Learning Video Representations from Correspondence Proposals

Xingyu Liu[1], Joon-Young Lee[2], Hailin Jin[2]
[1]Stanford University [2]Adobe Research

Code

Motivation

Correspondences between positions – Dynamic Component of Videos. Its pattern different than regular structured data:

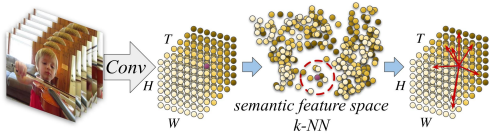
1. Correspondence have similar visual or semantic features.
Assumption underlying many computer vision tasks, such as image matching and flow estimation.

2. Correspondence can span arbitrarily long spatiotemporal ranges.
Spatially: fast motion/low frame rate; temporally: disappear and then re-appear in videos across a long time.

3. Potential correspondence in other frames are small in percentage.
Given a position, usually only small portion of positions in other frames can be potential correspondence. Other dissimilar positions can be safely ignored.

Challenges: sparsity, irregularity, feature space similarity

Our solution: Learn from Correspondence Proposals



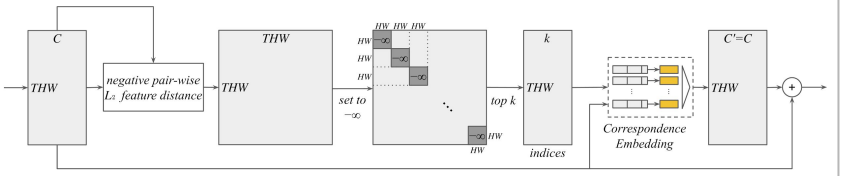
1. For each position, the **potential correspondence** are the ***k*-NN** in video representation tensor in feature space.
2. Each of the k feature pairs is processed **identically** and **independently** by a neural network (instantiated by **MLP**).
3. Then **max pooling** is applied to select the strongest response.

Output is the encoded representation of correspondence, i.e. dynamic component in videos.

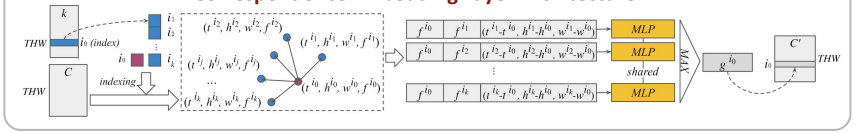
Suppose j-th k-NN of $(t^{i_0}, h^{i_0}, w^{i_0}, f^{i_0})$ is $(t^{i_j}, h^{i_j}, w^{i_j}, f^{i_j})$, $(t^{i}, h^{i}, w^{i}$ normalize to $[0, 1])$, then

$$g^{i_0} = \underset{j \in \{1,2,\ldots,k\}}{MAX} \{\zeta(f^{i_0}, f^{i_j}, t^{i_j} - t^{i_0}, h^{i_j} - h^{i_0}, w^{i_j} - w^{i_0})\}$$

CP Module Architecture



Correspondence Embedding Layer Architecture



Motion-centric Dataset Results

Jester

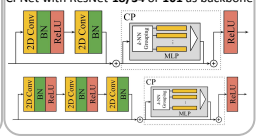
model	params (M)	val	test
BesNet [11]	37.8	-	94.23
MultiScale TRN [39]	22.8	95.31	94.78
TPRN [35]	22.0	95.40	95.34
MFNet [20]	41.1	96.68	96.22
MFF [19]	43.4	96.33	96.28
C2D Res34 baseline	20.3	84.73	-
CPNet Res34, 5 CP **(ours)**	**21.0**	**96.70**	**96.56**

Something-Something

model	params (M)	val top-1	val top-5	test top-1	test top-5
Goyal et al. [12]	22.2	51.33	80.46	50.76	80.77
MultiScale TRN [39]	22.8	48.80	77.64	50.85	79.33
Two-stream TRN [39]	46.4	55.52	83.06	56.24	83.15
C2D Res18 baseline	10.7	35.24	64.49	-	-
C2D Res34 baseline	20.3	39.64	69.61	-	-
CPNet Res18, 5 CP **(ours)**	**11.3**	54.08	82.10	53.31	81.00
CPNet Res34, 5 CP **(ours)**	**21.0**	**57.65**	**83.95**	**57.57**	**84.26**

Overall Architecture

CPNet with ResNet-**18/34** or **101** as backbone



Ablation Studies

(a) number of CP modules

model	top-1	top-5
C2D	56.9	79.5
1 CP	60.3	82.4
2 CPs	60.4	82.4
4 CPs	61.0	**83.1**
6 CPs	**61.1**	**83.1**

(b) Ablation on CP module's k values used in training and testing time.

top-1/top-5 accuracy		test $k=1$	test $k=2$	test $k=4$	test $k=8$	test $k=16$	test $k=32$
train	$k=1$	**59.9/82.3**	59.2/81.6	56.6/79.4	52.5/76.1	49.0/72.6	44.6/58.5
	$k=2$	59.1/81.8	**60.2/82.5**	59.6/81.8	56.9/80.1	53.0/77.1	48.9/73.5
	$k=4$	59.0/81.2	60.2/82.4	**60.5/82.6**	59.0/81.7	55.3/79.2	49.2/73.5
	$k=8$	53.4/76.3	56.8/79.5	59.6/81.9	**60.7/82.8**	59.7/82.1	57.0/80.3
	$k=16$	51.3/75.1	53.8/77.3	56.8/79.7	59.8/82.1	**60.6/82.8**	59.2/81.8
	$k=32$	52.6/76.6	53.8/77.7	55.5/79.1	58.2/80.8	60.0/82.2	**60.4/82.4**

(c) CP module positions

model	top-1	top-5
C2D	56.9	79.5
res_3	60.4	82.4
res_4	**60.8**	**82.8**
res_5	59.2	81.6

Kinetics-400 Results

ResNet-18 Model

model	params (M)	top-1	top-5
I3D Inception [3]	25.0	72.1	90.3
Inception-ResNet-v2 [2]	50.9	73.0	90.9
NL C2D ResNet-101 [33]	48.2	75.1	91.7
CPNet C2D ResNet-101 **(ours)**	**42.1**	**75.3**	**92.4**

ResNet-101 model

frame rate	1/12 of original frame rate				1/4 of original frame rate			
val configuration	1-clip, 1 crop		25-clip, 10 crops		1-clip, 1 crop		25-clip, 10 crops	
accuracy	top-1	top-5	top-1	top-5	top-1	top-5	top-1	top-5
C2D	56.9	79.5	61.3	83.6	54.1	77.4	60.8	83.3
C3D [28]	58.3	80.7	64.4	85.8	55.0	78.5	63.3	85.2
NL C2D Net [33]	58.6	81.3	63.3	85.1	55.3	78.6	62.1	84.2
ARTNet [32]	59.1	81.1	65.1	86.1	56.1	78.7	64.2	85.6
CPNet (Ours)	**61.1**	**83.1**	**66.3**	**87.1**	**57.2**	**80.8**	**64.9**	**86.5**

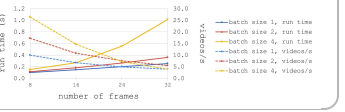
Toy Example: A Failing of several previous methods

- White 2×2 block moving on 32×32 black canvas, step size random between 7 and 9
  - Four labels of moving directions: *up*, *down*, *left*, *right*
  - Two layers of 3×3 conv allowed
  - Classification fails (with random guess): **insufficient receptive fields** or **lack positional info**

layer	I3D NL Net [33]	ARTNet [32]	TRN [39]	C2D CPNet **(ours)**
$conv_1$	$3\times3\times1, 16$	$3\times3\times3, 16$	$3\times3\times1, 16$	$3\times3\times1, 16$
	NL block	-	-	CP module
$conv_2$	$1\times1\times3, 16$ $3\times3\times1, 16$	SMART- $3\times3\times3, 16$	$3\times3\times1, 16$	$3\times3\times1, 16$
	pooling, fc	pooling, fc	pooling, temporal relation, fc	pooling, fc
train	27.8	26.8	27.1	**97.9**
val	26.4	25.9	26.9	**97.4**

Model Run Time

GTX 1080 Ti: 10.1 videos/s for frame length of 8, 3.9 videos/s for frame length of 32.



Visualization of CP Module