**The 3rd YouTube-8M Video Understanding Challenge**

※ 주최 : ICCV(International Conference on Computer Vision) – Google Research

※ more information : “YouTube 8M Large-Scale Video Understanding Workshop Page”

※ 기간 : June 28, 2019 ~ September 14, 2019  
 \* Rules Acceptance Deadline : September 7, 2019

**1. 목적**

: 중요하거나 특별한 이벤트들을 다루는 비디오에서 원하는 순간을 검색할 수 있는 기술을 발전시키는 것 ( = Temporal Concept Localization)

이 기술들은 향상된 비디오 검색(비디오 내에서의 검색 포함), 비디오 요약, 하이라이트 추출, 액션 순간 감지와 개선된 비디오 내용 안전 등의 많은 부분에 적용할 수 있다.

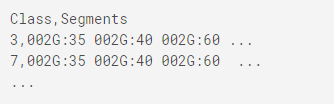
즉, 레이블이 실제로 나타나는 비디오에서 정확한 시간에 video-level 레이블들의 위치를 알아내는 작업을 하는 것이 목적이다.  
예를 들어서, 어느 시점에 고양이가 재채기를 하는지 정확한 지점을 파악하는 작업이다.

**2. 평가요소**: K = 100,000일 때 MAP(=Mean Average Precision)에 의하여 평가된다.

  
C : Class들의 수, P(k) : k 정지에서의 정밀도  
n : 각각의 Class 당 예측되는 Segment의 수  
rel(k) : rank k에서 항목이 (옳은) Class와 관련이 있으면 1, 이외에는 0으로 나타내는 지표 함수  
Nc : 각각의 Class에서 긍정적으로 레이블이 된 부분들의 수  
  
※ 평가되는 부분들

1. 각각의 Class에서 관련된 부분들을 예측하였는지.
2. 모든 Segment들이 사람과 관련되어 있지는 않지만, 오직 사람과 관련된 Segment들이 점수 기록에 사용될 것이다.  
   명확하게 관련되어 있지 않는 모든 Segment들은 (예를 들어서 Class를 포함하고 있거나 포함하고 있지 않거나) 점수를 기록하기 전에 예측 목록에서 제거될 것이다.
3. 공공의/사적의 분할된 Test는 Class 수준이 아닌 Segment 수준에서 수행될 것이다.

※ MAP@K의 파이썬 이행은 ‘youtube-8m’s github’에서 찾을 수 있을 것이다.  
  
**- 제출 파일 -** ‘sample\_submission.csv’에서 각각 1000개의 Class 값들에 대하여, 확신하는 순서로 정렬된 Class를 포함한다고 예측되는 100,000개의 Segment ID들을 예측할 수 있을 것이다.  
※ 파일 크기를 최소화하기 위해서는, segment 종료 시간이 아닌 각각의 Segment 예측에서 오직 비디오 id와 segment 시작 시간만 포함해야 한다. 그리고 파일은 header와 밑의 포맷을 따라야 한다.



※ 시작 코드는 ‘google/youtube8m github repo’에서 찾을 수 있다.

**3. 첨부파일 설명**

- frame-sample.zip : ‘train00’ 과 ‘train01’을 포함하는 frame-level 데이터의 샘플  
- validate-sample.zip : ‘validate00’ 과 ‘validate01’을 포함하는 validation set 데이터의 샘플  
- vocabulary.csv : 레이블 이름들과 그것들에 관한 설명에 대한 큰 데이터 사전  
- sample\_submission.csv : 올바른 포맷에서의 샘플 제출 자료  
- frame-level data  
: 지시를 따라서 개인 컴퓨터에 다운로드 가능하다.

: 총 크기는 1.53TB이다. (Large file warning!)

: 각각의 비디오는 아래의 성질을 가진다.  
a. id : 비디오에 따라서 유일하다. Train set에서는 유튜브 비디오 id이며, test/validation set에서는 익명이다.  
b. labels: 비디오들의 레이블 목록  
c. Each frame has rgb: 1024의 길이를 가진 실수 집합체이다.  
 d. Each frame has audio: 128의 길이를 가진 실수 집합체이다.

: validation set 비디오들의 부분집합들은 segment-level 레이블들이 주어진다. 게다가 id와 labels 그리고 frame level들의 특성들은 밑의 a~d과 같이 주어져서 설명된다.  
a. segment\_start\_times: segment 시작 시간들의 집합  
b. segment\_end\_times: segment 종료 시간들의 집합  
c. segment\_labels: segment 레이블들의 목록   
d. segment\_scores: segment 레이블들에 따라서 양 또는 음을 나타내는 이진수 값들의 목록

: 파일들은 [TFRecords](https://www.tensorflow.org/api_docs/python/tf/io#tfrecords_format_details) 포맷에 존재하며, TensorFlow 파이썬 리더들은 [github repo](https://github.com/google/youtube-8m)에서 이용 가능하다.

**4. 분석 순서**

1. 패키지 및 모듈 설치  
- numpy  
- pandas  
- os  
- tensorflow  
- YouTubeVideo (IPython.display)  
- pyplot (matplotlib)  
- TSNE (sklearn.manifold)  
- Counter (collections)  
- networkx  
- combinations (itertools)   
etc…

2. EDA (탐색적 자료 분석)

* 첨부파일에서 subsample인 ‘frame-level’과 ‘validate’데이터를 사용하여 데이터 탐색  
  - frame-level 파일로부터 Video-level 정보를 추출  
  - frame-level 데이터 읽기  
  - 레이블 탐색하기  
  - EDA (얻은 레이블 간의 관계들을 시각화하기)  
  - validate 데이터 읽기
* 첨부파일에서 ‘Vocabulary.csv’파일을 이용하여 데이터 탐색  
  - EDA (각각의 레이블들 간의 관계들을 시각화하기)  
   ( ex) plot, bar plot, correlations plot, etc)  
  - Vocabulary List
* WordCloud 만들기
* DiGraph를 통하여 카테고리 간의 관계 시각화

3. Modeling (모델링)  
 - train data, validation data, test data를 사용

4. Result : 지금까지의 분석 결과

위에서 언급했던 평가요소들에 관하여 제대로 분석이 이루어졌는지를 확인.