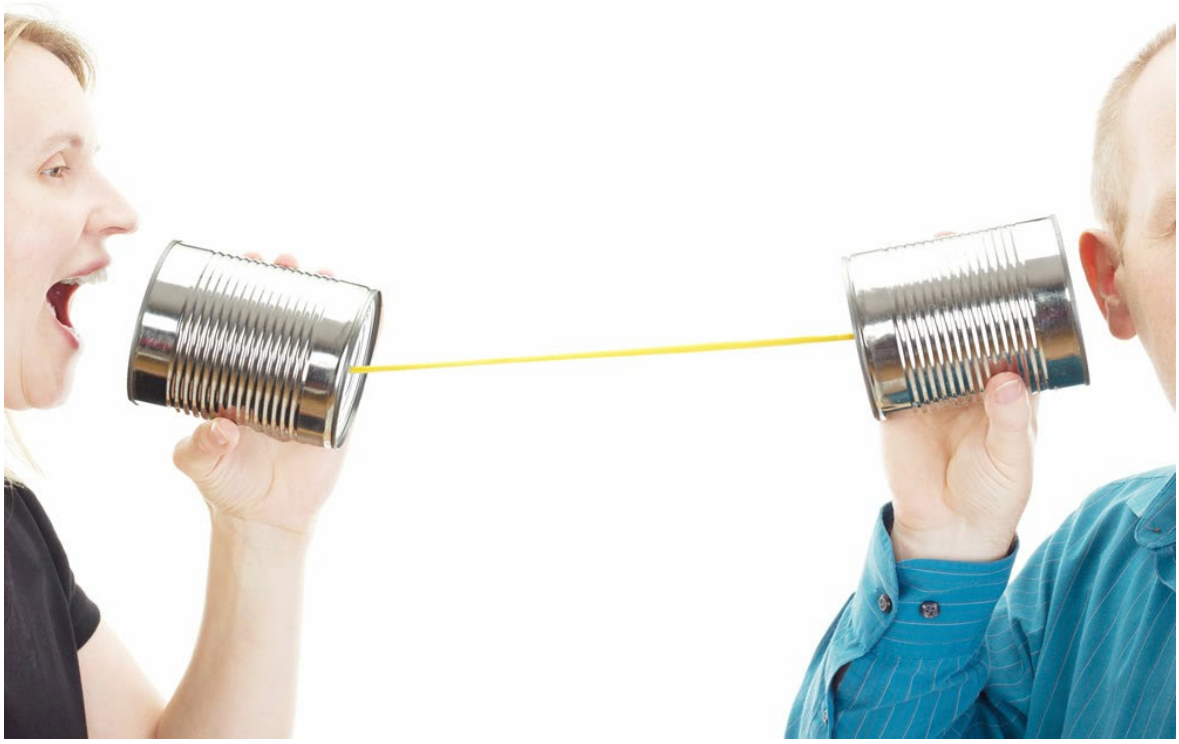


Audio and Video Analysis

Contents

Business Problem

SBIR Intern – Sirs, the submissions for this year are in.
SBIR Judge1 – How many?
SBIR Intern – 450
SBIR Judge1 – Alright. Judge2, you review 200, I will do 200 and we will give the rest for review to Judge3
SBIR Judge2 - Sounds good!

Does it really? This is just one scenario where an entity needs to review submissions/content which can be videos or audio. Other use cases can include an online video platform for the people where the videos need to abide by community guidelines, or analyzing movies to get insights like screen time, sentiment, and tone of the movie. What if there is a system in place that can load, transform, and present key insights from videos and audio without manually going through each one of them? Of course, the insights will be missing the instincts and decision making of a human being. There is a middle ground though. When the files are too large and many, filtering out them based on solid set criteria can save a lot of man hours while presenting with facts like video quality, framerate, number of frames, audio quality, frequent words, optical gradient flow of light, etc. most which the human eyes and ears cannot figure out.

The goal of Matrix is to come up with a similar system that will be performing video analysis on the frames of SBIR's submission videos and NLP on the text that will be converted from audio which in turn will be extracted from the videos. Once this model is in place, we presume that any entity with their own set of criteria will be able to filter out videos and audio for further use. The aim of this project is to extract sentiments from the audio, understand what kind of images are frequent in frames (employees, aerial views, animations man vs woman ratio, etc.) and what is the key terminology used that is common among the winners of this grant. This will enable the judges above to not go through each submission. Thus, if they want videos which have only a certain sentiment throughout, or videos which show a certain minimum participation of women, the system will filter those out and present for consideration.

Value Creation

This system will help achieve the following:

- A comprehensive model that will do everything from extracting the videos from source, converting to audio and text to analyzing and presenting key insights from them.
- Reduction in man hours to go through videos and audio for analysis.
- Automatic filtering of videos and audio based on *custom* criteria (video and sound quality, sentiments, duration, gradient flow, etc.)
- Detecting content that does not follow *custom* guidelines (graphic images, cuss words, negative sentiments, etc.)

Use Cases

The main use case we are focusing on here is to build and use the model to evaluate videos that are submitted for a competition. There are other use cases where this model is of relevance.

These include:

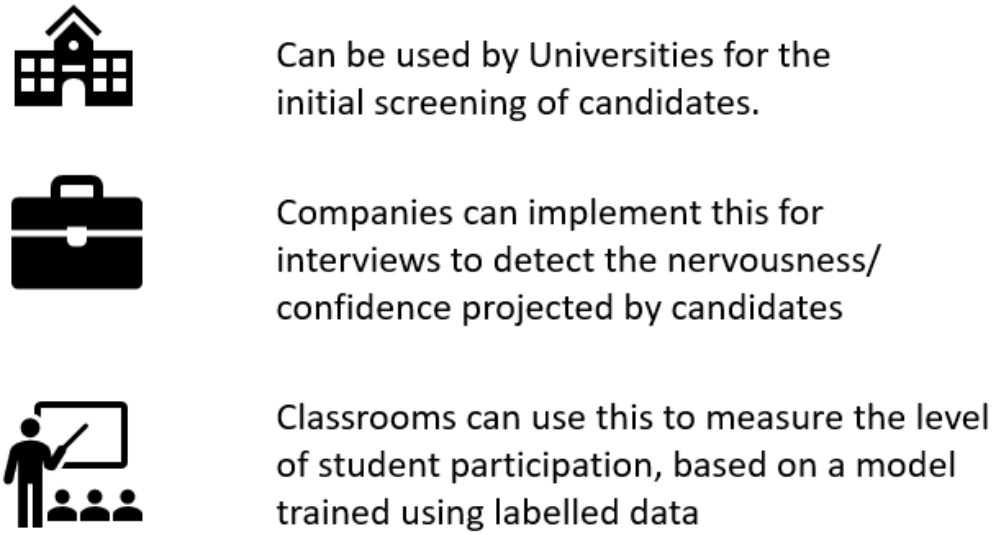


Fig: Other use cases

Road Map

The following steps will be taken to achieve the goal of the project

Data Sourcing (Extraction) - The data which includes primarily of the SBA (Small Business Administration) winners' videos will be taken from the website. Using some of the key words the non-winner videos will be collected.

Data Storage (Loading) - Local

Data Processing (Transformations) - Databricks would be used to access all the videos for transformation. The videos that are collected would be transformed into audio from which the text would be extracted. The images from every frame of the video will also be generated and used for the analysis.

Data Analysis: Various machine learning methods like NLP, emotions detection, image processing etc. would be applied to gauge the commonalities that the winner's video has as compared to the other videos.

Data Visualization: Since we deal with images, the scope of visualization would be limited. However, visuals like word cloud will be used to display the results of the text analysis

Business Solution: The final solution will have key insights about the winners and losers content included in their videos.

Data Information

Primary data is Winners and Losers' videos in mp4 format.

1. Steps to extract Winners videos:

- Get the xpath for the video links on the website https://www.sba.gov/offices/headquarters/ooi/resources/1428931 and save the links in a list.
- Use a function to parse the list and check if the links are from YouTube or Vimeo and use Google's YouTube API or vimeo_downloader to download the respective videos.

Snapshot of Winners videos extraction:

```
page = requests.get('https://www.sba.gov/offices/headquarters/ooi/resources/1428931')
page2 = requests.get('https://www.sba.gov/content/sba-growth-accelerator-winners')
# Parsing the page
tree = html.fromstring(page.content)
tree2 = html.fromstring(page2.content)

# Get element using XPath
link1 = tree.xpath(
'//html/body/div[3]/div[3]/div/div[2]/div[2]/div/div[2]/div/div/div/div/div/div/div/div/div/table/tbody/tr[*]/td[5]/a/@href'

link2 = tree2.xpath(
'/html/body/div[3]/div[3]/div/div[3]/div[1]/div/div[3]/div/div/div/div/div/div/div/div/div/table[1]/tbody/tr[*]/td[4]/p/u/a/

link3 = tree2.xpath(
'/html/body/div[3]/div[3]/div/div[3]/div[1]/div/div[3]/div/div/div/div/div/div/div/div/div/table[*]/tbody/tr[*]/td[4]/a/@hre
```

```
wLinks = link1 + link2 + link3
```

```
print("Number of winners links: ",len(wLinks))
print("Links:\n",wLinks)
```

```
Number of winners links:  275
Links:
 ['https://vimeo.com/343442714', 'https://youtu.be/t-034qZ2WG4', 'https://vimeo.com/302942190', 'https://vimeo.com/343477028/697
e039826', 'https://youtu.be/kgYKvucRzl8', 'https://vimeo.com/343459927', 'https://vimeo.com/343459480', 'https://youtu.be/Ku6ykS
liS4o', 'https://youtu.be/ucPKSXgi-WE', 'https://youtu.be/WGbXD_kdBXM', 'https://vimeo.com/343465853', 'https://youtu.be/832rzEL
DzqI', 'https://youtu.be/0aMxNZcgmzM', 'https://vimeo.com/250361787', 'https://youtu.be/_HQdfkea2Ng', 'https://vimeo.com/3434440
30', 'https://vimeo.com/342870401/40547f3f23', 'https://vimeo.com/343349336', 'https://youtu.be/QJGGRQqG9Jw', 'https://youtu.be/
A7K0WyKX6hI', 'https://youtu.be/3ImcUGgUyr8', 'https://youtu.be/KRmxvBfm3qo', 'https://youtu.be/8RUMmTX-sgI', 'https://youtu.be/
kb2J8syaNcU', 'https://youtu.be/apXZ_jMNm_E', 'https://vimeo.com/343306918', 'https://youtu.be/waN5Fqp-WfA', 'https://youtu.be/o
Gt06M_8hyU', 'https://youtu.be/ztWBopGObTw', 'https://youtu.be/iCTZ8n0fXmw', 'https://vimeo.com/343459964', 'https://youtu.be/nS
G4RYeQD-o', 'https://vimeo.com/343100552', 'https://youtu.be/Gfbe3OyHXk8', 'https://youtu.be/RbN00N9TR10', 'https://youtu.be/k1X
 hq4vG6s', 'https://youtu.be/9K3wqA3qJQE', 'https://youtu.be/ulrxqqw8tGQ', 'https://vimeo.com/343393539', 'https://vimeo.com/343
```

```
Mode: Command    Ln 1, Col 1    finaldraft.ipynb
```

2. Steps to extract Losers' videos:

- Use Google's YouTube and Vimeo API to retrieve links based on the keywords - 'SBA', 'Submission', 'Pitch', 'Award', 'Video'
- Filter the links based on location (we need the videos in the US only), duration (most submissions are in range 2 – 3 minutes)
- Download the videos that satisfy the criteria

Snapshot of Losers videos extraction:

- YouTube API:

First response page for YouTube:

```
_key = 'AIzaSyCV4vNL9avZMH8Zx-ZCQKGbpGgshGGhwsY'
tube = build('youtube', 'v3', developerKey = api_key)
uest = youtube.search().list(q= 'SBA+Growth+Accelerator+Competition+Submission+Pitch+', part = 'snippet', type = 'videos', maxRes
ponse = request.execute()
nt(response)
```

{'kind': 'youtube#searchListResponse', 'etag': '67ptAm1PigEDQLh_ekjUlzHmJNw', 'nextPageToken': 'CDIQAA', 'regionCode': 'US', 'pa
geInfo': {'totalResults': 735, 'resultsPerPage': 50}, 'items': [{'kind': 'youtube#searchResult', 'etag': 'g5jPSrmOZuCzPiS5pf-hjj
mLwjY', 'id': {'kind': 'youtube#video', 'videoId': 'QuBhLSi-bEw'}, 'snippet': {'publishedAt': '2015-07-15T13:56:22Z', 'channelId
': 'UCGgaBr13CNj0hniedqmqEXQ', 'title': 'SBA Growth Accelerator Fund Competition 2015', 'description': 'Small Business Administr
ation competition entry.', 'thumbnails': {'default': {'url': 'https://i.ytimg.com/vi/QuBhLSi-bEw/default.jpg', 'width': 120, 'he
ight': 90}, 'medium': {'url': 'https://i.ytimg.com/vi/QuBhLSi-bEw/mqdefault.jpg', 'width': 320, 'height': 180}, 'high': {'url':
'https://i.ytimg.com/vi/QuBhLSi-bEw/hqdefault.jpg', 'width': 480, 'height': 360}}, 'channelTitle': 'Macomb-OU Incubator', 'liveB
roadcastContent': 'none', 'publishTime': '2015-07-15T13:56:22Z'}}, {'kind': 'youtube#searchResult', 'etag': 'N9_53ScHTOgpFV6XTMS
aT6kBWGU', 'id': {'kind': 'youtube#video', 'videoId': 'Ia5TBHkkm9o'}, 'snippet': {'publishedAt': '2019-06-19T19:04:46Z', 'channe
lId': 'UCvFA53iHXCR2igRvkqEqjFQ', 'title': 'ReaKtor SBA 2019 Growth Accelerator Fund Competition', 'description': "REaKTOR Busin
ess Technology Innovation Center and Peninsula Technology Incubator's submission video for the SBA 2019 Growth Fund Competition.
", 'thumbnails': {'default': {'url': 'https://i.ytimg.com/vi/Ia5TBHkkm9o/default.jpg', 'width': 120, 'height': 90}, 'medium': {'

- Vimeo API

Vimeo API

```
client = vimeo.VimeoClient(
  token='52a53cc4f0f42dfc7e695b8c09859235',
  key='7b28253ac0c2ef37ce17666c5ec634dc120187e2',
  secret='fUdE4KhYT/M/CH+VgTnDHrHac5oi7v/DHqdkmAxSFlmQ5Usg/5VW9MV9I5741Utgi2vX7bICUMtUfb9HpjjAD5qWs6GrMooZPHySsSEbXX+SQTQ2ihQUib
)

vresponse = client.get('https://api.vimeo.com/videos?query=SBA+Growth+Accelerator+Competition&per_page=100')
response = vresponse.json()
```

```
# vimeo dictionary with duration
vimeo_ids = {}
for i in range(len(response['data'])):
    vimeo_ids[response['data'][i]['link']] = response['data'][i]['duration']

print("Job Finished")
```

```
Job Finished
```

Modifying the links that do not follow vimeo_downloader URL syntax:

```
# check for the following vimeo_ids (right hand side) once before executing this block.
vimeo_ids['https://vimeo.com/103911353'] = vimeo_ids.pop('https://vimeo.com/metrixcreate/growthacceleratorfund')
vimeo_ids['https://vimeo.com/231372792'] = vimeo_ids.pop('https://vimeo.com/user54654382/ffvc-2017-sba-growth-accelerator-video-
vimeo_ids['https://vimeo.com/133155287'] = vimeo_ids.pop('https://vimeo.com/406labs/sba-accelerator')
vimeo_ids['https://vimeo.com/133098034'] = vimeo_ids.pop('https://vimeo.com/maufic/sba')
```

Post the filtering, we have the following winners' and losers' data:

	Winners	**Losers**
Accessible Videos	**216**	**145**
Inaccessible Videos	**14**	**6**

Video Metadata:

Winners:

Count	Avg Size (mb)	Avg Framerate	Avg Frame Count	Total Size (gb)
216	42.37	27.34	139	15.96

Losers:

Count	Avg Size (mb)	Avg Framerate	Avg Frame Count	Total Size (gb)
145	32.63	29.87	144	7.6

Audio Metadata:

Winners:

Count	Avg Size (mb)	Recognized Chunks/Video	Unrecognized Chunks	Total Size (gb)
216	27	4.4	0.5	10.28

Losers:

Count	Avg Size (mb)	Recognized Chunks	Unrecognized Chunks	Total Size (gb)
145	46.8	4.08	1.1	8.6

Text Metadata:

Winners:

Count	Avg Words	Most Frequent Word	Unrecognized Files
216	160	'Entrepreneurs'	7

Losers:

Count	Avg Words	Most Frequent Word	Unrecognized Files
145	79	'Accelerators'	21

Architecture Diagram

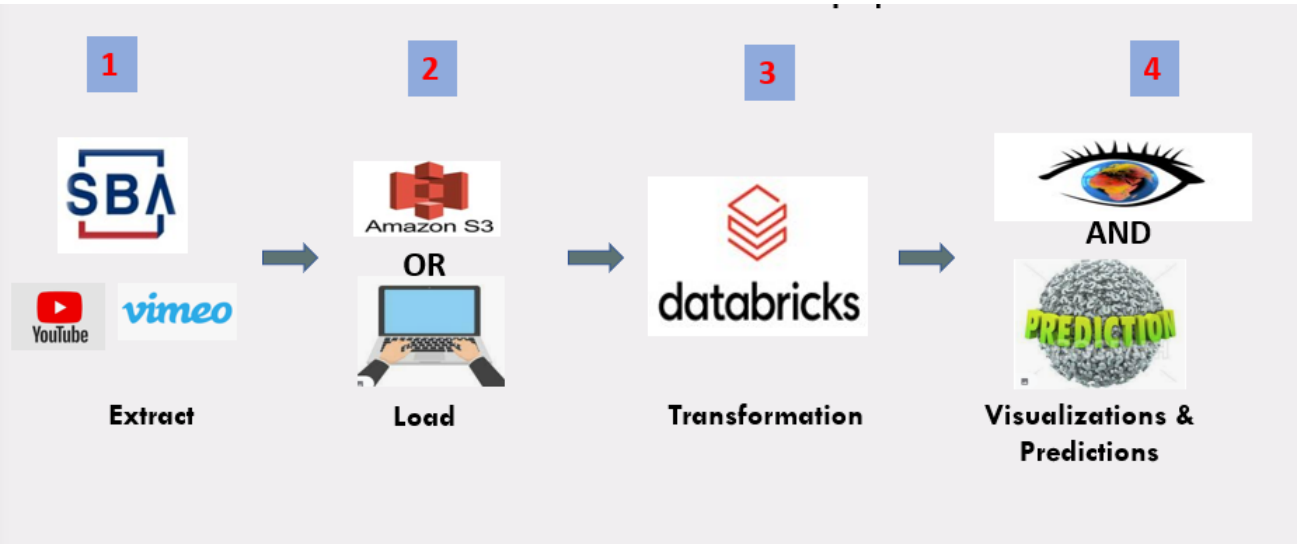


Fig: Architecture Overview

As depicted in the architecture overview diagram, there are 4 major components for the implementation of this project.

The first part is data acquisition. For this, most videos are sourced from the official website of SBA where the video links of pitches by the winners are posted. HTML scraping methodologies are used to extract the YouTube and Vimeo video links. For the next set of videos which mostly includes the startup pitches of other contestants that did not win the grant, we are relying on YouTube APIs (Application Programing Interface) to gather the links of videos based on relevant videos such as 'SBA Growth Accelerator Competition'.

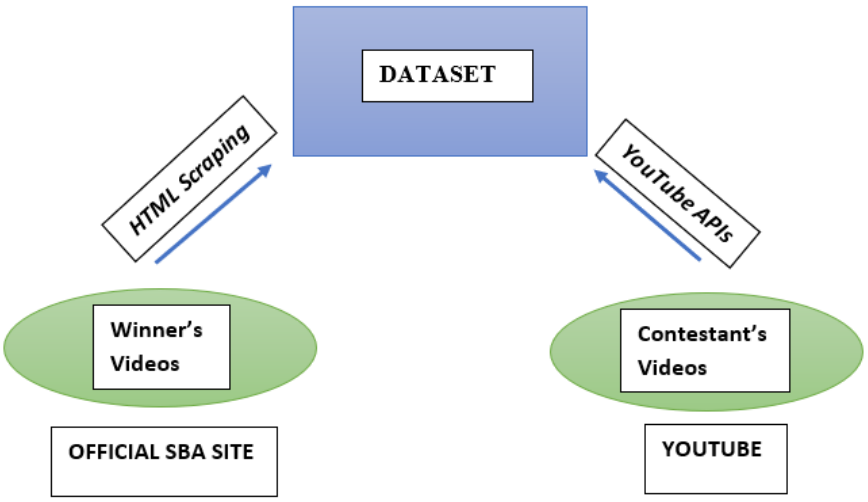


Fig: Structure of Dataset

Data

The second part deals with storage. The entire dataset which comprises of videos of winners as well as that of losers, once downloaded, needs to be stored securely to be later accessed for necessary transformations and ML algorithms. We have 2 options for this:

1. Amazon S3 – After analyzing the performance with S3 and local storage, for this particular use case, we decided with the local storage option

2. **Local storage** - Each of the SBA pitch video is about 2 minutes long and comes approximately 40 mb in size. So, storing the entire dataset in the local machine is also a feasible option. Local Storage usually refers to anything that is "on-premises". In this approach we have the following benefits:

- Speed is one of the main advantages to local storage.
- Storing data on external hard drives is faster than uploading to the cloud.
- We will also have full control of backups, which means better control of who accesses your data.

The third segment deals with data transformation. Since we are processing and transforming massive quantities of data and exploring the data through machine learning models, the most optimum solution is to use Databricks.

Databricks is built on top of Spark and adds exceptionally reliable and performant data pipelines. It offers a distributed cloud computing environment, and has provisions to code in Spark's native R, Scala, Python or SQL interface. Other main advantages offered by Databricks includes:

- Reliable and performant Data Lakes.
- Higher productivity and collaboration - Deploying work from Notebooks into production can be done instantly by just tweaking the data sources and output directories.
- Integrates easily with the whole Microsoft stack - Azure Databricks uses the Azure Active Directory (AAD) security framework. Using AAD allows easy integration with the entire Azure stack including Data Lake Storage (as a data source or an output), Data Warehouse, Blob storage, and Azure Event Hub. However, in this project we will not benefit much from this feature.
- Extensive list of data sources - Aside from those Azure-based sources mentioned, Databricks easily connects to sources including on premise SQL servers, CSVs, and JSONs. Other data sources include MongoDB, Avro files, and Couchbase.
- Suitable for small jobs too - While Azure Databricks is ideal for massive jobs, it can also be used for smaller scale jobs and development/ testing work. This allows Databricks to be used as a one-stop shop for all analytics work.
- Extensive documentation and support available - While Databricks is a more recent addition to Azure, it has existed for many years. Extensive documentation and support are available for all aspects of Databricks, including the programming languages needed.

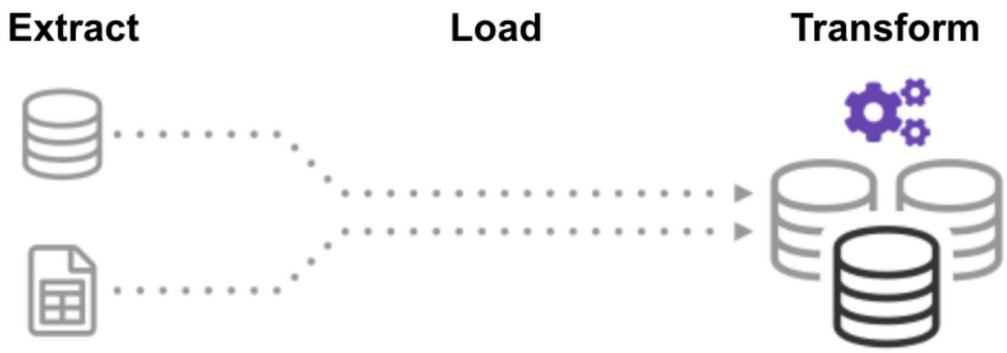
The last portion portrays the output in easily comprehensible formats. The output essentially consists of two parts:

1. Predictions: By using a subset of all the video data for both winning and losing groups as training dataset, the model is trained to predict the outcome of a pitch based on the video data that is supplied to the model. We are using a subset of the total data as test data to validate the model as well.
2. Visualizations: All the observations that we can gather by studying the winner's video as well as by comparison with the loser's video should be projected in an easy-to-understand fashion. We will be relying on NLP to analyze the text along with audio analysis, video sampling and image processing, and other object detection algorithms to arrive at conclusions.

Fig: A sample output of word cloud.

ELT

ELT, standing for Extract, Load and Transform, is the process involved in collecting data from different sources, loading the data and transforming it into a usable resource to fix the business problems.



Extraction

The most crucial aspect of ETL represents Extraction. Since this stage is responsible and linked to the success of subsequent processes. In most of the projects, data is collected and combined from different sources. Another vital aspect of Extraction involves data validation to ensure whether the data collected from the sources have the expected values. During a project, data is sourced from multiple sources, converted into a simplified format for further analysis, and stored in data warehouses.

The extraction for winners and losers are handled separately.

1. Data extraction of winner's videos:

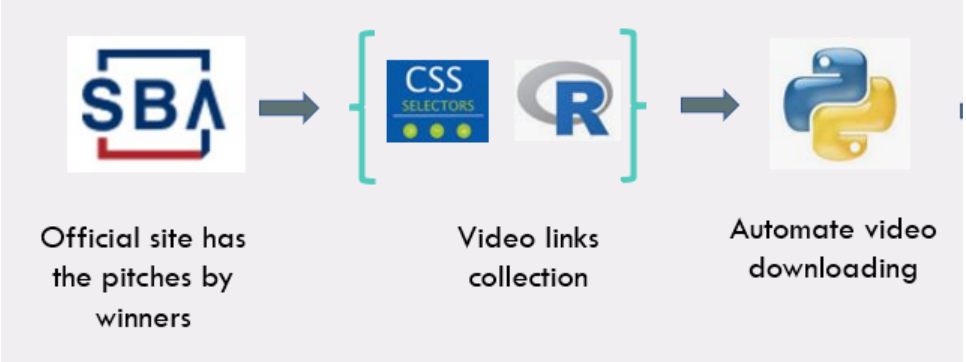


Fig: Data Extraction for winners

Data extraction begins from scraping the SBIR website, downloading the video submissions of hundreds of participants in our project. We have used CSS selectors and R programming to generate links to download the videos from variable xpaths. Our database has videos from 2 primary sources, i.e., YouTube and Vimeo. Links to the videos are stored in a string and then

independent downloading packages for each source to download the videos. We have downloaded the best video quality during the Extraction to ensure proper data transformation and stored the data with chronological naming.

2. Data extraction of loser's videos:

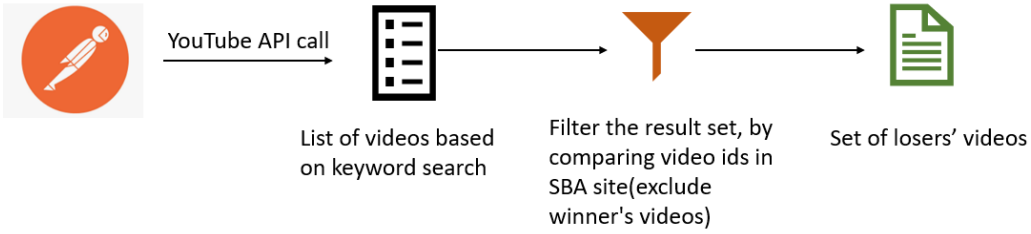


Fig: Data Extraction of other candidates

Postman is an HTTP client that tests HTTP requests along with a user interface through which we obtain respective responses. Google has provided APIs that interact with YouTube. We will be using an API call that provides the YouTube video IDs as response based on the list of keywords sent in the HTTP request.

The format for this request is as follows:

https://www.googleapis.com/youtube/v3/search?q={List of Keywords}&type=video&key={Google API Key}

Once all the set of videos are obtained, we will compare the video ids to that of the winners list which we obtained from official SBA site. These videos along with other search results that are not relevant will be excluded to give the final dataset of losers videos.

Storage

The videos once downloaded in the required quality and formats can be either uploaded to the cloud which is to upload in Amazon S3 buckets or can be stored in the local machine as well.

We are comparing the advantages of each of these to decide the most feasible option for our project.

Advantages of Cloud Storages (Amazon S3)	Advantages of Local Storage
High scalability	High Speed as there is no internet bandwidth dependency
High availability and disaster recovery capabilities	Constant connectivity to access your data
Highly secure during a file transfer session, i.e., when files are being uploaded to your server, files are protected via data-in-motion encryption technologies like SSL/TLS (in the case of FTPS or HTTPS) or SSH (in the case of SFTP).	You have complete control over how the data is stored, who has access, and information security protocols

Transformation

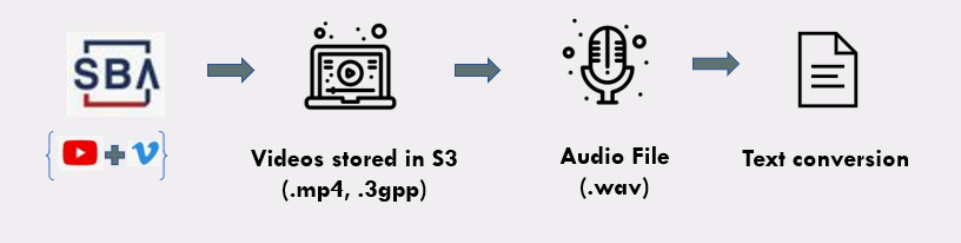


Fig: Data Transformation Overview

The extracted data is being broken down into 2 significant components, images and audio extraction. After extracting the data from web scraping and downloading the relevant videos, the data in the video is transformed into a set of images and audio transcripts. The data set contains more than 250 videos, and each video is comprised of 2 min length, divided further into around 80-100 frames with a time interval depending on the frame rate of each video. In total, there will be an image database of more than 25,000 with different image quality, in-built text, and human emotions, etc. After considering all these factors and linking them with the audio transcript, we have created a realistic dataset for a holistic analysis.

ML Techniques Used

For the analysis there will be 2 main components:

Image recognition - To extract the image frames from the videos and using OpenCV and Sci-kit Image libraries to understand what kind of images are persistent throughout the submission files, getting the male to female ratio of people featuring in the videos, detecting buildings, people or animations present in the videos, etc.

Audio and Text Analysis – To extract the audio from the videos and using SpeechRecognition and CNN (Convolutional Neural Network) model to analyze metrics like the general tone of the video, sentiments of speakers and frequent words used.

1. VIDEO FRAMES

- **STEP 1: Extract frames from all the videos**

Analysis of the frames within the video will be performed post extracting the frames from the videos. To do this the following steps were taken:

Step 0: Import OpenCV
Step 1: Get *framerate* of the video using cap (5)
Step 2: Parse through every *frame* using cap (1)
Step 3: If the *framerate* divides the *frame number,* write the *frame* as a '.jpg' file to the video's folder.

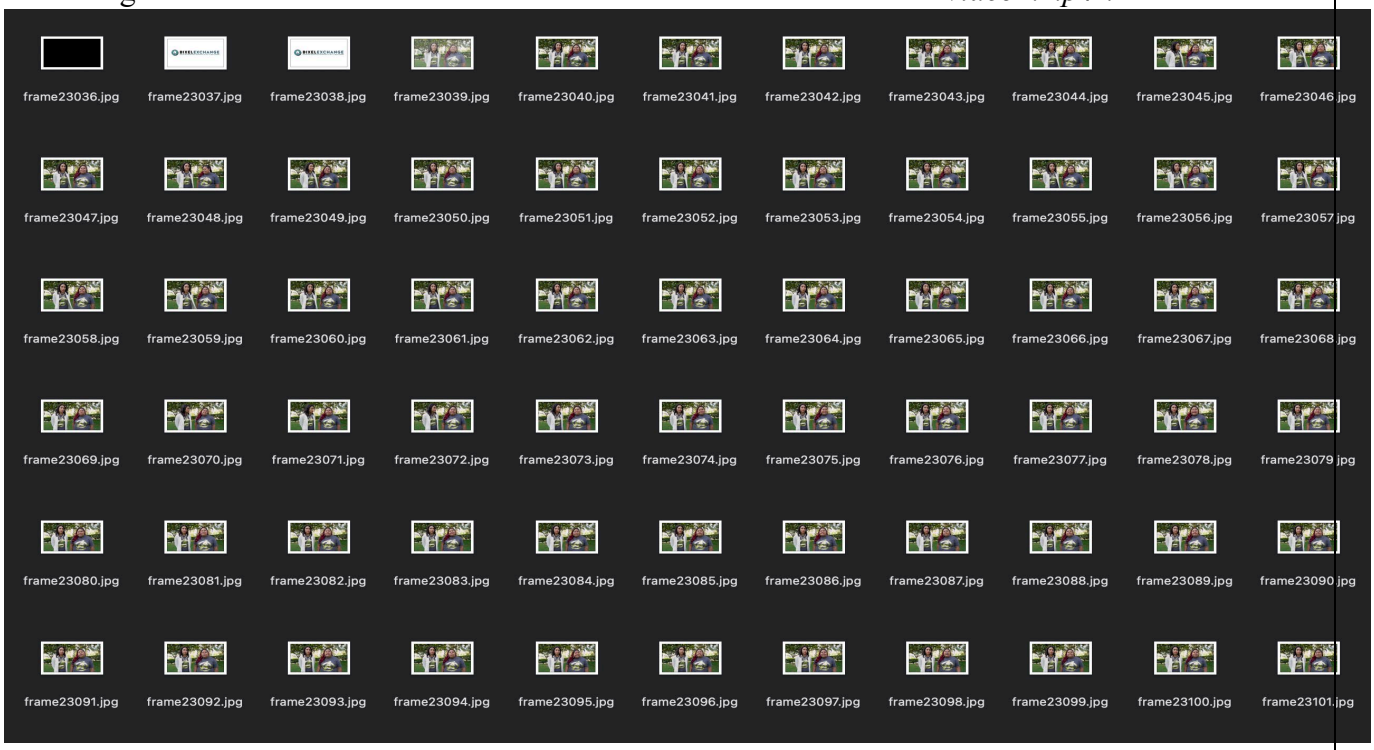
All the usable frames of the video are now ready to use.

Following is the code snippet for the extracting frames:

```
videoFileName = 'video' + str(i) +'.mp4'

cap = cv2.VideoCapture(videoFileName)# capturing the video from the given path
frameRate = cap.get(5) #frame rate
x=1
videofolder = re.sub('.mp4','',videoFileName)
os.mkdir(videofolder)
print('Creating folder',videofolder,'and saving frames...')
counter = 0
while(cap.isOpened()):
    video_path = os.path.join(path, videofolder)
    os.chdir(video_path)
    frameId = cap.get(1) #current frame number
    ret, frame = cap.read()
    if (ret != True):
        break
    if (frameId % math.floor(frameRate) == 0):
        filename ="frame%d.jpg" % count;count+=1
        cv2.imwrite(filename, frame)
```

Following screenshot holds all the frames extracted from a video named *'video1.mp4':*



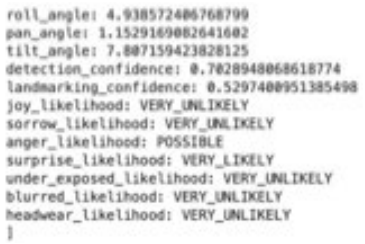
- **STEP 2: Use GCP to derive frames information**

GCP's Cloud Vision API provides an efficient method to extract details of an image. Following functionalities are available for image recognition:

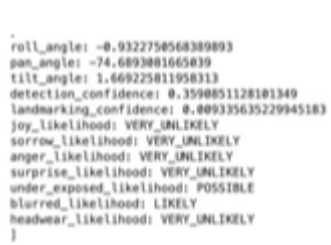
- Object Detection (presence of various objects within the frame like table, chair, windows, etc)
- Face recognition (position within the frame and likelihood emotions like joy, sorrow, anger, surprise, etc)
- Text Detection (words and numbers present within an image)

For this project's purpose, after reviewing the information returned by the different APIs, *Face_Detection* was chosen as the primary goal was to analyse the sentiments of people when talking in the video. The plan was to derive the 2 prominent emotions in both *Winners* and *Losers* videos and study how the emotions differ in them. The steps to get the 2 emotions:

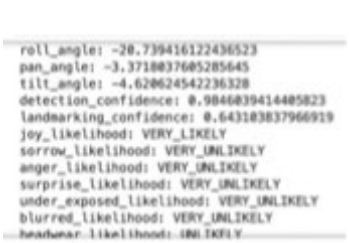
1. Send 8000 images (frames) via batch request to Cloud Vision APIs. GCP allows 8000 images per batch for an asynchronous API request. The limit is 300 batches per day. Total number of frames were ~32000, thus 4 batch request calls had to be made.
2. The response from *face_detection* is a protobuf variable storing all the details of the faces in the frames. To extract the likelihood of emotions, Protobuf needs to be converted to a dictionary format, post which using key-value pairs, the key (emotion) can be extracted by matching the value (likelihood) from the dictionary. Following are snapshots of the Protobuf variable returned by the Cloud Vision API:



Anger



UnderExposed



Joy

3. For a single video the likelihood of all the emotions will be stored in a dataframe with emotions as the column-name and their value as 1 if most likely and 0 if unlikely.
4. Following the same method, dataframes for all the videos will be created. From these dataframes, the columns with the highest number of 1s will be stored as the Prominent.Emotion1 column value and the second highest number of 1s will stored in Prominent.Emotion2 column in the video data frame. Once this is done, we will have a data frame that looks like the following:

Source	Video	Win/Lose	Prominent.Emotion1	Prominent.Emotion2
YouTube	video74.mp4	W	Joy	Surprise
Vimeo	video48.mp4	L	Joy	Surprise
Vimeo	video226.mp4	W	Joy	UnderExposed
Vimeo	video266.mp4	W	Joy	-
YouTube	video20.mp4	L	Joy	-
YouTube	video36.mp4	L	Joy	-
Vimeo	video254.mp4	W	Joy	Blurred
Vimeo	video108.mp4	W	Joy	Surprise
YouTube	video259.mp4	W	Joy	Blurred
Vimeo	video34.mp4	L	Joy	-

Here '-' means there weren't any other emotions detected by the API

2. AUDIO ANALYSIS

We are analyzing the audio present in the format in two different manners:

1. By converting the audio to text and employing text analysis
2. Converting the audio to images and performing CNN modelling

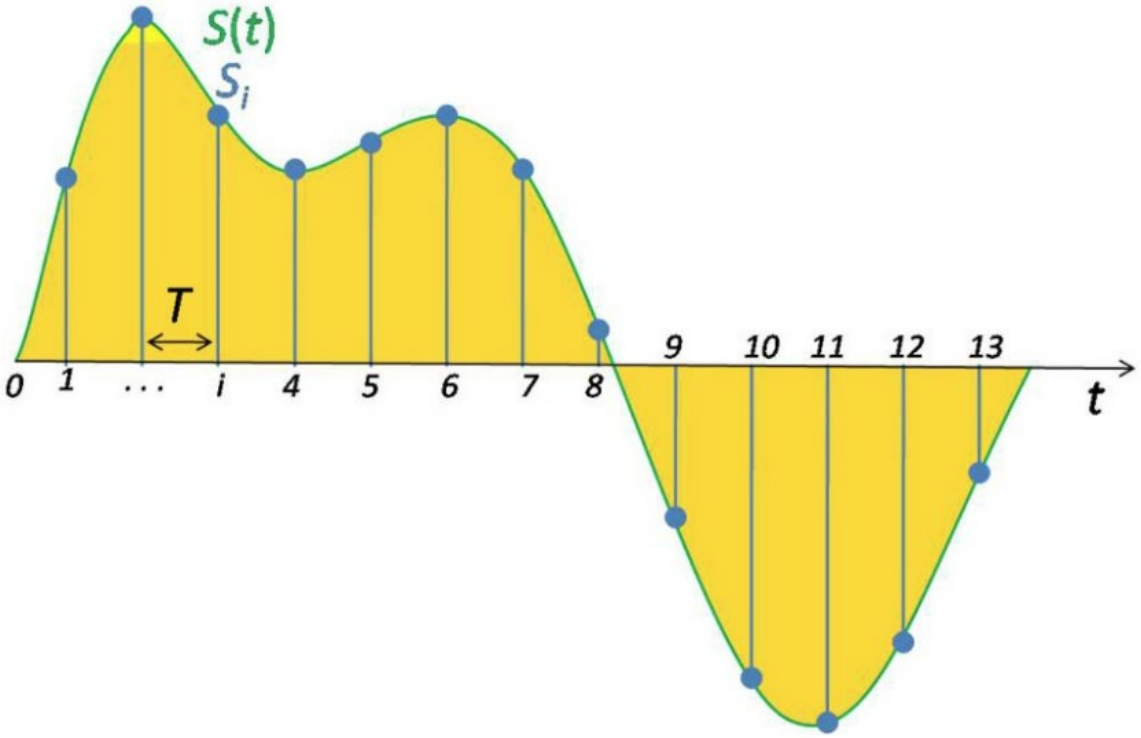
The first method is explained in detail in the text analysis part. In this section we will talk about the digital representation of sound.

Detecting sentiments through audio is one of the primary interests in such a competition to understand the confidence and tone of the participants. Listening to all the pitches and understanding the tone of the participants could be a strenuous task considering the volume of registrations. Even in the event of limited registrations, the interpretation of the tone and emotions conveyed through the pitches will be interpreted in different ways by different people leading to a difference of opinion. We often think of audio data as just data we interpret and process through our auditory system, but that doesn't have to be the only way

that we analyze and interpret audio signals. So, what is the alternative? Converting the audio files into a picture such that we can visually see and analyze the pitches than just hearing it. The visuals can be in the form of a waveform plot or a spectrogram.

Spectrogram is more helpful than a wave form since it reveals the hidden insights of an audio than the waveform. It is a representation of frequency over time in the form of a 2D graph and the third dimension of amplitude represented by color. It denotes the intensity of the signal at a frequency of time and helps in distinguishing between noise and the audio that will be of interest to interpret. All the audio that was generated from the video in the form of wav file was converted into respective spectrograms such that one spectrogram represents one audio file.

In physics, sound is a vibration that propagates as an acoustic wave, through a transmission medium such as a gas, liquid or solid. The sound signals often repeat at regular intervals so that each wave has the same shape. The height shows the intensity of the sound and is known as the amplitude. Here we are more interested in representing the audio digitally. Audio sampling is the process of transforming a musical source into a digital file. Sampling is a method of converting an analogue audio signal into a digital signal. While sampling a sound wave, the computer takes measurements of this sound wave at a regular interval called sampling interval.



Each of the intervals here is a sample. To get a context, a common sampling rate is about 44,100 samples per second. That means that a 10-second music clip would have 441,000 samples.

Although this is a representation of sound, here we consider spectrums that can give more details about sound waves.

Spectrograms

A spectrogram is a visual way of representing the signal strength, or “loudness”, of a signal over time at various frequencies present in a particular waveform. Not only can one see whether there is more or less energy at, for example, 2 Hz vs 10 Hz, but one can also see how energy levels vary over time.

Spectrograms are basically two-dimensional graphs, with a third dimension represented by colors. Time runs from left (oldest) to right (youngest) along the horizontal axis. The vertical axis represents frequency, which can also be thought of as pitch or tone, with the lowest frequencies at the bottom and the highest frequencies at the top. The amplitude (or energy or “loudness”) of a particular frequency at a particular time is represented by the third dimension, color, with dark blues corresponding to low amplitudes and brighter colors up through red corresponding to progressively stronger (or louder) amplitudes. To generate a spectrogram, a time-domain signal is divided into shorter segments of equal length. Then, the fast Fourier transform (FFT) is applied to each segment. The spectrogram is a plot of the spectrum on each segment. The Frame Count parameter determines the number of FFTs used to create the spectrogram and, as a result, the amount of the overall time signal that is split into independent FFTs.

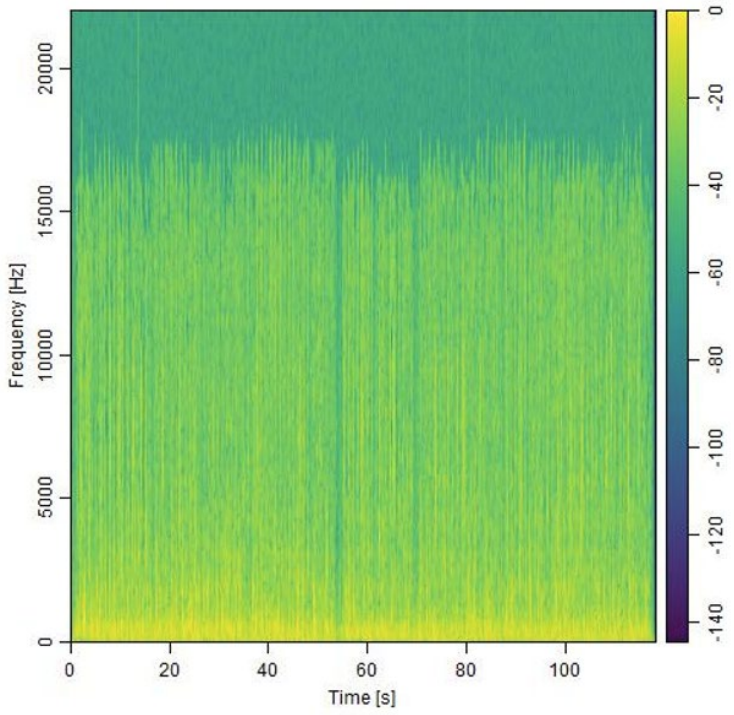


Fig: A sample spectrogram

We converted the losers and winners audio to spectrograms by using the package TuneR in R. We noticed the pattern distinctions in some of the videos that clearly show high energy (bright colors) spread across time in winners video compared to that of losers.

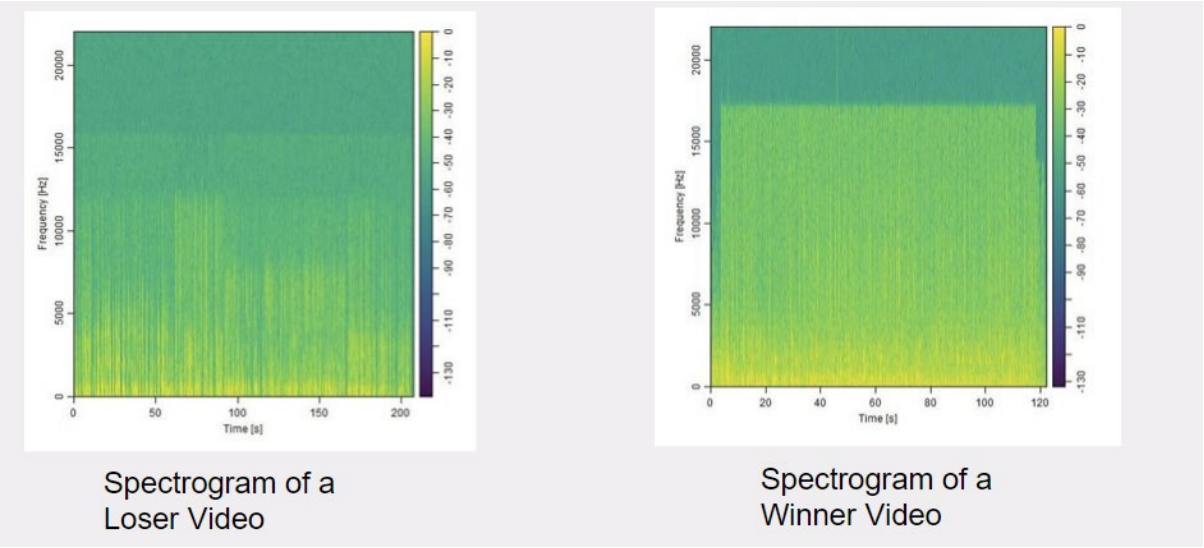


Fig: Winner and Loser spectrograms

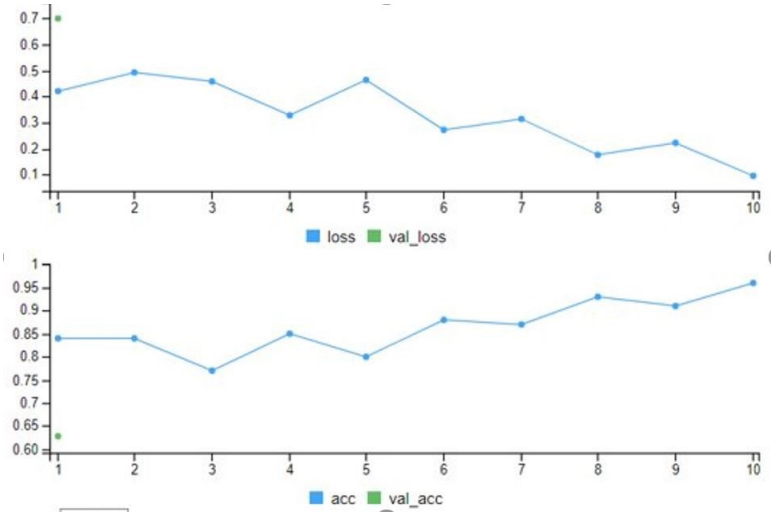
Once all the winners and losers' videos were converted into spectrograms they were used as image inputs to perform CNN analysis. A Convolutional Neural Network (CNN) learns to recognize patterns across spacial data. It has been proven to be successful in identifying objects, signs and sometimes even faces. This deep learning algorithm takes an image as an input to detect and assign importance to all the features of an image to differentiate between the other images. For example, CNN will recognize components of an image (lines, curves, etc.) and then combine these components to recognize objects/faces, etc. CNN are used for image recognition and classification due to its high accuracy and not overfit the data. Research shows that CNN performs well when it comes to validation data proving that the overfitting problem is not there.

Convolutional networks are composed of an input layer, an output layer, and one or more hidden layers. A convolutional network is different than a regular neural network in that the neurons in its layers are arranged in three dimensions (width, height, and depth dimensions). This allows the CNN to transform an input volume in three dimensions to an output volume. The hidden layers are a combination of convolution layers, pooling layers, normalization layers, and fully connected layers. CNNs use multiple conv layers to filter input volumes to greater levels of abstraction. This acts as an advantage when the data is augmented, and CNN can analyze the images from various angles and find out the similarities.

We used the following arguments and values in the CNN model

Function	Purpose	Value (if applicable)
sequential keras	The sequential keras model is used primarily. It is the easiest way to build a model in Keras. It allows us to build a model layer by layer. The 'add()' function is then used to add layers to our model. Our first 2 layers are Conv2D layers.	
layer_conv_2d	This layer creates a convolution kernel that is convolved with the layer input to produce a tensor of outputs.	
Filters	Filters is a set of learnable weights which are learned using the backpropagation algorithm. CNN do not learn a single filter. They learn multiple features in parallel for a given output.	32 as an initial filter to 512 filters
kernel size	kernel size is an integer or list of 2 integers, specifying the width and height of the 2D convolution window. It can be a single integer to specify the same value for all spatial dimensions.	Considering the images are spectrograms a kernel size of 3,3 is used.
Activation function	Activation function is the node that defines the output of the node given an input or a set of inputs. Some of the activation functions are Linear function, Sigmoid, Exponential, ReLU etc. Research shows that ReLU function is the most preferred for CNN. ReLU converges six times faster than tanh and sigmoid activation functions.	ReLU
Input shape	Input shape is the dimensionality of the input (integer) not including the samples axis. This argument is required when using this layer as the first layer in the model. It implies the height, weight and depth of the image.	(150, 150, 3)
Pooling	The max pooling method is used. It is nothing but a pooling operation that selects the maximum element of the feature map covered by the filter. Thus, as a result of doing max pooling, the output would contain more prominent features of the previous layer's feature map.	max pooling
Pool Size	Pool size involves the pooling operation like the filter to be applied to the feature maps. The pool size is usually smaller than the size of the feature map.	2,2
Layer Flatten	layer flatten is used finally to merge all the visible layers and reduce the file size. This makes the final output in a neat form to analyze and produce results.	

The model was created using the above-mentioned arguments and was used to train and validate the set of winners and losers' spectrogram



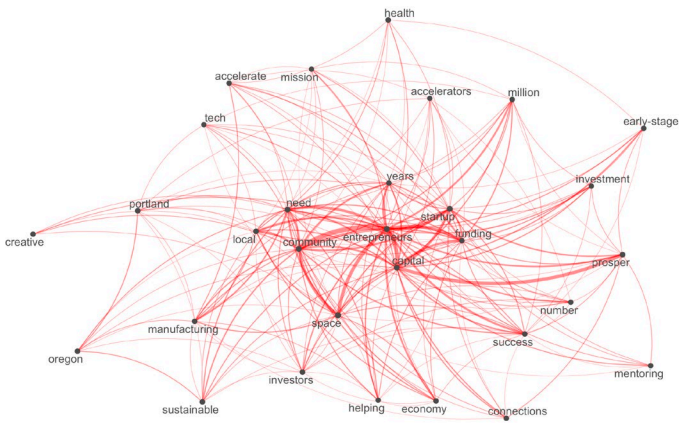
The performance of the model was observed through the loss and accuracy charts. Loss value implies how poorly or well a model behaves after each iteration of optimization. An accuracy metric is used to measure the algorithm's performance in an interpretable way. The accuracy of a model is usually determined after the model parameters and is calculated in the form of a percentage

It was observed that the accuracy of the model was improving, and the loss was reducing after each iteration such that the model finally gave an accuracy of 94 percent. This implies that we can distinguish between the winners' and losers' video. The model can be applied for future videos to see whether a video pitch could likely be a winner or a loser.

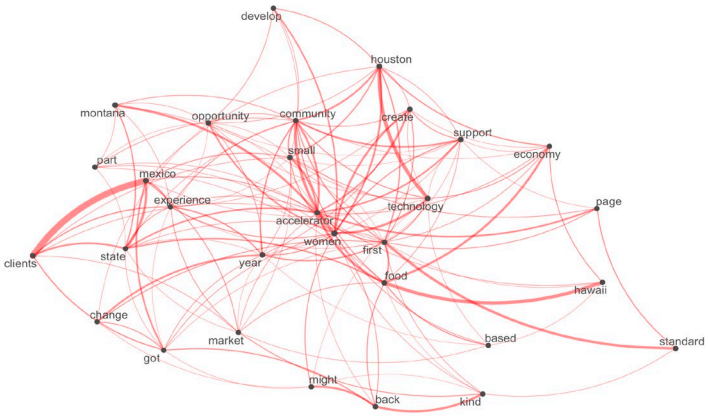
3. TEXT ANALYSIS:

We used R for text analysis. Using ‘quatnteda’ package we were able tyo derive the the most used words and phrases sorted by their ‘z-score’ values

Winners:



Losers:



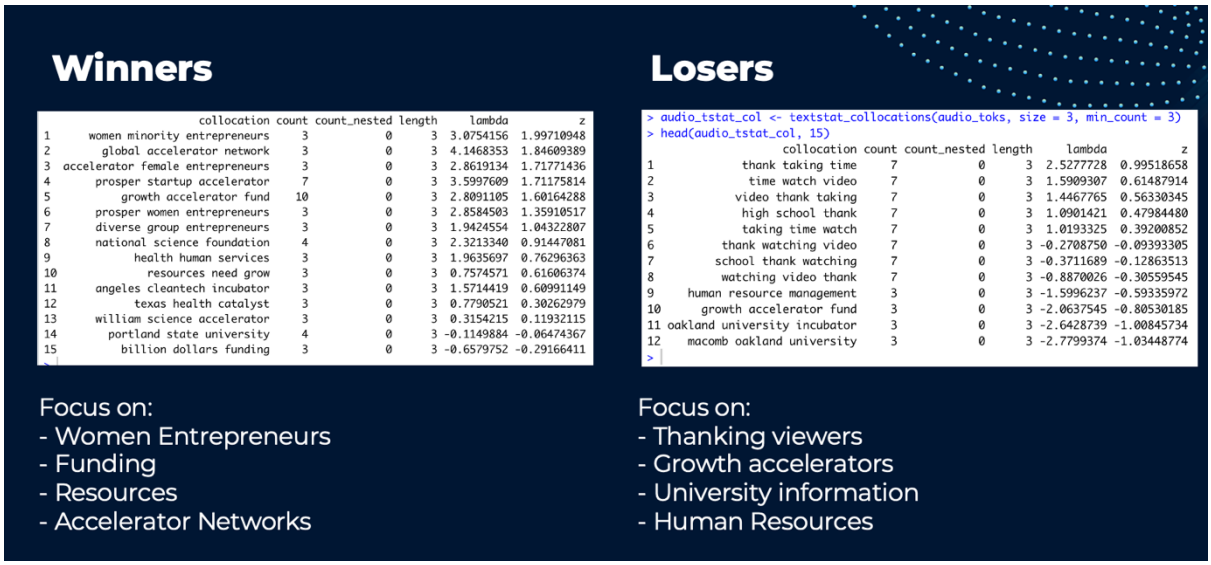
Common connected words:

- Community
- Funding
- Women

Common Connected Words:

- Accelerators
- Food
- Oklahoma

TextStat Collocation Analysis:



CHALLENGES

Every project comes with some challenges and so does this project too. Some of the challenges that are expected for the project are as follows

Video Sourcing: The SBA website has a set of videos from each year. However, these are the videos of only the winners of the Growth Accelerator Fund competition. The primary aim of the project being to find what defines the winners and distinguish them from the losers. It becomes a necessity to fetch the videos of the other participants too. Finding a set of key word lists that could extract all the videos of the participants and getting the relevant videos of the participants after feeding in the keywords will be challenging.

Conversion Validation: The are various formats of data conversion that is dealt at various stages. Every video will be converted to audio to extract the text. This would involve validation of data and quality at every step. As a part of a trial run for a few videos, it was found that the videos that are sourced do not take a uniform format. They differ between

3GPP or MP4 format. It was found that the accuracy of audio and text conversion of the videos were not perfect. This could lead to loss of information and misleading the analysis and findings of the project. Hence the data validation programmatically and to some extent manually could be involved which would be a cumbersome process.

Time and complexity: The SBA website have about 40 to 60 videos of the winners every year since 2014. Each of these videos spans 2 mins each. The images would be taken for every frame and each video must be converted to audio to further process into text. This also involves data cleaning like removing the unclear images and removing stop words before any machine learning model can be applied. It becomes even more challenging when it must be done for the other participants' video. Additionally, the time taken to run the image processing, NLP and other Neural network models to detect the emotions for each of the videos and summarizing them would be challenging given the expected volume of videos and within the period that we are targeting for completion of the project.

LEARNIGNS AND TAKEAWAYS:

Team Learnings:

- Team Management: We were able to split the work among ourselves strategically and addressed the issue as the overall project was gigantic and difficult for any 1 or 2 person to handle the code aspect of the project.

- Dedication: I was able to learn how dedicated my teammates are for the project and that motivated me to do put more hours trying to finish the project as much as I can.

- Time Management: We have kept deadlines after our project proposal and divided the project into chunks. We were optimistic in dividing the project but got stuck in few aspects due to other reasons.

- Contribution: I personally feel that I should have had knowledge of video analysis in prior to contribute more to the team and overall project.

Technical Learnings:

- Video analysis was new to each one of us and using GCP, analyzing a frame was very interesting as I was intrigued by the number of features GCP Cloud Vision API has and the minute details it captures.

- We have used R and Jupyter Notebook extensively and I was able to come through many new libraries and their use cases.

FUTURE WORK

- The next step would be to combine 3 machine learning models integrated under a functional API Keras, which can do the analyses under the 3 models, i.e.,
    - Audio,
    - Video, and
    - Text. A video passed through this functional API will give an output considering 3 aspects of the video.
- We have faced an issue in doing text Detection solely from images. Any text overlapped with an image or graphic has shattered the text detection. This needs to be addressed.
- Optimizing frames: Current video analysis captures and analyses all the frames in a video. For Ex: A video is 2 min length, and we capture frames after every second. This video comprises 120 frames. Although all these 120 won't be of much use. But, to build an optimized ML model, we need to capture only a few frames and remove the duplicates using a function. We will be using pyscene and pydetect functions.
- Electronic Music Data Analysis: Background music in all the videos will be analysed to understand the music used and its relevance.

REFERENCES

- OpenCV for Python: https://www.geeksforgeeks.org/python-process-images-of-a-video-using-opencv/
- Image and video processing in Python: https://www.sciencedirect.com/science/article/pii/S2405844019362206
- Using Google Cloud Video Intelligence and Cloud Vision APIs: https://www.qwiklabs.com/focuses/1831?parent=catalog
- Analyzing video using Python, OpenCV and NumPy: https://medium.datadriveninvestor.com/analyzing-video-using-python-opencv-and-numpy-5471cab200c4
- https://global.hitachi-solutions.com/blog/6-reasons-to-use-azure-databricks-today
- https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html