Why Not?
Tell us the Reason for Writer Dissimilarity

Chandranath Adak[1,2], Bidyut B. Chaudhuri[3,4], Chin-Teng Lin[2], Michael Blumenstein[2]
[1]Centre for Data Science, JIS Institute of Advanced Studies & Research, JIS University, Kolkata-700091, India
[2]Centre for AI, School of Computer Science, University of Technology Sydney, NSW-2007, Australia
[3]Techno India University, Kolkata-700091, India
[4]CVPR Unit, Indian Statistical Institute, Kolkata-700108, India
chandra@jisiasr.org

***Abstract*—Writer verification has drawn significant attention over the past few decades due to its extensive applications in forensics and biometrics. In traditional writer verification, handwriting similarity/dissimilarity analysis is mostly performed by extracting two feature vectors from two respective handwritten samples, followed by comparing them in relation to their similarity. In the state-of-the-art writer verification approaches, a distance metric is usually employed in terms of the similarity between two handwritten samples. If the distance between two handwritten samples is greater than a given threshold, then the samples are assumed to be written by two different writers, otherwise, they are considered to be due to the same writer. In this paper, for the very first time, we propose a model that generates English sentences to explain reasons for writer dissimilarity/similarity. First, our proposed model obtains features from handwritten images by employing a convolutional neural network, verifies the writer using a Siamese architecture, and generates English words using a recurrent neural network. Finally, these two networks are merged using an affine transformation to produce an explanatory sentence in support of writer similarity/dissimilarity. We evaluated our model on a handwritten numeral database of 100 writers and obtained promising results.**

***Index Terms*—Deep Neural Network, Explainable Artificial Intelligence, Handwriting Understanding, Writer Verification.**

I. Introduction

The eXplainable Artificial Intelligence (XAI) [1] has become a hot research area for the last couple of years, where the primary focus is to make machine learning models human-comprehensible [2], [3]. There are enormous opportunities and challenges in XAI as reported by Arrieta et al. in [1].

In this paper, we primarily work on "post-hoc explainability" where the machine provides an explanation after taking an action. This explanation can be of multiple types, such as textual, visual, featural, architectural, etc. [1]. For our study, we prefer the textual explanation. One popular example of such an explanation is "image caption generation" [3], [4], where a few descriptive lines in natural language are produced from an image.

Our current research aims to generate the text explanation for biometric authentication. We work with handwriting, which is a strong behavioral biometric and accepted as valid evidence in criminal courts of several countries [5]. In biometrics, forensics, and document image analysis domains, handwriting authentication or writer verification is a recognized topic of research [6]. Writer verification is a process where a questioned handwriting specimen is compared with the writing samples of a known writer-database [7]. Usually, in the writer verification task, we obtain two separate feature vectors from two writing samples and compare these two feature vectors based on a distance metric. If the metric is smaller than a certain threshold, then the two samples are assumed to be written by the same writer, otherwise, they are written by two different writers. The past methods in the literature [6]–[9] produce such binary outputs, either '0' (if same) or '1' (if different) by analyzing the writing characteristics inherently, which appear as a black-box [2] to the user. The objective of our research is to open up the black-box in order to attain justifiable and legitimate explanations for understanding the handwriting/writer [10].

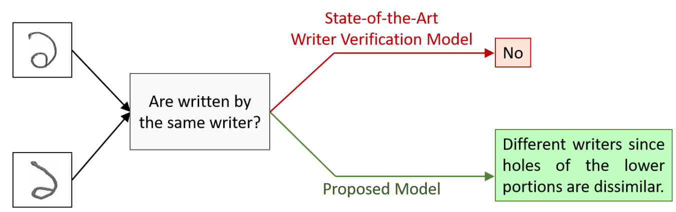

Fig. 1: Illustration of the explainability of our proposed model.

In this paper, by leveraging the writer-verification output, we generate a human-understandable text explanation of the reason behind the writer-similarity/dissimilarity. This is our major contribution in this paper. An illustration regarding the explainability of our proposed model is presented in Fig. 1.

In writer verification, deep neural network-based features are extracted from the handwriting sample, and a Siamese architecture [11] is employed for similarity learning. To explain the reason for writer similarity/dissimilarity, we generate text in the English language. We use here a recurrent neural network with an LSTM (Long Short-Term Memory) unit [12] for text generation, since this network works well in machine translation [13] and image caption generation [3], [4]. We perform our experiments on a Bengali offline handwritten numeral database [14].

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

Although we are motivated by image caption generation research, our work is markedly different from such caption generation. The image caption generator [4], [15] takes an input image, analyzes the objects in it, builds semantic relationships among the objects, and then produces the captioned text. On the other hand, our system takes two handwritten images in parallel and analyzes the reason for their writer similarity/dissimilarity, and finally, produces the textual explanation for authenticity.

To the best of our knowledge, our work is the earliest attempt of its kind in producing human-comprehensible justifications for biometric authentication. We did not come across any paper on this topic to the best of our exhaustive searching capacity (up to 10th April, 2020).

The rest of this paper is organized as follows. Section II describes the proposed methodology. Then Section III presents and analyzes the experimental results. Finally, Section IV concludes the paper.

II. Proposed Method

In this section, we discuss our proposed methodology in detail. As mentioned earlier in Section I, our system takes two handwritten images as input and generates the explanation for writer similarity/dissimilarity in an English sentence. In this paper, we call our proposed method "*ReasonTeller*". ReasonTeller takes assistance from two independently trained modules, (a) an image module and (b) a language module. The image module is responsible for understanding the handwritten image features, which aids the writer verification also. The language module is responsible for learning the sentence structure. We now discuss these two modules followed by a discussion about ReasonTeller.

A. Image Module

The objective of this module is to extract the features from handwritten samples in order to authenticate the authorship.

1) Feature Extraction: In computational handwriting analysis, the features are extracted first from a handwritten sample to represent the writing in a higher-dimensional real vector space.

From a handwritten image sample, we extract deep features due to their superior performance compared to hand-crafted features [16]. For feature extraction, we choose Inception-ResNet-v2 [17], since it works better than some other contemporary deep networks, such as Xception net, Inception-v4, etc. [17], [18]. The Inception-ResNet-v2 usually takes an input of size 299×299. In this paper, we work with a handwritten isolated numeral image (H), which is subsequently normalized into the size of 299×299. From the "average pooling" [17] layer of Inception-ResNet-v2 (say, IR-net), we obtain a 1536-dimensional feature vector (f_H) for each handwritten sample. This feature vector is first used in writer verification to authenticate a writer and latter engaged in ReasonTeller to provide the reason behind writer similarity/dissimilarity.

2) Writer Verification: In writer verification, an unknown handwritten sample is usually compared with a known one to check writing similarity [16]. Therefore, two handwriting inputs are fed here.

For the verification, we adopt the concept of a Siamese net [11], where two identical neural networks progress in parallel and produce deep feature vectors f_{Hi} and f_{Hj} from two different input images H_i and H_j, respectively [16]. The neural twins of the Siamese net are actually two IR-nets that share weights. The objective of the Siamese net is writer-similarity learning. Therefore, the Siamese twins are joined by a loss function $\mathcal{L}$, for which we use a contrastive loss function [19], given below.

$$\mathcal{L}(H_i, H_j, l) = \alpha(1-l)D^2 + \beta l\{\max(0, m-D)\}^2 \quad (1)$$

where, label $l = 0$ if H_i and H_j samples are written by the same writer, and $l = 1$, otherwise. Two scalar quantities α and β are fixed empirically as $\alpha = 0.5$ and $\beta = 0.5$. To compare H_i and H_j, the Euclidean distance (D) is used as a similarity metric, i.e., $D = D(H_i, H_j) = \|f_{Hi} - f_{Hj}\|_2$. The margin m is set as the average distance of all sample pairs in the training dataset.

To verify whether two samples H_i and H_j are written by two *different* writers or by the *same* writer, a threshold t_d is used. If $D(H_i, H_j)$ is less than t_d, then H_i and H_j are considered to be written by the same writer, otherwise, they are written by two different writers. Thus, from the writer verification block, we can generate an English word either "same" or "different" that is used in ReasonTeller. In Fig. 2, we present the writer verification block under the image module.

The image module is initially trained independently. We employ mini-batch gradient descent with momentum to optimize the cost. Regularization is also used to reduce overfitting. The hyper-parameters are set empirically.

B. Language Module

In this paper, our main objective is to generate an explanatory English sentence $S = \{S_0, S_1, \ldots, S_N\}$, which provides the human-understandable reason for writer similarity/dissimilarity. Therefore, we need a module that can generate natural language/text. We now discuss this language module.

To generate the natural language/explanatory sentence, we take some knowledge, especially the machine translation [13] idea, from the NLP (Natural Language Processing) domain [20]. For sequence tasks such as machine translation, the recurrent neural network is widely used since it performs reasonably well [13], [20]. Recently, the computer vision community has successfully applied recurrent architectures for image caption generation [4], [15]. Therefore, in our language module also, we employ the recurrent net.

For computational purposes, an English word ($S_i \in S$) is required to be represented in a numerical vector format. A word is initially encoded here using a one-hot-vector. However, the dimension of such a vector representation is

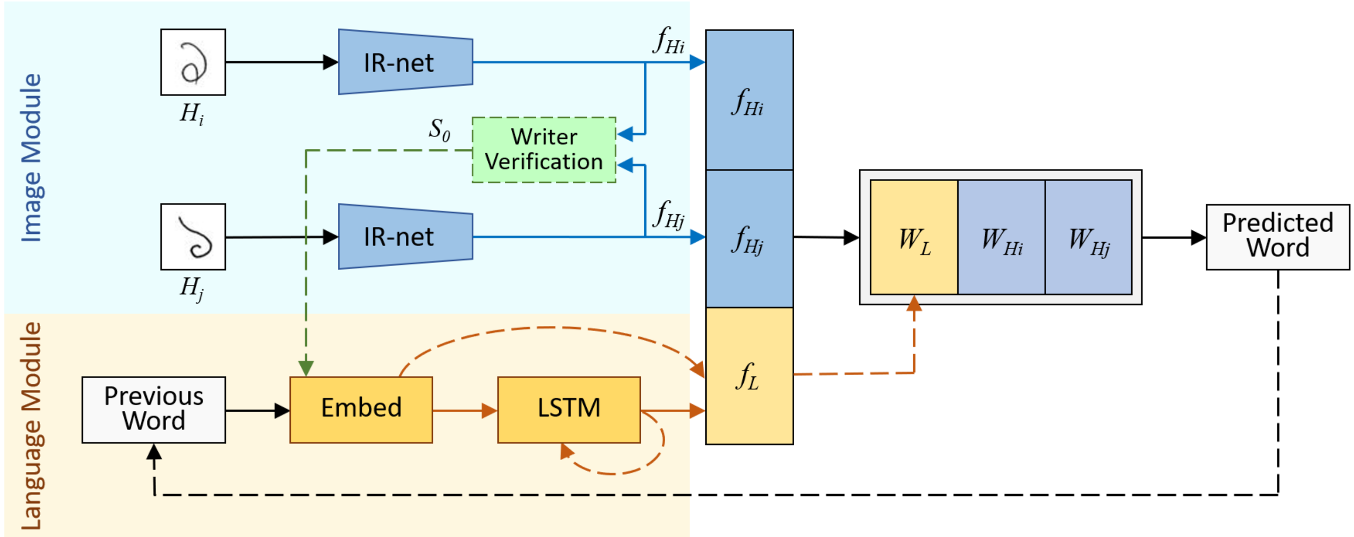

Fig. 2: Workflow of the proposed ReasonTeller model.

extremely large, i.e., equal to the dictionary size. Therefore, the one-hot-vector representation is further embedded into a lower dimension eS_i, where e is a weight matrix [21], [22].

We present our language module inside Fig. 2. In this module, at every time step t, the previous word ($S_{t-1} \in S$) of a sentence S is fed into the embedding layer to obtain an embedded word x_t ($= eS_{t-1}$). Now, the embedded word x_t is fed into an LSTM (Long Short-Term Memory) unit [12] of a recurrent neural net. We use LSTM due to its superior performance over some other recurrent units, such as a fully gated unit, minimal gated unit, etc. [23]. The embedding dimension and the size of the LSTM memory are both set to 512.

The LSTM is used to predict a word S_t of a sentence S at every time step t. We obtain a probability distribution p_t of all the words, at each time step t, by employing the $softmax$ function on the LSTM output, i.e., $p_t = softmax(\text{LSTM}(x_t))$, where $x_t = eS_{t-1}$.

The loss is defined below as the sum of negative log likelihood of the correct word at every step.

$$\mathcal{L}(S) = -\sum_t \log p_t(S_t) \qquad (2)$$

We use mini-batch gradient descent with momentum to minimize the cost. We empirically set the hyperparameters used in this module.

From the language module, we obtain some language feature f_L by concatenating the embedded word and LSTM output, which is also employed in ReasonTeller.

Similar to the image module, the language module is also trained independently during the initial phase of learning. Now, the ReasonTeller architecture is discussed below.

C. ReasonTeller

The ReasonTeller architecture merges the image module and language module to generate an explanatory sentence pertaining to writer similarity/dissimilarity.

By leveraging the writer verification block of the image module, we have obtained an English word (either "same" or "different") which is transferred to the language module in order to use it as the first word (S_o) of the prefix.

From the image module, we obtained the image features f_{Hi} and f_{Hj}, which are now combined with the language feature f_L obtained from the language module. For this combination, an affine transformation is performed, as follows.

$$p_S = softmax(W_{Hi}f_{Hi} + W_{Hj}f_{Hj} + W_L f_L + b) \qquad (3)$$

where, W_{Hi}, W_{Hj}, W_L are weight matrices, b is a bias vector, and p_S is a probability distribution over the predicted word. On one hand, W_{Hi} and W_{Hj} learn to produce a set of words with respect to the handwriting characteristics, on the other hand, W_L learns the word sequence in an English sentence.

As we mentioned earlier, initially the language module and image module are trained independently, and then combined in ReasonTeller to train further for producing a joint explanatory English sentence. We adopt the idea of [24] here, since the exclusive handling of the image and language modules perform quite well [15]. In the training stage of ReasonTeller, the language module has knowledge of the outcome of the writer-verification block, which is crucial information for producing English words pertaining to writer similarity/dissimilarity. The weights W_{Hi} and W_{Hj} are learned during the latter training of ReasonTeller. During training, the ground-truthed word is input to the language module, however, during testing, the predicted previous-word of the prefix is fed to the module.

To find the best natural language output, we use the beam search technique [4], [25] with a beam-size of 20.

The loss ($\mathcal{L}$) is defined here as the summation of the negative log-likelihood of the correct word at every step t.

$$\mathcal{L}(H_i, H_j, S) = -\sum_t \log p_{S,t}(S_t) \qquad (4)$$

During the training of ReasonTeller, we use mini-batch gradient descent with momentum to minimize the cost. For accelerating the training speed, we employ batch normalization [26]. Empirically, we fix the hyperparameters, such as initial learning rate = 10^{-2}, learning rate decay = 10^{-5}, and momentum = 0.9, on a tuning set.

III. Experiments and Discussion

In this section, first, we discuss the database employed for executing the experiments and then evaluating the system performance.

A. Database Employed

For our experimentation, we required a database which contains handwriting samples of multiple writers. Furthermore, it should contain some ground-truth information in English sentences regarding the explanation for writer similarity/dissimilarity.

In this paper, we employed the Bengali offline handwritten numerals of the *NewISIdb:HwC* database [14], which contains samples obtained from 100 writers. We used nine numeral classes $(1, 2, \ldots, 9)$ here as shown in Fig. 3. We refrained from using numeral '0', since it bears the least information for offline writer inspection [14]. In the rest of the paper, we call this database as DB_N. In DB_N, 100 writers wrote 11 copies of each of the nine numeral classes. Therefore, DB_N contains a total of 9900 $(= 100 \times 11 \times 9)$ handwritten numeral images. Further details regarding the generation of DB_N can be found in [14].

Fig. 3: A set of Bengali handwritten numerals from DB_N.

The *NewISIdb:HwC* database [14] does not contain any explanatory sentence-level ground-truth. Therefore, for ground-truth generation, we consulted 20 handwriting and linguistic experts having high professional proficiency in the English language. For every pair of writer's handwriting in DB_N, the experts stated their subjective reasons for similarity/dissimilarity. The experts were aware of the ground-truth information of the writers. Among 100 writers, we had 5050 pairs, out of which 4950 pairs were for dissimilar writers and 100 pairs for similar writers. For each of these 5050 pairs with nine separate numerals, we chose the 3 most logical and unbiased sentences as ground-truth explanatory sentences. Some brief details of DB_N are shown in TABLE I.

TABLE I: Brief Details of the Database Employed (DB_N)

	Writer Similarity	Writer Dissimilarity	Total
# Writer	***# Writer Pair***		
100	100	4950	5050
# Image	***# Explanatory Sentence***		
100×9×11 = 9900	100×9×3 = 2700	4950×9×3 = 133650	136350

B. Performance of Writer Verification

The language module of our proposed model depends on writer verification, as we discussed earlier in Section II (refer to Fig. 2). Therefore, we required a good writer verification system. For this, we evaluated the performance of our writer verification block and compared it with some state-of-the-art methods [7]–[9], [27], [28].

For writer verification, in the image module (refer to Fig. 2), the weights of the initial few layers of IR-net were trained on the ILSVRC dataset [29] and then transferred to the IR-net.

TABLE II: Writer Verification Performance on DB_N

Method	***Accuracy (%)***
Macro-micro features [27]	80.53
Texture [8]	93.89
LBP [28]	94.16
Contour-hinge [7]	94.71
Textural CNN-Siamese [9]	97.62
Proposed	**98.74**

In TABLE II, we analyze the performances of our writer verification block and some major state-of-the-art methods on DB_N. For the experimentation, DB_N was divided into training, validation, and test sets in the ratio of $3:1:1$. The writer verification performance was calculated in terms of accuracy (balanced) [16]. From this table, we can see our method performed the best and achieved 98.74% accuracy for writer verification.

C. Performance of ReasonTeller

The language module (refer to Fig. 2) was initially trained with the British National Corpus (BNC)[1] and the Flickr1M [30] image captioned text corpora. The learned weights of this language module were transferred during the entire training of ReasonTeller.

For the experimental analysis of our proposed ReasonTeller, we used 5-fold cross-validation. Besides random subsampling, here we ensured 990 pairs of dissimilar writers and 20 pairs of similar writers in each fold. Finally, we present the average performance from the 5 experiments.

For the performance measure, we used the BLEU score [31], which is commonly employed in machine translation and image caption generation papers [4], [15], [24], [31]. We employed BLEU-1, 2, 3, 4 here. Furthermore, we used another evaluation metric, i.e., METEOR [32] due to the recent criticism of BLEU [33].

In TABLE III, we present the performance of our proposed ReasonTeller in terms of BLEU and METEOR. In this table, we also present the BLEU score of "human" that was

[1]BNC: www.natcorp.ox.ac.uk, last retrieved on 30$^{\text{th}}$ Jan., 2020.

TABLE III: Performance of ReasonTeller on DB$_N$

Approach	*Dissimilarity/Similarity*	*BLEU-1*	*BLEU-2*	*BLEU-3*	*BLEU-4*	*METEOR*
ReasonTeller	Writer Similarity	63.23	40.45	26.92	18.67	17.90
	Writer Dissimilarity	65.73	43.57	29.20	20.81	19.35
	Total	64.26	42.12	28.56	19.89	18.43
Human	Writer Similarity	69.64	46.90	31.45	24.20	22.42
	Writer Dissimilarity	72.43	49.86	36.85	26.12	24.87
	Total	71.50	48.67	34.62	25.36	23.70

calculated by comparing one of the three experts' sentences against the other two. This was repeated for each of the three experts, and finally, we took their average BLEU score. For DB$_N$, ReasonTeller obtained the overall BLEU-4 and METEOR scores of 19.89 and 18.43, respectively, which was quite an encouraging performance compared to the "human"-based approach. The performance of writer-dissimilarity is better than writer-similarity.

In Fig. 4, we present some qualitative results of our proposed ReasonTeller.

D. Limitations of ReasonTeller

Our proposed system has a few limitations that are briefly discussed as follows.

(i) ReasonTeller has a dependency on the writer-verification block. Therefore, we require a very accurate verification strategy.

(ii) Our writer verification block does not perform well on low inter-variable and high intra-variable handwriting [16]. Therefore, ReasonTeller may generate false positive and false negative results in relation to low inter-variability and high intra-variability of individual writing. Some false positive and false negative results are shown in Fig. 5. This limitation can be overcome by incorporating some writer verification techniques [34] that can handle such issues.

(iii) There is another limitation in our study, whereby we compare two writers based on numerals of the same class. For example, we compare numeral '2' written by *Writer-A* with the numeral '$\mathfrak{2}$' penned by *Writer-B*. This limitation persists owing to the ground-truth information of DB$_N$. Our ReasonTeller model can work when trained on numerals of different classes for comparing two writers.

E. Comparison

We compared our writer verification block with some state-of-the-art methods in TABLE II, where our system performed the best.

For comparing our ReasonTeller, to the best of our searching capacity, we did not find any study of a similar kind to produce explanations for writer dissimilarity. Therefore, we were not able to perform any comparative analysis. However, in TABLE III, we provide the "human"-performance as a reference.

IV. Conclusion

In this paper, we propose a method for writer verification as well as for generating English text to understand the reasons for writer similarity/dissimilarity. In writer verification, we employed a deep neural network for handwriting feature extraction, and a Siamese net for similarity learning. For explanatory text generation, we used a recurrent net with an LSTM. To evaluate the performance of our system, we employed a Bengali offline handwritten numeral database. For writer verification, we obtained an accuracy of 98.21%. Our ReasonTeller system obtained an overall BLEU-4 score of 19.89 and a METEOR score of 18.43, which is quite promising with respect to the performance of a human.

In the future, we will endeavor to improve our system performance. Our current system takes the handwritten numeral as an input. Later, we will also attempt to handle character, word, and paragraph-level inputs to the system.

Acknowledgment

We heartily thank all the handwriting and linguistic experts for their fruitful consultation and inputs.

References

[1] A. B. Arrieta et al., "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI", Information Fusion, vol. 58, pp. 82-115, June 2020.

[2] P. Voosen, "How AI Detectives are Cracking Open the Black Box of Deep Learning", Science, AAAS, 6 July 2017.

[3] Y. Dong, H. Su, J. Zhu, B. Zhang, "Improving Interpretability of Deep Neural Networks with Semantic Information", CVPR, pp. 975-983, 2017.

[4] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, "Show and Tell: A Neural Image Caption Generator", CVPR, pp. 3156-3164, 2015.

[5] R. N. Morris, "Forensic Handwriting Identification: Fundamental Concepts and Principles", Academic Press Inc., 2000.

[6] L. Schomaker, "Writer Identification and Verification", In: N. K. Ratha, V. Govindaraju (eds) Advances in Biometrics, pp. 247-264, Springer, London, 2008.

[7] M. Bulacu, L. Schomaker, "Text-Independent Writer Identification and Verification Using Textural and Allographic Features", IEEE Trans. on PAMI, vol. 29, no. 4, pp. 701-717, 2007.

[8] R. K. Hanusiak, L.S. Oliveira, E. Justino, R. Sabourin, "Writer Verification Using Texture-based Features", IJDAR, vol. 15, no. 3, pp. 213-226, 2012.

[9] C. Adak, S. Marinai, B. B. Chaudhuri, M. Blumenstein, "Offline Bengali Writer Verification by PDF-CNN and Siamese Net", DAS, pp. 381-386, 2018.

[10] C. Adak, "A Study on Automated Handwriting Understanding", PhD Thesis, UTS, Australia, 2019.

[11] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, R. Shah, "Signature Verification Using a 'Siamese' Time Delay Neural Network", NIPS, pp. 737-744, 1993.

[12] S. Hochreiter, J. Schmidhuber, "Long Short-Term Memory", Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[13] N. Kalchbrenner, P. Blunsom, "Recurrent Continuous Translation Models", ACL, pp. 1700-1709, 2013.

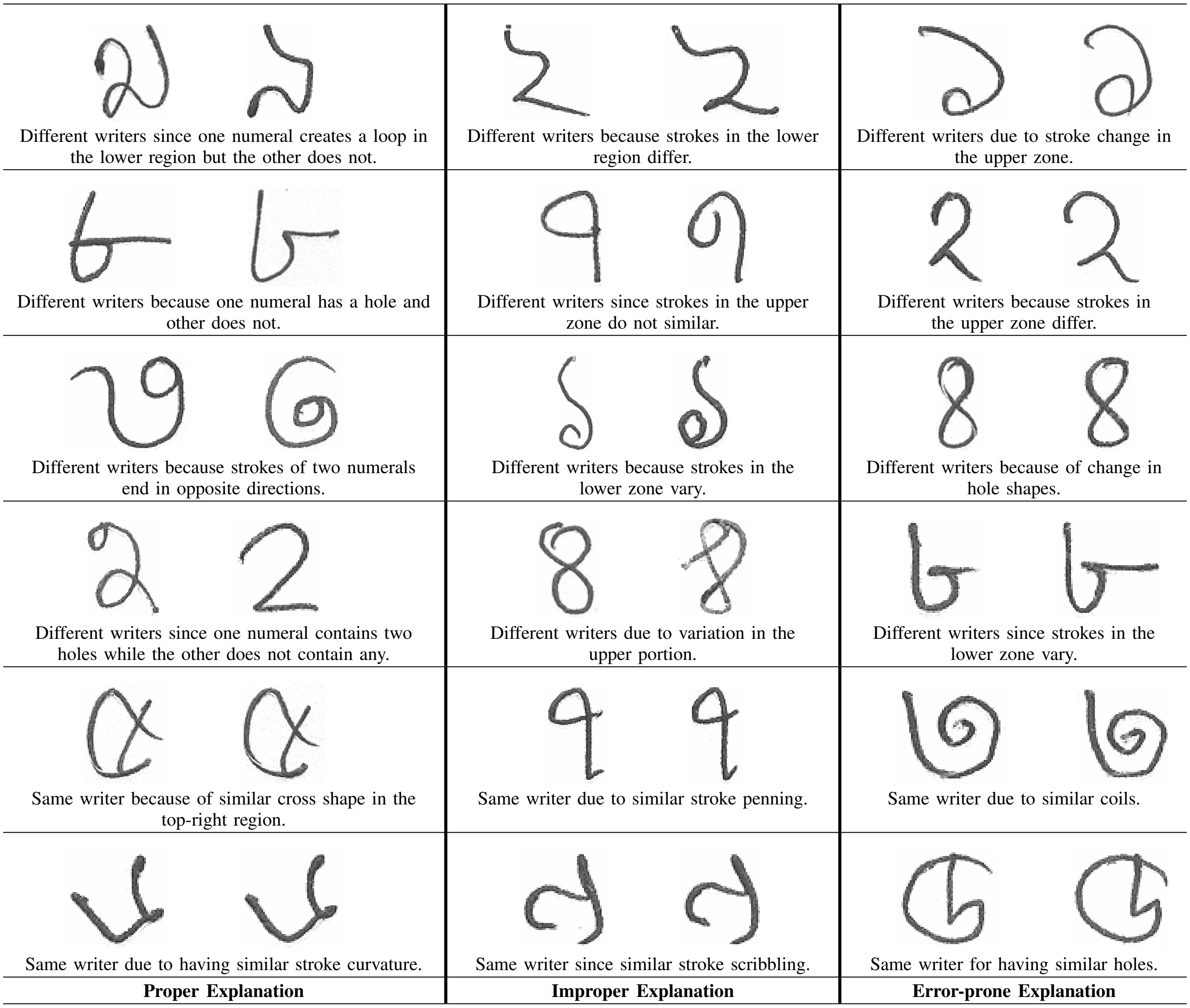

Fig. 4: Some qualitative results. Each box contains a pair of handwriting samples with ReasonTeller's explanation.

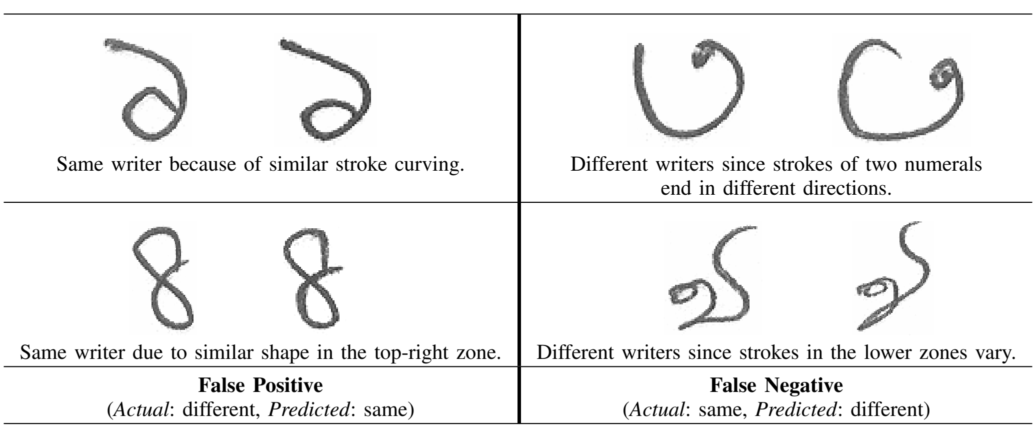

Fig. 5: Some qualitative results showing *false positive* and *false negative*. Each box contains a pair of handwriting samples with ReasonTeller's explanation.

[14] C. Adak, B. B. Chaudhuri, “Writer Identification from Offline Isolated Bangla Characters and Numerals”, ICDAR, pp. 486-490, 2015.
[15] M. Tanti, A. Gatt, K. P. Camilleri, “Where to put the Image in an Image Caption Generator”, Natural Language Engineering, vol. 24, no. 3, pp. 467-489, 2018.
[16] C. Adak, B. B. Chaudhuri, M. Blumenstein, “An Empirical Study on Writer Identification and Verification from Intra-Variable Individual Handwriting”, IEEE Access, vol. 7, pp. 24738-24758, 2019.
[17] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning”, AAAI, pp. 4278-4284, 2017.
[18] F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions”, CVPR, pp. 1800-1807, 2017.
[19] R. Hadsell, S. Chopra, Y. LeCun, “Dimensionality Reduction by Learning an Invariant Mapping”, CVPR, pp. 1735-1742, 2006.
[20] L. Deng, Y. Liu, “Deep Learning in Natural Language Processing”, Springer, Singapore, 2018.
[21] Y. Bengio, R. Ducharme, P. Vincent, C. Janvin, “A Neural Probabilistic Language Model”, Journal of Machine Learning Research, vol. 3, pp. 1137-1155, 2003.
[22] T. Mikolov, K. Chen, G. Corrado, J. Dean, “Efficient Estimation of Word Representations in Vector Space”, ICLR, 2013.
[23] G. Weiss, Y. Goldberg, E. Yahav, “On the Practical Computational Power of Finite Precision RNNs for Language Recognition”, ACL, pp. 740-745, 2018.
[24] L. A. Hendricks, S. Venugopalan, M. Rohrbach, R. Mooney, K. Saenko, T. Darrell, “Deep Compositional Captioning: Describing Novel Object Categories without Paired Training Data”, CVPR, pp. 1-10, 2016.
[25] S. Wiseman, A. M. Rush, “Sequence-to-Sequence Learning as Beam-Search Optimization”, Conf. on Empirical Methods in Natural Language Processing (EMNLP), pp. 1296-1306, 2016.
[26] S. Ioffe, C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”, ICML, pp. 448-456, ICML, 2015.
[27] S. N. Srihari, S.-H. Cha, H. Arora, and S. Lee, “Individuality of handwriting”, J. Forensic Sci., vol. 47, no. 4, pp. 856-872, 2002.
[28] S. Pal et al., “Performance of an Off-Line Signature Verification Method Based on Texture Features on a Large Indic-Script Signature Dataset”, Proc. DAS, pp. 72-77, 2016.
[29] O. Russakovsky et al., “ImageNet Large Scale Visual Recognition Challenge”, Int. Journal of Computer Vision, vol. 115, no. 3, pp. 211-252, 2015.
[30] M. J. Huiskes, M. S. Lew, “The MIR Flickr Retrieval Evaluation”, Int. Conf. on Multimedia Information Retrieval (MIR), pp. 39-43, 2008.
[31] K. Papineni, S. Roukos, T. Ward, W. J. Zhu, “BLEU: A Method for Automatic Evaluation of Machine Translation”, ACL, pp. 311-318, 2002.
[32] M. Denkowski, A. Lavie, “Meteor Universal: Language Specific Translation Evaluation for Any Target Language”, Workshop on Statistical Machine Translation (WMT), pp. 376-380, 2014.
[33] E. Sulem, O. Abend, A. Rappoport, “BLEU is Not Suitable for the Evaluation of Text Simplification”, ACL, pp. 738-744, 2018.
[34] C. Adak, B. B. Chaudhuri, C.-T. Lin, M. Blumenstein, “Intra-Variable Handwriting Inspection Reinforced with Idiosyncrasy Analysis”, arXiv:1912.12168, 2019.