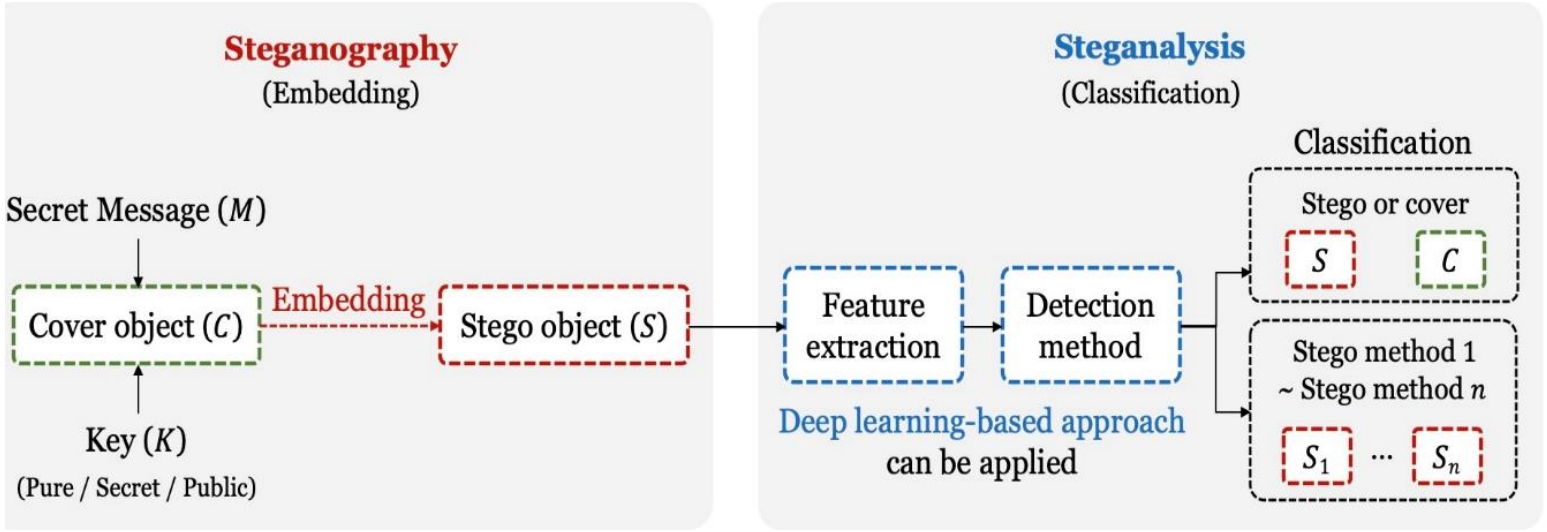
스테가노그래피

https://youtu.be/UA57BOyafSI

스테가노그래피 & 스테그아날리시스

- **스테가노그래피**
  - 이미지나 오디오 등의 데이터에 기밀 정보를 임베딩하여 기밀 정보의 존재 자체를 숨기는 것이 목적인 데이터 은닉기법

- **스테그아날리시스**
  - 스테가노그래피를 탐지하는 기술



스테가노그래피

- **데이터 암호화와 임베딩 과정으로 구성**

- **암호화**
  - 사용된 암호화 종류에 따라 일반 / 비밀키 / 공개키 steganography로 나뉨
    - 암호화 하면 기밀성 확보 가능
    - 그러나 숨기고자 하는 데이터의 크기 증가
      → 숨길 수 있는 메시지의 크기가 줄어들고 커버 오브젝트의 용량 증가
      → 스테가노그래피의 존재 여부가 발각되기 쉽다는 단점 존재

- **임베딩**
  - 암호화 후, 비밀 데이터를 커버 오브젝트에 숨김으로써 스테고 오브젝트 생성

- 다양한 임베딩 알고리즘 존재

스테가노그래피 임베딩 기법

- **적응형 방식**
  - 비밀 정보 삽입이 용이한 **(찾기 어려운) 지점에 삽입**
    → 성능이 더 좋음 (탐지는 어려움)
  - HUGO, S-UNIWARD 등

- **적응형 X 방식**
  - LSB, PVD 등 그냥 숨김

스테그아날리시스

- **종류**
  - 범용형 : 스테가노그래피로 생성된 스테고 오브젝트인지 아닌지를 탐지
  - 표적형 : 생성 알고리즘의 종류 탐지

- **특징 추출** (Feature extraction)과 **탐지** (Detection, classification) 과정으로 구성
  - **특징 추출**
    - 스테고 오브젝트에 존재하는 특징 값들을 추출
      → DCTR (Discrete Cosine Transform Residuals), GFR (Gabor FilteR bank) 등
  - **탐지**
    - 특징 추출 과정을 거친 데이터에 대해 통계 기반 탐지 기술 적용하여 분류 (범용형, 표적형 둘 다 가능)

스테가노그래피 및 스테그아날리시스를 위한 오픈 소스 도구

- 아래 표는 대표적인 오픈소스 도구들이며, 이외에도 깃헙에도 관련 툴들이 있음
- 실제 스테가노그래피 시나리오를 고려 (암호화를 수행 후 임베딩)하려면
  암호화 적용이 가능한 툴을 사용하거나 직접 입력 데이터를 가공해서 넣을 수 있는 형태의 툴 사용해야 할 듯

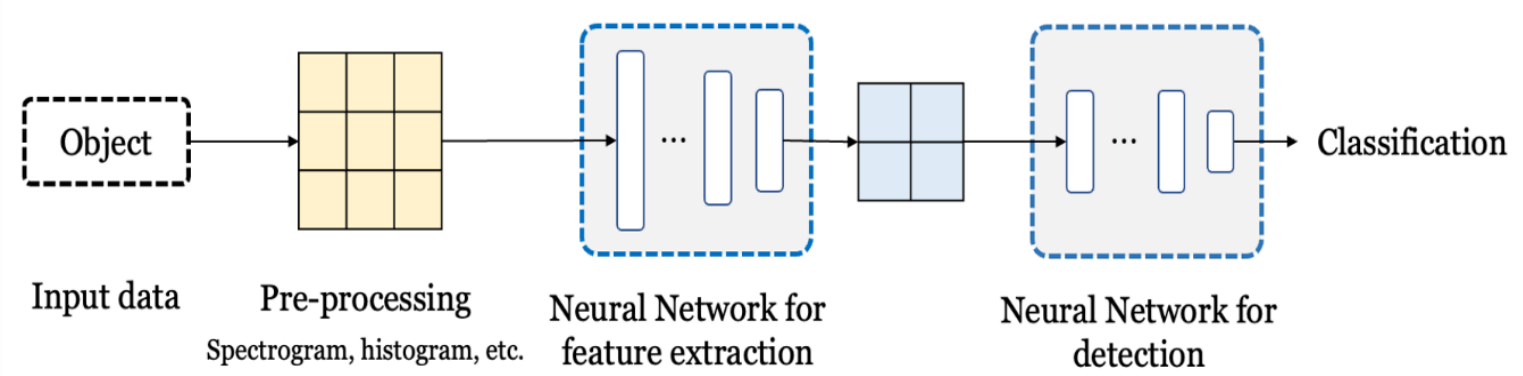
[표 1] 스테가노그래피 및 스테그아날리시스를 위한 오픈소스 툴

Method	Tool	Cover object type
Steganography	Openstego[12]	Image
	SteganPEG[13]	Image
	Steghide[14]	Image/Audio
Steganalysis	StegSpy[15]	Image
	StegDetect[16]	Image
	StegSecret[17]	Image/Audio/Video

딥러닝 기반의 스테그아날리시스

- CNN (Convolutional Neural Networks)과 **같은 딥러닝 모델 통해 자동으로 데이터 특징 추출**
  - 히스토그램, High Pass Filter (HPF) 등의 전처리 과정 거친 후, 모델에 입력하여 학습
  - 탐지를 위해서는 이진 분류 또는 다중 분류를 위한 학습 모델 필요

- **J-UNIWARD, S-UNIWARD, WOW 등과 같은 다양한 임베딩 방식을 대상으로 함**
- 특징 추출 및 탐지 과정에 딥러닝 기술 적용 가능



스테그아날리시스 연구 동향 (추가 예정)

- 많은 양의 비밀 데이터를 임베딩할 경우에는 높은 정확도, 적은 양의 데이터를 임베딩할 경우 낮은 정확도 달성
- **높은 정확도**를 갖는 모델 구현에 중점을 두며, 이를 위해 높은 차원의 데이터 특징을 추출하므로 **높은 계산 복잡도** 가짐
- 모델의 정확도가 아닌 **효율성에 중점**을 둔 연구들의 경우, **대부분 소규모의 데이터 셋** (5000~10000 정도)사용
- **대규모 데이터 셋의 경우 30000~100000개까지도 사용**
- 우리가 아는 한 임베딩 되는 비밀 데이터와 성능 간의 관계에 대한 분석은 수행되지 않음
- 실제 환경에서는 임베딩 되는 데이터에 암호화가 적용

  → 딥러닝 기반의 스테그아날리시스에 관한 대부분의 연구에서는 이를 고려하지 않고 무작위 일반 데이터를 임베딩

스테그아날리시스 구현 동향

- 다음과 같은 흐름이 있었음
  - Feature extraction, classification에 **딥러닝 적용**
  - **활성화 함수** 변경 (tanh, sigmoid 등을 Gaussian으로 변경한 사례 있음), **정규화** 적용
  - **데이터 셋** 추가 (섞어서 쓴 경우도 존재)
  - HPF와 같은 **전처리** 과정 추가
  - **네트워크 성능 향상**에 초점 (Residual 구조, Sub network 사용, Batch normalization 적용 등)
  - 낮은 페이로드 (임베딩 비율)에 대한 성능 향상 위해 **전이 학습**
    - 높은 페이로드로 학습한 후, 낮은 페이로드 데이터에 전이 학습
  - 새로운 **전처리** 적용 (예시 : 테일러 급수 전처리)

- 위와 같이 **구조적 개선, 데이터 셋 추가, 새로운 학습 기법 적용, 전처리 과정** 개선 등이 수행됨

스테그아날리시스 구현 동향 (추가 예정)

Ref.	네트워크 구조	Dataset					실험 환경	기타
		개수	크기	스케일	임베딩	특징 추출		
[1]	Dense+ BatchNorm	10000 개	512x512	gray	J-UNIWARD, nsF5, UERD 모두 0.1, 0.4	DCTR, GFR, PHARM	64GB RAM, 8vCPU cores, DELL PowerEdge sever	앙상블 구조
[2]	Conv2D+ BatchNorm+ Average Pooling	10000 개	256x256	gray	S-UNIWARD, WOW 모두 0.2, 0.4	DCTR + SRM (각각 진행 후 합침)	언급 X	Multi-domain feature 고려
[3]	Conv2D+ BatchNorm+ Average Pooling	임베딩 기법당 30000 개	256x256	gray	S-UNIWARD, WOW, PVD,LSB	SRM	언급 X	HPF 전처리, 계층적 구조로 여러 번 분류

[1] Detection of Image Steganography Using Deep Learning and Ensemble Classifiers
[2] Joint multi-domain feature learning for image steganalysis based on CNN
[3] 계층적 CNN 구조를 이용한 스테가노그래피 식별

스테그아날리시스 구현 동향 (추가 예정)

- 이외에도 다양한 연구들이 있으며, 해당 연구 사례들을 살펴보고 정리한 결과는 다음과 같음
- **네트워크 구조**
  - 대부분 Conv2D + BatchNorm + Average pooling 사용 (Maxpooling도 사용됨)
- **데이터 크기, 개수 및 스케일**
  - 256 또는 512, 10000~100000개, 흑백
  - 흑백이 대부분이어서 컬러 데이터 시도해보았으나 메모리 사용량이 너무 많아서 실행이 어려웠음
- **실험 환경**
  - 네트워크 구조 위주로 읽어서 아직 전부 파악하지는 못함 (64GB 쓴 경우 있음)
  - 실제 돌려본 결과 64GB RAM 이상 사용해서 코랩으로 돌렸고
    램 터질 때도 많아서 네트워크 크기나 데이터를 많이 늘릴 수 없었음
- **전처리**
  - 특징 추출 이전에 전처리를 하는 경우 많음
  - HPF가 대다수인 것 같음

- 이외에도 앙상블, 계층적 구조, 여러 도메인 고려하는 것처럼 **구조적인 개선**을 한 사례들이 많은 것 같음

현재 상황

- **LSB 임베딩 시도** → 컬러이미지만 되는 툴이었는데 컬러 데이터는 메모리 사용량이 많아서 포기
- 스테그아날리시스 데이터 셋을 발견하여, 그 중에서 **J-UNIWARD** 사용
  - **네트워크 구조는 구현 동향을 기반으로** 램이 터지지 않는 선에서 적당히 설정 (아직 튜닝 되지 않은 상태)
    - Batch normalization은 필수인 듯 해서 적용
    - 가장 많이 사용되는 average pooling 적용
    - Flatten 대신 Global average pooling 사용 (파라미터나 성능 측면에서 장점이 많으며, 요즘 트렌드라고 함)
  - **그러나 0.58~0.61에서 학습되지 않음** (언더 피팅으로 생각됨)
    - 동일 네트워크로 MNIST는 학습이 되어서, 데이터에 안 맞는 모델이거나 데이터가 적거나 데이터가 안 좋은 것으로 생각됨
- **문제 해결 중**
  - 데이터 증가
    → 메모리 때문에 30000개 이상 늘리기가 어렵고 새로 산 데스크탑이나 학교 컴퓨터에서 실행 불가
    (코랩에서는 동작 가능하며, 80GB 주는데 70GB 정도 사용하는 것 같음)
  - 모델 복잡도 증가
    → 메모리 때문에 어려움, 모델 구조를 다르게 한다거나 다른 방법을 생각해볼 예정
  - 데이터 노이즈 감소
    → 전처리는 아직 시도 X
  - Epoch 증가
    → 다른 방법 먼저 해보고 시도할 예정
  - 이외에도 더 찾아보는 중

향후 계획

- 연구 동향 더 살펴보기
- 임베딩 기법 이해 및 툴 사용
- 네트워크 세부 구조 파악
- 전처리 기법 공부 및 적용
- CNN 같은 이미지 처리를 위한 네트워크의 성능 향상을 위한 구조나 학습 기술들 파악

감사합니다.