0507 현재 진행상황

1. url 비교

1. 접속한 우리은행 화면에서 @->!이런 식으로 바꿔서 텍스트에 저장

2. 문서형식으로 저장된 사용자가 현재 접속한 url을 기존에 크롤링으로 저장한 url리스트들 과 비교한다.

3.

3-1. 일치하면 다음 procedure로 넘어감(텐서플로우 이미지 비교)

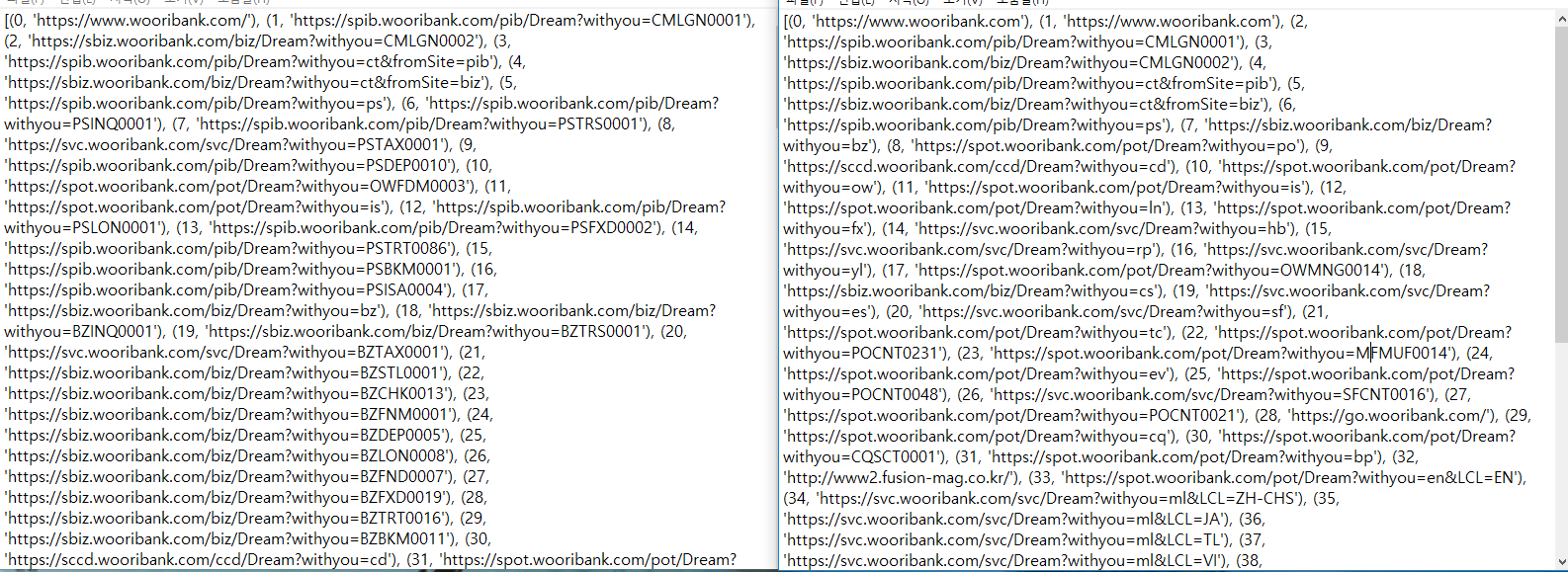
3-2. 일치하지 않으면 위조사이트라고 판단하여 알림

2. 학습시키기 위한 이미지 전처리

1. 웹사이트 크롤러로 이미지 저장(이름에서 /->@ 뭐 이런식으로 바꿔서 저장, 안 그러면 저장이 안됨)

2. 각 transaction에 대하여 같은 절차를 수행 (몇 단계 정도가 optimum?)

3. 한 단계가 끝날 때마다 같은 이미지 여러 개 복사해서 이미지 이름(url)으로 된 폴더에 저장 후 바로바로 학습시키기



<왼쪽은 <https://www.wooribank.com/>에 대한 트랜잭션 url>

<오른쪽은 <https://www.wooribank.com/>의 트랜잭션 url 중 하나인

<https://spib.wooribank.com/pib/Dream?withyou=CMLGN0001> >

-> 이 url에 해당하는 웹사이트 이미지를 데이터 셋으로 저장하여 바로바로 학습 시킬 수 있도록 한다.

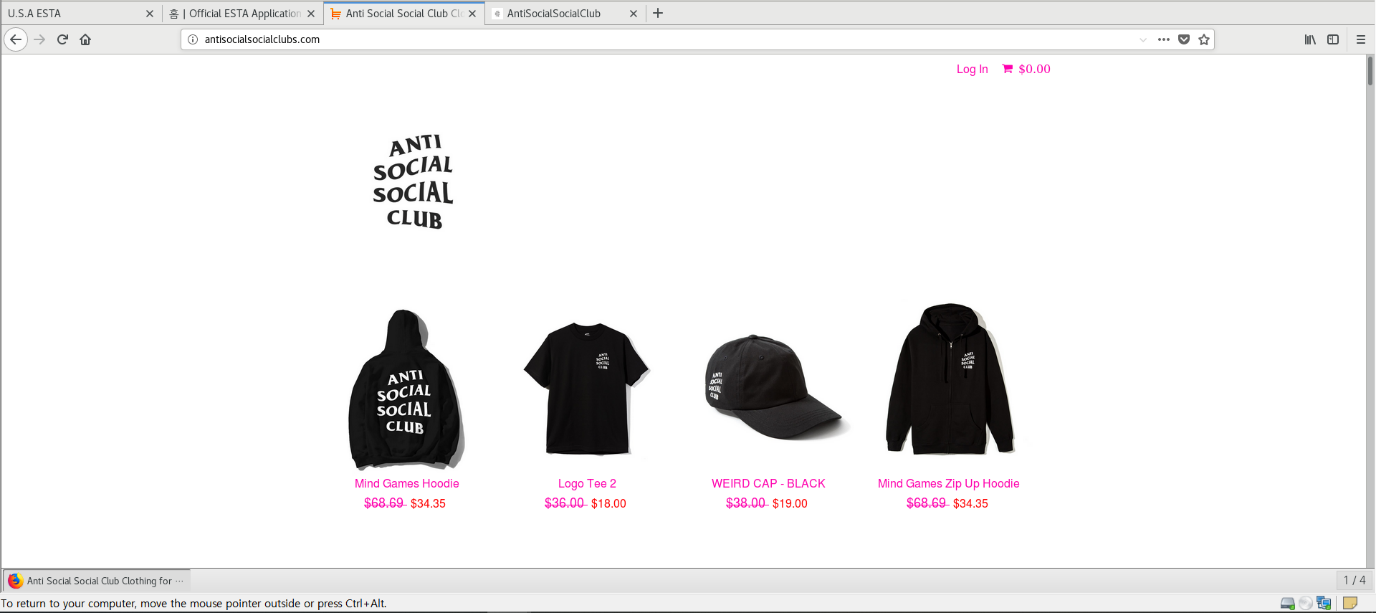


한 단계가 끝나면 바로 이미지로 저장하지 않고 리스트에 저장해 둔다.

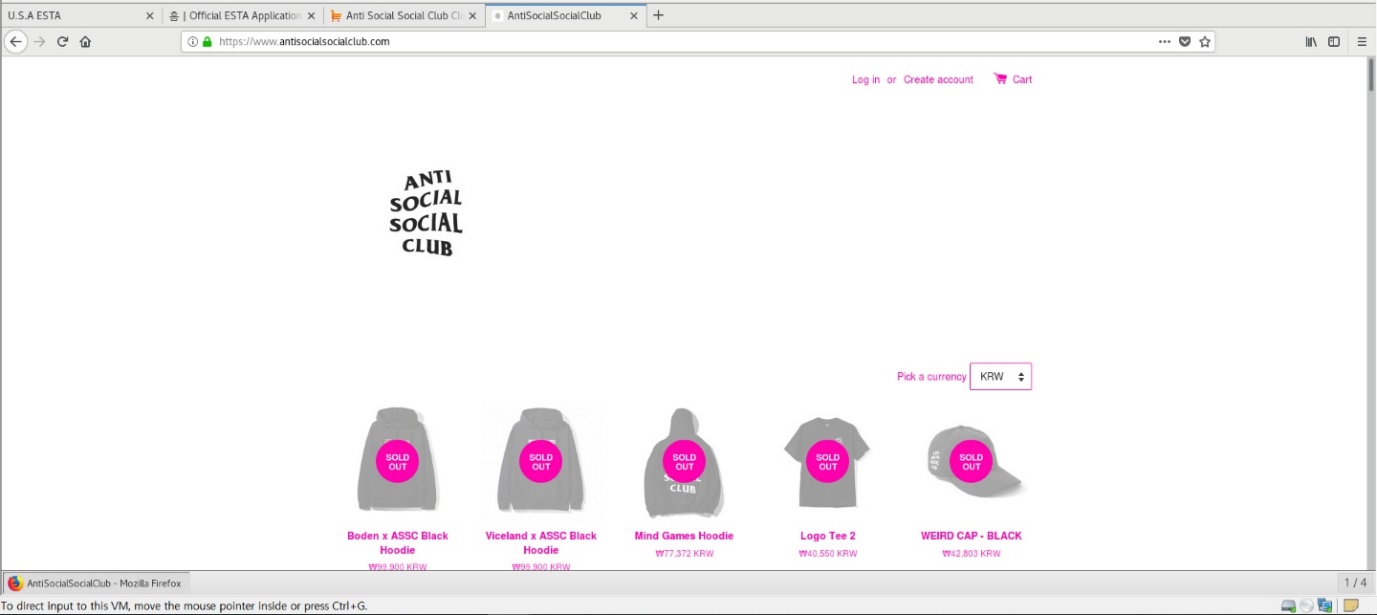


두 번째 단계에서 각 transaction에 대해 크롤링 한 후 이 단계에서 이미지로 저장한다.

3. 가짜 사이트 vs 진짜 사이트 탐지 -> 퍼센트 비교

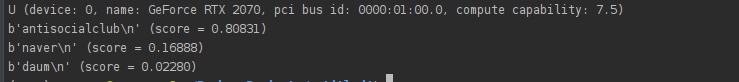


-> 가짜 이미지



-> 진짜 이미지

**- 1000번 학습**



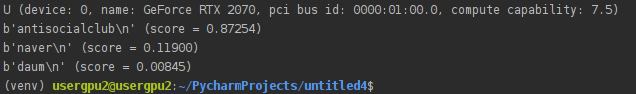
-> 가짜 이미지



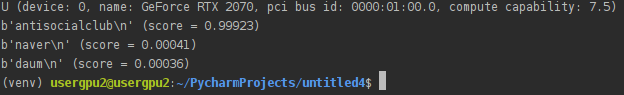
-> 진짜 이미지

**=> 가짜 80.8%, 진짜 99.6%로 진짜 이미지로 탐지한 결과가 진짜antisocial이미지로 학습된 모델에서 유사도가 더 높게 검출 되었다.**

**- 5000번 학습**



-> 가짜 이미지



-> 진짜 이미지

**=> 가짜 87.3%, 진짜 99.9%로 진짜 이미지로 탐지한 결과가 진짜antisocial이미지로 학습된 모델에서 유사도가 더 높게 검출 되었다.**

\* 더 해야 할 것

1. 사용자가 보고 있는 화면 캡쳐

2. 서버와 양방향 통신해서 데이터 주고받기

3. 알림 서비스(프로그램) 어느 정도 구현 시작

4. 몇 단계까지 크롤링 할 건지 정하고 그 단계까지 되도록 코딩

5. 가짜 사이트 더 찾아서 돌려보기

6. url 비교 코드 완성하기(금방 끝날 듯)