Coarse to Fine Multi-Resolution Temporal Convolution Network

Dipika et al.

Abstract

- TCN : for video segmentation
- Problem : over-segmentation error→なめらかさ&時間的一貫性を確保したい
- Insights : １新しいtemporal encoder-decoderの提案

  →追加の洗練化moduleの必要性を回避

  ：２ multiresolution feature-augumentation strategyで学習を強化

  →様々な時間解像度に対応

  ：３ 誤分類にペナルティを与えるaction lossの提案

Introduction

- 従来の時系列segmentationの標準framework：

  MS-TCN：順伝搬に従い拡大するtemporal cinvolutions
- 従来：追加のmodel学習・後処理の平滑化でMS-TCNを補強

↓

- 今回：新しいencoder-decoder C2F-TCNの提案 = coarse-to-fine ensemble of decoder

  従来のは強くない(decoder のデザイン&単純なbottleneckのため)

1. Action Loss：frame-levelではなくvideo-levelのloss→over-segmentation 減
2. Multi-resolution feature augmentation：ビデオシーケンスに特化した特徴量レベルでの拡張？
3. よい予測の不確かさを持たせる:間違った予測に対するover-confidenceを防ぐ

Introduction

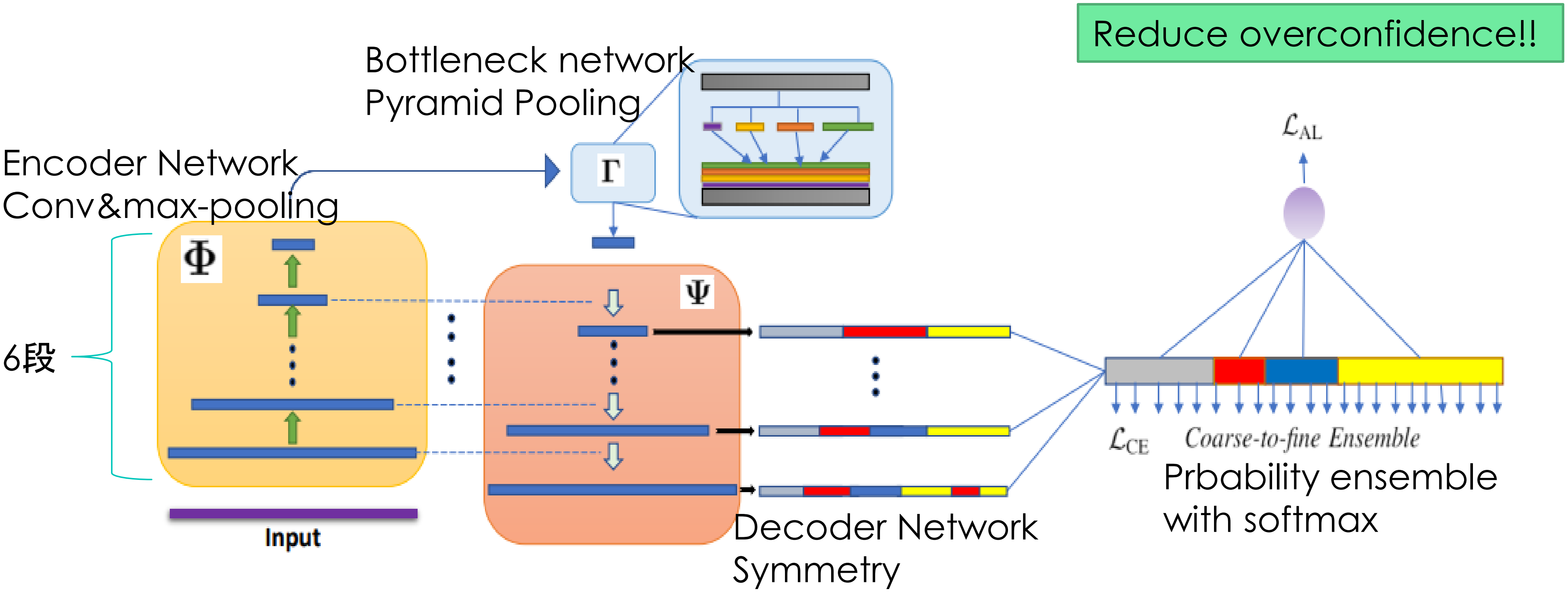
- Insights

1. 新しいencoder-decoder C2F-TCNの提案 : more calibrated, less fragmented, and accurate
2. multi-resolution temporal feature-level augmentation strategyの提案 : more accuracy
3. Action Lossの提案 : 誤分類に対するペナルティ

Related Work(未読)

- Action Recognition
- Action Segmentation

Methodology


Φ
Γ
Ψ
Input
$\mathcal{L}_{\mathrm{AL}}$
$\mathcal{L}_{\mathrm{CE}}$
Coarse-to-fine Ensemble

Reduce overconfidence!!

Bottleneck network
Pyramid Pooling

Encoder Network
Conv&max-pooling

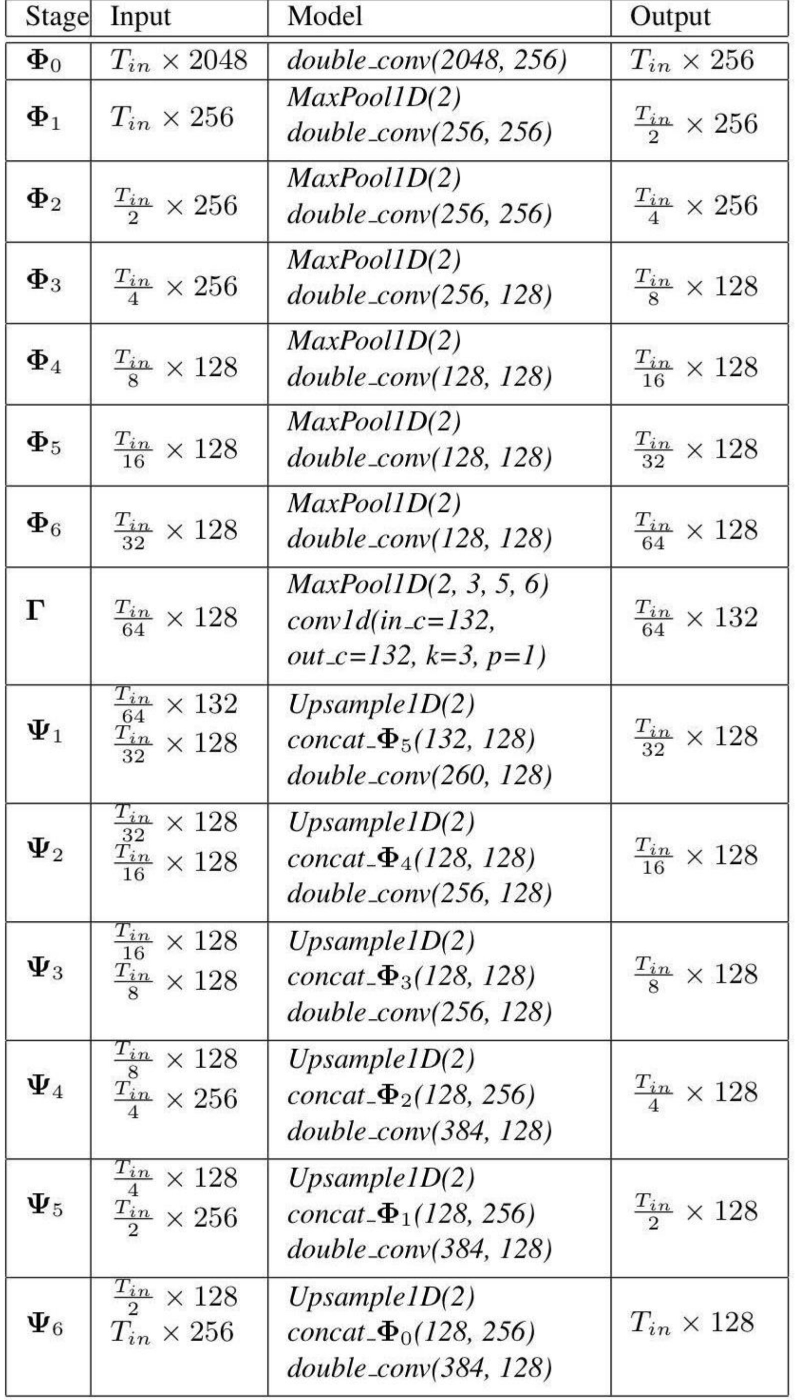
6段

Decoder Network
Symmetry

Prbability ensemble
with softmax

Figure 1. **Our segmentation architecture:** Depiction of the architecture of our model (**Φ : Γ : Ψ**). We utilize our multi-resolution features to produce *Coarse-to-fine Ensemble* predictions.

Encoder-Decoder Architecture M = (Φ, Γ, Ψ)



Stage	Input	Model	Output
$\mathbf{\Phi}_0$	$T_{in} \times 2048$	*double_conv(2048, 256)*	$T_{in} \times 256$

$\mathbf{\Phi}_1$	$T_{in} \times 256$	*MaxPool1D(2)*
		double_conv(256, 256)
			$\frac{T_{in}}{2} \times 256$

$\mathbf{\Phi}_2$	$\frac{T_{in}}{2} \times 256$	*MaxPool1D(2)*
		double_conv(256, 256)
			$\frac{T_{in}}{4} \times 256$

$\mathbf{\Phi}_3$	$\frac{T_{in}}{4} \times 256$	*MaxPool1D(2)*
		double_conv(256, 128)
			$\frac{T_{in}}{8} \times 128$

$\mathbf{\Phi}_4$	$\frac{T_{in}}{8} \times 128$	*MaxPool1D(2)*
		double_conv(128, 128)
			$\frac{T_{in}}{16} \times 128$

$\mathbf{\Phi}_5$	$\frac{T_{in}}{16} \times 128$	*MaxPool1D(2)*
		double_conv(128, 128)
			$\frac{T_{in}}{32} \times 128$

$\mathbf{\Phi}_6$	$\frac{T_{in}}{32} \times 128$	*MaxPool1D(2)*
		double_conv(128, 128)
			$\frac{T_{in}}{64} \times 128$

$\mathbf{\Gamma}$	$\frac{T_{in}}{64} \times 128$	*MaxPool1D(2, 3, 5, 6)*
		conv1d(in_c=132, out_c=132, k=3, p=1)
			$\frac{T_{in}}{64} \times 132$

$\mathbf{\Psi}_1$	$\frac{T_{in}}{64} \times 132$
	$\frac{T_{in}}{32} \times 128$
		Upsample1D(2)
		concat_*$\mathbf{\Phi}_5$(132, 128)*
		double_conv(260, 128)
			$\frac{T_{in}}{32} \times 128$

$\mathbf{\Psi}_2$	$\frac{T_{in}}{32} \times 128$
	$\frac{T_{in}}{16} \times 128$
		Upsample1D(2)
		concat_*$\mathbf{\Phi}_4$(128, 128)*
		double_conv(256, 128)
			$\frac{T_{in}}{16} \times 128$

$\mathbf{\Psi}_3$	$\frac{T_{in}}{16} \times 128$
	$\frac{T_{in}}{8} \times 128$
		Upsample1D(2)
		concat_*$\mathbf{\Phi}_3$(128, 128)*
		double_conv(256, 128)
			$\frac{T_{in}}{8} \times 128$

$\mathbf{\Psi}_4$	$\frac{T_{in}}{8} \times 128$
	$\frac{T_{in}}{4} \times 256$
		Upsample1D(2)
		concat_*$\mathbf{\Phi}_2$(128, 256)*
		double_conv(384, 128)
			$\frac{T_{in}}{4} \times 128$

$\mathbf{\Psi}_5$	$\frac{T_{in}}{4} \times 128$
	$\frac{T_{in}}{2} \times 256$
		Upsample1D(2)
		concat_*$\mathbf{\Phi}_1$(128, 256)*
		double_conv(384, 128)
			$\frac{T_{in}}{2} \times 128$

$\mathbf{\Psi}_6$	$\frac{T_{in}}{2} \times 128$
	$T_{in} \times 256$
		Upsample1D(2)
		concat_*$\mathbf{\Phi}_0$(128, 256)*
		double_conv(384, 128)
			$T_{in} \times 128$

Table 6: Encoder-Decoder Architecture M = ($\mathbf{\Phi}$, $\mathbf{\Gamma}$, $\mathbf{\Psi}$)

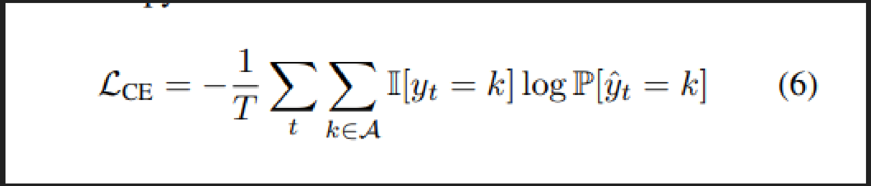
Multi-Resolution Feature Augmentation

- Not yet

Trining Loss

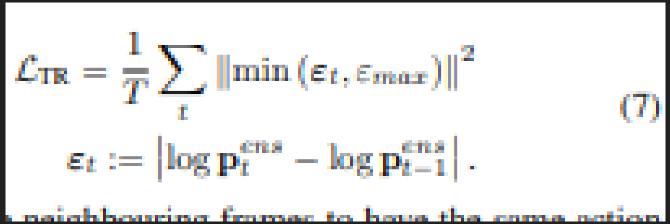
- 学習過程で使う3つのloss関数

1. Frame-level 交差エントロピー関数



$$\mathcal{L}_{\text{CE}} = -\frac{1}{T}\sum_{t}\sum_{k\in\mathcal{A}} \mathbb{I}[y_t = k] \log \mathbb{P}[\hat{y}_t = k] \qquad (6)$$

2. Transition loss

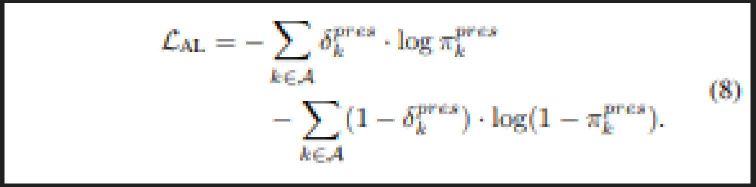


$$\mathcal{L}_{\text{TR}} = \frac{1}{T}\sum_{t} \left\|\min\left(\varepsilon_t, \varepsilon_{max}\right)\right\|^2$$

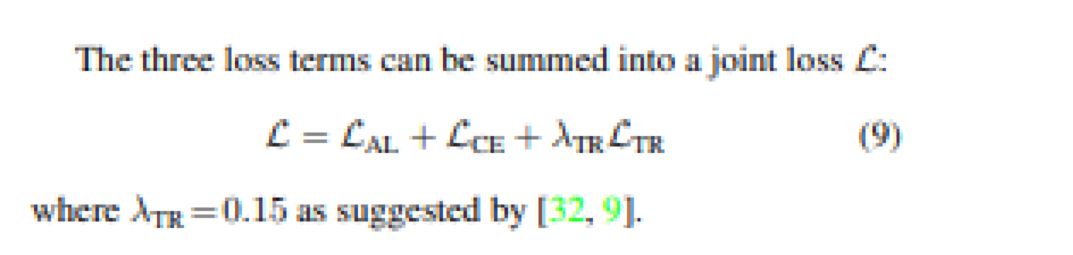
$$\varepsilon_t := \left|\log \mathbf{p}_t^{cns} - \log \mathbf{p}_{t-1}^{cns}\right|. \qquad (7)$$

neighbouring frames to have the same action

3. Video-level Action loss(複雑な行動の正しい判定)



$$\mathcal{L}_{\text{AL}} = -\sum_{k\in\mathcal{A}} \delta_k^{pres} \cdot \log \pi_k^{pres} - \sum_{k\in\mathcal{A}} (1-\delta_k^{pres}) \cdot \log(1-\pi_k^{pres}). \qquad (8)$$



The three loss terms can be summed into a joint loss $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_{\text{AL}} + \mathcal{L}_{\text{CE}} + \lambda_{\text{TR}}\mathcal{L}_{\text{TR}} \qquad (9)$$

where $\lambda_{\text{TR}} = 0.15$ as suggested by [32, 9].

- Complex Activity Recognition
- Calibration : over/under confidence の測定

$\hat{p}_t \in [0, 1]$:

$\hat{y}_t = \arg\max \mathbf{p}_t.$: 予測値のconfidence

Experiments

- Adam optimizer for 600 epochs
- Ensemble weight a = 1/4

Breakfast Actions

- 学習率(10^-4)
- 重み(3*10^-3)
- バッチサイズ(100)
- Base window(10)

50Salads

- 学習率(3*10^-4)
- 重み(10^-3)
- バッチサイズ(25)
- Base window(20)

GTEA

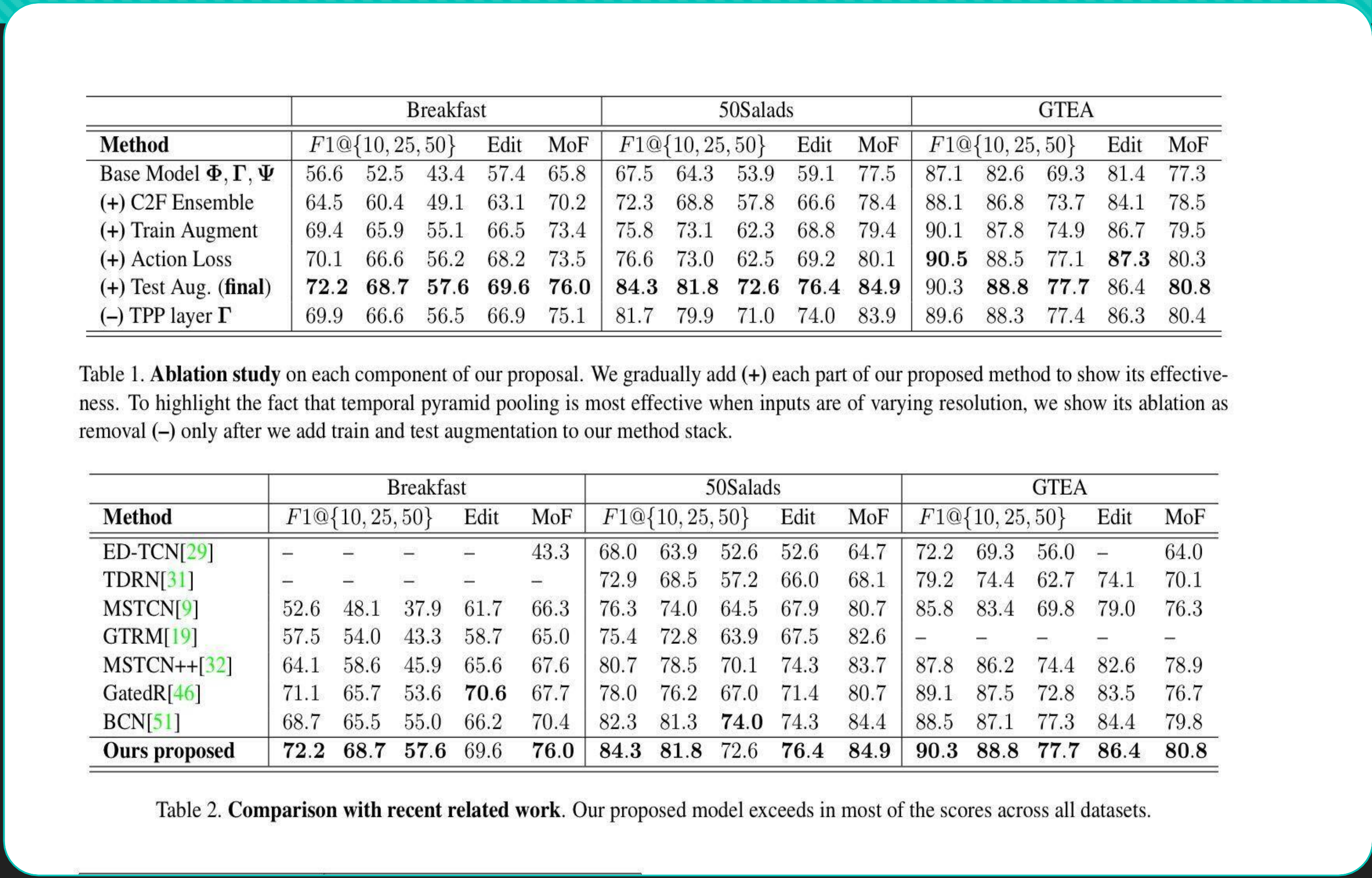
- 学習率(5*10^-4)
- 重み(3*10^-4)
- バッチサイズ(11)
- Base window(4)

Evaluation

- Mean-over-frames(MoF)
- Segment-wise edit distance(Edit)
- F1-scores with IOU

Ablation Studies & Model Analysis

- Base modelに段階的に要素を足していったらどんどん結果が向上



	Breakfast					50Salads					GTEA				
Method	$F1@\{10, 25, 50\}$			Edit	MoF	$F1@\{10, 25, 50\}$			Edit	MoF	$F1@\{10, 25, 50\}$			Edit	MoF
Base Model $\mathbf{\Phi}, \mathbf{\Gamma}, \mathbf{\Psi}$	56.6	52.5	43.4	57.4	65.8	67.5	64.3	53.9	59.1	77.5	87.1	82.6	69.3	81.4	77.3
(+) C2F Ensemble	64.5	60.4	49.1	63.1	70.2	72.3	68.8	57.8	66.6	78.4	88.1	86.8	73.7	84.1	78.5
(+) Train Augment	69.4	65.9	55.1	66.5	73.4	75.8	73.1	62.3	68.8	79.4	90.1	87.8	74.9	86.7	79.5
(+) Action Loss	70.1	66.6	56.2	68.2	73.5	76.6	73.0	62.5	69.2	80.1	**90.5**	88.5	77.1	**87.3**	80.3
(+) Test Aug. (**final**)	**72.2**	**68.7**	**57.6**	**69.6**	**76.0**	**84.3**	**81.8**	**72.6**	**76.4**	**84.9**	90.3	**88.8**	**77.7**	86.4	**80.8**
(–) TPP layer $\mathbf{\Gamma}$	69.9	66.6	56.5	66.9	75.1	81.7	79.9	71.0	74.0	83.9	89.6	88.3	77.4	86.3	80.4

Table 1. **Ablation study** on each component of our proposal. We gradually add **(+)** each part of our proposed method to show its effectiveness. To highlight the fact that temporal pyramid pooling is most effective when inputs are of varying resolution, we show its ablation as removal **(–)** only after we add train and test augmentation to our method stack.

	Breakfast					50Salads					GTEA				
Method	$F1@\{10, 25, 50\}$			Edit	MoF	$F1@\{10, 25, 50\}$			Edit	MoF	$F1@\{10, 25, 50\}$			Edit	MoF
ED-TCN[29]	–	–	–	–	43.3	68.0	63.9	52.6	52.6	64.7	72.2	69.3	56.0	–	64.0
TDRN[31]	–	–	–	–	–	72.9	68.5	57.2	66.0	68.1	79.2	74.4	62.7	74.1	70.1
MSTCN[9]	52.6	48.1	37.9	61.7	66.3	76.3	74.0	64.5	67.9	80.7	85.8	83.4	69.8	79.0	76.3
GTRM[19]	57.5	54.0	43.3	58.7	65.0	75.4	72.8	63.9	67.5	82.6	–	–	–	–	–
MSTCN++[32]	64.1	58.6	45.9	65.6	67.6	80.7	78.5	70.1	74.3	83.7	87.8	86.2	74.4	82.6	78.9
GatedR[46]	71.1	65.7	53.6	**70.6**	67.7	78.0	76.2	67.0	71.4	80.7	89.1	87.5	72.8	83.5	76.7
BCN[51]	68.7	65.5	55.0	66.2	70.4	82.3	81.3	**74.0**	74.3	84.4	88.5	87.1	77.3	84.4	79.8
Ours proposed	**72.2**	**68.7**	**57.6**	69.6	**76.0**	**84.3**	**81.8**	72.6	**76.4**	**84.9**	**90.3**	**88.8**	**77.7**	**86.4**	**80.8**

Table 2. **Comparison with recent related work**. Our proposed model exceeds in most of the scores across all datasets.