Movie Semantic Search

Project Overview

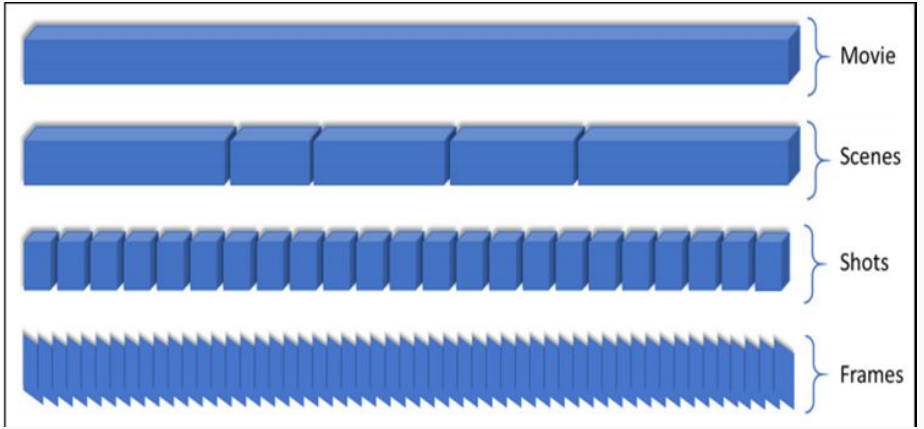
Contents

Introduction

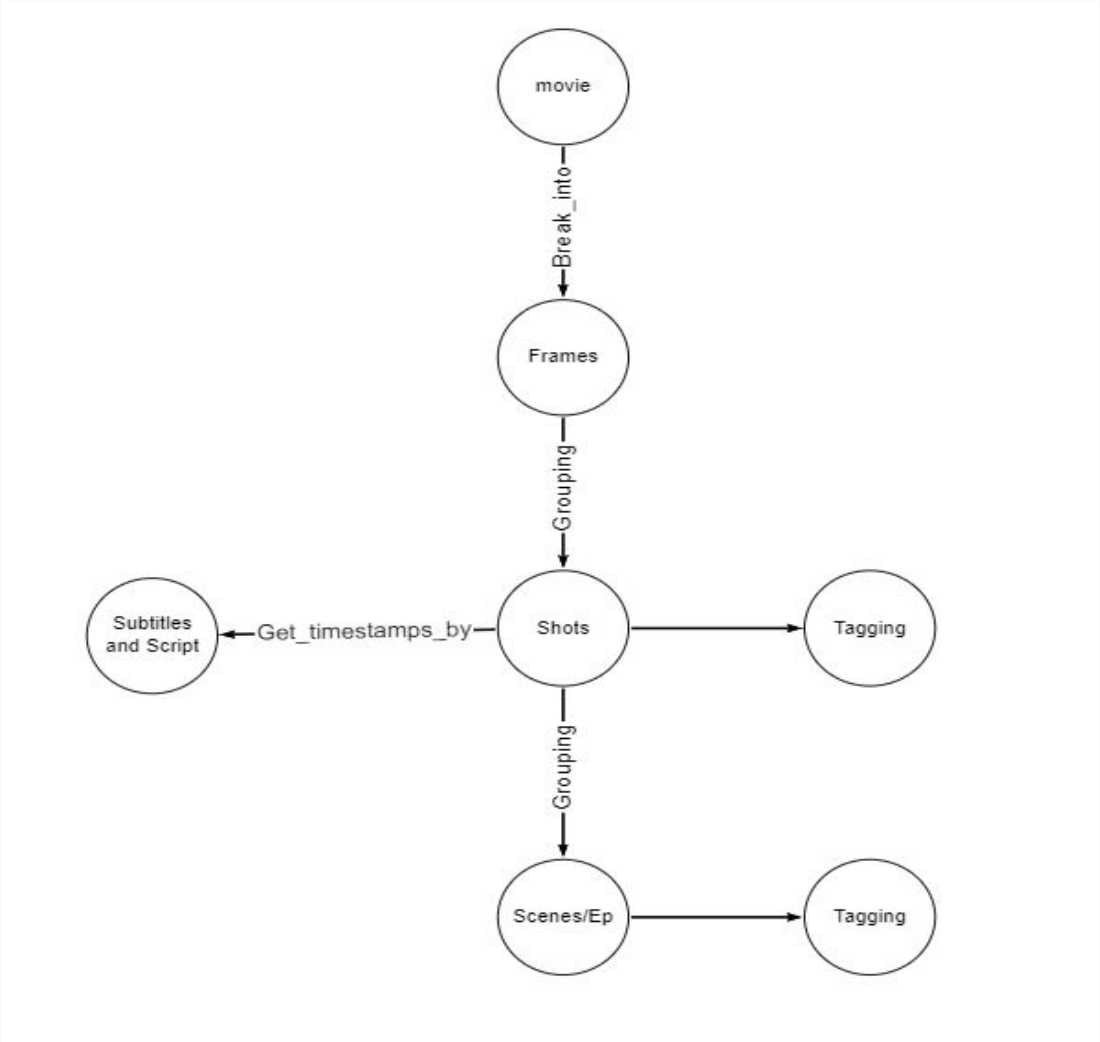
- The aim of our project is to develop a tool that acts as a semantic search engine for movies, making it easier to navigate movie scenes based on certain semantic elements such as actors, emotions, genre, and various other aspects.
- Our project will bring out a new way of movie traversal that will be much more useful for movie analysts.

Movie Breakdown

- A Shot is an unbroken sequence of frames recorded from same camera.
- Scene consists of one or more shots which are semantically co-related and share some physical settings or [resents a continuous action performed by the actors



Overview


movie
Break_into
Frames
Grouping
Subtitles and Script
Get_timestamps_by
Shots
Tagging
Grouping
Scenes/Ep
Tagging

Splitting into Frames

- We can break the movie into frames using *cv2.VideoCapture* function and create video capture object and store the frames using *.read* function as numpy array.
- After obtaining frames we write to a location using *cv2.imwrite* function.
- For a movie of size 2-3 GB we may obtain 80-100K frames, which may take up approximately 100GB.
- So we have used our college's supercomputer, as it is time taking and a CPU intensive process

Frames from the movie You’ve Got Mail

Grouping Frames into Shots

Colour Histogram

●A color histogram is a graphical representation of a distribution of colors within an image.

●The histogram is derived by counting the number of pixels of each of given set of color ranges.

●Color histograms are flexible and can be built for any kind of color space.

Example

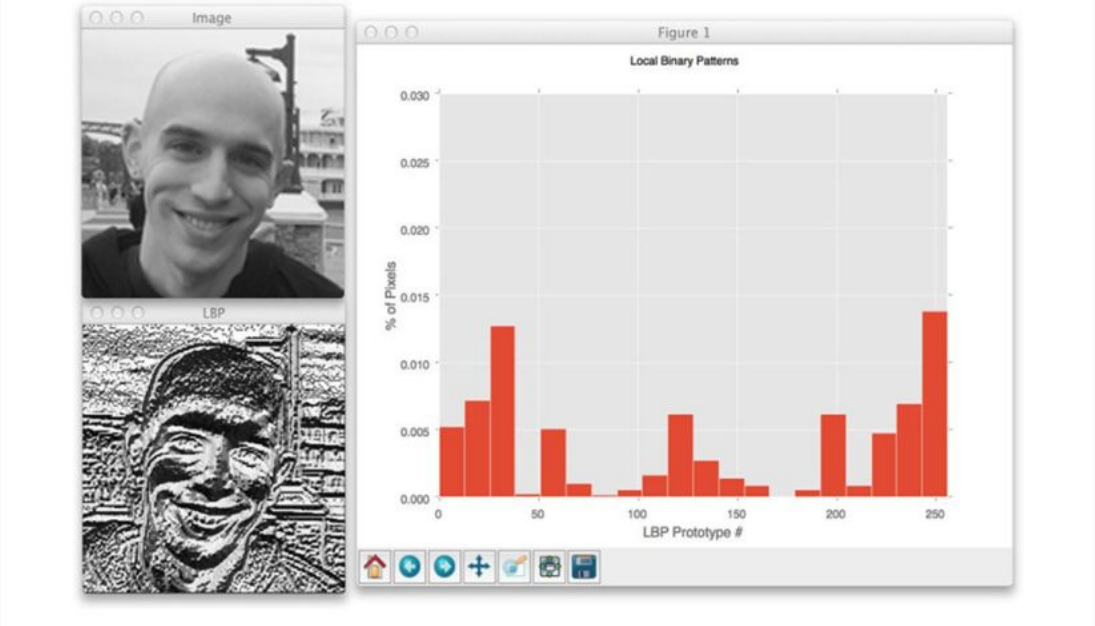
Features and Comparison

- Color Histograms have no concept of the shape of an object or the texture of the object.
- They show the statistical distribution of colors and the essential tone of an image.
- OpenCV has a built in method compareHist() which takes 2 images for which the color histogram is calculated using calcHist() and normalized the values between 0 and 1 before passing it in compareHist() function.

Local Binary Patterns

- Local Binary Pattern (LBP) is an effective texture descriptor for images which thresholds the neighboring pixels based on the value of the current pixel.
- It is a very powerful descriptor that detects all the possible edges in the image.
- The first step in constructing the LBP texture descriptor is to convert the image to grayscale and then perform the necessary computations.

The LBP 2D array has a *minimum value of 0* and a *maximum value of 255*, allowing us to construct a 256-bin histogram of LBP codes.



Combining Frames

- We iterated the movie taking 2 frames at a time.
- We compare the 2 frames based on the color histograms and structural similarity and obtain a score.
- If the obtained score is higher than a threshold value then we group the 2nd frame in the shot which already contains the 1st frame.
- Else we start a new shot starting from the 2nd frame.
- This process is repeated by taking the consecutive frames and obtaining the respective structural similarity and colour histogram scores.

Combining Frames: Continued

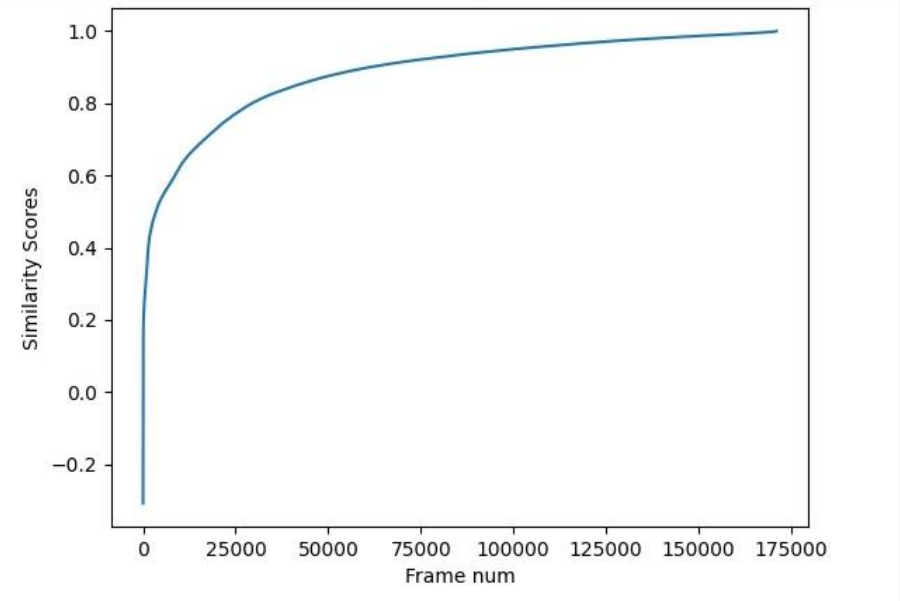
As discussed before, we have used 2 parameters: **Colour Histogram and Structural Similarity**

- We have predefined functions which returns the Colour Histogram and Structural Similarity of the given image.
- So, for the Histogram we have a function called “cv2.compareHist” which returns score between [0,1], 0 indicating both the images are similar.
- For, the Structural Similarity, we have a function called “structural_similarity” which returns score between [0,1] and 1 being both images are similar.

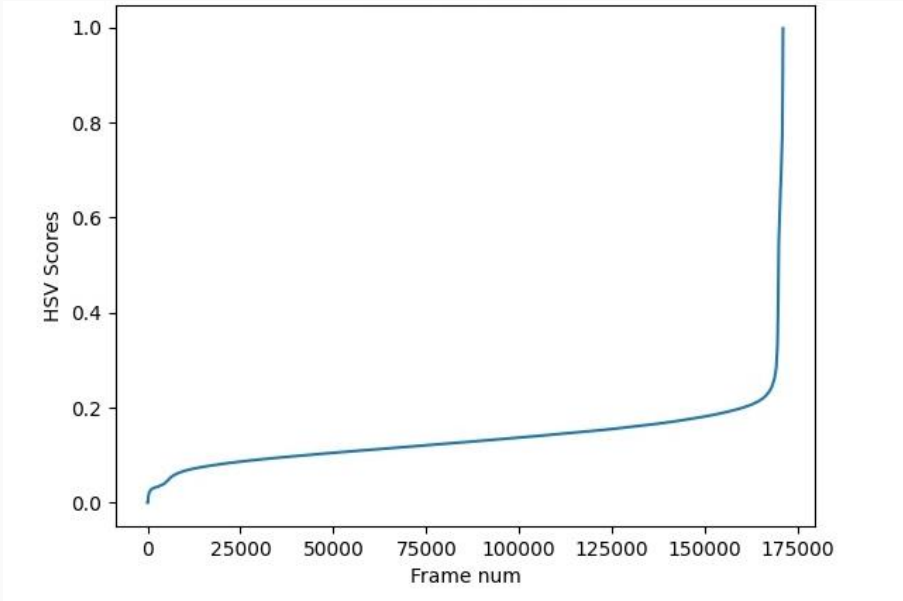
How did we decide on threshold?

- Scores of both Histogram and Similarity functions are taken and combined.
- We plotted a graph of both similarity and histogram scores and the change in slope of the graph gives us the correct score for that movie, so that the grouping of shots is done perfectly.

Graph plots of Histogram and Structural Similarity



Similarity values



Histogram values

Example of how grouping of frames work









Zero length shots

What are zero length shots?

- In the grouping of frames into shots, we have to compare 2 frames and place similar frames in a shot.
- We would be having a zero-length shot if the threshold value if very low and only a frames or very few frames are present in a shot.
- This is due to the threshold score obtained by comparing 2 frames(say they belong to same shot) is too low.

Handling zero-length shots

- To Handle zero-length shots we iterated over the shots that we got from the previous step.
- We then combined the zero-length shot with either the shot before or after based on the similarity of our zero-length with either of the two shots.
- The similarity was obtained by comparing the first frame of the zero-length shot with the last frame of the previous shot and the last frame of the zero-length shot with the first frame of the next shot.

Grouping into Scenes

We have followed 2 methods to group shots into scenes:

1. Using the Set Theory method discussed in the article, it is not very accurate as it defines a scene based on the objects it detected and if the objects detected, for example, are same in two shots(although two shots have different persons and the # of persons are same) then it has placed a shot of next scene in present scene.Hence, we have to perform celebrity face recognition to differentiate for such shots.

2. The second method is:

   To get scene boundaries by matching the dialogues from the subtitles and the script and extracting timestamp of scene from the subtitle file. To achieve this we should have all the movie scripts in a certain format only, and also extracting data from PDF's is not an easy task, thus we used scripts of html format.

```
1
00:01:58,991 --> 00:02:00,367
Good luck.

2
00:03:39,216 --> 00:03:42,136
Hi. I have an appointment with...

3
00:03:42,303 --> 00:03:43,637
...Emily Charlton?

4
00:03:44,763 --> 00:03:47,266
- Andrea Sachs?
- Yes?

5
00:03:47,433 --> 00:03:51,854
Great. Human Resources certainly has
an odd sense of humor. Follow me.

6
00:03:54,064 --> 00:03:56,901
Okay. So I was Miranda's
second assistant...
```

Subtitle File

9 INT. RUNWAY RECEPTION AREA -- DAY 9

Sleek, elegant, hard-edged chic. Behind the reception desk is an elegant logo that says RUNWAY. ANDY walks over.

ANDY
Hi, I have an appointment with Emily Charlton--

EMILY (O.S.)
Andrea Sachs?

(EMILY (and MIRANDA, later) pronounce ANDREA Ahn-DRAY-a. ANDY refers to herself as AN-dree-a.)

ANDY turns and sees a taller, thinner and, amazingly, more groomed CLACKER. This is EMILY. She looks the part of the sleek fashionista, but is propelled by a core of barely tamped down anxiety. She examines ANDY.

EMILY (CONT'D)
Human Resources certainly has a bizarre sense of humor.
(sigh, annoyed)
Follow me.

10 INT. RUNWAY HALLWAY -- DAY 10

Script File

Detecting Scene Boundaries

- The scripts we obtained were in the HTML format. We used the beautiful soup library to segment the HTML script file.
- We were able to extract details from the subtitle file using the pysrt module in python, which gave us the timestamps and dialogues separately using some predefined functions.
- Taking the first dialogue of every scene we match it with the dialogue in the subtitle using python Regex and get the timestamp of the dialogue from the subtitles which we assume to be the timestamp of the scene.

Comparison Function

- We used a simple comparison function where we take 2 sentences.
- We then take the sentence with shorter length of these 2 sentences.
- We then check how many words from the shorter sentence are present in the longer sentence and divide that by the length of the shorter sentence to get the similarity score.
- This proved to be very effective in matching the dialogues in the subtitle file with that of the script.

Output

```
Diag:  Hi. I have an appointment with...
Line:   5 EXT. ELIAS-CLARKE -- DAY 5
*********Scene Found***********
Diag:  Hi. I have an appointment with...
Line:    ANDY sees a tower looming in front of her. Elias-Clarke. Streaming into the building are the polished GIRLS we saw in
the opening... their heels click-clack on the concrete... They are the CLACKERS of Elias-Clarke. ANDY runs in.
1.0
Matched dialogue:    Hi, I have an appointment with Emily Charlton--
1.0
Diag:  Hi. I have an appointment with...
Line:   6 OMITTED 6
*********Scene Found***********
Diag:  Hi. I have an appointment with...
Line:
1.0
Matched dialogue:    Hi, I have an appointment with Emily Charlton--
1.0
Diag:  Hi. I have an appointment with...
Line:   7 INT. ELIAS-CLARKE ELEVATOR -- DAY 7
*********Scene Found***********
```

Method 2: Scene boundary detection using set theory method

Set Theory method

- Entirely based on object detection and set theory.
- Objects detected in each shot are to be stored.
- We compare the objects detected in each shot and then decide on the scene boundary.
- Object detection is carried out using YOLO.
- Face Recognition is done using the Amazon Web Services.

Algorithm for Set Theory method

We take the sliding window approach to find the scene boundary, first say we have 4 shots in the window which belong to the same scene, then say there is a 5^{th} shot which has to be compared. We take the individual intersection of 5^{th} shot with all the 4 shots and If the number of elements of intersection is greater than or equal to any of the total elements of sets of 4 shots, the shot is considered to be a part of the previous running scene and the window moves to the next shot

Working of Set Theory based Scene Boundary detection

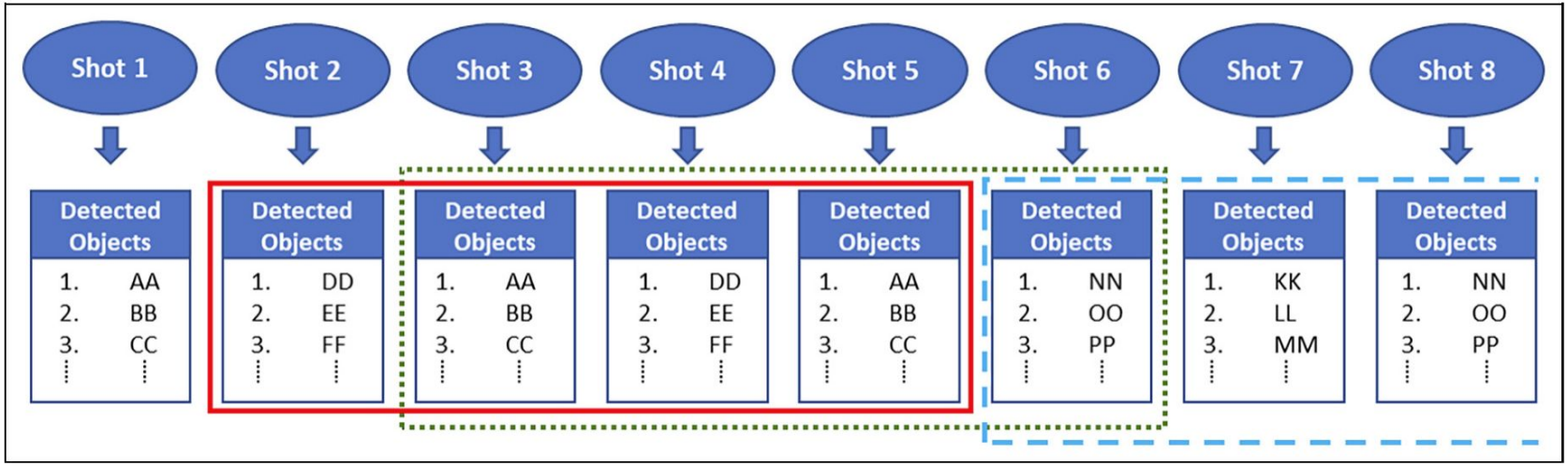


Figure 4. Scene boundary detection using sliding window approach.

Step 1: Object detection

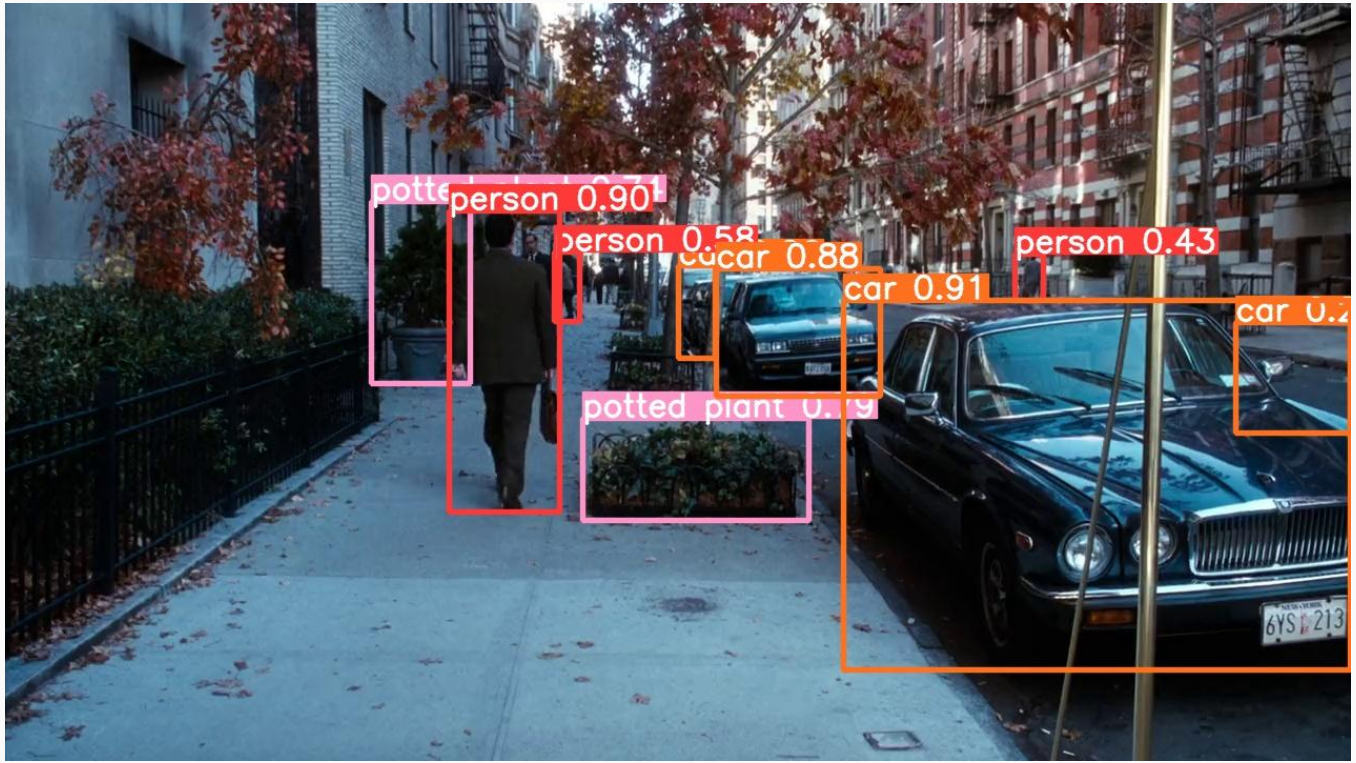
Object detection using YOLO

- YOLO v3 was the version which was mentioned in the paper which we have followed, and thus we started experimenting with v3.
- Then, we came across v5 and v8 versions of YOLO and also tried the object detection with it.
- We have observed that YOLO v8 can classify many classes of objects than v5 and v8.
- Then, we compared all the 3 versions and came to a conclusion that the number of classes which it can detect was increasing over the versions but the accuracy of v8 was a bit low compared to v3 as well as v5.
- Thus we took a trade off between accuracy and number of classes and choose the v5 version for the object detection.

Results of Object detection

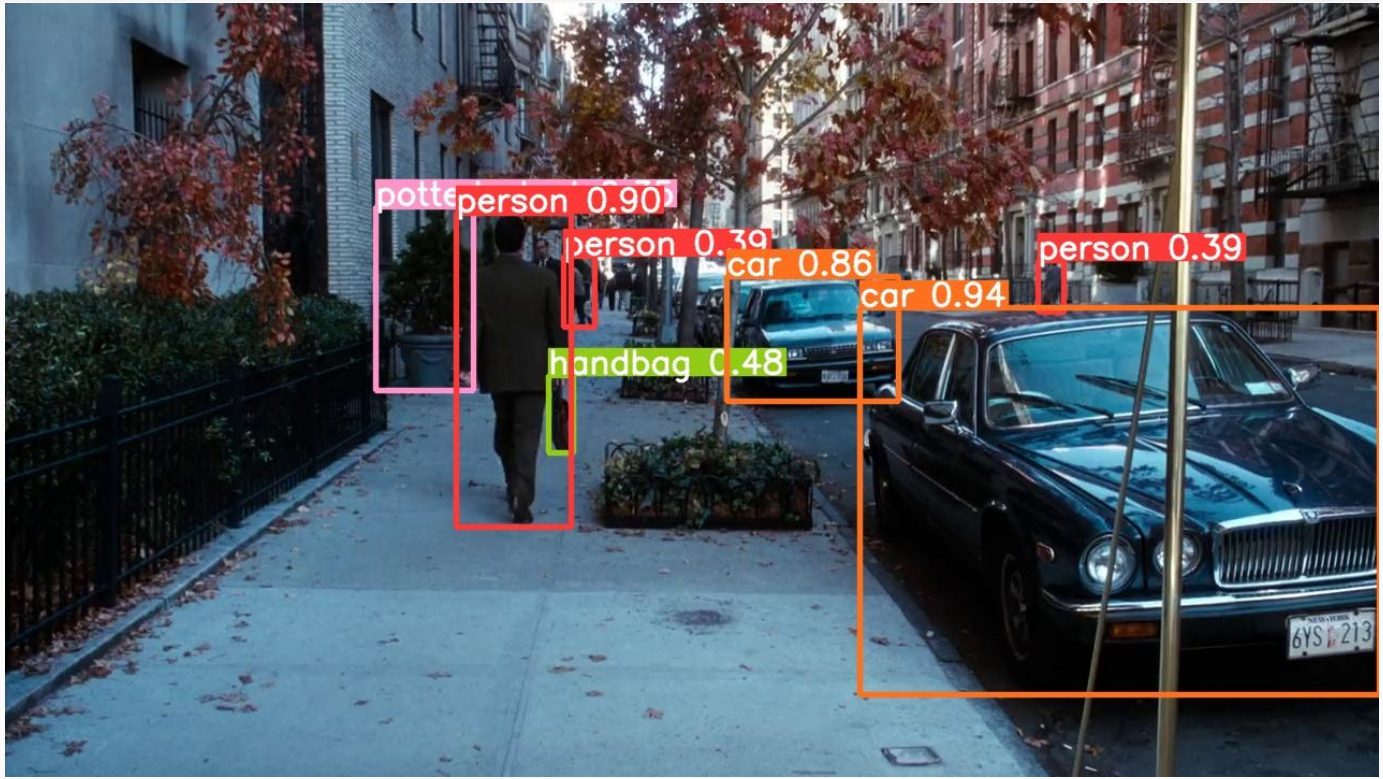
YOLO v3

Results of Object detection



YOLO v5

Results of Object detection



YOLO v8

Step 2: Face Recognition using AWS

Amazon Rekognition

- Celebrity Detection is a very tedious task as we have to train a model for every celebrity.
- We used an AWS service called ‘Amazon Rekognition’ to detect and recognise the celebrities in a frame.
- Rekognition contains pre-trained models which can detect celebrities from across the world which can be accessed by the user with the help of an API.
- The API returns data such as the position of detected faces, the celebrity identified, the confidence score of the recognition and other such data.
- However there are a few lesser known actors who cannot be detected by this model.

Results of Amazon Rekognition

Test Image

```
Detected faces for Test1.png
Name: Tom Hanks
Confidence: 96.6952133178711
KnownGender: Male
Position:
   Left: 0.25
   Top: 0.16
Info
   www.wikidata.org/wiki/Q2263
   www.imdb.com/name/nm0000158

Name: Dave Chappelle
Confidence: 99.88724517822266
KnownGender: Male
Position:
   Left: 0.24
   Top: 0.21
Info
   www.wikidata.org/wiki/Q40321
   www.imdb.com/name/nm0152638

2
PS C:\Users\nikhil\Desktop\Movie_Semantics\AWS>
```

Results

Results of Amazon Rekognition

Test Image

```
Detected faces for Test6.png
Name: Steve Zahn
Confidence: 99.94541931152344
KnownGender: Male
Position:
   Left: 0.22
   Top: 0.16
   www.imdb.com/name/nm0001872

Name: Meg Ryan
Confidence: 99.5303955078125
KnownGender: Female
Position:
   Left: 0.22
   Top: 0.19
Info
   www.wikidata.org/wiki/Q167498
   www.imdb.com/name/nm0000212

2
```

Results

Drawbacks of Set Theory method

- The Set theory method works under the assumption that once a scene breaks the next 3 shots belong to the same scene and performs grouping starting from the fourth shot.

- So, there might be a possibility that the shots taken in the window might not be a part of the same scene and thus this is a major drawback in the set theory.

- In the article, there is no proper implementation is not mentioned and also there is an ambiguity in the algorithm proposed.

- We found another implementation for scene boundary detection using tool called MovieNet.

Future Endeavors

Scene Boundary detection using MovieNet

- In the paper “MovieNet: A Holistic Dataset for Movie Understanding” we have found a new way of implementation, to find the scene boundary.
- They have considered different aspects of a movie to obtain the scene boundary
- Different parameters they considered were:
  - Actors present in the shot
  - Setting of the shot
  - Actions performed by actors
  - Audio similarity between shots

Tagging using AWS

Scene Level Tagging

We tag the following data for scenes:

1. Scene semantics through the script.
2. Songs played in the background.
3. Characters and the actors names in the scene.
4. Timestamp of a scene(start time).
5. Locations (INT/EXT followed by the place names).
6. Genre of the scene and also about the important characters and objects which may relate to the future scenes.

For shot level tags we can only have metadata like the objects and the people identified, (Each person should be stored as a unique identity)

only for the scene segmentation which is done in the article.

Tags such as timestamp for a shot(Hrs:Min:Sec:Frame.no) and also camera angles could be included.