Kindai University and Osaka Gakuin University at TRECVID 2020 AVS and ActEV Tasks

Daiki Mukai*, Ryosuke Utsunomiya*, Shunsuke Utsuki*, Kimiaki Shirahama*,
Takashi Matsubara† and Kuniaki Uehara‡
* Department of Informatics, Kindai University
† Graduate School of Engineering Science, Osaka University
‡ Department of Business Administration, Osaka Gakuin University
Contact: shirahama@info.kindai.ac.jp

***Abstract*—This paper presents our methods developed for Ad-hoc Video Search (AVS) and Activities in Extended Video (ActEV) tasks in TRECVID 2020. Our AVS method is based on embeddings that map visual features and textual descriptions into a common space [1], so that the relevance of each shot to a topic can be computed. The embedding model in our method is trained on Conceptual Captions dataset [2] that contains more than 3 million image-caption pairs. Visual features for images and the ones for shots in V3C1 dataset are extracted using ResNeXt WSL, which is pre-trained in a weakly-supervised fashion on 940 million social media images with 1500 noisy hashtags and fine-tuned using ImageNet dataset [3]. The MAP of the submitted result (F_P_C_D_kindai_ogu.20_1) is 0.133. Also, the progress subtask reveals that the performance of our this year's method is significantly higher than that of last year's one. This indicates the importance of using large-scale data to train an embedding model. Furthermore, our own experiments show an interesting relation of retrieval performances to the size of training data and the number of frames analysed for each shot.**

Our ActEV method focuses on capturing spatio-temporal relations among objects involved in an event. To this end, our method firstly detect objects in a segment of 100 frames using M2Det [4], which is a multi-scale object detection trained on MS COCO dataset [5]. The segment is then represented by a graph created by connecting objects that are spatially and temporally close to each other. Here, each object is characterised by a visual feature extracted with SE-ResNeXT-101 [6] trained on ImageNet dataset. Afterwards, our method builds a Spatial-Temporal Graph Convolutional Network (ST-GCN) [7] that abstracts the feature of each object in a graph into a higher-level one by integrating features of connected objects. Finally, such abstracted features are merged to examine the occurrence of each of 35 events. The submitted result (kindai_ogu_baseline) got a partial AUDC of 0.9682.

I. Introduction

We are continuously participating in TRECVID to make an objective performance comparison between our system and systems developed all over the world [8]. This year we participated in Ad-hoc Video Search (AVS) [9] and Activities in Extended Video (ActEV) tasks. For AVS task, we aim to examine the effect of the scale of training data on a retrieval performance. Specifically, models used in our last year's method [10] were mainly trained on 410K image-caption pairs contained in MS COCO dataset [5]. Compared to this, the model used in this year's model is trained on 3M image-caption pairs in Conceptual Captions (CC) [2]. Apart from the official evaluation, we examine a further larger training dataset that is the combination of CC and MS COCO datasets. In addition, we investigate the effect of analysing multiple frames on a retrieval performance. This shows an interesting result that the performance improvement by analysing multiple frames is significantly reduced when using a model trained on a large-scale dataset. In other words, such a model can reasonably predict contents that are invisible in the keyframe of a shot, and its performance is not so much improved by analysing more frames.

For ActEV task, an occurrence of an event is thought to be characterised by specific object appearances and their spatio-temporal relations. For example, an occurrence of the event "person_opens_trunk" should involve a characteristic pose of a person to open the trunk of a car, and a specific appearance of the car with the trunk open. In addition, the person and the car must be spatially close to each other. To capture such object appearances and spatio-temporal relations, a Spatial-Temporal Graph Convolutional Network (ST-GCN) [7] is used to abstract the visual feature of an object into a higher-level one by integrating features of spatially and temporally close objects. More concretely, each video segment is represented by a graph indicating the connectivities of objects that are spatially and temporally close to each other. Then, an ST-GCN is trained on such graphs so as to attain object feature abstraction that is useful for accurate event detection.

II. AVS Task

This section firstly presents our AVS method based on visual-semantic embeddings that project visual features and textual descriptions into a common space. Then, the results obtained by the official evaluation are described. Finally, the results acquired by our internal experiments are discussed.

A. Method

Our AVS method is based on VSE++ that is a simple but effective model for visual-semantic embeddings [1]. VSE++ consists of an image encoder that extracts a visual feature from an image, a text encoder that extracts a textual feature from a caption, and Fully-Connected (FC) layers that map the visual and textual features into a common space. A pre-trained Convolutional Neural Network (CNN) is usually used as the image encoder. The text encoder is implemented using a network consisting of a word embedding layer followed by a layer of Gated Recurrent Unit (GRU). Given a training dataset, the text encoder and FC layers are optimised so that the visual feature of an image and the textual feature of the corresponding caption are projected close to each other in the common space. In addition, the optimisation aims the projection where the projected feature of an image (or a caption) is distant from the projected features of irrelevant captions (or images). For more details, please refer to our last year's notebook paper [10].

We extend VSE++ used in our last year's method for the following two points: First, MS COCO dataset containing 410K image-caption pairs [5] and Flickr 30K dataset containing 30K image-caption pairs [11] were used last year. In contrast, this year we use a much larger dataset, Conceptual Captions (CC) containing more than 3M image-caption pairs [2], in order to build a more accurate VSE++ model. Second, we change the image encoder from ResNet152 [12] pre-trained on 1.28M images in ImageNet dataset [3] to ResNeXt-101 WSL (32x48d) [13], which is pre-trained in weakly-supervised fashion on 940M social media images with 1.5K noisy hashtags and fine-tuned using ImageNet dataset. As reported in [13], ResNeXt-101 WSL achieves much better performances than ResNet152. Thus, the former is expected to work as a much better image encoder in VSE++.

B. *Results*

Fig. 1 shows the ranking of methods developed for AVS task (fully automatic category). Each method is ranked based on its MAP over 30 queries in the main task. As shown in Fig. 1, the MAP of our submitted run is 0.133. Fig. 2 presents the ranking of methods for AVS's progress subtask. These methods were developed last year and this year. By comparing them on the same topics, one can see the performance improvement from last year. As can be seen from Fig. 2, the MAP of our this year's method is much higher than those of our four methods developed last year. This validates the importance of using a large-scale dataset to train VSE++.

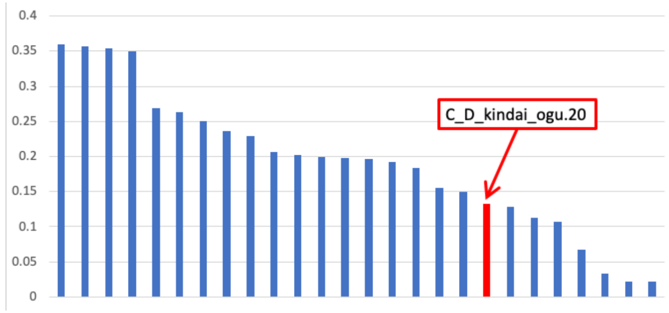

Fig. 1. Ranking of methods developed for AVS task (main task).

C. *Detailed Analysis*

We report additional results obtained by our own experiments on 30 topics in last year's AVS task. We aim at examining whether a further better performance is attained when using a larger dataset than CC. For this, CC including 3M image-caption pairs is combined with MS COCO dataset (newest version) including 0.6M image-caption pairs, so as to create a dataset with 3.6M pairs. In addition, it is traditionally said that analysing multiple frames in a shot improves a performance. Hence, the following two projections using VSE++ are compared. The first is to project the visual feature of the keyframe of a shot into the common space. The second projection starts with equidistantly sampling 10 additional frames from a shot in addition to the keyframe. Then, average-pooling is used to aggregate visual features of these 11 frames into a single feature, which is finally projected into the common space[1].

Fig. 3 shows a performance comparison among four variants of retrieval methods that are defined by a different combination of a training dataset and a number of analysed frames. Specifically, *CC-1* uses VSE++ trained on CC and only analyses the keyframe of a shot while the same VSE++ is used to analyse 11 frames in *CC-11*. Similarly, *CC+COCO-1* and *CC+COCO-11* use VSE++ trained on CC combined with MS COCO dataset, but the former analyses only the keyframe while the latter considers 11 frames. The rightmost set of four bars presents the MAPs of the above-mentioned variants. The comparison between *CC-1*'s MAP (0.091) and *CC+COCO-1*'s MAP (0.115) validates the effectiveness of using a larger training dataset than CC. By comparing *CC-1*'s MAP to *CC-11*'s MAP (0.127) verifies the significant performance improvement by analysing multiple frames in a shot. However, *CC+COCO-11*'s MAP is 0.128 that is slightly better than *CC+COCO-1*'s MAP. This means that, for *CC+COCO-1* that already achieves a high performance, analysing multiple frames only yields a very small improvement. A deeper investigation leads us to the conclusion that a high performance model like *CC+COCO-1* can perform reasonable prediction of contents which are invisible in the keyframe of a shot, so only analysing that keyframe is enough for such a model.

III. ActEV Task

This subsection firstly presents our event detection method based on an ST-GCN, and then shows the official result obtained by it.

A. *Method*

Fig. 4 illustrates our event detection method. It firstly extracts segments from a video by sliding a window of 100 frames with a stride of 50 frames. More precisely, each segment is a sequence of 20 frames that are sampled every five frames. Then, as shown in Fig. 4 (a), objects in every frame are detected using M2Det [4]. It performs fast, accurate multi-scale object detection by refining multi-level features extracted by a backbone network (in our case VGG16 [14]) into more representative multi-level, multi-scale features via

[1] We also tested average-pooling on projected features of 11 frames, and max-pooling instead of average-pooling. Among all the tested cases, average-pooling on visual features of 11 frames achieved the best performance.

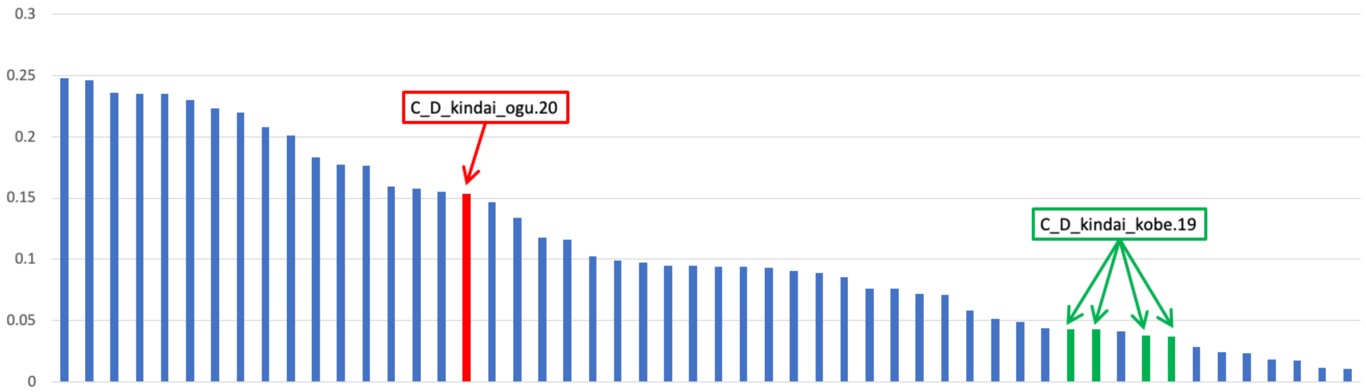

Fig. 2. Ranking of methods developed for AVS task (progress subtask).

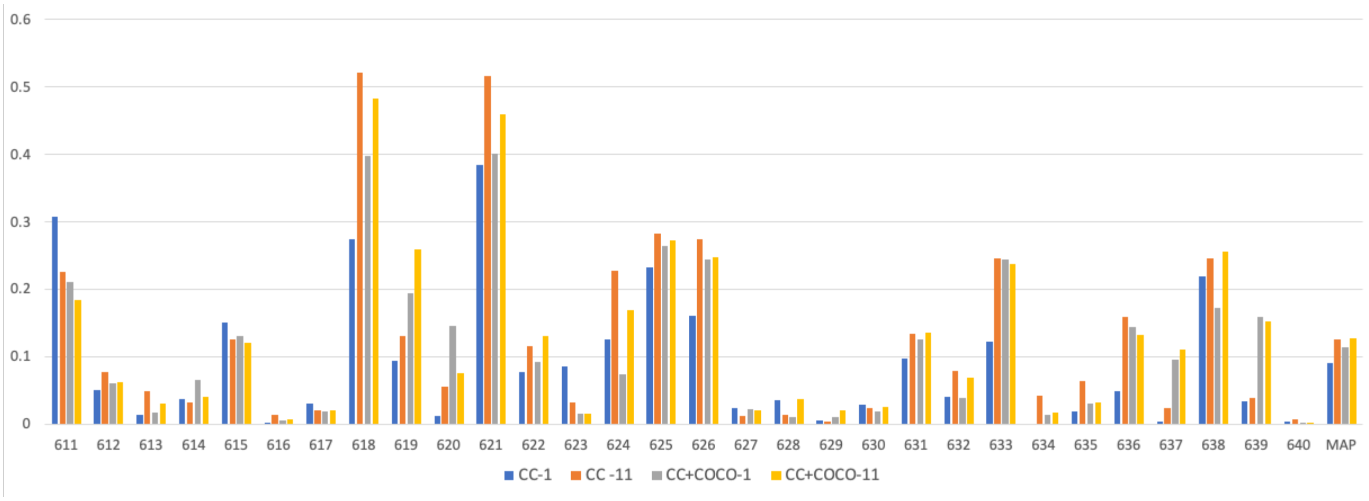

Fig. 3. Performance comparison among four methods defined by different combinations of training data and the number of frames analysed for each shot.

the Multi-Level Feature Pyramid Network (MLPFN). In particular, M2Det that we used is trained on MS COCO dataset [5] and detects 81 types of objects like person, car and bicycle. Finally, each of detected objects is represented by a feature extracted with SE-ResNeXT-101, which selectively weights feature maps in a channel-wise fashion to emphasise useful features for accurate image classification [6]. Especially, a 2048-dimensional feature extracted from the penultimate layer of SE-ResNeXT-101 is used to describe an object.

Next, a graph like the one depicted in Fig. 4 (a) is created for a segment in the following way: Objects in the same frame are connected to each other if the distance between the centres of their regions is less than 50 pixels. In addition, if two objects in two consecutive frames have very similar features, they are regarded as the same instance and connected to each other. This way, all the segments are represented by graphs.

Fig. 4 (b) shows the structure of our event detection model based on an ST-GCN, which extends the convolution operation on grid-structured data like images to graphs [7]. Specifically, pixels in a local image region form a grid structure which assigns a spatial order to those pixels, such as the top-left pixel, the top, second-leftmost pixel, and so on. Weights of a convolution filter are applied to pixels based on this spatial order. In contrast, nodes in a graph do not have such an order, and the ST-GCN defines it by mapping nodes into certain partitions. As shown in the red-coloured node and four blue-colored nodes in Fig. 4 (b), for each node v_i, the set of neighbouring nodes is defined as $B(v_i) = \{v_j | d(v_i, v_j) \leq 1\}$. We simply define the following two partitions for $B(v_i)$: The one only includes v_i, and the other includes the nodes directly connected to v_i.

The main idea of the ST-GCN is to define a convolution operation by applying different filter weights to nodes in a different partition. This is formulated as follows:

$$f_{out}(v_i) = \sum_{v_j \in B(v_i)} \frac{1}{|p(v_j)|} \, f_{in}(v_j) \, \mathbf{w}\left(p(v_j)\right), \qquad (1)$$

where $f_{in}(v_j)$ denotes the 2048-dimensional feature of the object corresponding to v_j. $p(v_j)$ indicates the partition to which v_j belongs, and $\mathbf{w}\left(p(v_j)\right)$ is the filter weights defined for $p(v_j)$. In addition, $|p(v_j)|$ is the cardinality of $p(v_j)$ and used to balance the contribution of the partition only including

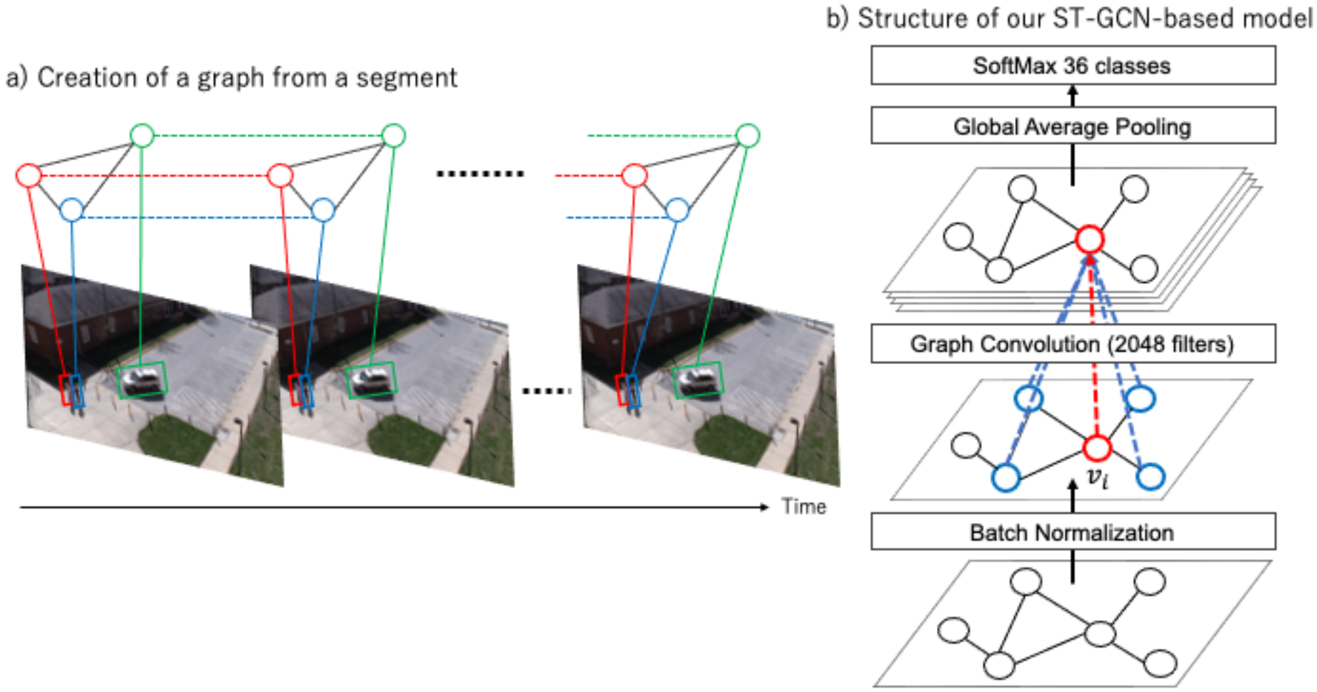
a) Creation of a graph from a segment
Time
b) Structure of our ST-GCN-based model
SoftMax 36 classes
Global Average Pooling
Graph Convolution (2048 filters)
v_i
Batch Normalization

Fig. 4. A overview of our event detection method based on a GCN.

v_i and the one including neighbouring nodes to the convolution operation. The resulting $f_{out}(v_i)$ can be thought as a value that abstracts the feature of the object corresponding to v_i and those of its surrounding objects. As shown in Fig. 4 (b), one feature map is formed by computing $f_{out}(v_i)$ for each node in the graph, and feature maps obtained using different filters can be stacked. As a result, the feature of each node is refined into a higher-level feature that considers not only the corresponding object but also its surrounding ones.

After the graph convolution described above, global average pooling is used to aggregate features of nodes in the graph into a single feature. Finally, this feature is fed into a softmax layer targeting at 36 classes. Here, 35 of these classes correspond to 35 events defined in ActEV task and the remaining one class expresses “no event happens”.

B. Results

The ST-GCN-based event detection model in Fig. 3 (b) is trained on $5,247$ segments extracted from the training partition in VIRAT V1 and V2 data. Adam optimiser with the initial learning rate 0.00001 is used to train the model with 300 epochs. Then, the model is applied to $3,956$ segments in the test partition. Finally, an interval where a certain event occurs is identified as a sequence of segments for which this event is continuously detected. Fig. 5 shows the ranking of event detection methods developed for ActEV task. Our method which gets a partial AUDC of 0.9682 is unfortunately ranked at the bottom. We will improve it by investing the issues described in the next section.

IV. Conclusion and Future Work

This paper introduced our methods developed for AVS and ActEV tasks in TRECVID 2020. For AVS task, we use VSE++ as the base model and examine the influence of using large-scale training data on retrieval performances. Especially, training dataset containing 3.6M image-caption pairs is created by combining CC with MS COCO, and leads to VSE++ that achieves a significantly improved performance. In addition, we get one interesting finding that such a high performance model can perform reasonable prediction of contents which are invisible in the keyframe of a shot. In other words, for this model, only analysing the keyframe is enough and analysing multiple frames yields a very slight performance improvement. We plan to improve our AVS method by adopting motion and acoustic features as well as the ones that characterise the transitions of frame-level visual features via a recurrent (or convolutional) neural network. Moreover, we aim to improve the processing of a caption (or topic) by explicitly modelling the correspondence between noun phrases in it and object regions in a frame.

For ActEV task, our method has much room for improvements. We describe some issues that are potentially useful for performance improvements. First, the current ST-CGN-based model uses a softmax layer that is useful for multi-class classification where an example belongs to one of mul-

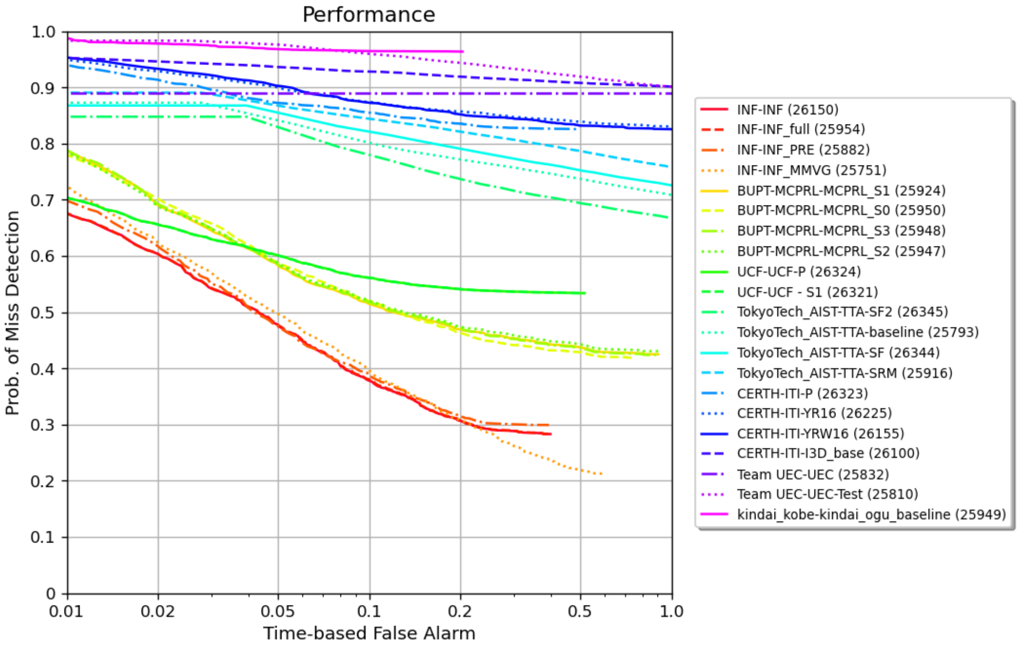

Fig. 5. Leaderboard for methods developed for TRECVID 2020 ActEV task.

tiple classes. But, multi-label classification where an example belongs to multiple classes is more suitable for ActEV task, because multiple events can occur in the same segment. We will pursue to incorporate multi-label classification into the ST-GCN-based model by adopting a binary cross-entropy loss. Second, overfitting may be one of the main reasons for the poor performance. To avoid it, we plan to utilise an attention mechanism in order to selectively emphasise objects (nodes) or feature dimensions that are relevant to detecting certain events. Finally, the current ST-GCN-based model uses a very simple configuration of filter weights by just grouping objects (nodes) into two partitions. A significant performance improvement could be obtained if filter weights could be configured based on more sophisticated partitions that consider object categories and temporal locations of frames.

References

[1] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, “Vse++: Improving visual-semantic embeddings with hard negatives,” in *Proc. of BMVC 2018*, 2018.

[2] P. Sharma, N. Ding, S. Goodman, and R. Soricut, “Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning,” in *Proc. of ACL 2018*, 2018, p. 2556–2565.

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[4] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, “M2det: A single-shot object detector based on multi-level feature pyramid network,” in *Proc. of AAAI 2019*, 2019, pp. 9259–9266.

[5] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *Proc. of ECCV 2014*, 2014, pp. 740–755.

[6] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proc. of CVPR 2018*, 2018, pp. 7132–7141.

[7] S. Yan, Y. Xiong, and D. Lin, “Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning,” in *Proc. of AAAI 2018*, 2018, pp. 7444–7452.

[8] G. Awad, A. A. Butt, K. Curtis, Y. Lee, J. Fiscus, A. Godil, A. Delgado, J. Zhang, E. Godard, L. Diduch, J. Liu, A. F. Smeaton, Y. Graham, G. J. F. Jones, W. Kraaij, and G. Quénot, “Trecvid 2020: comprehensive campaign for evaluating video retrieval tasks across multiple application domains,” in *Proceedings of TRECVID 2020*. NIST, USA, 2020.

[9] J. Lokoč, W. Bailer, K. Schoeffmann, B. Muenzer, and G. Awad, “On influential trends in interactive video retrieval: Video browser showdown 2015–2017,” vol. 20, no. 12, p. 3361–3376, 2018.

[10] K. Shirahama, D. Sakurai, M. Takashi, and K. Uehara, “Kindai university and kobe university at TRECVID 2019 avs task,” in *Proc. of TRECVID 2019*, 2019.

[11] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, “From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions,” in *Proc. of ACL 2014*, 2014, pp. 67–78.

[12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. of CVPR 2016*, 2016, pp. 770–778.

[13] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten, “Exploring the limits of weakly supervised pretraining,” in *Proc. of ECCV 2018*, 2018, pp. 185–201.

[14] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. of ICLR 2015*, 2015.