# Robust and Efficient Writer-Independent IMU-Based Handwriting Recognization

 $\begin{array}{c} \mbox{Jindong Li$^{1}$} [0000-0002-3550-1660], \mbox{Tim Hamann}$^{2}$} [0000-0003-3562-6882], \mbox{Jens} \\ \mbox{Barth}$^{2}$} [0000-0003-3967-9578], \mbox{Peter Kaempf$^{2}$}, \mbox{Dario Zanca$^{1}$} [0000-0001-5886-0597]}, \\ \mbox{and Bjoern Eskofier}$^{1}$} [0000-0002-0417-0336] \end{array}$ 

<sup>1</sup> Machine Learning and Data Analytics Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
<sup>2</sup> STABILO International GmbH, Germany

Abstract. Online handwriting recognition (HWR) using data from inertial measurement units (IMUs) remains challenging due to variations in writing styles and the limited availability of high-quality annotated datasets. Traditional models often struggle to recognize handwriting from unseen writers, making writer-independent (WI) recognition a crucial but difficult problem. This paper presents an HWR model with an encoderdecoder structure for IMU data, featuring a CNN-based encoder for feature extraction and a BiLSTM decoder for sequence modeling, which supports inputs of varying lengths. Our approach demonstrates strong robustness and data efficiency, outperforming existing methods on WI datasets, including the WI split of the OnHW dataset and our own dataset. Extensive evaluations show that our model maintains high accuracy across different age groups and writing conditions while effectively learning from limited data. Through comprehensive ablation studies, we analyze key design choices, achieving a balance between accuracy and efficiency. These findings contribute to the development of more adaptable and scalable HWR systems for real-world applications. The code is available at https://github.com/jindongli24/REWI.

**Keywords:** Online Handwriting Recognition · Time-series Analysis · Inertial Measurement Unit

#### 1 Introduction

Handwriting has been an essential way of recording and sharing information throughout human history. With advancements in technology, the demand for digitizing handwriting has increased. Handwriting recognition (HWR), a method for turning handwritten symbols into computer-readable text, has become an important area of research.

HWR is generally divided into two types: offline HWR and online HWR. Offline HWR, also known as optical character recognition, identifies handwriting from static images of handwritten text. This approach is widely used in various fields, including historical research [18] and healthcare [7]. On the other hand,

online HWR works with time-series data that captures dynamic handwriting features such as strokes, positions, directions, and speeds. This data is usually collected using touch screens and styluses on mobile devices [3], which limits the writing surface users can write on.

Another approach to online HWR uses pens equipped with inertial measurement units (IMUs) [21,1,14]. These sensors, including accelerometers and gyroscopes, capture pen movement without relying on the exact position of the tip, allowing the pen to function independently of external devices and on any surface. As IMU sensor costs continue to decrease, this method shows great potential for the application in online HWR. However, variations in handwriting styles can impact recognition accuracy, and sensor noise from rough surfaces, temperature changes, and digitization artifacts add further challenges. More significantly, gravitational acceleration also introduces noise, making motion tracking less accurate and complicating reliable HWR.

This paper introduces a sequence-to-sequence model for IMU-based online HWR that addresses challenges related to handwriting style variations and sensor noise. The model uses an encoder-decoder structure that combines a convolutional neural network (CNN) with a bidirectional long short-term memory (BiLSTM) network and integrates recent advancements in deep learning. We evaluate the proposed model by comparing it with several mainstream models and existing IMU-based HWR methods. Experimental results on datasets collected with an IMU-based pen show that our model outperforms others in terms of accuracy, data efficiency, robustness, and flexibility in WI HWR.

Section 2 reviews related work on IMU-based HWR and advanced mainstream models. Section 3 describes the datasets, the CNN-BiLSTM model, and the data augmentation methods that we used. Section 4 explains the training process and presents the experimental results in detail. Section 5 discusses key factors that improve IMU-based HWR. Finally, Section 6 provides the conclusions and suggests directions for future research.

## 2 Related Works

#### 2.1 IMU-based HWR

IMU-based HWR has been an area of research for decades. Early studies, such as [4,9], used dynamic-time warping-based algorithms to recognize digit data collected with IMU-based pens, achieving recognition rates above 90%. Later works, including [13,15], investigated LSTM-based models for recognizing individual English characters, reaching recognition accuracies of up to 99.68% and 79.01% on their respective datasets. While these methods achieved high accuracy, they rely on isolated character recognition, processing entire input signals to classify single characters. This character-by-character approach disrupts natural writing flow, making it less practical for real-world applications where people typically write word-by-word.

To handle more complex tasks, [22,16] employed CNN-LSTM models with connectionist temporal classification (CTC) [6] to recognize English and Ger-

man characters in a sequence-to-sequence format. These models were tested on datasets collected using the IMU-based pen developed by STABILO [21]. Although the methods achieved character error rates (CERs) of 27.8% and 17.97%, which are worse than earlier approaches, they paved the way for recognizing full words and even sentences, taking a step closer to practical applications.

## 2.2 Advancements in Deep Learning Architectures

In recent years, the introduction of ResNet [8] and Transformers [20] has significantly advanced the development of deep learning models. ResNet tackled the vanishing gradient problem using skip connections, making it possible to train very deep neural networks effectively. Transformers introduced self-attention mechanisms, which improved parallelization and enhanced the ability to model long-range dependencies compared to convolutional or recurrent neural networks.

Building on these innovations, Vision Transformer [5] applied a transformer-based architecture to image recognition by treating images as sequences of patches and leveraging self-attention to achieve state-of-the-art results. MLP-Mixer [19] showed that strong performance in vision tasks could be achieved without convolutions or self-attention, relying solely on multi-layer perceptrons (MLPs) to mix spatial and channel information. Swin Transformers [11,10] introduced a hierarchical architecture with shifted window-based self-attention, enabling efficient multi-scale modeling for various vision tasks. ConvNeXt [12] combined ideas from CNNs and Transformers, achieving state-of-the-art performance while maintaining the simplicity and efficiency of traditional CNNs. xLSTM [2] enhanced conventional LSTMs with memory augmentation and cross-layer parameter sharing, improving their ability to handle long-term dependencies and process sequential data effectively.

Although these models were originally designed for different tasks, their core principles e.g. self-attention, hierarchical structures, and memory augmentation, are well-suited for time-series data. Applying these techniques to IMU-based HWR could potentially enhance performance and open new research opportunities in the field.

#### 3 Methods

#### 3.1 Datasets

This paper uses a dataset collected with the IMU-based pen developed by STABILO [21]. The pen is equipped with two accelerometers, one at each end, along with a gyroscope, a magnetometer, and a force sensor, generating 13 output channels at a sampling rate of 100 Hz. Data collection included 984 recording sessions with participants of different ages and handednesses, resulting in 54,666 samples of English and German words of varying lengths. These words cover 59 character categories, including both upper- and lowercase letters for both languages.

The datasets are evaluated using two configurations: writer-independent (WI) and random splits, both following a 5-fold cross-validation approach. In the WI split, data are divided so that no writer appears in both the training and testing sets, ensuring handwriting styles remain independent. In the random split, samples are assigned randomly without considering writer identity.

To evaluate model robustness, we analyze subsets of the WI dataset based on participants' ages. Since children are still developing their handwriting skills, their handwriting patterns differ significantly from those of adults, particularly in areas such as writing speed and discontinuity due to high cognitive load. These patterns typically stabilize around 14 years old [17]. Based on this, we classify participants aged 12 and under as children to ensure that most writers represent typical children's handwriting patterns, and those aged 18 and older as adults. We assess model performance on both groups. Participants aged 13 to 17 were excluded because some had already developed mature and fluent handwriting, making them less representative of either the children or adult groups. Notably, the adult subset (47,992 samples) is significantly larger than the children subset (6,070 samples).

To evaluate data efficiency, we reduce the training sets to 50% and 25% of their original size while keeping the testing sets unchanged. Due to the significant imbalance between right- and left-handed samples (51,854 vs. 2,812, respectively), we exclude handedness-related comparisons from our analysis.

For commercial reasons, this dataset will not be publicly available. However, we also use the writer-dependent (WD) subset, where the WD subset is divided by words, and the WI subset only from the right-handed portion of the OnHW-words500 dataset [16] to further benchmark our model against various mainstream models. Due to the significant imbalance between right-handed and left-handed samples (25,218 vs. 1,000, respectively), left-handed portion of the OnHW dataset are excluded from our experiments.

#### 3.2 HWR Model

The sequence-to-sequence model uses an encoder-decoder structure, as shown in Fig 1. The encoder is a CNN that efficiently extracts and embeds input features, while the decoder is a BiLSTM network that captures contextual information in both directions. Both the CNN and BiLSTM can process inputs of varying lengths, providing flexibility for different handwriting data.

The encoder has three stages, each with a patch embedding layer followed by three convolutional blocks. The patch embedding layer uses a 1-D convolutional layer with a kernel size of 2 and a stride of 2, followed by a 1-D instance normalization layer. The first patch embedding layer converts the 13-channel input data into 128 channels, and each subsequent layer doubles the number of channels.

The convolutional blocks are based on the ConvNeXt block, which improves efficiency using grouped convolution and inverted bottlenecks. To reduce computational cost without sacrificing performance, each block includes a depth-dilated 1-D separable convolution. This layer uses a 1-D depthwise convolution with a kernel size of 5 that doubles the input dimension, followed by a 1-D pixelwise

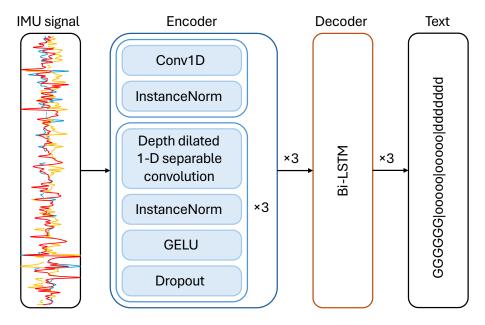


Fig. 1. Model architecture

convolution that reduces it back. This design creates a larger hidden space for better feature representation compared to standard depthwise separable convolutions while reducing complexity by omitting an extra layer found in the ConvNeXt block. Each depth-dilated 1-D separable convolution is followed by a 1-D instance normalization layer, a GELU activation function, and a dropout layer with a drop rate of 0.2.

The decoder has three BiLSTM layers, each with a hidden size of 128 and a dropout rate of 0.2. These layers are followed by a fully connected layer and a Softmax layer for pixelwise classification. The Softmax outputs are then processed by a greedy CTC decoder to produce the final results.

# 3.3 Data Augmentation

To improve the model's resistance to noise, we used four data augmentation techniques: AddNoise, Drift, Dropout, and TimeWarp. Each technique has a 25% chance of being applied to a given signal. TimeWarp only affects the time dimension, while the other methods are applied multiplicatively to keep the augmented signals' magnitude similar to the original signals. Examples of these augmentations are shown in Fig 2. After augmentation, the signals are also normalized.

- AddNoise: Adds Gaussian noise to the original signals.
- Drift: Divides the original signals into sections and applies random drift within each section.

- Dropout: Randomly replaces small segments of the signal with the last value preceding each segment.
- **TimeWarp**: Randomly adjusts the speed of sections of the original signal.

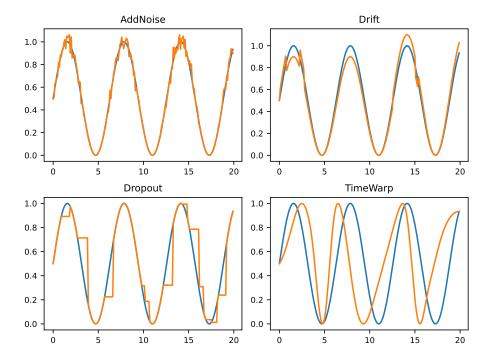


Fig. 2. Data augmentations

# 4 Experiments

The experiments are divided into five parts: comparison with previous methods, benchmarking against mainstream models, robustness evaluation, data efficiency evaluation, and an ablation study. In each part, models are trained using 5-fold cross-validation, and the best results from each fold's test set are used to calculate the final results.

Except for the CLDNN [22], which is trained using the method described in its original paper, all other models, including those from previous studies and mainstream models, are trained using the same following approach. We use the AdamW optimizer with a learning rate of 0.001 for 300 epochs. To improve convergence, a linear learning rate warm-up starts at 0.0001 for the first 30 epochs, followed by a cosine annealing schedule for the remaining epochs. The

training process uses the CTC loss function and a batch size of 64. Models are saved and evaluated every 5 epochs during training. All experiments are conducted on a computer with an AMD Ryzen 9 5950X processor and a GeForce RTX 3090 graphics card, using the PyTorch library for model implementation.

To assess model performance, we use two metrics: character error rate (CER) and word error rate (WER). These measure the proportion of errors, including substitutions, deletions, and insertions, relative to the total number of characters in the reference text at the character and word levels, respectively. Additionally, we evaluate model size and computational requirements by measuring the number of parameters (number of Params) and multiply and accumulate operations (MACs).

# 4.1 Comparison with Previous Works

In this section, we compare our model with previous works using both our dataset and the OnHW dataset [16]. For our dataset, we evaluate performance on the random and WI splits, while for the OnHW dataset, we use the WD and WI splits of the right-handed words500 subset. The comparison includes three models: CNN+BiLSTM [16], CLDNN, and our proposed model. Since the code for the previous models is not publicly available, we re-implemented them based on their published descriptions using PyTorch. We replicated the training and preprocessing strategy for CLDNN as described. However, the training details for CNN+BiLSTM were not clearly specified, making it difficult to reproduce the reported performance. Therefore, we trained CNN+BiLSTM using our training pipeline and data augmentation strategy, which may result in differences from the original reported results. Additionally, we include the previously published results for CNN+BiLSTM on the OnHW dataset for reference.

WI Random Models #ParamsMACs CER WER CER WER CNN+BiLSTM 0.0710 0.31420.15120.5213 0.40M153MCLDNN 0.07840.3291 0.15330.5112 0.75M291M 0.0367 0.17030.06920.2676 3.89M600MOurs

Table 1. Comparison with previous works on our dataset.

As shown in Table 1, our model outperforms previous works, achieving the lowest CER and WER on both the random and WI splits of our dataset. Specifically, it achieves a CER of 0.0367 and a WER of 0.1703 on the random split, and a CER of 0.0692 and a WER of 0.2676 on the WI split. In comparison, the error rates of the CNN+BiLSTM and CLDNN models are approximately twice as high. However, our model has a larger number of parameters and higher computational costs than these models.

Ours

WDWI Models #Params MACs WER CER WER CER CNN+BiLSTM (orig.) 0.17160.51950.27800.60910.40M153MCNN+BiLSTM 0.52430.40M153M0.15970.17160.4240**CLDNN** 0.16960.54040.17150.3948 0.75M291M

0.0746

0.1559

3.89M

600M

0.4551

**Table 2.** Comparison with previous works on OnHW dataset [16].

As shown in Table 2, on the OnHW dataset, our model achieves the best performance compared to previous works. It achieves the lowest CER and WER across both the WD and WI splits, with a CER of 0.1546 and a WER of 0.4551 for the WD split, and a CER of 0.0746 and a WER of 0.1559 for the WI split. Compared to the original CNN+BiLSTM results reported in the paper, our reimplementation using the same model architecture but our training pipeline and data augmentation strategy leads to better convergence and lower error rates. In the WI split, our model reduces the error rates by more than half, highlighting its effectiveness in handling challenges associated with unseen handwriting styles.

#### 4.2 Benchmark against Mainstream Models

0.1546

In this section, we benchmark our proposed model against various models using the same datasets as in the previous experiments. To efficiently extract high-level features while reducing output size and computational cost for the decoder, we use ResNet, ConvNeXt, MLP-Mixer, and Swin Transformer V2 as encoders combined with our BiLSTM decoder. To capture long-range dependencies, we use the mLSTM module of xLSTM as a decoder alongside our CNN encoder. For a fair comparison, we implement the mLSTM module in a bidirectional manner, allowing it to access both past and future information. Since the Transformer architecture can capture high-level information and has a global receptive field over the input, we evaluate it as an encoder, a decoder, and as a standalone HWR model. Since these models were not originally designed for HWR and vary in size, we adjust their hyperparameters to better suit the HWR task while maintaining a similar number of parameters across models. Additionally, because Transformer, MLP-Mixer, Swin Transformer V2, and Bi-mLSTM require a fixed input length, all inputs for these models are zero-padded to a length of 1024.

Table 3 shows that our model achieves the best CER and WER on the WI split, with values of 0.0692 and 0.2676, respectively, demonstrating its strong ability to generalize to unseen handwriting styles. It also delivers competitive performance on the random split, achieving a CER of 0.0367 and a WER of 0.1703. Compared to other models, our approach provides well-balanced performance across both splits, outperforming most other models. While the MLP-Mixer achieves the lowest error rates on the random split, it requires significantly

WIRandom Models MACs#Params WER CER WER CER Transformer 0.04430.20820.4042 509M0.11633.96MResNet (enc.) 0.03060.3026591M 0.14560.08053.97MConvNeXt (enc.) 0.0355 0.16960.0834 0.3205 3.86M600MMLP-Mixer (enc.) 0.02740.13390.10250.36243.90M802MTransformer (enc.) 0.04220.10020.3572 3.71M477M 0.1975SwinV2 (enc.) 0.0303 0.14910.07460.2924 3.88M601MTransformer (dec) 0.0387 0.18220.0828 0.3183 3.82M590M Bi-mLSTM (dec.) 0.04760.21920.09140.34844.10M625MOurs 0.03670.17030.06920.26763.89M600M

 Table 3. Benchmark against mainstream models on our dataset.

higher computational resources, with the highest MACs among all models. Interestingly, the MLP-Mixer ranks second worst on the WI split. This highlights the efficiency and robustness of our model in achieving strong performance while maintaining a favorable balance between accuracy and computational cost.

Table 4. Benchmark against mainstream models on OnHW dataset.

Models	WD		WI		// <b>D</b>	MAG
	CER	WER	CER	WER	#Params	MACs
Transformer	0.2615	0.6283	0.1139	0.2573	3.96M	509M
ResNet (enc.)	0.1294	0.4164	0.0850	0.1846	3.97M	591M
ConvNeXt (enc.)	0.1515	0.4657	0.0812	0.1791	3.86M	600M
MLP-Mixer (enc.)	0.1438	0.4659	0.0964	0.2149	3.90M	802M
Transformer (enc.)	0.1817	0.5242	0.1060	0.2303	3.71M	477M
SwinV2 (enc.)	0.1750	0.4983	0.0820	0.1814	3.88M	601M
Transformer (dec)	0.1396	0.4407	0.0864	0.1881	3.82M	590M
Bi-mLSTM (dec.)	0.2267	0.5749	0.0841	0.1803	4.10M	625M
Ours	0.1546	0.4551	0.0746	0.1559	3.89M	600M

Table 4 compares the performance of different models on the OnHW dataset. Our model delivers strong results, with a CER of 0.1546 and a WER of 0.4551 on the WD split, and the lowest CER of 0.0746 and WER of 0.1559 on the WI split. CNN-based models, such as ResNet and ConvNeXt, also perform well, with ResNet achieving the lowest CER and WER on the WD split. These models show better performance on the unseen WI split compared to Transformer-based models, which generally have higher CER and WER. These results highlight the advantage of CNN-based models in handling unseen handwriting styles while maintaining a good balance between performance and efficiency.

#### 4.3 Robustness Evaluation

In this section, we assess the robustness of the models on the adult (Adult) and children (Children) subsets of our WI dataset. Additionally, we evaluate how models trained on the adult subset perform when tested on the children's subset (Adult2Child). All models are tested using the same configurations as in the previous experiments.

Models	Adults		Children		Adult2Child	
	CER	WER	CER	WER	CER	WER
CNN+BiLSTM	0.1525	0.5086	0.6545	0.9669	0.6678	0.9982
CLDNN	0.1479	0.4793	0.4704	0.8331	0.3716	0.7783
Transformer	0.1247	0.4199	0.7584	0.9881	0.3350	0.7466
ResNet (enc.)	0.0891	0.3194	0.1775	0.4934	0.2841	0.6607
ConvNeXt (enc.)	0.0910	0.3330	0.3102	0.6557	0.2801	0.6612
MLP-Mixer (enc.)	0.1118	0.3715	0.7732	0.9835	0.3044	0.6824
Transformer (enc.)	0.1115	0.3727	0.4559	0.8390	0.2888	0.6681
SwinV2 (enc.)	0.0842	0.3109	0.5400	0.8634	0.2637	0.6368
Transformer (dec.)	0.0891	0.3252	0.2462	0.5431	0.2807	0.6630
Bi-mLSTM (dec.)	0.0974	0.3524	0.3079	0.6769	0.2995	0.7018
Ours	0.0752	0.2772	0.1748	0.4493	0.2678	0.6273

Table 5. Robustness evaluation.

Table 5 shows the robustness of our proposed model on the adult and children subsets of the WI split of our dataset. Our model achieves the best results on both subsets, with a CER of 0.0752 and a WER of 0.2772 for adults, and a CER of 0.1748 and a WER of 0.4493 for children. When trained on the adult subset, although our model is slightly behind the Swin Transformer V2, it still achieves the lowest WER and ranks among the top performers on the children subset. This demonstrates its ability to learn robust features and adapt to different handwriting styles across age groups. Notably, models like the Transformer and MLP-Mixer struggle with the children subset, likely due to the challenge of generalizing to different handwriting styles when trained on the adult dataset and the smaller size of the children subset.

## 4.4 Data Efficiency Evaluation

In this section, we assess the data efficiency of the models by training them on different portions of the training set. Using 5-fold cross-validation, each training set consists of four groups of data. To evaluate performance under varying data conditions, we train all models on 100% (four groups), 50% (two groups), and 25% (one group) of the original training set from our WI dataset.

Table 6 shows the data efficiency of various models when trained on different proportions of the WI dataset. Our model consistently outperforms all others

100% 50% 25% Models CER WER CER WER CER WER CNN+BiLSTM0.5213 0.2733 0.6833 0.38770.8208 0.1512CLDNN 0.7096 0.15330.51120.18680.56650.2688Transformer 0.1163 0.40420.17650.5318 0.19100.5615ResNet (enc.) 0.08050.30260.12560.41350.19780.5730 ConvNeXt (enc.) 0.1297 0.20050.5869 0.0834 0.3205 0.4336 MLP-Mixer (enc.) 0.1025 0.36240.14710.4622 0.2223 0.6213 Transformer (enc.) 0.10020.3572 0.1476 0.46920.22720.6295 SwinV2 (enc.) 0.07460.29240.1191 0.40810.19290.5728Transformer (dec.) 0.0828 0.31830.12530.42440.19100.5615Bi-mLSTM (dec.) 0.0914 0.34840.13560.45260.20770.5935Ours 0.06920.26760.11020.37630.18020.5235

**Table 6.** Data efficiency evaluation.

across all training set sizes, achieving a CER of 0.0692 and WER of 0.2676 on the full dataset (100%), 0.1102 and 0.3763 on 50%, and 0.1802 and 0.5235 on 25%. These results highlight our model's ability to maintain high accuracy even with less training data. While ResNet and SwinV2 perform well on larger training sets, their accuracy drops more noticeably as the dataset size decreases. Other models, such as the Transformer and MLP-Mixer, show even larger performance gaps, emphasizing the robustness and efficiency of our model in making effective use of training data.

#### 4.5 Ablation Study

In this section, we conduct an ablation study on our WI dataset using CLDNN as the baseline, as it has a similar architecture to our model. We evaluate the performance improvements achieved through incremental changes, including optimized training strategies, architectural enhancements, hyperparameter tuning, and data augmentation.

Table 7 summarizes the ablation study results, showing performance improvements from incremental modifications. Starting from the CLDNN baseline with a CER of 0.1533 and WER of 0.5112, each enhancement improves accuracy. Scaling the CNN and BiLSTM reduces error rates at the cost of more parameters and MACs. Optimized training strategies, instance normalization, GELU activation, and dropout rate adjustment further boost performance without increasing computational overhead. Notably, the dilated-depth 1-D separable convolution improves accuracy while reducing parameters and MACs. Data augmentation techniques, such as add dropout and time warping, also play a key role, lowering CER from 0.0839 to 0.0745 (-11.2%) and WER from 0.3098 to 0.2810 (-9.3%), achieving significant gains despite limited room for improvement. With all enhancements and fine-tuned dropout, the final model achieves a CER of 0.0692 and a WER of 0.2676, demonstrating the effectiveness of these modifications. In

Table 7. Ablation study.

Models	CER	WER	#Params	MACs
CLDNN	0.1533	0.5112	0.75M	291M
+ Training strategy	0.1305	0.4500	0.75M	291M
+ Reverse dimension order	0.1261	0.4470	0.92M	214M
+ 3× deeper CNN	0.1070	0.3924	3.05M	989M
+ Standalone embedding layer	0.1064	0.3851	3.95M	687M
+ Dilated-depth separable convolution	0.0997	0.3630	2.85M	467M
+ Instance normalization	0.0960	0.3635	2.85M	465M
$+~{ m GELU}$	0.0934	0.3526	2.85M	465M
$+ 2 \times$ wider BiLSTM	0.0860	0.3203	3.52M	551M
+ 3-layer BiLSTM	0.0830	0.3048	3.91M	602M
+ Remove hidden layer	0.0839	0.3098	3.89M	600M
+ Data augmentation	0.0745	0.2810	3.89M	600M
+ Dropout rate 0.2	0.0692	0.2676	3.89M	600M

summary, our final model reduces CER by 55% and WER by 48% compared to the baseline CLDNN. To achieve this performance improvement, it uses approximately 5.2 times more parameters (increasing from 0.75M to 3.89M) and 2.1 times more MACs (rising from 291M to 600M).

## 5 Discussion

This section examines the key factors behind our model's strong performance, with a focus on robustness, data efficiency, and flexibility.

## 5.1 Robustness

Handwriting styles vary widely among individuals, creating significant challenges for HWR. Since it is impossible to train an HWR model on data that includes every handwriting style in the world, evaluating models on WI datasets provides a more realistic measure of performance than WD or random-split datasets. WI datasets ensure that the handwriting styles in the training set do not overlap with those in the test set, better simulating real-world scenarios where a model must generalize to unseen writers.

As children are beginners in handwriting, their handwriting styles differ from those of adults, making HWR more challenging. However, they should not be excluded as potential users of HWR systems. Therefore, developing a solution that works well for both adults and children is essential for the success of an HWR system.

Additionally, while noise is common in IMU data and poses a challenge for HWR, this issue can be mitigated by manually introducing noise during training to improve the model's resistance to noise.

#### 5.2 Data Efficiency

More data generally improves the performance of deep learning models, including HWR. However, supervising contributors and removing faulty samples, such as typographical mistakes, makes collecting handwriting IMU data time-consuming and costly. Therefore, the ability to extract features efficiently from smaller datasets is crucial for real-world applications. This can be achieved through data augmentation, regularization, normalization, and learning rate scheduling, which help models converge better and perform well with limited training data.

#### 5.3 Flexibility

In real-world deployment, handwriting inputs vary in size, posing challenges for models that require fixed input dimensions. Transformer-based models, such as Swin Transformer V2 and xLSTM, rely on predefined input sizes, requiring padding or compression to meet these constraints. However, padding increases computational overhead, while compression can lead to information loss. In contrast, the CNN-BiLSTM design processes inputs of any size without resizing, improving computational efficiency and adaptability, making it more suitable for real-world HWR.

## 6 Conclusion & Outlook

In conclusion, our experiments show that our model consistently outperforms competitors on WI datasets, demonstrating strong robustness, data efficiency, and flexibility in HWR. These strengths make it well-suited for real-world deployment.

However, our evaluation is limited by the dataset, as we have not compared performance on left- and right-handed data and have observed significant improvements with larger datasets. Additionally, we have not investigated whether there are patterns of errors that could provide insights for further reducing the error rate. We also have yet to explore performance on sentence-level handwriting, which better reflects real-world use and could offer valuable directions for future research.

Since HWR is closely related to language, incorporating natural language processing techniques, such as multimodal pretraining, could potentially encourage models to learn more semantic features and further enhance performance. Exploring these approaches should lead to even more robust and intelligent HWR systems.

#### References

Alemayoh, T.T., Shintani, M., Lee, J.H., Okamoto, S.: Deep-learning-based character recognition from handwriting motion data captured using imu and force sensors. Sensors 22(20) (2022). https://doi.org/10.3390/s22207840, https://www.mdpi.com/1424-8220/22/20/7840

- Beck, M., Pöppel, K., Spanring, M., Auer, A., Prudnikova, O., Kopp, M., Klambauer, G., Brandstetter, J., Hochreiter, S.: xlstm: Extended long short-term memory. In: Thirty-eighth Conference on Neural Information Processing Systems (2024), https://arxiv.org/abs/2405.04517
- Carbune, V., Gonnet, P., Deselaers, T., Rowley, H.A., Daryin, A., Calvo, M., Wang, L.L., Keysers, D., Feuz, S., Gervais, P.: Fast multi-language lstm-based online hand-writing recognition. International Journal on Document Analysis and Recognition (IJDAR) 23(2), 89–102 (Jun 2020). https://doi.org/10.1007/s10032-020-00350-4, https://doi.org/10.1007/s10032-020-00350-4
- Choi, S.d., Lee, A.S., Lee, S.y.: On-line handwritten character recognition with 3d accelerometer. In: 2006 IEEE International Conference on Information Acquisition. pp. 845–850 (2006). https://doi.org/10.1109/ICIA.2006.305842
- 5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=YicbFdNTTy
- Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning. pp. 369–376. ICML '06, Association for Computing Machinery, New York, NY, USA (2006). https://doi.org/10.1145/1143844.1143891, https://doi.org/10.1145/1143844.1143891
- Hassan, E., Tarek, H., Hazem, M., Bahnacy, S., Shaheen, L., Elashmwai, W.H.: Medical prescription recognition using machine learning. In: 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC). pp. 0973–0979 (2021). https://doi.org/10.1109/CCWC51732.2021.9376141
- 8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
- 9. Jeen-Shing, W., Yu-Liang, H., Cheng-Ling, C.: Online handwriting recognition using an accelerometer-based pen device. In: Proceedings of the 2nd International Conference on Advances in Computer Science and Engineering (CSE 2013). pp. 231–234. Atlantis Press (2013/07). https://doi.org/10.2991/cse.2013.52, https://doi.org/10.2991/cse.2013.52
- Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., Wei, F., Guo, B.: Swin transformer v2: Scaling up capacity and resolution. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
- 11. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
- 12. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
- 13. Lopez-Rodriguez, P., Avina-Cervantes, J.G., Contreras-Hernandez, J.L., Correa, R., Ruiz-Pinales, J.: Handwriting recognition based on 3d accelerometer data by deep learning. Applied Sciences 12(13) (2022). https://doi.org/10.3390/app12136707, https://www.mdpi.com/2076-3417/12/13/6707

- Meißl, F., Eibensteiner, F., Petz, P., Langer, J.: Online handwriting recognition using lstm on microcontroller and imu sensors. In: 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA). pp. 999–1004 (2022). https://doi.org/10.1109/ICMLA55696.2022.00167
- Meißl, F., Eibensteiner, F., Petz, P., Langer, J.: Online handwriting recognition using lstm on microcontroller and imu sensors. In: 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA). pp. 999–1004 (2022). https://doi.org/10.1109/ICMLA55696.2022.00167
- Ott, F., Rügamer, D., Heublein, L., Hamann, T., Barth, J., Bischl, B., Mutschler, C.: Benchmarking online sequence-to-sequence and character-based handwriting recognition from imu-enhanced pens. Int. J. Doc. Anal. Recognit. 25(4), 385–414 (Dec 2022). https://doi.org/10.1007/s10032-022-00415-6, https://doi.org/10.1007/s10032-022-00415-6
- 17. Pontart, V., Bidet-Ildei, C., Lambert, E., Morisset, P., Flouret, L., ALA-MARGOT, D.: Influence of handwriting skills during spelling in primary and lower secondary grades. Frontiers in Psychology 4 (2013). https://doi.org/10.3389/fpsyg.2013.00818, https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2013.00818
- Seuret, M., van der Loop, J., Weichselbaumer, N., Mayr, M., Molnar, J., Hass, T., Christlein, V.: Combining ocr models for reading early modern books. In: Fink, G.A., Jain, R., Kise, K., Zanibbi, R. (eds.) Document Analysis and Recognition -ICDAR 2023. pp. 342–357. Springer Nature Switzerland, Cham (2023)
- Tolstikhin, I.O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., Lucic, M., Dosovitskiy, A.: Mlp-mixer: An all-mlp architecture for vision. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems. vol. 34, pp. 24261–24272. Curran Associates, Inc. (2021)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
- Wehbi, M., Hamann, T., Barth, J., Eskofier, B.: Digitizing handwriting with a sensor pen: A writer-independent recognizer. In: 2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 295–300 (2020). https://doi.org/10.1109/ICFHR2020.2020.00061
- Wehbi, M., Hamann, T., Barth, J., Kaempf, P., Zanca, D., Eskofier, B.: Towards an imu-based pen online handwriting recognizer. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) Document Analysis and Recognition – ICDAR 2021. pp. 289–303. Springer International Publishing, Cham (2021)