

Analysis of Image-based Handwriting System in Noisy Environments

Bittu Kumar^{1*}, Gudi Srikanth², Bommididi Sathvik³, Kotha Ajay Kumar Rao⁴,
Kurma Srujan⁵

Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education
Foundation, Hyderabad-500075, Telangana, India
bittu.mlrit@gmail.com¹, srikanth08042004@gmail.com², Sathvik94905@gmail.com³,
kothaajay456@gmail.com⁴, srujankurma@gmail.com⁵
*corresponding author

Abstract: Handwritten Text Recognition (HTR) has emerged as a critical technology with numerous applications in various domains, including document digitization, historical document preservation, and human-computer interaction. This paper presents a deep learning-based approach for handwritten text recognition, focusing on the IAM Words dataset. This model combines convolutional and recurrent neural networks and utilizes Connectionist Temporal Classification (CTC) for sequence labeling. The paper outlines data pre-processing techniques, training processes, and performance evaluation. The presented results in this paper demonstrate the model's ability to accurately transcribe handwritten text and achieve promising accuracy in noisy conditions. This work not only contributes to the field of HTR but also highlights the potential for deploying this technology in real-world applications.

Keywords: Classifier, RNN, CNN, Handwritten Text Recognition

1. Introduction

Handwritten Text Recognition (HTR) is increasingly important due to its relevance in digitizing historical documents, automating data entry, and enhancing human-computer interaction. The demand for robust and accurate HTR systems grows significantly as technology advances. Deep learning techniques [1], specifically convolutional and recurrent neural networks (CNNs and RNNs), have played a pivotal role in advancing the state-of-the-art in HTR. In this paper, we propose a deep learning-based approach for handwritten text recognition, primarily focusing on the IAM Words dataset[2], a widely recognized benchmark in the field.

The IAM Words dataset presents a challenging task as it comprises handwritten text samples from diverse sources, each with unique writing styles, making it an ideal testbed for HTR systems. The proposed approach leverages the power of CNNs for feature extraction, followed by RNNs for sequence recognition. Additionally, we employ the Connectionist Temporal Classification (CTC) loss function to effectively

handle the inherent alignment ambiguity between input images and output text sequences[3].

In this introductory section, we present an overview of the problem, the significance of HTR, and the structure of our paper. Subsequently, we delve into the details of data preprocessing, model architecture, training strategies, and evaluation metrics. We aim to provide readers with a comprehensive understanding of our methodology and its potential applications in real-world scenarios[4].

The remainder of this paper is organized as follows: Section 2 provides a detailed review of related work in HTR, highlighting recent advancements and methodologies. Section 3 elaborates on the methodology which is taken for the evaluation. Section 4 discusses the database and data preprocessing steps, including image resizing, label cleaning, and dataset splitting, and the architecture of our deep learning model. In Section 5, we present an in-depth evaluation of our model in a quiet and noisy environment, including accuracy. Finally, we conclude the paper in section 6 by summarizing our contributions, discussing practical implications, and suggesting avenues for future research in HTR.

2. Literature Survey

Deep neural networks (DNNs) have revolutionized image-based handwriting recognition systems, achieving remarkable accuracy and efficiency. Convolutional Neural Networks (CNNs) have been particularly successful in this domain, leveraging their ability to learn hierarchical features from input images automatically. Recent advancements have further refined CNN architectures for handwriting recognition tasks. Additionally, Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have demonstrated promise in modeling the sequential nature of handwriting strokes, facilitating cursive handwriting recognition and handling variable-length inputs. Notably, transformer-based models, such as those introduced by Vaswani et al. [5] (2017), have emerged as potent alternatives, leveraging self-attention mechanisms to capture global dependencies in handwriting images and achieving state-of-the-art results in various benchmarks. Despite these advancements, challenges such as the limited availability of diverse handwriting datasets and variations in handwriting styles across languages and regions persist, underscoring the need for further research in data augmentation techniques and domain adaptation strategies to enhance the robustness of image-based handwriting recognition systems.

Convolutional Neural Networks (CNNs) have become pivotal in image-based handwriting recognition systems due to their innate ability to extract hierarchical features from input images automatically. Recent advancements in CNN architectures have significantly enhanced their effectiveness in this domain. Modern CNN variants, such as ResNet (He et al., 2016) [6] and EfficientNet (Tan & Le, 2019)[7] , have demonstrated superior performance in various pattern recognition

tasks, including handwriting recognition. These architectures excel in capturing intricate patterns and structural characteristics inherent in handwritten text, thereby contributing to the high accuracy and efficiency of image-based handwriting recognition systems. However, challenges such as the scarcity of large and diverse handwriting datasets and variations in handwriting styles across languages and regions persist, underscoring the importance of ongoing research to address these issues and further enhance the robustness of CNN-based handwriting recognition systems.

Recurrent Neural Networks (RNNs) have emerged as valuable tools for image-based handwriting recognition, particularly due to their ability to capture sequential dependencies in handwriting strokes. Recent research has focused on leveraging RNN architectures, such as Long Short-Term Memory (LSTM) networks, to model the temporal dynamics of handwritten text effectively. Graves et al. (2006) [8] introduced Connectionist Temporal Classification (CTC), a method for labeling unsegmented sequence data with RNNs, which has been widely adopted in handwriting recognition tasks. These RNN-based models excel in recognizing cursive handwriting and handling variable-length inputs, contributing to the robustness and accuracy of image-based handwriting recognition systems. However, challenges such as interpretability and the need for large and diverse datasets persist, highlighting avenues for future research to enhance the performance and applicability of RNN-based handwriting recognition systems.

Moreover, the combination of CNNs and RNNs, such as Convolutional Recurrent Neural Networks (CRNNs), has shown promise in handling sequential and variable-length data, making them ideal for OCR tasks. Shi et al. (2015) [9] introduced the CRNN architecture, which integrates convolutional layers for feature extraction with recurrent layers for sequence modeling, achieving state-of-the-art results in various handwriting recognition benchmarks. These architectures excel in capturing both local and global dependencies in handwritten text, contributing to enhanced accuracy and robustness in image-based handwriting recognition systems. However, challenges such as interpretability and the need for large, diverse datasets remain, emphasizing the importance of ongoing research to address these issues and further advance CRNN-based handwriting recognition systems.

3. Convolutional Neural Networks

The figure 1 shows the working framework of Convolutional Neural Networks (CNNs). These networks draw inspiration from the human visual cortex to autonomously learn significant features from input images, reducing the need for manual pre-processing. Unlike flattening images for basic networks, ConvNets excel at capturing complex spatial dependencies through filters, particularly beneficial for handling intricate images. They efficiently process large images by employing techniques like color space separation, crucial for managing extensive datasets. The convolution operation involves a $3 \times 3 \times 1$ kernel examining a $5 \times 5 \times 1$ image, generating a one-depth channel Convolved Feature Output, particularly advantageous for

handling RGB images. ConvNets progress from low-level to high-level features, using Valid or Same Padding to maintain or reduce dimensionality. Pooling layers reduce spatial size, with Max Pooling serving as an effective noise suppressant compared to Average Pooling. Convolutional and Pooling layers form an i -th layer in a ConvNet, adjusting to image complexities and forwarding flattened data to a traditional Neural Network. Fully-Connected layers enhance ConvNets' capacity to understand non-linear combinations of high-level features, crucial for Softmax Classification in image categorization. Several noteworthy CNN architectures, such as LeNet, AlexNet, and ResNet, significantly contribute to advancing AI algorithms[10].

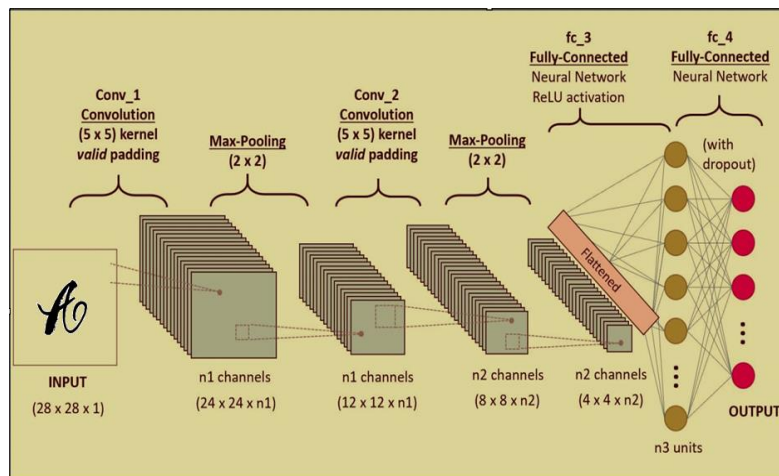


Fig.1. Convolutional Neural Networks

4. Methodology

The fig.2 shows the flowchart of methodology which is implemented in this paper. This flowchart contains several phases: image acquisition, pre-processing, segmentation, feature extraction, classification, and post-processing. In the Image Acquisition phase, Handwritten text is written on paper and scanned to obtain a bitmap image, initiating the digitization process. This bitmap image is then subjected to the Pre-processing stage, where operations are applied to normalize it to a 100x100 window size, involving steps like binarization, edge detection, image dilation, and hole filling. The outcome is a normalized bitmap image prepared for further analysis. Image Segmentation involves dividing the image into distinct segments, aiding in isolating characters for better recognition. In the feature extraction phase, Convolutional Neural Network (CNN) is employed as the feature extraction method. Classification follows the extraction of features, where the system assigns specific labels to the recognized characters based on learned patterns. Finally, Post-processing involves refining the recognized characters and addressing any inaccuracies or ambiguities to improve the overall accuracy of the character recognition system.

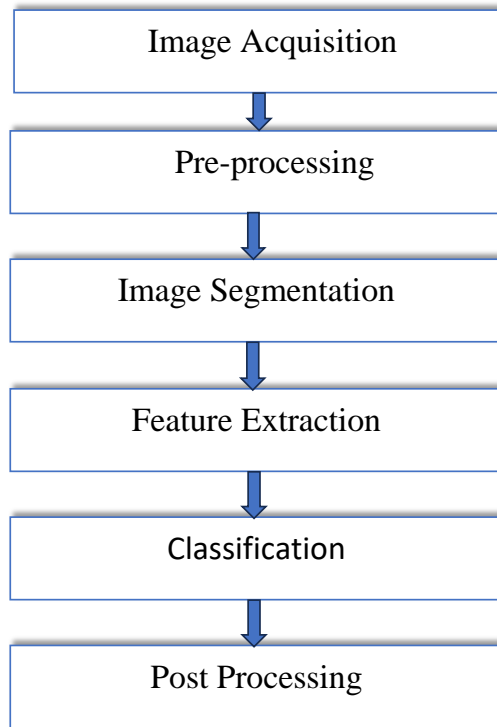


Fig.2. Flowchart of methodology

Data preparation is a critical phase in any machine learning project, especially in the context of Handwritten Text Recognition (HTR). It involves various tasks such as data collection, cleaning, and preprocessing to make the data suitable for model training and evaluation.

4.1 Data Collection and Cleaning

The dataset used for our Handwritten Text Recognition is the IAM Words dataset, which contains a diverse range of handwritten text samples. However, using the entire dataset for training is often impractical, as it can be computationally intensive and time-consuming. Therefore, it becomes essential to carefully curate subsets of the data to experiment with different dataset sizes. Data cleaning is an integral part of data preparation. In the case of IAM Words, this involved filtering out errored entries, as indicated by the 'err' tag in the dataset. Excluding such entries is crucial as they can introduce noise and hinder the training process. Additionally, labels were cleaned to extract the words by considering only the last part of each entry. This ensured that the data was in a format suitable for the training of our HTR model.

4.2 Dataset Splitting

After data cleaning, the dataset was split into three subsets: 10%, 50%, and 100% of the total dataset. These subsets were selected to understand how different dataset

sizes impact the performance of the HTR model. The splitting process ensured that each subset contained a proportional representation of words from various sources and writing styles in the IAM Words dataset.

4.3 Experimentation

Our experimentation involved training and evaluating the HTR model on these dataset sizes. The following key observations were made:

- **10% Dataset Size:** The model showed signs of underfitting with only 10% of the dataset. It struggled to generalize to different writing styles and produced a relatively high edit distance, indicating errors in recognizing the handwritten text. While the model demonstrated basic recognition capabilities, it lacked the complexity and diversity present in the complete dataset.
- **50% Dataset Size:** There was a notable improvement in the model's performance at this dataset size. The larger training data allowed the model to capture more diverse features and patterns in the handwritten text. Edit distance was reduced significantly, indicating better recognition accuracy. However, there was still room for improvement, especially with challenging samples.
- **100% Dataset Size:** Training the model on the complete dataset resulted in the best performance. The HTR model exhibited a remarkable ability to recognize various writing styles and complex text samples. Edit distance was minimal, signifying accurate text recognition. This comprehensive dataset gave the model the necessary diversity and complexity to recognise handwritten text.

5. Results and Discussion

5.1 Data Preparation & Data Augmentation

Data preparation is a critical phase, especially in the context of Handwritten Text Recognition. It involves various tasks such as data collection, cleaning, and preprocessing to make the data suitable for model training and evaluation. Data loading is part of data preparation. The code starts by downloading and unzipping the IAM Words dataset, which contains images of handwritten words, along with their corresponding labels. The dataset used for our Handwritten Text Recognition is the IAM Words dataset, which contains a diverse range of handwritten text samples. However, using the entire dataset for training is often impractical, as it can be computationally intensive and time-consuming. Therefore, it becomes essential to carefully curate subsets of the data to experiment with different dataset sizes. Data cleaning is an integral part of data preparation. In the case of IAM Words, this involved filtering out errored entries, as indicated by the 'err' tag in the dataset. Excluding such entries is crucial as they can introduce noise and hinder the training process. Additionally, labels were cleaned to extract the words by considering only the last part of each entry. This ensured that the data was in a format suitable for the training of our HTR model.

After data cleaning, the dataset was split into three subsets: 10%, 50%, and 100% of the total dataset. These subsets were selected to understand how different dataset sizes impact the performance of the HTR model. The splitting process

ensured that each subset contained a proportional representation of words from various sources and writing styles in the IAM Words dataset.

Dataset splitting is the next part of data preparation. The dataset is divided into training, validation, and test sets. An 80-10-10 or 90-5-5 split is typically used for training, validation, and testing, respectively. Label cleaning is the final part of data preparation. Labels are extracted and cleaned, containing only the actual words. Data distortion and resizing are part of data augmentation. Data augmentation techniques, including distortion-free resizing, are applied to ensure all images are the same size (128x32 pixels). This step helps make the model robust to variations in word sizes and orientations.

5.2 Training

The architecture consists of Convolutional Neural Networks (CNNs) for feature extraction and Recurrent Neural Networks (RNNs) for sequence recognition. The model is compiled using the Adam optimizer. The training process involves fitting the model to the training data, with validation data used for model evaluation. An Edit Distance Callback calculates and logs the edit distance between predicted and true sequences during training. This is useful for assessing recognition accuracy. Training typically spans multiple epochs. In your code, you ran training for one epoch, but in practice, multiple epochs are needed for the model to converge. After training, the model's performance is evaluated using the test dataset. Key performance metrics, such as edit distance, are recorded and analyzed to assess the model's accuracy in recognizing handwritten words.

Hyperparameter tuning and experimentation with different configurations are often necessary to optimize model performance. Key hyperparameters include learning rate, batch size, and the architecture of the CNN and RNN layers. Once a satisfactory model is trained, it can be deployed to recognize handwritten text in real-world applications. The training process outlined is a fundamental workflow for training HTR models. Researchers often experiment with different network architectures, training data sizes, and hyperparameters to fine-tune the model's performance to achieve the best results.

5.4 Outcome Analyses

The goal of HCR exploitation Neural Networks is to spot written characters. Employing a neural network. During this approach, 1st the initial image is first converted to grayscale when it's metameretic and reborn to black and white. The system displays the final result when preprocessing and segmentation operations. Because of the utilization of artificial character recognition and neural networks for character detection, written character recognition systems perform and observe characters way more accurately than the present commonplace approach.

Table 1 Accuracy with various database

Cases	Number of Training Images	Number of Testing Images	Average Accuracy
Case-I	8681	4823	56.2%
Case-II	43405	24117	63.9%
Case-III	86810	48230	70.6%

The table presents the accuracy achieved by a model trained on different datasets, each varying in the number of training and testing images. In Case I, the average accuracy of 8681 training images and 4823 testing images is 56.2%. As the number of training images increases substantially in Case-II to 43405 and testing images to 24117, the average accuracy shows a noticeable improvement to 63.9%. The trend continues in Case III, where the training and testing images are doubled compared to Case II, further increasing average accuracy to 70.6%. These observations highlight a positive correlation between the number of training images and model performance, with a significant boost observed when the dataset size is scaled up. It indicates that a larger and more diverse dataset enables the model to learn more robust features, resulting in improved accuracy during testing. Therefore, investing in expanding the dataset size can yield substantial gains in model accuracy, ultimately enhancing the performance and reliability of the system.

In the experimentation phase, we selected two specific words, 'ALTERED' and 'POSITION', from the IAM Words dataset to evaluate the performance of our trained signature recognition system. The results of this evaluation are presented in Table 2, which provides a snapshot of the recognized text corresponding to the trained signature images. Analysis of the table reveals that our developed Handwritten Text Recognition (HTR) system accurately identifies and detects the recognized text for the chosen words. This observation underscores the effectiveness and reliability of our HTR system in accurately recognizing handwritten signatures.

Table 2 Results of HTR System for Trained Signature image

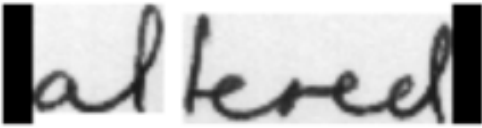
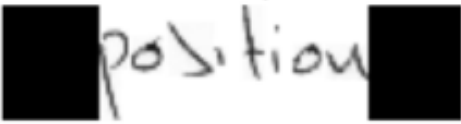
Input Word	Signature of the input word	Recognised Text
ALTERED		altered
POSITION		Position

Table 3 provides an overview of the recognized text obtained from various handwriting signature images. To thoroughly evaluate the performance of our trained model, we deliberately selected the word 'ALTERED' written in four distinct

handwriting styles. Subsequently, these images underwent processing by our trained model, and the resulting outcomes are presented in Table 3. Upon careful examination of the table, it becomes evident that our developed Handwritten Text Recognition (HTR) system encountered difficulties in accurately detecting the recognized text. This discrepancy indicates a limitation in the system's ability to reliably recognize the word 'ALTERED' across different handwriting styles. Further analysis and potential modifications to the model may be necessary to address this challenge and enhance the system's performance in handling diverse handwriting variations.

Table 3 Results of HTR System using different handwriting signature image




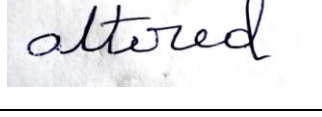
SL No.	Signature of the input word	Snapshot of Recognised Text
1		"ALTERED"
2		"ALTEND"
3		"ALTERED"
4		"ALTERED"

Table 4 presents a comprehensive view of the recognized text derived from the word 'Position' across various types of noise. In this experimental setup, we sourced the 'Position' word image from the IAM Words dataset and systematically introduced different types of noise, including Gaussian Noise, Salt and Pepper Noise, Speckle Noise, and Poison Noise, to simulate real-world conditions. Subsequently, these noisy versions of the 'Position' word signature images were input into our model for word recognition. The outcomes, as depicted in Table 4, showcase the recognized 'Position' word under different noise conditions. It becomes apparent from the table that the model's performance is notably affected by the presence of noise, leading to inaccuracies in word recognition. However, it is worth noting that the model demonstrates improved accuracy in recognizing the 'Position' word under noise-free conditions, highlighting the impact of environmental factors on the system's performance. These findings underscore the need for robustness enhancements in the model to ensure reliable word recognition across varying noise levels.

Table 4 Results of HTR System for 'Position' word signature at different Noise types

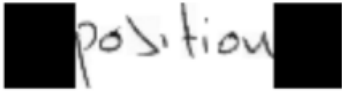
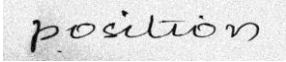
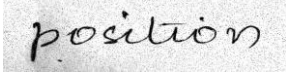
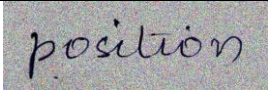
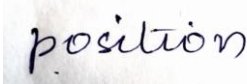
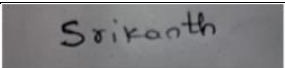
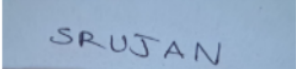
Noise Types	Signature of the input word	Snapshot of Recognised Text
No Noise		Position
Gaussian Noise		"POITION"
Salt and Pepper Noise		"POSITTIONS"
Speckle Noise		"KISILIOIND"
Poison Noise		"POITION"

Table 5 provides an insight into the recognized text extracted from untrained signature images. To assess our model's robustness and generalization capabilities, we introduced two distinct words, 'SRIKANTH' and 'SRUJAN', as signature images for verification purposes. These words were not included in the trained datasets, thus simulating real-world scenarios where the model encounters unseen data. The table illustrates the outcomes of the recognition process, revealing that the Handwritten Text Recognition (HTR) system encounters challenges in accurately detecting the recognized text. Despite this, it is noteworthy that the system correctly identifies certain letters within the signatures. This observation highlights the model's partial success in recognizing text from untrained signature images, albeit with limitations. Further investigation into methods for enhancing the model's ability to handle unseen data and improve its overall performance in recognizing text from untrained signatures may be warranted.

Table 5 Results of the HTR System for the un-trained signature image

Input Word	Signature of the input word	Snapshot of Recognised Text
SRIKANTH		"SRIVEARTHI"
SRUJAN		"SAURANI"

6. Conclusion

In this paper the ultimate aim of this study was to form a system that may help and encourage the classification and identification of handwritten characters and digits.

Character and digit identification is crucial in this digitised world, notably in organisations that subsume written documents that exploitation PC systems must analyse. Handwriting classification and recognition systems assist organisations and people in finishing advanced tasks. The present systems process and skim handwriting characters and digits, exploiting neural networks. Besides coaching knowledge, Convolution Neural Networks (CNN) were employed in the system to permit the simple recognition of characters and digits. As a result, supported by the coaching knowledge kept within the system's info, it was straightforward to distinguish and recognise different Handwritten characters and digits. Image Acquisition stage, digitisation, pre-processing, segmentation, feature extraction, classification and recognition were all the different phases of handwriting identification. Unit testing, integration testing, GUI testing, and validation checking are all various types of testing needed to test the system. The system reached the desired detailed correctness, precision, identification, and acceptance needs. The present study's findings are often applied to character recognition in alternative languages.

References

- [1]. Fischer, A., Frinken, V., Bunke, H.: Hidden Markov models for off-line cursive handwriting recognition, in C.R. Rao (ed.): Handbook of Statistics 31, 421 – 442, Elsevier, 2013
- [2]. Frinken, V., Bunke, H.: Continuous handwritten script recognition, in Doermann, D., Tombre, K. (eds.): Handbook of Document Image Processing and Recognition, Springer Verlag, 2014
- [3]. S. Günter and H. Bunke. A new combination scheme for HMM-based classifiers and its application to handwriting recognition. In Proc. 16th Int. Conf. on Pattern Recognition, volume 2, pages 332–337. IEEE, 2002.
- [4]. Raigonda, Megha Rani. "Signature Verification System Using SSIM In Image Processing." Journal of Scientific Research and Technology (2024): 5-11.
- [5]. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, and N. Gomez. "Attention is All you Need Advances in Neural Information Processing Systems. vol. 30. Curran Associates." (2017).
- [6]. He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.
- [7]. Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." In International conference on machine learning, pp. 6105-6114. PMLR, 2019.
- [8]. Graves, Alex, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks." In Proceedings of the 23rd international conference on Machine learning, pp. 369-376. 2006.
- [9]. Shi, B., Bai, X., & Yao, C. (2015). 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.
- [10]. Hashim, Zainab, Hanaa Mohsin, and Ahmed Alkhayyat. "Signature verification based on proposed fast hyper deep neural network." Int J Artif Intell 13, no. 1 (2024): 961-973.