Exploiting Feature Diversity for Make-up Temporal Video Grounding

Xiujun Shu∗
Tencent Youtu Lab
Shenzhen, China
xiujunshu@tencent.com

Wei Wen∗
Tencent Youtu Lab
Shenzhen, China
jawnrwen@tencent.com

Taian Guo∗
Tencent Youtu Lab
Shenzhen, China
taianguo@tencent.com

Sunan He
Tencent Youtu Lab
Shenzhen, China
sunanhe@tencent.com

Chen Wu
Tencent Youtu Lab
Shenzhen, China
chewu@tencent.com

Ruizhi Qiao
Tencent Youtu Lab
Shenzhen, China
ruizhiqiao@tencent.com

ABSTRACT
This technical report presents the 3rd winning solution for MTVG, a new task introduced in the 4th Person in Context (PIC) Challenge at ACM MM 2022. MTVG aims at localizing the temporal boundary of the step in an untrimmed video based on a textual description. The biggest challenge of this task is the fine-grained video-text semantics of make-up steps. However, current methods mainly extract video features using action-based pre-trained models. As actions are more coarse-grained than make-up steps, action-based features are not sufficient to provide fine-grained cues. To address this issue, we propose to achieve fine-grained representation via exploiting feature diversities. Specifically, we proposed a series of methods from feature extraction, and network optimization, to the model ensemble. As a result, we achieved 3rd place in the MTVG competition.

CCS CONCEPTS
• Computer systems organization → Embedded systems; Redundancy; Robotics; • Networks → Network reliability.

KEYWORDS
Temporal Grounding

ACM Reference Format:
Xiujun Shu, Wei Wen, Taian Guo, Sunan He, Chen Wu, and Ruizhi Qiao. 2022. Exploiting Feature Diversity for Make-up Temporal Video Grounding. In Proceedings of the 4th Person in Context Workshop (PIC ’22), October 14, 2022, Lisboa, Portugal. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3552455.3555818

∗The authors contributed equally to this research.
with 24 types in the ground truth moments descriptions as well.

2.1.1 Video-based Features. In this part, we focus on extracting video features that are more adapted for make-up scenes, we train video pre-training. To obtain the benefits of self-supervised learning on benchmarks like action recognition, the extracted video features can greatly facilitate downstream tasks.

2.1.2 MAE-based Features. Recently, self-supervised visual pre-training methods [4] [2] emerge, showing superior performance over supervised pre-training models. Among them, Masked Autoencoders (MAE) [9] demonstrate superior learning ability and scalability. Specifically, motivated by the autoencoding paradigm in BERT [6] in NLP, MAE adopts an asymmetric encoder-decoder architecture with visible patches encoding in the encoder and masked token reconstruction in the decoder. VideoMAE [13] further extends MAE to video and shows data-efficient learners for self-supervised video pre-training. To obtain the benefits of self-supervised learning in video classification, we utilize VideoMAE pre-trained in Something-Something [8] to extract video features inherent in make-up scenes.

2.1.3 Make-up Classification Features. As the existing pre-trained models, e.g., VideoSwin [11] and VideoMAE [13], are pre-trained on benchmarks like action recognition, the extracted video features are inevitably more suitable to general-purpose action scenes and not optimal for the specialized-purpose make-up scenes. To obtain video features that are more adapted for make-up scenes, we train a make-up classification model based on the descriptions provided on the YouMakeup dataset. Specifically, we take the “area” field with 24 types in the ground truth moments descriptions as well as “null” as the classification categories. We optimize a VideoSwinBase [11] model pre-trained in Something-Something [8] to solve the 25-class make-up area classification problem. We speculate that although the classes for each moment are not manually annotated, the optimization objective makes the model more applicable to make-up scenes, leading to better features for temporal grounding networks. Experiments demonstrate the advantages of make-up classification features.

2.1.4 CLIP-based Features. CLIP [12] is an efficient and scalable contrastive pre-training architecture, which predicts image-text correspondence in large-scale image-text pair datasets. Concretely, by learning on 400 million image-text pairs crawled from the Internet from scratch, CLIP learns state-of-the-art image representations, leading to amazing transfer learning capabilities on various downstream tasks. To take advantage of the abundant prior knowledge inherent in the CLIP model pre-trained with extensive natural image-text data, we extract frame features in make-up videos with CLIP and average the frame-level features within each second to obtain make-up video features.

2.2 Augmented Mutual Matching Network

In our approach, MMN [16] is regarded as the baseline method. Several methods are further proposed to optimize it.

2.2.1 Multi-scale Temporal Sensing. 2D-Map anchor-based method was first proposed in 2D-TAN [19], and then was further optimized [16, 18]. They model the temporal relations between video moments by a two-dimensional map, which indicates the starting time and end time of moments. However, 2D-Map splits the videos into a fixed number of clips, e.g., 128, which results in poor temporal resolution for long videos.

To address this issue, we propose an augmentation strategy named multi-scale temporal sensing. As shown in Fig. 1, the long video contains several textual descriptions. Each of them corresponds to specific video moments. During training, we randomly cut the video and get a clip that contains one or several complete textual descriptions. It should be noted that the start and end cutting points should be outside the textual moments to make them complete. Assume the video lasts 50 minutes and the clip length is only 5 minutes. As MMN firstly splits them into 128 pieces and each of them is pooled to get a feature. For the initial video, the temporal resolution is 23.4s (50/23.4s), but the temporal resolution of the cut clip is 2.34s (5/2.34s). It means the shorter the video is, the higher the temporal resolution is. By randomly clipping, the model has a stronger ability for modeling length-variable videos.

2.2.2 Direct Feature Concatenation. The extracted features are not used individually, but a combination of them can achieve better performance. A simple method is the direct feature concatenation, which is described as follows:

$$v = \text{Avg} (\text{Norm} (\text{Concat}(v_1, v_2, ..., v_k))) \quad (1)$$

where $v_k$ is the $k$th extracted feature of the whole video. “Concat” denotes the feature concatenation and “Norm” means the $L_2$ normalization. “Avg” denotes that all the features are split into 128 pieces and each of them is averaged.
Table 1: Comparison of features on YouMakeUp dataset.

| ID | Feature       | R1@0.3 | R1@0.5 | R1@0.7 | AVG  |
|----|---------------|--------|--------|--------|------|
| A  | C3D           | 30.63  | 20.84  | 18.04  | 20.50|
| B  | I3D           | 47.71  | 33.31  | 17.49  | 32.84|
| C  | VideoSwin-K400| 51.53  | 39.25  | 22.26  | 37.68|
| D  | VideoSwin-K600| 56.77  | 42.50  | 23.40  | 40.89|
| E  | VideoSwin-SS  | 51.09  | 38.27  | 20.81  | 36.72|
| F  | CLIP          | 51.85  | 38.49  | 21.85  | 37.40|
| G  | VideoMAE      | 35.49  | 24.25  | 11.43  | 23.72|
| H  | SwinMC        | 56.80  | 43.13  | 24.82  | 41.58|

Table 2: Experiments of feature combination. AUG: multi-scale temporal sensing. MC: Make-up classification features. FT: Fine-tune on the best model with A->H. FT*: Fine-tune using the learnable feature weighting.

| Feature       | R1@0.3 | R1@0.5 | R1@0.7 | AVG  |
|---------------|--------|--------|--------|------|
| C+E           | 60.15  | 37.02  | 27.98  | 45.05|
| C+D+E         | 61.70  | 48.15  | 29.30  | 46.38|
| B+C+D+E       | 60.06  | 47.14  | 28.42  | 45.21|
| D+E+F+G       | 60.15  | 45.97  | 27.34  | 44.49|
| B+C+D+E+AUG   | 61.35  | 48.06  | 28.83  | 46.08|
| B+C+D+E+F+G+H | 63.66  | 51.31  | 32.90  | 49.29|
| B+C+D+E+F+G+FT| 63.91  | 50.43  | 32.18  | 48.84|
| B+C+D+E+F+G+FT*| 64.22 | 51.50  | 33.03  | 49.58|

Before generating the 2D-Map, the feature \( v \) is firstly mapped into a small dimension with a fully connected layer \( f \).

\[
\hat{v} = f(v). \tag{2}
\]

By reducing the dimension of concatenated features to a fixed value, e.g., 1024, the MMN structure can remain unchanged. Although the feature concatenation is simple, its performance is quite satisfactory.

2.2.3 Learnable Feature Weighting. Besides the concatenation, we also propose a feature weighting method in which the weighting vector is learnable.

\[
v_k = \text{Avg}(\text{Norm}(v_k)), k \in [1, 2, 3, ..., K]. \tag{3}
\]

All the K features are independently mapped into the same dimension, e.g., 512, then they are weighted and summed.

\[
\hat{v} = \gamma \cdot \sum_{k=1}^{K} \hat{w}_k \cdot v_k, \tag{4}
\]

\[
\hat{w}_k = \frac{\exp(w_k)}{\sum_{k=1}^{K} \exp(w_k)}, \tag{5}
\]

where \( w_k \) is the \( k^{th} \) value of weight vector \( w = [w_1, w_2, ..., w_K] \). \( \gamma \) is a learnable weighting parameter.

The feature concatenation and weighting operations are both utilized in our approach. They could train different models and both achieve satisfying performance.

2.3 Model Ensemble

Based on the multiple powerful features and combination forms described above, we trained several different models with complementary information to improve the final results. Here we integrate them in two ways, i.e., intra-framework fusion and inter-framework fusion.

2.3.1 Intra-framework Fusion. We use the same framework to train different models, in which feature combinations and parameters are different. In this case, we can do the aggregation of 2D score map inside the framework. For example, the baseline method MMN [16] combines the contrastive map and iou map to obtain the final 2D score map, which is the key to getting the final forecast moment.

\[
S = (S_{\text{cons}} + 0.5 + \alpha)\beta + S_{\text{iou}}. \tag{6}
\]

Where \( S_{\text{cons}} \) and \( S_{\text{iou}} \) denote the contrastive map and iou map. \( S \) denote the final 2D score map of the current model, \( \alpha = 0.5 \) and \( \beta = 0.5 \) are small parameters.

To fuse the predicted results, We aggregate 2D score maps from different models. The 2D score maps are complementary and contribute to more accurate moments.

\[
S_{\text{combine}} = \sum_{k=1}^{n} S_k. \tag{7}
\]

Where \( n \) denotes the number of models, \( S_{\text{combine}} \) denotes aggregated 2D score map.

2.3.2 Inter-framework Fusion. For the models from different frameworks, score map sizes and types are heterogeneous and cannot be aggregated directly. We first save the candidates and scores of different models and then combine them to perform the NMS operation. It should be noted that scores of different frameworks need to be projected to the same magnitude before the NMS operation. This fusion strategy is useful because of the differences in granularity and information learned.

3 EXPERIMENTS

In this section, all the experiments have been conducted on the YouMakeup dataset, which contains 1680 videos for training, 280 videos for validation, and 420 videos for testing. As the labels of the test set are not available, we report the validation results on Rank1 accuracy with three IoUs@{0.3,0.5,0.7}. The average value is also reported.

3.1 Comparison of Diverse Features

In Table 1, we have evaluated several features extracted from different pre-trained models. On average, one feature is extracted every 1 second. Table 1 shows us that the performance of different features varies greatly. C3D behaves the worst and only achieves 20.50% on average Rank1. VideoSwin [11] behaves much better, especially the feature pre-trained on K600 achieves 40.89% on average Rank1. CLIP [12] can also achieve satisfying performance because it has been pre-trained on large-scale image-text datasets. SwinMC denotes the make-up classification feature. It achieves the best result and the average Rank1 accuracy attains 41.58%. This demonstrates the effectiveness of make-up classification. To increase diversity, all the features are utilized in our approach, but not independently.
### 3.2 Experiments of Feature Combination

Based on the analysis above, we combine several features other than using them independently. In Table 2, A→H refer to the features in Table 1. We can see that several concatenated features can achieve much better performance than single features. For example, “C+D+E” achieves 46.38% on average Rank1, which is much better than VideoSwin-K400(37.68%), VideoSwin-K400(40.89%), and VideoSwin-SS(36.72%). This is because different features bring different cues that are valuable for the MTVG task.

We also evaluate the multi-scale temporal sensing strategy, which is short of “AUG” in Table 2. It improves the R1@0.3 from 60.06% to 61.35%. The make-up classification feature, i.e., SwinMC, can also improve the performance significantly. It increases the R1@0.3 from 60.85% to 63.66%. In addition, fine-tuning is also useful. Specifically, we initialize the model’s parameters with the previous best model, e.g., “B+C+D+E+F+G”. Then we add more features, e.g., B+C+D+E+F+G+H, and fine-tune the model. In general, the fine-tuning process is quite fast and the model could achieve better performance, see FT/FT* in Table 2.

### 3.3 Fusion Results

In our approach, we fuse the results of 15 models. All models contain different complementary cues, we integrate their results as described in Section 2.3. As shown in Table 3, intra-framework fusion achieves 54.24% on average Rank1, which is improved by a large margin compared to the best single model. Limited by time, we only submitted the result of intra-framework fusion to the competition leaderboard. However, by integrating inter-framework fusion with intra-framework fusion, the performance can be further improved considerably, which demonstrated the effectiveness of our fusion strategy.

### 4 DISCUSSION

We have attempted to evaluate the SOTA methods in the TSGV fields. However, we regret to find that many latest methods have not released their codes. For those methods that have released their codes, e.g., proposal-based [17], regression-based [5], and span-based [10], they behave much worse than MMN [16] on the YouMakeUP dataset. It may be due to the different fine-grained levels between YouMakeUP and other public datasets. This need to be further studied in the future.

### 5 CONCLUSION

In this report, we present our approach for MTVG, the new task introduced in the 4th Person in Context (PIC) Challenge at ACM MM 2022. To relieve the gap between action-level and make-up features, we introduce three major components, i.e., diverse video modeling, augmented mutual matching network, and model ensemble. Our proposed method has achieved the 3rd place in MTVG test set.