Real-time CAPTCHA using Hand Gesture recognition for highly secure websites

D Lasya Priya, Beru Neha, B Vivek Sai Chinna, B Sai Sujay, DivyaPrabha K N
Computer Science Department, PES University, Bangalore, India
lasyad2k01@gmail.com, nehaberu9@gmail.com, viveksaichinna@gmail.com, saisujay3@gmail.com, divyaprabhamadhu@gmail.com

***Abstract*—Completely Automated Public Turing Tests to Tell Computers and Humans Apart (CAPTCHAs) are now almost a routine security measure which protect against unwanted and malicious bot programmes on the Internet. CAPTCHAs significance in Web Security is that it prevents sensitive information from scrapers, prevents dictionary attacks and DDoS attacks. There are several security threats on websites and it would be a major risk to the nation if defense websites or other classified material were exposed. Several algorithms are in place to solve CAPTCHAs automatically. The true definition of CAPTCHA is that it must be able to determine that humans, not computers, are attempting to get into a password-protected account. Our work provides an efficient and highly secure alternative to classic CAPTCHA.The objective is to essentially device a two-step "bot-proof" authentication process with camera access as a requirement. According to studies, implementing Multi-factor authentication makes a specific account 99.9% less likely to be penetrated, and similarly building a two-level CAPTCHA would surely improve the security of the user being attacked. The first level involves a Text-CAPTCHA and in the second level, a new CAPTCHA technique that recognizes user hand gestures in real time aids in preventing the possibility of algorithm breaking by attackers. Hand gesture detection was first implemented using Support Vector Machine (SVM) and improvised with Convolution Neural Network (CNN). Finally, Mediapipe assisted and produced faster and accurate results in real time.The authentication will be granted to the website if the human performs the given task successfully in the second level.**

***Index Terms*—CAPTCHA, Hand Gesture Recognition, SVM, CNN, Mediapipe, Web Security**

I. Introduction

This work proposes a simple, real-time and efficient two step bot-proof process of authentication. The complete architecture of our work is as shown in Figure 1. The user need to first login to the website and enter the credentials correctly. Text-based CAPTCHA [1] is used in the initial authentication phase to differentiate harmful bots from people. Background interference, noise lines, and geometric modifications are just a few of the resistance mechanisms that researchers have added to CAPTCHA in order to increase security. To make it harder, the alphabets and numbers are distorted and rotated. The computer generates a string of six characters. The user must submit an input that matches the computer-generated string to advance to the second level.The user gets two attempts to enter the right answer in either case, failing to do which, the access to the website will be denied. However, all of these protection systems have been dismantled thanks to the development of deep learning algorithms. The text-based CAPTCHA's security is frequently shielded by a number of mechanisms to hinder machine recognition [2]. The majority of prior attacks against text CAPTCHAs [1] have used a three-step strategy: preprocessing, segmentation, and recognition. Thus, the introduction of second level hand gesture recognition.

The second level is a hand gesture recognition algorithm based on computer vision. The goal of gesture recognition study is to develop a system that can identify and manage applications that use human motion. Gesture recognition systems can be divided into two basic categories: vision-based and data-glove based [3]. Because the user is required to carry a burden of cables and wires attached to the computer, this method obstructs the convenience and naturalness of user engagement. Data-Glove-based techniques use mechanical or optical sensors attached to a glove to transform finger flexions into electrical signals, which are then used to recognize hand position. A relatively established method is hand gesture recognition based on vision, which employs cameras to record films of scenes with motions before using algorithms on computers to recognise, extract, and classify the gesture elements in the photos. Detection, tracking, and recognition are the three core components of the majority of fully integrated hand interactive mechanisms that serve as the foundation of vision-based hand gesture recognition systems [4]..Using vision-based techniques for hand gesture detection is the most natural way to construct a human-computer interface.

The scope of this work is that it needs camera access to pass the second step of CAPTCHA successfully and hence it is mainly for websites that require high security. A camera captures a live video stream, which is then used to create a snapshot via an interface. Only a few hand motions have been programmed into the system. After that, it is given a test gesture, which the system attempts to recognise. Several algorithms were utilised and evaluated in order to find the most accurate algorithm.

SVM [5] and CNN [6] algorithms were implemented by generating our own dataset of around 2000 images containing six gestures. Each gesture differs based on the number of fingers displayed by the user. When compared to SVM, CNN has a higher accuracy of around 98.3%. Finally, MediaPipe [7] was used for second level CAPTCHA, which relies on OpenCV [8] for video data handling. Each frame of the webcam video capture is used to run the MediaPipe Hands process function in the implementation. The outcomes give each hand detected a 3D landmark model [9] for each frame. Palm Detection model and Hand Landmark model are used which gives fast, accurate and real-time results.

The significance of the research work are as follows. The work proposes a highly secure approach to providing security to users. The main target users are websites containing highly sensitive information that needs a more effective method to preserve information. This is a novel approach compared to other CAPTCHA methods in the present-day security market. No existing approach uses real-time gesture recognition to implement CAPTCHA. A variety of bots are constantly being created to defeat modern CAPTCHAs. Moreover, present day technology is not advanced enough to replace an actual human hand. This gives us an edge over other security methods as no bots can be developed against this method.

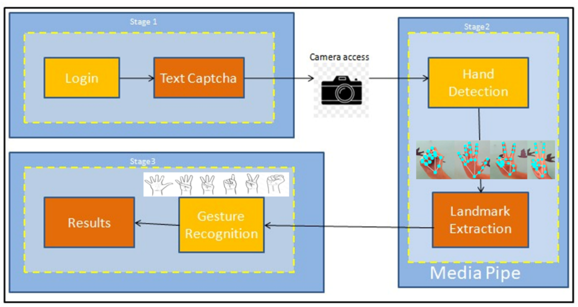

Fig. 1: High Level Design diagram

The remainder of the study is structured as follows. The second section examines the various hand gesture detection and identification techniques as well as the existing CAPTCHA approaches for safeguarding Web Pages. Section 3 presents the suggested methodology, which includes the SVM, CNN, and Mediapipe Algorithms. Results from Mediapipe and OpenCV are reported in section 4. In the last section, the work is concluded and future research direction is highlighted.

II. LITERATURE SURVEY

A comparison of various CAPTCHA algorithms for securing web pages is shown in [10]. CAPTCHAs come in a wide variety of forms, but they haven't always worked well. Even without a natural interface, traditional input devices can still be used to interact with computers. Images captured by web camera. Hand tracking and segmentation are the primary steps in any hand-based gesture recognition. Segmentation is a technique for identifying the image's related regions. Clients who wish to use a hand gesture must wear an information glove or use a web camera to capture a picture of their hand.

1. Text CAPTCHA—After entering the CAPTCHA text, users can log in to the secure website, making the content easy to view. High character collapsing may be difficult for users to understand.

2. Image CAPTCHAs are easy for users to complete since, after verifying the image, they can quickly access a secure website. We may now finally access the secure website. Bright backgrounds are the only drawback to image-based CAPTCHA.

3. Audio CAPTCHA is extremely helpful for blind persons. The audio-based CAPTCHA presents difficulties for blind users when the network is unreliable or offline.

Although an image-based CAPTCHA has been proposed as a replacement for content-based CAPTCHA, its limitations make its replacement challenging. We have seen that the most secure and user-friendly CAPTCHAs for distinguishing humans from machines are reCAPTCHA and NuCAPTCHA. Multiple strategies have been incorporated in the module to create barriers for users that use reCAPTCHA and NuCAPTCHA, as well as assaults involving optical character recognition and moving picture object recognition. People can easily recognise the arrangement of characters in reCAPTCHA and NuCAPTCHA, making them extremely useful.

In [11] the proposed model, the CAPTCHA is displayed on screen leading to image acquisition, pre-processing of the input, matching the content with the database and finally displaying the result. Pre-processing includes 3 steps- Image conversion is when the image input taken from camera is converted into binary image. Morphological Filtering filters out the noise from the binary image. There may be some parts of the hand showing 0 and background as 1. This step clears out all the noises. Edge detection gets the edge of the image to determine the shape Canny edge detection algorithm is used to eliminate the risk of multiple responses. 20 distinct people's hand gestures for training and testing the model, five individual gestures and hundred photos are taken and kept in the database.The model has given an accuracy of 80% with the algorithms which has been mentioned.

The steps followed in [12] is to first take a frame of video and performing hand segmentation, tracking and the does the recognition.Hand gesture segmentation analyses the photos and creates a Gaussian mixture model based on skin colours and additionally, it classifies hand movements using an AdaBoost classifier based on Haar features. The grayscale value change in the image is reflected by the Haar feature. The feature model is made up of the black and white rectangle portions. The key concept is to convert the original image into an integral vision, and the rectangle feature may be easily determined using two integral images. A total of 16000 hand gesture pictures are used in the experiment, which employs hand gestures ranging from 1 to 10 against a dynamic background. The test set then consists of 400 images from each of the 4000 categories of hand actions. Hand gesture identification has a 98.3 percent accuracy rate on average. However, the identification rate is poor since the hand motions for the numerals 7 and 9 are challenging. Using an AdaBoost classifier which is based on the Haar feature, hand gesture segmentation in a challenging environment is made achievable. The hand gesture area is acquired in real time utilising a camshaft algorithm for hand gesture tracking based on the movement of hand gestures and deformation data, and the hand gesture area is subsequently classified using a convolution neural network.

AlexNet layers that are fully connected are used to capture deep features in the study's PCA-based deep CNN reduced features method [13]. The gesture is recognised using a classifier built on a support vector machine. Using the enormous label ImageNet dataset, the CNN model was trained to handle the challenge of categorising images into several categories. In this work, hand motion photos of ASL gestures are classified using pre-trained AlexNet. This deep CNN was built using five convolutional layers, three max pooling layers, and three fully-connected layers which is the deep CNN's layer architecture. The MU dataset was created using five people and 36 gesture.Hand rotation and articulation are all varied in the gesture postures, while the lighting is changed in five different directions.The proposed technique's mean accuracy is 87.83 percent, which is higher than Alexnet's 'FC6' feature's mean accuracy.The suggested technique has a mean recognition accuracy of 13.97% and 3.81 percent greater than CNN and FCNN techniques, respectively.

In [7] Mediapipe offers a wide range of results within the holistic channel and it consists of three factors: pose, hand and face. The aggregate of 543 distinct landmarks is obtained using this holistic approach. The Mediapipe Python Module's Holistic Model for Mediapipe is what we are utilizing in which the function changes the image's BGR to RGB format before passing it to the model.process() function and storing the result. This function returns the concatenated array of all the arrays containing the key point coordinates of the holistic model. The first step was data collection through our own dataset for the training and testing of deep learning model. The next step was to define an array of 10 gestures that would be used to train our recognition algorithm. Then collecting the videotape data for each gesture and use the extract keypoints function to collect the array of key points for each frame. Eventually, the numpy array of key points collected is saved for each frame inside each video folder. The original dataset, which was split into training and testing data, served as the source for the dataset that was employed for testing. The LSTM model is successfully operating with respect to the test dataset with an accuracy of 90%.

III. Proposed Methodology

The suggested approach considers the difficulties that earlier models experienced and works to reduce them. We created a system that does not trade off performance for efficiency. Our initial approach involved hand detection, followed by segmentation of the palm and fingers, and then hand gesture identification. The separation of the hands from the background and the non-uniform background were two of the most significant problems encountered. Later, SVM and CNN algorithms were used to overcome this drawback. We created our own dataset using OpenCV which was used for the above algorithms mentioned. CNN produced better results compared to SVM but having the need to use a dataset and store data was inefficient and did not produce real-time results. Using Google's Mediapipe solution and the OpenCV package, which creates the essential algorithms for analysing the images and real-time movement recognition, we were able to solve that issue. It is used to recognise hand gestures in real time as well as to recognise landmarks on the hand.

For the detection and recognition of hand gestures, several methods were employed.

1. Hand detection

It is accomplished by using a typical camera to take pictures. To identify the hand region from the original image, use the quick and efficient background subtraction method. The hand can be distinguished from other moving objects by its colour. The skin's colour is determined using the HSV model. The hue, saturation, and value (HSV) values for the skin tone are 315, 94, and 37, respectively.

The following is the algorithm for colour segmentation (background subtraction method) using thresholding:

Take a picture of the gesture with the camera. Determine the skin colour HSV value range that will be utilised as threshold values. Change the image's colour space from RGB to HSV. Convert all pixels that are below the threshold values to white. Make the rest of the pixels dark. As an image file, save the segmented image. Before the gesture in the colour segmented image can be analysed, it must be recognised as a single object. This can be accomplished through the labelling and blob detection methods. Labelling is the process of assigning a unique integer number or label to each location for the purpose of identifying it. The adjacency relationships between pixels can be used to determine whether they belong to the same region. 4-adjacency and 8-adjacency are the two most common adjacency relationships.

Fingers and Palm Segmentation

Fingers and the palm can be segmented with ease using the palm mask. The hand's palm is covered by a mask, and the remaining fingers make up the remainder of the hand. Palm point: The palm point is the middle point of the hand. Utilizing the distance transform method, it is found. Each pixel records its distance from the closest boundary pixel. A matrix of 0s and 1s will be present. Consider The Maximal Radius' Inner Circle and Wrist Points and Palm Mask.

Hand Rotation - Once the palm point and wrist point have been established, an arrow can be drawn from the palm point to the middle point of the wrist line at the base of the hand. The arrow then changes direction to point north. Due to the hand image rotating in unison with the rotation, the hand gesture is invariant.

Fingers recognition

An algorithm for labelling is used to label the finger areas. The labelling procedure results in the discovery of noisy patches, which are then removed, if the number of pixels is too little. Only the sections of sufficient size are considered fingers and are retained. The smallest bounding box to enclose each remaining region is a finger.

Recognition and Detection of Thumb The centres of the fingers and the palm point are aligned. After that, the distances between these lines and the wrist line are calculated. The thumb will be seen in the hand image if the angle is less than fifty degrees. The similar centre is where the thumb rests. On the thumb that has been identified, the number 1 is written. The thumb does not appear in the image if all of the degrees are greater than fifty degrees. Finding and Identifying Other Fingers - The palm line is initially explored to locate and identify the other fingers. The lines at the wrist and the palm are parallel.

Hand gesture recognition

Once the fingers have been detected and identified, a straightforward rule classifier may be utilised to recognise the hand motion. The rule classifier predicts the hand motion based on the number and composition of detected fingers. Using the rule classifier is effective and powerful.

2. Support Vector Machine (SVM)

One of the supervised classification techniques is SVM. To make it function, a hyperplane is made with the greatest possible margin between it and the data, dividing each class of data. SVM is capable of handling learning tasks with a high number of features. That is the reason why we decided to use SVM as one of the algorithms in our data. When the sample can be separated linearly, SVM is the best classification plane that has been suggested. In order to ensure that the two types of categories can be recognised from one another without making any mistakes and that the interval between the two types is as wide as possible, the so-called optimal classification is required. The best classification surface is created by extending the best classification line into high-dimensional space. Because SVM has numerous distinct advantages in tackling small sample, nonlinear, and high dimensional pattern recognition problems, it is employed in gesture recognition.

The edges of the hand are located using Canny's Edge Detection. To extract pertinent characteristics from an edge detected image, Histogram Of Oriented Gradients (HOG) is used. The theoretical foundation for resolving illumination and gesture rotation interference issues is provided by the HOG features' ability to maintain a good invariance to the geometry and optical changes of a local image. The image is cropped to only contain the area that is of interest to us and pre-processed to lessen noise. To anticipate gesture, SVM model training uses recovered attributes.

3. Convolution Neural Network (CNN)

CNN is a Deep Learning algorithm that can take in an input image, rank various items within the image, and distinguish between them. Convolutional, pooling, and dense layers are present in CNN. Because it can extract the necessary feature values from the input image and because it can train on a huge number of samples, CNN is more widely used in the field of recognition and produces better results than other techniques. There is no need to use complex algorithms to extract and learn visual information when using a CNN to learn human gestures. Invariant features are permitted with little dislocation via the convolution and sub-sampling levels of a CNN.

The images in the dataset first go through pre-processing. To facilitate predictions, images from the dataset are transformed to a suitable colour palette. Then, the images are loaded and ready for the algorithm training. For improved outcomes, the hyperparameters are tuned. Table 1 lists the CNN hyperparameters. Training and test sets of images are created. The test data is used to determine whether the predictions are accurate, while the training set is used to develop the model. High precision is attained after several execution epochs. On providing the camera access, the results are displayed by processing and using the trained CNN model as shown in Figure 2.

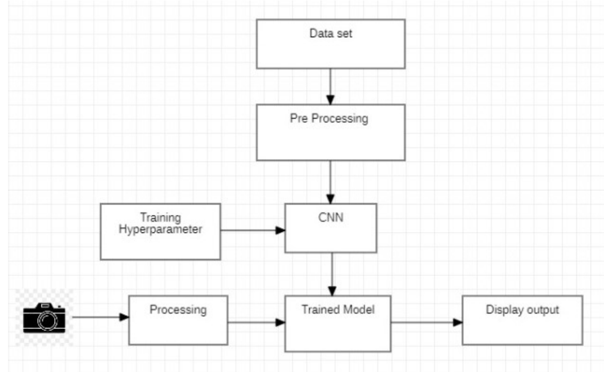

Fig. 2: Workflow of implementation using CNN

Table 1: CNN Hyperparameters

Training Parameters	**Neural Network**
Rate of Learning	0.1
Size of each Batch	64
Number of Epochs	5
Optimizer	Adam
Loss Function	Categorical Cross-entropy
Activation Function	Relu and Softmax
Dropout	0.5
Total Parameters	4104522

The pseudo-code and general implementation of the proposed work is as follows:

```
//User needs to access a highly secure website
//User first fills in login info and the presented text captcha
If (user login details or text captcha response) not right:
        Re-enter respective details
If (level 1 passed within 2 attempts):
        //Proceed to level 2
        User MUST provide camera access
        If (user replicates gesture image within 2 attempts):
                //Provided access to his/her account
        Else:
                //Denied access after level 2 captcha
Else:
        //Denied access after level 1 captcha
Exit
```

4. MediaPipe

It is an open-source framework for creating machine learning processing pipelines for time-series data, including audio and video. The free source MediaPipe framework is used to implement the suggested hand motion recognition system. The system extends MediaPipe Hands, which essentially comprises of a hand detection and a hand-keypoint component, with a further gesture-classification component. A real-time hand detection and hand tracking technology called MediaPipe Hands can forecast the skeleton of a human hand. 21 hand landmarks were detected and tracked, and the results are shown in Figure 1 for each x, y, and z axis. Although the user rarely makes movements in many applications, such as remote controls, the Hand Gesture Recognition is always operating in the background. The Hand Gesture Recognition system is set up to operate only when necessary, with a maximum frequency lower than that of hand keypoint generation and gesture classification. This is accomplished by employing the flow-control and stream-synchronization capabilities in MediaPipe. By doing this, the average amount of processing required is decreased but real-time performance is maintained across a variety of devices. To conserve computation and electricity, the hand gesture recognition executes hand detection at a lower frequency. When a hand is found, it is immediately tracked more frequently for improved accuracy and temporal resolution.

Real-time gesture recognition begins with the fundamental stage of finger detection. The aforementioned models are continuously fed frame-by-frame footage recorded from the user's camera. Figure 3 displays the outcomes of the detection and tracking of each of the 21 x, y, and z hand landmarks. With the help of the coordinates generated by the hand landmarks model, we can compare the relative positions of different parts of each finger to detect whether it is folded or open. It is used to accurately show the number of fingers open in real-time speed. OpenCV's landmark model to detect and count the number of fingers a user is trying to show. This is done with the help of the coordinates in the landmark model. It is used to detect and show which fingers are open.

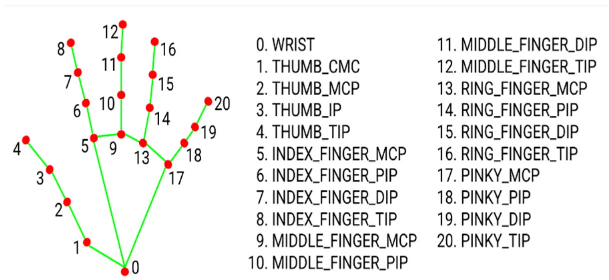

Fig. 3: Hand Landmark Model

The experimental results illustrate that the number of fingers are accurately shown with real-time speed. Even the issue of the user unsteady is taken care of with the accuracy of the model. Other issues such as unstable video input and other disturbances in the background are taken care of by the model.Each frame of the webcam video capture is used to run the MediaPipe Hands process function in the implementation.The outcomes give each hand detected a 3D landmark model for each frame.

Gesture is recognised by the user by taking note of all the open fingers. Open fingers are recorded at every frame using finger coordinates obtained with the landmark model.These detected finger names are displayed in real time on the screen to aid the user further. The user then captures the webcam input after recreating the gesture at any time the user is comfortable with, within the 20 second constraint. Using the above obtained information about the open fingers, we then verify whether the open fingers in the user's screen capture match the open fingers in the question.The set of detected open fingers is then compared to open fingers in the gesture. The user is supposed to recreate as shown in Figure 4 and Figure 5, in order to check if the answer in correct or wrong and provide authentication to the website if correct.

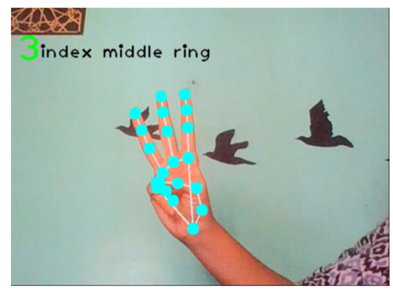

Fig. 4: fingers open in question: ['index','middle']
fingers open in screencapture: ['index','middle','ring']
WRONG ANSWER! TRY AGAIN.

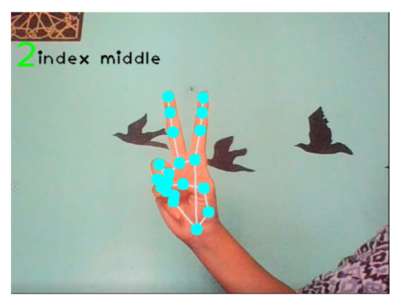

Fig. 5: fingers open in question: ['index','middle']
fingers open in screencapture: ['index','middle']
CORRECT ANSWER!

IV. RESULTS AND DISCUSSION

Six different hand gestures totaling 2450 photos were divided into training and testing data at the ration 80:20 for the SVM and CNN models, respectively. Images from the dataset are formatted to a suitable colour palate to make predictions easier. On passing dataset through the SVM model, the overall accuracy was about 89.14% and for CNN model, accuracy of 99.362% is achieved after multiple epochs of execution as shown in Figure 6. In terms of accuracy and recall, the CNN model is 1% more accurate than the SVM model, although both models' rates of precision are equal. Mediapipe module offered by python helps in plotting landmarks on a detected hand. It detects outline of the user's hand and was seen to be accurate most of the time for the right hand. The model has an average accuracy of 95.7% overall. Considering the fact that we altered it for only the right hand, it provides better accuracy. Since it gives real time gesture recognition, it proves to be a better alternative for SVM and CNN.

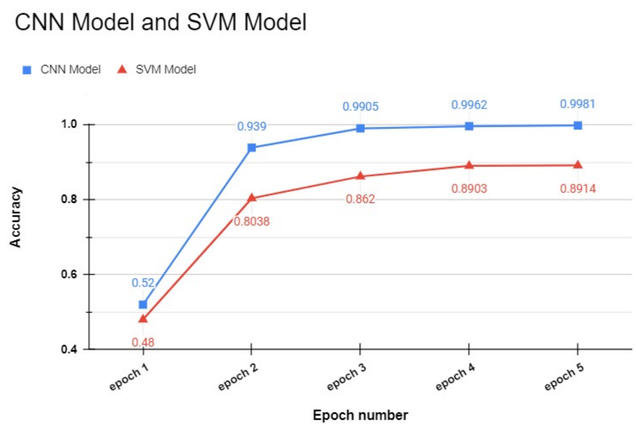

Fig. 6: Accuracy of SVM vs CNN Plot

Our analysis demonstrated that combining synthetic and real-world information produces the best results for the hand landmark model. We exclusively used real-world images for evaluation. Training using a big synthetic dataset lowers visual between frames in addition to improving quality. Based on this fact, we draw the conclusion that our real-world dataset can be increased for better generalisation. We're aiming for real-time performance on extremely secure websites. Multiple datasets and a variety of algorithms, including CNN, SVM, and finally mediapipe and OpenCV, were used in our tests. MediaPipe Hands is a high-quality hand and finger tracking system. It extrapolates 21 3D hand landmarks from a single frame using machine learning (ML). It offers a fantastic compromise between speed and quality. Only slight quality gains are made as model capacity is increased, but speed is drastically reduced.

V. CONCLUSIONS AND FUTURE WORK

Real-time vision-based hand gesture identification is one of the most challenging research issues in the field of human-computer interaction. To recognise hand movements, vision-based hand gesture identification can use generic video cameras found on a wide range of computers, tablets, and smartphones. Other interactive human-computer systems, such as data gloves, keyboards are less intuitive, efficient, and convenient than non-contact visual gesture detection.

An effective Hand Gesture Based CAPTCHA system was proposed and was implemented. In this method, a form containing a CAPTCHA image of a hand gesture from the database was displayed along with a message asking the user to copy the gesture. The user then made a gesture in front of the system's camera out of conscience, and the model checked to see if it matched the gesture in the image or not. As it is impossible for the robot to learn how to copy the identical move as displayed in the image on its own, using this approach would secure the identification of the user.

The experimental data demonstrates that our system will function better when the input image has good quality. It is safe and incredibly difficult to crack. This can also be expanded to a client-server-based architecture, where a CAPTCHA processing server is built up with an API to embed created CAPTCHA on different websites and carry out human or bot verification CAPTCHA.

REFERENCES

[1] Ping Wang, Haichang Gao, (Member, IEEE), Ziyu Shi, Zhongn Yuan, Jiangping Hu Simple and Easy: Transfer Learning-Based Attacks to Text CAPTCHA. IEEE(2020)

[2] Jun Chen,Xiangyang Luo,Yanqing Guo,Yi Zhang,and Daofu Gong A Survey on Breaking Technique of Text-Based CAPTCHA. Hindawi(2017)

[3] Lin Guo, Zongxing Lu , and Ligang Yao Human-Machine Interaction Sensing Technology Based on Hand Gesture Recognition: A Review. IEEE(2021)

[4] S.S.Rautaray and A.Agrawal, "Vision-based hand gesture recognition for human computer interaction: A survey,"Artif.Intell.Rev.,vol.43,no.1,pp.1–54,2015.

[5] Kai-ping Feng, Fang Yuan Static Hand Gesture Recognition Based on HOG Characters and Support Vector Machines. IMSNA(2013)

[6] Hsien-I Lin†, Ming-Hsiang Hsu, and Wei-Kai Chen Human Hand Gesture Recognition Using a Convolution Neural Network IEEE(2014).

[7] Sachin Agrawal, Agnishrota Chakraborty,M. Rajalakshmi Real-Time Hand Gesture Recognition System Using MediaPipe and LSTM. IEEE(2022).

[8] Ruchi Manish Gurav, Premanand K. Kadbe Real time Finger Tracking and Contour Detection for Gesture Recognition using OpenCV. ICIC(2015)

[9] Khawaritzmi Abdallah AHMAD, Dian Christy SILPANI and Kaori YOSHIDA Hand Gesture Recognition by Hand Landmark Classification. ISASE(2022)

[10] Shashank Awasthi, Arun Pratap Srivastava, Swapnita Srivastava ,Vipul Narayan."A Comparative Study of Various CAPTCHA Methods for Securing Web Pages."ICACTM(2019)

[11] Pooja Panwar1 , Monika1 , Parveen Kumar1,2 and Ambalika Sharma. "CHGR: Captcha generation using Hand Gesture Recognition."IEEE(2018)

[12] Pranjali Manmode, Rupali Saha, Manisha N. Amnerkar Real-Time Hand Gesture Recognition. IJSRCSIT(2021)

[13] Jaya Prakash Sahoo, Samit Ari, Sarat Kumar Patra "Hand Gesture Recognition using PCA based Deep CNN Reduced Features and SVM classifier". IEEE(2019)

[14] Okan Kop¨ ukl ¨ u¨ 1 , Ahmet Gunduz1 , Neslihan Kose2 , Gerhard Rigoll. Real-time Hand Gesture Detection and Classification Using Convolutional Neural Networks.IEEE(2019)

[15] Gyutae Park, V.K. Chandrasegar, JoongGun Park, Jinhwan Koh "Increasing Accuracy of Hand Gesture Recognition Using Convolutional Neural Network", IEEE(2022)

[16] Zhan, Felix. "Hand gesture recognition with convolution neural networks." 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI). IEEE, 2019.

[17] Anju S R, SubuSurendran, 2014, A Study on Different Hand Gesture Recognition Techniques, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH AND TECHNOLOGY (IJERT) Volume 03, Issue 04 (April 2014).

[18] Agrawal, M., Ainapure, R., Agrawal, S., Bhosale, S. Desai, S. S. (2020). Models for Hand Gesture Recognition using Deep Learning. 2020 IEEE 5th International Conference on Computing Communication and Automation, 2020,pp. 589-594, doi: 10.1109/ICCCA49541.2020.9250846.

[19] Andrea Tagliasacchi, Matthias Schroder, Anastasia Tkach, ¨ Sofien Bouaziz, Mario Botsch, and Mark Pauly. Robust articulated-icp for real-time hand tracking. In Computer Graphics Forum, volume 34, pages 101–114. Wiley Online Library, 2015
[20] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3593–3601, 2016.
[21] Kumar, B. P., Manjunatha, M. B. (2016).Performance analysis of KNN, SVM, and ANN techniques for gesture recognition system. Indian JSci. Technol., 9(1), 1-8
[22] Oudah, M., Al-Naji, A., and Chahl, J. (2020). Hand gesture recognition based on computer vision: a review of techniques. journal of Imaging, 6(8), 73.