



pindrop[®]

Improving Voice-Bio Using Free Data:
Audio-Visual Diarization of Youtube Videos

Final
Presentation

Desmond Caulley
*Georgia Institute of Technology
Research Intern, Pindrop*

About Me

- Born in Ghana, West Africa
- Grew up 20 miles east of Atlanta
- B.S at Cornell University (2016)
- M.S at Georgia Tech (2017)
- Ph.D Student at Georgia Tech

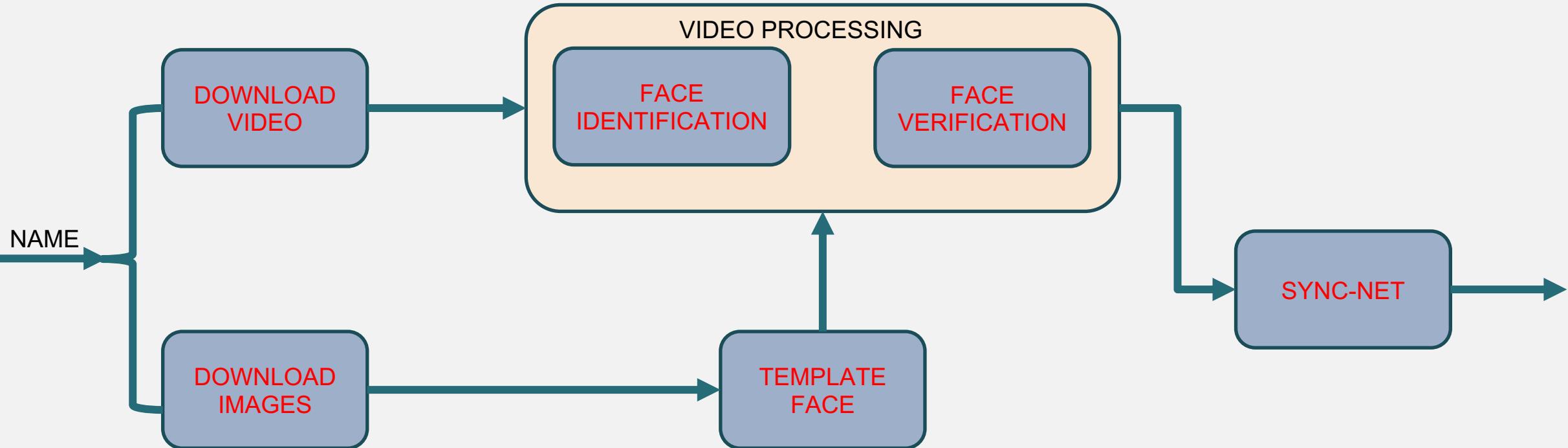


Summer Project Overview

- Free additional data to improve voice bio
- Audio-Visual Diarization
- Identify a person of interest in a video
- Verify that the person is talking



SYSTEM OVERVIEW



Step 1: Compiling List

- Celebrity List: Korean Celebrities
- Compiled list of 130 Celebrities
 - Actors, musicians, politicians



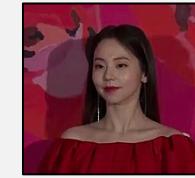
Step 2: Template Faces

Objective: We need a face with which to compare the video faces to verify person of interest

- Search celebrity face using google:
 - Download top 25 faces returned
 - Ex: **안소희 얼굴**

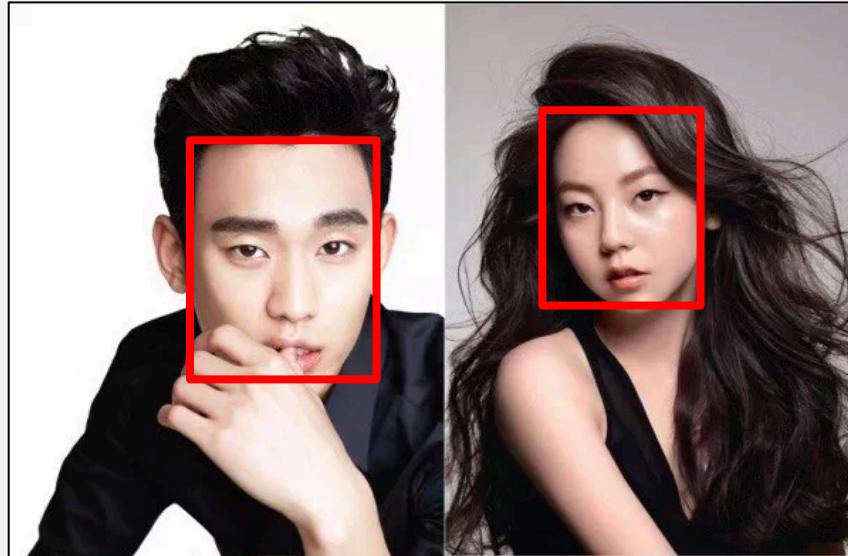
celeb
name

face



Creating Template Face

Task 1: Face Detection



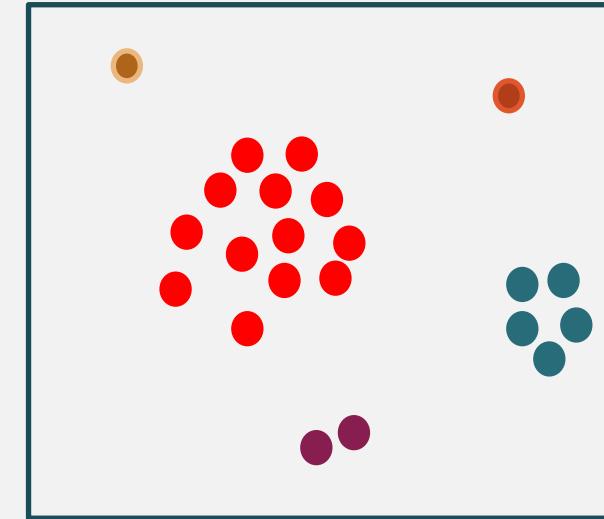
We ran this on all faces in all pictures

Creating Template Face

Task 2: DBSCAN Clustering

- Cosine scoring between the 2622-dimensional embedding
 - Threshold is set 0.4
 - Means if distance between two embeddings < 0.4 , then they can't be immediately be put together
- DBSCAN finds clusters with high density and builds from those

VGG Embedding Clustering



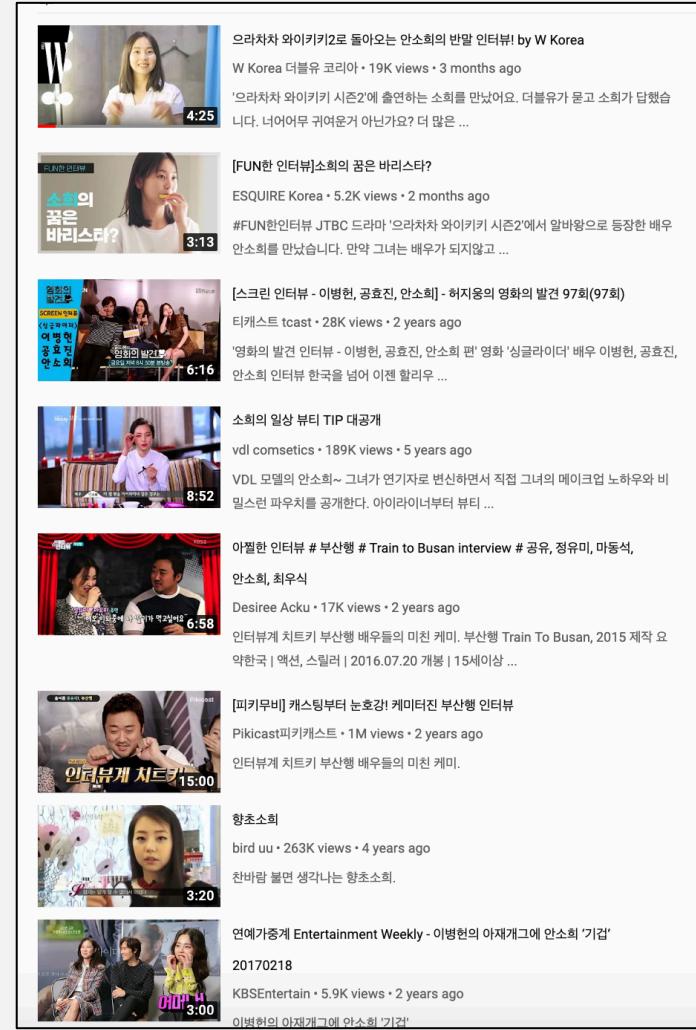
TAKE MODE TO GET LARGEST CLUSTER (Assuming it's the person of interest)

Store embedding for comparison later

Output: 2 2 1 0 3 2 2 2 -1 1 2 2 2 2 1 0 1 2 2 2 -1 2

Video Search and Processing

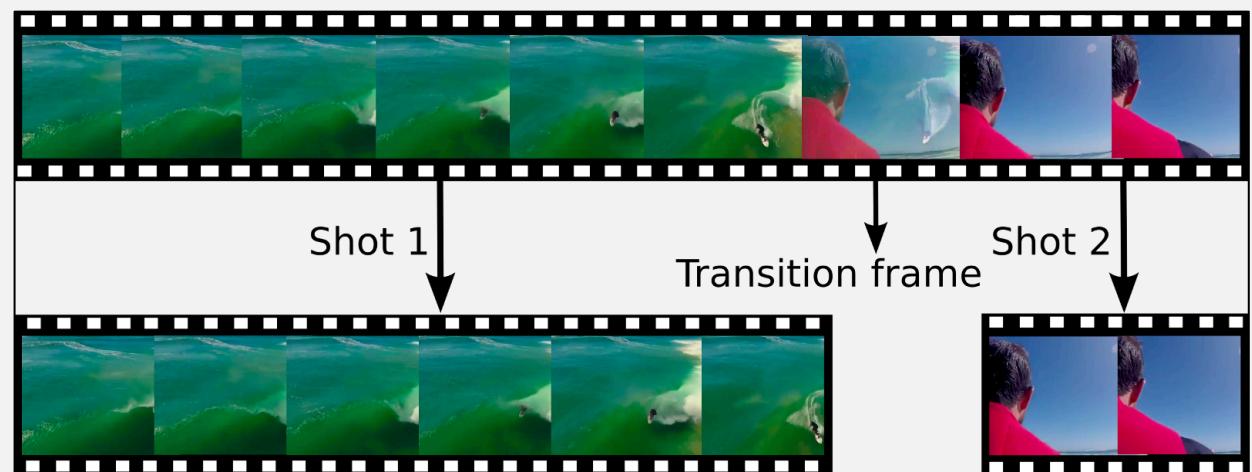
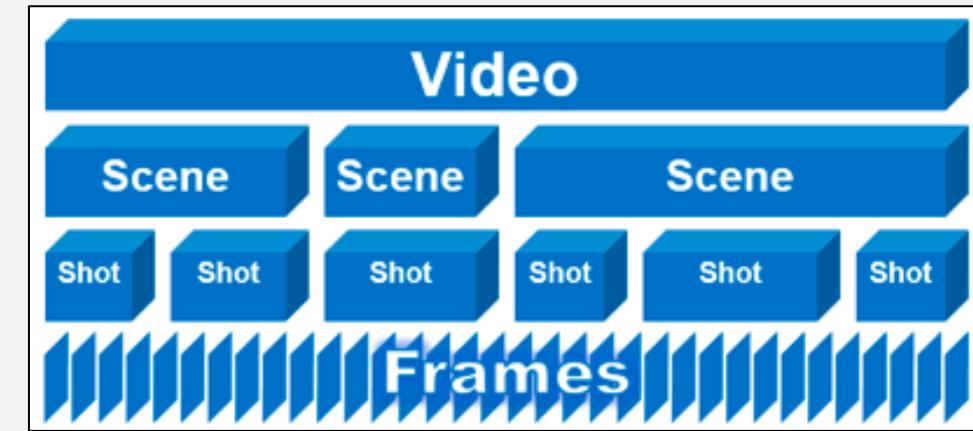
- Search youtube and download top search for celebrity
 - [Youtube-dl](#)
 - [Download top 30 videos](#)
- Search using Korean names
 - [Append word interview](#)
 - Ex: [안소희 인터뷰](#)



Step 3: Scene Detection

Objective: Identify scene changes in a video. Might signal to do face detection since people or objects present changes in a scene

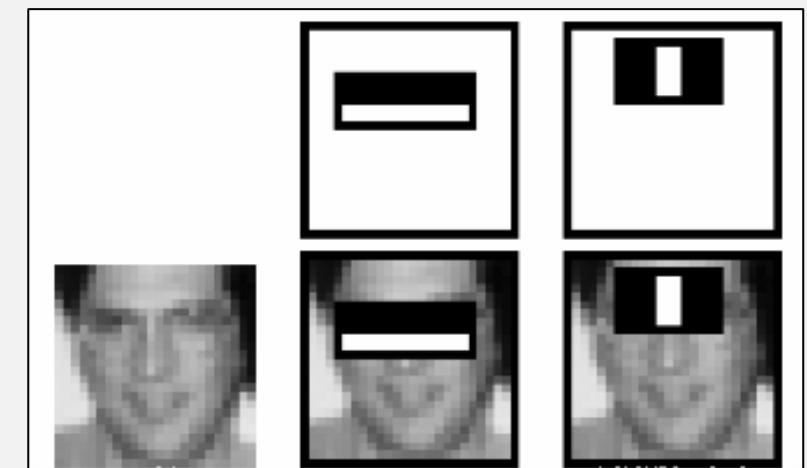
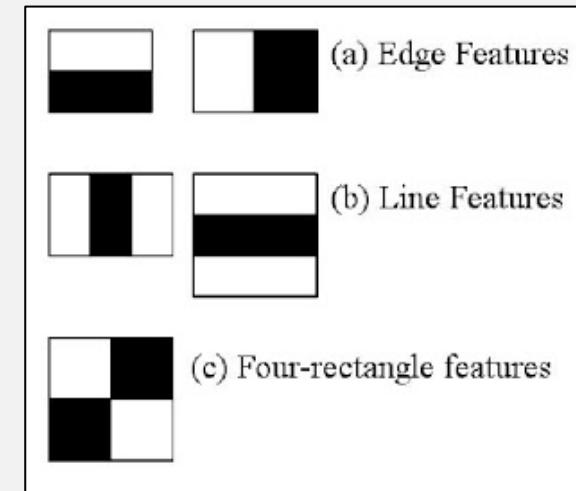
- Python pkg – `pyscenedetect`
- color histogram-based scene detection algorithm in the HSV/HSL color-space
- Store frames where scenes transitioned



Step 4: Face Detection (HAAR BASED)

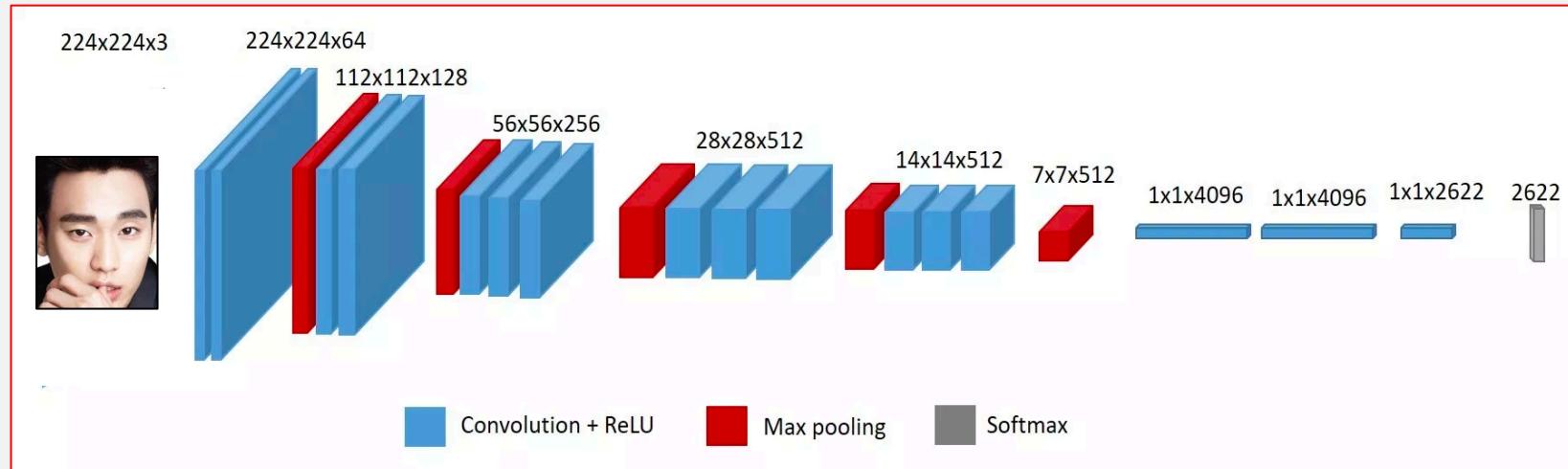
Objective: Identify faces in a particular frame before we can compare and check if it's person of interest

- Change image from RGB to gray scale.
 - Train on positive and negative images
- HAAR FEATURES – 16000+
 - 1st feature: region of eye darker than region of cheeks
 - 2nd feature: eye is darker than bridge of nose
- Weak Classifier (Adaboost)



Step 4: Face Verification

Objective: Compare detected face in video to template face learned from earlier step



- Extract features for detected face → `detected_face`
- Compare feature to template face → `template_face`
 - `val = cosine(detected_face, template_face)`
- If $\text{val} \geq 0.6$ → There is a match!!

WHAT NEXT?

Attempt 1: Detect and Verify and Track Using Histogram of Distribution

Find face in every frame and then do verification on that!!

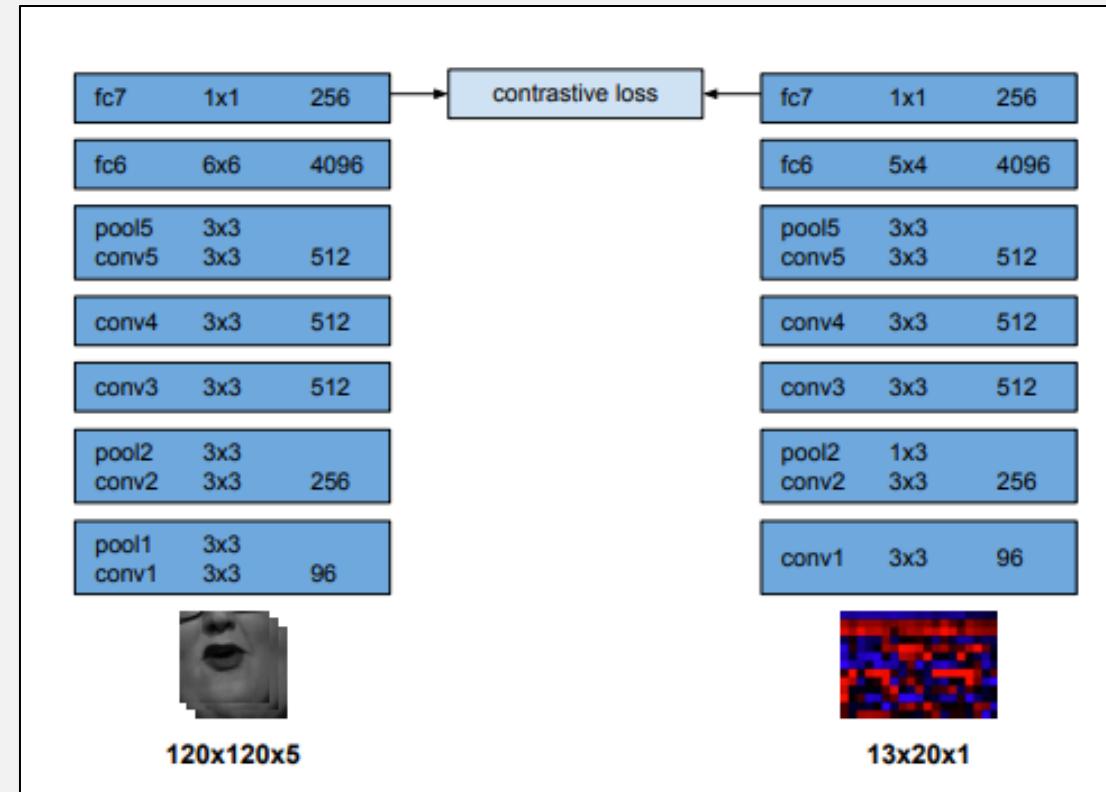
ISSUE:

- Keep histogram of pixels of that face
- Deviations from original histogram means we need to do face detect and verification

Step 5 : SyncNet

- Synchronize audio to lip movement
- Take Mel Frequency Cepstral Features from audio
 - Short term power spectrum on mel scale
- Take Images from video as input
- Train two models simultaneously.
 - One model for audio
 - One model for images on video frame

Syncnet



Models Trained Simultaneously

BUT WHAT IF THE CELEBRITIES ARE NOT
SOOO POPULAR?

OR LANGUAGE OF INTEREST DOES NOT
HAVE ENOUGH CELEBRITIES

TUNISIAN DATA

ISSUE:

BUT WHAT IF THE CELEBRITIES ARE NOT
SOOO POPULAR?

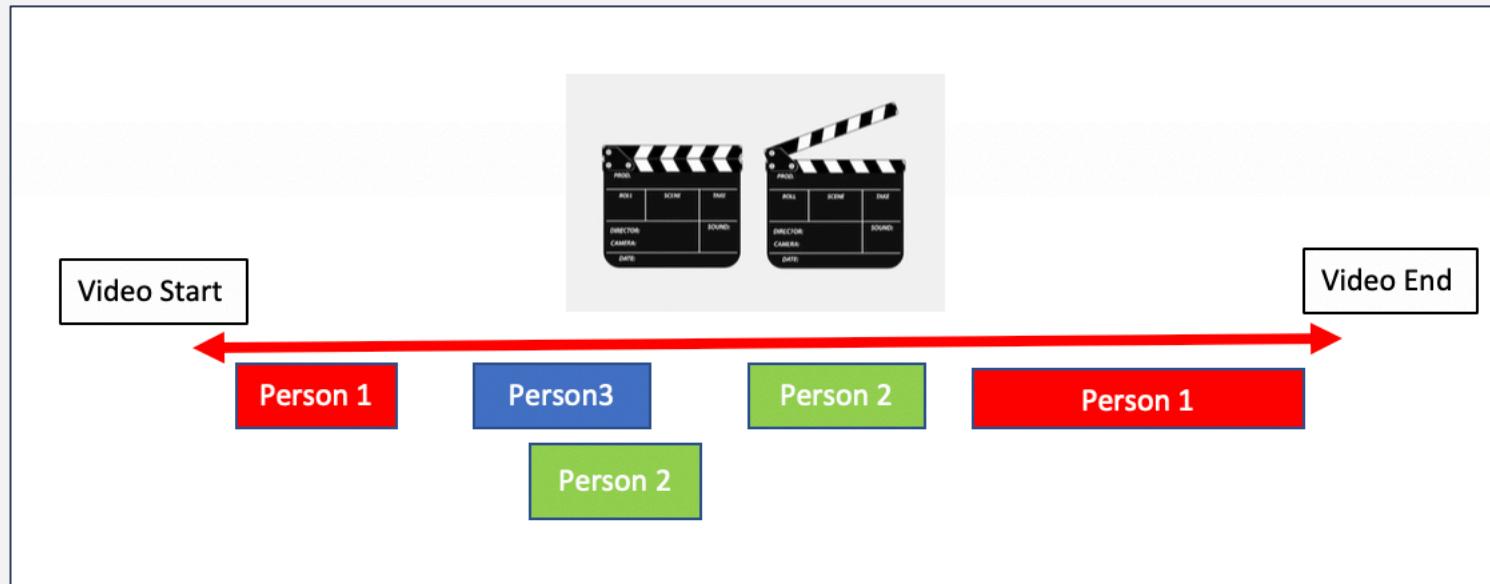
OR LANGUAGE OF INTEREST DOES NOT
HAVE ENOUGH CELEBRITIES

SOLUTION:

USE RANDOM PEOPLE

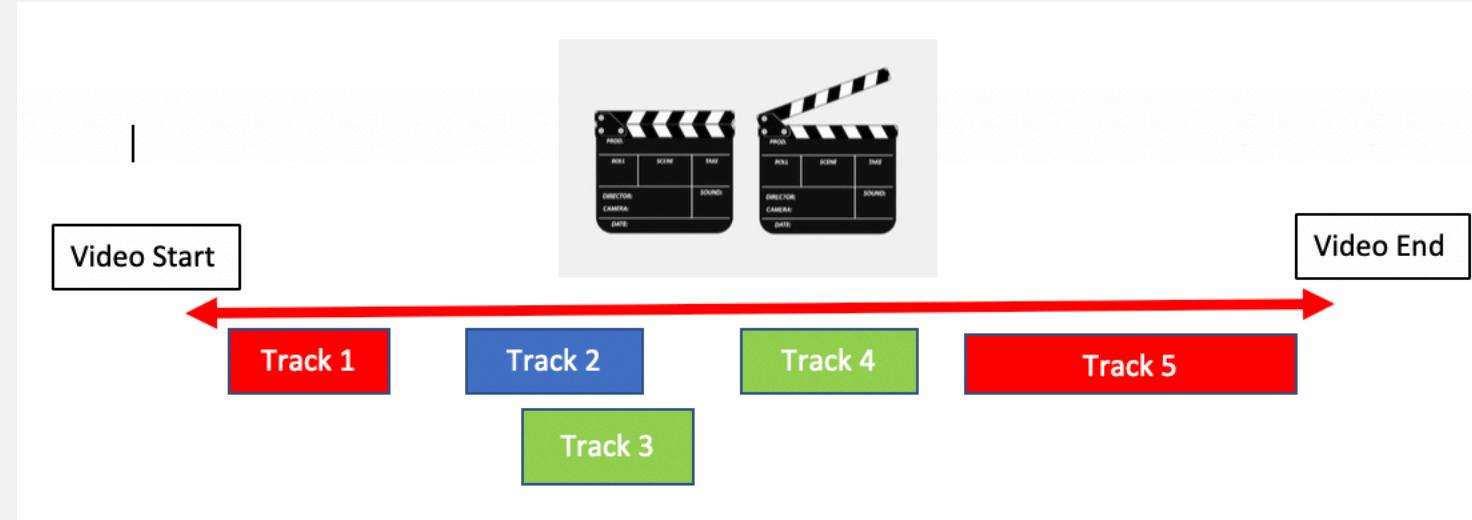
Step 1: Within Video Diarization

Face Tracking of all people in a video and then clustering them



Step 2: Sampling Faces From Each Track

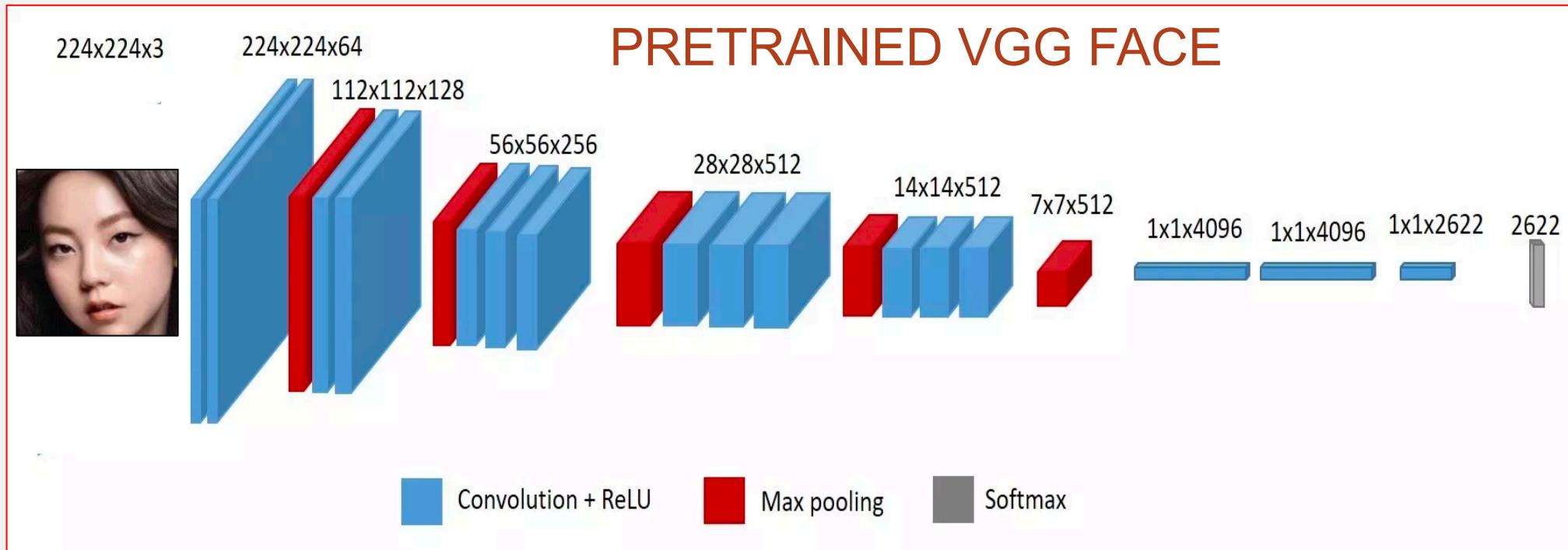
Select 50 frames and corresponding faces in each track



- Track1 → 50 random frames
- Track2 → 50 random frames
- Track3 → 50 random frames
- Track4 → 50 random frames
- Track5 → 50 random frames

Step 3: Extract Face Features

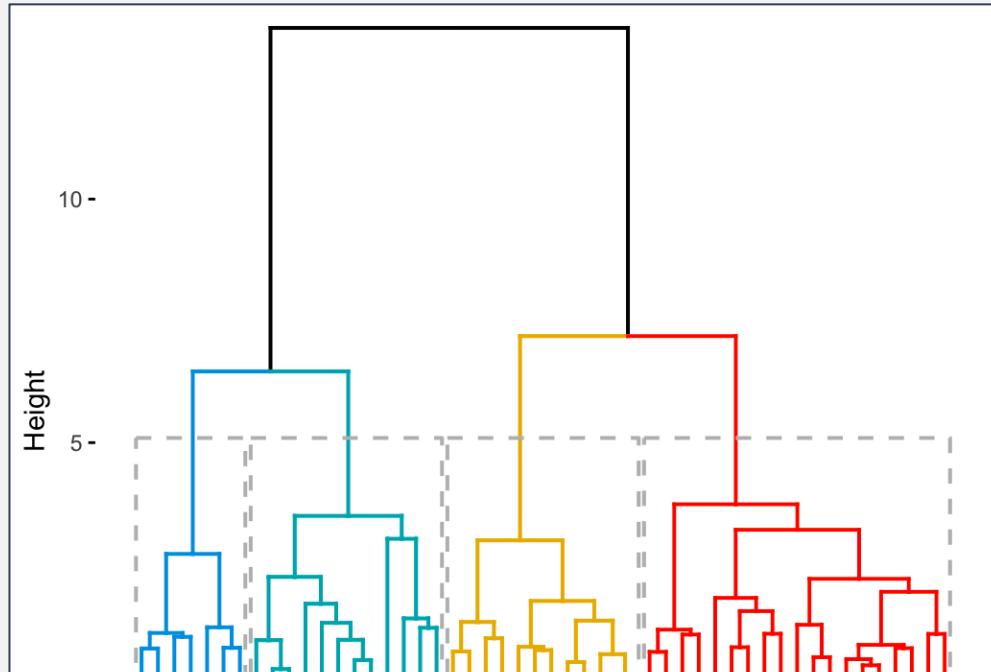
Extracting VGG Face Features for Each of
50 frames in Track



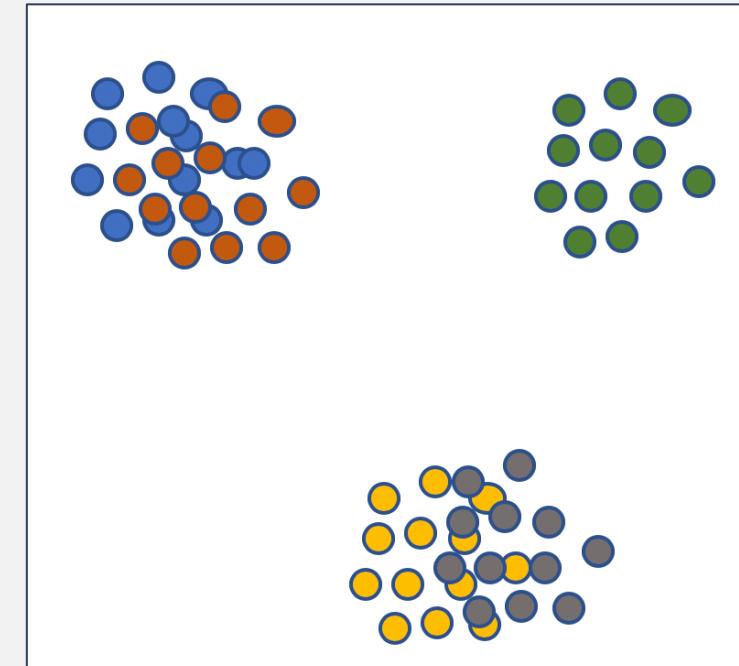
Step 4: Clustering the Tracks

Tracks for same Person Should be closer together

Agglomerative Clustering: threshold = 0.4



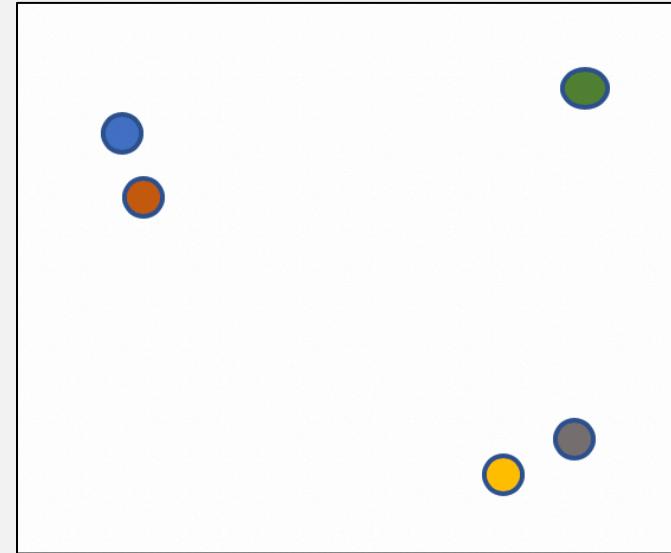
Each Color is a Track



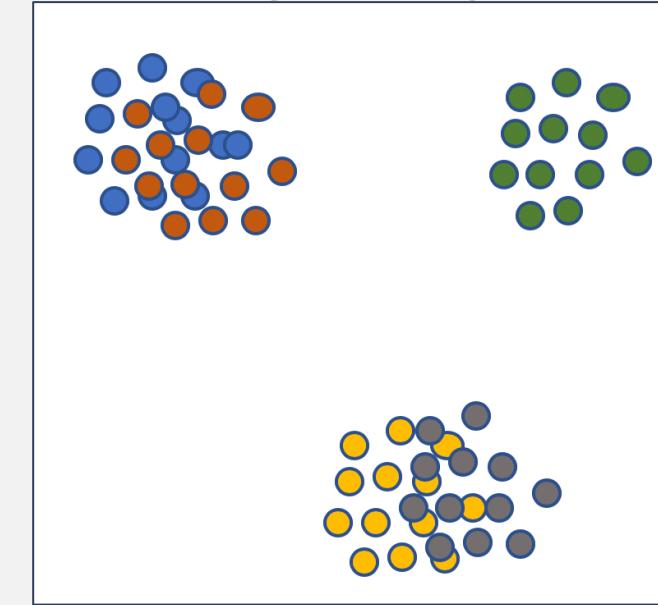
Alternative: Clustering the Averages

Take Average of Vectors in Each Track Before Clustering

Clustering Average



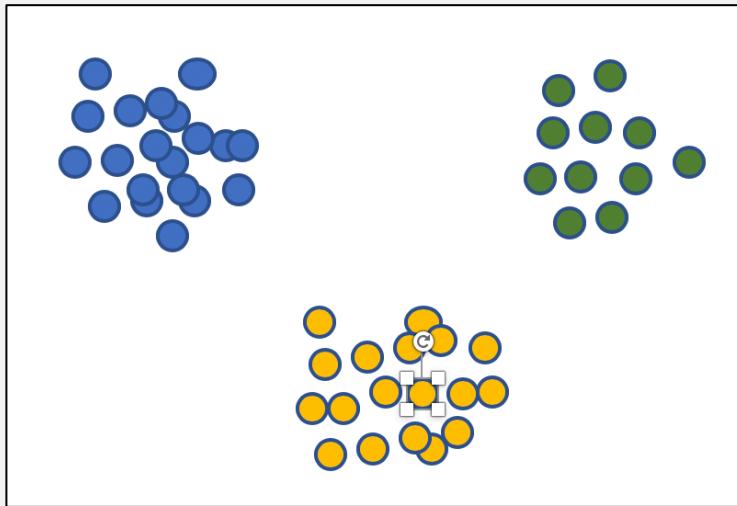
Clustering Every Vector



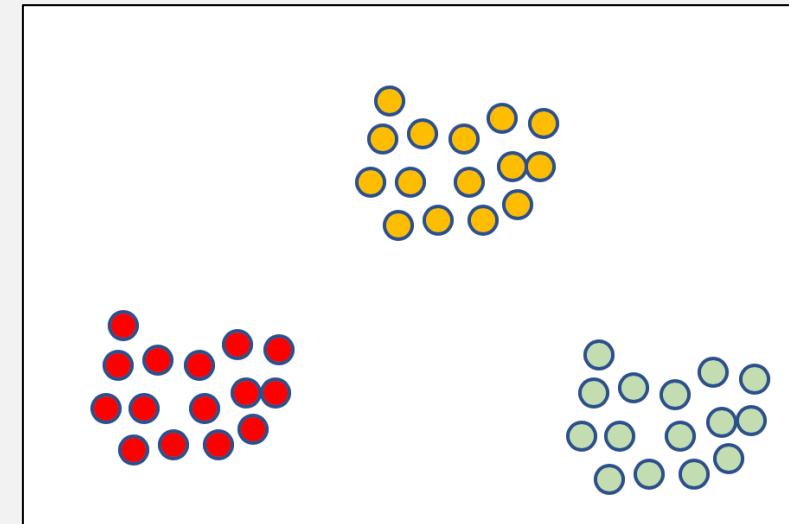
ISSUES

Same person appearing in different videos

Video 1



Video 2



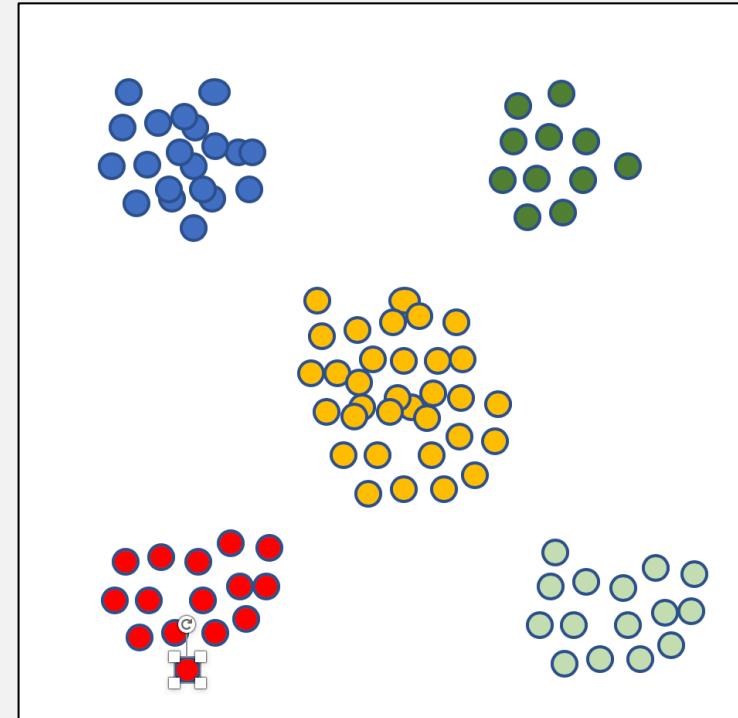
Training voice bio assumes unique voice per person.

We don't want to one voice being matched to two
“different” people

Solution

Across Video Clustering

Video1-Video2 Combined



MORE DEMO

THANKS!!
QUESTIONS?