* **코드 수정 사항**
  + GetTileInfo.py: requestTile(X,Y) 내의 코드들을 예외처리
    - 사유: JsonDecodeError가 계속해서 발생하는데, 코드 실행을 중단했다가 다시 실행하면 정상적으로 파일이 다운로드 됨
    - 해결법: **Exception 발생 시 다시 requestTile(X,Y) 호출**
  + ReadLog.py, GetSegments.py, MakeEdges.py: csv파일 저장 코드 변경
    - 사유: Window 운영체제에서 csv.writer를 사용할 경우 빈 행 발생
    - 해결법: **f = open(csv\_name, “w”, newline=””)**와 같이 변경
* **현재 segment가 선정되는 과정**

1. ./Tileinfo/의 {tileIdX, tileIdY}.json에서 ['geometry']['coordinates']의 'latitude'와 'longitude'를 추출하고, 메르카토르 좌표 X, Y로 변환해서 tileIdX, tileIdY와 함께 segment.csv에 저장한다.
2. ./Historylog/에 저장된 유저의 log 파일을 ReadLog.py를 통해서 변환한 뒤, ./log\_info/에 log{logname}\_info.csv에 저장한다.

* 이 때, log에 저장된 메르카토르 좌표를 위도, 경도 정보로 변환하고, 이를 Utils.py의 deg2num 함수로 16만큼 zoom해서 tileIdX, tileIdY를 계산하고 기록한다.

1. ./log\_info/의 log{logname}\_info.csv와 ./segments/의 segment.csv를 Candidate Segment Selection System을 이용해서 가장 가까운 segment를 선정하여 ./log\_segment\_info/user/{logname}\_segment\_info.csv에 저장한다.
   * **과정 요약**
2. Log의 위, 경도가 위치하고 있는 Tile을 찾는다.
3. 그 Tile에서 추출한 segment와 Log의 ‘tileIdX’, ‘tileIdY”를 비교해서 가장 가까운 segment를 찾는다.

* **미등록 구간 추출 방안**
  + **방안 1) anomaly detection을 활용하여 Tile에서 추출한 Segment와 Log에서 추출한 Segment(?)를 비교해서 이상치를 새로운 segment로 추가**
  + 방안 2) Log의 메르카토르 좌표 중 Segment의 메르카토르 좌표와 일치하지 않는 것들을 싹 다 segment로 만들기
  + 방안 3) 많은 사람들이 자주 다니는 길 중 segment로 등록되지 않은 곳만 등록
* **방안 1 수행 내용**

1. **Anomaly 학습**
   1. **Anomaly detection이란?**
      1. [딥러닝을 사용한 Anomaly Detection에 관한 거의 모든 것 (정의,설명,어려움,활용 방안,데이터셋,연구 흐름 등) :: ImagineOnTheGiants (tistory.com)](https://ffighting.tistory.com/entry/%EB%94%A5%EB%9F%AC%EB%8B%9D-Anomaly-Detection)

* Low-level anomaly가 미구간 경로 추출에 속하지 않을까?
  1. **정상 데이터를 학습하고 테스트 데이터에서 이상치 찾기?**
     1. [Anomaly Detection 개요： (1) 이상치 탐지 분야에 대한 소개 및 주요 문제와 핵심 용어, 산업 현장 적용 사례 정리 - 블로그 | 코그넥스 (cognex.com)](https://www.cognex.com/ko-kr/blogs/deep-learning/research/anomaly-detection-overview-1-introduction-anomaly-detection)
* Segment.csv는 현재 발견된 구간들(모두 정상 sample)이고, log{logname}\_info.csv는 발견되지 않은 구간이 포함된 데이터 셋이므로 **Semi-supervised (One-class) Anomaly Detection**을 이용해보자
* 문제점 1: 마땅한 label을 찾을 수 없음
  + **tileIdX, tileIdY:** 수많은 구간이 포함된 하나의 tile이기 때문에 label로 사용하기에는 마땅하지 않음
  + **seg\_id:** label이 너무 많아짐
* 문제점 2: Image classification이 주된 목적으로 보임
  + x, y 좌표 기반의 데이터를 어떻게 적용해야 할지 고민해야 함
  1. **Unsupervised anomaly detection?**
     1. [Unsupervised Anomaly Detection in Python | by Edwin Tan | Towards Data Science](https://towardsdatascience.com/unsupervised-anomaly-detection-in-python-f2e61be17c2b)
        1. [Isolation forest을 이용한 이상탐지 | Biohacker (partrita.github.io)](https://partrita.github.io/posts/isolation-forest/)
* label이 필요 없는 학습 방식
* 먼저 Semi-supervised Anomaly detection을 한 뒤에 시도해 볼 것

1. **Semi-supervised Anomaly Detection**
   1. [[논문 리뷰 및 코드구현] Deep One-Class Classification(Deep SVDD) (tistory.com)](https://wsshin.tistory.com/m/3), [[바람돌이/딥러닝] Deep SVDD 논문 및 코드 리뷰 : 네이버 블로그 (naver.com)](https://blog.naver.com/PostView.naver?blogId=winddori2002&logNo=222142873877&parentCategoryNo=&categoryNo=32&viewDate=&isShowPopularPosts=false&from=postView), [Deep One-Class Classification 분석 (tistory.com)](https://ys-cs17.tistory.com/50)
      1. 전처리: [[pytorch] MNIST dataset loading 하기 (teddylee777.github.io)](https://teddylee777.github.io/pytorch/pytorch-mnist-dataloader-loading%ED%95%98%EA%B8%B0), [PyTorch 데이터로더 이해하기 - Part 1 (promedius.ai)](https://blog.promedius.ai/pytorch_dataloader_1/), [[PyTorch] dataset과 dataloder 설명 및 custom dataset & dataloader 만들기 (tistory.com)](https://sanghyu.tistory.com/90)<https://towardsdatascience.com/unsupervised-anomaly-detection-in-python-f2e61be17c2b>
      2. Training set: Segment.csv, Test set: log{logname}\_info.csv
      3. 사용한 feature: **(잠시 보류)**

* **궁금한 점**

1. Tile.json의 [‘coordinate’]에서 segment를 추출하지 않고, Log의 mercX, mercY, latitude, longitude, tileIdX, tileIdY, direction을 이용해서 군집화 등등의 방법으로 직접 segment를 만들어내는 방식은 별로인가?
2. 찾아봤던 Anomaly detection 관련 모델들이 모두 이미지 분류에 사용됐는데, 이를 우리의 데이터셋에 어떻게 적용하는가?

* 데이터를 어떻게 전처리해야 하는가?