Noname manuscript No.
(will be inserted by the editor)

# Multilingual Handwritten Numeral Recognition Using a Robust Deep Network Joint with Transfer Learning

**Amirreza Fateh · Mansoor Fateh* · Vahid Abolghasemi**

**Abstract** Numeral recognition plays a crucial role in creating automated systems such as posting address sorting and license plate recognition. Nowadays, numeral recognition systems which have the capability of recognizing multiple languages are highly beneficial due to growing international correspondence and transactions, especially in multilingual countries where several languages are used simultaneously. Therefore, handwritten numeral recognition is more challenging than printed numeral recognition due to having different and complex handwriting styles. Hence, developing a multilingual handwritten system is considered as an important and debatable issue. We address this issue by proposing a language-independent model based on a robust CNN. Our proposed model is composed of language recognition and digit recognition, which aims to handle the recognition of multi-script images. We used transfer-learning in the proposed system to enhance the image quality and consequently the recognition performance. Extensive experiments were conducted to verify the effectiveness of both language and digit recognition procedures. The proposed system was tested with six different languages. The results showed an average accuracy of up to 99.8% for recognizing various languages and the associated digits. The robustness and design procedure of the proposed model created a cost-effective extension for recognition of handwritten numerals in other languages.

A. Fateh
Faculty of Computer Engineering, Shahrood University of Technology, Shahrood, Iran
E-mail: amirreza_fateh@shahroodut.ac.ir

M. Fateh
Faculty of Computer Engineering, Shahrood University of Technology, Shahrood, Iran
*corresponding author
E-mail: mansoor_fateh@shahroodut.ac.ir

V. Abolghasemi
School of Computer Science and Electronic Engineering, University of Essex, Colchester, CO4 3SQ, UK
E-mail: v.abolghasemi@essex.ac.uk

**Keywords** Deep Learning · Transfer Learning · Multilingual · Handwritten Numeral Recognition

## 1 Introduction

Recognition of handwritten numerals (digits) has attracted a lot of interest within the communities of computer vision and image processing. The main reason can be attributed to the key role of this program in various applications including automated sorting of postal mails, automated processing of bank cheques, automatic car plate recognition, digitisation of old historical handwritten documents, etc. [1–3]. This research topic lies under a broader field of handwritten character recognition. Both handwritten digit and character recognition suffer from the problems of diverse writing styles, strokes, and etc., compared to printed numeral or character recognition. A handwritten recognition system seeks to convert handwritten characters into machine readable formats. The importance of obtaining accurate digit recognition for using in sensitive areas such as bio-metric authentication and financial systems, motivates researchers to design more efficient handwritten numeral recognition systems. Most of the previous research has been carried out on English scripts because it is used as an international communication platform. However, it is also necessary to establish robust and accurate recognition systems for scripts of other languages, considering the current advances in digital age.

In the current decade, various machine learning techniques are used for handwritten numeral recognition, including Random Forests (RF), k Nearest Neighbor (kNN), Decision Tree (DT), Hidden Markov Model (HMM), local similarity and diversity preserving discriminant projection (LSDDP), fuzzy algorithms, etc. [1,3–6]. Some researchers have integrated these machine learning techniques with image processing methods to increase recognition accuracy and the system performance. The existing studies have used different features such as Gabor filters, Histogram of oriented Gradient (HOG), Discrete cosine transform (DCT), Wavelet, to name a few, with different classifiers. The emergence of deep learning has led to higher expectations for improving the performance of handwritten recognition systems. Deep learning has shown promising breakthrough in numerous machine learning applications. Hence, several works have focused on developing related techniques for the purpose of handwritten character recognition [7]. This paradigm shift, which has also been expanded for handwritten numeral recognition, is indebted to the adaption of cluster computing and GPUs and better performance by deep learning architectures. Various forms of deep neural networks have been introduced in the recent years, some of which are Recurrent Neural Networks (RNN), Convolutional Neural Network, Long Short-Term Memory (LSTM) networks, etc.

Despite rich literature on handwritten digit recognition of single-script (one language), few studies have worked on multi-script handwritten numeral recognition [8]. This problem has been addressed by previous related research by considering two main strategies. One strategy is to consider each language separately and then to apply the recognition of a every single language independently. In a work by Pal et al. [9], a modified quadratic classifier-based architecture was proposed for off-line recognition of handwritten digits of six popular Indian scripts. The other interesting strategy is to assemble and feed all script into a single model for recognition. This way, the number of classes to be recognised is multiplied by the number of languages used. Most of studies in this category attempt to find a solution by fusing classes of similar shape to decrease the the number of classes and consequently improving the performance [10]. In other words, those numerals posing relatively similar shapes are merged into one class. However, when the number of classes grows over ten, the techniques which are based on this approach would suffer from some limitations and low performance issue.

In this study, three major contributions are developed and presented: 1) we proposed a novel cascade multilingual approach that eliminated the existing drawback of low performance when number of classes were increased. As a solution, a language recognition module were first applied to the input handwritten images, and then the appropriate model is automatically selected for classification of the associated digits, 2) we proposed a novel robust CNN model with a special recognition rate, both for the recognition of languages and digits of input images. The proposed model can recognise handwritten digits in several languages, 3) we used transfer learning, equipped with an autoencoder, to improve the quality of low-resolution images prior to feeding the recognition stage, 4) the proposed model was assessed through six handwritten databases with different languages, i.e., English, Kannada, Urdu, Persian, Arabic, and Chinese, with significant diversity in writing styles, and 5) despite the multilingual support in the proposed model, it achieved higher accuracy than single language models, which showed the superior performance of the proposed method.

The rest of the paper follows this order review of related works is presented in Section 2. The proposed model is described in Section 3. Performance evaluation and experimental results are given in Section 4, followed by concluding remarks in Section 5.

## 2 Related Work

In this section, an overview of the related works is provided for handwritten numeral recognition. This includes methods based on handcrafted features, traditional classifiers, deep neural networks, sparse representation, transfer learning, and their combinations.

### 2.1 Classic approaches

One of the traditional classification methods used for handwritten digit recognition is support vector machine. In [11], vertical and horizontal edges, and directions of numerals were extracted from the input images. These were then combined with the Freeman chain code to form a great feature set to feed to the SVM for classification. This feature extraction regime avoids the need for normalisation of digits which ultimately improves the recognition performance.

In [12], handwritten character recognition was proposed for alphabets and digits of Persian and Arabic languages. The authors used fractal codes while taking advantage of the similarities between the characters. A multilayer perceptron (MLP) was designed as a classifier for this purpose in which the classification rate of 91.37% was achieved for digits recognition.

One of the pioneering works in handwritten numeral recognition was conducted in [13] based on a fuzzy model. The authors derived some fuzzy sets from features consisting of normalised distances obtained by using the Box approach. The recognition rates of 95% and 98.4% have been reported for Hindi and English numerals respectively.

In [14], a handwritten numeral recognition was proposed, using a hybrid feature set, and followed by a comparative analysis. In this research, the feature set was composed of multiple feature extraction approaches such as Box Method, Mean, Standard Deviation and Centre of Gravity. The classification was performed using a shallow neural network with the CHARS74 handwritten numerals database.

In a research based on shape analysis, Dash et al. [15] exploited the human perception for handwritten numeral recognition. They proposed a shape decomposition approach which breaks the digits into several parts depending on their shapes. Then, the classification is performed on the extracted parts instead. This method has been evaluated on four scripts, namely Odiya, Bangla, Arabic, and English individually.

### 2.2 Deep learning approaches

Within the last decade, deep learning has been dominantly used for various pattern classification applications, particularly in handwritten digit recognition [16,17]. In one of the primary related works [16], a neural network with two successive feature extraction stages was implemented and tested with the MNIST benchmark database. The results were promising and demonstrated the effectiveness of using deep networks in this context.

In [18], a novel multi-scale CNN was proposed for extracting spatial classification features for handwritten mathematical expression including digits. The classification performance was improved in this work by training the CNN with a combination of global max pooling and global attentive pooling.

Shopon et al. [17] used a deep convolutional neural network for classification of handwritten numerals of English and Bangla scripts. They improved the recognition accuracy by applying a data augmentation stage to produce blocky effects on the training images. Although two languages were considered for this research, their method should be applied to these languages, separately.

In [19], a neural network architecture was proposed for the numeral recognition of Gujrati scripts. Their method was based on a multi-layers feed-forward neural network for handwritten digits classification. The authors applied thinning and skew-correction in the pre-processing stage to improve the performance. The resulting average recognition rate was 82% which was not very high.

A customised CNN with minimum number of layers was proposed in [20] for the classification of Chinese handwritten numbers. Competitive performance compared to deeper networks with larger sizes and number of parameters (e.g., GoogLeNet and MobileNetV2).

### 2.3 Transfer learning approaches

Transfer learning is known as a machine learning approach that develops an existing model to work for a new task. It has shown extensive utilisation in many applications such as action recognition [21], prediction of image memorability [22], as well as handwritten digit recognition. The researchers mainly used transfer learning to mitigate the limitation of access to large-scale databases. In [23], a transfer learning approach was adopted for handwritten digit recognition based on both the multi-layer perceptron and convolutional neural network models to share the feature extraction process among five handwritten numerical datasets, namely, Tibetan, Arabic, Bangla, Devanagari, and Telugu. The authors found that transfer learning could significantly reduce the training time of the deep learning models, and slightly reduced the recognition accuracy.

An analysis of historical handwritten documents using transfer learning was addressed in [24]. Transfer learning was used to recognise features from heterogeneous datasets with available ground-truth and to share common properties with a new dataset with no ground-truth. This is done to skip the time consuming task of manual ground-truth creation. This study indicated that the transfer learning can be used for the task of transcribing handwritten titles of plays of the Italian Comedy, when trained on combinations of various datasets such as RIMES, Georges Washington, and Los Esposalles.

In a very recent study, ResNet50 architecture was used to train an ultra-deep neural networks for implementing multi-class image classification [25]. The method is based on transfer learning and uses various numbers of hidden layers to classify handwritten digits. In this method, the recognition accuracies at different classes are determined by epochs. In this study, the highest recognition rate obtained with MNIST database was reported to be 99%.

### 2.4 Sparse learning approaches

There exists another category of learning-based methods, called dictionary learning, which initially was introduced for sparse data representation. These methods have been rapidly utilised in pattern classification and categorization too [26]. In this regard, a dictionary learning method was proposed in [27] for recognition of Chinese handwritten numbers. The authors classified 15 Chinese numbers by introducing a novel dictionary learning method with improved discriminability.

To provide a high-level feature representation in a transform domain, an unspuervised feature learning procedure was proposed in [28]. Also, principle component analysis (PCA) was used as a post-processing scheme to transform the overcomplete dictionary obtained from unsupervised feature learning. The aim of this work was to reduce the number of bases in the unsupervised dictionary learning for real-time applications.

A dictionary learning approach was proposed in [29] for image classification, based on the Fisher discrimination criterion. The learned dictionaries are structured with columns categorized with the subject class labels. The best recognition error rate, reported in this work with USPS English handwritten database, was considered as 2.89%.

In [30], a novel sparse non-negative approximation approach was proposed for classifying face and handwritten images. The authors proposed a convex sparse coding problem which was then used with overcomplete discrete cosine transform (DCT) dictionaries to classify handwritten images. A classification accuracy of 94.52% with MNIST database was reported.

### 2.5 Combined approaches

With the help of genetic algorithm (GA), Yan et al. [31] presented an optimised neural network model for handwritten numeral recognition. They utilised GA to optimise and design the structure, weights, thresholds, the training ratio and momentum factor of neural network.

Chen et al. [32], proposed an adaptive fractional-order back-propagation (BP) neural network for handwritten digits recognition. Their proposed system was the combination of a competitive evolutionary algorithm called population extremal optimization and a fractional-order gradient descent learning mechanism. This algorithm was tested on MNIST English database where superior performance was achieved.

In a recent paper [33], combination of deep learning and dictionary learning was performed on English handwritten digits. The authors achieved accuracy of more than 99% with a multi-layered framework. Their proposed method works based on K-SVD (K-singular value decomposition) [34] but under a multi-layer CNN model.

Pramanik et al. [35], proposed a system for handwritten digit recognition of four Indic scripts including Devanagari, Bangla, Odiya and Telugu. Their method was based on CNN with pre-trained networks. The experiments in their research were conducted separately with each language. In order to minimise the misclassification which may occur due to similarity in shape, two languages i.e. Bangla and Odia were mixed and treated as one language with more number of classes. Moreover, the fusion of similar shaped numerals was carried out which led to a classification problem with 19 classes in total.

A multi-script recognition method was proposed in [36] for classification of postal codes. The authors attempted to combine features of numerals from four scripts with similar shapes. They considered Latin (English), Devanagari, Bangla, and Arabic (Urdu), leading to 25 classes in total. An image partitioning based on quad-tree was implemented for feature extraction. Then, they feed the extracted features to a SVM classifier for recognition of 25 handwritten numerals.

In a recent paper, Gupta and Bag [8] proposed a script independent method for recognition of handwritten numerals. They tested their method with databases of several Indic and non-Indic scripts. Their method was a class-based CNN model which avoided fusion of similar shaped classes to improve the performance.

A quantitative analysis of Deep CNNs for multilingual handwritten digit recognition was recently reported in [37]. In this work, the performance of ten state-of-the-art deep CNN methods mainly focused on common languages in the Indian sub-continent were analysed. The authors concluded that Inception-v4 method was superior considering the accuracy and computation time.

Recently, the largest multi-language handwritten digit database, named MNIST-MIX, was introduced [38]. MNIST-MIX can be seen as an extended version of English MNIST digit database comprised of multiple languages like Arabic, Bangla, Devanagari, English, Persian, Kannada, Swedish, Telugu, Tibetan, and Urdu, adopted from different sources Various data processing techniques were been applied during combination of these languages to make all data samples consistent. In this research, The results of applying a pre-trained LeNet model showed an average recognition accuracy of around 90%.

## 3 Proposed Approach

In this section, we describe different stages of the proposed approach to process the input handwritten images. The core of this model is a deep convolution neural network which classifies the input handwriting images. Due to the multilingual nature of the problem, the model should be able to capture diverse range of characteristics existing in every single language. In order to be more precise, we first build a preliminary model to work with three languages with diverse structures, namely Chinese, Arabic, and English. After tuning the model and ensuring high performance, we extend the model to operate with three more languages, i.e. Kannada, Persian, and Urdu. Such a design procedure, which is supported by our extensive experiments, ensures smooth development of the model for numeral recognition in any other language. Figure 1 depicts various stages and the full procedure of the proposed model. As seen in this figure, the proposed system is composed of two main stages, i.e. language recognition (LR) and digit recognition (DR). Our purpose is to design a system which is compatible with input images at various sizes and

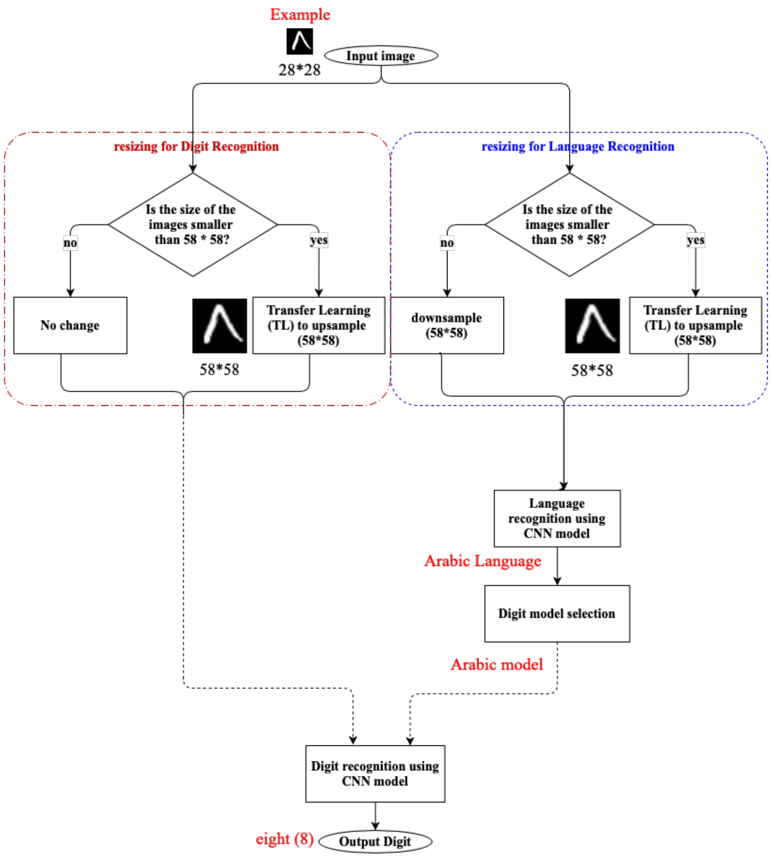



Fig. 1: Block diagram of the proposed model. Recognition of Arabic digit '8' image is demonstrated as an example.

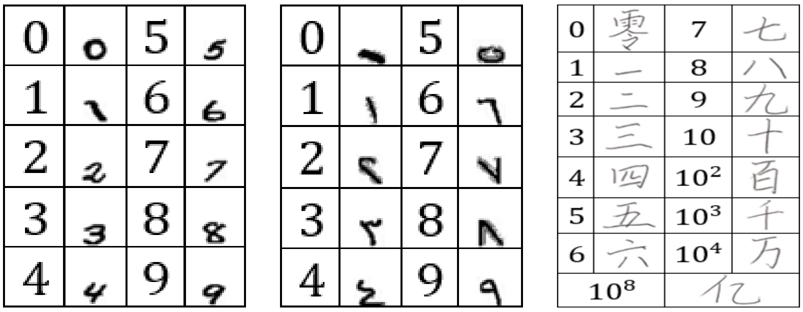
with different languages. Based on our exploration and observations within numerous existing databases, we found that a moderate size of $58 \times 58$ was suitable for input images. Hence, depending on the original size of the image, our proposed system performs a resizing operation (if needed), as seen in Figure 1. Simply upsampling the images with sizes smaller than $58 \times 58$ may lead to image quality degradation and thus reducing the recognition accuracy. In order to avoid this drawback, we enhance the quality of this image category by using a transfer learning procedure to avoid the image quality reduction. These steps will be explained in the following section in more details.

As mentioned earlier, the proposed system identifies the language associated with the input handwritten image, prior to recognising the actual digits. Afterward, the appropriate model is selected according to the identified language. Finally, deep learning is applied to the input image to recognise its corresponding class. The structure of this module is based on a convolutional neural network which has shown to be very effective for various image classification tasks.

In the remaining part of this section, we introduce the handwritten numeral databases that are used in this research. Then, different steps of both LR and DR blocks are explained in more details. Finally, the structure of the proposed deep neural network model, as well as the transfer learning technique, will be explained.

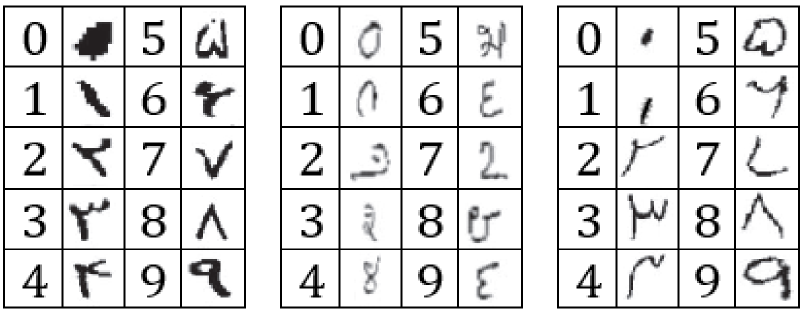
### 3.1 Databases

In this research, databases related to six different languages are considered to address the multi-linguistic feature of our proposed model. We considered well-established English (USPS [39]) and Arabic digits databases (MADBase [40]) to assess our method. The USPS database includes 7291 training samples and 2007 test samples of digits 0-9 in the form of grayscale images. Likewise, the MADBase includes 60,000 training samples and 10,000 test samples of dig-



(a) English (b) Arabic (c) Chinese

Fig. 2: Sample images from three different handwritten digits databases with their equivalent English digits.



(a) Persian (b) Kannada (c) Urdu

Fig. 3: Sample images from three different handwritten digits MNIST-MIX databases with their equivalent English values.

its 0-9. Sample images of these two databases are shown in Figures 2a and 2b. Furthermore, a publicly available database [27] are used that contains 15,000 Chinese handwritten numbers from 100 Chinese nationals. This database was collected by researchers at Newcastle University, United Kingdom, some sample images of which are represented in Figure 2c.

Other three databases used in this study are in Kannada, Persian, and Urdu languages (Figure 3). These databases belong to a recent collection of databases called MNIST-MIX [38]. The size of the images in these three databases is $28 \times 28$. The Persian database [41] is extracted from about 12,000 registration forms of two types, filled by undergraduate and senior high school students and consists of 60,000 training samples and 10,000 testing samples(Figure 3a). Kannada database [42] consists of 60,000 training samples and 10,000 testing samples, with the same data format with MNIST (Figure 3b). Urdu database [43] is collected from more than 900 individuals (Figure 3c). In addition to the above-mentioned databases, we used another Arabic database called HAND2020 [44]. This database has 72,000 images and is publicly available. Furthermore, we created a database of synthetic digits in English, Arabic, and Persian to evaluate the performance of the proposed transfer learning stage. We used the pillow library in Python to create this database. The images created in this database have a single grayscale channel all with size $58 \times 58$. This database includes 10,000 samples of synthetic digits 0-9 in form of grayscale images with 250 different font types.

### 3.2 Language Recognition

The first major step in our work is to build a language recognition model. Using this model, we will be able to feed each image to an appropriate digit recognition model according to the identified language. Previous studies showed that direct feeding of a multi-language script to a digit recognition model did not provide satisfactory results and decreased the classification accuracy. One of the main reasons for such malfunctioning can be attributed to the dramatic increase in the number of classes that are supposed to be recognised. As a solution, we propose using a language recognition model which seems necessary for categorising the images of different languages before using the main digit recognition process. Consequently, appropriate digit recognition parameters will be selected for the input images based on the identified language. Another great advantage of utilising the language recognition module in a multilingual system is that it adds the scalability and extra robustness to the system. Without the language recognition stage, whenever a new script (with different languages) is used, the entire digits recognition model needs to be trained from the beginning. However, by using the language recognition model, only part of the model, related to that specific language, needs to be retrained. In other words, the proposed handwritten recognition system can be expanded to function with more languages without the need for restructuring the entire system.

As mentioned at the beginning of this section, the size of images feeding into the language recognition model should be the same across all different languages. This requires up/down sampling of input images depending on the original image sizes. We have empirically chosen this universal size to be $58 \times 58$. In order to avoid quality degradation while upsampling the images, we propose using the transfer learning technique, where an autoencoder whose neurons' weights were pre-calculated, are used. Further details about this technique, as well as the structure of the autoencoder, are given in subsection 3.5.

Convolutional neural network (CNN) is a kind of neural network that is built from multiple layers, especially used for image recognition, image classification, and image analysis. There is the same set of operations in all CNN architectures, giving the input image to a convolutional layer and followed by a max-pooling layer to select the maximum element from the region of the feature. Besides, fully connected layers, activation functions, and softmax layer are involved in the basic model of CNN. Due to the similar structure of the input images of both language recognition and digit recognition, we used the same model for these stages with minor differences in some layers. Further details on the proposed CNN technique are given in subsection 3.4.

### 3.3 Digit Recognition

After recognizing the language of the input image, the second major step is digit recognition. Here, we explain our proposed recognition model which is based on a convolutional neural network. As seen from Figure 1, once the language associated with the input image is recognised, our system should decide on two important settings; pre-processing and selection of digit recognition model. In other words, the output of LR stage determines the required parameters and models for the next stage. In order to develop a successful deep learning system, one needs to prepare the input data for the network by using appropriate pre-processing operations (Figure 1). In DR stage, each image is taken from the database and is fed to one of the digit recognition models depending on the associated language. Our digit recognition model should be able to recognise digits within images of all databases with different sizes. The images from Arabic, Persian, Urdu, Kannada, and Chinese databases had all the same fixed sizes. The size of all images in Arabic, Persian, Urdu, and Kannada databases are $28 \times 28$, and that size is $64 \times 64$ for Chinese database. However, in the English database, the images had variable sizes, so we changed the size of all images to $192 \times 96$, it should be mentioned that 96 and 192 were considered as the maximum widths and lengths in the total images in this database respectively. Whenever an upsampling is required within the pre-processing of input images, we follow the same procedure explained above using transfer learning in order to maintain the image quality.

The structure of the proposed model is similar across all languages for digits recognition. However, the final model is trained separately with each language depending on the output of LR module. Moreover, all the network' layers of the CNN model for DR are the same across various languages except for the input layer which needs to be chosen differently depending on the language detected in the previous step. The output layer of the model associated to Chinese language should have 15 output neurons as of the number of classes. The advantage of having a LR module is that while input images can be in any of the six languages, the digit recognition model is automatically selected despite various image sizes, and character shapes of these languages which are relatively different from each other. If no language recognition system is in place, the final model would have to be obtained through training with all databases. This is not an efficient approach and causes problems such as complication in parameters tuning, increasing the number of classes to 64, and consequently reducing the performance and recognition accuracy. The advantage of utilizing a language recognition module in the proposed method is to first, identify the language of the input image, which is then forwarded to the corresponding trained digits recognition model of the same language.

Although there exists pre-trained models such as LeNet that could be used for digit recognition, we opt to design a new model for this purpose. Based on our observations, pre-trained networks perform inadequately, particularly when multi-language scripts are the focus of interest. We aim for a model to run on all languages with high accuracy. In the following, we explain different steps of the digit recognition module.

### 3.4 CNN Architecture

The proposed CNN architecture consists of three different types of layers; convection layers, pooling layers, and fully connected layers (also called dense layers). We will describe these layers in more detail as follows:

**Layer-1** is a convolutional layer with ReLU activation function. The input of this layer varies depending on the size of input images. For LR stage where the images have the size of $N \times N = 58 \times 58$, this layer is fixed. However, for DR stage the size of this layer is selected according to the detected language in LR stage. This is done in model selection blockset shown in Figure 1. For instance, in the model associated to Arabic database which has images with size of $N \times N = 28 \times 28$, this layer uses 64 filters with size of $F \times F = 3 \times 3$. In this layer, padding is $P = 0$, and stride is $S = 1$. The output of this convolutional layer has dimensions of $26 \times 26 \times 64$, which represents 64 matrices with 26 rows and 26 columns. Afterward, the ReLU activator function applies to each of these 64 output matrices.

**Layer-2** is a max-pooling layer that receives the output of the previous layer of size of $26 \times 26 \times 64$ as input. The pooling size is $2 \times 2$, padding is $P = 0$, and stride is $S = 2$. The output size of the max-pooling layer is $13 \times 13 \times 64$.

**Layers 3 and 5**, similar to the first layer, are convolutional layers that use the ReLU activation function. These two layers, like the first convolutional layer, use 64 filters with size $3 \times 3$, stride 1, and padding 0. Layers 4 and 6, like the second layer, are the subsampling layers, which precisely like the second layer, the pooling size is $2 \times 2$, stride 2, and padding 0.

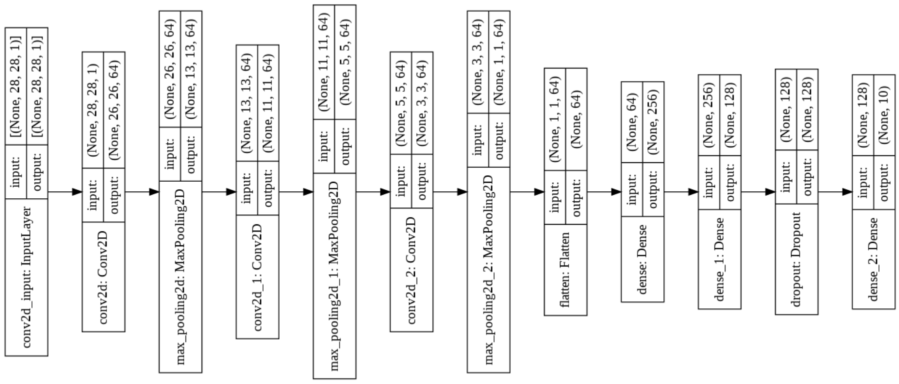



Fig. 4: Proposed CNN architecture for digit recognition.

**Flatten layer**: After 3 layers of convolutional and 3 layers of subsampling, the extracted properties are given as input to fully connected layers. Actually, fully connected layers perform classification based on the features extracted by the previous layers. However, a "Flatten" layer is required between the convolutional layer and the fully connected layer to convert matrix features into vectors that can be fed into a fully connected neural network classifier.

After flattening the features, they are given to two fully connected layers that have 256 and 128 neurons respectively along with a Sigmoid activation function. Then, a dropout that deactivates 0.25% of neurons is used to prevent overfitting. Due to the use of dense and convolutional layers of Keras themselves, many parameters plus weights are automatically tuned.

The last layer of the proposed model is called the classifier layer. It is responsible for classifying the input image into one of the existing classes. For all the languages except Chinese, the classifier layer has 10 neurons, while for the Chinese language, it has 15 neurons. Figure 4 illustrates the overall architecture of the proposed model for the Arabic database as an example.

### 3.5 Transfer learning

As mentioned at the beginning of this section, the size of the images in the Arabic and MNIST-MIX databases was smaller than that of the Chinese and English databases. The small size of the input images is an important limiting factor in extending the number of layers in the network and hence developing a deeper structure with higher discrimination power. As shown in Figure 4, after using three layers of convolution and three layers of max-pooling, only one feature remained that led to further elaboration and forced us to increase the number of convolution layers. On the other hand, we aimed to design a multilingual system capable of recognising digits of handwritten images at different sizes. If the image size can be enlarged, it may be easier to recognise the

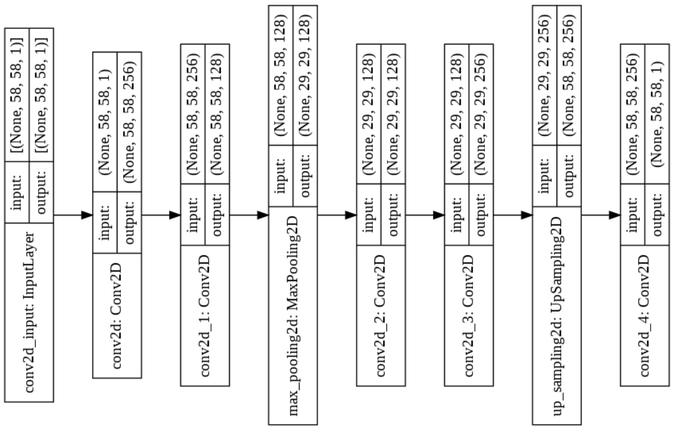



Fig. 5: Proposed autoencoder architecture.

digit inside the image. However, as the size of the images increases, their quality decreases. This is a major challenge in designing an efficient multilingual recognition system. In order to overcome this problem, we propose to use an autoencoder to increase the quality of images of databases where the quality of handwritten images is low.

Transfer learning is improving learning in a new task through transferring knowledge from a related task that has already been learned. It aims to improve learning in the target task by leveraging knowledge from the source task. It is a popular approach in deep learning where pre-trained models are used as the starting point on a new learning and classification task. The main advantage of transfer learning is that the learning process can be much faster, is more accurate and requires less training data.

We used deep autoencoder layers within the transfer learning model, the details of which are explained here. We used the images of the English database (USPS) for training the autoencoder. Due to the large size of the images in the English database, we can easily shrink the images using down-sampling. These images are considered as our original data. Lower resolution images are created out of this images with the fraction of 0.4 with respect to the original images. Lower resolution images are fed as input to the autoencoder. The autoencoder tries to increase the resolution of the images in each iteration of the training to reach the original resolution. As shown in Figure 5, the encoder side of the autoencoder includes two layers of convolution with 256 and 128 filters and $3 \times 3$ filter size, along with a max-pooling layer. At the decoder side, two convolution layers with 128 and 256 filters have been used. Further, there is an upsampling layer at the end of the autoencoder layers. Using this technique can increase the accuracy and the performance of the algorithm.

## 4 Experimental Results

In order to validate the effectiveness of the proposed system, we conducted extensive experiments. All six handwritten databases, introduced in section 3.1, were used in our experiments. For parameters tuning, we used the Chinese database with 15,000 samples, the Arabic database with 70,000 samples, and the English database with more than 9,000 samples. We have used Google Colab to implement and evaluate the proposed method. Google Colab is a free and powerful collaborative tool to run machine learning models. It has powerful hardware options such as GPU and TPU. We used Python version 3.7 to implement the proposed method. Next, the details of the parameters tuning are presented. Finally, the obtained results of each module along with performance metrics of the entire system are presented.

### 4.1 Parameters tuning

In order to achieve a model with optimal performance, the parameters like the numbers of convolution layers, convolution layer filters, dense layer neurons, and optimiser type need to be identified and tuned.

**Number of convolution layers:** It is important to find the number of convolution layers that lead to the highest accuracy across all databases with various languages. At the starting point, we used the Arabic database to evaluate the models and to check what parameters can provide highest recognition performance. This database has the smallest image sizes among all databases used in this study. Therefore, it is rational to claim that if a model can properly run on this database, it can be successfully applied to the other databases, too. Also, the model can be trained more quickly due to the smaller size of the Arabic database images. Moreover, A max-pooling layer is necessary after each convolutional layer of databases with large image sizes such as Chinese and English databases. The main reason is to reduce the computational complexity. As mentioned in section 3 and seen in Figure 4, only one feature is left to be added in the the convolutional layer again for the Arabic database, after using three convolutional layers and three max-pooling layers. Therefore, the maximum number of convolutional layers is selected as 3, followed by a max-pooling layer after each one.

**Number of convolution layer filters and dense layer neurons**: In order to find the appropriate number of filters, we have tested different values of the number of filters and neurons for each layer. A series of experiments were conducted with Arabic, Chinese and English databases under various conditions. Table 1 shows the accuracies for different values of the numbers of the convolutional layer filters and dense layer neurons with the Arabic database. The number of test data considered in this database is 10,000, and the number of training data is 60,000. Out of 60,000 training data, we considered 20% of the samples for validation. We also set the batch size to be 16 and the number of epochs to be 10. Then, six best models from Table 1 are considered

Table 1: The accuracy of different models on Arabic database.

| #of Conv. layer filters | #of first dense layer neurons | #of second dense layer neurons | Test accuracy |
|---|---|---|---|
| 32 | 64 | 64 | 98.72% |
| | 128 | 64 | 98.72% |
| | 128 | 128 | 98.7% |
| | 256 | 128 | 98.62% |
| | 256 | 256 | 98.72% |
| 64 | 64 | 64 | 98.87% |
| | 128 | 64 | 98.69% |
| | 128 | 128 | 98.8% |
| | 256 | 128 | **98.97%** |
| | 256 | 256 | 98.61% |
| 128 | 64 | 64 | 98.86% |
| | 128 | 64 | 98.7% |
| | 128 | 128 | **98.9%** |
| | 256 | 128 | 98.8% |
| | 256 | 256 | **98.88%** |
| 256 | 64 | 64 | 98.82% |
| | 128 | 64 | **98.94%** |
| | 128 | 128 | **98.92%** |
| | 256 | 128 | **98.92%** |
| | 256 | 256 | 98.82% |

Table 2: The accuracy of models on Chinese database.

| #of Conv. layer filters | #of first dense layer neurons | #of second dense layer neurons | Accuracy |
|---|---|---|---|
| 64 | 256 | 128 | 99.26% |
| 128 | 128 | 128 | 99.22% |
| 128 | 256 | 256 | 99.33% |
| 256 | 128 | 64 | 99.01% |
| 256 | 128 | 128 | 99.01% |
| 256 | 256 | 128 | 99.09% |

for training using the Chinese and English database. These results are given in Tables 2 and 3. For convolution layers, we evaluated the performance for different values between 256 to 32 to determine the number of filters. Also, we examined several values between 256 to 64 to determine the number of neurons in dense layers.

To obtain the final model, we have conducted another experiment. We tested all six models on three Arabic, Chinese and English databases. Then, the results were averaged and analysed. Table 4 shows the average accuracy of each model on three databases. The results in this table indicate comparable performances in each model. Moreover, as seen from Table 4, the first model achieved the highest average accuracy (98.52%), and can be considered as a candidate for the final model. Also, in Table 4, the variations among accuracies of different models are very small which confirm the robustness of our model against changes of the order of the databases in the parameter tuning module. Finally, the average accuracy of the final model on Arabic, Chinese and English

Table 3: The accuracy of models on English database.

| **#of Conv. layer filters** | **#of first dense layer neurons** | **#of second dense layer neurons** | **Accuracy** |
|---|---|---|---|
| 64 | 256 | 128 | 97.33% |
| 128 | 128 | 128 | 97.42% |
| 128 | 256 | 256 | 96.8% |
| 256 | 128 | 64 | 96.73% |
| 256 | 128 | 128 | 96.93% |
| 256 | 256 | 128 | 97.13% |

Table 4: The average accuracy of all six models on three databases.

| **#of Conv. layer filters** | **#of first dense layer neurons** | **#of second dense layer neurons** | **Avg. Accuracy** |
|---|---|---|---|
| 64 | 256 | 128 | **98.52**% |
| 128 | 128 | 128 | 98.51% |
| 128 | 256 | 256 | 98.33% |
| 256 | 128 | 64 | 98.22% |
| 256 | 128 | 128 | 98.28% |
| 256 | 256 | 128 | 98.38% |

Table 5: The accuracy of the final model on different databases.

| | **Arabic** | **Chinese** | **English** |
|---|---|---|---|
| **Accuracy** | 98.97% | 99.26% | 97.33% |

Table 6: The accuracy of the final model by different optimisers on Arabic language.

| **Optimiser** | **Accuracy** |
|---|---|
| Adam | **98.97%** |
| Adamax | 98.84% |
| RMSprop | 98.4% |
| SGD | 98.65% |

databases is shown in Table 5. Inspection of the results in these tables (1-5) revealed that the model with 64 convolutional layer filters, 256 first-dense and 128 second-dense layer neurons can be determined as the final model.

**Optimiser** is one of the important parameters that we can give to Keras and its correct choice increases the accuracy of the model is the type of optimiser. Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Table 6 shows the accuracy of the final model on the Arabic database when using any of these optimisers: Adam, Adamax, RMSprop, and SGD. As shown in Table 6, the model with Adam optimiser has the best accuracy. Therefore, we used this optimiser for our model.

Table 7: The accuracy of proposed Language Recognition model on all databases.

| database | Accuracy |
|---|---|
| Arabic, Chinese and English databases | **99.99%** |
| All six handwritten databases | 99.79% |

Table 8: The accuracy of proposed model with and without use of language recognition model on Persian and Arabic databases.

| Approach | Accuracy |
|---|---|
| With language recognition model | **99.08%** |
| Without language recognition model | 97.92% |

### 4.2 Recognition Performance

As mentioned in Section 3, one part of our work was to recognise the language associated to the input images. By correctly recognising the language, we were consequently able to forward the image to the right digit recognition model. In order to demonstrate scalability of the proposed method, we used six languages to run our model; English, Chinese, Arabic, Persian, Urdu, and Kannada. Hence, we first trained the system with all six different languages, and then tested our trained model. As shown in Table 7, using language recognition and digits recognition maintained the system performance at a high level. The average accuracy of the proposed multi-lingual system was negligibly dropped compared to the case where separate languages were used. The reason for this slight decrease in accuracy can be attributed to the similarity between some digits in different languages.

As shown in Figure 2b, 3a, and 3c, the image of digit "three" is relatively identical in Persian, Arabic, and Urdu languages. Therefore, the language recognition may fail in some rare cases. However, we believed that even if the language is incorrectly recognised, the accuracy of digit recognition and hence the overall performance is preserved. The reason is that in most cases (e.g. number 3) the digit has the same interpretation in all these languages. Therefore, the digits recognition stage performs properly independent of the misclassification in language recognition stage.

To verify the effects of language recognition model on languages with relatively similar alphabets, we trained our digit recognition model with two Arabic and Persian databases with (and without) the language recognition module. According to Table 8, the model accuracy was reduced by more than 1% when language recognition model was removed. This result reiterates the important role of language recognition. If all six languages are used without language recognition, the performance is significantly reduced due to increased number of classes.

### 4.3 Effects of using transfer learning

As mentioned in Section 3, an autoencoder was created within transfer learning using the English database as a benchmark to increase image resolution. Our proposed system was capable of detecting low-quality images with small size to be forwarded to the transfer learning module image enhancement. Here, we represent the results associated to Arabic database as an example which has low-quality images with size $28 \times 28$. As shown in Table 9, if we resize the images of Arabic database to $58 \times 58$, the accuracy of the proposed CNN model remains approximately constant on this database. However, by using the transfer learning technique, the accuracy of the proposed CNN model on the Arabic database increases to 90%.

The close structure of images of English and Arabic digits is the main reason for using the English database as a benchmark to train autoencoder. Also, the large size of English handwritten digit images and the ability to reduce their size with down-sampling is considered as another reason for using the English database. As shown in Figure 2, the structure of Chinese numerals is very different from that of English or Arabic digits. For this reason, the accuracy of the CNN model may be decreased by training the autoencoder with a Chinese database to increase the resolution of the Arabic database images. As shown in Table 9, the accuracy of the proposed CNN model on the Arabic database was higher when the autoencoder used the English database for training. Also, the reason for not using the transfer learning technique on English and Chinese databases was the lack of suitable databases that shared a similar structure to these two databases.

As stated in section 3.1, we also created a mixed set of synthetic digit images in English, Persian, and Arabic to further evaluate the effectiveness of transfer learning for image quality enhancement. This synthetic dataset was used to train the autoencoder. For comparison, we calculated the accuracy of Arabic digits recognition when the autoencoder was trained using three different databases, i.e. Chinese, Synthetic, and English. The results of this experiment are tabulated in Table 10. According to this table, the accuracy was slightly increased when synthetic images were used instead of Chinese handwritings. However, this is still lower than that for the autoencoder trained by the English handwritten database. We believe that this slight difference is due to the different structure of handwritten digits and synthetic digits. Also, the results of this experiment support the gained performance when transfer learning is used for image quality enhancement.

In addition to using the transfer learning technique for increasing the quality of Arabic database images, we also used the same approach for Persian, Urdu, and Kannada databases in which their corresponding images sizes were as small as $28 \times 28$. We used the same autoencoder trained by the English database for other databases. As shown in Table 16, the proposed method, which uses the transfer learning technique, achieved higher accuracy than the LeNet model. We examined different database for benchmarking and train-

Table 9: The accuracy of proposed CNN model on Arabic database using transfer learning.

| database | Accuracy |
|---|---|
| Arabic database | 98.97% |
| The resized Arabic database using simple upsampling | 98.99% |
| The resized Arabic database using transfer learning | **99.18%** |

Table 10: The accuracy of proposed CNN model on Arabic database with different autoencoder.

| database | Accuracy |
|---|---|
| The resized Arabic database using autoencoder with Chinese images | 98.17% |
| The resized Arabic database using autoencoder with synthetic digits images | 99.03% |
| The resized Arabic database using autoencoder with English images | **99.18%** |

ing our transfer learning model and observed that the best performance was provided by the English database.

### 4.4 Comparative Performance

In this subsection, we compared the performance of our proposed model with those of related methods. As mentioned in Section 3, we used six available databases to evaluate the performance of our model. However, most of our evaluations were focused on the Arabic database (due to low resolution images) to obtain model parameters based on a worst case scenario. Table 11 shows the confusion matrix for 10,000 Arabic database test data. The number of correct and incorrect predictions is given as a percentage with broken down numbers for each class. The recall and precision measures are calculated as:

$$Precision_i = \frac{M_{ii}}{\sum_i M_{ij}} \tag{1}$$

$$Recall_i = \frac{M_{ii}}{\sum_j M_{ji}} \tag{2}$$

where $M_{ii}$ represents the $i$-th diagonal element in Table 11 corresponding to $i$-th class. $M_{ij}$ and $M_{ji}$ are off-diagonal elements in upper and lower triangles, respectively. As shown in Figure 2b, the shape of digit 0 is very similar to 1 and 5 in Arabic handwriting. Therefore, the precision of these three digits is less than that of others. Also, the similarity of digit 2 to 3 causes a digit from 2 to be misrecognised in class digit 3.

Table 11: Confusion matrix for Arabic database.

| | | Classifier results | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Truth data | | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **Recall** |
| | **0** | 983 | 5 | 1 | 0 | 0 | 9 | 2 | 0 | 0 | 0 | **98.3%** |
| | **1** | 10 | 987 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | **98.7%** |
| | **2** | 3 | 2 | 991 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | **99.1%** |
| | **3** | 1 | 1 | 5 | 987 | 1 | 0 | 3 | 2 | 0 | 0 | **98.7%** |
| | **4** | 1 | 5 | 5 | 0 | 989 | 0 | 0 | 0 | 0 | 0 | **98.9%** |
| | **5** | 7 | 0 | 4 | 0 | 4 | 977 | 0 | 4 | 0 | 4 | **97.7%** |
| | **6** | 0 | 1 | 0 | 0 | 1 | 0 | 996 | 0 | 1 | 1 | **99.6%** |
| | **7** | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 999 | 0 | 0 | **99.9%** |
| | **8** | 2 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 996 | 0 | **99.6%** |
| | **9** | 0 | 0 | 1 | 0 | 1 | 2 | 2 | 0 | 2 | 992 | **99.2%** |
| | **Precision** | **97.61%** | **98.6%** | **98.31%** | **99.9%** | **99.19%** | **98.58%** | **99.3%** | **99.3%** | **99.5%** | **99.4%** | |

Table 12: Classification results on Arabic handwritten digits.

| **Method** | **Accuracy** |
|---|---|
| SRC [45] | 97.13% |
| LeNet-5 [47] | 88% |
| DPL [46] | 95.34% |
| InDPL [27] | 96.7% |
| VGG16 [48] | 98.1% |
| ResNet101 [49] | 97.2% |
| Proposed model without transfer learning | **98.97%** |
| Proposed model with transfer learning | **99.18%** |

Next, we compared the performance of the proposed model with other relevant techniques, particularly with that of models which were based on deep learning. As shown in Table 12, the results of three dictionary learning methods (SRC [45], DPL (dictionary pair learning) [46], and InDPL (incoherent dictionary pair learning) [27]), three CNN models (LeNet-5 [47], VGG16 [48], ResNet101 [49], and the proposed model with and without transfer learning are obtained in this experiment. Among all these methods, the proposed model achieved the highest accuracy. Moreover, VGG16 showed the closest accuracy to the proposed model. The main reason of observed lower performance with VGG or ResNet is that these models are pre-trained with natural images of large sizes, mainly for the object detection tasks, and not for the classification of handwritten digits with small-size images.

A second Arabic handwritten database called HAND2020 was also tested using the proposed method. According to Table 13, the CNN method, which was proposed in the original study with HAND2020, has achieved the recognition accuracy of 99.76% [44]. Their model was trained using 850 epochs to achieve this accuracy. However, as seen in this table, our proposed model has

Table 13: Classification results on HAND2020 database.

| Method | Accuracy |
|---|---|
| CNN model in [44] | 99.76% |
| Proposed model | **99.92%** |

Table 14: Classification results on Chinese handwritten digits.

| Method | Accuracy |
|---|---|
| SRC [45] | 96.28% |
| DLSI [50] | 97.8% |
| LC-KSVD1 [51] | 95.23% |
| LC-KSVD2 [51] | 95.24% |
| DPL [46] | 93.14% |
| InDPL [27] | 94.23% |
| Proposed model | **99.26%** |

Table 15: Classification results on English handwritten digits.

| Method | Accuracy |
|---|---|
| SRC [45] | 81.81% |
| DLSI [50] | 96.13% |
| LC-KSVD1 [51] | 91.25% |
| LC-KSVD2 [51] | 91.1% |
| DPL [46] | 96.68% |
| InDPL [27] | 97.17% |
| Proposed model | **97.33%** |

Table 16: Classification results of two different methods on MNIST-MIX handwritten numbers databases.

| Database | LeNet model | Proposed model |
|---|---|---|
| Persian [41] | 98.13% | **98.99%** |
| Urdu [43] | 97.3% | **98.23%** |
| Kannada [42] | 85.7% | **88.01%** |

achieved superior performance only after ten epochs. We believe that precise parameters adjustment used for different languages and using language recognition module in the proposed model, has a significant impact on obtaining a higher performance.

We also tested the proposed model on Chinese and English databases to further evaluate the performance of the model. As shown in Tables 14 and 15, the proposed model had the highest accuracy among all methods, which showed the high efficiency of our model.

Finally, Table 16 demonstrates the performance results of the proposed method on three MNIST-MIX databases. As shown, the proposed method had better accuracy than LeNet model.

## 5 Conclusion

In this paper, a multi-lingual approach was proposed for handwritten digit recognition. The proposed system is composed of two main modules: language recognition and digit recognition. In this innovative approach, which is based on a CNN architecture, our system first identified the language of input image and then decided the best model parameters for recognition of digits. Moreover, we proposed the use of transfer learning to unify the quality of images coming from different databases. This allows for a robust and consistent performance among various handwritten languages. We conducted extensive experiments to tune and optimise the parameters for achieving a superior performance. The results of our experiments with six different languages showed a high accuracy among other relevant techniques, even compared to those based on CNN structure. Utilizing transfer learning in the proposed model is one of the main reasons of achieving superior performance to previous models such as [44]. Also, in our model, the precise adjustment of the parameters for different languages had a great impact on achieving a high recognition accuracy.

As future work, we aim to extend this model to recognise multi-lingual handwritten characters (letters). Furthermore, we attempt to work on a procedure to set up a system to find the best hyperparameters, number of layers, with less computational burden.

## References

1. Fei Ye and Adrian G Bors. Learning joint latent representations based on information maximization. *Information Sciences*, 567:216–236, 2021.
2. Plamen P Angelov and Xiaowei Gu. Deep rule-based classifier with human-level performance and characteristics. *Information Sciences*, 463:196–213, 2018.
3. Xinjun Peng, De Chen, and Dong Xu. Hyperplane-based nonnegative matrix factorization with label information. *Information Sciences*, 493:1–19, 2019.
4. Xiaowei Gu, Plamen P Angelov, and José C Príncipe. A method for autonomous data partitioning. *Information Sciences*, 460:65–82, 2018.
5. Xiaowei Gu and Plamen P Angelov. Self-organising fuzzy logic classifier. *Information Sciences*, 447:36–51, 2018.
6. Zeng Yu, Tianrui Li, Guangchun Luo, Hamido Fujita, Ning Yu, and Yi Pan. Convolutional networks with cross-layer neurons for image recognition. *Information Sciences*, 433:241–254, 2018.
7. Marziye Rahmati, Mansoor Fateh, Mohsen Rezvani, Alireza Tajary, and Vahid Abolghasemi. Printed persian ocr system using deep learning. *IET Image Processing*, December 2020.
8. Deepika Gupta and Soumen Bag. Cnn-based multilingual handwritten numeral recognition: A fusion-free approach. *Expert Systems with Applications*, 165:113784, 2021.
9. U. Pal, N. Sharma, T. Wakabayashi, and F. Kimura. Handwritten numeral recognition of six popular indian scripts. *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, 2:749–753, 2007.
10. U. Bhattacharya and B. B. Chaudhuri. Handwritten numeral databases of indian scripts and multistage recognition of mixed numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):444–457, 2009.
11. Abdelhak Boukharouba and Abdelhak Bennia. Novel feature extraction technique for the recognition of handwritten digits. *Applied Computing and Informatics*, 13(1):19 – 26, 2017.

12. S. Mozaffari, K. Faez, and H. R. Kanan. Recognition of isolated handwritten farsi/arabic alphanumeric using fractal codes. In *6th IEEE Southwest Symposium on Image Analysis and Interpretation, 2004.*, pages 104–108, 2004.
13. M. Hanmandlu and O.V. Ramana Murthy. Fuzzy model based recognition of handwritten numerals. *Pattern Recognition*, 40(6):1840 – 1854, 2007.
14. Savita Ahlawat and Rahul Rishi. Off-line handwritten numeral recognition using hybrid feature set–a comparative analysis. *Procedia Computer Science*, 122:1092 – 1099, 2017. 5th International Conference on Information Technology and Quantitative Management, ITQM 2017.
15. Kalyan S. Dash, Niladri B. Puhan, and Ganapati Panda. Unconstrained handwritten digit recognition using perceptual shape primitives. *Pattern Anal. Appl.*, 21(2):413–436, May 2018.
16. K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th International Conference on Computer Vision*, pages 2146–2153, 2009.
17. M. Shopon, N. Mohammed, and M. A. Abedin. Image augmentation by blocky artifact in deep convolutional neural network for handwritten digit recognition. In *2017 IEEE International Conference on Imaging, Vision Pattern Recognition (icIVPR)*, pages 1–6, 2017.
18. Cuong Tuan Nguyen, Vu Tran Minh Khuong, Hung Tuan Nguyen, and Masaki Nakagawa. Cnn based spatial classification features for clustering offline handwritten mathematical expressions. *Pattern Recognition Letters*, 131:113 – 120, 2020.
19. Apurva A. Desai. Gujarati handwritten numeral optical character reorganization through neural network. *Pattern Recognition*, 43(7):2582 – 2589, 2010.
20. Rasoul Ameri, Ali Alameer, Saideh Ferdowsi, Vahid Abolghasemi, and Kianoush Nazarpour. Classification of handwritten chinese numbers with convolutional neural networks. In *2021 5th International Conference on Pattern Recognition and Image Analysis (IPRIA)*, pages 1–4, 2021.
21. Zan Gao, Leming Guo, Weili Guan, An-An Liu, Tongwei Ren, and Shengyong Chen. A pairwise attentive adversarial spatiotemporal network for cross-domain few-shot action recognition-r2. *IEEE Transactions on Image Processing*, 30:767–782, 2021.
22. Peiguang Jing, Yuting Su, Liqiang Nie, and Huimin Gu. Predicting image memorability through adaptive transfer learning from external sources. *IEEE Transactions on Multimedia*, 19(5):1050–1062, 2017.
23. Le Zhang. A transfer learning approach for handwritten numeral digit recognition. In *Proceedings of the 3rd International Conference on Software Engineering and Information Management*, ICSIM '20, page 140–145, New York, NY, USA, 2020. Association for Computing Machinery.
24. Adeline Granet, Emmanuel Morin, Harold Mouchère, Solen Quiniou, and Christian Viard-Gaudin. Transfer Learning for Handwriting Recognition on Historical Documents. In *7th International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, Madeira, Portugal, January 2018.
25. Kamalpreet Kaur, Renu Dhir, and Kuldeep Kumar. Transfer learning approach for analysis of epochs on handwritten digit classification. In *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)*, pages 456–458, 2021.
26. Meng Liu, Liqiang Nie, Xiang Wang, Qi Tian, and Baoquan Chen. Online data organizer: Micro-video categorization by structure-guided multimodal dictionary learning. *IEEE Transactions on Image Processing*, 28(3):1235–1247, 2019.
27. Vahid Abolghasemi, Mingyang Chen, Ali Alameer, Saideh Ferdowsi, Jonathon Chambers, and Kianoush Nazarpour. Incoherent dictionary pair learning: Application to a novel open-source database of chinese numbers. *IEEE Signal Processing Letters*, 25(4):472–476, 2018.
28. Ekachai Phaisangittisagul and Rapeepol Chongprachawat. Post-processing of unsupervised dictionary learning in handwritten digit recognition. In *2014 14th International Symposium on Communications and Information Technologies (ISCIT)*, pages 166–170, 2014.

29. Meng Yang, Lei Zhang, Xiangchu Feng, and D. Zhang. Sparse representation based fisher discrimination dictionary learning for image classification. *International Journal of Computer Vision*, 109:209–232, 2014.
30. Aboozar Ghaffari, Mahdi Kafaee, and Vahid Abolghasemi. Smooth non-negative sparse representation for face and handwritten recognition. *Applied Soft Computing*, 111:107723, 2021.
31. Tai-Shan Yan, Yong-Qing Tao, and Du-Wu Cui. Research on handwritten numeral recognition method based on improved genetic algorithm and neural network. In *2007 International Conference on Wavelet Analysis and Pattern Recognition*, volume 3, pages 1271–1276, 2007.
32. Min-Rong Chen, Bi-Peng Chen, Guo-Qiang Zeng, Kang-Di Lu, and Ping Chu. An adaptive fractional-order bp neural network based on extremal optimization for handwritten digits recognition. *Neurocomputing*, 391:260–272, 2020.
33. A. Montazeri, M. Shamsi, and R. Dianat. Using a new approach in deep dictionary learning to handwriting number classification. In *2020 25th International Computer Conference, Computer Society of Iran (CSICC)*, pages 1–8, 2020.
34. M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Sig. Process.*, 54(11):4311–4322, 2006.
35. Rahul Pramanik, Prabhat Dansena, and Soumen Bag. A study on the effect of cnn-based transfer learning on handwritten indic and mixed numeral recognition. In Suresh Sundaram and Gaurav Harit, editors, *Document Analysis and Recognition*, pages 41–51, Singapore, 2019. Springer Singapore.
36. Subhadip Basu, Nibaran Das, Ram Sarkar, Mahantapas Kundu, Mita Nasipuri, and Dipak Kumar Basu. A novel framework for automatic sorting of postal documents with multi-script address blocks. *Pattern Recognition*, 43(10):3507 – 3521, 2010.
37. Mohammad Reduanul Haque, Md. Gausul Azam, Sarwar Mahmud Milon, Md. Shaheen Hossain, Md. Al-Amin Molla, and Mohammad Shorif Uddin. Quantitative analysis of deep cnns for multilingual handwritten digit recognition. In M. Shamim Kaiser, Anirban Bandyopadhyay, Mufti Mahmud, and Kanad Ray, editors, *Proceedings of International Conference on Trends in Computational and Cognitive Engineering*, pages 15–25, Singapore, 2021. Springer Singapore.
38. Weiwei Jiang. MNIST-MIX: a multi-language handwritten digit recognition dataset. *IOP SciNotes*, 1(2):025002, August 2020.
39. Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.
40. S. Abdelazeem and E. El-Sherif. Modified arabic digits database.
41. Hossein Khosravi and Ehsanollah Kabir. Introducing a very large dataset of handwritten farsi digits and a study on their varieties. *Pattern recognition letters*, 28(10):1133–1141, 2007.
42. Vinay Uday Prabhu. Kannada-mnist: A new handwritten digits dataset for the kannada language. *arXiv preprint arXiv:1908.01242*, 2019.
43. Hazrat Ali, Ahsan Ullah, Talha Iqbal, and Shahid Khattak. Pioneer dataset and automatic recognition of urdu handwritten characters using a deep autoencoder and convolutional neural network. *SN Applied Sciences*, 2(2):1–12, 2020.
44. Pratik Ahamed, Soumyadeep Kundu, Tauseef Khan, Vikrant Bhateja, Ram Sarkar, and Ayatullah Faruk Mollah. Handwritten arabic numerals recognition using convolutional neural network. *Journal of Ambient Intelligence and Humanized Computing*, 11(11):5445–5457, 2020.
45. John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2008.
46. Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Projective dictionary pair learning for pattern classification. *Advances in neural information processing systems*, 27:793–801, 2014.
47. Ahmed El-Sawy, EL-Bakry Hazem, and Mohamed Loey. Cnn for handwritten arabic digits recognition based on lenet-5. In *International conference on advanced intelligent systems and informatics*, pages 566–575. Springer, 2016.

48. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
49. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
50. Ignacio Ramirez, Pablo Sprechmann, and Guillermo Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3501–3508. IEEE, 2010.
51. Zhuolin Jiang, Zhe Lin, and Larry S Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *CVPR 2011*, pages 1697–1704. IEEE, 2011.