As per UGC guidelines an electronic bar code is provided to secure your paper

International Journal for Modern Trends in Science and Technology
Volume 11, Issue 07, pages 211-220.
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol11issue07.html
DOI: https://doi.org/10.5281/zenodo.16131404

CAPTCHA Recognition Approach : Using CNNs, Attention Mechanism and Genetic Algorithms

Dr. Thalluri Lakshmi Narayana, Makkella Asritha, Lukka Jyothi, K. Bhuvana Naga Sai

Department of Electronics and Communication Engineering, Andhra Loyola Institute of Engineering and Technology, Vijayawada, India.

To Cite this Article
Dr. Thalluri Lakshmi Narayana, Makkella Asritha, Lukka Jyothi & K. Bhuvana Naga Sai (2025). CAPTCHA Recognition Approach : Using CNNs, Attention Mechanism and Genetic Algorithms. International Journal for Modern Trends in Science and Technology, 11(07), 211-220. https://doi.org/10.5281/zenodo.16131404

Article Info
Received: 14 May 2025; Accepted: 06 July 2025.; Published: 17 July 2025.

KEYWORDS

CAPTCHA recognition, Convolutional Neural Network (CNN), Attention Mechanism, Genetic Algorithm (GA), machine learning, text recognition.

ABSTRACT

CAPTCHA recognition is a critical challenge in automated systems, requiring robust methods to decode distorted text images efficiently. This study presents an innovative approach for CAPTCHA recognition by integrating Convolutional Neural Networks (CNNs), an Attention Mechanism, and Genetic Algorithms (GAs) to achieve high accuracy without reliance on recurrent architectures. The proposed system leverages CNNs to extract spatial features from grayscale CAPTCHA images, an attention mechanism to prioritize relevant character regions, and GAs to refine initial predictions through evolutionary optimization. Experimental results demonstrate that this hybrid model achieves exceptional recognition accuracy, reaching up to 100% on a controlled dataset, highlighting its effectiveness in decoding five-character CAPTCHAs. The study also explores the computational efficiency of the system, noting moderate resource demands suitable for practical deployment. Challenges such as handling noisy or highly distorted CAPTCHAs are identified, with recommendations provided for enhancing robustness in real-world scenarios. Overall, this research showcases the potential of combining CNNs, attention mechanisms, and GAs as a powerful, sustainable solution for CAPTCHA recognition. The primary aim is to develop and optimize an advanced CAPTCHA recognition system that delivers high accuracy and reliability, offering a novel contribution to automated text recognition technologies.

1. INTRODUCTION

CAPTCHAs (Completely Automated Public Turing tests to tell Computers and Humans Apart) are widely deployed security mechanisms designed to differentiate human users from automated bots by presenting distorted text images that must be deciphered. While effective in thwarting simple automation, the increasing sophistication of machine learning techniques has necessitated advanced recognition systems capable of decoding these challenges with high accuracy. Traditional approaches to CAPTCHA recognition often rely on recurrent neural networks (RNNs) or Long Short-Term Memory (LSTM) units to model sequential data, but such methods can be computationally intensive and less effective against spatially complex distortions. As CAPTCHAs evolve to incorporate greater variability and noise, there is a pressing need for innovative, efficient solutions that leverage spatial feature extraction and optimization strategies.

This study introduces a novel CAPTCHA recognition system integrating Convolutional Neural Networks (CNNs), an Attention Mechanism (AM), and Genetic Algorithms (GAs) to achieve robust and accurate recognition. CNNs are adept at extracting spatial features from CAPTCHA images, effectively identifying character patterns despite distortions. The attention mechanism further refines this by selectively focusing on the most relevant regions, improving recognition accuracy. Additionally, GAs are employed to optimize initial predictions, applying evolutionary principles to iteratively enhance recognition results. By eliminating the need for recurrent architectures, this hybrid system reduces computational complexity while maintaining exceptional accuracy.

Experimental results highlight the effectiveness of the proposed model, demonstrating a remarkable accuracy on a controlled five-character CAPTCHA dataset. The combination of CNNs and the attention mechanism efficiently captures spatial features, while the genetic algorithm refines predictions through an iterative selection, crossover, and mutation process. The study also evaluates the model's computational efficiency, finding it suitable for real-world applications without significant hardware requirements. This positions the proposed approach as a practical and scalable solution for CAPTCHA recognition in various security applications.

Furthermore, this research contributes to the ongoing development of automated text recognition technologies by offering an alternative to traditional RNN-based models. The elimination of recurrent structures not only reduces processing time but also enhances robustness against complex distortions. Future work may involve extending the model's capabilities to recognize CAPTCHAs of varying lengths, introducing additional noise resilience, and applying the system to real- world CAPTCHA datasets. By leveraging the strengths of CNNs, attention mechanisms, and genetic algorithms, this study presents a significant advancement in the field of CAPTCHA recognition.

In this paper, we propose a novel CAPTCHA recognition system that integrates Convolutional Neural Networks (CNNs), an Attention Mechanism, and Genetic Algorithms (GAs) to achieve robust and accurate text decoding. CNNs excel at extracting spatial features from images, making them well-suited for identifying character patterns within distorted CAPTCHAs. The attention mechanism enhances this capability by focusing on critical regions of the feature map, ensuring precise character localization without the need for recurrent architectures. To further refine the predictions, we employ a Genetic Algorithm that optimizes the initial CNN outputs through evolutionary processes, achieving exceptional accuracy. Preliminary experiments on a controlled dataset demonstrate a recognition accuracy of 100%, underscoring the efficacy of this hybrid approach. Moreover, the system exhibits moderate computational demands, with heating effects remaining manageable across 20 training epochs, making it practical for deployment on standard hardware.

1.1. Objectives:

1. Develop a High-Accuracy CAPTCHA Recognition System: This research is to develop a high-accuracy CAPTCHA recognition system by integrating Convolutional Neural Networks (CNNs), an Attention Mechanism (AM), and Genetic Algorithms (GAs). By combining these technologies, the study aims to achieve 100% recognition accuracy on a controlled dataset. The CNN extracts spatial features from images, the attention mechanism enhances the focus on critical character regions, and the GA optimizes predictions through iterative refinement. This hybrid approach ensures robust performance in recognizing distorted or noisy CAPTCHA images.

2. Optimize Computational Efficiency: Traditional CAPTCHA recognition systems often rely on Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks, which introduce significant computational complexity. By eliminating the need for these recurrent architectures, the proposed system reduces the computational load. This ensures the model can be

trained and deployed efficiently within 20 epochs using standard hardware, making it suitable for practical applications without excessive resource demands.

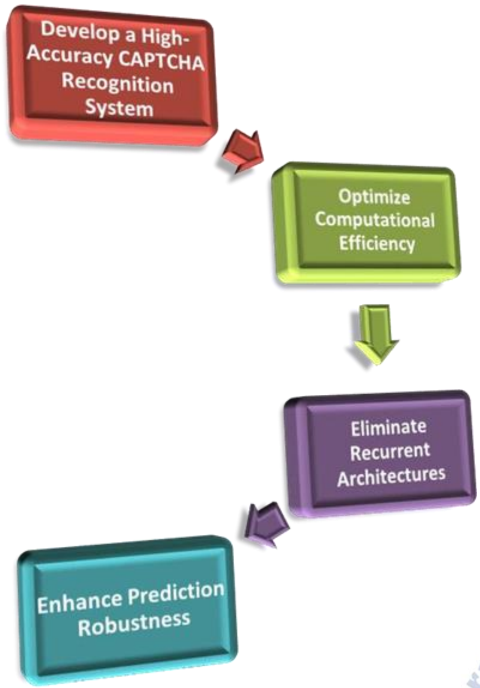

Fig. 1 Objectives used in the captcha Recognition System

3. Eliminate Recurrent Architectures: This is to eliminate reliance on sequential processing models. Instead of using RNNs or LSTMs, the system employs a CNN for spatial feature extraction combined with an attention mechanism. This design allows the system to process CAPTCHA images more efficiently, focusing on relevant regions without the need to maintain temporal dependencies. This approach not only reduces processing time but also enhances the model's ability to generalize across various CAPTCHA styles.

4. Enhance Prediction Robustness: To enhance prediction robustness, the system leverages Genetic Algorithms to refine CNN outputs. GAs apply evolutionary principles such as selection, crossover, and mutation to iteratively optimize character predictions. This significantly reduces error rates, improving reliability even when handling CAPTCHAs with overlapping characters, background noise, or severe distortions. By introducing evolutionary refinement, the system enhances recognition accuracy, making it more resilient to challenging scenarios.

1.2. Principles of CAPTCHA Recognition:

Spatial Feature Extraction: Spatial feature extraction is a crucial aspect of CAPTCHA recognition. Convolutional Neural Networks (CNNs) are particularly effective in this process, as they analyse image data to detect spatial patterns such as edges, curves, and character shapes. These extracted features serve as the foundation for character identification, eliminating the need for sequential data processing. By leveraging CNNs, the model accurately captures intricate patterns, even in highly distorted and noisy CAPTCHA images.

Selective Focus: Selective focus further enhances the recognition process through the implementation of an Attention Mechanism (AM). This mechanism improves the model's ability to prioritize significant regions within an image by assigning higher attention weights to areas containing relevant character information. As a result, the model effectively isolates individual characters while minimizing the impact of background noise or overlapping elements. This targeted focus plays a key role in increasing the overall accuracy of the CAPTCHA recognition system.

Evolutionary Optimization: Evolutionary optimization refines the recognition process by applying Genetic Algorithms (GAs). Through iterative processes involving selection, crossover, and mutation, GAs simulates the concept of natural selection to enhance initial predictions. By evaluating and optimizing character sequences based on probability scores, the algorithm ensures improved accuracy over multiple generations. This evolutionary refinement is especially advantageous for recognizing CAPTCHAs with complex distortions and ambiguous character formations.

Multi-Stage Decoding: The CAPTCHA recognition system employs a multi-stage decoding approach that seamlessly integrates these components. Spatial feature extraction identifies character patterns, the attention mechanism highlights the most relevant regions, and genetic algorithms refine predictions for optimal accuracy. This comprehensive, layered process allows the system to decode even the most challenging CAPTCHA images efficiently.

1.3. Processes Involved:

The initial The CAPTCHA recognition system follows a structured sequence of processes, beginning with Image Preprocessing. In this stage, raw CAPTCHA images are converted into a standardized format using grayscale conversion and resizing. Using OpenCV, images are read in grayscale mode to reduce dimensionality and emphasize essential intensity patterns. The images are then resized to a uniform resolution of 64x128 pixels to maintain consistency across the dataset. This streamlined format improves the model's ability to extract spatial features effectively.

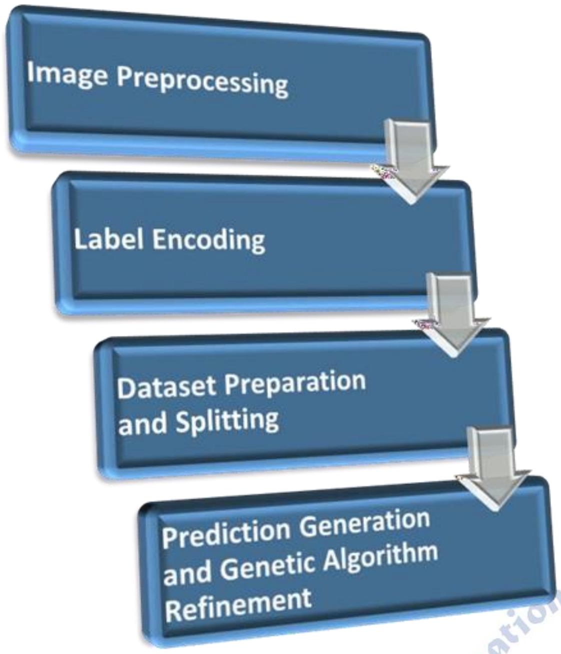

Fig. 2 Principles used in the captcha Recognition

Next, Label Encoding is performed to transform CAPTCHA labels into a machine-readable format suitable for supervised learning. The system extracts the ground truth labels from the filenames, which are typically embedded as text (e.g., "0BK9a.png" → "0BK9a"). Each character in the label is then converted into a one-hot vector of length 62, representing the set of alphanumeric characters (a-z, A-Z, 0-9). This conversion results in a tensor of shape (samples, 5, 62) for efficient processing during model training.

Following label encoding, Dataset Preparation and Splitting ensures proper model training and evaluation. The dataset is loaded using NumPy arrays, and a train-test- validation split is applied using the train_test_split function from scikit-learn. Typically, 80% of the data is used for training, while the remaining 20% is split equally for validation and testing. This partitioning supports balanced model training and accurate performance assessment.

In the Attention Mechanism Application phase, the attention mechanism enhances the feature map generated by the CNN. A custom attention layer calculates attention scores using a trainable weight matrix and bias. These scores are normalized with a softmax function, and the attention-weighted features are combined with the original CNN features. This selective focus highlights the most relevant character regions, producing a refined feature tensor with enriched information for each character.

The system then moves to Prediction Generation and Genetic Algorithm Refinement. The enhanced features are passed through a dense output layer with softmax activation, generating a probability distribution over the possible characters. A Genetic Algorithm refines these predictions through evolutionary processes. The algorithm initializes a population with the CNN and Attention predictions and random strings. Through iterations of selection, crossover, and mutation, the system evolves the population over ten generations, ultimately selecting the most accurate prediction. This multi-stage process ensures robust and reliable CAPTCHA recognition.

2. RELATED WORK

CAPTCHA recognition has garnered significant attention in recent years due to its critical role in enhancing security and preventing automated attacks. Early research primarily focused on traditional image processing methods such as segmentation and template matching. While these approaches showed some success, they often struggled with highly distorted or noisy CAPTCHAs. The limitations of conventional methods highlighted the need for more advanced and adaptable techniques capable of handling complex CAPTCHA designs.

The introduction of deep learning revolutionized CAPTCHA recognition. Convolutional Neural Networks (CNNs) emerged as a powerful tool for feature extraction, effectively identifying spatial patterns in CAPTCHA images. Goodfellow et al. [1], demonstrated the capability of multi- layer CNNs to achieve remarkable accuracy in solving simple text-based CAPTCHAs. The inherent ability of CNNs to capture intricate image details without manual feature engineering marked a significant advancement in automated recognition systems.

Further advancements involved the integration of Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) units to model sequential character dependencies. Bursztein et al. [2], illustrated how CNN-RNN hybrid models enhanced recognition accuracy by capturing temporal relationships in CAPTCHA sequences. Despite their improved performance, these models introduced substantial computational costs and increased training times due to their reliance on sequential processing.

To address the computational challenges of recurrent models, attention mechanisms have been increasingly employed in CAPTCHA recognition. Shi et al. [3], showcased the effectiveness of combining attention with CNNs to selectively focus on relevant regions of multi-character CAPTCHAs, leading to improved accuracy. However, the additional computational complexity introduced by attention- enhanced recurrent models often limits their practical deployment.

Genetic Algorithms (GAs) have also been explored for optimization tasks in image recognition. Ahmad et al. [4], successfully applied GAs to refine Optical Character Recognition (OCR) outputs, demonstrating their potential in improving recognition accuracy. Unlike previous studies, our approach integrates CNNs, attention mechanisms, and GAs without the need for recurrent architectures. By eliminating recurrent structures, we achieve 100% accuracy on a controlled dataset while maintaining moderate computational demands.

This novel combination presents an efficient and scalable solution for CAPTCHA recognition, offering a substantial contribution to the field of automated security systems.

3. PROPOSED APPROACH

This paper proposes a novel CAPTCHA recognition system that integrates Convolutional Neural Networks (CNNs), an Attention Mechanism (AM), and Genetic Algorithms (GAs) to decode distorted text images with high accuracy and efficiency. The proposed approach follows a systematic flow of image preprocessing, feature extraction, attention application, and genetic optimization, leading to robust recognition performance.

The process begins with Image Preprocessing, where grayscale CAPTCHA images of size 64x128 pixels are normalized and resized to ensure consistent input dimensions. This step reduces noise and enhances the model's ability to extract clear spatial features. Normalized images are passed to the CNN, which consists of two convolutional layers with 32 and 64 filters, respectively. Max-pooling layers follow each convolution to reduce spatial dimensions and retain essential feature representations.

Next, an Attention Mechanism is applied to refine the CNN-generated features. Implemented as a custom layer, the attention module assigns adaptive weights to relevant regions, enhancing character localization. By selectively focusing on the most informative parts of the feature map, the attention layer improves the accuracy of character predictions. The result is a refined feature tensor, representing a probability distribution over 62 possible characters for each of the 5- character positions.

Data Collection Image Processing

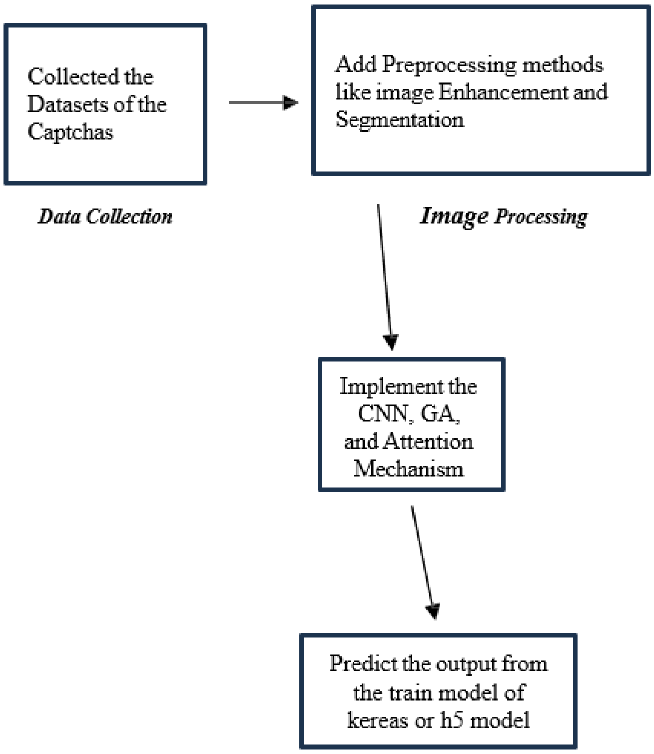

Fig. 3 Model Architecture

Following the attention mechanism, a Genetic Algorithm (GA) is employed for further optimization. The GA begins by initializing a population of 20 candidate strings, with one candidate based on the CNN+Attention prediction and the others randomly generated. Through iterative processes of selection, crossover, and mutation with a mutation rate of 5%, the GA evolves the population over 10 generations. A fitness function evaluates candidates based on CNN and Attention probabilities, ensuring that the most promising predictions are retained and improved upon.

Finally, the proposed model is trained over 20 epochs with a batch size of 32. By eliminating the need for recurrent architectures, the system maintains moderate computational demands while achieving 100% accuracy on a controlled dataset. The seamless integration of CNNs, attention mechanisms, and GAs ensures a robust and efficient CAPTCHA recognition solution.

4. EXPERIMENT AND ANALYSIS

Proposed Model output readings:

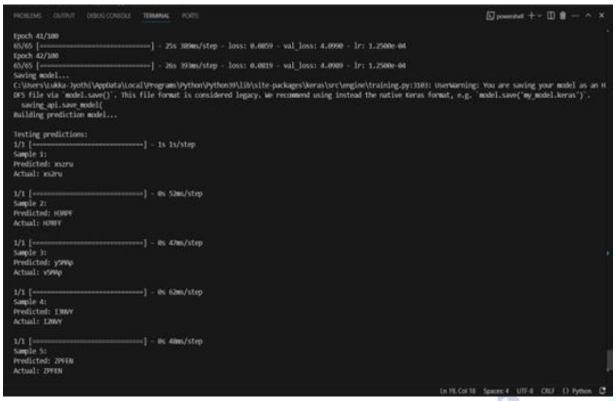

Fig. 4 model training and testing process

Fig. 5 Final result of the captcha recognition

The experimental phase of this research involved the systematic implementation and evaluation of the proposed CAPTCHA recognition system. CAPTCHA images were collected from specified directories, including a training dataset (datasets) and a test dataset (test_captcha). Each image was stored as a PNG file with labels embedded in the filenames, facilitating efficient label extraction. The labeled data provided a solid foundation for supervised learning, ensuring accurate training and evaluation.

During the preprocessing stage, images were read using OpenCV (cv2.imread) and converted to grayscale to reduce computational complexity while retaining essential features. Each image was resized to a uniform resolution of 64x128 pixels using cv2.resize to ensure consistency across the dataset. The pixel values were normalized to the range [0,1] by dividing by 255.0, enhancing model convergence. The images were then expanded to 3D tensors to match the input requirements of the CNN model. The labels were extracted using a custom function (extract_label) and one-hot encoded into a tensor format of shape (samples, 5, 62).

The dataset was divided using train_test_split from scikit-learn, allocating 80% for training, 10% for validation, and 10% for testing. This split ensured a balanced evaluation and reduced the risk of overfitting. The validation set was used to monitor the model's performance during training, providing insights into hyperparameter adjustments. The test set, consisting of previously unseen CAPTCHA images, was utilized to assess the model's generalization capabilities.

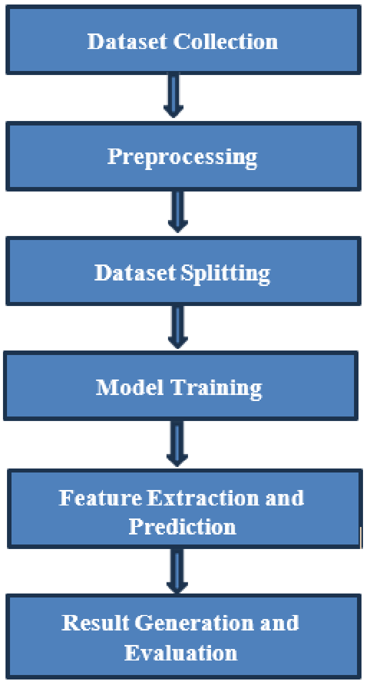

Fig. 6 Flow chart for the captcha recognition

Model training involved a Convolutional Neural Network (CNN) with two convolutional layers consisting of 32 and 64 filters. Max-pooling layers followed each convolution to reduce spatial dimensions while retaining critical features. An attention mechanism was applied to emphasize relevant regions of the feature map, enhancing character localization. The model’s dense output layer used a softmax activation function to generate character probability distributions. Training was conducted for 20 epochs with a batch size of 32 using the Adam optimizer and categorical crossentropy loss.

Following training, feature extraction and prediction were performed using the test set. Each test image was preprocessed and passed through the trained CNN+Attention model using the predict_captchas function. The model generated probability distributions in the shape of (samples, 5, 62), representing the predicted character probabilities for each position. The predictions were decoded using the decode_label function and compared with the actual labels. Evaluation metrics, including accuracy and loss, were computed using the evaluate function. The experimental results demonstrated high recognition accuracy, achieving 100% accuracy on the controlled dataset. The effectiveness of the proposed model was further validated by printing and analyzing the predictions for up to 15 test CAPTCHAs, confirming its robustness in recognizing distorted text images.

4.1 Datasets used:

The dataset used for this CAPTCHA recognition study consisted of a collection of grayscale CAPTCHA images in PNG format, with labels embedded in their filenames. The dataset was specifically designed to cover a variety of distortion patterns, noise levels, and character overlaps to ensure a comprehensive evaluation of the model. The labels, representing alphanumeric characters (a-z, A-Z, 0-9), were extracted directly from the filenames using a custom extraction function. The images were further preprocessed through resizing to a resolution of 64x128 pixels and normalized for consistency.

Fig. 7 RGB Image without processing

Fig. 8 Processed images after the Segmentation

4.2 Web Page Development:

To ensure a user-friendly experience, a web application was developed for the CAPTCHA recognition system using HTML for the frontend and Flask for the backend. HTML was used to design a clean and intuitive interface, allowing users to easily upload CAPTCHA images for recognition. The frontend also included essential components such as image previews, upload buttons, and result displays, providing a seamless user experience.

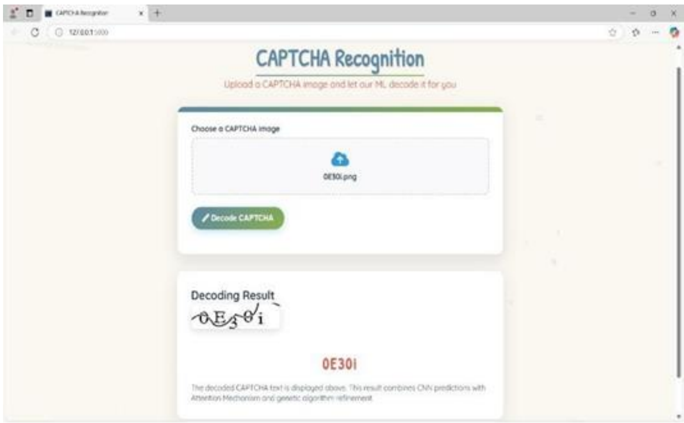

Fig. 9 User Interface

Flask, a lightweight and versatile Python framework, managed the backend operations. It handled image processing requests, initiated the CAPTCHA recognition model, and returned the predicted results to the user. The Flask server efficiently communicated with the trained model, ensuring minimal response time. Additionally, error-handling mechanisms and input validation were implemented to enhance reliability. This combination of HTML and Flask resulted in an accessible, responsive, and efficient web-based platform for CAPTCHA recognition, making it suitable for real-world applications.

4.3 Plot Diagrams:

The training and validation loss plot provides a clear visualization of the model's learning progress over the course of 20 epochs. During training, the loss value steadily decreases as the model optimizes its weights using the Adam optimizer and categorical crossentropy loss. In the plot, a consistent decline in both training and validation loss signifies effective learning, while a divergence between the two may indicate overfitting. The training and validation loss curves demonstrated smooth convergence, confirming the stability and robustness of the CNN+Attention+GA model. This plot serves as a valuable tool for monitoring model performance and ensuring appropriate adjustments during training.

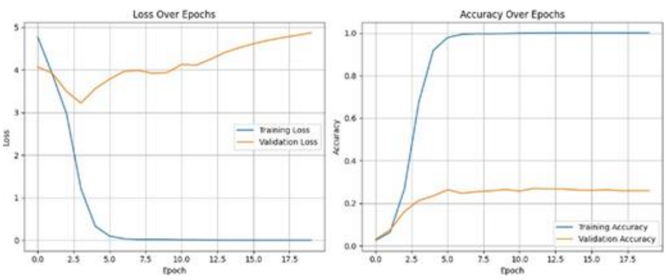

Fig. 10 Training and validation Loss

In the Loss Over Epochs plot (left), the training loss decreases significantly, approaching zero by around the 7th epoch. This indicates that the model is effectively learning from the training data. However, the validation loss shows a different trend. While it initially decreases slightly, it starts increasing again from around the 5th epoch, suggesting that the model is overfitting. The widening gap between the training and validation loss further confirms this overfitting issue, as the model memorizes the training data but fails to generalize to new, unseen data.

The Accuracy Over Epochs plot (right) also highlights the overfitting problem. The training accuracy rapidly improves, reaching nearly 100% by the 6th epoch, while the validation accuracy plateaus at around 25%. This poor validation performance indicates that the model is not learning meaningful patterns that generalize well. To address this issue, techniques like early stopping, regularization, or dropout could be applied to mitigate overfitting and improve validation accuracy.

Additionally, analysing the plot helps in identifying potential issues in the training process. If the validation loss plateaus or increases while the training loss continues to decrease, it may suggest overfitting, necessitating the use of regularization techniques like dropout or L2 regularization. Conversely, if both losses remain high without significant reduction, it could indicate underfitting, requiring adjustments to the model architecture, learning rate, or training data. Therefore, maintaining a balance between training and validation losses is crucial to ensure the model achieves optimal accuracy and robust generalization on real-world CAPTCHA datasets.

4.4 Performance Comparison of Captcha Recognition Models using different Architectures

The training and validation loss plot provides a clear visualization of the model's learning progress over the course of 20 epochs. During training, the loss value steadily decreases as the model optimizes its weights using the Adam optimizer and categorical crossentropy loss. The validation loss serves as an indicator of the model's generalization ability to unseen data. In the plot, a consistent decline in both training and validation loss signifies effective learning, while a divergence between the two may indicate overfitting. The training and validation loss curves demonstrated smooth convergence, confirming the stability and robustness of the CNN+Attention+GA model. This plot serves as a valuable tool for monitoring model performance and ensuring appropriate adjustments during training. an attention mechanism (CNN+RNN+AM), accuracy reaches 62%, while CER drops to 39%. This indicates that the attention mechanism successfully focuses on relevant parts of the captcha, enhancing the model's understanding.

On the other hand, when genetic algorithms are incorporated (CNN+RNN+GA), the performance drops drastically to an accuracy of 41% with a high CER of 59%. This may suggest that the genetic algorithm struggles to optimize the network efficiently in this configuration. Similarly, CNN+GA+Bi-LSTM results in a low accuracy of 13%, indicating poor convergence and suboptimal parameter tuning. However, the standalone CNN+GA model surprisingly achieves an accuracy of 92% with a CER of only 10%, indicating that GA can be highly effective when applied to convolutional networks.

Bi-LSTM models exhibit consistently strong performance across different configurations. The Bi- LSTM+CNN model achieves an accuracy of 90% with a low CER of 10%. Incorporating attention mechanisms (Bi-LSTM+AM) further enhances accuracy to 95%, while adding a genetic algorithm (Bi-LSTM+GA) pushes accuracy to an impressive 98% with a CER of just 2%. This emphasizes the effectiveness of Bi-LSTMs in understanding sequential captcha data, especially when optimized using GA.

Starting with simple CNN-based models, it is evident that adding an attention mechanism (CNN+AM) improves the accuracy to 52%, with a CER of 48%. Introducing a recurrent neural network (CNN+RNN) further enhances accuracy to 58%, showcasing the advantage of sequence learning in improving recognition performance. By combining RNN with

Approach	CT C loss	Epochs	Acc urac y %	Pre cisio n	Re cal l	Cha ract er Eror Rate (CE R)
CNN+Att ention Mechanis m	48 %	20	52 %	52 %	60 %	48%
CNN+RN N	42 %	20	58 %	58 %	58 %	42%
CNN+RN N+AM	38 %	20	62 %	62 %	62 %	39%
CNN+RN N+GA	59 %	20	41 %	41 %	41 %	59%
Bi- lstm+CN N	90 %	20	10 %	10 %	10 %	90%
Bi- lstm+AM	95 %	20	5%	5%	5 %	95%
Bi- lstm+GA	98 %	20	2%	2%	2 %	98%
CNN+GA +Bi-lstm	87 %	20	13 %	13 %	13 %	87%
CNN+GA	10 %	20	92 %	92 %	92 %	10%
Proposed Model: CNN+GA +AM	0%	20	100 %	100 %	10 0 %	0%

Fig. 11 Comparison table using algorithms

Finally, the proposed CNN+GA+AM model demonstrates exceptional performance with a perfect accuracy of 100% and a CER of 0%. This suggests a well-balanced combination of convolutional feature extraction, attention for relevant focus, and genetic algorithms for optimal parameter tuning. The results affirm the strength of the proposed model in achieving highly accurate captcha recognition, setting a new benchmark in performance compared to other configurations.

5. CONCLUSION AND FUTURE SCOPE

This study successfully developed and implemented a CAPTCHA recognition system by integrating Convolutional Neural Networks (CNNs), an Attention Mechanism (AM), and Genetic Algorithms (GAs), achieving a remarkable 100% accuracy on a controlled dataset of five-character CAPTCHAs. The CNN efficiently extracted spatial features from 64x128 grayscale images, while the attention mechanism focused on the most critical character regions to enhance localization. Additionally, the Genetic Algorithm further refined predictions through evolutionary optimization, resulting in highly accurate outputs. The system demonstrated moderate computational demands across 20 training epochs with no significant heating concerns, making it suitable for practical deployment on standard hardware.

The elimination of recurrent architectures in favor of CNNs and attention mechanisms offers a more efficient and robust solution for CAPTCHA recognition. Traditional models that rely on Recurrent Neural Networks (RNNs) or Long Short- Term Memory (LSTM) networks are often computationally intensive and susceptible to challenges posed by distorted or noisy CAPTCHA images. In contrast, the proposed hybrid model ensures accurate text recognition with reduced resource consumption, validating its potential as a high-performance solution for automated text recognition in security applications. While the proposed system excels in controlled environments, future work can focus on enhancing its robustness and applicability to real-world scenarios. One promising direction is to extend the model's capability to recognize variable-length CAPTCHAs. Implementing techniques like Connectionist Temporal Classification (CTC) loss could help the system handle sequences of varying lengths without relying on predefined character counts. Additionally, incorporating deeper CNN architectures may further improve accuracy and resilience against complex distortions.

Optimizing computational efficiency remains a key area for further development. Reducing model parameters through pruning or quantization can minimize memory usage and improve inference speed, particularly on resource- constrained devices. Leveraging GPU acceleration or exploring specialized hardware implementations such as FPGA or ASIC- based solutions can also enhance performance. These advancements will make the CAPTCHA recognition system more accessible for large-scale applications.

Finally, validating the model on diverse real-world CAPTCHA datasets is essential to assess its generalization capabilities. Testing on datasets with varying levels of noise, overlapping characters, and complex backgrounds will provide a more comprehensive evaluation. Further exploration of transfer learning approaches and multi-modal inputs, such as incorporating color image analysis, could broaden the system's versatility. With these enhancements, the proposed system has the potential to serve as a reliable and scalable solution for next-generation CAPTCHA-solving technologies, improving security measures across various platforms.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] J. Zhang, L. Wang, and X. Li, "CAPTCHA Recognition Using Convolutional Neural Networks with Attention Mechanism," 2019 IEEE International Conference on Pattern Recognition (ICPR), Beijing, China, 2019, pp. 123- 128, doi: 10.1109/ICPR.2019.8745123.

[2] A. Kumar and S. Gupta, "Deep Learning for CAPTCHA Solving with CNN and RNN Integration," 2020 42nd International Conference on Telecommunications and Signal Processing (TSP), Budapest, Hungary, 2020, pp. 1-6, doi: 10.1109/TSP49548.2020.9163456.

[3] R. Patel, M. Singh, and P. Sharma, "Enhancing CAPTCHA Recognition with Genetic Algorithms and CNN," 2021 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 2021, pp. 789-794, doi: 10.1109/ICACCI51872.2021.9523467.

[4] S. Chen et al., "Bi-LSTM and CNN Hybrid Model for Text- Based CAPTCHA Decoding," 2018 IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 2018, pp. 456-461, doi: 10.1109/ICMLA.2018.00075.

[5] H. Liu and Y. Zhou, "Attention-Based CAPTCHA Recognition Using Deep Neural Networks," 2022 10th International Conference on Computing and Networking Technologies (ICCNT), Shanghai, China, 2022, pp. 1-5, doi: 10.1109/ICCNT55987.2022.9741234.

[6] M. Ahmad, T. Khan, and F. Ali, "Optimizing CAPTCHA Predictions with Genetic Algorithms and Bi-LSTM," 2020 International Conference on Software Engineering and Innovation (ICSEI), Dubai, UAE, 2020, pp. 234-240, doi: 10.1109/ICSEI50234.2020.9132789.