KAIST Video Analytics

0. I set my data with below names

- Video: kaist.mp4
- File Path: Gdrive > /Lecture/Video_Analytics/
- Key JSON File Name: key.json

1. Initial Settings

- Video Intelligence API Package Download
- Environment Variable Setting
- Gcloud Setting

```
# PACKAGE INSTALL
!pip install --upgrade google-cloud-videointelligence
```

```
Requirement already up-to-date: google-cloud-videointelligence in /usr/local/lib/python3.6/dist-packages (1.8.0)
```

```
# GDRIVE CONNECTION
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee

Enter your authorization code:
··········
Mounted at /content/gdrive
```

```
Requirement already satisfied, skipping upgrade: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.6/dist-packages (
```

```
# SET DATA PATH
DATA_PATH = "/content/gdrive/My Drive/Lecture/Video_Analytics/"
```

```
Requirement already satisfied, skipping upgrade: urllib3<1.23,>=1.21.1 in /usr/local/lib/python3.6/dist-packages (
```

```
# CHECK FILE EXIST
!ls "{DATA_PATH}"
```

```
google-cloud-sdk-228.0.0-linux-x86_64.tar.gz  key.json
kaist.mp4                                     segment_df.csv
KAIST_Video_Analytics.ipynb                   shot_df.csv
```

```
# SET ENVIRONMENT VARIABLE
import os
os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = "/content/gdrive/My Drive/Lecture/Video_Analytics/key.json"
```

```
# CHECK ENVIRONMENT VARIABLE
print('Credendtials from environ: {}'.format(os.environ.get('GOOGLE_APPLICATION_CREDENTIALS')))
```

```
Credendtials from environ: /content/gdrive/My Drive/Lecture/Video_Analytics/key.json
```

2. Data Import

- Video Data Read

```
import io
from google.cloud import videointelligence
```

```
# READ VIDEO DATA
with io.open(DATA_PATH+"kaist.mp4", 'rb') as video:
    input_content = video.read()
```

3. Video Intelligence Request

```
# Detect labels given a file path
video_client = videointelligence.VideoIntelligenceServiceClient()
features = [videointelligence.enums.Feature.LABEL_DETECTION]

operation = video_client.annotate_video(features=features, input_content=input_content)

print('Processing video for label annotations:')

result = operation.result(timeout=1500)

print('Done')
```

```
Processing video for label annotations:
```

4. Results to CSV file

```
import pandas as pd

# GET SEGMENT RESULTS
segment_labels = result.annotation_results[0].segment_label_annotations

segment_list =[]
for i, segment_label in enumerate(segment_labels):
    for i, segment in enumerate(segment_label.segments):
        segment_dict = {}
        segment_dict['label'] = segment_label.entity.description
        segment_dict['category'] = '; '.join([category_entity.description for category_entity in segment_label.category_ent

        start_time = (segment.segment.start_time_offset.seconds +
                      segment.segment.start_time_offset.nanos / 1e9)
        end_time = (segment.segment.end_time_offset.seconds +
                    segment.segment.end_time_offset.nanos / 1e9)

        segment_dict['start_time'] = start_time
```

```
            segment_dict['end_time'] = end_time
            segment_dict['confidence'] = segment.confidence
            segment_list.append(segment_dict)


    # GET SHOT LEVEL RESULTS
    shot_labels = result.annotation_results[0].shot_label_annotations

    shot_list = []
    for i, shot_label in enumerate(shot_labels):
        for i, shot in enumerate(shot_label.segments):
            shot_dict = {}
            shot_dict['label'] = shot_label.entity.description
            shot_dict['category'] = '; '.join([category_entity.description for category_entity in shot_label.category_entities]

            start_time = (shot.segment.start_time_offset.seconds +
                          shot.segment.start_time_offset.nanos / 1e9)
            end_time = (shot.segment.end_time_offset.seconds +
                        shot.segment.end_time_offset.nanos / 1e9)

            shot_dict['start_time'] = start_time
            shot_dict['end_time'] = end_time
            shot_dict['confidence'] = shot.confidence
            shot_list.append(shot_dict)

    # LIST TO PANDAS DATAFRAME
    segment_df = pd.DataFrame(segment_list)
    shot_df = pd.DataFrame(shot_list)


    # SAMPLE OF DATA
    segment_df.head()
```

	category	confidence	end_time	label	start_time
0	person	0.772873	52.018633	presentation	0.0
1		0.553012	52.018633	media	0.0

```
    # SAMPLE OF DATA
    shot_df.head()
```

	category	confidence	end_time	label	start_time
0		0.664603	12.379033	media	5.872533
1		0.561513	18.051366	media	12.412400
2		0.785851	20.987633	media	18.084733

```
# DATA to CSV
segment_df.to_csv(DATA_PATH+"segment_df.csv")
shot_df.to_csv(DATA_PATH+"shot_df.csv")
```