|  |
| --- |
| [project proposal] |
| 아이디어 제안서 |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| **Team name** | R.U.Robot | |
| **Professor** | 손태식 | |
| **Project Manager** | 김형호 | 201620624 |
| **Team Member** | 이상민 | 201420673 |
| 조영준 | 201620617 |
| 송연주 | 201620642 |
|  |  |

The table of contents

**1. 전체 요약** 3

[**2.** **개발 목표** 4](#_Toc274482971)

[**3.** **동기 및 기대효과** 5](#_Toc274482972)

[**3.1** **기술적 기대효과** 5](#_Toc274482974)

[**3.2** **사회적 기대효과** 5](#_Toc274482982)

[**4.** **기술 동향** 6](#_Toc274482973)

[**4.1** **제품** 6](#_Toc274482974)

[**4.1.1** **인공지능 악용 사례** 6](#_Toc274483000)

[**4.1.2** **Invisible reCaptcha V3** 7](#_Toc274483000)

[**4.1.3** **이미지-텍스트 융합 Captcha** 7](#_Toc274483000)

[**4.2** **차별성 설명** 8](#_Toc274482975)

[**5.** **예상 결과물 및 요구조건 분석** 9](#_Toc274482976)

[**5.1** **시스템 구현 범위 (System boundary)** 9](#_Toc274482982)

[**5.2** **사용자 서술** 9](#_Toc274482983)

[**5.3** **입출력 (input/output)** 9](#_Toc274482984)

[**5.4** **기능적 요구사항 (functional requirement)** 9](#_Toc274482987)

[**5.5** **비기능적 요구사항 (nonfunctional requirement)** 10](#_Toc274482988)

[**5.5.1** **제품 요구사항** 10](#_Toc274483000)

[**5.5.2** **조직 요구사항** 10](#_Toc274483000)

[**5.5.3** **외부 요구사항** 10](#_Toc274483000)

[**6.** **개발 방법** 12](#_Toc274482989)

[**6.1** **개발 환경** 12](#_Toc274482995)

[**6.2** **개발에 활용되는 지식과 기술** 13](#_Toc274482996)

[**6.3** **위험 요소 및 대처 방안** 14](#_Toc274482997)

[**7.** **개발 일정** 16](#_Toc274482998)

[**7.1** **역할 분담** 16](#_Toc274482999)

[**7.2** **개발 일정** 17](#_Toc274483002)

[**8.** **참고 자료** 18](#_Toc274483003)

1. **전체 요약**

A CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is a type of challenge-response test used in computing to determine whether the user is human. However, as a machine learning evolves, this system is being threatened by image recognition and audio recognition. Although, upgraded Captcha such as reCaptcha V3 based on user interaction and image-text fusion captcha are proposed to solve this problem of the previous system, those are not widely used because of low dependability and low user-friendly. This project examines the reliability of Captcha by an Anti-Captcha automation AI (Artificial Intelligence) programme. The project is expected to demonstrate dangers of Artificial Intelligence and also gives vision on the improvement of Captcha.

Along the progress, the project encountered three threats and solved:

First, trained Machine Learning and Deep Learning models has to be stored. The models are stored in programme before released. However, image recognition will be execused in server.

Second, required Library has to be installed before using programme. The solution is including required Library in the programme itself.

Third, processing speed is affected by computing power of user. Whereas non-functional requirement specifies less than seven seconds for a whole processing image/audio recognition trial. The problem can be solved by processing in server so that programme can have controlled constant processing speed by excluding these change factors.

1. **개발 목표**

인공지능을 활용한 안티캡챠(Anti-captcha) 프로그램.

본인인증시에 자동입력방지문자를 인공지능을 활용하여 계산하고 미리 저장되어있던 개인정보를 자동으로 입력하는 프로그램을 개발한다. 본 프로그램을 통해 해당 웹페이지에 접근해 이미지와 오디오를 가져와 머신러닝을 통해 답을 추측한다. 이후 추측해낸 답과 기존에 알고있던 사용자 정보를 웹페이지에 자동으로 입력한다.

1. **동기 및 기대효과**

**3.1 기술적 기대효과**

현재 자동입력 방지문자는 인간의 인지능력을 활용하는 버전3까지 나와있으나 아직 버전3의 활용도는 높지않고, 버전2가 아직도 많이 쓰이고있다. 하지만, 본 프로젝트를 통해 자동입력 방지문자 버전2의 취약성을 알아보고, 자동입력 방지문자의 새로운 패러다임의 필요성을 제고한다. 또한, 프로젝트를 통해 발견한 취약점을 통해 자동입력 방지문자의 발전방향을 제시한다.

**3.2 사회적 기대효과**

인공지능이 발전함에 따라 이를 활용한 제품들이 만들어 지고 있다. 이와 함께 이를 악용하는 사례도 등장하기 시작하고 있다. 또한 정보화 시대가 진행되면서 과거 전문가들만 공유하고 있던 정보들을 일반인들도 손쉽게 접할 수 있게 되며, 인공지능의 경우 마찬가지로 초보자가 간단한 튜토리얼만으로 이를 활용한 응용프로그램을 제작할 수 있게 되었다. 이는 즉 인공지능을 악용하는것 역시 손쉽게 이루어 질 수 있다는 것이다. 그래서 본 프로젝트에서는 전문지식 없이 오픈소스로 공개된 인공지능 코드들을 이용하여 악성 프로그램을 제작하는 것을 목표로 한다. 그리고 이러한 일련의 과정들을 분석하여 추후 손쉽고 무분별하게 악용될 수 있는 인공지능으로부터의 보안성을 높이기 위해 어떠한 작업들이 필요한지 알아본다.

1. **기술 동향**

**4.1 제품**

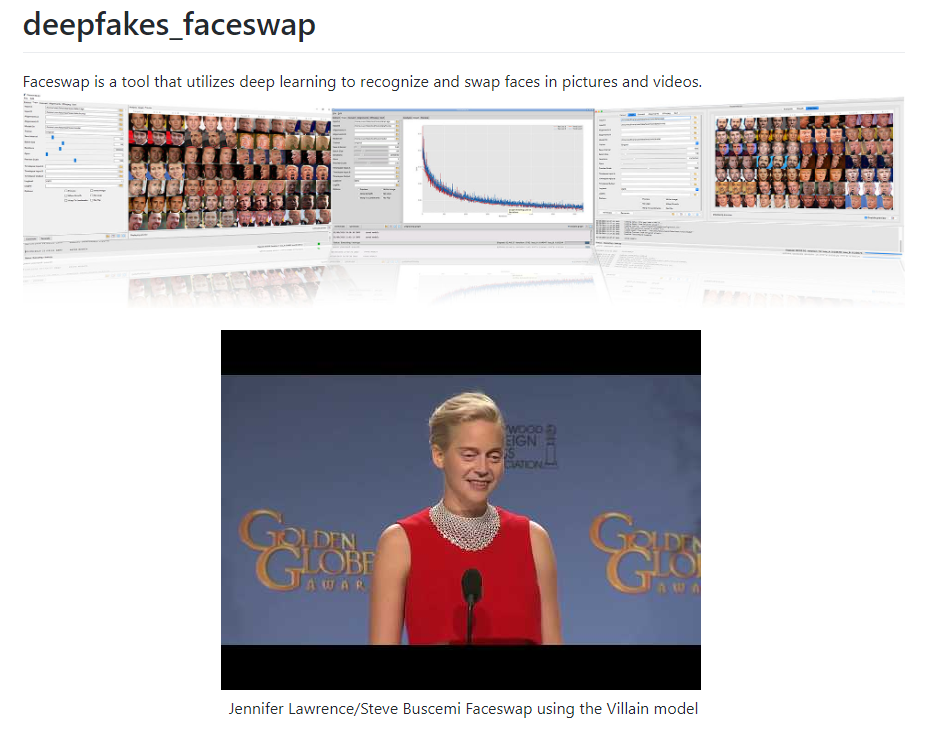
**4.1.1 인공지능 악용 사례**

한국 정보화 진흥원에서는 18년 8월 인공지능 악용에 따른 위협과 대응방안에 대한 보고서를 발표하였다. 보안에 영향을 미치는 인공지능의 6가지 속성으로 이중성, 효율성 및 확장성, 우수성, 익명성, 보급성, 취약성을 들었으며 이미 이러한 속성을 갖고 영향을 미치는 예시들이 생겨나고 있다.



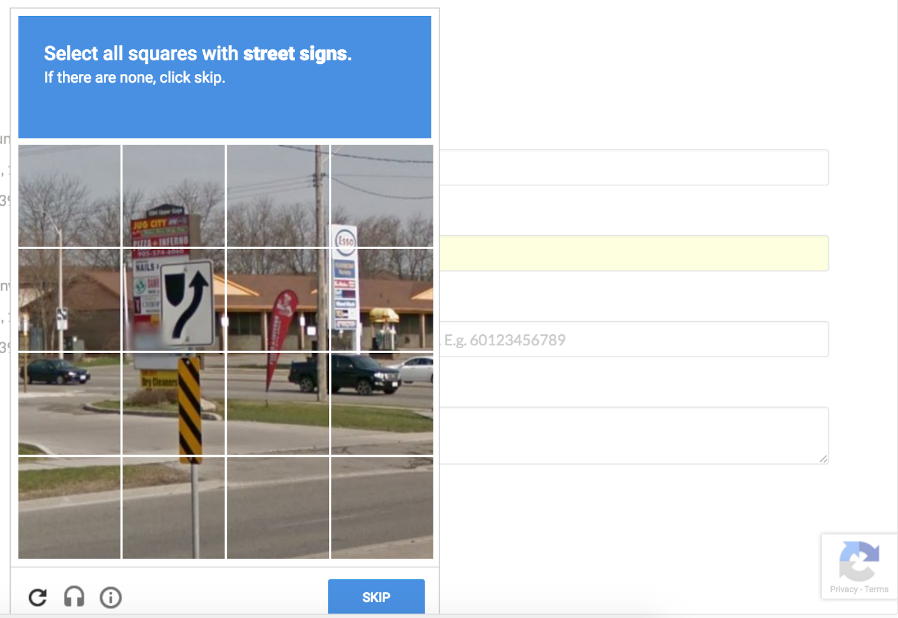
[그림 1] 인공지능 악용에 따른 위협과 대응 방안

여러 사례중 인공지능의 악용 가능성을 심각하게 일깨우며 공식적으로 여러 언론에 보도되었던 사건은 17년 말에 등장한 딥페이크 이다. 성인비디오에 얼굴 스와핑 알고리즘을 적용해 얼굴을 바꾸는 딥페이크는 초기엔 구별이 가능했지만 기술이 점차 확산되며 영상의 진위를 가리기 어려워지게 되었다. 이러한 문제점이 언론에 보도 되자 여러 온라인 사이트에서 가짜 영상물에 대한 제작과 보급을 엄중히 단속하기 시작했지만 해당 소스코드는 Github 에 공개되어 있는 상황이며 일반인들도 쉽게 다운받아 사용할 수 있다.



[그림 2] 딥페이크

**4.1.2 Invisible reCaptcha V3**



[그림 3] Invisible reCaptcha V3

Invisible reCaptcha V3는 사용자의 동작을 분석하여 로봇으로 판단이 되면 이미지 중에 특정 부분을 선택하는 창을 띄워 사람임을 확인하도록 한다. Invisible reCaptcha V3는 사용자들의 로그인 후 브라우징 습관을 이용하여 로봇인지를 파악하지만 자세한 원리는 보안상의 이유로 공개되지 않고있다.

**4.1.3 이미지-텍스트 융합 Captcha**

이미지-텍스트 융합 캡차 시스템은 기존의 이미지 기반 캡차들의 문제점을 보완하기 위해 인하대학교 정보보학과에서 고안해낸 새로운 캡차 방식이다. 이 시스템은 이미지와 캡차 답의 일부를 텍스트 힌트로서 융합시킨 이미지를 캡차 문제로 제공한다.



[그림 4] 이미지-텍스트 융합 캡차

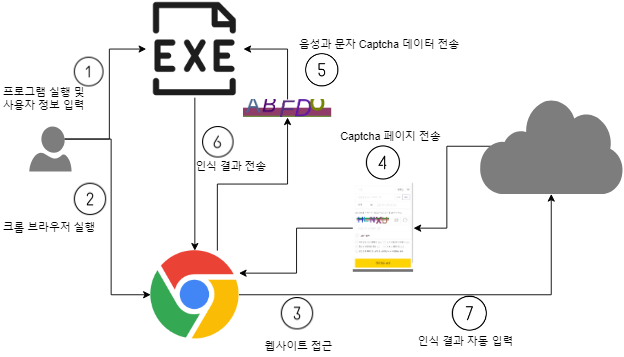
사용자는 주어진 힌트를 가지고 캡차 답을 유추할 수 있다. 그리고 텍스트 기반 캡차와 달리 정답의 일부만을 제공하기 때문에 OCR 소프트웨어 공격에 의해 캡차 답을 인식하지 못하게한다. 또한 인터넷 검색을 통해 이미지 검색을 실시하기 때문에 대량의 이미지를 확보하는 것이 가능하다. 그래서 봇 프로그램의 반복적인 시도를 이용한 시스템의 이미지 데이터베이스 구조파악이 쉽게 될 수 없다. 또한 캡자 답의 일부를 힌트로 부여하기 때문에 사용자의 정답률을 높일 수 있고 캡차 이미지 제공시에 위의 이미지를 전송함으로써 전송 비용의 낭비를 막을 수 있다.

**4.2 차별성 설명**

기존 대부분 연구들의 인공지능 발전 또는 인공지능 파훼라는 1차원적 목표에서 벗어나 인공지능의 위험성과 발전 방향을 제시하는 것을 목표로 삼고 있다.

1. **예상 결과물 및 요구조건 분석**


5. 2. **시스템 구현 범위**



[그림 5] 시스템 구조

* 윈도우 (windows 7 이상)
* 크롬
  + 디버깅이 가능한 Chrome 브라우저
  + Chrome Driver 설치
  1. **사용자 서술**
* 프로그램 및 필수 라이브러리 설치
* 사용자 정보 입력
  1. **입출력**
* 입력 : 사용자 정보 및 크롬과 크롬 드라이버 위치
* 출력 : Captcha 인증 화면에 사용자 정보와 자동입력 방지문자
  1. **기능적 요구사항**
* 사용자 정보 자동 입력
  + 사용자가 사전에 입력한 사용자 개인정보를 필요한 위치에 입력
* 자동입력 방지문자 자동 입력
  + 이미지 혹은 음성파일 자동 다운로드
  + 다운된 파일을 기준으로 문자 추출 후 화면상에 입력
  1. **비기능적 요구사항**
     1. **제품 요구 사항**
* 사용성 요구 사항
  + GUI는 단순하고 직관적(user friendly)이어야 한다.
  + 사용중 오류가 발생한다면 사용자에게 알려야한다.
* 효율성 요구 사항
  + 성능 요구 사항
    - 머신러닝 및 딥러닝 라이브러리 동작을 위한 최소 성능을 만족시켜야 한다.
    - 개인정보 자동입력은 3초 이내로 완료되어야 한다.
    - captcha 문자는 7초 이내로 인식 및 입력이 완료되어야 한다.
  + 공간 요구 사항
    - 프로그램의 실행 파일의 크기는 ??MB보다 작아야한다.
    - 학습 데이터는 그래프 파일에 저장한다.
    - 학습 데이터는 100MB보다 작아야한다.
    - 기능 수행 중이 아닐 때는 컴퓨터 자원을 매우 적게 사용해야한다.
* 신뢰성 요구 사항
  + 자동입력 방지문자의 인식률은 99%이상이어야 한다.
* 이식성 요구 사항
  + 프로그램은 사용자의 웹서비스 사용에 영향을 주지 않아야한다.
    1. **조직 요구사항**
* 배포 요구 사항
* 구현 요구 사항
  + 딥러닝 및 머신러닝 방식 이용 학습
  + python 기반 소스코드 작성
* 표준 요구 사항
  + python 패키지 및 github 표준 규약 준수
    1. **외부 요구사항**
* 상호운용성 요구사항
  + chrome 프로그램과 연동되어 자동으로 본인 인증을 진행해야 한다.
* 윤리적 요구사항
  + 본 프로그램은 연구용으로 범죄에 악용되지 않아야한다.
* 법적 요구사항
  + 사생활 요구 사항
    - 저장된 사용자의 개인정보를 외부로 유출하지 않는다.
  + 안전성 요구 사항
    - 저장된 사용자의 개인정보가 타 사용자에 의해 유출되지 않도록 한다.
    - 사용자의 개인정보는 암호화하여 저장하여야한다(파일 형식).

1. **개발 방법**



6. 1. **개발 환경**

|  |  |
| --- | --- |
| 내용 | 개발 환경 |
| 운영체제 | Windows 10 |
| 개발언어 | python |
| 사용된 주요 패키지 | sklearn(머신러닝), tensorflow(딥러닝), inception(딥러닝), pillow(이미지전처리), scipy(음성전처리), selenium(사용자정보자동화), PyQt5(GUI) |
| 버전 관리 시스템 | Git, SourceTree  <https://github.com/sm0514sm/BreakCaptcha> |

[표 1] 개발환경

|  |  |
| --- | --- |
| 운영체제 | Windows 10 |
| 크롬버전 | 74.0.3729.108 (x64) |
| 크롬드라이버 | 74.0.3729.6 |

[표2] 테스트 환경

* 1. **개발에 활용되는 지식과 기술**

Captcha에서 문자 이미지와 음성 데이터를 불러와 데이터를 전처리하여 머신러닝으로 학습시키기 쉽게 데이터를 가공하는 과정을 거쳐야한다. 먼저 [표 3]은 Pillow 패키지를 이용한 이미지 전처리 과정을 나타내고 있다. 문자 학습을 위해 5글자의 문자들을 개별로 나누는 것과 기계 인식 저하를 위해 생성되어있는 노이즈 라인을 제거하는 것이 이미지 전처리의 주요 목표이다. 이와 비슷하게 [표 4]는 scipy 패키지를 이용한 음성 데이터 전처리 과정을 나타내고 있다. 마찬가지로 5글자의 문자 음들을 개별로 나누는 것과 노이즈 제거 및 모든 길이 일정히 만드는 것이 음성 전처리의 주요 목표이다.

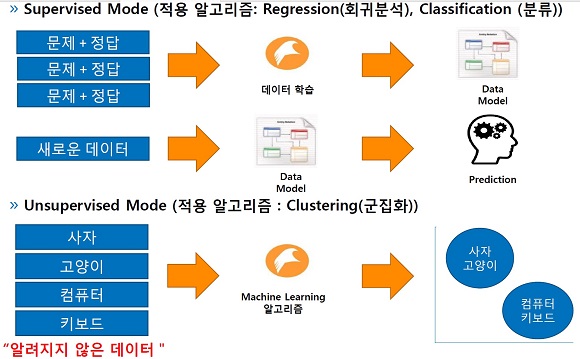
|  |  |
| --- | --- |
| 이미지 | 설명 |
| https://lh5.googleusercontent.com/klll4Jh4Bl7hXz17jetGFwNHV5Ne4VGwOvdcpJ1Hss_Ti_oPHZXaD1C8GEyA37C27kEcTOlD9pNpb0D57Q-zYs6T0JgC6t7BexEVb72mTK_Oxnu6s-yrAB7AfECqP-98IA3HkunY | 원본파일 |
| https://lh3.googleusercontent.com/PfADmM5ASN8KN4ZMi9cZepiyRu1o9R3xym5jm0X0ZXw1qiYGRG9oM3PxIkFbAdHLghnW9DcklhzM_9N_WUs6i-Hc9lJSjdSQVGSzKILE-QJfUvpTEUsd-uM-NcIhi-jGOH_mh0he | 블러제거 |
| https://lh5.googleusercontent.com/0IVaPauEsFExYmh5IL1078rkCoDxyiVCa44G6qvXBaJTpy5g-6NgtKXwequmzKQFLgMkmP6_WoTqcqueQ9eCg-VB5mFSXc4TylbhjPkqRX9dXK2iT0G723zfKb_UUEpAg0U3RTaK | 라인제거 |
| https://lh4.googleusercontent.com/F1-OQiSH-MHNonWEcd7ZERbKb1duTUuOrZhyPJw03jzAuNfiZ4Ub2DSy2Q9hkhELEvmm_-7wjA0wi6CXg6Y8oAWeKHJXCVwbs4FTeUD_LvL-OLQDzJWOs_raDqqEXLbkAcB81Gechttps://lh6.googleusercontent.com/Qtk9Ax-wEottcewPjYr212X9owmLgLPAQJclCNpft1QflEb1ffM7eR62EK7i3jdtypwaEc9fvw-GBKKv90JlcIlGj-7_jJbYRmkiAdNyfogMIbky11hjP2MbebnoCzC-tMBRglI_https://lh4.googleusercontent.com/L7Z8-_gkkZlLDaRX4V0LHX3ED84Y3prVnBxWdpjaexaXPkIlMZW68_oNTX_F4ca_QmeosGjYILaWsSgdWWMxldue2Xz6dlN0ui4JL4nK2qOD94XEc1xeSnM4EDAG4Qp1Q7jSLZiVhttps://lh6.googleusercontent.com/NZpTxY-6PCrDelV7FPvCcmyQ16ACmEDFfiAw_STV2COJReDpSMbZQlL5ucYH2Y6bHs5-M0WrlZjV8EO77vSEAXj6UNcCvw0m6_1VQ7xbgs3VimHYyz2HIP8gHAk6rr3BVIvxWghXhttps://lh4.googleusercontent.com/s8q1Ger_peIyaZYf27X0ypLJcrjYcWlxHSXEN3DOdHqc7YD057X5lxsYj1yq0EZ3SWHgHKr62BBUsxJ4ra4TnWHyy9oq6V-bEq0zLGBeQGZcdQsi_9T6hC2pWwuKInga9N9CwfqM | 문자별로 이미지 분류 |
| https://lh6.googleusercontent.com/DaLmN6Wsc-0DO_gC5Lf6PJBSNibckNtIWC03abUUZMGiFqDeqIR3T2pst5bYTIn_28dtT2Nspyg-Gal0npCJxYC9GhgY7Vqd2N78aHLV14WduirVdF-demAWXWJUoTKCUiyLa0CDhttps://lh3.googleusercontent.com/M8LlkHuLWJxWn5qkgaWa7TZ695TG1ZfXr77d-xJgC7sJeWSMBNqBlokxwv3BNjAR-nY2E_WW9YzoS-Ugv_qPy577GQGZNaGyCxeMtfj8J2kIhlUTXbAbBf0UIJM_U6Kd93BGEhyIhttps://lh4.googleusercontent.com/nsXGSlkoMHMMAt9mkiUPI_6-wI64SA7mFj4LkVGYBQ5hEvEdRaK4m4A7IS2Ii3TFiKUUUCc3lc5cC3EibDDvwo1iIDVD2lE9BNH2G4DD5GKGGZu-K2-RSishtocIuI4RayRTQWTwhttps://lh4.googleusercontent.com/mg2JkZ6hZ3i8o7OOs-Sw6EmhOGmY3UL4FaDaAgutGTCTSt4NN59PdGJ0RUJh2MQqFUe8H9DmRyYj117iuj74pXCkuxj4sKgFIzI17EOf6-w7_GeFVEXHOo4vWNOOhSHhXztdeHK6https://lh3.googleusercontent.com/8vlJgh9wRyUF7rVpbbRPCjz6S84O2M7A-czIjiJfVhMzGi4qyWUPr_bMWMfWuFRwFAbOUoCXTjW_3eW63xhGMj2Sims-JLzTkauZnk7vRzGdMPqz7WORx_rvvURdjkTKGGL8fS3Y | 흑백화 |

[표 3] 이미지 전처리(Pillow 패키지)

|  |  |
| --- | --- |
| 음성 데이터 | 설명 |
| 스크린샷이(가) 표시된 사진  자동 생성된 설명 | 원본데이터 |
|  | 음성 파일을 값으로 표현 |
|  | 문자 각각으로 나누기 |
|  | 패딩을 통하여 같은 길이로 만들기 |

[표 4] 음성 전처리

* 머신러닝 알고리즘(sklearn 패키지), 딥러닝 알고리즘(Tensorflow, Inception V3)



[그림 6] 머신러닝 작업 과정

머신러닝 알고리즘은 [그림 6]과 같은 과정을 거쳐 학습되고 예측된다. 문제와 정답을 train set으로써 설정되어 설정하려는 머신러닝 모델에 맞추어 데이터 학습이 진행되어 데이터 모델이 생성되고 새로운 데이터를 test set으로서 설정되어 생성된 데이터 모델에 적용되어 예측한다.

* 자동 입력, 이미지-음성 데이터 수집 (Selenium 패키지)

사용자 정보 및 Captcha 예상결과 자동 입력과 이미지, 음성데이터 자동 수집을 위해 Selenium 패키지를 사용한다. Selenium을 정상적으로 사용하기 위해서는 지원되는 브라우저와 해당 브라우저에 맞는 드라이버가 반드시 필요하다. 우리는 가장 널리 사용되고 있는 크롬 브라우저를 통해 작업을 진행하며 크롬 드라이버는 “http://chromedriver.chromium.org/downloads”에서 다운로드 받을 수 있다. 또한 해당 자동화 작동은 소스코드로부터 크롬을 자동화하는 것이 아닌 디버깅 모드로 열려있는 크롬에 크롬드라이버의 통신을 이용한 명령 전달로 자동화되기 때문에 크롬을 디버깅 모드로 열어야지 자동화 기능을 적용할 수 있다. 디버깅 모드로 여는 방법은 크롬 디렉토리 내부에서 다음 명령어를 입력한다. “chrome.exe --remote-debugging-port=9222 --user-data-dir=\"C:\ChromeTEMP\"”

**6.3 위험요소 및 대처 방안**

* 학습된 머신러닝 및 딥러닝 모델 적용 방법
  + 프로그램에 학습된 모델을 탑재하여 배포.
  + 문자 인식은 개별 서버를 두어 서버에서 실행.
* 필수 라이브러리 개별설치 필요 문제.
  + 프로그램 자체에 라이브러리 내장.
  + 머신러닝 및 딥러닝 인식의 경우 서버에서 동작.
* 유저 컴퓨팅 파워에 따라 문자 인식 속도 문제
  + 서버에서 진행.
  + 처리 최적화 및 전처리 입력 데이터 최소화
* 입력 자동화를 위한 필수 프로그램
  + 프로그램 자체에 내장
  + gUI 상에서 사용자에게 다운받도록 연결

1. **개발 일정[프로젝트 관리]**
   1. **역할 분담**

|  |  |
| --- | --- |
| **역할 분담** | |
| **김형호** | * **프로젝트 전체 진행 관리** * **문자 인식 라이브러리, Tesseract 테스트** * **딥러닝 알고리즘 학습 및 테스트** |
| **이상민** | * **Selenium을 통한 사용자 정보 자동입력** * **음성 STT, CSR API 테스트** * **이미지 전 처리, 모든 과정** * **이미지, 음성 데이터 다양한 머신러닝 알고리즘 학습 및 테스트** |
| **송연주** | * **GUI 설계 및 생성** * **음성 STT, 구글 speech API 테스트** * **음성 전 처리, 모든 과정** |
| **조영준** | * **GUI 상세 기능 구현** * **사용자 정보 저장 기능 구현** * **Selenium을 통한 captcha 이미지, 음성 데이터 다운로드** * **이미지 Test, Train 분류** |

[표 5] 역할 분담

김형호는 프로젝트 전체 관리, 딥러닝 테스트를 주로 맡았고 이상민은 자동입력, 이미지 전처리 과정, 머신러닝 테스트를 주로 맡았고 송연주는 GUI 설계, 음성 전처리 과정을 주로 맡았으며 조영준은 GUI 상세 기능과 이미지, 음성 데이터 다운로드를 맡았다.

* 1. **개발 일정**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | month | 3 | | | | 4 | | | | 5 | | | |
|  | week | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 기획 | 아이디어 회의 |  | | |  |  |  |  |  |  |  |  |  |
| 음성 처리 | 구글 API를 이용한 음성인식 |  |  |  |  |  |  |  |  |  |  |  |  |
| 클로바 API를 이용한 음성인식 |  |  |  |  |  |  |  |  |  |  |  |  |
| 음성파일 전처리 |  |  |  |  |  | | |  |  |  |  |  |
| 디버깅 |  |  |  |  |  |  |  | |  |  |  |  |
| 이미지 처리 | Tesseract 라이브러리 사용 |  |  |  |  |  |  |  |  |  |  |  |  |
| 이미지 전처리 연구 |  |  |  |  | |  |  |  |  |  |  |  |
| 문자 별 다양한 전처리 분류 |  |  |  |  |  | |  |  |  |  |  |  |
| 새로운 전처리 알고리즘 구상 |  |  |  |  |  |  |  |  |  | |  |  |
| 이미지 특징점 추출 |  |  |  |  |  |  |  | | | | | |
| 문서 작성 | 아이디어 제안서 |  |  |  |  |  |  | | |  |  |  |  |
| 설계서 |  |  |  |  |  |  |  | | | | |  |
| 최종보고서 |  |  |  |  |  |  |  | | | | | |
| 음성, 이미지 인식 | 기본 설정 머신러닝 테스트 |  |  |  |  |  |  |  |  |  | | | |
| 설정 변경 머신러닝 테스트 |  |  |  |  |  |  |  |  |  | | | |
| 클라이언트 | GUI |  |  |  |  |  |  |  |  |  |  |  |  |
| 사용자 정보 자동 입력 |  |  |  |  |  |  |  |  |  |  |  |  |

[표 6] 개발 일정

1. **참고 자료**
   1. <https://www.nia.or.kr/common/board/Download.do?bcIdx=20146&cbIdx=82618&fileNo=1>
   2. <https://en.wikipedia.org/wiki/Deepfake>
   3. <https://github.com/deepfakes/faceswap>
   4. <http://nownews.seoul.co.kr/news/newsView.php?id=20180222601022>
   5. <http://news.kbs.co.kr/news/view.do?ncd=3601037>
   6. 보안성을 향상시킨 이미지-텍스트 융합 캡차 자동 생성 시스템, 문광호, 인하대학교 대학원, 2011
   7. CAPTCHA에 사용되는 숫자데이터를 자동으로 판독하기 위한 Autoencoder 모델들의 특성 연구, 전재승, 고려대학교  정보보호대학원, 2017
   8. <https://m.blog.naver.com/netrance/110141647706>