

Transfer-Guided Hyperparameter Optimization for Hybrid Compact Convolutional Transformers in Handwritten Recognition

Muhammad Munsarif   [Universitas Muhammadiyah Semarang | m.munsarif@unimus.ac.id]

Richmond Ampah-Mensah  [Universitas Muhammadiyah Semarang | richmond.a.m@unimus.ac.id]

 Informatics, Universitas Muhammadiyah Semarang, Jl. Kedungmundu No.18, Kedungmundu, Kec. Tembalang, Kota Semarang, Jawa Tengah 50273, Indonesia.

Received: 29 June 2026 • Accepted: 08 September 2026 • Published: 21 September 2026

Abstract. Handwritten recognition remains sensitive to model architecture and hyperparameter selection. This study proposes a two-stage transfer-guided strategy in which CNN genetic algorithm (CNN-GA) hyperparameters are selected first and then transferred to Compact Convolutional Transformer (CCT), CCT-LSTM, CCT-biLSTM, and CCT-GRU models. The models are evaluated with 2 and 7 transformer encoder layers on an English handwriting dataset derived from NIST Special Database 19 and on MNIST. The reported CNN-GA baseline reaches 97.77% on English handwriting, while CCT-based models reach 98.96%–99.07% with 2 layers and 99.46%–99.63% with 7 layers. CCT-GRU gives the highest reported 2-layer English result at 99.07%, and CCT-LSTM gives the highest reported 7-layer result at 99.63%. On MNIST, plain CCT reaches the highest reported result, 99.55% with 7 layers. The findings show that hyperparameter transfer is useful, but recurrent hybridization remains dataset- and depth-dependent.

Keywords: Handwritten recognition, Compact Convolutional Transformer, Hyperparameter transfer, Genetic algorithm, CCT-LSTM, CCT-GRU

1 Introduction

Handwritten recognition has been a persistent challenge in document analysis and computer vision. It facilitates document digitization, form processing, educational technology, postal automation, banking applications, and historical archive analysis [Garrido-Munoz *et al.*, 2025]. The task is challenging due to the variability in stroke orders, character widths, slants, ligatures, and regional distortions for the same character [Agrawal *et al.*, 2024a]. The problem is also sensitive to preprocessing and segmentation because small boundary errors can change the visual evidence available to the classifier [Berriche *et al.*, 2024].

Convolutional Neural Networks (CNNs) significantly advanced contemporary handwritten identification capabilities by directly learning local stroke characteristics from image pixels. MNIST and EMNIST remain common benchmark families for handwritten recognition, while NIST-derived English handwriting datasets extend the evaluation landscape beyond digits by including handprinted letters and more challenging class variation [Cohen *et al.*, 2017; Baldominos *et al.*, 2019; Munsarif *et al.*, 2023]. Survey evidence indicates that MNIST has nearly reached saturation for numerous contemporary neural models, whereas more complex letter-based datasets continue to provide greater discriminative power for model comparison [Baldominos *et al.*, 2019].

Despite these advances, handwritten recognition is still affected by a practical model-design problem: high recognition accuracy depends on the choice of hyperparameters. The learning rate, batch size, optimizer, weight decay, ker-

nel size, activation function, number of convolutional layers, and number of training epochs can all influence convergence and generalization. Manual tuning is slow and difficult to reproduce. Exhaustive search is expensive. Evolutionary search can explore richer design spaces, but repeated training of candidate neural networks is computationally costly [Zhan *et al.*, 2022].

Vision transformers introduce another layer of opportunity and difficulty. The original Vision Transformer (ViT) showed that image classification can be framed as sequence modeling over image patches [Dosovitskiy *et al.*, 2020]. However, ViT-style models generally require careful training and substantial data. Compact Convolutional Transformer (CCT) was proposed to make transformer-style image classification more suitable for smaller datasets by replacing plain patch tokenization with convolutional tokenization and sequence pooling [Hassani *et al.*, 2021]. This is appealing for handwriting as characters are constructed from local strokes, loops, endpoints, and boundary patterns that convolutional tokenization may maintain prior to transformer encoding.

Recent advances in handwritten text recognition have shown the efficiency of hybrid CNN-transformer architectures. HTR-VT, for example, uses a CNN feature extractor to improve data-efficient transformer recognition on handwritten text lines [Li *et al.*, 2025]. Hybrid CNN-transformer models for handwritten digits have also been reported in recent literature [Agrawal *et al.*, 2024b]. These studies suggest that combining local convolutional features with attention-based representation learning is a promising design direction. What remains less clearly addressed is how to choose a reli-

able hyperparameter configuration for compact transformer and hybrid compact transformer models when full transformer tuning is expensive.

This study addresses that gap through a two-stage transfer-guided hyperparameter optimization strategy. The first stage uses CNN-GA optimization to identify an effective hyperparameter set. The second stage transfers the selected hyperparameters to CCT and hybrid CCT-deep learning models. The primary concept is not that CNN and transformer architectures are identical. Rather, it is that the convolutional tokenization and training process in CCT can use hyperparameters already validated on the same handwriting domain. This gives the transformer stage a disciplined starting configuration; the present experiment does not directly measure any reduction in target-model search cost.

The study makes four contributions:

1. It proposes a two-stage hyperparameter transfer strategy from CNN-GA optimization to compact transformer-based handwritten recognition.
2. It evaluates four architectures under the same transferred configuration: CCT, CCT-LSTM, CCT-biLSTM, and CCT-GRU.
3. It analyzes the effect of transformer depth by comparing two-layer and seven-layer encoder settings on English handwriting and MNIST.
4. It provides practical evidence that recurrent hybrid heads can improve CCT in some handwriting settings, but plain CCT remains stronger on the simpler MNIST digit task.

The article is structured as follows. Section 2 reviews relevant literature on handwritten recognition, compact transformers, hyperparameter optimization, and hybrid recurrent heads. Section 3 describes the proposed transfer-guided methodology. Section 4 specifies the experimental setup. Section 5 provides the quantitative and graphical results. Section 6 discusses the implications, limitations, and validity considerations. Section 7 concludes the research.

2 Related Work

2.1 Handwritten Recognition Benchmarks

MNIST remains one of the most recognizable handwritten digit benchmarks because it offers a standardized evaluation setting and a compact grayscale image format [Baldominos *et al.*, 2019]. Its value is historical and methodological: it enabled reproducible comparisons across many classifiers. However, the benchmark is also relatively simple for modern deep learning models. Prior survey evidence observed that MNIST and EMNIST performance is already close to 100% for strong methods, which makes incremental gains harder to interpret without careful analysis [Baldominos *et al.*, 2019].

NIST-derived handwriting resources provide broader sources of handprinted forms and isolated character images than digit-only benchmarks. EMNIST was derived from the NIST source material to create MNIST-style tasks that include letters [Cohen *et al.*, 2017]. The English handwriting dataset used in this study follows this broader character-recognition direction. In the earlier CNN-GA stage, the data were reported as 372,450 NIST-derived English handwriting records,

balanced through under-sampling, converted to grayscale, and split into training and testing with an 80:20 proportion [Munsarif *et al.*, 2023]. This dataset is more informative for the present study than MNIST alone because it includes richer character variation.

2.2 CNN Hyperparameter Optimization

The performance of CNNs is influenced by the interplay between architecture and training hyperparameters. A CNN with a poor learning rate, optimizer, kernel size, or layer configuration can underfit, converge slowly, or overfit even when the basic architecture is appropriate [Raiaan *et al.*, 2024]. Evolutionary algorithms are often used to search this space because they can encode candidate architectures and training choices as chromosomes, evaluate them through model fitness, and update populations through selection, crossover, and mutation.

Earlier work connected to the present study investigated Genetic Algorithm (GA) search for English handwritten recognition and showed that GA-based hyperparameter optimization can improve CNN model construction [Munsarif *et al.*, 2023]. A later CNN-GA evaluation expanded the search to 20 hyperparameters and found that the EX3 configuration was stronger than broader configurations for the English handwriting dataset [Munsarif *et al.*, 2023]. This finding is important because it indicates that a larger hyperparameter search space does not automatically produce a better image-recognition model.

Diversity-guided GA has also been proposed to reduce premature convergence in CNN optimization by controlling population diversity through adaptive parameter control and random injection [Fatyanosa and Aritsugi, 2021]. However, the model investigated here does not simply repeat CNN optimization. It uses the best CNN-GA configuration as a transferred design prior for compact transformer models. This changes the role of the optimization stage from final CNN selection to cross-architecture guidance [Sun *et al.*, 2020; Qian *et al.*, 2023; Chitty-Venkata *et al.*, 2022].

2.3 Vision Transformers and Compact Convolutional Transformers

ViT demonstrated that image classification can be performed by dividing an image into patches and processing the resulting patch sequence with a transformer encoder [Dosovitskiy *et al.*, 2020]. This design gives the model access to global self-attention, but the original formulation can be data-intensive. CCT addresses that issue by using convolutional tokenization, compact model structure, and sequence pooling [Hassani *et al.*, 2021]. Convolutional tokenization is especially relevant to handwriting because it can retain local stroke information before the transformer encoder models longer-range spatial relations.

The present work builds on this CCT design principle. Instead of replacing convolution with attention, it uses convolution as the tokenization and feature-extraction front end, then transfers CNN-GA hyperparameters into that stage. The transformer encoder then provides global representation learn-

ing over the tokenized character image [Dosovitskiy *et al.*, 2020; Hassani *et al.*, 2021].

2.4 Hybrid CCT with Recurrent Classifier Heads

Recurrent neural networks have long been used for sequence modeling. LSTM heads address long-term dependency learning through memory cells and gating mechanisms, while GRU subsequently streamlined recurrent gating via reset and update gates while preserving robust sequence modeling capabilities [Cho *et al.*, 2014]. In image recognition pipelines, recurrent heads may be employed after visual feature extraction when the visual representation is organized in a sequence. Research on hybrid CNN-GRU handwriting models indicates that GRU can substitute fully connected layers to enhance efficiency in handwritten digit recognition [Nguyen *et al.*, 2019].

This study assesses the efficacy of the same principle following CCT representation learning. Three hybrid models are examined: CCT-LSTM, CCT-biLSTM, and CCT-GRU. The comparison is important because recurrent heads add modeling capacity but also add complexity. The results therefore need to show not only which model is highest, but also when hybridization is useful.

2.5 Hyperparameter Transfer

Recent large-scale learning research has shown that hyperparameter transfer can reduce tuning cost when stable hyperparameters are learned on a proxy model and transferred to a larger target model. The μ Transfer framework, for example, transfers hyperparameters from smaller networks to larger networks under a particular parameterization [Yang *et al.*, 2022]. Depthwise hyperparameter-transfer research further studies transfer across depth and width for residual networks and vision transformers [Bordelon *et al.*, 2023].

The method in this paper is different in scope. It is an empirical transfer strategy for handwritten recognition, not a theoretical width-depth parameterization framework. The common motivation is the same: direct hyperparameter tuning is expensive, and useful hyperparameter structure can sometimes be transferred to reduce the search burden. Here, the source model is CNN-GA and the target models are CCT-based handwriting recognizers.

The present experiment does not include a CCT trained under a non-transferred configuration. Therefore, the study evaluates whether the transferred configuration can support high-performing CCT-based models, but it does not isolate the causal effect of hyperparameter transfer itself.

3 Proposed Method

3.1 Study Overview

The proposed strategy has two stages. Stage 1 identifies effective CNN hyperparameters using GA-based search on the handwriting domain. In this stage, CNN architecture and training candidates are encoded as chromosomes, initialized as a population, evaluated through CNN validation

accuracy as the fitness function, and updated through selection, crossover, and mutation [Sun *et al.*, 2020]. The available experiment record reports a crossover rate of 0.8, a mutation rate of 0.2, and 25 populations/generations in the CNN-GA stage [Munsarif *et al.*, 2023]. Stage 2 transfers the chosen hyperparameters to CCT and hybrid CCT-deep learning models [Hassani *et al.*, 2021]. The transferred values are used for convolutional tokenisation and learning, while the target architectures differ in the classifier component and transformer depth [Qian *et al.*, 2023].

Figure 1 provides the visual overview of the proposed method. To make the procedure explicit, Algorithm 1 summarizes the transfer-guided optimization process.

Algorithm 1. Transfer-guided hyperparameter optimization procedure.

1. Define the source CNN-GA search space and target CCT-based architectures.
2. Run the CNN-GA search on the handwriting domain and select the best hyperparameter set H^* from the available experiment record.
3. Transfer H^* to the convolutional tokenization and training configuration of the CCT-based target models.
4. Train CCT, CCT-LSTM, CCT-biLSTM, and CCT-GRU with two and seven transformer encoder layers under the same transferred configuration.
5. Evaluate the trained models using the reported training, validation, and test accuracy measures.

The transferred set H^* consists of the learning rate, weight decay, batch size, optimizer, kernel size, activation function, and number of convolution layers. The number of training epochs is not treated as a transferred architecture or optimizer hyperparameter in this article. Instead, the available experiment record used a fixed 74-epoch training budget for the target CCT-based models. The selected numeric values are shown in Table 1.

Table 1. Transferred hyperparameters used in the CCT-based models.

Category	Hyperparameter	Transferred value
Global	Learning rate	3.15647×10^{-3}
	Weight decay	0.0001
	Batch size	249
	Optimizer	Nadam
Layer	Kernel size	4×4
	Activation function	ReLU
Architecture	Number of convolution layers	4

The learning rate in Table 1 is rounded for readability from the GA-selected value 0.00315647054032801. The implementation-level value was preserved from the selected chromosome, but reporting it at full floating-point precision would imply a degree of scientific certainty that is not warranted by the experiment. Nadam was retained from the

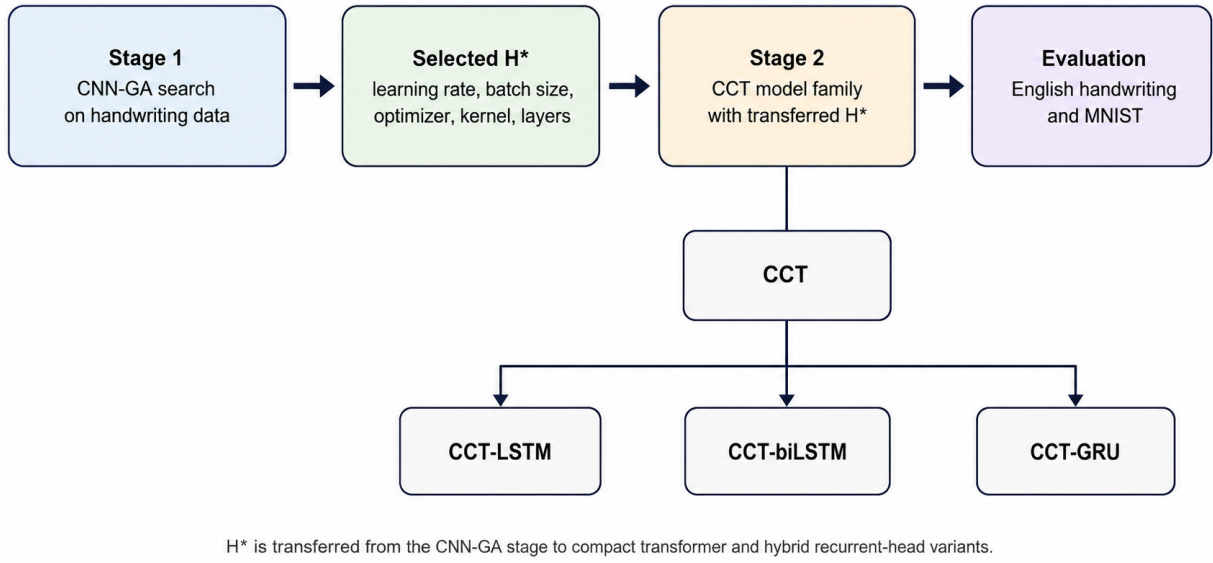


Figure 1. Two-stage transfer-guided hyperparameter optimization framework. Stage 1 identifies the CNN-GA source configuration; Stage 2 transfers compatible settings to CCT, CCT-LSTM, CCT-biLSTM, and CCT-GRU, which are evaluated with two and seven transformer encoder layers on English handwriting and MNIST. The workflow depicts the transfer procedure but does not by itself establish superiority over non-transferred CCT tuning.

selected configuration because it combines adaptive moment estimation with Nesterov momentum. The CCT tokenization stage follows the source CCT formulation with ReLU activation [Hassani *et al.*, 2021]. The study does not tune a new optimizer or activation function in the transformer stage. This preserves the transfer design and allows the comparison to focus on architecture and transformer depth.

3.2 CCT with Transferred Hyperparameters

The baseline target architecture is CCT. The model first applies convolutional tokenization to the input image. In this study, the tokenization stage uses the transferred convolutional design: four convolution layers, 4×4 kernels, and ReLU activation. The resulting representation is enriched with positional information and passed to the transformer encoder. Sequence pooling is then used before final classification [Hassani *et al.*, 2021].

$$x_0 = \text{MaxPool}(\text{ReLU}(\text{Conv2d}(x))) \quad (1)$$

$$x^L = f(x^0) \in \mathbb{R}^{b \times n \times d} \quad (2)$$

$$x^{L'} = \text{softmax}(g(x^L)^T) \in \mathbb{R}^{b \times 1 \times d} \quad (3)$$

$$z = x^{L'} x^L = \text{softmax}(g(x^L)^T) \times x^L \in \mathbb{R}^{b \times 1 \times d} \quad (4)$$

The CCT model is the most direct test of whether CNN-GA hyperparameters can support compact transformer design. It retains convolutional inductive bias while adding transformer-based global representation learning [Dosovitskiy *et al.*, 2020].

3.3 Hybrid CCT-LSTM

CCT-LSTM extends the CCT architecture by integrating an LSTM module in the encoded-token pathway before sequence pooling. The motivation is that the transformer output can be interpreted as a sequence of learned visual tokens. LSTM can then model sequential dependencies across this representation before classification.

This design tests whether memory-based sequence modeling improves the CCT representation for handwritten character recognition.

3.4 Hybrid CCT-biLSTM

CCT-biLSTM uses a bidirectional LSTM module in the encoded-token pathway. The bidirectional structure processes the token sequence in forward and backward directions. This may help when classification depends on spatial dependencies that are not naturally one-directional.

The inclusion of CCT-biLSTM is important because bidirectional recurrence is often assumed to be more expressive than unidirectional LSTM. The empirical results test whether that added expressiveness is beneficial in this specific handwriting setting.

3.5 Hybrid CCT-GRU

CCT-GRU integrates a GRU module in the encoded-token pathway before sequence pooling. GRU uses a simpler gating structure than LSTM and may provide a more compact recurrent classifier. This is useful when the goal is to improve sequence modeling without adding unnecessary complexity.

The CCT-GRU comparison is especially relevant because prior handwritten digit recognition research has used CNN-GRU models to reduce reliance on fully connected layers

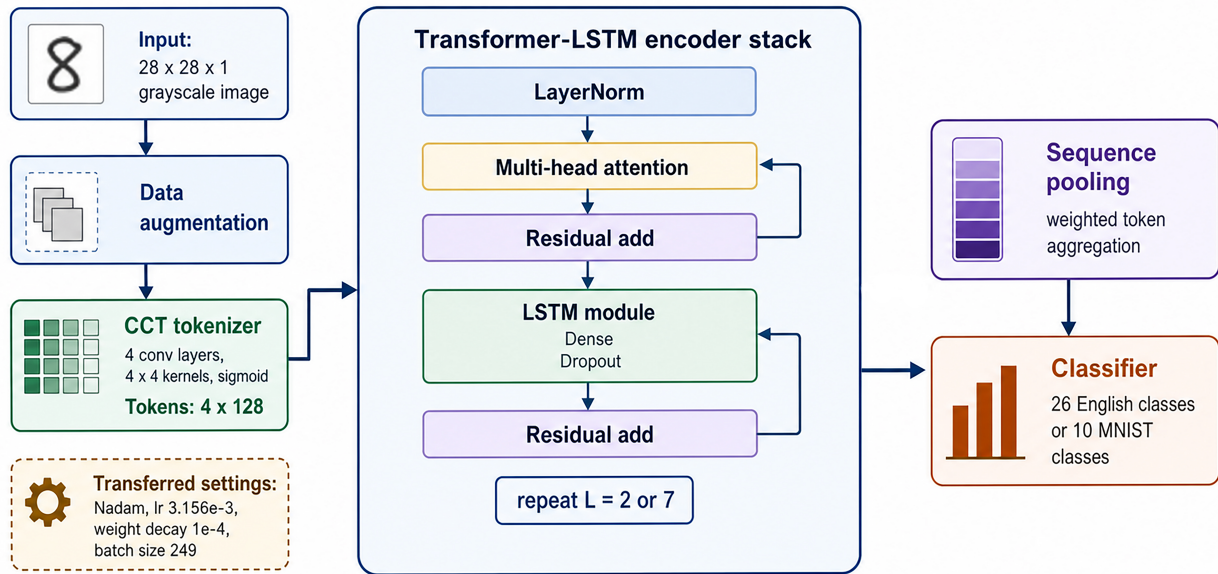


Figure 2. CCT-LSTM architecture with transferred convolutional and training hyperparameters. The LSTM module operates in the encoded-token pathway before sequence pooling and final classification.

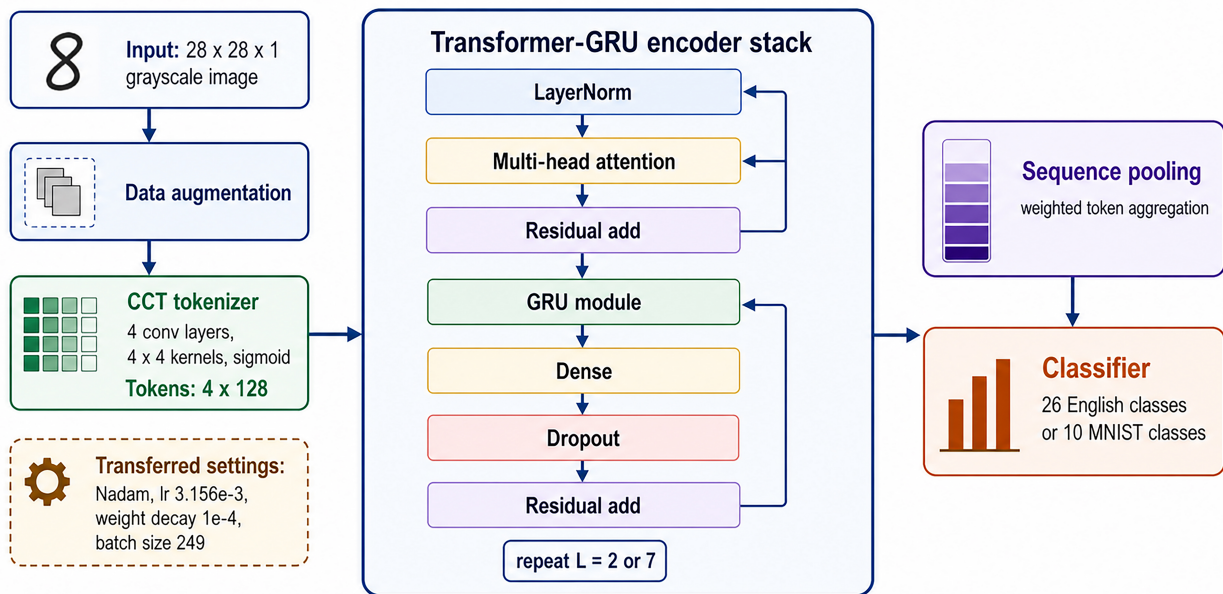


Figure 3. CCT-GRU architecture with transferred convolutional and training hyperparameters. The GRU module operates in the encoded-token pathway before sequence pooling and final classification.

[Nguyen *et al.*, 2019].

4 Experimental Setup

4.1 Datasets

The first evaluation dataset is an English handwriting dataset derived from NIST Special Database 19. The earlier CNN-GA stage reported 372,450 records, class imbalance across alphabet classes, under-sampling to balance the data, grayscale conversion, and an 80:20 training-testing split [Munsarif *et al.*, 2023]. The available record does not specify whether the split was writer-independent or writer-dependent. This point is therefore treated as a validity limitation rather than assumed. In the present CCT-based evaluation, training, validation, and test accuracy are reported for the English handwriting task.

For the Stage 2 CCT evaluation, the archived experiment record separately reports 70,000 grayscale character images, with 50,000 training and 20,000 testing samples, while also describing the allocation as 80:20. Because those statements are not arithmetically consistent, they are reported without reconciliation. The exact validation allocation, writer identifiers, and random split seed are not preserved.

The second dataset is MNIST. MNIST is used as a benchmark handwritten digit task and provides a useful comparison point because its recognition difficulty is lower than letter-based handwriting datasets for modern deep models [Baldominos *et al.*, 2019]. For MNIST, training and test accuracy are reported.

The study also includes private handwritten samples as a qualitative deployment check. These samples are not used as the basis for the main quantitative claims. The archived experiment record contains 100 private samples processed using contour segmentation, bounding boxes, cropping, resizing to 18×18 , and model-based character recognition; complete writer metadata and sample-level ground truth are not preserved. They are used to show how the trained models behave when character segmentation and word-level recognition are applied to practical handwritten inputs.

4.2 Evaluation Design

The study evaluates four architectures:

1. CCT.
2. CCT-LSTM.
3. CCT-biLSTM.
4. CCT-GRU.

Each architecture is evaluated with two and seven transformer encoder layers. The two-layer setting represents the shallower transformer configuration, while the seven-layer setting tests whether additional transformer depth improves representation quality under the same transferred hyperparameters. These are the two depth settings retained in the available experiment record; neither setting is claimed to be independently optimal.

Accuracy is reported in percent. For the English handwriting dataset, the reported metrics are training accuracy, validation accuracy, and test accuracy. For MNIST, the reported metrics are training accuracy and test accuracy. The

target CCT-based models were trained with the fixed 74-epoch budget available in the experiment record. No early-stopping evidence is available, so the results are interpreted as fixed-budget comparisons rather than natural-convergence comparisons. The main comparison is based on test accuracy because it reflects generalization more directly than training accuracy.

Several implementation details required for exact independent reproduction are not preserved in the available experiment record, including the complete convolutional channel progression, strides and padding, number of attention heads, feed-forward dimensions, recurrent hidden dimensions, complete dropout settings, loss-function implementation, initialization, learning-rate scheduling, checkpoint-selection procedure, software versions, and random seeds. These values are not inferred from a generic implementation.

The available record also does not report a non-transferred CCT control, a target-specific hyperparameter-search budget, Stage 2 wall-clock training time, GPU-hours, parameter count, FLOPs, inference latency, memory usage, or energy consumption. Therefore, the present experiment does not quantify a computational saving attributable to transfer.

4.3 Visual Evidence

The visual evidence is intentionally selective. The main paper uses six figure groups: the proposed two-stage framework, the CCT-LSTM architecture diagram, the CCT-GRU architecture diagram, paired training-performance panels for the strongest hybrid settings, a test-accuracy summary chart, and one private-sample recognition demonstration. This keeps the figures focused on the method, the strongest comparative evidence, and the practical recognition behavior.

5 Results

5.1 Training Behavior

The training evidence is summarized through the two strongest hybrid settings. The CCT-LSTM panels are retained because CCT-LSTM is the strongest seven-layer model on the English handwriting dataset. The CCT-GRU panels are retained because CCT-GRU is the strongest two-layer model on the same dataset. The paired training panels are therefore presented as one manuscript figure rather than as separate figures.

5.2 English Handwriting Results with two Transformer Layers

Table 2 reports the English handwriting results for the two-layer transformer setting.

The Stage 1 CNN-GA baseline reports 97.77% accuracy. In the two-layer CCT-based setting, all target models exceed the reported baseline, with test accuracies ranging from 98.96% to 99.07%. The differences among target models are narrow. CCT-GRU achieves the highest reported test accuracy at 99.07%. CCT-LSTM achieves the highest training and validation accuracy, but its test accuracy is 0.03 percentage

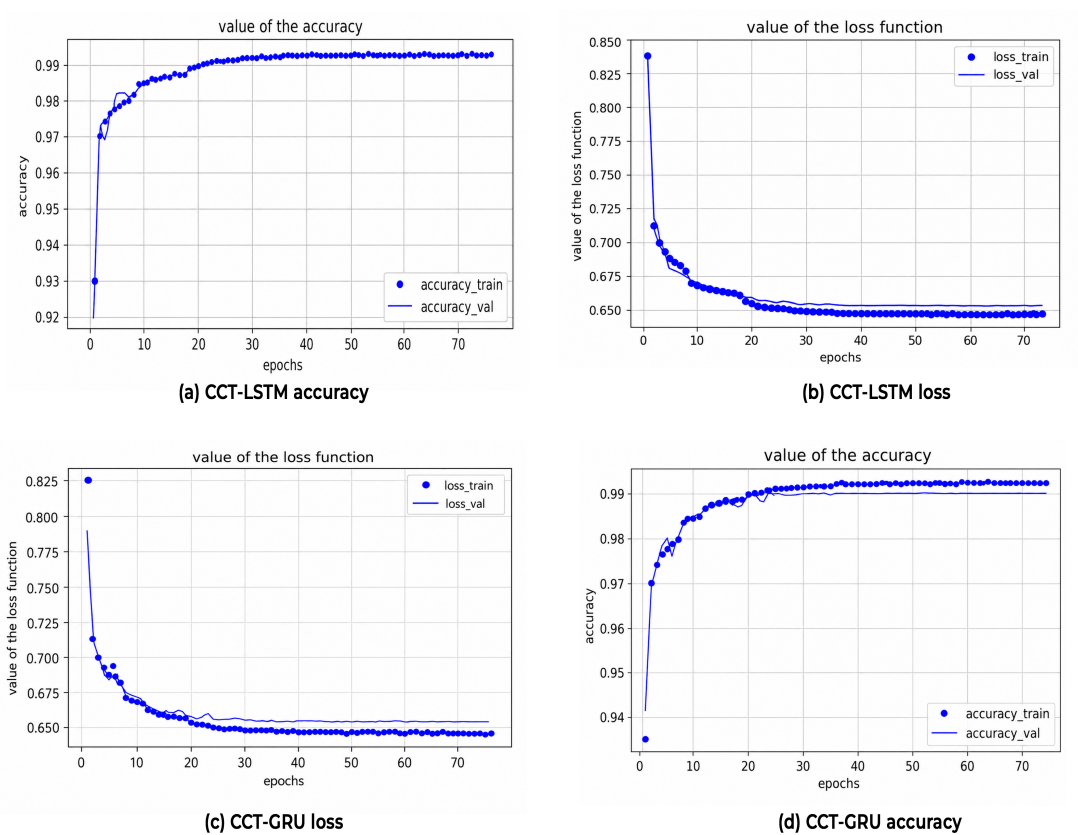


Figure 4. Paired training-performance panels for CCT-LSTM and CCT-GRU. Panels (a) and (b) show the CCT-LSTM accuracy and loss behavior. Panels (c) and (d) show the CCT-GRU loss and accuracy behavior.

Table 2. English handwriting recognition results with two transformer layers. N/R means not reported in the available Stage 1 CNN-GA baseline record.

Model	Train (%)	Validation (%)	Test (%)
Stage 1 CNN-GA			
baseline (EX3)	N/R	N/R	97.77
CCT	99.41	99.10	98.96
CCT-LSTM	99.46	99.13	99.04
CCT-biLSTM	99.34	98.99	98.98
CCT-GRU	99.40	99.04	99.07

points lower than CCT-GRU. Plain CCT reaches 98.96%, while CCT-biLSTM reaches 98.98%.

The result suggests that recurrent hybridization can be useful in the shallow transformer setting, but the specific recurrent unit matters. The GRU head gives the highest reported test performance. Because only one endpoint per configuration is available, the 0.03 percentage-point difference between CCT-GRU and CCT-LSTM should be read as an empirical ordering rather than statistical proof of architecture superiority.

5.3 English Handwriting Results with seven Transformer Layers

Table 3 reports the English handwriting results for the seven-layer transformer setting.

Table 3. English handwriting recognition results with seven transformer layers. N/R means not reported in the available Stage 1 CNN-GA baseline record.

Model	Train (%)	Validation (%)	Test (%)
Stage 1 CNN-GA			
baseline (EX3)	N/R	N/R	97.77
CCT	99.81	99.69	99.46
CCT-LSTM	99.86	99.72	99.63
CCT-biLSTM	99.74	99.59	99.47
CCT-GRU	99.80	99.64	99.57

With seven transformer layers, every target model also exceeds the reported Stage 1 CNN-GA baseline. CCT-LSTM gives the highest reported training, validation, and test accuracy. Its test accuracy reaches 99.63%, which is 0.17 percentage points higher than plain CCT and 0.06 percentage points higher than CCT-GRU. CCT-biLSTM improves over its two-layer result but remains close to plain CCT.

The result indicates that increasing transformer depth changes the relative value of the recurrent head. GRU gives the highest reported result in the shallow setting, while LSTM gives the highest reported result when the transformer encoder is deeper. This is important because it shows that architecture selection cannot be reduced to a single recurrent-unit preference. The most effective head depends on the feature representation produced by the transformer encoder. The 0.06 percentage-point CCT-LSTM/CCT-GRU difference is descriptive because repeated-seed variance is unavailable.

5.4 Depth Effect on English Handwriting

Table 4 compares the two-layer and seven-layer English handwriting test results.

Table 4. Effect of increasing transformer depth on English handwriting test accuracy.

Model	Two-layer (%)	Seven-layer (%)	Gain
CCT	98.96	99.46	+0.50
CCT-LSTM	99.04	99.63	+0.59
CCT-biLSTM	98.98	99.47	+0.49
CCT-GRU	99.07	99.57	+0.50

All four models improve when transformer depth increases from two to seven layers. The largest reported gain is observed for CCT-LSTM, which improves by 0.59 percentage points. The remaining models improve by approximately half a percentage point. The consistency of the reported gains shows a positive recorded depth difference for the English handwriting task under the transferred hyperparameter setting.

5.5 MNIST Results

Table 5 reports MNIST performance with two transformer layers. A Stage 1 CNN-GA baseline for MNIST under the same transferred configuration is not available in the experiment record, so the MNIST tables compare only the CCT-based target models.

Table 5. MNIST recognition results with two transformer layers.

Model	Train (%)	Test (%)
CCT	99.93	99.44
CCT-LSTM	99.93	99.34
CCT-biLSTM	99.92	99.28
CCT-GRU	99.93	99.37

Plain CCT obtains the highest reported two-layer MNIST test accuracy at 99.44%. CCT-GRU is the highest reported hybrid model at 99.37%, followed by CCT-LSTM at 99.34% and CCT-biLSTM at 99.28%. This differs from the English handwriting result, where a recurrent hybrid head improves the highest reported two-layer test accuracy. Table 6 reports MNIST performance with seven transformer layers.

Table 6. MNIST recognition results with seven transformer layers.

Model	Train (%)	Test (%)
CCT	99.95	99.55
CCT-LSTM	99.96	99.45
CCT-biLSTM	99.95	99.39
CCT-GRU	99.96	99.48

Plain CCT remains the highest reported MNIST model with 99.55% test accuracy. CCT-GRU again gives the highest reported hybrid result at 99.48%. CCT-LSTM reaches 99.45%, and CCT-biLSTM reaches 99.39%. Table 7 summarizes the MNIST depth effect.

Table 7. Effect of increasing transformer depth on MNIST test accuracy.

Model	Two-layer (%)	Seven-layer (%)	Gain
CCT	99.44	99.55	+0.11
CCT-LSTM	99.34	99.45	+0.11
CCT-biLSTM	99.28	99.39	+0.11
CCT-GRU	99.37	99.48	+0.11

All MNIST models improve by the same reported amount of 0.11 percentage points when depth increases from two to seven layers. The gain is smaller than on English handwriting, which is consistent with the lower complexity and stronger saturation of MNIST for modern deep models [Baldominos *et al.*, 2019].

5.6 Test-Accuracy Summary

Figure 5 summarizes the main test-accuracy patterns across architecture and transformer depth. It reinforces the tabular finding that deeper transformer settings improve all reported target models, while the leading architecture differs between English handwriting and MNIST.

5.7 Private Handwriting Demonstration

The private handwriting demonstration applies the trained models to word-level handwritten inputs. The process uses contour segmentation to separate foreground objects from background, bounding boxes to isolate characters, cropping, resizing to a common input size, and model-based character recognition. The qualitative recognition results are shown in Figure 6.

The private examples demonstrate that the trained models can be integrated into a practical recognition pipeline. They also reveal a realistic deployment issue: word-level recognition depends on segmentation quality and character-level consistency. Some word outputs are visually close but not exact, which is expected when character segmentation is applied to unconstrained handwritten words. For that reason, the private-sample evidence is treated as a qualitative demonstration rather than as a substitute for public-dataset testing.

6 Discussion and Validity Considerations

The results show that the transferred configuration can support high-performing CCT-based handwritten recognition; however, they do not isolate the causal effect of transfer. This interpretation is consistent with prior hyperparameter-transfer research, where a tuned configuration from one setting can reduce the search burden in another related model setting [Yang *et al.*, 2022; Bordelon *et al.*, 2023]. The English handwriting comparison provides the most direct evidence: the reported Stage 1 CNN-GA EX3 baseline is 97.77%, while the CCT-based target models report 98.96%–99.07% with two transformer layers and 99.46%–99.63% with seven transformer layers [Munsarif *et al.*, 2023]. The transformer-stage models therefore attain higher point accuracies than the reported

CNN-GA baseline, but the comparison does not establish that transfer itself causes the difference. The comparison should still be interpreted cautiously because the available CNN-GA baseline record does not provide the same training, validation, and test breakdown reported for the CCT-based models.

The method is coherent with the structure of CCT because CCT begins with convolutional tokenization rather than pure patch embedding [Hassani *et al.*, 2021]. Convolutional hyperparameters such as kernel size, activation function, and number of convolution layers therefore remain relevant when transferred from the optimized CNN source model. The most consistent empirical pattern is the depth effect: increasing transformer depth from two to seven layers improves every reported target-model test score. The gain is larger on English handwriting, where character variation is richer, than on MNIST, which is closer to saturation for modern neural models [Baldominos *et al.*, 2019].

Recurrent hybridization is dataset-dependent. On English handwriting, CCT-GRU gives the highest reported result with two transformer layers, while CCT-LSTM gives the highest reported result with seven transformer layers. A plausible interpretation is that GRU adds lower-overhead recurrence to shallow transformer features through reset and update gates [Cho *et al.*, 2014], while the deeper seven-layer transformer produces a richer hierarchical token representation that may benefit from the higher memory capacity and stricter gating of LSTM. MNIST does not show the same pattern: plain CCT gives the highest reported result under both depth settings, suggesting that the digit task is already well represented by the CCT encoder without an additional recurrent head [Baldominos *et al.*, 2019; Hassani *et al.*, 2021].

CCT-biLSTM does not become the leading model despite its bidirectional recurrence. This suggests that processing the token sequence in both directions does not automatically improve CCT-based handwriting classification. The CCT encoder and sequence pooling may already provide a sufficiently global representation [Hassani *et al.*, 2021], making bidirectional recurrence partly redundant. Another possibility is that the fixed transferred hyperparameters are better matched to the simpler recurrent heads. Additional model-size, latency, and repeated-run analysis would be required to separate these explanations.

These findings lead to a balanced model-selection message. CCT is a strong baseline when architectural simplicity and accuracy are both important, consistent with the motivation of compact transformer design for smaller image datasets [Hassani *et al.*, 2021]. CCT-GRU is a reasonable candidate for shallow transformer configurations on more complex handwriting data, while CCT-LSTM is more effective in the reported seven-layer English handwriting setting. More broadly, the two-stage strategy uses a domain-validated CNN-GA configuration as a controlled starting point for CCT-based models [Munsarif *et al.*, 2023; Hassani *et al.*, 2021]; the present experiment does not quantify a reduction in search cost or GPU-hours.

Several validity considerations remain. The English handwriting split is documented as 80:20, while the Stage 2 record also reports 50,000 training and 20,000 testing samples; these statements are not arithmetically consistent. The available record also does not specify whether the split is

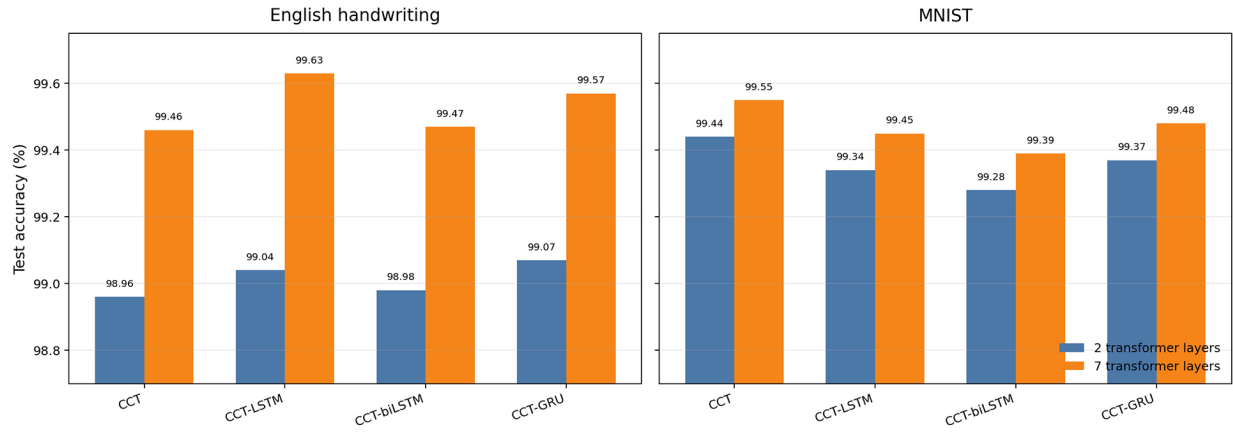


Figure 5. Test accuracy by architecture and transformer depth for the CCT-based target models. Exact values are reported in Tables 2–7; no uncertainty bars are shown because repeated runs were not retained.

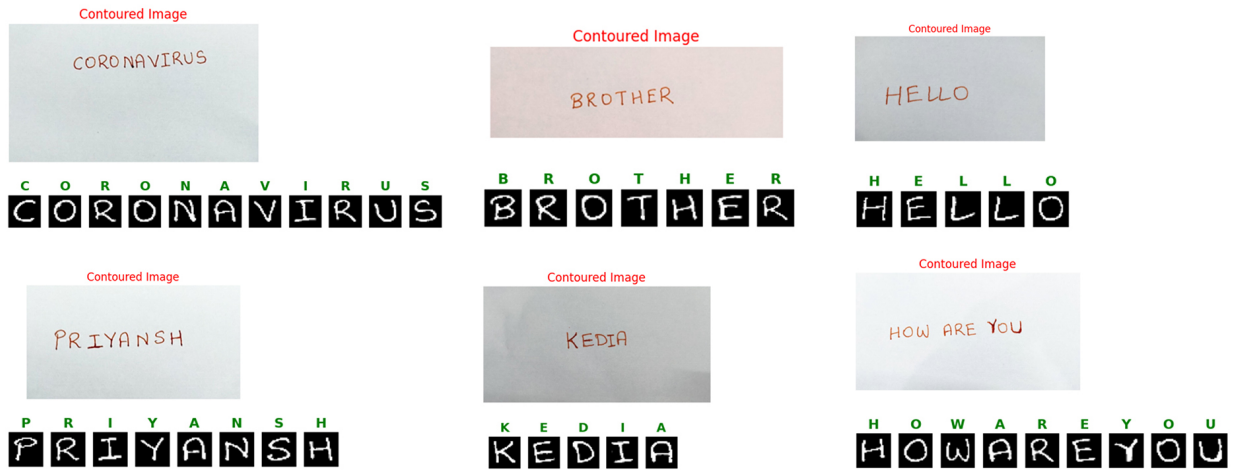


Figure 6. Private handwritten-word recognition examples using the proposed models. The examples illustrate the segmentation-recognition pipeline and are not used as a quantitative external-validation set.

writer-independent. This matters because NIST-derived handwriting data contain writer-level variation [Cohen *et al.*, 2017; Munsarif *et al.*, 2023]. If the split is writer-dependent, samples from the same writer may appear in both training and testing partitions, which can inflate recognition accuracy through writer-style memorization. The reported results should therefore be interpreted as dataset-level recognition performance under the documented split, not as confirmed writer-independent generalization.

The comparison focuses on accuracy because that is the metric available across the reported experiments. A fuller model-selection analysis would require parameter count, FLOPs, training time, inference latency, memory usage, and energy consumption. Therefore, the term Compact Convolutional Transformer is used as the architecture name from the CCT literature [Hassani *et al.*, 2021], not as a measured efficiency claim in this study. The available experimental record also reports point accuracies rather than repeated runs with different random seeds or standard deviations. Narrow differences such as 0.03% or 0.06% should therefore be read as empirical ordering, not statistical proof of architecture superiority.

Finally, the 74-epoch value should be interpreted as a fixed training budget available in the reported experiment record. Because early-stopping evidence is not available, the study does not claim that 74 epochs is the optimal training duration for each CCT-based architecture. The private handwritten demonstration is also qualitative; it illustrates practical segmentation-recognition behavior but does not replace controlled public-dataset testing. The main claim is therefore deliberately bounded: CNN-GA-selected hyperparameters can guide effective CCT-based model construction and enable a controlled comparison of architecture and transformer depth under the available experimental record.

7 Conclusion

This study presented a two-stage transfer-guided hyperparameter optimization strategy for handwritten recognition. A CNN-GA optimization stage identified a compact set of effective hyperparameters, and those values were transferred into CCT, CCT-LSTM, CCT-biLSTM, and CCT-GRU models. On English handwriting, the reported CNN-GA baseline is 97.77%, while the CCT-based models reach 98.96%–99.07% with two transformer layers and 99.46%–99.63% with seven transformer layers. CCT-GRU gives the highest reported two-layer test accuracy at 99.07%, and CCT-LSTM gives the highest reported seven-layer test accuracy at 99.63%.

The findings show that transformer depth consistently improves the reported target-model results, while recurrent hybridization is useful only under certain dataset and depth conditions. Plain CCT remains strongest on MNIST, reaching 99.44% with two layers and 99.55% with seven layers. The main contribution is a practical design strategy: CNN-GA hyperparameters can guide CCT and hybrid CCT construction when full transformer search is costly, but architecture choice should still be guided by dataset complexity, transformer depth, and the validity constraints identified in this study.

Future work should compare the transferred configuration with a documented non-transferred CCT baseline and an independently tuned CCT configuration under the same data partitions and computational budget. Repeated experiments with predetermined random seeds should report mean and standard deviation, while writer-aware split indices should be retained to evaluate unseen-writer generalization. Parameter count, FLOPs, training time, GPU-hours, memory usage, and inference latency should also be reported to test the computational motivation of the transfer strategy.

Declarations

Authors' Contributions

MM was the primary contributor and led the conceptualization, methodology, software implementation, experimentation, formal analysis, visualization, and original manuscript drafting. RAM contributed to research guidance, result interpretation, manuscript review, and editing. Both authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

The authors express their gratitude to the Informatika study program at Universitas Muhammadiyah Semarang for supplying the computational resources and facilities that made this research possible.

Funding

The authors received no specific funding or financial support for the research, authorship, and/or publication of this article.

Availability of data and materials

The public MNIST benchmark dataset can be accessed from its official website, while the English handwriting data used in the CNN-GA stage are derived from the NIST Special Database 19. The private handwritten samples shown in the qualitative demonstration are not included, as they are not used as the basis for the main quantitative claims.

References

- Agrawal, V., Jagtap, J., and Kantipudi, M. V. V. P. (2024a). Exploration of advancements in handwritten document recognition techniques. *Intelligent Systems with Applications*, 22:200358. DOI: 10.1016/J.ISWA.2024.200358.
- Agrawal, V., Jagtap, J., Patil, S., and Kotecha, K. (2024b). Performance analysis of hybrid deep learning framework using a vision transformer and convolutional neural network for handwritten digit recognition. *MethodsX*, 12:102554. DOI: 10.1016/J.MEX.2024.102554.

- Baldominos, A., Saez, Y., and Isasi, P. (2019). A survey of handwritten character recognition with mnist and emnist. *Applied Sciences*, 9(15):3169. DOI: 10.3390/APP9153169.
- Berriche, L., Alqahtani, A., and Rekik, S. (2024). Hybrid arabic handwritten character segmentation using cnn and graph theory algorithm. *Journal of King Saud University - Computer and Information Sciences*, 36(1):101872. DOI: 10.1016/J.JKSUCI.2023.101872.
- Bordelon, B., Noci, L., Li, M., Hanin, B., and Pehlevan, C. (2023). Depthwise hyperparameter transfer in residual networks: Dynamics and scaling limit. *arXiv*. DOI: 10.48550/arXiv.2309.16620.
- Chitty-Venkata, K. T., Emani, M., Vishwanath, V., and Somani, A. K. (2022). Neural architecture search for transformers: A survey. *IEEE Access*, 10:108374–108412. DOI: 10.1109/ACCESS.2022.3212767.
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734. DOI: 10.3115/v1/d14-1179.
- Cohen, G., Afshar, S., Tapson, J., and van Schaik, A. (2017). Emnist: An extension of mnist to handwritten letters. *arXiv*. DOI: 10.48550/arXiv.1702.05373.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv*. DOI: 10.48550/arXiv.2010.11929.
- Fatyanosa, T. N. and Aritsugi, M. (2021). An automatic convolutional neural network optimization using a diversity-guided genetic algorithm. *IEEE Access*, 9:91410–91426. DOI: 10.1109/ACCESS.2021.3091729.
- Garrido-Munoz, C., Rios-Vila, A., and Calvo-Zaragoza, J. (2025). Handwritten text recognition: A survey. *arXiv*. DOI: 10.48550/arXiv.2502.08417.
- Hassani, A., Walton, S., Shah, N., Abuduweili, A., Li, J., and Shi, H. (2021). Escaping the big data paradigm with compact transformers. *arXiv*. DOI: 10.48550/arXiv.2104.05704.
- Li, Y., Chen, D., Tang, T., and Shen, X. (2025). Htr-vt: Handwritten text recognition with vision transformer. *Pattern Recognition*, 158:110967. DOI: 10.1016/J.PATCOG.2024.110967.
- Munsarif, M., Noersasongko, E., Andono, P. N., and Soeleman, M. A. (2023). The evaluation of convolutional neural network and genetic algorithm performance based on the number of hyperparameters for english handwritten recognition. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 12(3):1250–1259. DOI: 10.11591/ijai.v12.i3.pp1250-1259.
- Nguyen, V., Cai, J., and Chu, J. (2019). Hybrid cnn-gru model for high efficient handwritten digit recognition. In *ACM International Conference Proceeding Series*, pages 66–71. DOI: 10.1145/3357254.3357276.
- Qian, X. *et al.* (2023). Knowledge transfer evolutionary search for lightweight neural architecture with dynamic inference. *Pattern Recognition*, 143:109790. DOI: 10.1016/J.PATCOG.2023.109790.
- Raiaan, M. A. K. *et al.* (2024). A systematic review of hyperparameter optimization techniques in convolutional neural networks. *Decision Analytics Journal*, 11:100470. DOI: 10.1016/J.DAJOUR.2024.100470.
- Sun, Y., Xue, B., Zhang, M., Yen, G. G., and Lv, J. (2020). Automatically designing cnn architectures using the genetic algorithm for image classification. *IEEE Transactions on Cybernetics*, 50(9):3840–3854. DOI: 10.1109/TCYB.2020.2983860.
- Yang, G. *et al.* (2022). Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv*. DOI: 10.48550/arXiv.2203.03466.
- Zhan, Z.-H., Li, J.-Y., and Zhang, J. (2022). Evolutionary deep learning: A survey. *Neurocomputing*, 483:42–58. DOI: 10.1016/J.NEUCOM.2022.01.099.