Vol.2013-CSEC-63 No.1
2013/12/9

インターネットにおける CAPTCHA の強度検証

小室進一郎† 木下俊之†

現在，WEB 上の CAPTCHA は，画像ノイズにより読み取りにくくした文字認識によるものが主に用いられている．そこで我々はこの文字認識による CAPTCHA を破るプログラムを開発し，その画像ノイズのレベルを変化させて認証強度や認証特性を調べる．

Intensity verification of CAPTCH in the internet

SHINICHIRO KOMURO† TOSHIYUKI KINOSHITA†

Now, a character recognition type CAPTCHA with some image noise is mainly used on Web system. In this work, we develop a program which breaks this character recognition type CAPTCHA, and investigate its authentication intensity and characteristics with changing level of the image noise.

1. はじめに

CAPTCHA とは，インターネットのサービスを利用するユーザが，人であるか悪意のあるプログラム（BOT）であるかを判別するための，チャレンジレスポンス型テストの一種である（図 1）．これは例えば，BOT が無料メールアカウントを大量に登録したり，ブログ等にコメントを大量に投稿したりすることを防ぐために用いられる．現在は，主に画像認識方式の CAPTCHA が利用されている，これは表示された画像に対して，点や線などのノイズを加えたものを表示し，人には正確に解答できても BOT は解答できないことにより両者を識別する方法である[1]．しかし，近年これらを破る技術が進み，ノイズ方式の CAPTCHA を突破する BOT が多くの研究者から指摘されている[2].

図 1 CAPTCHA 認証の例

2. BOT の概要

BOT は WEB サイトの，CAPTCHA の解答を入力するところに到達すると，CAPTCHA を破るプログラムに画像を送ってこれを，読み取らせ，突破する仕組みになっている（図 2）.

例として「Windows Live」a)サービスのアカウントを不正取得する BOT は, CAPTCHA を破ることに成功する確率は平均で 3 回に 1 回程度であり, 繰り返し実行して, CAPTCHA を突破している[3].

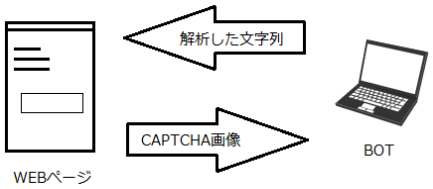


図 2 BOT の例

3. 関連研究

文字認証による CAPTCHA が突破されつつあることもあって，文字の利用から画像を利用した CAPTCHA へ移行する動きがある. 代表的なもは, Asirra[4]や CATCHA[5], がある Asirra や CATCHA では, 動物の画像を複数用意し, その中から特定の動物をユーザが選択するというチャレンジレスポンス型のテストを試みている．これは人には動物の絵を理解できるが，BOT には難しいことを利用した CAPTCHA である（図 3）.

† 東京工科大学大学院
Tokyo University of Technology

a) Microsoft Live は，米国 Microsoft Corporation の米国およびその他の国における登録商標です.

Vol.2013-CSEC-63 No.1
2013/12/9

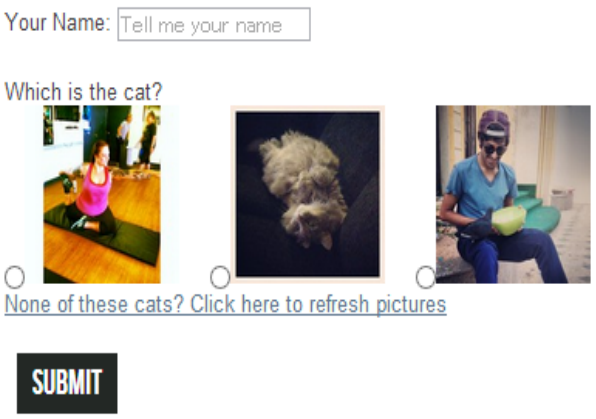


図 3　CHTCHA

4. 研究課題

本研究はプログラムにより文字認証 CAPTCHA の突破をはかり，突破率からこの CAPTCHA の強度検証を行う．本研究を行う上で Visual Studio C# 2010 Express b)及び OpenCV を用いてプログラムを作成した．現在 WEB 上で使われている文字認証方式の CAPTCHA をもとに，プログラムにより画像を取り込み（図 4 の①），取り込んだ CAPTCHA 画像にプログラムによって作成したフィルタをかけ，これによりプログラムに，より読み取りやすい CAPTCHA 画像を生成することが可能である．

5. フィルタの特徴

本研究ではノイズのかかった文字認識 CAPTCHA の画像のノイズを除去するフィルタを作成する．このフィルタをかける CAPTCHA に対して画像データを操作しやすくするために，取り込んだ画像に対して二値化を行う．二値化される画像は，ユーザにより輝度値を設定し，輝度値がユーザの設定した値よりも大きければ白，小さければ黒に二値化される（図 4 の②）．

ノイズ除去は，ユーザが画素数の個数のしきい値を設定し，画像中の一画素ごとに，周囲 8 マス（3×3 の場合）の中に黒い画素がしきい値以上であれば黒に，しきい値より少なければ白に書き換える（図 5）．このフィルタの範囲はユーザにより 3×3，5×5 に設定できる．またこれらのノイズ除去処理を全て行ったものを図 4 の③に示す．

これらのユーザによる設定を組み合わせることで CAPTCHA 画像のからノイズ除去を行える．ノイズを除去された画像は，OpenCV で作成したプログラムにより解析され画像の中に含まれる文字を検出する．

b) Microsoft および Visual Studio は，米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です．

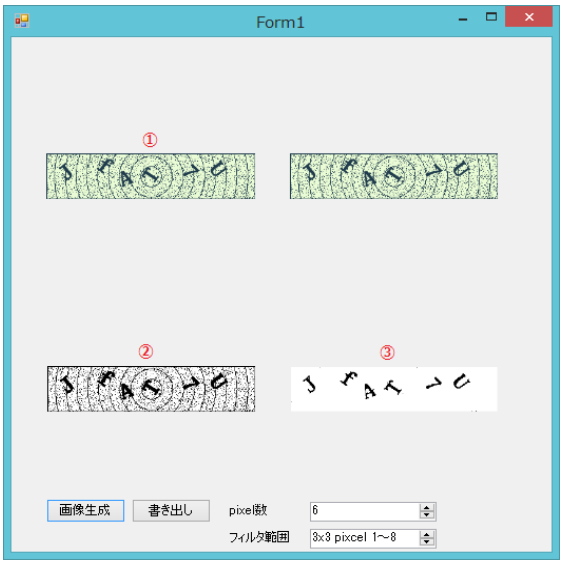
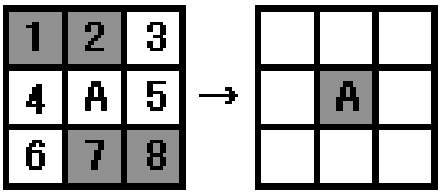
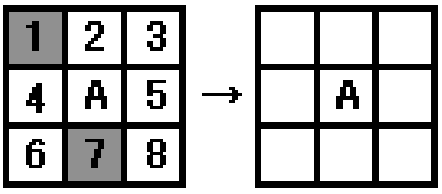


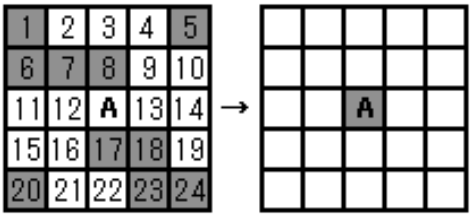
図 4　プログラムの動作



(a)　A が黒に設定なる場合



(b)　A が白に設定される場合



(c)　A が黒に設定される場合

図 5　A に注目した際のノイズ除去

6. 一致率の検証

6.1 OpenCV を用いた形状マッチング

フィルタによる効果の検証として，各文字アルファベットにノイズを 0%，25%，50%，75%付与したものを用意する．それらに対しノイズを付与する前の画像とノイズを付与した後の画像で形状マッチングを行い，一致率を調べる．これらはノイズ除去のフィルタをかけた画像とかける前で一致率に変化がどの程度見られるかを調べるためである．

形状マッチングは，テンプレート画像を入力画像の中から 1 ドット毎に調べ，最も高い一致率のスコアと場所を示すものになっている．またノイズを付与した文字を図 6 に示し，形状マッチングの例を図 7 に示す．またノイズを付与した文字の一致率の一部を表 1 に示す．

図 7 では左側のウィンドウの中から，テンプレート画像に一番形状がマッチしている部分を赤枠で示している．これらのプログラムを利用し，CAPTCHA 画像の中から特定させたい文字を形状マッチングさせ解析する．それにより得られた情報から CAPTCHA に含まれる文字を判別し，CAPTCHA を突破する．

図 6　各ノイズ付与

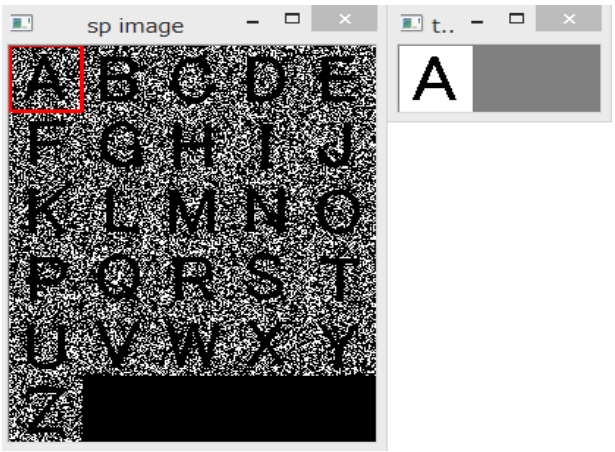


図 7　形状マッチング例

表 1 の結果から画像の中に付与されるノイズが高いほど一致率は下がる結果が得られた．またノイズを一切付与しない場合の数値は 1.0 で完全一致となる．

6.2 フィルタの検証

表 1 の結果から，ノイズを加えた文字はノイズを 25%加えたものと 75%加えたものでは一致率に 2 倍ほど差が出る結果がでた．そこで次に本研究で作成したフィルタを使い一致率に変化が出るかをフィルタサイズ毎に検証した．フィルタサイズ 3×3 の検証結果の一部を表 2 に示しフィルタサイズ 5×5 の検証結果の一部を表 2 に示す．またフィルタをかける前とかけた後で，最も一致率が上がった文字を図 8 に示す．

図 8　差がでたアルファベットの比較

表 1　各アルファベットの一致率

	ノイズ 25%	ノイズ 50%	ノイズ 75%
A	0.567	0.415	0.267
B	0.640	0.480	0.314
C	0.562	0.416	0.276
D	0.608	0.455	0.303
E	0.586	0.430	0.281

表 2　フィルタを用いた検証結果(3×3)

	ノイズ 25%	ノイズ 50%	ノイズ 75%
A	0.646	0.556	0.448
B	0.717	0.619	0.477
C	0.654	0.492	0.462
D	0.698	0.557	0.492
E	0.700	0.515	0.468

表 3　フィルタを用いた検証結果(5×5)

一致率	ノイズ 25%	ノイズ 50%	ノイズ 75%
A	0.558	0.438	0.405
B	0.615	0.505	0.423
C	0.546	0.341	0.380
D	0.594	0.466	0.442
E	0.552	0.403	0.420

表 2 及び表 3 の結果から，本研究のプログラムはフィルタをかける前よりも一致率を上げることができ，CAPTCHA の突破率を上げられるものと考えられる．特にノイズが 50%ほど付与されているものに関しては，多くの文字で一致率を 0.1 以上あげることに成功している．しかしながらフィルタの範囲を広げるとノイズ自体が消えにくくなるほか，文字自体が崩れてしまう欠点が多く見受けられた．また例外として”D”の一部で，フィルタをかける前よりもフィルタをかけた後で一致率が多少低下した．またノイズが 75%付与されている画像ではフィルタの範囲を 5×5 にした場合，ノイズが 50%付与されているものよりも一致率が高くなる場合がある．これはノイズを多く除去しようとした結果，文字自体に一部欠落が出てしまったためと思われる．文字が欠落している例を図 9 に示す．図 9 の中央の文字の上部で欠落が起きているのが確認できる．

図 9　フィルタによる文字の欠落

6.3 類似画像の検証

文字の中にはCとGやOとQといった，類似性のあるものが存在する，これらにフィルタをかけたとき，文字が崩れて誤ったアルファベットを検出しないかを検証し，その結果を図 10 に示す.

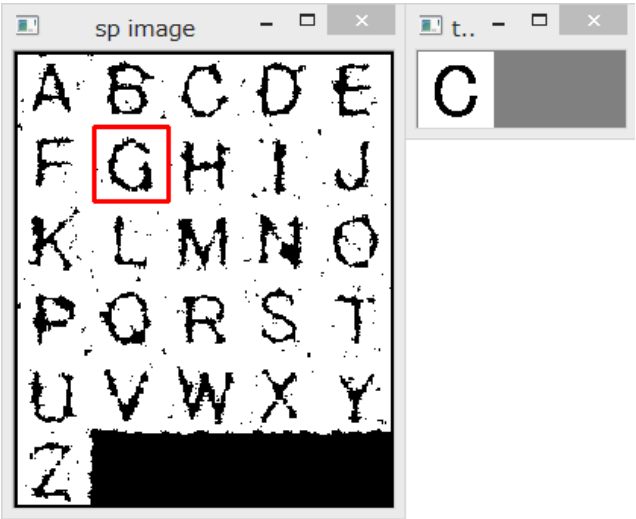


図 10 アルファベットの誤検出

図 10 の結果から，フィルタによって文字が崩れていてしまい，特定の文字を認識できなくなる場合がある．誤検出を防ぐ方法として，誤検出しやすい文字は，いくつかの検索用の画像を用意し最も一致率が高かったものを結果としてとる方法が考えられる.

6.4 実際の CAPTCHA への文字検出

実際に使用されているCAPTCHAに対して本プログラムを使用することでどの程度ノイズが除去でき，文字が検出されるかを確認する．今回使用するCAPTCHA画像はvBulletin[6]で使われるCAPTCHA画像を使用して検証している（図 11）．取り込んだCAPTCHA画像を読み取りやすく変換したものを図 12 に，読み取れた文字を図 13 に示し，読み取れなかったものを図 14 に示す.

図 11 vBullentin の CAPTCHA 画像

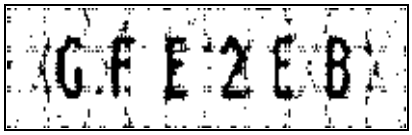


図 12 読み取りやすく変換された CAPTCHA 画像

(a) B の読み取り

(b) E の読み取り

図 13 読み取れた文字

(a) F の誤検出

(b) G の誤検出

図 14 読み取れなかった文字

図 13, 図 14 の結果から CAPTCHA に含まれる 6 文字のうち 4 文字が検出できず，実際に使用される CAPTCHA への精度は高いとは言えない結果になってしまった．考えられる原因として，文字のサイズが考えられる．また，CAPTCHA 画像の中に E という文字が 2 つあるが，本研究のプログラムは一致率が最大なものを返すため，1 つしか検出されなかったと考えられる.

7. 考察

7.1 フィルタについて

表 1，表 2，表 3 の結果から本研究で作成したプログラムのフィルタは文字の一致率を上げることに成功している，しかしながらフィルタをかけることで逆に一致率が低下する場合がある．図 9 で示した元の CAPTCHA 画像の欠落が原因と考えられる．これは検出したい文字の周りにノイズが無いのに，画素のしきい値を高く設定した時にCAPTCHA画像中の文字が元のサイズより縮小してしまうからである．これを避けるにはフィルタを 2 回通す方法などが考えられる.

7.2 文字検出について

OpenCV において作成したプログラムでの文字検出は，形状マッチングを用いたため文字サイズの異なる CAPTCHA に対しては，あまり精度が良くなかった，そこで CAPTCHA 画像に対してラベリングを行い，そこから文字の外接矩形調べ，そこで受け取った数値に合わせ，検索用の文字を作成する．これにより文字サイズの違いによる誤検出が防げるものと考えられる．

8. おわりに

本稿では文字認識による CAPTCHA を突破するプログラムを作成することで，WEB 上で使用されている CAPTCHA の突破を試みた．しかし現段階では文字の読み取りの精度が芳しくなく，プログラムの見直しが必要だと感じている．今後プログラムに変更を加える上で，CAPTCHA 画像に使われている文字の種類，サイズなどを解析する必要があると思われる．また同一の文字が使われていた時の検出方法なども加える必要がある．上記の内容を実装することで，本研究のプログラムの精度を上げ，CAPTCHA の認証制度を検証し，良い結果をだせるよう努力していこうと思う．

謝辞 インターネットにおける CAPTCHA の認証強度の作成にご協力頂いた皆様に，謹んで感謝の意を表する．

参考文献

[1] The Official CAPTCHA site,
http://www.captcha.net/
[2] Caca labs,
http://caca.zoy.org/wiki.PWNtcha
[3] IT pro:Windows Live Mail アカウント取得時の CAPTCHA を大量処理,
http://itpro.nikkeibp.co.jp/article/COLUMN/20080226/294681/
[4] Asirra
http://research.microsft.com/en-us/redmond/projects/asirra/
[5] CATCHA
http://syddev.com/catcha/
[6] vBulletin
http://www.vbulletin.com/
[7] 山本匠，西垣正勝，J.D.Tygar:機械翻訳の違和感を用いた CAPTCHA の提案，情報処理学会研究報告.CSEC，コンピュータセキュリティ, Vol37, pp.1-8(2009).
[8] schima.hatenablog.com
http://schima.hatenablog.com/entry/20091201/1259600665