electronics

MDPI

Article

Principal Component Analysis-Based Logistic Regression for Rotated Handwritten Digit Recognition in Consumer Devices

Chao-Chung Peng *, Chao-Yang Huang and Yi-Ho Chen

Department of Aeronautics and Astronautics, National Cheng Kung University, Tainan 70101, Taiwan; p46124358@gs.ncku.edu.tw (C.-Y.H.); p48101500@gs.ncku.edu.tw (Y.-H.C.)
* Correspondence: ccpeng@mail.ncku.edu.tw

Abstract: Handwritten digit recognition has been used in many consumer electronic devices for a long time. However, we found that the recognition system used in current consumer electronics is sensitive to image or character rotations. To address this problem, this study builds a low-cost and light computation consumption handwritten digit recognition system. A Principal Component Analysis (PCA)-based logistic regression classifier is presented, which is able to provide a certain degree of robustness in the digit subject to rotations. To validate the effectiveness of the developed image recognition algorithm, the popular MNIST dataset is used to conduct performance evaluations. Compared to other popular classifiers installed in MATLAB, the proposed method is able to achieve better prediction results with a smaller model size, which is 18.5% better than the traditional logistic regression. Finally, real-time experiments are conducted to verify the efficiency of the presented method, showing that the proposed system is successfully able to classify the rotated handwritten digit.

Keywords: Principal Component Analysis; logistic regression; rotated image; handwritten digit recognition

Citation: Peng, C.-C.; Huang, C.-Y.; Chen, Y.-H. Principal Component Analysis-Based Logistic Regression for Rotated Handwritten Digit Recognition in Consumer Devices. *Electronics* **2023**, *12*, 3809. https://doi.org/10.3390/electronics12183809

Academic Editors: Jungpil Shin, Md. Al Mehedi Hasan and Hoang D. Le

Received: 5 August 2023
Revised: 1 September 2023
Accepted: 5 September 2023
Published: 8 September 2023

1. Introduction

Computer vision has been widely used in many fields to enable image recognition [1]. As a result, there have been many applications of image recognition, such as automatic following car, license plate recognition, and facial recognition [2,3]. The work in [3] developed a Convolutional Neural Network (CNN)-based recognition system to recognize Peruvian license plates. The research developed a model with 100% accuracy, 0% failure, and 100% sensibility, with 100% specificity. It is a typical use of computer vision to perform image recognition. Among all of its uses, one of the most common applications in the field of consumer electronics is handwritten digit recognition, such as handwriting input on smartphones or smart pads.

A common use of the handwritten digit recognition system in common consumer mobile phones is shown in Figure 1. With a handwritten character "8", as illustrated in Figure 1a, the typing system can automatically recognize the digit. However, this experiment shows that handwritten digit recognition only works successfully when the digit is vertically well aligned. Recognition failure will be induced when the digit is subjected to apparent rotation, as shown in Figure 1b.

Therefore, the novelty of the research is that it proposes a handwritten digit recognition system with high accuracy for rotated images that significantly reduces the requirements of high computation complexity and memory, such as the Convolutional Neural Network (CNN)-based method shown in ref. [4]. Unlike the CNN-based pattern recognition methods, the proposed method is adequate for implementation in low-cost embedded systems or consumer electronics with restricted storage capacities. To develop such a low-cost consumer electronics system, it is necessary to choose appropriate methods of image pose correction and image recognition algorithms.

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

Figure 1. Handwritten digit recognition system in a mobile phone. The images in red blocks represent the recognition result of the system. (**a**) The system correctly recognizes the digit. (**b**) The system fails to recognize the digit subjected to rotations.

There are a lot of studies working on handwritten image recognition. For example, ref. [5] proposed a Deep Convolutional Self-Organizing Maps (DCSOM) network to learn unlabeled visual data. The experimental result showed that the proposed DCSOM had a high accuracy when predicting noise digits. Ref. [6] proposed an offline handwritten digit recognition system, which was trained using the MNIST dataset. The proposed system is a CNN-based recognition system, which is one of the most common types of computer vision model. Ref. [7] proposed a DeblurGAN-CNN model to recognize character images with noise, which is a system composed of two networks. The proposed system mentioned in ref. [7] uses different datasets to test the system, and the experimental results showed that the recognition system could recognize images with noise. The methods proposed by different researchers all have relatively high accuracy compared to basic methods. However, training such large models often takes long time. This issue is one of the most significant disadvantages of using these methods to perform image recognition.

Due to the long training and recognition time, lots of studies have tried to solve this problem. Ref. [8] focused on the detection Speed of Synthetic Aperture Radar (SAR), which became faster using faster region-based CNN (R-CNN). Though the accuracy of the proposed model is roughly the same as that of conventional R-CNN, the recognition speed is 8 times faster. Ref. [9] developed a CNN-based Range Partition (CRP), which satisfied the requirements of fast packet classification and online rule updating. As seen in the two studies, nowadays, developing recognition system with shorter training times and faster recognition speeds is one of the most important objectives.

Out of all of the methods of solving recognition problems, logistic regression is one of the most common [10]. The idea of this method is to use the concept of probability to solve the multiple-class classification problem, which can, thus, be used to enable recognition. As it is a basic but powerful method, there are several studies using logistic regression to perform different types of tasks. Ref. [11] used online logistic regression to approach the parameter estimation problem to replace Sequential Monte Carlo (SMC), while ref. [12] used logistic regression to approach the binary dynamical process used to reconstruct a network. Logistic regression is used in these articles for different purposes, which shows that it is a basic but powerful algorithm in terms of modeling. In this article, it will be used to perform digit image recognition.

The other important field of digit recognition is image straightening. Inspired by the ideas of [13,14], the proposed method applies PCA to image rotation to increase the accuracy of handwritten digit identification. The two previous studies showed that PCA can be used to perform image alignment. Using PCA to straighten the rotated images, the accuracy of recognition can be improved. However, it is found that PCA has a problem of orientation uncertainty. When an image is aligned via PCA, the image might be upside down, as PCA only uses the direction of the first principal component. To solve this problem, ref. [13] used a method of finding the direction and the angle of the rotation. In this research, PCA is not only used to reduce the dimension of the data, but also to

find appropriate axes to perform image rotation. Therefore, the result of the "principal component" via PCA has its physical meanings and is hard to replace with other feature extraction methods, such as Linear Discriminant Analysis (LDA). The LDA can produce a better data separation result, since it can maximize the separation between different classes and minimize it between the data with the same classes. However, the coordinates found via LDA are not guaranteed to be orthogonal, which may cause the digits to distort after rotation.

There are also different methods used to find the orthogonal coordinates to prevent distortion, such as Sparse Principal Component Analysis (SPCA) and Multilinear Principal Component Analysis (MPCA). However, they require more time and computation complexity compared to conventional PCA. Considering the real-time application in electronic devices with limited storage capacities, PCA will be the best candidate.

Unlike the paper in ref. [13], PCA is also utilized to reduce the dimension of the features to perform classification, as illustrated in refs. [15,16]. With a reduced data dimension, the training process can be much faster than traditional logistic regression. This speed allows the proposed method to use fewer parameters and reduces the storage capacities required to perform implementation. Similarly, although the identification accuracy in refs. [17,18] may have better results than the proposed method, it can be challenging to realize these methods in embedded systems or consumer electronic devices. In contrast, the proposed method has the advantage of seamless integration into real-time applications of constrained storage devices.

Based on this advantage, PCA is, thus, used in some studies that focus on high-dimensional data. Ref. [19] proposed a deep method based on PCA and DCNN to classify normal traffic and abnormal traffic, while ref. [20] used PCA in many different cases to verify that PCA and random projection can be used to perform dimension reduction. On the other hand, the low recognition time can make the real-time recognition frequency much higher. This fact helps improve the performance of a trained model when performing the real-time task.

In this article, an improved PCA is proposed to work with logistic regression, which solves the orientation uncertainty of traditional PCA. The model trained via PCA-based logistic regression has a much better performance than that trained via traditional logistic regression and Neural Networks (NN).

Experiments show that the model trained via the proposed method can successfully recognize the oblique handwritten digits that cannot be recognized on current mobile phones, as shown in the case demonstrated in Figure 1. Accordingly, the method's three main advantages are summarized as follows:

1. Better robustness of the trained model. The model trained via the proposed method can recognize a rotated image, which allows the model to work well, despite rotation issues affecting the incoming data.
2. Faster training process and higher real-time recognition frequency. Using PCA to reduce the data dimension, the model can recognize the image using much fewer features, which makes the model easier to train and require less memory storage to perform implementation.
3. Higher accuracy than many other classifiers. The PCA-based logistic regression presented in this article can solve the problem of orientation uncertainty, which leads to the higher accuracy of testing data.
4. Owing to the light computational complexity, the developed handwritten digit recognition algorithm can be realized in an embedded system or integrated into current consumer 3C devices.

Using the proposed PCA-based logistic regression method, the accuracy of testing using the rotated testing set can reach 77.5%. Compared to other algorithms trained via the MATLAB toolbox, the proposed method has a much faster training process and higher testing accuracy. The experimental result shows that the three contributions make the proposed method a potential tool for performing real-time recognition.

In Section 2.1, the research roadmap will be given to enable a clear understanding of the structure of the proposed method. The used methodology will be introduced, as will the processes of recognition. The detail of each method will then be explained in the rest of Section 2. The experimental platform description and associated result are presented in in Sections 3 and 4, respectively. The further comparison studies are considered and discussed in Section 5. Finally, the conclusion of the article is given in Section 6.

2. Methodology

2.1. Description of Research Roadmap

The structure of the PCA-based logistic regression proposed in this article is shown in Figure 2. The proposed method was divided into two parts: data pre-processing and logistic regression. The first part used PCA to straighten the image, smoothened the image using a convolution filter, and reduced the dimension of the whole dataset. The second part was logistic regression, which could efficiently identify the class of each image.

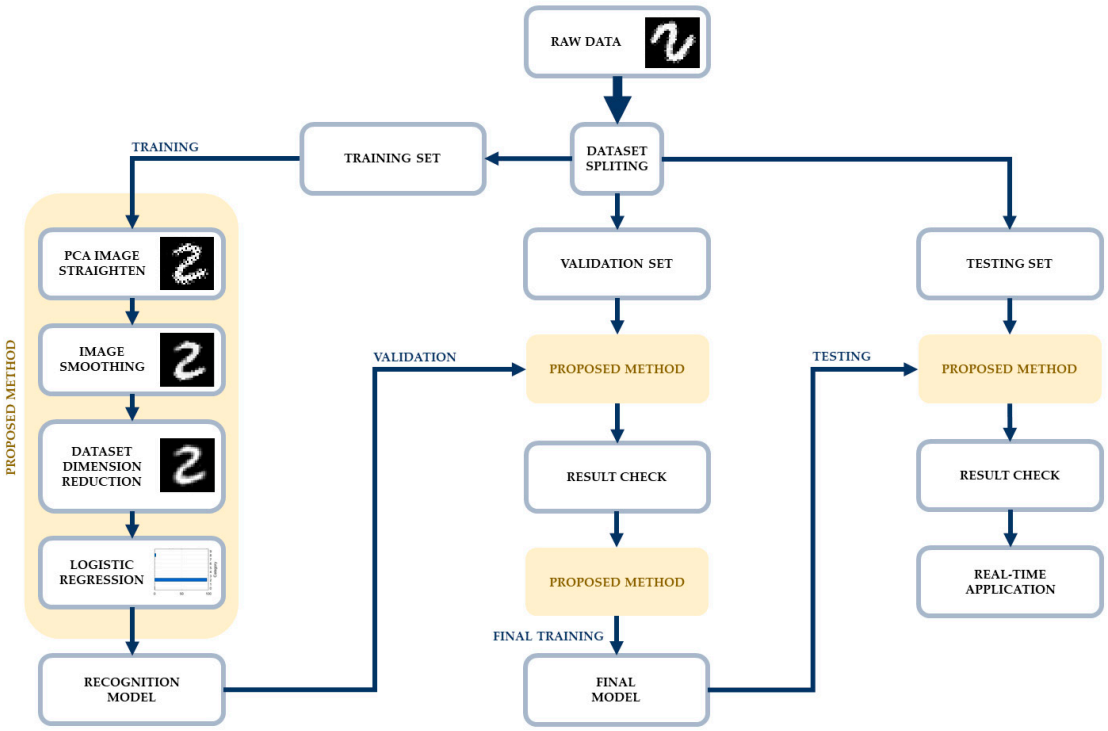


Figure 2. Research roadmap.

To straighten a rotated image, the coordinates of each pixel on the image were treated as data points. By finding the first and second principal components of all of the data points on one image, the two components were used as a new basis. This new basis spanned a subspace onto which the data points were mapped. Once the data points were mapped onto the subspace, the digit image composed of these data points became straight.

However, there was a downside to using data points in the subspace to draw a digit. As the new coordinate of each pixel might not have been an integer after mapping, the data points would deviate from the correct pixel after rounding. This issue led to some noise being present on the straightened digit image. To remove the noise caused by the image straightening, a convolution filter was used in this study. The details on how to identify the type of noise and choose the kernel are described in the methodology section.

Having a straightened digit image, the whole dataset dimension was reduced via PCA. Each image in MNIST had 28×8 pixels, which is a large number for model training. It was extremely inefficient to train a model with all 784 features derived from 60,000 images. To make the training more efficient, this study used PCA to find the principal components

of the whole dataset. The MNIST dataset was mapped onto the subspace spanned by sufficient principal components.

After all of the previous pre-processing, the dataset was used to train a model using logistic regression. The recognition model trained via the proposed method contained the data pre-processing and the matrix from logistic regression. It was tested via a validation set to check the result and determine necessary adjustments. The testing dataset was used to test the model after producing a proficient accuracy of validation. In the experiment, the validation set was not used again to perform testing. The original data were split into a training dataset, validation dataset, and testing dataset. The model after training was first evaluated using the validation set. If the validation accuracy was high enough, the model was then examined using the testing dataset.

The model was eventually used to perform real-time digit recognition. It was proved that the proposed method could be used to recognize different digits in the experiment. This result made the proposed method a potentially highly useful tool for usage in consumer electronics.

2.2. Logistic Regression

Compared to linear regression, logistic regression is a more advanced method with a logic function. The structure of the logistic regression is shown in Figure 3, which shows that the class with the highest probability will be considered to be the class to which the input belongs.

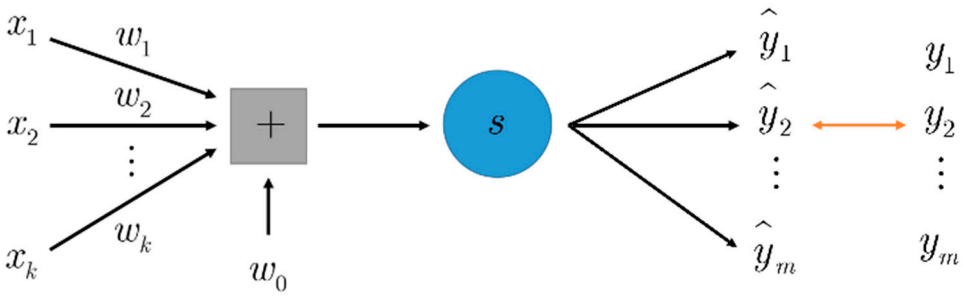


Figure 3. Structure of logistic regression.

The input is multiplied by a weighting matrix with a bias added, which will be placed into a logic function to generate a probability-based output [21]. The mathematic expression can be written as follows:

$$f_{\mathbf{w}_i}(\mathbf{x}) = s\left({\mathbf{w}_i}^T\mathbf{x}\right) = \hat{y}_i \quad , \quad s\left(\mathbf{w}_i^T\mathbf{x}\right) = \frac{e^{-\mathbf{w}_i^T\mathbf{x}}}{\sum\limits_{i=1}^{m} e^{-\mathbf{w}_i^T\mathbf{x}}} \tag{1}$$

To identify the difference between the output and the target, a loss function is required. The difference between two probability distributions can be found using maximum likelihood as follows:

$$L(\mathbf{W}) = \prod_{j=1}^{n}\prod_{i=1}^{m} s\left(\mathbf{w}_i^T\mathbf{x}_j\right)^{y_{ji}} \tag{2}$$

By taking the negative nature log of the likelihood, the equation becomes cross-entropic in nature [22].

$$\begin{aligned} -\ln L(\mathbf{W}) &= -\sum_{j=1}^{n}\sum_{i=1}^{m} y_{ji}\ln s\left(\mathbf{w}_i^T\mathbf{x}_j\right) \\ &= -\sum_{j=1}^{n}\sum_{i=1}^{m} y_{ji}\left(\mathbf{w}_i^T\mathbf{x} - \ln\sum_{i=1}^{m} e^{-\mathbf{w}_i^T\mathbf{x}_j}\right) \end{aligned} \tag{3}$$

In other words, finding maximum likelihood has the same purpose of finding minimum cross-entropy, which is shown in (3). Taking the first order of the a_{th} class that the j_{th} data belong to, it becomes

$$\frac{\partial - \ln L(\mathbf{W})}{\partial \mathbf{w}_a} = -\sum_{j=1}^{n}\sum_{i=1}^{m} y_{ji}\left(\frac{\partial \mathbf{w}_i^T \mathbf{x}}{\partial \mathbf{w}_a} - \frac{\partial \ln \sum_{i=1}^{m} e^{-\mathbf{w}_i^T \mathbf{x}}}{\partial \mathbf{w}_a}\right) \tag{4}$$

The minimum cross-entropy can be found by differentiating (4) on a term-by-term basis as follows

$$\frac{\partial \mathbf{w}_i^T \mathbf{x}_j}{\partial \mathbf{w}_a} = \begin{cases} \mathbf{x}_j & , \quad i = a \\ 0 & , \quad i \neq a \end{cases} \tag{5}$$

$$\begin{aligned} \frac{\partial \ln \sum_{i=1}^{m} e^{-\mathbf{w}_i^T \mathbf{x}_j}}{\partial \mathbf{w}_a} &= \frac{1}{\sum_{i=1}^{m} e^{-\mathbf{w}_i^T \mathbf{x}_j}} \frac{\partial \sum_{i=1}^{m} e^{-\mathbf{w}_i^T \mathbf{x}_j}}{\partial \mathbf{w}_a} \\ &= \frac{\mathbf{x} e^{-\mathbf{w}_a^T \mathbf{x}_j}}{\sum_{i=1}^{m} e^{-\mathbf{w}_i^T \mathbf{x}_j}} = s\left(\mathbf{w}_a^T \mathbf{x}_j\right)\mathbf{x}_j \end{aligned} \tag{6}$$

Notice that $\frac{\partial \mathbf{w}_i^T \mathbf{x}_j}{\partial \mathbf{w}_a} = \mathbf{x}_j$ only at the a_{th} class. Substitute both terms back to the cross-entropy equation as follows:

$$\begin{aligned} \frac{\partial - \ln L(\mathbf{W})}{\partial \mathbf{w}_a} &= -\sum_{j=1}^{n}\sum_{i=1}^{m} y_{ji}\left(\frac{\partial \mathbf{w}_i^T \mathbf{x}_j}{\partial \mathbf{w}_a} - \frac{\partial \ln \sum_{i=1}^{m} e^{-\mathbf{w}_i^T \mathbf{x}_j}}{\partial \mathbf{w}_a}\right) \\ &= -\sum_{j=1}^{n}\sum_{i=1}^{m} y_{ji}\left(\mathbf{x}_j(i=a) - s\left(\mathbf{w}_a^T \mathbf{x}_j\right)\mathbf{x}_j\right) \\ &= -\sum_{j=1}^{n} y_{ja}\left(1 - s\left(\mathbf{w}_a^T \mathbf{x}_j\right)\right)\mathbf{x}_j \\ &= -\sum_{j=1}^{n} \left(y_{ja} - s\left(\mathbf{w}_a^T \mathbf{x}_j\right)\right)\mathbf{x}_j \end{aligned} \tag{7}$$

This equation is the result of the a_{th} class to which the j_{th} data belong. Now, if we take the first-order differentiation of b_{th} class to which the j_{th} data do not belong, the result becomes

$$\begin{aligned} \frac{\partial - \ln L(\mathbf{W})}{\partial \mathbf{w}_b} &= -\sum_{j=1}^{n}\sum_{i=1}^{m} y_{ji}\left(\frac{\partial \mathbf{w}_i^T \mathbf{x}_j}{\partial \mathbf{w}_b} - \frac{\partial \ln \sum_{i=1}^{m} e^{-\mathbf{w}_i^T \mathbf{x}_j}}{\partial \mathbf{w}_b}\right) \\ &= -\sum_{j=1}^{n}\sum_{i=1}^{m} y_{ji}\left(0(i=b) - s\left(\mathbf{w}_b^T \mathbf{x}_j\right)\mathbf{x}_j\right) \\ &= -\sum_{j=1}^{n} y_{ja}\left(0 - s\left(\mathbf{w}_b^T \mathbf{x}_j\right)\right)\mathbf{x}_j \\ &= -\sum_{j=1}^{n} \left(y_{jb} - s\left(\mathbf{w}_b^T \mathbf{x}_j\right)\right)\mathbf{x}_j \end{aligned} \tag{8}$$

where $y_{ja} = 1$ and $y_{jb} = 1$. The result is the same, no matter which class of weight we differentiate. Thus, to perform the update of the matrix of all classes, we have

$$\mathbf{W}_{new} = \mathbf{W} - \eta \cdot \sum_{n=1}^{i} -\left[\mathbf{Y}_j - s\left(\mathbf{W}^T \mathbf{x}_j\right)\right]\mathbf{x}_j \tag{9}$$

The physical meaning is that the weights will be updated according to the error of the output, considering the learning rate. This approach can lead to the weighting matrix that yields the output with high accuracy.

2.3. Principal Component Analysis

In this article, image pre-processing will be focused on applying PCA to each of the digit images to perform image straightening and the whole dataset to perform data dimension reduction. To explain how PCA works with regard to image rotation, the article first employs a two-dimensional image dataset as an illustration example.

Let $\mathbf{P}$ be a two-dimensional dataset, with n data points having two features as follows:

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_n \end{bmatrix} \in \mathbb{R}^{2\times n} \quad , \quad \mathbf{p}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{10}$$

where x_i and y_i represent the pixel coordinate of a digit for the x-axis and y-axis, respectively. The idea of PCA is to find a new basis for a subspace of the space to which the original features belongs, which gives the largest variance of the projected feature.

This demand of maximum variance can be achieved by taking

$$\underset{\alpha_1\in\mathrm{R}^{m\times 1}}{\arg\max}\sum_{i=1}^{n}\left(\alpha_1^T(\mathbf{p}_i-\mu)\right)^2 = \underset{\alpha_1\in\mathrm{R}^{m\times 1}}{\arg\max}\sum_{i=1}^{n}\left(\alpha_1^T\mathbf{p}_{\mathbf{c}i}\right)^2 = \underset{\alpha_1\in\mathrm{R}^{m\times 1}}{\arg\max}\sum_{i=1}^{n}\left(\alpha_1^T\mathbf{p}_{\mathbf{c}i}\mathbf{p}_{\mathbf{c}i}^T\alpha_1\right) \tag{11}$$

where μ represents the mean value of (10), and α_1 represents the 1st principle axis.

Consider the unit vector constraint $\alpha_1^T\alpha_1 = 1$, imposing a Lagrange operator as follows:

$$\underset{\alpha_1\in\mathrm{R}^{m\times 1}}{\arg\max}\sum_{i=1}^{n}\left(\alpha_1^T\mathbf{p}_{\mathbf{c}i}\mathbf{p}_{\mathbf{c}i}^T\alpha_1\right) - \lambda_1\left(\alpha_1^T\alpha_1 - 1\right) \tag{12}$$

where the maximum value can be found by taking the partial derivative with respective to α_1 and λ_1, which leads to

$$\frac{\partial}{\partial\alpha_1}\underset{\alpha_1\in\mathrm{R}^{m\times 1}}{\arg\max}\sum_{i=1}^{n}\left(\alpha_1^T\mathbf{p}_{\mathbf{c}i}\mathbf{p}_{\mathbf{c}i}^T\alpha_1\right) - \lambda_1\left(\alpha_1^T\alpha_1 - 1\right) = 0 \Rightarrow \sum_{i=1}^{n}\left(\mathbf{p}_{\mathbf{c}i}\mathbf{p}_{\mathbf{c}i}^T\alpha_1\right) - \lambda_1\alpha_1 = 0 \tag{13}$$

and

$$\frac{\partial}{\partial\lambda_1}\underset{\alpha_1\in\mathrm{R}^{m\times 1}}{\arg\max}\sum_{i=1}^{n}\left(\alpha_1^T\mathbf{p}_{\mathbf{c}i}\mathbf{p}_{\mathbf{c}i}^T\alpha_1\right) - \lambda_1\left(\alpha_1^T\alpha_1 - 1\right) = 0 \Rightarrow \alpha_1^T\alpha_1 - 1 = 0 \tag{14}$$

Based on (13), it is indicated that

$$\mathbf{M}_c\alpha_1 - \lambda_1\alpha_1 = 0 \tag{15}$$

where $\mathbf{M}_c = \sum_{i=1}^{n}\left(\mathbf{p}_{\mathbf{c}i}\mathbf{p}_{\mathbf{c}i}^T\right)$ is the covariance matrix, which plays a key role in the PCA algorithm.

To calculate the second principal axis, consider the constraints

$$\alpha_2^T\alpha_2 = 1,\ \alpha_1^T\alpha_2 = 0 \tag{16}$$

Equation (16) shows that the principal axes are mutually orthogonal, which guarantees the shape and angle preservations during the rotation.

Based on (16), we considered another maximization problem using Lagrange operators as follows:

$$\underset{\alpha_2\in\mathbb{R}^{m\times 1}}{\arg\max}\alpha_2^T\mathbf{M}\alpha_2 - \lambda_2\left(\alpha_2^T\alpha_2 - 1\right) + \varphi_1\left(\alpha_1^T\alpha_2 - 0\right) \tag{17}$$

Taking the partial derivative of (17) w.r.t α_2 gives

$$2\mathbf{M}\alpha_2 - 2\lambda_2\alpha_2 + \varphi_1\alpha_1 = 0 \tag{18}$$

Multiplying α_1 on both side gives

$$\begin{aligned} &\Rightarrow 2\alpha_1^T\mathbf{M}\alpha_2 - 2\lambda_2\alpha_1^T\alpha_2 + \varphi_1\alpha_1^T\alpha_1 = 0 \\ &\Rightarrow \varphi_1\underbrace{\alpha_1^T\alpha_1}_{1} = 0 \\ &\Rightarrow \varphi_1 = 0 \end{aligned} \tag{19}$$

Therefore, from (15) and (18), we can conclude that

$$\mathrm{M}_c\alpha_1 = \lambda_1\alpha_1 \tag{20}$$

$$\mathrm{M}_c\alpha_2 = \lambda_2\alpha_2 \tag{21}$$

which leads to the standard "eigen-value" and "eigen-vector" problem.

So far, it can be seen that the first and second principal axes satisfy the orthogonal properties. Thus, these axes can be applied for 2D image rotation without causing pixel distortions. As a result, when the two eigen-vectors with the first two largest eigen-values are set to be the projections vectors, it is guaranteed that the two vectors are orthogonal, and the variances after the projection remain the largest.

For high-dimensional conditions, the PCA can be used to perform dimensional reduction. Based on the above derivation, we can find the following extension

$$\begin{bmatrix}\mathbf{M}_c\alpha_1 & \mathbf{M}_c\alpha_2 & \cdots & \mathbf{M}_c\alpha_m\end{bmatrix} = \begin{bmatrix}\lambda_1\alpha_1 & \lambda_2\alpha_2 & \cdots & \lambda_m\alpha_m\end{bmatrix} \tag{22}$$

where

$$\begin{cases}\alpha_i^T\alpha_j = 1 & if \quad i = j \\ \alpha_i^T\alpha_j = 0 & if \quad i \neq j\end{cases} \tag{23}$$

Recalling the singular value decomposition (SVD), we give a data set $\mathbf{P}_c \in \mathbb{R}^{n\times m}$, and we have

$$\begin{aligned}\mathbf{M}_c &= \mathbf{P}_c^T\mathbf{P}_c = \left(\mathbf{U}\Sigma\mathbf{V}^T\right)^T\left(\mathbf{U}\Sigma\mathbf{V}^T\right) \\ &= \mathbf{V}\Sigma^T\mathbf{U}^T\mathbf{U}\Sigma\mathbf{V}^T \\ &= \mathbf{V}\Sigma^T\Sigma\mathbf{V}^T\end{aligned} \tag{24}$$

where $\mathbf{V} := \begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_m\end{bmatrix}$ and $\Sigma^T\Sigma := diag(\sigma_1, \sigma_2, \cdots, \sigma_m)$, which represent the singular vector and singular value of $\mathbf{M}_c$.

Equation (24) can be re-represented by

$$\begin{bmatrix}\mathbf{M}_c\mathbf{v}_1 & \mathbf{M}_c\mathbf{v}_2 & \cdots & \mathbf{M}_c\mathbf{v}_m\end{bmatrix} = \begin{bmatrix}\sigma_1\mathbf{v}_1 & \sigma_2\mathbf{v}_2 & \cdots & \sigma_m\mathbf{v}_m\end{bmatrix} \tag{25}$$

Comparing (25) to (22), it can be seen that the eigen-values of the covariance matrix $\mathbf{M}_c$ are also the singular value of the covariance matrix $\mathbf{M}_c$. Moreover, for (24), the covariance matrix $\mathbf{M}_c$ can be written as the combination of matrices, that is

$$\mathbf{M}_c = \sigma_1\mathbf{v}_1\mathbf{v}_1^T + \sigma_2\mathbf{v}_2\mathbf{v}_2^T + \cdots + \sigma_m\mathbf{v}_m\mathbf{v}_m^T = \textstyle\sum_{i=1}^{m}\sigma_i\mathbf{v}_i\mathbf{v}_i^T \tag{26}$$

Equation (26) represents the essential concept of the matrix significance contribution and can be applied to perform dimension reduction, which will be introduced later in this paper.

Using the first two principal components is the same as using the first two eigenvectors as a new basis to perform a linear transformation. We applied the PCA using the first two principal components of dataset $\mathbf{P}$, and the result is shown in Figure 4.

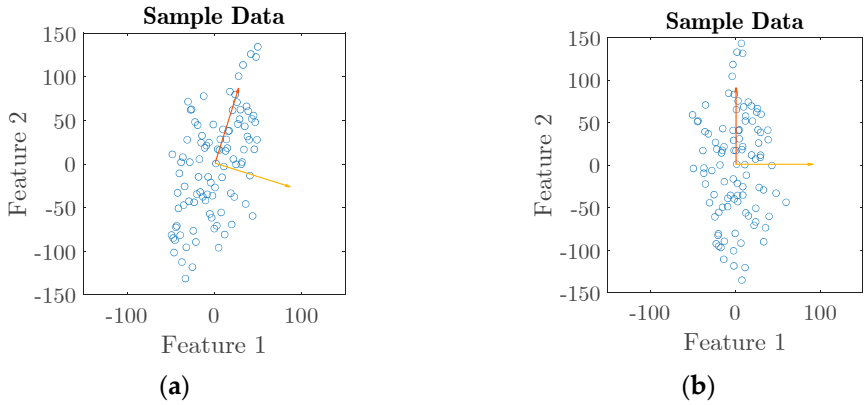


Figure 4. (**a**) is the raw dataset **P**, where (**b**) is the dataset with new basis. Linear transformation determined via setting the first two principal components, the two arrows, as the new basis.

It shows that the data points have been rotated after applying PCA, which is the key property that will be applied to the MNIST dataset to solve the image straightening problem. We noticed that the second principal component needed to be used as the negative basis to prevent the numbers from flipping.

2.4. Image Straighten Using PCA

As shown in the previous section, a two-dimensional dataset can be straightened via PCA. We implemented PCA using the MNIST dataset, and the result is shown in Figure 5. Regarding image rotation of the 2D cases, a computation trick is presented in Appendix A, which can greatly enhance the computation speed and is useful when performing real-time implementation.

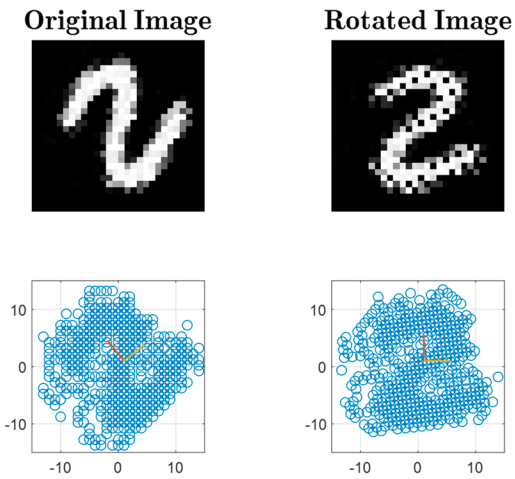


Figure 5. Image straightening of MNIST using PCA.

This result shows that it is possible to straighten the image via PCA if each pixel is viewed as a data point instead of a feature. However, there are two problems involved in using PCA to straighten an image. The first problem is orientation uncertainty. Using principal components found via traditional PCA as the new basis leads to digit straightening according to the direction of the components. However, the direction of the components might not be appropriate for some digits to follow. An improved PCA is proposed in this study, which is described in Section 2.5. The other problem concerns the new data point locations determined via PCA, which are composed of decimals rather than integers. As the image is composed of 28×28 pixels, the locations of the data points need to be integers. This issue causes some noise when the locations are rounded to integers. Therefore, an

appropriate smoothing method needs to be provided to work with PCA to perform image straightening. The image smoothing issue is discussed in Section 2.6.

2.5. PCA Improving

Although PCA can be used to straighten an image, the traditional PCA remains subject to some problems. The examples of the two types of problems are shown in Figure 6.

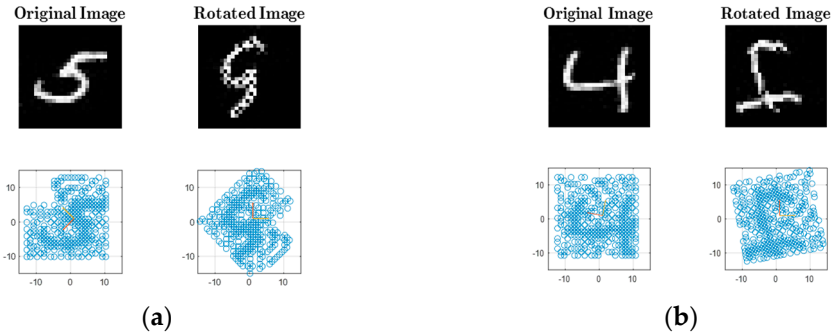


Figure 6. Examples of two main types of data in the error set: (**a**) represents the orientation uncertainty, and (**b**) represents the horizontal digit problem.

The first problem is orientation uncertainty. As the first principal component might not be pointing upward, the straightened image might be rotated. The second problem is horizontal digits. If the digit has a horizontal ellipse shape, the image might be rotated 90 degrees in a leading or lagging manner.

To solve these problems, the first step is to consider the principal component matrix to be a rotation matrix as follows:

$$\mathbf{U} = \begin{bmatrix} u_{1,1} & u_{1,2} \\ u_{2,1} & u_{2,2} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{27}$$

As the matrix $\mathbf{U}$ is the new basis of the projected subspace, the matrix will lead to the first issue if the component $u_{2,2}$ is negative. Therefore, the matrix should be multiplied by the negative component if $u_{2,2}$ is negative as follows:

$$u_{2,2} < 0 \Rightarrow \mathbf{U} = -\begin{bmatrix} u_{1,1} & u_{1,2} \\ u_{2,1} & u_{2,2} \end{bmatrix} \tag{28}$$

On the other hand, the pre-processing will encounter the second issue if the image is rotated more than 60 degrees. According to the practical observations, human-written digits are rarely rotated over 60 degrees. Furthermore, some digits may become similar when the rotation angle is too large, such as the digits "6"and "9". Therefore, it is reasonable to have a rotation angle threshold for the proposed method.

Whether the matrix is 90 degrees leading or lagging can be found based on the value of $\text{atan2}(u_{1,1}, u_{1,2})$. The image is rotated clockwise if the value is positive, which means that the angle of the rotation matrix should be positive, and vice versa.

$$\begin{aligned} \text{atan2}(u_{1,2}, u_{1,1}) > 0 &\Rightarrow \mathbf{U} = \mathbf{U}_{CW} \\ \text{atan2}(u_{1,2}, u_{1,1}) < 0 &\Rightarrow \mathbf{U} = \mathbf{U}_{CCW} \end{aligned} \tag{29}$$

Equation (29) shows how the rotation matrix is corrected. By correcting the matrix encountering the second problem in this way, the image can be correctly straightened. These two matrix corrections improve the performance of PCA. The model using this improved PCA has a 87.15% training accuracy, while the model using traditional PCA only has an 81.22% accuracy.

The proposed improved PCA prevents a normal digit from rotating due to image straightening. It makes the model performance better compared to using traditional PCA.

2.6. Image Smoothing

As mentioned in the previous section, the straightened images have some noises, which worsens the performance of the model. Therefore, it is necessary to remove those noises.

The noises are caused by missing pixels after straightening, as the data points might not be an integer after linear transformation. The rounding of image pixel values from decimals to integers leads to some holes on the digits, such as the rounding results, not being continuous. The holes become noises when the data points are seen as one image.

Considering the causes and distribution of noises, the noise type is very similar to pepper noise, which is often seen in computer vision studies. Therefore, an ideal filter for filtering the image should be a filter that can remove this kind of noise. There are two common filters used to perform image demising: the Gaussian filter and the median filter [23].

The way that convolution works is shown in Figure 7. A kernel, which is also known as filter, needs to be chosen to perform convolution. The kernel matrix is implemented on an image following a certain direction, normally from left to right and top to bottom. The matrix kernel and a part of an image with 3×3 pixels is multiplied on an element-by-element basis. The summation value of the part is used as the pixel value of the new image. For the number of pixels used as a filter, 3×3 pixels is the minimum number required to perform image smoothing. This result shows that the proposed recognition system is able to use fewer parameters to achieve high accuracy in terms of handwritten digit recognition.

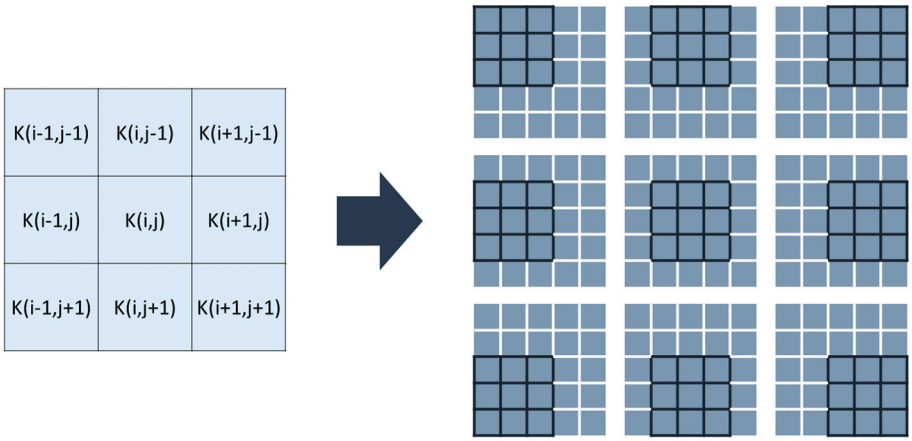


Figure 7. The convolution process of an image.

For a 3 by 3 Gaussian filter, the equation can be written in kernel form as follows:

$$\mathbf{K}_g = \frac{1}{\sum G(x,y)} \begin{bmatrix} G(-1,-1) & G(0,-1) & G(1,-1) \\ G(-1,0) & G(0,0) & G(1,0) \\ G(-1,1) & G(0,1) & G(1,1) \end{bmatrix} \tag{30}$$

where

$$G(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{31}$$

The kernel (30) is implemented on an image and generates the output via convolution. The result of this kernel is a low-pass filter, which works by giving different weights to each pixel to smoothen the image.

$$\mathbf{K}_m = \begin{bmatrix} I_{x-1,y-1} & I_{x,y-1} & I_{x+1,y-1} \\ I_{x-1,y} & I_{x,y} & I_{x+1,y} \\ I_{x-1,y+1} & I_{x,y+1} & I_{x+1,y+1} \end{bmatrix} \tag{32}$$

where $I_{x,y} = median(\mathbf{K}_m)$. For the median filter, the median value of the image that the kernel covered is taken to replace $I_{x,y}$, which is shown as (32). This approach can prevent the image's features from becoming smoothed too much while removing the values that are extremely different from those of its neighbors. Due to this property, the median filter is a more appropriate filter for removing impulse noise.

The image's appearance when filtered using each filter is shown in Figure 8. The image filtered using the median filter retains more features than that using the Gaussian filter. Therefore, this article chooses the median filter to work with PCA to perform image pre-processing.

Figure 8. Result of image smoothing with different filters.

3. Experiment Background

3.1. Data Image

Figure 9 shows the images of MNIST. The MNIST dataset has been well studied in recent years with regard to image recognition, especially handwritten digits. Therefore, the MNIST dataset is utilized to train and validate the proposed method in this paper. In this article, the training data will be divided into a training set and a validation set, which contain 50,000 and 10,000 images, respectively.

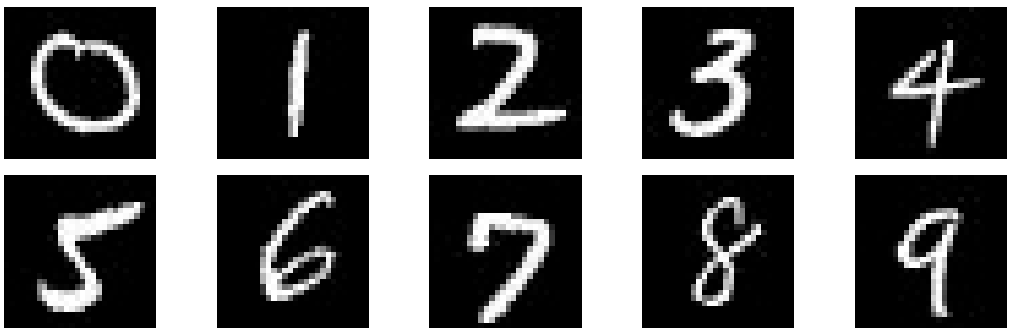


Figure 9. The images of MNIST.

The testing set will only be used to perform the final testing to ensure the objectivity of the testing result. To verify whether the model is able to recognize the rotated digits that are not similar to those of the training set, the testing set is manually rotated. Each of the figures is composed of 28×28 pixels.

3.2. Hardware Specification

The hardware specifications used in the experiment and real-time recognition are shown in Table 1. All of the models are trained using the same computer for a fair comparison. Despite the proposed model being trained via the computer, a much simpler computer or computation unit is sufficient to perform the training of the proposed model.

Table 1. Hardware specification for experiment.

Model	Information
System	Windows 10 22H2
Software	MATLAB 2023a
CPU	11th Gen Intel(R) Core(TM) i7-11800H @ 2.30 GHz
RAM	16 GB
Handwritten Board	Bamboo CTH-470

3.3. Real-Time Recognition System

The model is eventually used to create a real-time handwritten digit recognition system, as shown in Figure 10. The recognition system is built via MATLAB to enable the calculation and user interface, and a handwriting board is used to perform digit writing.

The system is connected as shown in Figure 11. The digit is written by the user using the handwriting board, which is the input of the recognition system. Both the recognition result and the handwritten digit display are calculated using the same computer explanation.

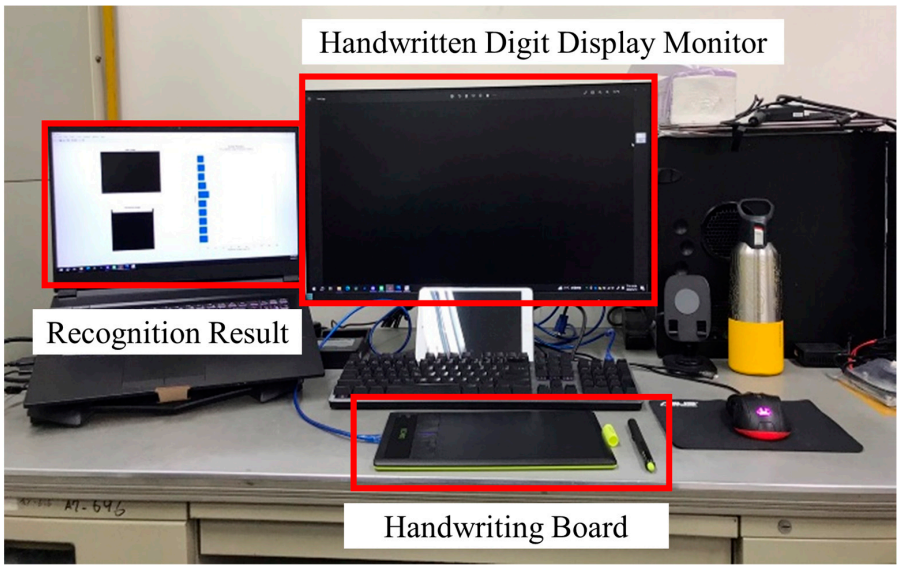


Figure 10. Real-time handwritten digit recognition system.

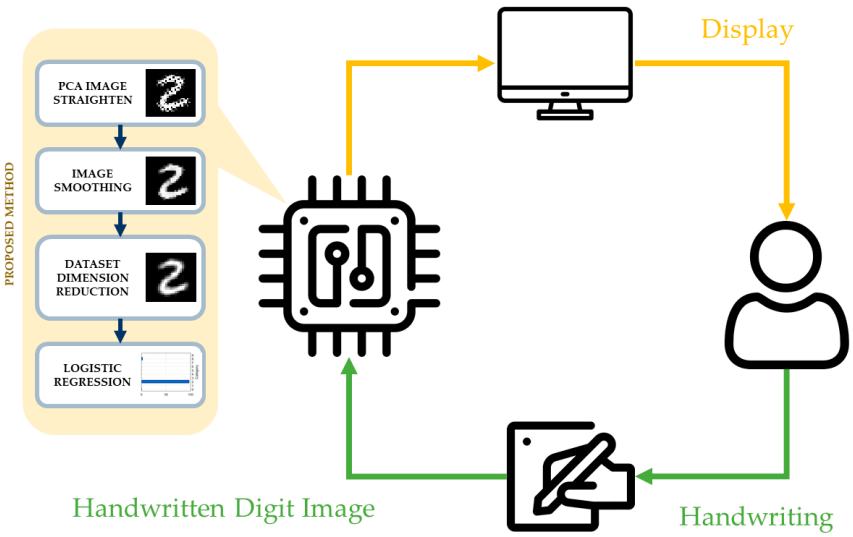


Figure 11. Real-time handwritten digit recognition system connection.

3.4. Comparison Study Platform

The results of other comparison methods shown in Table 1 are conducted via the MATLAB Classification Learner toolbox. The interface can be seen in Figure 12. The MATLAB toolbox is able to train different methods and optimize their recognition results.

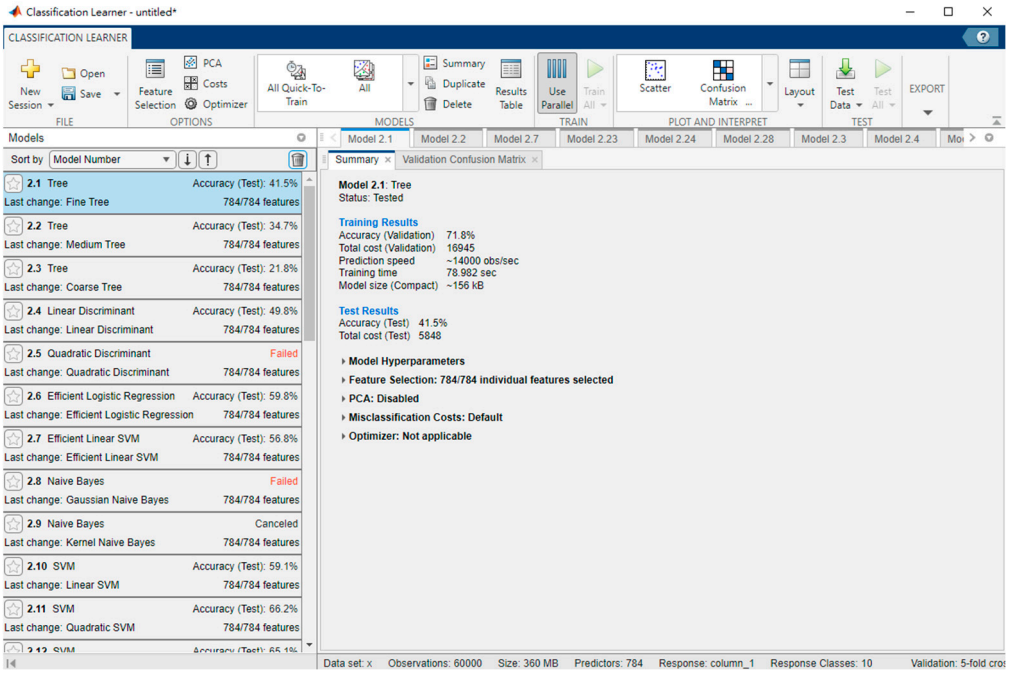


Figure 12. Snapshot of MATLAB Classification Learner.

4. Experimental Result

4.1. MNIST Pre-Processing

Before performing logistic regression, image pre-processing needs to be implemented on the images derived from MNIST. The result of the pre-processed testing data is shown in Figure 13. The pre-processed data have much reasonable digit orientation, which makes the digits easier for the model to recognize.

Figure 13. Image pre-processing of MNIST. The images on the top are the raw images derived from MNIST, while those at the bottom are the pre-processed images.

After image pre-processing, PCA is implemented on the whole dataset to perform dimension reduction. We recall (26), where a data were actually composed of all the principal component. Therefore, the first few principal components with larger singular values are sufficient to perform recognition. The result of the training with different numbers of features is shown in Figure 14. The training result using only 100 features would be as accurate as that using 784 features.

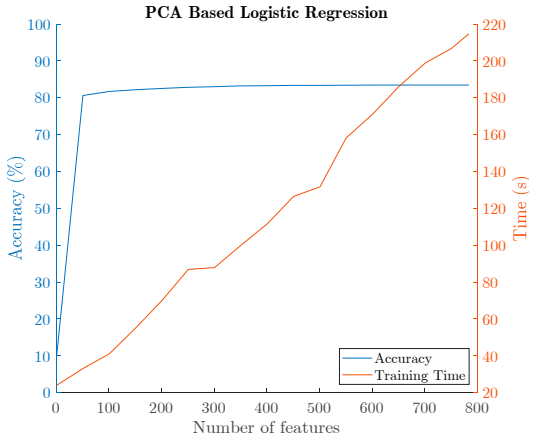


Figure 14. Training accuracy with different data dimensions.

Each of the principal components can be reconstructed to form a 28 by 28 image using each data component, as shown in Figure 15. The result shows the importance of the principal component that drops with the order of it, where the 784th component has nothing at all.

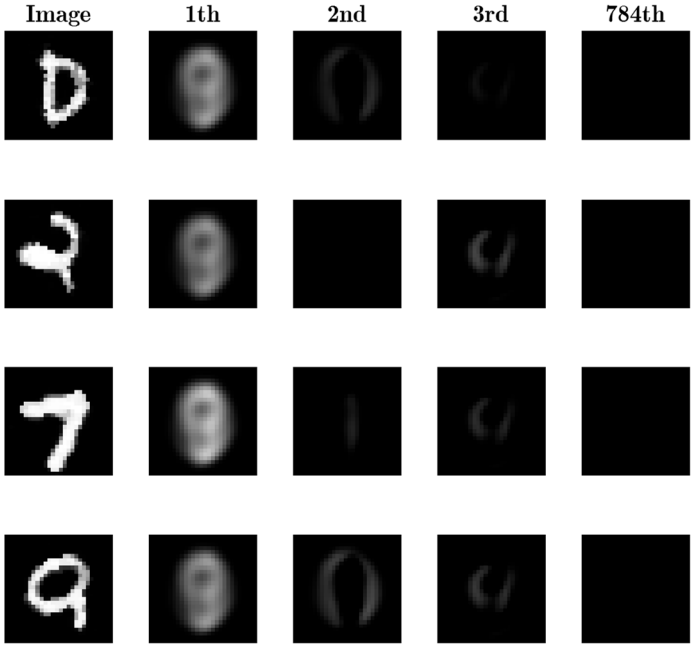


Figure 15. Reconstructed PCA image of different components.

As mentioned, the accuracy of the logistic regression using 100 features is as high as that using 784 features, the input images of which are shown in Figure 16, along with reconstructed images of different dimension subspaces. The m represents the dimension of the projected subspace of the original 784-dimensional feature space.

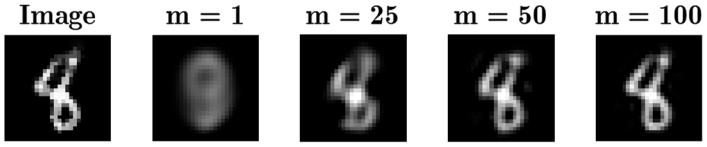
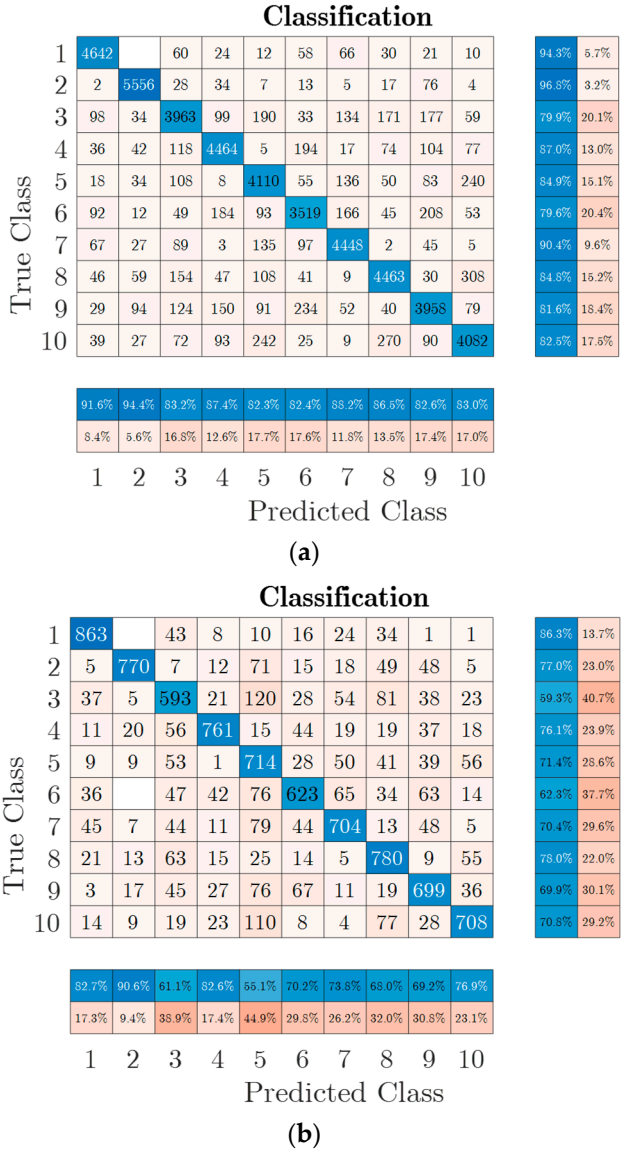


Figure 16. Reconstructed images of different dimension subspaces.

The result shows that the PCA can help to reduce the data to a sufficient dimension, which apparently reduces the training time.

4.2. MNIST Recognition Result

The final training method uses both training and validation sets to perform model training without rotation, while the testing set is rotated manually. The reason for keeping the training set in a non-rotated form is to make sure that all of the compared methods are fairly trained and used. The experimental result is shown in Figure 17.



(**a**)

(**b**)

Figure 17. Confusion tables: (**a**) training result and (**b**) testing result.

The training accuracy is 87.15%, while the testing accuracy is 73.50%. The accuracy is calculated by dividing the number of the successfully recognized images by the total number of MNIST images.

The result shows that even if the model is trained using raw images derived from MNIST, it can still recognize a rotated MNIST digit image. The result shows that the model trained via the proposed method can be used on consumer electronics, which are often required to be robust to the unseen rotated images in the training data.

The model trained via the proposed method can be used to recognize the digit that cannot be recognized on a mobile phone, as shown in Figure 18. The characters in red

blocks represent the recognition results for smart phones, while the number with the largest probability in the bar plot indicates the identification result of the proposed system. The correct digit can be successfully identified by applying the presented method. Therefore, the developed classification scheme is indeed a potential tool for the recognition of handwritten digit subjects to orientation uncertainties and can be further integrated into existing low-cost consumer electronic devices.

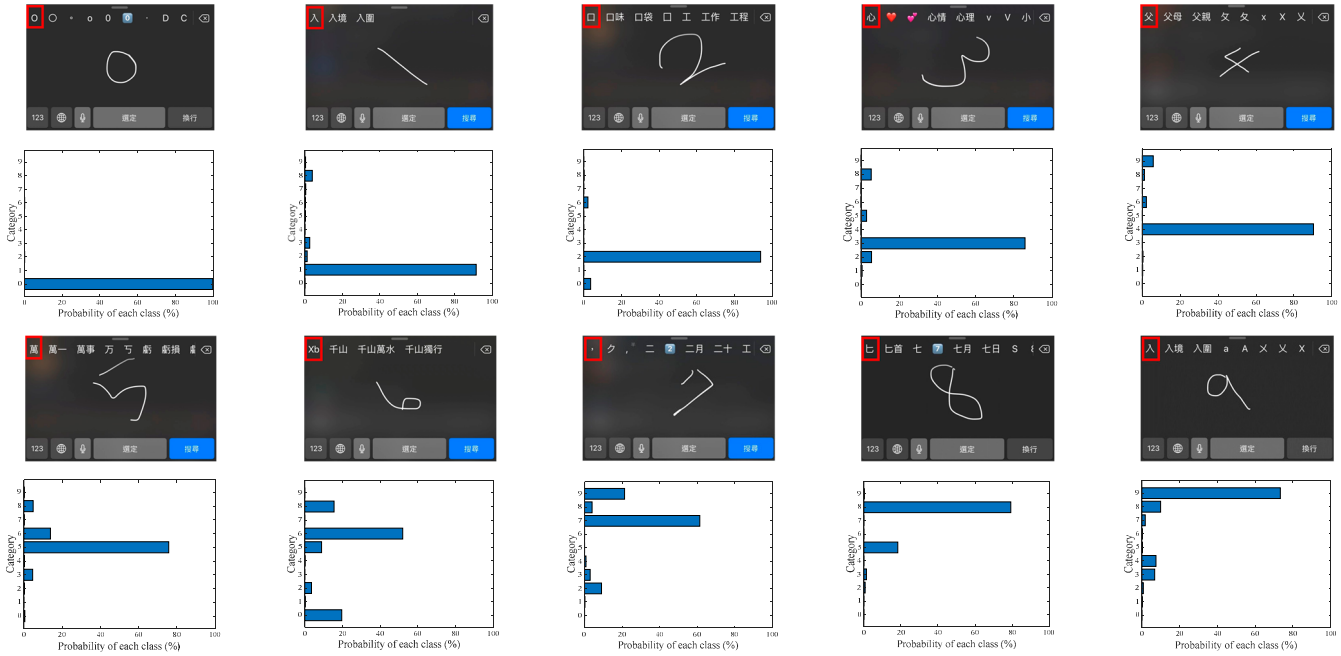


Figure 18. Handwritten digit recognition system comparisons.

5. Comparison Study

To verify how much better the proposed method is than the traditional logistic regression, a logistic regression model without image straightening is used to recognize the rotated images in this study. The testing accuracy of the traditional logistic regression is only 55%. The result shows that the proposed method, the testing accuracy of which is 73.5%, does improve the performance of the rotated digit recognition.

To further validate the effectiveness of the proposed method, different methods are applied to conduct a classification comparison. The associated statistical result is summarized in Table 2. Experiments shows that the PCA-based logistic regression has one of the highest testing accuracies for recognition while using the shortest training time. All of the models are trained using original MNIST training data and tested using rotated MNIST testing data.

Table 2. Accuracy comparison between different methods.

Model	Training Time (Sec)	Validation Acc (%)	Testing Acc (%)	Model Size (kB)
Fine Tree	78.98	71.8	41.5	156
Median Tree	73.60	61.2	34.7	115
Coarse Tree	58.84	36.0	21.8	107
Linear Discriminant	401.89	81.8	49.8	1000
Quadratic Discriminant	Failed			

Table 2. *Cont.*

Model	Training Time (Sec)	Validation Acc (%)	Testing Acc (%)	Model Size (kB)
Efficient Logistic Regression	631.88	83.9	59.8	6000
Efficient Linear Support Vector Machine	432.25	85.9	56.8	10,000
Gaussian Naïve Bayes	Failed			
Linear Support Vector Machine	7890.60	90.0	59.1	6000
Quadratic Support Vector Machine	9528.80	94.3	66.2	220,000
Cubic Support Vector Machine	12,029.00	95.0	65.1	218,000
Fine Gaussian Support Vector Machine	94,873.00	56.3	10.4	3,000,000
Median Gaussian Support Vector Machine	25,055.00	91.7	37.7	493,000
Coarse Gaussian Support Vector Machine	27,872.00	88.7	56.9	543,000
Fine K Nearest Neighbor	7052.60	90.9	61.2	360,000
Medium K Nearest Neighbor	21,222.00	91.2	62.3	360,000
Coarse K Nearest Neighbor	25,057.00	86.6	59.3	360,000
Cosine K Nearest Neighbor	28,134.00	90.8	57.5	360,000
Cubic K Nearest Neighbor	68,775.00	87.6	43.3	360,000
Weighted K Nearest Neighbor	36,165.00	91.5	62.5	360,000
Booster Trees	28,201.00	73.8	45.6	3000
Bagged Trees	47,474.00	91.9	64.2	60,000
Subspace Discriminant	32,226.00	32.3	51.9	75,000
Subspace K Nearest Neighbor	98,531.00	95.2	80.7	5,000,000
RUSBooster Trees	31,726.00	65.3	35.4	3000
Narrow Neural Network	38,601.00	85.8	45.5	181
Medium Neural Network	34,970.00	89.6	49.5	274
Wide Neural Network	36,564.00	94.2	66.7	740
Bilayered Neural Network	43,035.00	85.8	46.1	183
Trilayered Neural Network	43,716.00	86.1	48.2	184
Support Vector Machine Kernel	107,320.00	95.3	75.6	1,600,000
Logistic Regression Kernel	88,626.00	92.6	68.7	1,600,000
PCA-Based Logistic Regression	15.20	87.2	73.5	4628

In Table 2, most of the models cannot recognize the rotated digits as effectively as the proposed method. Except for the proposed model, there are only two models that have accuracies higher than 70%, the training times of which are 6000 times longer than that of the proposed method. The results show that the proposed method has a better recognition accuracy, as it uses much fewer parameters and spends less time on training, meaning that it shows great potential for application in systems with limited storage capacities.

The recognition results of the different rotated handwritten digits are shown in Figure 19. The result shows that the model can successfully recognize the rotated digit

for a real-time application, as the model classifies the digit via an extremely simple but efficient calculation.

As the low-cost recognition system can recognize a rotated digit that has never been trained during the training process, users of the system are able to write the digits without a limited angle restriction.

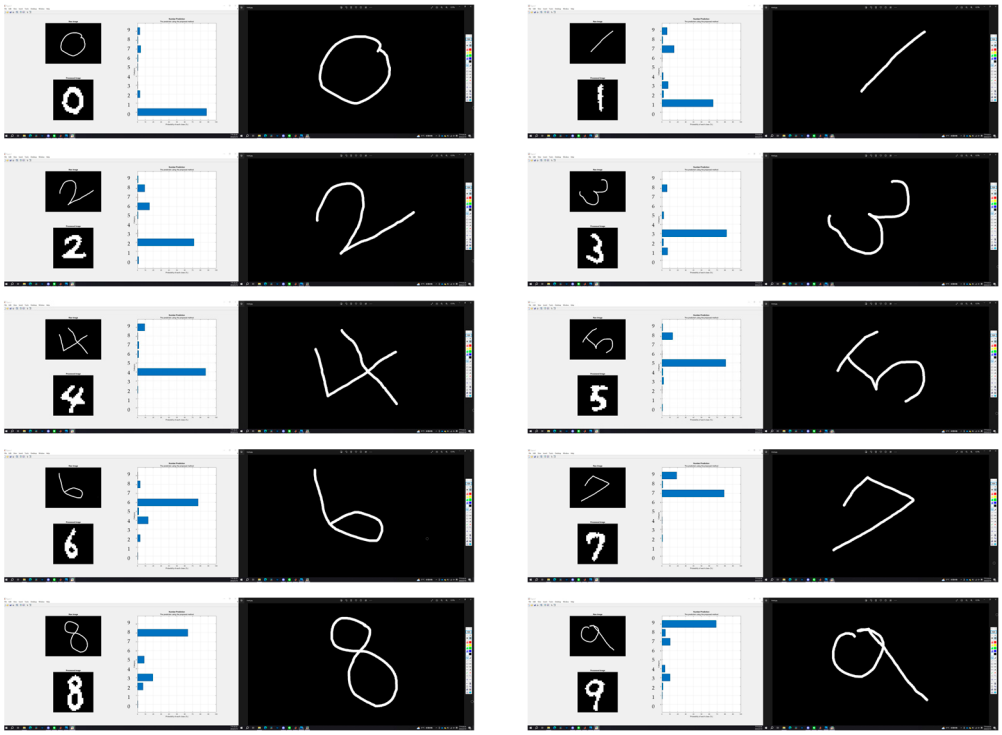


Figure 19. Result of the model's real-time application.

6. Conclusions

Nowadays, the handwritten digit recognition system is widely used in several kinds of 3C consumer electronics, including mobile phones, laptops, and tablet computers. However, when a handwritten number is rotated due to some human factors, the recognition system might fail to identify the digit. Therefore, this article proposes a PCA-based logistic regression to solve the digit rotation problem. The proposed method has four main contributions. Firstly, the PCA is proven to be able to perform image straightening. This ability can help to improve the practicability of the trained model, as the model can be used when the image is rotated. Secondly, the PCA helps to reduce the dimension of the whole dataset without affecting the accuracy. This ability leads to a lower training time and higher detection frequency, which can improve the performance of logistic regression and real-time detection tasks. Thirdly, an improved PCA for image orientation correction was presented, which makes the model's performance better than that of the model using traditional PCA. The improved PCA uses the rotated matrix to decide whether a digit has an orientation problem after pre-processing and revising the rotation matrix. The digits can be more appropriately straightened through this improvement. The first three contributions improve the performance of the model trained via the logistic regression and the process of training it. The proposed method leads to a more robust model that has the highest accuracy and shortest training time among the three compared models.

The main limitation of the proposed method is that the alignment process in the proposed method cannot address the rotation angle of the digits over 60 degrees. However, it is rare to find a digit that is rotated over 60 degrees written by humans. Furthermore, the proposed recognition has a high accuracy with the artificial rotated digit, as seen in

Figure 10 in this paper. Therefore, it is reasonable to have a rotation angle threshold for the proposed method.

Although there are some issues waiting to be solved in the future, such as improving the accuracy, this study shows that PCA is indeed a potential choice to perform data preprocessing to improve logistic regression, especially when the handwritten digits are subject to rotations. Finally, owing to the low computational complexity, the proposed method can be integrated into the current low-cost portable devices, and real-time recognition can be achieved. The attached demonstration (Supplementary Video S1) verified the effectiveness of the developed system.

Supplementary Materials: The following supporting information can be downloaded via the following link: https://www.mdpi.com/article/10.3390/electronics12183809/s1, Video S1: Real-Time Handwritten Digit Recognition.

Author Contributions: Conceptualization, C.-C.P.; Methodology, C.-C.P.; Software, C.-C.P. and C.-Y.H.; Validation, C.-Y.H.; Formal analysis, C.-C.P. and C.-Y.H.; Investigation, C.-C.P. and Y.-H.C.; Writing—review & editing, C.-C.P. and Y.-H.C.; Supervision, Y.-H.C.; Project administration, C.-C.P.; Funding acquisition, C.-C.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by the Ministry of Science and Technology under grant number MOST 111-2923-E-006-004-MY3.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author, upon reasonable request.

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

Appendix A

Since the PCA plays an important role in 2D image pose correction, in the following equation, a computation trick is presented, and it is further used in real-time implementation.

Recall a centralized 2D image data set as follows:

$$\{\mathbf{p}_{c1}, \mathbf{p}_{c2}, \cdots \mathbf{p}_{cN}\} = \left\{ \begin{bmatrix} x_{c1} \\ y_{c2} \end{bmatrix}, \begin{bmatrix} x_{c2} \\ y_{c2} \end{bmatrix}, \cdots, \begin{bmatrix} x_{cN} \\ y_{cN} \end{bmatrix} \right\}^T \subset \mathbf{P}_c \in \mathbb{R}^{N \times 2} \tag{A1}$$

For (A1), define a 2D covariance matrix using

$$\mathbf{S} = \frac{1}{N-1} \mathbf{P}_c^T \mathbf{P}_c \in \mathbb{R}^{2 \times 2} \tag{A2}$$

which can be factorized using SVD as follows:

$$\mathbf{S} = \mathbf{U}_s \Sigma_s \mathbf{V}_s^T \tag{A3}$$

Since $\mathbf{S}$ is a symmetric matrix, it gives $\mathbf{S}^T = \mathbf{S}$, and, therefore, we have $\mathbf{U}_s = \mathbf{V}_s$, where

$$\mathbf{U}_s = \left[\begin{array}{c|c} \mathbf{u}_1 & \mathbf{u}_2 \end{array} \right] = \left[\begin{array}{c|c} u_{1,1} & u_{1,2} \\ u_{2,1} & u_{2,2} \end{array} \right] = \left[\begin{array}{c|c} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{array} \right] \tag{A4}$$

Apparently, there are only two unknown decision variables. These variables are

$$\mathbf{U}_s = \left[\begin{array}{c|c} \mathbf{u}_1 & \mathbf{u}_2 \end{array} \right] = \left[\begin{array}{c|c} u_{1,1} & -u_{2,1} \\ u_{2,1} & u_{1,1} \end{array} \right] \tag{A5}$$

where u_{11} and u_{21} need to be solved. Put simply, $\mathbf{u}_2$ is available as long as $\mathbf{u}_1$ can be solved.

Since $\mathbf{U}_s^T = \mathbf{U}_s^{-1}$, it follows that

$$\begin{aligned} &\Rightarrow \mathbf{S}\mathbf{U}_s = \mathbf{U}_s\Sigma \\ &\Rightarrow \mathbf{S}\left[\begin{array}{c|c} \mathbf{u}_1 & \mathbf{u}_2 \end{array}\right] = \left[\begin{array}{c|c} \mathbf{u}_1 & \mathbf{u}_2 \end{array}\right]\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \\ &\Rightarrow \left[\begin{array}{c|c} \mathbf{S}\mathbf{u}_1 & \mathbf{S}\mathbf{u}_2 \end{array}\right] = \left[\begin{array}{c|c} \sigma_1\mathbf{u}_1 & \sigma_2\mathbf{u}_2 \end{array}\right] \end{aligned} \tag{A6}$$

Equation (A6) shows that the pairs (σ_1, σ_2) and $(\mathbf{u}_1, \mathbf{u}_2)$ are the eigen-values and eigen-vectors of $\mathbf{S}$, respectively.

Let $\mathbf{S} = \begin{bmatrix} s_{1,1} & s_{1,2} \\ s_{1,2} & s_{2,2} \end{bmatrix}$. As a result, the eigen-values (or singular values) can be easily solved by applying

$$\det\left(\begin{bmatrix} \sigma - s_{1,1} & -s_{1,2} \\ -s_{1,2} & \sigma - s_{2,2} \end{bmatrix}\right) = 0 \Rightarrow \sigma^2 - (s_{1,1} + s_{2,2})\sigma + s_{1,1}s_{2,2} - s_{1,2}^2 \tag{A7}$$

which gives

$$\sigma_{1,2} = \frac{\alpha \pm \beta}{2} \tag{A8}$$

where

$$\begin{aligned} \alpha &= (s_{1,1} + s_{2,2}) \\ \beta &= \sqrt{(s_{1,1} - s_{2,2})^2 - (2s_{1,2})^2} \end{aligned} \tag{A9}$$

So far, the eigen-values (or singular values) are available. Next, the corresponding eigen-vectors can be easily solved using

$$(\mathbf{S} - \sigma_1\mathbf{I})\mathbf{u}_1 = 0 \Rightarrow \begin{bmatrix} s_{1,1} - \sigma_1 & s_{1,2} \\ s_{1,2} & s_{2,2} - \sigma_1 \end{bmatrix}\begin{bmatrix} u_{1,1} \\ u_{2,1} \end{bmatrix} = 0 \tag{A10}$$

Through observation, we can find that the solution

$$\mathbf{u}_1 = \begin{bmatrix} u_{1,1} \\ u_{2,1} \end{bmatrix} = \frac{\begin{bmatrix} s_{1,2} \\ \sigma_1 - s_{1,1} \end{bmatrix}}{\left\|\begin{bmatrix} s_{1,2} \\ \sigma_1 - s_{1,1} \end{bmatrix}\right\|} \tag{A11}$$

can be applied to meet the requirement (A10) and, thus, it is

$$\mathbf{u}_2 = \begin{bmatrix} -u_{2,1} \\ u_{1,1} \end{bmatrix} \tag{A12}$$

Obviously, using the aforementioned equivalent representation, the singular value, as well as the singular vector, can be obtained via simple algebra computations without using SVD decomposition. This process can enhance the tool's real-time performance.

References

1. Bishop, C.M.; Nasrabadi, N.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4.
2. Salah, A.A.; Alpaydin, E.; Akarun, L. A selective attention-based method for visual pattern recognition with application to handwritten digit recognition and face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 420–425. [CrossRef]
3. Valdeos, M.; Velazco, A.S.V.; Paredes, M.G.P.; Velásquez, R.M.A. Methodology for an automatic license plate recognition system using Convolutional Neural Networks for a Peruvian case study. *IEEE Lat. Am. Trans.* **2022**, *20*, 1032–1039. [CrossRef]
4. Ahmed, S.S.; Mehmood, Z.; Awan, I.A.; Yousaf, R.M. A novel technique for handwritten digit recognition using deep learning. *J. Sens.* **2023**, *2023*, 2753941. [CrossRef]
5. Aly, S.; Almotairi, S. Deep Convolutional Self-Organizing Map Network for Robust Handwritten Digit Recognition. *IEEE Access* **2020**, *8*, 107035–107045. [CrossRef]

6. Li, J.; Sun, G.; Yi, L.; Cao, Q.; Liang, F.; Sun, Y. Handwritten Digit Recognition System Based on Convolutional Neural Network. In Proceedings of the 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), Shanghai, China, 25–27 August 2020; pp. 739–742.
7. Gonwirat, S.; Surinta, O. DeblurGAN-CNN: Effective Image Denoising and Recognition for Noisy Handwritten Characters. *IEEE Access* **2022**, *10*, 90133–90148. [CrossRef]
8. Li, Y.; Zhang, S.; Wang, W.Q. A Lightweight Faster R-CNN for Ship Detection in SAR Images. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
9. Zhang, X.; Xie, G.; Wang, X.; Zhang, P.; Li, Y.; Salamatian, K. Fast Online Packet Classification with Convolutional Neural Network. *IEEE/ACM Trans. Netw.* **2021**, *29*, 2765–2778. [CrossRef]
10. Watt, J.; Borhani, R.; Katsaggelos, A.K. *Machine Learning Refined: Foundations, Algorithms, and Applications*; Cambridge University Press: Cambridge, UK, 2020.
11. Leong, A.S.; Zamani, M.; Shames, I. A Logistic Regression Approach to Field Estimation Using Binary Measurements. *IEEE Signal Process. Lett.* **2022**, *29*, 1848–1852. [CrossRef]
12. Liu, Q.M.; Ma, C.; Xiang, B.B.; Chen, H.S.; Zhang, H.F. Inferring Network Structure and Estimating Dynamical Process from Binary-State Data via Logistic Regression. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 4639–4649. [CrossRef]
13. Rehman, H.Z.U.; Lee, S. Automatic image alignment using principal component analysis. *IEEE Access* **2018**, *6*, 72063–72072. [CrossRef]
14. Vretos, N.; Nikolaidis, N.; Pitas, I. A model-based facial expression recognition algorithm using Principal Components Analysis. In Proceedings of the 2009 16th IEEE International Conference on Image Processing (ICIP), Cairo, Egypt, 7–10 November 2009; pp. 3301–3304.
15. Garg, I.; Panda, P.; Roy, K. A low effort approach to structured CNN design using PCA. *IEEE Access* **2019**, *8*, 1347–1360. [CrossRef]
16. Akbar, M.A.; Ali, A.A.S.; Amira, A.; Bensaali, F.; Benammar, M.; Hassan, M.; Bermak, A. An Empirical Study for PCA- and LDA-Based Feature Reduction for Gas Identification. *IEEE Sens. J.* **2016**, *16*, 5734–5746. [CrossRef]
17. Zhong, Y.-W.; Jiang, Y.; Dong, S.; Wu, W.-J.; Wang, L.-X.; Zhang, J.; Huang, M.-W. Tumor radiomics signature for artificial neural network-assisted detection of neck metastasis in patient with tongue cancer. *J. Neuroradiol.* **2022**, *49*, 213–218. [CrossRef] [PubMed]
18. Cohen, T.; Welling, M. Group equivariant convolutional networks. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 2990–2999.
19. Yao, C.; Yang, Y.; Yin, K.; Yang, J. Traffic Anomaly Detection in Wireless Sensor Networks Based on Principal Component Analysis and Deep Convolution Neural Network. *IEEE Access* **2022**, *10*, 103136–103149. [CrossRef]
20. Yang, F.; Liu, S.; Dobriban, E.; Woodruff, D.P. How to Reduce Dimension With PCA and Random Projections? *IEEE Trans. Inf. Theory* **2021**, *67*, 8154–8189. [CrossRef] [PubMed]
21. Michalis Titsias RC AUEB. One-vs-each approximation to softmax for scalable estimation of probabilities. *Adv. Neural Inf. Process. Syst.* **2016**, *29*. [CrossRef]
22. Mannor, S.; Peleg, D.; Rubinstein, R. The cross entropy method for classification. In Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 7–11 August 2005; pp. 561–568.
23. Brownrigg, D.R. The weighted median filter. *Commun. ACM* **1984**, *27*, 807–818. [CrossRef]