Waseda_Meisei_SoftBank at TRECVID 2022

Kazuya Ueki[1,2], Yuma Suzuki[3], Hiroki Takushima[3],
Hideaki Okamoto[3], Hayato Tanoue[3], and Takayuki Hori[3,4]

[1] Department of Information Science, Meisei University,
Room 27-1809, Hodokubo 2-1-1, Hino, Tokyo 191–8506, Japan

[2] Faculty of Science and Engineering, Waseda University,
Room 40-701, Waseda-machi 27, Shinjuku-ku, Tokyo 162–0042, Japan

[3] AI Architect Department Technical Planning & Development Division,
Service Planning Technology Division, SoftBank Corporation
1-7-1 Kaigan, Minato-ku, Tokyo 105–7529, Japan

[4] Global Information and Telecommunication Institute, Waseda University,
Room 55-208, Okubo 3-4-1, Shinjuku-ku, Tokyo 162–0042, Japan

kazuya.ueki@meisei-u.ac.jp

Abstract. The Waseda_Meisei_SoftBank team participated in the ad-hoc video search (AVS), video-to-text (VTT), and activities in extended video (ActEV) tasks at TRECVID 2022 [1]. For this year's AVS task, we focused on several visual-semantic embedding approaches and submitted only fully automatic runs for both the main task and the progress subtask. Our best run for the main task achieved a mean average precision (mAP) of 28.2%, which was the highest accuracy among all submitted systems. This was the first time the Waseda_Meisei_SoftBank team participated in the VTT task. Our approach was to adopt SwinBERT, presented at CVPR2022, and implement it for fine-tuning with V3C1. The five automatic evaluation metrics yielded results of 0.415, 0.182, 0.037, 0.286, and 0.100 for CIDER, CIDER-D, BLEU, METEOR, and SPICE, respectively. The ActEV task was also our first. We proposed a system that combines 3D ResNet training with YOLOX and ByteTrack trained models and achieves a value of 0.9961 for P_{miss} @ $0.1R_{FA}$, 0.1080 for N_{MD} @ $0.1R_{FA}$, 0.9964 for $nAUDC$ @ $0.2R_{FA}$ for the primary activity and object detection (AOD) task, and 0.9829 P_{miss} @ $0.1R_{FA}$ and 0.9850 $nAUDC$ @ $0.2R_{FA}$ for a secondary activity detection (AD) task.

1 AVS Task

1.1 System Overview

Until last year, we had employed both concept-based and embedding methods; however, this year, we focused on embedding methods and submitted only the automatic systems. We developed the systems by combining several embedding methods, such as improved visual-semantic embeddings (VSE++) [2], a graph structured matching network (GSMN) [3], contrastive language-image pre-training (CLIP) [4], and self-supervision meets language-image pre-training (SLIP) [5]. Owing to the large video size and the attempts to extract features from multiple models, not all planned calculations could be completed. However, the best mean average precision was achieved by integrating only the results for which the calculations were completed.

1.2 Embedding Models

In the following section, we describe the details of the embedding models used to construct the system.

1. VSE++

   We used the implementation of VSE++[5] for training the models. To train the visual-semantic embedding, four image-caption datasets, Flickr8k [6], Flickr30k [7], MS-COCO [8], and Conceptual Captions [9], were used. The total number of image captions was 3,428,009. We used a gated recurrent unit (GRU) for the feature extraction from query sentences and the *ResNet-50*, *ResNet-101*, and *ResNet-152* models for the feature extraction from images. Owing to the large number of training data, 500,000 training data pairs and 50,000 validation data pairs were randomly selected for training the visual-semantic embedding models. We repeated this data-selection process 32 times for each of the three types of ResNet model, and then trained 96 embedding models.

2. GSMN

   The visual features of the GSMN were extracted using the bottom-up attention model[6] and the pre-trained bottom-up attention model provided. The bottom-up attention model is based on training *Faster R-CNN* with *ResNet-101*, using object and attribute annotations from the Visual Genome [10]. The GRU was used to extract the features from the text. To train the GSMN models, we used the GSMN implementation[7] and a total of 3,755,503 image-text pairs from Flickr8k, Flickr30k, MS-COCO, Conceptual Captions, and MSR-VTT [11]. Because of the large number of training data, we divided the training data and created nine models.

3. CLIP

   We did not train the models ourselves, but used the pre-trained models provided in the CLIP implementation[8]. We used eight types of pre-trained CLIP models: *RN50*, *RN101*, *RN50x4*, *RN50x16*, *RN50x64*, *ViT-B/32*, *ViT-B/16*, and *ViT-L/14*.

4. SLIP
   As with CLIP, we did not train the model on our own, but instead used publicly available pre-trained models[9]. We used two types of pre-trained SLIP models, (*ViT-Small* and *ViT-Base*), which were trained on YFCC15M. We also used *ViT-Base* models trained on CC3M or CC12M.

5. Diffusion Model
   We trained a diffusion model that generates an image embedding of CLIP conditioned on a text embedding of the CLIP *ViT-L/14* model. CC12M and a portion of LAION400M were used for training.

1.3 Inference Procedure

Using the model described in Subsection 1.2, we calculated the scores for V3C2, the test video dataset, based on whether it matched the query sentence. Because the image and

[5] `https://github.com/fartashf/vsepp`
[6] `https://github.com/peteanderson80/bottom-up-attention`
[7] `https://github.com/CrossmodalGroup/GSMN`
[8] `https://github.com/openai/CLIP`
[9] `https://github.com/facebookresearch/SLIP`

sentence vectors can be computed for all models, we use the cosine similarity between the frame images extracted from the videos and the query sentence to search for videos that match the query sentence. Images were extracted from the video every 10 frames, the similarity of the images to the query sentence was calculated, and the maximum value was used as the score for that video. For the diffusion model, we generated 1,000 image embeddings for one query and calculated the similarity between the generated images and the video frames. After calculating all scores for the test dataset, to obtain the final score, a min-max normalization was conducted for each model, the maximum value of which was 1.0, and the minimum value was 0.0. For each embedding method, all scores from multiple models were added and normalized again such that the maximum value for each method was 1.0 and the minimum value was 0.0. The final search result was determined using the score computed using the weighted sum of each embedding method.

1.4 Submissions and Results

Table 1. Our submitted runs for TRECVID 2022.

Run priority	Fusion weights					mAP
	VSE++	GSMN	CLIP	SLIP	Diffusion	
1	3	3	15	3	3	28.1
2	3	3	10	3	3	**28.2**
3	3	3	15	5	3	28.1
4	3	3	15	3	0	26.3

In this year's fully automatic systems, the test data were ranked according to the scores, which were calculated by simply adding the scores from the five different embedding methods multiplied by the fusion weights. The fusion weights were manually determined by evaluating the AVS tasks of 2019, 2020, and 2021 TRECVID. The accuracies of the fusion weights are listed in Table 1. As the reason for the highest fusion weights for the CLIP models, they had the highest precision and largest contribution over VSE++ and GSMN on the benchmark from last year. SLIP, which we newly introduced this year, is more accurate than CLIP; however, we set the integration weights lower than CLIP because we did not finish all feature extraction calculations and were unable to evaluate it sufficiently. In addition, the diffusion model introduced this year was given a lower fusion weight, partly because we had not yet obtained sufficient validation results, and only some of the models could be trained. However, the results of this year's benchmark show that priorities 1, 2, and 3, which introduced the diffusion model, had a higher mean average precision and contributed more than priority 4, which did not introduce the diffusion model. Because sufficient validation experiments could not be conducted, a detailed analysis and validation will be conducted in the future to confirm the effectiveness of this approach.

The results for all teams submitted to the main task are shown in Fig. 1. Among all systems submitted by all participating teams, the four systems we submitted were ranked within the top 1–4.

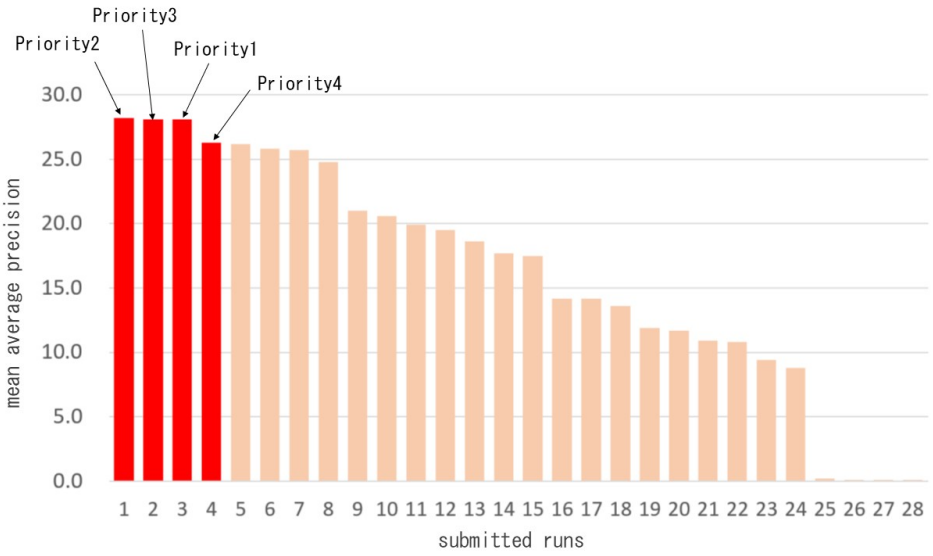

Fig. 1. Results of all fully automatic systems for all teams that submitted to the main task.

2 VTT Task

2.1 Introduction

Video-to-text (VTT) is a task to create a description in natural language after using the visual context and the voice context given from a video sequence. Image captioning is a task similar to VTT; however, VTT tasks require recognition in the temporal direction, in addition to recognition in the spatial direction required for image captioning. In addition, the input video has a variable length and contains numerous redundancies, which must be addressed. In this study, we conducted a pretraining on the VATEX dataset[12] based on SwinBERT[10][13], which is the latest method for conducting a VTT task, and applied a fine-tuning using the TRECVID-VTT dataset. The results were 0.415 for CIDER, 0.182 for CIDER-D, 0.037 for BLEU, 0.286 for METEOR, and 0.100 for SPICE.

2.2 Methods

In this section, we explain our VTT method. We pretrained the state-of-the-art SwinBERT model on the video captioning task and on the VATEX dataset and fine-tuned it on the TRECVID-VTT dataset. SwinBERT is a model comprising Video SwinTransformer[14] for video feature extraction and Transformer [15] Encoder for the decoder. Video SwinTransformer is an extension of SwinTransformer[16], which is an image-recognition model used for video-recognition tasks. SwinTransformer is based on the Transformer model proposed in natural language processing, and achieves a high accuracy and speed by improving the efficiency of the shifted window based multihead self-attention (SW-MSA) mechanism and its calculation method. SwinTransformer outputs feature maps that are compatible with the CNN architecture, and pre-trained models are provided with a wide variety of model scales. It is therefore also called a

[10] `https://github.com/microsoft/SwinBERT`

standard model for image recognition and is used to replace a CNN. With SwinBERT, video features are obtained using Video SwinTransformer, and captions are decoded based on the video features in the Transformer Encoder. To deal with long and redundant videos, SwinBERT avoids unnecessary calculations by creating a sparse attention mask using Transformer Encoder. The attention mask is also acquired through training, which allows it to autonomously determine redundant parts of the video.

2.3 Datasets

We used two datasets for this method, which are described in the following section.

TRECVID-VTT: The TRECVID-VTT dataset consists of datasets of the TRECVID VTT task provided from 2016 to 2021. There were 10,862 videos with a length of 3–5 s, and each video had two to five captions written by an annotator. The breakdown of the dataset was 6,475 Twitter Vine data, 1,009 Flickr data, and 3,376 Vimeo Creative Commons Collection (V3C) data. In the Twitter Vine data, 3,528 have captions between two and five sentences, and all others have five captions. V3C1 is a dataset that uses a portion of V3C data. The number of videos was 7,475, among which 2,008 were provided as test data for TRECVID2022. The 2,008 videos were captured by annotators based on the following perspectives:

- Who: Who is in the video? Does it describe specific objects or entities (people, animals, objects)?
- What: What are the objects and entities in the images doing? Does the video describe an action or state?
- Where: Where was the video shot? Does the video describe a location (geographical or architectural)?
- When: When was the video taken? Does the video describe the time of day, season, or other temporal elements?

VATEX: We used the VATEX dataset to pre-train the model. The VATEX dataset contains 41.3k videos with 10 captions in both English and Chinese. This dataset was used in this study because the video length is less than 10 s, the domain is close to that of the TRECVID-VTT dataset, and the quality is high.

2.4 Setups

The experimental setup is described in the following section. During the pre-training phase, we used the official training set of the VATEX dataset as the training dataset. We used the public test set as the test data for the evaluation. We randomly initialized the initial parameters of the Transformer Encoder of the decoder. We used AdamW as the optimization method, and the learning curve was a warm-up for 10% of the total learning steps and linear decay thereafter. For pretraining, we used an official implementation model. For fine-tuning, we used the TRECVID-VTT dataset as the training dataset and trained for an additional 15 epochs.

2.5 Results

The results for each evaluation metric among our submitted run files are shown in Table 2. We selected the run files submitted based on the validation data with high accuracy and annotations of the TRECVID-VTT dataset.

Table 2. Results of our submitted runs for TRECVID 2022 VTT task.

Runfile	CIDER	CIDER-D	BLEU	METEOR	SPICE
1	**0.415**	0.178	0.033	0.260	0.077
2	0.348	0.141	0.026	0.252	0.084
3	0.350	0.150	0.028	0.260	0.087
4	0.388	**0.182**	**0.037**	**0.286**	**0.100**

2.6 Conclusion

For the TRECVID VTT task, we submitted the results of fine-tuning SwinBERT using the TRECVID-VTT dataset. The five automatic evaluation metrics yielded results of 0.415, 0.182, 0.037, 0.286, and 0.100 for CIDER, CIDER-D, BLEU, METEOR, and SPICE, respectively.

3 ActEV Task

3.1 Overview

The Activities in Extended Video (ActEV) series of evaluations were designed to accelerate the development of robust, multi-camera, automatic activity detection systems in known and unknown facilities for forensic and real-time alerting applications. Activities in extended videos are temporally and spatially dispersed, requiring algorithms to detect and localize activities under a variety of collection conditions. We propose a system that combines 3D ResNet [17] training using YOLOX [18] and ByteTrack [19] trained models and achieved a value of 0.9961 for P_{miss} @ $0.1R_{FA}$, 0.1080 for N_{MD} @ $0.1R_{FA}$, and 0.9964 for $nAUDC$ @ $0.2R_{FA}$ for a primary Activity and Object Detection (AOD) task, and 0.9829 P_{miss} @ $0.1R_{FA}$ and 0.9850 $nAUDC$ @ $0.2R_{FA}$ for a secondary Activity Detection (AD) task.

3.2 Methods

This system comprises three steps: YOLOX for object detection, ByteTrack for tracking, and 3D ResNet for activity classification. YOLOX is a single-stage object detector that makes several modifications to YOLOv3 using a DarkNet53 backbone. Specifically, the head of YOLO is replaced with a decoupled head. ByteTrack eliminates the problem of non-detection by matching bounding boxes with low confidence values using a motion model that uses a queue called tracklets, which indicates the object being tracked. It also considers bounding boxes with low confidence values. 3D-ResNet is a ResNet constructed using a 3D convolution to consider the time axis. The prediction is made by placing videos created based on the tracking results obtained by YOLOX + ByteTrack into 3D ResNet, classifying the activities, and matching the results.

3.3 Experiments

YOLOX and ByteTrack were implemented in this study [11], and ByteTrack was implemented based on this [12]. YOLOX and ByteTrack were trained on the CrowdHuman

[11] `https://github.com/ifzhang/ByteTrack`
[12] `https://github.com/facebookresearch/pytorchvideo`

[20] and MOT17 [21] half-train sets. The training data for this system were labeled for only people. The first two steps used only the weights of the existing trained model. The output of ByteTrack was then used to generate the input video for 3D ResNet. The tracking results were joined together to form a single video and then stretched and resized to produce a video with a 256 × 256 pixel resolution. However, the training videos were preprocessed using the kitware-meva-training tracking annotation for the next processing step. 3D ResNet was trained on 53,027 square videos labeled for activity created from annotations of the above-mentioned kitware-meva-training tracking. The videos were normalized at the time of training, and two data augmentations, cropping to a pixel resolution of 224 × 224 and flipping the image horizontally, were then applied. The model parameters included a learning rate of 1e-5, batch size of 8, 20 epochs, a momentum optimizer, a momentum of 0.9, and a weight decay of 1e-4.

3.4 Results

Our results for the two tasks, AOD and AD, are shown in the table below.

Table 3. Our results for TRECVID 2022 ActEV task.

Activity and Object Detection (AOD)			Activity Detection (AD)	
P_{miss} @ $0.1R_{FA}$	N_{MD} @ $0.1R_{FA}$	$nAUDC$ @ $0.2R_{FA}$	P_{miss} @ $0.1R_{FA}$	$nAUDC$ @ $0.2R_{FA}$
0.9961	0.1080	0.9964	0.9829	0.9850

There are two major issues to be considered for this system. First, YOLOX + ByteTrack, as mentioned above, only detects people and ignores other objects such as cars. This system is composed of three steps, and if the system misses anything in the first step, the mistake cannot be rectified. Therefore, training with the appropriate labels, which we omitted in this study, is necessary. With the second method, 3D ResNet, the input for the model is compressed down to eight frames, and thus a large amount of information is missed. Because the video treated at this time is 5 min in length, it is essential to update to a model that can handle more frames.

Acknowledgments

This work was partially supported by JSPS KAKENHI (Grant Number 18K11362).

References

1. G. Awad, K. Curtis, A. A. Butt, J. Fiscus, A. Godil, Y. Lee, A. Delgado, J. Zhang, E. Godard, B. Chocot, L. Diduch, J. Liu, Y. Graham, G. Quénot, "An overview on the evaluated video retrieval tasks at TRECVID 2022," In Proc. of TRECVID 2022, 2022.
2. F. Faghri, D. J. Fleet, R. Kiros, and S. Fidler, "VSE++: Improved Visual-Semantic Embeddings," arXiv:1707.05612, 2017.
3. C. Liu, Z. Mao, T. Zhang, H. Xie, B. Wang, Y. Zhang, "Graph Structured Network for Image-Text Matching," In Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
4. A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," arXiv:2103.00020, 2021.

5. N. Mu, A. Kirillov, D. Wagner, S. Xie, "SLIP: Self-supervision meets Language-Image Pre-training," arXiv:2112.12750, 2021.
6. C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier, "Collecting Image Annotations Using Amazon's Mechanical Turk," Proc. of the NAACLHLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, pp.139–147, 2010.
7. P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," Transactions of the Association for Computational Linguistics. vol.2, pp.67–78, 2014.
8. T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," arXiv:1405.0312, 2014.
9. P. Sharma, N. Ding, S. Goodman, and R. Soricut, "Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning," Proc. of the 56th Annual Meeting of the Association for Computational Linguistics, pp. 2556–2565, 2018.
10. R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalanditis, L.-J. Li, D.A. Shamma, M.S. Bernstein, L. Fei-Fei, Y. Kalantidis, L.-J. Li, D.A. Shamma, M.S. Bernstein, and F.-F. Li, "Visual Genome : Connecting language and vision using crowdsourced dense image annotations," arXiv:1602.07332, 2016.
11. J. Xu, T. Mei, T. Yao, Y. Rui, "MSR-VTT: A Large Video Description Dataset for Bridging Video and Language," In Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
12. Wang, Xin and Wu, Jiawei and Chen, Junkun and Li, Lei and Wang, Yuan-Fang and Wang, William Yang, "VaTeX: A Large-Scale, High-Quality Multilingual Dataset for Video-and-Language Research" The IEEE International Conference on Computer Vision (ICCV) 2019.
13. Lin, Kevin and Li, Linjie and Lin, Chung-Ching and Ahmed, Faisal and Gan, Zhe and Liu, Zicheng and Lu, Yumao and Wang, Lijuan, "SwinBERT: End-to-End Transformers with Sparse Attention for Video Captioning" Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2022.
14. Liu, Ze and Ning, Jia and Cao, Yue and Wei, Yixuan and Zhang, Zheng and Lin, Stephen and Hu, Han, "Video Swin Transformer" Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2022.
15. Vaswani, Ashish and Shazeer, Noam and Parmar, Niki and Uszkoreit, Jakob and Jones, Llion and Gomez, Aidan N and Kaiser, Łukasz and Polosukhin, Illia, "Attention is All you Need" Thirty-first Conference on Neural Information Processing Systems(NeurIPS) 2017.
16. Liu, Ze and Lin, Yutong and Cao, Yue and Hu, Han and Wei, Yixuan and Zhang, Zheng and Lin, Stephen and Guo, Baining, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows" Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) 2021.
17. K. Hara, H. Kataoka, and Y. Satoh, "Learning spatio-temporal features with 3d residual networks for action recognition," In ICCV, 2017.
18. Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," arXiv preprint arXiv:2107.08430, 2021.
19. Y. Zhang, P. Sun, Y. Jiang, D. Yu, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," arXiv preprint arXiv:2110.06864, 2021.
20. S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, "Crowdhuman: A benchmark for detecting human in a crowd," arXiv preprint arXiv:1805.00123, 2018.
21. A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, " Mot16: A benchmark for multi-object tracking," arXiv preprint arXiv:1603.00831, 2016.