1 Video recognition

As introduced in the section of image recognition, recognition tasks could be simplified into a problem of calculating distances between different samples in the evaluated data set. This is because once the distance matrix is calculated, this distance matrix could be input into K-nearest neighbor algorithms or better approaches like Support Vector Machine for recognition. However, it is not so easy to compare two raw videos in quantitative way and thus difficult to calculate distances using raw video data. In order to resolve this problem, a compact representation similar to image for each video clip is needed.

The rest of this section is organized as follows. Firstly, two different representation f video are introduced followed by the distance calculation for each of these two representations. Once the task of calculating distances is finished, four types of kernels used in image recognition are tested. Finally, another method using concept attributes is presented.

1.1 Representations of videos

1.1.1 Bag of words (BoW)

Naive BoW Video consists of a series of consecutive frames and generally presents frames at rates ranging from 20 to 50 frames per second. To be simple, frames sampled from a video every one second could be used to represent this video. For instance, a two-minute video is then represented by 120 frames. With the same representation of image illustrated in section of image recognition, these 120 frames are then converted into 120 histograms. At last, these 120 histograms are stacked to represent this two-minute video.

A formal description of this approach to build bag of words model for videos go as below.

1. **Build vocabulary**
   Similar to image recognition, a vocabulary is needed to represent each frame of videos as a histogram. The vocabulary is built by applying Mini-Batch K-Means [3] clustering algorithm on SIFT features of sampled frames for all available videos. In the experiments, two vocabularies were built with the numbers of centroids being 2500 and 1000.

2. **Represent each video as a stack of histograms**

Let's say the number of sampled frame in a video is M, and each video is represented as a $1 \times V$ histogram, where V is the size of vocabulary. Then histograms of these frames from this video are stacked together to form a $M \times V$ matrix, and this matrix represents this video.

Following the above steps, videos are converted into a compact matrix with each row representing a frame. However, different videos are converted into matrices with different rows because of different durations. Such differences make it not easy to use general distance calculation formula to calculate video-to-video distance. The solution to this problem will be presented in next section about distance calculation.

Better BoW In naive Bow, SIFT feature are assigned to its nearest word, and the respective bin in histogram is increased by one. Two questions arise from this statement. Why SIFT feature can only be assigned to one word but not multiple words? Why the respective bin is increased by one but not some other values? To answer these two problems, soft assignment and different weighting schemes are introduced.

- Soft assignments

  Soft assignment allows that a visual feature could be assigned to multiple words rather than only one word. In doing so, more valuable information is retained during quantization process and thus might provide more discriminative power. An straightforward approach [2] is that the top N nearest words are selected for each visual feature. Let's say the size of vocabulary is K, and thus a K-dimensional vector $T = [t_1, t_2, ..., t_K]$ is used to represent an image. The algorithm to construct this vector goes as below.

  The above algorithm selects the top N nearest neighbors. What if N equals to the size of vocabulary? Here, instead of the original algorithm, Agaral and Triggs [1] proposed to use Gaussian mixture model (GMM) built from training data to perform assignment. Let's say the number of components in Gaussian mixture model is K. For each visual feature, GMM produces a K-dimensional vector representing posterior mixture-component membership probabilities. Finally, all these K-dimensional vectors are summed up to produce one final K-dimensional vector to represent the respective image.

- Weighting schemes

Algorithm 1 Build histogram with soft assignment

```
Given: vocabulary size $K$, words in vocabulary $[w_1, w2, ..., w_k]$, visual features $F$, parameter $N$
Initialize a $K$-dimensional vector $T = [t_1, t_2, ..., t_K]$ with all components $t_i = 0$

for $f \in F$ do
    Retrieve the top $N$ nearest words to $f$
    Put the indexes of the top N words in $W_N$ in sorted distances order
    $v \leftarrow 0$
    for $index \in W_N$ do
        $t_{index} \leftarrow t_{index} + \frac{1}{2^v}$
        $v \leftarrow v + 1$
    end for
end for

return $T$
```

1.1.2 Gaussian mixture models

1.2 Distance calculations

1.2.1 Aligned space-time pyramid matching

1.2.2 Distances between Gaussian mixture models

1.3 Other approach: concept attributes

References

[1] A. Agarwal and B. Triggs, “Hyperfeatures–multilevel local coding for visual recognition,” in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 30–43.

[2] Y.-G. Jiang, C.-W. Ngo, and J. Yang, “Towards optimal bag-of-features for object categorization and semantic video retrieval,” in *Proceedings of the 6th ACM international conference on Image and video retrieval*. ACM, 2007, pp. 494–501.

[3] D. Sculley, “Web-scale k-means clustering,” in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 1177–1178.