Learning Motion in Feature Space: Locally-Consistent Deformable Convolution Networks for Fine-Grained Action Detection

Khoi-Nguyen C. Mac[1]

Dhiraj Joshi[2], Raymond A. Yeh[1], Jinjun Xiong[2], Rogerio S. Feris[2], Minh N. Do[1]

[1]University of Illinois at Urbana-Champaign, [2]IBM Research AI

ILLINOIS

C3SR center for cognitive computing systems research

IBM

ICCV 2019 Seoul, Korea

Outline

Introduction

Fine-grained Actions

- Actions with high inter-class similarity [5, 6]
- Difficult to distinguish two different actions just from observing individual frames
- Heavily rely on motion, rather than mostly on appearance cues

(a) cut cheese (b) cut lettuce (c) place cheese into bowl

Figure 1: Examples of some fine-grained actions (frames) in 50 Salads dataset.

Fine-grained Action Detection Pipeline

- (Fine-grained) Action detection: given a video of action sequence, determine where an action segment starts/ends and categorize that action
- **Step one:** Spatio-temporal feature extraction (short-term)
  - Analyze *a few consecutive frames*
  - Traditional approaches: appearance stream (RGB) and motion stream (optical flow, IDT, MHI, *etc*.)
  - Our **focus**
- **Step two:** Long-temporal modeling
  - Models long-term dependency of *the whole video*
  - Using *extracted* short-term spatio-temporal features

Observation

- Two-stream approaches are computationally expensive (optical flow and multi-stream inference)
- Motion extracted by optical flow in pixel space suffers from noise [3, 4]
- Deformable convolution is flexible [1]
    - Adaptive receptive fields can focus on *important regions* in a frame → Motivates tracking important motion
    - Traditional optical flow tracks *all possible motion* (some are not necessary)

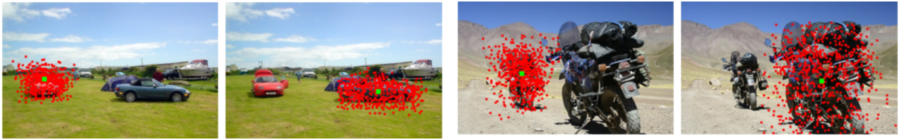


Figure 2: Adaptive receptive fields (red dots) of deformable convolutions w.r.t. activation units (green dots) [1].

Proposed Approach

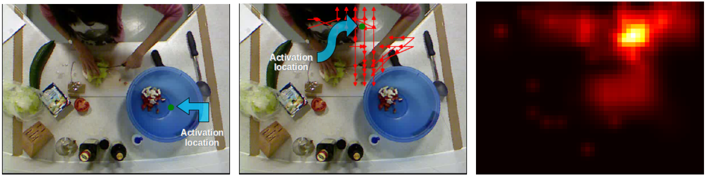
We propose: **Locally-Consistent Deformable Convolution** (LCDC)

- Learn temporal information in the feature space
- Exploit the property of adaptive receptive fields to extract motion of important regions
- Jointly model spatial and temporal components (single stream) effectively and efficiently with local coherency constraint
- As a byproduct, the framework produces rich spatio-temporal features for long-temporal models

Approach

More Observations

(a) frame at time $t-1$. (b) frame at time t. (c) masks of the person.



(d) no motion vectors found. (e) motion vectors found. (f) visualization of motion.

Figure 3: Visualization of difference of adaptive receptive fields for action *cutting lettuce* in 50 Salads dataset.

Network Architecture - Overview

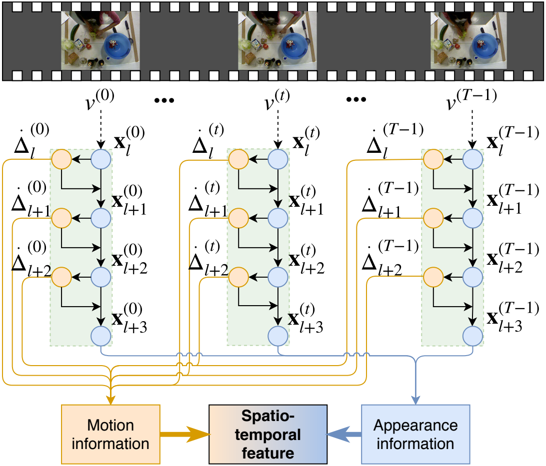


Figure 4: Network architecture of our proposed framework across multiple frames.

Deformable Convolutions

Standard convolutions

$$\mathbf{y}[n] = \sum_k \mathbf{w}[-k]\mathbf{x}\,[n+k]\,, \tag{1}$$

Deformable convolutions

$$\mathbf{y}[n] = \sum_k \mathbf{w}[-k]\mathbf{x}\left(n+k+\ddot{\Delta}_{n,k}\right), \tag{2}$$

- $\mathbf{w} \in \mathbb{R}^K$: convolutional kernel
- $n \in \mathbb{Z}^N$ and $k \in \mathbb{Z}^K$: signal and kernel indices (multi-dimensional)
- $\ddot{\Delta} \in \mathbb{R}^{N \times K}$: deformation offsets ($\ddot{\Delta}_{n,k} = (\mathbf{h}_k * \mathbf{x})[n]$)
- $(\cdot)$: index that requires interpolation ($\ddot{\Delta}_{n,k}$ is fractional)

Modeling Motion with Adaptive Receptive Fields

Adaptive receptive field at time t

$$\ddot{\mathbf{F}}^{(t)} \in \mathbb{R}^{N \times K} \qquad \text{where} \quad \ddot{\mathbf{F}}^{(t)}_{n,k} = n + k + \ddot{\Delta}^{(t)}_{n,k}, \tag{3}$$

Temporal modelling

$$\ddot{\mathbf{r}}^{(t)} = \ddot{\mathbf{F}}^{(t)} - \ddot{\mathbf{F}}^{(t-1)} = \ddot{\Delta}^{(t)} - \ddot{\Delta}^{(t-1)}. \tag{4}$$

$\ddot{\mathbf{r}}^{(t)} \neq 0$ only for deformable convolutions

Property: Given T input feature maps (spatial dimension $H \times W$), we can create

- T different $\ddot{\Delta}^{(t)}|_{t=0}^{T-1}$
- $T - 1$ motion fields $\ddot{\mathbf{r}}^{(t)}|_{t=0}^{T-2}$ with *the same* spatial dimension

Thus, we can model different motion at different positions n and time t.

Illustration of Difference of Receptive Fields

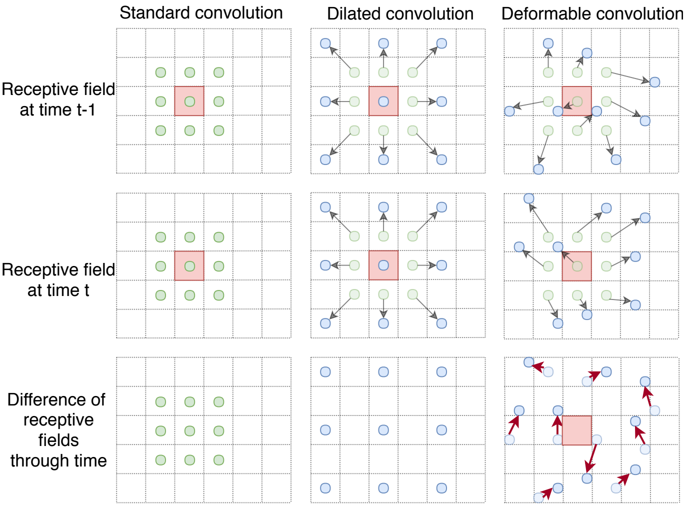
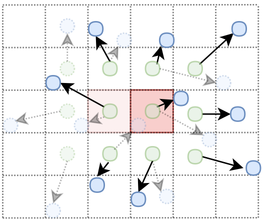


Figure 5: Temporal information modeled by the difference of receptive fields at a single location.

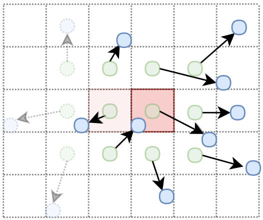
Consistency of $\ddot{\mathbf{r}}$

No guarantee of local consistency in receptive fields

- $\ddot{\Delta}_{n,k}$ corresponds to $\mathbf{x}[n+k] = \mathbf{x}[m]$
- Multiple ways to decompose m, *i.e.* $m = n + k = (n-l) + (k+l)$, for any l
- Therefore, one single $x[m]$ is deformed by multiple $\ddot{\Delta}_{n-l,k+l}$, with different l



Deformable convolution



Locally-consistent deformable convolution

Figure 6: Illustration of receptive fields at two consecutive locations (faded and solid red squares) in 2D at time t, with and without local coherency constraint.

Locally-Consistent Deformable Convolution

Locally-consistent deformable convolution (**LCDC**):

$$\mathbf{y}[n] = \sum_k \mathbf{w}[-k]\mathbf{x}\left(n + k + \dot{\Delta}_{n+k}\right). \tag{5}$$

for $\dot{\Delta} \in \mathbb{R}^N$. LCDC is a special case of deformable convolution where

$$\ddot{\Delta}_{n,k} = \dot{\Delta}_{n+k}, \quad \forall n, k. \tag{6}$$

We name this condition as **local coherency constraint**.

Interpretation of LCDC

Instead of deforming the receptive field as in Eq. (2), we can deform the input signal

$$\mathbf{y}[n] = \sum_k \mathbf{w}[-k]\tilde{\mathbf{x}}[n+k] = (\tilde{\mathbf{x}} * \mathbf{w})[n], \quad (7)$$

where

$$\tilde{\mathbf{x}}[n] = (D_{\dot{\Delta}}\{\mathbf{x}\})[n] = \mathbf{x}\left(n + \dot{\Delta}_n\right) \quad (8)$$

is a deformed version of $\mathbf{x}$ and $D_{\dot{\Delta}}\{\cdot\}$ is defined as the deforming operation by offset $\dot{\Delta}$

How to Produce $\dot{\Delta}$?

Recall that $\ddot{\Delta} \in \mathbb{R}^{N \times K}$ is learned via a convolution layer, *i.e.*

$$\ddot{\Delta}_{n,k} = (\mathbf{h}_k * \mathbf{x})[n] \tag{9}$$

Similarly, $\dot{\Delta} \in \mathbb{R}^{N}$ can also be learned via a convolution layer, *i.e.*

$$\dot{\Delta}_{n} = (\Phi * \mathbf{x})[n] \tag{10}$$

Property of $\ddot{\Delta}$ is carried over, *i.e.* $\dot{\Delta}$ can also model motion at different positions n and times t

Efficiency of LCDC

- $\dot{\Delta} \in \mathbb{R}^N$ only needs a kernel Φ, while $\ddot{\Delta} \in \mathbb{R}^{N \times K}$ requires K $\mathbf{h}_k|_{k=0}^{K-1}$
- Implementation-wise, given input feature map $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$
  - $\ddot{\Delta} \in \mathbb{R}^{(H \times W) \times (G \times K_h \times K_w \times 2)}$
  - $\dot{\Delta} \in \mathbb{R}^{H \times W \times 2}$
    - H and W: height and width of inputs
    - G: number of deformable groups
    - K_h and K_w: height and width of kernels
    - 2: offsets are 2D vectors
  - The reduction is $G \times K_h \times K_w$; proportional to the number of deformable convolution layers

Effectiveness of LCDC

LCDC can effectively model both appearance and motion information in a single-stream network

- Spatial information: $\mathbf{y} = (D_{\dot{\Delta}}\{\mathbf{x}\}) * \mathbf{w}$
- Temporal information: $\dot{\mathbf{r}}^{(t)} = \dot{\Delta}^{(t)} - \dot{\Delta}^{(t-1)}$ has a behavior equivalent to motion information produced by optical flow.

Proposition

Suppose that two inputs $\mathbf{x}^{(t-1)}$ *and* $\mathbf{x}^{(t)}$ *are related through a motion field, i.e.*

$$\mathbf{x}^{(t)}(s) = \mathbf{x}^{(t-1)}\left(s - o(s)\right), \tag{11}$$

where $o(s)$ *is the motion at location* $s \in \mathbb{R}^2$, *and* $\mathbf{x}^{(t)}$ *is assumed to be locally varying. Then the corresponding LCDC outputs with* $\mathbf{w} \neq 0$:

$$\mathbf{y}^{(t)} = (D_{\dot{\Delta}^{(t)}}\{\mathbf{x}^{(t)}\}) * \mathbf{w},$$
$$\mathbf{y}^{(t-1)} = (D_{\dot{\Delta}^{(t-1)}}\{\mathbf{x}^{(t-1)}\}) * \mathbf{w}$$

are consistent, i.e. $\mathbf{y}^{(t-1)} = \mathbf{y}^{(t)}$, *if and ony if* $\forall n$,

$$\dot{\mathbf{r}}_n^{(t)} = \dot{\Delta}_n^{(t)} - \dot{\Delta}_n^{(t-1)} = o\left(n + \dot{\Delta}_n^{(t)}\right). \tag{12}$$

Notice that in **pixel space**, $\mathbf{x}$ are input images and $o(s)$ is the optical flow at s. In **latent space**, $\mathbf{x}$ are intermediate feature maps and $o(s)$ is the motion of feature.

Proof.

With the connection of LCDC to standard convolution, under the assumption that $\mathbf{w} \neq 0$, we have:

$$\begin{aligned} \mathbf{y}^{(t)} &= \mathbf{y}^{(t-1)} \\ \Leftrightarrow D_{\dot{\Delta}^{(t)}}\{\mathbf{x}^{(t)}\} &= D_{\dot{\Delta}^{(t-1)}}\{\mathbf{x}^{(t-1)}\} \\ \Leftrightarrow \mathbf{x}^{(t)}\left(n + \dot{\Delta}_n^{(t)}\right) &= \mathbf{x}^{(t-1)}\left(n + \dot{\Delta}_n^{(t-1)}\right), \forall n. \end{aligned}$$

Substituting the LHS in the motion relation in Eq. (11), we obtain the following equivalent conditions $\forall n$:

$$\begin{aligned} \mathbf{x}^{(t-1)}\left(n + \dot{\Delta}_n^{(t)} - o(n + \dot{\Delta}_n^{(t)})\right) &= \mathbf{x}^{(t-1)}\left(n + \dot{\Delta}_n^{(t-1)}\right) \\ \Leftrightarrow \dot{\Delta}_n^{(t)} - o(n + \dot{\Delta}_n^{(t)}) &= \dot{\Delta}_n^{(t-1)} \\ \Leftrightarrow o\left(n + \dot{\Delta}_n^{(t)}\right) &= \dot{\Delta}_n^{(t)} - \dot{\Delta}_n^{(t-1)} = \dot{\mathbf{r}}_n^{(t)}. \end{aligned}$$

(since $\mathbf{x}^{(t)}$ is locally varying). □

Spatio-temporal Features

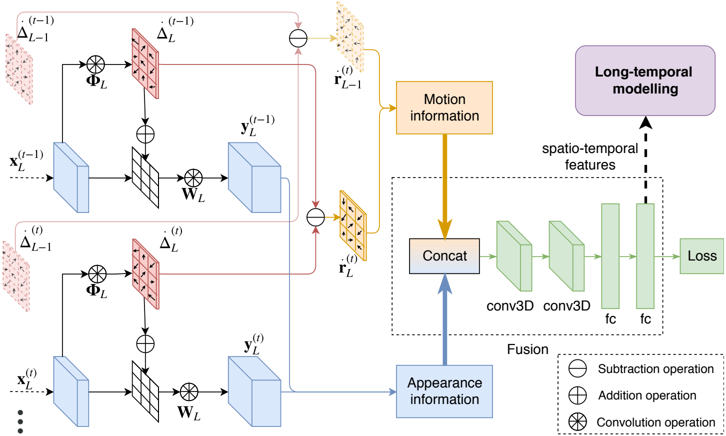


Figure 7: A more detailed view of our network architecture with the fusion module.

Experimental Results

Datasets

- **50 Salads Dataset** [7]: 50 salad making videos (5-10 minutes) with different granularity levels: *mid* (17 action classes) and *eval* level (9 action classes)
- **Georgia Tech Egocentric Activities (GTEA)** [2]: 28 videos (1 minute long) of 7 action classes. The camera in this dataset is head-mounted.

Baselines

- SpatialCNN [4]:
  - VGG-like model; learns both spatial and *short-term* temporal information
  - Spatial components: a RGB frame
  - Temporal components: corresponding MHI (the difference between frames over a *short* period of time)
- ST-CNN [4], DilatedTCN [3], and ED-TCN [3]:
  - Long-temporal modeling frameworks
  - ST-CNN: uses a single 1D convolution
  - DilatedTCN: stacked dilated convolutions
  - ED-TCN: encoder (pooling) and decoder (upsampling by repetition) framework

Metrics

- **Frame-wise accuracy**: evaluates whether a frame is correctly classified or not. Does not consider the temporal structure of the output.
- **Segmental edit score**: takes into account this problem by penalizing over-segmentation. It evaluates the ordering of actions without following specific timings.
- **F1@k score**[3]: also penalizes over-segmentation but ignores small time-shifting between the prediction and ground-truth.

	Model	Spatial comp	Temporal comp (short)	Long-temporal	F1@10	Edit	Acc
Mid	SpatialCNN [15]	RGB	MHI	-	32.3	24.8	54.9
	(SpatialCNN) + ST-CNN [15]	RGB	MHI	1D-Conv	55.9	45.9	59.4
	(SpatialCNN) + DilatedTCN [14]	RGB	MHI	DilatedTCN	52.2	43.1	59.3
	(SpatialCNN) + ED-TCN [14]	RGB	MHI	ED-TCN	68.0	59.8	64.7
	(SpatialCNN) + TDRN [16]	RGB	MHI	TDRN	(72.9)	(66.0)	(68.1)
	LCDC	RGB	Learned deformation	-	43.99	33.38	67.27
	LCDC + ST-CNN	RGB	Learned deformation	1D-Conv	60.01±0.42	51.35±0.12	68.45±0.15
	LCDC + DilatedTCN	RGB	Learned deformation	DilatedTCN	58.21±0.59	48.54±0.52	69.28±0.25
	LCDC + ED-TCN	RGB	Learned deformation	ED-TCN	**73.75±0.54**	**66.94±1.33**	**72.12±0.41**
Eval	Spatial CNN [15]	RGB	MHI	-	35.0	25.5	68.0
	(SpatialCNN) + ST-CNN [15]	RGB	MHI	1D-Conv	61.7	52.8	71.3
	(SpatialCNN) + DilatedTCN [14]	RGB	MHI	DilatedTCN	55.8	46.9	71.1
	(SpatialCNN) + ED-TCN [14]	RGB	MHI	ED-TCN	76.5	72.2	73.4
	LCDC	RGB	Learned deformation	-	56.56	45.77	77.59
	LCDC + ST-CNN	RGB	Learned deformation	1D-Conv	70.46±0.41	62.71±0.46	77.84±0.26
	LCDC + DilatedTCN	RGB	Learned deformation	DilatedTCN	67.59±0.42	58.97±0.55	78.29±0.29
	LCDC + ED-TCN	RGB	Learned deformation	ED-TCN	**80.22±0.21**	**74.56±0.70**	**78.90±0.25**

Table 1: Results on 50 salads dataset (*mid* and *eval*-level).

Model	Spatial comp	Temporal comp (short)	Long-temporal	F1@10	Edit	Acc
SpatialCNN [15]	RGB	MHI	-	41.8	-	54.1
(SpatialCNN) + ST-CNN [15]	RGB	MHI	1D-Conv	58.7	-	60.6
(SpatialCNN) + DilatedTCN [14]	RGB	MHI	DilatedTCN	58.8	-	58.3
(SpatialCNN) + ED-TCN [14]	RGB	MHI	ED-TCN	72.2	-	64.0
(SpatialCNN) + TDRN [16]	RGB	MHI	TDRN	(79.2)	(74.1)	(70.1)
LCDC	RGB	Learned deformation	-	52.42	45.38	55.32
LCDC + ST-CNN	RGB	Learned deformation	1D-Conv	62.23±0.69	55.75±0.94	58.36±0.45
LCDC + DilatedTCN	RGB	Learned deformation	DilatedTCN	62.08±0.85	55.13±0.79	58.07±0.30
LCDC + ED-TCN	RGB	Learned deformation	ED-TCN	**75.39±1.33**	**72.84±0.84**	**65.34±0.54**

Table 2: Results on GTEA dataset.

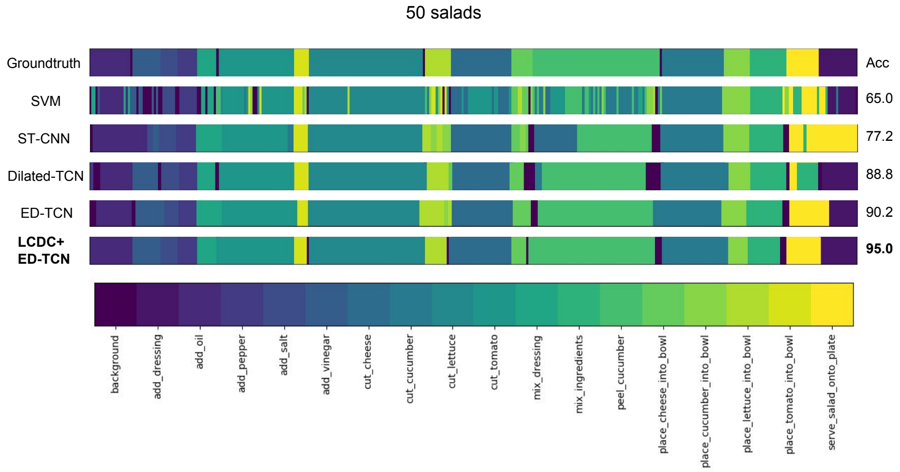


Figure 8: Comparison of segmentation results across different methods on a test video from 50 Salads dataset (*mid*-level).

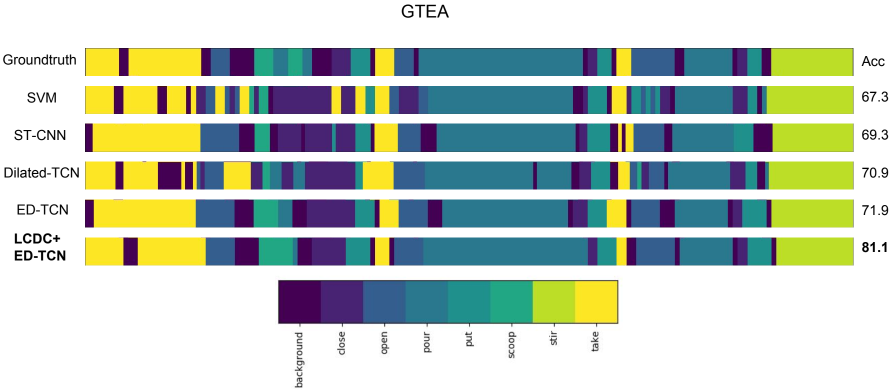


Figure 9: Comparison of segmentation results across different methods on a test video from GTEA dataset.

Ablation Study

- **SpatialCNN:** The features from [4], inputs are stacked RGB frame and MHI.
- **NaiveAppear:** Frame-wise class prediction using ResNet50 (no temporal information involved in this setup).
- **NaiveTempAppear:** Appearance stream with multiple input frames and ResNet50 backbone.
- **OptFlowMotion:** Motion stream that models temporal component using VGG-16.
- **TwoStreamNet:** The two-stream framework obtained by averaging scores from *NaiveTempAppear* and *OptFlowMotion*.
- **DC:** Deformable convolution network (ResNet50) (without local coherency constraint).
- **LCDC:** Our proposed approach.

Model	Spatial comp	Temporal comp (short)	Fusion scheme	Acc	Total params	Deform params
SpatialCNN	RGB (single)	MHI (multi)	Stacked inputs	60.99	-	-
NaiveAppear	RGB (single)	-	-	68.45	38.9M	-
NaiveTempAppear	RGB (multi)	Avg feat frames (multi)	-	71.52	38.9M	-
OptFlowMotion	-	OptFlow (multi)	-	25.67	134.1M	-
TwoStreamNet	RGB (multi)	OptFlow (multi)	Avg scores	71.82	173.0M	-
DC	RGB (multi)	Learned deformation (w/o local coherency) (multi)	3D-Conv	72.25	45.7M	995.5K
LCDC	RGB (multi)	Learned deformation (multi)	3D-Conv	**73.77**	**42.7M**	**27.7K**

Figure 10: Ablation study on 50 Salads dataset (Split 1, mid-level). “Single” and “multi” indicate the amount of input frames for spatial/temporal components.

Conclusion

We propose to model motion in feature space

To do so effectively, we introduce Locally-Consistent Deformable Convolution

Thank you for your attention

References I

[1] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei.
Deformable convolutional networks.
In *International Conference on Computer Vision (ICCV)*, 2017.

[2] Alireza Fathi, Xiaofeng Ren, and James M. Rehg.
Learning to recognize objects in egocentric activities.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[3] Colin Lea, Michael D. Flynn, René Vidal, Austin Reiter, and Gregory D. Hager.
Temporal convolutional networks for action segmentation and detection.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

References II

[4] Colin Lea, Austin Reiter, René Vidal, and Gregory D. Hager.
Segmental spatiotemporal CNNs for fine-grained action segmentation.
In *European Conference on Computer Vision (ECCV)*, 2016.

[5] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele.
A database for fine grained activity detection of cooking activities.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[6] Bharat Singh, Tim K. Marks, Michael Jones, Oncel Tuzel, and Ming Shao.
A multi-stream bi-directional recurrent neural network for fine-grained action detection.
In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

References III

[7] Sebastian Stein and Stephen J. McKenna.

Combining embedded accelerometers with computer vision for recognizing food preparation activities.

In *International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2013.