Hand Gesture Based sCaptcha

Pawan Dharmendra Mehta
dept. Computer Science
Lakehead University
Thunder Bay, Canada
pmehta5@lakeheadu.ca

Divyarajsinh Yadav
dept. Computer Science
Lakehead University
Thunder Bay, Canada
yadavd167@lakeheadu.ca

Siddhant Badola
dept. Computer Science
Lakehead University
Thunder Bay, Canada
sbadola@lakeheadu.ca

***Abstract*—CAPTCHA is a widely used security mechanism to prevent spammers from submitting automated forms. It is a simple test that tells humans and bots apart. Traditional CAPTCHA's are a combination of distorted text and numbers or an audio with extra noise. The theory is that humans can correctly identify such distorted text while a bot can't. However, with the advancement in the field of Artificial Intelligence it is only a matter of time that such CAPTCHA's will be cracked by bots. These CAPTCHA's wastes a lot of time and efforts of users.This report aims to explain the creation of a new experimental CAPTCHA that is claimed to provide greater security against spam bots than existing CAPTCHA systems. Report contains a summary of our project. In brief, our project is about using computer vision to detect sign language, specifically numbers 0 to 9 and thus detect numbers that can be used to solve number-based CAPTCHA puzzles to verify if the user is human or a bot. Using our project, we can provide a 2-layer security to verify users from bots as we use both hand gestures and CAPTCHA based numbers which is very hard to replicate artificially.**

I. Introduction

CAPTCHA stands for Completely Automated Public Turing test to tell Computers and Humans Apart. It ensures that the user is a human and not an automated machine. A typical CAPTCHA is a combination of distorted text and numbers as shown in figure 1.

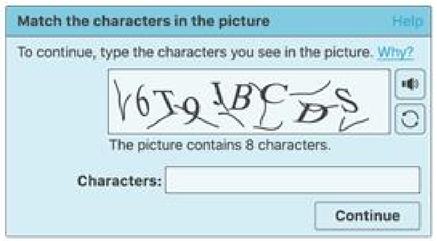

Match the characters in the picture
Help
To continue, type the characters you see in the picture. Why?
The picture contains 8 characters.
Characters:
Continue

Figure 1 A typical CAPTCHA (Source: cloudflare.com)

Gestures are a non-verbal form of communication making it an essential part of sign language. Hand gesture is a very powerful medium to communicate with deaf people, therefore it is more suitable solution for CAPTCHA implementation. Traditional CAPTCHA's are inaccessible to visually impaired individuals as they rely on visual perception. Even users without impairments can find it difficult to crack the text. To add more security to CAPTCHA, google introduced reCAPTCHA, it sources its text from real world images such as pictures of street address, text from old newspapers and so on [1]. They also introduced image recognition and checkbox CAPTCHA. Here the set of low resolution or blurry images are shown from where user has to select the images according to the question. These blurry images are also difficult to identify at certain times. Google acknowledges that its reCAPTCHA text is so complicated that even humans can solve it only 87% of the time [2]. All the issues related with traditional CAPTCHA can be solved by replacing it with a hand gesture recognition CAPTCHA.

Many individuals communicate primarily using sign language, however there has yet to be devised an artificial system capable of consistently recognizing such motions from films without any extra input. Findings The investigation revealed that it takes time for those who do not utilize sign languages in their daily lives to adjust to the new CAPTCHA at first, but that after a few successful recognitions, they are able to adapt quickly. gesture-based solution may be utilized as a general-purpose captcha system as well as on websites targeted towards sign language groups. With real-time simple gestures, we propose a

CAPTCHA system.

The system uses a camera to extract a user's real-time gesture; visual sign language for digits will be included to evaluate the reality of the real-time CAPTCHA. As long as the user has a hand, this CAPTCHA promotes usability for non-disabled users. With an emphasis on image recognition CAPTCHAs, the project aims to improve the security of picture-based Recognition systems. We will solely look at the creation of numerical digits using hand gestures in this study. With the aid of a camera, the user's real-time hand motions will be collected, and that picture frame will be matched with the CAPTCHA. If both photos match, it's safe to assume the user is human and not a bot.

II. Related Work

We have reviewed few papers regarding Hand gesture digit recognition and a couple of papers on hand gesture-based Captcha. In paper [3] hand gestures with black background were taken as input from the user. This input was then binarized with thresholding. Morphological operations were performed on the binarized image to remove hairline leaks and smoothen the image by filling the small gaps. The edge of the hand was detected with the help of canny edge detector. And finally, the pixel count of the edge was matched with the average calculated pixel for each category of image in the dataset. This approach works here because they have only used 1-5 digits. They have gathered all five gestures from 20 persons and achieved a decent accuracy of 80%. Once the pixel count of input image edge matches with that of the displayed image user is allowed to proceed further.

In another paper [4] for hand gesture digit recognition they used finger segmentation for classification. The captured image was converted into HSV image for skin segmentation. Then they used 8 neighbour method for masking of palm and wrist. Finally, the image was rotated vertically for simplicity. After that thumb was detected as the angles between the fingers and palm centre is less than 50 degree then it means that thumb is present in the image and if the angles are greater than 50 degree than that means that there is no thumb in the image. To differentiate rest of the fingers a palm line was drawn parallel to wrist line and the line was divided into four sections and a line drawn in the middle of each fingers and that line meets the palm line in section one, two, three and four then that will give index finger, middle finger, ring finger and little finger respectively. Hence, they used classification rule method to predict the output. They have used 1300 test data. The only limitation of this method was that they only used identical backgrounds. With this approach 96.69% accuracy was achieved.

In paper [5] they converted the input image into YCbCr format for skin segmentation. They also considered cases where face might also be present in the input so, after skin segmentation the face was removed with the help of ellipse method. Further the image was binarized with thresholding. On this binary image morphological transformations were performed to remove noise. The features such as palm line, palm line centre, and thumb were detected like they were detected in [4]. However, for thumb detection they had to use two angles 45 degree and 135-degree angle with palm centre as in this approach both hands were used for making digits. Finally, for digit classification they used valley points created between the fingers. Only 200 testing data were used and the accuracy achieved was 96%.

The paper which we preferred over others was [6] because in this research they considered inputs with all the three cases that are input with 2 hands, 2 hands and a face, and the third case with only one hand. In this approach skin was segmented with the help of YCbCr conversion as well. Edges were detected with the help of canny edge detector as it has two threshold values. And then the hand and face were localized with the help of Hough transformation. Face will definitely contain more white pixels than that offhand as hand has gaps in between fingers that are filled with black pixels. With the help of this difference hand was separated from the face. The hand was then vertically oriented with the help of ellipse method. The palm was masked by defining convex hull of skin pixels and fingers were extracted. The histogram of the fingers was plotted and the tip of each fingers were found. After that the distance between the tip of successive fingers were calculated and the length of each fingers were also measured and they devised an algorithm to classify recognized digit with the help of decision tree. This approach used 594 test data and achieved accuracy of 90.47%.

III. Proposed method

In our module, we randomly generate character set and ask the user to show a gesture that corresponds to that character. The user's gesture is taken and processed to determine if it represents the displayed character. If the gesture is correct, the CAPTCHA is solved, and the user is viewed as a human.

A random number is generated by the system and user has to replicate that number using his/her hand gesture, system will capture the image and the region of interest from the image is cropped. This cropped image is then converted into YCbCr format to segment the skin region (hand) from the image. The image is further smoothened with the help of gaussian blurring. To remove unwanted noise in the background by preserving the shape of hand, morphological opening transformations were performed. This pre-processed image is then fed to the train DenseNet121 model to predict the outcome. The predicted outcome of each input is appended to a list and the concatenated string of elements of list are then matched with the string of random generated number. If both these string matches then it confirms that user is human and not a bot so system will allow user to proceed further. So, basically it differentiates the robot and human and provides a good two layered security.

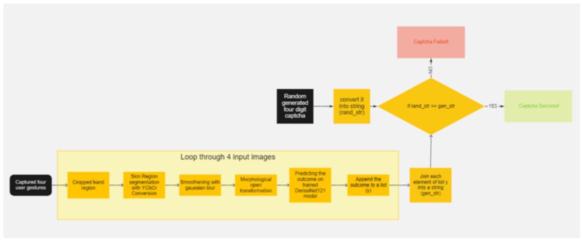

Captcha Failed!
NO
Random generated four digit captcha
convert it into string (rand_str)
if rand_str == gen_str
YES
Captcha Success!
Loop through 4 input images
Captured four user gestures
Cropped hand region
Skin Region segmentation with YCbCr Conversion
Smoothening with gaussian blur
Morphological open transformation
Predicting the outcome on trained DenseNet121 model
Append the outcome to a list (y)
Join each element of list y into a string (gen_str)

Figure 2 : Proposed method

IV. Experimental setup

We have experimented with a number of datasets publicly available. What we observed was that all of the datasets contained hand signal pictures with no background. At the moment we are still trying to implement hand gesture recognition with a background although the model is not very accurate, we plan to improve the accuracy in near future.

Following are the details of the dataset: -

- Image size: 100 x 100 pixels
- Colour space: RGB
- Number of classes: 10 (Digits: 0-9)
- Number of participants: 218
- Number of samples per participant: 10

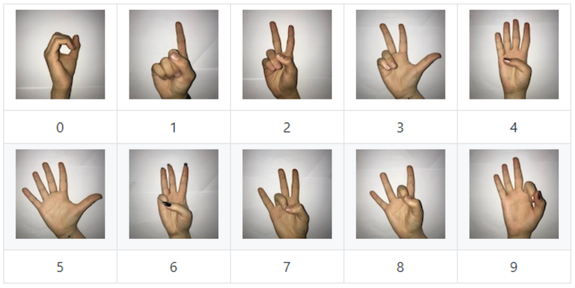


Figure 3 : Dataset Preview

We will use this data as the training dataset. We will then try by capturing the frames from a live video to test this data using real time data. For verification we have created our own images by converting live video of hand gestures into frames and labelling them to their respective number sign accordingly.

V. Results

Based on the papers reviewed regarding digit recognition based on hand gestures and some regarding captcha implementations using hand gesture concept. We followed [6] paper for processing images as the paper uses same kind of hand gestures as the one, we are using. We also tried converting image to HSV for skin detection as proposed in paper [4] but the results weren't as promising as the one with masked output of image converted to YCrCb which was proposed in [6]. A sample of HSV [8] and YcrCb [7] masked image is shown below:

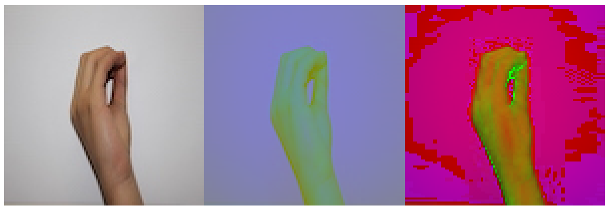


Figure 4 (a) Original Image, (b) YcrCb Image, (c) HSV Image

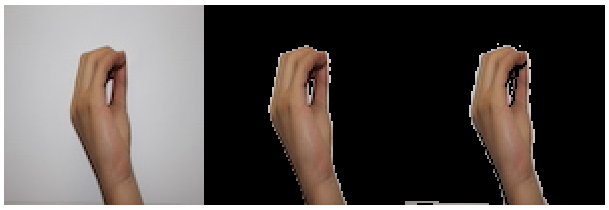


Figure 5 (a) Original Image, (b) YcrCb Image (c) HSV Image

As from the image it can be depicted that HSV masked image contains more noise than YcrCb masked image. However, we further smoothened both masked images with gaussian blur to reduce the noise. We can see that YCrCb masked image is more appealing.

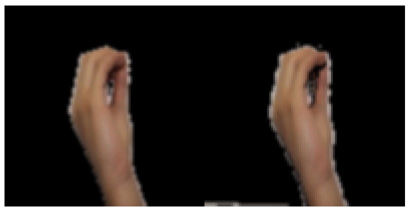


Figure 6 Gaussian Blurred images

(a) YcrCb Image (b) HSV Image

As there is still some noise in the image which was removed my morphological opening transformation as the input is a binary image.

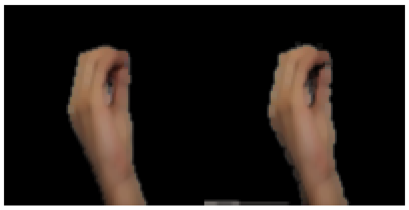


Figure 7 Morphological opening transformation

(a) YcrCb Image (b) HSV Image

Apart from applying image processing on some images of datasets we also applied these operations on our input images before applying these operations on whole dataset for training.

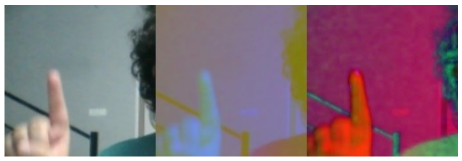


Figure 8 (a) Original Image (b) YcrCb Image (c) HSV Image

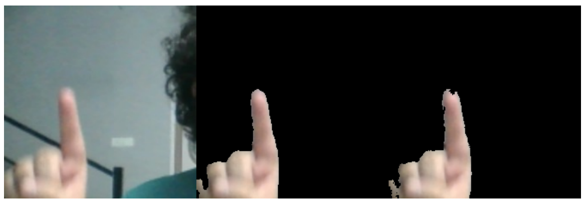


Figure 9 (a) Original Image (b) YcrCb (c) HSV

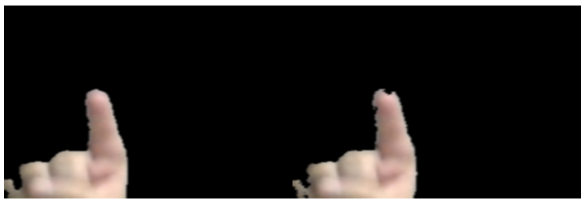


Figure 10 Gaussian Blurred images (a) YcrCb Image, (b) HSV Image

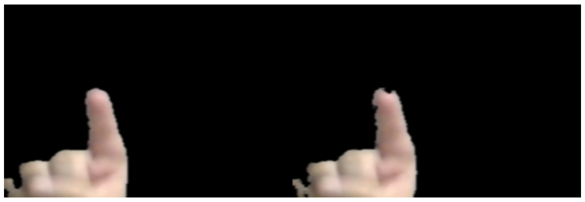


Figure 11 Morphological opening transform

Once we were satisfied then we processed whole dataset. The dataset was randomly split into 70% training data and 30% test data with the help of split folders library. We trained the dataset on two pretrained models ResNet50 and DenseNet121 as the dataset was small and these two have shown tremendous results even with small datasets.

However, we haven't used pretrained ImageNet weights as ImageNet is object centric and our dataset contains human hands which isn't an object thus, we trained the model with random weights. Unfortunately, Resnet50 model overfitted while Dense Net model have shown some outstanding results. Below are the results of the training and validation accuracies and the respective losses for 50 epochs. Approximately 98.7% of validation accuracy was achieved and validation loss dropped to 0.047.

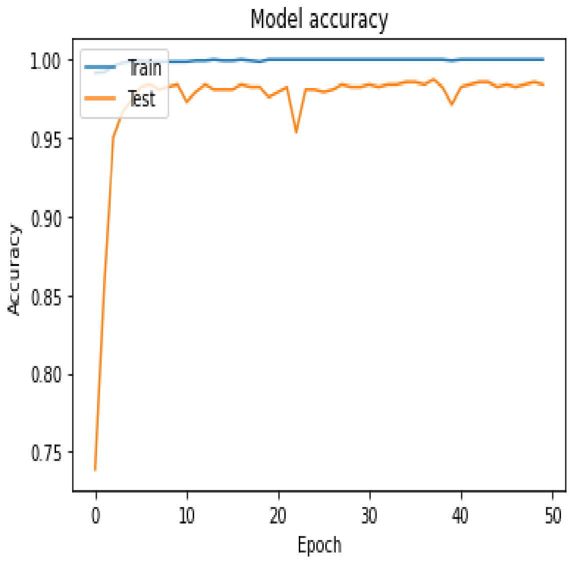


Figure 12: DenseNet121 Model Accuracy

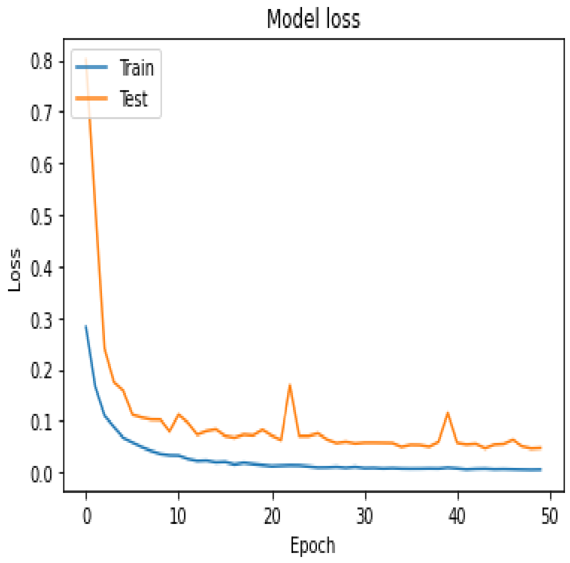


Figure 13: Model loss

For generation of captcha we used randint function of random library and generated 4-digit random number and user is asked to show the hand gesture for each number which were captured and saved one at a time. The saved images from the user inputs are further processed in the same fashion as the dataset and the output value of all the 4 images were predicted with the help of previous model and these outputs were stored in a list and all these values in the list are concatenated as a string with help of join function. Then this concatenated string is matched with the string form of captcha. If both the string matches it confirms that user is not a robot.

Show hand gesture for this captcha:

1467

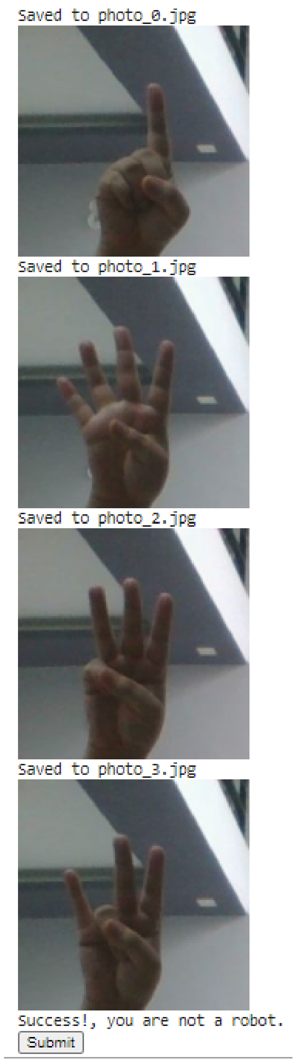
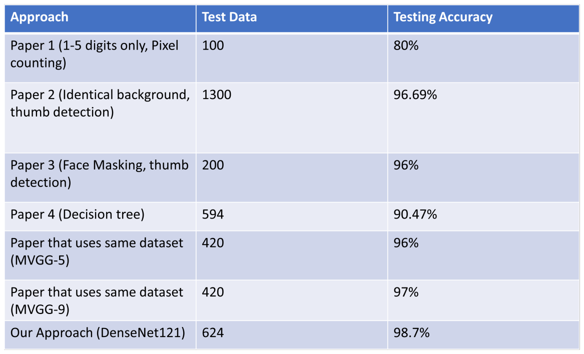

```
Saved to photo_0.jpg
Saved to photo_1.jpg
Saved to photo_2.jpg
Saved to photo_3.jpg
Success!, you are not a robot.
```

Figure 14 4 Input images from user

VI. Discussion

We decided to Sign Language Digits Dataset [9]. Which consists more than 2000 images and compared our results with other algorithms and methods. The first paper[3] used pixel matching of edges for classification which was the best approach for them as they only focused on 1-5 digits therefore the pixel in hand gestures edges will increase from 1 to 5 as there will be more fingers respectively. In our case we use all the digits from 0-9 and hand gestures of digit 3, 6, 7, 8, and 9 all will have 3 fingers so this approach will not work here. Another paper that we reviewed [5] used valley points created between the fingers for classification, this works for them because they used both the hands i.e. all ten fingers for digit classification. Both the other papers used datasets with same hand gestures as our dataset. One of the paper[4] used classification rule method for classification and achieved accuracy of 96.69% and the paper which we tried to replicate [6] used decision tree based classification and they used comparison based and Ellipse based methods for hand detection and from the 1396 hands discovered in the prior classification phase, the comparison-based approach detected 1262 hands. However, 1308 hands were found using the elliptical approach. The elliptical technique, based on these findings, is the superior choice for deployment.They have achieved 90.4 percent and 93.7 respectively. As there are so many pre-trained models available so we used few of those and couldn't achieve much accuracy with ResNet50 model but with DenseNet121 model trained with sgd optimizer we achieved 98.7% accuracy which is better than all the papers we reviewed.

We also compared results with a paper which uses CNN algorithm to train same dataset that we used [9]. The dataset was used to evaluate CNN methods, which are one of the most common image classification techniques and are used to categorise pictures with sign language. On the testing dataset, MVGG-5 had a 95 percent success rate, 96 percent with MVGG-9, and 97 percent with the structure after 15 training epochs. The two MVGG designs are the two best efficient methods. Where our method with DenseNet121 model, as mentioned earlier we reach up to 98.7 percent accuracy.



Approach	Test Data	Testing Accuracy
Paper 1 (1-5 digits only, Pixel counting)	100	80%
Paper 2 (Identical background, thumb detection)	1300	96.69%
Paper 3 (Face Masking, thumb detection)	200	96%
Paper 4 (Decision tree)	594	90.47%
Paper that uses same dataset (MVGG-5)	420	96%
Paper that uses same dataset (MVGG-9)	420	97%
Our Approach (DenseNet121)	624	98.7%

Figure 15 Comparison with reviewed papers

VII. Limitations

Some CAPTCHAs, however, do not satisfy the requirements for a successful CAPTCHA. A typical CAPTCHA asks the user to write letters or digits from a distorted picture on the screen. Recognizing images CAPTCHAs contain a slew of potential issues that have yet to be thoroughly investigated. It's tough for a tiny website to amass a huge picture dictionary, something an attacker doesn't have.

A pixel-based barrier that an intruder does not yet have exposure to and does not have a way of collecting additional fresh tagged pictures does not fulfil the criteria of a CAPTCHA.

Spambots may quickly defeat CAPTCHA using segmentation and vocabulary assaults. Converting the image to greyscale and removing the noise in the background Scanning activity files to get CAPTCHA phrases, as well as hacking activities such as the CAPTCHA-solving enterprise, which involves irritating other site users by sharing and requesting them to answer CAPTCHA. These are among the various ways that make the current CAPTCHA control scheme solvable by bots, allowing spammers and others to take advantage of the facilities.

This kind of CAPTCHA can only work when users device has a webcam and this also cannot work in dim or no lighting conditions.

VIII. Challenges

The most difficult aspect of our concept is separating the hand movements from the changing background. While we were able to obtain high training/testing accuracy with hand gestures with minimal or no background, we were unable to reach the same level of accuracy with hand gestures with diverse backgrounds.

The most difficult part was utilizing skin colour segmentation to separate the hand curves and borders. In addition, the majority of the datasets available for training the model are rather tiny. Also, skin colour segmentation was a challenging task for us as background and skin colour overlaps each other and there might be a chance of adverse effect on output. Skin colour segmentation is working well with exception of similar background.

We try to bind all functional quality in the oblong region with the minimum border to estimate the hand measurements. This area's measurements will be compared to those of a finger. We begin by establishing the skin pixels' feature vector; at least one edge of the vector space should be present in the most relevant rectangular region.

Another difficulty is quality of camera because most webcams have poor image quality, making sign number gesture identification more difficult. Similarly, in Google Collab, we struggled to start video capturing as it was new for us and even if we aren't conducting any computationally intensive image recognition operations, the entire graphics processing chain slowed and dragged along, and we were unable to handle more than one or two frames per second. However, we were able to make that work smoothly using the code available in colab snippet panel.

IX. Conclusion

To conclude, we can summarize by saying that we were able to make a working computer vision model that can detect hand sign number gestures. We can thus say that we have created a working model that can be used to verify human users from bots. Since we use numeric CAPTCHA to verify humans, it can be said that the system has a dual approach for verification. We first use the CAPTCHA so that only humans can read it and protect the system from OCR attacks. Followed by this, the user needs to make hand gestures according to the CAPTCHA code and only after the hand gestures are verified that the human passes the test.

The process is seamless and requires minimum input in comparison to other CAPTCHA systems like image CAPTCHAs where we need to give multiple inputs to pass the test.

Even though a major challenge we faced while developing the hand gesture detection model was that there was an issue segregating the hand gestures from dynamic backgrounds. We were still able to solve the issue considerably by using skin colour segmentation methods. We can try to improve the model for such a case but there are already very sophisticated models available that can perform the task with much higher precision. We can incorporate these open-source models to make our proposed project even more applicable in the field of human verification. Overall, we are presently happy with our results and hope that our proposed model can be applied in real applications.

X. References

[1] How CAPTCHAs work — What does CAPTCHA mean?, URL: www.cloudflare.com, 2021.

[2] Just what is the problem with CAPTCHA?, URL:www.canaxess.com.au, 2018.

[3] P. Panwar, Monika, P. Kumar and A. Sharma, ”CHGR: Captcha generation using Hand Gesture Recognition,” 2018 Conference on Information and Communication Technology (CICT), pp. 1-6, 2018.

[4] Zhi-hua Chen, Jung-Tae Kim, Jianning Liang, Jing Zhang, and Yu-Bo Yuan, “Real-Time Hand Gesture Recognition Using Finger Segmentation”, Hindawi Publishing Corporation,The Scientific World Journal, Volume 2014, Article ID 267872, 2014.

[5] Chidananda, H., Reddy, T. H., ”A natural approach to convey numerical digits using hand activity recognition based on hand shape features”, International Society for Optics and Photonics, In Second International Workshop on Pattern Recognition, Vol. 10443, p. 1044305, 2017.

[6] Ahmed BEN JMAA, Walid MAHDI, Yousra BEN JEMAA and Abdelhamid BEN HMADOU, “A new approach for digit recognition based on hand gesture analysis”, International Journal of Computer Science and Information Security, Volume 2, No. 1, 2009.

[7] HSV Skin color range, URL: https://stackoverflow.com/questions/8753833/exact-skin-color-hsv-range.

[8] Nalin Chhibber, January 26, 2018,” Skin Detection Using OpenCV Python”, URL: https://nalinc.github.io/blog/2018/skin-detection-python-opencv/.

[9] Mavi, A., (2020), “A New Dataset and Proposed Convolutional Neural Network Architecture for Classification of American Sign Language Digits”, arXiv:2011.08927[cs.CV],URL:https://github.com/ardamavi/S
Language-Digits-Dataset