Refinement of Boundary Regression Using Uncertainty in Temporal Action Localization

Yunze Chen[1,2]
chenyunze2018@ia.ac.cn
Mengjuan Chen[1]
chenmengjuan2016@ia.ac.cn
Rui Wu[3]
ruiwubuaa@gmail.com
Jiagang Zhu[1]
zhujiagang2015@ia.ac.cn
Zheng Zhu[1]
zhengzhu@ieee.org
Qingyi Gu[1]
qingyi.gu@ia.ac.cn

[1] Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences
Beijing, China

[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China

[3] Horizon Robotics
Beijing, China

Abstract

Boundary localization is a key component of most temporal action localization frameworks for untrimmed video. Deep-learning methods have brought remarkable progress in this field due to large-scale annotated datasets (e.g., THUMOS14 and ActivityNet). However, natural ambiguity exists for labeling an accurate action boundaries with such datasets. In this paper, we propose a method to model this uncertainty. Specifically, we construct a Gaussian model for predicting the uncertainty variance of the boundary. The captured variance is further used to select more reliable proposals and to refine proposal boundaries by variance voting during post-processing. For most existing one- and two-stage frameworks, more accurate boundaries and reliable proposals can be obtained without additional computation. For the one-stage decoupled single-shot temporal action detection (Decouple-SSAD) [11] framework, we first apply the adaptive pyramid feature fusion method to fuse its features of different scales and optimize its structure. Then, we introduce the uncertainty based method and improve state-of-the-art mAP@0.5 value from 37.9% to 41.6% on THUMOS14. Moreover, for the two-stage proposalproposal interaction through a graph convolutional network (P-GCN) [33], with such uncertainty method, we also gain significant improvements on both THUMOS14 and ActivityNet v1.3 datasets. Code and more details will be available at https://github.com/shadowclouds/Uty.

1 Introduction

With advanced data acquisition technology, people are enabled to record and share videos through various portable devices. The rapid growth of online video has driven the development of video analysis technology. An important branch is temporal action detection for

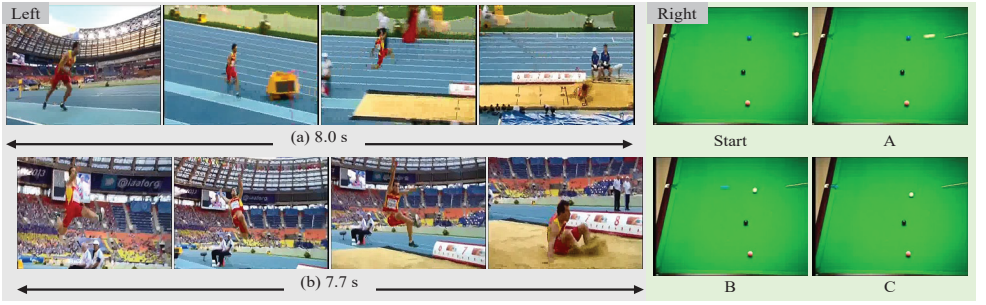


Figure 1: Ambiguous labels collected from THUMOS14. Left: (a) is labeled as Long Jump, where the action instance starts from preparing to run and ends in landing; (b) is a playoff of (a) with the latter part, starts from jumping to the highest point and ends in landing. Whereas, (b) is still labeled as a complete Long Jump as (a). Right: In Billiards, the interval between the start and end of the action (hitting) is only a fraction of a second, and therefore the end boundary is difficult to locate. Frequently, one of A, B, and C is selected as the stop time. A: The white ball moved for a short distance after having been hit by the stick. B: The white ball hits another. C: The white ball stops moving.

untrimmed videos. Temporal action proposal generation and action classification are its two major parts. In the former, the temporal boundary of an action instance is detected, whereas, in the latter, the category of such an instance is recognized. Compared with action classification, temporal action proposal generation is more fundamental and challenging, which has greater impacts on the final performance [1, 4, 18, 32].

As most temporal localization work makes extensive use of deep learning, the quality of the generated proposals output by the network relies heavily on the accuracy of boundary annotations. However, the definition and labeling of temporal boundaries involve ambiguities. For instance, categories are containing diverse action instances, or boundaries difficult to locate. Some ambiguous labels collected from THUMOS14 [13] dataset are shown in Figure 1.

In the left part of Figure 1, ambiguity in action definitions could generate unreliable proposals. For example, suppose we define Long Jump as starting from preparing to run and end in landing, the proposal starts from the highest point and ends in landing like left:(b) would be considered unreliable. However, previous algorithms assess the quality of a proposal based on overlap or classification score, which does not involve its reliability.

Meanwhile, traditional boundary regression loss [8] they use does not consider the ambiguities of labeling boundaries as shown in the right part of Figure 1, which may produce inaccurate boundaries. Thereby, affected by the ambiguity of boundary, unreliable proposals as well as inaccurate boundaries may be produced.

Recently, in object detection, a KL (Kullback-Leibler divergence) regression loss [9] was introduced to learn bounding box transformation and localization variance. However, this method did not consider the reliability of proposals, and it mainly focused on the two-stage object detector. Motivated by [9], we propose an improved KL loss for capturing 1D temporal boundary uncertainty in action localization better, in which the length of proposals varies. By such uncertainty modeling, we could reduce the influence of ambiguous annotations.

Moreover, the captured uncertainty is further used to select reliable proposals and to refine their boundaries during post-processing. Applying the uncertainty method to our pro-

posed one-stage framework two-branch SSAD (Tb-SSAD), and the two-stage framework P-GCN [33], we achieve remarkable improvement in both. The contributions of this paper are summarized as follows:

(1) To the best of our knowledge, we are the first to capture the uncertainty of boundary to reduce the impact of ambiguity labels for temporal action localization in videos. Specially, we introduce an improved KL loss to model uncertainty through a Gaussian distribution. The captured uncertainty is further used to select reliable proposals, and to refine their boundaries.

(2) We modify the one-stage framework [10] and propose a stronger baseline two-branch SSAD (Tb-SSAD) to evaluate our uncertainty method. When introducing such an uncertainty method to our framework Tb-SSAD, we improve the one-stage state-of-the-art mAP performance from 37.9% to 41.6% at a tIoU of 0.5 on THUMOS14. And for the two-stage framework P-GCN, we improve mAP performance from 49.1% to 50.4% at a tIoU of 0.5 on THUMOS14, and from 31.11% to 32.04% on ActivityNet v1.3.

2 Related Work

Action recognition. Action recognition is an important task for video understanding. Typically, a two-stream network [6, 26] learns appearance and motion clues from an RGB image and optical flow, respectively. Whereas the 3D network [23, 29] directly obtains the required messages from raw video sequences. In this paper, we use such pre-trained action models to extract feature sequences from untrimmed videos.

Temporal action detection. Temporal action detection methods can be divided into two categories: two-stage and one-stage. The former first generate action proposals and then classify them [31, 33], which achieve high performance. In contrast, one-stage methods integrate proposal generation and classification into an end-to-end structure, thus achieving higher efficiency. In [11], two branches were proposed: one for regression and another for classification, and a main stream fused with the two branches by hyper-parameters. Both one- and two-stage methods, may be affected by boundary ambiguity, generating unreliable proposals or inaccurate boundaries.

Uncertainty learning in deep neural networks (DNNs). Uncertainty learning involves model and data uncertainty. In the former, the uncertainty of model parameters is estimated using a given training set and can be alleviated by introducing additional training data [7, 12, 15]. This study is based on data uncertainty [16, 17], which originates from inherent noise in the training data, and thus adding additional datasets is meaningless [25]. Previous methods mainly focused on the uncertainty in the image task. For example, the ambiguity of bounding box labeling in object detection [9], and ambiguity caused by blurry pictures in face recognition [25]. For the action localization task, compared with the size of objects in an image, the duration of the action instance in the video varies dramaticallyfrom a fraction of a second to minutes [3], which is more difficult to locate. Meanwhile, there is not only the ambiguity of boundary labeling, but also the ambiguity of action definition as shown in Figure 1.

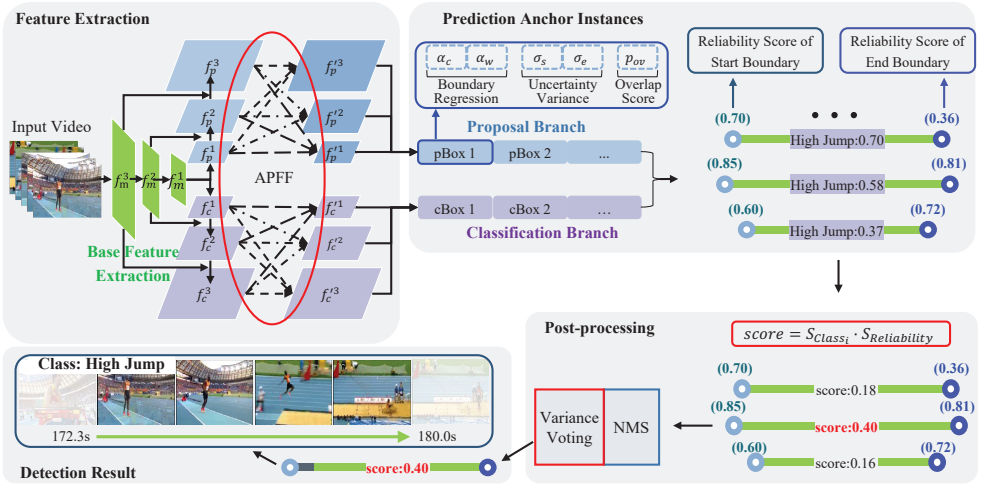


Figure 2: The framework of the proposed network. Given an untrimmed video, (1) pyramid features are generated by base feature extraction. This is followed by an adaptive pyramid feature fusion (APFF) method to generate final features for the proposal and classification branches. (2) Combining outputs of the two branches, we could obtain proposals with reliability score generated by boundary uncertainty variance, and final classification score (e.g. High Jump:0.70) obtained by multiplying the classification and overlap scores. (3) Multiplying the final classification score and reliability score of the start as well as end boundaries, we obtain the final criterion score for selecting proposals in NMS. Meanwhile, we use variance voting to revise the boundaries of the selected proposals.

3 Approach

Herein, we introduce details of our one-stage two-branch SSAD (Tb-SSAD) framework combining with the uncertainty method. As shown in Figure 2, the feature extraction module separately outputs final pyramid feature maps for proposal and classification branches, which contains base feature extraction and adaptive pyramid feature fusion (APFF) method. The prediction anchor instances module generates proposals with the final classification score and reliability scores by combining the output of the proposal and classification branches. Finally, post-processing selects reliable proposals and refines their boundaries to generate the detection result.

3.1 Base Feature Extraction

Action localization aims at detecting action instances in the temporal dimension. Given an input video, we select the two-stream method [26] to extract a temporal feature sequence $F = \{f_{t_n}\}_{n=1}^{T}$, where T is temporal length. Subsequently, following [11], we obtain $M = \{f_m^i\}_{i=1}^{3}$ from F through temporal pooling. With the base feature $M = \{f_m^i\}_{i=1}^{3}$, a base pyramid feature map $\{f_p^i\}_{i=1}^{3}$ for the proposal branch, and another feature map $\{f_c^i\}_{i=1}^{3}$ for

the classification branch are generated through FPN [20]:

$$f_{p,c}^{i}=\begin{cases}\mathcal{C}_1\left(f_m^i\right) & \text{, if } i=3\\ \mathcal{C}_2\left(\frac{1}{3}\mathcal{C}_3\left(f_m^i\right)+\frac{2}{3}\mathcal{S}\left(f_{p,c}^{i+1}\right)\right), & \text{if } 1\le i<3\end{cases} \tag{1}$$

where C1: three Conv-ReLU units, C2: Conv-ReLU-Conv, C3: ReLU-Conv-ReLU, S: kernel size 4, stride 2 for 2x up-sampling. These configurations are the same as in [11].

3.2 Proposal and Classification Branches

Adaptive Pyramid Feature Fusion. With base pyramid features $\{f_p^i\}_{i=1}^3$, $\{f_c^i\}_{i=1}^3$ separately for proposal and classification branches, following [27], we introduce an adaptive pyramid feature fusion (APFF) method for better correlating features of different scales, which could fuse pyramid features of different scales automatically. The APFF method is as follows:

$$f'^{l}=\alpha^l\cdot f^{1\to l}+\beta^l\cdot f^{2\to l}+\gamma^l\cdot f^{3\to l},\quad l=1,2,3 \tag{2}$$

where f'^{l}_{p} and f'^{l}_{c} are the final features for the proposal and classification branch, respectively. The notation $f^{n\to l}$ indicates that the scale of the feature map resizes from n to l. $f^1,f^2,f^3\in\{16,8,4\}\times 1024$. α^l, β^l, and γ^l are learnable weights of different scales for feature fusion. They are obtained from κ_α^l, κ_β^l and κ_γ^l through the softmax function:

$$\alpha^l=\frac{e^{\kappa_\alpha^l}}{e^{\kappa_\alpha^l}+e^{\kappa_\beta^l}+e^{\kappa_\gamma^l}}, \tag{3}$$

where κ_α^l, κ_β^l and κ_γ^l are obtained through 1×1 convolution layers from $f^{1\to l}$, $f^{2\to l}$ and $f^{3\to l}$.

Proposal Branch. The proposal branch is aimed at predicting the location of action instances. As in [24], a series anchors of different scale are generated to locate the default proposal position (a_c,a_w) through temporal pooling, where a_c and a_w are the default center and width, respectively. Using the feature map $f'^{l}_{p},l=1,2,3$ obtained above, each proposal outputs three predictions by 1D convolutions: 1) The proposal regression parameters $(\Delta a_c,\Delta a_w)$, which indicate the offset of the default temporal center and width. 2) The proposal uncertainty parameters (σ_s,σ_e), which denote the predicted variance of the start and end action boundaries. 3) The overlap score p_{ov}, which indicates the intersection-over-union (IoU) score between the proposal and its closest ground-truth segment. Therefore, using the default anchor position (a_c,a_w), the regressed boundaries are:

$$\begin{aligned}x_s&=a_c+\alpha_1 a_w\Delta a_c-\tfrac{1}{2}a_w e^{\alpha_2\Delta a_w}\\ x_e&=a_c+\alpha_1 a_w\Delta a_c+\tfrac{1}{2}a_w e^{\alpha_2\Delta a_w},\end{aligned} \tag{4}$$

where x_s and x_e denote the start and end position of the action, respectively. α_1 and α_2 are hyper-parameters and set to 0.1 in [11].

Classification Branch. The classification branch predicts the scores of C types of actions. For each anchor generated in the proposal branch, the classification branch outputs C action classification score $\mathbf{p}=[p_0,p_1,\ldots,p_C]$, which indicates the probability that the action instance belongs to C categories and one background.

3.3 Uncertainty Modeling

As mentioned above, the boundary ambiguity may generate unreliable proposals and inaccurate boundaries. To reduce the impact of such ambiguity, we need to locate the coordinates of action instances and estimate the uncertainty of boundaries. Specially, we construct a single Gaussian model as follows:

$$P(y^i|x^i) = \frac{1}{\sqrt{2\pi}\sigma^i} e^{-\frac{\left(y^i-x^i\right)^2}{2\left(\sigma^i\right)^2}}, \tag{5}$$

where σ^i indicates the boundary uncertainty variance, and y^i is the corresponding label of the location prediction x^i. Then, we estimate the learnable parameters $\hat{\Theta}$ that maximize the probability $P(y^i|x^i,\Theta)$ over N samples:

$$\hat{\Theta} = \arg\max_{\Theta} \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma^i} e^{-\frac{\left(x^i-y^i\right)^2}{2\left(\sigma^i\right)^2}}. \tag{6}$$

The logarithm of $\hat{\Theta}$ is:

$$\arg\max_{\Theta} \left\{ -\frac{N}{2}\ln 2\pi - \frac{N}{2}\ln\left(\left(\sigma^i\right)^2\right) - \frac{1}{2\left(\sigma^i\right)^2}\sum_{i=1}^{N}\left(x^i-y^i\right)^2 \right\}. \tag{7}$$

The term $\frac{N}{2}\ln 2\pi$ does not depend on the estimated parameters $\hat{\Theta}$. Hence, the Gaussian likelihood estimation can be regarded as a regression loss:

$$L'_{reg} \propto \frac{(x-y)^2}{2\sigma^2} + \frac{1}{2}\ln\left(\sigma^2\right), \tag{8}$$

where x and σ denote the action boundary and uncertainty variance, respectively. During training, we predict the log variance $r = ln(\sigma^2)$ instead of σ to avoid gradient exploding.

We observe that L'_{reg} may be negative for certain small values of σ^2. Because the temporal length of actions varies drastically, this situation happens sometimes and leads to training errors. Therefore, we modify the regression loss as:

$$L_{reg} = \frac{(x-y)^2}{2\sigma^2} + \frac{1}{2}\ln\left(\sigma^2+1\right). \tag{9}$$

This loss function is similar to the KL loss in [9] with Kullback-Leibler divergence. For simplicity, we call it improved KL loss. Generally, the ambiguous annotation only accounts for a small part of the total, so it often causes a large deviation between y and prediction x. During training, relatively large σ^2 is expected for large deviation. Thereby, the loss weight of the ambiguous sample would be reduced by the effect of σ^2.

3.4 Loss Function

The total loss consists of regression, classification, and overlap loss. The improved KL loss proposed above can be written as

$$L_{reg} = \frac{(x_{gs}-x_s)^2}{2\sigma_s^2} + \frac{1}{2}\log\left(\sigma_s^2+1\right) + \frac{(x_{ge}-x_e)^2}{2\sigma_e^2} + \frac{1}{2}\log\left(\sigma_e^2+1\right), \tag{10}$$

where (x_{gs}, x_{ge}) are the start and end boundaries of the ground truth closest to the proposal (x_s, x_e). For classification, the conventional multi-class cross-entropy loss is applied:

$$L_{cls} = -\log\left(\text{softmax}\left(p^k\right)\right), \tag{11}$$

where k denotes the label of the instance. The mean-square-error (MSE) loss is adopted for the overlap loss, and it is used to predict the IoU score:

$$L_{ov} = \left(p_{ov} - g_{ov}\right)^2, \tag{12}$$

where g_{ov} is the IoU between the proposal and its closest ground truth.

During training, the classification branch calculates L_{cls}, whereas the proposal branch provides L_{reg} and L_{ov}. Thus, the total loss can be obtained as

$$L = \delta \cdot L_{reg} + \varepsilon \cdot L_{cls} + \zeta \cdot L_{ov}, \tag{13}$$

where δ, ε, and ζ are hyper-parameters, and we set 0.5, 1, and 10 for the three, which are used for balancing three losses to a similar magnitude.

3.5 Post-processing

As proposal reliability cannot be obtained, the traditional localization process multiplies the classification and overlap scores as a detection criterion score for non-maximum suppression (NMS), where unreliable proposals with high classification or overlap scores may be selected. However, the proposed framework outputs the uncertainty variance (σ_s, σ_e), which denotes proposal reliability. During post-processing, (σ_s, σ_e) could be considered in conjunction with the classification and overlap scores for selecting more reliable proposals. The modified detection criterion S is

$$S = p_{ov} \times \max\left(p\right) \times S_{\text{Reliability}}, \tag{14}$$

where $\max\left(p\right)$ is the maximum classification score. The reliability score is as follows:

$$S_{\text{Reliability}} = \left(1 - g(\sigma_s^2)\right) \times \left(1 - g(\sigma_e^2)\right), g(\sigma^2) = \frac{e^{\sigma^2}}{\sum e^{\sigma^2}}. \tag{15}$$

where $g(\cdot)$ is a softmax function, used to normalize the reliability score. $\sum e^{\sigma^2}$ denotes the index summation of all input proposals' uncertainty variance. By adding reliability score to the detection criterion, more reliable proposals would be selected.

Moreover, we apply variance voting (var voting) [9] to produce more accurate boundaries. Specifically, we calculate the coordinates of proposals using neighbor locations weighted by the predicted uncertainty variance. Thereby, proposals with lower uncertainty variance and higher IoU are given larger weights as follows:

$$\begin{aligned} p_i &= e^{-(1-\text{Io}U(b_i,b))^2/\sigma_t} \\ x &= \frac{\sum_i p_i x_i / \sigma_{x,i}^2}{\sum_i p_i / \sigma_{x,i}^2}, \quad \text{subject to } IoU\left(b_i, b\right) > 0, \end{aligned} \tag{16}$$

where x_i denotes the coordinates of the temporal boundary, and σ_t is a hyper-parameter, which is 0.025 in this work. Combining the reliability score and variance voting with NMS, we could obtain more reliable proposals and precise boundaries.

4 Experiments

4.1 Experimental Settings

Datasets. We use two challenging datasets in the experiments: THUMOS14 [13] and ActivityNet v1.3 [10]. The former contains 13320 trimmed videos of 101 categories in the training set UCF-101 [28]. For the temporal action detection task, the training set is commonly used for pre-training, in which only 20 categories are temporarily labeled and involved. Besides, 200 validation and 213 test videos with temporal annotation are used for training and testing, respectively. ActivityNet [10] is another large-scale dataset used for action localization. It contains 10,000 training videos and 5,000 validation videos. There are 200 different action categories in ActivityNet v1.3. Following [33], we train the proposed framework on the training set and test it on the validation videos.

Implementations. The configuration of our Tb-SSAD is almost the same as Decouple-SSAD [11] for fair comparison. The number of training epochs is 30, with a batch size of 48, and the learning rate is 10^{-4}. The base feature $M = \{f_m^i\}_{i=1}^3 \in \{16,8,4\} \times 1024$, separately. And the dimensions of three feature maps generated by APFF for proposal and classification branches are the same as M.

As the results of Decouple-SSAD (De-SSAD) [11] on ActivityNet have not been published, we verify our uncertainty method using the two-stage framework P-GCN on ActivityNet. On THUMOS14, we combine the proposed methods with both Tb-SSAD and P-GCN to perform a comprehensive verification.

4.2 Comparison with State-of-the-art Methods

Table 1 shows the results of some state-of-the-art methods on THUMOS14, with tIoU ranging from 0.3 to 0.7. Comparable performance is achieved by combining Tb-SSAD with our uncertainty method. The proposed framework achieves 41.6% at a tIoU of 0.5, which leads to a gain of 3.7% and 1.8% compared with the advanced one-stage GTAN and two-stage DBG, respectively. Compared with our base Tb-SSAD, the highest gain corresponds to a tIoU of 0.5, where the mean Average Precision (mAP) performance of Tb-SSAD is improved from 38.0% to 41.6%.

For two-stage method P-GCN with I3D [2] backbone, which inputs proposals generated by BSN [18]. Performance at a tIoU of 0.5 gains from 49.1% to 50.4% on THUMOS14. As shown in Table 2, for ActivityNet v1.3, the baseline we reproduce is 26.90%, almost

One-stage Method	0.3	0.4	0.5	0.6	0.7
GTAN [22]	**56.9**	46.5	37.9	-	-
De-SSAD [11]	49.9	44.4	35.8	24.3	13.6
Tb-SSAD	49.9	45.5	38.0	28.0	16.5
Tb-SSAD+Uty	52.7	**48.1**	**41.6**	**31.5**	**19.3**
De-SSAD(Kinetics)	60.2	54.1	44.2	32.3	19.1
Tb-SSAD(Kinetics)	61.2	56.3	47.8	34.7	22.2
De-SSAD+Uty(Kinetics)	60.3	55.6	48.2	35.9	22.0
Tb-SSAD+Uty(Kinetics)	**64.3**	**58.9**	**49.9**	**38.0**	**24.2**

Two-stage Method	0.3	0.4	0.5	0.6	0.7
BSN [18]	53.5	45.0	36.9	28.4	20.0
MGG [21]	53.9	46.8	37.4	29.5	21.3
BMN [19]	56.0	47.4	38.8	29.7	20.5
DBG [5]	57.8	49.4	39.8	30.2	21.7
P-GCN [33]	63.6	57.8	49.1	-	-
P-GCN +**Uty**	**66.3**	**59.8**	**50.4**	**37.5**	**23.5**

Table 1: Action localization results on THUMOS14, measured by mAP(%) at different tIoU thresholds α, varies from 0.3 to 0.7. Uty represents our uncertainty method. Left: The performance of one-stage method. (Kinetics) denotes network pre-trained on Kinetics [14] dataset. Right: The performance of two-stage.

the same as 26.99% in P-GCN [33]. Compared with our baseline, we lead 1.75% gains from 26.90% to 28.65% by adding our uncertainty method. Whereas, when combined with video-level labels from UntrimmedNet [30] as well as BSN localization score [18], our base is 1.11% lower than the benchmark of P-GCN* in [33]. Based on our baseline 30.00% on average, we improve the performance to 32.04%, from 1.74% to 4.35% at tIoU of 0.95, and only 0.24% at a tIoU of 0.5. Commonly, our method achieves better performance at a higher tIoU threshold, where our method focuses on selecting more accurate proposals and revising their boundaries.

4.3 Ablation Study

On the left side of Table 3, we introduce three frameworks to evaluate the effectiveness of our uncertainty method and the APFF: Tb-SSAD that only contains two branches without APFF, Decouple-SSAD (De-SSAD) [11], and main stream (main) in Decouple-SSAD [11].
Uncertainty Method. De-SSAD and Tb-SSAD exhibit an improvement of 4.9% and 3.5%, respectively, by adding our uncertainty method. The main stream (main) increases by 1.8%, which is slightly lower than the improvement of the other two. This observation accords with the conclusion of Decouple-SSAD [11] that applying classification and location in a single branch may affect detection accuracy.

On the right side of Table 3, the effectiveness of three parts that consist of our uncertainty method are evaluated separately, which are improved KL loss, variance voting (var voting), and reliability score. It is seen that improved KL loss results in the greatest performance improvement, with a gain of approximately 2.5%. Var voting and reliability score lead to a gain of 1.1% in total (from 40.5% to 41.6% for mAP@0.5). The former revises the temporal boundary, whereas the latter takes the reliability of proposals into final selecting scores. Visible examples are shown in Figure 3.
Adaptive Pyramid Feature Fusion. Regarding APFF, the two-branch backbone Tb-SSAD leads to a gain of 5.7%, which is approximately twice as much as the improvement by a single main stream. However, the performance of De-SSAD drops from 35.8% to 35.5% by APFF. This is due to the conflict between APFF and the hyper-parameter that is used for fusing the two branches with its main stream. Therefore, we remove the main stream in our Tb-SSAD to make the model more concise.

5 Conclusion

In this paper, we propose an uncertainty method, which consists of improved KL loss, reliability score, and variance voting, for reducing the affects of boundary ambiguity. Our method is combined with both one- and two-stage networks, thereby improving the performance of

Method	0.5	0.75	0.95	Average
SSN [34]	39.12	23.48	**5.49**	23.98
P-GCN	42.90	28.14	2.47	26.99
ours(base)	43.94	28.80	1.63	26.90
ours(base)+Uty	**44.65**	**29.29**	4.23	**28.65**

Method(*)	0.5	0.75	0.95	Average
BSN*	46.45	29.96	8.02	30.03
P-GCN*	**48.26**	33.16	3.27	31.11
ours*(base)	47.93	32.57	1.74	30.00
ours*(base)+Uty	48.17	**33.79**	**4.35**	**32.04**

Table 2: Action localization results on the validation set of ActivityNet v1.3. The average mAP calculated by tIoU thresholds ranges from 0.5 to 0.95. Uty represents our uncertainty method. Left: The performance of methods without any external label. Right: methods combine with external video labels from UntrimmedNet (*) [30].

Figure 3: Detection results on THUMOS14 with the effects of variance voting and reliability score in post-processing. Top: Variance voting can revise proposals with inaccurate boundaries. Bottom: The reliability score can avoid selecting inaccurate proposals with a higher classification score. Without adding the reliability score, the inaccurate proposal with a CleanAndJerk score of 0.59 is selected, whereas the other two are filtered in NMS.

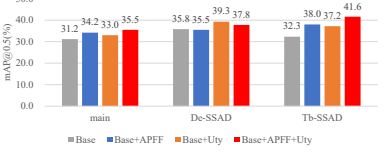
one-stage Tb-SSAD from 38.0% to 41.6%, and that of the two-stage P-GCN from 49.1% to 50.4% for mAP@0.5 on THUMOS14. On ActivityNet, P-GCN combined with the proposed method achieved an improvement from 30.00% to 32.04% on average mAP.

Acknowledgement

This work is supported in part by the National Natural Science Foundation of China under Grants 61673376. We would like to thank Tianwei Lin in Baidu for the help in completing this paper and Runhao Zeng in South China University of Technology for helpful discussions.

References

[1] Humam Alwassel, Fabian Caba Heilbron, Victor Escorcia, and Bernard Ghanem. Diagnosing error in temporal action detectors. In *2018 Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.



improved KL loss	×	✓	✓	✓	✓
Var voting	×	×	✓	×	✓
Reliability score	×	×	×	✓	✓
mAP@0.5 (%)	38.0	40.5	41.0	41.3	**41.6**

Table 3: Ablation results on THUMOS14 dataset. Left: Performance gain by adding our uncertainty method and APFF to three frameworks. Right: Effectiveness of each component in the uncertainty method based on Tb-SSAD.

[2] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017.

[3] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A. Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1130–1139, 2018.

[4] Peihao Chen, Chuang Gan, Guangyao Shen, Wenbing Huang, Runhao Zeng, and Mingkui Tan. Relation attention for temporal action localization. *IEEE Transactions on Multimedia*, 2019.

[5] Yabiao Wang Ying Tai Donghao Luo Zhipeng Cui Chengjie Wang Jilin Li Feiyue Huang Rongrong Ji Chuming Lin*, Jian Li*. Fast learning of temporal action proposal via dense boundary generator. In *AAAI Conference on Artificial Intelligence*, 2020.

[6] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1933–1941, 2016.

[7] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2015.

[8] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jagannath Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[9] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2883–2892, 2018.

[10] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970, 2015.

[11] Yupan Huang, Qi Dai, and Yutong Lu. Decoupling localization and classification in single shot temporal action detection. *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1288–1293, 2019.

[12] Shuya Isobe and Shuichi Arai. Deep convolutional encoder-decoder network with model uncertainty for semantic segmentation. *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pages 365–370, 2017.

[13] Yu-Gang Jiang, Jingen Liu, A Roshan Zamir, George Toderici, Ivan Laptev, Mubarak Shah, and Rahul Sukthankar. Thumos challenge: Action recognition with a large number of classes, 2014.

[14] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[15] 2017 Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *BMVC*, 2017.

[16] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NIPS*, 2017.

[17] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

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

[19] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. *2019 IEEE International Conference on Computer Vision (ICCV)*, pages 3888–3897, 2019.

[20] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017.

[21] Yuan Liu, Lin Ma, Yifeng Zhang, Wei Liu, and Shih-Fu Chang. Multi-granularity generator for temporal action proposal. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3599–3608, 2018.

[22] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 344–353, 2019.

[23] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5534–5542, 2017.

[24] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *ArXiv*, abs/1804.02767, 2018.

[25] Yichun Shi, Anil K. Jain, and Nathan D. Kalka. Probabilistic face embeddings. *2019 IEEE International Conference on Computer Vision (ICCV)*, pages 6901–6910, 2019.

[26] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[27] Di Huang Songtao Liu and Yunhong Wang. Learning spatial fusion for single-shot object detection. 2019.

[28] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[29] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2014.

[30] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6402–6411, 2017.

[31] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5794–5803, 2017.

[32] Runhao Zeng, Chuang Gan, Peihao Chen, Wenbing Huang, Qingyao Wu, and Mingkui Tan. Breaking winner-takes-all: Iterative-winners-out networks for weakly supervised temporal action localization. *IEEE Transactions on Image Processing*, 28(12):5797–5808, 2019.

[33] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. *2019 IEEE International Conference on Computer Vision (ICCV)*, pages 7093–7102, 2019.

[34] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. *2017 International Journal of Computer Vision*, 128:74 – 95, 2017.