착시현상을 이용한 이미지 CAPTCHA 프레임워크 제시

한준희[1]
[1]민족사관고등학교

Designing a novel image CAPTCHA framework based on illusion

Jooney Han[1]
[1]Korean Minjok Leadership Academy, Hoengseong-gun, Gangwon-do, Korea

요 약 최근 급속도로 발전하는 Vision AI 및 Multimodal AI를 악용하여 CAPTCHA의 파훼법이 개발됨에 따라 심각한 보안 문제가 발생하고 있다. CAPTCHA는 웹사이트에 접속하는 자동화 프로그램을 차단하기 위해 개발된 테스트로, 온라인 투표, 회원 가입 등 사람만이 접근할 수 있어야 하는 기능을 보호하기 위해 사용된다. 하지만 최근 발전하는 AI 기술을 사용하여 이를 자동으로 파훼하고, 크롤링 프로그램을 통해 데이터를 무단으로 수집하거나 악성 데이터를 주입하는 등의 피해가 발생하고 있다. 이에 본 연구에서는 이미지의 착시 현상을 활용하여 AI는 파훼 하지 못하지만, 사람은 문제없이 해결할 수 있는 새로운 CAPTCHA 프레임워크를 제시한다. 상술한 AI들은 이미지 내에 있는 사물들 뿐만 아니라 전체적인 분위기와 구도까지 파악하고 분석 할 만큼 성능이 발전된 상태이다. 따라서 위의 문제를 해결하기 위해 기존의 방식과는 다른, 착시 현상을 이용한 CAPTCHA 프레임워크를 개발하고, 이를 각각 AI와 사람이 어떻게 해결하는지 비교하였다. 그 결과 AI는 테스트를 파훼 하지 못하며, 사람은 큰 문제없이 통과함을 확인했다. 이에, 본 연구에서 제시하는 CAPTCHA 프레임워크가 AI 및 자동화 프로그램의 웹사이트 무단 접근을 효과적으로 방지함을 확인한다.

Abstract Recently, serious security issues have arisen due to the development of methods to bypass CAPTCHA by exploiting AIs. CAPTCHA, a test developed to block automated programs from accessing websites, is used to protect functions that should only be accessible to humans, such as online voting or registration. However, malicious use of advanced AI technologies to automatically bypass these CAPTCHAs are leading to unauthorized data collection through data crawlers, large-scale injection of malicious data, and much more. Therefore, this study presents a new CAPTCHA framework that uses optical illusions in images, which AI cannot decipher but humans can solve without any issue. AIs have been developed to the extent that they can understand and analyze not only the objects within an image but also the overall atmosphere and structure. To solve this issue, I have developed a CAPTCHA framework using optical illusions, different from conventional methods, and compared how both AI and humans solve it. As a result, it was confirmed that AI could not bypass the test, whereas humans passed it without significant difficulty. Thus, this study confirms that the novel CAPTCHA framework proposed effectively prevents unauthorized web access by AI and automated programs.

Key Words: CAPTCHA, Illusion, Multimodal AI, Stable Diffusion ControlNet

1. 서론

1.1 연구 배경

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart)는 HIP(Human Interaction Proof) 기술 중 하나로, 웹상에서 데이터를 무단으로 수집하거나, 사람만 접근할 수 있어야 하는 페이지를 접근하여 악성 데이터를 주입하는 등 다양한 피해를 끼치는 자동화 프로그램을 차단하고자 개발되었다. 캡차 (CAPTCHA)는 왜곡된 텍스트 인식, 논리 문제 해결과 같이 컴퓨터는 해결하지 못하지만, 인간은 매우 쉽게 해결할 수 있는 문제를 제시하여 인간을 구분한다. 하지만, 최근 급속도로 발전하고 있는 AI를 악용하여 높은 정확도로 이를 파훼하는 Solver 프로그램이 등장하기 시작했다. 왜곡된 텍스트를 사용하는 캡차는 CNN 모델에 의해 파훼된지 오래이며, 비교적 해결하기 어려운 이미지 선택 캡차도 최근 객체 인식 AI에 의해 파훼된 상태이다. [1] 또한, Flamingo, CLIP, VisualBERT, GPT-4, PaLI 등 텍스트, 이미지, 오디오의 다양한 입력을 동시에 처리할 수 있는 Multimodal AI의 등장으로 AI가 대부분의 캡차의 질문을 파악하고 해답을 도출할 수 있게 되었다. [2-6] 따라서 본 연구에서는 원래 의도와는 다르게 웹상에서 자동화 봇을 차단하지 못하고, 사용자들의 접근성만 저하시키는 현존하는 캡차의 문제점을 해결하고자 하였다. 이를 위해 착시 현상을 사용하여 AI는 해결하지 못하지만, 사람은 쉽게 해결할 수 있는 새로운 캡차 프레임워크를 개발하게 되었다.

1.2 선행 연구

캡차는 1997년 카네기 멜런 대학교에서 최초로 연구되어, 현재 웹상에서 보편적으로 사용되고 있는 캡차 서비스들로 발전되었다. 하지만 인터넷 상에서 사용되는 캡차의 96.48%를 차지하고 있는 서비스인 reCAPTCHA도 앞서 언급했듯이 파훼 프로그램이 존재하며, 이외에도 hCaptcha, Cloudflare Turnstile 등 다양한 캡차 프로그램의 파훼법이 등장한 상태이다. [1, 7-8]

이렇듯 캡차의 파훼법이 점차 개발됨에 따라, 이를 방지하기 위한 연구도 활발히 진행되었다. 첫 번째로 단순히 이미지 혹은 텍스트의 변형 (Distortion) 정도를 높이는 방식이다. 하지만 이는 AI의 캡차 성공 소요 시간을 늦출 수는 있으나, 사람도 동시에 어려움을 느끼게 되어 오히려 AI가 사람보다 빠른 속도로 캡차를 해결하게 된다. [9] 두 번째는 적대적 공격 (Adversarial Attack)을 캡차 이미지에 사용하는 방법이다. 적대적 공격은 이미지에 특정 노이즈를 추가하여 이미지 인식 모델이 사물을 오분류 하게끔 하는 기전이다. 하지만 이 마저도 이미지에서 적대적 노이즈를 사전에 제거하는 Denoising methods에 의해 이점이 사라졌다. [10-11]

2. 이론적 배경

2.1 캡차의 작동 원리 및 파훼법 분석

2.1.1 캡차의 종류와 작동 원리 분석

현존하는 캡차는 텍스트 기반, 이미지 기반, 오디오 기반, 그리고 사용자 패턴 기반으로 크게 5가지 종류로 나눌 수 있다. 왜곡된 텍스트를 인식해야 하는 텍스트 기반 캡차는 상술했듯 AI가 100%에 가까운 정확도로 파훼 가능하며, 변형 정도를 높인다고 해도 사람만 해결하기 힘들어지는 한계점에 봉착하였다. 하지만 그럼에도 불구하고 아직 이를 사용하는 웹사이트가 많아 보안 개선이 시급한 상황이다. [8]

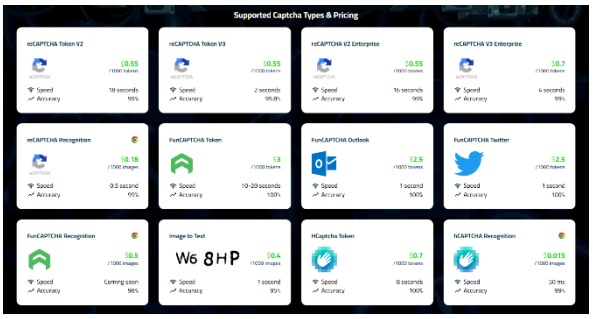
이미지 기반 캡차는 질문에 해당하는 이미지 타일을 선택하거나, 퍼즐을 맞추는 등 다양한 제시 패턴을 해결해야 한다. reCAPTCHA v2의 경우 1차 테스트인 체크박스 선택에서 판별이 불가능할 경우, 이미지 타일을 선택하는 캡차를 사용하며, hCaptcha, Arkose Matchkey 등 웹상에서 보편적으로 사용도는 대부분의 캡차 서비스도 이미지 기반 캡차를 사용중이다. 이는 이미지 기반 캡차가 텍스트 기반 캡차에 비해 보안이 상대적으로 뛰어나면서도, 사람이 해결하기에 크게 어렵지 않기 때문이라고 사료된다. 또한, 이미지 캡차의 경우 시각장애인, 난독증 환자, 눈 기능 저하 등 신체적인 문제로 인해 해결하지 못하는 상황이 발생하기에, 음성에서 들리는 문구를 받아 적어야 하는 오디오 캡차가 두 번째 옵션으로 제시되는 경우도 존재한다.

마지막으로, 사용자 패턴 기반 캡차는 웹사이트에 접근한 유저의 정보 및 행동 패턴을 토대로 사람 여부를 판단한다. 사용자 패턴 기반 캡차는 reCAPTCHA v2 및 v3가 대표적으로, 웹사이트 접속 시점으로부터 유저의 검색기록, 쿠키, 마우스 이동 양상 등을 분석한다. 이러한 방식은 사용자의 웹 접근성을 저하시키지 않는다는 장점이 있으나, 개인정보를 사용한다는 점에서 보안상의 우려가 제기되고 있다. [12]

2.1.2 캡차 Solver의 종류

악성 유저들은 자동화 프로그램을 사용하여 무단 데이터 수집을 비롯한 다양한 디지털 윤리에 어긋나는 행위를 시도한다. 이 과정에서 방해가 되는 캡차를 파훼하기 위한 Solver 프로그램이 다수 등장하게 되었다.

캡차 Solver는 굉장히 종류가 다양하고 유저들이 사용하기 쉽게 배포되어 있다. 당장 인터넷에 "Captcha Solver"라고 검색을 해도 최소 10가지 이상의 서비스가 나열되는 것을 볼 수 있다. 또한, 이러한 캡차 Solver들은 특정 캡차만 파훼할 수 있는 것이 아니라, 다양한 캡차의 파훼를 서비스로써 제공하고 있는 상황이다. 예를 들어, "CaptchaAI" 서비스는 reCAPTCHA v2, v3, hCaptcha, Solve Media, Invisible reCAPTCHA, 그리고 27,500 종류가 넘는 텍스트 캡차의 파훼법을 돈을 받고 제공하며, 2023년 8월 31일 기준 "1st CAPTCHA" 서비스는 [Fig. 1]에 나와 있듯 다양한 캡차를 100%에 수렴하는 정확도와 평균 5초 정도의 시간으로 파훼한다. [13-14]

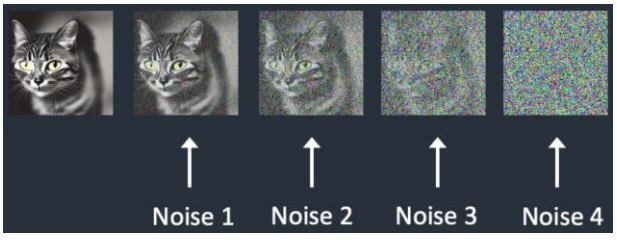


[Fig. 1] 1st CAPTCHA의 캡차 파훼 서비스

캡차 Solver 서비스는 크게 두 가지 종류가 있는데, 하나는 AI를 활용한 파훼 서비스이며, 두 번째는 외주를 통해 캡차를 해결하는 경우이다. "2Captcha" 서비스에 해당되는 두 번째 경우는 사용자가 "Worker"와 "Payer"로 분류된다. Worker는 해결 건수에 비례한 돈을 받으며 캡차를 해결하고, 이렇게 해결된 캡차를 api를 통해 Payer는 일정 비용을 지불하고 자동화 프로그램에서 사용하는 원리이다. 이런 방식의 서비스는 실제 인간이 캡차를 해결하는 것이기에 마땅한 해결책이 존재하지 않는다. 하지만 첫 번째 경우, 즉 AI를 활용한 파훼 서비스는 충분히 방지가 가능하다. 또한, 일부 캡차 Solver AI의 코드가 오픈소스로 공개되어, 이를 비용 부담 없이 사용할 수 있기에 악용의 위험성도 다분하다. 이에, AI를 활용한 캡차 파훼법의 방지에 대한 연구를 진행하게 되었다.

2.2 Stable Diffusion AI의 원리

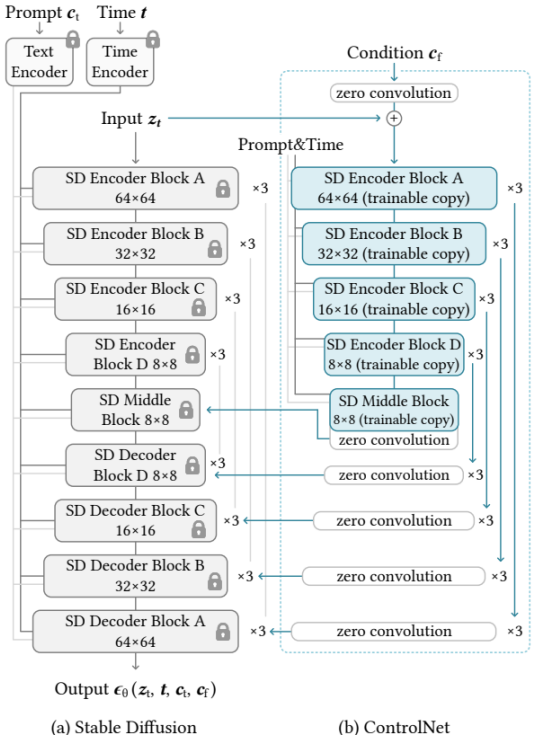
Stable Diffusion AI는 2022년 8월에 공개된 모델로, 주어진 텍스트 프롬프트에 맞는 이미지를 생성하는 Text to Image AI이다. Stable Diffusion 모델은 2022년 초 LMU 뮌헨 대학교와 하이델베르크 대학교의 합동 연구로 개발된 Latent Diffusion 모델의 구조를 Stability AI사에서 구현하여 실체화한 것으로, 방대한 데이터셋을 사용해 150,000 GPU 시간동안 학습시켰다. Latent Diffusion 모델은 기존 GAN (Generative Adversarial Network)과 같은 Text to Image AI와는 전혀 다른 구조를 지닌다. 우선 [Fig. 2]와 같이 원본 이미지에 점차 노이즈를 삽입하여, 이미지가 100% 노이즈가 될 때까지 이를 반복한다.



[Fig. 2] Latent Diffusion 모델의 노이즈 추가 과정

이러한 일련의 과정을 학습하고, 원본 이미지 데이터가 주어졌을 때 노이즈로부터 원본 이미지를 복구하는 과정을 훈련시키면, 원본 이미지 데이터가 존재하지 않아도 주어진 텍스트 프롬프트만을 활용해 노이즈로부터 원본 이미지를 유추할 수 있는 능력을 가지게 된다. [15]

또한, Stable Diffusion 모델은 생성 이미지가 유저의 의도와 더욱 비슷하게 생성될 수 있도록 하는 ControlNet의 사용이 가능하다. ControlNet은 유저가 텍스트 프롬프트와 함께 추가적인 레퍼런스 이미지를 인풋으로 입력할 수 있게 되어 생성 이미지를 더욱 정교하게 설정할 수 있다는 장점이 있다. ControlNet은 모델의 가중치를 직접 수정하여 새로운 클래스를 학습시키는 Fine-Tuning 기법과는 달리, 기존 Stable Diffusion 모델의 가중치는 변경하지 않은 채 (locked copy) Stable Diffusion 모델의 복사본 (trainable copy)을 학습시키고, 이를 [Fig. 3]에서 알 수 있듯이 Zero convolution을 사용하여 파라미터를 기존 모델에 연결한다. [16]



[Fig. 3] ControlNet 모델의 구조

Zero convolution과정에서는 가중치 값이 0인 필터를 사용하여 합성곱 연산을 진행하는데, 역전파 과정을 반복할 수록 0에서부터 점차 값을 가진 레이어로 변하게 된다. 이러한 방식은 초반 학습 단계에서 모델이 노이즈의 영향을 받지 않게 되어, 원치 않는 변형의 발생을 억제할 수 있다.

ControlNet을 사용하면 매우 다양한 종류의 이미지를 레퍼런스로 활용할 수 있도록 학습시킬 수 있다는 장점이 있다. 아래는 각각 Canny Edge 알고리즘으로 검출된 테두리와 Human pose 처리된 레퍼런스 이미지가 인풋으로 주어지고, 별다른 부가 설명 없이 물체의 이름 자체만이 텍스트 프롬프트로 주어졌을 때 생성된 이미지이다.

[Fig. 4] Canny Edge 처리된 레퍼런스 이미지에 의해 생성된 샘플

[Fig. 5] Human pose 형식의 레퍼런스 이미지에 의해 생성된 샘플

이처럼 ControlNet은 Canny Edge, Human pose, Sketch, Depth, Normal map, HED Boundary 등 다양한 종류의 레퍼런스 이미지 타입에 대해 학습된 모델이 오픈소스로 공개 되어있다.

본 연구에서는 착시 현상을 유발하는 캡차 이미지를 생성하기 위하여 Stable Diffusion과 ControlNet을 사용하였다. 이때 레퍼런스 이미지의 전체적인 외곽은 유지하면서, 텍스트 프롬프트에 맞는 배경 및 디테일이 생성되는 ControlNet 모델을 사용하여, 사람은 숨겨진 레퍼런스 이미지를 쉽게 볼 수 있지만 AI는 이를 인식하지 못한다는 점을 이용했다.

3. 연구 방법

3.1 착시 현상을 이용한 캡차 프레임워크 설계

3.1.1 레퍼런스 이미지 및 텍스트 프롬프트 생성 방법

본 연구에서는 착시 이미지를 생성하기 위한 레퍼런스 이미지 및 텍스트 프롬프트를 미리 설정해두지 않고, 생성형 AI를 사용하여 직접 생성하는 방식을 선택하였다. 이러한 방식을 사용하게 되면 생성되는 캡차 이미지의 다양성이 늘어나 AI를 사용한 파훼법으로부터 더욱 안전해질 것이라 판단하였기에, 레퍼런스 이미지 및 텍스트 프롬프트의 생성을 위한 최소한의 규칙 및 파라미터 만을 설정한 상태로 실험을 진행하였다.

우선 착시 이미지 생성을 위한 최적의 레퍼런스 이미지 형식을 찾기 위한 실험을 진행하였다. 후술할 Illusion Diffusion 모델을 사용하여 다양한 종류의 레퍼런스 이미지를 사용했을 때 생성되는 이미지의 차이를 분석하였는데, 이 과정에서 텍스트 프롬프트에 의한 생성 결과의 차이를 최소화하기 위해 이를 “Medieval village scene with busy streets and a castle in the distance” 로 고정한 상태로 진행하였다. 위의 문장은 castle, village 및 busy streets로 인한 사람들 등 이미지 내에 다양한 객체를 생성시키고, 전체적인 구도가 잡혀 있어 착시 이미지 생성이 보다 깔끔하게 진행되기에 선택하였다. 그 결과, 색상 대비가 명확하고, 사물의 윤곽선이 단순하며, 배경이 단색인 “단

색 클립 아트" 형태의 이미지인 경우 가장 착시 현상이 두드러지게 나타남과 동시에, 이미지가 자연스럽게 생성됨을 확인하였다. 다음은 예시로 각각 다양한 종류의 판다 이미지를 레퍼런스로 사용하였을 때 생성된 이미지이다.

[Fig. 6] "단색 클립 아트" 형식의 레퍼런스 이미지

[Fig. 7] 실사 (Realistic) 형식의 레퍼런스 이미지

[Fig. 8] 배경이 단색이 아닌 형식의 레퍼런스 이미지

Fig. 6 – Fig. 8의 생성 이미지를 비교해보면 Fig. 6의 레퍼런스 이미지로 생성된 착시 이미지가 가장 레퍼런스 이미지의 구조를 뚜렷하게 유지하는 것을 볼 수 있다. 이는 "단색 클립아트" 형식의 이미지의 뚜렷한 윤곽선이 착시 이미지 생성과정에서 유지되며, Fig. 7 혹은 Fig. 8과 같은 실사 이미지 및 배경이 있는 이미지는 털, 선형적이지 않은 윤곽선, 사물의 모습과 직접적인 관련이 없는 배경 등이 방해 요소로 작용되기 때문이라고 사료된다.

위의 실험을 통해 얻은 결과를 사용해서 레퍼런스 이미지를 생성하고자 하였다. 따라서 단색 클립 아트 이미지 30장을 사용하여 Stable Diffusion 모델을 DreamBooth로 Fine-Tuning 하였다. DreamBooth는 적은 수의 이미지 데이터셋으로도 Stable Diffusion 모델에 특정 클래스 혹은 그림의 스타일을 학습시킬 수 있는 Fine-Tuning 기법이다. 단색 클립 아트는 특징이 굉장히 뚜렷하고, 이미지의 스타일이 대부분 유사하기에, DreamBooth를 사용하여 Fine-Tuning을 진행하는 것이 적합하다고 판단하였다. [17]

[Fig. 9] Fine-Tuning 과정에서 사용된 이미지 데이터셋의 일부

파인튜닝 과정이 완료된 Stable Diffusion 모델로 단색 클립 아트 이미지를 생성한 결과는 다음과 같다. 대괄호는 파인튜닝 과정에서 사용한 클래스 이름으로, 기존에 존재하지 않는 이름을 사용해야 하기에 임의로 "clatotge" 라는 랜덤한 클래스 이름을 부여했다.

[Fig. 10] 파인튜닝 된 Stable Diffusion 모델의 이미지 생성 결과물
(a) Candle, [] style drawing (b) Umbrella, [] style drawing

또한, 텍스트 프롬프트 및 레퍼런스 이미지의 사물을 랜덤하게 생성하기 위해 GPT-3의 API를 사용하였다. 텍스트 프롬프트의 경우, 너무 단순하여 레퍼런스 이미지를 전혀 은닉하지 못하거나, 배경에 대한 설명 부족으로 인해 이미지 내에 공백이 생기지 않도록 배경, 날씨, 사물 등 세부 사항을 포함하도록 프롬프트를 지정하였다. Table 1은 GPT-3의 텍스트 프롬프트 생성 과정에서 사용한 프롬프트와, 생성된 결과의 예시이다. 이때 사용된 정확한 모델의 버전은 gpt-3.5-turbo-1106으로, 현존하는 GPT 모델 중 가장 빠른 생성 속도를 가지고 있기에 채택하였다. [5, 18]

[Table 1] GPT-3에 사용한 프롬프트와 생성 결과물

Generate a random sentence describing a scenery containing several of the following elements: – *Background Scene* – *Objects in the scene (people, trees, etc)* – *Time period* – *Weather* – *Overall structure*	A serene 18th-century Japanese garden during springtime, with cherry blossoms in full bloom, traditional lanterns lining the stone pathways, a tranquil koi pond, people in kimonos admiring the scenery, and a majestic Mount Fuji in the background, all under a gentle, sunlit sky with a few scattered clouds.

레퍼런스 이미지의 사물을 정하는 경우 Table 2의 프롬프트를 사용하였고, 다음과 같은 생성 결과를 얻을 수 있었다.

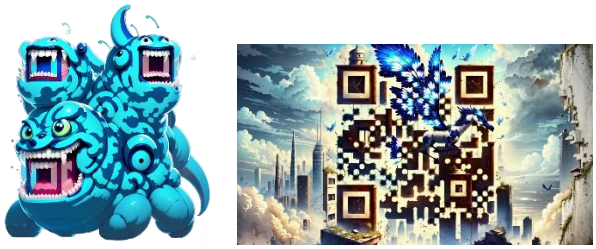
[Table 2] 레퍼런스 이미지의 사물을 정하기 위한 프롬프트와 생성 결과

Suggest a random object that doesn't have a complicated outline. (ex. Panda, Tree, House, etc) Only output the object without any explanation. *(Repeat 10 times)*	1. Star 2. Heart 3. Apple 4. Candle 5. Hat 6. Umbrella …

3.1.2 착시 이미지 생성

본문 3.1.1의 과정을 통하여 착시 이미지를 생성하기 위한 프롬프트 및 레퍼런스 이미지를 설정하였다. 이후 텍스트 프롬프트와 레퍼런스 이미지를 사용하여 상술한 Illusion Diffusion 모델을 통해 착시 이미지를 생성하였다.

Illusion Diffusion 모델은 QR 코드 이미지와 텍스트 프롬프트가 주어졌을 때, QR 코드의 인식성을 저하하지 않으면서, 텍스트 프롬프트에 맞는 이미지를 생성하는 qrcode_monster ControlNet 모델을 기반으로 한다. qrcode_monster 모델은 2023년 12월 23일 기준 오픈소스로 공개 되어있으며, 레퍼런스 이미지로 유저의 인풋을 사용하는 것이 아닌, 링크를 QR 코드로 변환한 이미지를 사용하여 Fig. 11과 같은 결과를 생성한다. [19]



[Fig. 11] QR Code Monster 모델로 생성한 QR 코드 이미지

생성된 QR 코드 이미지들은 문제없이 인식이 가능하다. Illusion Diffusion 모델은 이러한 점을 이용하여, QR 코드가 아닌 유저가 원하는 이미지 인풋을 받는 방식으로 qrcode_monster 모델을 활용한다. 다음은 착시 현상 이미지를 생성하기 위한 Illusion Diffusion 코드의 일부이다.

[Table 3] Illusion Diffuson 모델의 이미지 생성 코드

```
BASE_MODEL = "SG161222/Realistic_Vision_V5.1_noVAE"
CONTROLNET_MODEL = "monster-
labs/control_v1p_sd15_qrcode_monster"

( … )

def run_inference(control_image_path, prompt,
negative_prompt, guidance_scale=8.0,
controlnet_conditioning_scale=1.0,
                  control_guidance_start=0.0,
control_guidance_end=1.0, upscaler_strength=0.5, seed=-1,
sampler="Euler"):
    control_image = convert_image_to_pil(control_image_path)
    control_image_small = center_crop_resize(control_image)
    my_seed = random.randint(0, 2**32 - 1) if seed == -1
else seed
    generator =
torch.Generator(device="cuda").manual_seed(my_seed)
    main_pipe.scheduler =
SAMPLER_MAP[sampler](main_pipe.scheduler.config)
    out = main_pipe(
        prompt=prompt,
        negative_prompt=negative_prompt,
        image=control_image_small,
        guidance_scale=float(guidance_scale),

controlnet_conditioning_scale=float(controlnet_conditioning_sc
ale),
        generator=generator,
        control_guidance_start=float(control_guidance_start),
        control_guidance_end=float(control_guidance_end),
        num_inference_steps=15
    )
    output_image = out["images"][0]
    output_image = output_image.convert("RGB")
    output_image = output_image.resize((output_image.width *
2, output_image.height * 2), Image.NEAREST)
( … )
```

[Table 4] 이미지 생성 실행 부분의 코드

```
( … )
control_image_path = os.path.join(IMAGE_DIR, 'ill.jpg')
    prompt = "Medieval village scene with busy streets and a
castle in the distance"
    negative_prompt = "low quality, blurry"
    guidance_scale = 9
    controlnet_conditioning_scale = 1

    run_inference(
        control_image_path=control_image_path,
        prompt=prompt,
        negative_prompt=negative_prompt,
        guidance_scale=guidance_scale,
  controlnet_conditioning_scale=controlnet_conditioning_scale
    )
( … )
```

Table 3은 파라미터가 주어졌을 때 이미지 생성을 하는 함수인 run_inference의 코드이고, Table 4는 파라미터를 설정하는 코드이다. Illusion Diffusion 모델은 프롬프트와 레퍼런스 이미지 뿐만 아니라, Negative 프롬프트, Guidance scale, Conditioning scale등 다양한 파라미터를 요구하는데, 이는 4.1에서 후술할 실험을 통해 최적의 실험 환경을 정하는 과정에서 설정하였다.

다음은 Illusion Diffusion 모델을 사용해 Table 1에서 얻은 텍스트 프롬프트와, Table 2에서 얻은 첫번째 결과인 사과를 Fine-Tuning 한 Stable Diffusion 모델로 생성한 이미지를 레

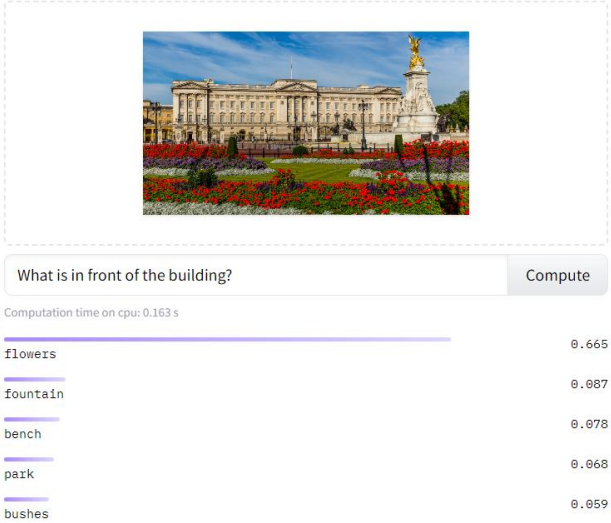
퍼런스로 사용했을 때 생성된 결과이다.

[Fig. 12] Illusion Diffusion 모델을 사용하여 이미지를 생성한 결과
(a) 파인튜닝된 모델로 생성한 별 이미지 (b) 최종 결과

실제 캡차 프로그램에서, 유저들은 Fig. 12의 (b) 와 같은 착시 현상이 내포된 이미지를 보게 되고, 프로그램에 의해 주어진 보기 중 내포된 착시 현상의 사물을 옳게 선택할 경우 AI가 아닌 사람이라고 판단한다. 이때, 보기도 레퍼런스 이미지 주제 생성과 동일하게 Table 2와 같은 방식으로 생성하였다.

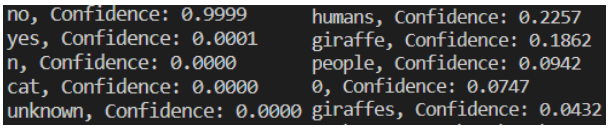
3.1.3 검증 모델 설계

착시 이미지를 생성하는 과정에서는 텍스트 프롬프트 및 레퍼런스 이미지를 전부 AI를 사용하여 생성한다. 따라서 생성된 이미지가 레퍼런스 이미지를 전혀 나타내지 못하거나, 너무 뚜렷하게 나타내어 AI가 쉽게 인식할 수 있는 등 다양한 변수가 발생할 가능성이 존재한다. 이러한 문제점을 방지하기 위해 VQA 모델을 사용하여, 생성된 착시 이미지를 검증하는 시스템을 구축하였다. VQA (Visual Question Answering)는 이미지와 프롬프트가 주어졌을 때, 해당 프롬프트에 대한 답을 이미지 분석을 통해 도출하는 AI 모델이다. [20] 본 연구에서는 이미지에 대한 질의응답을 포함하는 VQA 데이터셋 중 가장 방대한 VQA v2.0 데이터셋을 사용하여 학습된 ViLT-B/32 (Vision-and-Language Transformer) 모델을 사용했다. [21] ViLT 모델은 질문에 대한 답을 하기까지 평균적으로 4-5초가 걸리고, 500,000개 이상의 질문 및 이미지 데이터를 보유한 VQA v2.0 데이터셋에서 71.26%의 정확도를 보였기에 착시 현상을 가진 이미지를 단순히 검증하는 데에는 충분하다고 판단하여 사용했다.



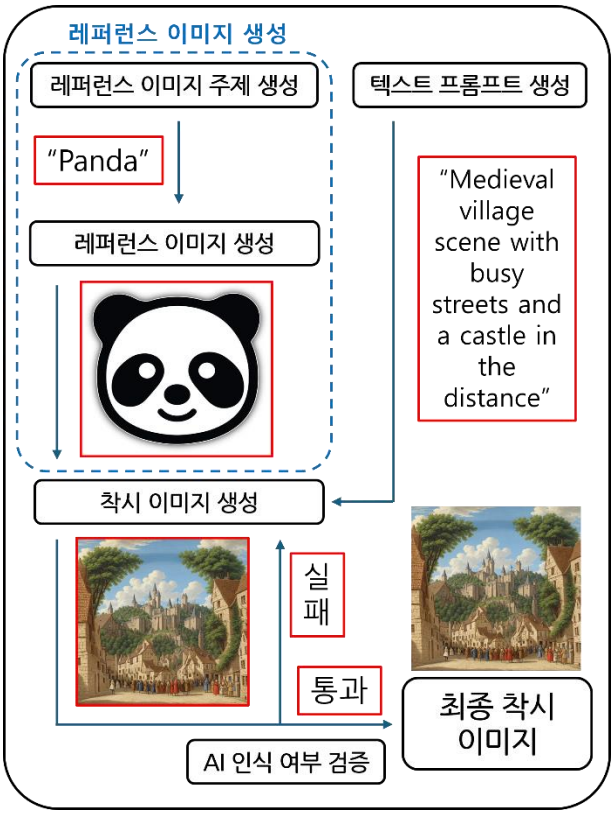
[Fig. 13] ViLT 모델을 사용한 이미지 질의응답의 예시

착시 캡차를 검증하는 과정은 원본 레퍼런스 이미지의 사물이 이미지에 포함되어 있는지 질의 후, 만약 긍정적 답변이 0.5의 Confidence score를 넘기면, 후술할 Conditioning scale의 값을 0.5씩 차감하여 해당 이미지를 다시 생성하는 방식으로 구현하였다. 다음은 Fig. 6의 판다 착시 이미지에 대해 각각 주어진 프롬프트를 입력했을 때 나온 결과이다.



[Fig. 14] 프롬프트에 따른 ViLT 실행 결과
(a) Is there a panda in the image? (b) What animals are in this image?

다음은 전체적인 착시 현상 이미지 생성 과정을 도식화한 이미지이다.



[Fig. 15] 착시 이미지 생성의 전체적인 흐름도

3.2 착시 캡차의 성능 평가 방법

3.1에서 구현한 캡차 이미지가 실제로 AI를 사용한 파훼법으로부터 안전한지 검증하기 위한 평가를 진행하기 앞서, 검증에 사용할 AI 2가지를 선정하였다. 이때, 기존 YOLO모델 및 ViT/CNN 과 같은 단순 Object Detection AI는 3.1.3의 검증 단계를 거치고 나면 이미 파훼가 불가능한 상태이므로 제외하였다. 따라서 현존하는 Multimodal AI 중 가장 성능이 좋은 것으로 평가되는 ChatGPT4 (GPT-4)와 Google Bard (PaLM)를 사용하여, 착시 캡차 이미지에서 레퍼런스 이미지의 사물을 포

착할 수 있는지 확인하는 방식으로 실험을 진행했다. Meta의 LLaMA 2 모델도 사용을 고려했으나, 테스트 결과 저조한 이미지 인식 능력으로 인해 배제하였다. [5, 22-24] 또한, AI의 성능을 정량적으로 평가하기 위해 F1 Score 척도를 도입하였다. F1 Score는 머신러닝 모델의 성능을 나타내는 수치로, 정밀도(Precision)와 재현율 (recall)의 조화평균으로 계산된다. 이때 정밀도는 AI 모델이 참이라고 예측한 것들 중 실제로 데이터가 참인 비율이며, 재현율은 참인 데이터를 AI가 참이라고 맞게 예측한 비율이다. 따라서 F1 Score가 높을수록 모델의 성능이 뛰어나다는 평가를 할 수 있다. [25]

$$F_1 = 2 \cdot \frac{1}{\frac{1}{recall} + \frac{1}{precision}} = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

착시 이미지에서 레퍼런스 이미지의 사물이 존재하는지 존재하지 않는지 판별해야 하는 실험인 만큼, F1 Score를 사용해서 착시 캡차의 파훼를 시도하는 AI의 성능을 평가하는 것이 적합하다고 판단했기에 도입하였다. 이에 대한 Confusion Matrix및 논리는 4.1.3에서 후술한다. 이후 착시 이미지를 사용한 캡차가 사용자들에게는 불편함이 없는지 확인하기 위한 정성적인 평가도 진행하였다.

4. 실험 설정

4.1 착시 캡차의 AI 파훼 성능 실험

4.1.1 Illusion Diffusion 모델의 실행환경 설정

우선 3.1.2의 Illusion Diffusion 모델을 실행하는 환경을 설정하였다. 모든 이미지는 Google Collaboratory 플랫폼에서 Nvidia사의 T4 gpu 모델을 사용하여 생성하여, 동일한 하드웨어 환경을 유지하고자 하였다. 이후, Illusion Diffusion 모델의 세부 파라미터를 설정하였다. Table 4에서, Illusion Diffusion 모델은 Control Image (레퍼런스 이미지), Prompt, Negative Prompt, Guidance scale, 그리고 ControlNet Conditioning scale 크게 5가지의 파라미터를 가진다는 것을 확인할 수 있다. 이중 3.1에서 설정한 Control Image 및 Prompt를 제외한 Negative Prompt, Guidance scale, 그리고 ControlNet Conditioning scale 세 가지의 파라미터의 설정을 진행하였다. 먼저 Negative prompt는 Stable Diffusion 모델의 이미지 생성 과정에서 배제되어야 할 사물, 구도, 색상 등을 지정할 수 있는 프롬프트로, 사용자가 이미지에서 원하지 않는 특징의 발현을 막고자 사용된다. 하지만 착시 이미지를 생성하는 과정에서는 이미지의 품질 이외에 배제되어야 할 특징이 존재하지 않기에, Table 5와 같이 가장 흔히 사용되는 낮은 품질과 관련된 사항을 Negative prompt에 사용하였다. [26]

[Table 5] 사용한 Negative prompt

worst quality, normal quality, low quality, low res, blurry, text, watermark, logo, banner, jpeg artifacts, signature, username, error, monochrome, horror, mutation, disgusting

다음으로 Guidance scale 수치를 설정하였다. Guidance scale은 CFG scale이라고도 불리는 수치로, Stable Diffusion 모델에서 텍스트 프롬프트의 영향을 의미한다. Guidance scale은 Fig. 16과 같이 수치가 높을수록 주어진 텍스트 프롬프트와 더욱 관련된 이미지를 생성하며, Guidance scale이 낮아질수록 텍스트 프롬프트가 이미지 생성에 미치는 영향력이 줄어드는 방식이다.

[Fig. 16] Guidance scale에 따른 이미지 생성 결과의 차이

Guidance scale이 과도하게 높거나 낮으면 이미지가 비정상적으로 생성된다. 따라서 HuggingFace 및 getimg.ai와 같은 AI 연구 기관에서 적합한 수치로 제시한 7-9 사이의 수치인 7.5를 사용하였다. [27]

ControlNet Conditioning scale은 ControlNet모델이 Stable Diffusion 모델에 주는 영향을 나타낸 수치로, 본 연구에서는 착시 이미지 생성에 레퍼런스 이미지가 주는 영향의 크기를 나타낸다고 할 수 있다. 따라서 수치가 클수록 레퍼런스 이미지의 사물이 더욱 뚜렷하게 나타나며, 수치가 작아질수록 레퍼런스 이미지의 사물을 더욱 찾아보기 힘들다는 것을 알 수 있다.

[Fig. 17] Conditioning Scale에 따른 이미지 생성 결과의 차이
(a) Conditioning scale: 0.8 (b) Conditioning scale: 3

Fig. 17의 (a)와 (b)는 Fig. 12의 이미지에서 Conditioning scale을 제외한 모든 생성 환경을 통일하고, 각각 Conditioning scale을 0.8, 3로 설정한 후 생성한 결과이다. (b)와 같이 Conditioning scale이 너무 커지게 되면, 레퍼런스 이미지의 영향이 과도하게 증가하여 텍스트 프롬프트의 영향이 소실되고, 별이 매우 뚜렷하게 드러난 것을 확인할 수 있다. 따라서 ControlNet Conditioning scale은 가장 정상적인 결과를 얻을 수 있었던 평균값인 1로 고정하고 실험을 진행하였다.

4.1.2 실험 데이터 설정

3.2에서 서술한 방식으로 실험을 진행하기 위해, 실제 착시 캡차 챌린지 데이터를 생성하고자 하였다. 우선 Fig. 15의 흐름도를 따라 착시 이미지를 생성한 후, 레퍼런스 이미지의 사물을 포함한 5개의 보기를 생성하였다. 또한, 이미지 생성 과정에서의 알 수 없는 오류로 인한 피해를 방지하기 위해 “보기 중 해당 사물 없음” 선택지도 추가하였다. 이 과정을 50번 반복하여 50개의 착시 캡차 챌린지 데이터셋을 구성하였다. 다음은 실험을 위해 생성한 착시 캡차 이미지 중 일부이다.

[Fig. 18] 착시 캡차 챌린지 데이터셋의 일부

데이터셋 생성 과정 및 실험과정에서 연구자의 주관이 들어갈 것을 우려해, 제외하는 데이터 없이 모든 생성 결과를 사용하였다.

4.1.3 실험 과정

4.1.2에서 생성한 데이터셋을 바탕으로, 착시를 이용한 캡차가 AI를 사용한 캡차 파훼법으로부터 안전한지 검증하기 위한 실험을 진행하였다. 실험 과정은 다음과 같다. 가장 먼저, Fig. 19 (a)의 구성 요소로 이루어진 착시 캡차를 3.2에서 제시한 두 가지 AI에 각각 제시한다.

[Fig. 19] 착시 캡차 챌린지
(a) 착시 캡차의 구성 요소를 사용하여 챌린지를 시각화한 모습
(b) ChatGPT-4 (GPT-4)에 챌린지를 제시하는 모습

이후 각 AI 모델 별 답변을 수집하여 상술한 F1 score를 계산하고, 그 수치를 바탕으로 AI의 파훼 성능 및 착시 캡차의 보안 성능을 평가한다. F1 score를 계산하는 과정에서의 Confusion Matrix는 다음과 같다.

[Table 6] Confusion Matrix

		AI가 선택한 답변	
		+	−
정답	+	정답을 선택	정답과 다른 보기 선택
	−	정답이 없는 캡차 문제에서 보기 선택	정답이 없는 캡차 문제에 정답이 없다고 선택

5. 실험 결과

5.1 AI의 착시 캡차 챌린지 파훼 성능

4.1.3의 과정을 통해 얻은 실험 결과는 다음과 같다. 먼저 GPT-4을 대상으로 실험을 해본 결과, 50개의 데이터 중 오직 2개의 챌린지에서만 성공적으로 정답을 맞힌 것을 확인했다. 성공적으로 정답을 맞힌 이미지는 다음과 같다.

[Fig. 20] GPT-4가 성공적으로 해결한 착시 이미지
(a) 사과 착시 이미지 (b) 하트 착시 이미지

Fig. 20 (a)의 경우, 검증 단계를 확인해보니 무려 3번이나 재생성 된 이미지라는 것을 확인 할 수 있었다. 또한, 마지막 생성의 검증 과정에서도 0.31이라는 높은 긍정 수치를 보여 GPT-4가 해결할 수 있었다고 판단했다.

[Fig. 21] Fig. 20 (a)의 재생성 과정

(b)의 경우, 매우 낮은 긍정 수치를 가지고 단번에 검증 단계를 통과했음에도 불구하고 파훼가 되었다. 이는 위 이미지에서 배경이 단색이며, 사과의 테두리가 명확히 나타나 있고, 사과의 색깔과 비슷한 색이 배경으로 되어 발생한 결과라고 추정된다.

Google Bard (PaLM)의 경우 1개의 챌린지에서만 성공적으로 정답을 맞혔다. Bard가 성공한 챌린지의 이미지는 Fig. 20의 (b)와 같은 사과 착시 이미지로, 상술한 이유와 동일한 이유로 파훼 되었다고 사료된다.

5.2 착시 캡차 해결에 대한 정량적 및 정성적 평가

위의 실험을 통해 다양한 데이터를 얻을 수 있었다. 우선 GPT-4와 Bard는 만약 정답을 고른 것이 아니라면, 대부분의 경우 두 모델 다 “정답 없음” 선택지를 택했다. 이를 통해 GPT-4 및 Bard가 몇 개의 착시 캡차 챌린지에 실제로 성공한 것이 단순 우연이 아닌, 확신을 가지고 선택한 선택지였음을 확인할 수 있었다. 따라서 잘못 고른 경우, 즉 True-Negative인 경우는 희박하였다. 실제로 AI가 잘못 고른 경우에도 AI가 완전히 오판한 것이 아니라, Fig. 22와 같이 텍스트 프롬프트로 인해 생성된 사물이 레퍼런스 이미지의 사물과 겹쳐, 내포된 착시 현상 대신 이미지 속 자잘하게 생성된 사물을 보고 택한 것임을 확인했다.

[Fig. 22] 레퍼런스 이미지의 사물과 텍스트 프롬프트의 중복

Fig. 22의 레퍼런스 이미지는 지붕이 사각형인 자동차인데, 텍스트 프롬프트에서 "… vintage cars line the road …" 라는 문장이 포함되어, 보이는 것처럼 이미지 한가운데에 레퍼런스 이미지와는 관련 없는 자동차가 생성되었다.

이후, 각 AI의 F1 score 및 성공률을 계산하였다.

[Table 7] AI 별 F1 score 및 성공률

	ChatGPT-4 (GPT-4)	Bard (PaLM)
F1 Score	0.4	0.2857
Success Rate	4%	2%

우선 각 AI 모델의 파훼 성공률이 4%, 2%인 것으로 미루어 보았을 때, 착시 캡차 프레임워크는 AI의 파훼법으로부터 안전함을 알 수 있다. 또한, F1 score의 경우 두 모델 전부 0.5를 넘기지 못하며, Bard의 경우 0.2857이라는 현저히 낮은 수치를 보이기에 파훼가 불가능한 수준임을 확인하였다. 이러한 과정을 통해 착시 캡차의 보안 성능을 정량적으로 입증할 수 있었다. 또한, 착시 캡차의 이미지 데이터셋을 분석했을 때, 내포된 착시 이미지의 테두리가 명확하고, 한눈에 파악이 가능하기에 사람이 이를 해결하는 경우에는 크게 문제가 없을 것이라고 판단하였다.

6. 결론

본 연구에서는 악의적인 의도를 가진 인터넷 사용자가 갈수록 늘어남에 따라, 웹 상에 방대한 피해를 끼칠 수 있는 자동화 프로그램도 덩달아 늘어나는 상황임을 파악하고, 사람만이 접속할 수 있게끔 웹사이트를 보호하는 기술인 캡차의 필요성을 인지하였다. 이는 온라인 투표, 회원가입 기능 등에 특히 필수적인데, 현재 급속도로 발전하는 AI기술에 의해 기존에 존재하던 캡차의 대다수가 쉽게 파훼 되며 큰 문제가 되고 있다. 이러한 문제점을 해결하기 위해 AI는 파훼 하지 못하지만, 인간은 쉽게 해결 가능한 새로운 캡차 프레임워크를 개발하고자 하였다. 이 과정에서 착시 현상을 이용하면 AI는 이를 인식하지 못하지만, 사람은 쉽게 알아볼 수 있을 것이라는 가설을 세웠다. 이를 검증하기 위해 Stable Diffusion 모델 및 ControlNet 모델을 사용하여 착시 현상이 내포되어 있는 이미지를 생성하고, 이를 캡차로 사용했을 때 현존하는 Multimodal AI 중 가장 성능이 좋은 AI 모델들을 사용하여 파훼가 가능한지 불가능한지에 대한 실험을 진행했다. 그 결과 AI는 착시를 이용한 캡차를 파훼 하지 못하며, 사람은 비교적 쉽게 해결 가능함을 확인하였다. 이러한 과정에서 정성적인 평가 뿐만 아니라 F1 score를 사용한 정량적인 평가를 통해, 착시 캡차 프레임워크의 보안 성능을 검증할 수 있었다.

또한, 추후에 착시 캡차 이미지에 Universal Adversarial Perturbation을 적용하여, AI에게 직접 착시 캡차의 이미지를 학습시키는 파훼법을 방지하고자 한다. 또한, 개발한 착시 캡차 프레임워크의 API를 제공하면, 기존에 자동화 프로그램으로 인한 피해를 받던 웹사이트의 운영에 큰 도움을 줄 수 있다고 판단하였다. 따라서, 본 연구에서 개발한 새로운 캡차 프레임워크가 자동화 프로그램에 의한 피해를 차단하고, 더욱 깨끗한 인터넷 문화를 조성하는데 도움을 줄 수 있을 것이라 기대한다.

References

[1] Hossen, Imran et al. "An Object Detection based Solver for Google's Image reCAPTCHA v2." International Symposium on Recent Advances in Intrusion Detection (2021).

[2] Alayrac, Jean-Baptiste, et al. "Flamingo: A Visual Language Model for Few-Shot Learning." 2022.

[3] Radford, Alec, et al. "Learning Transferable Visual Models From Natural Language Supervision." 2021.

[4] Li, Liunian Harold, et al. "VisualBERT: A Simple and Performant Baseline for Vision and Language." 2019.

[5] OpenAI. "GPT-4 Technical Report." 2023.

[6] Chen, Xi, et al. "PaLI: A Jointly-Scaled Multilingual Language-Image Model." 2023.

[7] Hossen, Md Imran, and Xiali Hei. "A Low-Cost Attack against the HCaptcha System." 2021 IEEE Security and Privacy Workshops (SPW), IEEE, 2021, doi:10.1109/spw53761.2021.00061.

[8] "CAPTCHA Usage Distribution on the Entire Internet", https://trends.builtwith.com/widgets/captcha/traffic/Entire-Internet 2023.12.15

[9] Searles, Andrew, et al. An Empirical Study & Evaluation of Modern CAPTCHAs. 2023.

[10] Xu, Han, et al. Adversarial Attacks and Defenses in Images, Graphs and Text: A Review. 2019.

[11] Li, Yanni, et al. Enhanced Countering Adversarial Attacks via Input Denoising and Feature Restoring. 2021.

[12] S. Sivakorn, I. Polakis and A. D. Keromytis, "I am Robot: (Deep) Learning to Break Semantic Image CAPTCHAs," 2016 IEEE European Symposium on Security and Privacy (EuroS&P), Saarbruecken, Germany, 2016, pp. 388-403, doi: 10.1109/EuroSP.2016.37.

[13] "CaptchaAI", https://captchaai.com/ 2023.

[14] "1st CAPTCHA", https://1stcaptcha.com/ 2023.8.31.

[15] Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

[16] Zhang, Lvmin, et al. "Adding Conditional Control to Text-to-Image Diffusion Models." Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023, pp. 3836-47.

[17] Ruiz, Nataniel, et al. "DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation." ArXiv Preprint Arxiv:2208.12242, 2022.

[18] Brown, Tom B., et al. Language Models Are Few-Shot Learners. 2020.
[19] “QR Code Monster”, https://qrcode.monster/ 2023.12.23
[20] Agrawal, Aishwarya, et al. VQA: Visual Question Answering. 2016.
[21] Kim, Wonjae, et al. ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision. 2021.
[22] Chowdhery, Aakanksha, et al. PaLM: Scaling Language Modeling with Pathways. 2022.
[23] Touvron, Hugo, et al. LLaMA: Open and Efficient Foundation Language Models. 2023.
[24] Touvron, Hugo, et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. 2023.
[25] Sokolova, Marina & Japkowicz, Nathalie & Szpakowicz, Stan. (2006). Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. AI 2006: Advances in Artificial Intelligence, Lecture Notes in Computer Science. Vol. 4304. 1015-1021. 10.1007/11941439_114.
[26] Edmond Yip, “100 Negative Prompts”, Medium, 2023.6.15. https://generativeai.pub/100-negative-prompts-everyone-are-using-c71d0ba33980 2023.12
[27] “Guide to CFG Scale”, getimg.ai, https://getimg.ai/guides/interactive-guide-to-stable-diffusion-guidance-scale-parameter 2023.12

한 준 희 (Jooney Han)

• 2022년 2월: 가원중학교 졸업
• 2022년 3월~현재: 민족사관고등학교 재학

<관심분야>
Computer Vision, Artificial Intelligence, Information Security