tvID: Identifying Characters in a TV Show

Philip Meyers

Goal

Locate and identify faces of characters in a TV show episode by using the script, subtitles, and video

Data Sources

Penny: Oh, you're inviting me over to eat?

Leonard: Uh, yes.

Penny: Oh, that's so nice, I'd love to.

Leonard: Great.

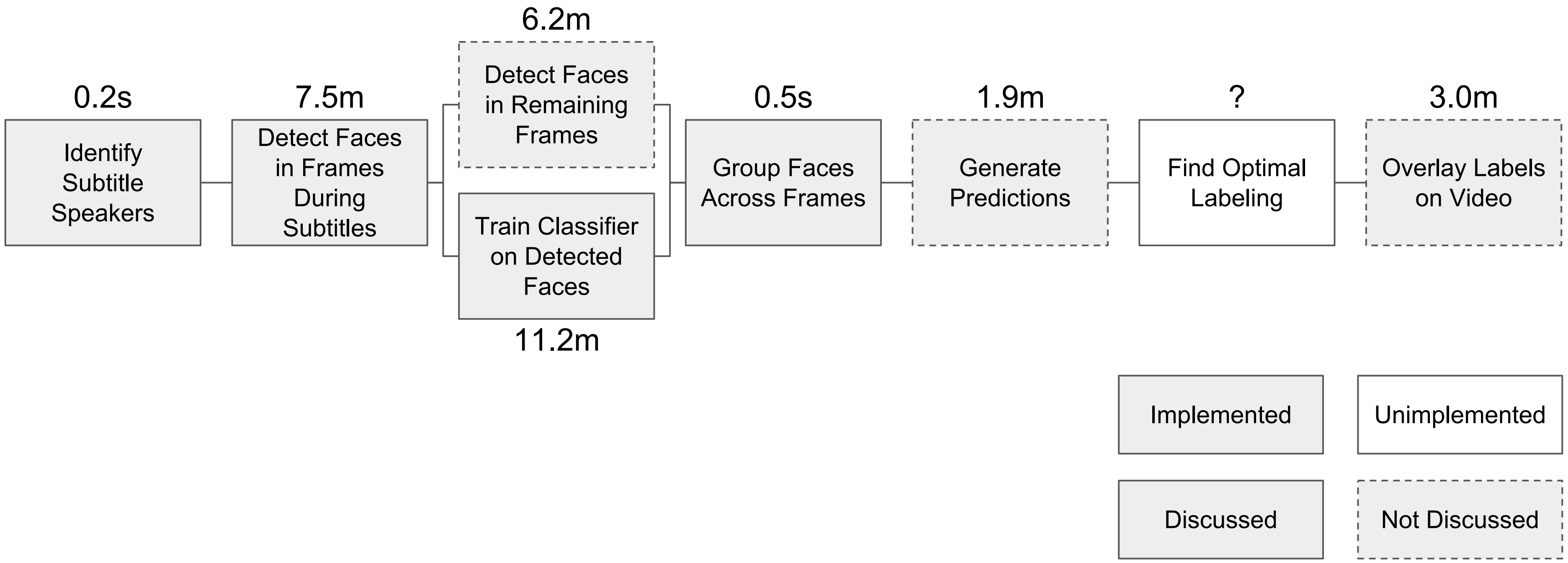
Script
(WHO, WHAT)

```
105
00:05:43,200 --> 00:05:44,900
You're inviting me over to eat?

106
00:05:47,300 --> 00:05:49,300
Oh, that's nice.
I'd love to.

107
00:05:49,500 --> 00:05:50,500
Great.
```

Subtitles
(WHAT, WHEN)

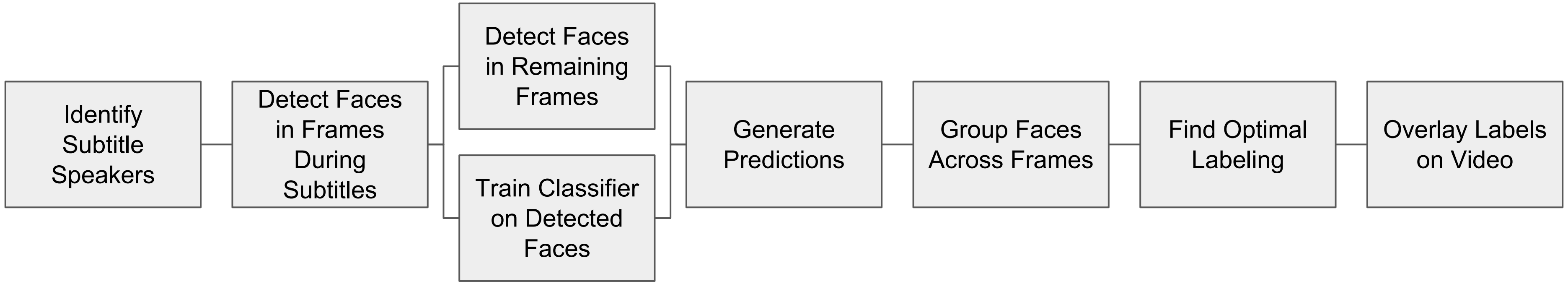
Video
(WHEN, WHO)

Pipeline


0.2s
Identify Subtitle Speakers
7.5m
Detect Faces in Frames During Subtitles
6.2m
Detect Faces in Remaining Frames
Train Classifier on Detected Faces
11.2m
0.5s
Group Faces Across Frames
1.9m
Generate Predictions
?
Find Optimal Labeling
3.0m
Overlay Labels on Video
Implemented
Unimplemented
Discussed
Not Discussed

Testbed: Intel i5 3.5GHz, Nvidia GTX 1080 Ti, 32GB RAM, 500GB SSD

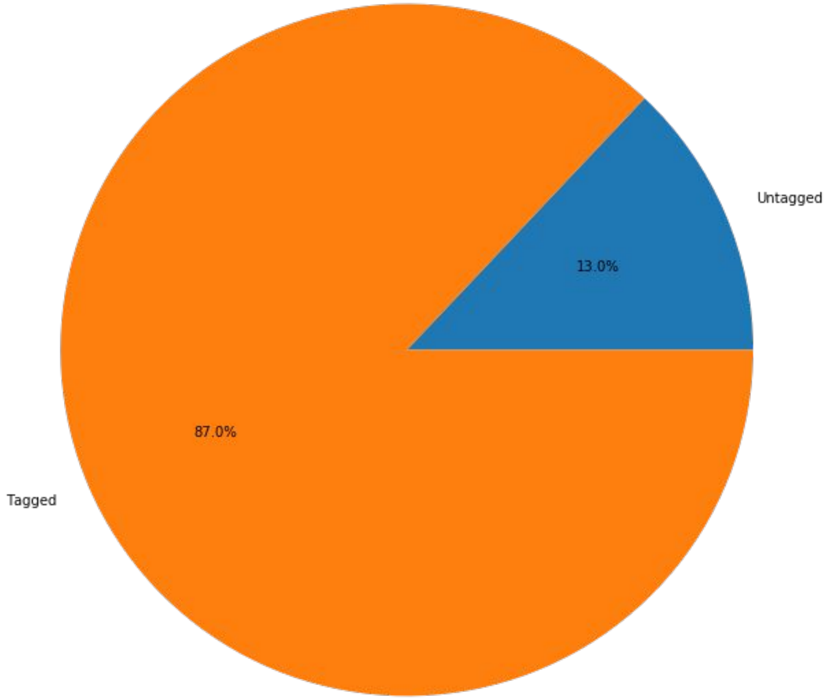
Pipeline


Identify Subtitle Speakers
Detect Faces in Frames During Subtitles
Detect Faces in Remaining Frames
Train Classifier on Detected Faces
Generate Predictions
Group Faces Across Frames
Find Optimal Labeling
Overlay Labels on Video

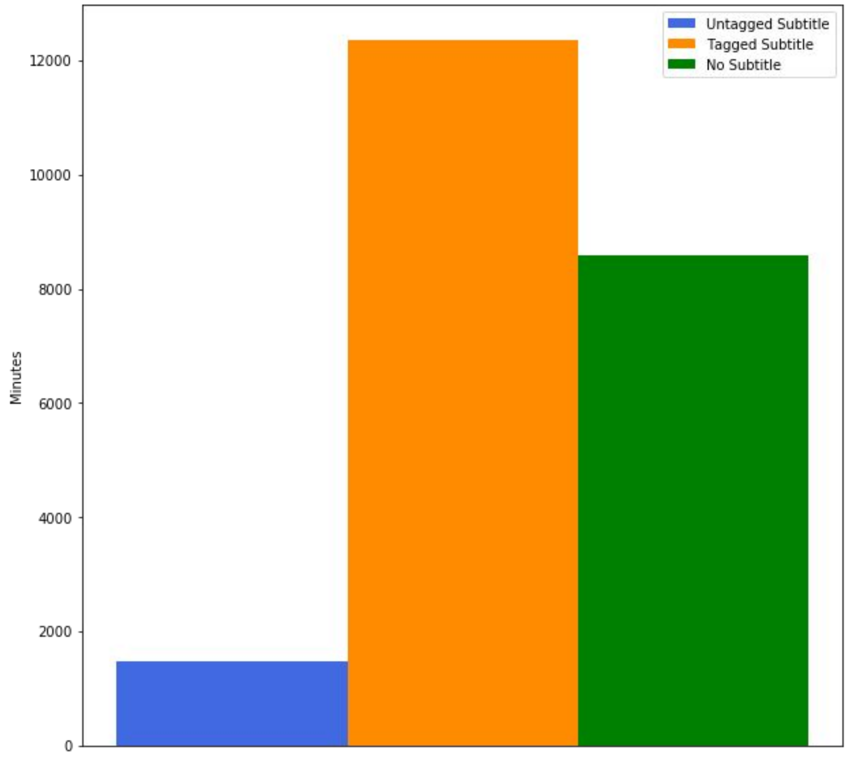
Testbed: Intel i5 3.5GHz, Nvidia GTX 1080 Ti, 32GB RAM, 500GB SSD

Identifying Subtitle Speakers

- Load dialogues from script and subtitle segments, discard non-character dialogues and multi-speaker subtitles
    - Scene: Sheldon and Leonard's apartment.
    - - Excuse me.
      - Hang on.
- Tokenize dialogue and subtitle text, remove stopwords
- Identify tokens that occur an equal number of times in dialogue and subtitles
- Find corresponding pairs of (dialogue, subtitle), vote for dialogue’s speaker as subtitle’s speaker
- For each subtitle, if all votes are for one speaker, choose that speaker



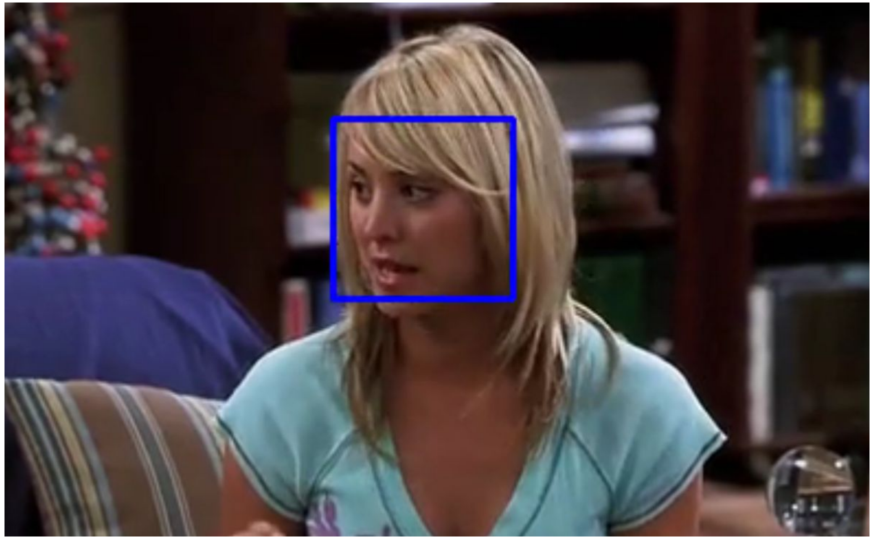
Proportion of tagged subtitles
(409 total segments)



Distribution of subtitle state over episode
(22 minutes long)

Detecting Faces

- Find locations of all faces in frames during subtitles with identified speakers
  - dlib CNN-based model
  - GPU acceleration faster than alternative CPU-based methods (HOG)
- Collect features of all detected faces
  - 128-dimension feature vectors



```
[-0.07777796  0.17063278  0.08688963
 -0.08402792 -0.09942919  0.14860559
 -0.18220419 -0.01301216 -0.04965496
 -0.0984873  -0.13330472  0.03754454
```

Training Classifier

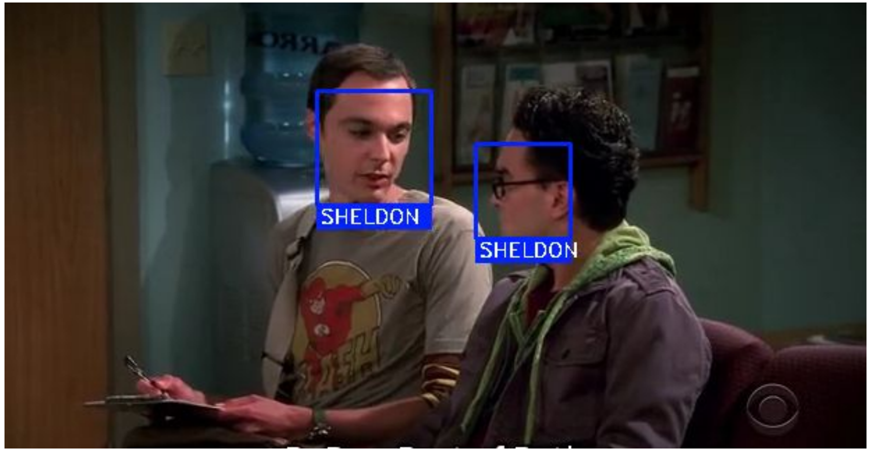
- Generate training data from face encodings using subtitle speaker as labels
- Weight samples to penalize frames with multiple faces
    - w(num_faces) = 1 / exp(1 - num_faces)
- Train SVM on face encodings and labels using sample weights

Grouping Faces Across Frames

- Face bounds identified individually in each frame → group nearby bounds across consecutive frames into a single face sequence
  - 60% area overlap, maximum lag of 0.5 seconds (12 frames for 24 fps video)
- Improves robustness against facial occlusion and noisy predictions

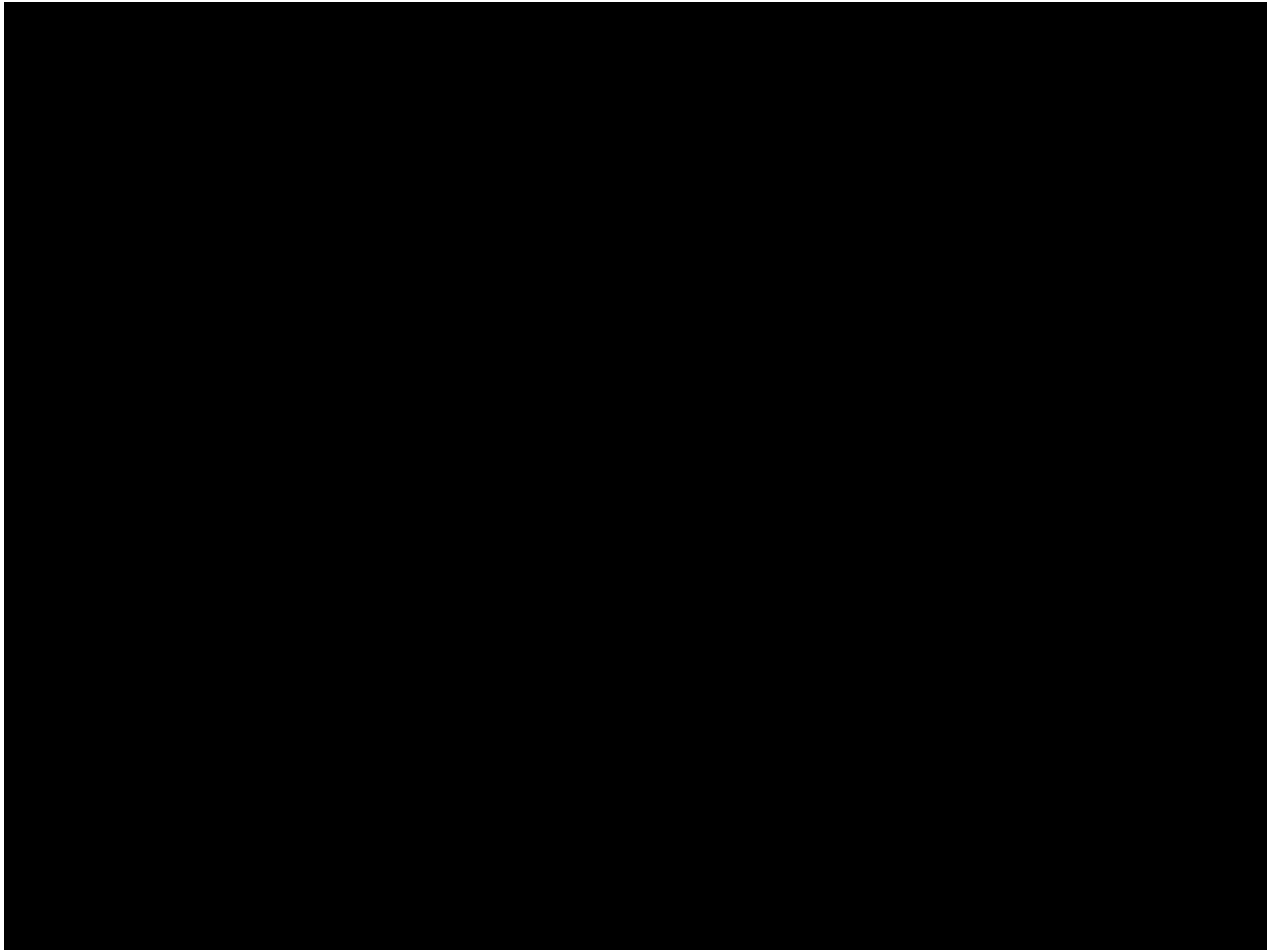
Finding Optimal Labels

- Each face sequence of length N frames has N predictions (probability distribution over C characters)
  - Reduce to single prediction (take max across all predictions and renormalize)
- Want to select label for each face sequence to maximize likelihood subject where no concurrent face sequences share the same label (i.e. the same face cannot be in two places at once)
- Constrained optimization problem
  - Linear programming



Results

- Challenge: finding the script, subtitles, and video for the same TV episode (while not getting a virus)
  - Scripts from fan-run websites/wiki
  - Subtitles from http://www.tvsubtitles.net/
  - Video from Google "Index of {TV SHOW} {SEASON NO}{EPISODE NO} 720p" searches
- Experiments
  - The Big Bang Theory S01E01
  - Seinfeld S01E05



Remaining Work

- Classifier
  - Explore alternative sample weighting methods and classifiers
    - Test impact of refining dataset to speed up training
  - Generate confusion matrix to quantitatively evaluate classifier performance
- Facial Grouping
  - Interpolate across missed frames
  - Use object tracking to group faces instead of boundary overlap
- Labeling
  - Find threshold for displaying label
  - Use linear programming to solve for optimal labels

Implementation Details

- 700 LOC
- Libraries Used
    - scikit-learn
    - scikit-video
    - OpenCV
    - face_recognition (https://github.com/ageitgey/face_recognition)
    - Numpy