# HYBRID SEQUENCE ENCODER FOR TEXT BASED VIDEO RETRIEVAL

**Xiang Wu**
Alibaba Group, China
weiyi.wx@alibaba-inc.com

**Da Chen**
Alibaba Group, China
chen.cd@alibaba-inc.com

**Yuan He**
Alibaba Group, China
heyuan.hy@alibaba-inc.com

**Hui Xue**
Alibaba Group, China
hui.xueh@alibaba-inc.com

**Mingli Song**
Zhejiang University, China
brooksong@zju.edu.cn

**Feng Mao**
Alibaba Group, China
maofeng.mf@alibaba-inc.com

**ABSTRACT**

This report presents our system developed for Ad-hoc Video Search(AVS) task in TRECVID 2019 as Team ATL. In this AVS task, we apply a hybrid sequential encoder which make use of the utilities of not only the multi-modal sources but also the feature extractors such as GRU, aggregated vectors, graph modeling, *etc.* Our motivation is mapping video embedding and language embedding into a learned semantic space. We observe that by combining different models and make use of their utilities for feature extraction, we can take better advantage of large batches and hard examples. Our models are trained on MSRVTT [1], IACC.3,TGIF and TRECVID2016 VTT datasets with different hybrid visual and text architectures. The final ensemble model achieves 0.163 infap and won the first place in this task.

***K*eywords** Multi-modal learning · Video retrieval · Graph network · Sequential model

## 1 Introduction

In this report, we present our hybrid sequential model for Ad-hoc Video Search (AVS) task in TRECVID 2019 [2] as team ATL. This is a task query video by text description in a zero shot manner.

Our hybrid sequential model is divided into two streams with visual module and text module. The two modules are applied to extract the embedding features from both modal, and presented in a common space. By a common space learning, the model can be trained.

There are lots of previous work has been proposed to solve this task with similar manner to our approach, VSE++ [3] improve image-semantic embedding inference by hard example mining with a improved marginal ranking loss. Word2VVec [4] learns to predict a deep visual feature of textual input based on multi-scale sentences. This type of methods embed multi-modal inputs from different domains into same feature space and apply a common space learning. Hence, to better represent the feature of different modal source input become essential.

In our work, we mainly focus on the optimization of visual and text embedding. Three different submodels *i.e.,* graph convolution model, sequence model and aggregated model, are applied along with their own strengths following a control gate as an automatic adjustment strategy.

To further improve the performance of our model, we use additional training data from IACC.3, MSRVTT, TGIF, and TRECVID2016 VTT. The result on MSRVTT is also reported along with each runs of our submit.

## 2 Related techniques and proposed Method

In this section, we first present a shot introduction about multi-modal learning, sequential models, followed by the details of the hybrid model applied in this task.

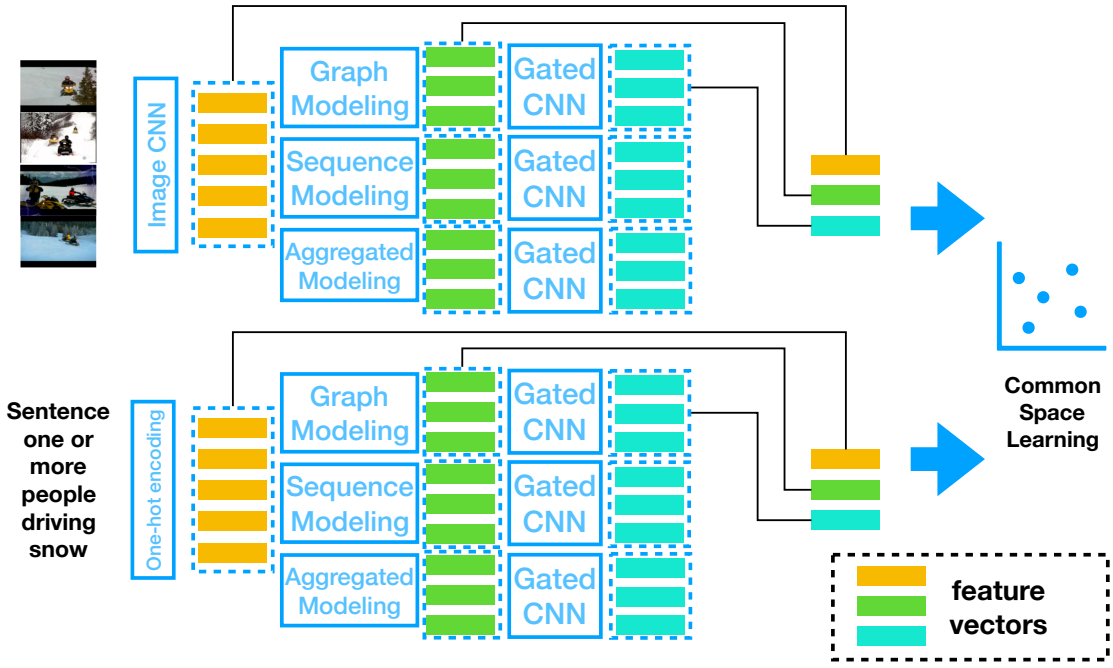


Figure 1: Overall architecture of the method in the final submission

### 2.1 Multi-modal learning

Multi-modal learning is a challenging task as it has different type of inputs such as visual information, audio, text, *etc.* There is a series of multi-modal learning methods tend to present the visual embedding and text embedding in a same feature space. Methods such as Vse++ [3], [5],Devise [6], *etc.* present the image embedding feature and text embedding feature in a same feature space. While there are methods [7, 8, 9] directly include video embedding features and text embedding features in a unique feature space.

State-of-the-art method *i.e.,* Dual-encoder method [9] applies a simple model to construct a common space and achieve the best performance in video retrieval task via multi-modal common space learning. However, Dual-encoder method only focus on current input video and its corresponding class, ignoring the whole distribution of the whole dataset with different classes and samples. Due to the video clipping and editing, temporal information is not consistently reliable and easy to mislead the encoder to properly extract the video feature from visual and text perspectives.

To solve this issue, in this competition, Vlad [10] based method is applied to embedded into the model so that the model can have a more general view among all training data. Meanwhile, to strengthen the robustness of the encoder, a graph convolutional network is included to better understand the hierarchy of the scenes in the video.

### 2.2 Sequential models applied for multi-modal learning

Video feature sequence classification is essentially the the task of aggregating video features, that is, to aggregate $N$ $D$-dimensional features into one $D'$-dimensional feature by mining statistical relationships between these $N$ features. The aggregated $D'$-dimensional feature is a highly concentrated embedding, making the classifier easy to mapping the visual embedding space into the label semantic space. It is common using recurrent neural networks, such as LSTM (Long Short-Term Memory Networks) [11] [12] [13] and GRU (Gated recurrent units) [14] [15], both are the state-of-the-art approaches for many sequence modeling tasks. However, the hidden state of RNN is dependent on previous steps, which prevent parallel computations. Moreover, LSTM or GRU use gate to solve RNN gradient vanish problem, but the sigmoid in the gates still cause gradient decay over layers in depth. It has been shown that LSTM has difficulties in converging when sequence length increase[16].There also exist end-to-end trainable order-less aggregation methods, such as DBoF(Deep Bag of Frame Pooling) [17].

### 2.3 Hybrid sequential model for our submission

In this section, the method applied in this competition is detailed introduced including the encoder for both visual content and text content, followed by the loss function of the model. The overall architecture of the method in the final submission is shown in Figure 1.

#### 2.3.1 Visual encoder

The proposed method is based on Dual-encoder method [9] with three individual sub-model ensemble 'on-the-fly' similar to the method proposed in [18]. As suggested in [9], the features of the video are extracted from different level. Given a video input, an ImageNet11k [19] pre-trained network is applied for feature extraction. The feature for a input video can be defined as $FN * D$ where $N$ is the number of frames and $D$ is the dimension of the extracted feature size for a single frame. Each frame is obtained from every 0.5 seconds. The first level encoding feature is conducted with an Mean pooling. The video feature is then represented by the average of features of all frames, and can be written as: $F1 = avg(FN * D)$.

**Second level embedding** The second level embedding feature is composed by three parts named by $F2g$, $F2s$, $F2a$. As shown in Figure 1, the feature extracted from pre-trained CNN model is put into three different modules, convolutional graph model [20] $F2g$, sequence model [21] $F2s$, and aggregation model $F2a$.

For convolutional graph model, it can effectively learn the hierarchical information of the video among frames. We observe that a typical video is composed by frames, scenes and events. After video clipping and editing, same event may distributed in a discrete distribution over frames in the video. With this observation, we apply Graph convolution and its forward passing to learn the whole video with a graph model. Frames, shots, events, scenes are treated as nodes of a graph connected with a adjacent matrix. $F2g$ can then be calculated as $F2g = avg(G(F))$ where $F$ is the feature from the pre-trained CNN, $G$ is the GCN function.

Sequence model has been proved to be robust to extract temporal information. As an action is normally shown in consecutive frames in a video, a sequence model can help to obtain a better representation of the videos with actions and help to improve the performance of video search. Comparing to LSTM, GRU has less parameter and is easier to train. In our model, we apply a bi-GRU model. bi-GRU has two GRU models with both direction on temporal dimension. With the input feature $F$, two GRU models can provide two features: $H1 = GRU(F, H)$ and $H2 = GRU(F, H)$. The final output from this model can then be calculated as $F2s = avg([H1, H2])$.

Aggregated model [22, 23] use global feature descriptor for the video which can aggregate evidence over the entire video about both the appearance of the scene and the motion. This is achieved by first dividing the descriptor space different cells using a vocabulary of "action words". Each video descriptor is then assigned to one of the cells and represented by a residual vector recording the difference between the descriptor and the anchor point. Vlad method can view on top of the whole distribution of the training samples and aggregate the key features of all classes. The output is a matrix $V$ , where $k - th$ column $V[\Delta, k]$ represents the aggregated descriptor in the $k - th$ cell. The columns of the matrix are then intra-normalized, stacked, and L2-normalized into a single descriptor $v$ of the entire video. $F2a$ is then calculated as $F2a = avg(V)$.

**Third level embedding** Based on our observation, not all videos need all three models introduced above. A automatic adjusting module is required. In our model, we add a gate convolution network behind each sub-model to control its output. Gate convolution network add a gated linear unit(GLU) [24] as a gate control unit which can be represented as:

$$\begin{aligned} H &= A * sigmoid(B) \\ A &= X * W + b \\ B &= X * V + c \end{aligned} \tag{1}$$

Where H is output of gate convolution,X is the output of each sub-model,W,b,V,c are learned parameters.

Hence,

$$\begin{aligned} F3g &= GatedConvolution(F2g) \\ F3s &= GatedConvolution(F2s) \\ F3a &= GatedConvolution(F2a) \end{aligned} \tag{2}$$

The visual embedding can then be concluded as:

$$\begin{aligned} V_g &= [F1, F2g, F3g] \\ V_s &= [F1, F2s, F3s] \\ V_a &= [F1, F2a, F3a] \end{aligned} \tag{3}$$

#### 2.3.2 Text encoder

The text encoder is similar to the visual encoder as shown in Figure 1. For all text input, a one-hot encoding is applied. Each word's one hot vector is timed by the work embedding matrix to get the dense vector. Word2vec is applied to initialize the embedding matrix [25] provided by [4], which is trained on English tags of 30 million Flickr images. The rest of the model for text feature is similar to visual model and also composed by features from different level $[F1, F2, F3]$.

$$\begin{aligned} S_g &= [F1, F2g, F3g] \\ S_s &= [F1, F2s, F3s] \\ S_a &= [F1, F2a, F3a] \end{aligned} \tag{4}$$

Once the embedding feature from both modal is obtained, common space learning is applied as dual-encoder [9]. A improved marginal ranking loss [3] is used and penalize the model according to the hardest negative examples.

## 3 Training, Results and Runs

In this section, we first introduce the training details of the model in the final submission along with the datasets that are applied to train in this work. We details our understanding of different features related to this task(*i.e.,* visual feature and text feature) and the feature extractors applied on them respectively. The experiment results on msr-vtt dataset is also reported followed by the details of four runs submitted for AVS task.

### 3.1 Training details

In this task, three models are tried along with different combination of them. Based on the test results and our experience during testing. GRU based dual encoder [9] is sensitive to the data with strong sequence information. It performs well when deal with data that the sequences are continuous such as short videos with continuous temporal information. However, most videos in this dataset include discrete visual information, *i.e.,* the visual clues are not evenly distributed on the timeline due to the camera motion, vision block, or video clips. As a result, as shown in Tab. 3, GRU based dual encoder does not show good performance on this dataset comparing to other methods. On the other hand, 'Netvald' focus more on visual content and aggregate them from the whole distribution of the video features. The aggregated vectors includes the views from different perspective of the video features with a big volume. Hence, it contains more information about the video and can achieve best performance in the test on this dataset. Similar to Netvlad, Graph based embedding method does not rely on the temporal information. It focuses on the hierarchical information of the video content. Based on our experience, it performs well on long range videos and has worse result comparing to Netvald as shown in Tab. 3. To make use of the utilities of all these models, we combine them together as a 'Hybrid' model. These three models are combined and controlled by a control gate. As a combined model, it has bigger volume and contains video feature from different clips, with different hierarchies and at different temporal points. This model achieves good performance in this dataset in our experiment. It inspires us to apply a similar combined 'hybrid' model for this AVS task.

For 'Hybrid' model:

$$\begin{aligned} Visual\ Feature &= GRU + Netvlad + DCGN \\ Text\ Feature &= GRU + Netvlad \end{aligned} \tag{5}$$

For 'Hybrid2' model:

$$\begin{aligned} Visual\ Feature &= GRU + Netvlad + DCGN \\ Text\ Feature &= GRU + Netvlad + DCGN \end{aligned} \tag{6}$$

| Hybrid Components | Hyper-parameters |
|---|---|
| GRU | Visual: Hidden size: 1024, Bidirectional |
| | Text: Hidden size: 512, Bidirectional |
| Netvlad | Visual: Cluster size: 32 |
| | Text: Cluster size: 16 |
| DCGN | Visual: Layer nr.: 4, Filter size: 512, Kernel size: 5 |
| | Visual: Layer nr.: 2, Filter size 128, Kernel size: 5 |
| Dual encoder | Visual: Kernel size: 2-3-4-5-6 |
| | Text: Kernel size: 2-3-4 |

Table 1: Training details

### 3.2 Datasets for training

In this section, the datasets applied for training in this competition are introduced. The details of the datasets are listed in Table 2 [26].

| Dataset Name | Clips | Sentences |
|---|---|---|
| MSRVTT | 10k | 200k |
| TGIF | 100k | 120 |
| IACC.3 | 335,944 | 30 |
| TRECVID2016 VTT | 200 | 400 |

Table 2: Training datasets

### 3.3 Visual feature

In this subsection, we include the understanding of video visual features with our visual feature extractor set up.

Deep convolutional neural network(CNN) has been proved to be a sufficient feature extractor for visual content in many computer vision tasks such as object detection [27, 28], segmentation [29, 30], few-shot learning [31, 32], *etc.* Currently, pre-trained networks on ResNet [33], inceptionNet [34], *etc,* are widely applied on different tasks. Current pre-trained networks are mostly pre-trained on ImageNet dataset [35] with 1000 single-labeled classes and 1 million images. Although it is sufficient for many tasks, 1000 classes is still quite limited comparing to millions of classes in the wild which leads to the information loss of the feature extractor when it faces some novel tasks. To solve this problem, fine-tuning is applied to fit the specific task. This method, however, is easy to lead to a over-fitting or under-fitting model because of the volume and complexity of the fine-tuning dataset.

As datasets for videos are all have limited volume and complex information, we observe that it is not sufficient to fine-tune the network on these dataset for initial feature extractor. As a result, in this competition, we applied a large scale dataset [19] to train a ResNet152 network [33] as the feature extractor for the submit model. As a more generalized dataset, it has been proved to fit many computer vision tasks even without fine-tuning [19]. This dataset includes 11,797,630 training images, covering 11,221 categories. Moreover, we include the model trained on labeled scene data and concatenate this feature with the features obtained from the model trained on 11k dataset [19]. Based on the experimental result, this strategy provides a more generalized and robust feature extractor to our task.

### 3.4 Results and Runs

#### 3.4.1 Evaluation on MSRVTT [1]

We evaluated different methods and models on MSRVTT dataset [1] with different metrics *i.e.,* $R@K$(K = 1,5,10), Median rank(Med_r) and mean Average Precision(mAP). $R@K$ is the percentage of test queries for which at least one relevant item is found among the top $K$ retrieved results. Med_r is the median rank of the first relevant item in the search results. Higher $R@K$ and mAP indicates better performance of the model, while the higher Med_r mean, the worse the performance is.

As shown in Tab. 3, four different models are tested on MSRVTT dataset to help us to get the insight of the relation between the performance on video tasks and different models.

| Methods | Rank | $R@1$ | $R@5$ | $R@10$ | Med_r | mAP |
|---|---|---|---|---|---|---|
| GRU [9] | 4 | 7.7 | 22.0 | 31.8 | 32 | 0.155 |
| Netvlad [22] | 2 | 7.6 | 22.3 | 31.9 | 31 | 0.156 |
| Graph [20] | 3 | 7.5 | 21.9 | 31.0 | 34 | 0.153 |
| Hybrid | 1 | 7.8 | 22.5 | 32 | 31 | 0.158 |

Table 3: Results tested on MSRVTT dataset

#### 3.4.2 Submission runs

For TRECVID2019 ad-hoc video search(AVS) task this year, we submit four runs and the result for each run is shown in Tab. 4. As presented in the table, 'Hybrid' model is better than 'Hybrid2' model as text content has more stable temporal sequential information and this information is more important than others. Hence, as shown in the table, GRU based model performs better in text encoding. Based on these tests, the final version of the our model combines two Hybrid models and acheives the best performance in this task.

| Runs | Method | Results |
|---|---|---|
| 1 | Hybrid | 0.161 |
| 2 | (Hybrid + Hybrid2)/2 | 0.163 |
| 3 | GRU | 0.098 |
| 4 | Hybrid2 | 0.157 |

Table 4: Results for four runs in TRECVID 2019 AVS task.

## 4 Conclusion

In this report, we address the problem of large scale video search via text. Methods including classical sequential techniques and state-of-the-art baselines are analyzed based within the ad-hoc video search(AVS) task in TRECVID 2019. We, as team ATL, detailed present the model that achieve the first place in this task in TRECVID 2019, including the architecture of the model and the training details of it. As a result, a hybrid sequence encoder can make use of the utilities of different methods such as GRU, Vlad, GCN, and achieve good performance in this task.

## References


[1] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[2] George Awad, Asad Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, Andrew Delgado, Alan F. Smeaton, Yvette Graham, Wessel Kraaij, and Georges Quénot. Trecvid 2019: An evaluation campaign to benchmark video activity detection, video captioning and matching, and video search & retrieval. In *Proceedings of TRECVID 2019*. NIST, USA, 2019.

[3] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improved visual-semantic embeddings. *arXiv preprint arXiv:1707.05612*, 2(7):8, 2017.

[4] Jianfeng Dong, Xirong Li, and Cees GM Snoek. Predicting visual features from text for image and video caption retrieval. *IEEE Transactions on Multimedia*, 20(12):3377–3388, 2018.

[5] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *European conference on computer vision*, pages 15–29. Springer, 2010.

[6] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013.

[7] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. *arXiv preprint arXiv:1904.01766*, 2019.

[8] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a text-video embedding from incomplete and heterogeneous data. *arXiv preprint arXiv:1804.02516*, 2018.

[9] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9346–9355, 2019.

[10] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*, pages 392–407. Springer, 2014.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[12] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852, 2015.

[13] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015.

[14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[15] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. *arXiv preprint arXiv:1511.06432*, 2015.

[16] Shuai Li, Wanqing Li, Chris Cook, Ce Zhu, and Yanbo Gao. Independently recurrent neural network (indrnn): Building a longer and deeper rnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5457–5466, 2018.

[17] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[18] Xu Lan, Xiatian Zhu, and Shaogang Gong. Knowledge distillation by on-the-fly native ensemble. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7528–7538. Curran Associates Inc., 2018.

[19] Baoyuan Wu, Weidong Chen, Yanbo Fan, Yong Zhang, Jinlong Hou, Junzhou Huang, Wei Liu, and Tong Zhang. Tencent ml-images: A large-scale multi-label image database for visual representation learning. *arXiv preprint arXiv:1901.01703*, 2019.

[20] Feng Mao, Xiang Wu, Hui Xue, and Rong Zhang. Hierarchical video frame sequence representation with deep convolutional graph network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[21] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[22] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. pages 5297–5307, 2016.

[23] Noise learning for weakly supervised segment classification in video. `https://static.googleusercontent.com/media/research.google.com/zh-CN//youtube8m/workshop2019/c_14.pdf`. Accessed: 2019-10-29.

[24] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 933–941. JMLR. org, 2017.

[25] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[26] Ad-hoc video search dataset. `https://github.com/li-xirong/avs`. Accessed: 2019-10-29.

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015.

[28] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 3, 2017.

[29] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2980–2988. IEEE, 2017.

[30] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2018.

[31] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, pages 4077–4087, 2017.

[32] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *ICLR*, 2019.

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE, 2016.

[34] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[35] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE, 2009.