G-TAD Refinement

Anonymous CVPR submission

Paper ID *****

1. Abstract

2. Introduction

As more and more videos are published on online platforms such as facebook, youtube, tik tok, people start paying more attention to video understanding. Popular video understanding tasks include video classification, video captioning, action detection, etc. We are going to work on the temporal action localization (TAL) problem, which is one of the most challenging tasks among them. TAL is to find the starting time and end time of an action from an untrimmed video along with the category of the action.

The input to our algorithm is a sequence of video snippet features extracted from a pre-trained image recognition model. The annotations of each video sequence is a set of N annotations which consists of a set of start time, end time and action class. The temporal action localization task is to predict M possible actions $\Phi = \{\phi_m = (\hat{t}_{s,m}, \hat{t}_{s,m}, \hat{c}_m, p_m)\}$ for each video. The $(\hat{t}_{s,m}, \hat{t}_{s,m})$ are the predicted temporal start and end time of the $m_t h$ predicted action. $\hat{c}_m$ and p_m are the predicted action category and the confidence score. We use G-TAD as the backbone model and refine on top of it. G-TAD model focuses on the localization problem which is to only predict the time window of each action in the videos. The category of each action is predicted by a different model called UntrimmedNets. In G-TAD, the extracted features will be first passed into a graph convolutional network (GCN) model for context feature encoding. Then the generated features will then be passed into Sub-Graph of Interest Alignment (SGAlign) to be rearranged in an embedded space so that it can be used for detection. Lastly, three fully connected layers are applied and outputs classification score and the regression score.

We came up with three ideas which we will try and experiment and see if any of them will make improvement.

1. Fine-tune the parameter in the $K(th)$ Nearest Neighbor(KNN). In G-TAD, the KNN is used to construct the semantics edges. And k was set to be 3. We are going to finetune this hyperparameter.
2. Try to fine tune the Graph Convolutional Network architecture. The Graph Convolutional Network model is used to learn temporal and semantics context.
3. Learn the classification task through the model. The G-TAD model did not conduct the classification tasks. Instead, it used the result from the UNet directly.

In the Technical Approach, we will explain in more details.

3. Related Work

Temporal action localization has been an active research area which is not as mature as other vision tasks. This is partly due to video tasks are generally challenging. The methods to solves this types of problem can be roughly divided into two main types: the traditional methods and the deep learning methods.

3.1. Traditional Methods

Traditionally, the localization algorithm is built upon hand-crafted features using methods such as SIFT (ScaleInvariant Feature Transform) [6], HOG (Histogram of Oriented Gradients) [2] and etc. SIFT algorithm can extract large collection of local feature vectors from an image. And those feature vectors are invariant to image translation, scaling and rotation, and partial invariant to illumination changes as well as local geometric distortion. HOG extracts feature vectors by calculating the occurrence of respective horizontal and vertical gradients in localized portion of an image. And the extracted feature vectors are invariant to geometric and photometric transformations. However, they are variant to object orientation.

3.2. Deep learning Methods

After 2014, deep learning methods became more and more popular. There are two different ways: the two-stage localization method and the one-stage localization method. The two-stage divide the problem into two tasks: temporal action proposal generation and then classification. Popular methods include sliding window [7], temporal actionness

grouping [9], temporal unit regress network [3], boundary sensitive network [5], boundary-matching network [4], etc. The one-stage method on the other hand tried to solve two tasks at the same time.

3.3. Self-supervised learning by cross-modal audio-video clustering [1]

Literature in perceptual studies have shown that visual and audio modalities are highly correlated. This paper leverages the connection to tackle two challenges in action recognition: a) exorbitant cost of manually labeling data, and b) vague definition of suitable label space, by using one modal as a supervisory signal for the other. Specifically, unlabeled data from one modal is passed through an encoder to generate feature vectors, which are then clustered with cluster assignment as pseudo labels. Then the pseudo labels are passed to the other modal to train the encoder of the other modal. This novel approach outperformed state-of-the-art large-scale fully-supervised models for video action recognition. Further, features generated from the learned encoder can also be used for other tasks such as Temporal Action Localization. In particular, the authors compare original G-TAD performance with that using features from their model and observed improvement across the board.

For this project, we do not have time to duplicate the results from this model, but combining audio and video is a promising and interesting direction overall. This also suggests other ways to improve G-TAD, in addition to modifying its architecture, and that is to use better pre-processed input.

3.4. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition [8]

Following the original G-TAD paper, we will use the pretrained video action recognition model from the temporal segment network (TSN) work to extract video snippet features as input. This paper proposed a network to help improve effectiveness for learning video representation over long temporal structure by using deep temporal convolutional neural networks and spatial convolutional neural networks that captures information from the entire video. Also, it tried to address the limited training example issue by leveraging some good practices for video model training, including (1) cross-modality pretraning (2) regularization and (3) enhanced data augmentation. Another strategy this paper used was sparse sampling, which enables end-to-end learning under a time and computing budget

4. Method

The original G-TAD work focused on the localization problem. It proposed a new framework which mainly consists of two parts, the GCNeXt and SGAlign. It combines the temporal context, multi-level semantic context learned using Graph Convolutional network models, and the original input features. And then pass it to the SGAlign to be rearranged in an embedded space so that it can be used for detection.

Specifically, the Input data used by G-TAD is video snippet features extracted from a pre-trained image recognition model. Then, the features of every sigma consecutive frames will be averaged to construct a snippet. It takes the video snippets as nodes in the graph, snippet-snippet correlations as the edges, and actions associated with context as target sub-graphs. The graph convolutional network (GCN) model used has a DeepGCN-like structure. To learn the semantics context, it uses a dynamic semantic graph where edges are learned from the node features and a fixed temporal graph where edges are predefined according to the snippets' temporal order to get the temporal graph. which learns the features of each node by aggregating its context and dynamically updates the edges in the graph.

For the classification task, it used classification results from the UntrimmedNets as inference scores. The UntrimmedNet is a framework to perform weakly supervised action recognition and detection on untrimmed videos without temporal annotations of action labels. One key challenge with video action recognition is that a lot of work has to use manually trimmed video to achieve good results which makes large-scale learning difficult. What's more, annotating the video may be subjective. The UntrimmedNet addressed this but tried to directly predict the class without relying on the annotation end-to-end. Although THUMOS14 includes temporal annotations, the UntrimmedNets was not trained on them (only on the labels). Feature extraction step in this paper is very similar to TSN and the extracted features are the feed thought classification and selection modules to yield video level classification.

After reviewing the original G-TAD work, we have 3 potential areas for refinements.

4.1. Attention

Firstly, we want to explore if there is any opportunity to modify GCNeXt blocks. We added an single headed self attention layer in the GCNeXt blocks. The self-attention layer allows inputs to interact with each other which generates better representation of long sequences with long-range dependencies. We used a single headed self attention layer. The input $X \in \mathbb{R}^{\ell \times d}$ is used to derive the query, value and key:

$$v = Vx_i \;\; i \in \{1, \ldots, \ell\} \tag{1}$$
$$k = Kx_i \;\; i \in \{1, \ldots, \ell\} \tag{2}$$
$$q = Qx_i \;\; i \in \{1, \ldots, \ell\} \tag{3}$$

And we will need to learn the weight matrix V, K, Q.

4.2. Cosine Similarity

Secondly, another area was to rethink the sub graphs, specifically how we can better define semantic edges. By examining the code, we found that G-TAD uses KNN to detect similarity between features. In the G-TAD paper, L2 norm is used to measure the distance. However, there is no proof that why L2 norm is chosen and why it is the best option. In order to check if there is alternative metric which could outperform L2 norm, We explored a different distance measure cosine similarity. Cosine similarity of two non-zero vectors x and y is

$$cos(x, y) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\|\|\vec{y}\|} \tag{4}$$

4.3. K Furthest

Lastly, we would like to see how much KNN really helps with the performance. In the paper, author used the KNN to find the most similar edges. If this is really useful, then, if we use most different edges, we would expect the performance gets affected. Hence, we add an opposite sign to the distance to find the furthest points and expect to see a performance drop. If so, then we are convinced that the original methodology works.

5. Dataset

We plan to use the THUMOS-14 dataset for our project. This dataset contains untrimmed temporal localization video data on 20 human action classes collected from youtube. We will keep the same 200 video validation set and 213 video testing set in order to compare with the original G-TAD performance result. We are also planning to use the same set of features as the original G-TAD paper which were based on a temporal segment network (TSN). The TSN helps us simplify the data preprocessing process by providing a novel framework for video action recognition leveraging long-range temporal structure modeling. Using the pretrained features also significantly reduced our computational cost and save our time for thinking about potential areas of improvement to the model itself.

6. Experiments, Results and Discussion

We examined the provided code for GTAD. In the code two datasets are used, a training set of size 1389 and a validation set of size 1626. Input datapoints are feature vectors extracted from TSN. There is no separate test set. Instead validation set is used as test set to evaluate metrics (mAP). This does raise our concern since normally test set should be a separate dataset completely untouched during training. Further, in the code there is no usage of cross-validation. We have thought about adding cross-validation, but due to time and compute budget, it was not possible

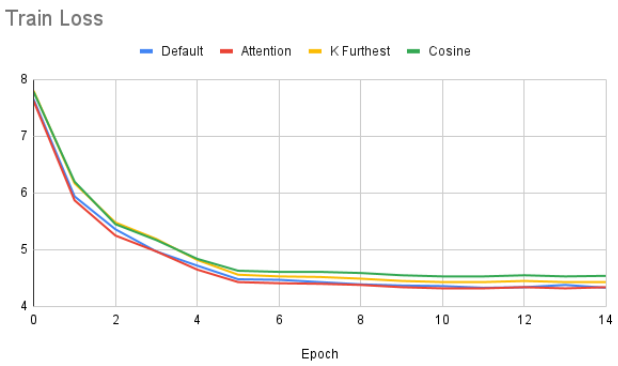

Figure 1

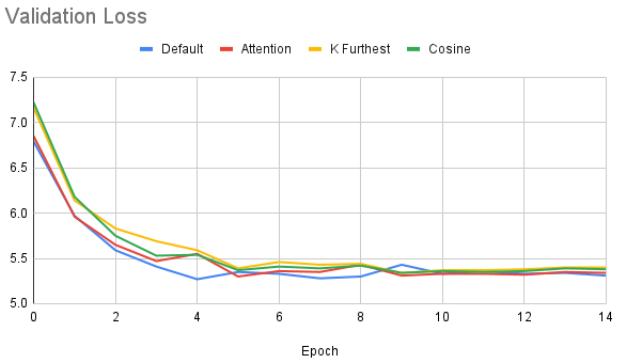

Figure 2

for us to collect a larger dataset and given that the provided training dataset is relatively small, splitting it further significantly degraded the performance when we tried. Hence we decided to follow the original setup of GTAD without modification. Hyper-parameter values such as learning rate are taken from the paper if specified, otherwise we used defaults in the code.

For comparison, we first looked at the training loss and validation loss for each method. Figure 1 shows training loss across different methods, and likewise Figure 2 compares validation loss across methods. Both training loss and validation loss stablized after around 10 epochs. We observed that the training loss and validation loss are very similar across methods. This is especially surprising for the case where we take the furthest instead of closest K neighbors when building the graph. It seems to suggests that the structure of the graph proposed by GTAD is irrelevant in terms of minimizing the loss. This observation could be due to small training data size, but nonetheless it was unexpected and questionable to us.

On the other hand, while the losses are similar, the mAP metric does have slightly different values when using dif-

Threshold	Cosine	Attention	K Furthest	Default
0.3	0.4659543782	0.4805293685	0.4912162624	0.5002606373
0.4	0.3876065338	0.4128165642	0.4307155699	0.4390325922
0.5	0.3056658125	0.3333948466	0.3579568977	0.3650816923
0.6	0.2080119919	0.2378373517	0.2627669732	0.2709449166
0.7	0.1125672113	0.1415298727	0.1514491303	0.1744530723

Figure 3

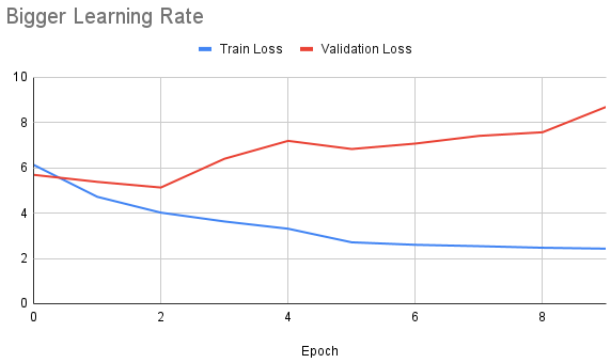

Figure 4

ferent methods, in particular for higher IOU threshold. The results is summarized in Figure 3. This again concerns us as it indicates that the loss may not be a good proxy for mAP metric. In fact if using the default hyper-parameter values from the code (not from the paper), we observed better mAP values (see Table 1) even though the validation loss starts diverging after 1st epoch (see Figure 4). There is no theoretical proof provided by the paper justifying the proposed loss function, and from our experiments we are not completely convinced the effectiveness of chosen loss.

Putting that aside, based on the mAP values, default outperforms all variations overall, but is very close to K-Furthest. Adding attention layer does not help with performance judging from the mAP values, though it could also be due to the fact that we did not have enough budget to train it with more epochs or do hyper-parameter tuning.

We further looked at some specific videos in the test dataset and below are plots of the action intervals predicted under each method against the ground truth.

7. Summary

While GTAD proposed novel ideas and showed improved performance in their paper, when walking through their implementation we had several concerns regarding its methodology. And we were not able to find difference in the performance in different methodologies. For this project we did not have time to investigate further to either corroborate or disprove our conjecture. If we had more time, we would

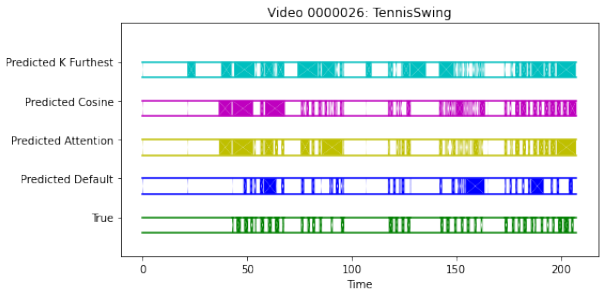

Figure 5

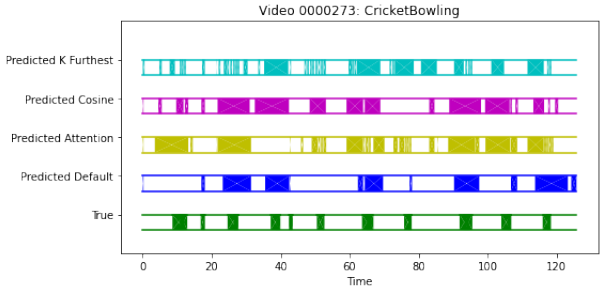

Figure 6

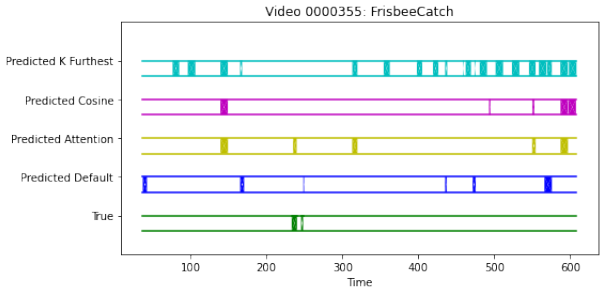

Figure 7

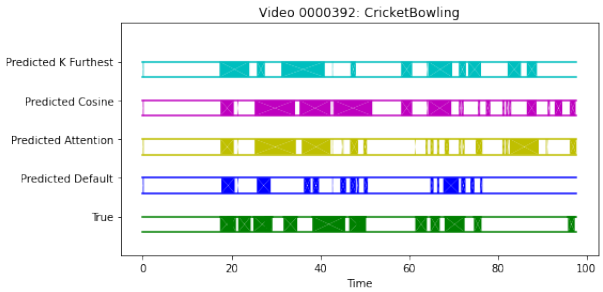

Figure 8

examine code from other papers to obtain a better understanding of methods for temporal action localization and a fair comparison across methods.

Table 1. mAP on THUMOS14 data with G-TAD

IoU Threshold	mAP in our run	mAP in G-TAD paper
0.3	0.581	0.573
0.4	0.519	0.513
0.5	0.432	0.430
0.6	0.334	0.327
0.7	0.236	0.228

8. Contributions and Acknowledgements

The work of this paper leveraged the code from the gtad paper at this link https://github.com/frostinassiky/gtad.

References

[1] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. *Advances in Neural Information Processing Systems*, 33:9758–9770, 2020. 2

[2] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005. 1

[3] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *Proceedings of the IEEE international conference on computer vision*, pages 3628–3636, 2017. 2

[4] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3889–3898, 2019. 2

[5] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 2

[6] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999. 1

[7] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1049–1058, 2016. 1

[8] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 2

[9] Yuanjun Xiong, Yue Zhao, Limin Wang, Dahua Lin, and Xiaoou Tang. A pursuit of temporal accuracy in general activity detection. *arXiv preprint arXiv:1703.02716*, 2017. 2