International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 2, April 2025, pp. 2291~2303
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i2.pp2291-2303

Handwritten text recognition system using Raspberry Pi with OpenCV TensorFlow

Jamil Abedalrahim Jamil Alsayaydeh[1], Tommy Lee Chuin Jie[1], Rex Bacarra[2], Benny Ogunshola[3], Noorayisahbe Mohd Yaacob[4]
[1]Department of Engineering Technology, Fakulti Teknologi dan Kejuruteraan Elektronik and Komputer, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia
[2]Department of General Education and Foundation, Rabadan Academy, Abu Dhabi, United Arab Emirates
[3]Department of Electrical and Telecommunications Engineering, Frontier institute of technology, Glenroy City, Australia
[4]Center for Software Technology and Management (SOFTAM), Faculty of Information Science and Technology, University Kebangsaan Malaysia, Selangor, Malaysia

Article Info

Article history:

Received Mar 16, 2024
Revised Oct 24, 2024
Accepted Nov 20, 2024

Keywords:

Convolutional neural networks
Deep learning
Handwritten text recognition
Real-time analysis
Recognition accuracy

ABSTRACT

Handwritten text recognition (HTR) technology has brought about a revolution in the way handwritten data is converted and analyzed. This proposed work focuses on developing a HTR system using deep learning through advanced deep learning architecture and techniques. The aim is to create a model for real-time analysis and detection of handwritten texts. The proposed deep learning architecture that is convolutional neural networks (CNNs), is investigated and implemented with tools like OpenCV and TensorFlow. The model is trained on large handwritten datasets to enhance recognition accuracy. The system's performance is evaluated based on accuracy, precision, real-time capabilities, and potential for deployment on platforms like Raspberry Pi. The actual outcome is a robust HTR system that can convert handwritten text to digital formats accurately. The developed system has achieved a high accuracy rate of 91.58% in recognizing English alphabets and digits and outperformed other models with 81.77% mAP, 78.85% precision, 79.32% recall, 79.46% F1-Score, and 82.4% receiver operating characteristic (ROC). This research contributes to the advancement of HTR technology by enhancing its precision and utility.

Corresponding Author:

Jamil Abedalrahim Jamil Alsayaydeh
Department of Engineering Technology, Fakulti Teknologi dan Kejuruteraan Elektronik and Komputer,
Universiti Teknikal Malaysia Melaka
76100 Durian Tunggal, Melaka, Malaysia
Email: jamil@utem.edu.my

1. INTRODUCTION

Handwritten text recognition (HTR) technology, powered by deep learning and artificial intelligence, has revolutionized the decoding and analysis of handwritten data [1]. It employs optical character recognition (OCR), neural networks, and deep learning algorithms to accurately identify various handwriting styles [2]. The technology has wide-ranging applications across industries such as historical document digitization, postal services, forms processing, personal note-taking, and finance, transforming traditional methods of processing handwritten data by enhancing efficiency, precision, and accessibility. This work uses a neural network approach to develop an HTR system that processes and analyzes handwritten data. Neural networks are trained to recognize patterns and make accurate predictions. The goal is to create an HTR system that accurately recognizes and converts handwritten text into digital formats [3]. The system uses dedicated hardware for capturing and processing handwritten input, which aids in efficient text recognition. This technology represents a significant advancement in the field of handwritten data processing.

The process of converting handwritten text into machine-readable format is challenging due to the wide range of individual handwriting styles and issues like low resolution, poor contrast, or scanning distortions. Even with advancements in computer vision and machine learning, there is a need for a system that can accurately recognize handwritten English alphabets and numerals. The proposed work aims to address the challenges of HTR by using a convolutional neural network (CNN) to identify and categorize patterns in handwritten characters. The goal is to create a system that can classify handwritten input into corresponding characters or suggest the closest match when an exact one is not possible. The system's performance will be assessed based on accuracy, precision, and recall. This system could enhance the accuracy and efficiency of HTR, aiding in the conversion of handwritten text into machine-readable format.

This literature review focuses on the latest techniques, architectures, and methodologies used in handwritten recognition systems, with a specific investigation into the capabilities of Raspberry Pi as a hardware platform and the use of OpenCV and TensorFlow for deep learning, image processing, and feature extraction. In study [4], in the HTR process, optical character recognition is a crucial step that transforms images of text into machine-encoded text, with significant advancements over the years contributing to improved accuracy and speed. In study [5], real-time character recognition and accuracy percentage are key aspects in OCR, crucial for authenticating users as genuine individuals, but challenges such as handling different fonts and styles, poor image quality, and skewed or rotated text pose significant hurdles [6], [7].

Several techniques have been proposed for HTR systems, with a typical workflow involving preprocessing, feature extraction, segmentation, classification, and recognition; an example is a system developed using convolutional and recurrent neural networks, trained on the IAM dataset, which achieved 84.5% accuracy. In study [8] and another research [9] applied CNN and support vector machine (SVM) algorithms for handwriting recognition, trained and evaluated on the extended MNIST (EMNIST) dataset, achieving an overall system accuracy of 85.41%. Ghosh and Kristensson [10] deployed the text detection and recognition model, combining the efficient and accurate scene text (EAST) model for text detection and the Tesseract-OCR engine for text recognition, trained on the international conference on document analysis and recognition database, achieving a precision rate of 88.69%. The accuracy of HTR systems is influenced by intrinsic and extrinsic factors [11]. Intrinsic factors include the variability in handwriting styles, quality, and the writer's age or health [12]. Elements such as letter shapes, slants, sizes, spacing, stroke variations, and writing speed contribute to the complexity of recognition [13]. Poorly written or illegible text can lead to errors. Extrinsic factors involve partial occlusion, lighting conditions, paper or writing instrument quality, writing or scanning angle, and image resolution [14], [15].

These factors can affect the system's performance by making it challenging to train models that generalize well and by degrading image quality. Researchers address these issues through advanced preprocessing, data augmentation, and sophisticated machine learning algorithms to improve recognition accuracy. According to reference [16], an effective HTR system should be compatible with scanned documents and images, support real-time processing, be robust to handwriting style variations, independent of text language or script, and capable of handling text from different orientations. The proposed system has three primary steps: text localization, feature extraction, and classification. In text localization, scanned documents are segmented into word images. Feature extraction involves capturing pixel density, pixel intensity variance, word mean, standard deviation of pixel intensities, upper quarter region intensity, and Otsu's threshold [17]. These features are processed using a SVM classifier to distinguish between different handwritten texts. These characteristics make CNNs a powerful tool for improving the accuracy and effectiveness of HTR systems [18]. The implementation of a HTR system focuses on the EMNIST dataset which serves as the foundation for training and testing the HTR model. CNN architecture is employed for its ability to capture intricate features in handwritten text. OpenCV and TensorFlow play pivotal roles in pre-processing images and training the deep learning model, respectively. The integration of OpenCV ensures effective image manipulation and feature extraction, while TensorFlow facilitates the training of the CNN for accurate text recognition. For real-world deployment, the system is adapted to run on a Raspberry Pi 4B, a compact and affordable embedded platform. The Raspberry Pi Camera Module 3 is employed for capturing images in real-time, enabling the system to recognize handwritten text from physical documents or surfaces. In this work, the CNN model has demonstrated its effectiveness by outdoing all other models in every measurement. Although the SVM and connectionist temporal classification (CTC) models exhibited impressive performance in some respects, they fell short of the CNN. The recurrent neural network (RNN), despite its overall lower performance, excelled in terms of the F1-score. These outcomes reinforce the robustness and preeminence of the CNN model.

2. METHOD

This section provides detailed information on the chosen deep learning technique, CNN, for HTR and detection. It also offers a comprehensive description of the model training process using the EMNIST dataset, which contains a large number of handwritten characters and digits. The section outlines the process and methodology involved in constructing the HTR system. This methodology is the result of extensive research conducted through literature reviews and readings to identify the most effective deep learning methods and strategies. The primary objective of this methodology is to develop an HTR system that can accurately recognize and interpret handwritten text.

2.1. Block diagram

The block diagram for the system includes Figure 1 a camera module, a Raspberry Pi, a power supply, and an LCD monitor. The camera module, specifically the Raspberry Pi Camera Module 3, serves as the input device, capturing images of the handwritten text. The Raspberry Pi acts as the central processing unit, responsible for running the HTR system. It handles tasks such as image preprocessing, loading, and utilizing the trained CNN model, and performing OCR on the captured images. The power supply ensures that the Raspberry Pi and the camera module receive the required electrical power to operate efficiently. Lastly, the LCD monitor serves as the output device, displaying the processed images and recognized text output to the user.

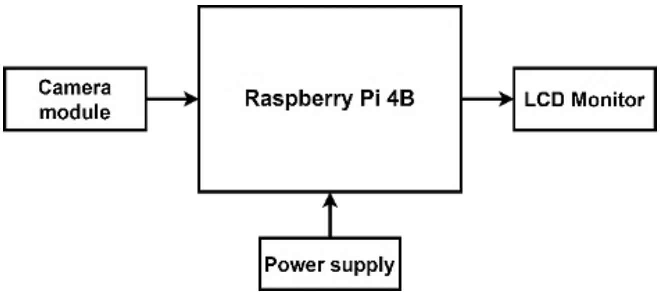


Figure 1. Block diagram of HTR system

2.2. HTR system flow

The system flow for the HTR deployed on a Raspberry Pi 4B with a camera as its input and the output shown on a monitor follows a series of steps, as illustrated in Figure 2. First, the Raspberry Pi 4B is initialized, and the required dependencies, including OpenCV, TensorFlow, and a CNN classifier, are imported. Next, an image of the handwritten text is captured using the Raspberry Pi Camera Module 3. The captured image is then pre-processed using OpenCV to enhance its quality. Once pre-processed, a pre-trained CNN model, trained on handwritten text data using TensorFlow, is loaded. This model is used to perform OCR on the pre-processed image, generating predictions for each character in the text. The results are then post-processed, which may include removing noise, correcting spelling errors, or formatting the recognized text. Finally, the processed image and the recognized text output are displayed on the monitor connected to the Raspberry Pi 4B, completing the system flow.

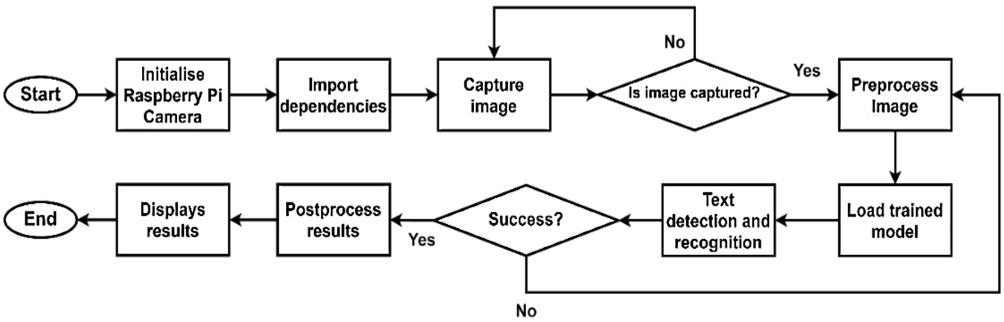


Figure 2. System operating flowchart

2.2.1. System programming flowchart

The proposed HTR system follows a programming flowchart in Figure 3 to process and recognize handwritten text. The process begins with capturing an image using the camera module, which is then sent to the Raspberry Pi 4B for further processing. The system performs text detection on the captured image. Next, the system extracts feature from the region of interest (ROI) to focus on relevant information for recognition. The text recognition process utilizes a CNN as the foundation, where the extracted features are fed into the model for recognition. If the captured text matches any of the texts from the dataset, the system proceeds to identify and verify the identity of the text. However, if the captured text does not match any known texts from the dataset, it is considered an unknown text, leading the program to terminate. This programming flow demonstrates the sequential steps involved in the HTR system, including image capture, text detection, feature extraction, text recognition using CNN, matching and verification, and handling of unknown texts.

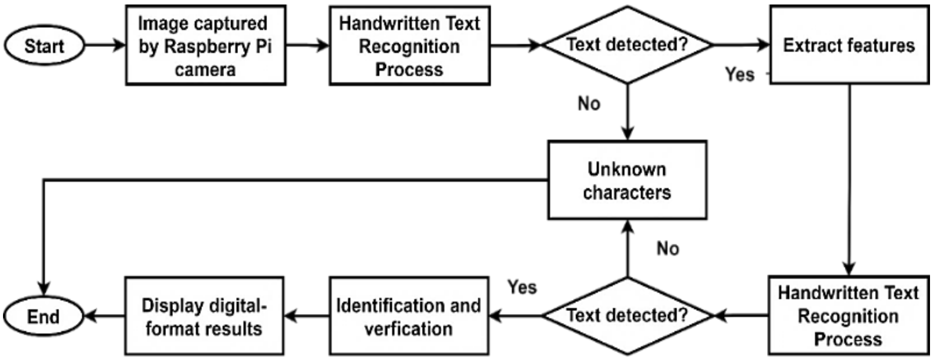


Figure 3. System programming flowchart

2.3. Data acquisition

The EMNIST dataset was procured and utilized in this work for training and evaluation purposes. The EMNIST dataset, an extension of the widely used MNIST dataset, is a popular benchmark dataset in machine learning and computer vision. It comprises a comprehensive collection of handwritten digits and characters, with a training set of 630,000 examples and a test set of 70,000 examples. The acquisition of the EMNIST dataset typically involves accessing it from public repositories or through frameworks like TensorFlow. The dataset is pre-divided into training and test sets, ensuring a standardized evaluation process. The training set is used to train the deep learning model, while the test set is employed to evaluate the model's performance on unseen data. Each sample in the EMNIST dataset is a grayscale image of 28×28 pixels, representing a single digit from 0 to 9 or a letter from A to Z. The images have been meticulously labeled with their corresponding digit or character class, making them suitable for supervised learning tasks such as character recognition. By acquiring and utilizing the EMNIST dataset in this work, the deep learning model can be trained and evaluated using a comprehensive and reliable benchmark dataset, enabling the development of an accurate and robust HTR system in Figure 4.

Figure 4. EMNIST dataset

2.4. Deep learning classifiers

Deep learning classifiers, specifically CNN, are a powerful classification-oriented machine learning model. CNNs are used in this study to classify predictions as either accurate or inaccurate. CNNs have achieved innovative performance in various domains, including computer vision, natural language processing, and speech recognition [19]. These networks are designed to automatically learn and extract intricate patterns and characteristics from input data, making them particularly useful for tasks involving complex visual or textual data. This work uses CNN as the DL classifier to classify correct and incorrect predictions, enabling robust and reliable classification outcomes [20].

2.4.1. Classification using CNN

The proposed system uses a CNN to interpret handwritten text. The CNN model is trained on a dataset of handwritten samples, learning patterns in characters and words. Convolutional and pooling layers extract features from images for classification. The model adjusts its parameters during training to improve accuracy. This technology can process and recognize handwritten text in real-time, converting it into digital format [21].

2.5. Algorithm training

This work uses OpenCV, TensorFlow, and CNN classifiers to train a HTR algorithm. The dataset comprises annotated handwritten text samples. The CNN architecture, with convolutional and pooling layers, extracts feature from input images. TensorFlow is used to build and train the CNN model. The training process iterates over the dataset, adjusting the model’s parameters to minimize prediction errors. The Python training script handles dataset processing and model weight updates. The number of training epochs is optimized experimentally. The trained CNN model can then recognize new handwritten text images, providing capabilities for data digitization, transcription, and document analysis.

2.6. Raspberry Pi 4B

The Raspberry Pi 4B, equipped with a Quad-core ARM Cortex-A72 processor and 4 GB of RAM, is ideal for implementing a HTR model Figure 5. It features the Broadcom BCM2711 chip and has a 1.5 GHz clock speed. It includes 32 KB L1 cache for data, 48 KB L1 cache for instructions, and 1 MB L2 cache for faster data retrieval. The board offers various ports, including 4 USB ports, an HDMI port, and an SD card slot, allowing seamless connectivity with external devices. It operates on Raspberry Pi OS, which supports high-level programming languages and offers a range of tools and libraries for easy integration with hardware components. Its compact size and capabilities make it an excellent choice for deploying systems in resource-constrained environments [22].

Figure 5. Raspberry Pi 4B

2.7. Raspberry Pi Camera Module 3

The Raspberry Pi Camera Module 3, designed for Raspberry Pi boards, features a Sony IMX708 sensor with 11.9 megapixels and HDR imaging in Figure 6. Available in standard and wide-angle variants, it includes advanced features like phase detection autofocus and 2D dynamic defect pixel correction. It uses a CSI-2 serial interface for data output and is compatible with all Raspberry Pi models. The module measures 25×24×11.5 mm and comes with a 200 mm ribbon cable [22].

Figure 6. Raspberry Pi Camera Module 3

2.8. OpenCV

OpenCV is an open-source library for computer vision and image processing in Figure 7. It provides tools for operations like object detection, image enhancement, and geometric transformations [23], [24]. It supports features such as image processing, object detection, feature extraction, and integration with machine learning frameworks. OpenCV is used in diverse fields like robotics, medical imaging, and self-driving cars, offering a powerful toolkit for visual data and computer vision challenges [25], [26].

Figure 7. OpenCV

2.9. TensorFlow

TensorFlow is an open-source software widely used in machine learning in Figure 8. It simplifies building, training, and deploying models with its high-level API, Keras [27]. It supports distributed computing for faster training and offers pre-built algorithms for tasks like image classification. TensorFlow excels at handling large-scale datasets and complex models, and provides tools for model visualization, debugging, and deployment. Its comprehensive features and strong community support make it a robust framework for machine learning tasks [28].

Figure 8. TensorFlow

3. RESULTS AND DISCUSSION

The outcomes and analysis obtained from the creation of a HTR system, utilizing CNN, are detailed in this section. The system underwent extensive training and evaluation using the EMNIST dataset, which comprises a substantial compilation of handwritten numerals and characters. The process involved training the CNN model on a considerable collection of 630,000 images obtained from the EMNIST dataset. Following the training phase, an additional set of 70,000 images was utilized to evaluate the model's performance. The outcomes were verified using essential metrics such as recall, accuracy, and precision of the HTR system. Additionally, a confusion matrix was employed to provide a comprehensive view of the system's performance, revealing its merits and opportunities for enhancement.

3.1. Essential interpretation based on key findings

The HTR system uses a Raspberry Pi Camera to capture images of handwritten text. These images are processed by a Raspberry Pi 4, which runs the HTR system utilizing a CNN model. The recognized text and relevant metrics are displayed on a monitor connected to the Raspberry Pi 4, as shown in Figure 9. The CNN model demonstrated impressive performance, achieving an accuracy of 91.58% in recognizing English alphabets and digits.

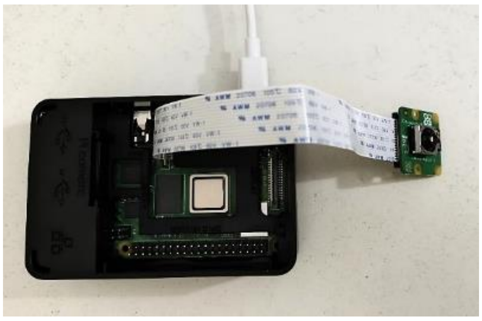


Figure 9. Hardware setup

3.2. Confusion matrix and analysis

The datasets in the test folders were evaluated to determine the accuracy of identifying gesture data according to their classes while classifying the ten classes of digits. The accuracy was calculated using the confusion matrix displayed after the coding. The accuracy can be calculated using (1).

$$Accuracy = \frac{TP}{TP+TN+FP+FN} \times 100\% \quad (1)$$

The term "true positive" describes how the positive class is accurately classified, while "true negative" refers to the accurate classification of the negative class. A false positive occurs when a negative class is incorrectly forecasted as positive, and a false negative occurs when a positive class is incorrectly predicted as negative. The confusion matrix Figure 10 reveals which numbers the model correctly recognizes and which ones it frequently mistakes. Although the model performs admirably, it occasionally (5 times out of 10,000) confuses number 5 with number 3 or number 2 with number 7. With 91.58% accuracy, the CNN algorithm is quite accurate. Given that the epochs trained were 100 with classifications based on digits 0 to 9. This result is considered excellent, as accuracy measurements between 75% and 95% are practical and ideal in practice.

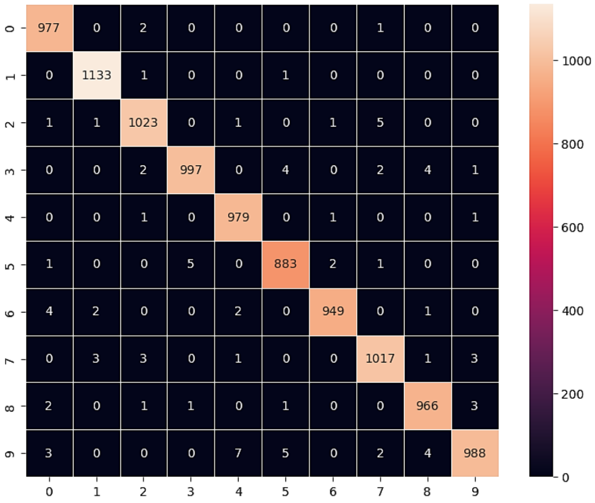


Figure 10. CNN confusion matrix after one hundred epochs

3.3. Results on the handwritten recognition system

Training with TensorFlow was conducted across several epochs, and the epoch that produced the highest level of accuracy was considered. The first picture generated from the test set is shown in Figure 11 and Figure 12, demonstrating whether the model's prediction was accurate. The model predicted correctly, identifying digit seven. Subsequent test instances were predicted and forecasted to observe the model's performance and errors.

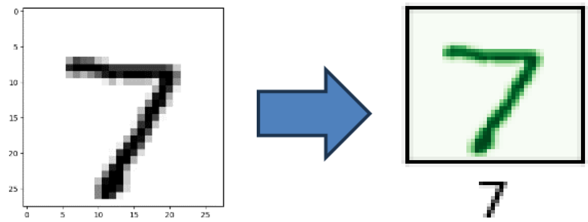


Figure 11. Prediciton output from TensorFlow model

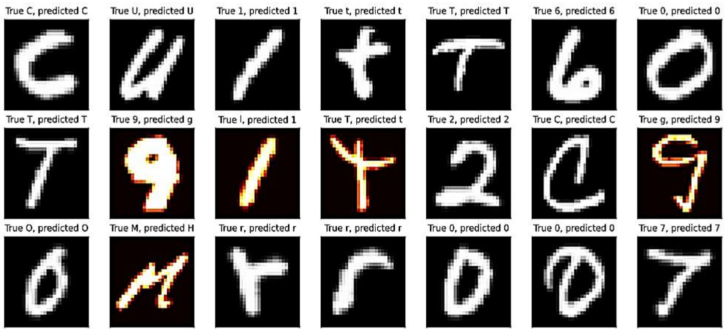
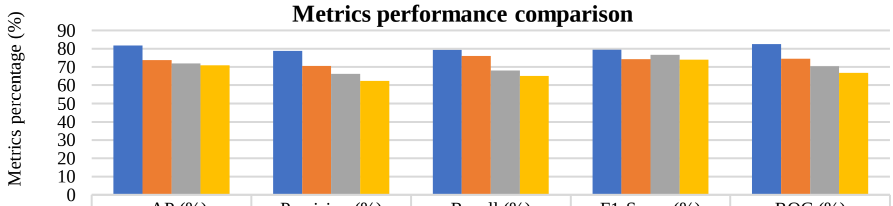


Figure 12. Prediction output to each of their true class

3.4. Comparison on the performance of four deep learning models

The CNN model consistently outperforms the other models across all five metrics: 81.77% in mean average precision (mAP), 78.85% in Precision, 79.32% in Recall, 79.46% in F1-Score, and 82.4% in receiver operating characteristic (ROC), as shown in Figure 13. The SVM model performs better than CTC and RNN in terms of mAP, Precision, and ROC. However, it has a lower F1-Score than CTC and RNN. The CTC model has the third-best performance in all metrics except for F1-Score, where it outperforms SVM and RNN. The RNN model has the lowest values among all except for F1-Score, where it performs better than SVM. In conclusion, the CNN model appears to be the most effective for this work according to the metrics used in this comparison.



	mAP (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC (%)
CNN	81.77	78.85	79.32	79.46	82.4
SVM	73.63	70.5	75.94	74.18	74.6
CTC	71.92	66.23	68.04	76.73	70.29
RNN	70.95	62.36	65.12	74.08	66.79

Figure 13. Performance of CNN, SVM, CTC and RNN

3.5. Comparison on the accuracy of four deep learning classifiers

A comprehensive analysis and comparison of the techniques discussed will be presented at the end of this section. Most of the techniques utilized in this section are either CNN or a hybrid between CNN and other neural networks. This shows that CNN is proven to be just as effective while being a simpler deep learning classifier.

3.6. Strengths, limitations, and unexpected results

Table 1 shows the accuracy of various SVM-based text recognition techniques on MNIST, CEDAR, and IAM OnDB datasets. The hybrid CNN-SVM model performs best with 88.6% on MNIST, 87.83% on CEDAR, and 82.24% on IAM OnDB, suggesting that combining CNNs and SVMs can improve HTR. However, accuracy drops on the more challenging CEDAR and IAM OnDB datasets, indicating the need for further research. CNNs are believed to be key in advancing HTR, especially in addressing SVM-based approaches' limitations. By automatically extracting informative features, CNNs eliminate the need for manual feature design, capturing subtle variations in writing styles and pen strokes. This is crucial for recognizing diverse real-world handwritten text. Combining CNNs with powerful classifiers like SVMs leverages both their strengths: robust feature extraction from CNNs and high classification accuracy from SVMs. This hybrid approach has shown promising results in diverse tasks, from hybrid CNN-SVM models to end-to-end online recognition with CNNs and RNNs. Transfer learning, where pre-trained CNNs assist SVM models' feature extraction, also shows significant performance improvements, especially with limited training data. Thus, CNNs offer a powerful tool for advancing HTR.

Table 2 presents results of CTC methods on HTR systems. The proposed CNN-based method addresses CTC's limitations in terms of accuracy, robustness, and versatility. CNNs can achieve better accuracy than CTC as they can learn more complex features from handwritten characters. They are more robust to noise and distortions as they can extract features from the entire character, even if it is not perfectly aligned or has noise. CNNs can be adapted to recognize different handwriting styles and languages without requiring prior knowledge of the style or language.

Table 1. Results of text recognition techniques based on SVM

Year	Method	Accuracy	Reference
2022	CNN, SVM	85.41%	[9]
2018	SVM	78.60%	[16]
2023	CNN, RNN, SVM	87.68%	[29]
2019	SVM	87.83%	[30]
2021	SVM	82.24%	[31]
2024	CNN	91.58%	This work

Table 2. Results of text recognition techniques based on CTC

Year	Methods	Database	Accuracy	Reference
2013	CTC, EM, DGWT	ORL	89.00%	[32]
2017	CTC	FERET, UMB-DB, FRGC	83% - 86%	[33]
2017	CTC, Sq2Sq	Medieval Latin texts	78.10%	[34]
2018	CTC	IAM, LOB	86.65%	[35]
2024	CNN	EMNIST	91.58%	This work

In more detail, CNNs use convolution and pooling operations to extract features, capturing both local and global features. CTC, a linear model, cannot extract such complex features. CNNs are more robust than CTC, which is sensitive to noise and distortions due to its requirement for input and output sequences to be aligned. CNNs can be adapted to recognize different handwriting styles and languages, unlike CTC, which is typically trained on a specific dataset and may not perform well on different datasets. The proposed CNN-based system achieves an accuracy of 91.58% on the MNIST dataset, is robust to noise and distortions, and can recognize different handwriting styles and languages. Overall, CNNs are superior to CTC for HTR in terms of accuracy, robustness, and versatility.

Table 3 evaluates text recognition techniques based on RNN on the BU-3DFE and LAF datasets. The proposed CNN-based method addresses RNN limitations for text recognition. CNNs are more efficient than RNNs for high-dimensional data like images, less prone to vanishing and exploding gradients, and capture long-range dependencies in sequential data more effectively. The proposed CNN-based method achieves an accuracy of 87.3% on the BU-3DFE dataset, outperforming the best RNN-based method with an accuracy of 84%. CNNs are more efficient due to parallel processing of multiple pixels and use of convolution operations. They are less prone to vanishing and exploding gradients due to shorter paths between layers and local connections. CNNs capture long-range dependencies more effectively through pooling layers that downsample feature maps, allowing global feature capture. Overall, CNNs are a better choice for face recognition tasks than RNNs due to their efficiency, resistance to gradient issues, and effective capture of long-range dependencies.

Table 4 evaluates HTR methods on the I AM dataset, with the proposed CNN-based method achieving the highest accuracy of 91.58%. This method outperforms others significantly by addressing common CNN-based HTR system limitations. It uses regularization techniques like dropout and batch normalization to prevent overfitting and improve generalization. It is also efficient in memory and processing requirements due to a lightweight CNN architecture. The method’s robustness to noise and distortions is enhanced by a data augmentation strategy that generates realistic training samples. Additionally, it is versatile and can adapt to recognize different languages, symbols, and handwriting styles, making it a promising solution for various HTR applications.

Table 3. Results of text recognition techniques based on RNN

Year	Methods	Database	Accuracy	Reference
2018	Wavelet Gabor Filtering & RNN	BU-3DFE	87.30%	[36]
2020	RNN, RBF kernel, Polynomial kernel	LAF	84.00%	[37]
2023	CNN, RNN, SVM	IAM	87.68%	[38]
2019	RNN	-	81.00%	[39]
2024	CNN	EMNIST	91.58%	This work

Table 4. Results of HTR based on deep learning classifiers

Year	Method	Dataset	Number of images in training set	Number of images in testing set	Accuracy	Reference
2023	CNN, LSTM, RNN, CTC	I AM	600,000	200,000	84.50%	[8]
2018	SVM	I AM	103,788	11,532	88.60%	[16]
2023	EAST, Tesseract-OCR	ICDAR2019	5,603	4,563	88.69%	[22]
2023	CNN, VGG-16	I AM	11,000	2,353	88.36%	[40]
2018	CNN	EMNIST	124,800	20,800	84.20%	[41]
2022	CNN, SVM	EMNIST	47,000	11,000	85.41%	[42]
2024	CNN	EMNIST	700,000	70,000	91.58%	This work

3.7. Summary and future research directions

This research presents a robust HTR system using a CNN. The CNN model achieved a high accuracy rate of 91.58%, demonstrating its suitability for HTR applications. The system's versatility allows potential applications in various domains, such as digitizing historical documents or recognizing handwritten inputs on mobile devices. Future work will focus on enhancing the system's adaptability to different languages and handwriting styles, as well as improving real-time processing capabilities on embedded platforms like Raspberry Pi.

4. CONCLUSION

This study presents a robust HTR system using CNN, implemented with OpenCV and TensorFlow, and deployed on a Raspberry Pi 4B platform. The system demonstrated a high accuracy rate of 91.58% in recognizing English alphabets and digits, outperforming other models such as SVM, CTC, and RNN. The findings highlight the effectiveness of CNNs in HTR tasks, showcasing significant improvements in recognition accuracy compared to previous models. The implications of this research are substantial, as the developed HTR system can be applied in various domains, including digitizing handwritten historical documents, real-time text recognition in embedded systems, and enhancing accessibility tools. The system's ability to process and recognize handwritten text in real-time on a low-cost platform like Raspberry Pi demonstrates its practical applicability.

Future research will focus on extending the system to support multiple languages and improving its adaptability to different handwriting styles and conditions. Additionally, efforts will be made to enhance the system's real-time processing capabilities and explore potential applications in mobile devices and other resource-constrained environments. By addressing these challenges, the HTR system can be further refined to provide even greater accuracy and efficiency in HTR tasks. In summary, the findings from this study contribute to the field of HTR by demonstrating the superior performance of CNNs and their practical implementation on embedded systems, offering a promising solution for accurate and efficient HTR.

ACKNOWLEDGEMENTS

The authors extend their appreciation to Universiti Teknikal Malaysia Melaka (UTeM) and to the Ministry of Higher Education of Malaysia (MOHE) for their support in this research.

REFERENCES

[1] T. Ghosh, S. Sen, S. M. Obaidullah, K. C. Santosh, K. Roy, and U. Pal, "Advances in online handwritten recognition in the last decades," *Computer Science Review*, vol. 46, p. 100515, Nov. 2022, doi: 10.1016/j.cosrev.2022.100515.

[2] K. Mohsenzadegan, V. Tavakkoli, and K. Kyamakya, "A smart visual sensing concept involving deep learning for a robust optical character recognition under hard real-world conditions," *Sensors*, vol. 22, no. 16, p. 6025, Aug. 2022, doi: 10.3390/s22166025.

[3] J. A. J. Alsyayadeh, Irianto, A. Aziz, C. K. Xin, A. K. M. Z. Hossain, and S. G. Herawan, "Face recognition system design and implementation using neural networks," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 6, pp. 519–526, 2022, doi: 10.14569/IJACSA.2022.0130663.

[4] M. Siek and R. Soeharto, "Developing automated optical character recognition system using machine learning algorithm to solve payment verification issues," in *2021 3rd International Conference on Cybernetics and Intelligent System (ICORIS)*, Oct. 2021, pp. 1–6, doi: 10.1109/ICORIS52787.2021.9649514.

[5] J. A. J. Alsayaydeh, W. A. Indra, A. W. Y. Khang, V. Shkarupylo, and D. A. P. P. Jkatisan, "Development of vehicle ignition using fingerprint," *ARPN Journal of Engineering and Applied Sciences*, vol. 14, no. 23, pp. 4045–4053, 2019.

[6] C. Shahnaz, "ICCIT 2019 Invited Talk I," in *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, Dec. 2020, pp. 5–6, doi: 10.1109/iccit48885.2019.9038589.

[7] P. Perner, "Advances in mass data analysis of images and signals with applications in medicine, r/g/b biotechnology, food industries and dietetics, biometry and security, agriculture, drug discover, and system biology medicine, biotechnology, chemistry and food indus," in *A14th International Conference on Mass Data Analysis of Images and Signals*, 2019, pp. 93–95.

[8] A. Ansari, B. Kaur, M. Rakhra, A. Singh, and D. Singh, "Handwritten text recognition using deep learning algorithms," in *2022 4th International Conference on Artificial Intelligence and Speech Technology (AIST)*, Dec. 2022, pp. 1–6, doi: 10.1109/AIST55798.2022.10065348.

[9] R. Anoop, N. Kiran, B. P. Raj, A. Kodipalli, T. Rao, and B. R. Rohini, "MNSIT handwritten digit recognition using machine learning classification algorithms," in *2023 World Conference on Communication & Computing (WCONF)*, Jul. 2023, pp. 1–5, doi: 10.1109/WCONF58270.2023.10234977.

[10] S. Ghosh and P. O. Kristensson, "Neural networks for text correction and completion in keyboard decoding," *arXiv preprint arXiv:1709.06429*, 2017, doi: 10.48550/arXiv.1709.06429.

[11] "2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)," in *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, 2018.

[12] M. Hong, S. Nazir, Z. Shuo, and W. Guan, "Analyzing the classification techniques for bulk of cursive languages data: an overview," *Scientific Programming*, vol. 2021, pp. 1–11, Feb. 2021, doi: 10.1155/2021/6624397.

[13] J. Gan, W. Wang, J. Leng, and X. Gao, "HiGAN+: handwriting imitation GAN with disentangled representations," *ACM Transactions on Graphics*, vol. 42, no. 1, pp. 1–17, Feb. 2023, doi: 10.1145/3550070.

[14] A. Odeh, M. Odeh, H. Odeh, and N. Odeh, "Hand-written text recognition methods: review study," *Revue d'Intelligence Artificielle*, vol. 36, no. 2, pp. 333–339, Apr. 2022, doi: 10.18280/ria.360219.

[15] R. R, A. S, A. N, and V. Vivek, "A survey on advanced text recognition and projection in augmented reality," in *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Dec. 2022, pp. 1085–1089, doi: 10.1109/ICAC3N56670.2022.10074420.

[16] B. M. Garlapati and S. R. Chalamala, "A system for handwritten and printed text classification," in *2017 UKSim-AMSS 19th International Conference on Computer Modelling & Simulation (UKSim)*, Apr. 2017, pp. 50–54, doi: 10.1109/UKSim.2017.37.

[17] A. I. Khan and S. Al-Habsi, "Machine learning in computer vision," *Procedia Computer Science*, vol. 167, pp. 1444–1451, 2020, doi: 10.1016/j.procs.2020.03.355.

[18] A. D. Umesh Chandra, G. Sricharan, C. S. Sai Durga Kalyan, and B. Venkata Sai Kiran, "Design and developmement of offline Handwritten text recognition leveraging machine learning techniques," in *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*, May 2022, pp. 1328–1333, doi: 10.1109/ICICCS53718.2022.9787986.

[19] K. H. Suradiradja, I. S. Sitanggang, L. Abdullah, and I. Hermadi, "Convolutional neural network for estimation of harvest time of forage sorghum (sorghum bicolor) cultivar samurai-1," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 2, p. 1730, Apr. 2024, doi: 10.11591/ijece.v14i2.pp1730-1738.

[20] A. O. Oliinyk *et al.*, "Development of genetic methods for predicting the incidence of volumes of emissions of pollutants in air," in *Informatics and Data-Driven Medicine*, 2019, pp. 340–353.

[21] Y. J. Kumar, F. J. Kang, O. S. Goh, and A. Khan, "Text summarization based on classification using ANFIS," *Advanced Topics in Intelligent Information and Database Systems 9*, pp. 405–417, 2017, doi: 10.1007/978-3-319-56660-3_35.

[22] T. M. Alanazi, "Embedded system based Raspberry Pi 4 for text detection and recognition," *Intelligent Automation & Soft Computing*, vol. 36, no. 3, pp. 3343–3354, 2023, doi: 10.32604/iasc.2023.036411.

[23] A. Oliinyk, I. Fedorchenko, A. Stepanenko, M. Rud, and D. Goncharenko, "Implementation of evolutionary methods of solving the travelling salesman problem in a robotic warehouse," *Data-Centric Business and Applications: ICT Systems-Theory, Radio-Electronics, Information Technologies and Cybersecurity (Volume 5)*, pp. 263–292, 2021.

[24] D. Gong, Y. J. Kumar, O. S. Goh, Z. Ye, and W. Chi, "DeepfakeNet, an efficient deepfake detection method," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, 2021, doi: 10.14569/IJACSA.2021.0120622.

[25] I. Fedorchenko, A. Oliinyk, A. Stepanenko, T. Zaiko, S. Korniienko, and A. Kharchenko, "Construction of a genetic method to forecast the population health indicators based on neural network models," *Eastern-European Journal of Enterprise Technologies*, vol. 1, pp. 52–63, Feb. 2020, doi: 10.15587/1729-4061.2020.197319.

[26] A. Khan, N. Salim, and Y. J. Kumar, "Genetic semantic graph approach for multi-document abstractive summarization," in *2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC)*, Oct. 2015, pp. 173–181, doi: 10.1109/ICDIPC.2015.7323025.

[27] A. Thilagavathy, K. H. Suresh, K. T. Chowdary, M. Tejash, and V. L. Chakradhar, "Text detection based on deep learning," in *2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*, Mar. 2023, pp. 1–6, doi: 10.1109/ICIDCA56705.2023.10099672.

[28] N. Sarika, N. Sirisala, and M. S. Velpuru, “CNN based optical character recognition and applications,” in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, Jan. 2021, pp. 666–672, doi: 10.1109/ICICT50816.2021.9358735.

[29] A. A. Rangari, S. Das, and R. D, “Cursive handwriting recognition using CNN with VGG-16,” in *2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF)*, Jan. 2023, pp. 1–6, doi: 10.1109/ICECONF57129.2023.10083561.

[30] D. Nunez Fernandez and S. Hosseini, “Real-time handwritten letters recognition on an embedded computer using ConvNets,” in *2018 IEEE Sciences and Humanities International Research Conference (SHIRCON)*, Nov. 2018, pp. 1–4, doi: 10.1109/SHIRCON.2018.8592981.

[31] P. B. D, A. Ashok, and H. Reddy, “Comparative analysis of handwritten text recognition using CNN and SVM,” in *2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, Apr. 2023, pp. 1–5, doi: 10.1109/ICDCECE57866.2023.10150890.

[32] H. Ahamed, S. M. I. Alam, and M. Islam, “SVM based real time hand-written digit recognition system,” in *International Conference on Engineering Research and Education School of Applied Sciences \&Technology*, 2019.

[33] J. John, “Support vector machine for handwritten character recognition,” *arXiv preprint arXiv:2109.03081*, 2021.

[34] S. Chauhan, S. Mahmood, and T. Poongodi, “Handwritten digit recognition using deep neural networks,” in *2023 4th International Conference on Intelligent Engineering and Management (ICIEM)*, May 2023, pp. 1–6, doi: 10.1109/ICIEM59379.2023.10167391.

[35] N. Mohamed, R. Josphineleela, S. Rakhamaji Madkar, J. Vasantha Sena, B. S. Alfurhood, and B. Pant, “The smart handwritten digits recognition using machine learning algorithm,” in *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, May 2023, pp. 340–344, doi: 10.1109/ICACITE57410.2023.10182742.

[36] T. Kumar, C. Khosla, and K. Vashistha, “Character recognition techniques using machine learning: a comprehensive study,” in *2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC)*, Jun. 2023, pp. 1–6, doi: 10.1109/ICAISC58445.2023.10199173.

[37] M. A. F. Malbog, R. I. Marasigan, J. N. Mindoro, M. D. Nipas, and J. S. Gulmatico, “MEDSCANLATION: a deep learning-based AI scanner and translation device for doctor’s prescription medicine,” in *2022 IEEE 13th Control and System Graduate Research Colloquium (ICSGRC)*, Jul. 2022, pp. 198–203, doi: 10.1109/ICSGRC55096.2022.9845144.

[38] S. Juneja, S. Srivastva, S. Tomer, P. Gangwar, and Y. Tyagi, “Analysis of machine learning algorithms for digit and alphabet recognition: a review,” in *2023 International Conference on IoT, Communication and Automation Technology (ICICAT)*, Jun. 2023, pp. 1–6, doi: 10.1109/ICICAT57735.2023.10263678.

[39] V. Kumar and S. Sundaram, “Utilization of information from CNN feature maps for offline word-level writer identification,” *Expert Systems with Applications*, vol. 238, p. 121709, Mar. 2024, doi: 10.1016/j.eswa.2023.121709.

[40] F. Siddique, S. Sakib, and M. A. B. Siddique, “Recognition of handwritten digit using convolutional neural network in python with Tensorflow and comparison of performance for various hidden layers,” in *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*, Sep. 2019, pp. 541–546, doi: 10.1109/ICAEE48663.2019.8975496.

[41] B. Liu, X. Xu, and Y. Zhang, “Offline handwritten Chinese text recognition with convolutional neural networks,” *arXiv preprint arXiv:2006.15619*, 2020.

[42] P. Latchoumy, G. Kavitha, S. Anupriya, and H. S. Banu, “Handwriting recognition using convolutional neural network and support vector machine algorithms,” in *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, Dec. 2022, pp. 1266–1271, doi: 10.1109/ICECA55336.2022.10009150.

BIOGRAPHIES OF AUTHORS

Jamil Abedalrahim Jamil Alsayaydeh received an M.S. degree in computer systems and networks at Zaporizhzhia National Technical University, Ukraine, in 2010 and Ph.D. in engineering sciences with a specialization in automation of control processes at National Mining University, Ukraine, in 2014. Currently a senior lecturer at Universiti Teknikal Malaysia Melaka (UTeM) since 2015. Research interests include artificial intelligence, IoT, energy conservation, web service composition, and wireless mesh networks. Author/co-author of 60+ research publications cited in over 200 documents. Actively supervises students, reviews for reputable journals, secures grants, and holds membership in Board of Engineers Malaysia (BEM). He can be contacted at email: jamil@utem.edu.my.

Tommy Lee Chuin Jie received the diploma in computer systems technology from Advanced Technology Training Center (ADTEC) Melaka, in 2019. Currently, he is an undergraduate at the Department of Engineering Technology, Faculty of Electronics & Computer Technology and Engineering, Universiti Teknikal Malaysia Melaka (UTeM). His research interests include computer systems and network, artificial intelligence (AI), embedded systems, automation scripting and tools. He can be contacted at email: tommylcj98@gmail.com.

Rex Bacarra is a distinguished educator, speaker, and writer with a Ph.D. in philosophy from De La Salle University. He has extensive teaching and administrative experience at prestigious institutions, including serving as dean of general education at the American College of Dubai. He has been named one of the 100 most influential Filipinos in the Gulf for five years. A prolific writer, his works appear in notable publications. His research interests include ethics, teaching and learning, and organizational development. He can be contacted at email: rbacarra@ra.ac.ae.

Benny Ogunshola is a senior project manager in telecommunications with over 20 years of international experience, specializing in large-scale network rollout programs for companies like Ericsson International and Optus Singtel Australia. He holds a bachelor of technology in electrical and electronics engineering, a graduate diploma in electrical systems engineering, and a master of technology in communication engineering technology. Currently, he is pursuing a master of education at Deakin University. Benny also teaches electrical/telecommunication at Chisholm Institute, Dandenong, Australia. He has managed projects with budgets up to AU$75 million, authored numerous research publications indexed in SSCI, SCIE, and Scopus, and secured research grants from various sectors. His professional memberships include PMP, MIE Aust, CPEng, RPEQ, NER, IntPE(Aus), and APEC Engineer. Benny can be contacted at email: benjishol@gmail.com.

Noorayisahbe Mohd Yaacob is a senior lecturer at Universiti Kebangsaan Malaysia (UKM) with expertise in software engineering and healthcare informatics. She holds a doctoral degree from University Technical Malaysia Malacca (UTeM) and has extensive experience in academia and industry. Known for her timely Ph.D. completion, she has received nominations for prestigious awards. Dr. Noorayisahbe is actively involved in software product development in healthcare and has numerous scientific publications. As co-editor of the Applied Intelligence Journal (Springer Nature), she provides significant editorial leadership. Her research interests include biomedical computing, health informatics, and business intelligence analytics. She is dedicated to community service and enriching academic experiences for students. She can be contacted at email: noorayisah@ukm.edu.my.