iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** IX **Month of publication:** September 2023

DOI: https://doi.org/10.22214/ijraset.2023.55608

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IX Sep 2023- Available at www.ijraset.com

Improving CAPTCHA Recognition for Enhanced Web Scraping

Srujan Mahesh U N[1], Sujata Priyambada Mishra[2]
[1]*Student*, [2]*Assistant Professor, Electronics and communication*, R V College of Engineering, Bengaluru

Abstract: ***Web scraping capabilities are greatly improved by CAPTCHA recognition, which enables automated systems to get past security barriers and access valuable data. This journal presents a comprehensive study and analysis of techniques aimed at improving split CAPTCHA recognition for effective web scraping. The research investigates two approaches, including optical character recognition, and image processing techniques, to tackle the challenges posed by increasingly complex CAPTCHAs.***
Index Terms: OCR, CAPTCHA, pytessaract, Recurrent neural networks, Convolutional neural networks

I. INTRODUCTION

Web scraping has become an indispensable tool for extracting valuable data from the vast expanse of the internet. With the proliferation of online content, businesses, researchers, and developers rely on web scraping techniques to gather data for a variety of uses, such as market analysis, sentiment analysis, and data-driven decision-making. However, web scraping encounters significant challenges, one of which is the presence of CAPTCHAs, specifically designed to prevent automated access and protect the integrity of websites. For web scraping to be successful and overcome these obstacles, CAPTCHA recognition is an essential feature. CAPTCHAs, short for "Completely Automated Public Turing test to tell Computers and Humans Apart," are designed to differentiate between human users and automated bots. They typically involve visually distorted characters or images that humans can decipher, but pose difficulties for automated systems. Thus, CAPTCHA recognition techniques are essential to bypass these security measures and automate the data collection process. However, there are a number of issues with CAPTCHA recognition for web scraping that must be resolved. Firstly, CAPTCHAs are constantly evolving and becoming increasingly complex to defeat automated systems. This necessitates the development of robust and adaptable recognition algorithms capable of handling a wide range of CAPTCHA types. Secondly, since web scraping tasks frequently require processing a significant amount of data in real-time, the precision and effectiveness of CAPTCHA recognition algorithms are crucial. Moreover, CAPTCHAs should be recognized promptly to avoid delays in the web scraping process. Lastly, there is an ethical dimension to CAPTCHA recognition, as it raises concerns about the potential misuse of automated systems for malicious purposes. In light of these challenges, this journal presents a comprehensive study and analysis of techniques aimed at enhancing CAPTCHA recognition for web scraping. By exploring various approaches to optical character recognition and including image processing techniques, the research aims to address the complexities of CAPTCHA recognition and provide insights into improving the efficiency and accuracy of web scraping processes.

A. Optical Character Recognition

OCR (Optical Character Recognition) makes it possible to automatically extract and decipher textual information from images or scanned documents. OCR makes it possible to digitize printed or handwritten text, automate data entry processes, improve accessibility for blind people, and make text analysis easier in a variety of fields. For the most part, OCR methods involve pre-processing the image to make the text more visible, segmenting the text into individual characters or words, and using machine learning algorithms for pattern recognition techniques to recognize and extract the text and convert it into editable or searchable formats. To increase accuracy and handle complex situations, advanced OCR techniques may include deep learning models, language models, and post- processing steps.

B. Pytesseract

The popular Python library Pytesseract for optical character recognition (OCR) offers a simple interface to the Tesseract OCR engine. Google created Tesseract, an open-source OCR engine renowned for its reliability and accuracy. Developers can incorporate Tesseract's features into their Python applications with ease thanks to Pytesseract. It can handle multiple languages and supports text extraction from different image formats. The extensive customization options provided by Pytesseract include defining page segmentation modes, specifying language models, and modifying image preprocessing parameters. Pytesseract makes the process of extracting text from images easier.

II. EXISTING MODELS

The issue of CAPTCHA recognition has been approached using a variety of methodologies. A notable methodologies for CAPTCHA recognition are based on machine learning techniques. Using a sizable dataset of labeled CAPTCHA images and their corresponding solutions, a model is trained using this method. CNNs (Convolutional neural networks) and RNNs (recurrent neural networks) are two commonly used machine learning algorithms for CAPTCHA recognition. CNNs are excellent at extracting features and can successfully learn intricate patterns from CAPTCHA images. In contrast, RNNs are effective at spotting sequential dependencies in CAPTCHAs that involve character ordering or spatial relationships. To ensure robustness against different CAPTCHA designs, it is necessary to train the model on a diverse and representative dataset.

The enormous popularity of deep learning models, particularly deep neural networks, in image processing and computer vision is a result of their capacity to automatically learn features directly from raw pixel data. Convolutional neural networks (CNN), recurrent neural networks (RNN), and generational adversarial networks (GAN) are examples of deep learning models that have demonstrated remarkable success in a variety of tasks, including object detection, semantic segmentation, image generation, and image classification. The preprocessing of an image followed by feature extraction techniques is another widely used technique to extract the distinctive properties of the CAPTCHA elements, such as letters, numbers, or symbols. Frequently used features include those that are based on gradients, textures, or statistics. In order to classify the CAPTCHA image and predict the characters or symbols, these extracted features are then fed into classification algorithms like Support Vector Machines (SVM), Neural Networks (NN), or Random Forests (RF). Overall, there are pros and cons to each approach, even though machine learning and also for image processing-based methods have shown promise in CAPTCHA recognition. Further investigation is needed to explore hybrid methodologies that take the best features of both paradigms and address the changing problems posed by contemporary CAPTCHA designs. Innovation in this area will also be fueled by the creation of more complex CAPTCHA designs and the ongoing competition between security and recognition technologies.

III. METHODOLOGY

In most of the OCR techniques available preprocessing defines the output, accuracy and efficiency of the model. This model is no exception to that. Two pre-processing techniques are used for detection in Level 2 of preprocessing and a common preprocessing is done in level 1 of preprocessing.

A. Level I Preprocessing

First, the original CAPTCHA image is converted to grayscale in level 1 preprocessing. By removing the colour information, this conversion makes the image simpler by producing a single-channel image where each pixel denotes the brightness of the light.

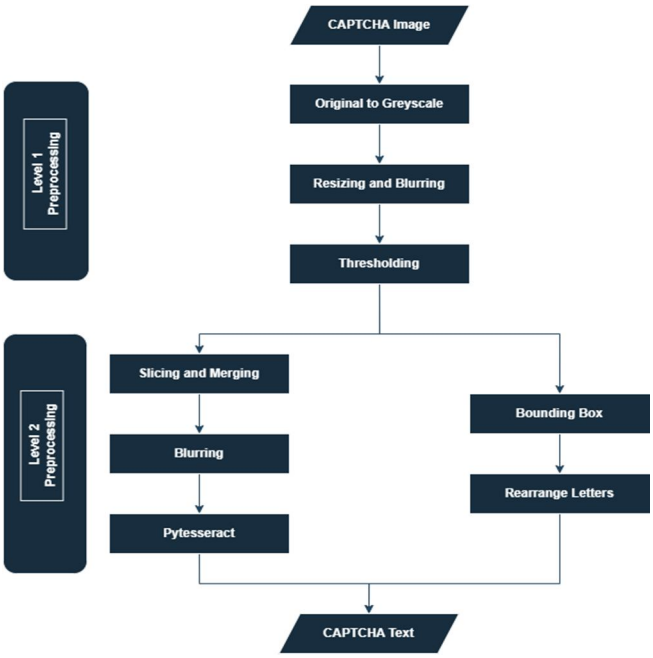


Fig. 1. Methodology Flow Chart

The resulting image is then shrunk and blurred. Resizing is done to adjust the image's size so that it can be processed and used for analysis. Blurring the image can improve the OCR engine's ability to accurately recognize the characters by reducing noise and smoothing the edges.

By applying a threshold to the image, thresholding trans- forms the image into a binary representation. Based on its intensity, this process categorizes each pixel as either foreground (text) or background. By selecting a suitable threshold value, the contrast between the text and background

B. Level II Preprocessing

Level 2 of preprocessing employs two different preprocessing techniques for detection, namely the bounding box technique and the slicing technique, to extract CAPTCHA text from CAPTCHA images.

1) *Slicing Technique:* The slicing method is used to im- prove character recognition even more. In order to do this, the CAPTCHA image must be divided into separate character segments and then combined. Additionally, blurring can be used to lessen background noise and enhance character clarity. The characters are then found and extracted from the preprocessed image slices using the Pytesseract library. Pytesseract recognises the characters and transforms them into machine- readable text using cutting-edge algorithms and trained models. By addressing issues with noise, irregularities, and character segmentation, these steps make it possible to more reliably extract the CAPTCHA characters for processing and analysis.
2) *Bounding Box Technique:* The bounding box technique to extract individual characters from the CAPTCHA image. The bounding box is located using contours as the first step in this procedure. Outlines that enclose continuous regions of the same pixel intensity are called contours. The contours in the thresholded image can be used to pinpoint each character's boundaries.

The characters are rearranged according to their x coordinates once the bounding boxes have been obtained. The characters are now arranged in the proper order as they do in the CAPTCHA thanks to this rearrangement. It is possible to accurately extract the characters for further character recognition and analysis by sorting the bounding boxes according to their x coordinates.

IV. IMPLEMENTATION

To implement the models for CAPTCHA recognition using Python Jupyter Notebook on Visual Studio Code was used. These IDEs provide a convenient environment for writing, running, and debugging Python code. Regarding the dynamic dataset, where the image is loaded every time from a URL, the first step would be to fetch the image from the URL using Python's requests library.

The requests python package is applied to directly download the images from the given URL because the dataset consists of images that are dynamically loaded from a URL. With the aid of these libraries, you can dynamically load the images during runtime by programmatically fetching them. After the images have been downloaded from the URL, Each picture has undergone our CAPTCHA recognition models. Python code is used for integration, depending on the particular models you are employing.

By fetching the data from the web scraping model multiple times and using each CAPTCHA recognition model, The comparison of accuracy and efficiency of different models is done. This comparison allows you to assess the performance of each model in terms of recognizing CAPTCHAs fetched dynamically from URLs.

Python provides a rich ecosystem of libraries and tools for implementing CAPTCHA recognition models. By utilizing the appropriate IDE, image retrieval from URLs, preprocessing techniques, and machine learning/deep learning models, you can build a robust system capable of recognizing CAPTCHAs dynamically loaded from URLs.

V. RESULTS AND DISCUSSION

As per the results, the Slicing method, which segments the CAPTCHA image into distinct characters or symbols, has a higher accuracy rate of about 90-93%. This means that compared to the Bounding Box method, which achieves an accuracy rate of 84-86%, it more accurately recognizes and predicts the characters or symbols in the CAPTCHA. The accuracy is determined by how often the data could be obtained from the web scraping model with the CAPTCHA being detected. contrast to the Slicing technique, the Bounding Box method processes data more quickly, on average with a 650 millisecond delay. This suggests that the Bounding Box method, which involves using bounding boxes to identify and extract the characters or symbols, is computationally more effective and can process CAPTCHA images faster.

In addition, even though the Slicing method provides greater accuracy, it is a marginally longer processing time. The Bounding Box method, on the other hand, provides faster processing at the expense of a small amount of accuracy. The decision between these approaches ultimately comes down to the particulars of the application, such as the preferred ratio of accuracy to speed.

VI. CONCLUSION

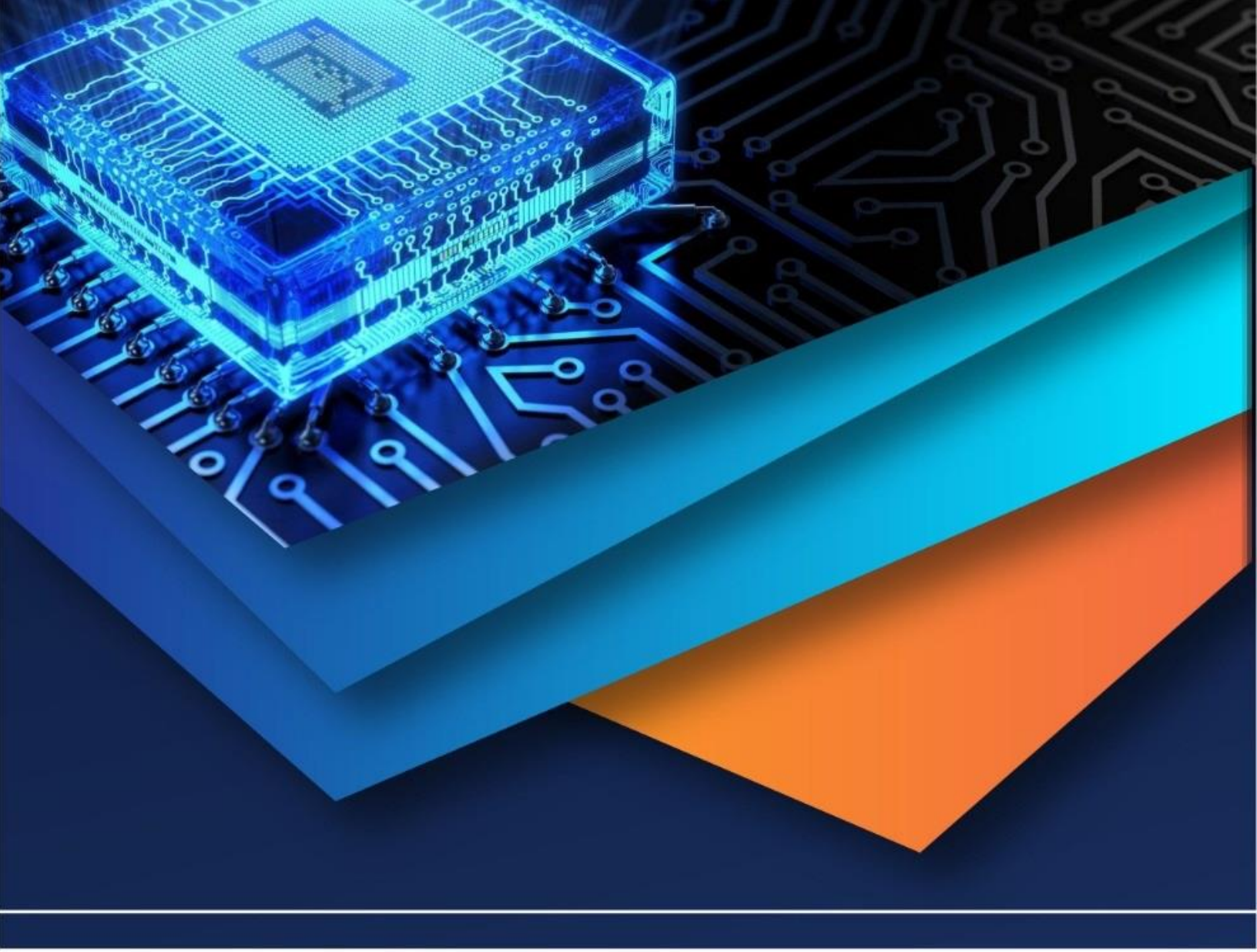
In conclusion, this journal paper presented and compared two methods for CAPTCHA recognition: the slicing method and the bounding box technique. This study demonstrates the merits of the slicing method for dynamically loaded CAPTCHA recognition over the bounding box method. The slicing approach showed superior accuracy and efficiency, making it a promising method for applications like automated systems, security protocols, and data processing pipelines that call for robust and quick CAPTCHA recognition. These findings offer both researchers and practitioners useful insights on the field of web scraping, offering a more efficient and accurate approach to overcome CAPTCHA challenges in data collection tasks. Future studies could concentrate on further refining the slicing technique and incorporating it with other cutting-edge methods to improve CAPTCHA recognition performance.

VII. FUTURE SCOPE

It has been found that the slicing method outperformed the bounding box method in terms of accuracy and processing speed offers intriguing opportunities for future research and development. Future areas of focus can be on investigating cutting-edge methods for enhancing Slicing method's ac- curacy, investigating alternate preprocessing strategies, and increasing the method's efficiency through parallel processing or distributed computing. The paper can advance CAPTCHA recognition methods by addressing these issues and offer insightful information to researchers and practitioners in the field.

REFERENCES

[1] Kumar, M., Jindal, M.K. & Kumar, M. A Systematic Survey on CAPTCHA Recognition: Types, Creation and Breaking Techniques. Arch Computat Methods Eng 29, 1107–1136 (2022). https://doi.org/10.1007/s11831-021- 09608-4
[2] Wang, Jing, Jiaohua Qin, Xuyu Xiang, Yun Tan, and Nan Pan. CAPTCHA recognition based on deep convolutional neural network. Math. Biosci. Eng 16, no. 5 (2019): 5851-5861.
[3] Ma, Wentao, Jiaohua Qin, Xuyu Xiang, Yun Tan, Yuanjing Luo, & Neal N. Xiong. Adaptive median filtering algorithm based on divide and conquer and its application in CAPTCHA recognition. Comput., Mater. Continua 58, no. 3 (2019): 665-677.
[4] Wang, Yao, Yuliang Wei, Yifan Zhang, Chuhao Jin, Guodong Xin, and Bailing Wang. Few-shot learning in realistic settings for text CAPTCHA recognition. Neural Computing and Applications 35, no. 15 (2023): 10751- 10764.
[5] Shu, Yujin, and Yongjin Xu. End-to-End Captcha Recognition Using Deep CNN-RNN Network. In 2019 IEEE 3rd Advanced Information Man- agement, Communicates, Electronic and Automation Control Conference (IMCEC), pp. 54-58. IEEE, 2019.
[6] Wang, Zhong, and Peibei Shi. CAPTCHA recognition method based on CNN with focal loss Complexity 2021 (2021): 1-10.
[7] Xiangfeng Lin, Linfu Li, and Yu Ren Deep learning captcha recognition for mobile based on TensorFlow, Proc. SPIE 12587, Third International Seminar on Artificial Intelligence, Networking, and Information Technol- ogy (AINIT 2022), 125871J (22 February 2023)
[8] A. M. Zhang, Application of optimized convolutional neural network in image recognition of complex verification code, Journal of University of information engineering 22.06 (2021)
[9] Z. F. Guo, and Y. Qiu.]textitResearch and implementation of character graphic verification code recognition algorithm based on SVM. computer programming skills and maintenance. 12 (2021): 163–165.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY