Determining Video Similarity With Object Detection

Ruochen Liu
Department of Computer Science
Columbia University
rl3160@columbia.edu

Directed Research Report - Spring 2021
Advisor: Prof. John Kender

May 18, 2021

Abstract

Videos, particularly news videos, are likely to contain similar frames if they share the same topics. To determine the similarity between a pair of videos, it is essential to first accurately identify key frames. Objection detection offers a viable way to accomplish this task. This paper will demonstrate a system that automate the video similarity comparison process using object detection.

Keywords— Video similarity, frame comparison, object detection, key frame extraction, video segmentation, computer vision, cross-cultural analysis

1 Introduction

In the era of internet and portable devices, videos are one of the most popular carriers to convey information. News videos, in particular, have clear topics. Reporters from different news agencies usually appear at the same locations during the same event. It is also common for some news agencies to relay others' footage when their reporters cannot access scenes in time. As a result, news videos reporting the same event may contain similar frames even if the sources are distinct.

The interest of this paper is to study the similarity between news videos, especially the ones coming from distinct cultural backgrounds. Comparing videos frame by frame and pixel by pixel is straightforward, but this process consumes an enormous amount of computation power. Consider a 3-minute long video and extract one frame per second, producing 180 frames in total. Comparing two such videos will result in $180 \times 180 = 32,400$ comparisons, not to mention longer videos. It is possible to utilize visual feature extractor such as VGG-NETS [6] and then compute the distance between each pair of vectors to reduce the computation cost during comparison. However, this optimization is trivial when the number of comparisons is huge. Feeding each image through a feature extractor also takes a significant amount of time. In addition, pixel by pixel comparison is too strict for determining similarity. Given the same scene shot at two different angles, the two frames may be very different in terms of pixel arrangements. Pixels or low-level features thus are not ideal for general similarity comparison.

To reduce the number of comparisons, it is vital to first extract key frames from each video. A key frame is a shot that can best represent a scene or a segment of a video. The other frames can be seen as "duplicates" since they are only slightly different from the key frames. With key frames, the number of comparisons can drop drastically. A 3-minute long video may only contain 10 key frames, and it will only require $10 \times 10 = 100$ comparisons between two such videos.

The question now is how to extract key frames quickly and accurately. Manual labelling is the most reliable way to accomplish this task, but this method is infeasible when a study like cross-cultural analysis requires a substantial amount of videos. To automate the extraction process, one solution is to utilize semantic information or high-level features contained in frames. Objects along with their positions and sizes are examples of such features. YOLO [5], a real-time object detector, is ideal for extracting these high-level features fast without sacrificing much accuracy. A system can identify key frames by comparing object features between frames and then aligns different sets of key frames to determine the similarity. This paper will cover all these steps in the following sections.

2 Methods

2.1 Data

There are in total 108 videos in the database. Most of the videos are news reports and they cover a range of different topics or events. These videos are separated into three sets for different experiments:

1. 50 videos are about AlphaGO, a go-playing computer program that competed with human players

2. 50 are about Change'e 5, a Chinese lunar explorer launched in 2020

3. The rest of the 8 cover events such as the crash of Malaysia Airlines Flight 370 (MH370), Coronavirus, and 2020-2021 Thailand protests

All videos are downloaded from YouTube using the YouTube Data API [8], YouTube's official searching API, and the youtube-dl [9], a third-party YouTube video downloader. The former returns a list of video ids given a query and the latter follows this list to download corresponding videos. Note that the YouTube Data API requires a developer key.

This paper attempts to limit search results to news videos only, but at this moment the YouTube Data API does not support filters that can fulfil this requirement. As a workaround, this paper limits the video duration to 4 minutes, a length corresponding to the short video category considered by the API. There is an additional constraint to limit the spoken language in videos to English.

After downloads complete, each video is then split one frame per second using FFmpeg [1], a command-line based video and audio editor. As a result, the 108 videos generate around 25 thousand frames in total. Below are some examples by topics:

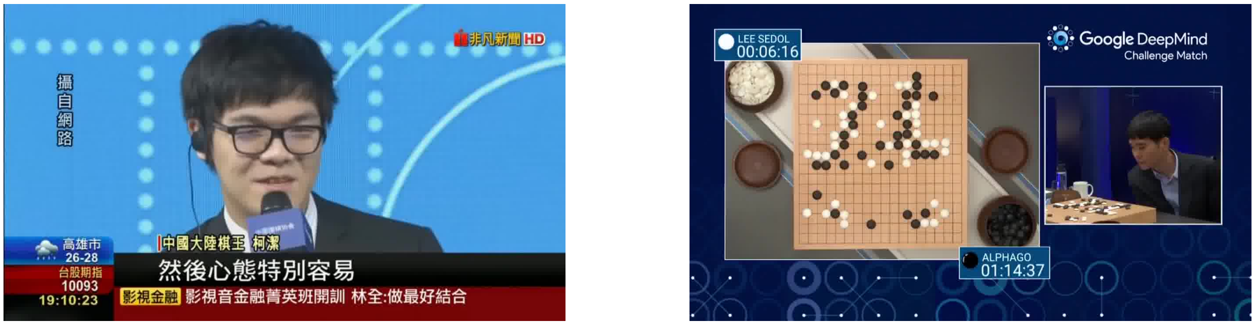

Figure 1: AlphaGO example frames

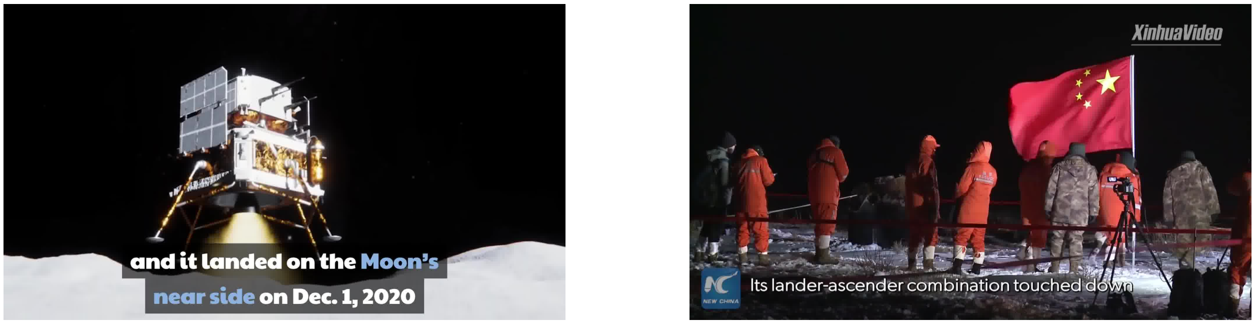
and it landed on the Moon's
near side on Dec. 1, 2020
XinhuaVideo
Its lander-ascender combination touched down

Figure 2: Chang'e-5 example frames

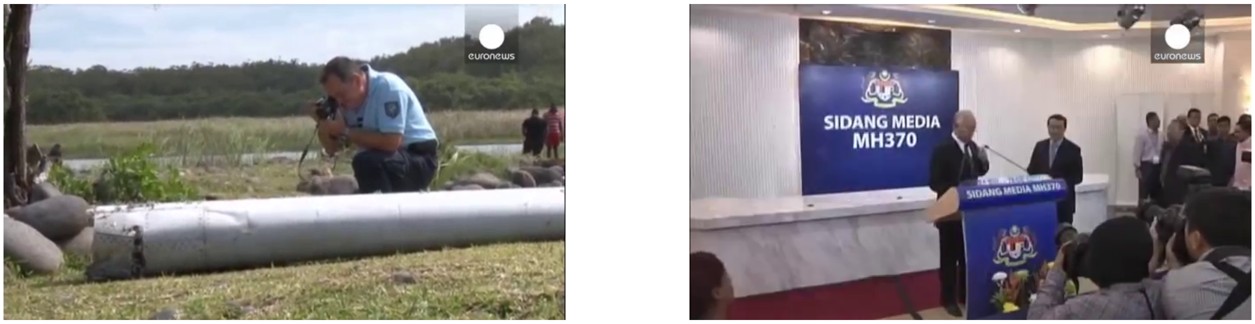

Figure 3: MH370 example frames

Dr Tedros Adhanom Ghebreyesus
Director-General of WHO
I am speaking to you on the eve of a holiday season unlike any other we have known.

Figure 4: Coronavirus example frames

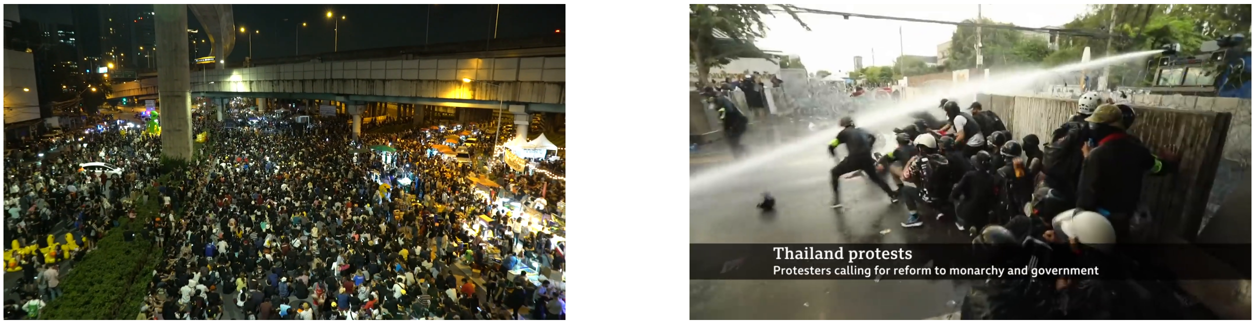

Figure 5: Thailand protest example frames

2.2 Object Detection

The object detection task contains two sub-tasks:

- Localization: where an object lies on an image
- Classification: what category that an object belongs to

To accomplish these two sub-tasks, an object detector requires three modules in the following sequence:

1. a region extractor to propose bounding boxes or locations where objects of interest may exist
2. an feature extractor to extract visual features from each region,
3. and a classifier to predict the class or label of each identified objects.

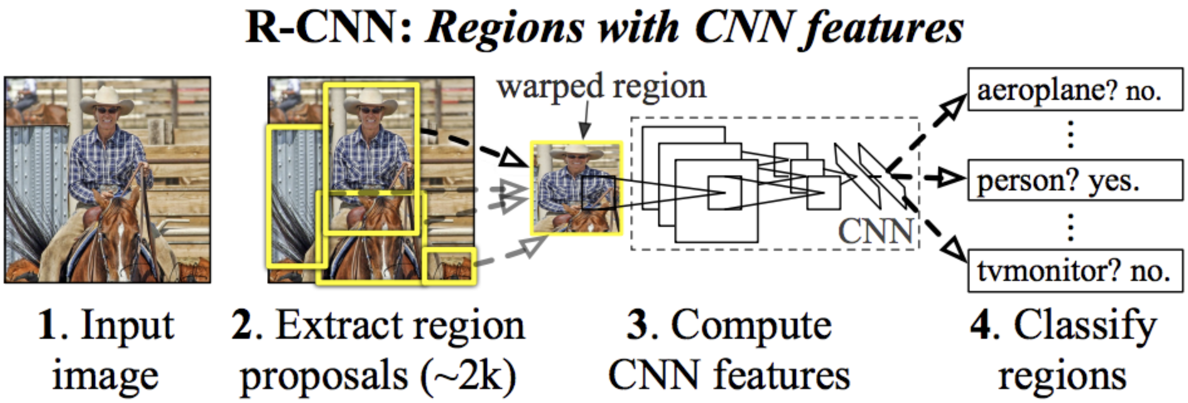

Figure 6: Three modules of an object detector. R-CNN (Regions with CNN features) implements this workflow.

YOLO (You Only Look Once) [5] is a real-time object detector that unifies all these three components into a single model. It divides images into regions, predicts bounding boxes and probabilities for each region, and then labels objects appearing in bounding boxes with confidence higher than a predetermined threshold. This paper uses 0.6 as the confidence threshold.

Comparing with other object detectors such as R-CNN (Regions with CNN features) [2], YOLO claims to be faster without sacrificing too much accuracy. Since the goal of this paper is to build a system that automatically determines video similarity, the implementation and training of a YOLO model is out of the scope of this paper. Instead, this paper adopts an existing Keras implementation (the 3rd version of YOLO), keras-yolo3 [3], on GitHub.

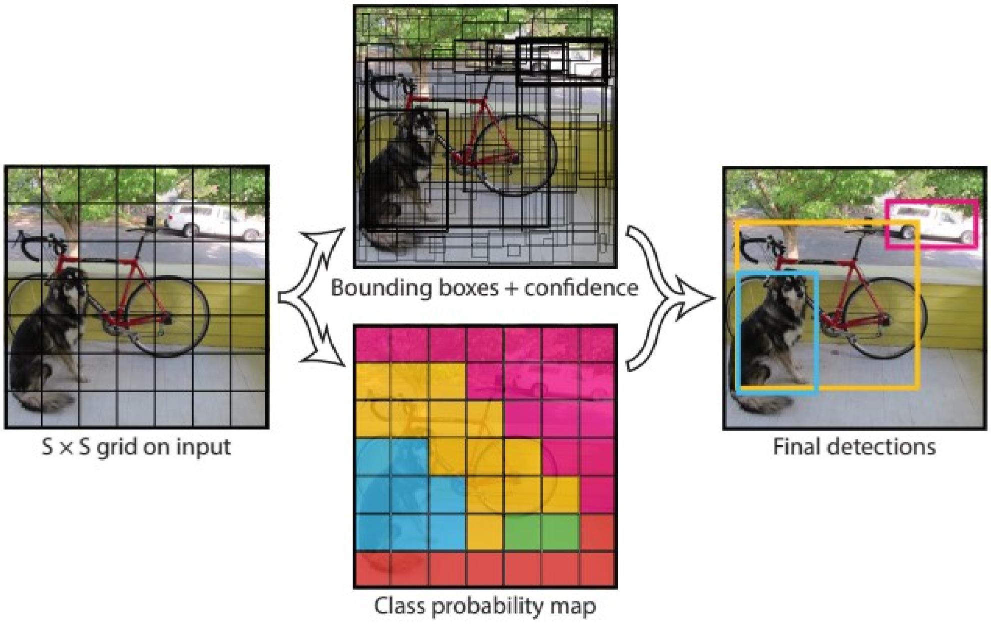

Figure 7: YOLO workflow

YOLO is trained on COCO (Microsoft Common Objects in Context) [4], an image dataset designed for object detection. COCO contains 330K images and 80 categories of daily objects such as people, dogs, and airplanes.

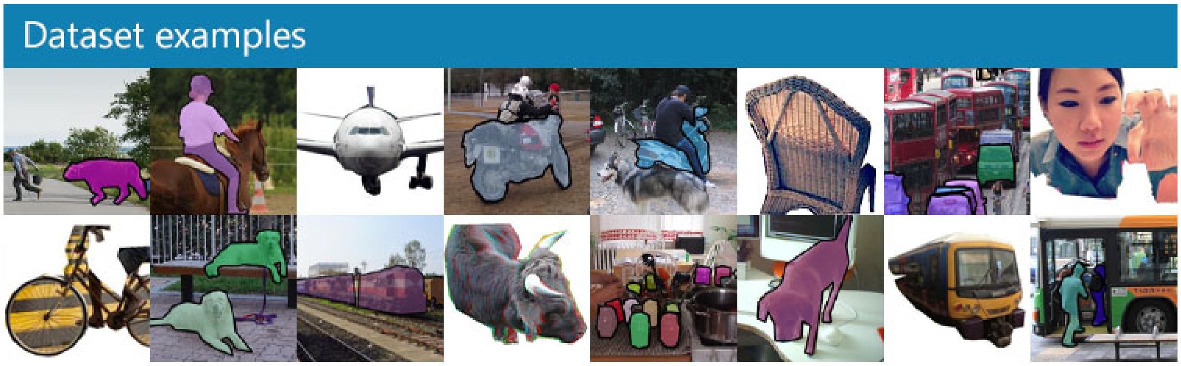

Figure 8: COCO dataset examples

COCO allows YOLO to detect overlapping objects and even infer labels from object parts. For example, YOLO considers a hand or leg as a single person. YOLO does make mistakes when the objects are uncommon. Figures below draw predicted bounding boxes and labels on input frames:

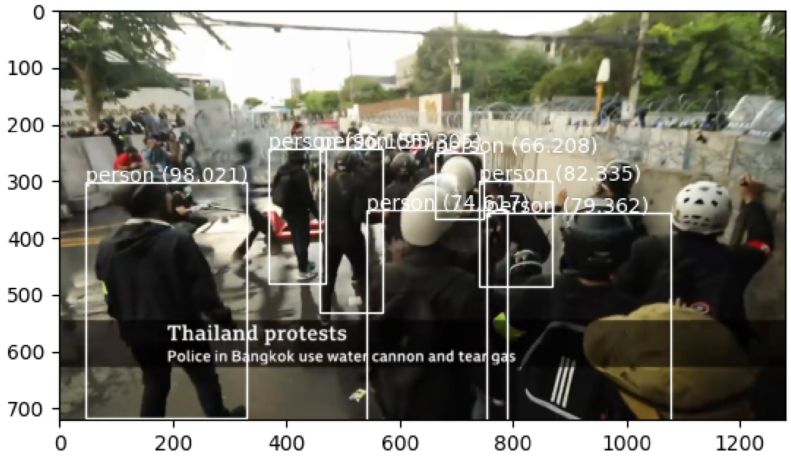

Figure 9: YOLO predictions of overlapping objects

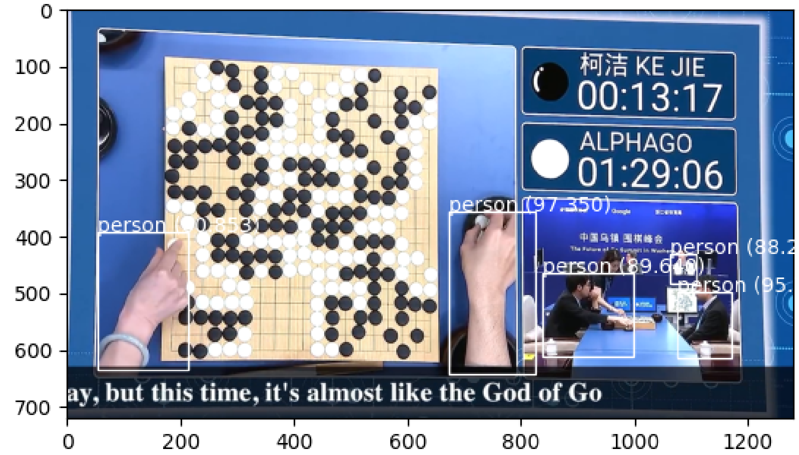

Figure 10: YOLO predictions of object parts

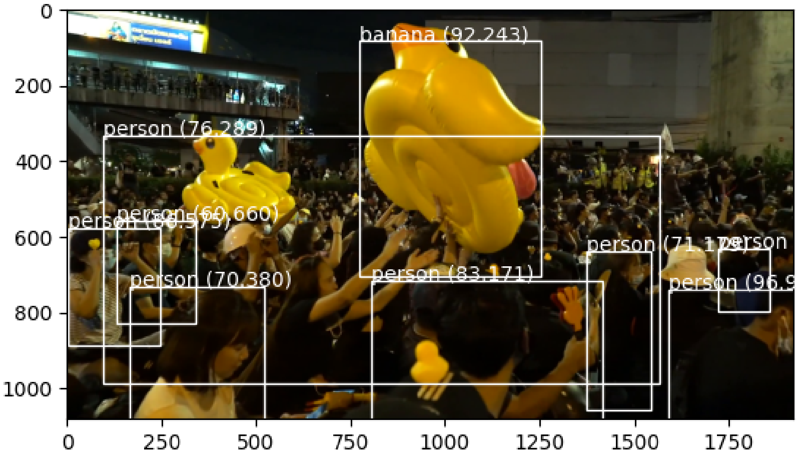

Figure 11: YOLO mistakes on uncommon objects

2.3 Video Segmentation

A video consists of at least one segment. A news video for example may switch between two cameras, one in the studio and the other on the scene. Each switch marks the start or the end of a segment since scenes change drastically.

A key frame is a frame that best summarizes a video segment. All the other frames in the same segment only differ slightly from the key frame. A frame at start and one in the middle may carry the same amount of information. For simplicity, this paper considers the very first frame of each segment as a key frame.

This paper divides videos based on changes in

- object positions
- or object counts

among key frames.

2.3.1 IoU-Based Segmentation

IoU (Intersection Over Union) is a metric that measures the overlap between two objects or specifically their bounding boxes.

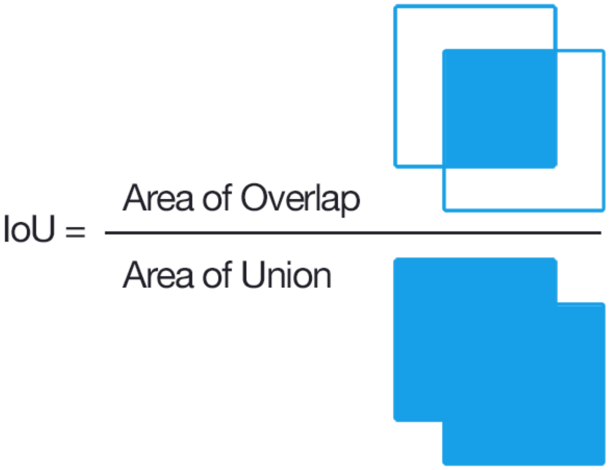

Figure 12: IoU (Intersection Over Union) metric

The IoU-based segmentation goes through two stages to determine the similar score of each frame pair:

1. For each object present in the first frame, find the closest object with the same label existing in the second frame in terms of Euclidean distance.
2. Obtain the two corresponding bounding boxes given by YOLO and compute the IoU score.

The final similarity score is the average IoU score. If this score is lower than a threshold, the two frames belong to different segments. The second frame, or the one with a later timestamp, becomes the key frame of the new segment. This paper uses 0.6 as the threshold.

2.3.2 Frequency-Based Segmentation

The frequency-based segmentation focuses on variations in object counts. Such differences may indicate a change of scene. This paper puts frames into distinct bins in terms of their object counts. Frames in the same bin have the same or similar numbers of a specific object. For example, frames containing three or four people belong to the same bin while two frames having one and two people respectively fall into different bins. The choice of bins will be explained in the experiment section. If two frames appear in different bins, a new segment appears with the second frame, or the one with a later timestamp, as its key frame.

2.3.3 Segmentation Output

Running each segmentation method above will generate a JSON list of key frames sorted in chronological order. An example (output by the IoU-based segmentation) may look like the following:

```
[
    ...,
    7,
    12,
    19,
    ...
]
```

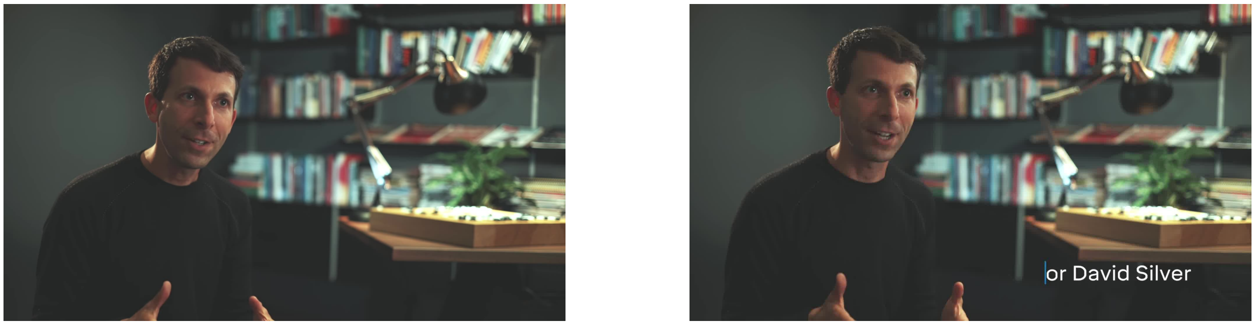

Figure 13: Example segment 1: frame 7 and 11

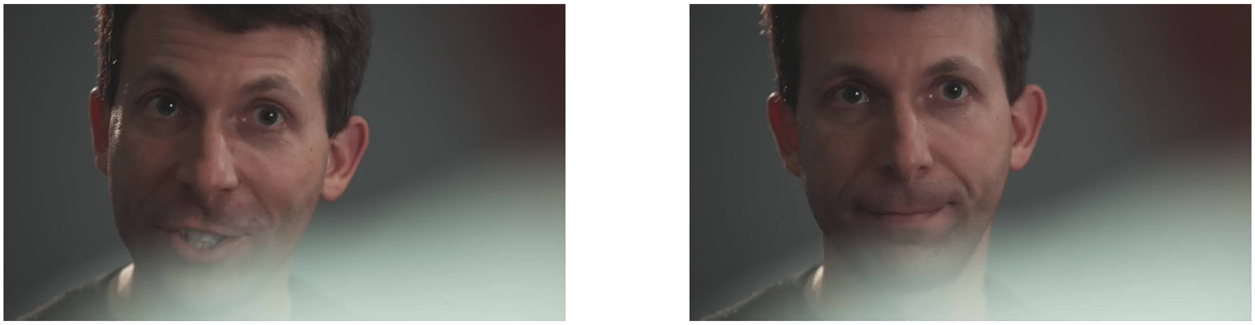

Figure 14: Example segment 2: frame 12 and 18

3 Experiments

3.1 YOLO Performance

Since the adopted YOLO implementation is not official, it is essential to determine its performance in the first place. The official one is written in Darknet that has compatibility issue during experiments described in this paper.

This experiment runs the implementation over the third video set with 8 videos or 1076 frames in total. Time to load the pre-trained weight is excluded. Results show that this implementation takes around 3.5 seconds to process an individual frame, or equivalently 0.27 frame per second. The implementation outputs bounding boxes along with their confidences and labels simultaneously. A 4-minute long video (maximum video length in this paper) will take around 15 minutes. This paper considers segmentation as a preprocessing process, whose results can be reused during the video comparison stage. Hence, the performance is acceptable.

3.2 Most Significant Object

Not all objects are helpful to indicate a scene change. Whether or not there is a bottle on a desk probably does not matter. In addition, focusing on the most significant object(s) reduces the workload during video comparisons.

This experiment uses the first two sets of videos, 50 about AlphaGo and 50 about Chang'e 5. Through data exploitation, the results show that people are the most common objects in both topics. The counts are dominating. All later experiments thus only consider people for simplicity.

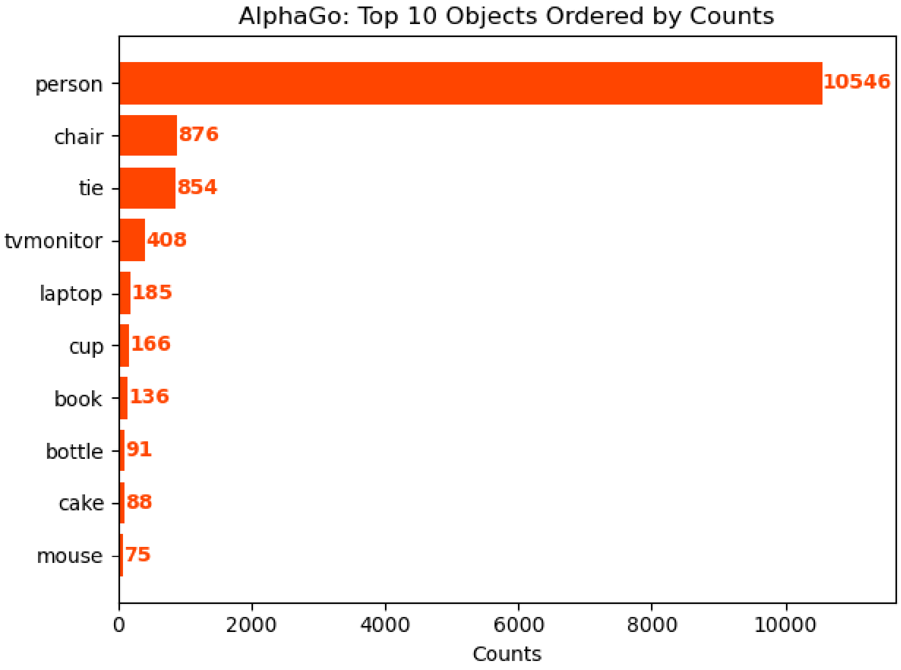

Figure 15: Top 10 most common objects in AlphaGo videos

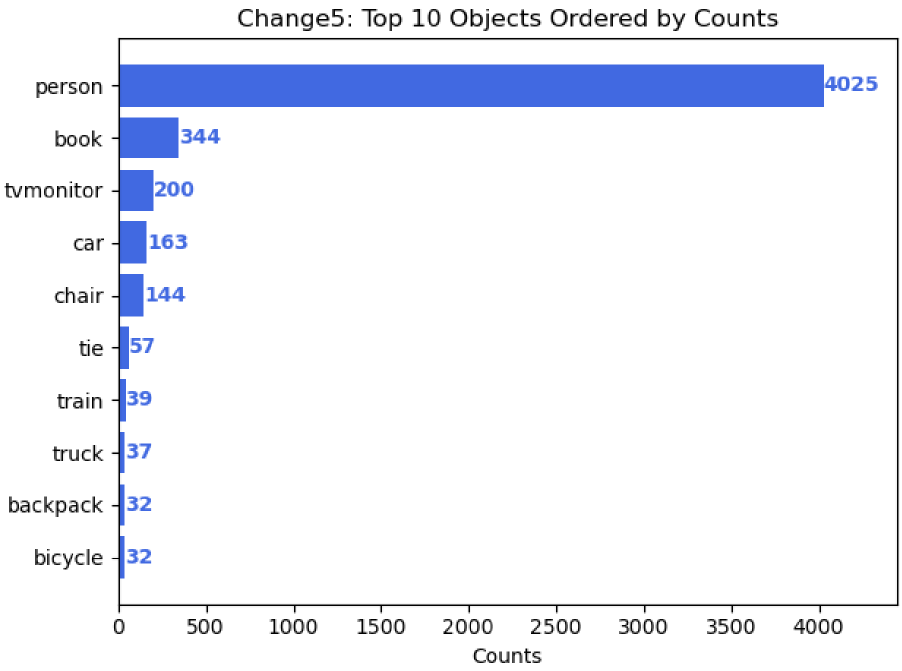

Figure 16: Top 10 most common objects in Chang'e-5 videos

3.3 Bin Selections for Frequency-Based Segmentation

3.3.1 Frequency Distribution of People

This experiment uses the 100 videos from the first two video sets again to generate a frequency distribution for each of the two topics. Both distributions demonstrate the same pattern:

1. a big jump between counts of frames containing zero and one person,
2. frames with a single person dominate both distributions,
3. a noticeable difference between counts of frames containing one and two people,
4. and a long tail for each distribution.

Following this pattern, it is rather clear to separate frames having zero, one, and two people into distinct bins. However, it is hard to group the other frames due to their relatively low frequencies and varying distributions over different topics.

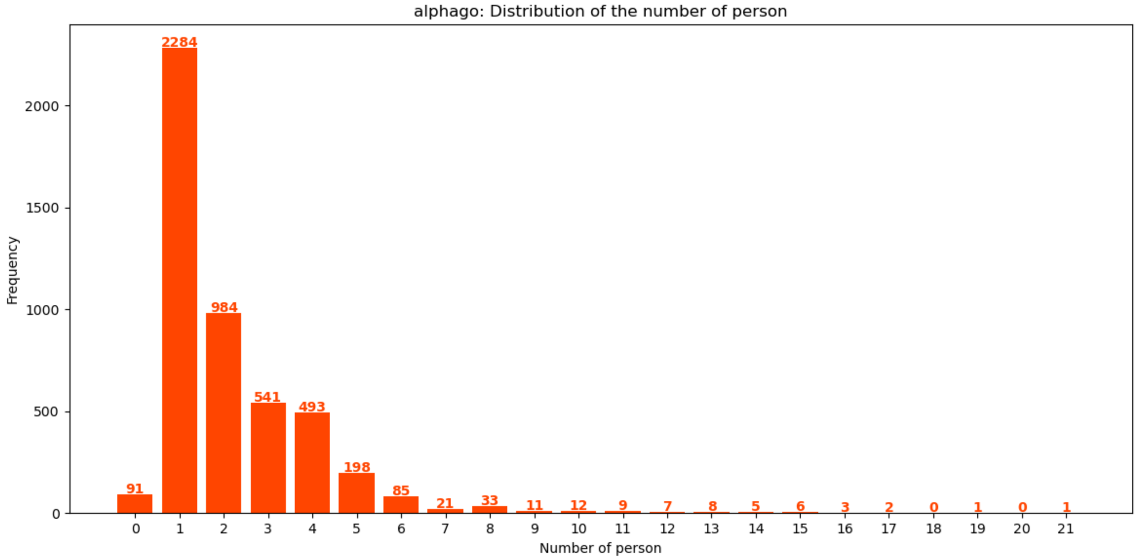

Figure 17: Frequency distribution of people in AlphaGo videos

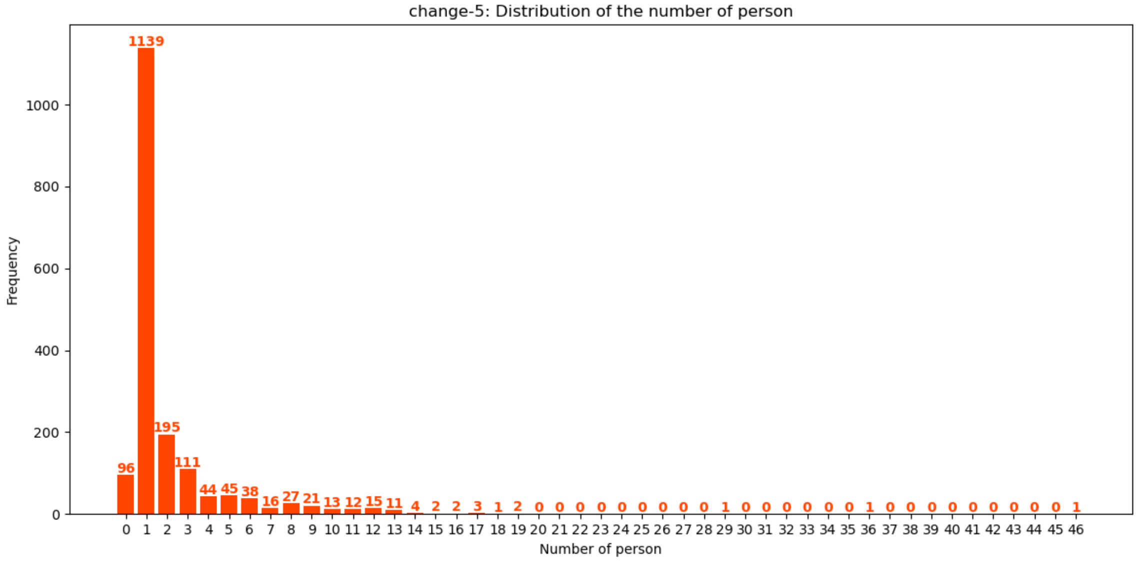

Figure 18: Frequency distribution of people in Chang'e-5 videos

3.3.2 Roles of People

To confirm the bin selections above and determine the rest of the bins, this part of experiment explores 10 of the 100 videos to understand the relationship between the counts and roles of people in frames. Below are some observations:

- 0 person: background scenes and non-human objects describe the overall events and topics

- 1 person: usually appears during an interview or announcement; people are the topic focus
- 2 people: indicate interactions such as conversations; people and their interactions tell the event
- $\geq$ 3 people: people become part of scenes; both background scenes and people present describe the topics

That is, people are primary information sources when there are one or two of them. Background scenes and other objects have higher weightings in story-telling if zero or more than two people appear. The observations above suggest 4 bins, which agree with the previous bin selections.

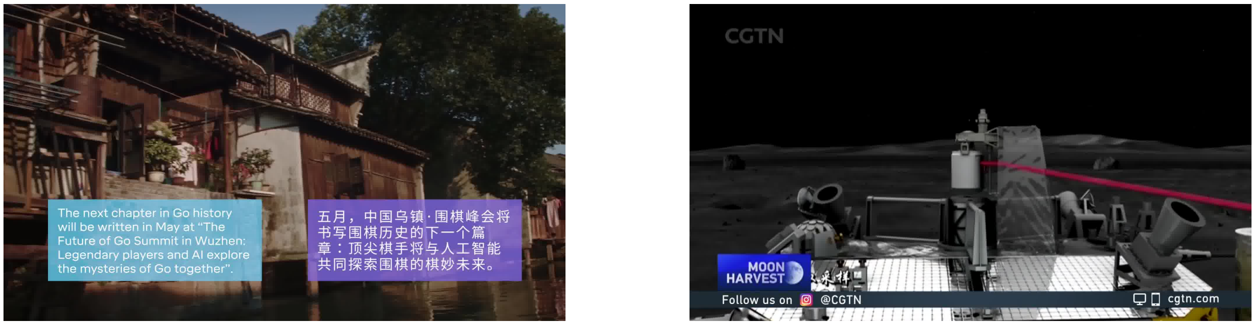
The next chapter in Go history will be written in May at "The Future of Go Summit in Wuzhen: Legendary players and AI explore the mysteries of Go together".
五月，中国乌镇·围棋峰会将书写围棋历史的下一个篇章：顶尖棋手将与人工智能共同探索围棋的棋妙未来。
CGTN
MOON HARVEST
Follow us on @CGTN
cgtn.com

Figure 19: Frames without people

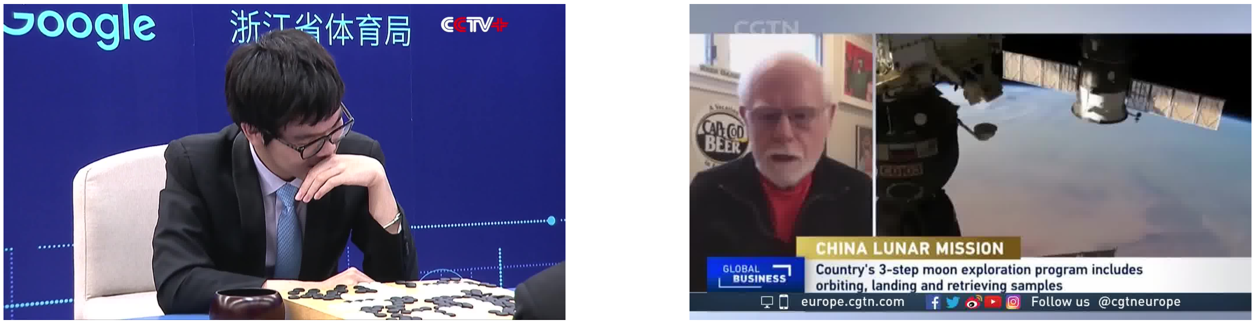

Figure 20: Frames with a single person

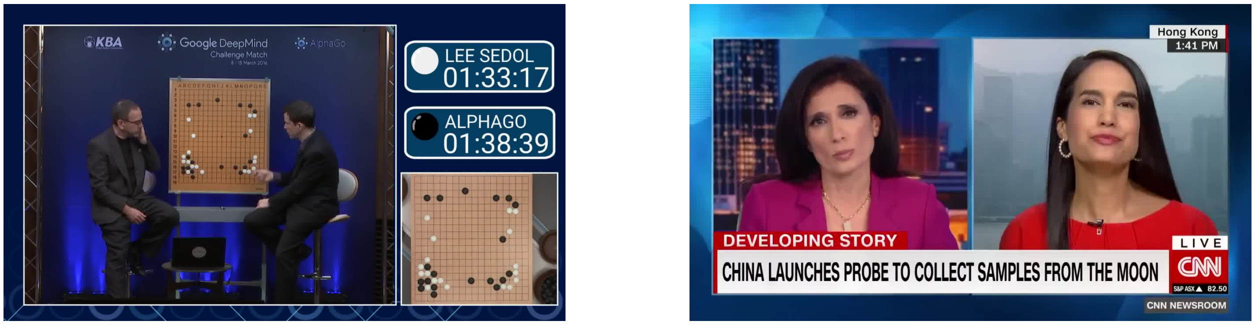

Figure 21: Frames with two people

Figure 22: Frames with three or more people

3.4 Comparison of Segmentation Methods

This experiment aims to determine which of two segmentation methods extracts key frames/video segments with a higher accuracy.

3.4.1 Ground Truth

The ground truth are also JSON lists of key frames. The split choices are arbitrary, usually based on scene or object size changes. 11 AlphaGo videos are manually labelled to perform this experiment.

3.4.2 Evaluation

Dynamic Time Warping (DTW) is a metric measuring the similarity between two temporal sequences [7]. It tries to match the two sequences with minimal cost under four constraints [7]:

1. Each index from a sequence must match with at least one index from the other sequence.
2. The very first indices of both sequences must match with each other.
3. The very last indices of both sequences must match with each other.
4. The mapping of indices from the first sequence to the second must be monotonically increasing.

Since the lists of key frames are sorted in chronological order, they can be seen as temporal sequences, making DTW an ideal evaluation metric.

For the cost function `d(x,y)`, this paper uses a “one-dimensional” IoU function inspired by the standard/two-dimensional one. That is, this cost function considers intersections and overlaps between temporal segments instead of object bounding boxes. A lower cost indicates a better accuracy/performance.

A DTW implementation may look like the following [7]:

```
int DTWDistance(s: array [1..n], t: array [1..m]) {
    DTW := array [0..n, 0..m]

    for i := 0 to n
        for j := 0 to m
            DTW[i, j] := infinity
    DTW[0, 0] := 0

    for i := 1 to n
        for j := 1 to m
            cost := d(s[i], t[j])
            DTW[i, j] := cost + minimum(DTW[i-1, j  ],    // insertion
                                        DTW[i  , j-1],    // deletion
                                        DTW[i-1, j-1])    // match

    return DTW[n, m]
}
```

3.4.3 Results

Video ID	IoU-Based Cost	Frequency-Based Cost	Video Description
1	38.5	26.4	News report and interview
2	54	32.3	Explanation of Go strategies
3	19	1	Explanation of Go strategies
4	33	16.9	News report
5	27.6	19.1	News report
6	28.2	27.6	Documentary and interview
7	105	107	Documentary
8	40	47	Go match
9	30	31	Interview
10	22	30.2	Interview
11	17	35.7	Documentary Trailer

Video 1 to 6 favors the frequency-based segmentation while video 7 to 11 shows the IoU-based segmentation a better method. However, video 8 and 11 are uncommon comparing to the other videos:

- Video 8: The primary camera focuses on hand movements on a Go board, not people as in a typical news videos.
- Video 11: Since this video is a trailer, scenes change rapidly. A scene usually lasts one or two frames. This trait makes video 11 different from standard news videos.

Excluding these two special cases, the results favor the frequency-based segmentation as it performs better in 3 more videos than the IoU-based one.

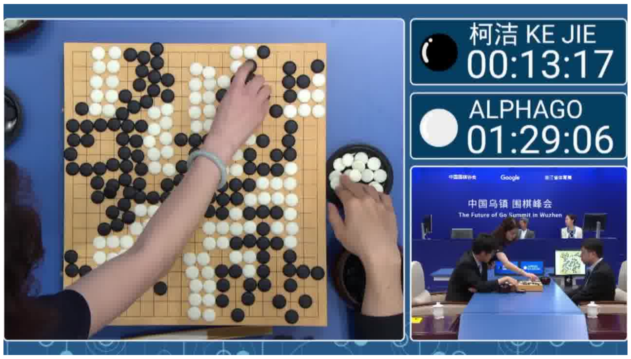

Figure 23: Example frame from video 8.

4 Conclusion

This paper demonstrates a system that automates video segmentation and similarity comparison using object detection. Experiments show that people are the most common objects regardless of topics and the frequency-based segmentation outperforms the IoU-based method. The system can follow exactly the same steps as in the evaluation section to compute similarity scores for different videos. Future work can examine frequency distribution of people or other objects in depth. Distribution variations may provide hints to cultural differences.

All codes are available on GitHub.

References

[1] FFmpeg. *FFmpeg*. URL: `https://www.ffmpeg.org/`.

[2] Ross B. Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *CoRR* abs/1311.2524 (2013). DOI: `https://arxiv.org/pdf/1311.2524.pdf`.

[3] keras-yolo3. *keras-yolo3*. URL: `https://github.com/experiencor/keras-yolo3`.

[4] Microsoft. *COCO*. URL: `https://cocodataset.org/#home`.

[5] Joseph Redmon and Ali Farhadi. "YOLOv3: An Incremental Improvement". In: *CoRR* abs/1804.02767 (2018). DOI: `https://arxiv.org/pdf/1804.02767.pdf`.

[6] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations (ICLR)* (2015). DOI: `https://arxiv.org/pdf/1409.1556.pdf`.

[7] Wikipedia. *Dynamic time warping*. URL: `https://en.wikipedia.org/wiki/Dynamic_time_warping`.

[8] YouTube. *Search:list*. URL: `https://developers.google.com/youtube/v3/docs/search/list`.

[9] youtube-dl. *youtube-dl*. URL: `https://github.com/ytdl-org/youtube-dl`.