Revue d'Intelligence Artificielle
Vol. 38, No. 2, April, 2024, pp. 523-529
Journal homepage: http://iieta.org/journals/ria

Decoded-ViT: A Vision Transformer Framework for Handwritten Digit String Recognition

Vanita Agrawal, Jayant Jagtap, MVV Prasad Kantipudi*

Symbiosis Institute of Technology (SIT), Symbiosis International (Deemed University) (SIU), Lavale, Pune 412115, Maharashtra, India

Corresponding Author Email: mvvprasad.kantipudi@gmail.com

Copyright: ©2024 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/ria.380215

Received: 6 October 2023
Revised: 11 November 2023
Accepted: 12 December 2023
Available online: 24 April 2024

Keywords:
historical documents, DIDA dataset, handwritten digit string recognition, vision transformer, LIME

ABSTRACT

In the era of digitization, handwritten document recognition has several applications, like historical information preservation, postal address recognition, etc. The conservation and analysis of priceless cultural treasures depend heavily on the handwritten digit string recognition from historical documents. The prominent challenges in recognition are writing style variations, noise, distortions, and limited data. This paper suggests a novel method for overcoming the difficulties in reading complex, fading, and old handwritten documents that contain digit strings. The goal is to create a reliable and effective system that automatically recognizes digit strings from ancient manuscripts, helping to digitize records. Hence, this paper proposes a robust vision transformer framework to identify handwritten digit strings without segmentation of digits from uncleaned images of smaller datasets. The proposed method is a four-step procedure comprised of preprocessing, feature extraction through tokenization, recognition using the attention mechanism of a vision transformer, and outcome decoding using a beam search decoder. The performance of the proposed method is compared with the hybrid approach consisting of a Convolutional Neural Network and Long Short-Term Memory (CNN-LSTM). The proposed method achieved word accuracy of 56% with a loss below 0.6 in less time. The results show that the proposed model is a fast learner and can be used in real-time scenarios where results are expected in less time. The proposed deep learning model performance explanation is also discussed in this paper with the help of the Local Interpretable Model-agnostic Explanations (LIME) technique. The results of this study impact the digitization of postal services. The generalization of the proposed method by providing Software-as-a-Service (SaaS) for real-time applications is explored as a future research direction.

1. INTRODUCTION

1.1 Background

The conversion of handwritten historical documents into digital format is the need of the day so that readers can access them easily and quickly through the digital library. The main components of historical documents are text, images, and digit strings. Digit strings in historical documents could correspond to various information kinds, including dates, population count, measurements, page numbers, etc. Figure 1 shows the sample Swedish handwritten historical document [1, 2], and the red color boxes are the digit strings present in the document.

In India, Raksha Bandhan is celebrated as an expression of responsibility among siblings. On this day every year, sisters wrap their brothers' wrists with a talisman known as a rakhi. As they stay in different locations, rakhis are sent via courier. Sometimes delivery gets delayed because of the number of couriers and difficulty understanding the address on the envelope. Also, in rural areas, information about birth and death is sent via postcards. Hence, digitizing postal addresses will speed up the sorting and tracking letters based on location, resulting in smooth delivery and avoiding delays. Postal addresses consist of alphabetical words and digit strings similar to historical documents. Examples of digit strings in postal letters are house numbers, Postal Index Number (PIN) codes, or Zone Improvement Plan (ZIP) codes, as shown in Figures 2 (a) and 2 (b) (source: https://www.huppme.com/gift/personalizedhandwritten-letters/).

Figure 1. Sample digit strings present in the handwritten historical document

(a) Postal letter address

(b) Courier address

Figure 2. House number and pin codes

The state-of-the-art method used in computer vision is CNN because of its efficiency in image processing [3-6]. However, the images collected from historical documents are challenging to recognize because of distortions such as noise and varying writing styles. If a technique works on the historical dataset, it can work on real-time data, such as pin codes written in mailing addresses. Hence, a robust, efficient, and reliable method is needed to recognize digit strings from such images.

1.2 Motivation

The work by Dosovitskiy et al. [7] presented the Vision Transformer (ViT) and proved that, if assessed against conventional convolutional neural networks, transformers might excel competitively on image recognition tasks. An intriguing use of transformer design in computer vision is handwritten digit recognition through ViT. However, the ViT lacks locality inductive bias and hence needs larger datasets for training [8]. Lee et al. [9] proposed two mechanisms, shifted patch tokenization (SPT) and locality self-attention (LSA), in ViT architecture so that it could be used on smaller datasets. SPT provides a broader receptive field, leading to the embedding of more spatial features and hence resulting in the boosting of locality inductive bias. At the same time, LSA forces tokens to concentrate on those with strong relationships by making attention to work locally.

1.3 Objective

As the available benchmark historical handwritten digit string dataset, DIDA [1, 2], is small, the work by Lee et al. [9] motivated us to use ViT for handwritten digit string recognition (HDSR). To make the technique useful for HDSR, optimization of hyperparameters is required. Also, for HDSR without segmentation of digits, a methodology is needed to decode the output of ViT. Hence, this paper proposes a method that optimizes the hyperparameter and decodes the output. The proposed framework is named Decoded-ViT. To show that the attention mechanism of Decoded-ViT is better, the results are compared with the CNN-LSTM method [10]. Also, to visualize why there is a difference between Decoded-ViT and CNN-LSTM prediction results, the Local Interpretable Modelagnostic Explanations (LIME) method [11] is used.

2. RELATED WORK

Due to the varying length and writing style of digits, HDSR is problematic [12]. The early methods proposed by the studies [13-15] segment the digit string into single digits. They used multiobjective genetic algorithms for feature selection. Saabni [16] proposed extreme learning machines (ELM) for fast recognition of digits and used sliding window protocol for digit segmentation. The author used the ADA-boosting technique to improve accuracy. Ma et al. [12] proposed using anchor boxes to segment digit strings. But it isn't easy to correctly separate handwritten digits from one another.

Recently, various deep learning methods without segmentation have been proposed for HDSR. The studies designed a residual network and took advantage of the recurrent neural network (RNN) and CTC architecture for HDSR. Hochuli et al. [19] suggested using the two classifiers without segmentation. One to predict the length of a string and another to recognize a string. They trained four CNNs on synthetic data. Aly and Mohamed [20] used a series of hybrid principal component analysis networks (PCANet) and support vector machine (SVM) classifiers for HDSR. The multiple stages of the hybrid model avoided the need for segmentation. Neto et al. [21] presented a model based on a handwritten text recognition workflow. They extracted the features using a CNN. The extracted features are propagated using the bidirectional gated recurrent unit neural network (BGRU), and the loss value is computed through the CTC. The decoding of the model output is done with the help of a beam search decoder. Zhan et al. [22] proposed an architecture based on CNN and CTC. They used multiple dense blocks for feature extraction.

Though much work has been done on HDSR, historical documents are only a little considered. Also, most research is done based on state-of-the-art CNN techniques. Hence, this paper uses vision transformers for the HDSR from handwritten historical documents.

3. PROPOSED METHOD

The inspiration to use vision transformers on smaller datasets came from the approach [9]. The proposed technique, Decoded-ViT for HDSR, is shown in Figure 3.

It is a four-step process with main components: pre-processing, feature extraction, recognition through a vision transformer, and final output decoding with the help of a beam search decoder.

Pre-processing is the series of actions to prepare the data for a deep learning model. It creates the conditions enabling the framework to learn efficiently and produce precise predictions.

One kind of deep learning architecture intended for image identification applications is the Vision Transformer. It

divides pre-processed images into small portions, applies cognitive processing using transformer blocks, and then applies what it learns to produce precise predictions on the images' contents.

Figure 3. The proposed architecture of Decoded-ViT for HDSR

The predictions are then passed to the CTC beam search decoder. The CTC beam search decoder is similar to figuring out a problematic jigsaw with ambiguous component boundaries. The beam search effectively investigates multiple-word formation scenarios for tokens, evaluating and fine-tuning until it locates the most probable transcription of the input.

3.1 Decoded-ViT model

In the Decoded-ViT model, the image is initially downsized during the pre-processing stage, so they are all the same size. Then, the image is normalized based on the mean and standard deviation and converted to grayscale.

After pre-processing, the image patching is done. Patching is done sequentially through patch embedding, concatenation, and positional embedding. The SPT method is used for patching. In the SPT method, first, the images are moved in four directions: left-up, right-up, left-down, and right-down. Then, the features from the four images are concatenated, and patches are generated. As the transformer works on a one-dimensional sequence vector, the two-dimensional patch feature map is flattened, and positional information is extracted using positional embedding.

The one-dimensional embedded tokens are then passed through the transformer encoder layer. The transformer encoder has two components: the attention layer and the Multilayer perceptrons (MLP) layer. Layer normalization is done at the start of both layers. As the dataset is small, the LSA method is used.

The output of the attention layer is given to the feedforward neural network. The data is sequentially processed by a feedforward layer, gaussian error linear unit (GeLU), and another feedforward layer. The activation function used is the GeLU because it combines the functionality of rectified linear activation unit (ReLU) and dropout. The outcome of the transformer layer is forwarded to the MLP head. Then, the logarithmic softmax function is applied for correct output and gradient computation.

The output generated by the transformer is encoded; hence, the CTC beam search decoder decodes the result and converts it to the final digit string. Table 1 shows the summary of the model.

Table 1. Vision transformer model summary

Layer (type)	Output Shape	Param #
Rearrange	[32, 16, 160]	0
LayerNorm	[32, 16, 160]	320
Linear	[32, 16, 128]	20608
SPT	[32, 16, 128]	0
LayerNorm	[32, 17, 128]	256
Linear	[32, 17, 384]	49152
Softmax	[32, 8, 17, 17]	0
Linear	[32, 17, 128]	16512
LSA	[32, 17, 128]	0
LayerNorm	[32, 17, 128]	256
Linear	[32, 17, 256]	33024
GELU	[32, 17, 256]	0
Linear	[32, 17, 128]	32896
FeedForward	[32, 17, 128]	0
Transformer	[32, 17, 128]	0
Identity	[32, 17, 128]	0
LayerNorm	[32, 17, 128]	256
Linear	[32, 17, 81]	10449

Choosing the optimal hyper-parameter value for the model is the most crucial task. Optuna [23] is a hyper-parameter optimizing approach for black-box optimization techniques. Using a fitness function that returns a numerical value, Optuna, a black-box optimizer, assesses the behavior of the hyper-parameters and decides where to sample within subsequent experiments. The optuna trial object specifies the nature and scope of the necessary hyper-parameter adjustments. The parameters tuned were patch size, dim, depth, heads, optimizer, learning rate, dropout, etc.

4. EXPERIMENTAL SETUP

The experiments were done on an HP Z8 G4 workstation with an HP P24v G5 GPU connected to NVIDIA RTX A4000. The model is built on the Pytorch library, and the optimization algorithm Adam, with a learning rate of 0.001, updates network weights.

4.1 Dataset

The DIDA dataset [1, 2] is used to evaluate the proposed method, as it is a historical handwritten digit dataset. The images of DIDA are uncleaned. Two hundred fifty-two thousand eight hundred sixty photos cropped from historical documents make up the DIDA dataset. The writing styles, sizes, orientations, widths, and layouts of the digits in the DIDA dataset vary. The handwritten digit string recognition from the DIDA dataset is challenging because the papers and ink are from the nineteenth century. Multiple degradations and artifacts were produced due to document age and distortions. Background fluctuation, tear, and smearing are artifacts in the document image. Different priests wrote the digits in copperplate, cursive, and Gothic styles. Also, the images are

not cleaned and contain noise. The DIDA dataset's 12k digit string representations are utilized for testing. The dataset was downloaded from https://didadataset.github.io/DIDA/. The dataset's sample images are displayed in Figure 4.

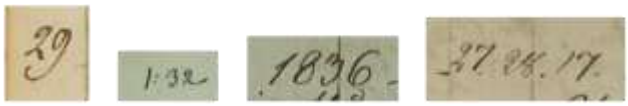


Figure 4. Sample images from the DIDA dataset

The 12000 images of the digit strings are divided alphabetically into two parts: 9600 for training and 2400 for testing. Of the 9600 training images, 1920 images are randomly split for validation. The images are resized and normalized before passing to the model.

4.2 Performance metrics

The model's performance is evaluated using loss, character accuracy, word accuracy, and time. The model output is passed through the logarithmic softmax function, and CTC is used for the loss computation. The character accuracy computes the Levenhstein distance between the prediction and actual labels. At the same time, the word accuracy is calculated by matching the whole digit string.

The results of the suggested method are contrasted with those of the CNN-LSTM model to demonstrate the advantages of the vision transformer model. The CNN-LSTM model used for comparison was proposed by Shi et al. [10] to recognize scene text. The best hyperparameter values suitable for the DIDA dataset are also found through Optuna for CNN-LSTM.

The LIME explainable AI method is used to visualize the features that lead to the prediction result. Based on the weights in the model, the images utilized for evaluation are split into superpixels, which are subsequently visualized [24].

4.3 Results and discussion

The proposed model (Decoded-ViT) achieved a character accuracy of 90.30%, 85.31%, and 83.73% in the training, validation, and testing phases. As per the survey, the only experiment on the DIDA 12k digit string dataset was by Kusetogullari et al. [2]. Their proposed method is DIGITNET, where the DIDA single-digit images are used for training, and digit string images are used for testing. They achieved an accuracy of 75.96%. The outcomes show that the Decoded-ViT architecture can work on smaller datasets and achieve higher accuracy for digit string recognition of historical documents.

Word accuracy achieved by the Decoded-ViT is 70.15%, 55.75%, and 56.00% in the training, validation, and testing phases. The recognition results of Decoded-ViT on a few images of the DIDA dataset are shown in Figure 5.

Figure 6 shows the training loss of the Decoded-ViT and CNN-LSTM, along with the approximate time taken for computation. After 21hr35min, the training loss computed by CNN-LSTM was 0.36, whereas the Decoded-ViT achieved that much loss in 17hr20min. Hence, the overall computation time of Decoded-ViT is less than CNN-LSTM.

The training and validation loss comparison of Decoded-ViT and CNN-LSTM are shown in Figures 7 (a) and 7 (b), respectively.

Figure 5. Recognition result of DIDA handwritten digit string dataset

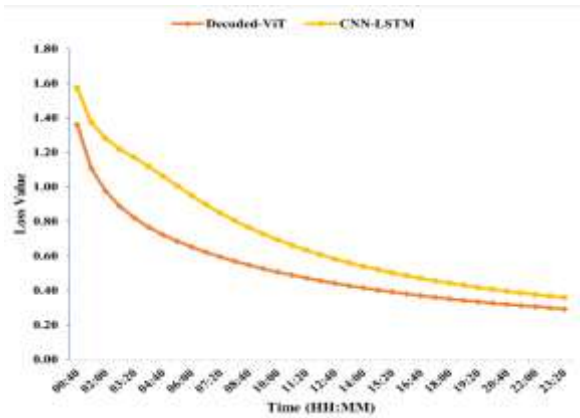
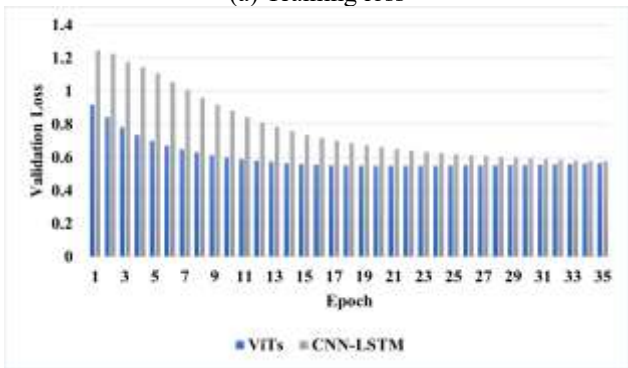


Figure 6. Training loss and time taken by Decoded-ViT and CNN-LSTM

(a) Training loss



(b) Validation loss

Figure 7. Loss comparison of Decoded-ViT and CNN-LSTM

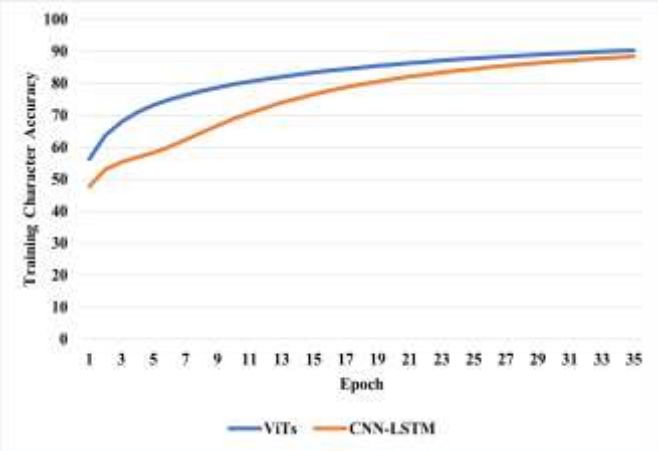


Figure 8. Training character accuracy comparison of Decoded-ViT and CNN-LSTM

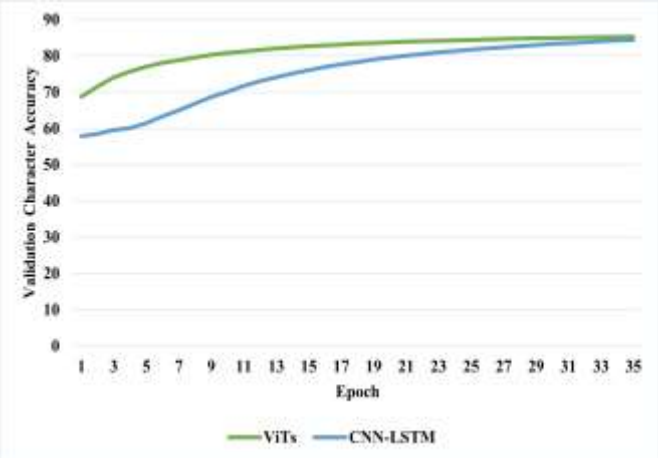


Figure 9. Validation character accuracy comparison of Decoded-ViT and CNN-LSTM

The charts in Figure 8 and Figure 9 show the difference in the training and validation character accuracy of Decoded-ViT and CNN-LSTM, respectively.

The graphs in Figure 10 and Figure 11 show how Decoded-ViT and CNN-LSTM differ in training and validation word recognition correctness, respectively. The outcomes show that the proposed Decoded-ViT model is a fast learner and can be used in any real-time scenario.

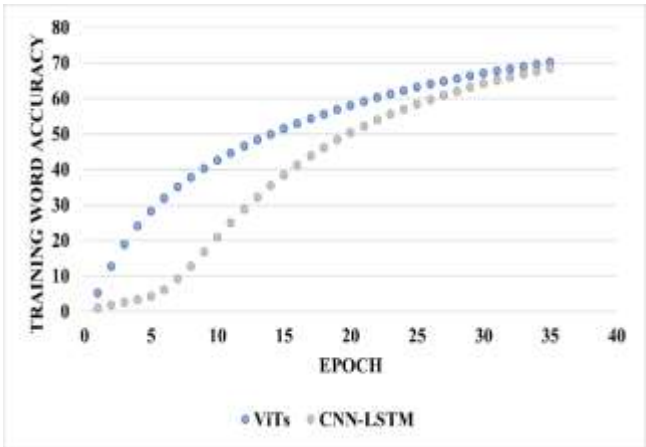


Figure 10. Training word recognition accuracy comparison of Decoded-ViT and CNN-LSTM

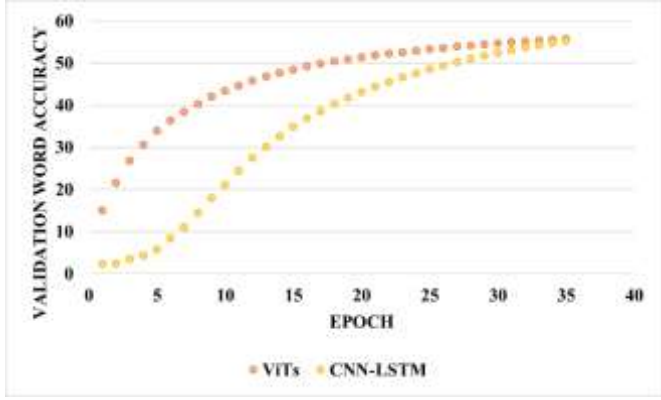
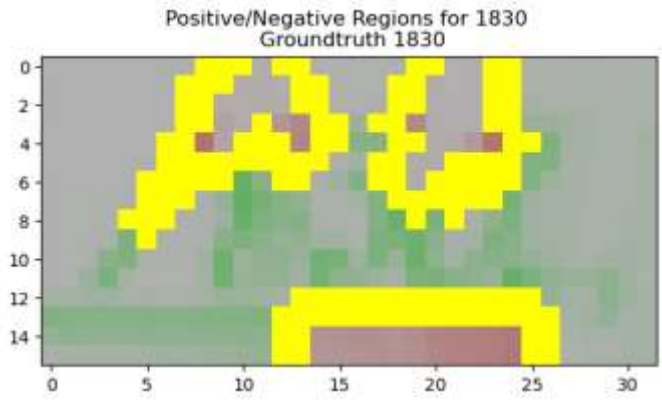
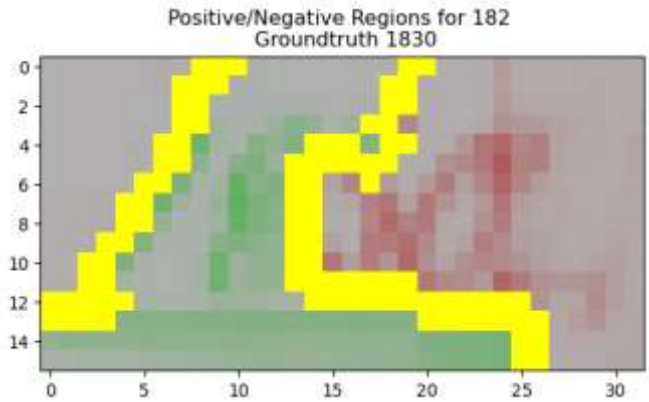


Figure 11. Validation word recognition accuracy comparison of Decoded-ViT and CNN-LSTM

The LIME method visualizes the features that led to the prediction output. The LIME method creates local data by minor random adjustments after sampling comparable data points. Then, the pre-trained proposed model is used on these local data to extract the most relevant features that led to the prediction result. Figures 12 (a) and 12 (b) show the output of the LIME explanation function on the Decoded-ViT and the CNN-LSTM model. The green color indicates the image's positive region, which has the highest value in prediction output. The red color shows the negative features, which, if absent or containing a low value, will have less impact on prediction results. Yellow acts as a boundary between negative and positive regions.



(a) ViT model prediction



(b) CNN-LSTM model prediction

Figure 12. LIME model explainer output for HDSR on DIDA dataset

The difference in prediction results of both methods indicates that when the dataset is small and accurate results are expected in less time, one can prefer the proposed Decoded-ViT model. The model's capacity to generalize across various writing styles is a positive finding. It functions well despite dealing with differences in how people write the exact string of numbers. This shows that the model's robustness results from its ability to capture essential features and its lack of sensitivity to minor variations in handwriting.

It's vital to remember that, like a human reader, the recognition system could have trouble understanding digits that are not accurately written. Adding more varied examples of unclear handwriting to the training set could help overcome this difficulty. Details of the model's practical use can be obtained by evaluating it on handwritten examples from real-world scenarios that differ from the training data.

A web app, Software as a Service (SaaS), delivers software where the model is available online. SaaS apps use an identical framework to service several users. Using an internet browser, users can use the application included in the proposed model without installing or updating it on their devices. The user needs to upload a handwritten digit string image, and the SaaS app will show the image's digitized output along with explanations for the results. Figure 13 shows the proposed app's output. The solid box indicates the current functionality of the SaaS app, and the dotted box shows the components that will be added in the future. The SaaS provider will update and implement new features, and users instantly receive the most recent version without doing anything.

Figure 13. Generalization and explainability of the proposed method for real-time use

5. SUMMARY

Different writing styles, irregular digit spacing, and possible overlaps make handwritten digit string recognition difficult. The research objective was to examine whether the vision transformer architecture can be used to recognize such handwritten digit strings. Hence, the vision transformer framework was altered to collect contextual dependencies and tokenized digit sequences. The proposed model showed impressive capacity to record long-term dependencies in handwritten digit patterns. It is essential to strike a balance to successfully represent digit sequences, as demonstrated by experiments with tokenization precision.

6. CONCLUSIONS

Recognizing handwritten digit strings from historical records is a challenging but crucial undertaking with many uses in digitizing and safeguarding historical material. Automating the process of identifying handwritten digit strings in ancient manuscripts, as proposed in this paper, has come a long way, thanks to developments in deep learning and computer vision techniques. As a result of the integration of the vision transformer and beam search decoder proposed in this paper, adequate recognition of digit strings is possible even in complex and deteriorated handwritten texts. The proposed method avoids locality inductive bias drawback and performs better on smaller datasets. The results show that the proposed architecture can better recognize handwritten digit strings from historical documents in less time. The proposed framework can deal with differences in writing style, ink color, and other flaws, typically in historical records. The proposed work also covers two main aspects of artificial intelligence: generalization and explainability. The model's availability in the form of SaaS made it worthwhile for different domain users. It will also help researchers to update the model as per user needs.

The study's outcomes contribute to theoretical knowledge, impact technology, and education policy decisions, and provide helpful advice for putting more efficient handwritten digit string recognition systems into practice. The importance comes from the possible game-changing effects in multiple industries where precise and context-aware digit string recognition is essential.

Despite these developments, there are still difficulties, mainly when working with severely damaged or unreadable papers, including smudged digits or overlapped numerals. Additional investigation and creativity are needed to increase the recognition models' resilience in these situations. Handwritten digit string recognition is anticipated to become ever more vital as technology develops, uncovering the depth of knowledge in ancient manuscripts and enhancing the understanding of history and cultural heritage. More comprehensive advancements in digitizing multilingual handwritten documents will result from future research. The robustness and generalizability of the model may be improved by enlarging the dataset to incorporate a broader range of handwriting styles, cultural variances, and demographic representations. Additional metrics like precision, recall, and F1-score should be investigated in future research to assess the model's performance in various classes and gain a deeper understanding of its behavior. To guarantee the model's dependability in real-world situations, further studies could

examine its sensitivity to hostile inputs and evaluate how well it performs in various settings.

REFERENCES

[1] Kusetogullari, H., Yavariabdi, A., Hall, J., Lavesson, N. (2020). DIDA: The largest historical handwritten digit dataset with 250k digits. https://github.com/didadataset/DIDA/, accessed on Jun. 13, 2021.

[2] Kusetogullari, H., Yavariabdi, A., Hall, J., Lavesson, N. (2021). DIGITNET: A deep handwritten digit detection and recognition method using a new historical handwritten digit dataset. Big Data Research, 23: 100182. https://doi.org/10.1016/j.bdr.2020.100182

[3] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778. https://doi.org/10.1109/CVPR.2016.90

[4] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700-4708. https://doi.org/10.1109/CVPR.2017.243

[5] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-9. https://doi.org/10.1109/CVPR.2015.7298594

[6] Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., Xiao, B. (2020). Deep high-resolution representation learning for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(10): 3349-3364. https://doi.org/10.1109/TPAMI.2020.2983686

[7] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv Preprint arXiv: 2010.11929. https://doi.org/10.48550/arXiv.2010.11929

[8] Neyshabur, B. (2020). Towards learning convolutions from scratch. Advances in Neural Information Processing Systems, 33: 8078-8088. https://doi.org/10.48550/arXiv.2007.13657

[9] Lee, S.H., Lee, S., Song, B.C. (2021). Vision transformer for small-size datasets. arXiv Preprint arXiv: 2112.13492. https://doi.org/10.48550/arXiv.2112.13492

[10] Shi, B., Bai, X., Yao, C. (2016). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(11): 2298-2304. https://doi.org/10.1109/TPAMI.2016.2646371

[11] Ribeiro, M.T., Singh, S., Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135-1144. https://doi.org/10.1145/2939672.2939778

[12] Ma, Y., Guo, J., Wei, W. (2019). An exceedingly fast model for low resolution handwritten digit string recognition. In 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT), pp. 282-288. https://doi.org/10.1109/ICCSNT47585.2019.8962475

[13] Oliveira, L.S., Lethelier, E., Bortolozzi, F., Sabourin, R. (2004). A new approach to segment handwritten digits. In EPRINTS-BOOK-TITLE.

[14] Ribas, F.C., Oliveira, L.S., Britto Jr, A.S., Sabourin, R. (2013). Handwritten digit segmentation: A comparative study. International Journal on Document Analysis and Recognition, 16(2): 127-137. https://doi.org/10.1007/s10032-012-0185-9

[15] Vellasques, E., Oliveira, L.S., Britto Jr, A.S., Koerich, A.L., Sabourin, R. (2008). Filtering segmentation cuts for digit string recognition. Pattern Recognition, 41(10): 3044-3053. https://doi.org/10.1016/j.patcog.2008.03.019

[16] Saabni, R. (2015). Ada-boosting extreme learning machines for handwritten digit and digit strings recognition. In 2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC), pp. 231-236. https://doi.org/10.1109/ICDIPC.2015.7323034

[17] Zhan, H., Wang, Q., Lu, Y. (2017). Handwritten digit string recognition by combination of residual network and RNN-CTC. In Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, Proceedings, Part VI. Springer International Publishing, 24: 583-591. https://doi.org/10.1007/978-3-319-70136-3_62

[18] Zhan, H., Lyu, S., Tu, X., Lu, Y. (2019). Residual CRNN and its application to handwritten digit string recognition. In Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, Proceedings, Part V. Springer International Publishing, 26: 49-56. https://doi.org/10.1007/978-3-030-36802-9_6

[19] Hochuli, A.G., Oliveira, L.S., Britto Jr, A.S., Sabourin, R. (2018). Handwritten digit segmentation: Is it still necessary? Pattern Recognition, 78: 1-11. https://doi.org/10.1016/j.patcog.2018.01.004

[20] Aly, S., Mohamed, A. (2019). Unknown-length handwritten numeral string recognition using cascade of PCA-SVMNet classifiers. IEEE Access, 7: 52024-52034. https://doi.org/10.1109/ACCESS.2019.2911851

[21] Neto, A.F.D.S., Bezerra, B.L.D., Lima, E.B., Toselli, A.H. (2020). HDSR-Flor: A robust end-to-end system to solve the handwritten digit string recognition problem in real complex scenarios. IEEE Access, 8: 208543-208553. https://doi.org/10.1109/ACCESS.2020.3039003

[22] Zhan, H., Lyu, S., Lu, Y., Pal, U. (2021). DenseNet-CTC: An end-to-end RNN-free architecture for context-free string recognition. Computer Vision and Image Understanding, 204: 103168. https://doi.org/10.1016/j.cviu.2021.103168

[23] Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.O. A next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2623-2631.

[24] Önler, E. (2022). Explaining image classification model (CNN-based) predictions with lime. World Journal of Advanced Engineering Technology and Sciences, 7: 275-280. https://doi.org/10.30574/wjaets.2022.7.2.0176