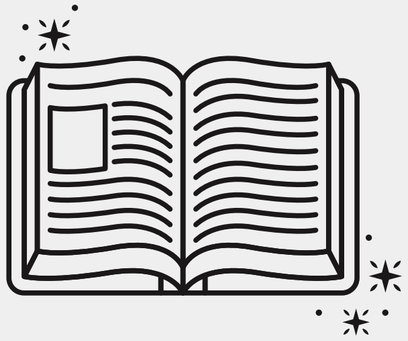
Team 15

Story Graph

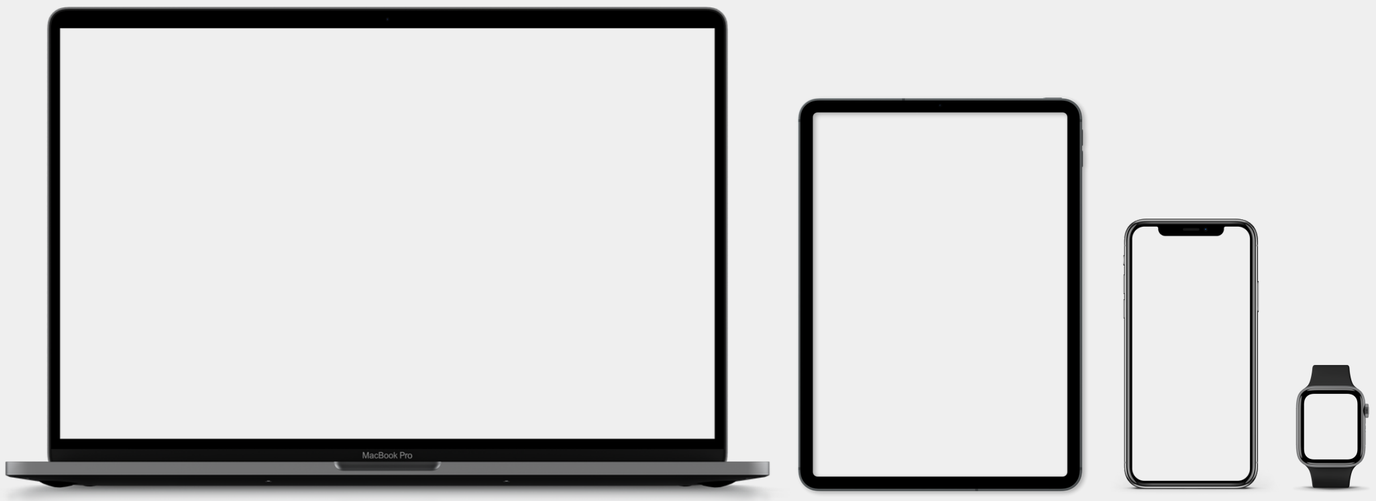
Visualizing Character Interactions as a Timeline



Sreenya Chitluri
2020102065

Sankeerthana Venugopal
2020102008

Problem Statement

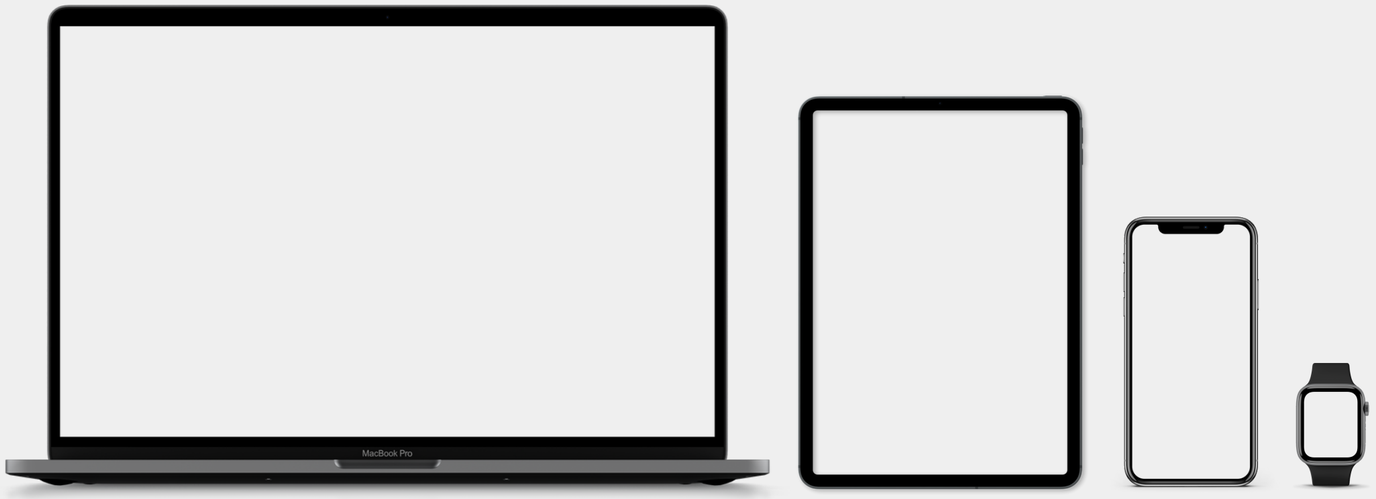


- Grouping similar shots together to create what is called a shot thread. These are shots that are not continuous but have coherency in terms of angle of camera, characters on screen etc.

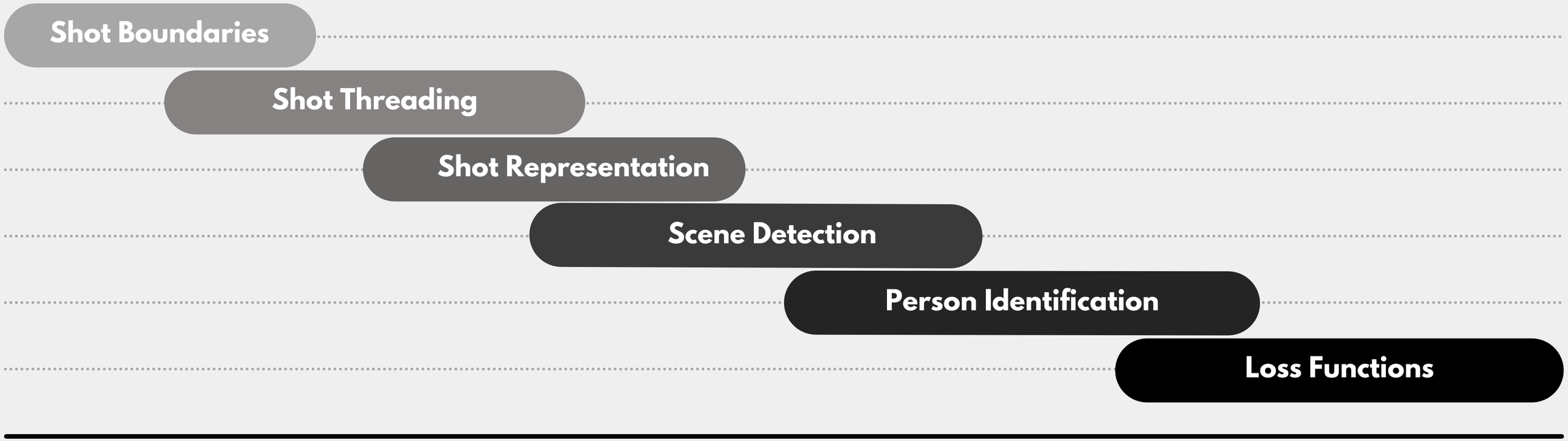
- Scene detection is used to isolate the different scenes of an episode.

- Constructing a graphical representation of the narrative structure that captures the interactions between the characters and progression of events over time.

Implementation Details

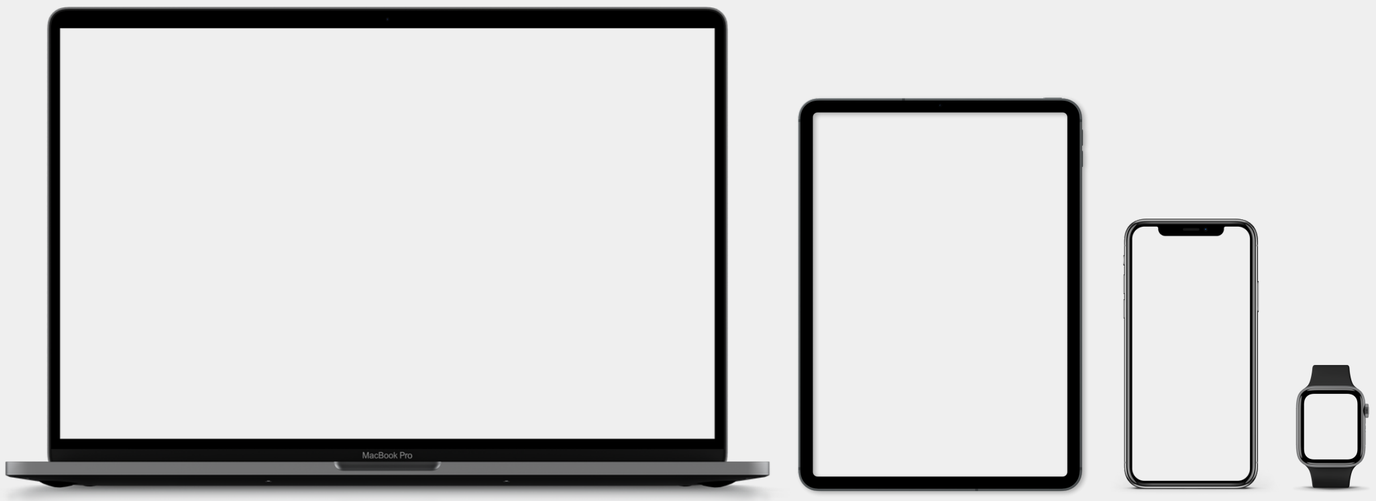


Roadmap


Shot Boundaries
Shot Threading
Shot Representation
Scene Detection
Person Identification
Loss Functions

Part 1

Shot Boundaries



Examples of different shots

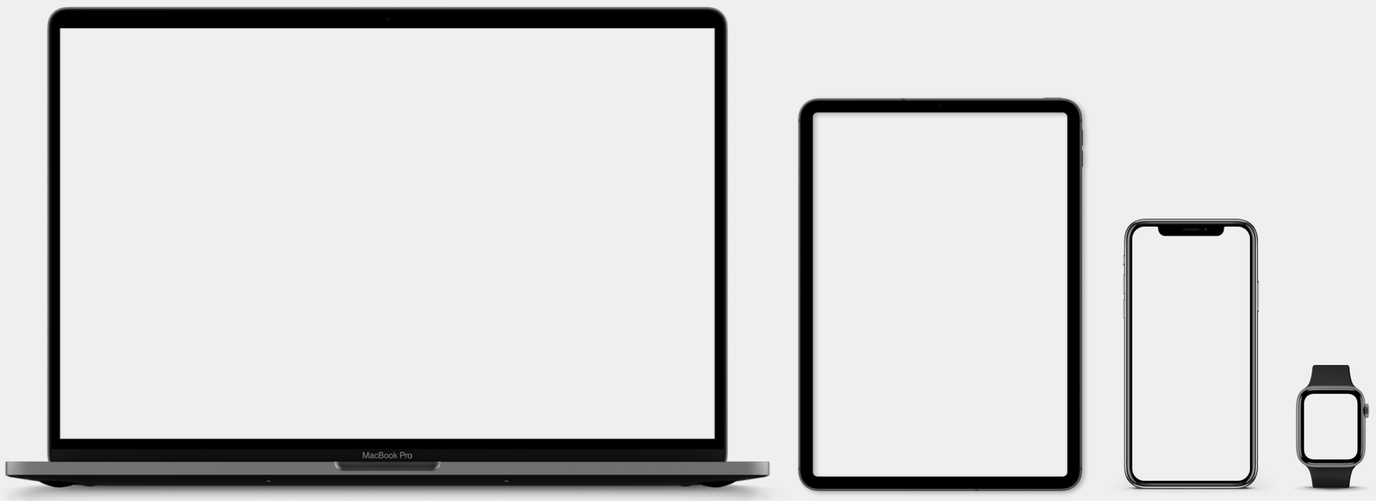
01 Uses Displaced Frame Differencing technique between consecutive frames to identify shots.

02 Used later to detect the scene boundaries. Assumed that the scene boundaries can only occur at shot boundaries.

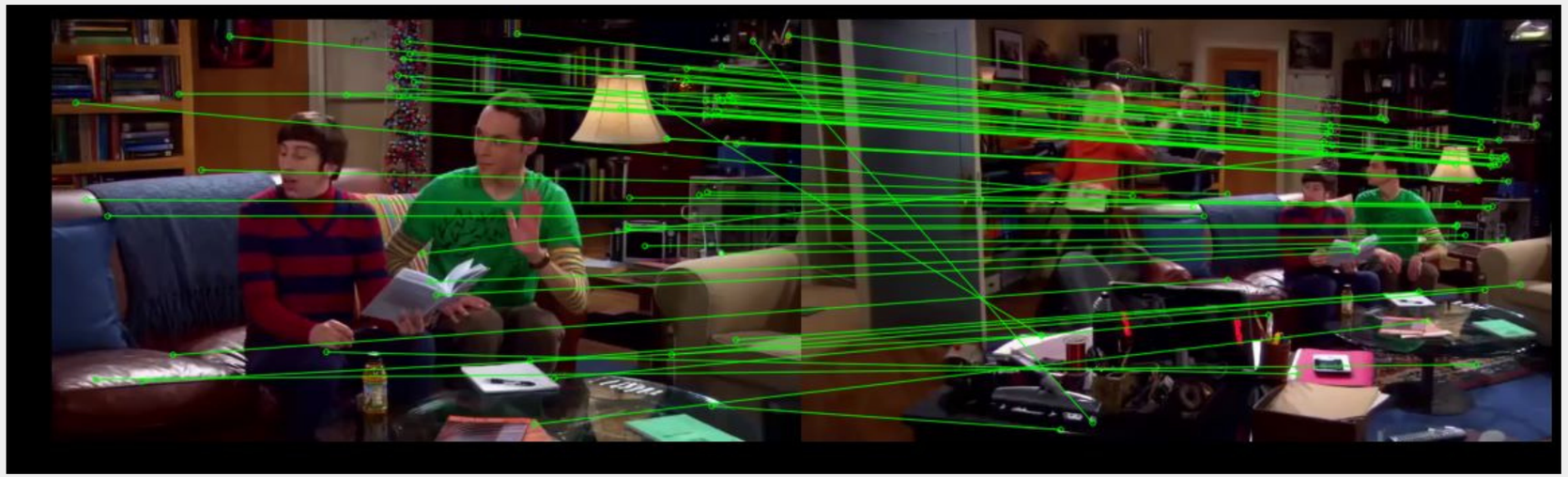
03 All the shots are stored in a dictionary with the initial frame and the frame number with the shot number as the key.

Part 2

Shot Threading

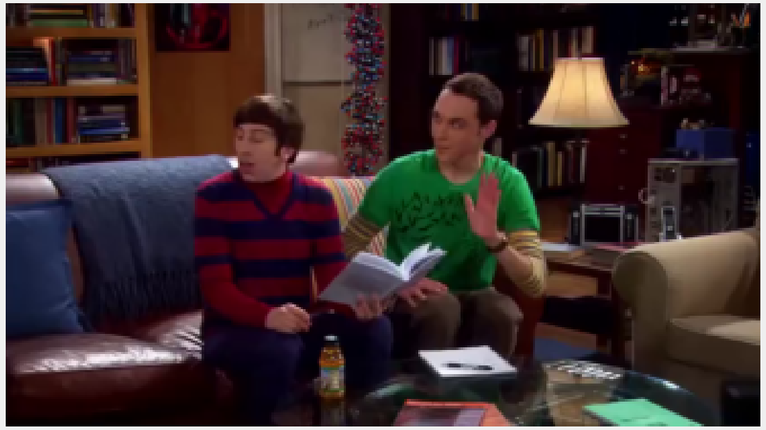


Shots that involve the same set of characters at the same location, typically belong to the same scene. Shot threading links shots that appear similar to each other using SIFT matching.
We compute the similarity between the last frame of every shot and the first frame of 25 subsequent shots and set a threshold on the number of SIFT matches to decide whether two shots are part of a thread



SIFT matches on the last frame of shot 8 and the first frame of shot 17

Shot 8



Shot 17

Shot 20

Shot 25

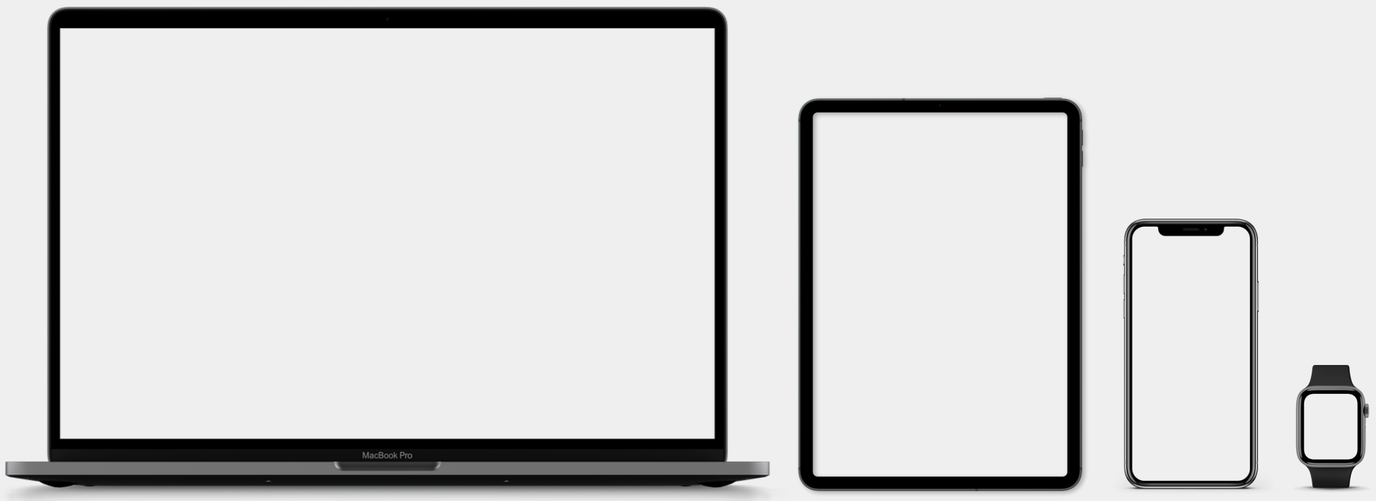
Shot 27

Examples of shots from a single thread

It is observed that they are all the same viewpoints/have overlapping characters and features, despite being not consecutive.

Part 3

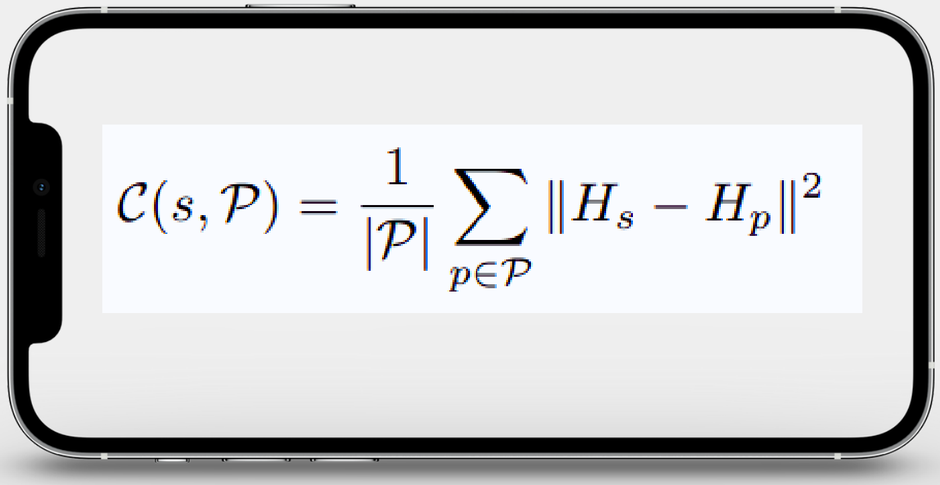
Shot Representation



Shot Representation

Distance of the shot from a set of other shots is given by the following formula.

01 Each shot is represented by the mean RGB histogram of the starting frame of the shot.



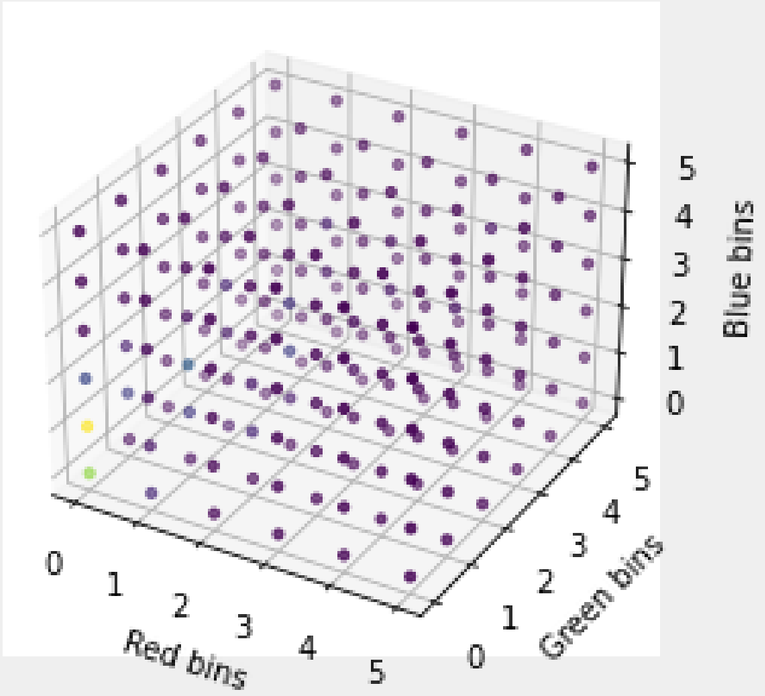
02 We store it in 6x6x6 bins where each dimension is divided into units of 42 each.

03 This can be used to calculate distance from other shots, each represented by a histogram.

04 The Bhattacharya histogram distance has been used in the project.

Results

Input shot



Histogram representation

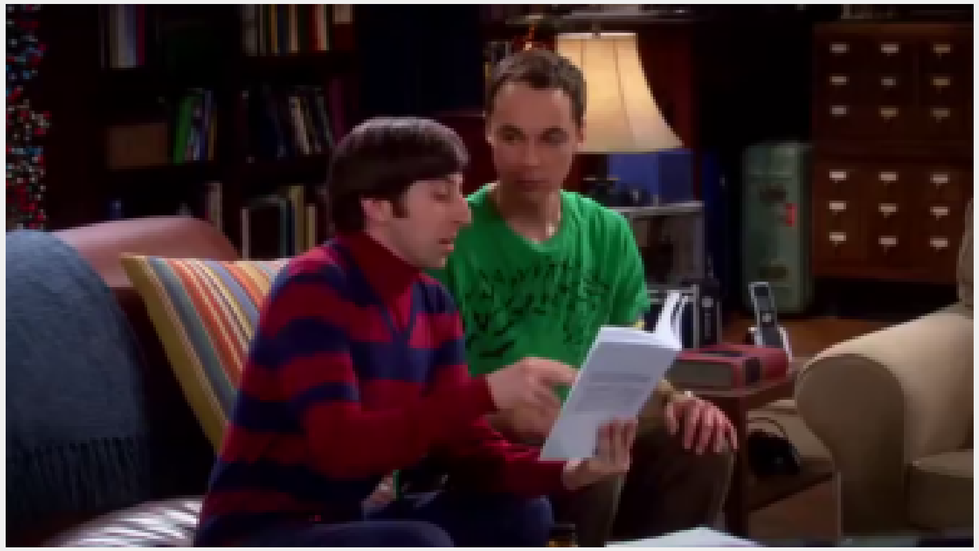
Results

Shot 4

Thread- [0, 2, 4, 9, 11, 13, 21]

Distance between the thread and the shot:
0.1173010422744623 8

Shot 0



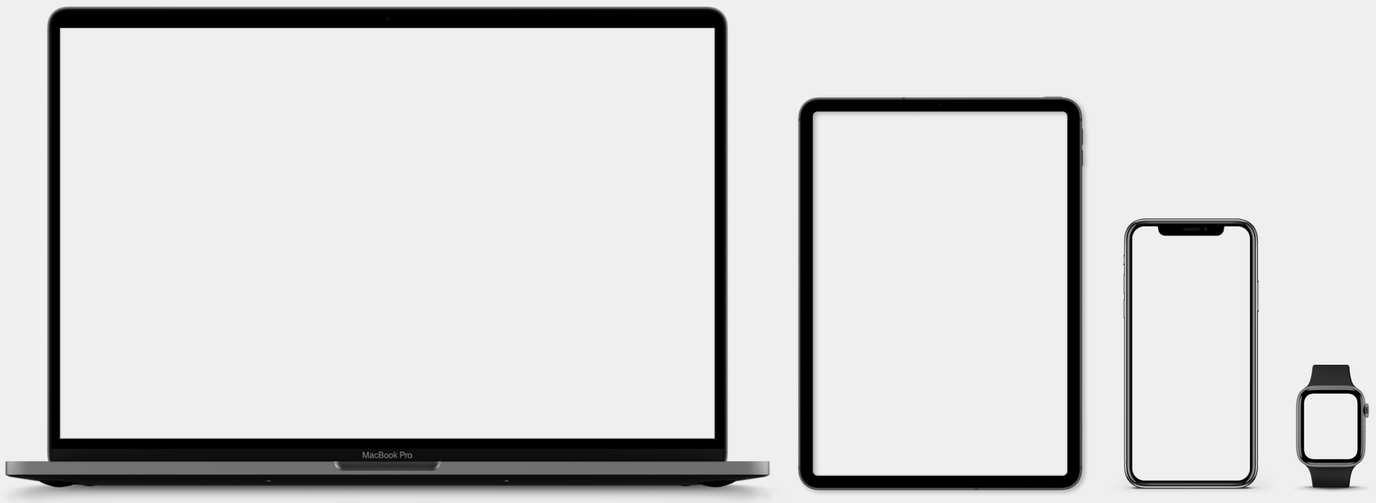
Thread- [43, 44, 45, 46]

Distance between the thread and the shot:
0.7995434756765227

It is observed that if the shot is threaded to the set the distance is lesser than if it isn't.

Part 4

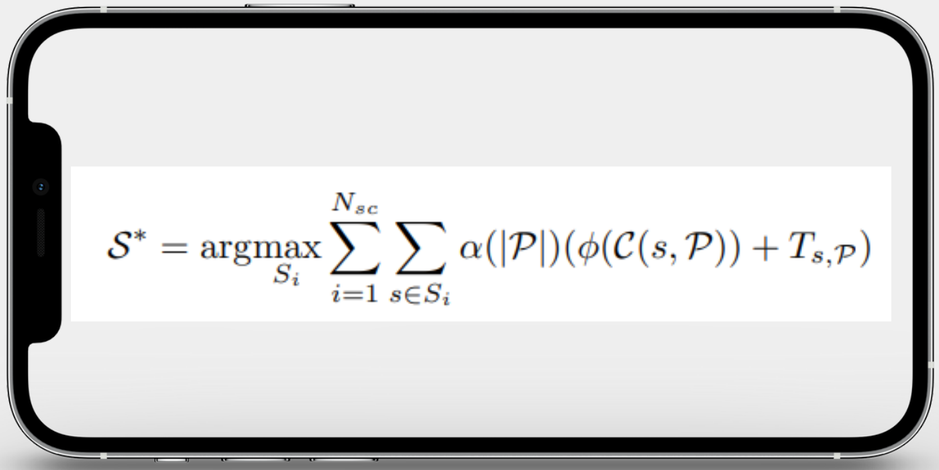
Scene Detection



A scene typically refers to a group of shots that remain coherent in location, people and general appearance

01 Let NSc be the number of scenes. The problem can be formulated as optimally grouping all the shots into sets of scenes S*i*. Essentially we perform an exhaustive search through all combinations of sets of shots till we get the maximum score.

02 $\phi(\mathcal{C}(s,\mathcal{P}))$ is the color based shot similarity score between the shot s and the shots that precede s in the scene

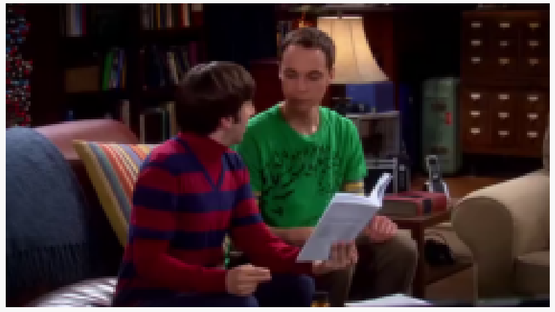
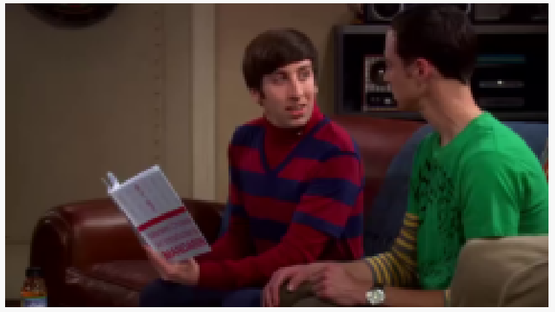
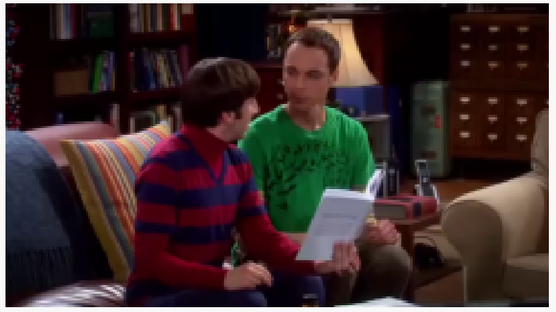
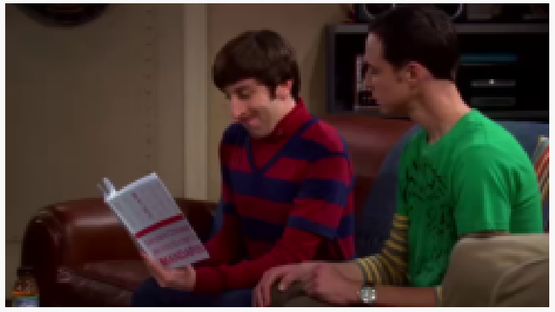


03 $T_{s,\mathcal{P}}$ captures whether there is a thread between s and any shot from *P*

04 $\alpha(n) = 1 - 0.5\,(n/N_l)^2$ is a decay factor that prevents the scene from growing too large





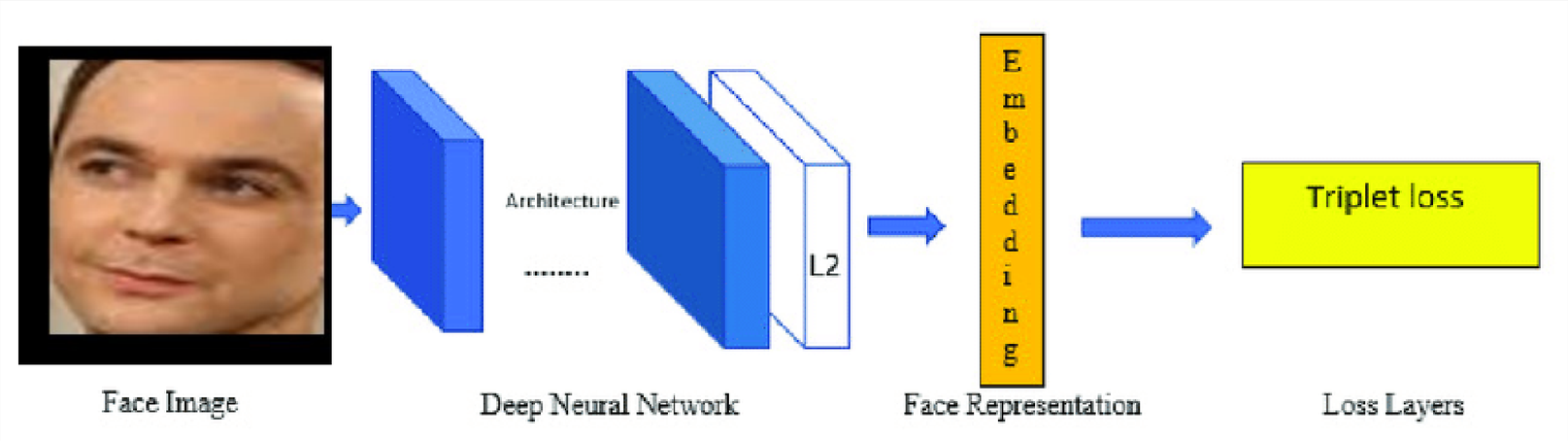


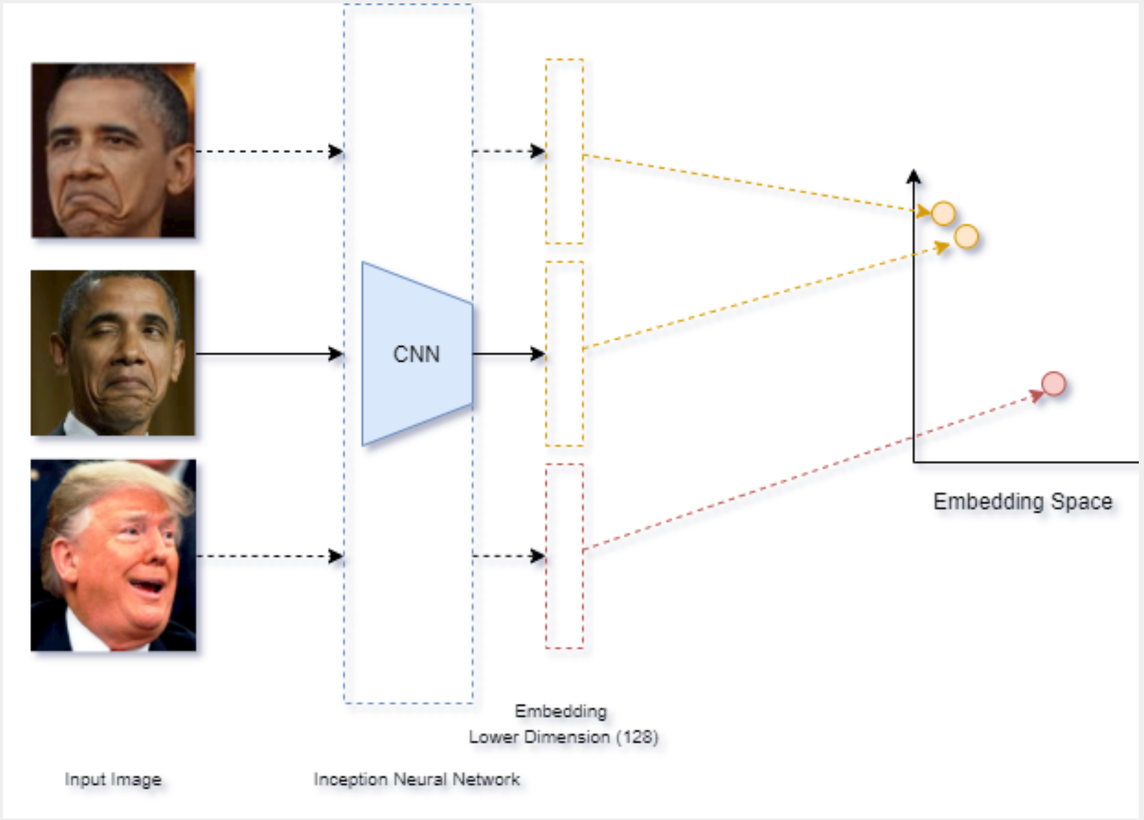
Shots from different scenes

Part 5

Person Identification

Face Recognition with Face**NET**





01 Trained with 6 different images of a character

02 Extract faces from the frames and generate an embedding vector using the model

03 Feed the test image through Facenet, get the embedding and find the closest face distance to any of the known faces

04 Used for the calculation of character-pair co-occurrence score

Results

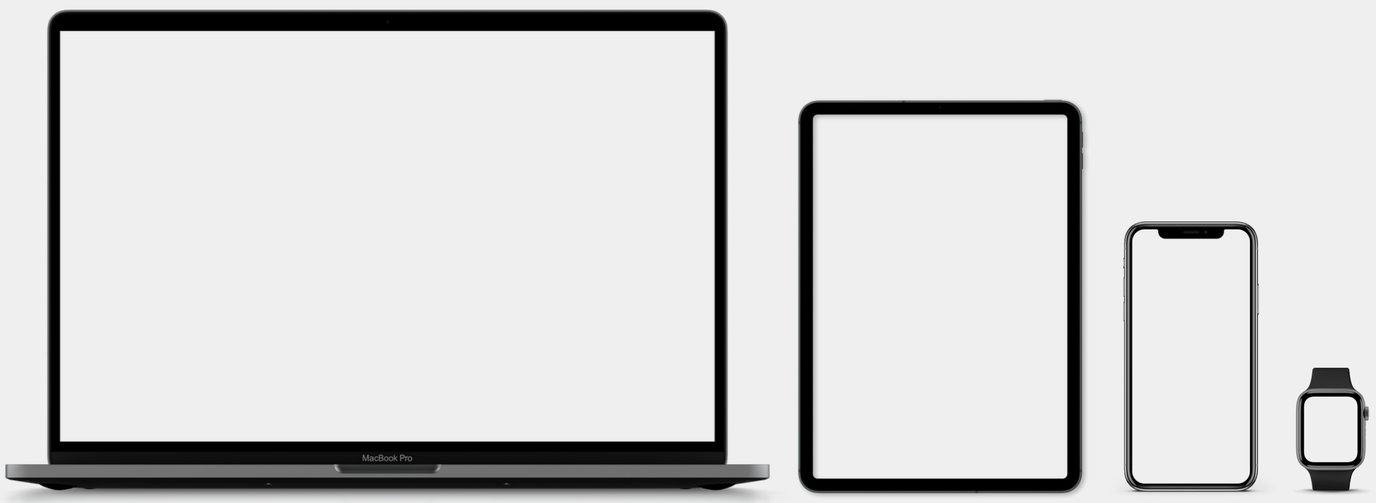
Predicted: ['sheldon']

Predicted: ['penny']

Predicted: ['leonard']

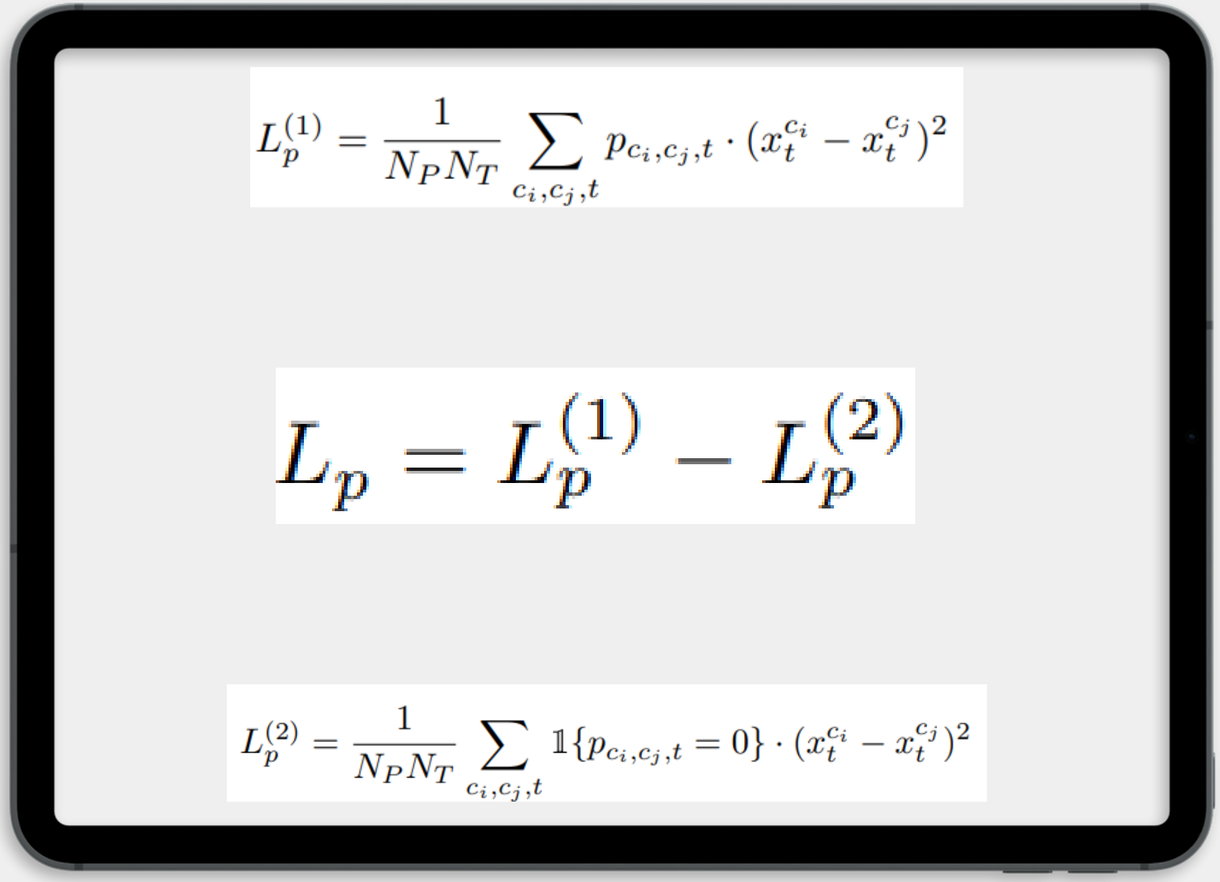
Part 6

Loss Functions



Proximity Loss

Controls the distance between the x coordinates of all characters in each scene.



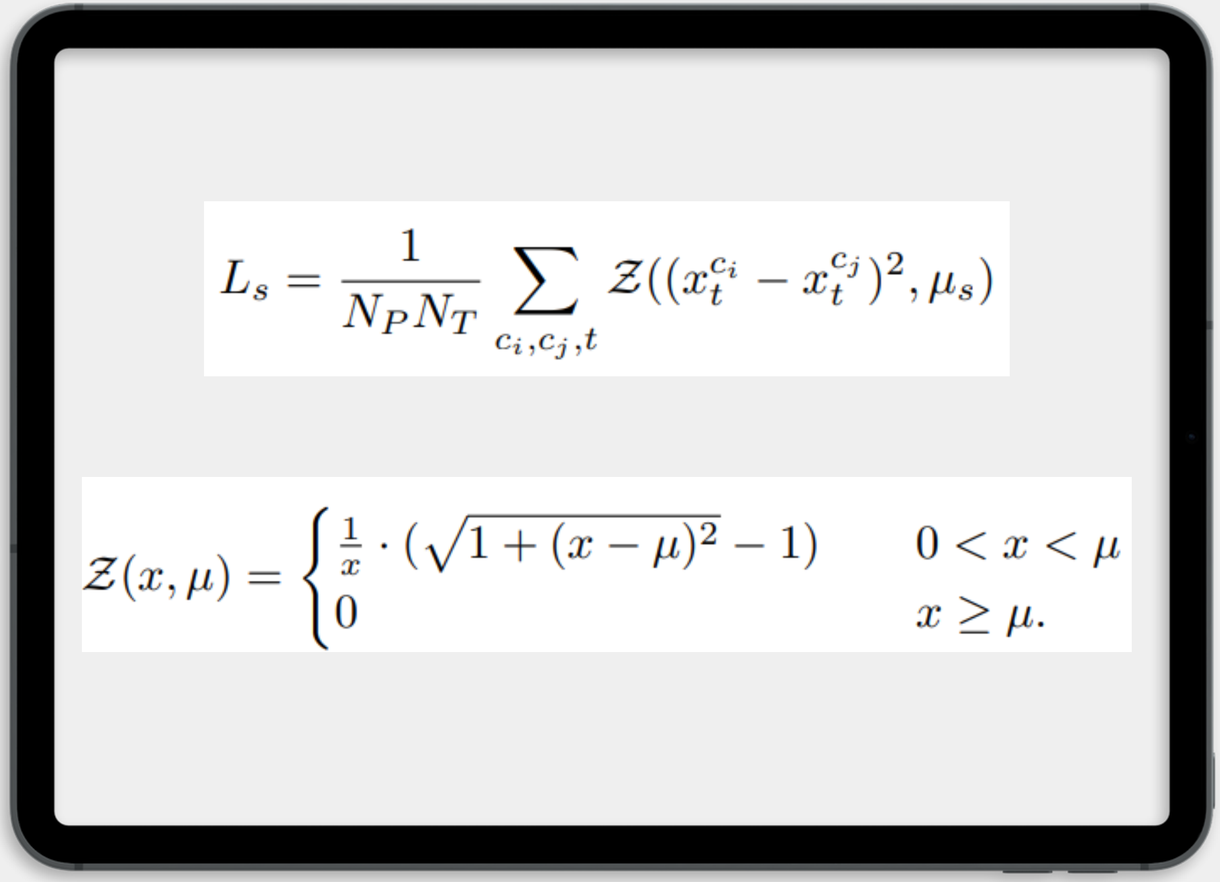
01 The co-occurence score is a defined quantity that checks whether the characters are interacting in a scene using the geometric mean of the number of shots they appeared in per scene.

02 The L1 loss ensures that if the earlier score is high, the lines are closer in that scene for the 2 characters.

03 The L2 loss ensures that if the co-occurance score is zero, then the coordinates are not close.

Minimum Separation Loss

This is to increase the readability of the story graph. It ensures that the lines of two characters do not overlap.



$$L_s = \frac{1}{N_P N_T} \sum_{c_i, c_j, t} \mathcal{Z}((x_t^{c_i} - x_t^{c_j})^2, \mu_s)$$

$$\mathcal{Z}(x, \mu) = \begin{cases} \frac{1}{x} \cdot (\sqrt{1 + (x - \mu)^2} - 1) & 0 < x < \mu \\ 0 & x \geq \mu. \end{cases}$$

01 If the input value to the Z function is less than a certain decided mu value, then the function penalizes it and hence needs more optimization.

02 If there is enough gap between the lines, then the function returns a 0 which is lesser than the other value it can return.

03 muS is taken to be 0.3 units in our case.

Straight Lines

$$L_l = \frac{1}{N_C N_T} \sum_{c,t} (x_t^c - \mu_{\setminus x_t^c})^2$$

$$\mu_{\setminus x_t^c} = \frac{1}{N_T - 1} \sum_{q \neq t} x_q^c$$

The straight lines loss function tries to keep the lines straight

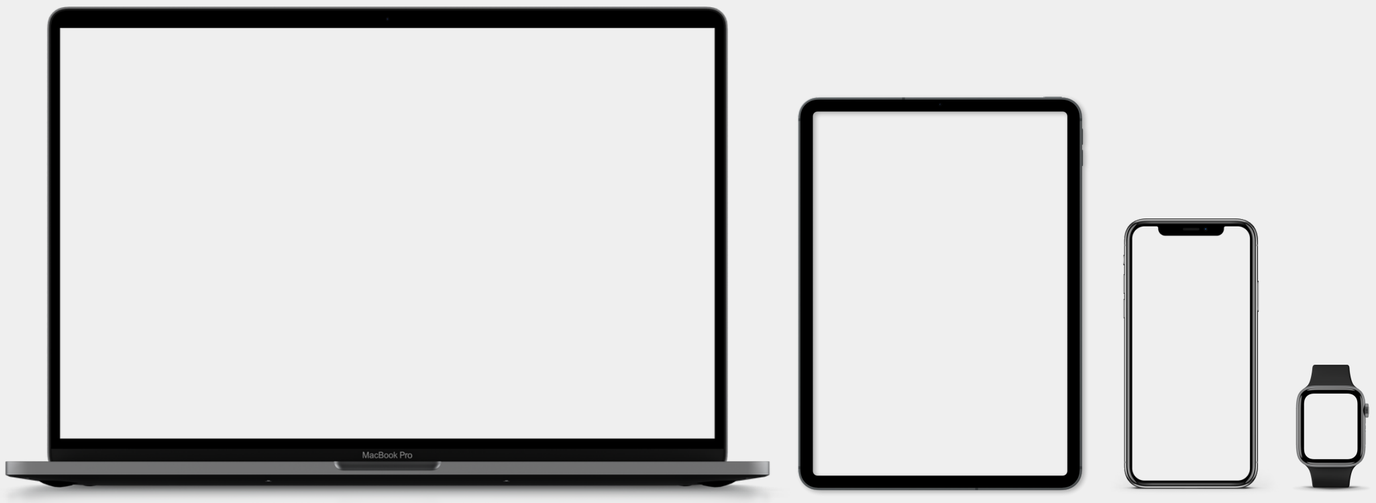
Crossings

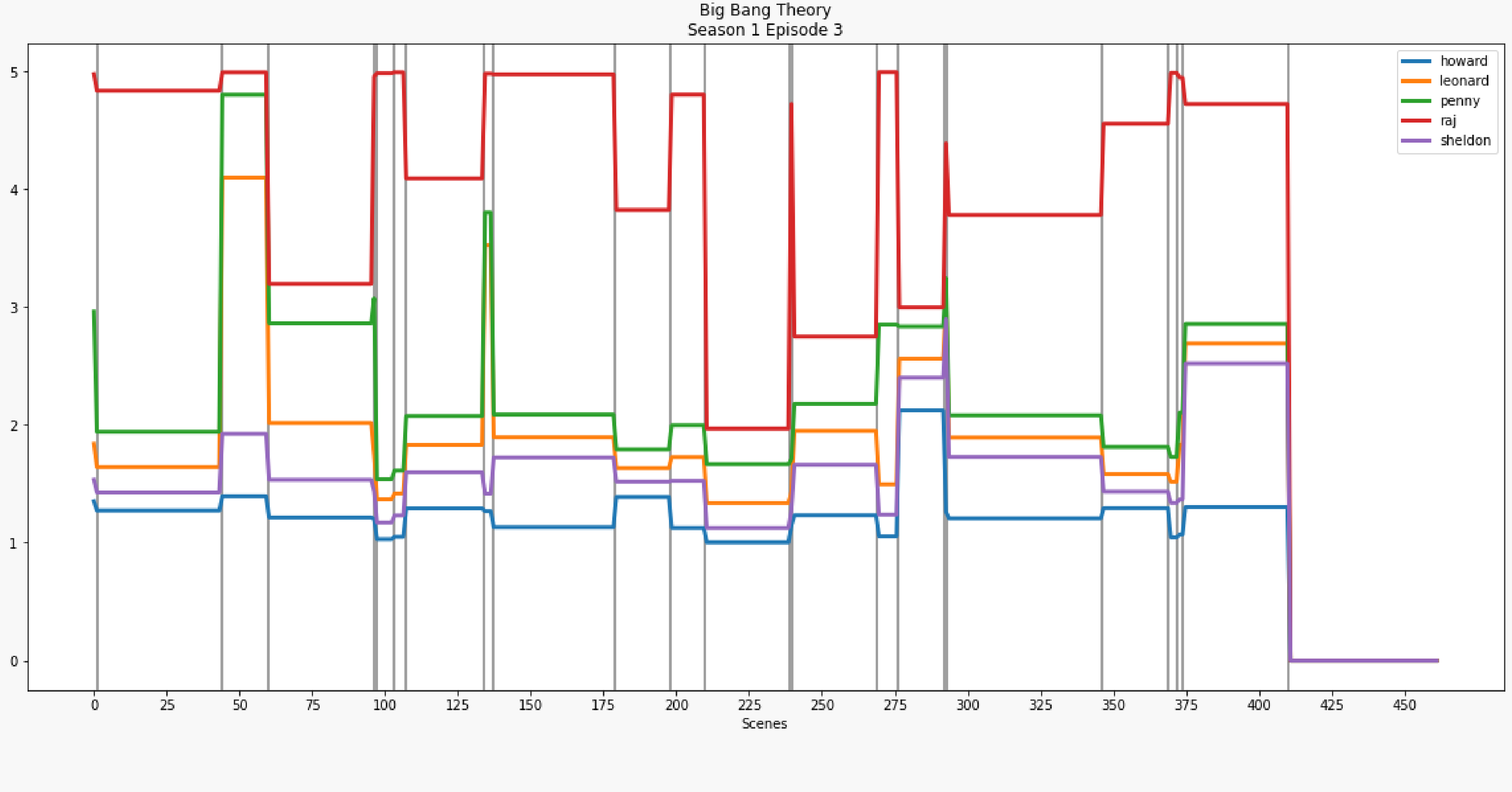
$$L_c = \frac{1}{N_P N_T} \sum_{c_i, c_j, t} \mathcal{H}((x_t^{c_i} - x_t^{c_j})(x_{t+1}^{c_i} - x_{t+1}^{c_j}), \mu_c)$$

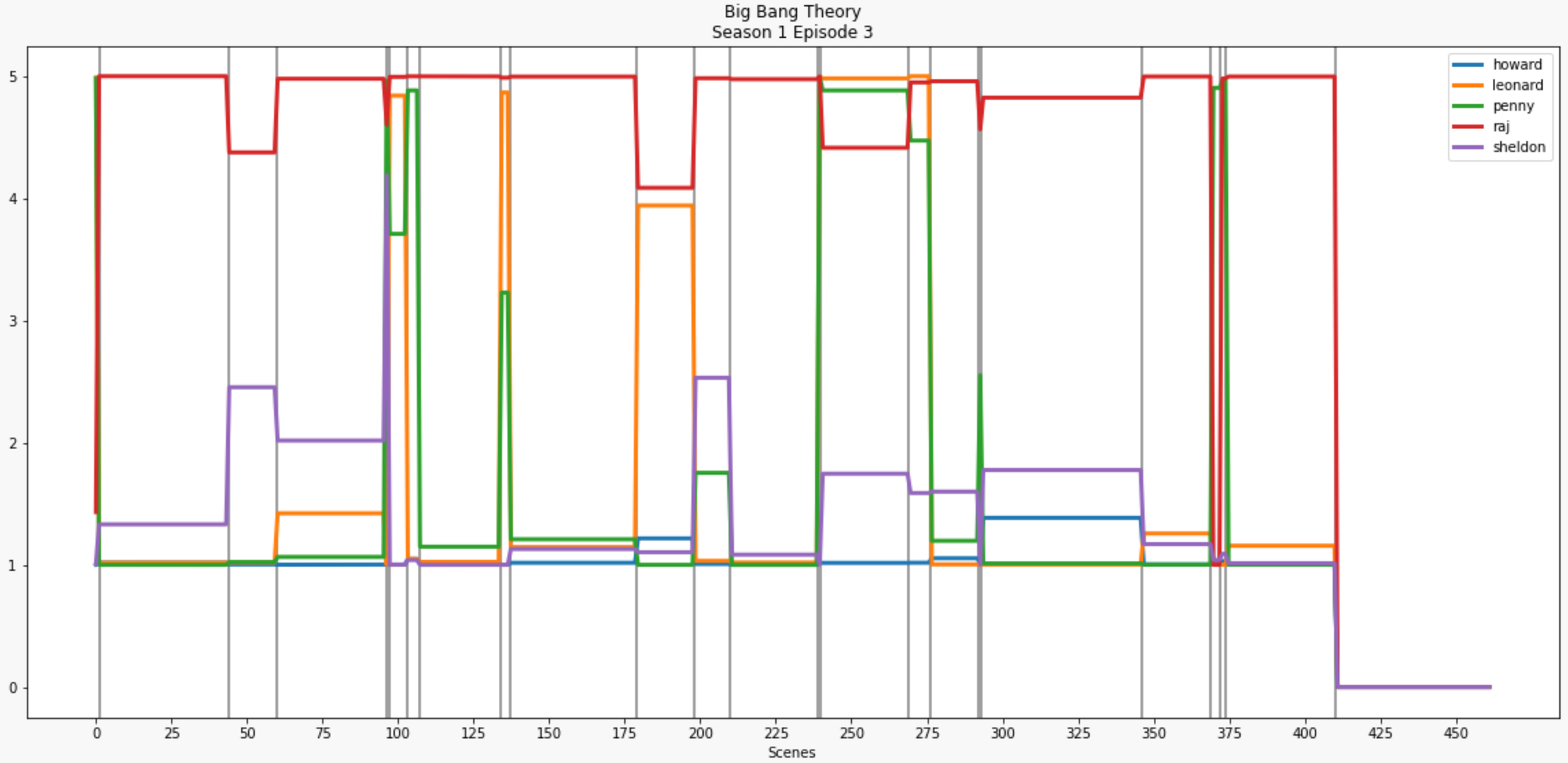
$$\mathcal{H}(x, \mu) = \begin{cases} \sqrt{1 + (x - \mu)^2} - 1 & x < \mu \\ 0 & x \geq \mu. \end{cases}$$

Minimizes the number of crossings between lines of different characters

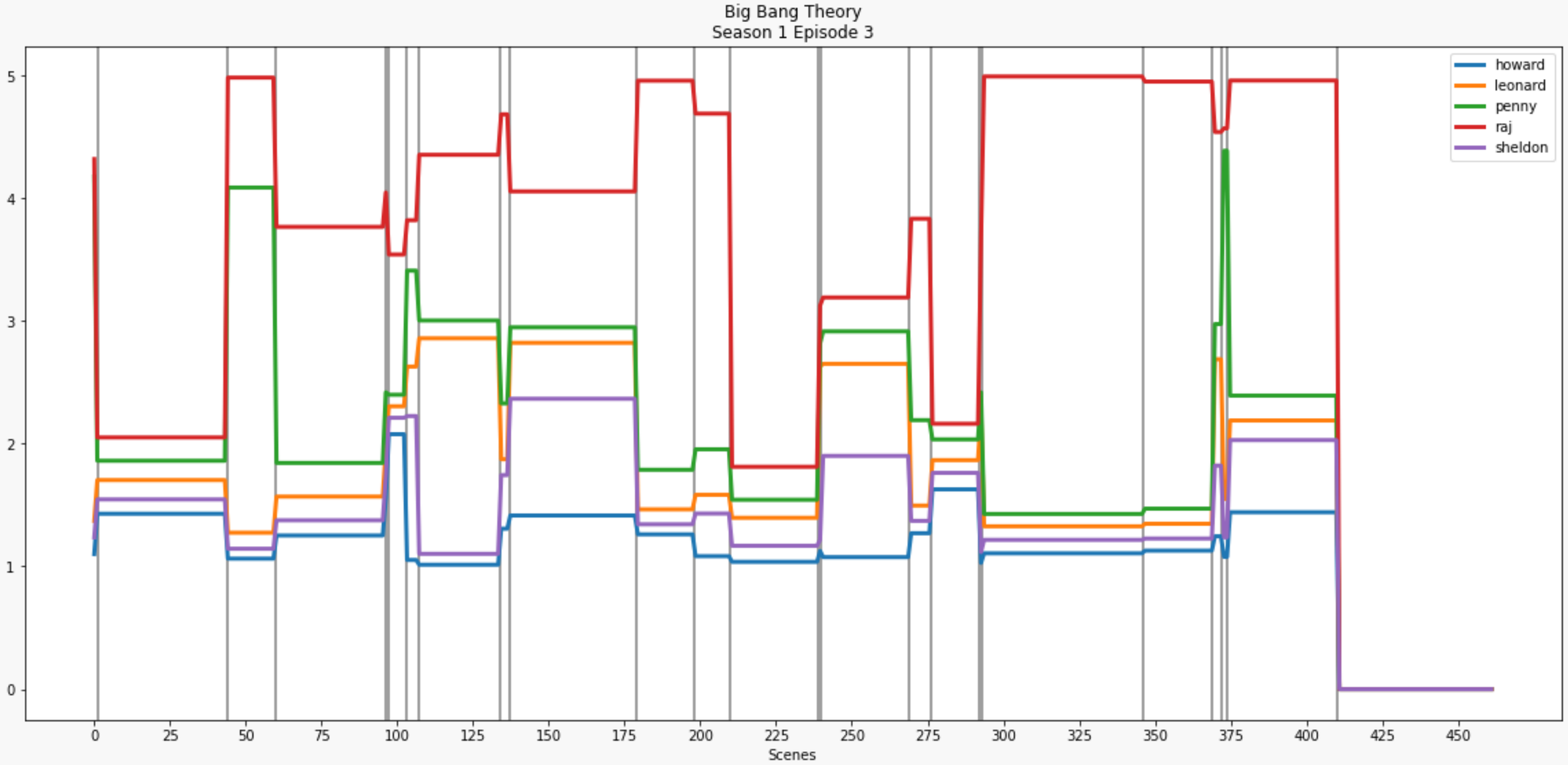
Final Results




Big Bang Theory
Season 1 Episode 3
howard
leonard
penny
raj
sheldon
0
1
2
3
4
5
0
25
50
75
100
125
150
175
200
225
250
275
300
325
350
375
400
425
450
Scenes



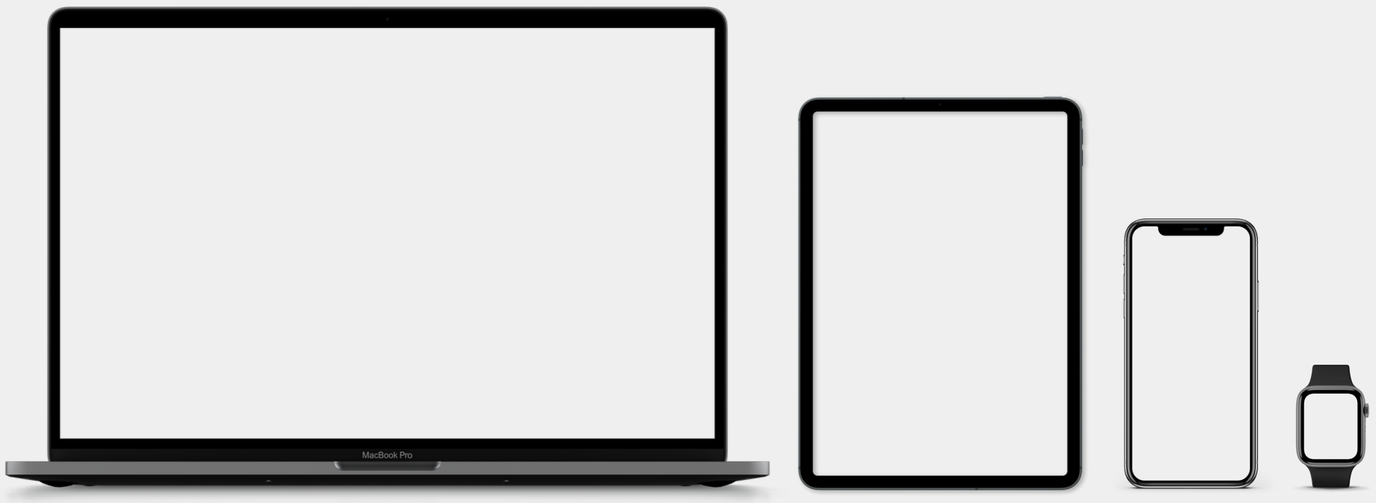
Experiment 1 - Only proximity loss
We observe that when characters interact a lot in a scene,
they overlap.



Experiment 2 -Weighted loss function

Loss = 0.5*proximity + 0.3*minSep + 0.1*(crossings + straightLines)

Evaluation



Move

Normalised maximum coordinate movement for all charecters

$$\text{Move} = \frac{1}{N_C} \sum_{c} \left(\max_{t} x_t^c - \min_{t} x_t^c \right)$$

Maximum Separation

Average (over scenes) of the maximum separation between any two characters

$$\text{MaxSep} = \frac{1}{N_T} \sum_{t} \max_{c_i} \left[\min_{c_j, j \neq i} (|x_t^{c_i} - x_t^{c_j}|) \right]$$

Crossings

Number of times two of the character lines cross

$$\#\text{Cross} = \sum_{c_i, c_j, t} \mathbb{1}\{(x_t^{c_i} - x_t^{c_j})(x_{t+1}^{c_i} - x_{t+1}^{c_j}) < 0\}$$

When given equal weight to all losses

Big Bang Theory Season 1 Episode 3		
Move	Separation	Crossings
2.38646833819401 66	3.1268951918846666	0

Loss = 0.5*proximity + 0.3*minSep + 0.1*(crossings + straightLines)

Big Bang Theory Season 1 Episode 3		
Move	Separation	Crossings
2.00974295904251 13	2.7078560073191698	0

When we have only proximity loss

Big Bang Theory Season 1 Episode 3		
Move	Separation	Crossings
3.112905039211459	3.8876041816406 24	72


Thank You
Computer Blindness
Sreenya Chitluri
Sankeerthana Venugopal
MacBook Pro