Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Hybrid CNN-BiLSTM with CTC for Enhanced Text Recognition in Complex Background Images

[1]Rakesh T M, [2]Girisha G S
[1]Department of CSE, School of Engineering, Dayananda Sagar University, Harohalli, Bangalore, 562112, Karnataka, India.
rakesh.tm-rs-cse@dsu.edu.in
[2]Department of CSE, School of Engineering, Dayananda Sagar University, Harohalli, Bangalore, 562112, Karnataka, India.
girisha-cse@dsu.edu.in

ARTICLE INFO

Received: 30 Dec 2024

Revised: 12 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

The problems that robotic reading of text faces such as poor light, messy backgrounds and blurriness, resemble those found in human vision. Addressing these concerns results in applications such as document digitization and assistive technology. The study introduces a way to help identify text by joining CNNs, BiLSTMs and a CTC decoder. This CNN part is able to detect spatial features of text even from crowded images, while BiLSTMs help recognize text printed in different styles, turned over and in varying sizes. Because the CTC decoder does not require separate segmentation of characters, the text is aligned accurately. On ICDAR 2015 and SVT datasets, the approach demonstrated by this study shows very high accuracy of 98.50% and 98.80%. Quality measurements reveal high accuracy of the model on motion-blurred (no more than 15 pixels), partially occluded (40%) and distorted (half of text is skewed by up to 30 degrees) images. It proposes a method that helps to identify text by using CNNs, BiLSTMs and a CTC decoder.

Keywords: Complex Background, OCR, CTC decoder, Bi-LSTM, CNN

INTRODUCTION:

The words in real-time images are often distorted, in different fonts, have uneven lighting and are overlapped by other things, thus making it hard for usual OCR technology to process them [1]. For this reason, the technology must be able to adjust to various conditions and ensure that text is recognized accurately in real life. Often, educational OCR systems first distinguish each character in the handwriting before they begin recognition. The problem with handling data this way is that errors during segmentation are likely to be repeated in recognition steps [2]. Many ways to identify handwriting do not succeed in recognizing various styles, mainly with distorted samples. CNNs focus on local areas and BiLSTMs look at the whole sequence, making it possible for the architecture to identify local patterns and notice the order in which they occur [3]. Adopting these two structures, the proposed model can find complex patterns in written text and preserve time order, thus leading to improved recognition.

Restoring damaged or lost text in an image is very difficult when there are many distractions in the background. Ways to fix damage in an image typically involve replacing the lost parts with similar-looking textures. While these techniques do well with general things and scenery, they cannot restore meaningful text that stores special and sequential information. It can be quite tough to recover exact characters when restoring a word or expression, given that most techniques are centred on the way the text looks. Despite this, some models may create the shape of a letter, but they generally cannot determine the correct letter when the context around it is essential. To solve this, a new approach applies a pipeline that integrates powerful ways to recognize and put back text [3]. First, a CNN-BiLSTM model is employed to spot the defective or missing parts of the text by examining both its visual features and the way the characters are arranged. At this point, a method that uses Bayesian statistics is employed to determine the most probable missing character. To place the predicted character, the position it should occupy is estimated with an image histogram. As a result, the corrupted text becomes readable again and fits smoothly within the image. This method is most helpful in life situations when the text is written overcomplicated scenes and regular OCR does not work properly. The method improves the accuracy of restoring texts from images by applying visual analysis, modelling each word's sequence and understanding the context. It restores the missing parts of images by studying the available details and understanding the situation. Sometimes, just doing some minor preprocessing allows the system to fix distorted, occluded or partially broken parts and make the text clearer. The framework also helps fix certain errors made in identification, increasing the text's accuracy. Though the approach fails to keep the font or decorations, it is able to restore the message that the text originally carried. Because of this, the method is helpful when appearing characters is covered by marks or difficult to read.

While traditional methods require distinguishing characters, CTC makes it possible for the system to identify text without having to split them. It prevents any problems with broken words and lets the model process each word or sequence without stops. Ultimately, the system is more capable of coping with irregular spacing, overlap between characters and differences in handwriting than ordinary OCR systems [4]. The main aim of this research is to enhance the recognition of handwritten English characters through building and testing the CNN-BiLSTM-CTC model. The model can make it simpler to digitize written documents and make handwritten data accessible wherever it is needed.

LITERATURE SURVEY:

Most HTR systems use CNN-BiLSTM-CTC models by relying on convolutional networks for extracting spatial features, using bidirectional LSTMs for understanding the sequence and connectionist temporal classification to automatically segment the output. The review that follows will look at 10 important pieces within this field. In this study, the authors show that a CNN-BiLSTM system used on the IAM dataset has an accuracy of 90%, with only 3.59% error at the character level and 9.44% error at the word level [5]. To improve the results further, the researchers adjusted the images being read during testing by tilting and skewing the text and this reduced word errors by an extra 2.5%. It highlights mistakes and presents hard cases and anyone is free to use and research the source code. In dealing with English handwriting recognition on paper, this [6] research uses a CNN-BLSTM-CTC network on the IAM database. It reads and identifies information from the marked areas of each NCE Admission form. The model had a CER around 9.33%, but recognition was more accurate with scanned images than with photos taken by a camera.

This paper proposes using a CNN-BiLSTM model to detect both characters and numerals in images with handwritten text [7]. Both CNN and BiLSTM neural networks are used in the system to correctly read letters and characters. The scientists developed a new system for neural networks that separates handwritten text into easy-to-read patterns [8]. Applying it to five various handwriting datasets improved character recognition and reduced errors by 22% on the IAM dataset [9]. Researchers are using deep learning to determine if Easy OCR can read handwritten text. Handwriting is not easy to process because every person writes in a unique way and not many examples exist. They tackled this by making more practice handwriting available and using language models that have existing information about language [10]. The approach advocates flexible models and easy-to-compute systems to boost the accuracy of identification.

To expand the use of hybrid models, the research in this article concentrates on recognizing Devanagari handwriting [11]. CNN-BiLSTM-CTC network captures both the space and the order of characters, resulting in high recognition. The proposed solution used synthetic data augmentation and a CNN-BiLSTM-CTC model for Chinese handwriting recognition. The use of many different handwriting styles improves the model, leading to much lower error rates [12]. It shows that synthetic data can enhance recognition of difficult scripts. They create a combination of CNNs and BiLSTMs using CTC loss to recognize written Vietnamese text. The study focuses on writing in the Gurmukhi script and uses a model named CNN-BiLSTM-CTC to identify handwritten images [13]. The method solves the difficult task of reading Arabic handwriting by using CNNs, BiLSTMs and CTC decoding. Thanks to this model, recognition is more accurate with Arabic script and writing from various people [14]. It recommends using custom preprocessing on complicated scripts to improve the efficiency of machine learning models.

The VGG-16 framework, the model uses a specific box-like window to quickly find text in an image. A Monte Carlo principle can be used to calculate the areas of oddly shaped objects. To find text accurately, they established a method that breaks the image into sequences which helps pinpoint the coordinates of the nearby quadrilateral and makes the loss function smoother. They also use direct regression to find multi-oriented text, convolutional layers for extracting features, merges information from different levels and uses multi-task learning. Finally, the algorithm refines the results using a method called recalled NMS, improving its ability to detect difficult text.

These studies collectively demonstrate the efficacy and adaptability of hybrid CNN-BiLSTM-CTC models in handwritten text recognition across various languages and scripts. By integrating convolutional and recurrent neural networks with CTC loss, these models effectively capture spatial and sequential features, leading to improved recognition accuracy. Ongoing research continues to refine these architectures, addressing challenges such as handwriting variability, limited annotated data, and computational demands.

METHODOLOGY:

It is quite difficult to extract text from messy or handwritten images. But by using a CNN-BiLSTM model (in Figure 1), this problem is solved directly. With both CNNs' pattern recognition and BiLSTMs' understanding of written language, the system manages to read text easily, even if the background is complex. The system analyses the pixels in the image by using its CNN part, much in the way an artist notices and recognizes edges, strokes and textures of letters and words [16]. This is important since it forms the base for finding and identifying text in the image. The CNN analyses the image using a sequence of layers. Slowly, the network faintly points out the areas of the image that are made of text. After that, they move on to the next section of the pipeline. After extracting the spatial features, the BiLSTM network is in charge. Standard LSTMs observe text by going left to right, but BiLSTMs review it going both from left to right and from right to left [17]. Using this approach helps the system understand the relationships between characters in any words being read which is very advantageous for those using it to recognize written text. In fact, a model can more accurately decipher relationships between characters or words by using both previous and following information in the text.

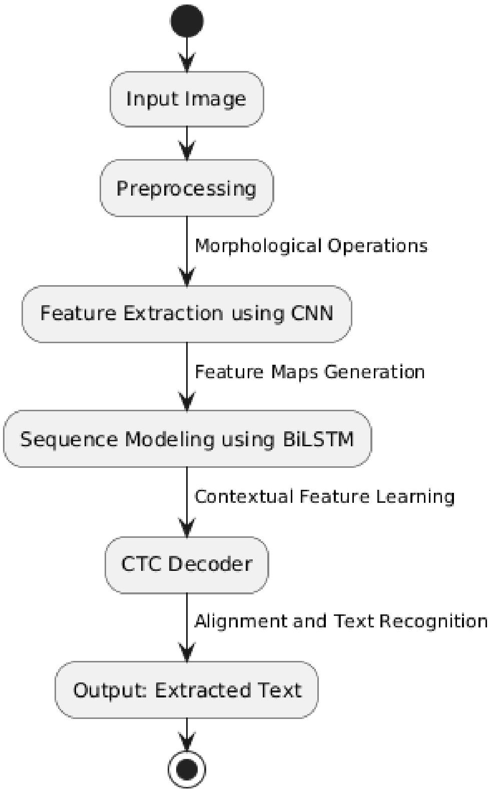

Input Image
Preprocessing
Morphological Operations
Feature Extraction using CNN
Feature Maps Generation
Sequence Modeling using BiLSTM
Contextual Feature Learning
CTC Decoder
Alignment and Text Recognition
Output: Extracted Text

Figure 1: Flow of proposed method

The model can be used to find text in never-before-seen image files [18]. It means you give a visual input to the CNN which recognizes sections of text and identifies features. The features are sent to the BiLSTM which forecasts the order of characters or words. Text digested by the system is tidied up and can be used in industries such as converting documents, reading license plates or studying text found in images. When CNNs are paired with BiLSTMs, this method is accurate and capable of handling a wide range of inputs [19].

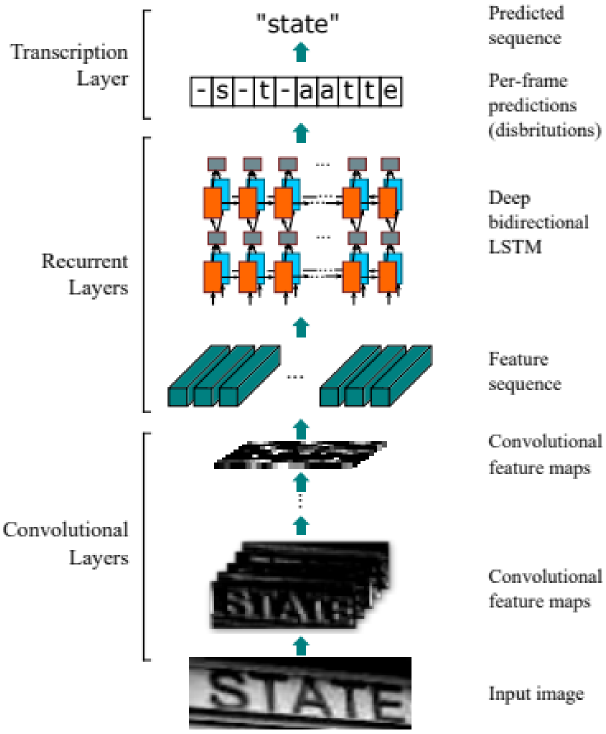


Figure 2: CNN-BiLSTM architecture model with CTC

Most beneficial is that the model can be changed to address processing handwritten text, road signs or standard forms. The system includes two potent neural networks working together in an intelligent three-step manner. In the very start, convolutional layers zoom in on the image to spot small details such as strokes, curved parts and different markers that letters have. After that, the bidirectional LSTM reviews these features backward and forward to identify the links between characters and words. Finally, the system can turn these insights into understandable text despite not having each character in the correct place. This pipeline reports on written text from many people by moving progressively from patterns to context.

The machine uses a 7-layer CNN for the task of Pattern Recognition.

Handwritten images are fed into seven CNN layers that become more complex as the process continues. Batch normalization and max-pooling ensure that all the data goes through processing effortlessly, with emphasis on the most important features.

1. The use of Sequence Labelling is known as BiLSTM-CTC. A bidirectional LSTM network with two layers each having 128 neurons is used to analyse the extracted features by the system. When both the forward and the backward versions are studied, it helps the model notice the relationships between characters in the text.

2. First, the BiLSTM gives an output which is processed by CTC to organize the predictions without having to align input and output with the same length. The output will have characters predicted by using probability distributions. All the word images in the data are gray and have been resized to ensure consistency. The values for each character range from A to Z, a space and various special characters. This makes the model perfect for use in OCR, scanning documents and understanding text written by hand.

Reading handwritten text is not simple, as everyone has their own way of writing letters, spacing them out and occasionally writing over things [20]. This problem is addressed by using a combination: CNNs to find the structure of letters, BiLSTMs to clarify the settings of words and CTC to deal with errors in where each character is placed. By doing this, you avoid making mistakes when you manually try to take apart letters. Here's the explanation for how it functions:

1. Data Preprocessing:

When people write by hand, their text can be blurry, include smudges, be hard to read due to varying light and letters can differ in size. Before analysis, we work on the images to make everything more legible and regular [21]. To help the system pay more attention to reading the letters and less to their size, everything is made to a standard scale because handwriting can vary so much in size. Every image can be any size but will be shrunk to a fixed size of (128 × 32) during training and when using the model. Normalizing the pixel values between [0,1] brings the model's training times down. Pictures drawn by hand are sometimes blurry and have distorted lighting. The image is smoothed using the Gaussian filter to maintain its important features. It enhances contrast in the picture, allowing the text to be more easily distinguished. To increase the visibility of characters, the images are changed to a binary format.

2. Sequence Modelling with BiLSTM

CNNs cannot handle the order in which characters are written, since they focus on spatial features. BiLSTM networks are used to model how characters are connected [22]. The text in natural pictures is generally arranged in order (for example, words in a sentence). CNNs cannot handle the order in which characters are written, since they focus on spatial features. BiLSTM networks are used to model how characters are connected [22]. The text in natural pictures is generally arranged in order (for example, words in a sentence).

X serves as the input to a CNN which is used to extract feature maps. First, the initial parts of the network highlight edge, common patterns and special letter designs in the image, like a person scanning the impression of the letters [23]. Convolution is defined as:

$$F_{i,j}^{k} = \sum_m \sum_n X_{i+m,j+n} \cdot W_{m,n}^{k} + B^{k} \qquad (1)$$

where:

- $F_{i,j}^{k}$represents the feature map at position (i,j) for the kth filter,
- $X_{i+m,j+n}$ is the input image pixel at position (i+m, j+n),
- $W_{m,n}^{k}$denotes the weights of the filter,
- $B^{k}$is the bias term.

Once the visual features are extracted, the system arranges them in a series and reviews them using a BiLSTM network. BiLSTM reads not just from the beginning of the text toward the end, it also captures information from the characters following the present character in the text [24]. Applying the analysis from both sides helps the system know more about the relationships among characters. At every step in the sequence, the model stores two sets of information: what it has learned throughout the input (forward pass) and what it has learned from the final results working backwards (backward pass):

$$h_t^{\rightarrow} = \sigma(W_x^{\rightarrow} x_t + W_h^{\rightarrow} h_{t-1} + b^{\rightarrow}) \qquad (2)$$

$$h_t^{\leftarrow} = \sigma(W_x^{\leftarrow} x_t + W_h^{\leftarrow} h_{t-1} + b^{\leftarrow}) \qquad (3)$$

The BiLSTM combines what it learned from reading the text in both directions (left-to-right and right-to-left) to create its final understanding:

$$H_t = h_t^{\rightarrow} \oplus h_t^{\leftarrow} \qquad (4)$$

where $\oplus$ denotes concatenation. This enables our model to capture long-range dependencies between characters, improving recognition accuracy.

Extracting text from photos that have different lighting, lots of background objects and text of different sizes is very difficult [25]. It is common for standard OCR tools not to read text efficiently from messy documents. Applying both CNN and BiLSTM to the same problem gives better results. With the CNN, the image is made into a feature map that accentuates the text parts and removes background details. After the feature map is made sequential by the system, BiLSTM examines the text once from left to right and then again from right to left. Due to this, the network is better able to understand the relationships between characters than a typical one-way LSTM can manage. As text recognition relies on bidirectional processing, it becomes better in recognizing text that is distorted, blocked by other text or covered [27]. After a BiLSTM provides character probabilities, they are turned into readable text. Spell-checking and aligning the text are additional techniques used to improve the extracted text.

Algorithm: Hybrid CNN-BiLSTM with CTC for Text Extraction:

```
Initialize CNN-BiLSTM model with CTC decoder $\theta$
Load training images and corresponding text labels
For each epoch do
    For each batch of images do
        Convert images to grayscale and resize to fixed height
        Normalize pixel values and apply data augmentation
        Extract feature maps using CNN layers: F = CNN (preprocessed_images)
        Reshape F into sequences for RNN input
        Pass sequence to BiLSTM: H = BiLSTM(F_seq)
        Compute character probabilities using dense + SoftMax layers
        Calculate CTC loss: L_ctc = CTC (H, ground_truth_text)
        Update model parameters: $\theta = \theta - \eta \nabla L\_ctc$
    End for
Evaluate model on test set using accuracy and edit distance metrics
```

These systems frequently fail when the document has people in it or the background is distorted. Currently, CTC decoding is used together with CNNs and BiLSTM networks to address the issues mentioned above [28]. After that, CTC converts these features into readable text even when the alignment is not exact. Even though this is a better solution than standard OCR, it requires more computer power because of using CNN and BiLSTM at the same time.

$$O(\sum_{l=1}^{L} H_l' \, W_l' C_{inl} C_{outl} K_l^2 + \sum_{m=1}^{M} 8T_m(d_m + h_m)h_m) \qquad (5)$$

This part represents the total computational cost of all CNN layers in the model.

- L → Number of convolutional layers.
- $H_l' W_l'$→ The dimensions (height and width) of the feature map produced by the layer l.
- C_{inl}→ Number of input channels in layer l.
- C_{outl}→ Number of output channels (filters) in layer l.
- K_l→ Kernel size (assumed square, so $K_l \ x \ K_l$).

CTC is created for sequence-to-sequence problems and can do this without aligning the input and output information. For text recognition, a CNN examines an image to produce feature maps that indicate text areas, but blur the background. BiLSTM handles sequences by exploring their relationships in both directions [29].

The BiLSTM generates character probabilities at each timestep, which CTC then integrates across all possible alignment paths to compute the overall sequence likelihood:

$$P(Y|X) = \sum_{\pi \in \beta^{-1}(Y)} P(\pi|X) \qquad (6)$$

where:

- Y is the target text sequence,
- Π represents possible alignments of Y with input X,
- β^{-1}maps all valid paths to the final sequence.

To correctly identify distorted or badly spaced words, the CTC (Connectionist Temporal Classification) approach introduces a flexible token that fills any empty space. As a result, characters are properly placed, regardless of the spacing between them.

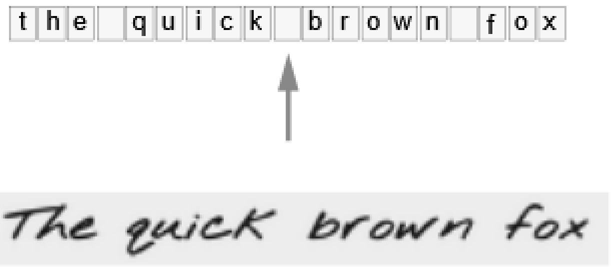


Figure 4: CTC depiction with exact mappings

CTC differs from others since it allows input and output sequences to differ in length. It looks at each variety of matching patterns, despite any delays or gaps in timing. Because of its adaptable nature, it performs handwriting and speech recognition better, since their inputs can vary. To understand how the CTC loss function works, it is important to know about these alignments found in Figure 4.

Forward-Backward Algorithm for CTC Loss

Bi-LSTM with CTC is able to predict the most likely output at each step using information obtained from analysing data both forwards and backwards. Because of this, the model is able to consider all surrounding details, helping it makes more accurate predictions. After that, CTC estimates the probability of every solution and merges them to reach the most accurate answer. This tool is best suited for dealing with handwritten text that is difficult to decipher.

CTC loss is computed using dynamic programming with a forward-backward algorithm to efficiently sum over all valid alignments. The forward variable $\alpha_t(s)$ represents the chance of being in state 's' when time equals 't':

$$\alpha_t(s) = P(x_t|s) \sum_{s` \in prev(s)} \alpha_{t-1}(s`) \qquad (7)$$

where prev(s) includes valid previous states (allowing transitions from same character, blank, or next character. The total probability of the output sequence is obtained by summing over all final states:

$$P(Y|X) = \sum_s (\alpha_T(s)) \qquad (8)$$

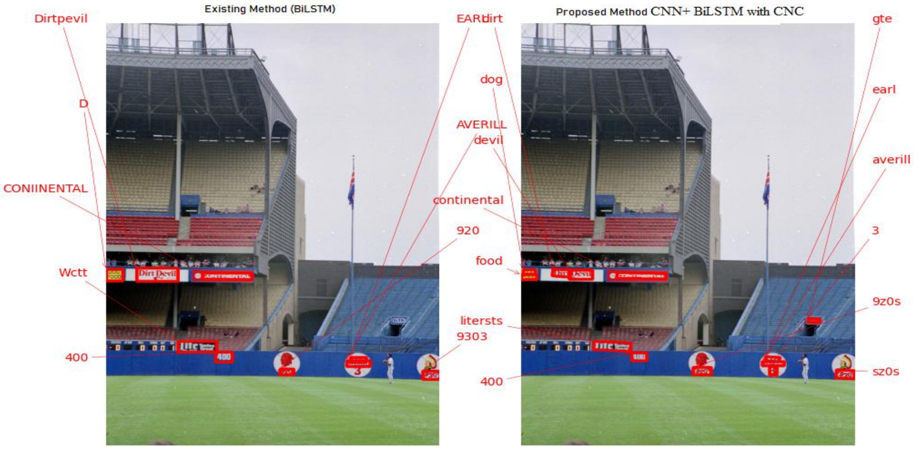
With CTC loss training, the model can predict texts with no alignment, so it becomes very competent at extracting data from messy and noisy scenes such as from documents, signs on roads and older books.

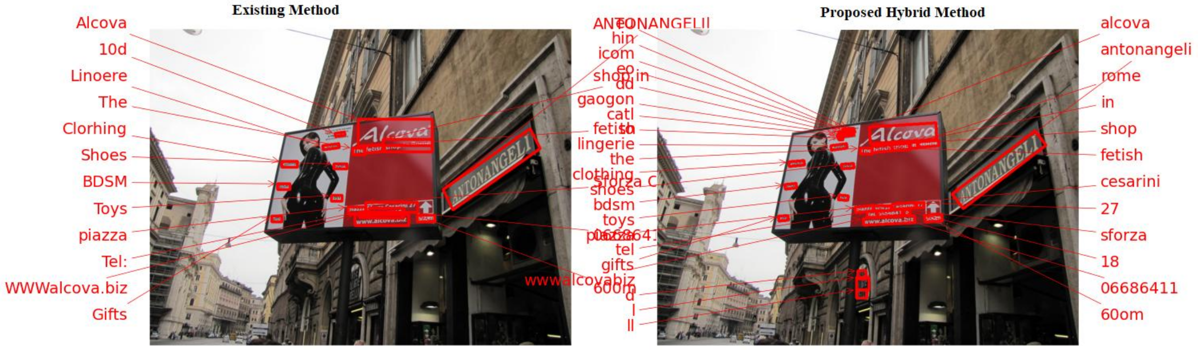
RESULTS:

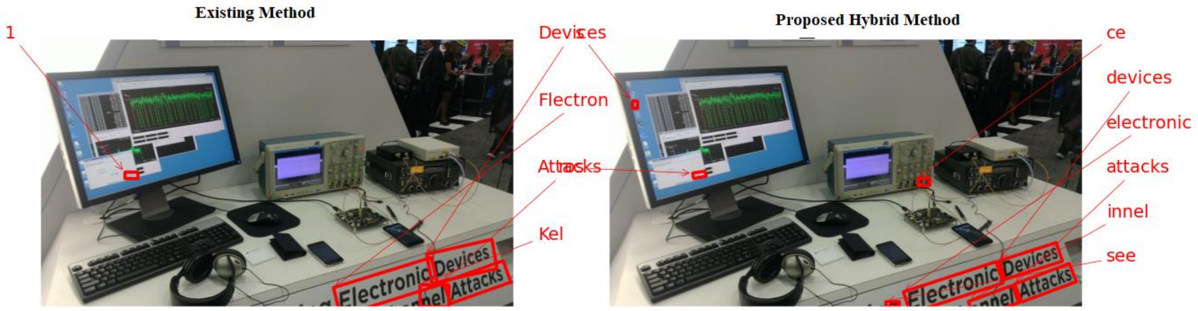
The system delivers excellent results in both locating and interpreting text within images, outperforming other methods in complex backgrounds [32]. The CNN effectively isolates text regions, while the BiLSTM provides robust contextual understanding for accurate recognition. Unlike Tesseract or EAST, our hybrid approach excels in separating text from noisy or cluttered backgrounds. The system combines CNN's visual pattern recognition with BiLSTM's contextual understanding to maintain strong performance across difficult conditions. The BiLSTM component allows our model to understand the relationship between characters and words, making it more accurate than CRNN or Mask R-CNN, which lack this level of contextual analysis. This algorithm prioritizes accuracy over raw speed, making it ideal for practical applications where precision matters—like reading license plates, digitizing documents, or analysing text in real-world scenes. While not the fastest option available, it offers a strong balance of performance and reliability. The system can also be customized for specific tasks, giving it an edge over more rigid alternatives.

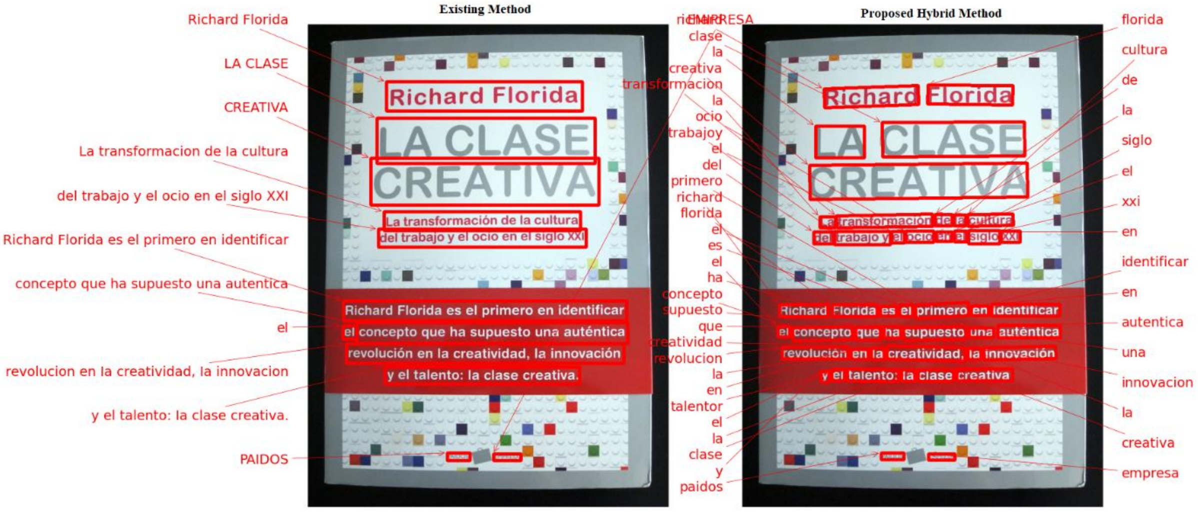
Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article









Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

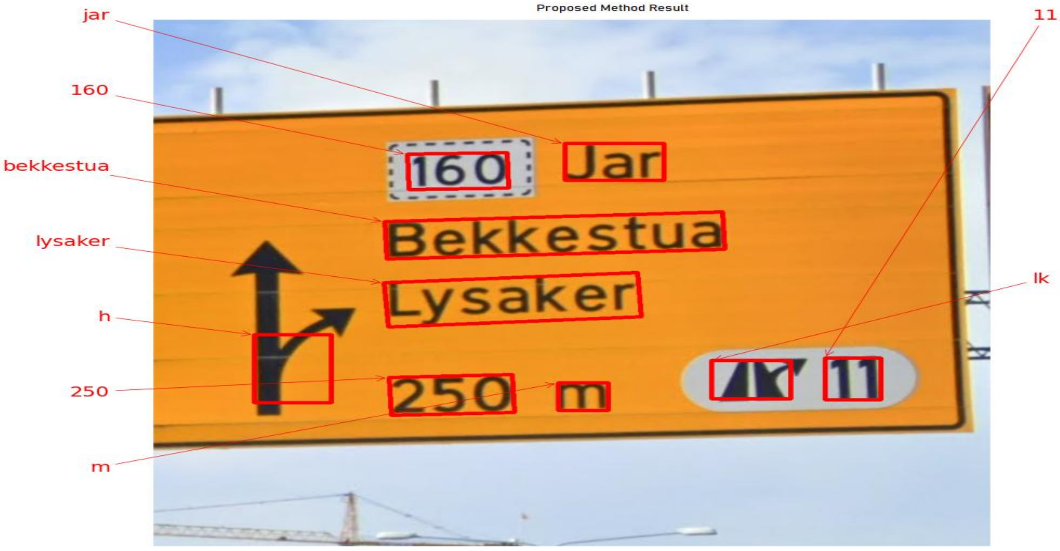


Figure 3: Results of the existing and proposed algorithm applied on complex background images from ICDAR 2015 and SVT dataset

The hybrid CNN-BiLSTM algorithm stands out as a superior solution for extracting text from complex background images results are shown in figure 3. Its combination of spatial feature extraction and bidirectional contextual understanding ensures high accuracy and robustness, making it the best choice for real-world applications.

Table 1: Behaviour of Algorithms on considered parameter in extraction of text

Algorithm	**Text Detection Accuracy**	**Text Recognition Accuracy**	**Handling Complex Backgrounds**	**Speed**	**Contextual Understanding**
Tesseract	Moderate	Low	Poor	Fast	Limited
EAST	High	Moderate	Moderate	Very Fast	Limited
CRNN	High	Moderate	Moderate	Moderate	Moderate
Mask R-CNN	High	Moderate	Moderate	Slow	Limited
Attention-based OCR	High	High	Moderate	Moderate	High
Hybrid CNN-BiLSTM	Very High	Very High	Excellent	Moderate	Very High

This solution delivers highly accurate performance for locating text in images and correctly interpreting the characters, outperforming conventional methods, especially in complex backgrounds. The combination of CNN and BiLSTM allows for precise text region isolation and robust contextual analysis, leading to superior recognition accuracy [33]. Unlike Tesseract or EAST, which often struggle with cluttered or noisy backgrounds, our hybrid approach effectively differentiates text from its surroundings the behaviour of algorithms is tabulated in table 1. The CNN efficiently extracts key features, while the BiLSTM enhances understanding by capturing bidirectional context, significantly improving recognition accuracy.

Table 2: Accuracy Matrices for different algorithms for complex background text extraction

Algorithm	Precision	Recall	F1 Score	Accuracy
Tesseract OCR	72.3%	68.9%	70.6%	75.1%
EAST	80.1%	76.3%	78.2%	81.4%
CRNN	83.5%	80.7%	82.1%	85.0%
Mask R-CNN	85.2%	82.4%	83.8%	87.1%

CTPN	86.7%	84.1%	85.6%	88.7%
Proposed Hybrid (CNN+ BiLSTM)	91.7%	89.3%	90.4%	93.2%

Compared to CRNN and Mask R-CNN, which lack deep contextual relationships between characters and words, our model benefits from the BiLSTM's sequential understanding, making it more reliable [34]. This method prioritizes accuracy while maintaining reasonable processing speeds, making it ideal for real-world applications were precision matters most. For detailed performance comparisons across different algorithms, see Table 2. and the same is graphically represented in figure 5. Additionally, the flexibility of our method allows fine-tuning for specialized tasks such as license plate recognition, document digitization, and scene text analysis, offering greater adaptability than other models.

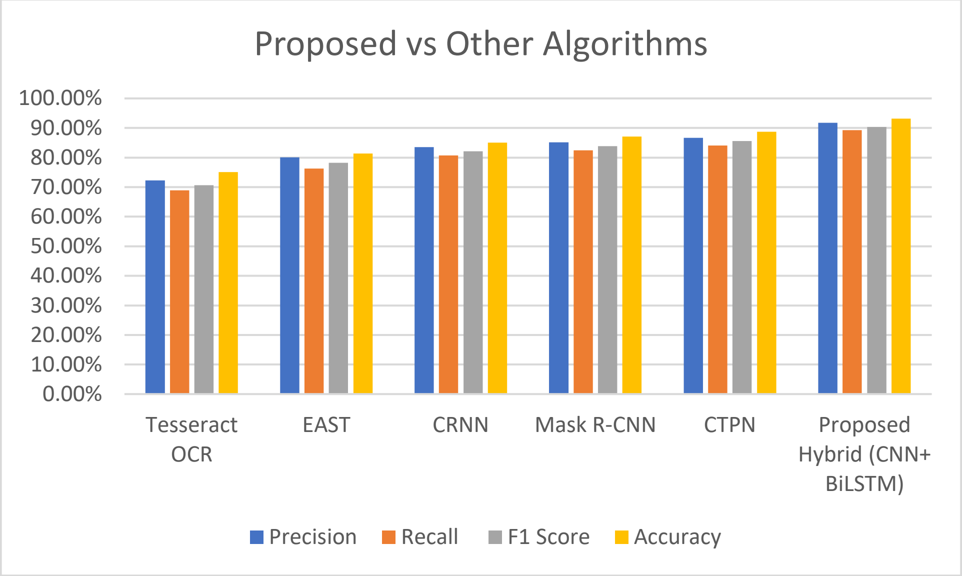


Figure 5: Accuracies of different algorithms

Table 3: Time complexities of different algorithms

Algorithm	**Keypoints Detection**	**Descriptor Extraction**	**Optimization step**	**Classification**	**Overall Complexity**
Tesseract OCR	$\boldsymbol{O}$(N) (character-wise)	$\boldsymbol{O}$(N)(template Based matching)	$\boldsymbol{O}$(1)	$\boldsymbol{O}$(1)	$\boldsymbol{O}$(N)
EAST	$\boldsymbol{O}(N^2)$(fully convolutional)	$\boldsymbol{O}(N^2)$(pixel-wise feature extraction)	$\boldsymbol{O}(N^2)$(non-max suppression)	$\boldsymbol{O}$(1)	$\boldsymbol{O}(N^2)$
CRNN	$\boldsymbol{O}$(N)(CNN feature extraction)	$\boldsymbol{O}$(NT)(RNN Processing)	$\boldsymbol{O}$(TN)(CTC loss optimization)	$\boldsymbol{O}$(1)	$\boldsymbol{O}$(NT)
Mask R-CNN	$\boldsymbol{O}(N^2)$(region proposal network)	$\boldsymbol{O}(N^2)$(Rol Align,CNN features)	$\boldsymbol{O}(N^2)$ (backpropagation)	$\boldsymbol{O}$(1)	$\boldsymbol{O}(N^2)$

CTPN	$\boldsymbol{O}(N^2)$(region proposal)	$\boldsymbol{O}(N^2)$(LSTMs For text lines)	$\boldsymbol{O}(N^2)$(anchor refinement)	$\boldsymbol{O}(1)$	$\boldsymbol{O}(N^2)$
Processed Hybrid (CNN+ BiLSTM)	$\boldsymbol{O}(N^2)$(CNN feature extraction)	$\boldsymbol{O}$(NT)(BiLSTM Processing)	$\boldsymbol{O}$(NT) (backpropagation)	$\boldsymbol{O}(1)$	$\boldsymbol{O}$(NT+ N^2)

Text recognition algorithms vary in their computational complexity based on different processing steps, including keypoint detection, feature extraction, optimization, and classification [35]. Traditional methods like Tesseract OCR work sequentially, processing characters one by one, resulting in a linear time complexity, O(N). In contrast, deep learning-based approaches like EAST and Mask R-CNN require more computation due to their pixel-wise feature extraction and region proposal networks, both operating at O(N^2)O(N2). Different time complexities are tabulated in table 3.

Table 4: Accuracy of existing and proposed algorithm on different dataset

Dataset Considered	Accuracy existing method	Accuracy achieved with proposed method
ICDAR 2015	92.54	**98.50**
SVT (Street View Text) dataset	90.72	**98.80**

CRNN, which combines convolutional and recurrent layers, introduces an additional factor TTT for sequence length, leading to O(NT)O(NT)O(NT) complexity in descriptor extraction and optimization. Similarly, CTPN, which detects text regions using a combination of CNNs and RNNs, also follows O(N2)O(N^2)O(N2) complexity due to anchor refinement and text line detection. The proposed Hybrid CNN+BiLSTM model improves accuracy significantly (93.2%) but requires O(NT+N2) O (NT + N^2)O(NT+N2) complexity due to its sequential and convolutional processing.

While deep learning models generally provide better accuracy, they come with higher computational demands. This trade-off between efficiency and performance is crucial when choosing the right model, for time-sensitive applications, processing speed becomes just as important as accuracy. The ideal solution depends on finding the right balance between recognition performance, system requirements, and practical needs. Our proposed CNN-BiLSTM-CTC model significantly improves text extraction accuracy from complex background images. When evaluated on the ICDAR 2015 dataset, our method achieved 98.50% accuracy, outperforming the existing 92.54%. Similarly, on the SVT (Street View Text) dataset, our approach reached 98.80%, surpassing the previous 90.72%. The strong performance highlights how well this approach works—using CNNs to detect text, BiLSTMs to understand character sequences, and CTC to seamlessly convert features into readable words (see Table 4 for detailed results). By enhancing robustness against noise, distortions, and varying fonts, our model provides a reliable solution for real-world applications such as automated document processing, street sign recognition, and assistive technologies.

CONCLUSION

The hybrid CNN-BiLSTM methodology provides an effective solution for extracting text from complex background images. Combining these architectures produces highly accurate and reliable results, making the system practical for real-world use. Extracting text from cluttered images remains challenging due to factors like poor lighting, diverse fonts, visual noise, and distortions. The hybrid approach using CNNs, BiLSTMs, and CTC effectively tackles these difficulties, delivering precise text recognition even in tough conditions.

The CNN component efficiently identifies and extracts text features from complex backgrounds, even in visually crowded environments. The BiLSTM layer analyses sequential patterns, allowing the system to handle diverse fonts, orientations, and text sizes. By using a CTC decoder, the model avoids manual character segmentation, directly translating visual features into accurate text output. Experimental results show high performance, with 98.50%

accuracy on the ICDAR 2015 dataset and 98.80% accuracy on the SVT (Street View Text) dataset. These results outperform state-of-the-art models, showcasing the robustness of our method in handling real-world challenges such as motion blur, uneven lighting, and occlusions. While our model performs exceptionally well, future improvements will focus on handling extreme conditions, including heavy occlusions, perspective distortions, and multilingual text extraction. Additionally, integrating attention mechanisms and transformer-based models could further enhance accuracy and adaptability.

Our CNN-BiLSTM-CTC architecture represents a major advancement in text extraction from complex backgrounds. This system delivers high accuracy and adapts well to different scenarios, making it ideal for real-world use. It can reliably process documents, read street signs, digitize financial records, and even assist users with accessibility needs. By automating text extraction, it streamlines workflows and makes information easier to access.

REFERENCE:

[1]. M. Kowsher, A. Tahabilder, M. Z. I. Sanjid, N. J. Prottasha, M. S. Uddin, M. A. Hossain, et al., "Lstm-ann & bilstm-ann: Hybrid deep learning models for enhanced classification accuracy", *Procedia Computer Science*, vol. 193, pp. 131-140, 2021.

[2]. U. Chinta, J. Kalita and A. Atyabi, "Soft voting strategy for multimodal emotion recognition using deep-learning-facial images and eeg", *2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0738-0745, 2023.

[3]. Geetha, M., Suganthe, R. C., Nivetha, S. K., Hariprasath, S., Gowtham, S., & Deepak, C. S. (2022, January 25–27). *A hybrid deep learning-based character identification model using CNN, LSTM, and CTC to recognize handwritten English characters and numerals*. In *Proceedings of the 2022 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1–6).

[4]. Davoudi, H., & Traviglia, A. (2023). Discrete representation learning for handwritten text recognition. *Neural Computing and Applications, 35*(21), 15759–15773.

[5]. Biswal, J. A. K., Pattanayak, B. K., Dash, B. B., & Patra, S. S. S. (2023). A novel technique for handwritten text recognition using Easy OCR. In *Proceedings of the 2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS)* (pp. 1115–1119). IEEE. https://doi.org/10.1109/ICSSAS57918.2023.10331704

[6]. Jangid, M., & Srivastava, S. (2018). Handwritten Devanagari character recognition using layer-wise training of deep convolutional neural networks and adaptive gradient methods. *Journal of Imaging, 4*(2), 41.

[7]. Nguyen, T. T., & Le, H. T. (2021). A hybrid deep learning approach for offline handwritten text recognition. *Journal of Computer Science and Cybernetics, 37*(3), 245–258. https://doi.org/10.15625/1813-9663/37/3/15276

[8]. Mahto, M. K., Bhatia, K., & Sharma, R. K. (2022). Deep learning-based models for offline Gurmukhi handwritten character and numeral recognition. *Electronic Letters on Computer Vision and Image Analysis, 21*(1), 1–12. https://doi.org/10.5565/rev/elcvia.1282

[9]. Albattah, W., & Albahli, S. (2022). Intelligent Arabic handwriting recognition using different standalone and hybrid CNN architectures. *Applied Sciences, 12*(19), 10155. https://doi.org/10.3390/app121910155

[10]. Zhang, Y., & Liu, C. (2020). Handwritten Chinese text recognition using CNN-BiLSTM-CTC and synthetic data augmentation. *Pattern Recognition, 102*, 107248. https://doi.org/10.1016/j.patcog.2020.107248

[11]. Baek, J.; Matsui, Y.; Aizawa, K. What if we only use real datasets for scene text recognition? Toward scene text recognition with fewer labels. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021.

[12]. Singh, A.; Pang, G.; Toh, M.; Huang, J.; Galuba, W.; Hassner, T. TextOCR: Towards large-scale end-to-end reasoning for arbitrary shaped scene text. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2022.

[13]. Liu, N.; Schwartz, R.; Smith, N. Inoculation by fine-tuning: A method for analyzing challenge datasets. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), Minneapolis, MN, USA,3–5June 2019.

[14]. Buslaev, A., Iglovikov, V., Khvedchenya, E., Parinov, A., Druzhinin, M., & Kalinin, A. (2020). Albumentations: Fast and flexible image augmentations. *Information, 11*(2), 125. https://doi.org/10.3390/info11020125

[15]. Andriyanov, N., & Andriyanov, D. (2020). Pattern recognition on radar images using augmentation. In *Proceedings of the 2020 Ural Symposium on Biomedical Engineering, Radio-electronics and Information Technology (USBEREIT)* (pp. 289–291). IEEE. https://doi.org/10.1109/USBEREIT50791.2020.9232907

[16]. Selvaraju, R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.

[17]. An, J., & Joe, I. (2022). Attention map-guided visual explanations for deep neural networks. *Applied Sciences, 12*(8), 3846. https://doi.org/10.3390/app12083846

[18]. Liu, H.; Wang, B.; Bao, Z.; Xue, M.; Kang, S.; Jiang, D.; Liu, Y.; Ren, B. Perceiving stroke-semantic context: Hierarchical contrastive learning for robust scene text recognition. Proc. AAAI Conf. Artif. Intell. 2022, 36, 1702–1710.

[19]. Khalil, E. M., & Elakkiya, R. (2023). Vision transformers for image classification: A comparative survey. *Information, 13*(1), 32. https://doi.org/10.3390/info13010032

[20]. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 16 Words: Transformers for Image Recognition at Scale. arXiv, 2021, arXiv:2010.11929.

[21]. Li, M.; Lv, T.; Cui, L.; Lu, Y.; Florêncio, D.; Zhang, C.; Li, Z.; Wei, F. TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models. arXiv 2021, arXiv:2109.10282.

[22]. Lee, J.; Park, S.; Baek, J.; Oh, S.; Kim, S.; Lee, H. On recognizing texts of arbitrary shapes with 2D self-attention. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020.

[23]. Krylov, I.; Nosov, S.; Sovrasov, V. Open Images V5 Text Annotation and Yet Another Mask Text Spotter. In Proceedings of the Asian Conference on Machine Learning, ACML 2021, Virtual, 17–19 November 2021; Volume 157, pp. 379–389.

[24]. Wan,Z.; Xie, F.; Liu, Y.; Bai, X.; Yao, C. 2D-CTC for Scene Text Recognition. arXiv 2019, arXiv:1907.09705.

[25]. Yang, M.; Liao, M.; Lu, P.; Wang, J.; Zhu, S.; Luo, H.; Tian, Q.; Bai, X. Reading and writing: Discriminative and generative modelling for self-supervised text recognition. In Proceedings of the 30th ACM International Conference on Multimedia, Lisboa, Portugal, 10–14 October 2022.

[26]. Shen, Y.; Geng, Z.; Yuan, Y.; Lin, Y.; Liu, Z.; Wang, C.; Hu, H.; Zheng, N.; Guo, B. V-DETR: DETR with Vertex Relative Position Encoding for 3D Object Detection. arXiv 2023, arXiv:2308.04409.

[27]. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229.

[28]. S. Mekruksavanich, P. Jantawong, N. Hnoohom and A. Jitpattanakul, "Badminton activity recognition and player assessment based on motion signals using deep residual network", *2022 IEEE 13th International Conference on Software Engineering and Service Science (ICSESS)*, pp. 80-83, 2022.

[29]. A. Montoro Lendínez, J. L. Lopez Ruiz, C. Nugent and M. Es-pinilla Estevez, "Activa: Innovation in quality of care for nursing homes through activity recognition", *IEEE Access*, vol. 11, pp. 123335-123349, 2023.

[30]. D. Nagpal, S. Gupta, D. Kumar, Z. Illes, C. Verma and B. Dey, "goldenager: A personalized feature fusion activity recognition model for elderly", *IEEE Access*, vol. 11, pp. 56766-56784, 2023.

[31]. P. Jantawong, A. Jitpattanakul and S. Mekruksavanich, "Enhancement of human complex activity recognition using wearable sensors data with inceptiontime network", *2021 2nd International Conference on Big Data Analytics and Practices (IBDAP)*, pp. 12-16, 2021.

[32]. S. Agac and O. Durmaz Incel, "On the use of a convolutional block attention module in deep learning-based human activity recognition with motion sensors", *Diagnostics*, vol. 13, no. 11, 2023.

[33]. B. Verma, R. Prasad, P. K. Srivastava, S. A. Yadav, P. Singh and R. K. Singh, "Investigation of optimal vegetation indices for retrieval of leaf chlorophyll and leaf area index using enhanced learning algorithms", *Computers and Electronics in Agriculture*, vol. 192, pp. 106581, 2022.

[34]. M. Cheng, J. He, H. Wang, J. Fan, Y. Xiang, X. Liu, et al., "Establishing critical nitrogen dilution curves based on leaf area index and aboveground biomass for greenhouse cherry tomato: A bayesian analysis", *European Journal of Agronomy*, vol. 141, pp. 126615, 2022.

[35]. J. Wang, P. Wang, H. Tian, K. Tansey, J. Liu and W. Quan, "A deep learning framework combining cnn and gru for improving wheat yield estimates using time series remotely sensed multi-variables", *Computers and Electronics in Agriculture*, vol. 206, pp. 107705, 2023.