Advance Access publication on 1 September 2021
https://doi.org/10.1093/comjnl/bxab117

Long Multi-digit Number Recognition from Images Empowered by Deep Convolutional Neural Networks

Muhammad Asif*[1], Maaz Bin Ahmad[2], Shiza Mushtaq[3], Khalid Masood[4], Toqeer Mahmood[5] and Arfan Ali Nagra[6]

[1]*Department of Computer Science, Lahore Garrison University, Lahore, Pakistan*
[2]*College of Computing and Information Sciences, PAF KIET, Karachi, Pakistan*
[3]*Department of Computer Science, Lahore Garrison University, Lahore, Pakistan*
[4]*Department of Computer Science, Lahore Garrison University, Lahore, Pakistan*
[5]*Department of Computer Science, National Textile University, Faisalabad, Pakistan*
[6]*Department of Computer Science, Lahore Garrison University, Lahore, Pakistan*
**Corresponding author: drmuhammadasif@lgu.edu.pk*

Scanning images and converting the scanned information into digital format is an active research area. Scanning is an automated, fast and efficient process as compared to the traditional data entry, and the resultant converted data is more accurate. Recognizing digits from the scanned images is a challenging task. To address this issue, most of the existing techniques perform multiple individual steps that are localization, segmentation and recognition. Some researchers also focused on adopting a unified approach that combined these three steps for multi-digit recognition of up to five digits. To cope with the modern requirements, a unified multi-digit recognition technique capable of recognizing more than five digits is the need of the hour. Considering this necessity, a unified multi-digit recognition approach is presented in the current study that can recognize sequences up to 18 digits long. The proposed technique is based on a deep convolutional neural network algorithm that performs two basic functions. First, it localizes and extracts the region of interest in the image, and then it performs multi-digit recognition. The proposed algorithm recognizes sequences of up to 18 characters that makes it one of the preferred recognition techniques among the existing algorithms. The proposed technique is compared with state-of-the-art techniques and is proved to be superior and robust. The experiments are performed on two datasets, and overall accuracy up to 98% is achieved.

Keywords: multi-digit recognition; deep learning; CNN; MNIST; SVHN

Received 1 August 2020; Revised 30 April 2021; Accepted 13 June 2021
Handling editor: Suchi Bhandarkar

1. INTRODUCTION

Deep learning has become a vital area in data science in the past few years [1]. It is a sub-field of artificial intelligence (AI) in which models are trained on convolutional neural network (CNN) features instead of hand-crafted features to perform classification tasks [2–9]. The CNN features are extracted by the neural network architecture directly from video, images, text or sound. In computer vision and image processing domains, several remarkable applications have been developed by using CNN such as speech recognition, image and pattern recognition, object detection, automated image colorization, vehicle classification, image segmentation, text detection, traffic signs recognition, lane classification and many more [10–19].

Multi-digit number recognition from snapshots belongs to the optical character recognition (OCR) domain [20–22]. OCR can be utilized to build numerous applications, such as house number recognition, license plate recognition, digitization of handwritten documents, automatic passport, social security number reading and many more. The success of such applications is highly correlated with the performance of the recognition technique. Multi-digit recognition from snapshots is still a challenging task because of the large variety in the visual appearance of digits due to a wide range of styles, colors, fonts, character arrangement and orientation. The complexity of the multi-digit number recognition problem further increases due to environmental conditions like occlusions, shadows and lighting along with image acquisition factors including focus blurs, camera resolution, lens distortion and motion [21]. To exemplify said issues, a few sample images from the Street View House Numbers (SVHN) dataset [23] are given in Fig. 1.

Several techniques exist to recognize digits from images. Some approaches focus only on single-digit recognition and others target multi-digit recognition. Several researchers [24–28, 30, 31] presented handwritten digit recognition techniques from digital images. These techniques are evaluated on MNIST handwritten digit dataset [32]. Figure 2 shows the sample images from MNIST handwritten digit dataset [33].

In the field of multi-digit recognition, there are several schemes [21, 28–31, 34–43] targeting the recognition of handwritten numbers, telephone numbers, vehicle identification numbers (VINs), player jersey number and house number from the images. In the literature, most of the proposed techniques for multi-digit recognition adopted localization, segmentation and recognition as individual steps. However, currently, the researchers are focusing on unified approaches by combining the localization, segmentation and recognition steps for multi-digit recognition. The limitation of existing unified techniques is that these are evaluated on SVHN dataset having images with sequence length up to five digits.

Currently, the requirements for multi-digit recognition have been changed from smaller to longer sequences, e.g. bank cheque and IBAN numbers, account numbers, social security numbers, passport numbers, etc. To cope with the modern requirements, multi-digit recognition techniques capable of recognizing larger sequences are required. Considering this fact, a unified deep learning-based multi-digit recognition approach is presented in this work that is capable of recognizing longer sequences up to 18 digits precisely. To recognize multi-digits from the printed documents, the proposed technique firstly performs the region of interest extraction and localization operations thereafter executes the process of multi-digit recognition. Experimental results indicate that the proposed multi-digit recognition technique achieved 98% accuracy that is reasonably higher in comparison with other state-of-the-art techniques in the literature. The prime contributions of the proposed multi-digit recognition technique are summarized as under:

- The proposed technique can precisely recognize multi-digits from the printed documents after the region of interest extraction.

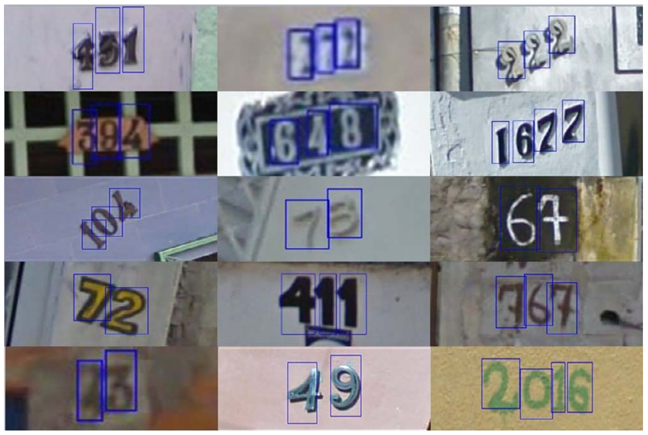

FIGURE 1. Sample images from SVHN database.

FIGURE 2. Sample images from MNITS test dataset [21].

- The proposed technique is capable of recognizing multi-digits sequences up to 18 digits accurately.
- The integration of three steps, i.e localization, segmentation and recognition reduces the overall computational complexity.
- Extensive experiments performed on two different datasets show the effectiveness of the proposed technique and the obtained results are compared with the existing methods.

The rest of the article is structured as follows. Section 2 discusses the work done in this area. Section 3 describes the proposed unified multi-digit number recognition technique. Experimental analysis is performed in section 4. Finally, Section 5 concludes the article.

2. RELATED WORK

In the literature, several techniques have been proposed for the recognition of numbers from images. These techniques can be classified from several aspects, e.g. based on the dataset used, based on AI methods applied, based on single-digit or multi-digit recognition capability, etc. Even in these classifications, there may exist sub-classified techniques. For example, classification based on the dataset used can be further classified as a handwritten dataset or printed document dataset. In this paper, the classification of recognition techniques is performed with respect to their single-digit or multi-digit recognition capability. Multi-digit recognition techniques are further sub-classified as either unified or non-unified techniques. The main categories of digit recognition techniques are as follows:

2.1. Single-digit recognition techniques

Kevin *et al.* [24] proposed a two-stage system based on SIFT (oriented edge filters applied to a small patch) and PMK-SVM (operates on features vectors) approach on a single digit using MINST and NORB datasets, which achieved an accuracy of 62.9% and 65%, respectively. Ranzato *et al.* [26] proposed an unsupervised method for learning approach on single-digit using MINST and Caltech 101 datasets and the achieved accuracy was 54%. Ranzato *et al.* [27] proposed an energy-based model for unsupervised learning of sparse over-complete representations. They used the MINST dataset and achieved 60% accuracy. Yang and Pu [28] proposed a technique for the recognition of handwritten multi-digits for mobile devices with the help of CNN, and achieved 99.07% top-1 single digit accuracy on the MNIST database. In this approach, after pre-processing, the image is segmented into digit patches, which are then fed to the CNN for digit recognition.

Nguyen *et al.* [29] proposed the hybrid CNN-GRU model for highly efficient handwritten digit recognition. They used a combination of CNN and gate recurrent units (GRU), where GRU replaces the fully connected layer part of CNN to improve the overall accuracy at low computational complexity. They utilized MNIST dataset containing 70,000 images to evaluate the model and claimed 99.21% accuracy. Shovon *et al.* [30] presented a multi-layer CNN-based technique to recognize the handwritten Bangla number. They used the NumtaDB dataset having images of 85000+ digits to evaluate the performance of the technique. They achieved 98.96% accuracy. Islam *et al.* [31] suggested a handwritten digits recognition approach based on Artificial Neural Network (ANN). They extracted 28,000 images of digits from the MNIST dataset and achieved an overall accuracy of 99.60%. Table 1 summarizes the single-digit recognition techniques.

2.2. Multi-digit recognition techniques

2.2.1. *Non-unified techniques*

These techniques address this problem in multiple individual steps, i.e localization, segmentation and recognition. Ciresan *et al.* [25] proposed a deep CNN-based approach for multi-digit recognition using multiple datasets. They achieved the best accuracy of 72% for the GTSRB dataset. They emphasized that the combination of multiple different DNN columns into multiple columns DNN (MCDNN) further decreases the error rate by 30–40%. Yamaguchi *et al.* [34] proposed a telephone numbers recognition system from signboards images. They also presented a skew and slant correction technique for digit recognition. In this work, initially, digits are extracted from the image. After that, directional features are acquired from the extracted digits and finally pattern matching is performed. This technique gives correct digit extraction and recognition rate up to 99.2% and 98.8%, respectively, on single digit from 1332 signboards images.

Shah *et al.* [35] proposed a technique for chess number identification or VIN from snapshots. In this technique, characters are segmented from the embossed VIN and are fed to the OCR module that exploited the ANN. Netzer *et al.* [36] proposed a technique to recognize digits from real-world images with the help of unsupervised feature learning procedures. They exploited numerous existing unsupervised feature learning techniques including stacked sparse auto-encoders and *k*-means-based systems to learn features in the training phase from the training data. These techniques produce a fixed-length feature vector from the input image that is used to train the SVM classifier. The proposed technique combined three stages: detection, segmentation and classification. It gives 90.6% accuracy for SVHN digit publicly available database. Sermanet *et al.* [37] exploited a CNN to learn features from the pixels of the images for classification of house number digits instead of using handcrafted features to reach human-level performance. This technique provides 95.10% accuracy for the SVHN dataset, which is 4.5% better than Netzer [36].

Goodfellow *et al.* [38] presented maxout model to enhance the performance of dropout's fast approximate model averaging methodology and to facilitate optimization by dropout. This model achieves 97.53% accuracy on single-character recognition for the SVHN dataset. Jeon *et al.* [39] presented a system for real-time multi-digit recognition. They exploited deep learning for embedded system and evaluated the system using MNIST handwritten dataset along with some self-created data. The system obtained 98.23% accuracy. Liu and Bhanu [40] proposed a pose-guided region-based CNN (R-CNN) approach to recognize the jersey number of players in sports. They used a self-generated dataset of 3567 images having 6293-digit instances. They claimed 94.09% recognition accuracy. Cubuk *et al.* [41] highlighted that existing techniques of learned augmentation have systematic disadvantages, i.e. increase in training complexity and computational cost. To address these issues, a technique is proposed that reduces search space and allows it to be trained on actual tasks instead of having a separate proxy task. The authors reported that the proposed technique outperformed the existing automated augmentation techniques on ImageNet, SVHN and CIFAR-10/100 datasets.

TABLE 1. Summary of single-digit recognition techniques.

Paper	Approach	Datasets (accuracy)	Remarks
[24]	Two-stage system based on SIFT and PMK-SVM	MNIST (62.9 %) & NORB (65%)	**Pros:** The use of two successive stages of feature extraction make it more effective. **Cons:** The performance of supervised convolutional networks on Caltech 101 dataset is inadequate due to the over parameterization.
[26]	Unsupervised method for learning	MNIST & Caltech 101 (54%)	**Pros:** The technique is applicable for scenarios where lack of labeled dataset exists. **Cons:** The accuracy of the technique is quite low and is not suitable to apply in real-time scenarios.
[27]	Unsupervised method for learning	MNIST (60%)	**Pros:** There is no pre-processing stage in this technique that makes it faster. It can be useful in scenarios where labeled data is not available. **Cons:** The noisy images can degrade the recognition performance.
[28]	CNN	MINST (99.07%)	**Pros:** The human level accuracy of the technique demonstrates its superiority and makes it highly suitable for real-time scenarios. **Cons:** The use of canny edge detector in pre-processing stage may increase the computational complexity of the technique.
[29]	CNN-GRU	MINIST (99.21%)	**Pros:** The high accuracy of the technique makes it highly appropriate for real-time applications.
[30]	CNN	NumtaDB (98.96%)	**Limitation:** The scope of technique is limited to Bangla numbers only.
[31]	ANN	MNIST (99.60%)	**Pros:** The high accuracy of the technique makes it highly appropriate for real-time applications.

2.2.2. *Unified techniques*

These techniques perform detection, segmentation and classification in a single step. Goodfellow *et al.* [21] presented a technique that combined the detection, segmentation and classification steps with the help of a deep CNN. This approach gives 96.03% accuracy on the dataset for recognizing complete street numbers. This technique can only recognize complete street numbers up to five digits. Zhong *et al.* [42] targeted SVHN to develop a digital recognition system based on Deep Convolutional Generative Adversarial Networks (DCGANs). The approach recognizes sequence numbers without splitting digits. Firstly, they used CNN for character features extraction. After that, a CNN was built for digits recognition. To improve the recognition rate up to 95.56%, they used DCGAN that helped to enhance the resolution of the number. Liu *et al.* [43] developed an architecture to recognize multi-digit sequences using CNN encoder and long short-term memory (LSTM) decoder. In the design of the CNN encoder, the dropout and batch normalization processes are used to avoid overfitting. The main purpose of using LSTM is to address exploding and vanishing gradients. The model achieved 92.53% accuracy on the SVHN dataset. Table 2 summarizes the multi-digit recognition techniques.

In nutshell, most of the existing multi-digit recognition techniques are non-unified, i.e. they address recognition problems in multiple separate steps, i.e localization, segmentation and recognition. In addition, there also exist some unified approaches that perform localization, segmentation and recognition in a single step. The main limitation of existing unified approaches is their focus on short sequences length up to five digits. To cope with the modern requirements, a unified multi-digit recognition technique capable of recognizing more than five digits is need of the hour. Considering this necessity, a unified multi-digit recognition approach is presented in the current study that can recognize up to 18 digits sequences.

3. PROPOSED MULTI-DIGIT NUMBER RECOGNITION SYSTEM

The block diagram of the proposed multi-digit recognition system is shown in Fig. 3. The entire system consists of two main stages: (i) a region of interest selection stage that locates individual multi-digit number sequence in a large image and (ii) a recognition stage that performs multi-digit recognition. The numerous steps of the system are presented in the following subsections.

3.1. Problem description

Multi-digit number recognition from a photograph is an exceptional type of sequence recognition. Let us consider an image

TABLE 2. Summary of multi-digit recognition techniques.

Paper	Approach	Non-unified/ unified	Datasets (accuracy)	Remarks
[25]	Deep CNN	Non-unified	MNIST (41%), NIST SD 19 (30-80%), CASIA-HWDBI (26%), CASIA-OLHWDBI (35%), CIFAR 10 (39%), GTSRB (72%) & NORB (46%)	**Pros:** Variety of datasets has been used to evaluate the technique. **Cons:** The overall accuracy of the technique on different datasets is not up to the mark.
[34]	Skew and slant method	Non-unified	Self-generated (98.8%)	**Pros:** The accuracy of the system is high. The use of sparse hierarchical features, which are locally shift invariant, makes it reliable. **Cons:** The invariant feature extraction process makes this system complex.
[35]	OCR using ANN	Non-unified	Characters of VIN (95.49%)	**Pros:** Gives high value for correct identification rate. It can be used to develop automatic VIN recognition application. **Cons:** Even a single-digit false recognition of VIN could be catastrophic.
[36]	Unsupervised feature learning - SVM classifier	Non-unified	SVHN (90.6%)	**Pros:** Extensive evaluation of the system is done both on handcrafted and learned features. It can be applicable for scenarios where lack of labeled dataset exists. **Cons:** The system performance lags well behind the human performance.
[37]	CNN	Non-unified	SVHN (95.10%)	**Pros:** Automatically learn a unique set of features. Its performance is better than unsupervised learning methods (kmeans, auto-encoders). **Cons:** Though the performance is better but still less than human learning.
[38]	Maxout networks	Non-unified	MNIST (99.55%), CIFAR-10 (90.62%), CIFAR- 100 (61.43%) & SVHN (97.53%)	**Pros:** This method has valuable attributes both for model averaging and optimization with dropout. The method is evaluated on four standard datasets. **Cons:** The maxout activation has higher number of trainable parameters.
[21]	Deep CNN	Unified	SVHN (96% (Sequence), 97.84% (Per digit)) & reCAPTCHA (99.8%)	**Limitation:** This technique can only recognize complete street number up to five digits.
[42]	DCHAN	Unified	SVHN (95.56%)	**Limitation:** The use of multiple CNNs demands more computation power.
[43]	CNN & LSTM	Unified	SVHN (92.53% (Sequence), 97.88% (Per digit))	**Limitation:** The use of both CNN and LSTM makes it computationally intensive.

'I' consisting of a multi-digit sequence, as shown in Fig. 4. The task is to identify the multi-digit sequence 'S' in an image. Suppose the sequence 'S' is composed of multiple digits, i.e. $S = \{D_1, D_2, ..., D_N\}$. Where '$N$' indicates the length of the sequence. For correct sequence recognition, every digit 'D_i' should be correctly predicted.

3.2. Pre-processing

Pre-processing is an essential step in computer vision and pattern recognition problems. It is a collection of techniques that boost the quality of an input image and removes unwanted entities such as noise from the images. The importance of pre-processing can be established from the fact that it removes impurities from the input data and enhances the quality of forthcoming processes on the image, i.e. segmentation, feature extraction, classification or recognition. In this work, localization of individual multi-digit number sequence is manually performed through cropping. The input images fed to the system may vary in size. To establish a base size for all images, resizing is performed using a bi-cubic interpolation over 4×4 pixel neighborhood [44]. To eliminate the noise from the input

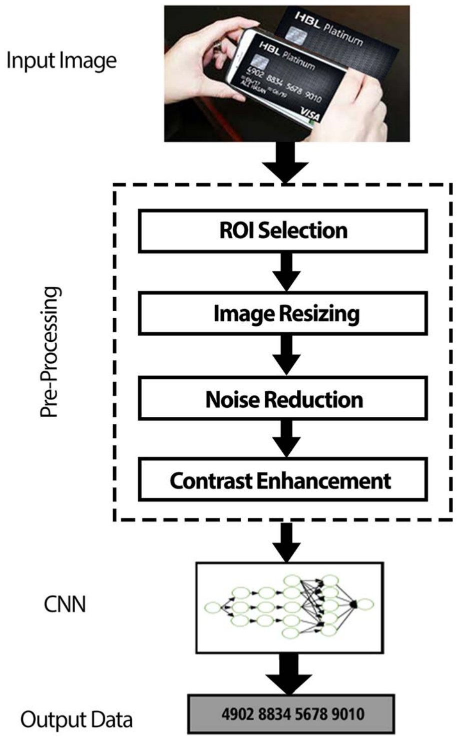

FIGURE 3. Proposed system block diagram.

FIGURE 4. Image with multi-digit sequence.

image, a Gaussian blur (also known as Gaussian smoothing) operation is performed with kernel size 5×5 [45]. An adaptive histogram equalization at 8×8 block size is performed to improve the contrast of the input image [46].

3.3. Probabilistic model for sequence

Let 'I' indicate the input image and 'S' denote the output sequence. The task is to train a probabilistic model $P(S|I)$ that maximizes the $\log P(S|I)$ on training data. To model sequence 'S', it is described as a set of 'N' discrete random variables $\{D_1, D_2, ..., D_N\}$ that indicates sequence elements. The sequence length 'L' is also considered as a random variable. It is supposed that there is no dependency among digits of sequence, so the likelihood of a certain sequence $S = \{D_1, D_2, ..., D_N\}$ can be calculated through Eq. 1.

$$P(S|I) = P(L = N|I)\Pi_{j=1}^{N} P(D_j|I). \tag{1}$$

There are small possible values for each discrete random variable. Each digit variable has 10 probable values ranging from 0 to 9. On the other hand, there are 18 possible values for random variable $L(0, 1, 2, ..., 17)$. Hence, it is possible to indicate each random variable with a softmax classifier. The classifier takes features as an input which are extracted from an image 'I' using CNN. In this work, a random variable 'F' is used to represent these features. The value of the random variable 'F' is known, provided 'I'. In this model, $P(S|I) = P(S|F)$. A stochastic gradient descent (SGD) is used to train a model that maximizes the $logP(S|I)$ on the training data. Softmax model for each discrete random variable exploits the similar backpropagation learning rule, except the missing digit case. For each test image, the model predicts the sequence.

$$S(L, D_1, D_2, ..., D_L) = argmax_{(L,D_1,D_2,...,D_L)} logP(S|I) \tag{2}$$

The argmax can be linearly calculated. It can be calculated independently for each digit. For a sequence of length 'L', the running sum of the digit log probabilities plus $\log P(L|I)$ gives the complete log probability.

3.4. CNN architecture

The CNN architecture comprises eight convolutional layers and two fully connected or dense layers as depicted in Fig. 5. The convolutional and fully connected layers are denoted with CVL and FCL, respectively. The layer number is indicated by a subscript, e.g. FCL_1 denotes the first fully connected layer. All the connections are made in a forwarding direction and go from the current to the next layer. To add the non-linearity, ReLU activation function [47] is applied to the outputs of all convolutional and fully connected layers. The utilization of ReLUs in this network not only fastens the training process several times but it also permits to proceed deeper in case of vanishing gradient problems [48].

Table 3 provides the details about the layers of CNN. The first, second, third and fourth convolutional layers (CVL_1, CVL_2, CVL_3 and CVL_4) filter the input with 48, 64, 128 and 160 kernels of size 5×5, respectively. In rest of the four layers (CVL_5, CVL_6, CVL_7 and CVL_8), 192 kernels of size 5×5 are used to filter the input. Each convolutional layer incorporates max pooling and subtractive normalization. The window of size 2×2 is used for max pooling. To maintain the representation size, zero padding is used before performing each convolution operation. The value of stride is adjusted in such a way that for two consecutive convolutional layers, it alternates between 2 and 1. It means that half of the layers do not decrease the spatial size of the representation. Overall, there are eight convolutional layers and all of these use padding uses padding, max pooling and dropout regularization layer to drop random units from the network to avoid over-fitting. The dropout layer drops the neurons with a likelihood of $1 - p$, where p indicates the likelihood of retained neurons. The dropped-out neurons have no role in both forward and backward pass, i.e. all the connections belonging to dropped-out neurons are detached

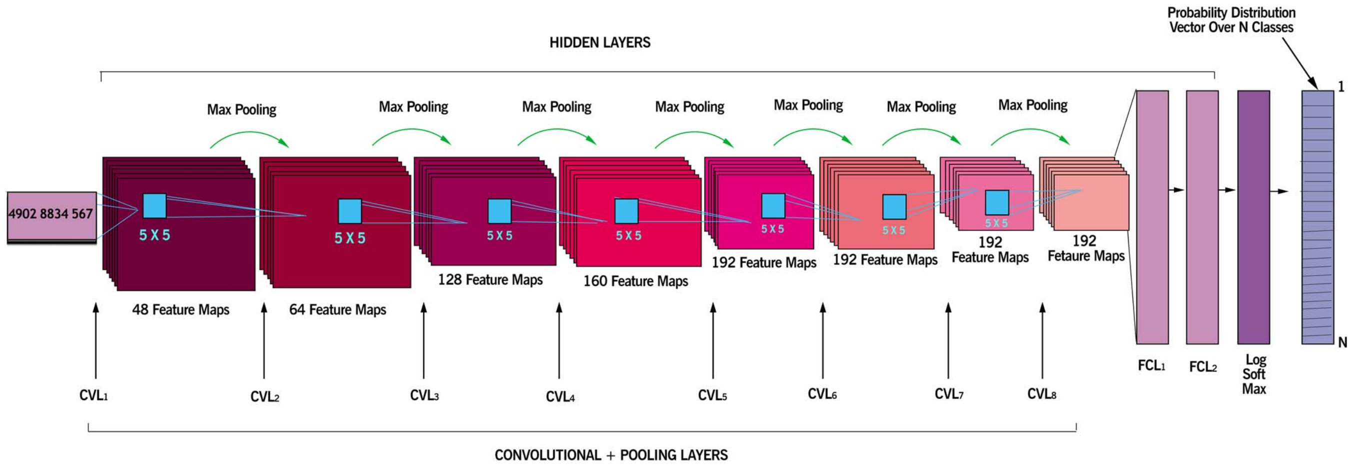

FIGURE 5. DCNN architecture used for the multi-digit recognition.

TABLE 3. CNN configuration.

Layer no.	Layer type	No. of Kernals	Kernel Size	Padding	Max pooling	Stride	Drop out
1		48	5×5	2	2×2	2	Yes
2		64	5×5	2	2×2	1	Yes
3	Convolutional	128	5×5	2	2×2	2	Yes
4	layer	160	5×5	2	2×2	1	Yes
5		192	5×5	2	2×2	2	Yes
6		192	5×5	2	2×2	1	Yes
7		192	5×5	2	2×2	2	Yes
8		192	5×5	2	2×2	1	Yes
9	Fully						
10	connected						
11	Softmax						

in a training stage. After the completion of the training stage, the weights of dropped-out neurons are restored to their initial value for the upcoming training stage. In the testing phase, all of the neurons are utilized without dropping them. However, to maintain the balance between the expected values of neurons in the training and testing phase, a weighting factor p is used.

Layers 9 and 10 are fully connected layers that consist of the ReLU activation function [47] to add non-linearity. The output of the last fully connected layer (FCL_2) is fed to the N-way softmax function, which gives the probability of N class labels. In this work, the value of N is set to 18 for the classification of images up to 18 digits. Hence, the probability vector of size 1×N (where each vector element corresponds to a class of dataset) is obtained. 'Log softmax' is used as a softmax function that scales the vector elements in the range of (0, 1). The sum of vector elements 1. In this way, it allocates probability distribution to each class.

To initialize the weights of all layers, Gaussian distribution with zero mean and 0.01 standard deviation is used. The biases belonging to CVL_2, CVL_4, CVL_6, CVL_8, FC_1 and FC_2 are initialized with constant '1' while for the remaining layers CVL_1, CVL_3, CVL_5 and CVL_7, the value is set to '0'. For all iterations, the learning rate is kept constant i.e., 0.0001.

The SGD with back-propagation (entropy loss) is exploited to maximize the multinomial logistic regression objective function during the training process. SGD algorithm is frequently used to train neural networks and it performs well in learning discriminative linear classifiers with a convex loss function, e.g. Logistic Regression, SVM, etc. The two main benefits that motivate us to utilize SGD are efficiency and ease to implement with the provision of network-tuning options including rate decay, learning rate, number of iterations, etc. Some drawbacks of using SGD are its requirements of hyper-parameters including regularization parameters and the number of iteration/epochs.

Eq. 3 is used to update the SGD parameters for each training sample x_i and label y_j:

$$\psi = \psi - \lambda.\nabla_{\psi} Z(\psi; x_i, y_j), \quad (3)$$

where 'Z' represents the objective function that SGD will optimize (multinomial logistic regression in this work) and ψ indicates DCNN model parameters, i.e. biases and weights.

The backpropagation technique is mainly used to train ANNs along with some optimization strategy, e.g. SGD. In this technique, gradients are computed with respect to all parameters as the input updates these while trying to minimize the objective function. This technique needs the actual class label to calculate the gradients. To compute the gradients with respect to the loss function, the chain rule is applied iteratively. The backpropagation technique consists of three main phases including forwarding pass, backward pass and gradients computation in case of parameters (weights and biases) [49, 50]. The derivatives are not computed for those layers that do not have any parameters like pooling layers.

Forward pass: In this phase, data is forwarded to calculate all $J's$, where J is a function of input x. Its mathematical representation is given in Eq. 4:

$$J^{k+1} = Q(J^k), \tag{4}$$

where k denotes layer number, and $J = f(x_i)$

Backward Pass: In this phase, data is moved to determine all derivative's (δ's) of loss function w.r.t $J's$. Eq. 5 gives its mathematical representation:

$$\delta_i^k = \frac{dL}{dJ_i^k} = \Sigma_j \frac{dL}{dJ_i^k}.\frac{dJ_i^{k+1}}{dJ_i^k} = \Sigma_j \delta_j^{k+1}\left(\frac{dJ_j^{k+1}}{dJ_i^k}\right), \tag{5}$$

where i indicates layer number, j denotes input sample, and L represents the loss function. Eq. 5 recursively attempts to minimize the loss function.

Derivative with respect to parameters:

For the layers having parameters, Eq. 6 demonstrates the computation of derivatives of cost function w.r.t parameters:

$$\frac{dJ}{d\psi^k} = \Sigma_j \frac{dL}{dJ_j^{k+1}}.\frac{dJ_i^{k+1}}{d\psi^k} = \Sigma_j \delta_j^{k+1}\left(\frac{dJ_j^{k+1}}{d\psi^k}\right), \tag{6}$$

where layer parameters (weights and biases) are represented with ψ. Figure 6 portrays backpropagation for a neural network with K sequentially connected layers and parameters. Each layer takes loss value L from the output layer (loss layer) and back propagates it. Eqs. 4 and 5 are used to generate the forward ($J's$) and backward messages (δ's). The gradient of loss w.r.t to layers parameters ψ is given by $\frac{dE}{d\psi}$.

4. EXPERIMENTAL ANALYSIS

For experimental analysis, NVidia GeForce GTX 1080 Ti graphic card [51] is used. It consists of a GPU with 3584 CUDA cores that operate at 1480 MHz and goes up to 1584 MHz in Boost mode. The GPU accesses an 11.2 GB memory through a 352-bit memory interface. The memory operates at a 1376 MHz clock.

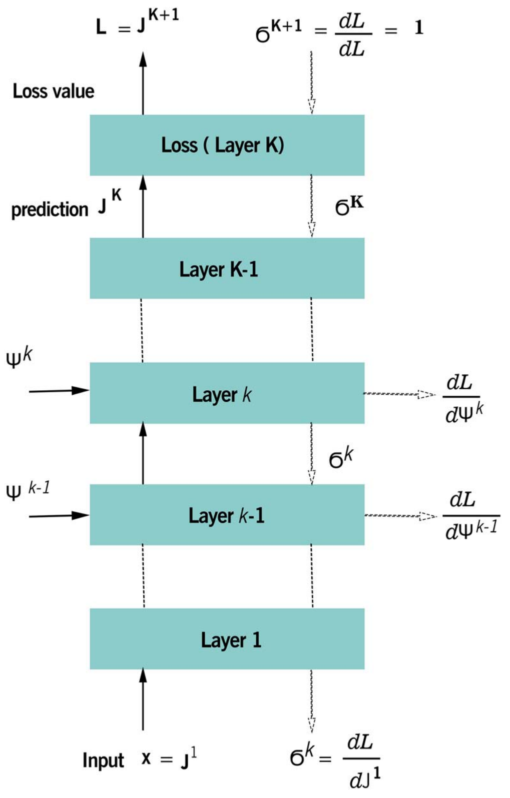

FIGURE 6. Backpropagation Algorithm.

For performance evaluation of the proposed solution, four parameters, i.e. precision, recall, F-Score and accuracy are computed. These parameters are detailed as under:

Precision (P): This is the ratio of the correctly predicted positive values (TP) to the total predicted positive values ($TP + FP$). Eq. 7 is used to compute this metric.

$$Precision(P) = \frac{TP}{TP + FP} \tag{7}$$

Recall (R): This is the ratio of TP to all samples in positive or actual class ($TP + FN$). It can be determined with the help of Eq. 8.

$$Recall(R) = \frac{TP}{TP + FN} \tag{8}$$

F-Score: This is the harmonic mean of the Recall and Precision. The parameter considers both false positives (FP) and false negatives (FN). Eq. 9 is used to calculate this parameter.

$$F - Score(F) = \frac{2 \times P \times R}{R + P} \tag{9}$$

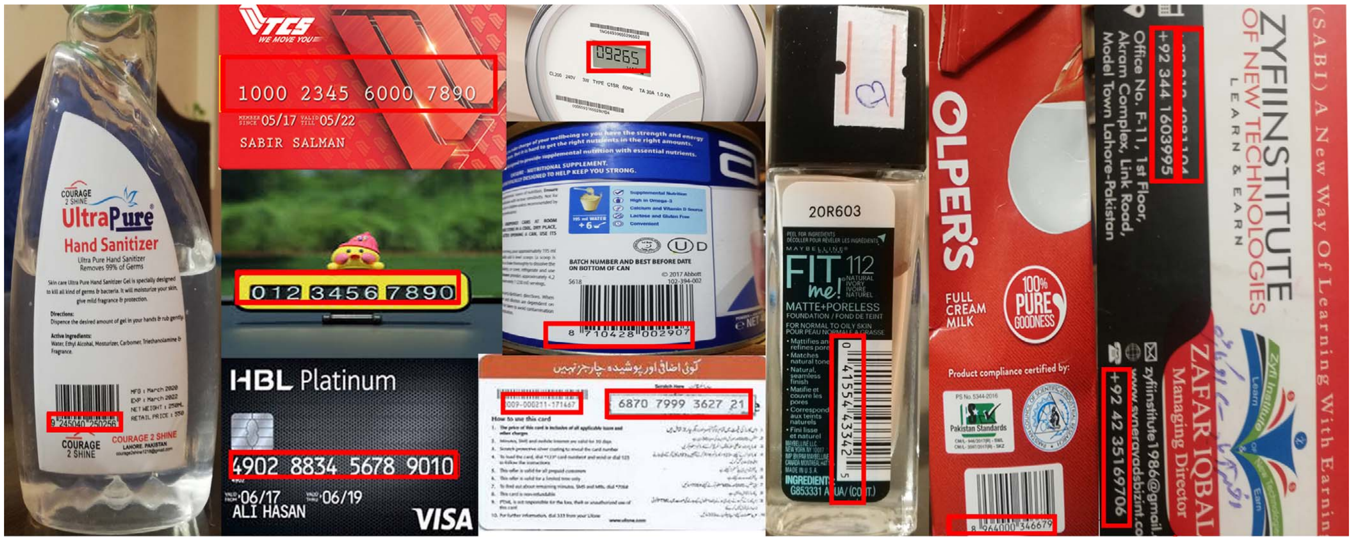

FIGURE 7. Sample captured images from dataset.

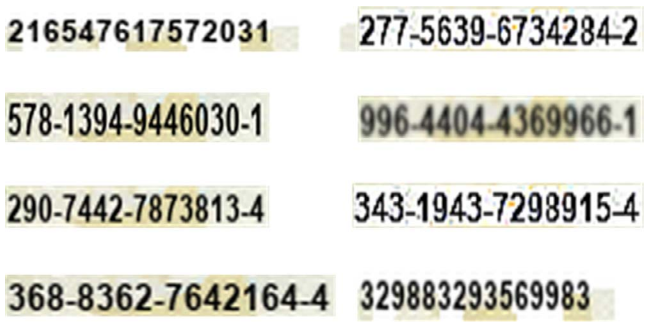

FIGURE 8. Sample synthetic images from dataset.

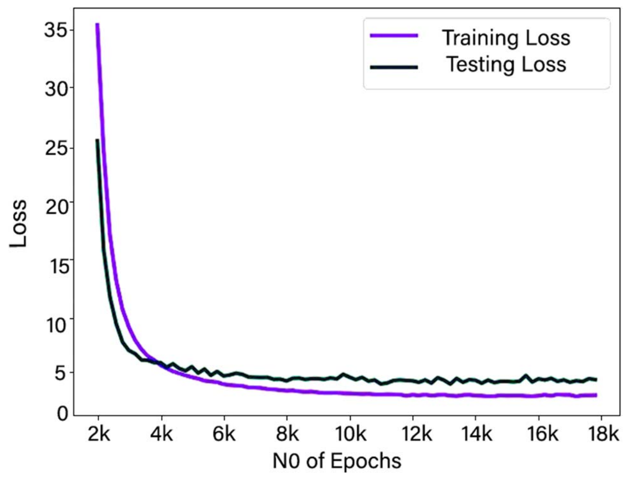

FIGURE 9. Classifier loss vs number of epochs (multi-digit long sequence dataset).

Accuracy: This is a ratio of correctly predicted samples to the total samples. Eq. 10 is used to calculate the accuracy.

$$Accuracy = \frac{TP + TN}{TN + TP + FP + FN} \tag{10}$$

The training of deep neural networks demands a huge amount of data. The publicly available SVHN dataset consists of images with multi-digit sequences up to five digits. To target multi-digit sequences of more than five digits with the help of CNN, data collection becomes the main challenge. This section first describes the creation of a dataset that consists of multi-digit sequences up to 18 digits. After that, experimental results are discussed on a multi-digit long sequence dataset. Finally, the performance of the model is evaluated on the SVHN dataset and its comparison is made with some existing techniques.

4.1. Multi-digit long sequence dataset

Dataset is created by capturing images of daily usage products, items, or common objects that have multi-digit sequences. A total of 10,000 images are captured in different environmental conditions. Figure 7 shows sample captured images of the dataset. Moreover, data is also synthesized by randomly generating sequences in the range of 1-digit length to max 18-digit lengths. For synthetic image generation, a random function is used to create a dictionary that consists of 100 thousand multi-digit sequences. This dictionary is further used to create the dataset by writing multi-digit sequences from it to plain images of size 300×54. For sequence writing on plain images, different combinations of font, style and size are used from the subset

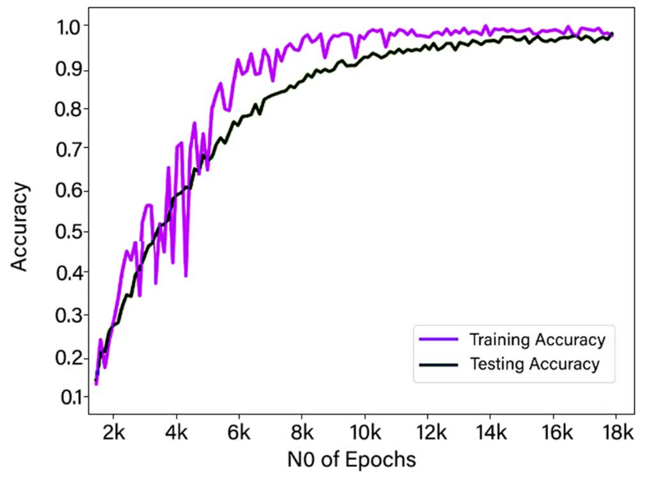

FIGURE 10. Classifier accuracy vs number of epochs (multi-digit long sequence dataset).

of fonts (Arial, Times New Roman, Calibri, San Serif), styles (Regular, Bold, Italic, etc.) and sizes (10, 11, 12, 13, 14, 15). To add more variations in the dataset, filtering (smoothing and sharpening) and morphological operations (erosion, dilation, opening and closing) are performed with varied window sizes (3×3, 5×5 and 7×7) on created images. As a result, 40,000 images are generated. Figure 8 shows some synthetic images of the dataset.

4.2. Training and testing for multi-digit long sequence dataset

The dataset is divided into two parts: one for training and the other for testing. The training set consists of 70% of the dataset and the testing set comprises the rest of the 30%. Data labeling is performed by drawing a small rectangular bounding box that contains individual character bounding boxes. After that, the bounding box is expended by 30% in both the x- and y-direction, the image is cropped to that bounding box and it is resized to the resolution of 310×64 pixels. In the second stage, the bounding box is again cropped to 300×54 pixels using a random location within the 310×64 pixel image. This means that there are several randomly generated boxes for each training sample that may increase the size of the dataset. Without the data augmentation, we may lose about half a percentage point of accuracy. The resolution of the training and validation images is 300×54.

Figure 9 shows that training loss is decreased with the increase in the number of epoch. It is also evident that for the first 1000 epochs, the training loss is very high. As the number of epochs increases, the overall loss decreases. From epochs 2000 to 10000, the training loss decreases rapidly. It is also observed that the classifier converges at 14,000 epochs and for the rest of the epochs, the loss remains almost the same.

Figure 10 shows the accuracy of the model with respect to the number of epochs. It is observed that for 3000 epochs, the testing accuracy is approximately 35%. At 6000 epochs, the accuracy reaches up to 75% and for the rest of the iterations, the classifier converges slowly. At 14,000 epochs, the classifier converges to about 98% accuracy and for the rest of the epochs, the classification accuracy remains almost the same.

Figure 11 illustrates some examples of challenging inputs that are incorrectly recognized.

4.3. Training and testing for public SVHN dataset

The SVHN dataset consists of about 73K core training images. Data labeling is performed by drawing a small rectangular bounding box that contains individual character bounding boxes. After that, the bounding box is expanded by 30% in both the x and the y directions. The image is cropped to that bounding box and it is resized to the resolution of 64×64 pixels. In the second stage, the bounding box is again cropped to 54×54 pixels using a random location within the 64×64 pixel image. This means that there are several randomly generated boxes for each training sample that may increase the size of the dataset. Without the data augmentation, we may lose about 0.5% accuracy. The resolution of the training and validation images is 54×54. The experimental results show that the trained model gives 96.87% accuracy in recognizing complete street numbers.

Table 4 is the comparison placeholder of the proposed work with the existing unified techniques at the SVHN dataset. It indicates that the proposed technique outperforms the existing work for SVHN reorganization at the sequence level. The evaluation of the proposed technique is also made at the character level for the SVHN dataset to compare it with existing character-level recognition techniques. Table 5 shows that 98.05% overall accuracy is achieved.

Table 6 is the comparison placeholder of the proposed work with the existing techniques at the SVHN dataset at the character level. The results show that the proposed technique outperforms the existing character-level recognition techniques for SVHN. Based on the above discussion, it is clear that a deep CNN can recognize and localize multi-digit sequences with decent performance.

5. CONCLUSION

In this research, a multi-digit recognition system is proposed that identifies long sequences of numeric information in the images and converts those to the digital format. There are various stages in the proposed technique and several operations are performed on the input images. Firstly, a dataset with long sequences of digits (up to 18 characters) is created, as the publicly available dataset SVHN contains images of only 5 or fewer digits sequences. Secondly, several pre-processing operations such as image resizing, noise reduction and image

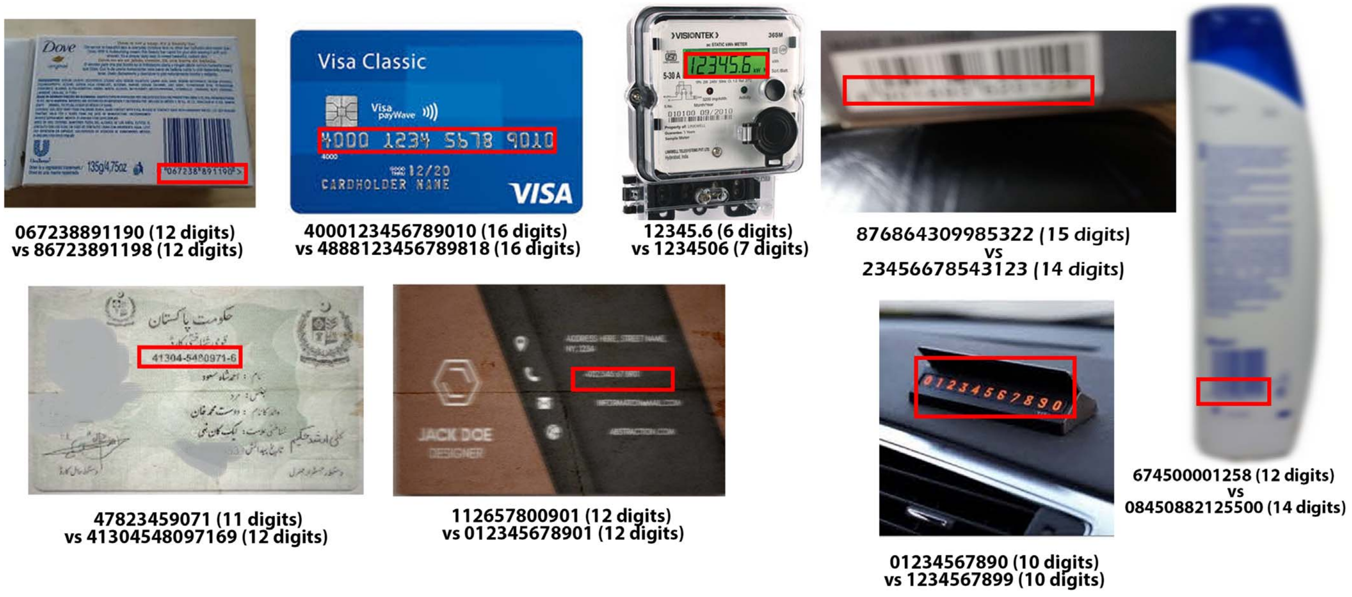
067238891190 (12 digits)
vs 86723891198 (12 digits)
4000123456789010 (16 digits)
vs 4888123456789818 (16 digits)
12345.6 (6 digits)
vs 1234506 (7 digits)
876864309985322 (15 digits)
vs
23456678543123 (14 digits)
47823459071 (11 digits)
vs 41304548097169 (12 digits)
112657800901 (12 digits)
vs 012345678901 (12 digits)
01234567890 (10 digits)
vs 1234567899 (10 digits)
674500001258 (12 digits)
vs
08450882125500 (14 digits)

FIGURE 11. Examples of incorrectly transcribed sequences (transcription vs ground truth).

TABLE 4. Comparison with unified approaches.

Technique	Accuracy
Liu *et al.* [43]	92.53%
Zhong *et al.* [42]	95.56%
Goodfellow *et al.* [21]	96.0%
Proposed	96.87%

TABLE 5. Results for SVHN dataset.

Precision	Recall	F-score	Accuracy
0.98	0.97	0.98	0.9805

enhancement are performed. For recognition, the state-of-the-art CNN with fine-tuned hyper-parameters is designed that achieved 98% classification accuracy on both of the datasets. The proposed technique can be used in several applications such as the reading of numeric data from passports, scanning of bank cheques, information in the visiting cards, etc. In the future, in addition to the digits, handwriting scripts and signatures would also be converted into a digital format. For signatures, a decoding algorithm can be proposed that will decode the alphabets of the signature and can reproduce the identity from the signature.

TABLE 6. Comparison with non-unified approaches.

Technique	Accuracy
Netzer *et al.* [36]	90.60%
Sermanet *et al.* [37]	95.10%
Goodfellow *et al.* [38]	97.53%
Liu *et al.* [43]	97.88%
Proposed	98.05%

5. DATA AVAILABILITY

The data underlying this article are available in Github (https://github.com/System-CTL/Dataset).

REFERENCES

[1] LeCun, Y., Bengio, Y. and Hinton, G. (2015) Deep learning. *Nat. Int. J. Sci.*, 512, 436–444.

[2] Bosse, S., Maniry, D., Muller, K.-R., Wiegand, T. and Samek, W. (2018) Deep neural networks for no-reference and full-reference image quality assessment. *IEEE Trans. Image Process.*, 27, 206–219.

[3] Jin, K.H., McCann, M.T., Froustey, E. and Unser, M. (2017) Deep convolutional neural network for inverse problems in imaging. *IEEE Trans. Image Process.*, 26, 4509–4522.

[4] Lezoray, O., Charrier, C., Cardot, H. and Lefevre, S. (2008) Machine learning in image processing. *EURASIP J. Adv. Signal Process.*, 927950, 1–4.

[5] Gavat, I. and Militaru, D. (2015) Deep Learning in Acoustic Modeling for Automatic Speech Recognition and Understanding: An Overview. In *Int. Conf. on Speech Technology and*

Human-Computer Dialogue (SpeD). Bucharest, Romania, October 14–17, pp. 37–44. IEEE, New York, USA.

[6] Deng, L., Hinton, G. and Kingsbury, B. (2013) New Types of Deep Neural Network Learning for Speech Recognition and Related Applications: An Overview. In *Int. Conf. on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada*, May 26–31. IEEE, New York, USA.

[7] Zhang, Y.-D., Satapathy, S.C., Zhu, L.-Y., Grriz, J.M. and Wang, S.-H. (2020) A seven-layer convolutional neural network for chest CT based COVID-19 diagnosis using stochastic pooling. *IEEE Sens. J*., 1–1.

[8] Zhang, Y.-D. *et al.* (2020) Advances in multimodal data fusion in neuroimaging: overview, challenges, and novel orientation. *Inf. Fusion*, 64, 149–187.

[9] Tahir, M., Taj, I.A., Assuncao, P.A. and Asif, M. (2020) Fast video encoding based on random forests. *J. Real Time Image Process.*, 17, 1029–1049.

[10] Wang, X., Zhang, W., Wu, X., Xiao, L., Qian, Y. and Fang, Z. (2019) Real-time vehicle type classification with deep convolutional neural networks. *J. Real Time Image Process.*, 16, 5–14.

[11] Particke, F., Kolbenschlag, R., Hiller, M., Patio-Studenckiand, L. and Thielecke, J. (2017) Deep Learning for Real-Time Capable Object Detection and Localization on Mobile Platforms. In *IOP Conf. Series: Materials Science and Engineering, (AIAAT 2017), Hawaii, USA*, August 30–September 2, IOP, Bristol, United Kingdom.

[12] Dong, Z., Wu, Y., Pei, M. and Jia, Y. (2015) Vehicle type classification using a semi supervised convolutional neural network. *IEEE Trans. Intell. Transport. Syst.*, 16, 2247–2256.

[13] Carrio, A., Sampedro, C., Rodriguez-Ramos, A. and Campoy, P. (2017) A review of deep learning methods and applications for unmanned aerial vehicles. *Hindawi Journal of Sensors*.

[14] Wang, W., Xie, E., Song, X., Zang, Y., Wang, W., Luy, T., Yu, G. and Shen, C. (2019) Efficient and Accurate Arbitrary-Shaped Text Detection with Pixel Aggregation Network. In *Int. Conf. on Computer Vision (ICCV), Seoul, Korea (South)*, October 27–November 2. IEEE, New York, USA.

[15] Xu, Y., Duan, J., Kuang, Z., Yue, X., Sun, H., Guan, Y. and Zhang, W. (2019) Geometry Normalization Networks for Accurate Scene Text Detection. In *Int. Conf. on Computer Vision (ICCV), Seoul, Korea (South)*, October 27–November 2. IEEE, New York, USA.

[16] Liao, M., Wan, Z., Yao, C., Chen, K. and Bai, X. (2020) Real-Time Scene Text Detection with Differentiable Binarization. In *Proc. of the AAAI Conf. on Artificial Intelligence*, 34, 7, 11474-11481. AAAI, Palo Alto, California, USA.

[17] Zhang, C., Liang, B., Huang, Z., En, M., Han, J., Ding, E. and Ding, X. (2019) Look More Than Once: An Accurate Detector for Text of Arbitrary Shapes. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA*, June 15–20. IEEE, New York, USA.

[18] Tiany, Z., Shuz, M., Lyux, P., Lix, R., Zhoux, C., Shenx, X. and Jiay, J. (2019) Learning Shape-Aware Embedding for Scene Text Detection. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA*, June 15–20. IEEE, New York, USA.

[19] Wang, X., Jiang, Y., Luo, Z., Liu, C., Choi, H. and Kim, S. (2019) Arbitrary Shape Scene Text Detection with Adaptive Text Region Representation. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA*, June 15–20. IEEE, New York, USA.

[20] Asif, M., Ahmad, M.B., Taj, I.A. and Tahir, M. (2018) A generalized multi-layer framework for video coding to select prediction parameters. *IEEE Access*, 6, 25277–25291.

[21] Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S. and Shet, V. (2013) Multi-digit number recognition from street view imagery using deep convolutional neural networks. [Online] arXiv preprint, arXiv:1312.6082 (accessed July 23, 2020).

[22] Asif, M., Taj, I.A., Ziauddin, S.M., Ahmad, M.B. and Raza, A. (2016) An efficient inter prediction mode selection scheme for advanced video coding based on motion homogeneity and residual complexity. *IEEJ Trans. Electr. Electron. Eng.*, 11, 760–767.

[23] The Street View House Numbers (SVHN) Dataset. [Online]. http://ufldl.stanford.edu/housenumbers/ (accessed June 31, 2020).

[24] Jarrett, K., Kavukcuoglu, K., Ranzato, M. and LeCun, Y. (2009) What is the best multi-stage architecture for object recognition? In *Int. Conf. on computer vision, Kyoto, Japan*, September 29–October 2, 2009, pp. 2146–2153. IEEE, New York, USA.

[25] Ciresan, D., Meier, U. and Schmidhuber, J. (2012) Multi-column Deep Neural Networks for Image Classification. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA*, June 16-21, pp. 3642–3649. IEEE, New York, USA.

[26] Ranzato, M.A., Huang, F.J., Boureau, Y.-L. and LeCun, Y. (2007) Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition, Minneapolis, MN, USA*, June 17–22, pp. 1–8. IEEE, New York, USA.

[27] Ranzato, M.A., Poultney, C., Chopra, S. and LeCun, Y. (2006) Efficient learning of sparse representations with an energy-based model. In *The 19th Int. Conf. on Neural Information Processing Systems*, pp. 1137–1144. MIP Press, Cambridge, Massachusetts, USA.

[28] Yang, X. and Pu, J. (2015) MDig: Multi-digit recognition using convolutional nerual network on mobile. Stanford [Online]. https://web.stanford.edu/class/cs231m/projects/final-report-yang-pu.pdf (accessed June 2, 2020).

[29] Nguyen, V., Cai, J. and Chu, J. (2019) Hybrid CNN-GRU Model for High Efficient Handwritten Digit Recognition. In *The 2nd Int. Conf. on Artificial Intelligence and Pattern Recognition, August 2019*, pp. 66–71.

[30] Shovon, M.M.I., Kamruzzaman, M. and Kundu, M.K. (2020) Recognition of Handwritten Bangla Number Using Multi Layer Convolutional Neural Network. In *IEEE Region 10 Symposium (TENSYMP)*, June 5–7, 2020, Dhaka, Bangladesh. IEEE, New York, USA.

[31] Islam, K.T., Mujtaba, R., T. R. G and Nweke, H.F. (2017) Handwritten Digits Recognition with Artificial Neural Network. In *Proc. of the Int. Conf. on Engineering Technologies and Technopreneurship*, September 18–20, 2017, Kuala Lumpur, Malaysia.

[32] LeCun, Y., Cortes, C. and Burges, C.J. C. The mnist database of handwritten digits. [Online]. http://yann.lecun.com/exdb/mnist/ (accessed July 23, 2018).

[33] Steppan, J. MnistExamples.png [Online]. https://commons.wikimedia.org/w/index.php?curid=64810040 (accessed July 23, 2020).

[34] Yamaguchi, T., Nakano, Y., Maruyama, M., Miyao, H. and Hananoi, T. (2003) Digit Classification on Signboards for Telephone Number Recognition. In *IEEE 7th Int. Conf. on Document Analysis and Recognition (ICDAR)*, pp. 359–363. IEEE, New York, USA, Edinburgh, UK.

[35] Shah, P., Karamchandani, S., Nadkar, T., Gulechha, N., Koli, K. and Lad, K. (2009) OCR-based Chassis-Number Recognition Using Artificial Neural Networks. In *IEEE Int. Conf. on Vehicular Electronics and Safety (ICVES), Pune, India*, November 11–12, pp. 31–34. IEEE, New York, USA.

[36] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B. and Ng, A. (2011) Reading Digits in Natural Images with Unsupervised Feature Learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, pp. 1–9. NIPS, San Diego, CA, USA.

[37] Sermanet, P., Chintala, S. and LeCun, Y. (2012) Convolutional Neural Networks Applied to House Numbers Digit Classification. In *Int. Conf. on Pattern Recognition (ICPR 2012), Tsukuba, Japan*, November 11–15, pp. 3288–3291. IEEE, New York, USA.

[38] Goodfellow, I.J., Warde-Farley, D., Mirza, M., Courville, A. and Bengio, Y. (2013) Maxout Networks. In *Proc. of the Int. Conf. on Machine Learning*, Atlanta, Georgia, USA, June 17–19. DBLP Computer Science Bibliography.

[39] Jeon, H.-M., Nguyen, V.D. and Jeon, J.-W. (2018) Real-Time Multi-Digit Recognition System Using Deep Learning on an Embedded System. In *12th Int. Conf. on Ubiquitous Information Management and Communication, Langkawi, Malaysia*, January 5–7, 2018, pp. 1–6. ICPS Proceeding. Association for Computing Machinery, New York, NY, USA.

[40] Liu, H. and Bhanu, B. (2019) Pose-Guided R-CNN for Jersey Number Recognition in Sports. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops, Kyoto, Japan*, June 16–17, 2019, pp. 1–10. IEEE, Long Beach, CA, USA.

[41] Cubuk, E.D., Zoph, B., Shlens, J. and Le, Q.V. (2020) RandAugment: Practical Automated Data Augmentation with a Reduced Search Space. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops*. Seattle, WA, USA, IEEE, New York, USA.

[42] Zhong, J., Gao, J., Chen, R. and Li, J. (2019) Digital Recognition of Street View House Numbers Based on DCGAN. In *The 2nd Int. Conf. on Image and Graphics Processing*, pp. 19–22. Singapore, Association for Computing Machinery, New York, NY, USA.

[43] Liu, X., Dengy, Y., Sunz, Y. and Zhou, Y. (2018) Multi-digit Recognition with Convolutional Neural Network and Long Short-term Memory. In *The 14th Int. Conf. on Natural Computation, Fuzzy Systems and Knowledge Discovery*, pp. 1187–1192, July 28–30. Huangshan, China, IEEE, New York, USA.

[44] Resize Image: [Online]. https://pythonexamples.org/python-opencv-cv2-resize-image/ (accessed June 2, 2020).

[45] Deng, G. and Cahill, L.W. (1993) An Adaptive Gaussian Filter for Noise Reduction and Edge Detection. In *IEEE Conf. Record Nuclear Science Symposium and Medical Imaging Conference, San Francisco, CA, USA*, October 31–November 6. IEEE, New York, USA.

[46] Pizer, S.M., Amburn, E.P., Austin, J.D., Cromartie, R., Geselowitz, A., Greer, T., Romeny, B.H., Zimmerman, J.B. and Zuiderveld, K. (1987) Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*, 39, 355–368.

[47] Hahnloser, R.H., Sarpeshkar, R., Mahowald, M.A., Douglas, R.J. and Seung, H.S. (2000) Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405, 947–951.

[48] Glorot, X., Bordes, A. and Bengio, Y. (2011) Deep sparse rectifier neural networks. In *Proc. of the 14th Int. Conf. on Artificial Intelligence and Statistics*, *Nature*, Fort Lauderdale, FL, USA, April 11–13.

[49] Qayyum, A., Anwar, S.M., Awais, M. and Majid, M. (2017) Medical image retrieval using deep convolutional neural network. *Neurocomputing*, 266, 8–20.

[50] Nawi, N.M., Ransing, R.S., Salleh, M.N.M., Ghazali, R. and Hamid, N.A. (2010) An Improved Back Propagation Neural Network Algorithm on Classification Problems. In *Int. Conf. on Database Theory and Application, Bio-Science and Bio-Technology*, vol. 118, pp. 177–188. Springer, Berlin, Heidelberg.

[51] Geforce GTX 1080 Ti: [Online]. https://www.nvidia.com/en-us/geforce/products/10series/geforce-gtx-1080-ti/ (accessed June 2, 2020).