×

SANDBOX

Minerva University

December 2021
Seoul, South Korea

© SANDBOX NETWORK Inc.

2020

Contents

1. Team Introduction
2. Project Focus in two areas
3. Thumbnail Analysis
    a. Detailed Project Focus
    b. Approach
    c. Deliverable
    d. Reflection
4. Video Analysis
    a. Detailed Project Focus
    b. Approach
    c. Deliverable
    d. Reflection
5. Wrap-Up

© SANDBOX NETWORK Inc.

Team click.ai

Top row (from left to right)

Saad Bin Ihsan, Nico Gankhuyag, Chris Fok, Eisha

Bottom row

Steven H. Yang

Not in the picture

Hugo Siu

© SANDBOX NETWORK Inc.

2020

Project Focus

Thumbnail

- Project significance
- Approach
- Successes and improvements
- Deliverable
- Team reflection

Video

- Time Stamp of Popular Comments
- Sound analysis
- Learning Outcome
- Improvements

© SANDBOX NETWORK Inc.

Thumbnail Analysis

© SANDBOX NETWORK Inc.

Project significance

Why is Thumbnail significant in the creator economy?

- Human eyes analyze an image in less than **13 milliseconds**
- User spend only **1.8s** looking once
- Thumbnail is the focus of **82%** of browsing and
- The **biggest influence** to watch content as per Netflix

What are in Thumbnails?

- Color
- Text
- Faces
- Optional: localization

© SANDBOX NETWORK Inc.

Approach - **Data Science Project Life Cycle**

1. **Business Understanding**
   a. Netflix case study ✔️
   b. Project plan ✔️
2. **Data understanding**
   a. Data collection ✔️
   b. Data analysis 🚧

3. Data preparation (feature engineering + selection)
4. Modelling (training)
5. Evaluation
6. Deployment

© SANDBOX NETWORK Inc.
2020

Data Collection (success)

Dependent variable

- YouTube API
- View counts, like/dislike counts, favourite counts, comment counts

Colors - independent variable

- OpenCV
- Average hue, saturation, Value

Text - independent variable

- EAST Algorithm
- Scene text percentage cover

Faces - independent variable

- Haar Cascade Classifier
- Number of faces

© SANDBOX NETWORK Inc.

Data Collection (improvement)

Dependent variable

- Cannot simply compare all videos' thumbnail
- Need to control for extraneous variables

Colors - independent variable

- Mean value not conclusive → standard deviation
- Need number of colors and color distribution

Text - independent variable

- Bounding boxes are accurate estimates
- Need fine-tuning

Faces - independent variable

- Not successful
- Need more accurate algorithms like YOLOv5

© SANDBOX NETWORK Inc.

2020

Deliverable (For more detailed information, please refer to the GitHub)

```python
from googleapiclient.discovery import build

def get_stats(video_id):

    # create youtube resource object
    youtube = build("youtube", "v3", developerKey = "AIzaSyDsD5jELu-4jyFRYpeUfOiueSuuBMXz7aA")

    # get the video statistics
    request = youtube.videos().list(part='statistics', id=video_id)
    response = request.execute()

    # return None if request has no result, e.g. private video
    if not response['items']:
        return None

    items = response['items'][0]

    viewCount = items['statistics']['viewCount']
    likeCount = items['statistics']['likeCount']
    dislikeCount = items['statistics']['dislikeCount']
    favoriteCount = items['statistics']['favoriteCount']
    commentCount = items['statistics']['commentCount']

    return viewCount, likeCount, dislikeCount, favoriteCount, commentCount

def add_apidata(df):

    for index, row in df.iterrows():
        stats = get_stats(row["video_id"])

        if stats is None:
            df.loc[index, 'view_count'] = np.nan
            df.loc[index, 'like_count'] = np.nan
            df.loc[index, 'dislike_count'] = np.nan
            df.loc[index, 'favorite_count'] = np.nan
            df.loc[index, 'comment_count'] = np.nan

        else:
            df.loc[index, 'view_count'] = stats[0]
            df.loc[index, 'like_count'] = stats[1]
            df.loc[index, 'dislike_count'] = stats[2]
            df.loc[index, 'favorite_count'] = stats[3]
            df.loc[index, 'comment_count'] = stats[4]

    return df

df = add_apidata(df)
df.head()
```

	date	video_id	thumbnail	view_count	like_count	dislike_count	favorite_count	comment_count	hue	saturation	val
0	2020-07-22	___8hOuoAKw	https://i.ytimg.com/vi/___8hOuoAKw/maxresdefau...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1	2020-08-20	___NoMi5pp0	https://i.ytimg.com/vi/___NoMi5pp0/sddefault.jpg	33654	1004	18	0	141	52.15	45.87	98.
2	2021-07-27	__1e1QJ8-y8	https://i.ytimg.com/vi/__1e1QJ8-y8/maxresdefau...	6645	260	10	0	60	89.38	170.28	67
3	2020-05-14	__3fHmFbnhU	https://i.ytimg.com/vi/__3fHmFbnhU/maxresdefau...	44244	396	17	0	109	56.60	84.42	131.
4	2020-05-06	__4sPuqw6s0	https://i.ytimg.com/vi/__4sPuqw6s0/maxresdefau...	111873	1250	69	0	2	59.55	137.95	136.
5	2021-02-18	__566nRGAt4	https://i.ytimg.com/vi/__566nRGAt4/maxresdefau...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
6	2021-04-11	__5V-i5MfRw	https://i.ytimg.com/vi/__5V-i5MfRw/maxresdefau...	274835	8994	174	0	561	30.75	48.92	185.
7	2021-07-31	__6NlUmDAMA	https://i.ytimg.com/vi/__6NlUmDAMA/maxresdefau...	48019	729	7	0	206	90.20	58.65	156.
8	2020-05-14	__7ba5-FNAc	https://i.ytimg.com/vi/__7ba5-FNAc/maxresdefau...	108761	1192	54	0	480	78.09	141.49	151.
9	2020-06-09	__arsuxE_P8	https://i.ytimg.com/vi/__arsuxE_P8/maxresdefau...	28586	222	12	0	59	50.72	86.79	125.
10	2020-05-14	__bV2M-ZoqE	https://i.ytimg.com/vi/__bV2M-ZoqE/maxresdefau...	240531	6247	79	0	106	15.24	91.36	65.
11	2021-06-11	__CcmYGPCXY	https://i.ytimg.com/vi/__CcmYGPCXY/maxresdefau...	186861	2304	51	0	291	85.12	97.43	203.
12	2021-02-27	__cPg_qlA8c	https://i.ytimg.com/vi/__cPg_qlA8c/maxresdefau...	89349	940	24	0	151	62.73	102.19	132.
13	2020-05-26	__cwYjeVyRA	https://i.ytimg.com/vi/__cwYjeVyRA/maxresdefau...	20346	156	9	0	42	92.02	53.55	94.
14	2021-02-04	__d1g67sGd4	https://i.ytimg.com/vi/__d1g67sGd4/maxresdefau...	22070	525	2	0	85	83.38	102.02	161.
	2021-										

© SANDBOX NETWORK Inc.

Reflection - **Challenges**

- **Steep learning curve**
  - Data science project life cycle
  - New module: YouTube API + OpenCV
  - New algorithms: text detection + face detection

- **Constraints**
  - Assignment deadlines + different workload
  - ***Accurate data collection is crucial before moving on***

- **Needed more support**
  - concrete expectation + directions
  - support for technical onboarding

© SANDBOX NETWORK Inc.

Reflection

Learnings

- Opportunity to work with Sandbox
- Code collaboration on GitHub and feedback

Opportunity

- Continue working on the project to ***refine the accuracy of data (priority) + include more image-related data***
- Possibly work with Sandbox to train a thumbnail-processing model and deploy in the business

© SANDBOX NETWORK Inc.

Video Analysis

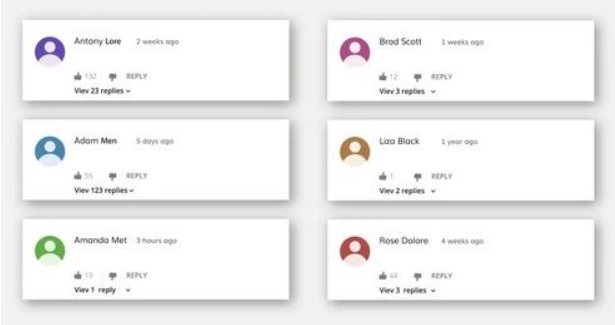
© SANDBOX NETWORK Inc.

Project significance

- Feedback is a major aspect in improving future video performance
- Thus, the analysis of feedback data available through Youtube api is extremely important.

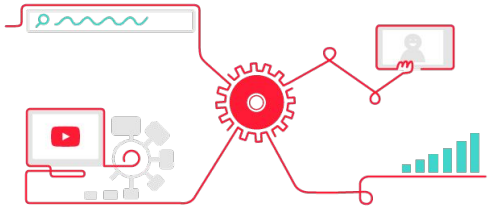
Comment Analysis

- Using timestamps as a measure for traction at different points in a video, we can try to analyse the more popular portions.
- This helps us know which parts are better received by the audience



© SANDBOX NETWORK Inc.

Approach



Data Collection using the Youtube Api

- Extracting all comments using the Youtube api
- Parsing the data to collect all the timestamps
- Using a min-heap to find the 5 ranges of 10 seconds which have the most traction within the video

```
class Ranges():
    def __init__(self,ranges,amount):
        self.ranges = ranges
        self.amount = amount
    def __lt__(self,other):
        return self.amount<other.amount

def clip_parts(times,tophowmany,gif_size):
    start = 0
    end = gif_size
    ranges = []
    if len(times) == 0:
        return ranges
    for a in range(0, max(times)+1,gif_size):
        current_range = Ranges((start,end),len([i for i in times if i >= start and i <= end]))

        if len(ranges) == tophowmany:
            heapq.heappush(ranges,current_range)
            heapq.heappop(ranges)
        else:
            heapq.heappush(ranges,current_range)

        start += gif_size
        end += gif_size
    return ranges
```

© SANDBOX NETWORK Inc.

Gif creation using moviepy

- Downloading the video using the yt-dlp module
- Storing the video as a clip object and using the previously calculated ranges to make the required gifs

```
def clipper(ranges, title = ''):

    for i in ranges:
        test = VideoFileClip("test.mp4")
        if i.ranges[0] < test.duration:
            test = (VideoFileClip("test.mp4").subclip((i.ranges[0]),(i.ranges[1])).resize(0.3))
            test.write_gif(f'{title}test{i.ranges}.gif',program = 'ffmpeg')
            test.close()
        test.close()
```

© SANDBOX NETWORK Inc.

Conclusions

Usefulness of Gif Creation

- Highly useful with a large data set, pointed out areas well received by the audience

For example:

© SANDBOX NETWORK Inc.

Conclusions (continued)

Useful… to an extent

- Even though we get a good amount of information with which areas perform well, it's difficult to put it into place without a large enough dataset
- Smaller youtubers with not as many timestamps will not benefit from this

How it could be used

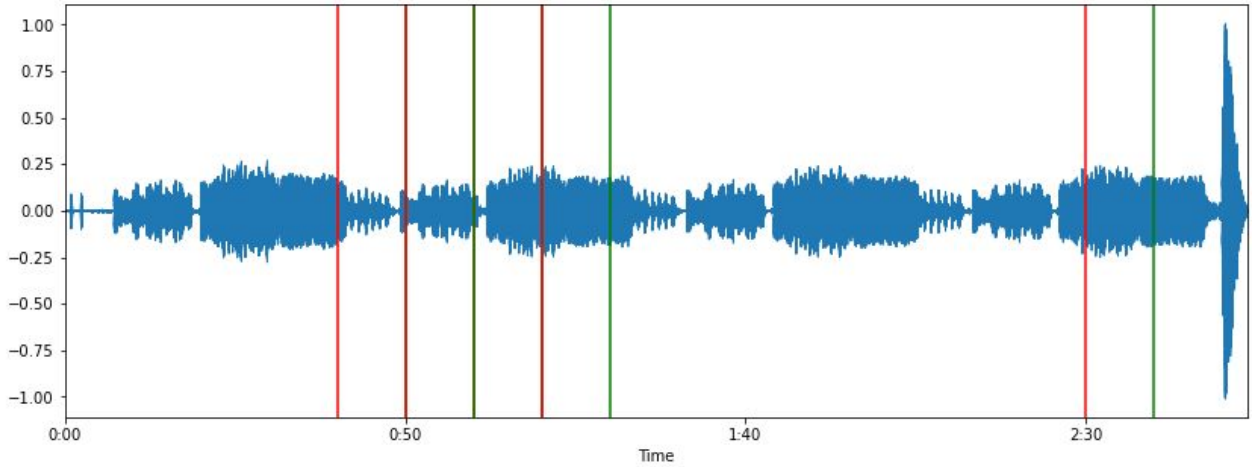
- By looking at larger more popular videos within the same genre and identifying the best parts, we can help smaller youtubers know which areas are better received by the audience.

© SANDBOX NETWORK Inc.

Sound Analysis

Initial exploration

- To see whether there's correlation between the audio levels of the video and the portions of the video that are more popular



© SANDBOX NETWORK Inc.

2020

Approach

Using the librosa module to read audio data (TensorFlow)

- Firstly, the audios for the corresponding videos were downloaded using yt-dlp and converted to .wav format manually.
- Then the read functionality for wav files with librosa was used
- Alongside the waveplot to show the amplitude graph, the ranges were highlighted using matplotlib's vertical line feature

```
for i in range(10):

    import librosa
    x , sr = librosa.load(f"{i}.wav")
    print(type(x), type(sr))#<class 'numpy.ndarray'> <class 'int'>print(x.shape, sr)#(94316,) 22050

    %matplotlib inline
    import matplotlib.pyplot as plt
    import librosa.display
    plt.figure(figsize=(14, 5))
    librosa.display.waveplot(x, sr=sr)
    for j in ranges[i]:
        plt.axvline(x=j[0][0], color = 'red')
        plt.axvline(x=j[0][1], color = 'green')
    plt.show()
    print(x)
```

© SANDBOX NETWORK Inc.

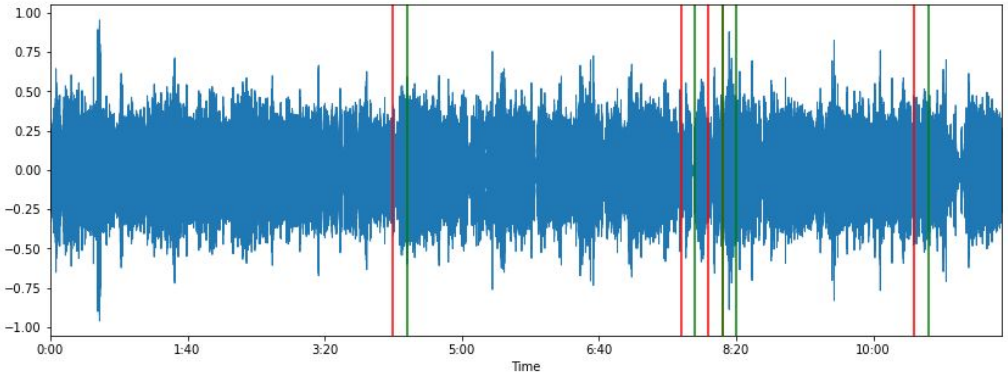
Sound Analysis

Exploration in volume analysis

- Using runningman video as an example

Findings

- results obtained are not significant



© SANDBOX NETWORK Inc.

Sound Analysis

Exploration in sound pitch analysis + confidence

- crepe module
- processes frequency, pitch and confidence

Findings

- useful in evaluating whether youtuber is confident during speeches
- **overlapping** between background and foreground (speech) sounds

time	frequency	confidence
0	0	nan
0.01	1964.225	0.057465
0.02	1961.278	0.075938
0.03	35.996	0.062172
0.04	145.578	0.113675
0.05	163.862	0.10396
0.06	103.483	0.215041
0.07	105.621	0.179718
0.08	161.095	0.286962
0.09	157.39	0.188892
0.1	338.356	0.12647
0.11	173.464	0.133835
0.12	351.153	0.265045
0.13	348.9	0.241365
0.14	347.037	0.256739
0.15	201.131	0.456456
0.16	206.721	0.554744
0.17	209.429	0.201914
0.18	233.71	0.611275
0.19	237.796	0.75827
0.2	241.967	0.665832
0.21	246.281	0.604705
0.22	1024.513	0.520397
0.23	1040.344	0.500229

© SANDBOX NETWORK Inc.

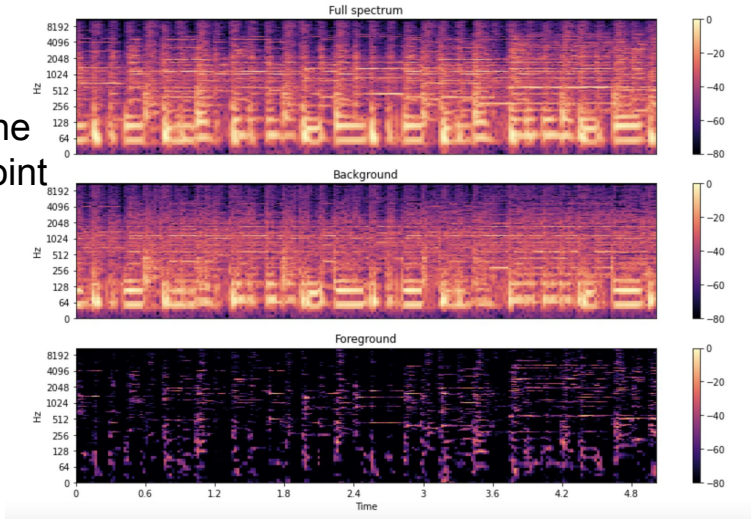
Sound Analysis

Exploration in vocal separation for sound pitch analysis

- sound pitch and effect on popularity
- overlapping between background and foreground (speech) sounds

Findings

- librosa separates the two spectrums
- however, there are many pitches occurring at the same time so results obtained still cannot pinpoint one certain sound



© SANDBOX NETWORK Inc.

Conclusions

Problems

1. ability to separate vocal and background music
    - pitch and volume

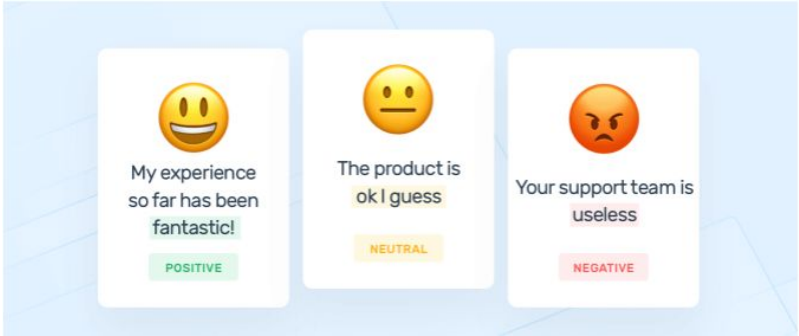
© SANDBOX NETWORK Inc.

Further possible exploration

Comment analysis

- One thing that extracting gifs doesn't tell us is the exact response of the viewers towards the video which we can possibly do using a sentiment analysis on the comments to see the general response and by filtering out the negative responses, see what exactly was criticized and what could have been improved more!

https://monkeylearn.com/sentiment-analysis/



© SANDBOX NETWORK Inc.

Wrap-Up

© SANDBOX NETWORK Inc.

Special Thanks to Joseph, Sungsan, and Jisu from Sandbox Network, Inc. Dahyun from Minerva University.

© SANDBOX NETWORK Inc.