A Similarity Alignment Model for Video Copy Segment Matching

Zhenhua Liu[1]*, Feipeng Ma[1,2]*, Tianyi Wang[1]*, Fengyun Rao[1]†
[1]WeChat of Tencent, [2]University of Science and Technology of China
mafp@mail.ustc.edu.cn, {edinliu, tyewang, fengyunrao}@tencent.com

Abstract

With the development of multimedia technology, Video Copy Detection has been a crucial problem for social media platforms. Meta AI hold Video Similarity Challenge on CVPR 2023 to push the technology forward. In this report, we share our winner solutions on Matching Track. We propose a Similarity Alignment Model(SAM) for video copy segment matching. Our SAM exhibits superior performance compared to other competitors, with a 0.108 / 0.144 absolute improvement over the second-place competitor in Phase 1 / Phase 2. Code is available at https://github.com/FeipengMa6/VSC22-Submission/tree/main/VSC22-Matching-Track-1st.

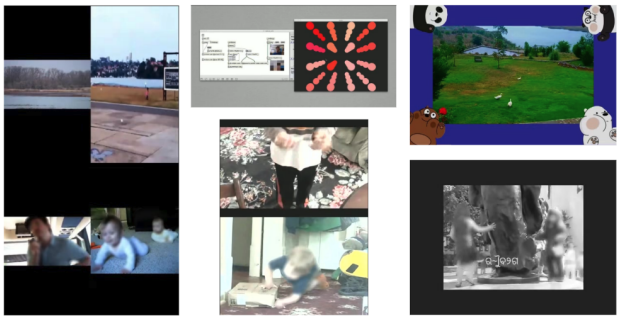

Figure 1. Typical edits in stacked frames and extra edges. A plenty of video contains 2-4 scenes, as showed in the first two columns. And there is also extra edges showed in the last column.

1. Introduction

In the past decade, the development of information technology has led to a shift in the main carrier of information from text to images and then to videos. Moreover, with the rise of User-generated Content (UGC), the producer of information has shifted from Occupationally-generated Content (OGC) to UGC. As a result, a large number of videos have emerged on social media platforms and have been widely shared, leading to the increasingly important and challenging problems of video copyright protection. The video copy detection task can always be divided into two parts: the descriptor task, which is used to recall similar videos, and the matching task, which is used to locate the copied segment. In this report, we summarize our work on the Matching Track of the Meta AI Video Similarity Challenge.

For the matching task, two important problems arise. The first is deciding which feature to use: the embedding or the similarity matrix. The second is determining how to match the video copy segments. We first consider the feature. It is reasonable choice to share features among the matching task and the descriptor task, as two task are highly correlated. Extracting features independently for each task would require double the computing resources, so sharing features can reduce the computation cost. As to the model input, the advantage of using an embedding for the matching task is that it contains more information and can be used for further tuning [3] . However, the drawback is that the matching model must be changed when the descriptor model is changed, as two different embedding models may have little correlation with each other. While the similarity matrix is more robust as changes of embedding model does not change the characteristic of similarity matrix, and even the use of different embedding models can expand the limited annotations. Finally, we choose the similarity matrix as model input.

As to the video copy segments, we found several academic approaches. First is the Temporal Network(TN) [7], a graph-based method takes matched frames as nodes and similarities between frames as weights of links to construct a network. This is also the baseline, but we found this method is hard to modify and optimize. Similarity Pattern Detection (SPD) [5] adopts detection method to direct output the result. We attempted to use this method, but encountered difficulties in optimizing the model with limited annotations. We also explored TransVCL [3], but ultimately decided to against it. Because this method relies on frame embedding as input, which does not align with our project's objectives. Most importantly, we discovered that the pri-

*Equal contribution.
†Corresponding author.

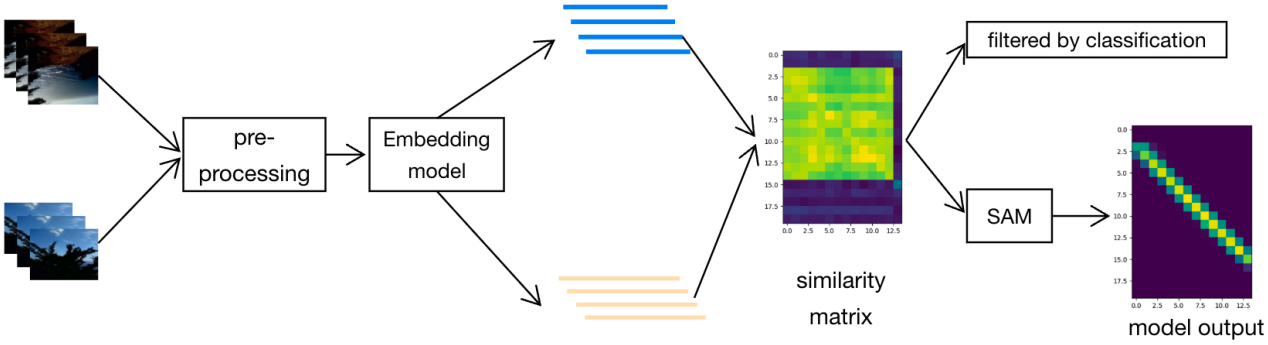

Figure 2. Our Pipeline: Preprocess input video by extracting frames, splitting scenes, and removing edges. Use embedding model(s) to generate embeddings. Generate similarity matrix for (query, reference) videos, filter negative recalls using a small classification model. Use SAM model to output noiseless matching score matrix.

mary challenge here is not simply outputting the results in an end-to-end way, but rather obtaining a cleaner matching relationship in comparison to the raw similarity matrix input. To address this, our SAM was specifically designed to take a similarity matrix as input and output a score matrix with the same resolution, with significantly improved matching relationships.

2. Method

In this section, we will introduce the whole pipeline we used to develop our result. As shown in figure 2. Our pipeline include the preprocessing, embedding extraction, similarity matrix filtering, and the SAM model processing.

2.1. Preprocessing

Video frames were extracted at one frame per second, but many videos contained multiple scenes in one frame or extra edges. Canny [1] edge detection and frame pixel standard deviation feature was used to address this issue. First, we average the edge detection results from multiple frames to get more robust edges. Then we use frame pixel standard deviation feature to find potential background. The whole processing is done recursively by: 1) A split images function divides a video into segments based on: a) the vertical or horizontal edges that extend across the frame; b) a low pixel standard deviation zone that split video vertically or horizontally. 2) A edge erase function that remove low variant parts of videos. With a video input, it stops when the processing parts doesn't change in size or has too low resolution. Figure 3 shows the feature we used in preprocessing. Figure 1. reveals the typical edits in stacked frames and extra edges.

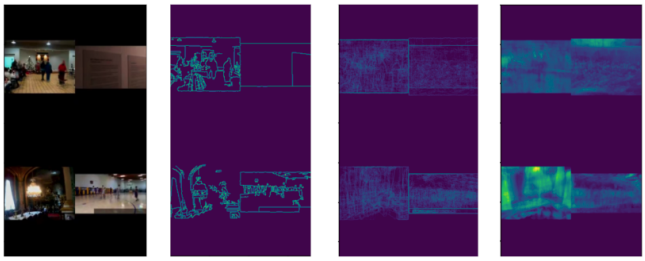

Figure 3. The frame processing details. a is one frame of a query video. b is the Canny [1] edge result, which is noisy to locate all edges of the stacked video. c is the average Canny result of multiple frames, the edge is more clear than single frame. d is standard deviation of frame pixels values. Our frame preprocessing is base on feature c and d

2.2. Embedding Model

As previously mentioned, our method utilizes an embedding model from the Descriptor track, using a similarity matrix as the embedding. This approach allows our model to be resilient to changes in the embedding, and multiple embeddings of the same (query, reference) pair can be utilized as a form of data augmentation.

To maximize efficiency and recall, a recall process is employed to identify potential copied video pairs. This process only considers the similarity of descriptor features and utilizes a low recall threshold, resulting in a high number of potential matches being identified but is not copied at all. We utilized a classification process that takes a similarity matrix as input and outputs the probability that a video pair is duplicated. Our chosen classification model is the MobileNet-v3 [4], pre-trained by Image-net. This approach successfully removed 95% of the recalled samples without

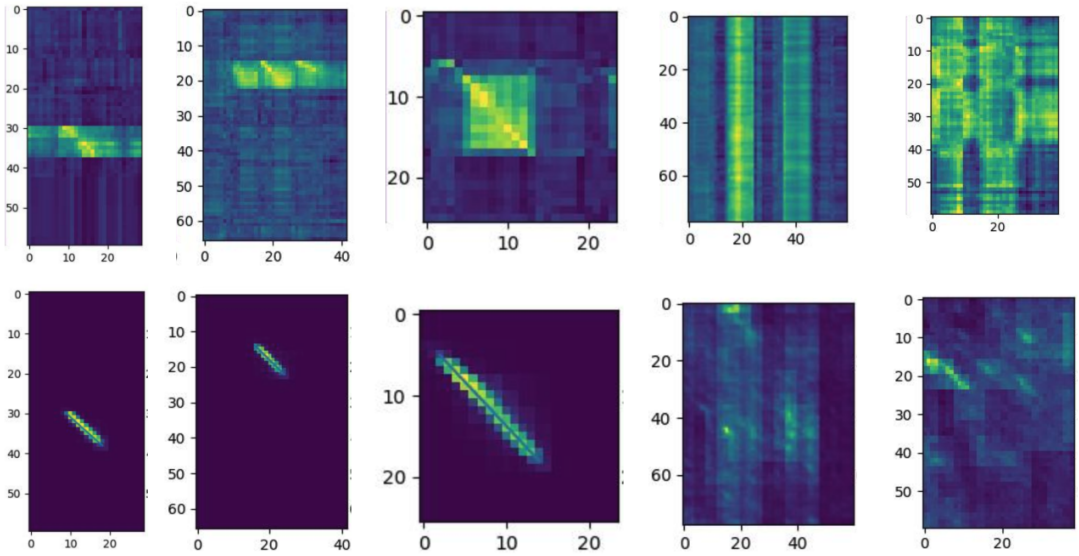

Figure 4. The input similarity matrix is compared with the SAM output matching score. The first row is the input similarity matrix, the second row is the SAM matching score. The first three column are sample successfully matched, the last two column are samples that not recognize copy result.

impacting overall performance.

2.3. Similarity Alignment Model

We choose the similarity matrix as model input. The query/reference longer than 128 seconds is truncated, while the query/reference video shorter than 128 are padding to 128 with zero embedding. So our SAM model takes (128, 128) resolution similarity matrix as input. For query videos that has been splited into multiple frames in preprocessing. We choose the segment with max top matching similarity as target pair. Other segments with lower matching similarity are discarded.

To build a model that output matching relationship, one possible approach is to use models such as key point detection or semantic segmentation. The key idea here is that the model should both learn global information which helps to recognize the real matching parts and the local information for percise detection. So we choose the high resolution network(HRnet-w18) [6] as our backbone. There are two major changes to the model: 1) Our model outputs the same resolution feature maps as the input. We do it by setting the first two convolution stride to (1,1). 2) The target output has been changed to a heat-map generated by annotations to accurately reflect the real matching relationship. Figure 4. shows some model detection result and the post-processing result.

2.4. Postprocessing

The SAM model outputs a matching relationship matrix, but post-processing is required before submitting the final result. This involves: 1) using a filter threshold to remove false positive matches, 2) identifying multiple detections with the Connected Components algorithm, 3) and detecting the matching relation with RANSAC [2] regression, which is effective for detecting linear relationships in video copies. The final submit score s is an ensemble of SAM score predictions:

$c = max(1/coef, coef)$

$s = mean(score) - \alpha * std(score) - abs(c-1)/10$

Where $coef$ is the slope of RANSAC regression. $score$ is the matched score list for points which is first filter by score threshold t, then filter by RANSAC regression inner points. α is the weights for score variance penalty. For the final submission, we ensemble results with parameter (t, α) equal to (0.35, 0.5), (0.1, 1.25), (0.001, 2).

3. Experiments

The SAM model was trained using 1 A100 GPU. After applying the classification filter, there were 10,000 pairs of samples, and their labels were generated through annotations. The resolution of the similarity matrix used for training the SAM model is 128x128, and the batch size is 64. As mentioned earlier, four embedding models from the descriptor track were used to generate the similarity matrix. Therefore, the training sample size for each round is approximately 20,000 pairs. It takes around 3 hours to finished

training.

To generate the final submission, the two cross-validation models were ensembled by averaging their predicted scores. As to different embedding model, the SAM is directly evaluated on their PCA ensembled feature similarity matrix. On Matching Track,we got the first place on both phases with a 0.108 μAP / 0.144 μAP absolute improvement over the second-place competitor in Phase 1 / Phase 2.

User or teams	Phase 1 μAP	Phase 2 μAP
do something more(Ours)	**0.9290**	**0.9153**
CompetitionSecond	0.8206	0.7711
cvl-matching	0.7727	0.7036
Zihao	0.5687	-

Table 1. Leaderboard results on Matching Track.

4. Conclusion

This report introduces the SAM for video copy segment matching. By modify the structure and target of high resolution network, Our model takes similarity matching matrix as input, output high quality video segment matching score. Based on this model, We get 1^{st} rank on VSC 2022 Matching Track.

References

[1] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 2

[2] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 3

[3] Sifeng He, Yue He, Minlong Lu, Chen Jiang, Xudong Yang, Feng Qian, Xiaobo Zhang, Lei Yang, and Jiandong Zhang. Transvcl: Attention-enhanced video copy localization network with flexible supervision. *arXiv preprint arXiv:2211.13090*, 2022. 1

[4] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019. 2

[5] Chen Jiang, Kaiming Huang, Sifeng He, Xudong Yang, Wei Zhang, Xiaobo Zhang, Yuan Cheng, Lei Yang, Qing Wang, Furong Xu, et al. Learning segment similarity and alignment in large-scale content based video retrieval. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1618–1626, 2021. 1

[6] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 3

[7] Hung-Khoon Tan, Chong-Wah Ngo, Richard Hong, and Tat-Seng Chua. Scalable detection of partial near-duplicate videos by visual-temporal consistency. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 145–154, 2009. 1