Exploring a Deep Learning-based Solution for hCAPTCHA Challenge Solver

Vinod Bisen
University of the Western Cape
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Omowunmi E. Isafiade
University of the Western Cape
47 PUBLICATIONS **204** CITATIONS
SEE PROFILE

Exploring a Deep Learning-based Solution for hCAPTCHA Challenge Solver

Vinod D. Bisen
Department of Computer Science
University of the Western Cape
CapeTown, South Africa
4284956@uwc.ac.za

Omowunmi E. Isafiade
Department of Computer Science
University of the Western Cape
CapeTown, South Africa
https://orcid.org/0000-0002-3028-6180

Abstract— **Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) is used to prevent websites from bot attacks. Image-based CAPTCHA services such as hCAPTCHA image challenges are being solved by using deep learning approaches such as Residual Network (ResNet) and You Only Look Once (YOLO). Despite the progress that has been made so far, there is a need for more research in this domain. This paper utilized YOLO v3, which is a deep learning-based object detection algorithm that falls under the family of Convolutional Neural Networks (CNNs). The architecture of YOLO v3 consists of 53 convolutional layers and utilizes a variant of Darknet-53 as its backbone, with the capability to successfully solve image-based CAPTCHA challenges. This work systematically gathered standard hCAPTCHA image challenges from the hCAPTCHA server, encompassing ten distinct categories. The solution resulted in the successful resolution of 5,867 out of 6,000 challenges. The system's performance was evaluated using a confusion matrix and by testing it against more than 300 image-based CAPTCHA challenges per day, through a web page created for login purposes integrated with hCAPTCHA, which achieved 98% accuracy. The system runs on a local machine with only CPU:2 Intel core i5-4300U CPU(1.90GHz) processors, 16 GB of RAM with no GPU devices, and is also time efficient, taking only an average of 3.5 seconds to crack a challenge. To the best of the researchers' knowledge, no research has utilized the automated hCAPTCHA challenge solver approach implemented in this study using YOLO.**

Keywords— hCAPTCHA, Darknet-53 YOLO, Neural Network, Artificial Intelligence, Automation Testing

I. INTRODUCTION

The CAPTCHA is a crucial tool in distinguishing between genuine human users and automated bots. Traditional CAPTCHAs, such as text-based or image-based challenges, have been effective in preventing unauthorized access but have become vulnerable to sophisticated machine learning attacks [1]. Among these, hCAPTCHA stands out for its enhanced user-friendliness and privacy features. However, with the rise of machine learning algorithms capable of decoding traditional CAPTCHAs, there is a pressing need for more advanced and accurate solutions [2,3]. This paper explores the use of deep learning, specifically the YOLO v3 algorithm, to effectively solve hCAPTCHA challenges [4,5].

The effectiveness of CAPTCHAs in combating bots is undeniable. However, their implementation can inadvertently lead to adverse effects on user experience. Several drawbacks associated with CAPTCHAs include:

a) Disruption and Frustration: CAPTCHAs have the potential to disrupt user activities and induce frustration due to their intrusive nature [6].

b) Accessibility Issues: Certain types of CAPTCHAs pose barriers for users with visual impairments or those reliant on assistive devices, thereby excluding a segment of the population from accessing online services [6][7].

c) Browser Compatibility: Compatibility issues may arise with certain CAPTCHA types, resulting in inconsistencies across different web browsers and potentially impeding user interactions [6][7].

d) Usability Concerns: Some users may encounter difficulties in understanding or completing CAPTCHAs, leading to usability concerns, and hindering the overall user experience [8].

These factors highlight the importance of carefully considering the implications of CAPTCHA implementation to ensure that security measures do not compromise user accessibility and satisfaction.

hCAPTCHA, created by researchers at the University of Maryland [5], aims to enhance user-friendliness and privacy compared to traditional CAPTCHAs. Figures 1(A) and 1(B) depict an instance of hCAPTCHA widgets integrated within a webpage. Upon users selecting the "I am human" checkbox, the challenge presented in Figure 1(B) is displayed on the screen for users to resolve [9].

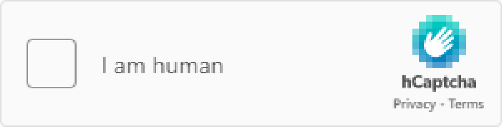
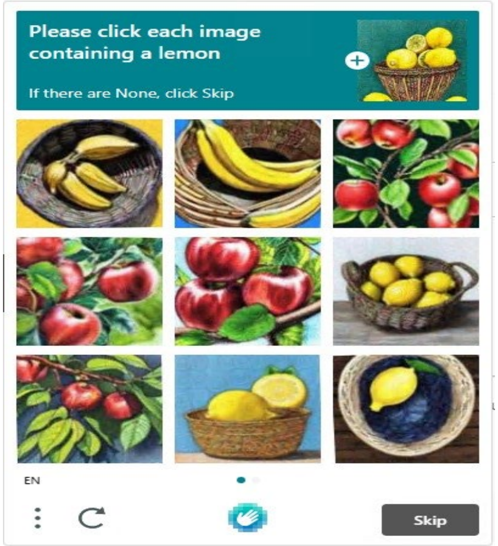


FIGURE 1(A): HCAPTCHA "I AM HUMAN" WIDGET [9].



Please click each image containing a lemon

If there are None, click Skip

FIGURE 1(B): HCAPTCHA IMAGE CHALLENGE WIDGET [9]

Image-based CAPTCHA-solving techniques involve various stages in automating the process of solving CAPTCHAs. This research implemented its proposed solution using the high-level architecture summarised in Figure 2, which is further discussed as follows:

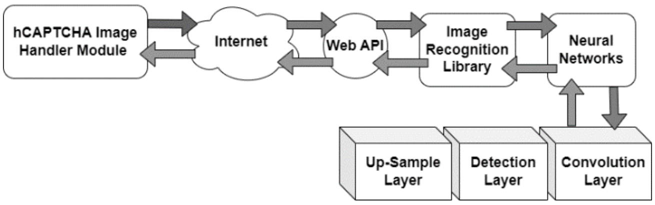


FIGURE 2: SCHEMATIC REPRESENTATION OF THE PROPOSED AUTOMATED HCAPTCHA SYSTEM.

- hCAPTCHA Application Programming Interface (API) Image Handler Module: This module is for login and operates from the testing environment, accessing the API to detect and process hCAPTCHA images. This module will also transmit hCAPTCHA images to the web APIs.
- Internet Connectivity: The test system must be connected to the internet to access the web APIs.
- Web APIs: The Web APIs hosted on a local server will activate the deep learning model to identify the images presented.
- Image Recognition Library: This library will establish and manage the neural network model necessary for image recognition. It includes algorithms and business logic to initialize the neural network and gather responses.
- Neural Network: This is the deep learning model designed to support the YOLO algorithm in recognizing an image. The neural network is trained with various sets of images to generate the required pattern.
- Convolutional Layer: This layer applies convolution operations to the input image to extract features. It uses filters to scan the image, creating feature maps that highlight various aspects like edges, textures, and patterns.
- Detection Layer: This layer is responsible for predicting bounding boxes, object classes, and confidence scores.
- Up-Sample Layer: This layer increase the spatial resolution of the feature maps, enabling finer-grained detection.

YOLO uses an approach that applies a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. The bounding boxes are weighted by the predicted probabilities. [10] [11]

While several authors have addressed the challenge of solving hCAPTCHA [4, 5, 9, 12], this research aims to identify areas for potential enhancement, such as accuracy and training time through the implementation of YOLOv3 on 6,000 image challenges.

The remainder of this paper is structured as follows: Section 2 presents the related research and further justifies the contribution of this work. Section 3 provides details on the research methodology. Section 4 presents the results and detailed analysis. Finally, Section 5 concludes the paper and provides future recommendations.

II. LITERATURE REVIEW

CAPTCHAs have evolved significantly over the years, with solutions ranging from text-based and audio-based to image-based challenges [4, 6, 11, 17]. Text-based CAPTCHAs, although widely used, often pose accessibility issues and can be circumvented by advanced optical character recognition (OCR) techniques [9,12]. Audio CAPTCHAs, designed for visually impaired users, require the accurate transcription of distorted spoken words [40, 42], but they are not foolproof against sophisticated speech recognition systems.

The emergence of image-based CAPTCHAs, such as hCAPTCHA, introduced more complex challenges involving object recognition tasks. These CAPTCHAs present users with images that require identification of specific objects, leveraging human visual perception to thwart automated attacks [5]. However, the advancement in deep learning, particularly in computer vision, has led to the development of models capable of solving these challenges with high accuracy.

Among these models, YOLO (You Only Look Once) has shown remarkable performance in real-time object detection tasks [28]. YOLO v3, in particular, combines speed and accuracy, making it a promising candidate for solving hCAPTCHA challenges. This study leverages YOLO v3 to enhance the accuracy and efficiency of automated hCAPTCHA solvers.

A. Categories of CAPTCHA

CAPTCHAs are tests designed to determine if a user is human. They are categorized based on the type of distortion, such as letters, numbers, or pictures. Modern CAPTCHAs fall into five primary categories, which are:

1) Text-based CAPTCHAs: These are challenges that require users to interpret and enter text characters correctly to prove they are human. These typically involve distorted or obscured letters or words that users need to identify and input into a form. While effective, they can be difficult to interpret for both bots and humans as shown in Figure 3 [9,12,29]. Methodologies to solve text-based CAPTCHAs typically involve OCR techniques, which use algorithms to analyze and interpret the visual patterns of the distorted text. These algorithms may include preprocessing steps to enhance image quality, segmentation techniques to separate individual characters, feature extraction methods to identify distinguishing characteristics of each character, and pattern recognition algorithms to match the characters against a database of known characters. Additionally, machine learning approaches, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), may be trained on labeled datasets of CAPTCHA images to learn to recognize and transcribe the characters more accurately. Adversarial training methods may also be used to improve the robustness of OCR systems against attempts to evade detection.[29]

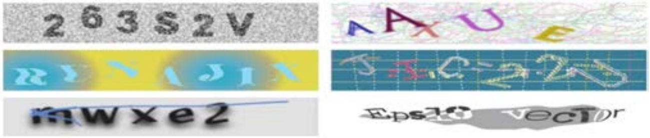


FIGURE 3: SAMPLE CATEGORIES OF TEXT-BASED CAPTCHA CHALLENGE WIDGETS [9,12,29]

2) Picture-based CAPTCHAs: These challenges require users to identify and interact with images to prove they are human [29,35]. These may involve tasks like selecting images that contain specific objects to match or identify similar images. They are easier for people but tough for bots because they need to understand and classify the images as shown in Figure 4.

Methodologies to solve picture-based CAPTCHAs typically involve computer vision techniques, which use algorithms to analyze and interpret the visual content of the images. These algorithms may include:

- Feature Extraction: Extracting visual features from the images, such as shapes, colors, textures, and edges [29].
- Object Detection: Detecting and locating objects within the images using techniques like Haar cascades, Histogram of Oriented Gradients (HOG), or deep learning-based object detection models like Faster R-CNN or YOLO (You Only Look Once) [40].
- Image Classification: Classifying images into predefined categories based on the presence or absence of specific objects or features. This may involve training machine learning models, such as convolutional neural networks (CNNs), on labeled datasets of images [40].
- Semantic Segmentation: Segmenting images into different regions that corresponds to different objects or background elements. This helps in identifying and isolating the relevant objects within the images [41].
- Pattern Recognition: This involves recognizing patterns or shapes within the images to identify specific objects or arrangements [42].

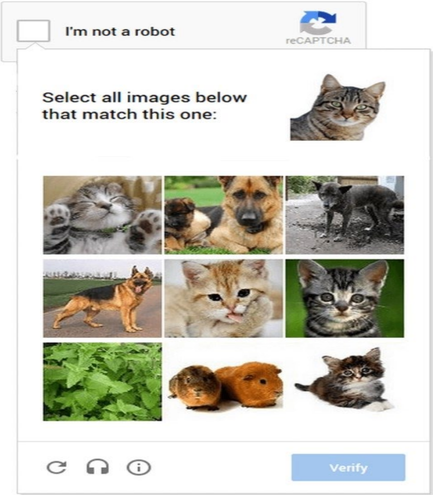


FIGURE 4: SAMPLE CATEGORIES OF IMAGE-BASED CAPTCHA CHALLENGE [29].

3) Sound CAPTCHA: These are the challenges that involve listening to and deciphering audio recordings to prove that the user is human. Typically, these audio recordings contain distorted or obscured spoken words or sequences of numbers that the user must accurately transcribe, often designed for the visually impaired. Users listen to and type a series of letters or numbers. Sound CAPTCHA can be tricky for both humans and bots [29,43].

The approaches to solving sound CAPTCHAs typically involve using machine learning algorithms, such as speech recognition systems, to analyze and interpret the audio content. These algorithms may employ techniques like signal processing, feature extraction, and pattern recognition to identify and transcribe spoken words or numbers. Additionally, deep learning approaches, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), may be employed to improve the accuracy of audio transcription [40,43].

4) Math or Word Problems: Math or word problems CAPTCHAs are challenges that require users to solve mathematical equations or answer questions involving language comprehension to prove they are human. These challenges typically involve simple arithmetic operations, logical reasoning, or language understanding tasks. Bots struggle because they have a hard time understanding and answering these types of questions [9, 12, 44]. The approach to solving math or word problems CAPTCHAs typically involves natural language processing (NLP) techniques for word problems and mathematical computation algorithms for math problems [44].

5) No CAPTCHA: This refers to a system that aims to distinguish between human users and automated bots without requiring any explicit action or challenge from the user. Instead of presenting users with traditional CAPTCHA challenges, such as distorted text or image recognition tasks, "No CAPTCHA" solutions rely on passive methods to detect human-like behavior and distinguish it from automated activity [9,12]. Methodologies to implement "No CAPTCHA" solutions typically involve behavioral analysis, machine learning, and heuristic algorithms to identify patterns indicative of human interaction such as mouse movements, and keyboard input.

In summary, CAPTCHAs are important for stopping bots, even though they can be difficult for users. Different types of CAPTCHAs balance security and user-friendliness.

B. Existing Research on CAPTCHA

Several researchers have attempted to propose solutions around CAPTCHA systems. For example, the Asirra CAPTCHA [13] relied on distinguishing between images of dogs and cats in 2007, but in 2008, Golle et al. [14] developed a machine learning classifier that could solve Asirra challenges with a 10.3% success rate.

Another example is the Human interactive proof (HIP) algorithm called ARTiFACIAL, The ARTiFACIAL CAPTCHA scheme, proposed by Rui et al. [15], requires users to identify faces and facial features in distorted images. However, Zhu et al. [16] demonstrated successful attacks against several image CAPTCHA schemes, including ARTiFACIAL, and provided guidelines for designing more robust schemes based on their findings.

Sivakorn et al. [17] attacked an earlier version of image reCAPTCHA v2 by leveraging online image annotation services, shedding light on the workings of reCAPTCHA's risk analysis engine.

In 2019, Weng et al. [18] demonstrated deep learning-based attacks against real-world image CAPTCHA services, highlighting their vulnerability to automation. Recently, Hossen et al. [19] proposed a low-cost attack mechanism against the hCAPTCHA System. The authors introduce an automated system capable of effectively bypassing hCAPTCHA challenges. The results demonstrate that their system achieves an accuracy rate of 95.93% in successfully solving these challenges. Moreover, the average time spent in cracking a single challenge is 18.76 seconds. It is worth noting that their attack was executed from a docker instance equipped with only 2GB of RAM, 3 CPUs, and no GPU devices. However, Hossen et al. [20] proposed a solution that demonstrated that automated programs equipped with Deep Neural Network (DNN) image classifiers and off-the-shelf image recognition services' vision APIs could solve reCAPTCHA v2's image challenges. The authors achieved an online success rate of 83.25%. This success rate represents the highest reported to date, and on average, it takes about 19.93 seconds (including network delays) to crack a single challenge.

Considering the aforementioned research, it is evident that several aspects need further consideration and investigation such as dependency on an existing online dataset for testing, Furthermore, ensuring accuracy is a significant factor driving research in this area due to the sensitivity of this domain of interest.

This paper distinguishes itself by proposing an automated deep learning system that can effectively solve hCAPTCHA challenges with a high level of accuracy using YOLO v3. Unlike previous studies that relied on tools like Asirra, ARTiFACIAL, Cortcha, reCAPTCHA, and hCAPTCHA with various methods such as Partial Credit Algorithm, HIP algorithm, Image classifier, R-CNN/Faster-RCNN, and ResNet-18, this work introduced the YOLO family version 3 for hCAPTCHA solver. While other authors depended on existing datasets like Pix GRIS and ImageNet, this research built its custom dataset and trained the proposed system. This means we did not rely on online datasets, which helps ensure better accuracy. The proposed system can run on a regular computer without a GPU and cracks a challenge on time compared to previous research.

TABLE 1: COMPARISON OF RELATED RESEARCH ON CAPTCHAS.

Ref	*Year*	*Research Focus*	*CAPTCHA*	*Model*	*Data Set*
[13]	2007	A CAPTCHA that Exploits Interest-Aligned Manual Image Categorization	Asirra	Partial Credit Algorithm and Token Buckets	Asirra database
[14]	2008	Machine Learning Attacks Against the Asirra CAPTCHA	Asirra	Partial Credit Algorithm and Token Buckets	Asirra database
[15]	2004	ARTiFACIAL: Automated Reverse Turing test using FACIAL features	ARTiFACIAL	Face detection Model	Pix dataset
[16]	2010	Attacks and Design of Image Recognition CAPTCHAs	Cortcha	Face detection Model	Pix dataset
[17]	2016	Deep Learning for Semantic Image CAPTCHAs Solution	reCAPTCHA	Image classifier	GRIS,Alchemy,Clarifai,TDL
[18]	2019	Towards Understanding the Security of Modern Image Captchas and Underground Captcha-Solving Services	Selection-based captcha Slide-based captcha Click-based captcha	R-CNN Faster-RCNN	ImageNet
[19]	2021	A Low-Cost Attack against the hCaptcha System	hCAPTCHA	ResNet-18	ImageNet
This work	2024	Exploring A Deep Learning-based hCAPTCHA Challenge Solver	hCAPTCHA	YOLO v3 DarkNet-53, CNN	Self-curated dataset

III. METHODOLOGY

This study employs a deep learning-based approach to solve hCAPTCHA challenges, utilizing the YOLO v3 algorithm for real-time object detection. The methodology consists of the following steps:

- *Data Collection and Preprocessing:* An automated script was developed to collect images from Google Images, categorized into 10 specific classes commonly used in hCAPTCHA challenges. Each image was resized to 416x416 pixels to ensure uniformity.
- *Model Training:* The collected dataset, consisting of approximately 13,000 images, was divided into training, validation, and test sets in a 70:20:10 ratio. The YOLO v3 model was trained on this dataset using a GPU-based system to optimize learning efficiency and accuracy.
- *Challenge Solving:* The trained model was integrated into a web API designed to automate the hCAPTCHA-solving process. This API captures hCAPTCHA images, processes them using the YOLO v3 model, and identifies the objects to solve the challenge.
- *Evaluation:* The system's performance was evaluated based on accuracy and response time, using a confusion matrix and precision-recall metrics to analyze the results. The average time spent to solve a single hCAPTCHA challenge was also recorded to assess the model's efficiency.

This structured approach ensures a comprehensive evaluation of the YOLO v3 model's capability in solving hCAPTCHA challenges, highlighting its potential for real-world applications.

A. Object Detection Metrics and Non-Maximum Suppression

Object detection metrics are measures used to evaluate how well an object detection model performs. These evaluation metrics help assess the detection accuracy and localization precision of objects in images when using YOLOv3 [21].

- Mean Average Precision (mAP): It measures how accurately the model detects objects across different categories. This metric is commonly used in object detection tasks to evaluate the overall performance of the model [23].

- Intersection over Union (IoU): It measures the overlap between the predicted bounding boxes and the ground truth bounding boxes. IoU is crucial for determining the accuracy of object localization [21].
- Non-maximum Suppression (NMS) is a post-processing mechanism implemented in object detection to improve the results by cutting down on the number of overlapping boxes and making the detections more accurate. Usually, object detection systems give multiple boxes for the same thing, each with its confidence score. NMS gets rid of extra boxes that cover the same thing, keeping only the most accurate ones [25].

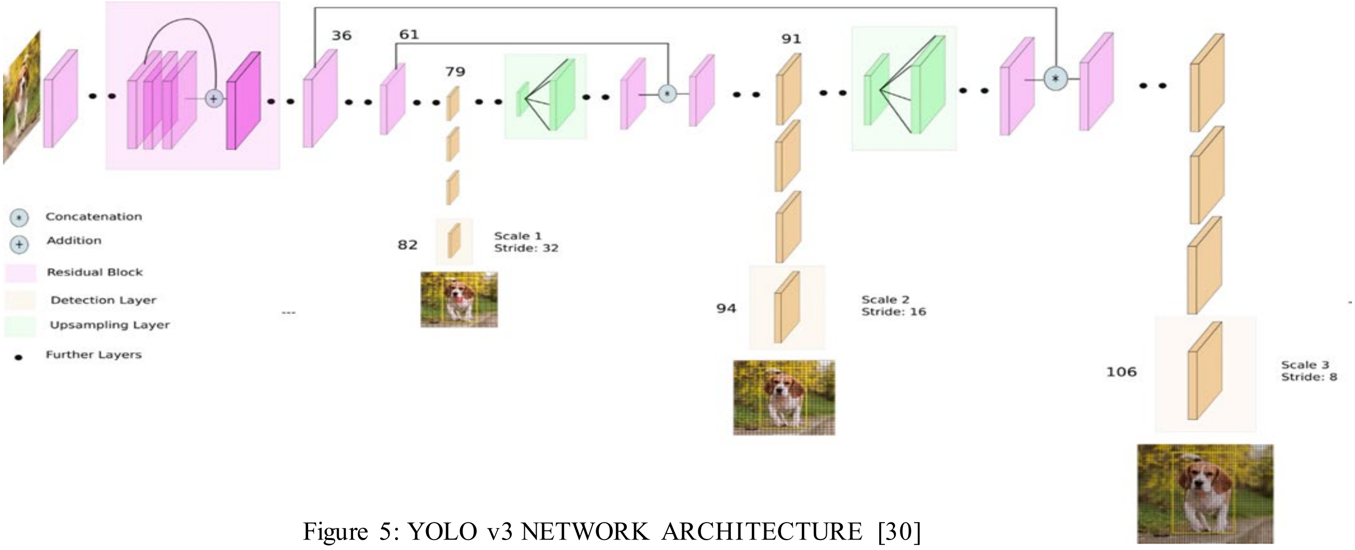


Figure 5: YOLO v3 NETWORK ARCHITECTURE [30]

B. Bounding box prediction

In YOLOv3, each bounding box gets a score based on how likely it is to contain an object. This score is figured out using logistic regression. The box that best matches the real object gets a score of 1, while the others get a score of 0. If no box is matched with an object, it only affects the classification loss, not the loss related to figuring out where the object is or how confident the model is in its prediction [24].

C. Class Prediction

Instead of using SoftMax for classification, YOLO uses binary cross-entropy [24,28] to train separate logistic classifiers for each label. The task is treated as a multi-label classification. It uses scale prediction to guess boxes at different scales. This helps the model to detect accuracy with small objects [24,28].

D. You Only Look Once (YOLO) v3 Architecture.

YOLO v3 is an object detection algorithm used in computer vision. Figure 5 presents the architecture of YOLO v3. It detects objects within an image using a deep neural network architecture comprised of several layers, including residual blocks, detection layers, up-sampling layers, and further layers. The following presents a simplified explanation of how each of the components highlighted in Figure 5 contributes to object detection:

- Residual Blocks: These are the basic building blocks of the neural network. Residual blocks help in learning features from the input image by passing it through multiple layers. They enable the network to capture complex patterns and details within the image.
- Detection Layer: The detection layer takes feature maps generated by previous layers and predicts bounding boxes, confidence scores, and class probabilities for potential objects within the image. This layer is responsible for identifying and localizing objects.
- Up-sampling Layers: These layers are used to increase the spatial resolution of feature maps. They help in preserving fine details and improving the accuracy of object localization. These layers are particularly useful in detecting small objects within the image.

- Further Layers: These additional layers in the network help in refining the predictions made by the detection layer. They perform tasks like adjusting bounding box coordinates, refining class probabilities, and filtering out false positives. These layers contribute to the overall precision of object detection.

Overall, by combining these different types of layers and leveraging the depth of the network (106 layers in the case of YOLOv3), the model can effectively detect and localize objects of various sizes and classes within the input image [25,26].

E. Implementation Details

In this paper, we propose a web API that can solve the hCAPTCHA challenge. This API internally implements artificial neural network-based learning for the identification of objects from a given image. Internally this network will be trained for multiple commonly referred objects in an hCAPTCHA.

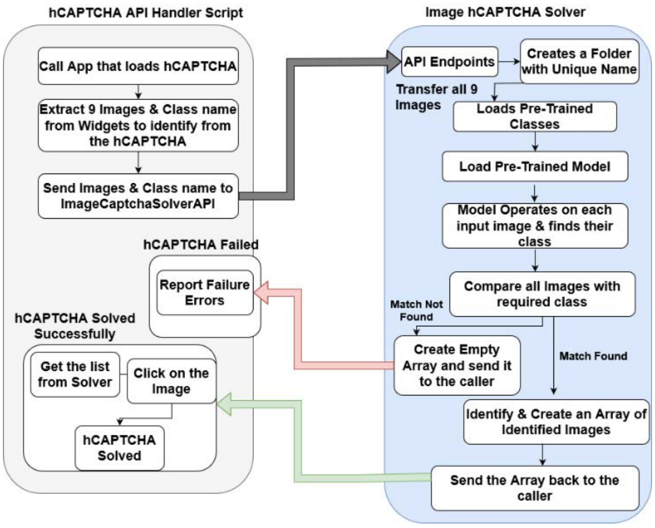

hCAPTCHA API Handler Script
Call App that loads hCAPTCHA
Extract 9 Images & Class name from Widgets to identify from the hCAPTCHA
Send Images & Class name to ImageCaptchaSolverAPI
hCAPTCHA Failed
Report Failure Errors
hCAPTCHA Solved Successfully
Get the list from Solver
Click on the Image
hCAPTCHA Solved
Image hCAPTCHA Solver
API Endpoints
Creates a Folder with Unique Name
Transfer all 9 Images
Loads Pre-Trained Classes
Load Pre-Trained Model
Model Operates on each input image & finds their class
Compare all Images with required class
Match Not Found
Create Empty Array and send it to the caller
Match Found
Identify & Create an Array of Identified Images
Send the Array back to the caller

FIGURE 6: THE WORKFLOW OF THE HCAPTCHA SOLUTION ARCHITECTURE

Figure 6 presents the solution architecture of the proposed automated hCAPTCHA solver, which comprises three fundamental steps:

Step 1 - Capturing the Challenge: To tackle the challenge, first a user will use the application to open hCAPTCHA. This is followed by selecting 9 images and their corresponding class names from the widgets on hCAPTCHA. These are sent to the ImageCaptchaSolver API.

Step 2 - Resolving the Challenge: When the ImageCaptchaSolver gets the images and class names, it creates a folder with a special name and transfers all 9 images into it. Then it loads pre-trained classes and a model. This model works on each image to find its class. Once the class is found, it compares all the images to the required class.

Step 3: Submitting and Verifying the Solution: After the second step, the system decides if there is a match or not. There are two potential outcomes after the third step, which are:

Outcome 1: No Match Found

If the system fails to identify a match, it returns an empty array to the initiator, signaling a "No match found" error to the user interface. This indicates an unsuccessful hCAPTCHA attempt.

Outcome 2: Match Found

In cases where a match is detected, the system assembles an array containing the recognized images. This array is then returned to the caller. This array is utilised as a guide, directing browser automation to click on images corresponding to the identified class. This process effectively resolves the hCAPTCHA challenge.

F. *Data Collection and Preprocessing*

To facilitate the data collection process for testing purposes, this work developed an automated login module using Python. The module is designed to log in to Google images, retrieving images from specified categories, and resizing them to dimensions of 416 x 416. Subsequently, the images are organised into a structured folder system optimized for data learning purposes.

After data collection and preparation are completed, this work employed a GPU-based learning system to conduct the learning process. This system generates a file named "best.pt," representing the optimal learning point achieved during the process. We utilize this file within our CAPTCHA solver framework to recognize input images from the hCAPTCHA challenger.

In total, we collected approximately 1300 images for each of the 10 categories namely: bed, bonsai tree, car, cat, crumpled paper ball, cup of orange juice, motorcycle, tree, turtle, and zebra. These images were labelled using an open-source tool. The image categories were then reorganized into three main folders, namely Training, Validation, and Testing, allocating 70%, 20%, and 10% of the images to each respective folder. In essence, our learning process leveraged a dataset comprising 13,000 images. The learning process is outlined in Table 2, presenting the varying number of epochs to obtain the best results. From Table 2, the best results were obtained at epoch 310, after which there was no further improvements observed.

TABLE 2: TRAINING RESULTS ON DATASETS WITH VARYING EPOCHS.

Epoch Cycles	*Images per Category*	*Total Images*	*Training Duration*	*Observations*
25	1000	10000	0.19 Hours	--
50	1000	10000	0.38 Hours	--
75	1000	10000	0.60 Hours	--
100	1000	10000	0.75 Hours	--
125	1000	10000	0.94 Hours	--
150	1000	10000	1.13 Hours	--
175	1000	10000	1.33 Hours	--
200	1000	10000	1.51 Hours	--
300	1000	10000	2.26 Hours	--
310	1000	10000	2.37 Hours	Best Results Obtained
479	1000	10000	6.41 Hours	No Improvements observed

Processing 10,000 images required approximately 2.37 hours to complete 310 epochs using a GPU-based system equipped with a processor i9 13900KF 128GB DDR5 RTX 3070 Ti 8GB GDDR6X 1TB Ultra-Fast NVME SSD configuration.

G. *Evaluation Metrics*

The evaluation metrics help assess the detection accuracy and localization precision of objects in images during the experiment. The following metrics were used in this study:

- Confusion Matrix: This measures the number of true positives, true negatives, false positives, and false negatives generated by the model. It helps to understand the performance of the model in terms of correct and incorrect classifications [31,33].
- F1-Confidence Curve: The F1-Confidence Curve combines the F1 score with the confidence scores. The F1 score is a metric that balances precision and recall. This curve helps assess the model's performance while

considering both the confidence of the predictions and their accuracy. A higher F1 score indicates better overall performance in the detection [31, 46].

- Precision-Confidence Curve: This curve illustrates the relationship between the precision of the model's predictions and the confidence scores assigned to those predictions. It helps understand how the model's precision varies with different confidence thresholds. [31, 46]
- Precision-Recall Curve: The Precision-Recall Curve shows the trade-off between precision and recall for different confidence thresholds. It helps evaluate the model's ability to make accurate detections while considering its completeness in capturing all relevant objects. [31, 46]

These evaluation metrics provide insights into the performance of the YOLO v3 model implemented, to understand its strengths and weaknesses in object detection and classification tasks. [31,33,46]

IV. RESULTS AND DISCUSSIONS

The evaluation of the implemented YOLO v3-based hCAPTCHA solver demonstrated a high accuracy rate of 98%, successfully solving 5867 out of 6000 challenges. The average time taken to solve each challenge was 3.5 seconds, indicating the system's efficiency.

The confusion matrix revealed that the model achieved high confidence levels across most categories, with minor confusion observed in similar object classes. The precision-recall curve further confirmed the model's robustness, maintaining a high precision level even at lower recall thresholds.

These results validate the effectiveness of YOLO v3 in solving image-based CAPTCHA challenges, highlighting its potential for enhancing security measures against automated bot attacks. The successful implementation of the solution on a CPU-based system without GPU acceleration demonstrates the feasibility of deploying this solution in resource-constrained environments.

The confusion matrix in Figure 7 elucidates the confidence levels associated with the classification results. In Figure 7, the vertical line (y-axis) shows the category predicted by the model, while the horizontal line (x-axis) shows the actual (true) category. For instance, upon inspecting the matrix, we found that the recognition confidence for the first category, "bed," stands at 97%. Conversely, for the second category, "bonsaitree," the confidence level is 86%, with a 3% confusion rate with the "tree" category. Similarly, the "crumpledpaperball" category exhibits confusion primarily with "turtle" and "tree," with a mere 0.01% confusion rate.

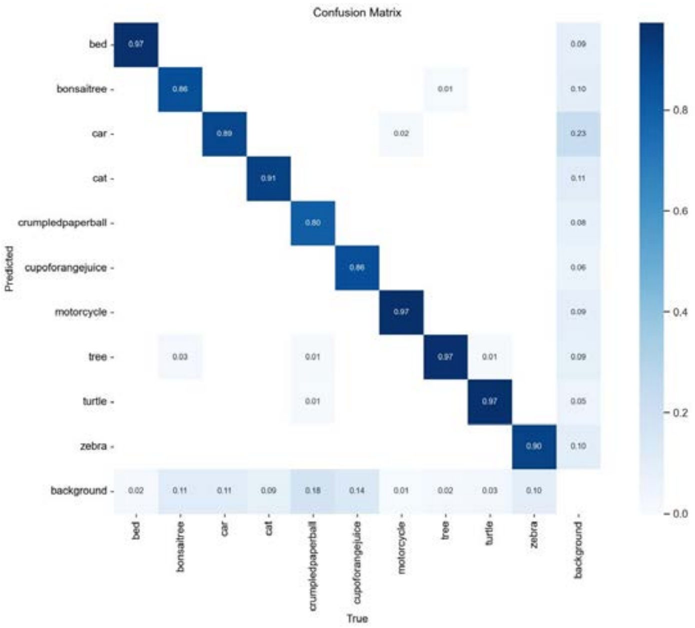


FIGURE 7: CONFUSION MATRIX WITH PREDICTED AND TRUE VALUES.

Figure 8 presents the F1 confidence curve for the implemented learning using YOLOv3 and represents the relationship between the F1 score and the confidence threshold used for the detection. The F1 confidence curve helps in selecting an optimal confidence threshold that balances precision and recall. By analyzing the F1 confidence curve, one can determine the confidence threshold that maximizes the F1 score, thereby achieving the best trade-off between precision and recall. Figure 8 illustrates the confidence levels of the individual categories relative to the total confidence across all categories, ranging from 0.88 to 0.473.

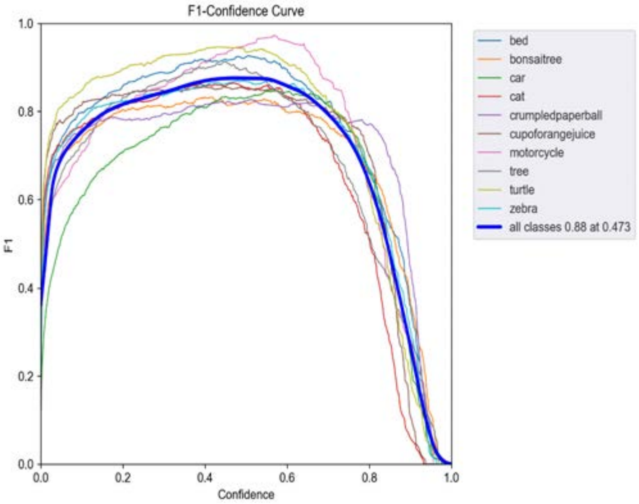


FIGURE 8: F1 CONFIDENCE CURVE.

The Precision-Confidence Curve obtained, as shown in Figure 9, provides insights into the trade-off between precision and confidence levels in the detection. A higher precision indicates fewer false positives, meaning that the model is more accurate in detecting objects. By examining this curve, one can determine the optimal confidence threshold that maximizes precision while maintaining an acceptable level of recall. Figure 9 concurrently maps the values of precision against confidence, with precision on the y-axis and confidence on the x-axis. All classes initially start at a precision of 1.00 and gradually decrease to 0.941.

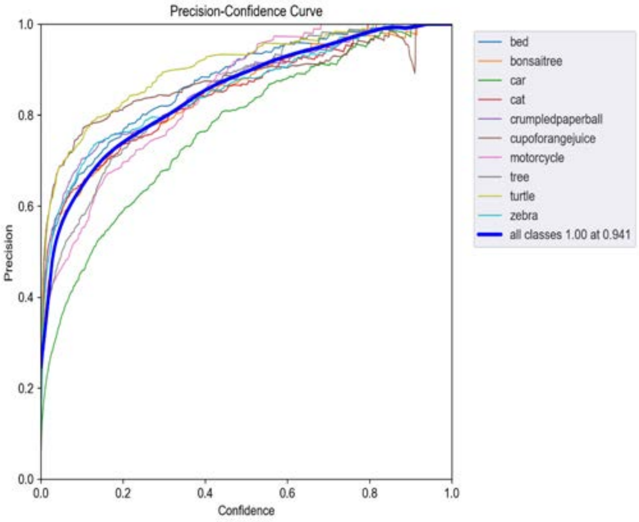


FIGURE 9: PRECISION-CONFIDENCE CURVE.

The Precision-Recall Curve obtained, as shown in Figure 10, helps in evaluating and fine-tuning the performance of the model. In Figure 10, precision is represented on the y-axis while recall is depicted on the x-axis, starting from a value of 0.915 for mean Average Precision (mAP) at a threshold of 0.5.

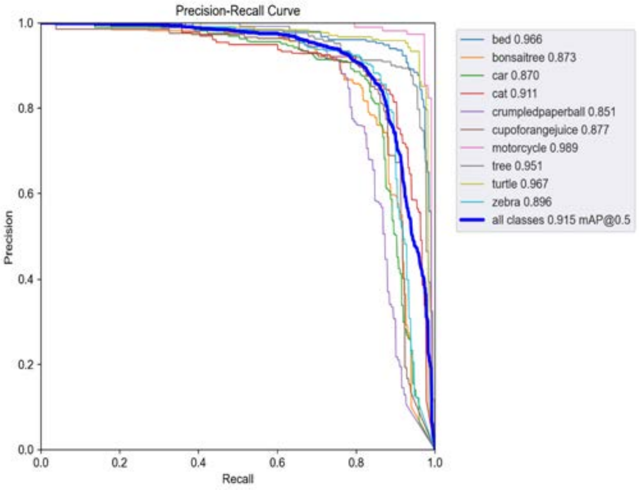


FIGURE 10: PRECISION-RECALL CURVE.

The results obtained in the experiment highlight the efficacy of our approach, showcasing an impressive accuracy rate of 98% in successfully solving hCAPTCHA challenges. Furthermore, the system exhibits efficiency with the average time required to crack a single challenge being a mere 3.5 seconds. Notably, these accomplishments were achieved using a system configuration comprising a CPU with 2 Intel Core i5-4300U processors, 16 GB of RAM, and no GPU devices. However, testing the solution on a different dataset would be necessary to further justifies the effectiveness of the proposed solution.

A. *Category Frequency*

Figure 11 illustrates the Category and Frequency details of the images used in the experiment, with categories represented on the x-axis and their corresponding frequencies on the y-axis.

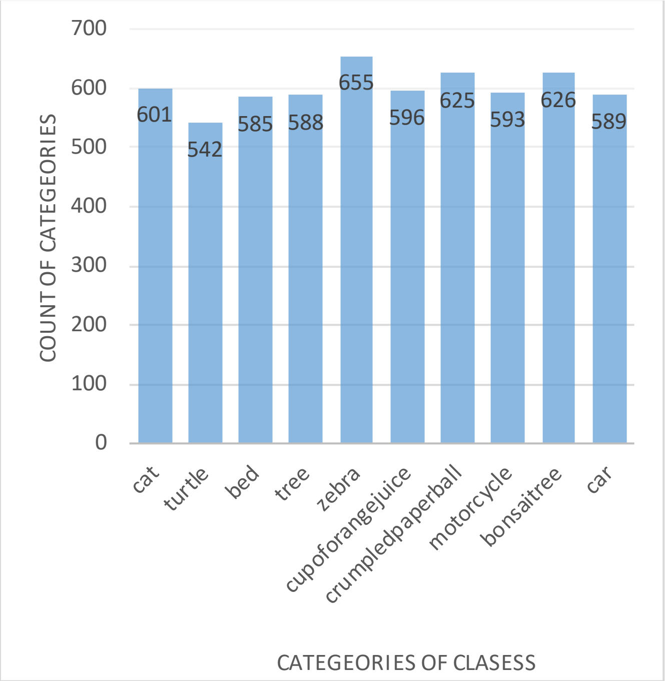


FIGURE 11: FREQUENCY OF DIFFERENT CATEGORIES OBSERVED DURING THE CHALLENGE.

B. *Average Response*

Figure 12 illustrates the average response time in seconds, denoting the duration required by the Image CAPTCHA Solver to resolve a challenge, with the y-axis representing time and the x-axis indicating the categories. As evident from Figure 12, the average response time for the system to resolve a challenge is 3.5 seconds.

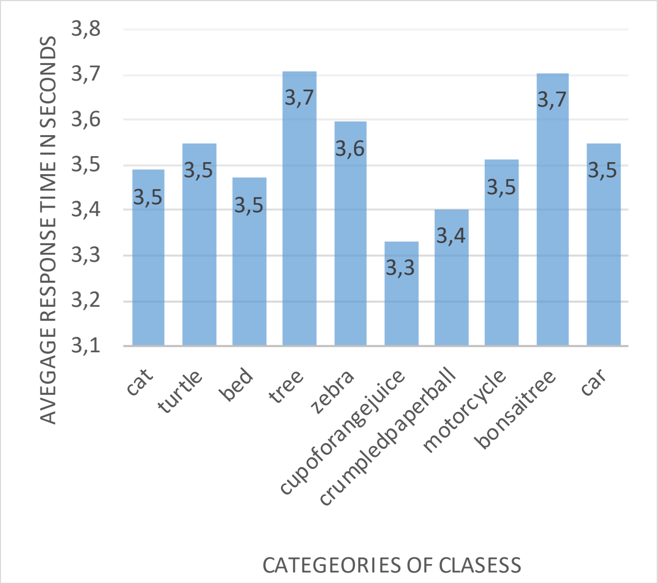


FIGURE 12: AVERAGE RESPONSE TIME FOR CRACKING OR SOLVING A CHALLENGE PER CATEGORY.

C. Accuracy

The system's accuracy was evaluated by analyzing successful challenges completed versus failures encountered. As depicted in Figure 13, out of 6000 challenges presented, the system successfully resolved 5867, while encountering 133 failures. This results in a remarkable 98% success rate in overcoming challenges, which is commendable. However, as part of future exploration, there is a need to further expand the categories and further test the scalability of the system.

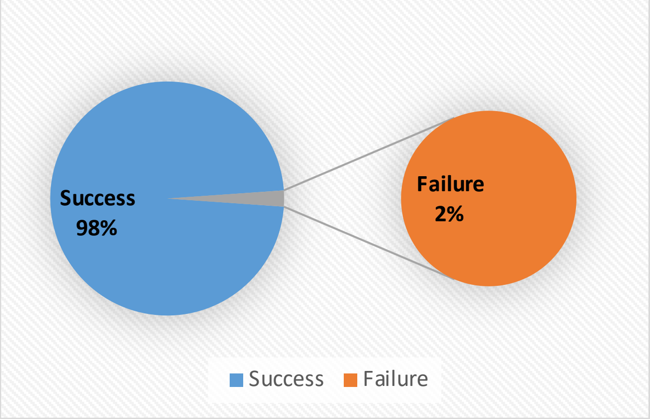


FIGURE 13: A DEPICTION OF THE ACCURACY OF THE SYSTEM DENOTING SUCCESS RATE VS FAILURE RATE.

V. SUMMARY AND CONCLUSION

This work presents a low-resource and high-success rate solution on the hCAPTCHA service using YOLO v3. The implemented automated hCAPTCHA breaker solves ten different categories and the average time taken to successfully crack a single challenge is approximately 3.5 seconds. The solution presented involved carefully curated comprehensive datasets to eliminate the dependency on existing static datasets due to the dynamic nature of hCAPTCHA. The dataset encompasses 10 categories, comprising roughly 1000 images per category. The implemented solution achieved a 98% accuracy. However, the training time to get the best results was over 2 hours.

In the future, we aim to compare our findings with other higher versions of YOLO and advanced models. Additionally, in terms of scalability, expanding the dataset to include more diverse object classes will enhance the model's generalizability and robustness.

REFERENCES

[1] M. M. Elbalky, A. Medhat, Tawfeek, and H. M. Mousa. “A comprehensive Study for Different Types of CAPTCHA Methods and Various Attacks”. Journal of Emerging Technologies and Innovative Research JETIR June 2021, Volume 8, Issue 6. SSRN: https://ssrn.com/abstract=3873233

[2] V. Ragavi, G. Geetha, “CAPTCHA and its applications”. Journal of Computer Science Engineering and Information Technology Research (JCSEITR) ISSN(P) 2250-2416 ISSN(E) Applied Vol. 4, Issue 1, Feb 2014, 11-16.

[3] S. B. Samuel and D. B. Nicholas, B. Sajal. “I am Totally Human: Bypassing the reCAPTCHA”. 2017 13th International Conference on Single-Image Technology and Internet-Based Systems (SITIS), USA.

[4] X. Ling-Zi and Z. Yi-Chun. “A Case Study of Text-Based CAPTCHA Attacks”. In International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Beijing 2012 IEEE.

[5] Z. Xu, Q. Yan, F. R. Yu, V. C. M. Leung. “A Survey of Adversarial CAPTCHAs on its History, Classification and Generation”. Nov 2023, ACM 37, 4, Article 111.

[6] G. Ruti and Idan. “CAPTCHA: Impact on User Experience of Users with Learning Disabilities”. The Academic College of Tel-Aviv–Yaffo, Yaffo, Israel Interdisciplinary Journal of e-Skills and Lifelong Learning · December 2016.

[7] G. Ombretta, “A study on Accessibility of Google ReCAPTCHA Systems”. DOI: https://doi.org/10.1145/3524010.3539498 OASIS22: Open Challenges in Online Social Networks, Barcelona, Spain, June 2022.

[8] Y. Jeff and S. E. A. Ahmad. “Usability of CAPTCHAs Or usability issues in CAPTCHA design”. School of Computing Science Newcastle University, UK. Proceedings of the 4th Symposium on Usable Privacy and Security, SOUPS 2008, Pittsburgh, Pennsylvania, USA, July 23-25, 2008.

[9] S. Sivakorn, J. Polakis, A. D. Keromytis. “I'm not a human: Breaking the Google reCAPTCHA”. Columbia University, New York NY,USA 2016.

[10] J. Schmidhuber. “Deep Learning in Neural Networks: An Overview Technical Report”. Technical Report IDSIA-03-14 / arXiv:1404.7828 v4 [cs.NE] (88 pages, 888 references), 8 October 2014.

[11] A. Searles, Y. Nakatsuka, E. Ozturk, A. Paverd, G. Tsudik, A. Enkoji. “An Empirical Study & Evaluation of Modern CAPTCHAs”. arXiv:2307.12108v1 [cs.CR] 22 Jul 2023.

[12] V. P. Singh, P Pal. “Survey of Different Types of CAPTCHA”. (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2), 2014, 2242-2245.

[13] J. Elson, J. R. Douceur,J. Howell, and J. Saul. “Asirra: A captcha that exploits interest-aligned manual image categorization”. Conference: Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007, pages 366–374.

[14] P. Golle. “Machine learning attacks against the asirra captcha”. In Proceedings of the 15th ACM Conference on Computer and Communications Security, CCS '08, pages 535–542, New York, NY, USA, 2008. ACM.

[15] Y. Rui and Z Liu. “Artifacial: automated reverse turing test using facial features”. Multimedia Systems, 9:493–502, 06 2004.

[16] B. B. Zhu, J. Yan, Q. Li, C. Yang, J. Liu, N. Xu, M. Yi, and K. Cai . “Attacks and design of image recognition captchas”. In Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS'10, pages 187–200, New York, NY, USA, 2010. ACM.

[17] S. Sivakorn, I. Polakis and A. D. Keromytis. “I am robot: (deep) learning to break semantic image captchas”. 2016 IEEE European Symposium on Security and Privacy, Mar 2016.

[18] H. Weng, B. Zhao, S. Ji, J. Chen, T. Weng, Q. He, and R. Beyah. “Towards understanding the security of modern image captchas and underground captcha-solving services”. Big Data Mining and Analytics, 2:118–144, 06 2019.

[19] M. I. Hossen, H. Xiali. “A Low-Cost Attack against the hCaptcha System” arXiv:2104.04683v1 [cs.CR] 10 Apr 2021.

[20] M. I. Hossen, Y. Tu, M. F. Rabby,M. N. Islam, H Cao, and X Hei. “An Object Detection based Solver for Google's Image reCAPTCHA v2”. In 23rd International Symposium on research in attacks, intrusions and defenses (RAID 2020) (pp. 269-284).

[21] J. Redmon, S. Divvala, R. Girshick and A. Farhadi. “You Only Look Once: Unified, Real-Time Object Detection”. University of Washington, Allen Institute for AI, Facebook AI Research http://pjreddie.com/yolo/ arXiv:1506.02640v5 [cs.CV] 9 May 2016.

[22] J. Terven and D. M. Cordova-Esparza. “A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and Beyond”. arXiv:2304.00501v1 [cs.CV] 2 Apr 2023.

[23] J. Redmon, A. Farhadi. “YOLO9000: Better, Faster, Stronger”. Conference: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). arXiv:1612.08242v1 [cs.CV] 25 Dec 2016.

[24] J. Redmon, A. Farhadi. “YOLOv3: An Incremental Improvement”. The University of Washington IEEE Trans. Pattern Anal. 2018, 15, 1125–1131.

[25] K. J. Oguine, O. C. Oguine, H. Bisallah. “YOLO v3: Visual and Real-Time Object Detection Model for Smart Surveillance Systems(3s)”. Department of arXiv:2209.12447v1 [cs.CV] 26 Sep 2022.

[26] Q. C Mao, H. M. Sun, Y. B. Liu, and R. S. Jia. “Mini-YOLOv3: Real-Time Object Detector for Embedded Applications”. Shandong University of Science and Technology, Qingdao 266590, China 10.1109/ACCESS.2019.2941547, IEEE Access.

[27] S.V. Viraktamath, M. Yavagal, R. Byahatti. "Object Detection and Classification using YOLOv3". Dept. of Electronics and Communication Engineering SDM College Of Engineering and Technology Dharwad, India. Volume 10, Issue 02 (February 2021), IJERT, 2278-0181, Paper ID: IJERTV10IS020078.

[28] L. Zhao and S. Li. "Object Detection Algorithm Based on Improved YOLOv3". Laboratory of Modern Power System Simulation and Control and Renewable Energy Technology, Ministry of Education, Northeast Electric Power University, Jilin 132012, China; Received: Published: 24 March 2020.

[29] N.Tariq, F.A.Khan, S.A.Moqurrab and G.Srivastava. "CAPTCHA Types and Breaking Techniques: Design Issues, Challenges, and Future Research Directions". Brandon University, Canada arXiv:2307.10239v1 [cs.CR] 16 Jul 2023

[30] S.R. Ibadov, B.Y. Kalmykov , R. Ibadov and R.A. Sizyakin. "Method of Automated Detection of Traffic Violation with a Convolutional Neural Network", The European Physical Journal Conferences 224:04004, 2019.

[31] E. Beauxis-Aussalet and L. Hardman: "Visualization of Confusion Matrix for Non-Expert Users", In proceedings of IEEE Vis, Pp. 1-2, 2014

[32] S. Visa, B. Ramsay, A Ralescu and E. V. D. Knaap. "Confusion Matrix-based Feature Selection". Conference: Proceedings of The 22nd Midwest Artificial Intelligence and Cognitive Science Conference 2011, Cincinnati, Ohio, USA, April 16-17, 2011.

[33] Y Yin, H. Li, W. Fu. "Faster-YOLO: An accurate and faster object detection method", Digital Signal Processing (IF 2.9) Pub Date: 2020-05-04 , DOI:10.1016/j.dsp.2020.102756.

[34] M. M..Elbalky, M. A. Tawfeek and H M. Mousa. "A comprehensive Study for Different Types of CAPTCHA Methods and Various Attacks". Menoufia University, Shebin Elkom 32511, Egypt, 2021 JETIR June 2021, Volume 8, Issue 6

[35] S. Deosatwar, S. Deshmukh, V. Deshmukh, R. Sarda and L. Kulkarni. "An Overview of Various Types of CAPTCHA". In book: Information and Communication Technology for Intelligent Systems, Proceedings of ICTIS 2020, Volume 1 (pp.261-269).

[36] V. Rathi and R. Suryawanshi. "Different Types of CAPTCHA: A Literature Survey" Open Access International Journal of Science and Engineering. P.G.M.C.O.E Pune, Volume 3Special Issue 1 March 2018ISO 3297:2007 Certified ISSN (Online) 2456-3293.

[37] M. Kumar and M. Sahib. "Detailed Analyses of Different Types of CAPTCHA". International Journal of Management, Technology And Engineering Volume 8, Issue VIII, AUGUST/2018 ISSN NO : 2249-7455.

[38] G. Aishwarya, V. S. Kumar, SNS Rajalakshmi. "A Study on CAPTCHA Techniques & Its Applications". Coimbatore, Tamil Nadu, India - 641 049 IJSRD - International Journal for Scientific Research & Development Vol. 5, Issue 08, 2017 ISSN (online): 2321-0613.

[39] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu. "Object detection with deep learning: A review". January 2019, IEEE Transactions on Neural Networks and Learning Systems PP(99):1-21, PP(99):1-21.

[40] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation". in CVPR, 2014.

[41] K. Fukushima. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". Biological cybernetics, 36(4):193–202, 1980

[42] N. Tariq and F. A Khan. "Match-the-sound captcha". In Information Technology-New Generations, pages 803–808. Springer, 2018.

[43] S. Acharya, R. Basak and S. Mandal. "Solving Arithmetic Word Problems Using Natural Language Processing and Rule-Based Classification". March 2022 International Journal of Intelligent Systems and Applications in Engineering 10(1):87-97.

[44] I. Akrout, A. Feriani , M. Akrout. "Hacking Google reCAPTCHA v3 using Reinforcement Learning". University of Toronto.arXiv:1903.01003v3 [cs.LG] 18 Apr 2019.