ResNet18-BLSTM-CTC-Based OCR Model for Extracting Technical Parameter Data from Digital Television Analyzer Screenshots

Henrian Robby Fakhriannur[1*], Alva Hendi Muhammad[2]

[1,2]Department of Informatics, Faculty of Computer Science, Universitas Amikom Yogyakarta, Sleman, DI Yogyakarta, Indonesia
E-mail: [1*]robbyfakhriannur@students.amikom.ac.id, [2]alva@amikom.ac.id

(Received: 30 May 2026, revised: 11 Jun 2026, accepted: 12 Jun 2026)

Abstract

Monitoring digital television broadcasting stations requires accurate extraction of technical measurement parameters from screenshot images generated by television analyzer devices. In current practice, parameters such as frequency, signal level, bandwidth, modulation, guard interval, code rate, Fast Fourier Transform (FFT), pilot pattern, provider name, and video standard are often read and recorded manually, which may cause transcription errors when the images contain small characters, low resolution, noise, or visually similar symbols. This study proposes an Optical Character Recognition (OCR) system based on a hybrid ResNet18-Bidirectional Long Short-Term Memory-Connectionist Temporal Classification (ResNet18-BLSTM-CTC) architecture to improve technical parameter extraction from digital television analyzer screenshots. ResNet18 is used to extract visual character features, BLSTM models sequential relationships from both directions, and CTC decodes character sequences without explicit character segmentation. The implementation includes image acquisition, Region of Interest (ROI) selection, cropping, grayscale conversion, resizing into 48 x 48 pixel character images, model training, fine-tuning using hard labels, comparative evaluation, and web-based deployment using Streamlit. After fine-tuning, the proposed model achieved 96.58% accuracy, 0.0342 Character Error Rate (CER), 0.9253 macro F1-score, and reduced errors from 80 to 55. Additional comparison with Tesseract, EasyOCR, and CRNN-CTC, as well as testing on a different analyzer dataset, shows the practical relevance of the proposed OCR workflow for digital television technical monitoring.

Keywords: Optical Character Recognition, ResNet18, Bidirectional Long Short-Term Memory, Connectionist Temporal Classification, Deep Learning

I. INTRODUCTION

Digital television broadcasting requires accurate technical monitoring to ensure that the transmitted signal complies with the required standards and does not interfere with other frequency spectrum users. In Indonesia, radio frequency spectrum monitoring is carried out by the Radio Frequency Spectrum Monitoring Center, including the Class II Radio Frequency Spectrum Monitoring Center of Banjarmasin. One of its responsibilities is to conduct monitoring, measurement, inspection, and interference handling related to the use of radio frequency spectrum, including digital television broadcasting services.

In the monitoring process, technical parameters such as frequency, signal level, bandwidth, modulation, guard interval, code rate, Fast Fourier Transform (FFT), pilot pattern, and other related parameters are obtained from measurement devices such as television analyzers. These measurement results are commonly stored as screenshot images. However, the values displayed in these images are still often read and recorded manually into reports or spreadsheet files. This manual transcription process may lead to human errors, especially when dealing with small characters, low-resolution images, noise, similar character shapes, and variations in screenshot quality. Errors in reading technical parameters may affect the accuracy of monitoring reports and reduce the reliability of decision-making in spectrum supervision.

To address this problem, an automatic data extraction model is needed to convert text information from measurement screenshot images into structured digital data. Optical Character Recognition (OCR) is a technology that enables text recognition from image data. Recent developments in OCR have widely adopted deep learning approaches because they

DOI: 10.34148/teknika.v15i2.1501

can automatically learn visual patterns and character features from images. Convolutional Neural Network (CNN)-based models are effective for extracting spatial features, while recurrent-based models are useful for modeling sequential character relationships.

This study proposes a deep learning-based OCR model using a hybrid ResNet18-Bidirectional Long Short-Term Memory (BLSTM)-Connectionist Temporal Classification (CTC) architecture. ResNet18 is used as a feature extractor to capture visual patterns from character images, following the effectiveness of CNN-based OCR feature extraction reported in previous studies [1], [2]. BLSTM is applied to understand character sequences from both forward and backward directions [3], while CTC is used to decode character sequences without requiring explicit character segmentation [4]. This combination is expected to improve the accuracy and efficiency of text extraction from digital television measurement screenshots.

The specific objective of this study is to develop and evaluate a domain-specific OCR workflow that can transform TV analyzer screenshots into structured technical data with minimal manual transcription. The main contributions are: (1) the construction of a character-level OCR dataset from digital television measurement screenshots, (2) the design of a ResNet18-BLSTM-CTC recognition model for technical characters, (3) fine-tuning using hard-label samples derived from error analysis, and (4) integration of the trained model into a Streamlit-based verification and Excel reporting workflow.

Implementation begins with data acquisition and preparation, including ROI selection, cropping, grayscale conversion, resizing, model training, fine-tuning, and Streamlit deployment. The model is evaluated using accuracy, CER, macro F1-score, and error analysis to identify misclassified technical characters.

This study is domain-specific because digital television analyzer screenshots contain mixed numbers, units, abbreviations, and small symbols that differ from printed documents or natural-scene text. By automating the extraction workflow, the proposed OCR system supports faster and more consistent technical reporting.

II. RELATED WORKS

A. Deep Learning-Based OCR

Several OCR studies show that convolutional and recurrent architectures are capable for text recognition in visual pictures that have interference, tiny symbols, and uneven arrangements. Regarding Arabic document recognition [1], CNN-BLSTM-CTC was used by researchers to perform character identification [3]. BiLSTM and CTC are also helpful for sequence-based text extraction because BiLSTM and CTC explain how symbols relate to each other [5]. Text is decoded by these systems without using explicit segmentation, which makes the process simpler. CNN and BiLSTM approaches are also used for general text extraction from visual pictures [2]. Attention-based recurrent networks and adaptive augmentation have shown benefits for scene text and visually complex text recognition [4], [6], while deep CRNN backbones support recognition in natural scene images [7]. These findings support the use of a ResNet18-BLSTM-CTC architecture for the proposed OCR model.

B. OCR for Technical and Real-World Images

Previous works have also applied OCR to technical documents, water meter readings, identity cards, license plates, and other real-world objects. Relay protection setting sheet recognition demonstrates the relevance of combining detection and recognition stages for technical documents [8]. Water meter reading studies show the usefulness of text recognition for measurement-related images [9], while identity card extraction highlights the importance of structured text detection and recognition in real-world documents [10]. ROI selection, cropping, preprocessing, and robust feature extraction are also emphasized in document text localization using Mask R-CNN [11]. The present study differs from these applications because it focuses on screenshots produced by TV Analyzer devices, which contain mixed numerical values, technical abbreviations, units, uppercase and lowercase letters, and symbols related to digital television broadcasting parameters.

C. Research Gap

Although deep learning-based OCR has been widely studied, limited research specifically addresses technical parameter extraction from digital television measurement screenshots. Such images contain small fonts, limited resolution, visual noise, and similar character shapes such as O/0, V/v, S/5, and g/q. Therefore, this study contributes an OCR model and a web-based workflow designed for technical monitoring and reporting in digital television broadcasting stations.

To support the comparative analysis, this study considers several OCR baselines and recent OCR directions. Tesseract is used as a classical OCR engine baseline, while EasyOCR represents a practical deep learning-based OCR engine used in real-world recognition tasks [12]. CRNN-CTC and CNN-based approaches are commonly applied in sequence recognition and license plate or technical text recognition [7], [8], [13]. Transformer-based OCR and feature extractor search studies further indicate recent progress toward robust text recognition architectures [14-17]. Additional OCR studies on handwritten, low-resource, and non-Latin scripts also show that performance is affected by script characteristics, image quality, and evaluation setting [18-24].

III. RESEARCH METHODOLOGY

A. Research Design

This research is categorized as applied and experimental research. It is applied research because the proposed OCR model is designed to solve a practical problem in the monitoring and reporting of digital television broadcasting parameters. It is experimental research because the model is

DOI: 10.34148/teknika.v15i2.1501

trained, tested, evaluated, and improved through fine-tuning using a prepared dataset. The study uses a quantitative approach because the performance of the model is measured using numerical metrics, including accuracy, Character Error Rate (CER), macro F1-score, and the number of prediction errors. Detailed research stages and outputs are listed in Table 1.

Table 1. Research Stages and Outputs

Stage	Description	Output
Problem identification	Identifying the risk of manual transcription errors from TV analyzer screenshots.	Research problem and OCR requirement
Dataset collection	Collecting screenshot images containing digital television technical parameters.	Raw screenshot dataset
Image preprocessing	Applying cropping, ROI selection, grayscale conversion, resizing, segmentation, and normalization.	Standardized character images
Model design	Designing the ResNet18-BLSTM-CTC OCR architecture.	Proposed OCR model
Training and fine tuning	Training the initial model and improving its performance through fine tuning.	Optimized OCR model
Evaluation	Measuring model performance using character-level metrics and error analysis.	Quantitative performance results
System implementation	Deploying the OCR model in a web-based application using Streamlit.	Practical OCR application

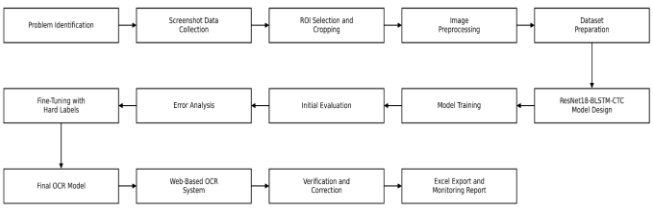


Figure 1. Research Workflow of the Proposed ResNet18-BLSTM-CTC OCR System

Figure 1 shows a sequential workflow from problem identification and data preparation to model training, fine tuning, web-based OCR implementation, verification, and Excel report export.

B. Dataset Collection

Dataset materials for this research are derived from screenshots captured during digital television broadcasting measurement tasks using a TV analyzer tool (Table 2). Within these screenshots, technical particulars like channel, frequency, signal level, bandwidth, modulation, guard interval, code rate, Fast Fourier Transform (FFT), pilot pattern, broadcasting provider name, and video standard are located (Figure 2). The dataset is organized into various classifications because this organization makes the gathering procedure easier.

Table 2. Dataset Categories and Technical Parameters

Category	Technical Parameter	Description
A	Channel, frequency, and signal level	Contains the basic measurement information used to identify channel allocation and signal strength.
B	Bandwidth	Contains channel bandwidth information obtained from the measurement display.
C	Modulation, guard interval, pilot pattern, and FFT	Contains digital transmission configuration parameters.
D	Code rate	Contains forward error correction configuration used in DVB-T2 transmission.
E	Broadcasting provider name and video standard	Contains service/provider information and video format details.

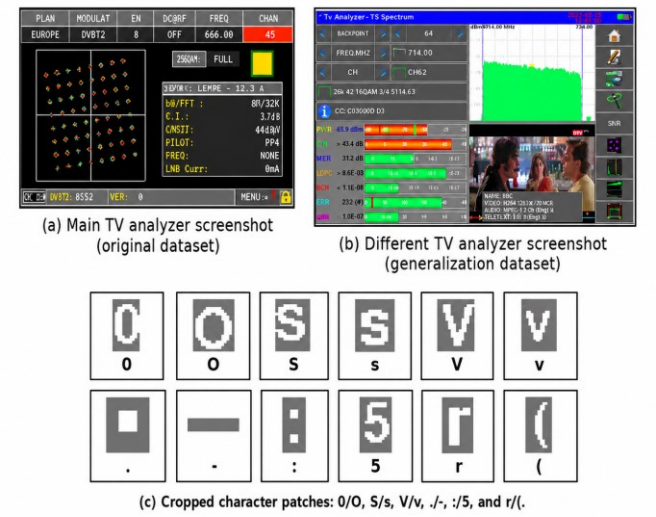


Figure 2. Main and Different TV Analyzer Dataset Samples with Cropped Character Patches

C. Image Preprocessing

Image preprocessing is performed to improve the quality and consistency of the OCR input. Since the original screenshots may contain irrelevant areas, different resolutions, noise, and small characters, preprocessing is required before

training and testing the model. The preprocessing stage includes cropping, Region of Interest (ROI) selection, grayscale conversion, resizing, character segmentation, and normalization. The segmented character images are resized into 48 x 48 pixels.

The cropped Region of Interest (ROI) is resized into a standardized character image using Formula 1.

$$I_{\text{resized}} = \text{Resize}(I_{\text{ROI}}, 48 \times 48) \quad (1)$$

where I_ROI is the cropped text region and I_resized is the standardized 48 x 48 pixel character image.

Pixel normalization is applied after grayscale conversion using Formula 2.

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (2)$$

where x is the original pixel value, x_min and x_max are the minimum and maximum pixel values, and x_norm is the normalized pixel value.

A detailed image preprocessing pipeline is shown in Figure 3 below. Also, the results for the segmented characters dataset used in this research are depicted in Figure 4.

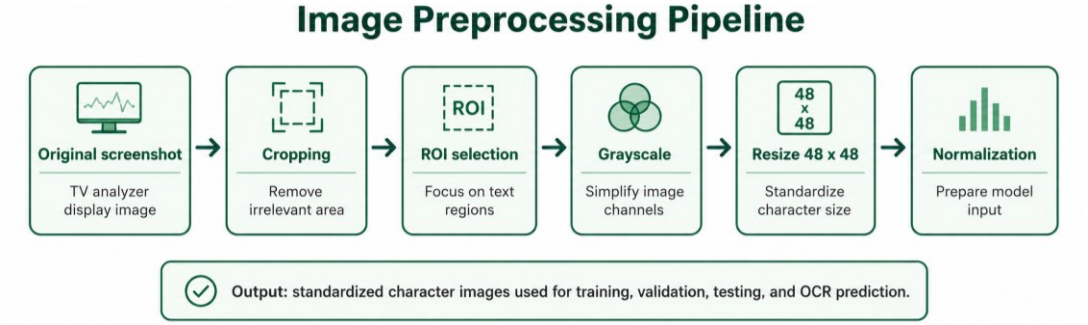


Figure 3. Image Preprocessing Workflow

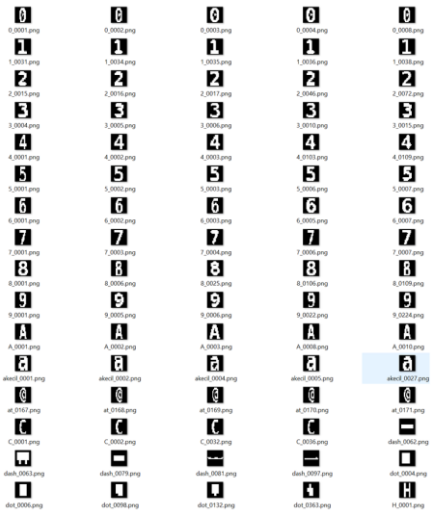


Figure 4. Sample Segmented Character Dataset

D. Dataset Preparation

After preprocessing, the dataset is divided into training, validation, and testing subsets (Table 3). The training data are used to update the model parameters, the validation data are used to monitor model performance during training, and the testing data are used to evaluate the final model on unseen data.

Table 3. Dataset Subset Distribution

Subset	Number of character images	Purpose
Training	13,620	Used to train the OCR model and update its parameters.
Validation	1,610	Used to monitor model performance during training and fine tuning.
Testing	1,610	Used to evaluate the generalization ability of the final model.

E. Proposed Model Architecture

A hybrid ResNet18-BLSTM-CTC architecture is utilized by this OCR model because the framework integrates multiple layers for recognition (Figure 5). ResNet18 functions as the visual feature extractor which gathers character designs like edges, strokes, curves, and local structures. ResNet18-BLSTM-CTC architecture changes the extracted feature map into a sequence description so that BLSTM can process the data. Contextual associations between characters are learned by BLSTM when those characters are scrutinized from both forward and backward directions. CTC operates as the decoding system that produces the final character sequence because explicit character segmentation is not required by this method.

The ResNet18 feature extraction stage is represented by Formula 3.

$$Z = R_{\theta_R}(X) \quad (3)$$

where Z is the extracted feature map and theta_R denotes the learnable parameters of ResNet18.

The residual block used in ResNet18 is expressed by Formula 4.

$$H(x) = F(x, W_i) + x \quad (4)$$

where x is the input feature, F(x, W_i) is the residual mapping, W_i denotes the network weights, and H(x) is the residual block output.

The model workflow can be summarized consistently as X -> F -> S -> H -> P -> Y, where X is the input image, F = ResNet18(X) is the extracted visual feature map, S is the reshaped feature sequence, H = BLSTM(S) is the bidirectional sequence representation, P is the character probability distribution produced before CTC decoding, and Y is the extracted text. This architecture is suitable for technical measurement screenshots because the images contain small characters, numerical values, abbreviations, symbols, and visually similar character shapes.

The BLSTM sequence representation combines the forward and backward hidden states using Formula 5.

$$h_t = [h_{\text{forward}}(t); h_{\text{backward}}(t)] \quad (5)$$

where h(t) is the combined hidden representation at time step t, h_forward(t) is the forward hidden state, and h_backward(t) is the backward hidden state.

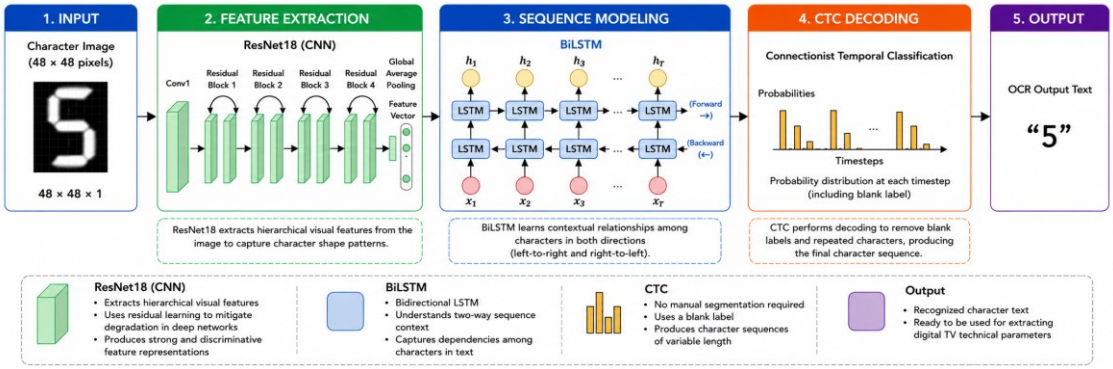


Figure 5. ResNet18-BLSTM-CTC OCR Architecture

IV. RESULTS AND DISCUSSION

A. Model Training and Fine Tuning

A prepared character-level dataset is utilized so that the model can be trained correctly. Many scholars state that the model identifies specific character patterns from preprocessed images while the CTC loss is decreased during this training period. Because validation performance is monitored constantly, the process shows convergence which stops the model from overfitting. Fine tuning is performed after the first training stage is done because this activity enhances the model performance for those characters that look almost identical or are often categorized incorrectly. The fine tuning stage is important because the dataset contains characters such as O and 0, V and v, S and 5, and g and q, which have similar visual structures.

During training, CTC estimates the probability of a target sequence by considering all valid alignment paths as shown in Formula 6.

$$P(y \mid X) = \sum_{\pi \in \mathcal{B}^{-1}(y)} P(\pi \mid X) \quad (6)$$

where X is the input feature sequence, y is the target label sequence, pi is an alignment path, B is the mapping function that removes repeated and blank labels, and T is the sequence length.

The CTC loss minimized during training is defined in Formula 7.

$$L_{\text{CTC}} = -\log P(y \mid X) \quad (7)$$

where L_CTC is the Connectionist Temporal Classification loss.

The reduction of prediction errors after fine tuning is calculated using Formula 8.

$$\text{Error Reduction} = \frac{\text{Errors}_{\text{before}} - \text{Errors}_{\text{after}}}{\text{Errors}_{\text{before}}} \times 100\% \quad (8)$$

where Errors_before is the number of errors before fine tuning and Errors_after is the number of errors after fine tuning. In this study, error reduction equals ((80 - 55) / 80) x 100% = 31.25%.

Table 4. Model Training Configuration

Parameter	Value
Model	Custom ResNet18 + BLSTM + CTC
Input size	48 x 48 pixels
Image format	Grayscale image + CSV label
Epochs	150
Batch size	32
Framework	PyTorch
Computation device	CUDA-supported device

The OCR model was trained using the ResNet18-BLSTM-CTC architecture on character images converted to a grayscale format with a resolution of 48 x 48 pixels (Table 4). The training and validation losses decreased at the beginning of training and remained stable afterwards (Figure 6). The validation and character accuracies and the F1-scores also increased rapidly at the beginning of training and plateaued at high values.

The Character Error Rate (CER) curve further confirms the convergence behavior (Figure 7). The CER dropped significantly at the beginning of training and then remained low, which indicates that the model was able to reduce character-level errors during the learning process.

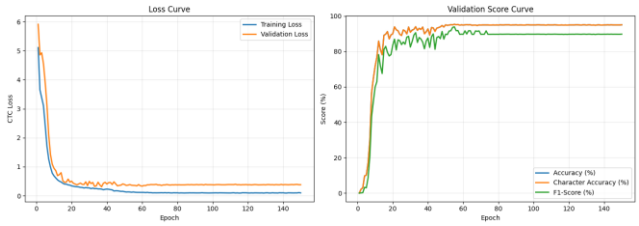


Figure 6. Model Training and Validation Performance Curves

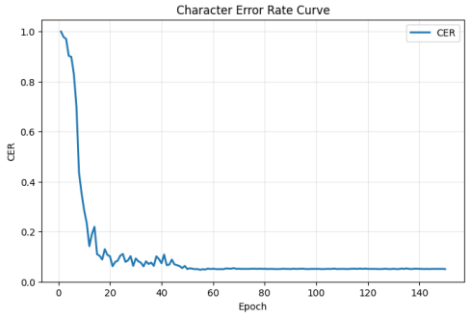


Figure 7. Character Error Rate Curve During Training

B. Initial Model Testing Results

For testing purposes, 1,610 character images that were not used in the training of the model were utilized. With these characters, the model was able to correctly classify 1,530 characters and made 80 errors. The initial model achieved an accuracy of 95.03%, a character accuracy of 95.03%, a character error rate (CER) of 0.0497, and a macro F1-score of 0.8743 (Table 5 and 6). Given these results, it is safe to say that the initial model was reliable for classification purposes, despite some errors made in the classification of visually similar characters.

Table 5. Initial Model Testing Metrics

Metric	Value
Accuracy	95.03%
Character Accuracy	95.03%
Character Error Rate (CER)	0.0497
Macro F1-score	0.8743
Total test characters	1,610
Correct predictions	1,530

Table 6. Initial Model Misclassification Examples

No.	Actual Character	Predicted Character
1	4	g
2	4	g
3	a	0
4	a	0
5	a	0
6	@	0
7	@	0
8	@	0
9	:	-
10	:	5

C. Confusion Matrix Evaluation

The performance of the OCR model is evaluated using a variety of quantitative metrics. Accuracy, and character accuracy, measure the overall correctness of the model in predicting characters. Character Error Rate measures the number of errors in predicting characters, with lower rates indicating better performance. Macro F1-score is used to evaluate the balance of model performance for each character class. In addition, a confusion matrix and error analysis are utilized to evaluate the number of each type of error made by the model (Figure 8).

Accuracy is calculated using Formula 9.

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}} \quad (9)$$

Character Error Rate (CER) is calculated using Formula 10.

$$CER = \frac{S+D+I}{N} \quad (10)$$

where S is substitution error, D is deletion error, I is insertion error, and N is the total number of ground-truth characters.

Precision, recall, F1-score, and macro F1-score are calculated using Formula 11, 12, 13, and 14.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (11)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (12)$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}} \quad (13)$$

$$\text{Macro F1} = \frac{1}{C}\sum_{i=1}^{C} F\,1_i \quad (14)$$

Where TP is true positive, FP is false positive, FN is false negative, and C is the number of character classes.

DOI: 10.34148/teknika.v15i2.1501

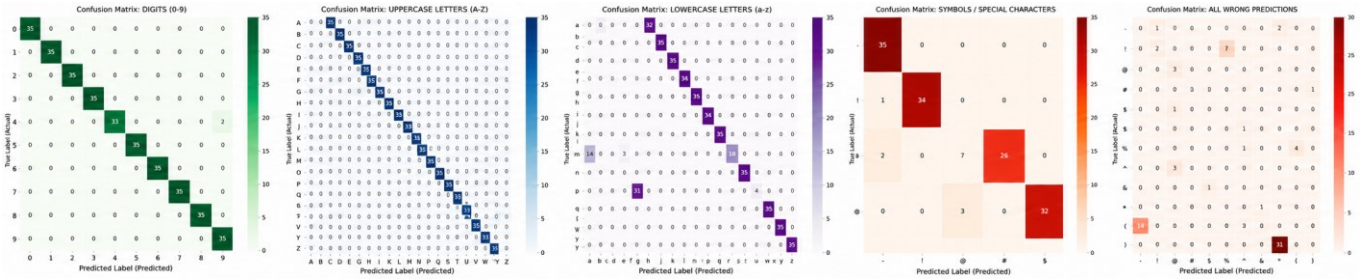


Figure 8. Initial Model Confusion Matrix Evaluation

D. Fine Tuning on Hard Labels

To address the classification inaccuracies observed during the initial testing and confusion matrix evaluation of the model, a fine-tuning stage was performed on a dataset that contained hard labels. These hard labels were used to force the model to learn more visually distinguishable features for character pairs that were often misclassified. The fine-tuning dataset included synthetic data, augmented training data, real data, synthetic hard labels, and real augmented error-label data (Table 7).

Table 7. Fine-Tuning Dataset Sources

No.	Fine-tuning data source	Number of samples
1	finetune_synthetic_error_label	12,800
2	synthetic_hard	4,940
3	train_aug	4,840
4	synthetic	3,480
5	finetune_real_aug_error_label	1,400
6	real	360
	Total	**27,820**

E. Model Evaluation After Fine-Tuning

After fine-tuning on hard labels, the optimized model was evaluated using the same 1,610 testing characters to ensure a fair comparison with the initial model. The evaluation focused on accuracy, character accuracy, CER, macro F1-score, total errors, confusion matrix behavior, and remaining misclassification patterns (Table 8). The performance comparison before and after fine-tuning is shown in Table 9 and Figure 9 below.

Table 8. Model Evaluation After Fine-Tuning

Metric	Value after fine-tuning
Accuracy	96.58%
Correct predictions	1,555
Total test characters	1,610
CER	0.0342
Character Accuracy	96.58%
Macro F1-score	0.9253
Total errors	55

Table 9. Performance Comparison Before and After Fine-Tuning

Metric	Before fine-tuning	After Fine-Tuning	Change
Accuracy	95.03%	96.58%	+1.55 percentage points
Character Accuracy	95.03%	96.58%	+1.55 percentage points
CER	0.0497	0.0342	-0.0155
Macro F1-score	0.8743	0.9253	+0.0510
Total errors	80	55	-25 errors

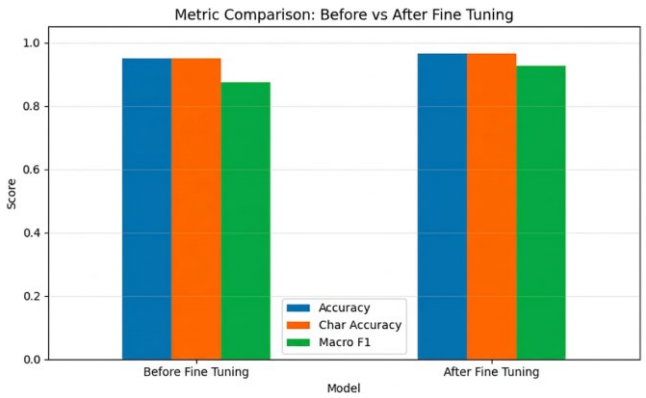


Figure 9. Metric Comparison Before and After Fine-Tuning

The confusion matrix after fine-tuning shows that most class predictions remained concentrated along the diagonal (Figure 10 and 11). Some remaining errors occurred in specific visually similar pairs, such as dot/dash, colon/ 5, S/s, V/v, and r/(. A few difficult classes still required additional data or augmentation. Based on the top error-pair analysis after fine-tuning, the most frequent remaining errors were v->V (31 cases), S->s (13 cases), r->(5 cases), r->C (3 cases), .->- (2 cases), and:->5 (1 case).

DOI: 10.34148/teknika.v15i2.1501
TEKNIKA, Volume 15(2), July 2026, pp. 303-313
ISSN 2549-8037, EISSN 2549-8045

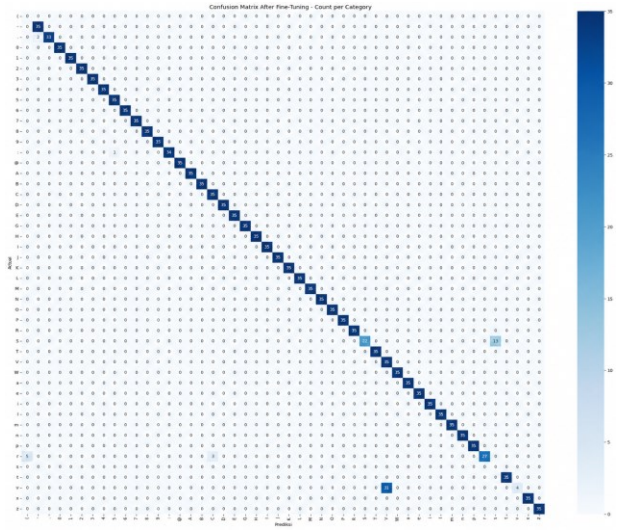


Figure 10. Confusion Matrix After Fine-Tuning

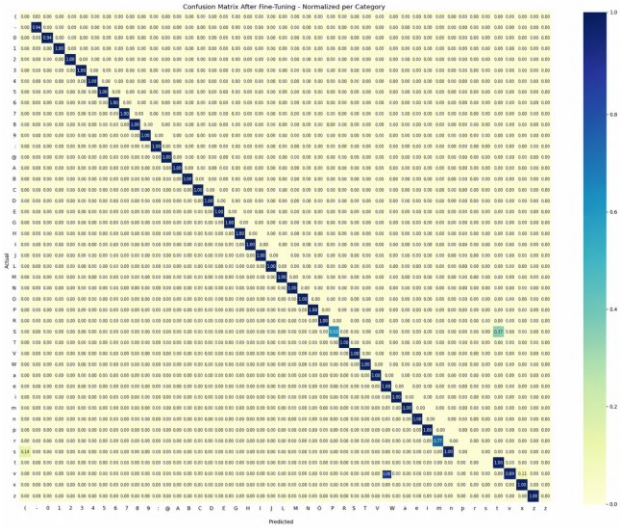


Figure 11. Normalized Confusion Matrix After Fine-Tuning

The precision, recall, and F1-score chart for each character category shows that most classes achieved high and balanced scores after fine-tuning (Figure 12). The few classes with low scores are the difficult characters, indicating that the main limitation of the model is on a few specific classes rather than on all the classes in the dataset.

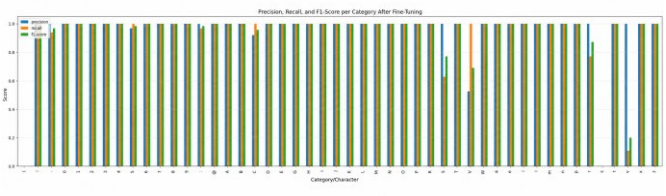


Figure 12. Character-Level Precision, Recall, and F1-Score Results

Although the model performance improved after fine-tuning, several errors were still observed. The following Table 10 and Figure 13 show examples of the remaining misclassified characters after fine-tuning. These cases mainly involve characters with small strokes, similar shapes, and uppercase-lowercase ambiguity.

Table 10. Remaining Misclassification Examples After Fine-Tuning

No.	Actual Character	Predicted Character
1	:	5
2	.	-
3	.	-
4	r	(
5	r	c
6	r	(
7	r	(
8	r	c
9	r	(
10	r	(
11	r	c
12	S	s
13	S	s
14	S	s
15	S	s
16	v	V

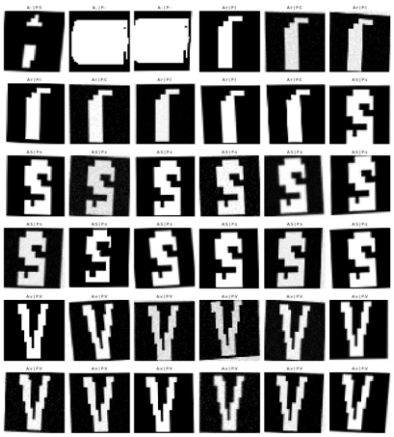


Figure 13. Remaining Misclassified Character Examples After Fine-Tuning

Table 11. Comparative and Generalization Evaluation of OCR Methods

Method	Dataset	N	Acc.	CER	Macro F1
Tesseract	Main	1,610	16.77%	1.3168	0.2204
EasyOCR	Main	1,610	48.26%	0.5404	0.3674
CRNN-CTC	Main	1,610	97.83%	0.0217	0.9775
ResNet18-BLSTM-CTC	Main	1,610	96.58%	0.0342	0.9253
ResNet18-BLSTM-CTC	Different	1,995	93.38%	0.0662	0.9210

Table 11 shows that the proposed ResNet18-BLSTM-CTC model remained competitive on the main TV analyzer dataset

DOI: 10.34148/teknika.v15i2.1501

and maintained stable recognition performance on a different TV analyzer dataset. The lower accuracy on the different analyzer dataset indicates a domain shift caused by differences in screenshot layout, character appearance, resolution, and display condition. Tesseract and EasyOCR performed considerably lower because they were used as off-the-shelf OCR engines without domain-specific training or fine tuning.

Overall, the results demonstrate that the ResNet18-BLSTM-CTC architecture is effective in extracting the technical characters within the screenshots of the digital TV measurement device. The fine-tuning of the model on hard labels led to an improvement in accuracy, a decrease in the character error rate, a higher macro F1-score, and a reduction in errors from 80 to 55. Therefore, the proposed model is suitable for supporting more accurate and efficient OCR-based technical data extraction.

F. Web-Based OCR System Implementation

The trained OCR model was implemented in a Streamlit-based web application to support practical extraction of technical parameters from digital television measurement screenshots.

The web application can support users to upload television screenshots in JPG, JPEG, PNG, and BMP formats. The software application can support an integrated workflow for users including the upload of TV screenshots, preview of screenshots, generation of OCR text, verification and correction of OCR text, storage of OCR text, and finally, export of OCR text (Figure 14).

After the images are selected, the system displays thumbnails and file names so that users can verify the input before processing (Figure 15). When the Generate OCR button is pressed, the application applies preprocessing and character recognition using the trained ResNet18-BLSTM-CTC model. The system then displays administrative data, technical multiplexing parameters, and broadcasting program information in edit forms (Figure 16). This workflow is intended to keep human verification inside the application so that OCR errors can be corrected before the data are saved or exported into the final report (Figure 17).

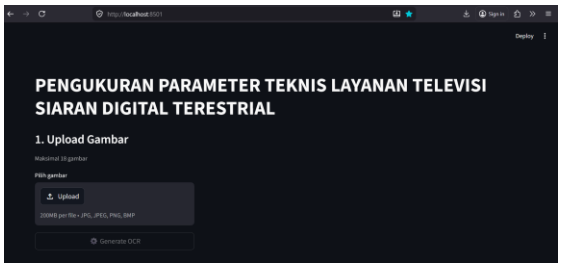


Figure 14. Main Upload Page of the Streamlit-Based OCR Application

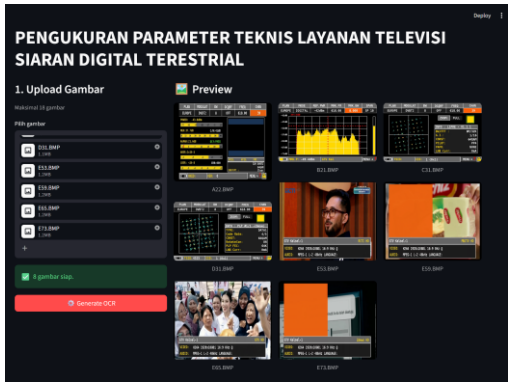


Figure 15. Uploaded Image Preview and Batch Processing Interface

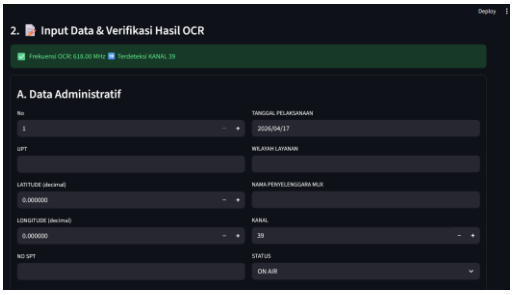


Figure 16. Administrative Data Verification Form Generated from OCR Results

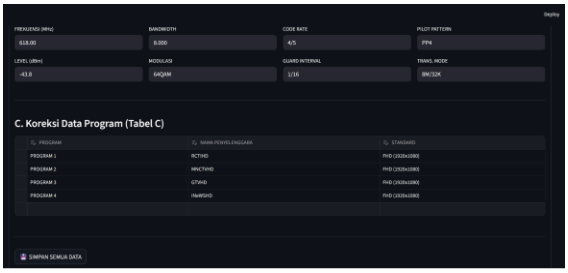


Figure 17. Technical Parameter and Program Data Correction Form

G. Excel Data Export Result

Beyond the web interface, the OCR system also includes a Microsoft Excel export option (Figure 18 and 19). This provides users with the OCR results in a structured format containing administrative data, multiplexing technical parameters, and broadcasting program identification data. This feature supports the automation of OCR results into digital television monitoring reports, eliminating the need for manual retyping.

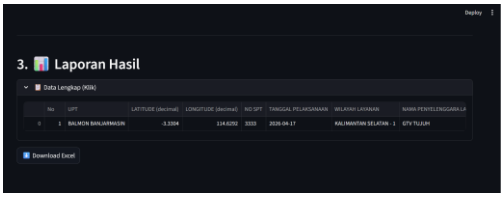
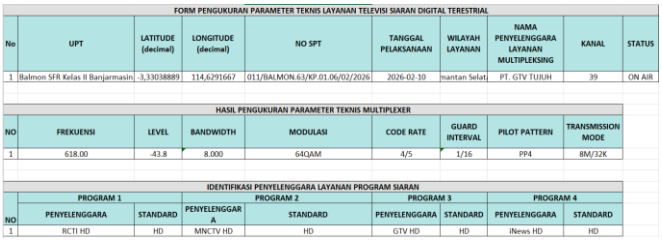


Figure 18. Summary Report Page Before Downloading the Output

DOI: 10.34148/teknika.v15i2.1501



FORM PENGUKURAN PARAMETER TEKNIS LAYANAN TELEVISI SIARAN DIGITAL TERESTRIAL

No	UPT	LATITUDE (decimal)	LONGITUDE (decimal)	NO SPT	TANGGAL PELAKSANAAN	WILAYAH LAYANAN	NAMA PENYELENGGARA LAYANAN MULTIPLEKSING	KANAL	STATUS
1	Balmon SFR Kelas II Banjarmasin	-3,33038889	114,6291667	011/BALMON.63/KP.01.06/02/2026	2026-02-10	mantan Selat	PT. GTV TUJUH	39	ON AIR

HASIL PENGUKURAN PARAMETER TEKNIS MULTIPLEXER

NO	FREKUENSI	LEVEL	BANDWIDTH	MODULASI	CODE RATE	GUARD INTERVAL	PILOT PATTERN	TRANSMISSION MODE
1	618.00	-43.8	8.000	64QAM	4/5	1/16	PP4	8M/32K

IDENTIFIKASI PENYELENGGARA LAYANAN PROGRAM SIARAN

NO	PROGRAM 1		PROGRAM 2		PROGRAM 3		PROGRAM 4	
	PENYELENGGARA	STANDARD	PENYELENGGARA	STANDARD	PENYELENGGARA	STANDARD	PENYELENGGARA	STANDARD
1	RCTI HD	HD	MNCTV HD	HD	GTV HD	HD	INews HD	HD

Figure 19. Excel Output Containing Administrative, Technical, and Program Data

V. FUTURE WORK

Although the proposed model achieved high recognition performance, several challenges remain. Most residual errors were associated with visually similar characters, including dot/dash, colon/5, S/s, V/v, and r/(. Future work will focus on expanding the dataset and improving augmentation strategies for difficult classes to enhance model robustness. Domain adaptation, federated optimization, and CNN-Transformer integration may also be explored to improve generalization across different digital television analyzer devices [14], [25-27]. Furthermore, attention mechanisms and self-supervised character-level distillation have the potential to reduce class ambiguity and minimize labeling effort while maintaining recognition accuracy [28], [29].

VI. CONCLUSION

This study developed a deep learning-based OCR model for extracting technical parameter data from digital television analyzer screenshots. The proposed ResNet18-BLSTM-CTC architecture integrates visual feature extraction, bidirectional sequence modeling, and CTC decoding, enabling character recognition without requiring manual character segmentation. The trained model was implemented in a Streamlit-based web application that supports image upload, OCR processing, data verification, correction, storage, and Excel export.

Experimental results demonstrated strong recognition performance. Before fine tuning, the model achieved 95.03% accuracy, a character error rate (CER) of 0.0497, a macro F1-score of 0.8743, and 80 errors out of 1,610 testing characters. After fine tuning with hard labels, accuracy improved to 96.58%, CER decreased to 0.0342, macro F1-score increased to 0.9253, and the number of recognition errors was reduced from 80 to 55, corresponding to a 31.25% error reduction.

Overall, the proposed ResNet18-BLSTM-CTC approach effectively improves the accuracy and efficiency of technical data extraction from digital television measurement screenshots, demonstrating its potential for practical automation of measurement reporting and technical data management.

REFERENCES

[1] L. Mosbah, I. Moalla, T. M. Hamdani, B. Neji, T. Beyrouthy, and A. M. Alimi, "ADOCRNet: A Deep Learning OCR for Arabic Documents Recognition," *IEEE Access,* vol. 12, no. January, pp. 55620-55631, 2024, doi: 10.1109/access.2024.3379530.

[2] M. Sinthuja, C. G. Padubidri, G. S. Jayachandra, M. C. Teja, and G. S. P. Kumar, "Extraction of Text from Images Using Deep Learning," presented at the Procedia Computer Science, 2024.

[3] S. Mahadevkar, S. Patil, and K. Kotecha, "Enhancement of handwritten text recognition using AI-based hybrid approach," *MethodsX,* vol. 12, Jun. 2024 2024, doi: 10.1016/j.mex.2024.102654.

[4] A. A. A. Alshawi, J. Tanha, and M. A. Balafar, "An Attention-Based Convolutional Recurrent Neural Networks for Scene Text Recognition," *IEEE Access,* vol. 12, pp. 8123-8134, 2024, doi: 10.1109/access.2024.3352748.

[5] B. H. Nayef, S. N. H. S. Abdullah, R. Sulaiman, and A. M. Saeed, "Text Extraction with Optimal Bi-LSTM," *Computers, Materials & Continua,* vol. 76, no. 3, pp. 3549-3567, 2023, doi: 10.32604/cmc.2023.039528.

[6] B. Imane, A. Alae, K. Ghizlane, and M. Mrabti, "Enhancing Arabic handwritten word recognition: a CNN-BiLSTM-CTC architecture with attention mechanism and adaptive augmentation," *Discover Applied Sciences,* vol. 7, no. 5, 2025, doi: 10.1007/s42452-025-06952-z.

[7] A. A. Chandio, M. Asikuzzaman, M. R. Pickering, and M. Leghari, "Cursive Text Recognition in Natural Scene Images Using Deep Convolutional Recurrent Neural Network," *IEEE Access,* vol. 10, pp. 10062-10078, 2022, doi: 10.1109/access.2022.3144844.

[8] X. Lv, S. Gao, G. Miao, and Z. Chen, "Relay Protection Setting Sheet Detection and Recognition Approach Based on YOLOv8-CRNN-CTC," *IEEE Access,* vol. 14, no. January, pp. 19961-19969, 2026, doi: 10.1109/access.2026.3654663.

[9] B. V. Nguyen *et al.*, "Water Meter Reading Based on Text Recognition Techniques and Deep Learning," *IEEE Access,* vol. 13, no. February, pp. 41422-41434, 2025, doi: 10.1109/access.2025.3547225.

[10] G. Suddul and J. F. L. Seguin, "A custom-built deep learning approach for text extraction from identity card images," *International Journal of Informatics and Communication Technology,* vol. 13, no. 1, pp. 34-41, Apr. 2024 2024, doi: 10.11591/ijict.v13i1.pp34-41.

[11] P. Kiatphaisansophon, D. Wanvarie, and N. Cooharojananone, "Efficient Text Bounding Box Identification Using Mask R-CNN: Case of Thai Documents," *IEEE Access,* vol. 12, pp. 49306-49328, 2024, doi: 10.1109/access.2024.3383911.

[12] A. Sarhan *et al.*, "Egyptian car plate recognition based on YOLOv8, Easy-OCR, and CNN," *Journal of Electrical Systems and Information Technology,* vol. 11, no. 1, p. 32, 2024, doi: 10.1186/s43067-024-00156-y.

[13] G. Pandey, K. K. C, N. Lamichhane, and U. Subedi, "CNN based System for Automatic Number Plate Recognition," *Journal of Soft Computing Paradigm,* vol. 5, no. 4, pp. 347-364, Dec. 2023 2023, doi: 10.36548/jscp.2023.4.002.

[14] A. K. A. Abdulsalam, F. Saeedi, and M. A. Ambusaidi, "Robust Automatic License Plate Recognition Using Synthetic Data and Transformer-Based Deep Learning Model," *IEEE Access,* vol. 13, no. September, pp. 182890-182901, 2025, doi: 10.1109/access.2025.3621237.

[15] S. Fang, Z. Mao, H. Xie, Y. Wang, C. Yan, and Y. Zhang, "ABINet++: Autonomous, Bidirectional and Iterative Language Modeling for Scene Text Spotting," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 45, no. 6, pp. 7123-7141, Jun. 2023 2023, doi: 10.1109/tpami.2022.3223908.

[16] H. Zhang, Q. Yao, J. T. Kwok, and X. Bai, "Searching a High Performance Feature Extractor for Text Recognition Network," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 45, no. 5, pp. 6231-6246, May 2023 2023, doi: 10.1109/tpami.2022.3205748.

[17] C. Xue, J. Huang, W. Zhang, S. Lu, C. Wang, and S. Bai, "Image-to-Character-to-Word Transformers for Accurate Scene Text Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 45, no. 11, pp. 12908-12921, Nov. 2023 2023, doi: 10.1109/tpami.2022.3230962.

[18] S. A. Raza, M. S. Farooq, U. Farooq, H. Karamti, T. Khurshaid, and I. Ashraf, "A Convolutional Neural Network Based Optical Character Recognition for Purely Handwritten Characters and Digits," *Computers, Materials & Continua,* vol. 84, no. 2, pp. 3149-3173, 2025, doi: 10.32604/cmc.2025.063255.

[19] H. M. R. U. Rehman, S. A. Haider, H. Faisal, K. Y. Yoo, M. Z. Jhandir, and G. S. Choi, "A Novel Framework for Saraiki Script Recognition Using Advanced Machine Learning Models (YOLOv8 and CNN)," *IEEE Access,* vol. 13, no. April, pp. 56843-56860, 2025, doi: 10.1109/access.2025.3551747.

[20] F. Yasir and M. Kazmi, "Acceleration of Urdu Optical Character Recognition on Zynq UltraScale+ MPSoC Using Deep Convolutional Neural Network," *IEEE Access,* vol. 13, no. August, pp. 135538-135557, 2025, doi: 10.1109/access.2025.3593294.

[21] C. Mai, P. Penava, and R. Buettner, "Oracle Bone Inscription Character Recognition Based on a Novel Convolutional Neural Network Architecture," *IEEE Access,* vol. 12, no. December, pp. 197021-197034, 2024, doi: 10.1109/access.2024.3521319.

[22] Prathwini, A. P. Rodrigues, P. Vijaya, and R. Fernandes, "Tulu Language Text Recognition and Translation," *IEEE Access,* vol. 12, no. January, pp. 12734-12744, 2024, doi: 10.1109/access.2024.3355470.

[23] N. Khan *et al.*, "Systematic Literature Review of Machine Learning Models and Applications for Text Recognition," *IEEE Access,* vol. 13, no. August, pp. 177647-177670, 2025, doi: 10.1109/access.2025.3618109.

[24] E. Vidal, A. H. Toselli, A. Ríos-Vila, and J. Calvo-Zaragoza, "End-to-End page-Level assessment of handwritten text recognition," *Pattern Recognition,* vol. 142, Oct. 2023 2023, doi: 10.1016/j.patcog.2023.109695.

[25] X. Q. Liu, X. Y. Ding, X. Luo, and X. S. Xu, "ProtoUDA: Prototype-Based Unsupervised Adaptation for Cross-Domain Text Recognition," *IEEE Transactions on Knowledge and Data Engineering,* vol. 36, no. 12, pp. 9096-9108, 2024, doi: 10.1109/tkde.2023.3344761.

[26] H. Ren, J. Deng, X. Xie, X. Ma, and Y. Wang, "FedBoosting: Federated learning with gradient protected boosting for text recognition," *Neurocomputing,* vol. 569, Feb. 2024 2024, doi: 10.1016/j.neucom.2023.127126.

[27] Y. Zhang and G. Li, "Improving Handwritten Mathematical Expression Recognition via Integrating Convolutional Neural Network With Transformer and Diffusion-Based Data Augmentation," *IEEE Access,* vol. 12, no. May, pp. 67945-67956, 2024, doi: 10.1109/access.2024.3399919.

[28] R. Malhotra and M. T. Addis, "Handwritten Amharic Word Recognition With Additive Attention Mechanism," *IEEE Access,* vol. 12, no. August, pp. 114645-114657, 2024, doi: 10.1109/access.2024.3444897.

[29] T. Guan, W. Shen, X. Yang, Q. Feng, Z. Jiang, and X. Yang, "Self-supervised Character-to-Character Distillation for Text Recognition," presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023. [Online]. Available: https://github.com/TongkunGuan/CCD.