Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 55 Issue: 12:2022
DOI10.17605/OSF.IO/KADQV

OFFLINE WRITER IDENTIFICATION USING CNN AND RNN: A DEEP LEARNING APPROACH

DEEPA BENDIGERI

Assistant Professor, Department of Information Science and Engineering, SDM College of Engineering and Technology, Dharwad, India. Corresponding Author Email: deepabendigeri12@gmail.com

Dr. GOPAL A. BIDKAR

Professor Department of Electronics and Communication Engineering, SDM College of Engineering and Technology, Dharwad, India. Email:gabidkar@rediffmail.com

Dr. JAGADEESH D.PUJARI

Professor and Head, Department of Information Science and Engineering, SDM College of Engineering and Technology, Dharwad, India. Email:jaggudp@gmail.com.

Abstract

Pattern recognition and artificial intelligence have a lot to say about how to identify a writer. To identify the writers, conventional methods rely on hand-craft features. Convolutional Neural Networks (CNNs) have recently emerged as the most effective method for classifying Writers. The Recurrent Neural Network (RNN) models the spatial relationship between the sequences of fragments in order to make the local fragment features better at telling them apart. So, this work, suggests CNN and RNN models for identifying offline writers. The proposed method used IAM dataset of offline handwritten english script for experimentation. The proposed method achieved 90% accuracy to classify 690 writers using CNN and 93% of accuracy using RNN. The proposed method can generate efficient and robust writer identification based on different scale, different orientation and both.

Keywords: Writer Identification, Offline Analysis, Convolution Neural Networks, Deep Learning Recurrent Neural Network, IAM Database

1. INTRODUCTION

Writing style is a biometric trait that may be used to authenticate or identify a person. It is possible to identify the author of a piece of writing based on its distinctive writing style, which cannot be imitated. For a variety of purposes, people recognition via the analysis of handwritten texts is used to identify who produced a document, detect fakes and fraud, and identify who authorsied holographic checks, wills, letters, and other documents [1]. It's common for current studies that analyze complicated text structures to extract properties including complete pages, text, and paragraphs, as well as words [5][6][7], and signatures [8][9]. In order to get a high verification rate, it is necessary to work with sources that are quite complex.

In order to determine the authorship of a piece of handwriting, one must use a method known as writer identification. Forensic document analysis and historical manuscript research both benefit from its usage [2]. To determine who has written a piece of handwritten document, conventional approaches look at its form or texture [4]. Lot of images for each sample to produce a statistically valid feature vector. So, the majority of research concentrates on determining who authored something by analyzing page-level pictures of papers containing numerous paragraphs or phrases [3].

In order to identify the author of an unknown piece of documents, a database of reference texts must be compared to the given unknown document. Searching for a collection of design features in the unknown document is how writer identification is accomplished. Text-dependent or text-independent methods of author identification may be used depending on the relationship between the query document and the reference document. It is necessary for the writer to write same text in order to find their identity using the text-dependent technique. A text-dependent approach is also known as signature verification [10]. On the other hand, the text-independent technique recognizes the author of a given document regardless of what is contained in the document. If you want to extract strong characteristics, you will need several samples. In order for a text-free system to function properly, a specific minimum quantity of text must be provided. Online and offline techniques may be divided down into text-dependent and text-independent approaches depending on how handwriting is learnt. For example, the online technique saves x and y coordinates (as well as pressure and tilt) that are subsequently utilized to define a writer in terms of their spatial-temporal information. Using the offline approach, a scanned picture of handwritten data is used to identify who authored it, based on a variety of allograph traits. In this study, the offline author identification is taken into consideration that does not rely on text. Offline author identification may be divided down into the following classes, according to literature [11]. Approaches based on texture, form, and deep learning.

It may be possible to determine who wrote a piece of information using deep learning. Caffe Net, which comprises three fully-connected layers and five convolutional layers, has been used in the past to extract deep features. A two-stream neural network with shared weights may be used to detect who wrote what when images of English lines and single Chinese characters are compared [12]. The cluster indices of the clustered SIFT descriptors collected from 3232 picture patches are used to train a neural network. Using a VLAD trained on a triplet network to detect the degree of similarity between image patches, the global feature vector of each document is encoded. Complex attributes are extracted using the conditional Auto Encoder. Use of the VLAD feature-learning algorithm for offline identification of authors based on word images is demonstrated and encoded. Deep features are extracted from character images using convolutional neural networks, and these features are then integrated with other areas and deep features to build global characteristics that can identify the author. It is possible to identify English and Arabic writers by analyzing a large number of representations of image patches using the Image Net architecture. There are many ways to construct a feature vector that may be used to compare the handwriting styles of different people.

Vineet Kumar and Suresh Sundaram [13] (2022) came up with a revolutionary way to analyze handwritten word images in order to determine the author of a text. The proposed method does not depend on text, and the size of word graphics is not limited in any way by the proposed technology. The SIFT approach is used at the beginning to extract many

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 55 Issue: 12:2022
DOI10.17605/OSF.IO/KADQV

key locations with varying levels of detail (comprising allograph, character or combination of characters). Then, a CNN network trained to build feature maps matching a convolution layer gets these crucial points. Because SIFT key points exist at many scales, it is feasible to create feature maps of various sizes. This problem is solved by using the histogram of gradients to represent the feature map. There are more and more filters per convolution block with increasing network depth. Meaning that getting histogram features from each feature map generated by a convolutional algorithm adds both time and complexity. The weights of a specific CNN layer's feature maps are learned using an entropy-based technique during training. To test the practicality of the system architecture, publically available datasets CVL and IAM [13] are been utilized.

Handwritten word graphics and neural networks may be used to identify the author in a 2021 research by Sheng He and Lambert Schomaker [14]. The system is built using a combination of fragment-based attributes and global context. A global average pooling step is used to extract information from the neural network's tail about the global context. To determine the sequence of local and fragment-based features, a low-level deep feature map is used. This map's handwritten style is clearly visible. Researchers may be able to get a better understanding of the spatial connections between various components by using an RNN. The concept behind the global context residual recurrent neural network (GR-RNN) technique is that local fragments and global context provide complementary information. Four public data sets were used to test the proposed technique, and the results show that it is capable of cutting-edge performance. Gray-scale images are more successful in training neural networks than binary or contour images. Texture information is essential to identifying the author [14].

H T Nguyen et al. [15] (2020) have developed a deep-learning approach for the identification of authors without the need of specifying attributes ahead of time. Each character and its subregions are authored by a distinct author when a CNN is taught to extract local attributes. In order to train the CNN, photos from the training set must be selected at random. The global features are created by concatenating the local features that were collected from each pair of photos. Tuples are chosen at random and the process is repeated for each training session. It is akin to teaching the CNN to identify the author even when there is no text to analyze to create a huge number of training patterns. Utilizing the JEITA-HP library of offline patterns, an attempt was made to write Japanese characters by hand using the characters. With 200 characters, appropriately 100 writers with the accuracy of 99.97% were identified. Even though few characters were used; the proposed technique was still 92.8 percent or 93.8 percent accurate for 100 writers or 400 writers. Fire maker and IAM databases were used to do more tests on offline handwritten English text in the database. Using just one page per author to train on, this technique correctly classified over 900 writers at a rate of 91.81 percent. At the end, the proposed method exceeded the previous best result, which was based on hand crafted features and computational techniques of clustering. That the proposed technique works even with handwritten English text [15] is evidence of its effectiveness.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 55 Issue: 12:2022
DOI10.17605/OSF.IO/KADQV

Comparison of triangulation embedding and VLAD encoding was made by Christlein and Andreas in 2017. Decor relation and Exemplar SVMs, as well as generalized max pooling, are examined as alternatives to sum pooling. ICDAR13 and KHATT datasets were utilized to construct new criteria based on these methodologies [16].Yang et al. proposed online text-independent author identification [17] (2016). Through the removal of one or more strokes from a single original online character pattern, this technique generates a wide range of synthetic patterns. CNNs determine the likelihood of each character using the false patterns that the original pattern created. The last step is to do a statistical analysis to figure out who wrote the article. This technique is quite accurate, but averages can only be trusted to the extent that character patterns in the papers themselves are reliable. Drop Segment technique is tough to use with handwritten text [17].Christlein et al [18] used a different method (2017). Using the local features of CNNs and the super-vector encoding of Gaussian Mixture Models (GMM), they created global features. This approach performed better than Fiel's and Sablatnig’s [19] (2015). For the record, CNNs beat traditional local descriptors like SIFT.

Fiel and Sablatnig [19] (2015) used CNNs to learn about the unique properties of the area they were studying. Since the layer just above it provides enough information to identify the author, their technique eliminates the need for the final, fully connected layer. An image's mean vector may be computed by summing up all of its local feature vectors, and this can then be used to identify the author. The database and the preprocessing procedure are critical to the success of this strategy.

This paper presents a Deep learning technique for offline author identification combining CNN and RNN. Section 2 discusses about proposed method combining CNN and RNN. Section 3 discusses about Results and Discussion. Section 4 discusses about conclusion remarks.

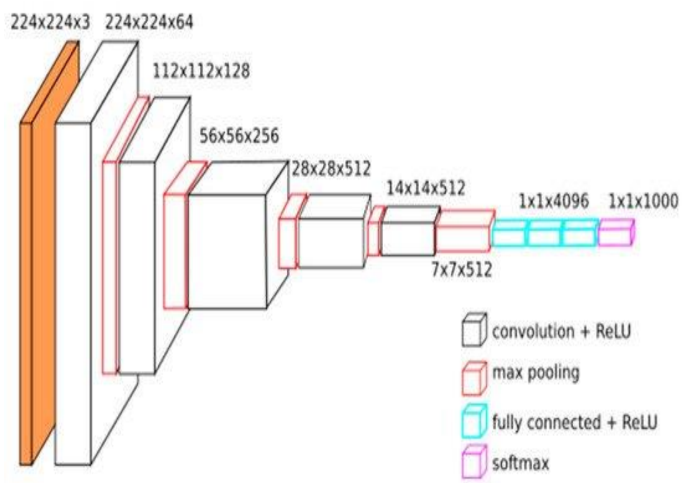
2. PROPOSED METHOD

2.1 Writer Identification Using Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) was employed in this experiment. It contains a maximum pooling layer, 16 hidden convolutional layers, and layers that are completely linked. Figure1 shows a structure of a Convolutional Neural Network. ReLu is used in both the convolution and fully connected layers, whereas SoftMax initiation capacity is utilised in the yield layer to determine the probability of each class. This research uses Tensor Flow-supported Keras for deep learning, and scikit-learn for order and evaluation. As a result, less standard codes can be written. In addition, this aids us in doing research more quickly.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 55 Issue: 12:2022
DOI10.17605/OSF.IO/KADQV

Figure 1: The structure of a Convolutional Neural Network (CNN)



Convolutional levels between pooling and normalisation are common in the CNN network. The convolution between inputs and filter groups is computed at every stage of convolution. "Convolutional," "Batch Normalization," "Rectifier Linear Units," "Pooling," and "Completely Linked" are all terms used to describe CNN's setup. There are five convolutional layers and three fully connected layers in Alex Net, which is a CNN that's previously been trained. There are just four convolutional layers and one fully linked layer in this network. Between the convolutional layers were stages of batch regularisation, ReLU, and pooling.

2.1.1. Convolutional Layer

This level was used to mix together the results of descending filters both horizontally and vertically. For every filter, the result of the weights and inputs is calculated, and later, the term "bias" was added. Here, 4 convolutional levels were used, each with its own set of hyper parameters. The equations for calculating the output dimension of each convolutional level and the number of weights per filter are shown in Eq. (1) and Eq. (2).

$$Outputsize = \frac{W_1 - F + 2P}{S} + 1 \tag{1}$$

$$Weight = FXFX3 \tag{2}$$

2.1.2. Batch Normalization Layer

The mini-group is used to process all of the input channels in this level, which then equalises them all by subtracting the mini-group average and dividing by the mini-group normal aberration. In the following stage, level shifts the input by a learnable offset, and

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 55 Issue: 12:2022
DOI10.17605/OSF.IO/KADQV

it calculates the magnitude using a learnable scaling factor. As a result, CNN operates more quickly and is less susceptible to network initialization [21].

$$X = \frac{X_1 - \mu_E}{\sqrt{\sigma} + e} y_i = yx + \beta \tag{3}$$

2.1.3. Rectifier Linear Units (ReLU)

With the basic input running through the threshold process, negative numbers are changed to zero and positive numbers are retained. This is also referred to as a neuron's activation function, and this layer increases the Non-linear properties of the model [23].

$$f(x) = (0, x) \tag{4}$$

2.1.4. Pooling layer

This level works like downsampling, which divides the input into rectangular sections for pooling and calculates the values of each section. This also makes the output look better and stops local changes. There are three different ways to pool resources: maximum, minimum, and normal pooling. Here, the output of the convolutional level is split up using the supreme pooling level. The Supreme pooling level figures out the highest value in each section. Eq (5) gives the formula for figuring out the output dimension of the pooling level.

$$Output\ size = \frac{W_1 - P_1 + 2P}{S} + 1 \tag{5}$$

2.1.5. Fully Connected Layer

While CNN final phase uses the classifier level, this level uses skilfull weighted linkages to catalogue previously mined features. The bias vector and the weight matrix are both increased as a result of this. An N-dimensional vector is created from the output of the level above, which determines how many divisions are used for cataloguing.

2.2. Writer Identification Using Recurrent Neural Network (RNN)

An artificial neural network may be used in many different ways. Artificial neural networks have evolved throughout time to specialize in diverse sorts of data analysis. Recurrent neural networks (RNNs) have been designed for the processing of sequence data, while CNNs have been developed for the processing of matrix-type data, such as pictures [20], [22]. Traditional feed-forward neural networks (FFNNs) only accept input from samples that they have already been exposed to. RNNs, on the other hand, use data that they accumulate through time as well as previously used samples as input. [xx1, xx2... xkk] is the input sequence. The length of the sequences may affect the value of k. The RNN model creates a hidden state [h1, h2.....hkk] for each step.

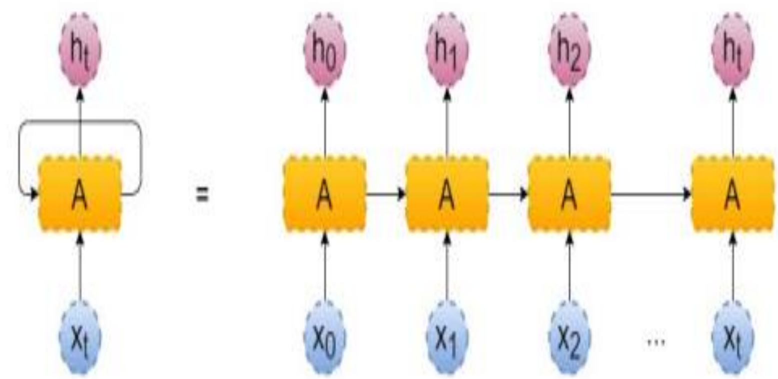
Where, htt−1 is the previous hidden state and $xxtt$ is the current input, the activation of the hidden state at time t can be written as:

$$\text{h}tt = (xxtt, \text{h}tt-1) \tag{6}$$

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 55 Issue: 12:2022
DOI10.17605/OSF.IO/KADQV

RNNs feature a repeating layer, unlike standard FFNNs. It is via this layer that the FFNN-generated state information is saved and re-applies to the network. As a result, RNNs are capable of remembering what they've already learned [23]. The RNN may be seen in Figure 2, which shows the RNN representation.

Figure 2: Recurrent neural network representation



A RNN [24] is a natural extension of feed-forward neural networks to sequences is the RNN [27]. The RNN can be used to encode a variable-length sequence into a fixed-length vector representation given an input sequence of [x1, x2,..., xk] (different samples may have different values for k). A hidden state is created at every time step, resulting in a hidden sequence of [h1, h2... hk]. As a function of the current input xt and the preceding hidden state ht1, the activation of the hidden state at time step t is calculated.

$$h_t = f(x_t, h_t - 1) \quad (7)$$

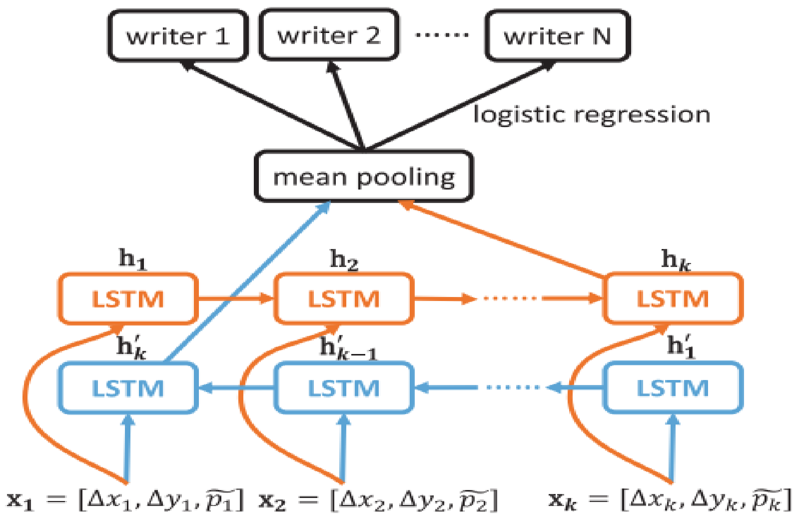
The output sequence [y1, y2,..., yk] produced at each time step by the formula yt = g(ht) can be used for sequence-to-sequence tasks (like speech and handwriting recognition) [25]. In this study, sequence classification is taken into account; therefore only one output is generated based on the final hidden state, hk. In other words, because the transition function f is applied recursively, the input sequence is encoded into a fixed-length vector hk. Figure 3, which show the RNN model for writer identification.

2.2.1 Long Short-Term Memory

The RNN is extraordinarily deep because it keeps track of activations at every step in time. To ensure the success of RNN, the recurrent unitf is critical. Long-term reliance may be learned using LSTM [26] and it is immune to the vanishing gradient issue.An input gate i_t, a forget gate f_t, and an output gate o_t are present in an LSTM for time step t.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 55 Issue: 12:2022
DOI10.17605/OSF.IO/KADQV

Figure 3. RNN model for writer identification



$$c_t = i_t \odot \check{c}_t + f_t \odot \quad (8)$$

$$h_t = O_t \odot tanh(c_t) \quad (9)$$

Where

$$i_t = sigm(W_i x_t + U_i h_{t-1} + b_i) \quad (10)$$

$$f_t = sigm(W_f x_t + U_f h_{t-1} + b_f) \quad (11)$$

$$O_t = sigm(W_o x_t + U_o h_{t-1} + b_o \quad (12)$$

$$c_t^{\vee} = tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (13)$$

Where b is the bias vector, W is the input-to-hidden weight matrix, U is the state-to-state recurrent weight matrix. The operation stands for the vector product, element by element. The concatenation of the LSTM's hidden state is (ct, ht). As indicated in Eq. (8) and Eq. (9), the long-term memory is stored in ct, and the updating of ct is controlled by the forget gate and input gate, while the updating of ht is controlled by the output gate as indicated in Eq. (10-13).

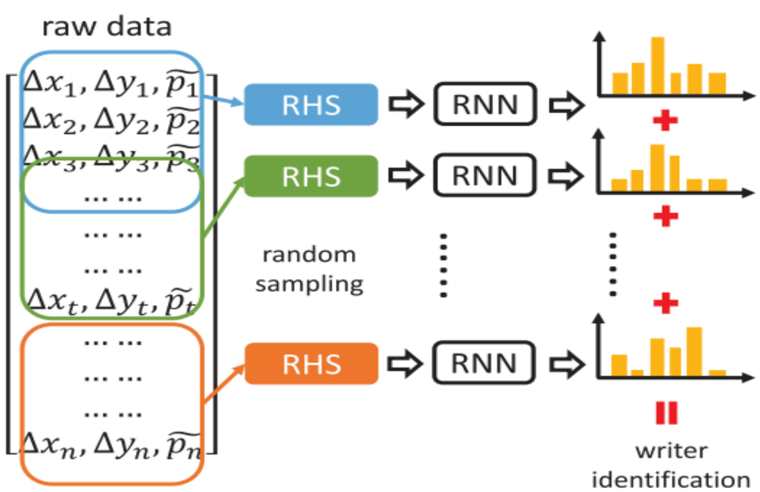
2.2.2 Writer Identification with RNN

In order to identify a writer, he or she must first sign up for the system and show some handwriting data. A big each writer may only have a little amount of training data to work with, making it difficult to identify them. For example, a single line of text or a single page may be allocated to each writer. The technique known as RHS is used to choose small subsequences at random to tackle this difficulty. To generate a large enough training dataset for RNNs, many RHSs can be chosen at random as necessary. According to Figure 4 a writer's raw data is randomly selected and the RNN model is taught to classify

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 55 Issue: 12:2022
DOI10.17605/OSF.IO/KADQV

each RHS on its own. This is followed by a final ensemble-based prediction, which yields a final writer identification result. In order to correctly identify the author, a single RHS is insufficient. The performance of a large number of randomly selected RHSs, on the other hand, may be swiftly and dramatically improved. True and imaginary strokes are used in the RHS, which is not a real "stroke" in handwriting. An RHS sequence is nothing more than a collection of three-dimensional vectors. For example, there is no need to know how to split apart words or letters in handwriting. The identity of the individual whose handwriting style can be found in RHS cannot be identified. Because of this, the RNN model is able to identify these characteristics by making smart judgments about whether or not a specific stroke in RHS will be remembered or forgotten.

Figure 4. Proposed end-to-end writer identification system



Randomly selected RHSs are found in the raw data. For each RHS, the RNN model is given a posterior probability histogram for each individual RHS (for different writers). In the end, the choice is based on the average of all the histograms. LSTMs are pre-tuned for the job of identifying the author by automatically focusing on the RHS for distinctive qualities. The approach proposed, as seen in Figure4, is based on raw data and can identify the author from beginning to end. The methodology is more broad and effective than earlier feature-based methods that were created by hand [27]. Pre-processing and feature engineering may be done without the requirement for human intervention. The technique outperforms earlier methods because of end-to-end training, sufficient training data.

3. RESULTS AND DISCUSSION

The experimental results of offline writer identification using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) models are shown below:

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 55 Issue: 12:2022
DOI10.17605/OSF.IO/KADQV

Table 1: various parameters/variables used for writer identification

Parameters /Variables	Values
Data Set	IAM Data Set
Number of writers/Classes	680
Number of documents per writers	25
Each documents is divided into Number of image documents per writers	8 sub image 25 x8=200 image/writers
Total Data Set	680 x200=136000 Samples
% of Training and Testing from total Data set	70% -Training ,30%-Testing
Number of Samples in Training Data set	680 x140x5x4 =1904000 Samples
Number of Samples in Testing Data set	680 x60x5x4 =816000 samples
Different Orientation	0° , 10° , 20°, 30° ,40°
Scale	Zoom in=.5,.75 Zoom out=1.5,2
Max No.of epoch	70
Num of layers	16 layers
Convolution filters size	3x3
Pooling	Max polling
Pooling filters size	2x2
Classifier	Soft max

The above table 1 explainsvarious parameters/variables used for writer identification using CNN and RNN used for conducting this experiment.

3.1 Writer identification using CNN

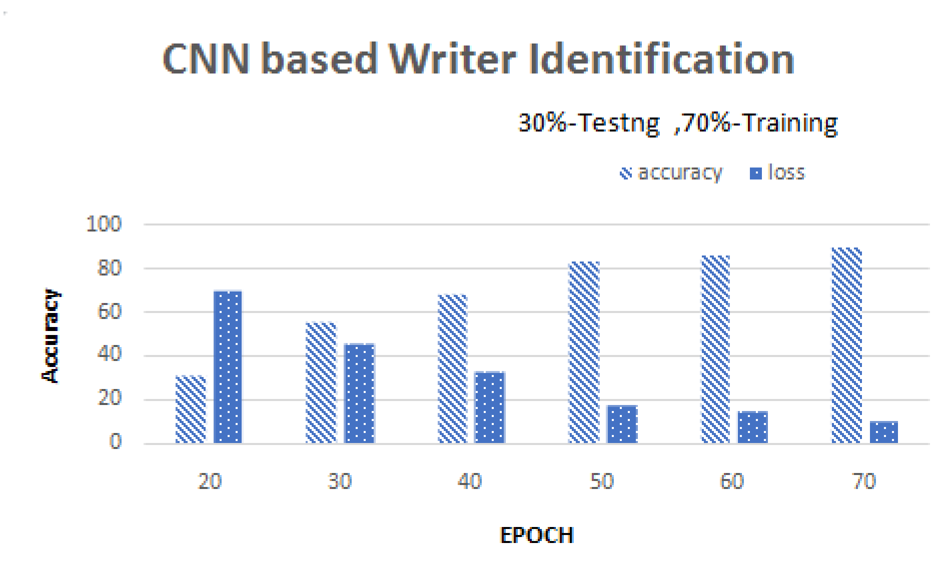
Table 2 depicts, Experiments conducted on IAM dataset with 680 writers and each writer writes 25 documents and each document is divided into 8 sub image. Number of image document per writer is 200 image documents per writers. Each document is rotated in 5 different orientation and 4 scale factor.70% of the 200 image document per writer are used for training and 30% of the 200 image document per writer are used for testing. The average accuracy and standard deviation is computed scale wise, orientation wise and overall combination. From this, the result is analysed for three inferences i.e., the system is Sensitive to Scale, Sensitive to Orientation and Sensitive to Scale and Orientation. From Table2, it is observed that as Standard Deviation is small, the proposed method is robust, invariant to scale, orientation and both. For every training Epoch increases accuracy will also increase. Figure 5 depicts Accuracy Graph of CNN based Writer Identification.For the value of 70 in Epoch, 90% accuracy is achieved for writer identification using CNN.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 55 Issue: 12:2022
DOI10.17605/OSF.IO/KADQV

Table 2: Average Accuracy of all Writers with different Orientation and Different Scale using CNN

Orientation → Scale ↓	0°	10°	20°	30°	40°	Average accuracy in %	Standard Deviation
Zoom in .5	90.1	90.1	90	90.2	90.1	90.1	0.070710678
Zoom in .75	90.1	89.9	89.8	89.9	89.9	89.92	0.109544512
Zoom out 1.5	89.9	89.88	89.9	89.9	89.9	89.896	0.008944272
Zoom out 2	89.9	89.99	89.8	89.99	89.8	89.896	0.095026312
Average	90	89.9675	89.875	89.9975	89.925		
Standard Deviation	0.1154	0.10045	0.082916	0.1415097	0.12583057		

Figure 5: Accuracy Graph of CNN based Writer Identification



3.2 Writer identification using RNN

Table 3depicts, Experiments conducted on IAM dataset with 680 writers and each writer writes 25 documents and each document is divided into 8-sub image. Number of image document per writer is 200 image documents per writers. Each document is rotated in 5 different orientation and 4 scale factor.70% of the 200 image document per writer are used for training and 30% of the 200 image document per writer are used for testing. The average accuracy and standard deviation is computed scale wise, orientation wise and overall .From these result analysis we done 3 inferences that the proposed system is Sensitive to Scale, Sensitive to Orientation and Sensitive to Scale and orientation. From this Table 3 we observed that Standard Deviation is small so we can say that the algorithm is robust invariant to scale, orientation and both. For every training, as Epoch increases,

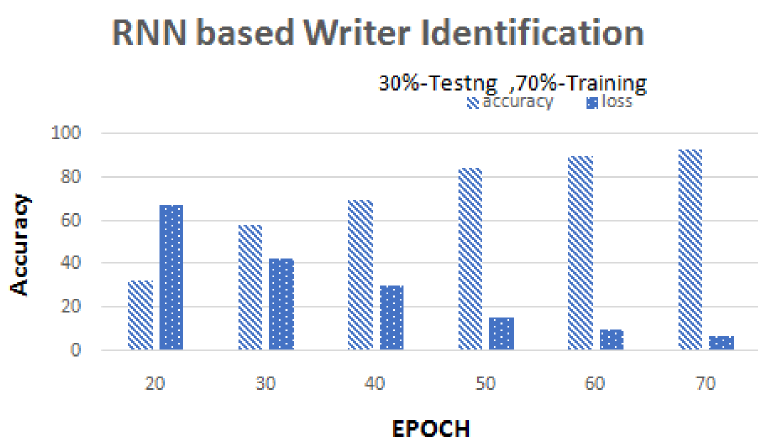
Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online):0493-2137
E-Publication: Online Open Access
Vol: 55 Issue: 12:2022
DOI10.17605/OSF.IO/KADQV

accuracy also increases. Figure 6 depicts Accuracy Graph of RNN based Writer Identification. For the value of 70 in Epoch, we are getting 90% of accuracy for writer identification Using RNN.

Table 3: Average Accuracy of all Writers with different Orientation and Different Scale using RNN

Orientation → Scale ↓	0°	10°	20°	30°	40°	Average accuracy in %	Standard Deviation
Zoom in .5	93	93.1	93	93.1	93	93.04	0.054772256
Zoom in .75	93.1	93	93.1	93.2	93.1	93.1	0.070710678
Zoom out 1.5	92.9	92.9	92.8	92.9	92.9	92.88	0.04472136
Zoom out 2	92.9	92.8	92.9	92.9	92.8	92.86	0.054772256
Average	92.975	92.95	92.95	93.025	92.95		
Standard Deviation	0.095743	0.129099	0.111803	0.15	0.12909944		

Figure 6. Accuracy Graph of RNN based Writer Identification



4. CONCLUSION

Offline writer identification is presented using CNN and RNN. The experiments were conducted on IAM data set with 680 writers. Each character pattern's sub-regions and the whole character patterns were used in the CNN-based model to derive local characteristics. To create global characteristics, they were merged in three ways. A modest number of samples were utilized to create a large number of training patterns throughout CNN's training process. The RNN approach was then proposed as an end-to-end framework for author identification by directly dealing with offline handwriting data. Classifying a random set of hybrid strokes was done using an RNN model with LSTM's bidirectional, from CNN 90% of accuracy is achieved for writer identification and using RNN 93% of accuracy is achieved. The proposed method can generate efficient and robust for writer identification based on different scale, different orientation and both of given handwritten documents. In future, the same experiment can be conducted for different languages with multiple scripts.

ACKNOWLEDGMENT

I would like to express my deep gratitude to Dr. Gopal A Bidkar and Dr. Jagadeesh D Pujari, my research supervisors, for their patient guidance, enthusiastic encouragement and useful critiques of this research work.

I would also like to thank Dr. P.SHiremath sir, for his advice and directionwhich helped me to keep my research progress on schedule.Finally, I wish to thank my family and my colleagues for their support and encouragement throughout my study.

REFERENCES

1. R. Morris, R. N. Morris, “Forensic Handwriting Identification: Fundamental Concepts and Principles,” Academic Press London, 2nd ed. UK, 2020.
2. Marcelli, A. Parziale and C. De Stefano, “Quantitative evaluation of features for forensic hand writing examination”, *In Proceedings of the 13th International Conference on Document Analysis and Recognition(ICDAR),* Tunis, Tunisia, Aug.2015, pp. 1266–1271,doi:10.1109/ICDAR.2015.7333952.
3. M. Bulacu and L. Schomaker, “Text-independent writer identification and verification using textural and allographic features,” *IEEE Trans. Pattern Analysis machine intelligence,* vol.29, no.4, pp.701–717, Apr.2017, doi: 10.1109/TPAMI.2007.1009.
4. R. K. Hanusiak, L. S. Oliveira, E. Justino, and R. Sabourin, “Writer verification using texture-based features,” *International Journal on Document Analysis and Recognition. (IJDAR)*, vol. 15, pp. 213–226, May.2011, doi:10.1007/s10032-011-0166-4
5. J. L. Vasquez, A. G. Ravelo-García, J. B. Alonso, M. K. Dutta and C. M. Travieso, “Writer identification approachby holistic graphometric features using off-line handwritten words,” *Neural Computing and Application.,* vol. 32, pp. 1–14, Oct.2020, doi:10.1007/s00521-018-3461-x.
6. J. Chu, M. A. Shaikh, M. Chauhan, L. Meng, S. Srihari, “Writer Verification using CNN Feature Extraction,”*In Proceedings of the 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*,Niagara Falls, NY, USA, Aug. 2018; pp. 181–186,doi:10.1109/ICFHR-2018.2018.00040.
7. S. He, and L. Schomaker, “Deep adaptive learning for writer identification based on single hand written word images,” *Pattern Recognition*, vol. 88, pp. 64–74, Apr.2019, doi:10.1016/j.patcog.2018.11.003.
8. D.Impedovo, G.Pirlo and R.Plamondon,“Handwritten signature verification: New advancements and openissues,” *In Proceedings of the International Conference on Frontiers in Handwriting Recognition, Bari,Italy,* Sep.2012; pp. 367–372,doi: 10.1109/ICFHR.2012.211
9. L. G. Hafemann, R. Sabourin, and L. S. “Oliveira, Offline handwritten signature verification—Literature review,” *In Proceedings of the 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA),* Montreal, QC, Canada, 28 November–1 December 2017; pp. 1–8. Doi:10.1109/IPTA.2017.8310112.
10. Sharma and S. Sundaram, “A novel online signature verification system based on GMM features in a DTW framework,” *IEEE Transactions on Information Forensics and Security,* vol.12, no.3, pp.705–718, Nov.2016. Doi:10.1109/TIFS.2016.2632063.
11. Y. J. Xiong, Y. Lu, and P. S. Wang, “Off-line text-independent writer recognition: A survey,” *International Journal of Pattern Recognition and Artificial Intelligence,* vol. 31, no. 05, pp. 1756008, Jan.2017, doi:10.1142/S0218001417560080.

12. L. Xing, Y. Qiao, “Deep writer: A multi-stream deep CNN for text independent writer identification,” *International Conference on Frontiers in Handwriting Recognition (ICFHR),* pp.584-589, Aug.2016, doi:https://arxiv.org/pdf/1606.06472.

13. V. Kumar and S. Sundaram, “Offline Text-Independent Writer Identification based on word level data,” *Computer Vision and Pattern Recognition,* July.2022, doi:https://arxiv.org/pdf/2202.10207.

14. S. He and L. Schomaker, “Global-Context Residual Recurrent Neural Networks for Writer Identification,” *Computer Vision and Pattern Recognition,* April.2021, doi:https://arxiv.org/abs/2104.05036.

15. H. T. Nguyen, C. T. Nguyen, T. Ino, and M. Nakagawa, “Text-independent writer identification using convolutional neural network,” *Computer Vision and Pattern Recognition Pattern Recognition Letters,* vol. 121, pp. 104-112, Sep.2020, doi:https://arxiv.org/abs/2009.04877.

16. Christlein and Andreas, “Encoding CNN Activations for Writer Recognition,” *Computer Vision and Pattern Recognition,* Jun.2018, doi: 10.1109/DAS.2018.9.

17. W. Yang, L. Jin, M. Liu, “Deep WriterID: An End-to-End Online Text-Independent Writer Identification System,” *IEEE Intelligent Systems,* vol. 31, no. 2, pp. 45–53, Dec.2016, doi:https://arxiv.org/abs/1508.04945.

18. V. Christlein, D. Bernecker, A. Maier, and E. Angelopoulou, “Offline writer identification using convolutional neural network activation features,” *Pattern Recognition,* vol. 9358, pp. 540-552, Dec.2015, doi:10.1007/978-3-319-24947-645.

19. S. Fiel, R. Sablatnig, “Writer identication and retrieval using a convolutional neural network,” *International Conference on Computer Analysisof Images and Patterns,* pp. 26-37, 2015,doi:10.1007/978-3-319-23117-4_3

20. G. Li, M. Liu, Q. Sun, D. Shen, and L. Wang, “Early diagnosis of autism disease by multi-channel CNNs,” *Machine. Learning Medical Imaging,* vol. 11046, pp. 303–309, Sep.2018, doi: 10.1007/978-3-030-00919-935.

21. S. Ioffe, and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *Computer Science and Machine Learning,* Mar.2015, doi:https://arxiv.org/abs/1502.03167.

22. A.T. Kabakuş, "A Comparison of the State-of-the-Art Deep Learning Platforms: An Experimental Study," *Sakarya University Journal of Computer and Information Sciences,* vol.3, no.3, pp.169-182, Dec.2020, doi:10.35377/saucis.03.03.776573.

23. Z. Özer, “The Effect of Normalization on the Classification of Traffic Comments,” Ph. D. Thesis, KarabükUnv. Grad. Sch. of Nat. and App. Sci, Dept. of Computer Engineering, Karabük, Turkey, 2019.

24. I.Sutskever, O. Vinyals, and Q. Le, “Sequence to sequence learning with neural networks,” *In ProcComputational and Language.Conf.,* 2014,doi:10.48550/arXiv.1409.3215.

25. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, “Connection is ttemporal classification: Labelling Un segmented sequence data with recurrent neural networks,” *Proc. International Conference on Machine Learning,* pp. 369–376, 2006, doi:10.1145/1143844.1143891.

26. F. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with LSTM,” *Neural Comput.,* vol. 12, no. 10, pp. 2451–2471, Oct.2000, doi: 10.1049/cp: 19991218.

27. N. S. Bae and N. Memon, “Online signature verification on mobile devices,” *IEEE Transaction on Information Forensics and Security,* vol. 9, no.6, pp. 933–947, Jun.2014, doi:10.1109/TIFS.2014.2316472.