情報保護學會論文誌
第19卷 第6號, 2009. 12

색상 정보를 이용한 문자 기반 CAPTCHA의 무력화*

김 성 호,[1†] 양 대 헌,[1‡] 이 경 희[2]
[1]인하대학교 정보보호연구실, [2]수원대학교

Breaking character-based CAPTCHA using color information*

SungHo Kim,[1†] DaeHun Nyang,[1‡] KyungHee Lee[2]
[1]Information Security Research Laboratory, INHA University,
[2]The University of Suwon

요 약

오늘날 캡차(CAPTCHA)는 계정 생성, 광고, 스팸 메일 등 자동화된 소프트웨어 대리자에 의한 다양한 공격을 방어하는데 널리 사용되고 있다. 초기 캡차의 문자들은 왜곡이 심하지 않아서 사용자들은 캡차 문자들을 쉽게 인식할 수 있었다. 이런 이유 때문에 이미지 처리, 인공지능 등의 다양한 기술들을 사용하여 많은 캡차들이 쉽게 무력화 되었다. 이에 대한 대안으로 캡차에 노이즈를 추가하거나 캡차 문자를 왜곡함으로써 문자 기반의 캡차 공격을 어렵게 만들었지만 캡차에 노이즈를 추가하거나 캡차 문자를 왜곡하는 것은 사용자들이 캡차 문자를 읽는 것을 더 어렵게 만들었다. 캡차의 가독성을 보완하기 위하여 몇몇 캡차들은 서로 다른 색상을 가진 문자를 사용했다. 그러나 서로 다른 문자의 색상을 사용하는 것은 캡차를 공격하기 원하는 공격자에게 이점을 제공했다. 이 논문에서는 색상을 기초로 문자열 캡차의 인식 성공률을 높일 수 있는 방법을 제안한다.

ABSTRACT

Nowadays, completely automated public turing tests to tell computers and humans apart(CAPTCHAs) are widely used to prevent various attacks by automated software agents such as creating accounts, advertising, sending spam mails, and so on. In early CAPTCHAs, the characters were simply distorted, so that users could easily recognize the characters. From that reason, using various techniques such as image processing, artificial intelligence, etc., one could easily break many CAPTCHAs, either. As an alternative, By adding noise to CAPTCHAs and distorting the characters in CAPTCHAs, it made the attacks to CAPTCHA more difficult. Naturally, it also made users more difficult to read the characters in CAPTCHAs. To improve the readability of CAPTCHAs, some CAPTCHAs used different colors for the characters. However, the usage of the different colors gives advantages to the adversary who wants to break CAPTCHAs. In this paper, we suggest a method of increasing the recognition ratio of CAPTCHAs based on colors.

Keywords: CAPTCHA, Support Vector Machine, RGB Color

I. 서 론

캡차(CAPTCHA, Completely Automated Public Turing test to tell Computers and Humans Apart)는 컴퓨터와 사람을 구별 할 수 있게 해주는 튜링 테스트의 일종으로써, 주로 컴퓨터 소프트웨어에 의한 자동화된 공격을 차단할 때에 사용된다. 현재 캡차는 웹 서비스의 자동 계정 생성, 카페 자동 가입 신청, 블로그 댓글의 스팸광고, 투표 봇, 게시판 글의 자동 등록 등을 차단하는데 유용하게 쓰이고 있다[1].

접수일(2009년 7월 1일), 수정일(2009년 9월 19일),
게재확정일(2009년 10월 30일)
* 이 논문은 인하대학교의 지원에 의하여 연구되었음.
† 주저자, night12th@isrl.kr
‡ 교신저자, nyang@inha.ac.kr

그러나 초기 문자 기반 캡차의 약점들이 드러나면서 Yahoo의 EZ-gimpy CAPTCHA[2], Windows Live Hotmail CAPTCHA[3], Windows MSN CAPTCHA[4], Google의 Gmail CAPTCHA[5] 등이 주로 공격을 당하게 되었다. 이에 사용된 주된 공격 방법은 캡차의 문자열을 하나씩 분리해 내어, 분리된 문자를 여러 가지 분류 알고리즘들을 이용하여 캡차 문자를 인식하는 것이었다. 이에 대한 방어책으로 캡차의 문자들을 연결시키고 문자의 형태를 왜곡하거나 노이즈를 추가하여 캡차 문자의 분리를 어렵게 만들었다. 그 결과로 공격 성공률은 낮아졌지만 캡차 문자의 가독성도 크게 떨어졌다. 그 대안으로 여러 가지 이미지 기반의 캡차가 만들어 지기도 하였고[6], 또는 캡차 문자의 색상을 다르게 하여 가독성을 보완하기도 하였다.

그러나 그 중 캡차 문자의 색상을 다르게 하는 방법은 공격자에게 이점을 제공한다. 캡차의 문자들이 서로 다른 색상 값을 가지고 있을 경우 문자의 대표 색상을 기준으로 하여 캡차 문자의 분리를 쉽게 할 수 있다. 물론 이런 캡차들은 기존의 방법들을 사용하여 어느 정도의 공격이 가능하다. 이 논문에서는 그런 약점을 노출하고 있는 캡차들 중, 문자가 색상으로 충분히 구분되는 캡차들에 한하여, 추가적인 문자열 분리 방법을 적용하여 문자열 캡차의 인식 성공률을 높일 수 있는 방법을 제시할 것이다.

II. 공격대상의 특징

이 논문에서는 특별한 특성을 가지는 캡차에 한하여 사용될 방법을 서술한다. 그 특성은 다음과 같이 한정한다.

- 문자 기반의 영문자 및 숫자를 포함한 캡차이다.
- 캡차의 문자가 각각 색상으로 구별 된다.
- 노이즈와 문자열을 서로 분리 해 낼 수 있다.

캡차의 한 문자에 대해 동일 색상 계열의 그라데이션(gradation) 효과를 적용시켰다면, 색상 보정 처리 후 그 문자가 하나로 분리 되게 하는데 어려움이 없겠으나, 다른 색상 계열로 그라데이션 효과를 적용시켰다면 문자열 분리 과정 시에 한 문자가 2개 이상으로 분리 될 것이라고 예상된다. 그래서 이 논문에서는 한 문자가 동일 색상 계열로 구성되어 있는 것으로 한정한다.

2.1. 기존 방법의 문제점

공격 대상으로 정한 캡차들은 기존 방법으로도 쉽게 공격 할 수 있다.

[그림 1(a)]의 경우, 배경 노이즈를 제거하고, 제거된 배경색인 하얀색을 기준으로 하거나 색상 히스토그램을 이용하면 [그림 1(b)]처럼 문자들을 분리 할 수 있다. 그러나 [그림 1(c)]와 같이 수직 투영이 어려운 경우가 발생 할 수 있다. [그림 1(c)]의 경우 문자 '7' 과 'Z' 사이를 강제로 나눌 수도 있으나, 그럴 경우 한쪽 문자가 잘려 나가므로 문자를 정확히 나눌 수 없는 경우가 발생한다.

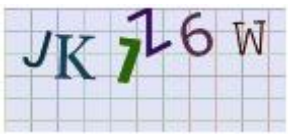
수직 투영 방법이 어려운 [그림 2(a)]의 경우 검정 픽셀의 연결성을 이용하면 정확이 나눌 수 있다. [그림 1]처럼 배경 노이즈를 제거한 후 배경을 흰색, 문

(a) 원본 이미지

(b) 수직 투영 적용 성공

(c) 수직 투영 적용 실패

[그림1] 수직 투영 기술의 적용 예



(a) 원본 이미지

(b) 검정 픽셀의 연결성을 이용하여 분리 성공

[그림 2] 검정 픽셀의 연결성을 이용한 분리 예

(a) 원본 이미지

(b) 수직 투영 및 검정 픽셀의 연결성을 이용할 경우 분리에 성공하는 예(JZ)와 실패하는 예(XJ, RK)

그림 3. 수직 투영 및 검정 픽셀의 연결성을 이용한 예

자를 검정색 처리를 한다. 그 후 검정색 픽셀로 연결된 픽셀들을 모으는 방법으로 나누면 캡차 문자를 정확이 나눌 수 있다[7].

그러나 [그림 3(a)]와 같이 문자가 완전히 붙어 있는 경우, 수직 투영 및 검정 픽셀의 연결성을 모두 이용해도 두 문자('X'와 'J')를 정확하게 나누기 힘들다. [그림 3(a)]처럼 문자 X와 J가 붙어 있는 경우, 두 문자의 폭을 기준으로 강제로 분리할 수도 있으나, 정상정인 글자 이미지로 분리하기 어려워지고, 그 결과로 분류 성공률이 낮아질 것이다.

[그림 1]과 [그림 2]를 해결한 방법(수직 투영 및 색상 히스토그램 및 검정 픽셀의 연결성을 이용한 분리)으로도 어느 정도의 공격 성공률을 보장 할 수는 있으나, [그림 3]과 같은 유형이 많이 나타난다면 높은 인식 성공률을 보장 할 수 없다. 정확하게 나누는 것을 목표로 하는 이유는 추후에 사용될 분류 알고리즘의 분류 성공률을 높이기 위해서이다. 그래서 [그림 3]과 같은 유형을 해결하기 위하여 추가적인 분리 방법을 생각하게 되었다.

III. 색상에 기초한 문자 분리

[그림 1], [그림 2], [그림 3]과 같은 캡차는 색상으로 캡차 문자들을 구별 할 수 있다는 큰 특징을 가지고 있다. 문자가 색상으로 구별된다는 의미는 각 글자마다 유사한 RGB 값들을 포함하고 있다는 의미이다. 이 특징을 이용하여 유사한 RGB 색상을 가지고 있는 픽셀들을 서로 연결하여 글자들을 분리해 낸다는 것이 주된 방법이다. [그림2]와 같은 경우에는 이미지에 대하여 이진 역상 처리(기준으로 정한 색상 임계값 이상이면 흰색, 이하이면 검정색으로 이미지 색상을 변경하는 것)를 하면 배경은 흰색, 글씨는 검정색의 이진 형태를 가지기 때문에 검정색 픽셀을 연결하여 글자를 나누는 것이 매우 용이하다. 색상에 기초하여 의미는 나누는 방법도 이와 유사하나 RGB의 경우에는 픽셀 당 Red, Green, Blue 총 3가지의 속성을 가지고 있기 때문에 조금 더 확장된 방법이 필요하다.

3.1 색상을 이용한 캡차 문자열 인식 과정

이 논문에서 대상으로 한 캡차 문자열 인식의 진행 및 처리 과정은 다음과 같다.

1단계) 캡차 이미지의 전처리 과정 처리(노이즈 및 배경색 제거)

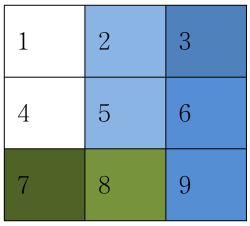
1) 캡차 이미지에 이진 역상 처리를 한다. 처리 후 글자의 경우 검정색이 되고 배경은 흰색이 된다.

2) 처리된 이미지와 원본 이미지를 이용하여 문자의 색상을 복원한다. 이진 역상 처리된 이미지와 원본 이미지의 크기는 항상 같다. 그리고 또한 캡차 내 문자열의 위치도 항상 같다. 이 점을 이용하여 문자의 색상을 복원한다. 이진 역상 처리된 이미지에서 검정색 픽셀의 좌표를 (a,b)라고 하면, 원본 이미지의 좌표(a,b)에는 글자의 색상 값이 존재한다. 이 값을 서로 치환하거나 원본 이미지에서의 값을 이진 역상 처리된 이미지의 좌표(a,b)에 할당 한다. 이 과정을 이진 역상 처리된 이미지의 모든 검정색 픽셀에 적용시키면 이진 임계값 처리된 이미지는 색상 값을 가진 캡차 문자열을 얻을 수 있다.

2단계) 색상 보정 처리 과정

1) 이미지의 모든 열에 대하여 대표 값을 찾는다. 이 값을 열의 대표 색상 값이라고 정한다. 대표 색상 값은 해당열의 모든 색상 값들 중 픽셀의 빨강, 초록, 파랑 값의 합이 가장 작은 것으로 정한다. 그리고 색의 유사도 측정이 기준이 될 임계값을 정한다.

2) 전처리 과정을 통하여 얻어진 이미지에서 색상 값이 (255,255,255)가 아닌 좌표를 순차적으로 찾는다. 색상 값 (255,255,255)는 흰색 픽셀을 나타내고, 이 값의 위치는 전처리 과정을 통과여 얻어진 이미지의 바탕을 나타낸다. 색상 값이 (255,255,255)가 아닌 좌표는 전처리 과정을 통하여 얻어진 이미지 캡차의 색상 문자열 위치를 나타낸다. 검색된 좌표가 흰색 픽셀이 아닐 경우 다음과 같은 처리를 진행한다. 다음과 같은 처리를 전처리 과정을 통하여 얻어진 이미지의 모든 픽셀에 적용 시킨다



[그림 4] 좌표(a,b)를 둘러싼 픽셀의 색상 예

2-1) 해당 좌표를 (a,b)라고 할 때 좌표를 둘러싼 픽셀의 이미지 값들을 가져온다.

2-2) 좌표(a,b)를 둘러 싼 픽셀과 열의 대표 값 총 9개의 색상 값들 중 좌표(a,b)의 색상 값을 서로 비교하여 색의 농도가 가장 진하면서 유사한 색상을 가지는 값을 찾아 좌표(a,b)의 픽셀 값에 할당한다. 유사도는 현재 좌표의 픽셀의 색상 요소 값과 비교 대상이 되는 색상 요소 값의 차를 구한 다음 절대 값을 구한다. 그 후에 요소 중 하나 이상이 임계값을 초과하면 유사도가 없다고 판단한다. 예를 들어, 임계값은 60이라고 가정할 때, [그림 4]의 5번 픽셀 색상 값과 7번의 픽셀 색상 값의 차의 절대 값은 (63,81,192) 이다. 여기서 임계값을 넘는 요소가 2개 이상 발생했으므로 유사도가 없다고 판정한다. [그림 4]의 3번의 픽셀 값과 5번의 픽셀 값의 차의 절대 값은 (59,51,39)이다. 여기서 임계값을 넘는 요소가 존재하지 않으므로 유사 색상이라고 판단한다. 진한 색상 값 찾기는 정의된 유사도에 따라 진한 색상은 유사한 색상의 값들 중 빨강, 초록, 파랑 요소의 합을 구한 후 그 합이 가장 작은 값을 진한 색상이라고 판단한다.

3단계) 문자 분리 과정

1) 색상 보정 처리 과정을 마친 이미지에 대하여 색상 값이 (255,255,255)가 아닌 픽셀을 찾는다. 즉, 캡차 문자열이 존재하는 픽셀을 찾으면 다음과 같은 처리를 한다. 다음과 같은 처리는 모든 픽셀에 적용시킨다.

1-1) 해당 좌표를 (a,b)라고 할 때 좌표를 둘러싼 픽셀의 이미지 값들을 가져온다. 그중 유사 색상이라고 판단되고 가장 진한 색을 가지고 있는 좌표를 (a′,b′)를 선택한다.

1-2) 선택한 좌표 (a′,b′)를 기준으로 색상의 유사도가 같다고 판단되는 좌표들을 서로 연결하여 좌표들의 모음을 구한다. 좌표들을 연결 할 때 색상의 유사도가 다르면 끊어진 것으로 간주하고 좌표들의 모음을 모두 구했으면 선택된 좌표들의 색상 값을 모두 (255,255,255)로 할당한다.

2) 캡차 문자열의 길이의 가정에 따라 다음과 같이 처리 한다.

2-1) 고정 길이 문자열 캡차 라고 가정 할 경우 : 색상의 유사도 측정 시 사용한 임계값을 줄여가면서 해당 고정 길이의 숫자만큼 문자가 분리될 때 까지 문자 분리 과정의 1) 과정을 반복하여 진행한다. 문자열 길이와 분리 이미지의 수가 동일하면 분리 성공으로 간주하고 다음 단계로 넘어가고 그렇지 않으면 실패로 간주한다. 여기서 임계값의 최솟값은 0으로 한다.

2-2) 비 고정 길이의 문자열 캡차 라고 가정 할 경우 : 문자 분리 과정의 1) 처리를 1회만 진행하고 다음 과정으로 넘어간다.

4단계) 분류 알고리즘을 통한 문자열 인식

위의 3단계)을 통하여 얻어진 분리된 이미지를 분류 알고리즘에 적용시킨다. 분류 알고리즘을 통하여 인식된 문자들을 획득하여 최종적으로 캡차의 문자열을 추측한다.

3.2 색상 보정 처리의 필요성 및 임계값에 대한 고찰

한 문자를 구성하고 있는 모든 픽셀의 색상 값 중 동일 요소(빨강, 초록, 파랑)의 최댓값과 최솟값의 차이를 색상 폭이라고 정의한다고 하면, 한 문자를 구성하고 있는 색상 픽셀의 색상 값 범위의 폭은 글자마다 서로 다양하다. 글자 획의 폭이 넓고 색이 뚜렷하게 나타 날 경우 해당 글자를 구성하는 모든 픽셀의 색상 값의 최댓값과 최솟값의 차이가 적게 나는 반면, 글자 획의 폭이 좁고 글자가 흐릿할 경우에는 해당 글자를 구성하는 모든 픽셀의 색상 값의 최댓값과 최솟값의 차이가 크게 발생한다.

[그림 5]에서 문자 'X' 의 경우 색상 값의 폭이 보정 전에는 (61,49,50)이였으나, 보정 후 (48,41,41)로 범위의 폭이 좁아진 것을 확인 할 수 있다. 이와 같이 보정처리를 하면 색상의 유사도 판단을 위한 임계값의 크기를 좁게 할 수 있어 색상의 연결성으로 문자를 쉽게 분리 해 낼 수 있다.

[그림 5] 보정 전과 보정 후 비교사진

색상 픽셀들을 서로 연결하여 캡차 문자를 분리 할 때 가장 중요한 점은 문자 픽셀의 색상 값의 범위를 한정하는 것이다. 색상의 유사도 판단을 위하여 임계값을 설정하는데 이 임계값을 너무 작게 잡으면 색상 픽셀의 연결성이 약해져서 문자가 끊어진다. 그렇다고 임계값을 너무 크게 잡으면, 색상 픽셀의 연결성이 강해져서 각각의 문자를 분리해 내지 못한다. 하나의 문자를 구성하고 있는 색상 픽셀 값의 범위를 좁게 할 수 있다면 한다면, 캡차의 문자 분리 성공률을 높일 수 있다. [표 2]와 [표 3]에서 확인 할 수 있듯이 색상 보정 처리를 하면 캡차 문자를 구성하고 있는 픽셀의 색상 값의 범위의 폭을 줄일 수 있다.

이 문서에서 이진 역상처리에 요구되는 임계값과 색의 유사도 판단의 기준이 되는 임계값이 필요하다. 이 값들의 경우 정형화된 계산 방법이 존재하지 않는다. 그러나 캡차 문자열이 고정된 길이라면 캡차 문자열의 길이만큼 문자가 분리 될 때 까지 임계값을 조절하면서(임계값을 단계적으로 하향하여 적용) 캡차 문자열을 분리 하여 인식 성공률을 높일 수 있다. 그러나 임의의 길이 캡차 문자열일 경우 임계값을 조절하기 어렵다. 이와 같은 경우에는 캡차 문자열의 분리 과정을 다수 실행 한 후 그 결과를 바탕으로 최적의 임계값을 지정하여야 한다.

[표 2] [그림 5]의 문자 X 의 색상 값의 변화

문자 X	색상 보정 전	색상 보정 후	범위 변화
Red	93 ~ 154	93 ~ 141	61 → 48
Green	42 ~ 91	42 ~ 51	49 → 41
Blue	90 ~ 140	90 ~ 131	50 → 41

표 3. [그림 5]의 숫자 5 의 색상 값의 변화

숫자 5	색상 보정 전	색상 보정 후	범위 변화
Red	38 ~ 59	38 ~ 56	21 → 18
Green	8 ~ 13	8 ~ 12	5 → 4
Blue	46 ~ 60	46 ~ 57	14 → 11

IV. 실험 및 성능 분석

이 논문에서는 기존 방법에 대하여 실증적인 검증을 위해 다음과 같이 실험 프로그램을 만들었다.

- 이미지 처리: OpenCv, OpenCsharp
- 분류 알고리즘(SVM 사용): libSVM, SVM.NET
- 메인 프로그램 제작 시 사용한 언어: C#
- 테스트에 사용한 컴퓨터: intel Pentium D 3GH, RAM 2G

이미지 처리를 위해 대표적인 이미지 처리 라이브러리인 OpenCV를 사용하였다. 메인 프로그램을 C#으로 제작하였고, C#에서 OpenCV 라이브러리를 사용하기 위해 OpenCsharp을 추가로 사용하였다. 분류 알고리즘은 SVM(support vector machine)을 사용하였다[8]. SVM의 모델을 생성하기 위해서 네이버 계정 생성 페이지에서 사용되는 캡차 이미지 약 1500개, 네이버 카페 가입 신청 페이지에서 사용되는 캡차 약 1700개 정도를 수집하였고, 수집한 캡차 이미지를 바탕으로 SVM 모델 파일을 각각 생성하였다.

4.1 네이버 계정 생성 페이지에서 사용된 캡차에 대한 시뮬레이션 결과

네이버 계정 생성 페이지에서 사용된 캡차의 특징은 [그림 6]과 같다.

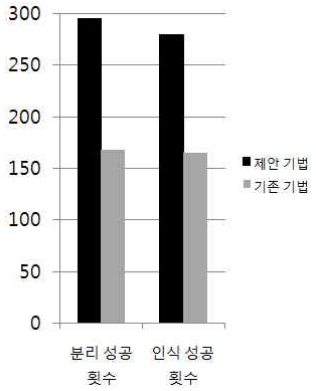
- 캡차 문자열의 길이가 6인 이미지가 반복적으로 발생하였다.
- 문자열의 왜곡은 심하지 않으나 문자의 회전은 발생한다.
- 배경 및 노이즈와 문자열을 분리 해 낼 수 있다.

[그림 6] 네이버 회원 가입 시 사용된 캡차 이미지의 예

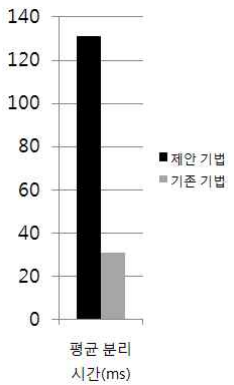
즉 색상을 기준으로 배경 및 노이즈를 제거하는 데 문제가 발생하지 않았다.

네이버 계정 생성 페이지에서 사용된 캡차의 경우 길이가 6인 캡차 문자열이 반복적으로 발생하여, 캡차 이미지 문자열 길이가 고정이라고 가정하고 테스트를 실행하였다. 비교 대상은 색상에 기초 하지 않는 검정 픽셀 연결성을 이용한 캡차 문자열 분리 방법을 사용하였다. 이 테스트에서는 최적화 되지 않은 검정 픽셀 연결 분리 방법을 사용하였다. 그리고 테스트에 사용된 캡차 이미지 사이즈의 평균은 대략 160×60 픽셀이었다. 테스트 캡차 이미지는 SVM 모델을 제작할 때 사용 되지 않는 캡차 이미지 300개를 사용하였다.

제안 기법을 기초로 한 분리 방법으로는 총 300개의 이미지 중 295개가 성공하였고, 기존 기법의 경우 168개가 성공 하였다. 캡차의 문자열이 고정이라고 가정 하였는데도 상당이 높은 분리 성공률이 나타났다. 색상 분리 성공 개수에서 비 색상 분리 개수를 뺀 수 127은 캡차 이미지에서 문자열이 붙어있는 것이 존재하는 이미지 개수이다. 분류 성공 횟수는 SVM을 사용하여 분류에 성공한 횟수, 즉 캡차 문자열 인식 성공 개수를 의미한다. 제안 기법을 사용한 분류의 경우 295에서 15가 줄어든 280이 나왔고, 기존 기법의 경우 3회가 줄어든 165가 나왔다.



(a) 분리 및 인식 성공



(b) 문자열 분리 성공 시에 소요된 평균 분리 시간

[그림 7] 네이버 회원가입 캡차에 대한 테스트 결과

평균 분리 시간은 캡차 이미지에서 문자를 분리해 낸 시간만 계산하였다. [그림 7]에서 보다시피, 제안 기법은 평균 131ms, 기존 기법은 31ms가 걸렸다. 제안 기법의 경우 색상 값들에 대한 연산이 색상 보정 및 색상 연결을 이용한 분리 과정에서 많이 발생하여 시간이 길어졌고, 기존 기법의 경우 전처리 과정 및 분리 과정의 연산이 제안 기법 보다 적어서 빠른 시간에 처리 할 수 있었다.

최종적으로 색상에 기초로 한 분류 성공률 즉 캡차 문자열 인식 성공률은 93%였고, 기존 기법의 경우 55%였다. 두 방법의 성공률 차이는 38%이나, 기존 기법을 최적화 한다면 두 성공률의 차이는 더 좁아질 것이다. 그러나 기존 기법의 경우 최적화를 해도 나눌 수 없는 이미지 들이 존재한다. 이미지가 랜덤으로 나오기 때문에 비 색상으로 처리할 수 없는 이미지의 발생 빈도를 가늠하기 어렵지만, 이 부분을 고려하지 않는다면 성공률을 높일 수는 없을 것이다.

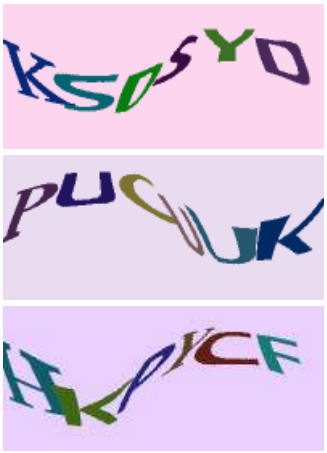
색상에 기초한 방법에서 분리 실패보다 분류 실패 횟수가 더 많이 발생하였다. 이와 같은 결과는 SVM 모델 데이터를 생성할 때 사용한 캡차 이미지의 개수가 적어서 발생하였다고 생각된다. 그리고 색상 보정 실패 및 불안전한 색상 분리 결과로 인하여 발생한 부분도 있다고 생각된다.

4.2 네이버 카페 가입 신청 페이지에서 사용된 캡차에 대한 시뮬레이션 결과

네이버 카페 가입 페이지에서 사용된 캡차의 특징은 다음과 같다.

- 캡차 문자열의 길이가 6인 이미지가 반복적으로 발생하였다.
- 문자의 왜곡 및 회전이 상당히 존재하였으나 노이즈는 존재하지 않았다.
- 대부분의 캡차 문자열이 서로 붙어 있었으나 색상으로 구분 되어 있었다.

네이버 카페 가입 신청 페이지에서 사용된 캡차의 경우도 길이가 6인 캡차 문자열이 반복적으로 발생하



[그림 8] 네이버 카페 가입 신청 시에 사용된 캡차의 예

여 캡차 이미지의 문자열 길이가 고정이라고 가정하고 테스트를 실행하였다. 비교 대상은 색상에 기초 하지 않는 검정 픽셀 연결성을 이용한 캡차 문자열 분리 방법을 사용하였다. 이 테스트에서도 최적화 되지 않은 검정 픽셀 연결 분리 방법을 사용하였다. 그리고 테스트에 사용된 캡차 이미지 사이즈의 평균은 대략 200×90 이었다. 테스트에 사용된 캡차 이미지는 SVM 모델을 제작할 때 사용 되지 않는 캡차 이미지 300개를 사용하였다.

테스트에 사용된 300개의 캡차 이미지에서 문자열이 모두 연결되어 있어 검정 픽셀 연결 분리 방법을 적용 한 결과 모두 실패하였고 그로 인하여 캡차 문자열을 분리하는데 걸린 평균 분리 시간을 비교 할 수 없었다. 색상 정보를 이용한 경우, 분리 성공이 292, 분류 성공이 270, 즉 캡차 문자열 인식 성공률이 90%였다. 캡차의 문자열이 고정이라고 가정 하였는데도 상당이 높은 분리 성공률이 나타났다. 네이버 카페 가입 신청 페이지에서 사용된 캡차에 대하여 비 색상 방법을 좀 더 최적화 한다면 성공 횟수는 증가할 것으로 예상되나 위와 같은 캡차의 속성을 가지고 있다면 그 상승폭은 매우 작을 것이라고 예상된다.

V. 결 론

이 논문에서는 텍스트 기반의 캡차 중 특정한 대상을 한정하여 추가적 캡차 문자열 분류 방법인 색상에 기초한 캡차 문자 분리 방법을 제시하였다. 기존의 방법에 추가하여 성공률을 높이는 방법으로, 기존 방법은 성능이 떨어진다는 것을 나타내지는 않는다. 그러나 특정 성질을 띠는 캡차에 대해서 높은 공격 성공률을 보인다. 캡차 이미지의 특성을 이용하여 캡차 문자열 분리 성공률 및 인식 성공률을 높이는 것이 주된 목표이다.

보안할 사항으로는 첫째 붙어있는 문자를 분리하였을 경우 잘려나간 문자 이미지를 복구하는 것이다. 이것을 해결한다면 분류 알고리즘의 성공률 향상될 것으로 기대한다. 둘째는 색상 보정을 안정적으로 또는 정확하게 하는 것이다. 이 처리를 최적화 한다면 문자열 분리 실패를 줄이고 처리 속도를 줄일 수 있을 것이다.

테스트에 사용된 캡차에 대한 인식 성공률이 높게 나왔다. 캡차에 대한 취약점 때문에 웹 서비스에 대한 공격이 바로 일어나긴 힘들어 보인다. 캡차는 보안성을 강화한 것들 중 하나의 방법이기 때문이다. 그러나 보안 요소의 한 부분인 캡차가 공격이 취약하므로 캡차에 대한 추가적인 보안성 강화가 필요할 것이다.

참 고 문 헌

[1] Carnegie Mellon University, “The Official CAPTCHA Site,” http://www.captcha.net/

[2] G. Mori and J. Malik, “Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA,” Proc. of the Computer Vision and Pattern Recognition (CVPR) Conference, IEEE Computer Society, pp. 134-141, June 2003.

[3] S. Prasad, “Microsoft Live Hotmail Under Attack by Streamlined Anti-CAPTCHA and Mass-mailing Operations,” http://securitylabs.websense.com/content/Blogs/3063.asp, 2008.

[4] J. Yan and A.S. El Ahmad, “A Low-cost Attack on a Microsoft CAPTCHA,” Proceedings of the 15th ACM Conference of Computer and Communications Security, pp. 543-554, Oct. 2008.

[5] S. Prasad, “Google’s CAPTCHA busted in recent spammer tactics,” http://securitylabs .websense.com/content/Blogs/2919.aspx, 2008.

[6] 강전일, 맹영재, 김군순, 양대헌, 이경희, “복수의 이미지를 합성하여 사용하는 이미지 기반의 캡차와 이를 위한 안전한 운용 방법,” 정보보호학회논문지, 18(4), pp. 153-165, 2008년 8월.

[7] J.B. Fiot and R. Paucher, “The Captchacker Project Home Page,” http://code.google.com/p/captchacker

[8] S.R. Gunn, “Support Vector Machines for Classification and Regression,” Image Speech & Intelligent Systems Group, May 1998.

<著者紹介>

김 성 호 (Sung Ho Kim) 학생 회원
2009년 2월: 인하대학교 컴퓨터 공학과 졸업
2009년 9월~현재: 인하대학교 정보통신공학과 석사 과정
<관심분야> 시스템 보안, 네트워크 보안

양 대 헌 (Dae Hun Nyang) 정회원
1994년 2월: 한국과학기술원 과학기술 대학 전기 및 전자 공학과 졸업
1996년 2월: 연세대학교 컴퓨터 과학과 석사
2000년 8월: 연세대학교 컴퓨터 과학과 박사
2000년 9월 ~ 2003년 2월: 한국전자통신연구원 정보보호연구본부 선임연구원
2003년 2월 ~ 현재: 인하대학교 정보통신대학원 조교수
<관심분야> 암호 이론, 암호 프로토콜, 인증 프로토콜, 무선 인터넷 보안

이 경 희 (Kyung Hee Lee) 정회원
1993년 2월: 연세대학교 컴퓨터과학과 학사
1998년 8월: 연세대학교 컴퓨터과학과 석사
2004년 2월: 연세대학교 컴퓨터과학과 박사
1993년 1월 ~ 1996년 5월: LG소프트(주) 연구원
2000년 12월 ~ 2005년 2월: 한국전자통신연구원 선임연구원
2005년 3월 ~ 현재: 수원대학교 조교수
<관심분야> 바이오인식, 정보보호, 컴퓨터비전, 인공지능, 패턴인식