

Facilitating Form Filling For Visually Impaired By Automatic Detection Of Captcha In Web Forms

Ms. Priyanka , Assistant Professor, Department of MCA, BBDIT, Ghaziabad, **Dr. Harleen Kaur**, Assistant Professor, FMIT, JAMIA HAMDARD, New Delhi, **Mr. Dileep Kumar**, Assistant Professor, Department of MCA, BBDIT, Ghaziabad.

Abstract:

The massive use of the Internet and WWW by the government and commercial organizations for automation of work has resulted in extensive proliferation of online computing. Almost every procedure be it related to railway reservations or other governmental or entertainment related work requires the user to filling an online form. To avoid the forms being filled by bots or malicious computers the form designers usually use a cryptic image known as CAPTCHA which is human readable and the form can be submitted only when the user has decrypted the captcha and filled in the text represented by that captcha. While this is a security feature and prevents the bombarding of servers by automatically filled forms, this also acts as an impediment for the visually impaired genuine users. It also slows down genuine employees who have to do data entry work for an organization. To overcome this handicap for the genuine users we propose a solution for automatic detection and text conversion of captcha. This solution uses image processing and optical character recognition to detect captcha and convert it to text.

Keywords: CAPTCHA, bypassing Captcha, detection of captcha, spammers, crowd-sourcing, optical character recognition.

Introduction:

CAPTCHA(Completely automated public Turing test to tell computer and human apart Captcha is widely used over the world wide web to prevent the automated system(like bot) to scrape a data from web sites.

A CAPTCHA is a program that can generate and tests the human can pass but computer programs cannot. A CAPTCHA is basically just an implementation of function where it is easy to take input and compute the result. A CAPTCHA can be thought of in a simple terms as “ARE YOU A HUMAN?” test.

Initially it is a images which has some words or letters, embedded in it.

It is a challenge response test used to ensure that the challenge is generated by a human not by a computer. Users are asked to read some distorted characters (or some special images) and type the string in order to

ensure that the user is human. Automation is real problem for web application, automate attacks can exploits public web services for several purposes- just like placing thousands of message on blogs, spam comments, forums, guest books and wikies, linking to forged sites for identity theft(Phishing), promoting a product, flooding a site with useless comment. As a newspaper printed out, 200 billion spams are sent every day. The spammers are using third parties to solve CAPTCHA called crowd sourcing.

Captcha is a challenge based response test used to ensure that the response is generated by a person not by computer. Users are asked to read and type a string of distorted characters in order to ensure that the user is human not a computer trying to access a website or account.

CAPTCHA provides us a secure web environment but sometimes it slow down the system where faster processing is more important than security. CAPTCHA is used where securities are more important, but we need to bypass the captcha where execution is more important. For example, a form being filled by a visually impaired person or an IRCTC agent performing many reservations, then we need to bypass the CAPTCHA.

Examples:



Fig. 1

In this paper we propose a technique for automatic detection of captcha and converting it to text.

Brief History:

Turing [1] in 1950 proposed a test for AI in which a computer must fool a panel of humans into believing the machine is human. Blum, Ahn, Langford(2000)[2] proposed a class of ATT(called human interactive proof HID), which Hopper(2001)[3] describes as a protocol “ that allows a human to prove something to a computer.

Hopper and Blum (2001)[4] propose a HIP called secure Human Identification Protocol or Humanoid, in which computer must verify a human membership in a group without requiring a password , biometric data, electronic key or any other physical evidence.

Blum,Ahn,Langford (2000)[5] propose a “Completely Automated Public Turing Test To Tell Humans and Computer Apart” in which the computer must be able to generate and grade “ A test that most human can pass.”

Example: EzGimpy-deter to spam attacks. First used by yahoo (2001).

Types of Captcha:

- Text based-Gimpy, Gimpy-r, Google captcha.
- Graphics based- Bongo Pix.
- Audio based.

Applications of Captcha:

- Site registration
- Comments
- Online polls and
- Passwords

2. REALTED WORK

CAPTCHA was first introduced in 1997 when Andrei Broder devised the CAPTCHA method. In the same year, Altavista web site used this method to distinguish between computer programs and human user. In this method, a distorted English word was shown to the user and the user was asked to type it



(Fig. 1).

Distortion was needed so that OCR programs could not recognize the word As mention above CAPTCHA can be classified in two ways OCR based and non OCR based.

2.1 OCR-Based CAPTCHA Methods

In OCR-based methods, the image of a word with distortion and various pictorial effects is shown to the user and he/she asked to type that word. Due to presence of various pictorial effects, the computer will encounter

problems in the recognition of these words and only a human user can recognize the word. But these methods usually result in dissatisfaction of users. On the other hand, efforts have been made for attacking these methods. Examples of these methods include Gimpy, Handwritten CAPTCHA, and Persian/Arabic CAPTCHA.

2.2. Non-OCR-Based CAPTCHA Methods

In contrast, we can point to Non-OCR-based methods which are easier to work with than OCR-based ones. Examples of these methods include PIX, Text-to-Speech method, and Drawing CAPTCHA. Various methods have been proposed for breaking up the CAPTCHA. It can be done with the help of the segmentation and character recognition techniques. This can be done with the help of the Neural Network using divide and conquer approach. To provide defense against such techniques resistance method for CAPTCHA have also been proposed that includes Font tricks, Noise, Color model and Overlap.

Problem Statement:

The aim of this paper is to propose a method for automatic detection of the text in a captcha image for facilitation of work.

Proposed Solution for Detecting Captcha:

For this purpose I will use image processing operations to convert captcha images into simpler ones which can easily be converted to text by OCR software. The steps involved in the proposed methods are as follows:

Step 1: Determine the average color intensity of the image.

Step 2: Mark all pixel as white or black. This is done to remove any background noise in captcha.

Step 3: In step two there may have been generated some gaps so we eliminate horizontal gaps.

Step 4: Eliminate vertical gaps.

Step 5: It might be that steps 3 & 4 lead to inadvertent removal of some valid portion of captcha so we attempt to fill the region between two consecutive black regions in the image produced after step 4.

Step 6: Repeat the step 5.

Step 7: There may be false positives produced in step 6 (by filling in regions) which need to be cleared so we clear any false positives.

Step 8: The image produced in step 7 has characters distinctly visible but not aligned. So we find and align the characters.

Step 9: Feed this image produced in step 8 with aligned characters to optical character recognition software.

Step 10: Write the output of OCR software to a text file.

So we see that using simple image filtering loops based on observation of how a human being would approach the image and some existing OCR software the CAPTCHA has been detected and converted to text.

The steps are illustrated in fig. 2.

Algorithms:

Algorithm for automated Captcha:

```
String automatedcaptcha(inputimage)
{
String str;
Alignedimage=preprocess(inputimage);
Str=converttotext(Alignedimage);
return str;
}
```

Algorithm for Preprocessing:

```
Image preprocess(inputimage)
{
image BWimage,Reimage,
Fillimage1,Fillimage2,Prunedimage,Aligned
image;

int avgintensity;

avgintensity=determine_avg_intensity(input
image);

BWimage=mark_White_Black(avgintensity,
inputimage);//remove noise

Reimage=reconstruct(BWimage);

Fillimage1=fillregion(Reimage);

Fillimage2=fillregion(Fillimage1);

Prunedimage=removefalseposition(Fillimag
e2);

Alignedimage=alignedcharacter(Prunedima
ge);
```

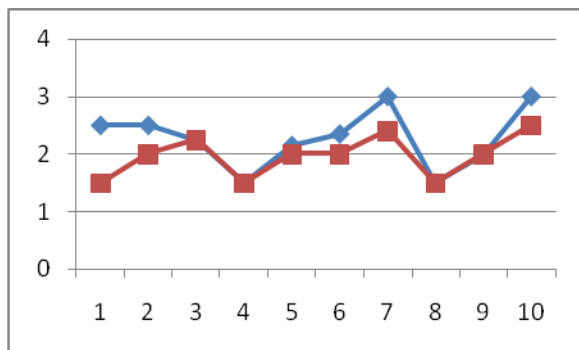
```
return Alignedimage;}
```

Algorithm for Extracting Character Using OCR:

```
String converttotext(image alignedimage)
{
String str;
Str=OCRModule(alignedimage);
}
```

Results:

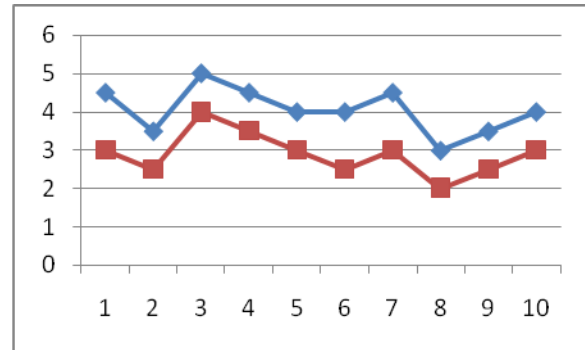
Following graphs illustrate the gain in speed obtained by using above software for form filling. Typical email account registration form is considered in both the cases and the experiment is repeated for 50 users using general usb based internet connection. The relative speeds for both the cases : not using this software and using this software are depicted.



For Non- Visually Impaired Users

Red: With Software

Blue: Without Software



For Visually Impaired Users

Red: With Software

Blue: Without Software

Future Scope:

Researchers at University of California, Berkeley, have developed computer programs that can automatically solve simple Captchas with 83 percent accuracy. Researchers are also in the process of developing audio Captchas that would make Captchas independent of visual perception. The current focus seems to be on human-generated Captchas that present a distorted figure of an animal at an aberrant angle. These types of picture Captchas are exceptionally difficult to solve with software.

Captcha development continues on both the offensive (programs and systems that defeat Captchas) and defensive (improved Captchas) sides. Both sides are using advanced image processing as well as dictionary-style attacks. This sort of “arms race” between researchers seeking more secure Captchas and the hackers, spoofer, and spammers trying to defeat the Captchas is likely to continue for some time.

Conclusion:

From the use of software for automatically detecting captcha and converting them to text we found a considerable gain in speed

obtained by both the visually impaired and non-visually impaired users. This gain was around 25% to 40% in terms of form completion time.

Acknowledgement:

This research paper is made possible through the help and support from everyone, Including : Teachers, family, friends, and in essence, all sentient beings.

Especially, please allow me to dedicate my acknowledgment of gratitude toward the following significant advisors and contributors:

First and foremost, I would like to thank Dr. Harleen Kaur for his most support and encouragement. He kindly read my paper and offered invaluable detailed advices on grammar, organization, and the theme of the paper.

Second, I would like to thank as well as all the other faculty members who have taught me about image processing and captcha over the past three years of my pursuit of the master degree.

Finally, I sincerely thank to my friends, who provide the advice. The product of this research paper would not be possible without all of them.

References:

- [1] [Alan Turing](#) in his 1950 paper "[Computing Machinery and Intelligence](#)," which opens with the words: "I propose to consider the question, 'Can machines think?'"
- [2] Blum, M., Ahn, L. von, & Langford, J. (2000). The CAPTCHA Web site. Retrieved November 11, 2005 from <http://www.captcha.net>.
- [3] Ahn, L. von, Blum, M., Hopper, N. J., & Langford, J. (2003). *CAPTCHA: Using hard AI problems for security*. Retrieved October 10, 2005 from

<http://www.captcha.net>

[4] N. J. Hopper and M. Blum, "Secure human identification protocols," in Proc., ASIACRYPT '01, LNCS 224, pp. 52–66, 2001.

[5] M. Blum, L. A. von Ahn, J. Langford, and N. Hopper, "The CAPTCHA Project: Completely Automatic Public Turing test to tell Computers and Humans Apart." <http://www.captcha.net>, November 2000.

[6] Luis von Ahn, Manuel Blum, Nicholas J. Hopper and John Langford. TheCAPTCHA Web Page: <http://www.captcha.net>. 2000.

[7] Luis von Ahn, Manuel Blum and John Langford. Telling Humans and Computers Apart (Automatically) or How Lazy Cryptographers do AI. To appear in Commu-nications of the ACM.

[8] Mihir Bellare, Russell Impagliazzo and Moni Naor. Does Parallel Repetition Lower the Error in Computationally Sound Protocols? In 38th IEEE Symposium on Foundations of Computer Science (FOCS' 97),

[9] A. L. Coates, H. S. Baird, and R. J. Fateman. Pessimial Print: A Reverse Turing Test. In Proceedings of the International Conference on Document Analysis and Recognition (ICDAR' 01), pages 1154-1159. Seattle WA, 2001.

[10] <http://www.lafdc.com/captcha>

[11] K. Chellapilla, P. Simard. Using Machine Learning to Break Visual Human Interaction Proofs (HIPs)

[12] Y. Lecun, L. Bottou, Y. Benschio, P. Haffber. Gradient-Based Learning Applied to Document Recognition. Proc. of the IEEE, November 1999.

[13] Mike O'Neill. Neural network for Recognition of Handwritten Digits.

[14] L. Bottou, Y. Benschio, Y. Lecun. Global Training of Document Processing Systems using Graph Transformer Networks.

- [15] Y. Lecun, L. Bottou, Y. Benschio. Reading checks with multilayer graph transformer network.
- [16] P. Simard, D. Steinkraus, J. Platt. Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis.
- [17] C. Burges, O. Matan, Y. LeCun, J. Denker, L. Jackel, C. Stenard, C. Nohl, J. Ben. Shortest Path Segmentation: a method for training a Neural Network to Recognize Character Strings.
- [18] D. You, G. Kim. An approach for locating segmentation points of handwritten digit strings using a neural network.
- [19] G. Mori, J. Malik. Recognizing Objects in Adversarial Clutter: Breaking a Visual Captcha.

IJERT

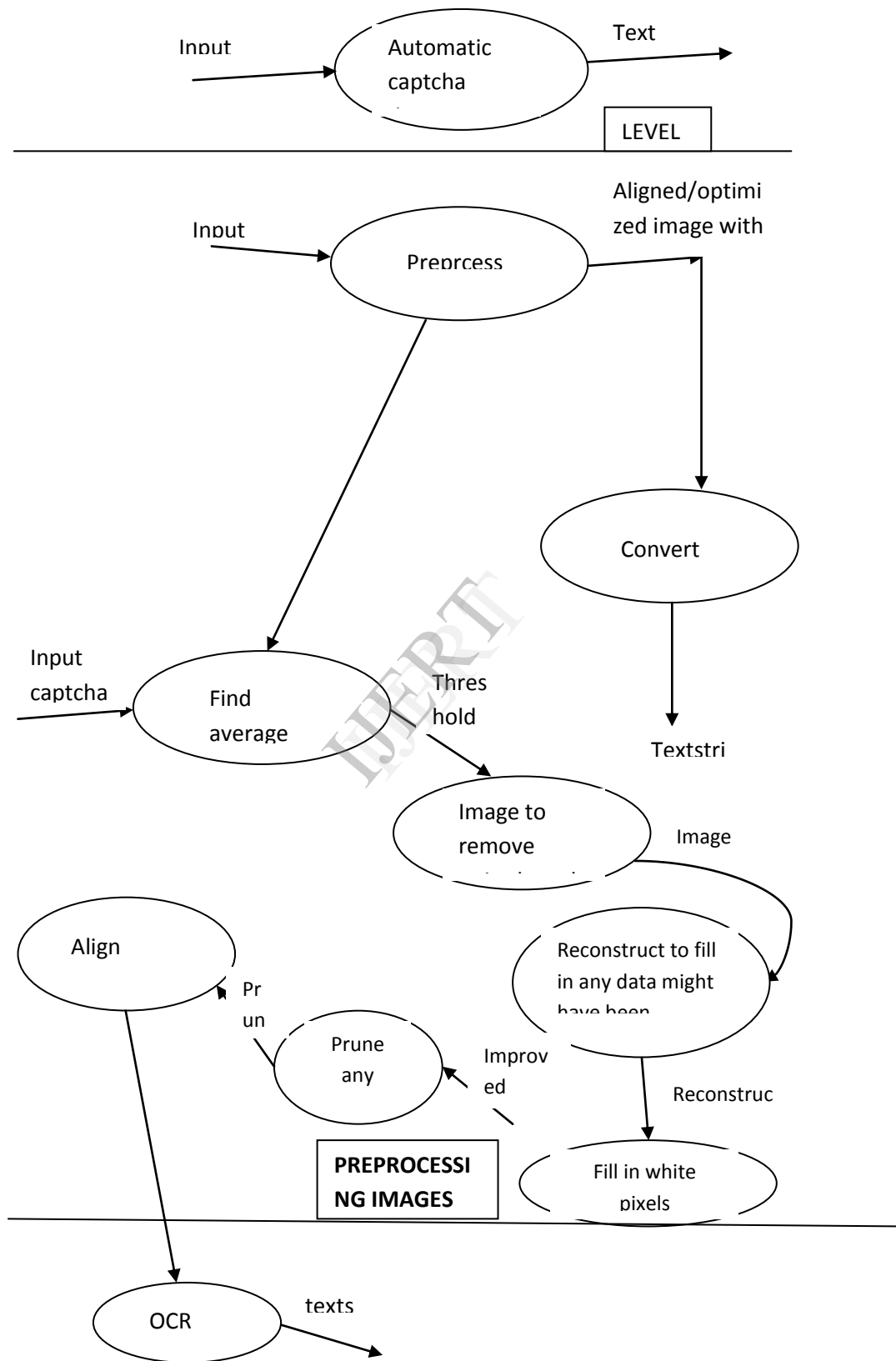


Fig. 2