Indexing and Retrieval of Video Lectures

Progress Seminar

by

Narayan Kunal
Roll No. 18CS60R56

under the supervision of

Prof. K Sreenivasa Rao
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur

Overview

1. Introduction
2. Objective
3. Experimental Setup
4. Terminology
5. Ground Truth
6. Slide Extraction
7. Result
8. Topic/Subtopic
9. Conclusion
10. Future Work

Introduction

- Digital video has become a popular storage and exchange medium due to the rapid development in recording technology, improved video compression techniques and high-speed networks in the last few years.
- As a result, there has been a huge increase in the amount of multimedia data on the Web. Therefore, for a user it is nearly impossible to find desired videos without a search function within a video archive.
- The requested information may be covered in only a few minutes, the user might thus want to find the piece of information he requires without viewing the complete video.

Objective

- Most of the video retrieval and video search systems such as YouTube, Bing and Vimeo rely on available textual metadata such as title, genre, person, and brief description, etc.
- Generally, this kind of metadata has to be created by a human to ensure a high quality, but the creation step is rather time and cost consuming.
- Our target is to generate metadata automatically using video analysis tool.

Experimental Setup

- Course Details
  - Name : Cloud Computing
  - Instructor : Prof Soumya Kanti Ghosh
  - Institute : IIT Kharagpur
  - Link : `https://nptel.ac.in/courses/106105167/`
  - Numbers of lectures : 40
- Tools used
  - OpenCV for video processing
  - Pytesseract for OCR

Terminology

- Mean Square Error(MSE) : MSE is the cumulative squared error between the two frame F1 and F2.

$$MSE(F1, F2) = \frac{1}{MN} \sum_{n=1}^{M} \sum_{m=1}^{N} [F2(n, m) - F1(n, m)]^2$$

- Manhattan Distance : The manhattan distance is the simple sum of the horizontal and vertical components.

$$MH(a, b) = |x_1 - y_1| + |x_2 - y_2| + ... + |x_n - y_n|$$
$$\text{where } a = (x_1, x_2, x_3, ..., x_n), b = (y_1, y_2, y_3, ..., y_n)$$

- Euclidean Distance : It is the straight-line distance between two pixels. the euclidean distance between two points $a = (a_x, a_y)$ and $b = (b_x, b_y)$ is defined as :

$$d(a, b) = \sqrt{(b_x - a_x)^2 + (b_y - a_y)^2}$$

Terminology(contd..)

```python
def euclidean_dist(A,B):
    return np.sqrt(np.sum((A.astype("float") - B.astype("float"))**2))

def manhattan_dist(A,B):
    return np.sum(abs(A - B))

def mse(imageA, imageB):

    err = np.sum((imageA.astype("float") - imageB.astype("float")) ** 2)
    err /= float(imageA.shape[0] * imageA.shape[1])

    return err
```

Figure: Python Code

Ground Truth

- This ground truth is from 15 minute clip of Lecture 6 of cloud computing. All analysis has been done based on this.

Time	00:00	00:01	00:08	01:25	01:51	02:54	03:51
Slide No	1	2	3	4	5	6	7

Time	04:11	04:13	04:57	05:19	05:27	06:21	06:23
Slide No	8	9	10	11	12	13	14

Time	07:11	07:38	07:43	08:49	08:50	08:53	09:02
Slide No	15	16	17	18	19	20	21

Time	09:24	09:56	10:02	10:56	11:42	11:47	13:56	14:11
Slide No	22	23	24	25	26	27	28	29

Table: Slide Number vs Timestamp

Slide Example

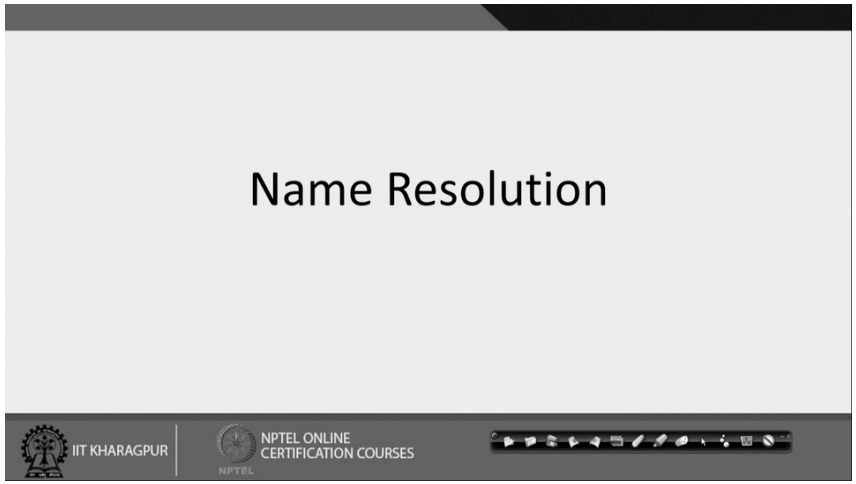


Figure: Topic

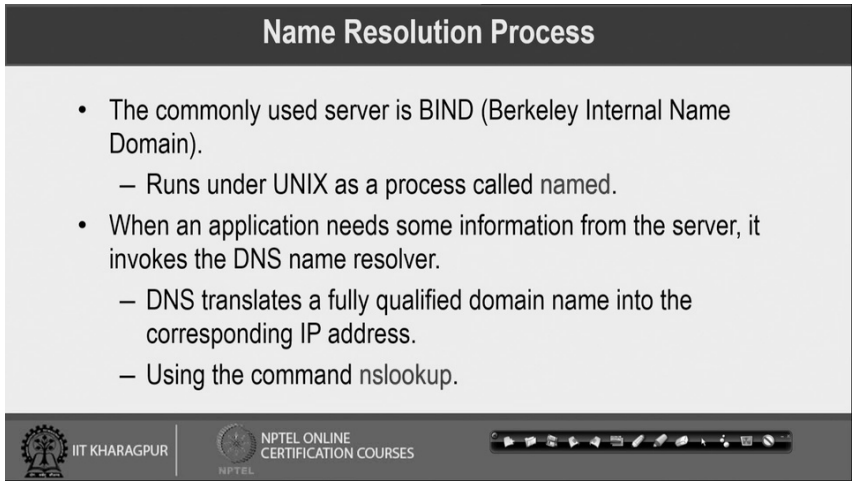


Figure: Subtopic

Slide Extraction

- Convert all frame to gray to make it two dimensional.
- Create canny edge for each frame.
- Calculate MSE, Manhattan Distance and Euclidean Distance between consecutive frames.
- Plot graph of two consecutive frame on different measures like MSE, Manhattan Distance and Euclidean Distance.
- Choose threshold based on above plots.
- Store frame if that frame crossed chosen threshold

Slide Exraction Approach

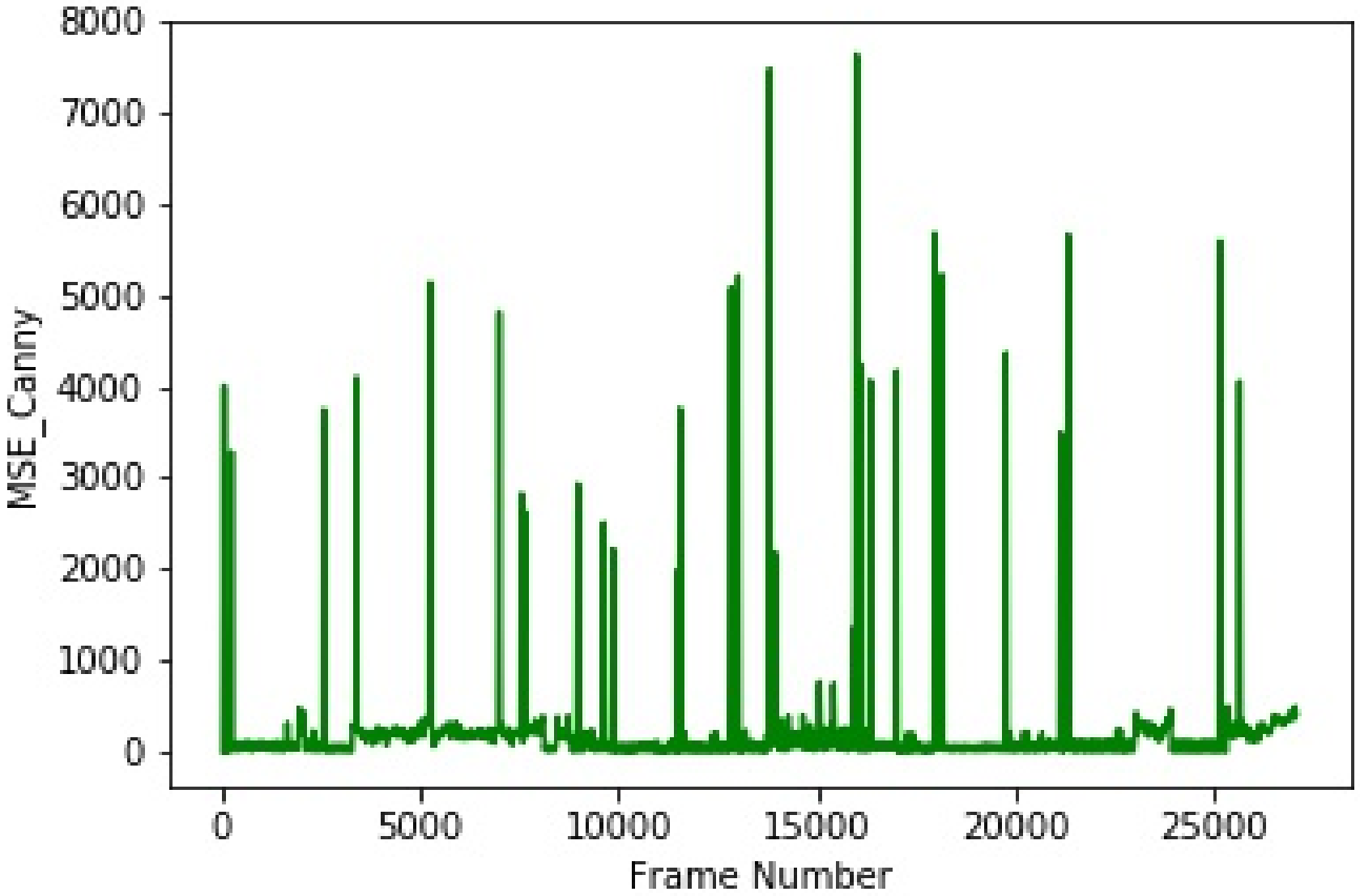
```
capture = cv2.VideoCapture('test1.mp4')
fps = int(capture.get(cv2.CAP_PROP_FPS))
fnum,count = 0,0

ret,prev = capture.read()
prev = cv2.cvtColor(prev,cv2.COLOR_BGR2GRAY)
prev_canny = cv2.Canny(prev.copy(),60,120) #creating canny edge

if ret:
    fnum = 1

while capture.isOpened() and ret:
    ret1,cur = capture.read()
    if ret1 == True:
        if fnum%fps==0:
            count += 1
            cur = cv2.cvtColor(cur, cv2.COLOR_BGR2GRAY)
            cur_canny = cv2.Canny(cur.copy(), 60, 120)
            mean_square_error = mse(cur_canny,prev_canny)
            if mean_square_error > 1500:
                name = 'MSE_Canny_threshold_1500/slide_'+str(fnum)+'.jpg'
                cv2.imwrite(name,prev_canny)
            prev_canny = cur_canny
        fnum += 1
    else:
        break

capture.release()
cv2.destroyAllWindows()
```

Canny Edge Snapshot

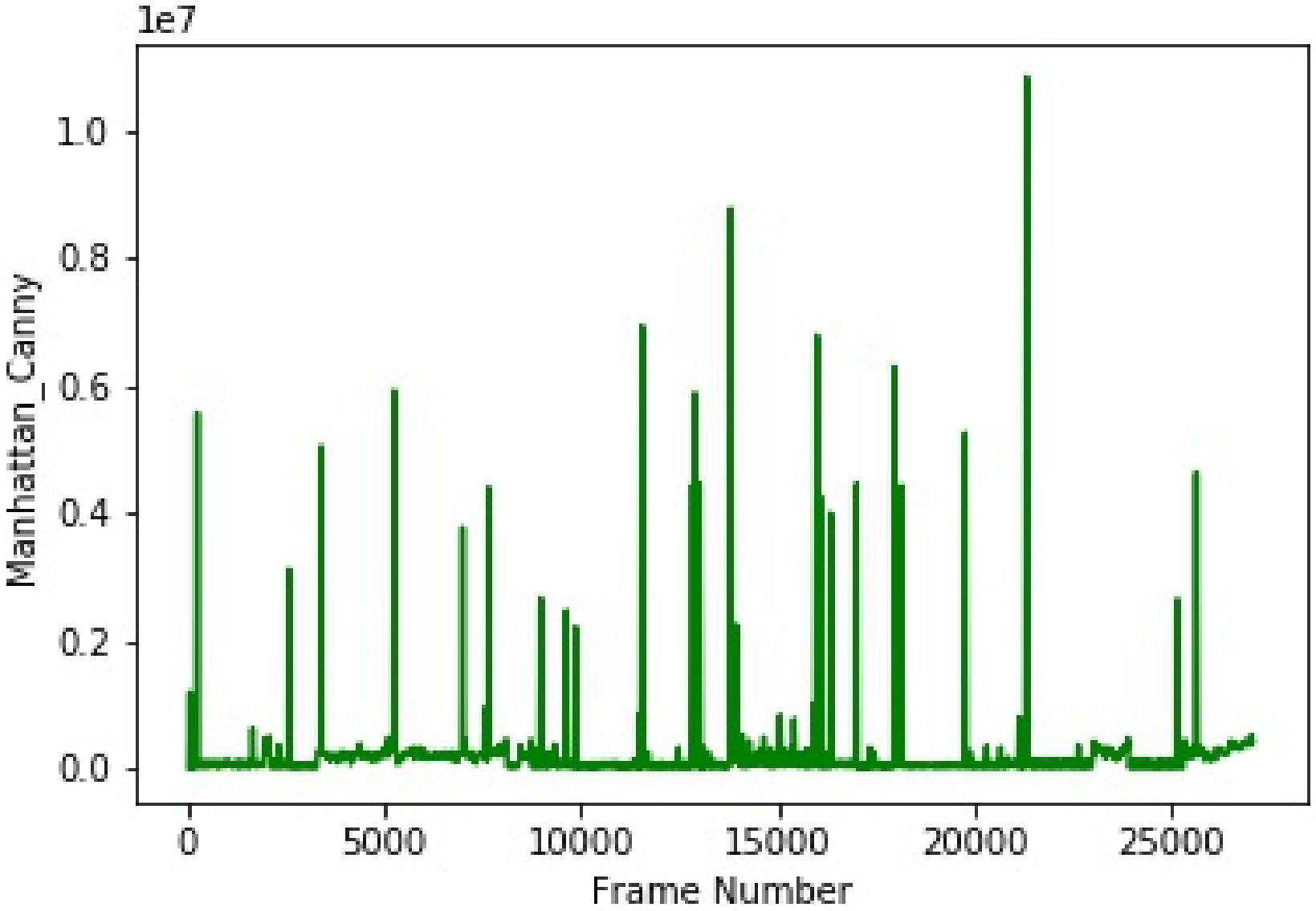
Name Resolution Process

- The commonly used server is BIND (Berkeley Internal Name Domain).
  - Runs under UNIX as a process called named.
- When an application needs some information from the server, it invokes the DNS name resolver.
  - DNS translates a fully qualified domain name into the corresponding IP address.
  - Using the command nslookup.

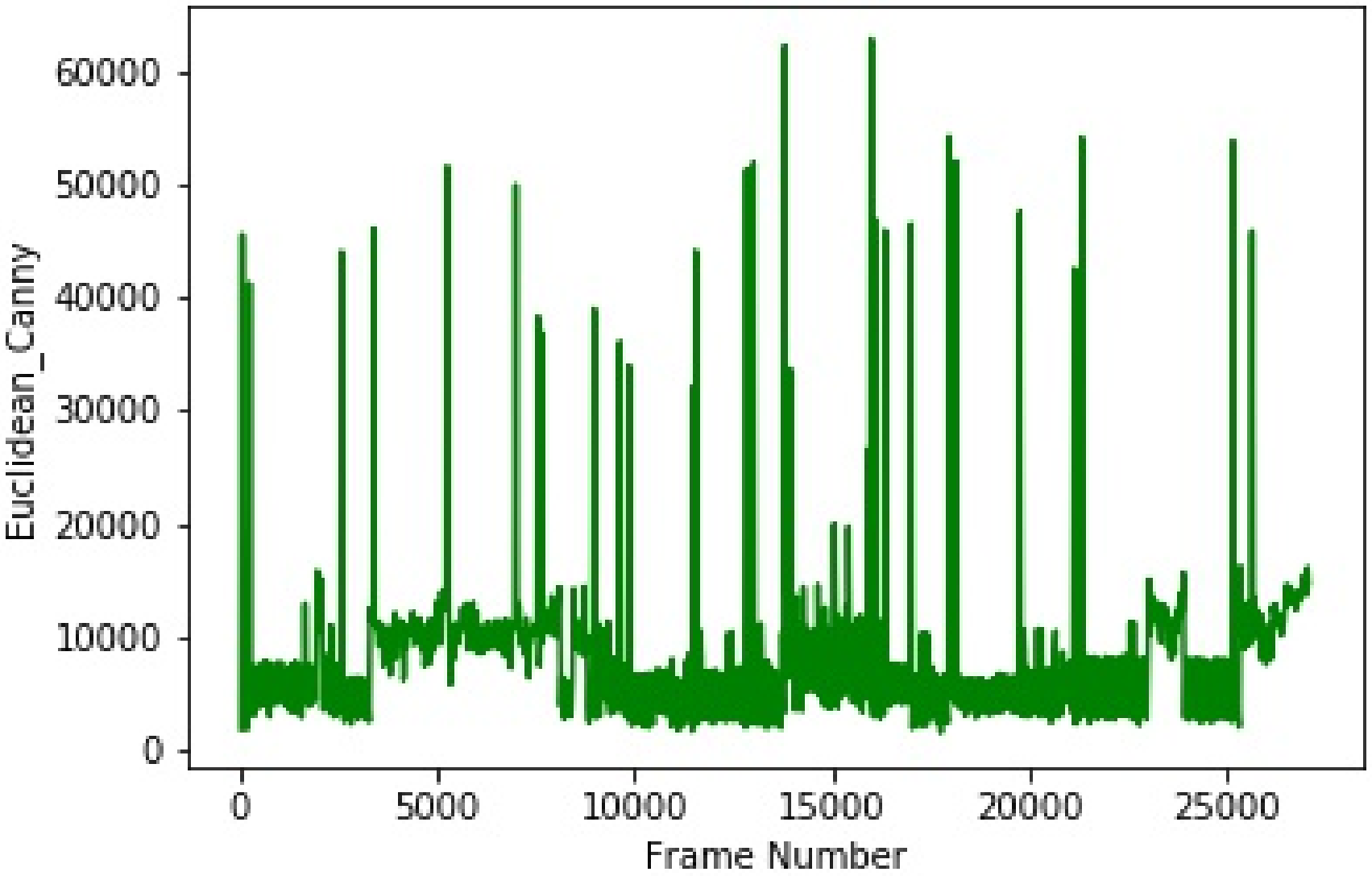
Mean Square Error Plot



Manhattan Distance Plot



Euclidean Distance Plot



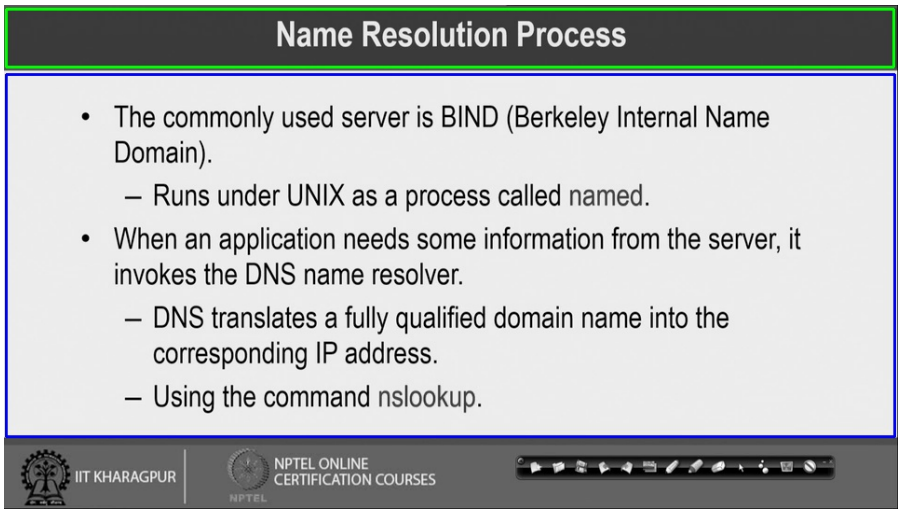
Result

- Result is for same video with different threshold for diffrent measure.

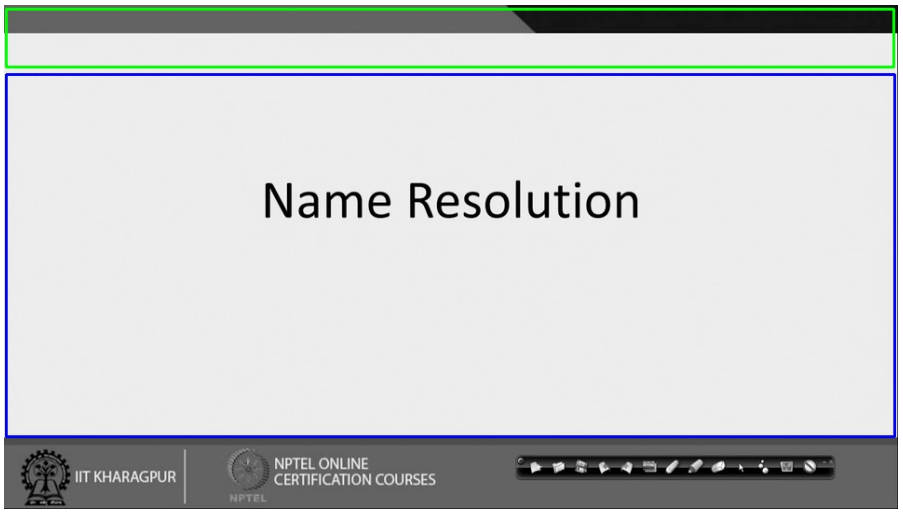
Measure	Threshold	No of Slides	No of Slides Detected	No of Slides Correctly detected
Mean Square Error	1100	29	31	29
	1500	29	29	29
	2000	29	28	28
Manhattan Distance	800000	29	29	28
	1000000	29	27	26
	2000000	29	25	25
Euclidean Distance	20000	29	31	29
	30000	29	28	28
	35000	29	26	26

Topic vs Subtopic

- Blue region is for Topic and Green region is for subtopic.
- OCR will be performed on respective region.



(a) Subtopic



(b) Topic

Conclusion

- Course presentation should be consistent. Latex based would be preferrable.
- Manual fine tuning of threshold is required.
- Region for Topic/Subtopic is dependent on slide format.
- Need to run on different lecture to see the effectiveness of the method.

Future Work

- We will analyse on different course where presentation is consistent.
- Topic/Subtopic selection based on text extraction(OCR) of Topic/Subtopic area.
- Exploration of DNN based approach.

References I

[1] J. Adcock, M. Cooper, L. Denoue, H. Pirsiavash, and L. A. Rowe, "Talkminer: A search engine for online lecture video," in *Proceedings of the 18th ACM International Conference on Multimedia*, ser. MM '10. New York, NY, USA: ACM, 2010, pp. 1507–1508. [Online]. Available: http://doi.acm.org/10.1145/1873951.1874263

Thank You