# Digit Recognition Using Composite Features With Decision Tree Strategy

Chung-Hsing Chen[1,2], Ko-Wei Huang[1] *

[1] Department of Electrical Engineering, National Kaohsiung University of Science and Technology, Kaohsiung City (Taiwan)
[2] Plustek Inc., Taipei City (Taiwan)

Received 11 February 2022 | Accepted 21 September 2022 | Published 14 December 2022



## Abstract

At present, check transactions are one of the most common forms of money transfer in the market. The information for check exchange is printed using magnetic ink character recognition (MICR), widely used in the banking industry, primarily for processing check transactions. However, the magnetic ink card reader is specialized and expensive, resulting in general accounting departments or bookkeepers using manual data registration instead. An organization that deals with parts or corporate services might have to process 300 to 400 checks each day, which would require a considerable amount of labor to perform the registration process. The cost of a single-sided scanner is only 1/10 of the MICR; hence, using image recognition technology is an economical solution. In this study, we aim to use multiple features for character recognition of E13B, comprising ten numbers and four symbols. For the numeric part, we used statistical features such as image density features, geometric features, and simple decision trees for classification. The symbols of E13B are composed of three distinct rectangles, classified according to their size and relative position. Using the same sample set, MLP, LetNet-5, Alexnet, and hybrid CNN-SVM were used to train the numerical part of the artificial intelligence network as the experimental control group to verify the accuracy and speed of the proposed method. The results of this study were used to verify the performance and usability of the proposed method. Our proposed method obtained all test samples correctly, with a recognition rate close to 100%. A prediction time of less than one millisecond per character, with an average value of 0.03 ms, was achieved, over 50 times faster than state-of-the-art methods. The accuracy rate is also better than all comparative state-of-the-art methods. The proposed method was also applied to an embedded device to ensure the CPU would be used for verification instead of a high-end GPU.

## Keywords

Decision Tree, E13B Fonts, Feature Extraction, Image Classification, Multilayer Perceptron.

DOI: 10.9781/ijimai.2022.12.001

## I. Introduction

It is important to determine whether a scanned image for recognition can be used without employing a magnetic ink reader. Several studies on using images for character recognition and handwritten characters have been conducted. However, recently researched architecture, such as deep learning, is not suitable for embedded devices because the high computational resources and elapsed time cannot meet the requirements of a compact embedded system.

### *A. Motivation*

The scanner market has begun to shrink, and scanner manufacturers have turned to special-purpose scanners or readers. Considering the common barcode reader on the market as an example, barcode recognition may be achieved with mobile phones or software; however, in commercial applications, customers prefer a device that does not need to rely on the computing power of the cash register to directly read the barcode. The identification is implemented on the reader. For the cash register, the reader is just a HID (Human Input Device), just like someone helping you type. This application scenario is used on magnetic ink character recognition (MICR) as well. We want to design a device like a MICR reader, which means that we need to complete image recognition and HID output on a single-board controller. This requirement necessitates a simple algorithm that does not require additional GPU or NPU devices—the primary goal of the current research. Assuming that the recognition speed of each character is 50 ms, the total recognition time of one check can be computed to be 1,400 ms. This value is more than one second and hence, does not meet the recognition time requirement. The desired computation time is not achievable using the technologies proposed in recent years. Hence, smaller data features and simpler recognition decisions are needed to accomplish the desired computation time.

We cannot use the image of a character to reduce the amount of information required for recognition. Instead, a method must be developed to reduce the amount of required information for quick calculations. Before using an image as input data for machine learning or deep learning, 1050 pixels need to be processed. This value is

* Corresponding author.
E-mail addresses: I110154101@nkust.edu.tw (C. H. Chen), elone.huang@nkust.edu.tw (K. W. Huang).

greater than the 784 pixels of MNIST's handwritten digital database [1]. Additionally, more explicit features are needed to classify the input data, which can reduce the required amount of input data during classification and facilitate faster decisions and classification. The specifications for checks were defined by the International Standards Organization in 1977, and the latest version is ISO 1004-1:2013, comprising 10 numbers and 4 symbols. E13B is one of the two primary MICR fonts used for printing checks and other payment documents. This font is popular in North America and most of Asia, while the other major MICR font, CMC7, is the standard font used in most of Europe and South America. The E13B font uses characters with unique characteristics designed to produce a distinctive pattern when scanned by a magnetic reader. Both the CMC7 and E13B are read using the unique magnetic characteristics of the font. The characters of E13B are read by detecting the strength of the magnetic signal in a continuous "waveform" pattern from left to right, while CMC7 is read like a barcode, performing a series of "on/off" tests from left to right.

The first and second rows consist of 1, 2, 3, 4, 5, and 6, 7, 8, 9, 0, respectively, as shown in Fig. 1.

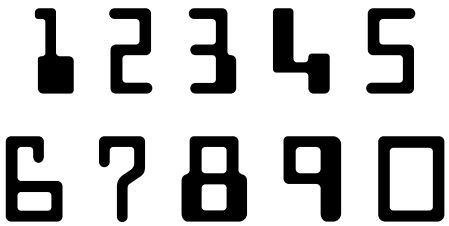


Fig. 1. Illustration of E13B font digits.

The E13B [2] font is specifically defined for MICR, comprising ten digits (0–9) and four symbols. The symbols comprise three separate squares or rectangular symbols with specific arrangements, as shown in Fig. 2.

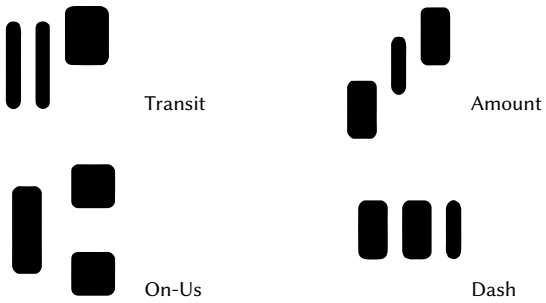



Fig. 2. Illustrations of the E13B font symbols.

In this study, we propose a method to identify all the characters of E13B using feature recognition and relevant correlations to prove the validity and accuracy of the proposed method. Five sets of control groups are applied using the same sample neural network model for training: a four-layer and five-layer MLP [3]–[5] network structure, LeNet-5, Alexnet and hybrid convolutional neural network–support vector machine (CNN-SVM) control groups [6]. Artificial neural networks, such as the relatively new LeNet-5 [7]–[11] have made considerable progress in character recognition.

The neural networks of the five control groups used herein have all ten recognition categories consistent with E13B digital fonts. In this study, we use statistical features [12]–[20] such as image density, pattern, and target relative position features, using a decision tree diagram [21] to achieve classification.

### B. Contribution

A low-energy-consuming algorithm is the main contribution of this research. For the sake of accuracy, advanced research continuously increases the depth of the neural network and the input parameters, but this research direction cannot meet the requirements of special-purpose scanners. Restricted by the hardware of special-purpose scanners, computational requirements and I/O requirements of image data limit the feasibility of neural networks for use in image recognition. Although this study uses a more traditional approach to image recognition, the text features are captured more creatively, and the decision tree is not trained by machine learning but by general logic because of the need to work around the design in cases where there is a limit on computational resource consumption.

We used a laptop with an i5-2520m 2.5Ghz CPU and 4GB RAM for verification. The process in this experiment used the CPU instead of GPU to demonstrate the cost-effectiveness of the proposed method. Our proposed method obtained all test samples correctly, with a recognition rate close to 100%. A prediction time of less than one millisecond per character, with an average value of 0.03 ms, was achieved, over 50 times faster than state-of-the-art methods. The accuracy rate is also better than all comparative state-of-the-art methods.

The remainder of this paper is organized as follows. The literature review is presented in Section II. The research methodology is described in Section III. The performance evaluation is outlined in Section IV. Finally, conclusions and suggestions for future research are provided in Section V.

## II. Literature Review

Text recognition can be divided into several stages. In addition to the pre-processing of the image and the layout segmentation of the text [22], past research papers can be roughly divided into two aspects, one is feature extraction and the other is classifier. Fumitaka and Shridhar used the static Zoning topo to represent the contour of a character [23]. According to the direction of the contour, it is divided into four groups, namely horizontal, vertical, and diagonal in both directions (45° and 135°). The number of contours in each group is its features.

Singh and Budhiraja proposed several features for recognizing handwritten Gurmukhi text, such as projection histogram features, zoning, distance profile features, and background directional distribution features [24].

Verma and Aki published a study on feature extraction methods and classification techniques used in OCR systems [25]. The features used in the study were statistical features and structural features. In their research, statistical feature techniques include zoning, moments, projection histograms, n-tuples, crossing and distances etc. Structural features included convexities, concavities, number of end points, number of holes, etc.

Dimpy et al. proposed a feature extraction of pneumonia data using DensenNet-169 architecture [26]. The original image database was a 3-channel image that was resized from 1024 x 1204 to 224 x 244 pixels. The resizing reduces the need for heavy computation and speed up the processing. After feature extraction through DenseNet-169, a one-dimensional vector of 50,176 x 1 was obtained and input to a different classifier. The findings of this study showed that the best classifier model was SVM (RBF kernel) with an AUC value of 0.80022.

Aimin Yang et al. proposed a feature extraction approach for recognizing tumors by using a local binary model algorithm for image preprocessing [27]. The local binary patten and convolutional neural network algorithm are used to image and extract features from tumor CT images in the medical field, and the recognition rate of this method for medical images was 99.7%.

Lehal proposed a powerful font-independent Gurmukhi OCR system [28]that used four classifiers. The first two classifiers are a binary tree classifier and k-NN classifier. They operate sequentially and use structural features for feature extraction. The third classifier is an SVM using a Gabor filter with a vector size of 189. The fourth classifier is also an SVM, which operates on certain structural and statistical features and achieved a 98.18% accuracy.

Kobayashi et al. proposed the use of histogram of gradient (HOG) to extract candidate features from an image located in a grid. Moreover, they applied principal component analysis to obtain vectors [29]. This method used linear SVM to detect the pedestrian/non-pedestrian and achieved an accuracy of 99.3%.

Singh et al. proposed the use of the Gabor filter for handwritten Gurmukhi character recognition [30]. They performed 5-fold cross-validation on the entire database using the RBF-SVM classifier and achieved 94.29% accuracy.

Shawon et al. proposed a Bengali handwritten letter recognition using 76 features and MLP as a classifier [31]. The feature set developed for recognizing handwritten characters of the Bengali alphabet consists of 24 shadow features, 16 centroid features and 36 longest-run features. The recognition accuracy of the MLP designed to process this feature set was 86.46% and 75.05%, respectively, on the training set and test set samples. This method is useful for the development of a complete OCR system for handwritten Bengali text.

Rajinikanth et al. proposed a meta-heuristic algorithm to solve the multi-thresholding of RGB scale images by using entropy value [32].

Acharya et al. published a new image dataset for the Devanagari script called the Devanagari Handwritten Character Dataset (DHCD) [33]. It consisted of 92,000 images across 46 unique classes of characters of Devanagari script segmented from handwritten files. In addition, they proposed a deep learning architecture (CNN) for the recognition of these characters. The accuracy of the proposed system was 98.47% using the DHCD dataset.

Ramadhan et al presented a comparative analysis of the accuracy and process length of each algorithm [34]. The use of K-Nearest Neighbor (KNN) and Decision Tree (DT) algorithms to detect DDoS attacks was analyzed. In addition, they used the CICIDS2017 dataset, which consists of the world PCAP data format. The results of their study showed that the accuracy of the DT algorithm in detecting DDoS attacks was higher than the KNN value algorithm. The accuracy of DT was 99.91%, while the accuracy of k-NN was was 98.94%.

Assegie et al. proposed a method for recognition of handwritten numbers using a decision tree classification model [35]. Decision tree classification is a machine learning method that uses predefined labels from past known sets to determine or predict classes for future datasets for which the class labels are unknown. They used a standard Kaggle digits dataset to train and recognize handwritten digits using a decision tree classification model. This experiment was trained using a Kaggle dataset containing 42,000 rows and 720 columns. The method had an accuracy of 83.4%. Based on its accuracy, we can see that a decision tree classification model for handwritten number recognition is quite efficient.

Ahlawat et al. proposed a hybrid model of CNN-SVM, a hybrid model of a powerful convolutional neural network (CNN) and supporting machine (SVM) [36], for handwritten digit recognition in the MNIST dataset. A hybrid model of CNN-SVM was proposed for handwritten digit recognition that utilized automatic feature generation of CNN and output prediction using SVM. Experimental results show that their proposed method achieved a classification accuracy of 99.28% on the MNIST dataset.

Barbhuiya et al. proposed a sign language recognition system using Alexnet and VGG16 for feature extraction with SVM as a classifier [37]. The system uses Alexnet and VGG16 to pre-train the American Sign Language dataset, and then uses SVM to pre-train the feature classification, enabling the classification of gestures. The proposed American Sign Language recognition system, has been compared with some state-of-the-art recognition approaches, including both a random 70–30 and leave-one-subject-out validations. The proposed method using the modified AlexNet and SVM classifier has a recognition accuracy of 99.82%, which is the highest among the compared methods using a random 70–30 cross-validation.

Varun et al. proposed a multi-classifier for classifying imbalanced financial news datasets [38]. The proposed architecture uses the SMOTE method to generate similar synthetic samples, which can balance the original imbalanced dataset. Past research has proven that imbalanced datasets have adverse effects on machine learning, thus making the author ensure that the dataset was balanced in his architecture. The proposed architecture was compared with other compared architectures, with the worst recognition accuracy of 34% and the best recognition accuracy of 99%. The proposed architecture achieved a recognition accuracy of 100% with a dataset balanced by random forest with SMOTE.

Manju et al. proposed an RGB and RGB-D static gesture recognition method [39] using the fine-tuned VGG19 model. The fine-tuned VGG19 model uses feature concatenation layers from RGB and RGB-D images to improve the accuracy of the neural network. The proposed model was compared with different CNN models, such as VGG16, CaffeNet, VGG19, and Inception V3, which were not fine-tuned. It achieved 94.8% accuracy using the test results of the ASL dataset. The maximum recognition accuracy of these four models was 88.15%, much lower than that of the proposed model.

## III. Methodology

In this study, 24,000 E13B digits were extracted from 1,079 check samples. The treatment and control groups were used to verify the recognition rates. The treatment group utilized the multi-feature method and directly classified the 24,000 samples to obtain the recognition results. The control group worked with a two-layer and three-layer MLP, LeNet-5, AlexNet and CNN-SVM using the k-Fold cross-validation method, where $k$ was 2, 5, and 10 [40].

### A. Collecting Data

For this study, 1,079 checks were collected from related companies, and 24,000 characters were extracted from them to train the AI model. The study methodology did not require a physical copy of the checks owing to its feature characteristic recognition.

The check is an important accounting document of expenditure or income in the company; it is confidential information within each company. We have legally collected nearly 2,000 Taiwan check samples from our customers through Plustek Inc. After discarding the bad samples, the remaining 1,079 checks were used in this study. These check samples are based on company confidentiality principles, even though Plustek Inc. has eliminated the customer's astute data before providing the check images for this study. In principle, these check images cannot be made public, but images and labels that have been cut into single characters can be provided free of charge for academic purposes or licensed to other for-profit organizations for commercial use.

Using the control group of the neural network of the MLP model, we applied the $k$-fold cross-validation method by dividing the data into $k$ groups. Three groups with $k$ = 2, 5, and 10 were used to validate the accuracy and speed of the proposed method. Fig. 3 represents the checks used in Taiwan; the information on the MICR is shown as the

Fig. 2 that identified as “Transit” check number ”Transit” ”OnUs” bank’s routing number ”OnUs” interchange code ”Transit” account number ”Transit” ”Amount” amount ”Amount”.

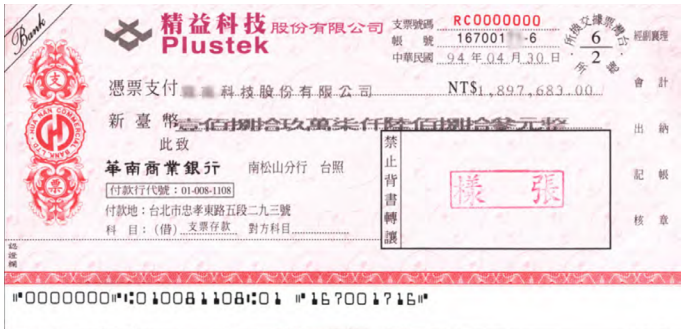



Fig. 3. Checks used in Taiwan. (The amount and the respective symbol are not displayed in this image because the check has not been cashed and processed in the clearing house).

Fig. 4 shows a check used in the United States. The information on the MICR is shown as the Fig. 2 that identified as “Transit” check number “Transit” “OnUs” bank’s routing number ”OnUs” account “Transit” “Amount” amount “Amount”. Although the format of Taiwan’s check is different from that of the United States, it conforms to the ISO 1004-1:2013 standard. In Taiwan, the check format has an additional clearing house code after the bank code because Taiwan uses a computerized exchange system to automatically exchange each clearing house’s notes.


Fig. 4. Checks used in the United States.

## B. Pre-Processing

The following algorithm is used for pre-processing, where the background is separated from the foreground, the foreground is the text, and the background is the non-text area. HTGS(k) is a statistical histogram, whose value can be evaluated using the formula given below, where $k$ is the grayscale ranging from 0 to 255, $p$ represents each pixel, and gray is the grayscale value that utilizes the average of the R, G, and B channels. The HTGS(k) can be calculated as indicated in Eq. (1):

$$HTGS(k) = |\{p \mid k = \lfloor (Gray(p) + LastGray(p))/2 \rfloor, \; 50 < Gray(p) < Mean, \; |Gray(p) - LastGray(p)| > 40\}| \quad (1)$$

where LastGray represents the grayscale value of the previous image and Mean represents the average grayscale value of the image. The obtained value of HTGS(k) is equal to $k_{max}$, which is the binarization threshold of the entire area.

## C. Characters in Digit Recognition

### 1. Statistical Features

The statistical feature of this method utilizes the surface density component to divide an image into four equal blocks: upper left (UL), lower left (LL), upper right (UR), and lower right (LR), as shown in Fig. 5. The density refers to the ratio of the black dots to the total area. The number of black dots in the upper-left block is denoted by nUL. The density in the upper left block (dUL) can be obtained by dividing the quarter of the total area (Width*Height) with nUL. The formulae are as follows:

1. Area = $|(Width/2)| * |(Height/2)|$
2. nUL = black pixel count in Upper Left Block.
3. dUL = nUL/Area; Density in Upper Left Block.
4. nLL = black pixel count in Lower Left Block.
5. dLL = nLL/Area; Density in Lower Left.
6. nUR = black pixel count in Upper Right Block.
7. dUR = nUR/Area; Density in Upper Right.
8. nLR = black pixel count in Lower Right Block.
9. dLR = nLR/Area; Density in Lower Right.

A description of the examined objects and tools used during the experiment, and the factors that could affect the experimental results are discussed in subsequent sections.

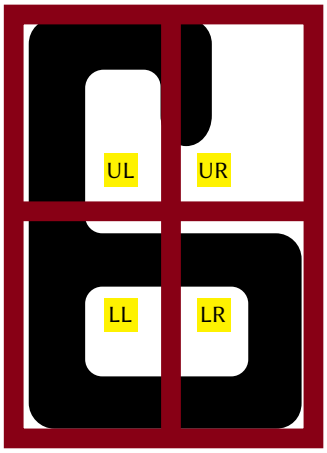



Fig. 5. Schematic of statistical features.

### 2. Pattern Features

The E13B font used in the check is composed of ten numbers ranging from 0 to 9 and four symbols. The numbers and symbols are identified separately in this study. The pattern features were analyzed by calculating the total number of black pixels in an area. Six probes were designed for the digit characters 3, 5, 6, 8, and 0. These feature labels distinguish the five-digit characters, as shown in Fig. 6. $Probe_6$ detects the blank area of the number 6 in the upper-right corner. There is a visible empty block with an E13B print on number 6, which indicates that the block should not contain any black pixels.

Fig. 6. Schematic of geometric features.

In Table I, the coordinates of the rectangles are denoted by (X’, Y’, W’, H’), where X’ and Y’ are the coordinates of the leftmost and highest points of the rectangles, respectively, and W’ and H’ are the widths and heights of the rectangles, respectively. The width and height of image are represented by W and H, respectively. $Probe_0$, $Probe_8$, $Probe_3$-Upper, $Probe_3$-Lower, $Probe_5$, and $Probe_6$ denote the number of black pixels in the six rectangles, respectively.

TABLE I. Reference Labels for Each Coordinate

| Label | Coordinates (X', Y', W', H') |
|---|---|
| $Probe_0$ | W / 3, H / 4, |
| | W - ((W / 3) * 2), W - (H / 4) * 2 |
| $Probe_8$ | (W / 5) * 2, H / 3, W / 5, H / 12 |
| $Probe_3$-*Upper* | 0, (W / 4), W / 2, H / 7 |
| $Probe_3$-*Lower* | 0, (H / 6) * 4, W / 2, H / 7 |
| $Probe_5$ | W / 2, (H / 6), W / 2, H / 6 |
| $Probe_6$ | W - (W / 4), 0, W / 4, H / 4 |

## 3. Decision Tree

The recognition of numbers uses hybrid features from density features and geometric features, based on those decision conditions to create a decision tree model. As shown in Fig. 7, the three numbers 0, 2, 5 have unique features that can be easily classified, while the remaining numbers are divided into two groups using density features. The first group consists of 3, 7, 9, and the second group consists of 1,4,6,8. The detection area of $Probe_0$ is binarized without any black pixels, i.e., the threshold value of $Probe_0$ binarization is calculated using the HTGS method. If the result is not zero, the second set of feature conditions is compared. The number 2 is the only one of the ten numbers whose dLL is larger than its dUL and whose dUR is larger than its dLR. Hence, if the candidate being tested has the above-mentioned features, then the number is classified as 2.

If the result is not 0 or 2, a third set of feature conditions is compared. The next feature condition is the geometric feature $Probe_3$-Lower and $Probe_5$, and both are zero. If the candidate has this feature, then the candidate can be predicted to be number 5.

The fourth set of feature conditions divides the numbers into two groups, that is, (3, 7, 9) and (1, 4, 6, 8). The fourth decision condition is that the dUR must be larger than the dLL. If this condition is met, the number is in the first group. If the condition is not met, the number is in the second group.

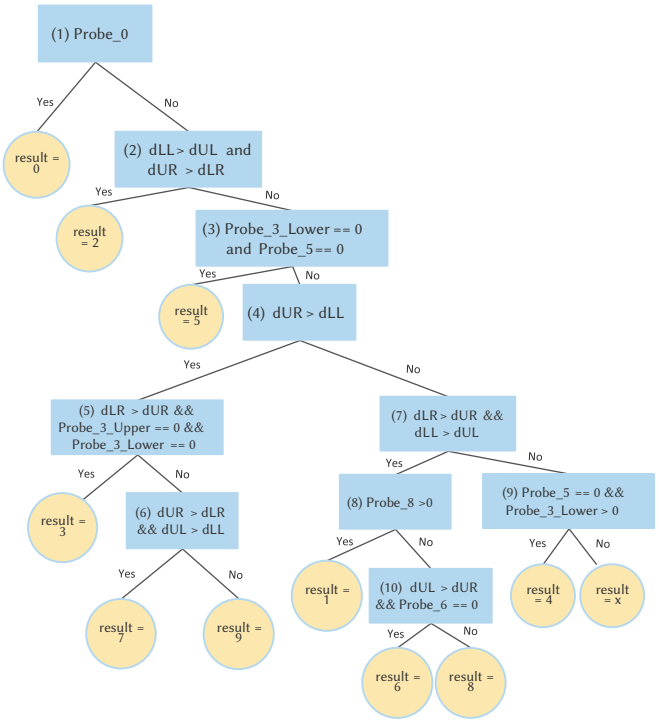



Fig. 7. Decision tree for digit recognition.

The fifth set of feature conditions divide the remaining numbers into two groups. The first group consists of 3 and the second consists of 7 and 9. The condition is that the dLR must be larger than the dUR. The sixth feature conditions are that dUR must be greater than dLR and dUL must be greater than dLL. If the condition is met, the candidate is predicted to be 7 and if not, it is predicted to be 9.

For the group of numbers 1, 4, 6, 8, a seventh set of feature conditions is used. The condition is that the dLR must be larger than dUR and dLL must be larger than dLR. The numbers 1, 4, 6, 8 are divided into two groups, 1, 6, 8 and 4. The next group is 1, 6, 8, and the decision condition is the eighth group of the feature conditions , which is the geometric feature detection area $Probe_8$. If there is a black pixel in $Probe_8$, the candidate can be predicted as number 1. If there is no black pixel, then the number can be either 6 or 8. A tenth set of feature conditions to be passed is that the dUL must be larger than the dUR and the geometric feature detection area $Probe_6$ is zero. If both these conditions are met, the candidate is predicted to be 6 and if not, it is predicted to be 8.

For the last number 4, the ninth set of feature conditions is compared, wherein if the values of the geometric feature $Probe_5$ and $Probe_3$-Lower are zero, the candidate can be predicted to be 4. If this set is not valid, the candidate is not an E13B number and will predicted to be not a digit number that will be classified into the x.

The following illustrates the decision path for each number:

1. Number 0 : (1)
2. Number 1 : (1), (2), (3), (4), (7), (8)
3. Number 2 : (1), (2)
4. Number 3 : (1), (2), (3), (4), (5)
5. Number 4 : (1), (2), (3), (4), (7), (9)
6. Number 5 : (1), (2), (3)
7. Number 6 : (1), (2), (3), (4), (7), (8), (10)
8. Number 7 : (1), (2), (3), (4), (5), (6)
9. Number 8 : (1), (2), (3), (4), (7), (8), (10)
10. Number 9 : (1), (2), (3), (4), (5), (6)

According the pseudo code as showing in Algorithm 1. From the above decision-making, it can be inferred that number 0 is compared once, number 2 is compared twice, number 5 is compared three times, number 3 is compared five times, numbers 1, 4, 7, and 9 need to be compared six times, and numbers 6 and 8 require seven comparisons. Therefore, it can be concluded that the maximum number of comparisons for this decision tree is seven, the minimum number of comparisons is one, and the average number of comparisons is 4.6.

## D. Symbol Recognition

The difference between symbols and digits is that symbols are divided into three targets, whereas digits are not. Therefore, the comparison methods differ. The comparison method applied in this study comprises the following steps. First, the selected symbolic targets are placed in a queue of length 3 from left to right, and the relative positions of targets 1 and 2 are compared. If they are valid, the next target is compared. If not, target 1 is discarded, target 2 moves to the position of target 1, and target 3 moves to the position of target 2. This comparison model test is repeated.

The conditions for symbol comparison are detailed in Table II, and the comparison flow is illustrated in Fig. 8.

**Algorithm 1**: Pseudo code for proposed decision tree

```
Input: statistics features: dUL dLL dUR dLR and
geometric features : Probe0 Probe8 Probe3-Upper Probe3-Lower Probe5 Probe6
Output: classification result of input data
if number of Probe0 == 0 then
    result = "0" ;
else if density in Lower Left > density in Upper Left and density in Upper Right > density in Lower Right then
    result = "2" ;
else if number of Probe3-Lower ==0 and number of Probe5==0 then
    result = "5" ;
else if density in Upper Right > density in Lower Left then
    if density in Lower Right > density in Upper Right and number of Probe3-Upper == 0 and number of Probe3-Lower ==0 then
        result = "3" ;
    else if density in Upper Right > density of Lower Right and density Upper Left > density in Lower Left then
        result = "7" ;
    else
        result ="9" ;
else if density in Lower Right > density in Upper Right and density in Lower Left > density in Upper Left then
    if number of Probe8>0 then
        result = "1" ;
    else if density in Upper Left > density in Upper Right and number of Probe6 == 0 then
        result = "6" ;
    else
        result ="8" ;
else if number of Probe3-Lower >0 and number of Probe5==0 then
    result = "4" ;
else
    something else;
return result
```

TABLE II. Relative Feature Chart

| Item | Conditions |
|---|---|
| Condition 1 | The distance between Target 2 and 1 is greater than 2 and less than 10. |
| Condition 2 | Height of Target 1 is greater than that of Target 2, and height of Target 2 is greater than that of Target 3. Additionally, the width of Target 3 is greater than that of both of Targets 1 and 2. |
| Condition 3 | The correlation height of Target 1 is greater than that of Targets 2 and 3, and the y-axis position of Target 1 is greater than that of Targets 2 or 3. |
| Condition 4 | The relative y-axis distance of Target 3 is less than that of Target 2, and relative y-axis distance of Target 2 is less than that of Target 1. |
| Condition 5 | The correlation height of Target 1 is greater than that of Target 3, and the correlation width of Target 2 is greater than that of Target 3. |

## IV. Experimental Results

The accuracy rate of character recognition was calculated using the above-mentioned characteristics along with the decision-making process. The complete data set consisted of 24,000 E13B characters. The control experiment was conducted with the same data set in Two Layer MLP, Three Layer MPL, LetNet-5 [7], AlexNet [11] and CNN-SVM [36]. The data of this study and the control group were compared to verify the validity and performance of the proposed method.

### *A. Comparison Algorithms*

#### *1. Multilayer Perceptron*

The control group is divided into two groups of multi-layer perceptron hidden layers; in two sets and three sets of hidden layers, respectively. The input is a 28 × 28 pixel image, and uses one applicable channel at a time. There are 784 neurons (1 × 28 × 28 input neurons); each input layer is between 0 and 1, and a pixel value between 0-255 is converted to a floating-point number between 0 and 1. Next, the 28 × 28 matrices are converted into a single layer before input, and the 28 columns are converted into a 784 one-dimensional array, referred to as the first input layer. In the first set of the two hidden layers, as shown in Fig. 9, 256 neurons are arranged in the second hidden layer, and ten neurons in the third hidden layer. The fourth layer acts as the output layer. As shown in Fig. 10, three hidden layers are arranged in the second control group, with 512 neurons in the second hidden layer, 256 neurons in the third layer, and ten neurons in the fourth layer. The fifth layer acts as the output layer. The output consists of digits ranging from 0–9, and thus, according to the softmax function, the final output contains ten neurons.

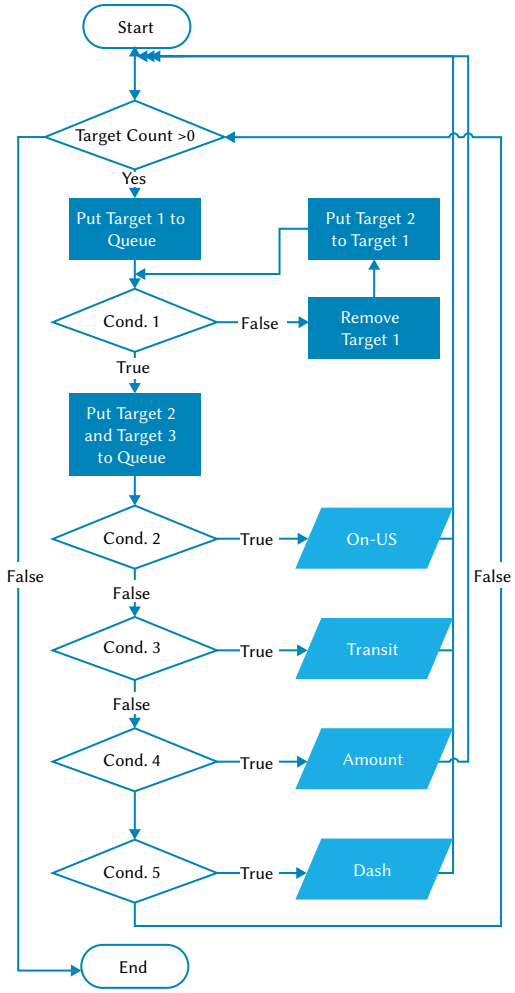



Fig. 8. Flow chart of proposed algorithm.

Neural networks mimic the chain of responses generated by the stimulation of the brain's nerve cell. Transferring neurons between each layer eventually yields an output result.

### 2. LeNet-5

LeNet-5 is a convolutional network algorithm proposed by LeCun in 1998 [7]. This network is also the basis of today's deep learning models. We will use this algorithm as another control group for our experiments. The first layer of LeNet-5 is a 32*32 grayscale image, implying that it is a two-dimensional array with only one channel, which is different from the three channels of an RGB image. The second layer is a 2 x 2 Max pooling layer with a stride of 2. The output of this layer is 14 x 14 x 6. The third layer consists of 16 convolutions of size 5 x 5, with a stride of 1. The fourth layer is a pooling layer that is the same as the second layer with a stride of 2. The output layer is the fifth layer with 120 convolution kernels. The sixth layer is a fully connected layer, and the hidden layer has 84 neural nodes. The last layer consists of 84 hidden nodes corresponding to 10 outputs.

### 3. Alexnet

Alexnet is a CNN proposed by Alex Krizhevsky in 2012 [11] and it won the ImageNet LSVRC competition in the same year. We used this algorithm as another control group for our experiments. The AlexNet architecture has eight layers and uses a total of five convolutional layers and three fully connected layers, which is deeper than the LeNet-5 model. The first to fifth layers are convolutional layers, where a Maxpooling layer of size 3 x 3 and stride of 2 is used after the first, second and fifth convolutional layers . The input layer is larger than

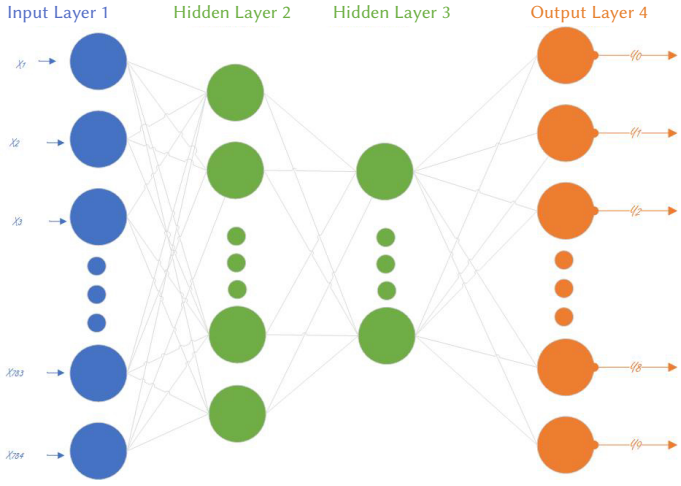



Fig. 9. Two Hidden Layers Multilayer Perceptron Structure Chart.

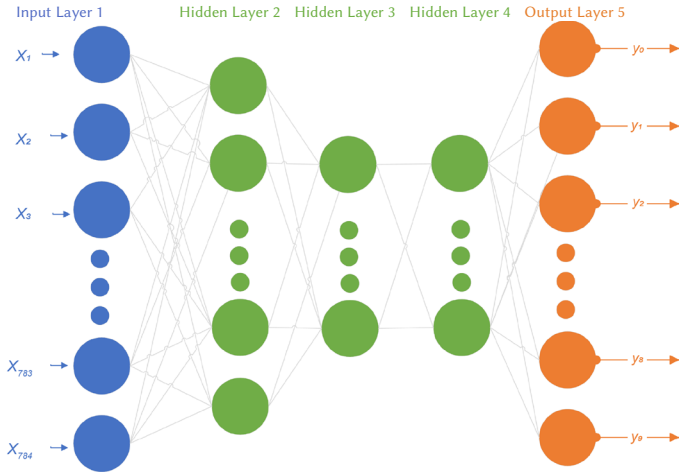



Fig. 10. Three Hidden Layer Multilayer Perceptron Structure Chart.

LeNet-5 and can input 224*224 pixel color images. Unlike LeNet-5, which adopts average pooling, the stride is smaller than the size of the mask, and 2 < 3 can repeat the inspection of features, avoiding important features being discarded during pooling, resulting in better feature calculation results. The sixth, seventh, and eighth layers are fully connected layers. Although the E13B samples in this study were not large images, this method was used as a control group to highlight the comparison of computational power and accuracy.

### 4. Hybrid CNN-SVM

Ahlawat and Choudhary [36] proposed the hybrid CNN-SVM architecture. As this architecture was proposed in recent years, we incorporated this architecture as a control group in our study, in addition to using MLP, LeNet-5, and Alexnet. This architecture uses CNN to extract the features of the image after two convolutions. The input, a 28 x 28 single-channel image, is taken through the first layer using a 5 x 5 filter for convolution operations. The output from this stage is six 24 x 24 feature maps, which are then input to the second layer. Alternatively, the 5 x 5 filter can be used for convolution operations to obtain sixteen 24 x 24 feature maps. A total of 576 neurons were flattened after convolution as a classification feature for the SVM.

## B. Comparison Results

As shown in Table III, 24,000 characters were extracted from the 1,079 check specimens. The number of correct characters recognized in this study was 24,000, whereas the number of incorrect characters

recognized was zero. Table IV shows the results of the two-layer MLP. The recognition rate of the control group at k = 2, 5, and 10 was calculated to be 96.29% on average, while the experimental group recognition rate was 100%. This shows the superiority of the proposed method compared to the two-layer MLP model, which is trained by the artificial neural network in accordance with the accuracy of character recognition.

TABLE III. Treatment Group Accuracy Results

| Test Model | Tests | Errors | Accuracy |
|---|---|---|---|
| Proposed Method | 24,000 | 0 | 100% |

TABLE IV. Two Layer MLP Accuracy Result

| *k* | Total Tests | Number of Errors | Accuracy |
|---|---|---|---|
| 2 | 24,000 | 63 | 99.74% |
| 5 | 24,000 | 1816 | 92.43% |
| 10 | 24,000 | 794 | 96.69% |
| | | Average: | 96.29% |

Table V shows the results of the three-layer MLP. The recognition rate of the control group at k = 2, 5, and 10 was calculated to be 99.64% on an average, while the experimental group recognition rate was 100%. This shows the superiority of the proposed method compared to the three-layer MLP model, which is trained by the artificial neural network in accordance with the accuracy of character recognition.

TABLE V. Three Layer MLP Accuracy Result

| *k* | Total Tests | Number of Errors | Accuracy |
|---|---|---|---|
| 2 | 24,000 | 73 | 99.70% |
| 5 | 24,000 | 125 | 99.48% |
| 10 | 24,000 | 59 | 99.75% |
| | | Average: | 99.64% |

Table VI shows the results of LeNet-5. The recognition rate of the control group at k = 2, 5, and 10 was calculated to be 99.83% on an average, while the experimental group recognition rate was 100%. This shows the superiority of the proposed method compared to the LeNet-5 model, which is trained by the artificial neural network in accordance with the accuracy of character recognition.

TABLE VI. LeNet-5 Accuracy Result

| *k* | Total Tests | Number of Errors | Accuracy |
|---|---|---|---|
| 2 | 24,000 | 120 | 99.5% |
| 5 | 24,000 | 3 | 99.99% |
| 10 | 24,000 | 2 | 99.99% |
| | | Average: | 99.83% |

As shown in Table VII, the recognition rate of the control group at k = 2, 5, and 10 was calculated to be 99.99% on an average, while the experimental group recognition rate was 100%. This shows the superiority of the proposed method compared to the Alexnet model, which is trained by the artificial neural network in accordance with the accuracy of character recognition.

TABLE VII. AlexNet Accuracy Result

| *k* | Total Tests | Number of Errors | Accuracy |
|---|---|---|---|
| 2 | 24,000 | 2 | 99.99% |
| 5 | 24,000 | 4 | 99.98% |
| 10 | 24,000 | 3 | 99.99% |
| | | Average: | 99.99% |

Table VIII shows the results of the CNN-SVM. The recognition rate of the control group at k = 2, 5, and 10 was calculated to be 99.99% on an average, while the experimental group recognition rate was 100%. This shows the superiority of our method compared to the CNN-SVM hybrid model, which is trained by the artificial neural network in accordance with the accuracy of character recognition.

TABLE VIII. CNN-SVM Accuracy Result

| *k* | Total Tests | Number of Errors | Accuracy |
|---|---|---|---|
| 2 | 24,000 | 3 | 99.99% |
| 5 | 24,000 | 2 | 99.99% |
| 10 | 24,000 | 2 | 99.99% |
| | | Average: | 99.99% |

Table IX and Table X show a comparison between the average character recognition time of the treatment and control groups. The average recognition time per character for the treatment group was 0.03 ms, and the average character recognition time for the fastest control group was 1.60 ms per character. The speed of character recognition in the treatment group was significantly higher than that of the control group.

TABLE IX. Elapsed Time for Recognition of This Study

| Test Model | Elapsed time Per Character (ms) |
|---|---|
| Proposed Method | 0.03 |

TABLE X. Elapsed Time for Recognition of the Focus Group

| Test Model | *k* = 2 | *k* = 5 | *k* = 10 | Average |
|---|---|---|---|---|
| Two Layer MLP | 1.54 | 1.58 | 1.67 | 1.60 |
| Three Layer MLP | 1.95 | 1.77 | 1.78 | 1.83 |
| LeNet-5 | 1.79 | 2.73 | 4.08 | 2.87 |
| AlexNet | 66.23 | 63.71 | 68.00 | 65.98 |
| CNN-SVM | 1.25 | 2.73 | 4.72 | 2.9 |

## V. Conclusion

Because E13B is a font for machine reading, traditional feature recognition can be used for stable and convenient font recognition. This approach reduces reliance on hardware, enhances the judgment of exceptions, and reduces the computational cost of the recognition algorithm. Moreover, the method proposed in this study is suitable for embedded platforms or thin clients. A small sample can yield a reasonably high accuracy rate for fixed fonts; the only drawback is that the training process has a long lead time, including real-time corrections of exceptional cases. The aim of this study is to determine the accuracy and speed of character recognition of E13B fonts. The so-called adequate sample eliminates the standard stamps or signature-related targets in the check, and only retrieves E13B. Additionally, filtering noise and partitioning contents were not within the scope of this study and were thus excluded. One limitation of the proposed method is that if the last decision point in the decision tree does not match the feature combination shown in Fig. 7, the output value will be x. However, as x is not a number between 0 and 9, such an output will cause errors during runtime. Using decision trees and comprehensive features has the advantage of speed, but this approach cannot output any confidential information. If there is an exception in the decision tree, it will output x. The addition of exception handling is therefore necessary but can be disadvantageous. In future research, this comprehensive feature can be used for shallow and low-dimensional deep learning. The trained model will be able to perform basic functions of accurate identification with high output accuracy. It will also be faster than general high-dimensional deep networks. The decision tree used in this study also constitutes the identification core of a dual classifier. Such an approach mitigates the drawback of decision tree results that cannot be classified correctly under exceptional conditions and enables better identification.

## Acknowledgement

Plustek Inc. provided the samples for this study at no cost. We thank Bob Lin, General Manager at Plustek Inc., for his constant support, and ADView Technology for providing an Nvidia GPU for E13B model training, making this study possible. In addition, the identification framework of this study has obtained the Republic of China Patent No. M617631. The Plustek iKnow application software and SDK have been imported, which can be used by other software developers for authorization. This work was supported in part by the Ministry of Science and Technology, Taiwan, R.O.C., under the grant ID: MOST 110-2222-E-992 -006 -.

## References

[1] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE signal processing magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[2] X. Corporation, "Generic micr fundamentals guide," Xerox Corporation, 2012.

[3] A. Choudhary, S. Ahlawat, R. Rishi, "A binarization feature extraction approach to ocr: Mlp vs. rbf," in *International Conference on Distributed Computing and Internet Technology*, 2014, pp. 341–346, Springer.

[4] I. B. Cruz, A. Díaz Sardiñas, R. Bello Pérez, Y. Sardiñas Oliva, "Learning optimization in a mlp neural network applied to ocr," in *Mexican International Conference on Artificial Intelligence*, 2002, pp. 292–300, Springer.

[5] A. Choudhary, R. Rishi, S. Ahlawat, "Off-line handwritten character recognition using features extracted from binarization technique," *Aasri Procedia*, vol. 4, pp. 306–312, 2013.

[6] A. F. Agarap, "An architecture combining convolutional neural network and support vector machine for image classification," *arXiv preprint arXiv:1712.03541*, 2017.

[7] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient- based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[8] Z. Zhong, L. Jin, Z. Xie, "High performance offline handwritten chinese character recognition using googlenet and directional feature maps," in *2015 13th International Conference on Document Analysis and Recognition*, 2015, pp. 846–850, IEEE.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[10] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[11] A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[12] N. Sharma, B. Kumar, V. Singh, "Recognition of off- line hand printed english characters, numerals and special symbols," in *2014 5th International Conference- Confluence The Next Generation Information Technology Summit*, 2014, pp. 640–645, IEEE.

[13] I. O. for Standardization, "Information processing — magnetic ink character recognition — part 1: Print specifications for e13b," International Organization for Standardization, 2018.

[14] Y. Yang, X. Lijia, C. Chen, "English character recognition based on feature combination," *Procedia Engineering*, vol. 24, pp. 159–164, 2011.

[15] M. Rani, Y. K. Meena, "An efficient feature extraction method for handwritten character recognition," in *International Conference on Swarm, Evolutionary, and Memetic Computing*, 2011, pp. 302–309, Springer.

[16] S. B. Moussa, A. Zahour, A. Benabdelhafid, A. M. Alimi, "New features using fractal multi-dimensions for generalized arabic font recognition," *Pattern Recognition Letters*, vol. 31, no. 5, pp. 361–371, 2010.

[17] H. Bay, T. Tuytelaars, L. V. Gool, "Surf: Speeded up robust features," in *European conference on computer vision*, 2006, pp. 404–417, Springer.

[18] L. Wang, S. Bi, X. Lu, Y. Gu, C. Zhai, "Deformation measurement of high-speed rotating drone blades based on digital image correlation combined with ring projection transform and orientation codes," *Measurement*, vol. 148, p. 106899, 2019.

[19] K. K. Shreyas, S. Rajeev, K. Panetta, S. S. Agaian, "Fingerprint authentication using geometric features," in *2017 IEEE International Symposium on Technologies for Homeland Security*, 2017, pp. 1–7, IEEE.

[20] T. Kobayashi, "Bfo meets hog: feature extraction based on histograms of oriented pdf gradients for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 747– 754.

[21] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[22] J.-M. Park, C. G. Looney, H.-C. Chen, "Fast connected component labeling algorithm using a divide and conquer technique," *Computers and Their Applications*, vol. 4, no. 20, p. 0, 2000.

[23] F. Kimura, M. Shridhar, "Handwritten numerical recognition based on multiple algorithms," *Pattern recognition*, vol. 24, no. 10, pp. 969–983, 1991.

[24] P. Singh, S. Budhiraja, "Feature extraction and classification techniques in ocr systems for handwritten gurmukhi script–a survey," *International Journal of Engineering Research and Applications*, vol. 1, no. 4, pp. 1736–1739, 2011.

[25] R. Verma, D. J. Ali, "A-survey of feature extraction and classification techniques in ocr systems," *International Journal of Computer Applications & Information Technology*, vol. 1, no. 3, pp. 1–3, 2012.

[26] D. Varshni, K. Thakral, L. Agarwal, R. Nijhawan, A. Mittal, "Pneumonia detection using cnn based feature extraction," in *2019 IEEE international conference on electrical, computer and communication technologies*, 2019, pp. 1–7, IEEE.

[27] A. Yang, X. Yang, W. Wu, H. Liu, Y. Zhuansun, "Research on feature extraction of tumor image based on convolutional neural network," *IEEE access*, vol. 7, pp. 24204–24213, 2019.

[28] G. S. Lehal, "Optical character recognition of gurmukhi script using multiple classifiers," in *Proceedings of the international workshop on multilingual OCR*, 2009, pp. 1– 9.

[29] T. Kobayashi, A. Hidaka, T. Kurita, "Selection of histograms of oriented gradients features for pedestrian detection," in *International conference on neural information processing*, 2007, pp. 598–607, Springer.

[30] S. Singh, A. Aggarwal, R. Dhir, "Use of gabor filters for recognition of handwritten gurmukhi character," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 5, 2012.

[31] A. Shawon, M. J.-U. Rahman, F. Mahmud, M. A. Zaman, "Bangla handwritten digit recognition using deep cnn for large and unbiased dataset," in *2018 International Conference on Bangla Speech and Language Processing*, 2018, pp. 1–6, IEEE.

[32] V. Rajinikanth, S. Kadry, R. González-Crespo, E. Verdú, "A study on RGB image multi-thresholding using kapur/tsallis entropy and moth-flame algorithm," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 5, no. 2, 163-171, 2021.

[33] S. Acharya, A. K. Pant, P. K. Gyawali, "Deep learning based large scale handwritten devanagari character recognition," in *2015 9th International conference on software, knowledge, information management and applications*, 2015, pp. 1–6, IEEE.

[34] I. Ramadhan, P. Sukarno, M. A. Nugroho, "Comparative analysis of k-nearest neighbor and decision tree in detecting distributed denial of service," in *2020 8th International Conference on Information and Communication Technology*, 2020, pp. 1–4, IEEE.

[35] T. A. Assegie, P. S. Nair, "Handwritten digits recognition with decision tree classification: a machine learning approach," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 5, pp. 4446–4451, 2019.

[36] S. Ahlawat, A. Choudhary, "Hybrid cnn-svm classifier for handwritten digit recognition," *Procedia Computer Science*, vol. 167, pp. 2554–2560, 2020.

[37] A. A. Barbhuiya, R. K. Karsh, R. Jain, "Cnn based feature extraction and classification for sign language," *Multimedia Tools and Applications*, vol. 80, no. 2, pp. 3051–3069, 2021.

[38] V. Dogra, S. Verma, N. Jhanjhi, U. Ghosh, D.-N. Le, *et al.*, "A comparative analysis of machine learning models for banking news extraction by multiclass classification with imbalanced datasets of financial news: Challenges and solutions.," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, no. 3, 2022.

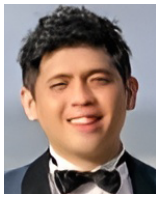
[39] M. Khari, A. K. Garg, R. G. Crespo, E. Verdú, "Gesture recognition of

RGB and RGB-D static images using convolutional neural networks.," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 7, pp. 22–27, 2019.

[40] J. D. Rodriguez, A. Perez, J. A. Lozano, "Sensitivity analysis of k-fold cross validation in prediction error estimation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 3, pp. 569–575, 2009.

Chung-Hsing Chen

C.H Chen received his master's degree at the Department of Information Management, National Sun Yat-Sen University, in 2006. Currently, he is the Director of Research and Development Department of Plustek Inc. His current research interests mainly include, network applications, embedded systems and AI image recognition.



Ko-Wei Huang

Ko-Wei Huang received his PhD from the Institute of Computer and Communication Engineering, Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, in 2015. He is currently an Associate Professor at the Department of Electrical Engineering, National Kaohsiung University of Science and Technology, Taiwan. His current research interests mainly include data mining, deep learning, evolutionary computing, and medical image processing.