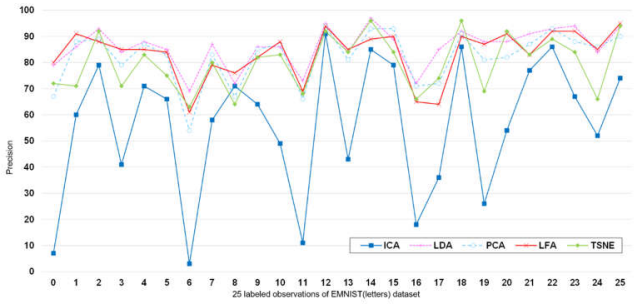


Figure 12. Results of precision and recall evaluation by feature extraction algorithm in k-nearest neighbors (KNN). (**a**) Precision measurement data using KNN. (**b**) Recall measurement data using KNN.

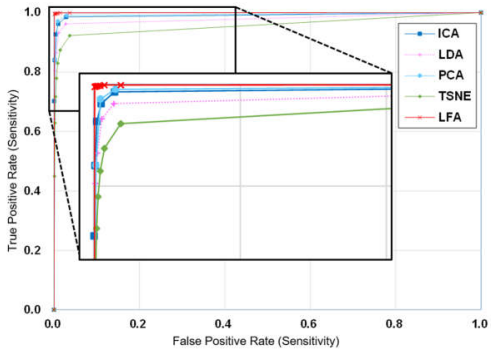


(**a**)

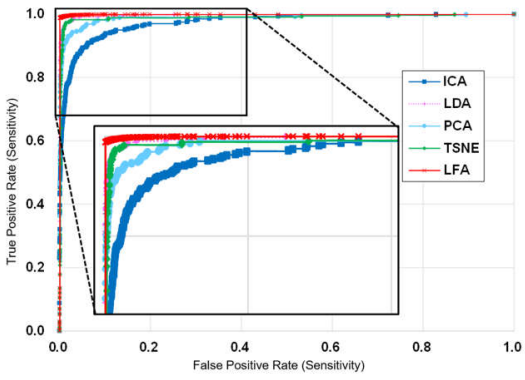


(**b**)

Figure 13. Results of precision and recall evaluation by the feature extraction algorithm in a support vector machine (SVM). (**a**) Precision measurement data using SVM. (**b**) Recall measurement data using SVM.



(**a**)



(**b**)

Figure 14. Receiver operating characteristic (ROC) curve by the feature extraction algorithm. (**a**) ROC curve using k-nearest neighbors. (**b**) ROC curve using a support vector machine.

5. Conclusions

In this paper, we proposed a line-segment feature analysis algorithm using input dimensionality reduction for handwritten text recognition. It suggests a dimensionality reduction algorithm to compensate for the problems that might be caused by the linear combination generated by PCA used for dimensional reduction algorithms. Unlike PCA, this newly suggested algorithm uses all the information of linear segments with all features to generate new features by identifying the main features and then orthogonally projecting all the remaining features. This new algorithm extracts contours, lines, and faces from the given data and then identifies the types of line segments and sums them up. Through this process, it generates a 1-D vector with a size of 512 elements. LFA uses 3×3 and 5×5 filters to extract features from line-segment information. These filters have serial numbers of 0, 1, 2, 4, 8, 16, 32, 64, and 128 as parameters. The 3×3 filter derives information based on the type of line segment, such as point, vertical, horizontal, and diagonal. On the other hand, the 5×5 filter distorts the features of linear segments to derive information. Each derived information is in the format of a 1-D vector with a size of 256 elements, and, by merging these data, LFA derives data with a size of 512 elements.

To evaluate the performance of the algorithm suggested in this study, machine learning (KNN and SVM) with the EMNIST database was used. To increase the reliability of this experiment, LDA, ICA, and TSNE were also performed. The results show that LFA achieved 97.5% accuracy with KNN and 98.9% with SVM, and PCA achieved 96.6% and 93.8% with KNN and SVM, respectively. We obtained insight into the potential of LFA in this experiment. Therefore, we are dedicated to designing an optimal learning model for the LFA algorithm as well as working on the improvement and future direction of the LFA algorithm by using various data (emotion, face, etc.). In addition, we will continue to enhance the LFA, so that it can relate to various learning neural networks that are currently used.

Author Contributions: C.-M.K. and E.J.H. conceived and designed the framework. K.C. and R.C.P. performed the experiments and analyzed the results. All authors contributed to writing and proofreading the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the GRRC program of Gyeonggi province [GRRC KGU 2020-B03, Industry Statistics and Data Mining Research].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yoo, H.; Han, S.; Chung, K. A Frequency Pattern Mining Model Based on Deep Neural Network for Real-Time Classification of Heart Conditions. *Healthcare* **2020**, *8*, 234. [CrossRef] [PubMed]
2. Shin, D.H.; Chung, K.; Park, R.C. Prediction of Traffic Congestion Based on LSTM through Correction of Missing Temporal and Spatial Data. *IEEE Access* **2020**, *8*, 150784–150796. [CrossRef]
3. Baek, J.W.; Chung, K. Context Deep Neural Network Model for Predicting Depression Risk Using Multiple Regression. *IEEE Access* **2020**, *8*, 18171–18181. [CrossRef]
4. Shin, D.H.; Chung, K.; Park, R.C. Detection of Emotion Using Multi-Block Deep Learning a Self-Management Interview App. *Appl. Sci.* **2019**, *9*, 4830. [CrossRef]
5. Kim, C.M.; Hong, E.J.; Chung, K.; Park, R.C. Driver Facial Expression Analysis Using LFA-CRNN-Based Feature Extraction for Health-Risk Decisions. *Appl. Sci.* **2020**, *10*, 2956. [CrossRef]
6. Govindan, V.K.; Shivaprasad, A.P. Character recognition-a review. *Pattern Recognit.* **1990**, *23*, 671–683. [CrossRef]
7. Trier, Ø.D.; Jain, A.K.; Taxt, T. Feature extraction methods for character recognition-a survey. *Pattern Recognit.* **1996**, *29*, 641–662. [CrossRef]
8. Mohammed Aarif, K.O.; Roruran, S. OCR-Nets: Variants of Pre-trained CNN for Urdu Handwritten Character Recognition via Transfer Learning. *Procedia Comput. Sci.* **2020**, *171*, 2294–2301.
9. Pramanik, R.; Bag, S. Shape decomposition-based handwritten compound character recognition for Bangla OCR. *J. Vis. Commun. Image Represent.* **2018**, *50*, 123–134. [CrossRef]

10. Goodfellow, I.J.; Bulatov, Y.; Ibarz, J.; Arnoud, S.; Shet, V. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *arXiv* **2013**, arXiv:1312.6082.
11. Coates, A.; Carpenter, B.; Case, C.; Satheesh, S.; Suresh, B.; Wang, T.; Ng, A.Y. Text detection and character recognition in scene images with unsupervised feature learning. In Proceedings of the 2011 International Conference on Document Analysis and Recognition, Beijing, China, 18–21 September 2011; pp. 440–445.
12. Gan, J.; Wang, W.; Lu, K. In-air handwritten Chinese text recognition with temporal convolutional recurrent network. *Pattern Recognit.* **2020**, *97*, 107025. [CrossRef]
13. Sanchez, J.A.; Romero, V.; Toselli, A.H.; Villegas, M.; Vidal, E. A set of benchmarks for Handwritten Text Recognition on historical documents. *Pattern Recognit.* **2019**, *94*, 122–134. [CrossRef]
14. Wang, Z.R.; Du, J.; Wang, J.M. Writer-aware CNN for parsimonious HMM-based offline handwritten Chinese text recognition. *Pattern Recognit.* **2020**, *100*, 107102. [CrossRef]
15. Li, Z.; Wu, Q.; Xiao, Y.; Jin, M.; Lu, H. Deep Matching Network for Handwritten Chinese Character Recognition. *Pattern Recognit.* **2020**, *107*, 107471. [CrossRef]
16. Ashiquzzaman, A.; Tushar, A.K.; Rahman, A.; Mohsin, F. An efficient recognition method for handwritten arabic numerals using cnn with data augmentation and dropout. In *Data Management, Analytics and Innovation*; Springer: Singapore, 2019; pp. 299–309.
17. Sampath, A.K.; Gomathi, N. Handwritten optical character recognition by hybrid neural network training algorithm. *Imaging Sci. J.* **2019**, *67*, 359–373. [CrossRef]
18. Wang, S.; Cai, G.; Zhu, Z.; Huang, W.; Zhang, X. Transient signal analysis based on Levenberg–Marquardt method for fault feature extraction of rotating machines. *Mech. Syst. Signal Process.* **2015**, *54*, 16–40. [CrossRef]
19. Shivakumara, P.; Tang, D.; Asadzadehkaljahi, M.; Lu, T.; Pal, U.; Anisi, M.H. CNN-RNN based method for license plate recognition. *CAAI Trans. Intell. Technol.* **2018**, *3*, 169–175. [CrossRef]
20. Lodhi, B.; Kang, J. Multipath-DenseNet: A Supervised ensemble architecture of densely connected convolutional networks. *Inf. Sci.* **2019**, *482*, 63–72. [CrossRef]
21. Park, S.S.; Chung, K. MMCNet: Deep learning–based multimodal classification model using dynamic knowledge. *Pers. Ubiquitous Comput.* **2019**. [CrossRef]
22. Zhang, R.; Nie, F.; Li, X.; Wei, X. Feature selection with multi-view data: A survey. *Inf. Fusion* **2019**, *50*, 158–167. [CrossRef]
23. Rong, D.; Xie, L.; Ying, Y. Computer vision detection of foreign objects in walnuts using deep learning. *Comput. Electron. Agric.* **2019**, *162*, 1001–1010. [CrossRef]
24. Kim, H.J.; Baek, J.W.; Chung, K. Optimization of Associative Knowledge Graph Using TF-IDF Based Ranking Score. *Appl. Sci.* **2020**, *10*, 4590. [CrossRef]
25. Kim, J.C.; Chung, K. Knowledge expansion of metadata using scriptmining analysis in multimedia recommendation. *Multimed. Tools Appl.* **2020**. [CrossRef]
26. Rehman, U.; Mahmud, S.; Chang, Y.K.; Jin, L.; Shin, J. Current and future applications of statistical machine learning algorithms for agricultural machine vision systems. *Comput. Electron. Agric.* **2019**, *156*, 585–605. [CrossRef]
27. Tao, J.; Sun, G. Application of deep learning based multi-fidelity surrogate model to robust aerodynamic design optimization. *Aerosp. Sci. Technol.* **2019**, *92*, 722–737. [CrossRef]
28. Lago, J.; Ridder, F.D.; Schutter, B.D. Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. *Appl. Energy* **2018**, *221*, 386–405. [CrossRef]
29. Tappert, C.C.; Suen, C.Y.; Wakahara, T. The state of the art in online handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **1990**, *12*, 787–808. [CrossRef]
30. Arica, N.; Yarman-Vural, F.T. Optical character recognition for cursive handwriting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 801–813. [CrossRef]
31. Hazra, T.K.; Singh, D.P.; Daga, N. Optical character recognition using KNN on custom image dataset. In Proceedings of the Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON), Bangkok, Thailand, 16–18 August 2017; pp. 110–114.
32. Zanchettin, C.; Bezerra, B.L.D.; Azevedo, W.W. A KNN-SVM hybrid model for cursive handwriting recognition. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Brisbane, Australia, 10–15 June 2012; pp. 1–8.

33. Park, S.S.; Baek, J.W.; Jo, S.M.; Chung, K. Motion Monitoring using Mask R-CNN for Articulation Disease Management. *J. Korea Converg. Soc.* **2019**, *10*, 1–6.
34. Cao, B.; Zhao, J.; Yang, P.; Yang, P.; Liu, X.; Qi, X.; Simpson, A.; Elhoseny, M.; Nehmood, I.; Muhammad, K. Multiobjective feature selection for microarray data via distributed parallel algorithms. *Future Gener. Comput. Syst.* **2019**, *100*, 952–981. [CrossRef]
35. Ku, B.H.; Kin, G.T.; Min, J.K.; Ko, H.S. Deep convolutional neural network with bottleneck structure using raw seismic waveform for earthquake classification. *J. Korea Soc. Comput. Inf.* **2019**, *24*, 33–39.
36. Wu, C.; Fan, W.; He, Y.; Sun, J.; Naoi, S. Handwritten character recognition by alternately trained relaxation convolutional neural network. In Proceedings of the International Conference on Frontiers in Handwriting Recognition, Heraklion, Greece, 1–4 September 2014; pp. 291–296.
37. Yao, S.; Chang, Y.; Qin, X.; Zhang, Y.; Zang, T.; Tianxu, Z. Principal component dictionary-based patch grouping for image denoising. *J. Vis. Commun. Image Represent.* **2018**, *50*, 111–122. [CrossRef]
38. Zhao, W.; Du, S. Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4544–4554. [CrossRef]
39. Li, S.Z.; Yu, B.; Wu, W.; Su, S.Z.; Ji, R.R. Feature learning based on SAE–PCA network for human gesture recognition in RGBD images. *Neurocomputing* **2015**, *151*, 565–573. [CrossRef]
40. Zhang, M.; Khan, S.; Yan, H. Deep eigen-filters for face recognition: Feature representation via unsupervised multi-structure filter learning. *Pattern Recognit.* **2019**, *100*, 107176. [CrossRef]
41. Liu, Y.; Zhang, G.; Xu, B. Compressive sparse principal component analysis for process supervisory monitoring and fault detection. *J. Process Control* **2017**, *50*, 1–10. [CrossRef]
42. Kang, T.J.; Eo, S.H.; Cho, H.J.; Donatelli, R.E.; Lee, S.J. A sparse principal component analysis of Class III malocclusions. *Angle Orthod.* **2019**, *89*, 768–774. [CrossRef]
43. Park, S.B.; Oh, S.K. Radial Basis Function Neural Networks Classifier for Face Recognition: A Comparative Studies Using Two-Dimensional Preprocessing Algorithms. *J. Korean Inst. Intell. Syst.* **2019**, *29*, 104–110. [CrossRef]
44. Cohen, G.; Afshar, S.; Tapson, J.; van Schaik, A. EMNIST: An extension of MNIST to handwritten letters. *arXiv* **2017**, arXiv:1702.05373.
45. Egghe, L. The measures precision, recall, fallout and miss as a function of the number of retrieved documents and their mutual interrelations. *Inf. Process. Manag.* **2008**, *44*, 856–876. [CrossRef]
46. Soleymania, R.; Granger, E.; Fumera, G. F-measure curves: A tool to visualize classifier performance under imbalance. *Pattern Recognit.* **2020**, *100*. [CrossRef]