

# 추출된 부분이미지의 방향결정을 이용한 효율적인 CAPTCHA 의 설계

## Design of Efficient CAPTCHA System By Determining the Orientation of Cropped Sub-images

정우근

Chung Woo-Keun

부산대학교 컴퓨터공학과

wkchung@pusan.ac.kr

### ABSTRACT

Completely Automated Public Turing test to tell Computers and Humans Apart ) 은 스팸이나 로봇에 의한 사이트 자동가입, 계정 생성방지 도구로써, 인간의 우수한 가독성을 통해 특정 언어 또는 그림을 해독할 수 있는 특성을 이용한 것으로 일반적으로 컴퓨터 프로그램이 해독하기 어려운 기호, 글자 등을 재입력하도록 하여 스팸을 위한 자동화 도구 등을 무력화 시키는 보안 기술이다. 기존에 사용되어왔던 CAPTCHA 시스템은 텍스트 기반 시스템이다. 하지만 기존에 사용되었던 텍스트 기반의 CAPTCHA 방식은 웹봇 또는 이미지프로세싱, 머신 러닝 또는 문자 인식 알고리즘을 통하여 쉽게 통과할 수 있는 것으로 알려졌다. 본 논문에서는 기존에 사용되었던 Text 기반의 CAPTCHA 시스템과는 달리 이미지 기반의 새로운 CAPTCHA 시스템을 제안한다. 본 논문에서 제안하는 이미지 기반의 CAPTCHA 시스템은 일반적으로 촬영한 사진에서 서브 이미지를 추출하여, 서브 이미지를 0, 90, 180, 270도로 4가지의 방향중 임의적으로 회전을 통하여 사용자에게 교정을 요하는 CAPTCHA 시스템이다. 본 논문에서 서브 이미지를 이용한 CAPTCHA 시스템을 제안한다.

KEYWORDS CAPTCHA, Sub-Image

## 1 서론

인터넷의 도래로 많은 웹 페이지 이용 및 생성 및 서비스가 제공되고 있으며 많은 사용자들이 웹 페이지 이용 및 웹 페이지에서 제공되는 서비스를 이용하고 있는 실정이다. 많은 서비스가 제공되고 있는 만큼 현재 인터넷의 사용자는 또한 급속도로 증가하고 있다. 또한 사용자들이 급속도로 증가하고 있는 것과 비례하여 많은 홈페이지들의 생성이 기하급수 적으로 증가하고 있다. 인터넷 사용자들은 홈페이지에 존재하는 게시판을 이용하여 게시판 이용자들 간의 정보 교환 및 서로의 의견을 내놓기도 하고 또한 토론을 하기도 한다. 하지만 사용자들은 웹봇을 이용하여 게시판에 가공적인 사용자 등록을 통한 무분별한 스팸 정보 기입이 빈번하게 일어나는 것을 볼 수가 있다.

이러한 무분별한 정보기입을 막기 위해 사용자 등록 또는 게시판에 정보를 기입할 경우, 웹봇과 실제 사용자를 식별하기 위하여 CAPTCHA 시스템을 도입하였었다. CAPTCHA( Completely Automated Public Turing test to tell Computers and Humans Apart ) 은 이러한 홈페이지의 무분별한 자동가입, 계정 생성방지 도구로써, 실사용자를 구분 지을수 있는 방법이었다.

또한 HIP( Human Interaction Prrof ) [2] 방식또한 CAPTCHA 시스템과 동일한 시스템이다. 기존에 존재하는 CAPTCHA, HIP 방식은 텍스트 기반의 시스템이었다. 하지만 텍스트 기반의 CAPTCHA, HIP 시스템들은 머신러닝 [6]을 통하여 쉽게 통과될 수 있는 것으로 알려졌다 [6]. 기존에 존재하는 텍스트기반 CAPTCHA 시스템은 그림 1과 같다. 그림 1에 a) 와 같은 시스템은 너무 명확한 문자로 CAPTCHA 시스템이 제공되어서



그림 1. 가장 일반적인 글자를 이용한 CAPTCHA 시스템. a) 와 같은 경우 너무 명확한 문자로 제공되어서 이미지 프로세싱 또는 문자인식을 통하여 쉽게 통과 할 수 있다. b)와 같은 경우 이미지 프로세싱이 통과하기엔 어려운 점이 있으나, Projection-based segmentation algorithm 을 사용하면 문자열을 덮고 있는 선들을 제거할 수가 있다. c)와 같은 경우 사용자마저도 통과 하기 힘든 CAPTCHA 시스템을 제공하고 있다. 이와 같이 문자 인식을 통한 시스템은 컴퓨터 가 통과할 수 있는 시스템 및 사용자에게 어려움이 많은 CAPTCHA 시스템을 제공한다.

문자 인식 프로그램 또는 이미지 프로세싱을 통하여 웹봇이 쉽게 통과할 수 있다[10]. b) 와 같은 예제는 문자 인식 프로그램을 통하여 인식하기 힘든 점이 있으나, Projection-based segmentation 알고리즘을 통하여 문자 들을 덮고 있는 선을 제거 할 수 있다[3]. b) 에 존재하고 있는 선을 제거한다면 a) 와 같은 예제가 되므로 문자 인식 프로그램을 통하여 웹봇이 쉽게 통과할 수 있다. c) 와 같은 경우는 글자위에 존재하는 선을 더 굵게 하여 CAPTCHA 시스템이 제공되었으며, Projection-based segmentation 알고리즘을 이용하여 선들을 제거할 수 없지만, 사용자에게도 제공되는 CAPTCHA 시스템내에 있는 문자를 인식할 수 있는 어려움이 커짐에 따라, 효과적인 CAPTCHA 시스템을 제공하지 못한다[3].

최근 이미지 기반 CAPTCHA 시스템이 개발되었다[4]. 그림 3 와 같은 이미지 기반의 CAPTCHA 시스템은 원안에 이미지를 담아서, 임의적으로 원을 회전시킨뒤 사용자에게 이미지 교정을 요하는 문제이다. 이 시스템은 사용자에게 보다 쉬운 CAPTCHA 시스템을 제공하였으나, 상하가 구분이 없는 사진은 사용자에게 어려움을 제공한다. 상하 구분이 없는 사진이 사용자에게 어려움을 제공하는 이유로 인하여 이 시스템에 사용되는 데이터 수집에 제한이 주어진다.

그림 2 은 Assira CAPTCHA 시스템이다. Assira CAPTCHA 시스템[5]은 각각 개와 고양이가 담겨있는 총 12 장의 사진을 제공한 다음, 그중에서 고양이의 사진을 선별해내는 것이다. Assira 시스템에서는 컴퓨터는 동일한 색을 지닌 개와 고양이의 선별이 어렵다고 말했다. 또한 Assira CAPTCHA 시스템은 사람은 30초안에 99.6%의 확률로 시스템을 통과할 수 있다고 말했다. 하지만 Assira CAPTCHA 는 머신러닝을 통하여 82.7% 로 Assira CAPTCHA 시스템을 패스하였다[11]. 이렇게 이미지 기반의 CAPTCHA 시스템중에서 특정 사물이 존재하는 사진을 선별해내는 CAPTCHA 시스템은 머신 러닝을 통하여 쉽게 통과될 수 있다고 보여진다.

이와 같이 현재 이미지 기반의 CAPTCHA 시스템은 사용자에게 편의를 제공함과 동시에 컴퓨터는 통과하기 어려운 시스템을 제공한다고 생각 하였으나, 머신 러닝, 수직, 수평기법, 이미지 프로세싱등 기타 알고리즘을 사용하여 통과할 수 있었다.

그림 4는 공원에서 일반적으로 촬영한 사진이다. 그림 4에 존재하는 공원의 사진에서 2개의 서브 이미지 a),b)를 추출하였다. 서브 이미지 추출시 이미지에서 임의적으로 위치를 선정하여 서브 이미지를 추출한다. 이렇게 추출된 서브 이미지를 4가지의 방향 0, 90, 180 그리고 270 4가지의 방향으로 임의적으로 회전을 가하여

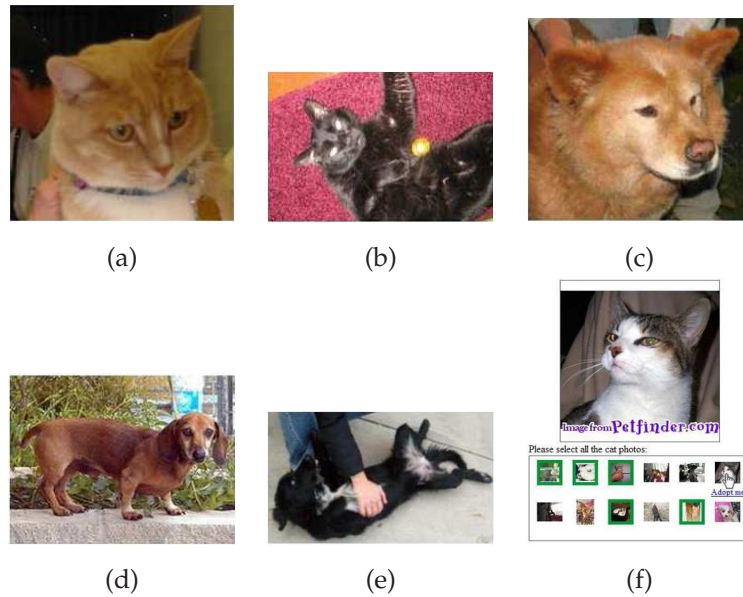


그림 2. Assira CAPTCHA 시스템이다. Assira CAPTCHA 시스템은 12 장의 사진을 제공한 후, 12 장의 사진에서 고양이의 사진을 선별해 내는 것이다. 각 사진마다 Adopt me 를 클릭하였을 경우 확대된 사진을 제공하여 사용자의 편의를 제공한다. Assira CAPTCHA 시스템은 사람은 30초 안에 99.6%의 확률로 시스템을 통과할 수 있다고 하였다. (a)-(e)는 Assira 에서 발췌한 대표적인 개, 고양이 이미지 5 개 이며, (f)는 Assira 시스템이다.

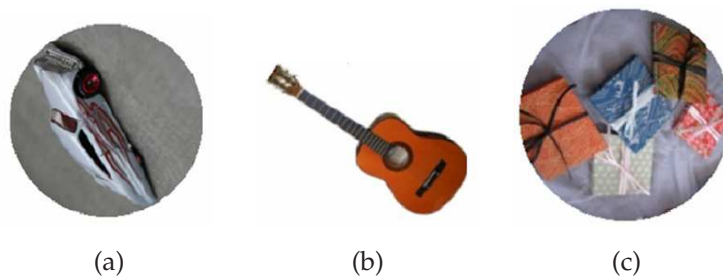


그림 3. 이미지 기반의 CAPTCHA 시스템이다[4]. 원안에 이미지를 담아서 임의적으로 회전시킨 후 사용자에게 교정을 요하는 문제이다. 이미지 기반의 시스템이라 사용자에게 편의를 제공할 수 있으나, 상하 구분이 없는 사진의 경우 사용자에게 어려움을 제공하고, 데이터 수집에 제한적이다.(a) 와 같은 예제는 정확하게 방향으로 잡을 수 있는 그림이며, (b)의 경우 방향은 잡을 수 있지만 정확한 방향이 다소 모호한 경우, (c)는 이미지에 특별한 의미적 개체(semantic cue)가 없어서 전혀 방향을 가늠할 수 없는 경우

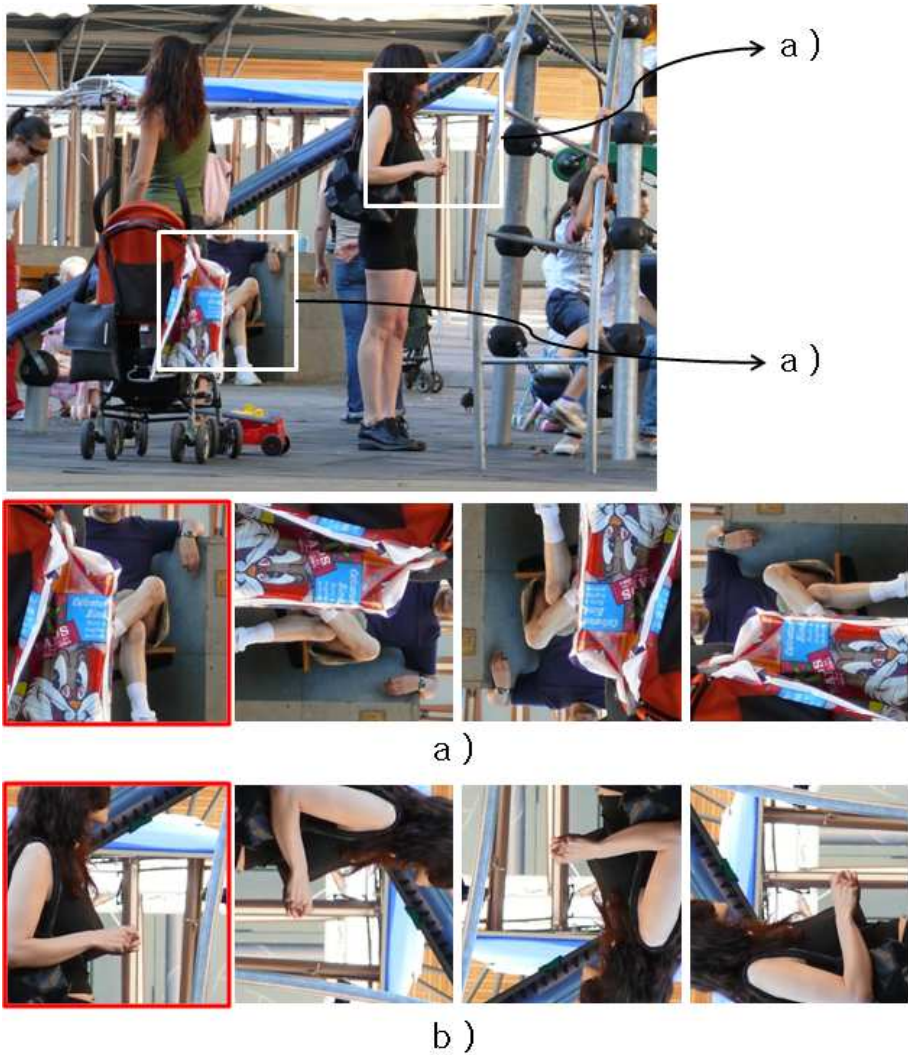


그림 4. 일반적인 사진이 주어지면, 본 논문에서 제시한 CAPTCHA 시스템은 임의적으로 서브 이미지를 추출한다. a) 와 같은 서브 이미지를 추출할 경우 0, 90, 180 그리고 270 도 와 같은 4 가지의 방향으로 임의적으로 이미지 회전을 통하여 CAPTCHA 시스템을 제공한다.

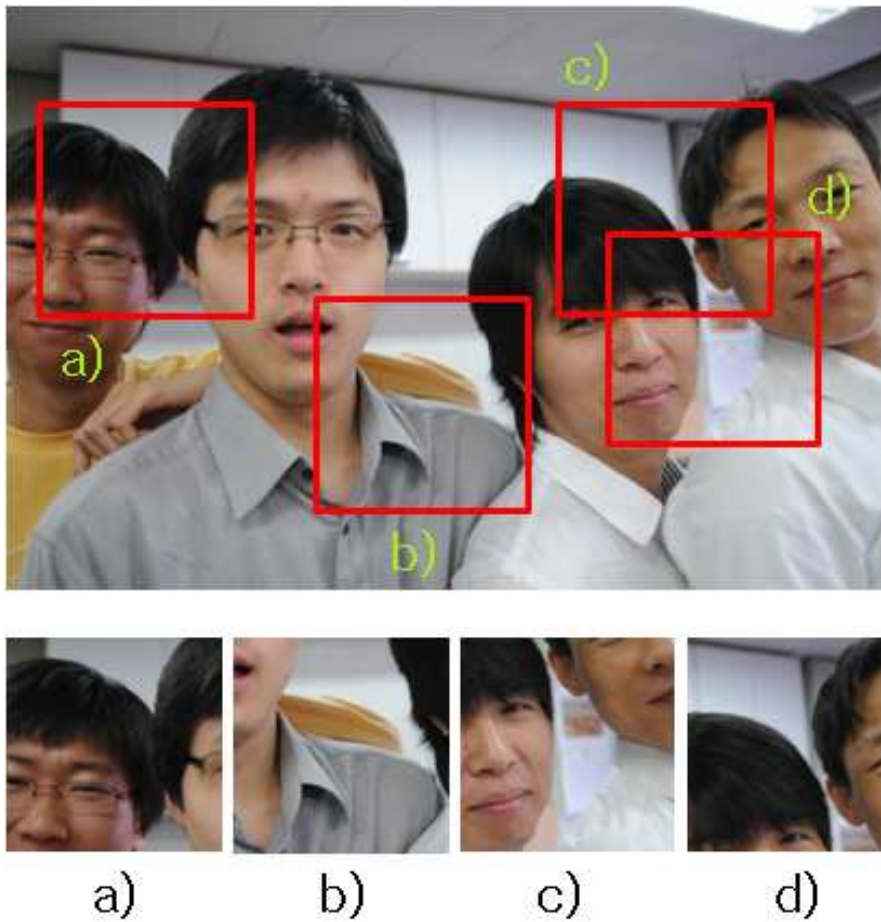


그림 5. 한 장의 사진에서 여러 장의 서브 이미지를 추출한 것이다. 본 예제는 한장의 사진으로 부터 a), b), c), d)와 같이 총 4장의 서브 이미지를 추출하였으며, 얼굴인식을 기피하기 위하여 서브 이미지에는 사람의 얼굴 전체가 들어간 사진은 필터링을 통하여 제외 시켰다. 4장의 사진에는 사람 얼굴이 부분적으로 들어가 있으나, 얼굴 인식 프로그램으로는 인식할 수 없으며, 사람은 이러한 의미있는 사진을 쉽게 교정할 수가 있다.

사용자에게 올바른 교정을 요하는 것이 본 논문에서 제공하는 서브 이미지 기반의 CAPTCHA 시스템이다. 그림 4 하단에 보이는 서브 이미지 a)에 대하여 4가지의 방향 0, 90, 180, 270도의 회전시킨 4가지의 서브 이미지를 보여주고 있다. b) 역시 마찬가지이다. 사람의 경우 a)와 b) 같은 서브 이미지로 하여금 올바른 교정을 요한다면 쉽게 교정을 할 수가 있다.

## 2 시스템 구조

본 단락에서는 본 논문에서 제시한 시스템의 전반적인 구조와 시스템을 소개한다. CAPTCHA 시스템이 가져야 할 기본적인 요소는 자동생성 및 데이터의 다양성, 간편한 User Interface와 쉬운 난이도 그리고 웹봇 방지이다. 즉, 실사용자를 쉽게 판별할 수 있는 사용자는 통과 하기 쉬운 CAPTCHA 시스템을 제공하며, 컴퓨터에게는 어려운 CAPTCHA 시스템을 제공하는 것이다[15]. 텍스트기반의 CAPTCHA 시스템은 텍스트의 조합으로 자동 생성과 다양한 데이터를 만들 수 있다. 그러나 이미지 기반 CAPTCHA에서는 이러한 조합에 의한 생성이 불가능하기 때문에 많은 이미지 데이터들이 필요하다. 본 논문에서 제시하는 CAPTCHA 시스템은 기존의 방식과는 달리 일반 사진의 일부분 즉, 전체 이미지가 아닌 서브 이미지를 사용한다. 이는 본 시스템의 가장 핵심 아이디어로서 하나의 이미지(사진)내에서 많은 량의 서브이미지 추출이 가능하므로 데이터의 다양성이 보장된다. 본 논문에서 제시한 CAPTCHA 시스템의 데이터로 일반적으로 촬영한 1,850장의 사진이 있다. 우리는 이 1,850 장의 사진에서 약 100,000 개의 서브 이미지를 추출할 수가 있다.

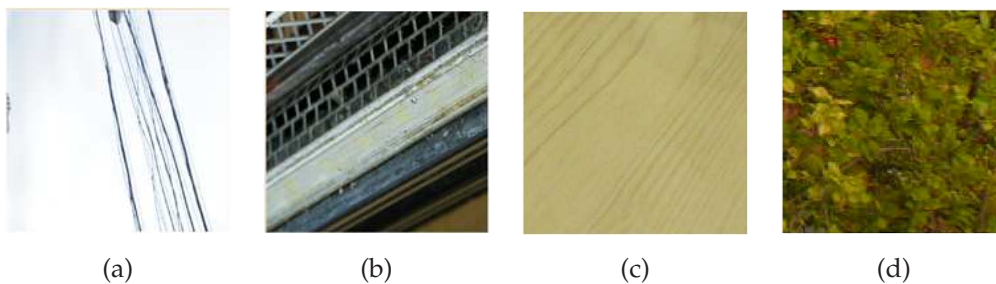


그림 6. 서브 이미지의 추출이 사용자에게 혼란을 야기한 서브 이미지들이다. a)와 같은 사진은 우리가 흔히 볼 수 있는 전선줄 사진이다. 사용자는 이러한 서브 이미지는 사용자가 이미지 교정에 어려움을 겪는 사진이다. b) 역시 건물의 일부분을 담은 사진이지만 사용자가 이미지 교정에 어려움을 겪는 사진이다. c)와 같은 예제는 가정에서 흔히 볼수 있는 마루 바닥을 뜻하는 사진이나 이미지 교정에 어려움을 겪는 사진이다. d)와 같은 경우도 식물을 촬영한 사진이란걸 알 수 있으나 정확한 이미지 교정은 어렵다.

시스템의 동작 방식은 추출한 서브 이미지를 0, 90, 180, 270도로 랜덤하게 회전시키고 사용자는 간편한 마우스 클릭을 이용하여 추측되는 원래의 Orientation으로 수정하는 것이다. 서브 이미지 추출 시 이미지의 비율로 서브 이미지를 추출하기 때문에, 이미지의 크기와는 무관하게 추출할 수가 있다. 하지만 서브 이미지의 추출하여 CAPTCHA 시스템을 제공할 시 사용자에게 혼란을 야기한 이미지가 추출될 수가 있다. 그림 6과 같은 사진은 사용자에게 혼란을 야기 시키는 이미지라 할 수가 있다. 그림 6에 있는 사진의 경우에는 너무 가까운 거리에서의 촬영된 이미지 또는 이미지를 구성하고 있는 색상의 비율 그리고 모호한 Edge의 방향에 의해 인식하기 힘든 문제가 발생된다. 그림 14는 일반적으로 촬영한 2개의 사진을 볼수가 있다. 이 두 사진에는 색상의 분포도가 고르지 못하다. (a)에 존재하는 빨간 박스에 따라서 서브 이미지를 추출할 경우 사용자에게 어려움을 야기시키는 이미지가 될 것이며, b)에 존재하는 빨간 박스 또한 마찬가지이다. 제안하는 시스템에서는 이러한 모호함을 줄

이기 위하여 이미지를 구성하고 있는 색상의 비율, 너무 가까운 촬영 거리(Zoom 고려)를 자동 Curation 한다. 뿐만 아니라, 이미지 프로세싱을 통한 웹봇 방지를 위하여 사람의 전체 얼굴이 나오는 서브이미지는 얼굴인식 프로그램[20]을 자동으로 필터링한다.

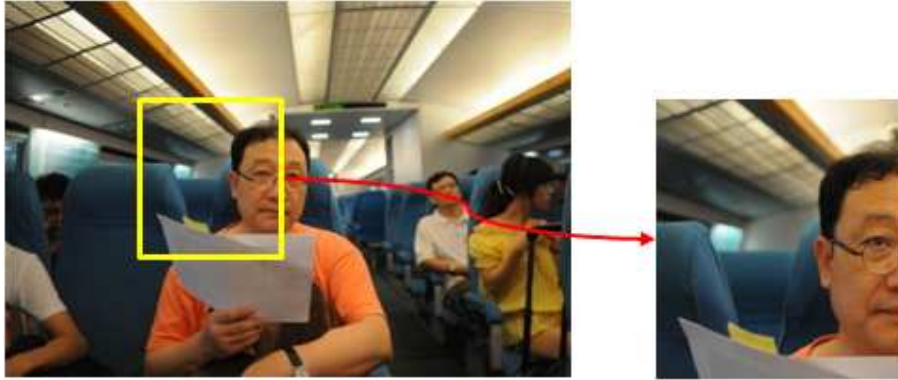


그림 7. 좌측의 이미지의 경우 컴퓨터는 얼굴인식을 통하여 이미지에 회전을 가하여 올바른 교정을 할 수 있지만, 우측의 경우 얼굴인식 프로그램을 통하여서는 얼굴을 인지 할수가 없으므로 컴퓨터는 올바른 교정을 할 수 없지만, 사람에게게는 통과하기 쉬운 이미지를 제공한다.

그림 7은 얼굴 인식 프로그램을 사용하여 컴퓨터가 얼굴 인식을 통하여 쉽게 교정할 수 없는 서브 이미지를 추출한 예제이다. 그림 5는 얼굴인식 프로그램을 사용하여 서브 이미지 추출시 얼굴 전체의 모습이 담겨져 있는 사진을 회피하여 서브 이미지를 추출한 것이다. 본 예제는 한장의 사진으로 부터 총 4장의 사진을 추출하였으며, 4장의 사진에는 전부 얼굴의 부분적인 모습이 들어가 있어 의미있는 사진이 되어 사용자에게는 교정이 쉬운 이미지가 될수 있으며, 컴퓨터에게는 어려운 이미지로 될수 있다. 또한 옷과 같은 색상 분포도가 적은 서브 이미지의 경우는 필터링을 통하여 서브 이미지로써 추출되지 않도록 하였다. 본 시스템에서는 서브 이미지 추출시 색상 분포도가 적은 서브 이미지를 추출하지 않기 위하여, 서브 이미지에 존재하는 색상을 총 25가지의 색상으로 변환한 다음, 그 중에서 한 가지의 색상이 55% 이상이면 서브 이미지를 추출 하지 않기로 하였다[12]. 서브 이미지 추출한 뒤, 25가지의 색상으로 변환하였을 경우 성능면에서도 향상된 성능을 보이며 사람이 가장 선별하기 쉬운 색상의 갯수는 25가지이므로 서브 이미지에 존재하는 색상을 25가지로 변화하여 필터링을 제공하였다.

일반적으로 회전된 이미지를 찾는 많은 방법들이 공개되어 있다[19, 9]. 기존에 있는 방법들은 머신 러닝[14, 16] 모서리, edge 등으로 찾는 방법, 사진 내에 존재하는 수직, 수평들을 탐지하여 찾는 방법, 벡터머신을 이용하여 하늘을 Detection 하는 방법, 사진 내 얼굴을 인식[17]하여 회전된 이미지의 원래 방향을 찾았다. 그러나 본 시스템에서는 전체 이미지가 아닌 서브이미지를 사용하기 때문에 또한 서브 이미지 추출시 앞서 설명하였던 25 색상 테이블을 이용하여 하늘과 같은 단일 색상또는 색상 균포가 균일하지 않은 것은 서브 이미지로 제공하지 않기 때문에 기존의 이미지 교정 방법으로는 자동으로 방향을 찾기 힘들다.

그림 8은 본 논문에서 제시한 서브 이미지 기반의 CAPTCHA 시스템에서 사용자에게 제공되는 순서를 나타내고 있다. 서브 이미지 기반의 CAPTCHA 시스템은 총 4가지로 분류할 수 있다. 첫 번째는 DB1, DB2에 존재하는 이미지들을 임의적으로 이미지를 선택한다. 임의적으로 선택된 이미지에서 서브 이미지를 임의적 좌표에서 비율에 따른 크기에 따라 서브 이미지를 추출한다. 추출된 서브 이미지에 대하여 본 시스템에서 제공되는 필터링을 통하여 얼굴 인식 또는 색상 분포도에서 만족하지 못할 경우 다시 임의적으로 이미지를 선택하여 서브 이미지를 추출한다. 필터링에서 통과되어 서브 이미지가 갖추어지면 서브 이미지에 대하여 사용자가 쉽게 이미지 회전을 가할 수 있는 User Interface를 제공한다. User Interface가 갖추어지면 본 시스템은 N 개의 서브

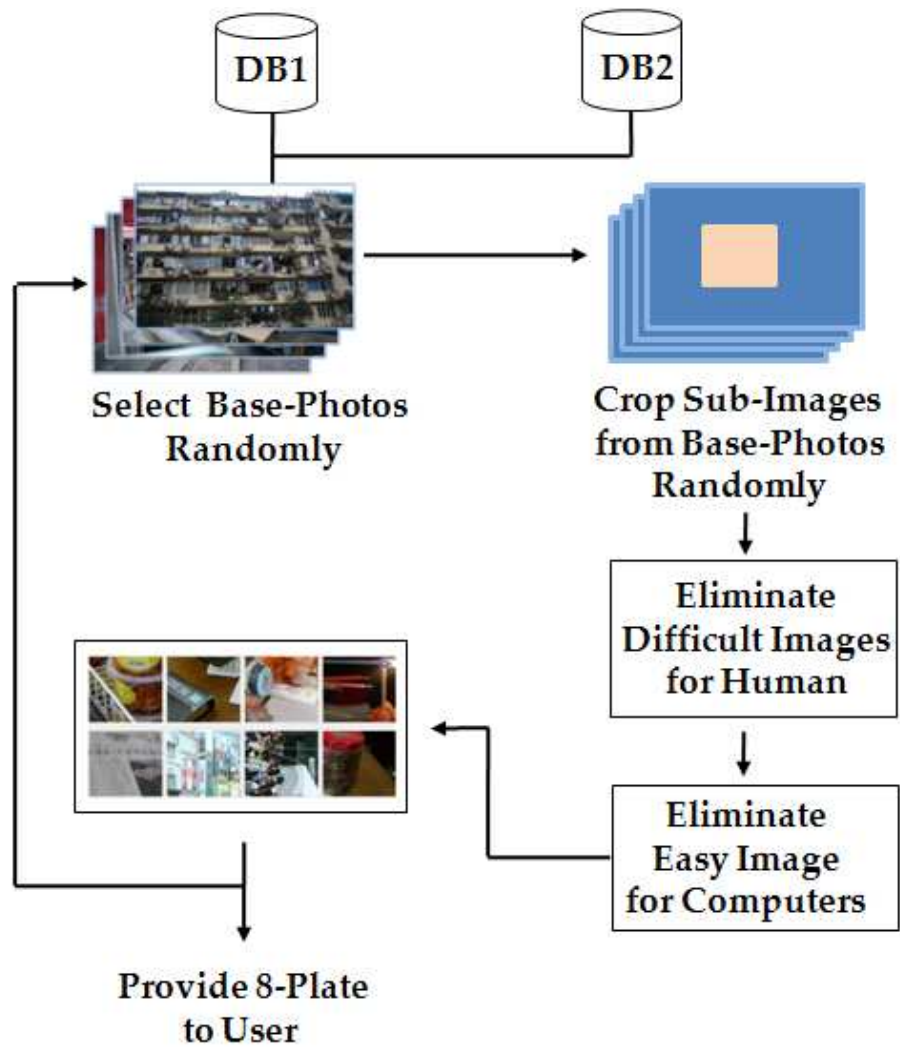


그림 8. 본 논문에서 제시한 CAPTCHA 시스템에서 서브 이미지가 제공되는 순서도를 보여주고 있다.



그림 9. 이미지의 크기에서 10%의 배율 크기로 서브 이미지를 추출하여 CAPTCHA 시스템을 제공한 것이다. 지금 제공된 것은 필터링을 거치지 않고 제공된 것이다. 좌측 2개의 서브 이미지는 색상 분포도가 균일하지 않거나와 사람이 인식 할 수 없는 서브 이미지로 이루어져 있다. 또한 우측의 2개의 서브 이미지는 사람의 얼굴이 포함되어 있어 컴퓨터에게는 쉽게 통과할 수 있는 서브 이미지를 제공하고 있다.



이미지를 제공하여 N-1 개의 이미지에 대하여 교정에 성공 하였을 경우 CAPTCHA 시스템에서 패스하였다고 가정한다. DB1 에 해당하는 사진은 여행을 다니면서 찍은 대부분의 사진이 풍경용 사진이며, DB2 에는 일반적

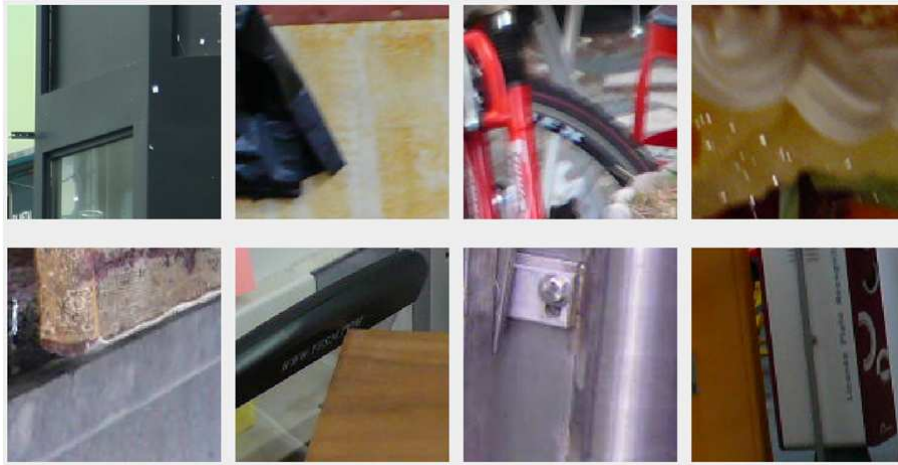


그림 10. 본 논문에서 제시한 CAPTCHA 시스템에서 서브 이미지 추출시 10%의 비율로 추출하여, CAPTCHA 시스템을 제공한 것이다.

으로 촬영한 사진들이다. 하지만 대부분의 사진들이 많은 수직, 수평 선들을 가지고 있으며, 대부분의 사진의 윗부분이 밝거나, 하늘을 포함하고 있다. 하지만 본 시스템에서 서브 이미지를 추출하기 때문에 수직 수평 선분이 있다 하여도, 이미지의 회전을 가한 이미지에도 똑같은 수직 수평 선분이 존재하며 컴퓨터는 이러한 수직, 수평 선분을 통하여 이미지를 교정하기는 어렵다 하지만 사람의 경우 이 수직, 수평 성분이 아닌 사진이 내포하고 있는 의미로 이미지 교정에 임하기 때문에 교정에 큰 어려움은 없다. 또한 하늘과 같은 윗부분이 밝거나 단일 분포 색상을 내포하고 있는 경우 25 색상 테이블을 이용하여 서브 이미지로 하여금 제공하지 않기 때문에 컴퓨터 역시 이미지 교정에 어려울 것이다.

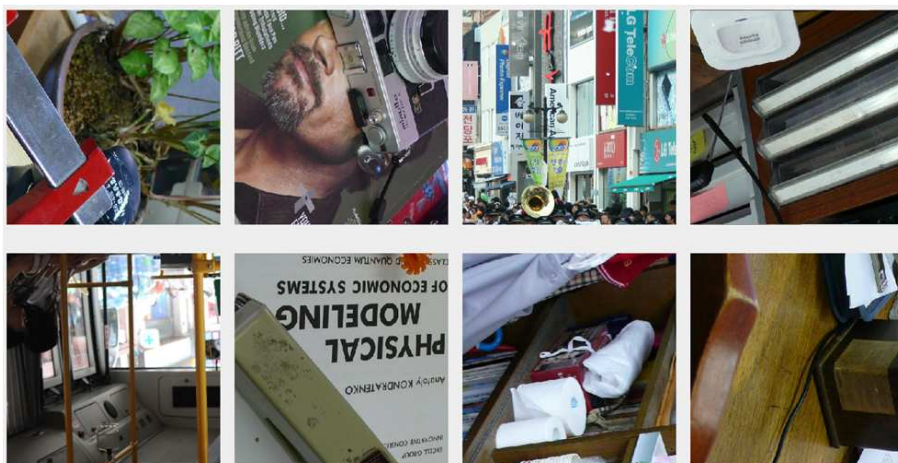


그림 11. 본 논문에서 제시한 CAPTCHA 시스템에서 서브 이미지 추출시 30%의 비율로 추출하여, CAPTCHA 시스템을 제공한 것이다. 사용자가 CAPTCHA 에 임하기 전 CAPTCHA 시스템의 스냅샷이다.

그림 9와 같은 경우는 필터링을 통하지 않고 서브 이미지 기반의 CAPTCHA 시스템을 제공한 것이다. 즉, 얼굴 인식, 25 컬러 색상 테이블을 통하여 필터링을 하지 않고 CAPTCHA 시스템을 제공한 것이다. 그림 9 좌측 2개의 서브 이미지는 색상 분포도가 균일하지 않거나 사람이 전혀 인식할 수 없는 즉, 의미가 내포되지 않은 서브 이미지로 교정이 있어서 사용자가 어려움을 느끼게 되는 예제이며, 우측의 2개의 이미지는 얼굴 인식 프로그램을 사용하지 않고 서브 이미지를 추출하여 컴퓨터에게는 비교적 쉬운 교정을 할 수 있는 예제가 될 수 있다.

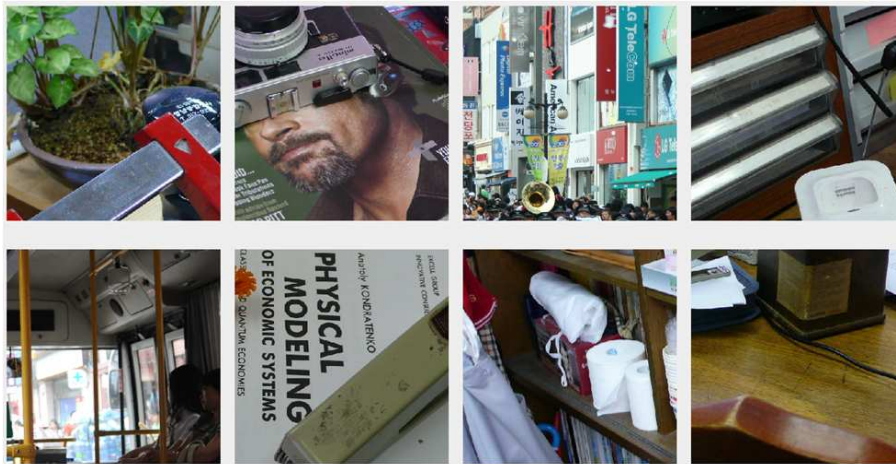


그림 12. 11에서 제공된 CAPTCHA 시스템에서 올바른 이미지 교정을 한 뒤의 스냅샷이다.

본 논문에서 제공한 서브 이미지 기반의 CAPTCHA 시스템이다. 그림 10의 경우 색상 분포도, 얼굴 인식을 통한 필터링을 통하여 서브 이미지를 추출한 것이다. 하지만 너무 작은 비율로 추출한 결과 사용자가 의미 있는 서브 이미지를 찾기 힘들거나, 교정에 있어서 어려움을 제공하는 예제이다. 그림 11와 같은 예제는 원 이미지에서 30%의 확률로 서브 이미지를 추출하였다. 10% 비율로 서브 이미지를 추출한 것 보다 서브 이미지내에 의미가 내포되어 있으며, 사용자가 이미지 교정에 있어서 큰 어려움을 못느끼는 예제가 될 수 있으며, 그림 12은 사용자가 이미지 교정을 통하여 서브 이미지 전체에 대하여 정답을 제공한 예제이다.

### 3 실험

본 단락에서는 본 논문에서 제시하였던 서브 이미지 기반의 CAPTCHA 시스템을 통하여 실험한 결과에 대해서 알아보도록 하겠다. 본 논문에서 제시한 서브 이미지에 실험은 약 50 명가량의 사용자들을 통하여 시행되었다. 실험에서 제공된 서브 이미지 기반의 CAPTCHA 시스템은 서로 다른 8장의 이미지에서 서브 이미지를 추출한 뒤, 각 이미지에 회전을 가한 뒤 제공되었다. 서브 이미지를 추출할 시 총 10%-40% 까지의 비율로 서브 이미지를 추출 하여 총 7가지 크기의 서브 이미지 크기에 대하여 실험 하였다. 일반적인 사진에서 35% 크기의 비율로 서브 이미지를 추출하여 서브 이미지 기반의 CAPTCHA 시스템에서의 성공률은 약 97.4%의 성공률을 보였으며, 사용자가 CAPTCHA 시스템을 수행하는데 걸리는 시간은 약 9.2초가량이 소요되었다.

그림 9에서 알수있듯이 일반적인 촬영된 사진에서 10% 비율의 크기로 서브 이미지를 추출하였을때, 색상 분포도 균일 하지 못하고 또한 서브 이미지 내에 의미가 전혀 내포되어 있지 않아 사용자가 이미지 교정에 있어서 큰 어려움을 느낀다고 판단되었다. 30%이상의 서브 이미지 추출은 사용자에게 이미지 교정에 어려움을 느끼지 못하였으며, 컴퓨터에게 쉽게 통과할 수 없는 이미지를 추출되어 30%가 적당하다고 판단하였다. 그래프

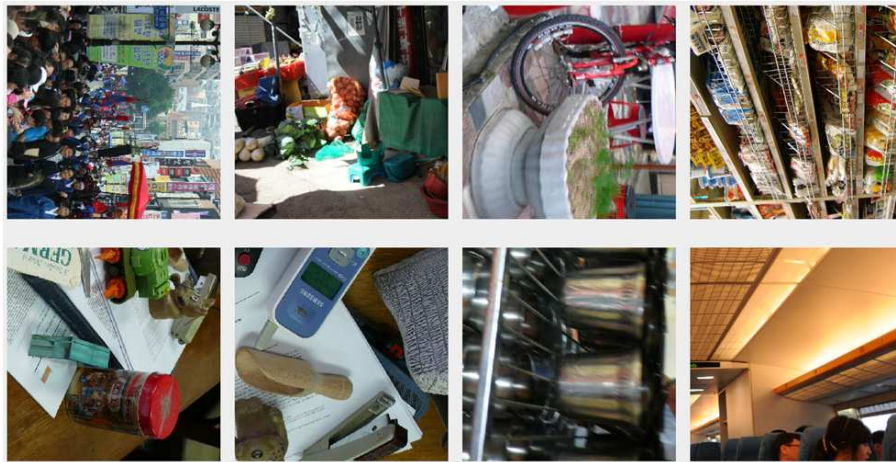
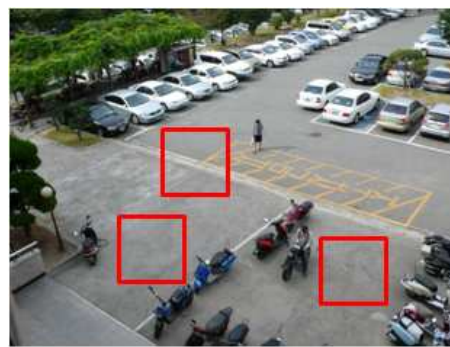


그림 13. 서브 이미지의 추출시 40%의 확률로 추출한 CAPTCHA 시스템이다. 필터링을 통하여 제공된 서브 이미지이며, 사용자가 CAPTCHA 시스템에도 임하였을때 의미 있는 사진을 쉽게 찾을 수 있는 서브 이미지의 크기를 가지고 있다.



(a)



(b)

그림 14. 일반적으로 촬영한 사진들이다. 이 두 사진에는 색상의 분포도 고르지 않다. 만약, 사진에 보이는 빨간 박스들이 서브 이미지로 추출되어, 본 논문에서 제시한 CAPTCHA 시스템에 서브 이미지로 제공된다면, 사용자들에게는 통과하기 쉬운 CAPTCHA 시스템을 제공하지 못한다.

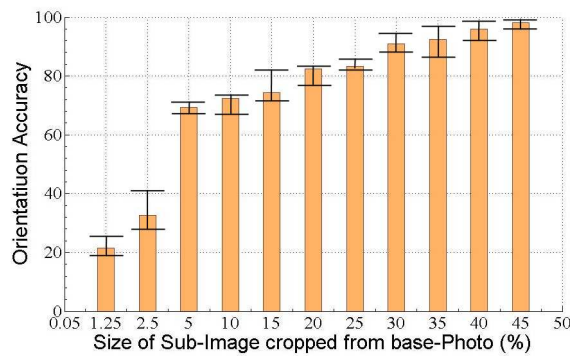


그림 15. 실험 그래프

SIZE	0.1	0.15	0.20	0.25	0.30	0.35	0.40
tester 1	77.4	80.2	85.7	84.9	94.9	97.6	98.5
tester 2	73.4	76.4	80.7	82.1	88.8	98.6	98.1
tester 3	78.4	79.1	80.1	87.2	88.1	97.7	100.0
tester 4	75.1	74.5	83.9	86.3	91.5	98.4	98.2
tester 5	72.6	81.2	80.1	83.2	91.1	96.8	98.1
tester 6	75.9	78.2	79.4	86.9	93.5	97.5	100.0
tester 7	74.9	75.1	82.4	85.7	88.3	98.7	98.5
tester 8	75.5	74.3	81.2	81.2	94.8	96.4	100.0
tester 9	73.9	80.2	78.5	81.2	88.1	96.4	98.1
tester 10	74.6	77.3	79.2	87.1	93.1	97.2	98.4
Average	75.1	77.6	81.1	84.4	91.2	97.4	98.7

표 1. 10 명의 실험 참가자들이 8 개의 서브 이미지로 이루어진 CAPTCHA 시스템을 실험한 테이블이다.

15에서 5% 비율로 추출된 서브 이미지의 경우 CAPTCHA 시스템에 임하였을 경우는 정답률이 0%에 가까웠고 서브 이미지의 비율이 증가됨에 따라서 정답률이 높아 지는 것을 볼수 있으나, 30%에 가까울수록 증가량이 적어지는 것을 알수가 있다. 그리하여 30%가 사용자에게는 이미지 교정에 있어서 쉽게 통과할 수 있으며, 컴퓨터에게는 어려운 CAPTCHA 시스템을 제공할 수 있다는 것을 알 수가 있었다. 우리는 이러한 실험을 통하여 기본적으로 각 이미지에 임의적으로 맞출 수 있는 확률은 다음과 같다는 것을 알 수가 있다.

표 1에서는 10 명의 사용자에게서 일반적으로 촬영한 사진에서 7 가지의 비율로 추출하였을 때, 각 비율에 따른 사용자의 성공률을 나타내고 있다. 표 2에서는 일반적인 이미지에서 각 비율에 따라 추출된 서브 이미지에 따른 사용자의 이미지 회전의 갯수, 성공률과 시간을 나타내고 있다. 표 2에서는 각 사용자들은 비율의 최소일때는 이미지에 대한 의미를 전혀 찾을 수 없으므로 이미지에 대한 교정이 거의 없는 것으로 나타났다. 하지만 이미지의 크기가 증가됨에 따라 이미지에 대한 의미가 조금씩 부여됨으로 사용자는 이미지에 대한 의미가 부족하므로 계속해서 이미지에 대한 교정을 시도하며 이미지에 대한 올바른 교정을 요하는 것으로 나타났다. 또한 이미지에 대한 교정이 많아 짐으로 시간도 같이 증가하는 것을 알 수 있다. 하지만 30%에 가까울 수록 서브 이미지에 대한 의미가 명확하게 부여가 되고 사용자 역시 몇번의 교정만으로도 올바른 해답을 찾을 수가

Image SIZE	Accuracy %	Clicks	Time(Sec.)
0.10	75.1	0.81	6.2
0.15	77.6	1.12	10.4
0.20	81.1	1.15	14.3
0.25	84.4	1.46	16.4
0.30	91.2	1.41	13.7
0.35	97.4	1.14	9.2
0.40	98.7	1.11	6.5

표 2. 서버 이미지의 크기에 따른 평균 성공률, 평균 클릭수, 시간을 나타낸 표이다. 서버 이미지의 크기에 따라서 성공률은 증가하고 있으나, 클릭수는 30%까지 증가하다가 감소하고 있는 것을 나타내고 있다. 서버 이미지 추출시 너무 작은 비율로 추출시 의미 있는 이미지를 찾기 힘들어서 클릭의 수가 적은 것을 나타내지만, 서버 이미지 비율이 증가되므로 의미가 부여 되기 때문에 클릭의 횟수가 늘어가면서 비율이 어느 정도 확립이 되면은 사람이 인지 할수 있는 의미가 부여된 서버 이미지가 추출될 수 있기 때문에 클릭의 횟수가 적어지고 그에 따른 성공률도 증가되는 것을 알 수 있다.

있으므로, 클릭의 횟수가 줄어 들고 시간도 줄어 드는 것을 알수가 있다. 이에 따라 본 시스템은 클릭의 횟수가 낮아지는 지점 또는 시간이 감소하는 시점을 서버 이미지 추출의 최대 비율로 정하였다.

본 논문에서 제공한 서버 이미지 기반의 CAPTCHA 시스템을 기본적인 CAPTCHA 시스템의 취지에 가까운 시스템으로 만들기 위해 즉, 사용자에게는 쉬운 컴퓨터에게는 어려운 CAPTCHA 시스템을 제공하기 위하여, 머신 러닝[1, 7] 실험을 통하여 컴퓨터에게는 더욱 어렵다는 실험 결과를 얻어 내었다. 머신 러닝 실험에 트레이닝 셋은 4,000장의 사진으로 구성 되어 있다. 4000 장의 사진은 동일한 사진 1,000장의 사진을 0, 90, 180 그리고 270 이미지 회전을 통한 각각 1000 장 씩으로 구성되어 있다. 이 4,000장의 사진으로 트레이닝 셋을 만들때 각 사진에서 이미지가 가지고 있는 특정 값들을 추출해서 트레이닝 셋을 만들었다[8]. 4000장의 사진으로 트레이닝 셋을 만들고 테스트 셋은 서버 이미지로 이루어진 400장의 사진이다. 이 400장의 서버 이미지 역시 4가지의 방향이 100장씩 포함된 400장의 사진이다. 이 400장의 서버 이미지를 4가지의 비율로 서버 이미지를 출력하여 실험에 임하였다. 4 가지의 비율은 10, 30, 50, 100%로 이루어져 있다. 10-30%의 비율로 추출된 서버 이미지는 머신 러닝을 통하여 68% 미만의 성공률을 보였으며, 전체 이미지인 100%의 사진은 85.10%의 성공률을 보였다. 68%의 성공률을 보이는 전체 이미지의 머신 러닝의 결과는 일반적인 사용자가 본 시스템에 20%의 비율로 서버 이미지로 구성된 CAPTCHA 시스템을 임한 것과 마찬가지로 되므로 결국 머신 러닝으로도 우리의 시스템은 통과할 수 없다는 것을 입증 하였다. 100%의 전체 이미지를 머신 러닝에 테스트한 결과 85.10%의 결과를 보였으나, 본 시스템에서 제공되는 CAPTCHA 시스템은 서버 이미지의 기반으로 제공되기 때문에 전체 이미지에 대한 머신 러닝에 대한 결과는 고려하지 않아도 된다[18, 13].

#### 4 결론

본 논문에서는 스팸이나 로봇에 의한 사이트 자동가입, 계정 생성방지 도구로써 서버 이미지 기반의 새로운 CAPTCHA 시스템을 제안하였다. 본 논문에서 제안한 CAPTCHA 시스템은 일반적으로 촬영된 사진에서 사진의 크기에 따라 일정한 비율로 서버 이미지를 추출한 뒤, 이 서버 이미지에 0, 90, 180, 270도 4 가지의 방향중 임의적으로 방향을 선택하여 회전을 가하여 사용자에게 CAPTCHA 시스템을 제공하는 것이었다. 본 시스템에서 제공되는 서버 이미지 기반의 CAPTCHA 시스템은 우리는 실험을 통하여 일반적으로 촬영된 사진에서 이

Image SIZE	Accuracy	Precision	Recall
10%	65.66	31.68	32.32
30%	66.92	35.96	41.41
50%	71.72	43.16	41.41
100%	85.10	68.18	75.76

표 3. 머신 러닝을 통한 결과를 나타낸 표이다. 10-30% 비율의 서브 이미지에서는 68% 미만의 성공율을 보였다. 100% 비율의 크기 즉, 전체 이미지에 대한 머신 러닝 결과는 85.10%의 성공률을 보였으나 본 시스템에서 제공되는 CAPTCHA 시스템은 서브 이미지로 제공되기 때문에 이 결과에 대해서는 고려하지 않아도 된다.

미지 크기에 따른 비율에 따라서 서브 이미지를 추출하여, 실험자들을 통하여 사용자에게는 의미가 있는 서브 이미지를 제공할 수 있고, 컴퓨터에게는 통과 하기 어려운 비율을 실험을 통하여 알수 있게 되었다. 또한 우리는 컴퓨터에게는 통과 하기 어려운 CAPTCHA 시스템을 만들기 위하여 수직, 수평 방법, 얼굴 인식등을 피하기 위하여 필터링 기능도 구성하였다. 본 시스템에서 제공되는 필터링은 컴퓨터가 쉽게 파악할 수 있는 얼굴 인식을 피하기 위하여 서브 이미지 추출시 얼굴이 포함되어 있는 서브 이미지는 추출하지 않도록 하였으며, 사용자에게 또는 자동 이미지 교정 시스템을 회피하기 위하여 색상 분포도가 균일 하지 않은 서브 이미지는 25 색상 테이블을 이용하여 추출하지 않도록 하였다. 우리는 실험을 통하여 일반적인 사진에서 서브 이미지를 추출하여 사용자에게 제공하였을 때, 사용자는 이미지에 대한 의미를 알고, 또한 사진을 찍는 사람들은 의미가 있는 사진을 찍기 때문에 CAPTCHA 시스템에 임하는 사용자들은 본 논문에서 제시한 시스템을 통과할 수 가 있는 것이다. 우리는 또한 현재 존재하고 있는 이미지 CAPTCHA 시스템이 머신 러닝으로 통과 할수 있는 것을 알고 머신 러닝으로 서브 이미지 기반의 CAPTCHA 시스템을 테스트 해보았다. 트레이닝 셋은 1,000 장의 동일한 사진이 4 가지 방향 0, 90, 180, 270 도로 이루어진 4,000 장이 사진이 트레이닝 셋으로 구성되었다. 테스트 셋으로는 서브 이미지의 100 장의 사진을 4 가지 방향으로 구성되어 머신 러닝을 통하여 실험에 임하였다. 원 이미지의 사진으로 테스트 했을 경우 85.10%의 성공률을 보였다. 하지만 본 시스템에서 제시한 CAPTCHA 시스템은 서브 이미지 기반으로 이루어져 있기 때문에 고려 하지 않아도 되며 50% 비율로 추출한 서브 이미지로 테스트한 결과 71%의 성공률을 보였다. 71%의 성공률은 서브 이미지의 비율이 10%에서 나타나는 성공률이므로 머신 러닝을 통하여서도 본 논문에서 제시한 서브 이미지 기반의 CAPTCHA 시스템을 통과 할 수 없다는 것을 증명하였다.

추구 연구로써는 머신 러닝을 통하여 사진 Sample 10 만장을 트레이닝 셋으로 이용하여 좀 더 평균적인 실험을 해볼 예정이고, 본 논문에서 제시되었던 서브 이미지 기반의 CAPTCHA 시스템을 웹 서비스를 제공하여 좀 더 많은 실험자들을 통하여 다양한 실험 결과를 얻을 계획이다.

### 참고 문헌

1. Shumeet Baluja, Automated image-orientation detection: a scalable boosting approach, *Pattern anal. appl.* **10** (2007), 247-263.
2. Kumar Chellapilla, Kevin Larson, Patrice Simard, and Mary Czerwinski, Designing human friendly human interaction proofs (HIPs), Proc. of the SIGCHI conference on Human factors in computing systems, 2005, pp. 711-720.
3. Huang SY et al., A projection-based segmentation algorithm for breaking MSN and YAHOO CAPTCHAs, Proc. of the International Conference of Signal and Image Engineering, 2008.
4. Rich Gossweiler, Maryam Kamvar, and Shumeet Baluja, What's up CAPTCHA?: a CAPTCHA based on image orientation, Proc. of the 18th international conference on World wide web, 2009, pp. 841-850.

5. J. Howell J. Elson, JR. Douceur and J. Saul, Asirra: a CAPTCHA that exploits interest-aligned manual image categorization, Proc. of the 14th ACM conference on Computer and communications security, 2007, pp. 366–374.
6. P Simard K Chellapilla, Using machine learning to break visual human interaction proofs (HIPs), Advances in Neural Information Processing Systems, 2004.
7. Hong-Jiang Zhang Lei Zhang, Mingjing Li, Boosting Image Orientation Detection with Indoor vs. Outdoor Classification, Proc. of the Sixth IEEE Workshop on Applications of Computer Vision, 2002, pp. 95–99.
8. David G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, *International journal of computer vision* **60** (2004), 91–110.
9. Jiebo Luo and Matthew Boutell, A Probabilistic Approach to Image Orientation Detection via Confidence-Based Integration of Low-Level and Semantic Cues, CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 9, IEEE Computer Society, 2004, p. 141.
10. G. Mori and J. Malik, Recognizing objects in adversarial clutter: breaking a visual CAPTCHA, Proc. of Computer Vision and Pattern Recognition., June 2003.
11. Golle Philippe, Machine learning attacks against the Asirra CAPTCHA, Proc. of the 15th ACM conference on Computer and communications security, 2008, pp. 535–542.
12. B. G. Prasad, K. K. Biswas, and S. K. Gupta, Region-based image retrieval using integrated color, shape, and location index, *Comput. vis. image underst.* **94** (2004), 193–233.
13. et al. S. M. Bileschi, Towards component-based car detection, In ECCV Workshop on Statistical Learning and ComputerVision, 2004.
14. Lyu Siwei, Automatic image orientation determination with natural image statistics, Proc. of the 13th annual ACM international conference on Multimedia, 2005, pp. 491–494.
15. V. Srikanth, C. Vishwanathan, Udit Asati, and N. Ch. Sriman Narayana Iyengar, Think-an image based CAPTCHA mechanism (testifying human based on intelligence and knowledge), Proc. of the International Conference on Advances in Computing, Communication and Control, 2009, pp. 421–424.
16. A. Vailaya, H. Zhang, Changjiang Yang, Feng-I Liu, and A.K. Jain, Automatic image orientation detection, *IEEE trans. on image processing* **11** (2002), 746–755.
17. Lei Wang, Xu Liu, Lirong Xia, Guangyou Xu, and Alfred Bruckstein, Image orientation detection with integrated human perception cues (or which way is up), Proc. of ICIP, vol. 2, Sept. 2003, pp. 539–42.
18. Yongmei Wang and Hongjiang Zhang, Content-based image orientation detection with support vector machines, IEEE Workshop on Content-Based Access of Image and Video Libraries, 2001.
19. Yongmei Michelle Wang and Hongjiang Zhang, Detecting image orientation based on low-level visual content, *Computer vision and image understanding* **93** (2004), 328 – 346.
20. Ming-Hsuan Yang, David J. Kriegman, and Narendra Ahuja, Detecting Faces in Images: A Survey, *IEEE transactions on pattern analysis and machine intelligence* **24** (2002), 34–58.