May – June 2020
ISSN: 0193-4120 Page No. 7651 - 7658

Feature Extraction with MLP and CNN in Writer Identification

Vinita Patil
Research Scholar, Dept. of ECE,
GSSS Institute of Engineering and Technology, Mysuru
Email:vinita278@yahoo.co.in

Rajendra R Patil
Professor and Head Dept. of ECE,
GSSS Institute of Engineering and Technology, Mysuru
Email:rajendra@gmail.com

Article Info
Volume 83
Page Number:7651 - 7658
Publication Issue:
May-June 2020

Article History
Article Received: *19 November 2019*
Revised: *27 January 2020*
Accepted: *24 February 2020*
Publication: *18 May 2020*

***Abstract*:**
In this work, features from the handwritten documents are extracted using two methods, namely, multilayer perceptrons and convolutional neural networks. Features extracted from these two models were used to define the states of a hidden markov model. Performance of the models were tested on two datasets, namely, VTU-WRITER and IAM datasets. The VTU-WRITER dataset is a custom created dataset by collecting the handwritten documents exclusively for this research work. The performance of the two models namely, MLPHMM and CNNHMM are compared with the hidden markov chain model that has singular values as the features. Baum-Welch algorithm is used to determine parameters of HMM models.

Index Terms—Hidden markov models, Convolutional neural network, Writer identification, Multi layer perceptron etc.

I. INTRODUCTION

Authentication is one of the important factors in order to grant access to a facility or service to certain people. The authentication is important in the sense that only few people are allowed to access the facility or service and all other should be prevented from getting the access. To grant the authentication for access verification is important. Verification can be either manual or automatic. Manual verification includes documents verification or confirmation by any other individual like comparing the photographs or signatures. Automatic verification includes processing the information by an automatic system or AI (Artificial Intelligence) system. Examples of automatic verification include passwords, electronic access with ID cards or PIN. Data that is collected from a person is processed by an electronic system. However, these access credentials can be forgotten by a person or can be stolen by others. Hence it is imperative to make the system fail proof so that only authorized personal are allowed the access to a facility, service or application. Biometrics is one such kind of data that is specific to a person, but cannot be stolen or forged by others that easily.

The other part of the writer identification system to accurately predict the writer is type of classifiers used. The type of classifier is very important to convert the features into the probabilities that the text belongs to a writer. A writer with highest probability score is the most probable writer. This is known as top-1 accuracy. If we consider two top probabilities so that the writer could be one of the two, then it is known as top-2 accuracy. In a similar way, one can predict 10 writers with top 10 probabilities so that writer could be one among the 10 writers. The accuracy of top-10 will be higher and top-1 will always be lower than top-2, top-5 and top-10.

Hidden Markov Models (HMM) are modeled on the basis of creating multiple states in the input. In the present case, each line of text is an image and hence image is split into various blocks. Each block can be converted into features. Based on the probabilities of change of states of the model, the identification is performed. These probabilities can be optimized for a set of inputs and constraints using Baum-Welch or Viterbi algorithm. There are various ways to define the features in the HMM.

Pixel intensities of each block can be used to define each state of HMM. Since the text image is divided into blocks, each block is represented with pixel intensities of image of that block. If the block size is 60 rows and 200 columns, then there would be 12,000 pixels. If the image has 3 color channels then the number of pixel values would be 36,000 pixel values for each block. But for the case of writer identification, color image can be converted to gray image so that the number of pixels in that block remains 12,000.

May – June 2020
ISSN: 0193-4120 Page No. 7651 - 7658

1. Each pixel is not a true representative of the state or features.
2. When image is stretched, rotated or scaled, pixel intensities change.
3. Size of observation vectors is huge like 12,000 in the above example.

For each block that represents a state, singular values can be determined. Singular values can be derived with the method of SVD. Number of singular values that can be derived from SVD would be equal to number of columns. That means if the size of the block is 60 rows x 200 columns, then 200 singular values can be extracted which may be denoted by S.

In the development of ANN, foremost development to be given the due importance is the introduction of the concept of a perceptron by Rosenblatt in 1957. Perceptron is the simplest form of feed forward neural network. When the perceptron was introduced, it had only one layer of neurons and the inputs were fed to the output directly via the weighted connections. Initially the perceptron's gained popularity, but eventually, it waa not suitable for multi class problems. This was one of the drawbacks of single layer perceptron (SLP) models and to overcome this, multilayer perceptron (MLP) became alternative model in 1960s and it has become one of the widely used prediction models in neural network topologies. Machine learning (ML) and Deep learning (DL) are two branches of data science. These methods are useful in the design of algorithms that learn patterns from the data. Though ML and DL are two different branches, deep learning is a sub field of machine learning. The major difference lies in the definition of feature extraction. If the features are extracted manually and then classified by a discriminator, then it is treated as Machine learning. On the contrary, if the features are extracted by the algorithm itself, then it is known as Deep learning. Deep learning majorly developed in the field of ANNs. It can also be treated that deep learning is also a sub field of ANNs which are inspired by neuron behavior in biological neural networks.

While there are many deep learning methods, convolutional neural networks (Conv Net or CNN) are a kind of feed forward network. CNNs are deep networks like MLPs. As discussed before, feed forward networks can also be treated as MLPs.

The online and offline handwritten recognition methods are explained in [1-4]. The online handwritten recognition focuses mainly on the dynamic characteristics like pressure applied, speed of writing, direction of stroke, pauses taken etc [5-7]. Since online recognition considers these dynamic characteristics of writing, it is treated as more robust than the offline methods [8-9]. There are many other important methods used in the handwriting recognition or writer identification [10-14]. Deep learning methods like CNN (Convolutional neural network) [15] are becoming popular and has been on the raise in the application of document analysis [16-18] and wirter identification [19-22].

II. MLPHMM AND CNNHMM ALGORITHM

Features are extracted from the text images using MLP and CNN and used these features for classification of text images to associate each image with most probable writer index using the HMM models. In the previous work [ ], feature extraction was performed with HMM based SVD values and used Baum-Welch algorithm for classification. In this work, feature extraction is performed with two ANN algorithms. They are:

- Multi Layer Perceptron (MLP)
- Convolutional Neural Network (CNN)

The text image is input to the MLP and CNN in two separate models and features are extracted. The extracted features are in the form of one dimensional vector and two dimensional matrix in MLP and CNN respectively.

The micro features can be extracted in two ways:
1. Features extracted as one dimensional vector - MLP
2. Features input as two dimensional matrix - CNN

Features can be in the form of pixel intensities, KLT coefficients, DCT coefficients, PCA components or SVD values or using the ANNs. In this work, features are extracted using MLP and CNN models.

2.1 MLPHMM Model

As discussed above, MLPHMM model has two parts in it. The first part of layers being feature extraction layer with MLP and the second part of layers is classification layer with HMMBW. MLP is used as micro feature extraction layer and HMMBW is used as classification layer. Fig. 1 shows the micro feature extraction procedure for MLP. In the present model, the fully connected layers are formed by MLP. Every input is connected to every neuron in the hidden layer and also every neuron in one hidden layer is connected on other neurons in the next hidden layer.

It can be observed from Fig. 1 that each state (shown as non-overlapping states) is subjected to MLP and feature vector of size 128 is extracted for each state. Every state is now represented by an MLP feature vector. This one dimensional feature vector is input to HMMBW for classification. Instead of SVD values, MLP feature vectors are used as feature on HMMBW model. One dimensional feature vector of size 128 is labeled with the writer identification for all the states of that image. Like this, there may be several samples of text for each writer. Each sample of state of the text is converted to one dimensional feature vector using MLP and labeled with a writer index.

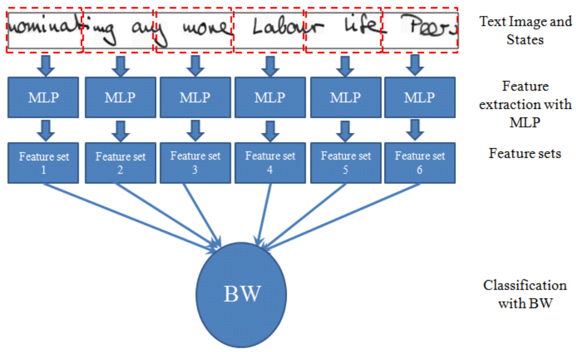

Fig. 1 Feature Extraction with MLP and classification with BW

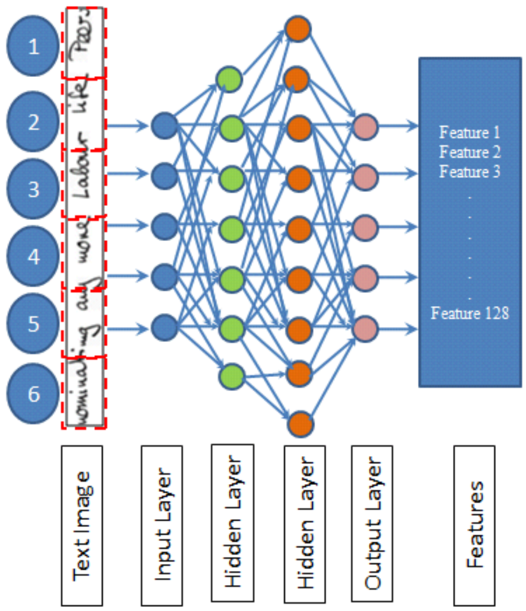

Fig. 2 Feature extraction with MLP

It can be observed from Fig. 2 that feature vector which is extracted from the text image is input to input layer of the MLP. Size of input layer is equal to the size of feature vector of the text image,. For example, if the size of the text image 50x1000 (gray image), then number of feature are 50,000. Therefore, size of input layer in the MLP is 50,000. In an MLP, there are hidden layers followed by the input layer. Number of hidden layers in MLP may be three or more. In the present research, three hidden layer are used in the MLP. Number of neurons in hidden layer is 2/3 times the input layer that is 33,333 neurons in the first hideen layer. Size of third hidden layer is set to be two thirds the size of adjacent input layer. Size of output layer is equal to number of writers in the dataset if the MLP is used as a classifier. Size of second hidden layer would be 22,222 and number of neurons in subsequent hidden layer would be 11,111. Hence the number of weights in the first hidden layer would become 50,000 x 33,333. That means there will be 1.67 x 10^9. Since there are other hidden layers in the subsequent layers, number of weights to be optimized will also be very high. In case of IAM dataset, number of neurons in output layer is set to 657 and that in VTU-WRITER dataset is set to 100 if the MLP is used as a classifier, but in this case, MLP is used only as feature extractor and hence the number of neurons in the output layer is set to 128.

2.2 CNNHMM Model

Another hybrid model CNNHMM can be used to classify the writers in a similar way the MLPHMM was used. In this model, the deep learning MLP model which was used for feature extractor was replaced with another deep learning model, namely, CNN. Similar to MLP, there are feature extraction layers and classification layers in CNN. But the classification layer of CNN was not used as a classifier, but used only as a feature extractor. In addition to automatic feature extraction layers of CNN, there is also another feature extractor, namely, HMM model which is input with the features extracted from the CNN. The states of HMM are defined with the 128 feature vector extracted from the CNN.
Advantage of using CNN in place of MLP as feature extractor is the huge number of weights in MLP which increases the computation time for optimization of weights enormously. Since the text image is in two dimensional form, it is converted to one dimensional vector in MLP and hence the size of input vector is very large which subsequently adds to the large number of weights. If the input text image is input in two dimensional form, then number of weights in the model can be drastically reduced. This is the approach followed in CNN and hence CNN is much faster than MLP and also CNN captures two dimensional features effectively.
State of the HMM are defined with the blocks created from the text image both in horizontal and vertical direction as shown in Fig. 3.

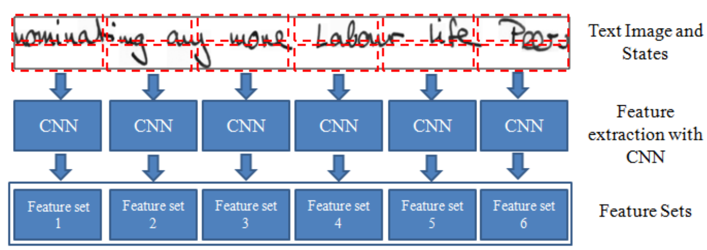

Fig. 3 Feature Extraction with CNN and classification with BW

In the example shown in Fig. 3, each state or overlapping/non-overlapping block is processed in CNN and is converted to a feature vector of size 128. As mentioned in the previous section, the DL models are used only as feature extractors here and not as classifiers. Therefore each block is converted to a feature vector of size 128.

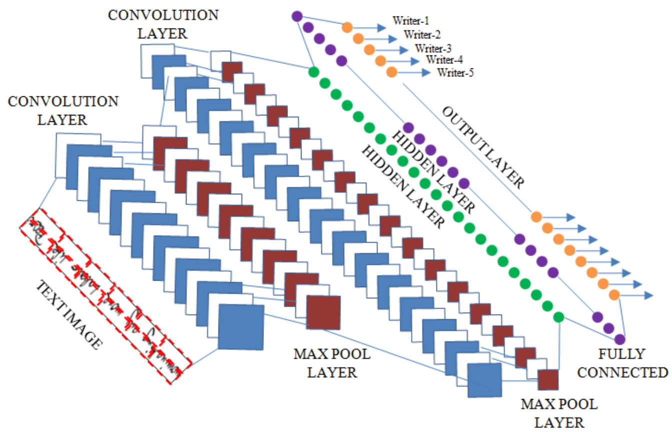

Fig. 4 Feature Extraction with CNN

CNN has the input of two dimensional features from the blocks extracted from the text image. Features are singular values of overlapping or non-overlapping blocks. Every hand written text is re-sized to 50 x 1000 pixels. The number of overlapping states in each of the image is set to 41 vertically and 951 horizontally. The size of each sub image is 10 x 50. Various layers involved in CNN are given below:

- **Input Layer:** Size of input layer is 50 x 1000 and has a set of input values of image.
- **Convolutional layer:** In convolution layer, a region is convoluted with a filter.
- **ReLU layer: O**utput for a region in convolution is presented to a ReLU layer which will apply the activation function on the input and it outputs a maximum of (0, x).
- **Down sampling layer**: Down sampling is performed to reduce the size of the output from ReLU layer.
- **Fully connected layer**: output from down sampling layer is input to the fully connected layer after flattening. It outputs 128 probabilities which may be considered as the feature vector for the text image of a writer.

2.3 MLPHMM Algorithm

Algorithm used in this research work is shown in Fig. 5 and is as follows:

1. Image is split into multiple blocks. The blocks can be non overlapping or overlapping.
2. A matrix is formed with the pixel intensities of each state.
3. For each state extract the features using MLP. MLP is provided with the softmax layer at output with 128 classes.
4. Foe every state extract the features using MLP.
5. Likewise for every image, extract the features for all the blocks.
6. Form a single column vector by concatenation of all features.
7. Create train set and test set.
8. Train HMM with Baum Welch algorithm.
9. Test the HMM with test set.

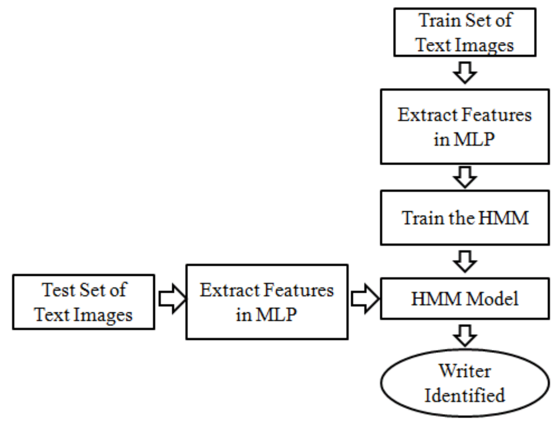

Fig. 5: Block diagram of proposed MLPHMM model

2.4 CNNHMM Algorithm

Algorithm used in this research work is shown in Fig. 6 and is as follows:

1. Image is split into multiple blocks. The blocks can be non overlapping or overlapping.
2. A matrix is formed with the pixel intensities of each state.
3. For each state extract the features using CNN. CNN is provided with the softmax layer at output with 128 classes.
4. Foe every state extract the features using CNN.
5. Likewise for every image, extract the features for all the blocks.
6. Form a single column vector by concatenation of all features.
7. Create train set and test set.
8. Train HMM with Baum Welch algorithm.
9. Test the HMM with test set.

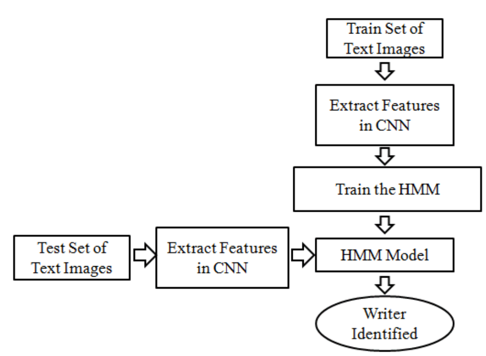

Fig. 6: Block diagram of proposed CNNHMM model

May – June 2020
ISSN: 0193-4120 Page No. 7651 - 7658

III. Simulation Results

Accuracy of MLPHMM and CNNHMM models for different combinations of train and test sets of custom made VTU-WRITER dataset is shown in Table 1. It can be noticed that MLPHMM and CNNHMM have produced results that are not comparable with HMMBW for improved performance.

Table 1: MLPHMM and CNNHMM model results

Number of Samples per writer	Total number of training samples	Total number of test samples	Ratio of test to train sets	Total Matches	Total Mismatches	HMMBW Accuracy	MLPHMM Accuracy	CNNHMM Accuracy
1	100	780	7.80	130	650	16.67%	15.34%	17.23%
2	200	680	3.40	280	400	41.18%	40.32%	43.21%
3	300	580	1.93	300	280	51.72%	52.00%	51.35%
4	400	480	1.20	300	180	62.50%	59.00%	64.78%
5	500	380	0.76	270	110	71.05%	68.00%	72.87%
6	600	280	0.47	270	10	96.43%	94.00%	95.58%

The MLPHMM has shown improved results when the number of samples was increased from one to six in the train set. It has yielded a best accuracy of 94% when the number of samples was increased to size. Similarly, the CNNHMM model also demonstrated an improved performance when the number of samples was increased. When the number of samples in the train set was one, the accuracy yielded by CNNHMM was 17.23% and it increased to 95.58% when the sample were increased to six.

It can be observed that though the CNNHMM model performed better than MLPHMM, accuracy was lower than that of the statistical model HMMBW. Both MLPHMM and CNNHMM did not perform well compared to that of HMMBW. This is due to the reason that features extracted from MLP and CNN are based on the probabilities of the writer indices instead of defining the features based on the parameters of the model like in an inception model. Hence this approach of defining the features in HMMBW with MLP and CNN features did not yield better results as expected. However before concluding on the effectiveness of this approach, more experiments are conducted to confirm if the approach of using MLP and CNN features in HMM indeed does not result in the improvement in accuracies.

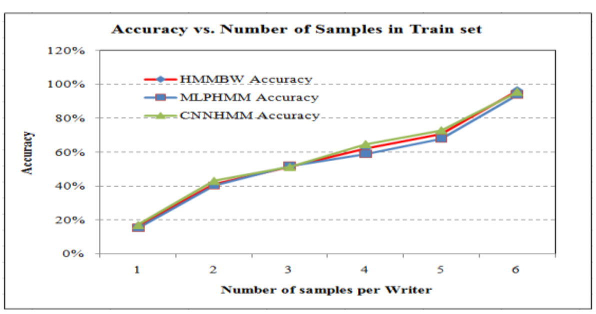

Fig. 7: MLPHMM and CNNHMM accuracy with respect to Sample Size

In next section, the number of states in the HMM model is varied to verify the improvement in the accuracy of MLPHMM and CNNHMM. Fig. 7 shows the improvement in the accuracy as number of samples is increased from to one to six per writer. It can be noticed that MLPHMM and CNNHMM have not yielded any better results than HMMBW. In some cases, it performed worse than the HMMBW.

Fig. 8 shows the relationship of accuracy with the test to train ratio. It can be observed from the plot that there is an inverse relationship between the accuracy and test to train ratio. As the test to train ratio increases, the accuracy falls. That means, the as the number of samples in the train set is reduced, the accuracy of model prediction reduces. All the three models, namely, HMMBW, MLPHMM and CNNHMM show the same trend. The accuracy falls below 20% when number of samples is only 100 in the train set.

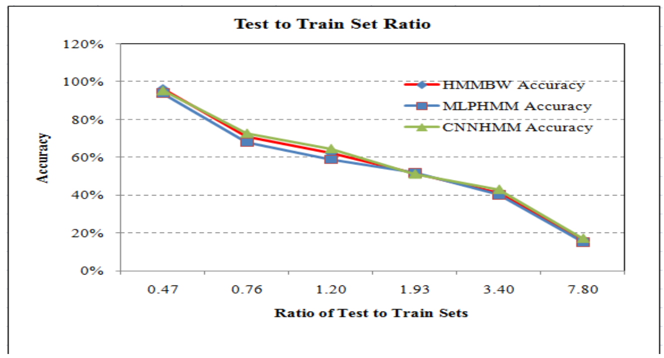

Fig. 8: MLPHMM and CNNHMM accuracy with respect to test to train ratio

Table 2 shows the accuracies achieved by HMMBW, MLPHMM and CNNHMM when the number of overlapping states is varied from 50% to 100%. In all the above models, the percentage of overlapping states is 50% of maximum possible states.

Table 2: MLPHMM and CNNHMM accuracy with respect to number of states

Number of states as percentage of maximum possible overlapping states	HMMBW Accuracy	MLPHMM Accuracy	CNNHMM Accuracy
50%	96.43%	94.00%	95.58%
60%	96.90%	94.85%	95.65%
70%	97.60%	96.35%	97.50%
80%	98.10%	97.95%	98.30%
90%	99.20%	98.50%	98.95%
91%	99.90%	99.10%	99.25%
92%	99.30%	99.20%	99.30%
93%	99.10%	99.20%	99.20%
94%	98.90%	98.65%	98.95%
95%	98.90%	98.60%	98.75%
100%	98.40%	98.30%	98.90%

May – June 2020
ISSN: 0193-4120 Page No. 7651 - 7658

It can be noticed from the Table 2 that the maximum accuracy is achieved with HMMBW when the percentage of overlapping states is 91%. At 91% of maximum possible states, the accuracy achieved is 99.9% for the VTU-WRITER dataset. The test to train ratio was set at 0.47. Similarly, the accuracy achieved by MLPHMM and CNNHMM are 99.20% and 99.30%. These accuracies are less than that of the HMMBW.

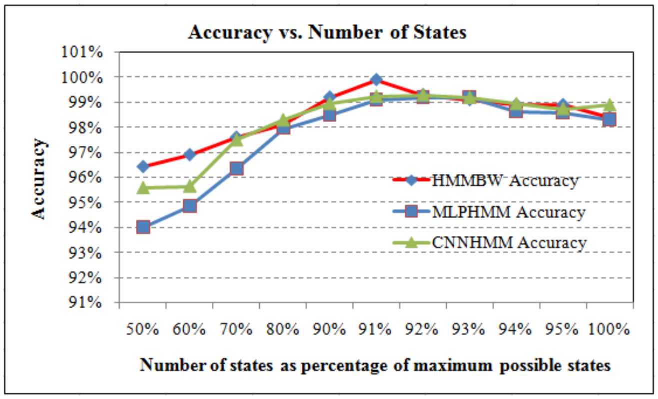

Fig. 9: MLPHMM and CNNHMM accuracy with respect to percentage of states

Both MLPHMM and CNNHMM yielded an accuracy of 99.20% and 99.30% at 92% of the states. However, the performance of MLPHMM and CNNHMM are not better than that of HMMBW, but it is almost similar to the HMMBW model. In this experiment it is confirmed that accuracies will improve as the number of states are increased and test to train ratio are decreased. It is required now to find another improvement over the current hybrid model to take the accuracies beyond HMMBW models. The reason behind higher accuracy of HMMBW model compared to that of the MLPHMM and CNNHMM is, the HMMBW model extracts the features based on the actual information content which is extracted as singular values from the data through a deterministic or mathematical approach. In case of MLPHMM and CNNHMM, the features are extracted using probabilistic approach. Hence the deterministic features with HMMBW has yielded better accuracy than the probabilistic features with MLPHMM and CNNHMM.

2.5 Comparison of HMMMLP and HMMCNN with other methods on IAM Dataset

Performance of MLPHMM and CNNHMM are compared with other methods for IAM dataset. The methods used by Bensefia et al [23-26], Marti et al. [27], Hertel and Bunke [28] and Xing and Qiao [29] are already compared in the previous work [ ] for IAM dataset with HMMBW. However, in this section also, these methods are compared with MLPHMM and CNNHMM as well. The models were run with:

- Test to train ratio of 0.47
- 91% of maximum possible states for MLPHMM
- 92% of maximum possible states for CNNHMM

Table 3 shows the comparison of other four methods with HMMBW, MLPHMM and CNNHMM.

Table 3: Accuracies of Various Models on IAM Dataset

Authors	Database	Features	Classifier	Accuracy	Language
Bensefia et al.	IAM	Graphemes extracted from cursive handwriting	Cosine similarity	95.00%	English
Marti et al.	IAM	Height of writing zones, Distance between connected components	kNN	92.00%	English
Hertel and Bunke	IAM	Height of writing zones, Distance between connected components	ANN	92.00%	English
Xing and Qiao	IAM	Raw pixels by CNN	CNN	98.80%	English
HMMBW	IAM	States based HMMBW	BW	99.80%	English
MLPHMM	IAM	States based MLP	BW	99.60%	English
CNNHMM	IAM	States based CNN	BW	99.75%	English

It can be noticed from Table 3 that MLPHMM and CNNHMM, which is marked in blue color and HMMBW with green color, have the better accuracies in writer identification compared to all other methods. MLPHMM and CNNHMM had performed equally well on IAM dataset, but the HMMBW has highest accuracy compared to MLPHMM and CNNHMM. MLPHMM was at 99.60% and CNNHMM was at 99.75%. There was lift of 0.15% in CNNHMM over MLPHMM method.

IV. CONCLUSIONS

In this work SVD based HMM model and other ANN methods like MLP and CNN are developed and simulated for writer identification. The features are fed to the HMM models. For each state in the HMM model, the features are extracted using MLP and CNN. Though the hybrid methods MLPHMM and CNNHMM have shown much better performance than the existing methods, its performance was not better than the HMMBW. The HMMBW has outperformed the MLPHMM and CNNHMM models, since the features in MLPHMM and CNNHMM are probability based whereas in HMMBW, the features are dependent on the actual quantities of information content of the image like singular values. Hence the concept of hybrid model has worked for smallest test to train ratio of 0.47 and for 91% and 92% of overlapping states, respectively, there is still some room for improving models better than HMMBW.

TEST Engineering & Management

May – June 2020
ISSN: 0193-4120 Page No. 7651 - 7658

REFERENCES

1. Fornes, A., Llados, J., Sanchez, G., Bunke, H. (2008).Writer Identification in Old Handwritten Music Scores. In: 8th IAPR Workshop on Document Analysis Systems, 347—353.
2. Sas, J. (2006) Handwriting Recognition Accuracy Improvement by Author Identification. In: L. Rutkowski et al. (eds.), ICAISC 2006, LNAI 4029, 682--691.Springer, Heidelberg.
3. Chaudhry, R., Pant, S. K. (2004) Identification of authorship using lateral palm print—a new concept. J.ForensicScience International, volume (141), 49--57.
4. Schomaker, L. (2007) Advances in Writer Identification and Verification. In: 9th International Conference on Document Analysis and Recognition (ICDAR‘07), volume (2), 1268--1273.
5. Plamondon, R., Lorette, G. (1989) ―Automatic Signature Verification and Writer Identification—The State of the Art,‖ Pattern Recognition, vol. 22, no. 2, pp. 107-131.
6. Leclerc, F., Plamondon, R. (1994) Automatic signature verification: The state of the art 1989-1993. In Progress in Automatic Signature Verification edited by R. Plamandon, World Scientific Publ. Co., pp. 13-19.
7. Gupta, S. (2008).Automatic Person Identification and Verification using Online Handwriting. Master Thesis. International Institute of Information Technology Hyderabad, India
8. Schlapbach, A., Marcus, L. Bunke, H. (2008) A writer identification system for on-line whiteboard data, Pattern Recognition Journal 41 (2008) 23821–23897.
9. Schomaker, L. (2007) Advances in Writer identification and verification, in: Ninth International Conference on Document Analysis and Recognition (ICDAR).
10. A. Rehman, S. Naz, and M. I. Razzak, “Writer identification using machine learning approaches: a comprehensive review,” Multimedia Tools and Applications, pp. 1–43, 2018.
11. A. Brink, J. Smit, M. Bulacu, and L. Schomaker, “Writer Identification Using Directional Ink-Trace Width Measurements,” Pattern Recognition, vol. 45, no. 1, pp. 162–171, 2012.
12. T. Gilliam, R. C. Wilson, and J. A. Clark, “Scribe Identification in Medieval English Manuscripts,” in 20th International Conference on Pattern Recognition (ICPR), 2010, pp. 1880–1883.
13. D. Fecker, A. Asit, V. Margner, J. El-Sana, T. Fingscheidt, V. Margner,¨ J. El-Sana, and T. Fingscheidt, “Writer Identification for Historical Arabic Documents,” in 2014 22nd International Conference on Pattern Recognition (ICPR), 2014, pp. 3050–3055.
14. S. He and L. Schomaker, “Delta-n Hinge: Rotation-Invariant Features for Writer Identification,” in 2014 22nd International Conference on Pattern Recognition (ICPR), 2014, pp. 2023–2028.
15. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in Advances In Neural Information Processing Systems 25. Curran Associates, Inc., 2012, pp. 1097—-1105.
16. T. Bluche, H. Ney, and C. Kermorvant, “Feature Extraction with Convolutional Neural Networks for Handwritten Word Recognition,” in 2013 12th International Conference on Document Analysis and Recognition, 2013, pp. 285–289.
17. M. Jaderberg, A. Vedaldi, and A. Zisserman, “Deep Features for Text Spotting,” in Computer Vision – ECCV 2014. Springer International Publishing, 2014, vol. 8692, pp. 512–528.
18. F. Wahlberg, T. Wilkinson, and A. Brun, “Historical Manuscript Production Date Estimation Using Deep Convolutional Neural Networks,” in 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), 2016, pp. 205–210.

May – June 2020
ISSN: 0193-4120 Page No. 7651 - 7658

19. S. Fiel and R. Sablatnig, “Writer Identification and Retrieval Using a Convolutional Neural Network,” in Computer Analysis of Images and Patterns: 16th International Conference, CAIP 2015, Valletta, Malta, September 2-4, 2015, Proceedings, Part II, G. Azzopardi and N. Petkov, Eds. Springer International Publishing, 2015, pp. 26–37.
20. V. Christlein, D. Bernecker, A. Maier, and E. Angelopoulou, “Offline Writer Identification Using Convolutional Neural Network Activation Features,” in Pattern Recognition: 37th German Conference, GCPR 2015, Aachen, Germany, October 7-10, 2015.
21. L. Xing and Y. Qiao, “DeepWriter: A Multi-Stream Deep CNN for Textindependent Writer Identification,” in 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), 2016, pp. 584–589
22. Y. Tang and X. Wu, “Text-Independent Writer Identification via CNN Features and Joint Bayesian,” in 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), 2016, pp. 566–571.
23. Bensefia, A., Paquet, T., Heutte, L. (2005) ―A Writer Identification and Verification System,‖ Pattern Recognition Letters, vol. 26, no. 10, pp. 2080-2092.
24. Bensefia, A., Paquet, T., Heutte, L. (2005) Handwritten Document Analysis for Automatic Writer Recognition,‖ Electronic Letters on Computer Vision and Image Analysis, vol. 5, no. 2, pp. 72-86.
25. Bensefia, A., Nosary, A., Paquet, T., Heutte, L. (2002)-Writer Identification by Writer‘s Invariants,‖ Proc. Eighth Int‘l Workshop Frontiers in Handwriting Recognition, pp. 274-279.
26. Bensefia, A., Paquet, T., Heutte, L. (2003) ―Information Retrieval Based Writer Identification,‖ Proc. Seventh Int‘l Conf. Document Analysis and Recognition (ICDAR), pp. 946-950.
27. Marti, U.V., Messerli, R., Bunke, H. (2001) ―Writer Identification Using Text Line Based Features,‖ Proc. Sixth Int‘l Conf. Document Analysis and Recognition (ICDAR), pp. 101-105.
28. Hertel, C., Bunke, H. (2003) ―A Set of Novel Features for Writer Identification,‖ Proc. Fourth Int‘l Conf. Audio and Video-Based Biometric Person Authentication, pp. 679687.
29. L. Xing and Y. Qiao, “DeepWriter: A Multi-Stream Deep CNN for Textindependent Writer Identification,” in 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), 2016, pp. 584–589