

## Research Article

# Recognition of Persian/Arabic Handwritten Words Using a Combination of Convolutional Neural Networks and Autoencoder (AECNN)

Sara Khosravi  and Abdoloh Chalechale 

*Department of Computer Engineering and Information Technology, Razi University, Kermanshah, Iran*

Correspondence should be addressed to Abdoloh Chalechale; [chalechale@razi.ac.ir](mailto:chalechale@razi.ac.ir)

Received 24 January 2022; Revised 3 March 2022; Accepted 7 March 2022; Published 8 July 2022

Academic Editor: Ramin Ranjbarzadeh

Copyright © 2022 Sara Khosravi and Abdoloh Chalechale. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Despite extensive research, recognition of Persian and Arabic manuscripts is still a challenging problem due to the complicated and irregular nature of writing, wide vocabulary, and diversity of handwritings. In Persian and Arabic words, letters are joined together, and signs such as dots are placed above or below letters. In the proposed approach, the words are first decomposed into their constituent subwords to enhance the recognition accuracy. Then the signs of subwords are extracted to develop a dictionary of main subwords and signs. The dictionary is then employed to train a classifier. Since the proposed recognition approach is based on unsigned subwords, the classifier may make a mistake in recognizing some subwords of a word. To overcome this, a new subword fusion algorithm is proposed based on the similarity of the main subwords and signs. Here, convolutional neural networks (CNNs) are utilized to train the classifier. An autoencoder (AE) network is employed to extract appropriate features. Thus, a hybrid network is developed and named AECNN. The known Iranshahr dataset, including nearly 17000 images of handwritten names of 503 cities of Iran, was employed to analyze and test the proposed approach. The resultant recognition accuracy is 91.09%. Therefore, the proposed approach is much more capable than the other methods known in the literature.

## 1. Introduction

Handwritten character recognition systems have recently been used in different areas such as recording and analyzing personal information and administrative forms, helping the blind to read, reading postal addresses of envelopes and sorting them out automatically, and processing bank checks [1]. A review of the literature indicates that recognition results of various printed texts of different languages such as Persian and Arabic have converged on an acceptable rate. However, the existing algorithms and applications are unable to achieve an acceptable accuracy on manuscripts. Thus, more studies should be conducted to design more efficient recognition algorithms.

Considering the available data, text recognition can be either online or offline. Offline recognition of manuscripts is more difficult than online recognition because the latter

benefits from different pieces of information such as the number of strokes, directions of strokes, writing speed in each stroke, and the pressure and duration of each stroke on a surface. These pieces of information are unavailable in offline recognition, which is based on the scanned image of a text [2, 3]. Diversity of writing styles and handwriting is another challenge in the recognition of manuscripts.

Generally, most of the conventional methods of text recognition include three major steps to design an efficient recognition system [4]: preprocessing the input words to enhance the quality of input raw data (through normalization, making images binary, noise elimination, dimension reduction, etc.), feature extraction (including 8-directional feature extraction [5], profiling [6], structural features [7], statistical and geometrical features, etc.), and finally classification through machine learning techniques such as support vector machine (SVM) [8, 9], fuzzy logic [10],

K-Means [11], neural networks (NNs) [12], integration of classifiers [13], and other evolutionary computing techniques. According to the machine vision approach, image classification is based on appearance and feature. In the feature-based method, classification is performed through the features extracted from images [14].

Most of the abovementioned methods are regarded as shallow learning techniques, which usually include a combination of extracted features and trainable classifiers. In such methods, the performance and efficiency of a recognition system depend on the researcher's creativity and innovation. Besides, finding an efficient method for a proper feature extraction has now become an essential issue in the image classification field. Unlike the automated feature extraction methods, most of the common feature extraction techniques are time-consuming and do not show satisfactory result.

On the contrary, there are deep learning (DL) methods [15–18], which have recently attracted many researchers conducting a plethora of studies on them. DL techniques are considered a subcategory of machine learning methods [19], which benefit from high-speed computer processors and a large amount of data to train large-scale NN and solve artificial intelligence problems. These algorithms differ from the abovementioned framework by providing an alternative solution. They have performed well in text recognition without using any feature extraction or preprocessing techniques. The distinct advantage of deep learning methods over shallow learning techniques is the automated extraction of features without considering the information. Therefore, DL methods can discover complicated structures and distinguish between large-scale datasets [20].

The first DL architectures for computer vision were implemented using artificial neural network (ANN) in the 1980s [21]. Recently, DL has been applied in many artificial intelligence (AI) applications; among its usages, we can mention emotion recognition [22, 23], coral classification [24], detecting robotic grasps [25, 26], natural language processing [27], biometric authentication based on finger knuckles and fingernails [28], brain tumor segmentation [29], and various usages in computer networks such as network traffic classification [30] and network intrusion detection systems [31]. The most important aspect of the DL is automatic feature extraction and using these features in the next layers of the network [32]. Some of exploited types are the autoencoders (AEs), the convolutional neural networks (CNNs), the recurrent neural networks (RNNs), the recursive neural networks, and the deep belief networks. This study focuses on the CNN and the AE.

The convolutional neural network is inspired by human learning strategy. The human brain recognizes objects visually whereby, examining similar images of an object, it obtains the ability to recognize objects it has not seen before. The ability to automatically extract important features of an object leads to high performance in CNN. Applications of CNNs include in-scene text recognition to obtain image encrypted information [33] and improving handwritten mathematical symbols classification [34]. Moreover, Steganalysis on JPEG images was introduced in [35]. This allows

identifying hidden embedding information without calculating the predefined image properties utilizing an experimental-based procedure.

An autoencoder is a special type of ANN which is used for optimal encoding in the learning process. Instead of training the network and predicting the target value for a particular input, an autoencoder rebuilds its inputs. Therefore, the output vector will have the same dimension as the input vector [36]. Among the usages of the autoencoders, we can mention deep feature learning for the medical image analysis [37], handwritten recognition [38, 39], and anomaly detection [40, 41].

In recent years, there has been research using a combination of CNNs and autoencoders. Dastider et al. [42] presented a framework to predict the disease severity of COVID-19 based upon the integration of CNNs and AEs. They considered both spatial and temporal features of the lung ultrasound (LUS) frames. Besides, deep convolutional autoencoder (CAE) has been proposed by Seyfioglu and Gurbuz [43]. This research is for radar-based activities recognition such as border security and control, pedestrian identification for automotive safety, and remote health monitoring.

The proposed recognition system is based on interconnected components. The words are first decomposed into their constituent subwords to enhance the recognition accuracy. Then the signs of subwords are separated to develop a dictionary of main subwords and signs. In the proposed system, a CNN is employed to categorize different classes. In such networks, the feature extraction of input images is performed automatically through the filter coefficient defined on the entire image. The filter coefficient of the entire image generates unnecessary and redundant features. Therefore, an autoencoder network is employed, prior to the CNN classifier, to extract effective features to improve the generated features. The CNNs are utilized to train the classifier. An autoencoder network is employed to extract appropriate and necessary features; thus, hybrid network was developed and named AECNN. After performing preprocessing and segmentation operations on a word, the recognition of subwords is done in the testing phase. Then the subwords are integrated. The integration of generated subwords should be based on how much the dictionary words resemble a corresponding word. A powerful algorithm is proposed for subword fusion based on the similarity to the error modification approach. Our contribution can be considered in three categories:

- (1) A combination of the CNNs and the AEs, which is called AECNN for the recognition of Persian and Arabic words.
- (2) Creating a dictionary of the main subwords and signs to increase the recognition accuracy.
- (3) Proposing a new subword fusion algorithm based upon the similarity of the main subwords and signs.

In this paper, Section 2 includes the writing features of Persian words. Section 3 addresses the proposed convolutional neural network and autoencoder based on deep

learning. Section 4 discusses the proposed feature extraction approach through an AE network, a CNN architecture, and an AECNN in full details. Section 5 deals with the research dataset. The results of simulating the proposed method are analyzed and compared with those of other studies in Section 6. Finally, Section 7 presents the conclusion and provides some strategies for future studies.

## 2. Writing Features of Persian/Arabic Words

Regarding the selection of appropriate methods for text recognition, it is necessary to learn Persian (or Arabic) writing rules. Hence, the following includes brief explanations of writing features in Persian/Arabic. For more clarity, in the following the English equivalents of Persian/Arabic words and characters are given in parentheses.

- (i) Unlike Latin texts, Persian/Arabic texts start from right to left and are written on the contour line, which is a horizontal line along the connected parts of the text having usually the largest number of word pixels.
- (ii) Persian consists of 32 characters, whereas Arabic contains 28 characters. Depending on the positions of characters, they appear in four different forms (at the beginning, in the middle, at the end, and separate). Table 1 indicates the different forms of Persian and Arabic alphabets with respect to their positions in words.
- (iii) In some writing styles of Persian texts, two or more letters are merged in a way where the resultant form looks nothing like the constituent letters. For instance, the combination of “ل” and “ا” is shown as “لا.”
- (iv) The only factor differentiating between the shapes of similar letters such as {“ث”, “ت”, “پ”, “ب”} (Se, Te, Pe, and Be), {“ح”, “خ”, “ج”, “چ”} (Khe, He, Che, and Jim), {“ذ”, “ز”} (Dal and Zal), and {“ر”, “ز”, “ز”} (Zhe, Ze, and Re) is the presence or absence of dots as well as the number of position of dots.
- (v) There are dots in 18 Persian characters and 15 Arabic characters. These dots may appear above or below the contour line. The number of these dots varies from one to three.
- (vi) In Persian/Arabic, words are divided into smaller units called subwords. In fact, these subwords are joined together to form the words. Figure 1 shows some samples of words with different subwords.
- (vii) Letters “ز”, “چ”, “پ”, and “گ” (Pe, Che, Zhe, and Gaf) exist in the Persian alphabet; however, the Arabic alphabets lack them.

## 3. Deep Networks

Deep learning is a subcategory of machine learning. The DL algorithm has a key role in object recognition. The following subsections address two deep learning techniques briefly.

**3.1. Convolutional Neural Networks.** Convolutional neural networks are known as one of the most famous deep

TABLE 1: Persian characters in four different positions.

Character	Characters positions			
	Beginning	Middle	End	Separate
1 (Alef)	ا	ا	ا	ا
2 (Be)	ب	ب	ب	ب
3 (Pe)	پ	پ	پ	پ
4 (Te)	ت	ت	ت	ت
5 (Se)	ث	ث	ث	ث
6 (Jim)	ج	ج	ج	ج
7 (Che)	چ	چ	چ	چ
8 (He)	ح	ح	ح	ح
9 (Khe)	خ	خ	خ	خ
10 (Dal)	د	د	د	د
11 (Zal)	ذ	ذ	ذ	ذ
12 (Re)	ر	ر	ر	ر
13 (Ze)	ز	ز	ز	ز
14 (Zhe)	ژ	ژ	ژ	ژ
15 (Sin)	س	س	س	س
16 (Shin)	ش	ش	ش	ش
17 (Sad)	ص	ص	ص	ص
18 (Zad)	ض	ض	ض	ض
19 (Ta)	ط	ط	ط	ط
20 (Za)	ظ	ظ	ظ	ظ
21 (Ayn)	ع	ع	ع	ع
22 (Ghayn)	غ	غ	غ	غ
23 (Fe)	ف	ف	ف	ف
24 (Ghaf)	ق	ق	ق	ق
25 (Kaf)	ک	ک	ک	ک
26 (Gaf)	گ	گ	گ	گ
27 (Lam)	ل	ل	ل	ل
28 (Mim)	م	م	م	م
29 (Noon)	ن	ن	ن	ن
30 (Waw)	و	و	و	و
31 (He)	ه	ه	ه	ه
32 (Ye)	ی	ی	ی	ی

learning methods. CNNs were developed by LeCun et al. earlier than two decades ago (1990) [44]. They were then developed by other researchers. A CNN is a multilayer neural network which can benefit simultaneously from an automated feature extractor and trainable classifier [45]. In recent years, CNNs have most widely been used in solving machine vision problems such as pattern recognition, object detection, or speech recognition [1]. They have also been widely used in Chinese handwritten text recognition [46]. Nevertheless, researchers are still trying to achieve a rapid and efficient training process and quick classification for CNNs.

CNNs outperform other deep learning methods in managing big data and sharing their weights [23]. These networks include a powerful learning technique by utilizing a multilayer structure and a specific type of supervised feedforward networks, in which layers are arranged differently for different applications. The back-propagation (BP) algorithm is usually employed to train and adjust the network parameters [47, 48].

Generally, a CNN consists of three major layers: convolutional layer, pooling layer, and fully connected layer. All of these layers are interconnected through weights. In this network, the convolutional layer and the pooling layer act as

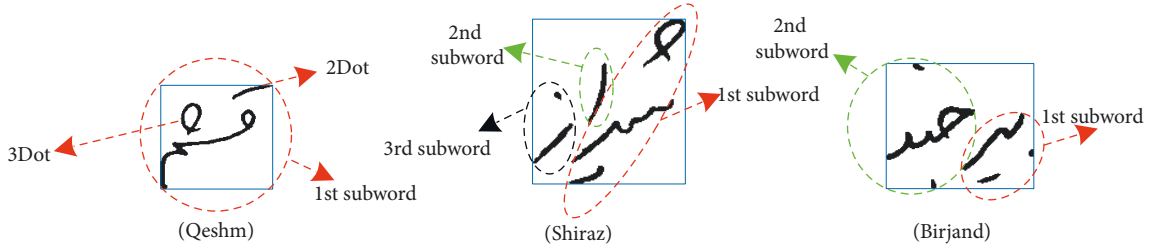


FIGURE 1: Some samples of words with different subwords.

feature extractors, whereas the fully connected layer acts as a classifier.

These layers consist of too many parameters, increasing the computing load and making the training process time-consuming. So far, some solutions have been provided for this problem. One solution is to reduce the number of connections in layers by employing basic CNN models such as GoogLeNet and AlexNet, in which the layer structure differs from the common type. The network depth can increase by adding more convolutional layers and benefiting from small convolutional filters [19].

There have also been some hybrid models, benefiting from the integration of shallow learning methods with deep learning techniques. In such models, features are usually extracted through a deep learning method. Then the differentiated features are given to a shallow learning technique such as an SVM for classification [19]. An application of such hybrid methods was discussed in the paper reviewed by [49]. In that paper, text detection was performed by integrating a CNN with RNN. In another study, a CNN was integrated with the Markov hidden model (HMM) for feature extraction in the recognition of handwritten words (based on two types of segmentation through letters and the sliding window) [50]. It was shown that features extracted through a CNN were more efficient than hand-crafted features in recognition. In the paper reviewed by [51], the particle swarm optimization (PSO) was integrated with a stochastic gradient descent (SGD), used as an alternative solution to optimize and enhance the network training process, based on the idea that the BP algorithm is prone to certain limitations on CNN training.

**3.2. Autoencoders.** An autoencoder is a type of neural network, in which the network input is the same as the network output. It also benefits from an unsupervised learning technique SGD for training [52, 53]. The autoencoder network consists of an encoder and a decoder. The encoder maps the input data  $x \in [0, 1]^{n+1}$  onto a latent coded space  $y \in [0, 1]^{m+1}$ , in which  $m$  is smaller than  $n$  most of the time. The encoder output is obtained through the following equation [53, 54]:

$$y = f(Wx + b). \quad (1)$$

In this equation,  $W$  and  $b$  are the weight matrix and mapping bias, respectively. Moreover,  $f$  is the activator function, which can be either linear or nonlinear. Then the decoder generates the reconstructed data  $\hat{x} \in [0, 1]^{n+1}$  by

mapping the latent coded space onto the main input space through the following equation:

$$\hat{x} = f(W'y + c). \quad (2)$$

In this equation,  $W'$  and  $c$  are named the weighted matrix and reverse mapping weight bias, respectively. The autoencoder tries to converge  $x$  on  $\hat{x}$  by adjusting weights and biases. The difference between  $x$  and  $\hat{x}$  is called the loss function, the value of which can be minimized to train the network [54].

For the retrieval of the input signal, an autoencoder should extract the important features of the input signal from the autoencoder output. Furthermore, if the number of autoencoder units is smaller than the input signal size, it is possible to present a compressed and low-dimensional display of the high-dimensional input signal [52].

#### 4. The Proposed Recognition System

Figure 2 shows a schematic view of the proposed recognition system, in which the training phase includes all of the images undergone through preprocessing steps including completion, noise elimination, and image size normalization.

The proposed recognition system is based on interconnected components. Since the dataset images are words, a segmentation step is required to divide words into interconnected components, which are hereinafter referred to as subwords. After this step, the subwords are labeled to generate a dictionary of subwords for network training (as shown in the following equation):

$$L = \{s_1, \dots, s_i, \dots, s_N\}. \quad (3)$$

Accordingly,  $s_i$  is the  $i$ th subword in the dictionary set  $L$ , and  $N$  indicates the total number of subwords.

In the proposed system, a CNN was employed to categorize different classes. In such networks, the feature extraction of input images is performed automatically through the filter coefficient defined on the entire image. The filter coefficient of the entire image generates unnecessary and redundant features. Therefore, an autoencoder network was employed, prior to the CNN classifier, to extract effective features to improve the generated features. The dimensions of features generated by the autoencoder are smaller than those of the input image. These features also contain pixel information.

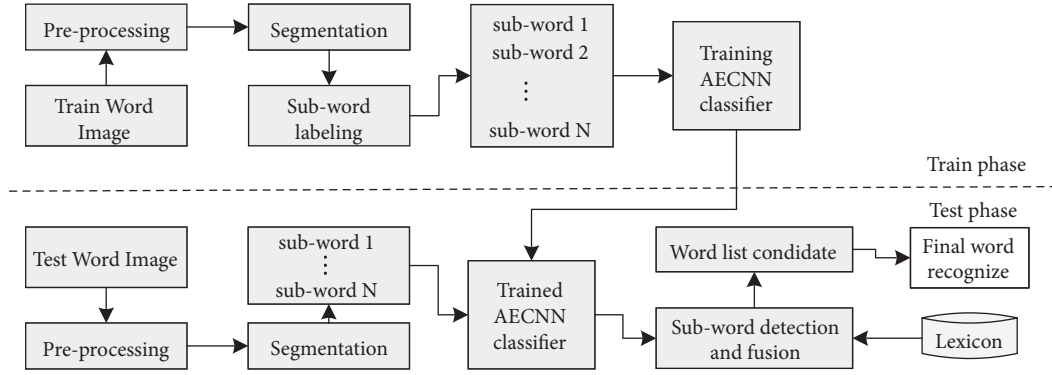


FIGURE 2: The block diagram of the proposed system.

$$\forall (0 < x < w, 0 < y < h) P(x, y) \in \{0, 1\}, \quad (4)$$

$$\forall (0 < x' < w', 0 < y' < h') F(x', y') \in [0, 1]. \quad (5)$$

In equation (4),  $P(x, y)$  indicates the value of pixel  $(x, y)$  of a binary image pertaining to a subword in  $w \times h$  dimensions. The pixel value is either one or zero. In fact, the pixel shows either the background (zero) or a subword outline (one). Equation (5) states the image output after being operated by the autoencoder in the  $w' \times h'$  dimensions. Furthermore,  $F(x', y')$  shows the value of pixel  $(x', y')$  ranging in the continuous span  $[0, 1]$  ( $h' < h$  and  $w' < w$ ).

After performing preprocessing and segmentation operations on a word, the recognition of subwords is done in the testing phase. Then the subwords are integrated. The integration of generated subwords should be based on how much the dictionary words resemble a corresponding word. In the proposed method, the recognized subwords are divided into main subwords and signs through the two following equations:

$$\text{SignSW} = \{\text{Dot}, \text{Zigbar}, \text{Hamze}, \text{Mad}\} \quad (6)$$

$$\text{MainSW} = U - \text{SignSW}. \quad (7)$$

In these equations,  $\text{SignSW}$  is the set of sign subwords including Dot, Zigbar, Hamza, and Mad. Moreover,  $\text{MainSW}$  is the set of main subwords including all of the subwords ( $U$ ) except for the sign subwords.

After dividing subwords into main and sign subwords based on the number of main subwords, it is necessary to determine how similar the combination of recognized subwords is to the dictionary words in order to introduce the most similar word/words. If several words have the highest rate of similarity, the similarity of secondary subwords is taken into account to reach the final decision, resulting in a unified output. All of the previous steps are discussed in detail here.

**4.1. Preprocessing.** Preprocessing is an important step in recognition systems. It prepares images for the next steps and puts them in the same conditions. The preprocessing step includes the following operations.

**4.1.1. Image Completion.** The dataset images have white backgrounds, although the background should be black in image processing operations when objects (subword outlines) are white. For this purpose, the supplementary image is determined through the following equation:

$$P'(x, y) = \begin{cases} 1 & \text{if } P(x, y) = 0 \\ 0 & \text{if } P(x, y) = 1. \end{cases} \quad (8)$$

In the above equation,  $P(x, y)$  and  $P'(x, y)$  show the main image and supplementary image, respectively.

**4.1.2. Noise Elimination.** There are usually tiny noise points on scanned images. They should be eliminated. The area filter of 10-pixel threshold was employed to eliminate these unwanted objects.

**4.1.3. Size Normalization.** The recognition system input images should be of the same size. Since the proposed system is based on subwords, each of them was normalized to the same size ( $50 \times 50$ ) after word segmentation and generation of subwords.

**4.1.4. Contour Line Detection and Gradient Modification.** In Persian, the contour line is a line on which letters and words are written and signed. Sometimes, the position of certain signs (such as dots) to the contour line can change the meaning of a word or even a sentence. For instance, different positions of a dot can completely change the meanings in “برو” meaning “go” and “ورن” meaning “do not go.” In Persian handwritten texts, contour line detection is usually done at the level of sentences because the word-level contour line detection is usually prone to errors.

This paper employed a method resembling the one used by [55] based on the horizontal histogram profile with a few changes to minimize the detection error. First, a moving-average filter of length  $L$  ( $L = 7$ ) was utilized to soften the horizontal histogram. Given the fact that Persian words are written on the contour line, the local maximums of the selected horizontal histogram are on the contour line. However, these local maximums, usually existing nearly above or below the image, are sometimes not on the contour line in Persian handwritten words. Figure 3 shows some

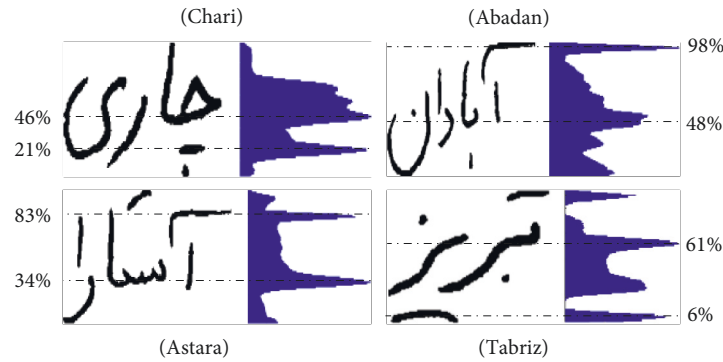


FIGURE 3: Incorrect and correct contour lines based on the maximum histogram.

samples of Persian handwritten words on the right and wrong contour lines based on the maximum histogram.

To avoid selecting an inappropriate contour line, it should be ensured that the maximum histogram is not placed 25% above or below the image. If the maximum value exists in these areas, the next maximum value is checked until the correct contour line is finally obtained. In other words, the maximum histogram is considered from 25% to 75% in the middle of the contour line image.

An unwanted gradient sometimes appears when words are written or images are scanned. Like the method used in [55], the images were rotated from  $-5$  to  $+5$  degrees with an angle of  $.5$  degrees to modify the unwanted gradient. Then the contour line histogram was determined. The largest histogram shows the best rate of rotation, which should be applied to images.

**4.2. Subword Segmentation and Labeling.** For subword segmentation and labeling, first the eight-connected neighboring method is used to identify and divide connected objects. Next, signs are removed from the connected components. After removing the signs, the similar ones are labeled by their representatives. For example, subwords such as “ب”, “پ”, “ت”, and “ث” (Be, Pe, Te, and Se) have the same outline and after removing of the dots are categorized in the group “ب.”

The rules of labeling the proposed interconnected components are as follows:

- (1) The beginning, middle, ending, and separate “ب”, “پ”, “ت”, and “ث” (Be, Pe, Te, and Se) are put in the “ب” (Be) group.
- (2) The beginning and middle “ن” and “ی” (Noon and Ye) are put in the “ب” (Be) group.
- (3) The beginning, middle, ending, and separate “ح”, “ج”, “خ”, and “گ” (He, Jim, Che, and Khe) are put in the “ح” (He) group.
- (4) The beginning, middle, ending, and separate “د” and “ذ” (Dal and Zal) are put in the “د” (Dal) group.
- (5) The beginning, middle, ending, and separate “ر”, “ز”, and “ژ” (Re, Ze, and Zhe) are put in the “ر” (Re) group.
- (6) The beginning, middle, ending, and separate “س” and “ش” (Sin and Shin) are put in the “س” (Sin) group.

- (7) The beginning, middle, ending, and separate “ص” and “ض” (Sad and Zad) are put in the “ص” (Sad) group.
- (8) The beginning, middle, ending, and separate “ط” and “ظ” (Ta and Za) are put in the “ط” (Ta) group.
- (9) The beginning, middle, ending, and separate “ع” and “غ” (Ayn and Ghayn) are put in the “ع” (Ayn) group.
- (10) The beginning and ending “ف” and “ق” (Fe and Ghaf) are put in the “ف” (Fe) group.
- (11) The beginning, middle, ending, and separate “ک” and “گ” (Kaf and Gaf) are put in the “ک” (Kaf) group.
- (12) The rest of Persian letters act as their beginning, middle, ending, and separate labels.
- (13) A dot, two dots, and three dots are put in the “1D”, “2D”, and “3D” groups, respectively.
- (14) The separated upper parts of “ک” and “گ” (Kaf and Gaf) are put in the “1ZB” and “2ZB” groups, respectively.
- (15) The Mad is put in the “Mad” group.
- (16) The Hamza is put in the “HZ” group.
- (17) The subword “ل” (La) is written like لا (La) by some people. Therefore, they are put in two groups of “ل” (La), one with “لا” (La) and the other one with two subwords “لا” (Lam) and “ا” (Alef).

According to the 17th rule, words with “ل” (La) such as “آق قلا” (Agh ghala) can be considered in two modes (“آق قلا” and “آق قلا”). Therefore, the number of words increased from 503 to 531 in the dataset. Table 2 shows the rules of labeling the interconnected components for the alphabet.

After applying the labeling rules to the interconnected components of 531 words in the dataset, 328 classes of subwords were obtained with different samples. These images were employed to train the classifier. The secondary classes included “1D”, “2D”, “3D”, “ZB”, “Mad”, and “HZ” known as the sign subwords. Table 3 shows the subwords of different written forms of “پولادشهر” (Poladshahr) along with some other sample images.

**4.3. The Dataset of Main and Sign Subwords.** After labeling the interconnected components and creating different classes of subwords, the dataset of main subwords and

TABLE 2: The labeling rules of the proposed interconnected components.

Characters		Characters positions				Label
		Beginning	Middle	End	Separate	
ا (Alef)		✓	✓	✓	✓	ا (Alef)
ب (Be)	پ (Pe)	✓	✓	✓	✓	ب (Be)
ت (Te)	ث (Se)	✓	✓	✓	✓	ب (Be)
ن	ی	✓	✓			ب (Be)
ح (He)	ج (Jim)	✓	✓	✓	✓	ح (He)
چ (Che)	خ (Khe)	✓	✓	✓	✓	ح (He)
د (Dal)	ذ (Zal)	✓	✓	✓	✓	د (Dal)
ر (Re)	ز (Ze)	✓	✓	✓	✓	ر (Re)
ژ (Zhe)		✓	✓	✓	✓	ر (Re)
س (Sin)	ش (Shn)	✓	✓	✓	✓	س (Sin)
ص (Sad)	ض (Zad)	✓	✓	✓	✓	ص (Sad)
ط (Ta)	ظ (Za)	✓	✓	✓	✓	ط (Ta)
ع (Ayn)	غ (Ghayn)	✓	✓	✓	✓	ع (Ayn)
ف (Fe)	ق (Ghaf)	✓	✓			ف (Fe)
ف (Fe)				✓	✓	ف (Fe)
ق (Ghaf)				✓	✓	ق (Ghaf)
ک (Kaf)	گ (Gaf)	✓	✓	✓	✓	ک (Kaf)
م (Mim)		✓	✓	✓	✓	م (Mim)
ن (Noon)				✓	✓	ن (Noon)
و (Waw)		✓	✓	✓	✓	و (Waw)
ه (He)		✓	✓	✓	✓	ه (He)
ی (Ye)				✓	✓	ی (Ye)

signs was generated. This dataset can be employed to integrate subwords and recognize the final word. For the generation of the main subword dataset, all of the words should be labeled by the rules of labeling the proposed interconnected components, discussed in the previous section. Regarding the sign subword dataset, it is necessary to determine the word under a sign and the position of the word to the contour line. The positions of “ZB,” “Mad,” and “HZ” are not important. The positions of sign subwords are checked to the contour line only for “1D,” “2D,” and “3D” classes. The signs placed below the contour line are labeled “D,” and those placed above the contour line are labeled “U.”

The rules of codifying the dataset of sign subwords are as follows:

- (1) The letters “ن,” “ط,” “خ,” “غ,” “ف,” “ض,” “ز,” and “ذ” (Noon, Zad, Fe, Ghayn, Khe, Za, Ze, and Zal) are labeled as “1D-U.”
- (2) The beginning, middle, and ending “ب” and “ج” (Be and Jim) are labeled as “1D-D.”
- (3) The beginning, middle, and ending “ق” and “ت” (Ghaf and Te) are labeled as “2D-U.”
- (4) The beginning and middle “ی” (Ye) are labeled as “2D-D.”
- (5) The beginning, middle, and ending “ث” and “ش” (Se and Shin) are labeled as “3D-U.”
- (6) The beginning, middle, and ending “چ” (Che) are labeled as “3D-D.”

Table 4 shows main subwords and signs of some samples in the dataset. Table 5 exhibits the number of main subwords of all the words existing in the dataset.

#### 4.4. The Autoencoder Network Used as the Feature Extractor.

As discussed earlier, autoencoder networks consist of an encoder and a decoder. They are employed to reconstruct the input image in the decoder output by adjusting weights and network biases. They are classified as unsupervised network training. The interesting feature of these networks is that they first create a compressed and coded vector of the image, based on which the input image is constructed. This vector can be used as the feature vector containing appropriate locational-pixel information in the CNN classifier. Since the CNN input is an image, the generated feature vector should be transformed into a 2-dimensional vector so that it can be injected into the CNN.

The proposed autoencoder network has 625 neurons in the encoder layer and an activator function in the rectified linear unit (ReLU). It also has 2500 neurons (the number of pixels on the input image) in the decoder layer and a sigmoid activator function. The generated feature vector has 625 components in the encoder output, which will be of size 25\*25 after being transformed into a 2-dimensional form. The SGD algorithm was employed along with the MSE function to train the autoencoder network. All of the training images of subwords, generated in the previous section, were utilized as the autoencoder network input and output. The training algorithm adjusts the network weights and biases to converge the network output on the network input. After training the network, the autoencoder output includes the feature vector and new training images used in the CNN. Figure 4 shows the proposed autoencoder network along with the input, output, and coded images.

4.5. CNN Architecture. Regarding the architecture of the network used in this study, it can be stated that the first layer is a convolutional layer including 32 filters of size 4\*4 and

TABLE 3: The resultant subwords of "پولادشهر."



















Subwords	Class	Sample		
Dots	1D			
	2D			
	3D			
پو (Po)	پو			
ال (la)	ال			
	ل			
لا (la)	ا			
د (d)	د			
شهر (shaher)	هرس			

TABLE 4: Main subwords and signs of some samples in the dataset.

Word	Main subword	Sign subword
آبیک (Abyek)	ب، پ، ک، ا	Mad, 1D-D, 2D-D, 1ZB
بیرجند (Birjand)	ب، پ، ر، ح، ب، د	1D-D, 2D-D, 1D-D, 1D-U
کرم‌ان‌شاه (Kermanshah)	ک، ر، م، ا، ب، س، ا، ه	1ZB, 1D-U, 3D-U
گرگان (Gorgan)	ن، ک، ر، ک، ا	2ZB, 2ZB, 1D-U
یزد (Yazd)	ب، ر، د	2D-D, 1D-U

TABLE 5: The number of main subwords in the dataset.

Number of main subwords	Number of words
1	36
2	156
3	169
4	121
5	38
6	8
7	3
Number of all words	531

images of size  $25 \times 25$ , which is the very autoencoder output. This layer employs ReLU to eliminate the negative values ( $C_1$ ). The pooling layer of  $2 \times 2$  window size and MaxPooling function is put after it ( $P_1$ ). Then another convolutional layer including 32 filters of size  $2 \times 2$  and ReLU ( $C_2$ ) are put along with a pooling layer in the same way as the previous step ( $P_2$ ). Softmax is an activator function employed to generate the output classes at the end of a fully connected layer with 328 neurons and ReLU (FC). Then the output layer of 328

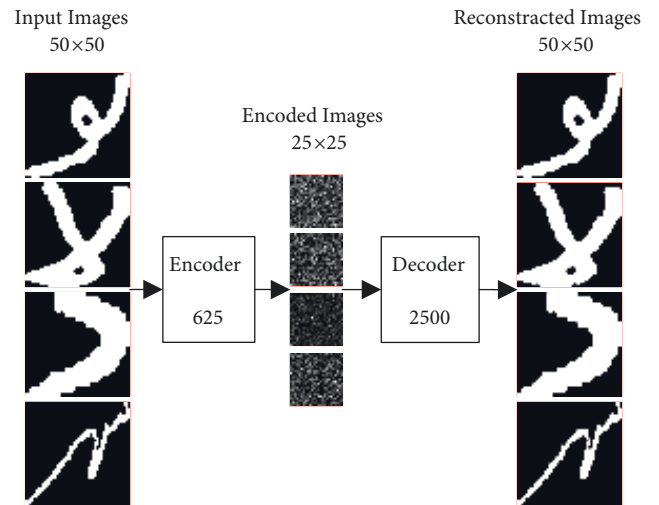


FIGURE 4: Feature extraction through the proposed autoencoder network.

neurons was used for different classes of subwords. Tables 6 and 7 show the local and global parameters of the proposed CNN, and Figure 5 indicates its architecture.

**4.6. Subword Classification Using the AECNN.** The proposed AECNN was employed to classify different subwords. This network consists of an AE network and a CNN in a subsequence. It performs feature extraction and classification simultaneously. First, all of the training images of subwords are put into the AE network. After training this network and reaching an appropriate MSE, the encoder output vector is



TABLE 6: The local parameters of the proposed CNN.

Layers	Filter no.	Filter size	Input layer size	Output layer size	Activation function	No. of parameters
Convolution ( $C_1$ )	32	4×4	25×25	22×22	ReLU	544
Pooling ( $P_1$ )	32	2×2	22×22	11×11	MaxPooling	0
Convolution ( $C_2$ )	32	2×2	11×11	10×10	ReLU	160
Pooling ( $P_2$ )	32	2×2	10×10	5×5	MaxPooling	0
Fully connected ( $FC$ )	328	1×1	5×5	328×1	ReLU	262728
Output layer	328	1×1	328×1	328×1	Softmax	107912

TABLE 7: The global parameters of the proposed CNN.

Parameters	Value
Image size	25*25
Optimizer	SGD
Batch size	200
Max epochs	40
Learning rate	0.001
Output classes	10

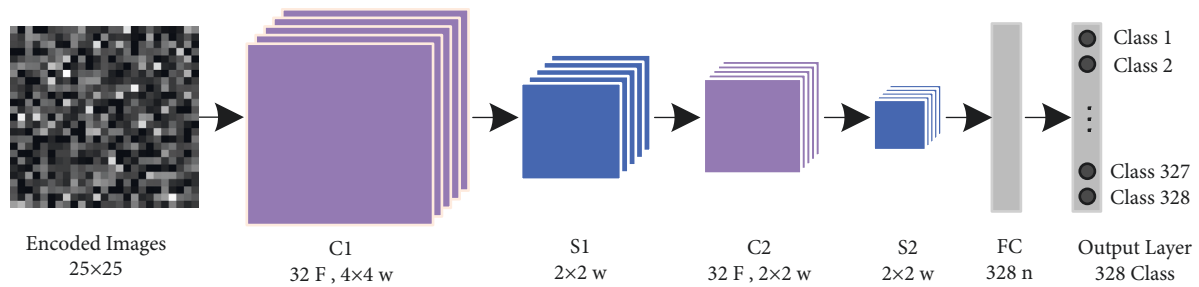


FIGURE 5: The architecture of the proposed CNN.

made two-dimensional and regarded as the CNN input images, including features affecting the recognition of subwords. The CNN is trained with these images and classifies different subwords. In fact, the process of training AECNN consists of two steps: training the AE network and training the CNN. Figure 6 shows the proposed AECNN.

#### 4.7. Integration of Subwords and Recognition of Final Words.

As discussed, the goal of the proposed recognition system is to detect an input word. As a result, all subwords of a word should be first recognized and then integrated correctly to form the input word of interest. Given labeling different letters of the same group, some words of the candidate dataset might belong to the input word. The signs of subwords might be needed to select them. It is also probable that the classifier may make mistakes in recognizing main subwords and signs and mistake main subwords for signs, and vice versa. As a result, the subword integration algorithm should predict these conditions and be resistant to them. The steps of the proposed integration algorithm are as follows:

- (1) Dividing the recognized subwords into main subwords and signs.
- (2) Integrating the main subwords in order of recognition and naming them MS according to the following equation:

$$MS = \{ms_1, \dots, ms_i, \dots, ms_n\}. \quad (9)$$

In the above equation,  $ms_i$  indicates the  $i$ th subword, and  $n$  shows the whole number of subwords.

- (3) Determining the similarity between MS and the number of datasets with three different assumptions:

- (3.1) The number of detected main subwords is correct: determining the similarity between MS and the words of dataset with  $n$  subwords (Table 5) and storing the highest rate of similarity as  $mss_1$  with corresponding words.
- (3.2) The number of detected main subwords is larger than the correct number of main subwords: According to equation (10), a new set is defined and named  $MS'$ , which is created by eliminating each  $ms_i$  subword:

$$MS' = \begin{cases} \{ms_2, ms_3, \dots, ms_n\} \\ \{ms_1, ms_3, \dots, ms_n\} \\ \vdots \\ \{ms_1, ms_2, \dots, ms_{n-1}\}. \end{cases} \quad (10)$$

Then the similarity of each row of  $MS'$  and words of the dataset with  $n - 1$  main subwords

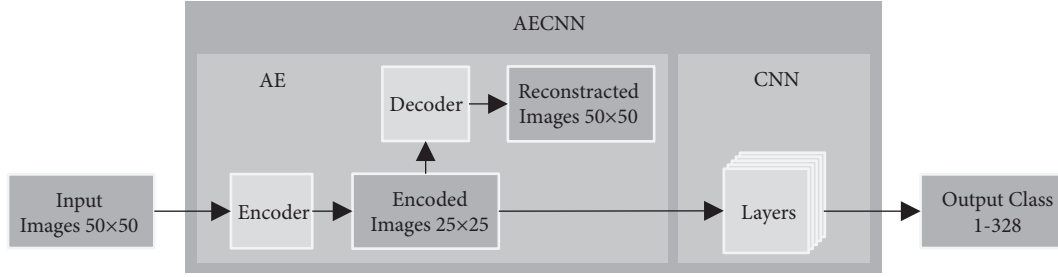


FIGURE 6: The proposed AECNN.

is determined, and the highest similarity rate is saved as  $mss_2$  with corresponding words.

- (3.3) The number of detected main subwords is smaller than the correct number of main subwords: According to equation (11), a new set is defined and named  $MS''$ , created by adding  $\emptyset$  to the position of each subword  $ms_i$ .

$$MS' = \begin{cases} \{\emptyset, ms_1, ms_2, \dots, ms_n\} \\ \{ms_1, \emptyset, ms_2, \dots, ms_n\} \\ \vdots \\ \{ms_1, ms_2, \dots, ms_n, \emptyset\}. \end{cases} \quad (11)$$

Then the similarity between each row of  $MS''$  and words of the dataset with  $n + 1$  main subwords is determined, and the highest rate of similarity is saved as  $mss_2$  with corresponding words.

After determining three maximum similarity rates ( $mss_1, mss_2, mss_3$ ), the similarity vector  $MSS$  is obtained from the following equation:

$$MSS = \{mss_1, mss_2, mss_3\}. \quad (12)$$

Equation (13) indicates the similarity percentage between two subword sets  $A$  and  $B$  of the same size:

$$Similarity(A, B) = \frac{\alpha}{n} \times 100. \quad (13)$$

In the above equation,  $\alpha$  and  $n$  show the number of matching subwords of the two sets and the total number of subwords, respectively.

- (4) Generating the sufficiency vector  $suf$ , indicating how fit a word is to be selected, by multiplying the similarity vector  $MSS$  into the sufficiency coefficient vector  $F$ , according to equation (15):

$$F = \{f_1, f_2, f_3\}. \quad (14)$$

$$Suf = MSS.F = \{mss_1.f_1, mss_2.f_2, mss_3.f_3\}. \quad (15)$$

In equation (14),  $f_1, f_2, f_3$  indicate the fitness coefficients corresponding to  $mss_1, mss_2, mss_3$ , respectively.

- (5) Detecting the word or words corresponding to the highest value of fitness. If there are several candidate

words, the next step is taken. Otherwise, the final word is printed, and the algorithm is terminated.

- (6) Detecting the contour line of the input word.  
 (7) Dividing the image into upper-contour and lower-contour sections.  
 (8) Labeling the upper-contour and lower-contour sign subwords as  $U$  and  $D$ , respectively.  
 (9) Integrating the sign subwords in order of detection and naming them as  $SS$  according to the following equation:

$$SS = \{ss_1, \dots, ss_i, \dots, ss_n\}. \quad (16)$$

In the above equation,  $ss_i$  is the  $i$ th sign subword, and  $n$  shows the total number of sign subwords.

- (10) Determining the similarity between  $SS$  and the candidate sign subwords, detecting and printing the final word based on the highest rate of similarity, and terminating the algorithm.

Considering the algorithm assumptions allows the correction of errors if a main subword is detected wrongly in the classification step. The values of fitness coefficients are also considered  $f_1, f_2, f_3$  controlling the fair selection of three assumptions. To determine their values, it should be taken into account that the first assumption is much stronger than the second ( $f_1 > f_2, f_3$ ), and the second and third assumptions are not superior to one another,  $f_2 = f_3$ . These coefficients should also be selected in a way to keep the value of  $suf$  between zero and 100. As a result, the fitness coefficient vector  $(1, f, f)$  can be determined. To select the value of  $f$ , the proposed recognition accuracy should be checked for different values of  $f$ , which can be seen in Figure 7. Accordingly, the highest accuracy was obtained in  $f = x$ ; therefore, the fitness coefficient vector is determined as  $\{1, .9, .9\}$ .

Figure 8 shows the steps of the proposed subword fusion algorithm for the word "تبریز" (Tabriz). Accordingly, the AECNN classifier output includes the following subwords: "Mad", "1", "ررD", "2", "ررD", and "1D." The first subword was detected as "Mad" instead of "2D." After dividing subwords into main subwords and signs, the set  $MS = \{"ر", "رر", "ررر"\}$  was obtained. For the first assumption, the similarity of  $MS$  and all words of the dataset should be determined to two main subwords, the number of which is 156 according to

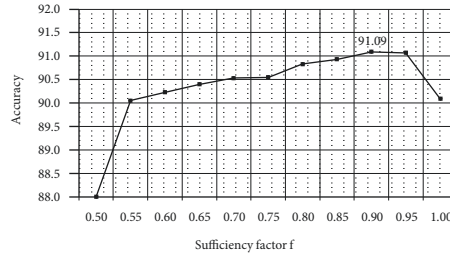


FIGURE 7: Classification accuracy based on different values of (f).

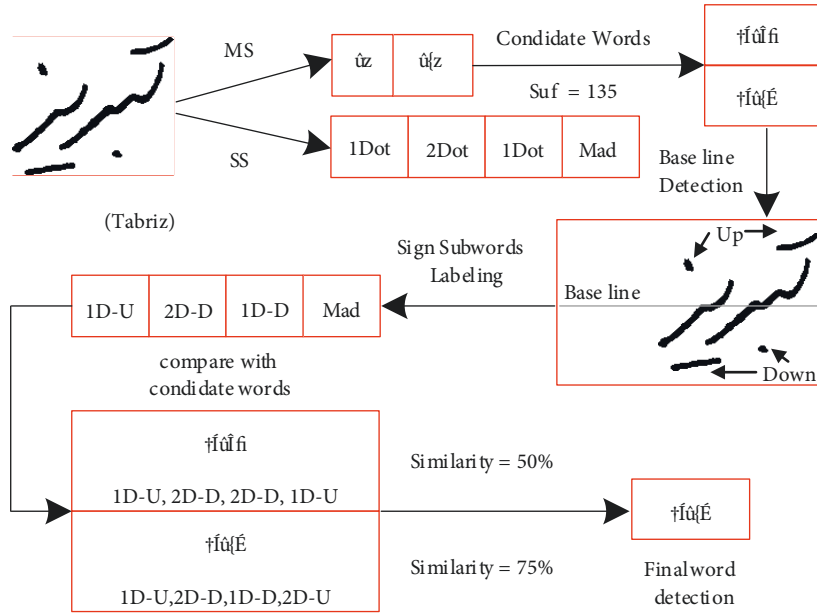


FIGURE 8: Subwords fusion steps and final word detection.

Table 6. Finally, “تبریز” (Tabriz) and “نریزن” (Neyriz) were the most similar words with  $mss_1 = 100$ . Regarding the second assumption,  $MS'$  was first developed as follows:

$$MS' = \begin{cases} \{ "رب" \} \\ \{ "زب" \} \end{cases} \quad (17)$$

Then the similarity of each row of  $MS'$  with all words of the dataset with one main subword was obtained as 36 according to Table 6. After that, no candidate words were found for this assumption; thus,  $mss_2 = 0$ . Regarding the third assumption,  $MS''$  should first be developed as follows:

$$MS'' = \begin{cases} \{ "رب", "زب", "" \} \\ \{ "ر", "زب", "" \} \\ \{ "", "زب", "زب" \} \end{cases} \quad (18)$$

Then the similarity of each line of  $MS''$  and all words of the dataset with three main subwords was obtained as 169 according to Table 6. Finally, “آبریز” (Abriz) was the most similar to the first row of  $MS''$  with  $mss_3 = 33$ . Thus, it was

selected as the candidate of the third assumption. Therefore, the similarity vector was  $MSS = \{100, 0, 33\}$ , which was obtained by multiplying the fitness coefficient functions into  $Suf = \{100, 0, 25\}$ . Based on the highest sufficiency coefficient, “تبریز” (Tabriz) and “نریزن” (Neyriz) were selected. The other steps of the algorithm should also be taken because the result was not unique. After detecting the contour line through the method introduced in the preprocessing section, the upper and lower signs were labeled as  $U$  and  $D$ ; then  $SS = \{ "1D\_U", "2D\_D", "1D\_D", "Mad" \}$  was created. Finally, the similarity of  $SS$  and candidate sings was determined. The signs of “تبریز” (Tabriz) were  $\{ "1D\_U", "2D\_D", "1D\_D", "2D\_U" \}$ , which were 75% similar, and those of “نریزن” (Neyriz) were  $\{ "1D\_U", "2D\_D", "2D\_D", "1D\_U" \}$ , which were 50% similar. Hence, “تبریز” (Tabriz) was selected as the final output.

### 5. Dataset

In this study, the widely used and known dataset Iranshahr was employed. It includes the names of 503 Iranian cities in

TABLE 8: Properties of Iranshahr dataset.

Dataset name	Classes	General			Samples	Dataset	No. of images	
		Dimension	Format	Resolution			Training set (%)	Testing set (%)
Iranshahr	503	Different	Binary	96 dpi	30–40	17000	85	15

different handwritings. All of the images were scanned with 96 dots per inch on a gray surface [54–57]. For the name of each city, there were nearly 30 to 40 different images. Totally, there were over 17000 images. In the training and testing section, the images were divided randomly into two groups: 85% of images for training and the other 15% for testing. Table 8 shows the research dataset in detail.

## 6. Evaluation of the Proposed Algorithm

A Hewlett-Packard (hp) computer was utilized to evaluate our approach with the 64-bit Windows 10 Home Operating System, installed memory (RAM) of 8.00 GB, and Intel (R) Core (TM) i7-6500U CPU. For data processing, the MATLAB and Statistics Toolbox Release 2021a were employed. The recognition results of the proposed method can be analyzed in two aspects: (1) what effects the autoencoder network have on the recognition accuracy and (2) what percentage of recognition accuracy pertains to the integration of the proposed subwords to generate the final words. The ability of any technique to recognize the word is measured by a parameter named “accuracy” and is demonstrated by

$$\text{Accuracy} = 100 \times \frac{TP + TN}{TP + TN + FP + FN}, \quad (19)$$

where TP, FP, TN, and FN are True Positive, False Positive, True Negative, and False Negative, respectively.

As discussed in the integration of subwords, the fitness coefficient vector was determined as  $\{1, f, f\}$ , controlling the fair selection of the three assumptions. Figure 7 shows the recognition accuracy of the proposed method based on different values of  $f$ . If  $f = .5$ , it means that it is probably as twice as the first assumption, and  $f = 1$  shows the sameness of the three assumptions. Accordingly, the highest accuracy was 91.09%, which was achieved in  $f = 0.9$ .

First, the CNN and the AECNN were compared in the same structure and parameters to analyze the effect of the autoencoder network on the recognition accuracy. Table 9 shows the comparison results, indicating the effect of the autoencoder network on the extraction of appropriate features and proper recognition of Persian handwritten words through the CNN. In fact, the accuracy was improved by 3%.

The proposed recognition system is based on subwords, which are integrated with each other in the final step to create the final words. Table 10 indicates the recognition accuracy of subwords without being integrated into other subwords to show the share of integration in the final accuracy.

Accordingly, the recognition accuracy of subwords was 85.93% in CNN; however, it was 86.68% in AECNN. According to the results of Tables 9 and 10, it is concluded that the

TABLE 9: A comparison between CNN and AECNN.

Method	Accuracy (%)
CNN	89.04
AECNN	91.09

TABLE 10: Subword recognition accuracy.

Method	Accuracy (%)
CNN without subwords fusion	85.93
AECNN without subwords fusion	86.68

proposed subword integration algorithm increased the recognition accuracy of CNN by 3.11%. It also increased the recognition accuracy of AECNN by 4.41%. As a result, although the Persian word recognition algorithm is based on subwords, the subword integration process is considered an error correction operation, which is an important part of the system.

If the problem is considered only from the prospective of the classifier, it is perceived that the recognition system is merely based on words, which were directly classified through CNN and AECNN with different parameters. In this case, the recognition accuracy would be similar to the results of Table 10 or even lower because there are 328 classes of subwords. However, there are 503 classes of words, and increasing the number of classes usually decreases the accuracy. This shows that dividing words into smaller components such as subwords or dividing words into main subwords and signs and classifying them into smaller components can significantly increase the recognition accuracy. However, such a classification can increase complexity and necessitates the existence of powerful algorithms in the component integration section in an effort to generate the final word. Table 11 shows some of the common mistakes in the classification step along with modifications by dividing subwords and detecting the final word.

Table 12 shows the recognition accuracy of the proposed method separately for each number of subwords. Accordingly, the lowest recognition accuracy (89.71%) came from words with one main subword, and the highest recognition accuracy (100%) came from words with seven subwords. Moreover, increasing the number of subwords relatively enhanced the recognition accuracy because a larger number of subwords in a word can increase the probability integrating subwords and correcting errors successfully. For instance, if the input word is only a subword with a recognition error, the integration of subwords will have no chance to correct the errors.

In Table 13, the recognition accuracy of the proposed method was compared with those of other methods on

TABLE 11: Classification errors and modifications by subword integration.

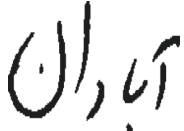
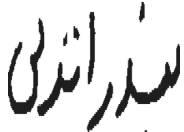
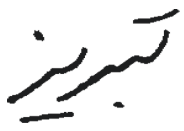
Input Image	Output Classifier	Detected Word
 (Abadan)	All: "1D", "ن", "ا", "د", "1D", "با", "ا", "2D" Main: "ن", "ا", "د", "با", "ا" Sign: "1D", "1D", "2D"	<b>Max Main Similarity: (100, 75, 67)</b> <b>Candidate:</b> ("آرادان", "باحکبران", "آبادان") <b>Score: (100, 75, 67) × (1, 0.9, 0.9) = (100, 67, 60)</b> <b>Detected Word/Words : "آبادان"</b>
 (Bandaranzali)	All: "سبید", "1D", "1D", "ر", "ا", "بر", "1D", "1D", "لی" Main: "سبید", "ر", "ا", "بر", "لی" Sign: "1D", "1D", "1D", "1D"	<b>Max Main Similarity: (80, 50, 33)</b> <b>Candidate:</b> ("وارآباد", "زامهرمر", "بیدرابرلی") <b>Score: (80, 50, 33) × (1, 0.9, 0.9) = (80, 45, 30)</b> <b>Detected Word /Words: "بندرانزلی"</b>
 (Tabriz)	All: "1D", "2D", "بر", "1D", "بیر", "Mad" Main: "بیر", "بر" Sign: "1D", "2D", "1D", "Mad" Sign by baseline: "1D-U", "2D-D", "1D-D", "Mad"	<b>Max Main Similarity: (100, 0, 33)</b> <b>Candidate: ("ابربر", "-", "بیربر")</b> <b>Score: (100, 0, 33) × (1, 0.9, 0.9) = (100, 0, 30)</b> <b>Detected Word /Words:</b> ("نیریز", "تبریز") <b>Word Sign (Tabriz): ("1D-U", "2D-D", "1D-D", "2D-U")</b> <b>Word Sign (Neyriz): ("1D-U", "2D-D", "2D-D", "1D-U")</b> <b>Sign Similarity: (75, 50)</b> <b>Detected Word: "تبریز"</b>

TABLE 12: Accuracies of the proposed method based on the number of subwords.

Number of main subwords	Accuracy (%)
1	89.71
2	90.14
3	90.77
4	90.79
5	92.34
6	94.59
7	100.00
Average accuracy	91.09

TABLE 13: Comparing the proposed method with the others.

Authors	Techniques	Accuracy (%)
Dehghan et al. [55]	HMM	63.00
Broumandnia et al. [56]	M-band packet wavelet	75.50
Younessy and Kabir [57]	RNN	83.90
Ghadikolaie et al. [54]	RNN	84.30
Proposed method	AECNN	91.09

Iranshahr dataset. Accordingly, the recognition accuracy of the proposed AECNN was 91.09%, indicating the efficiency of this method in comparison with other techniques.

## 7. Conclusion and Future Studies

Due to the complicated nature of the Persian and Arabic languages, in the proposed approach, words were divided into main subwords and sign subwords. Such an action increased the recognition accuracy. Regarding the feature extraction, the intrinsic characteristic of an autoencoder was employed to extract effective and useful features of subword images. After converting them into two-dimensional versions, they were classified through a CNN. The integration of an autoencoder and a CNN in an AECNN increased the recognition accuracy. Moreover, a powerful algorithm was proposed for subword fusion based on the similarity to the error modification approach. This algorithm increased the recognition accuracy by 4.41%. Due to the high similarity between Persian and Arabic, the proposed system can be used for Arabic handwritten word recognition. An important limitation of this research is that the proposed method is not applicable to other languages such as English, due to the intrinsic characteristics of Persian and Arabic languages which are very deferent from English-like languages.

In the future, we will focus on developing the proposed AECNN architecture by increasing the number of layers and deepening the network. Furthermore, it is possible to adopt a metaheuristic training algorithm for AECNN architecture in

order to reduce the training duration and enhance the network accuracy.

## Data Availability

The data used to support the findings of this study are available from the authors upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] k. younis and a. khateeb, "Arabic hand-written character recognition based on deep convolutional neural networks," *Jordanian Journal of Computers and Information Technology*, vol. 3, no. 3, p. 186, 2017.
- [2] M. Shafii, "Optical character recognition of printed Persian/Arabic documents," Doctoral Thesis, University of Windsor Canada, Canada, 2014.
- [3] H. Q. Ung, C. T. Nguyen, K. M. Phan, V. T. M. Khuong, and M. Nakagawa, "Clustering online handwritten mathematical expressions," *Pattern Recognition Letters*, vol. 146, pp. 267–275, 2021.
- [4] C.-L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition: Investigation of normalization and feature extraction techniques," *Pattern Recognition*, vol. 37, no. 2, pp. 265–279, 2004.
- [5] X. Qu, W. Wang, K. Lu, and J. Zhou, "Data augmentation and directional feature maps extraction for in-air handwritten Chinese character recognition based on convolutional neural network," *Pattern Recognition Letters*, vol. 111, pp. 9–15, 2018.
- [6] H. Soltanzadeh and M. Rahmati, "Recognition of Persian handwritten digits using image profiles of multiple orientations," *Pattern Recognition Letters*, vol. 25, no. 14, pp. 1569–1576, 2004.
- [7] N. Shanthi and K. Duraiswamy, "A novel SVM-based handwritten Tamil character recognition system," *Pattern Analysis & Applications*, vol. 13, no. 2, pp. 173–180, 2009.
- [8] L. S. Bernardo, R. Damaševičius, V. H. C. De Albuquerque, and R. Maskeliūnas, "A hybrid two-stage SqueezeNet and support vector machine system for Parkinson's disease detection based on handwritten spiral patterns," *International Journal of Applied Mathematics and Computer Science*, vol. 31, no. 4, pp. 549–561, 2021.
- [9] A. H. Alkilani and M. I. Nusir, "Off-line handwritten verification model for processing bank checks based on truncated-SVD and support vector machine (SVM)," in *Proceedings of the 2021 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pp. 37–42, IEEE, Amman, Jordan, November 2021.
- [10] M. Zare, M. Jampour, A. S. Arezoomand, and M. Sabouri, "Handwritten recognition based on hand gesture recognition using deterministic finite automata and fuzzy logic," in *Proceedings of the 2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA)*, pp. 93–99, IEEE, Tehran, Iran, March 2019.
- [11] N. Sakamat, *A guided hybrid k-means and genetic algorithm models for children handwriting legibility performance assessment*, PhD thesis, Universiti Teknologi MARA, Malaysia, 2021.
- [12] Y.-C. Wu, F. Yin, and C.-L. Liu, "Improving handwritten Chinese text recognition using neural network language models and convolutional neural network shape models," *Pattern Recognition*, vol. 65, pp. 251–264, 2017.
- [13] Z. Tamen, H. Drias, and D. Boughaci, "An efficient multiple classifier system for Arabic handwritten words recognition," *Pattern Recognition Letters*, vol. 93, pp. 123–132, 2017.
- [14] S. Chen, G. Liu, C. Wu, Z. Jiang, and J. Chen, "Image classification with stacked restricted Boltzmann machines and evolutionary function array classification voter," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 4599–4606, IEEE, Vancouver, BC, Canada, July 2016.
- [15] J. Gan, W. Wang, and K. Lu, "Compressing the CNN architecture for in-air handwritten Chinese character recognition," *Pattern Recognition Letters*, vol. 129, pp. 190–197, 2020.
- [16] Y. Weng and C. Xia, "A new deep learning-based handwritten character recognition system on mobile computing devices," *Mobile Networks and Applications*, vol. 25, no. 2, pp. 402–411, 2019.
- [17] M. Mhiri, C. Desrosiers, and M. Cheriet, "Convolutional pyramid of bidirectional character sequences for the recognition of handwritten words," *Pattern Recognition Letters*, vol. 111, pp. 87–93, 2018.
- [18] M. Taghizadeh and A. Chalechale, "A comprehensive and systematic review on classical and deep learning based region proposal algorithms," *Expert Systems with Applications*, vol. 189, Article ID 116105, 2022.
- [19] R. Sabzi, Z. Fotoohinya, A. Khalili et al., "Recognizing Persian handwritten words using deep convolutional networks," in *Proceedings of the 2017 Artificial Intelligence and Signal Processing Conference (AISP)*, pp. 85–90, IEEE, Shiraz, Iran, October 2017.
- [20] R. S. Alkhalwaldeh, M. Alawida, N. F. F. Alshdaifat, W. Z. a. Alma'aitah, and A. Almasri, "Ensemble deep transfer learning model for Arabic (Indian) handwritten digit recognition," *Neural Computing & Applications*, vol. 34, no. 1, pp. 705–719, 2022.
- [21] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and Cooperation in Neural Nets*, pp. 267–285, Springer, 1982.
- [22] H. Yang, J. Han, and K. Min, "Distinguishing emotional responses to photographs and artwork using a deep learning-based approach," *Sensors*, vol. 19, no. 24, p. 5533, 2019.
- [23] F. Wang, S. Wu, W. Zhang et al., "Emotion recognition with convolutional neural network and EEG-based EFDMs," *Neuropsychologia*, vol. 146, Article ID 107506, 2020.
- [24] A. Mahmood, M. Bennamoun, S. An et al., "Deep learning for coral classification," in *Handbook of Neural Computation*, pp. 383–401, Elsevier, 2017.
- [25] P. Shukla, N. Pramanik, D. Mehta, and G. Nandi, "Generative model based robotic grasp pose prediction with limited dataset," *Applied Intelligence*, vol. 52, pp. 1–15, 2022.
- [26] E. G. Ribeiro, R. de Queiroz Mendes, and V. Grassi Jr, "Real-time deep learning approach to visual servo control and grasp detection for autonomous robotic manipulation," *Robotics and Autonomous Systems*, vol. 139, Article ID 103757, 2021.
- [27] I. Lauriola, A. Lavelli, and F. Aioli, "An introduction to deep learning in natural language processing: Models, techniques, and tools," *Neurocomputing*, vol. 470, pp. 443–456, 2022.

- [28] H. Heidari and A. Chalechale, "Biometric authentication using a deep learning approach based on different level fusion of finger knuckle print and fingernail," *Expert Systems with Applications*, vol. 191, Article ID 116278, 2022.
- [29] R. Ranjbarzadeh, A. B. Kasgari, S. J. Ghoushchi, S. Anari, M. Naseri, and M. Bendechache, "Brain tumor segmentation based on deep learning and an attention mechanism using MRI multi-modalities brain images," *Scientific Reports*, vol. 11, no. 1, pp. 1–17, 2021.
- [30] S. Izadi, M. Ahmadi, and A. Rajabzadeh, "Network traffic classification using deep learning networks and bayesian data fusion," *Journal of Network and Systems Management*, vol. 30, no. 2, pp. 1–21, 2022.
- [31] R. Atefinia and M. Ahmadi, "Network intrusion detection using multi-architectural modular deep neural network," *The Journal of Supercomputing*, vol. 77, no. 4, pp. 3571–3593, 2020.
- [32] S. Di Cataldo and E. Ficarra, "Mining textural knowledge in biological images: Applications, methods and trends," *Computational and Structural Biotechnology Journal*, vol. 15, pp. 56–67, 2017.
- [33] Z. Lei, S. Zhao, H. Song, and J. Shen, "Scene text recognition using residual convolutional recurrent neural network," *Machine Vision and Applications*, vol. 29, no. 5, pp. 861–871, 2018.
- [34] I. Ramadhan, B. Purnama, and S. Al Faraby, "Convolutional neural networks applied to handwritten mathematical symbols classification," in *Proceedings of the 2016 4th International Conference on Information and Communication Technology (ICoICT)*, pp. 1–4, IEEE, Bandung, Indonesia, May 2016.
- [35] D. Bashkirova, "Convolutional neural networks for image steganalysis," *BioNanoScience*, vol. 6, no. 3, pp. 246–248, 2016.
- [36] W. Wang, Y. Huang, Y. Wang, and L. Wang, "Generalized autoencoder: A neural network framework for dimensionality reduction," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 490–497, Columbus, OH, USA, June 2014.
- [37] M. Chen, X. Shi, Y. Zhang, D. Wu, and M. Guizani, "Deep feature learning for medical image analysis with convolutional autoencoder neural network," *IEEE Transactions on Big Data*, vol. 7, no. 4, pp. 750–758, 2017.
- [38] M. Shopon, N. Mohammed, and M. A. Abedin, "Bangla handwritten digit recognition using autoencoder and deep convolutional neural network," in *Proceedings of the 2016 International Workshop on Computational Intelligence (IWCI)*, pp. 64–68, IEEE, Dhaka, Bangladesh, December 2016.
- [39] J. Almotiri, K. Elleithy, and A. Elleithy, "Comparison of autoencoder and Principal Component Analysis followed by neural network for e-learning using handwritten recognition," in *Proceedings of the 2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, pp. 1–5, IEEE, Farmingdale, NY, USA, May 2017.
- [40] H. D. Nguyen, K. P. Tran, S. Thomassey, and M. Hamad, "Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management," *International Journal of Information Management*, vol. 57, Article ID 102282, 2021.
- [41] P. S. Vasafi, O. Paquet-Durand, K. Brettschneider, J. Hinrichs, and B. Hitzmann, "Anomaly detection during milk processing by autoencoder neural network based on near-infrared spectroscopy," *Journal of Food Engineering*, vol. 299, Article ID 110510, 2021.
- [42] A. G. Dastider, F. Sadik, and S. A. Fattah, "An integrated autoencoder-based hybrid CNN-LSTM model for COVID-19 severity prediction from lung ultrasound," *Computers in Biology and Medicine*, vol. 132, Article ID 104296, 2021.
- [43] M. Seyfioglu and S. Gurbuz, "Deep convolutional autoencoder for radar-based human activity recognition," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 99, 2018.
- [44] A. Alani, "Arabic handwritten digit recognition based on restricted Boltzmann machine and convolutional neural networks," *Information*, vol. 8, no. 4, p. 142, 2017.
- [45] G. Tong, Y. Li, H. Gao, H. Chen, H. Wang, and X. Yang, "MA-CRNN: A multi-scale attention CRNN for Chinese text line recognition in natural scenes," *International Journal on Document Analysis and Recognition*, vol. 23, no. 2, pp. 103–114, 2019.
- [46] G. Xie, K. Yang, and J. Lai, "Filter-in-Filter: Low cost CNN improvement by sub-filter parameter sharing," *Pattern Recognition*, vol. 91, pp. 391–403, 2019.
- [47] C. Wu, W. Fan, Y. He, J. Sun, and S. Naoi, "Handwritten character recognition by alternately trained relaxation convolutional neural network," in *Proceedings of the 2014 14th International Conference on Frontiers in Handwriting Recognition*, Hersonissos, Greece, September 2014.
- [48] S. J. Ghoushchi, R. Ranjbarzadeh, S. A. Najafabadi, E. Osgooei, and E. B. Tirkolaee, "An extended approach to the diagnosis of tumour location in breast cancer using deep learning," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 1–11, 2021.
- [49] T. Bluche, H. Ney, and C. Kermorvant, "Feature extraction with convolutional neural networks for handwritten word recognition," in *Proceedings of the 2013 12th International Conference on Document Analysis and Recognition*, Washington, DC, USA, August 2013.
- [50] H. M. Albeahdili, T. Han, and N. E. Islam, "Hybrid algorithm for the optimization of training convolutional neural network," *International Journal of Advanced Computer Science and Applications*, vol. 1, no. 6, pp. 79–85, 2015.
- [51] T. Liu, Z. Li, C. Yu, and Y. Qin, "NIRS feature extraction based on deep auto-encoder neural network," *Infrared Physics & Technology*, vol. 87, pp. 124–128, 2017.
- [52] W. Luo, J. Li, J. Yang, W. Xu, and J. Zhang, "Convolutional sparse autoencoders for image classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 7, pp. 3289–3294, Jul 2018.
- [53] G. Li, S. Peng, C. Wang, J. Niu, and Y. Yuan, "An energy-efficient data collection scheme using denoising autoencoder in wireless sensor networks," *Tsinghua Science and Technology*, vol. 24, no. 1, pp. 86–96, 2019.
- [54] M. F. Y. Ghadikolaie, E. Kabir, and F. Razzazi, "Sub-word-based offline handwritten farsi word recognition using recurrent neural network," *ETRI Journal*, vol. 38, no. 4, pp. 703–713, 2016.
- [55] M. Dehghan, K. Faez, M. Ahmadi, and M. Shridhar, "Handwritten Farsi (Arabic) word recognition: A holistic approach using discrete HMM," *Pattern Recognition*, vol. 34, no. 5, pp. 1057–1065, 2001.
- [56] A. Broumandnia, J. Shanbehzadeh, and M. R. Varnoofaderani, "Persian/Arabic handwritten word recognition using M-band packet wavelet transform," *Image and Vision Computing*, vol. 26, no. 6, pp. 829–842, 2008.
- [57] M. F. Younessy and E. Kabir, "A new classifier based on Recurrent neural network using multiple binary-output networks," *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 17, no. 3, pp. 63–69, 2015.