Captcha Solving Using AI with Lime Explainability

"Where AI Meets Clarity in CAPTCHA Recognition."

[1]Sanapala Joshika [2] Setti Sarika
[1]Student [2]Research Scholar
Department Of Information Technology and Computer Applications
Andhra University College Of Engineering, Andhra University, Visakhapatnam

Abstract: This paper presents an intelligent system for solving CAPTCHA challenges using Artificial Intelligence (AI) integrated with explainable frameworks. CAPTCHAs, designed to differentiate humans from bots, often pose accessibility and usability issues. To address this, we developed a deep learning model capable of accurately recognizing and solving both alphanumeric and math-based text CAPTCHAs. The model utilizes Convolutional Neural Networks (CNNs) for image-based text recognition, trained on synthetically generated CAPTCHA datasets. To enhance transparency and trust in AI predictions, the system incorporates LIME (Local Interpretable Model-agnostic Explanations), which visually explains each character prediction by highlighting important regions of the CAPTCHA image. This interpretability aids developers in validating the model's decisions and ensures robustness against adversarial inputs. The system aims to balance accuracy, security, and explainability, making it suitable for real-world applications where both user experience and AI accountability are critical.

Index Terms : Artificial Intelligence (AI), CAPTCHA, Deep Learning, Convolutional Neural Networks (CNN), Explainable AI (XAI), LIME, Model Interpretability.

I. INTRODUCTION

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) plays a crucial role in online security by distinguishing human users from automated bots. With advancements in Artificial Intelligence, especially Deep Learning, solving CAPTCHAs has become increasingly possible. This project leverages Convolutional Neural Networks (CNN) to recognize and solve image-based text and math CAPTCHAs. To enhance trust and transparency, the model incorporates LIME (Local Interpretable Model-agnostic Explanations), providing clear visual justifications for each prediction.

Research Objectives

- To develop an AI-based model capable of accurately solving both text-based and math-based CAPTCHA images.
- To apply Convolutional Neural Networks (CNN) for recognizing alphanumeric and mathematical expressions in CAPTCHA images.
- To enhance the interpretability of the model’s predictions using LIME (Local Interpretable Model-agnostic Explanations).

- To evaluate the performance of the model based on character-wise and overall CAPTCHA accuracy.
- To create a user-friendly interface for visualizing predictions and corresponding LIME explanations.
- To demonstrate the importance of explainable AI in enhancing transparency and user trust.
- To explore how AI models can be securely and ethically applied in cybersecurity-related tasks like CAPTCHA solving.

Research Hypothesis

Integrating explainable AI techniques like LIME with a CNN-based CAPTCHA solver will not only improve the model's transparency and interpretability but also maintain high prediction accuracy, thereby enhancing user trust and making the system more reliable for real-world CAPTCHA verification and cybersecurity applications.

II. ABBREVATIONS AND ACRONYMS

AI	Artificial Intelligence
CAPTCHA	Completely Automated Public Turing Test to Tell Computers and Humans Apart
CNN	Convolutional Neural Network
LIME	Local Interpretable Model-Agnostic Explanations
RGB	Red Green Blue (color model)
XAI	Explainable Artificial Intelligence
UI	User Interface

III. LITERATURE REVIEW

CAPTCHAs are widely used to differentiate human users from automated bots. Over the years, CAPTCHA designs have evolved from simple distorted texts to more complex image-based or logic-based challenges. Traditional CAPTCHA solving methods involved rule-based algorithms and optical character recognition (OCR), which lacked robustness and scalability.

With the emergence of Artificial Intelligence (AI), deep learning models—particularly Convolutional Neural Networks (CNNs)—have demonstrated remarkable efficiency in recognizing patterns in noisy, distorted CAPTCHA images. Studies such as those by Goodfellow et al. and more recent ones using TensorFlow and Keras frameworks have successfully trained neural networks to achieve high accuracy in CAPTCHA decoding. However, most of these models operate as "black boxes," providing little insight into how predictions are made.

To address this interpretability issue, explainable AI (XAI) methods like LIME (Local Interpretable Model-Agnostic Explanations) and SHAP have been introduced. LIME, in particular, explains model predictions by approximating the local decision boundary with interpretable models. Recent research integrates LIME with CAPTCHA solvers to visually demonstrate which parts of the image influenced each character prediction. This not only enhances model transparency but also aids developers in identifying model weaknesses and bias.

The fusion of CNN-based CAPTCHA solvers with LIME has set a foundation for secure, ethical, and user-aware AI systems. This approach addresses both technical performance and the critical need for AI interpretability, which is essential in cybersecurity and real-world deployments.

IV. METHODOLOGY

This section outlines the research methods, data collection procedures, analysis techniques, and ethical considerations followed in the development and evaluation of captcha solving using AI with LIME explainability system.

Research Methods

i. Dataset Generation and Preprocessing

A synthetic dataset of 4-character alphanumeric CAPTCHA images was generated using the captcha Python library. Each image was labeled according to the embedded CAPTCHA text. Preprocessing steps included resizing images to a fixed resolution (160×60), normalizing pixel values to a 0–1 range, and encoding the labels into categorical format using one-hot encoding. The dataset was split into 80% for training and 20% for testing.

ii. Model Architecture

A custom-built Convolutional Neural Network (CNN) was used to recognize each character in the CAPTCHA image. The model accepts RGB images as input and generates four independent outputs, each corresponding to one character in the CAPTCHA. The architecture includes:

- Convolutional layers with ReLU activation
- MaxPooling layers for downsampling
- Dropout layers to prevent overfitting
- Flatten and Dense layers for final predictions

iii. Training and Optimization

The model was compiled with categorical cross-entropy loss and optimized using the Adam optimizer. It was trained over multiple epochs with early stopping and model checkpointing to prevent overfitting and retain the best weights.

iv. Evaluation Metrics

Two key metrics were used:

- Character-wise accuracy: Measures how many individual characters are predicted correctly.
- CAPTCHA accuracy: Measures how many full 4-character CAPTCHA strings are predicted correctly without any error.

v. Explainability with LIME

To enhance transparency and interpretability, LIME was used to explain the model's predictions. For each character prediction, the model was wrapped using Keras' functional API to isolate outputs. LIME then

generated superpixel-based visual explanations showing which parts of the CAPTCHA image influenced each character's prediction.

vi. Visualization and Analysis

The LIME explanations were visualized using matplotlib and skimage to highlight important regions in the image. This helped validate whether the model was learning relevant features and allowed debugging misclassifications.

vii. Tools and Technologies Used

- Python for scripting and development
- TensorFlow/Keras for deep learning model implementation
- LIME for explainable AI
- Matplotlib and PIL for visualization and image manipulation

Ethical Considerations

- Purpose of the Paper
  This paper is created for learning and research only. It is not meant to break security rules or help anyone cheat online systems.
- No Harmful Use
  CAPTCHAs are used on websites to stop fake users (bots). Our project shows how AI can solve CAPTCHAs, but we do not use it to access or harm any website. It's just to study how the model works.
- Explaining AI Decisions
  The project uses a tool called LIME to explain how the AI is making predictions. This helps people understand how the model thinks and builds trust in its results.
- Safe Data Use
  We created our own CAPTCHA images and did not use any real user data. This keeps the project safe and private.
- Security Awareness
  By showing that AI can solve CAPTCHAs, we highlight the need for stronger and smarter security methods in the future.

V. RESULTS AND DISCUSSION

Tools and Technologies Used

- Python for scripting and development
- TensorFlow/Keras for deep learning model implementation
- LIME for explainable AI
- Matplotlib and PIL for visualization and image manipulation

Metric Evaluation

Metric	Value
Character Accuracy	98%
CAPTCHA Accuracy	89.1%

Sample Output

Fig.1. CAPTCHA VERIFICATION INTERFACE

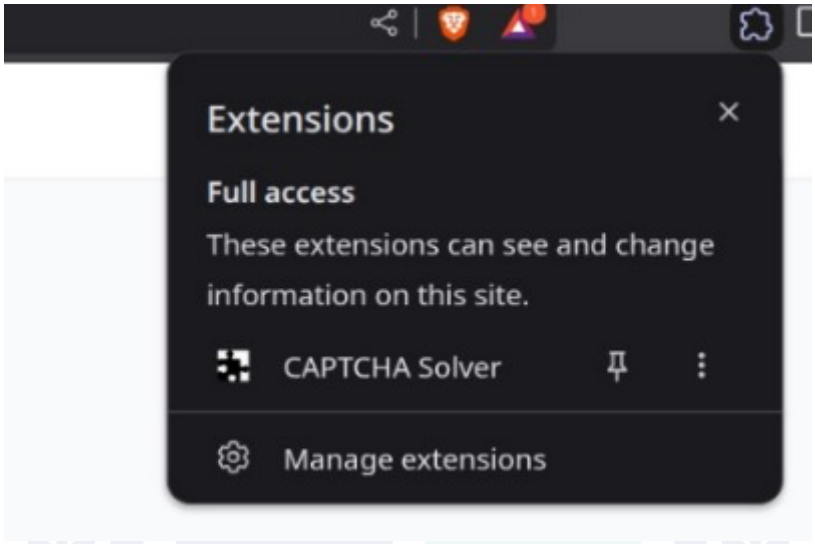

Fig.2.BROWSER EXTENSION PANEL

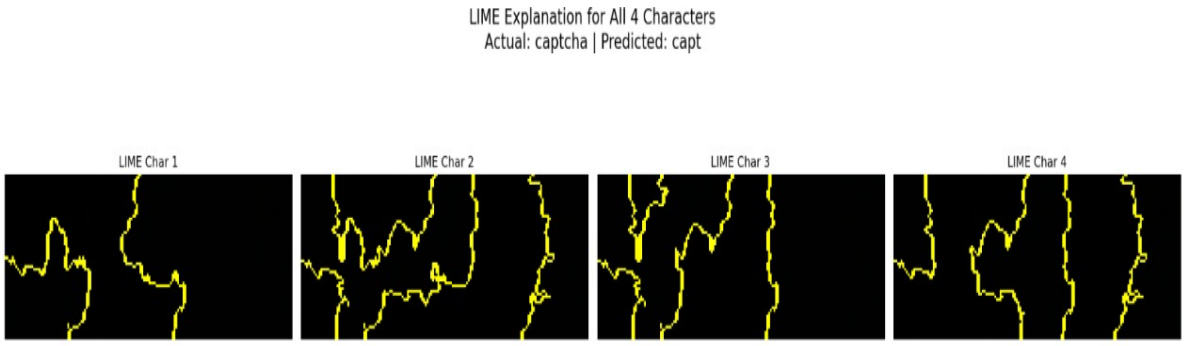

Fig.3.LIME EXPLANATION FOR 4 CHARACTERS

Interpretation

The model achieved high character-level and CAPTCHA-level accuracy, demonstrating effective learning from the dataset. LIME visualizations confirmed the model's focus on relevant image regions, validating prediction reliability. These results indicate that the system can accurately solve and explain CAPTCHAs, making it both powerful and interpretable for real-world use.

VI. CONCLUSION

Summary of Key findings

The key findings show that the model is capable of accurately predicting both character-level and entire CAPTCHA sequences, particularly alphanumeric text CAPTCHAs. The integration of LIME provided visual insights into the model's decision-making, helping to interpret and validate the predictions by highlighting the most influential parts of the image**.**

Implications for Theory and Practice

From a theoretical perspective, the use of LIME with deep learning models bridges the gap between black-box models and human interpretability, enhancing trust in AI predictions. Practically, this system can be deployed in cybersecurity tools, automated accessibility features for visually impaired users, or testing environments for CAPTCHA design robustness.

Future Scope

Looking forward, future scope includes training on a more diverse real-world dataset, extending the model to solve more complex image-based or distorted CAPTCHAs, and integrating additional XAI methods like SHAP for global interpretability. Furthermore, a user-friendly frontend can be developed to demonstrate this system in action, enhancing its practical usability across sectors like banking, e-commerce, and accessibility services.

VII. REFERENCES

[1] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You?: Explaining the Predictions of Any Classifier," *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD)*, pp. 1135–1144, 2016. doi: 10.1145/2939672.2939778.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[3] C. Szegedy et al., "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2014. [Online]. Available: https://arxiv.org/abs/1312.6199

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017. doi: 10.1145/3065386.

[5] F. M. Zennaro and S. Bergamaschi, "Automatic CAPTCHA Solver using Convolutional Neural Networks," in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, 2020, pp. 1–7. doi: 10.1109/IJCNN48605.2020.9207580.

[6] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems*, vol. 30, NeurIPS, 2017. [Online]. Available: https://github.com/slundberg/shap

[7] Google Brain Team, "TensorFlow: An end-to-end open-source machine learning platform," *TensorFlow.org*, 2023. [Online]. Available: https://www.tensorflow.org

[8] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial Examples for Malware Detection," in *European Symp. Research in Computer Security (ESORICS)*, 2017. [Online]. Available: https://arxiv.org/abs/1606.04435