Automatic Naming TV serial characters

Computer Vision Project - endeval

Members :

1. Anjali (2019201012)
2. Apurva (2019201069)
3. Amogh (2019201071)
4. Himani (2019201081)

Objective of the project

The objective of this work is to label television or movie footage with the identity of the people present in each frame of the video.

The new technique used in this project is that we make the use of subtitles + script to know which speaker is speaking at a particular moment. Along with which speaker detection technique is used to remove ambiguity from the detected faces.

Related Work:

- "Hello! My name is… Buffy" – Automatic Naming of Characters in TV Video, *Mark Everingham, Josef Sivic and Andrew Zisserman Department of Engineering Science, University of Oxford*

Methodology

- Creating the CSV File for timestamp and respective names of actor speaking
- Creating training data
- Training
- Prediction

Creating the CSV File

For each entry in srt file :

- Create entry in csv file.
- Get the start and end timestamp.
- Map the dialogues using Dynamic Time Warping with the script file and extract the name of the person speaking.
- Insert this into the new entry created

Training stage

- Frame creation
- Feature Extraction
- Dataset creation
- Training

Training stage contd...

1. Divide the video into frames so that we have a set of images now

2. The dataset is created using the images that have single person's face detected in the image

    where (1 bounding box detected) :

        Face_recog.vector + csv-name

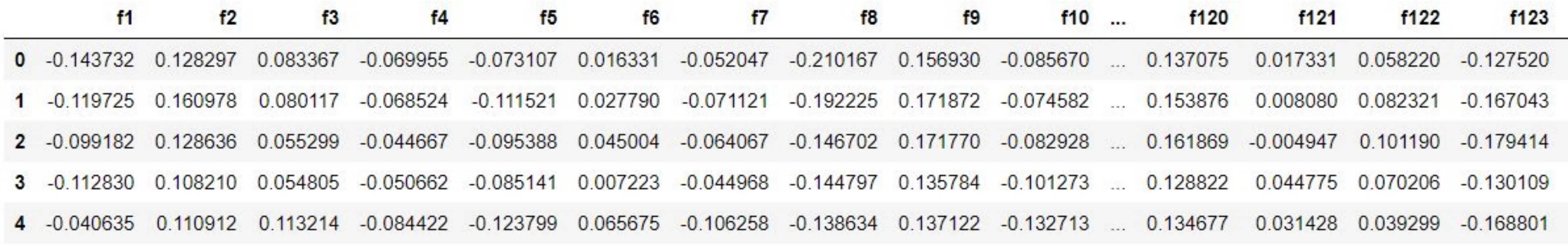
3. Train this data using the model (SVM classifier)

4. Save this model for later use for predicting

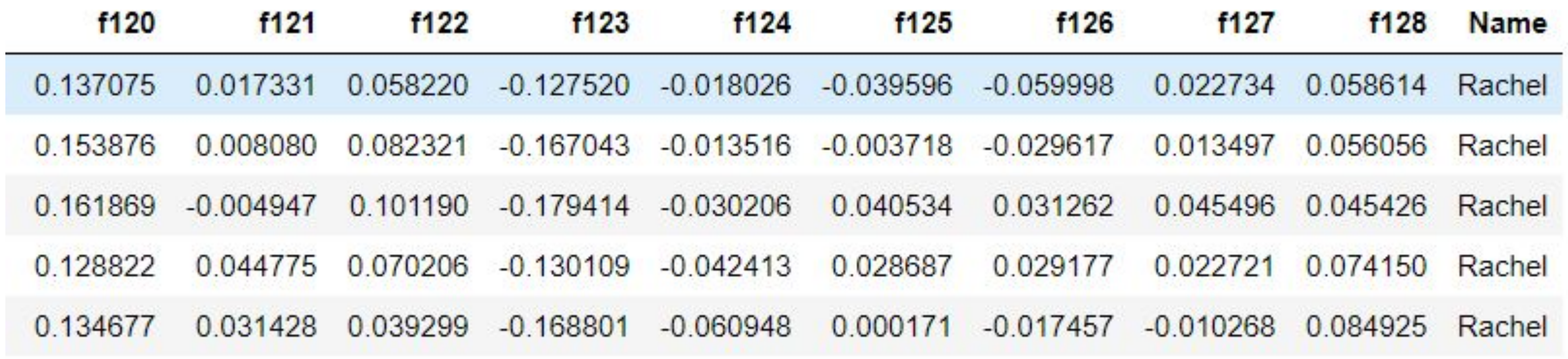
Video Processing (Extracting Descriptors)

- Aim is to extract descriptors out of the face tracks for matching.
- face_encodings()

Training data:



	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	...	f120	f121	f122	f123
0	-0.143732	0.128297	0.083367	-0.069955	-0.073107	0.016331	-0.052047	-0.210167	0.156930	-0.085670	...	0.137075	0.017331	0.058220	-0.127520
1	-0.119725	0.160978	0.080117	-0.068524	-0.111521	0.027790	-0.071121	-0.192225	0.171872	-0.074582	...	0.153876	0.008080	0.082321	-0.167043
2	-0.099182	0.128636	0.055299	-0.044667	-0.095388	0.045004	-0.064067	-0.146702	0.171770	-0.082928	...	0.161869	-0.004947	0.101190	-0.179414
3	-0.112830	0.108210	0.054805	-0.050662	-0.085141	0.007223	-0.044968	-0.144797	0.135784	-0.101273	...	0.128822	0.044775	0.070206	-0.130109
4	-0.040635	0.110912	0.113214	-0.084422	-0.123799	0.065675	-0.106258	-0.138634	0.137122	-0.132713	...	0.134677	0.031428	0.039299	-0.168801



f120	f121	f122	f123	f124	f125	f126	f127	f128	Name
0.137075	0.017331	0.058220	-0.127520	-0.018026	-0.039596	-0.059998	0.022734	0.058614	Rachel
0.153876	0.008080	0.082321	-0.167043	-0.013516	-0.003718	-0.029617	0.013497	0.056056	Rachel
0.161869	-0.004947	0.101190	-0.179414	-0.030206	0.040534	0.031262	0.045496	0.045426	Rachel
0.128822	0.044775	0.070206	-0.130109	-0.042413	0.028687	0.029177	0.022721	0.074150	Rachel
0.134677	0.031428	0.039299	-0.168801	-0.060948	0.000171	-0.017457	-0.010268	0.084925	Rachel

Training Results and model Used:

Model	**Accuracy**
SVM	0.79
SVM(C=20)	0.83
Random Forest	0.91
LogisticRegression	0.78
LogisticRegression(C=20)	0.80
LogisticRegression(C=0.01)	0.66

Final Predictions

- Frame by frame run

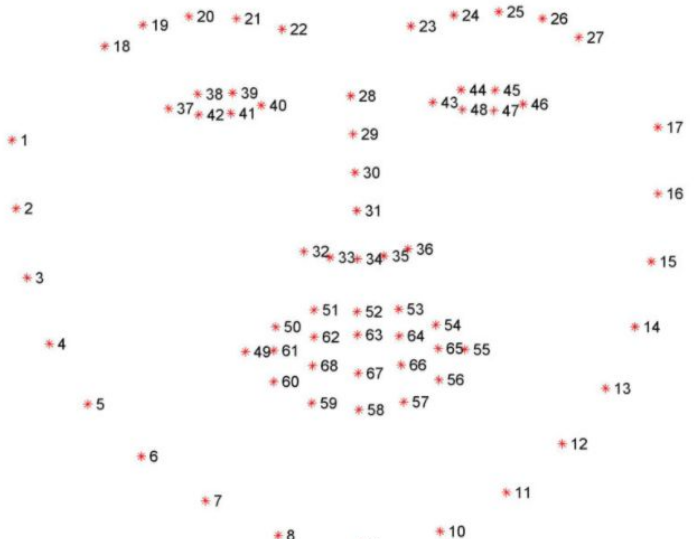
```
For each frame :
    For each bounding box:
        name = model.predict
        display name on the box
```

Additional approach for feature vector

- Facial Landmarks detection -> 68 facial coordinates
- The high level description of facial encodings can be:
  - the color, size and slant of eyes
  - Bridge of nose
  - Nose to chin
  - Jaw angle
  - Gap between eyebrows

Assuming that these remain same for a person

irrespective of the facial expressions.

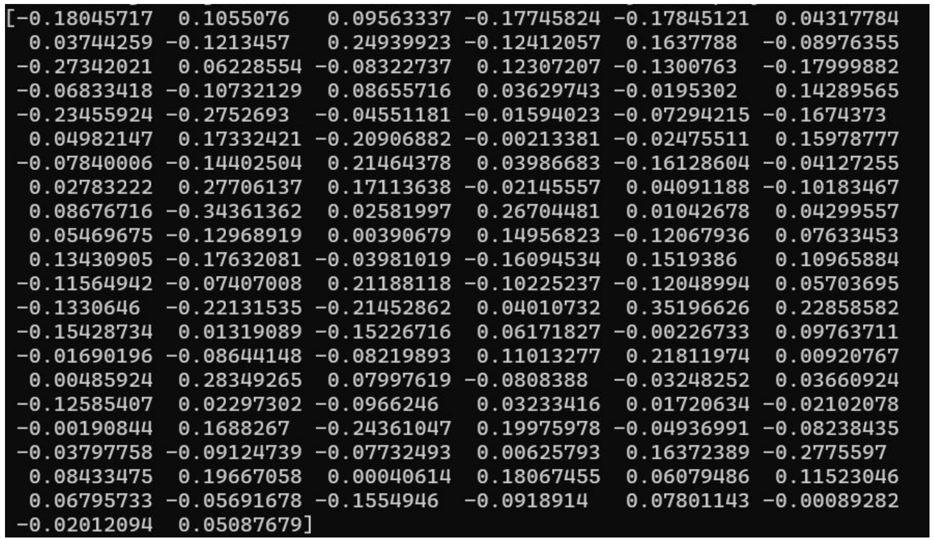


Libraries and functions used

dlib.get_frontal_face_detector() - to fetch the bounding box for face

face_recognition.face_encodings() - convert the facial image to vector of features

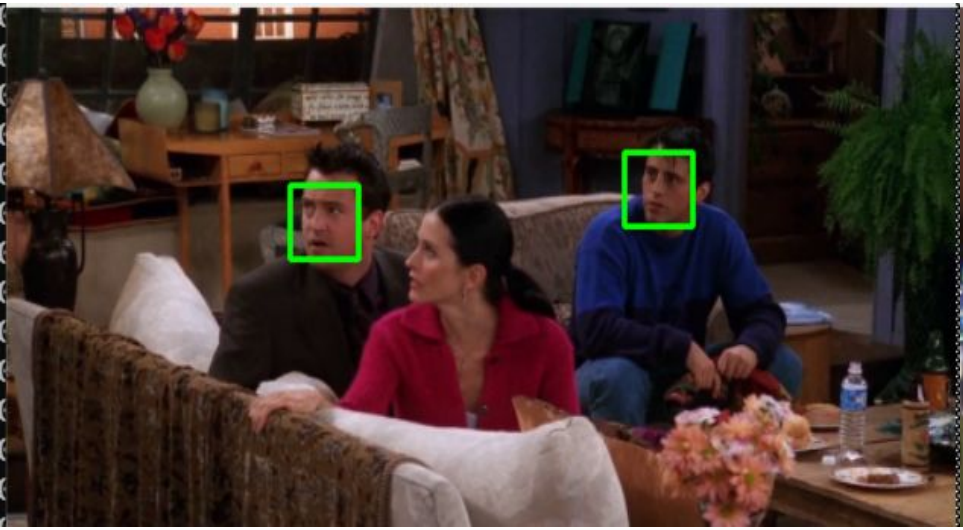
Image Feature Vector using facial_encodings

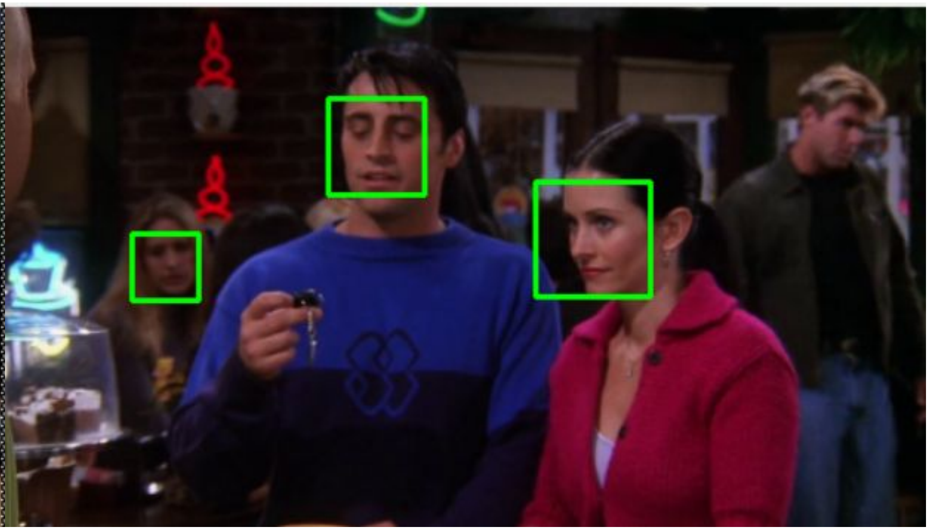


```
[-0.18045717  0.1055076   0.09563337 -0.17745824 -0.17845121  0.04317784
  0.03744259 -0.1213457   0.24939923 -0.12412057  0.1637788  -0.08976355
 -0.27342021  0.06228554 -0.08322737  0.12307207 -0.1300763  -0.17999882
 -0.06833418 -0.10732129  0.08655716  0.03629743 -0.0195302   0.14289565
 -0.23455924 -0.2752693  -0.04551181 -0.01594023 -0.07294215 -0.1674373
  0.04982147  0.17332421 -0.20906882 -0.00213381 -0.02475511  0.15978777
 -0.07840006 -0.14402504  0.21464378  0.03986683 -0.16128604 -0.04127255
  0.02783222  0.27706137  0.17113638 -0.02145557  0.04091188 -0.10183467
  0.08676716 -0.34361362  0.02581997  0.26704481  0.01042678  0.04299557
  0.05469675 -0.12968919  0.00390679  0.14956823 -0.12067936  0.07633453
  0.13430905 -0.17632081 -0.03981019 -0.16094534  0.1519386   0.10965884
 -0.11564942 -0.07407008  0.21188118 -0.10225237 -0.12048994  0.05703695
 -0.1330646  -0.22131535 -0.21452862  0.04010732  0.35196626  0.22858582
 -0.15428734  0.01319089 -0.15226716  0.06171827 -0.00226733  0.09763711
 -0.01690196 -0.08644148 -0.08219893  0.11013277  0.21811974  0.00920767
  0.00485924  0.28349265  0.07997619 -0.0808388  -0.03248252  0.03660924
 -0.12585407  0.02297302 -0.0966246   0.03233416  0.01720634 -0.02102078
 -0.00190844  0.1688267  -0.24361047  0.19975978 -0.04936991 -0.08238435
 -0.03797758 -0.09124739 -0.07732493  0.00625793  0.16372389 -0.2775597
  0.08433475  0.19667058  0.00040614  0.18067455  0.06079486  0.11523046
  0.06795733 -0.05691678 -0.1554946  -0.0918914   0.07801143 -0.00089282
 -0.02012094  0.05087679]
```

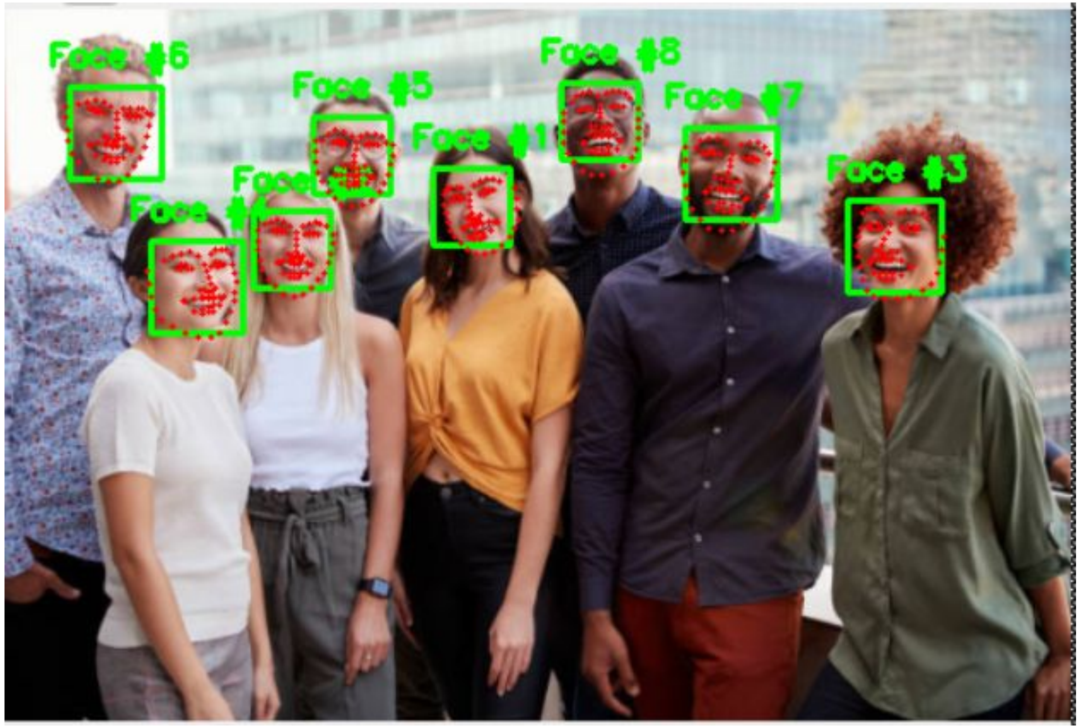
Size of each feature vector: 128

Face Detector Output

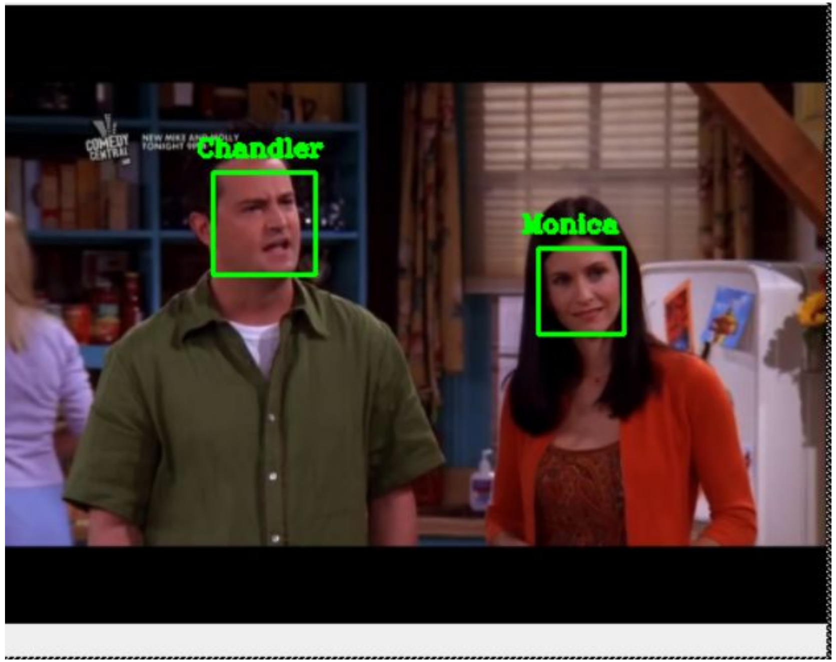


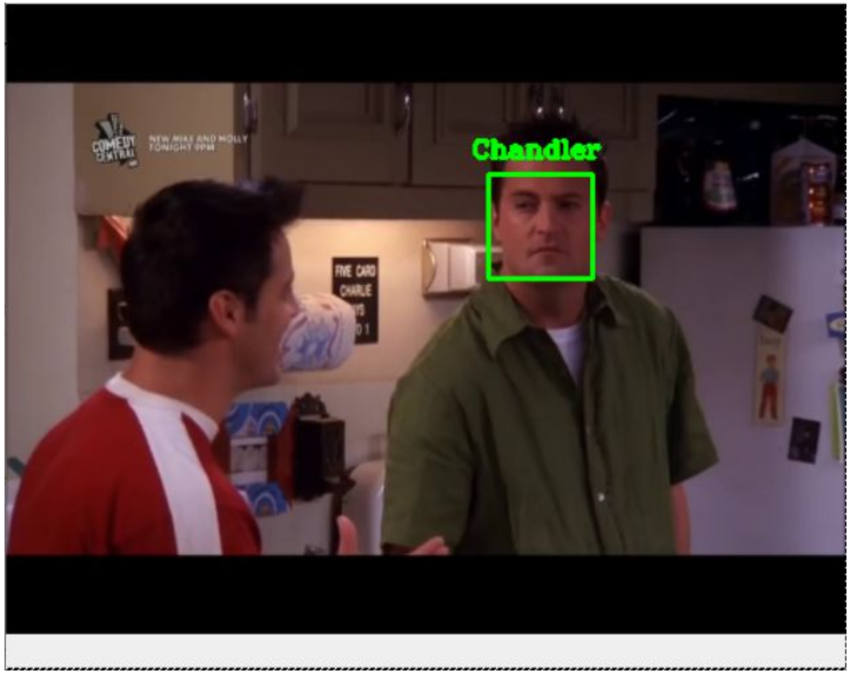


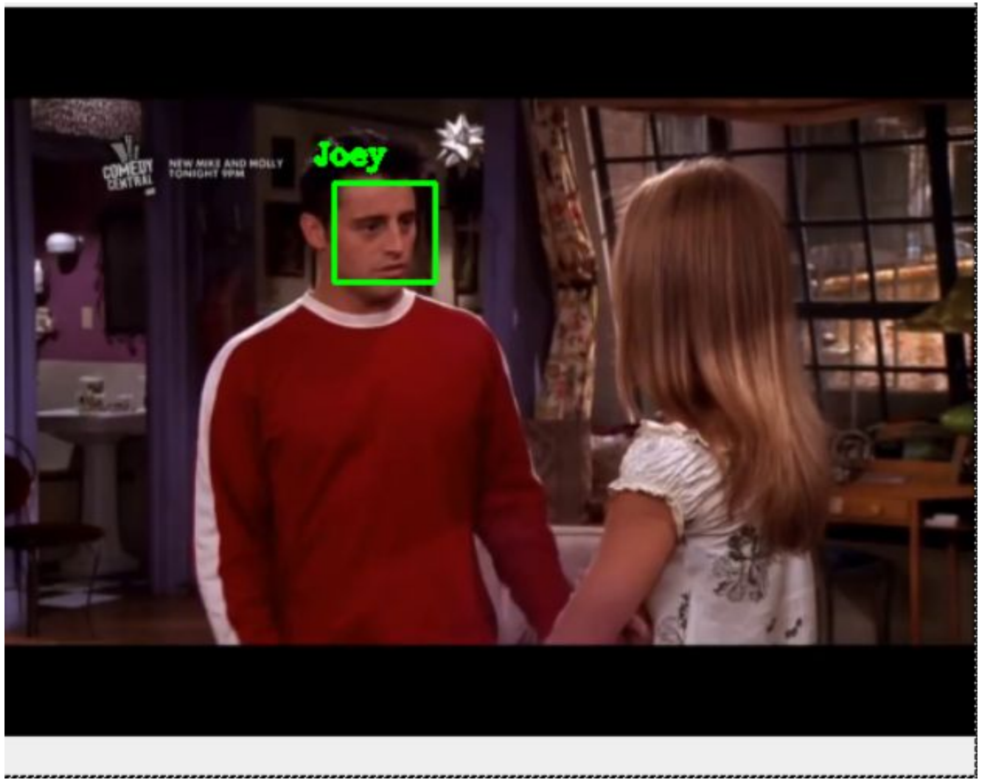
Facial Landmarks Detections



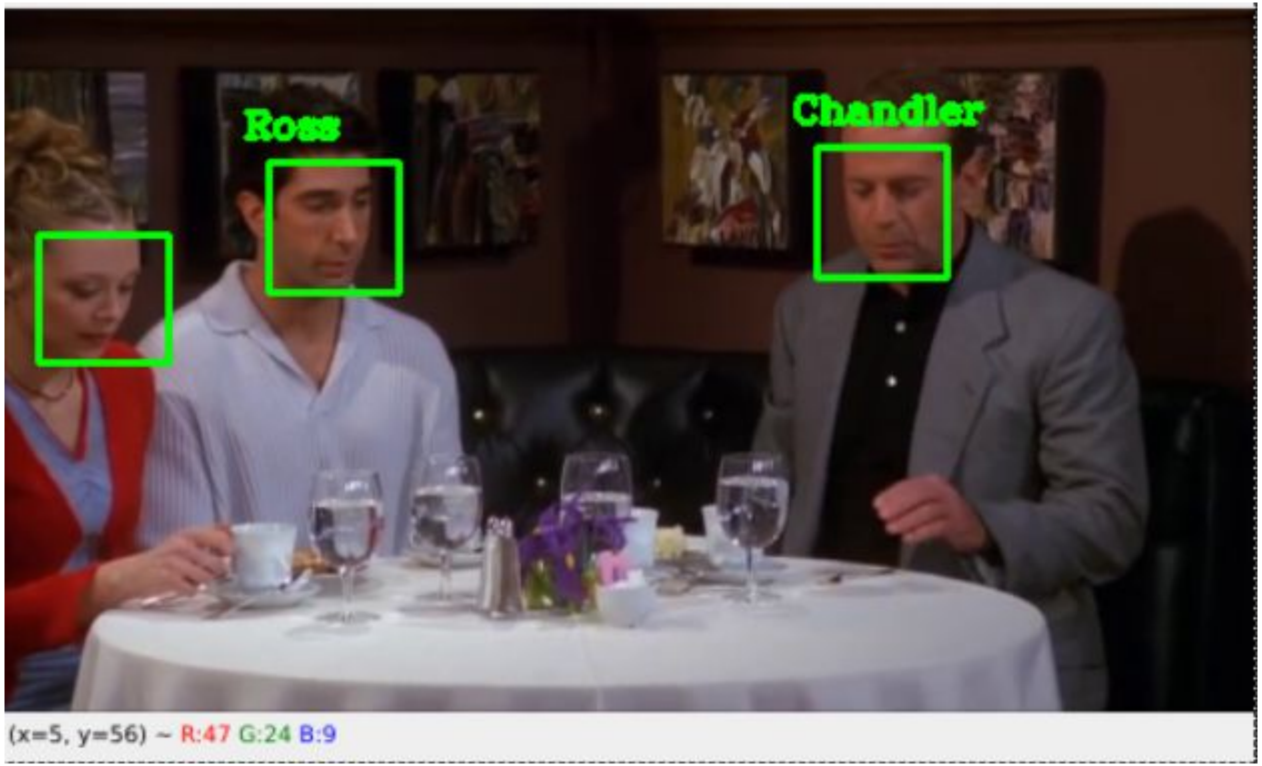
Classification Results:






Joey
COMEDY CENTRAL
NEW MIKE AND MOLLY TONIGHT 9PM

Misclassification Results:



(x=5, y=56) ~ R:47 G:24 B:9

Future Work:

- The detection method and appearance models used here could be improved, for example by bootstrapping person-specific detectors.
- Later this can be extended to Non-Linear models since they don’t have fixed size feature vector.
- Mechanism for error correction can be introduced, where we can keep a threshold.

Thank you!