Temporal 3D ConvNets using Temporal Transition Layer

Ali Diba, Mohsen Fayyaz, **Vivek Sharma***, A.Hossein Karami, M.Mahdi Arzani, Rahman Yousefzadeh, Luc Van Gool

https://vivoutlaw.github.io/

Goal of the paper

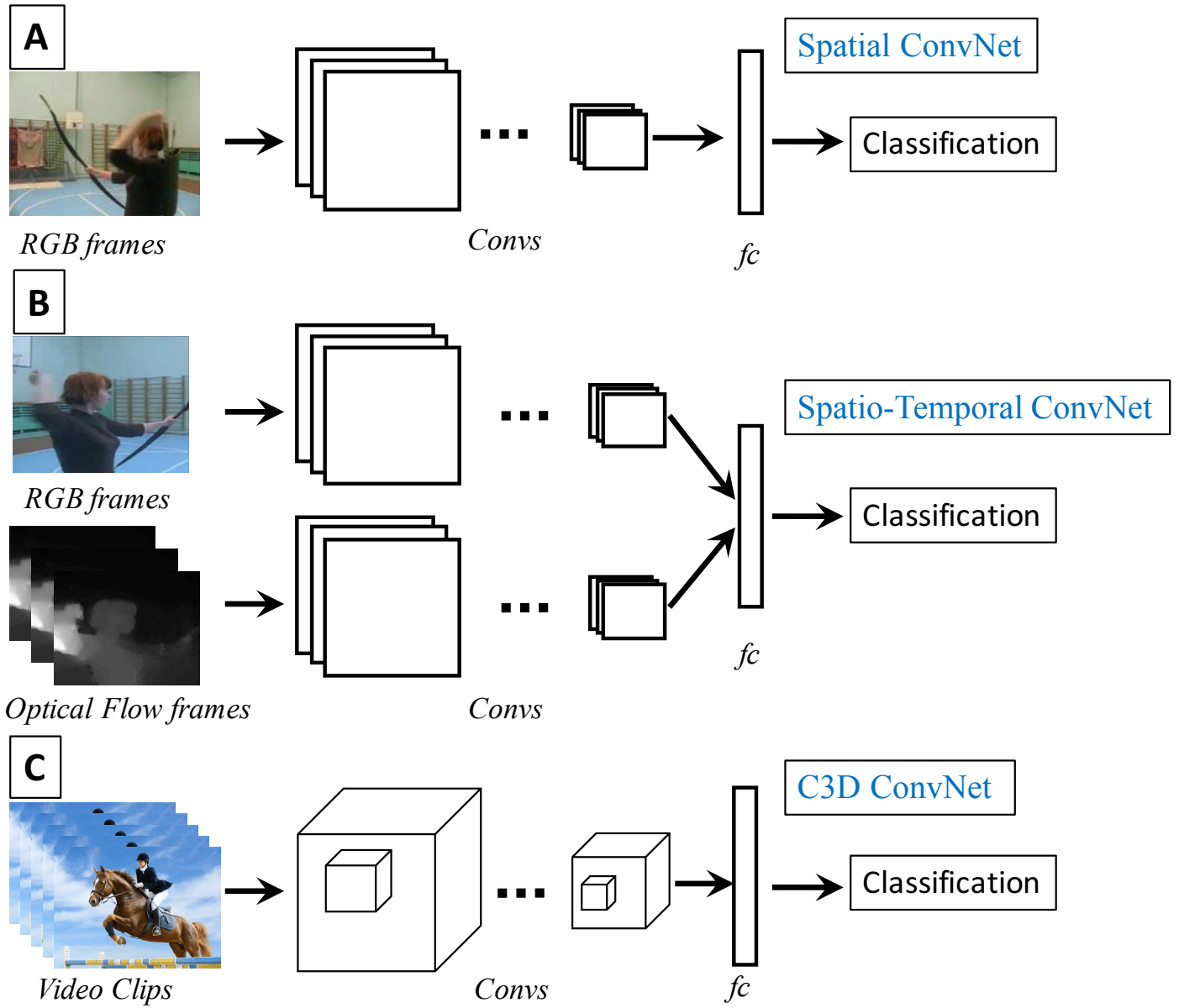
- In this paper, we address an effective utilization of the temporal cues available in the videos.
- We introduce a new temporal layer that models variable temporal depths.
  - That captures short, mid, and long term dynamics for a better video representation.

Introduction

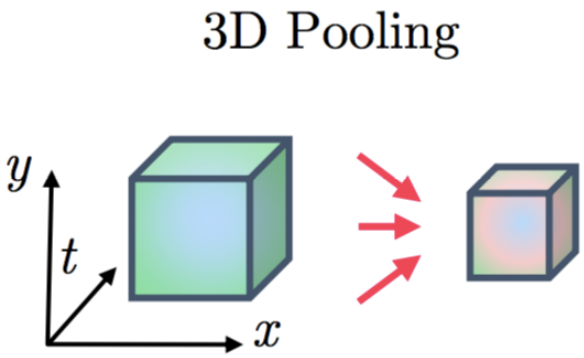
- Exploiting temporal information is advantageous for video classification.
- Current 3D ConvNets fail to exploit the long-range temporal information
  - Because of fixed 3D homogeneous kernel depths.
- Other complicating aspects of 3D ConvNets include:
  - More model parameters
  - Requires extra large labeled datasets
  - Usage of demanding optical flow maps.
- This calls for efficient methods.

- Contributions:
  - A new temporal layer that models variable temporal convolution kernel depths, namely “Temporal Transition Layer (TTL)”
  - We extend DenseNet to incorporate 3D filters and pooling kernels, namely DenseNet3D.
  - We deploy our TTL layer in DenseNet3D. We refer our architecture as “Temporal 3D ConvNets (T3D)”.
- We evaluate T3D on UCF101, HMDB51 and Kinetics datasets.

Related Work

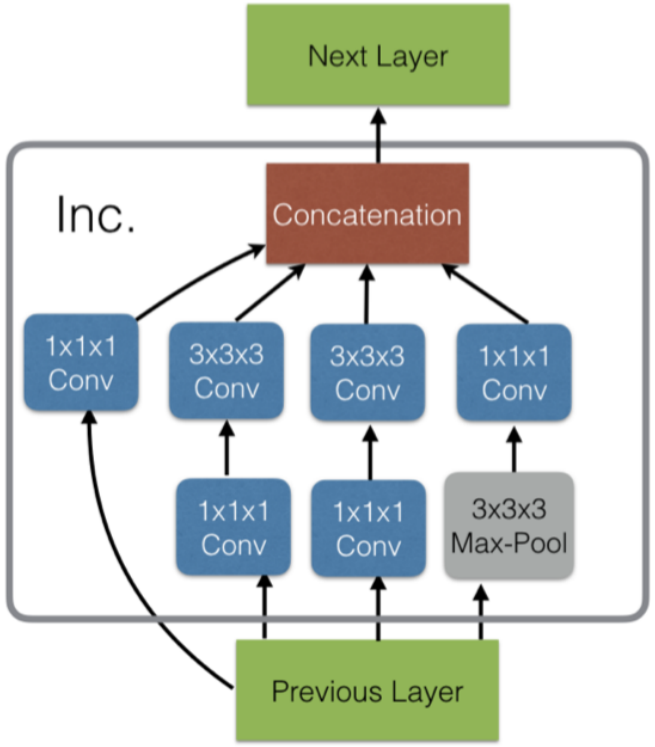


[A] A.Karpathy,G.Toderici,S.Shetty,T.Leung,R.Sukthankar, and L. Fei-Fei. Large-scale video classification with convo- lutional neural networks. In CVPR, 2014.
[B] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.
[C] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional net- works. In ICCV, 2015



3D Pooling

2016



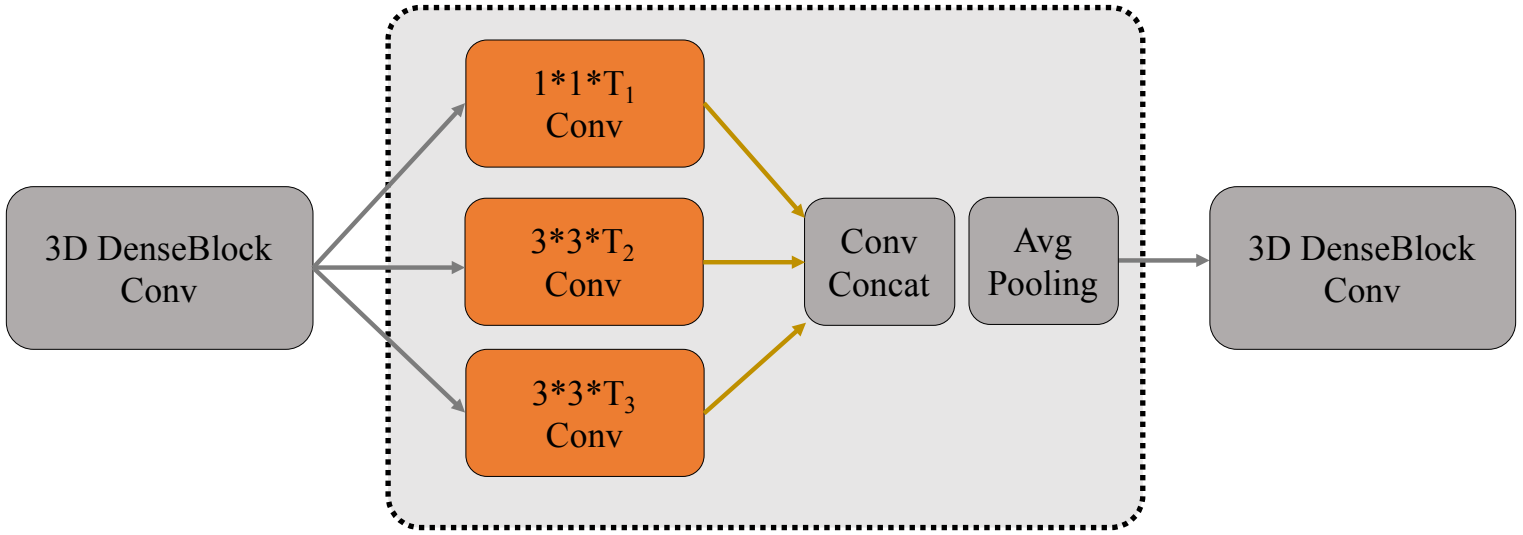
I3D Inception module

2017

C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016.
J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

Temporal Transition Layer (TTL)

- TTL capture short, mid, and long term dynamics for a video representation.
- It is inspired from GoogLeNet.
- It consists of several 3D Convolution kernels, with diverse temporal depths.
- We employ TTL in our DenseNet3D architecture.

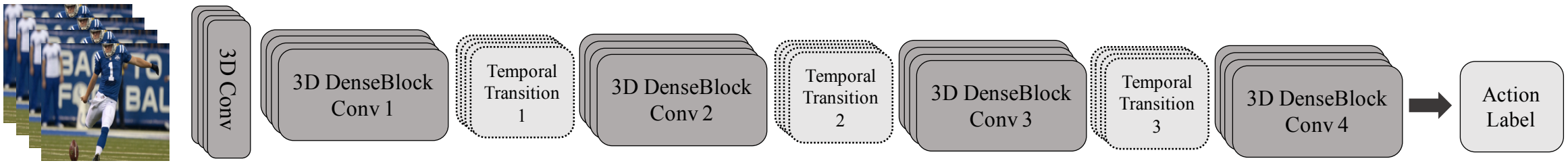


3D Temporal Transition Layer

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

DenseNet3D

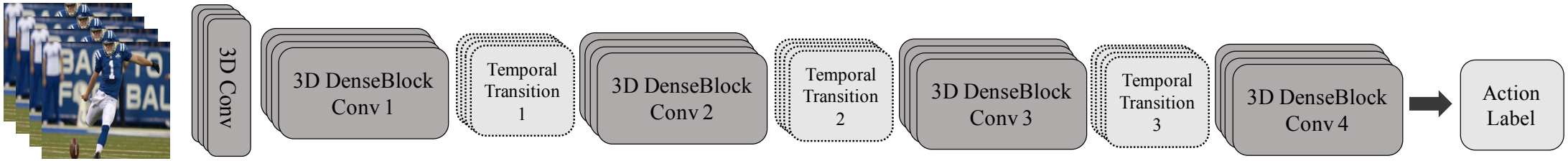
- We use the DenseNet architecture and replace the 2D filters and pooling kernels, by 3D filters and pooling kernels.
- Reasons for choosing DenseNet architecture.
  - Highly parameter efficient architecture
  - Its dense knowledge propagation
  - Its state-of-the-art performance on image classification tasks

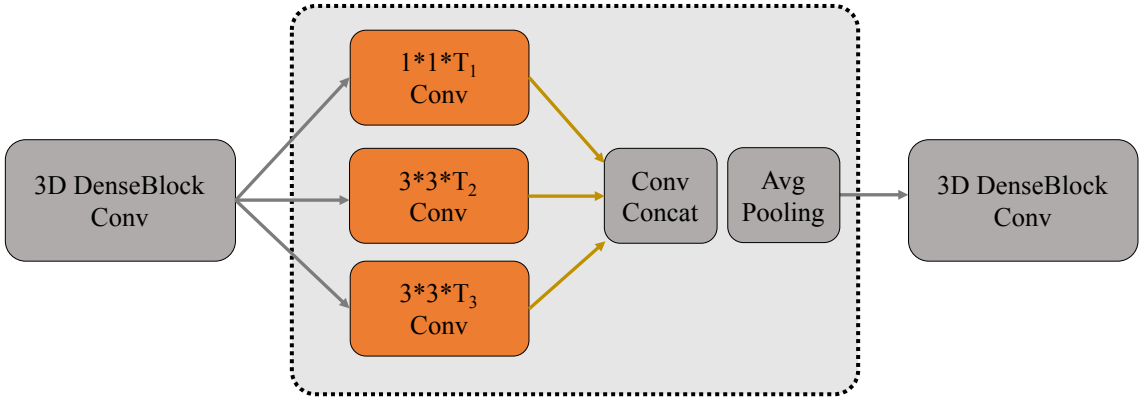


G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In *CVPR*, 2017

Temporal 3D ConvNets (T3D)

- We replace the transition layer in DenseNet3D architecture by TTL layer.
- We refer our new architecture as Temporal 3D ConvNets.





3D Temporal Transition Layer

Layers	Output Size	DenseNet3D-121	T3D-121	T3D-169
3D Convolution	112 × 112 × 16	7 × 7 × 3 conv, stride 2		
3D Pooling	56 × 56 × 16	3 × 3 × 3 max pool, stride 1		
3D Dense Block (1)	56 × 56 × 16	$\begin{bmatrix} 1 \times 1 \times 1 \\ 3 \times 3 \times 3 \end{bmatrix}$ × 6 conv	$\begin{bmatrix} 1 \times 1 \times 1 \\ 3 \times 3 \times 3 \end{bmatrix}$ × 6 conv	$\begin{bmatrix} 1 \times 1 \times 1 \\ 3 \times 3 \times 3 \end{bmatrix}$ × 6 conv
Transition/TTL (1)	56 × 56 × 16	1 × 1 × 1 conv	$\begin{bmatrix} 1 \times 1 \times \mathbf{1} \\ 3 \times 3 \times \mathbf{3} \\ 3 \times 3 \times \mathbf{6} \end{bmatrix}$ conv → concat	$\begin{bmatrix} 1 \times 1 \times \mathbf{1} \\ 3 \times 3 \times \mathbf{3} \\ 3 \times 3 \times \mathbf{6} \end{bmatrix}$ conv → concat
	28 × 28 × 8	2 × 2 × 2 avg pool, stride 2		
3D Dense Block (2)	28 × 28 × 8	$\begin{bmatrix} 1 \times 1 \times 1 \\ 3 \times 3 \times 3 \end{bmatrix}$ × 12 conv	$\begin{bmatrix} 1 \times 1 \times 1 \\ 3 \times 3 \times 3 \end{bmatrix}$ × 12 conv	$\begin{bmatrix} 1 \times 1 \times 1 \\ 3 \times 3 \times 3 \end{bmatrix}$ × 12 conv
Transition/TTL (2)	28 × 28 × 8	1 × 1 × 1 conv	$\begin{bmatrix} 1 \times 1 \times \mathbf{1} \\ 3 \times 3 \times \mathbf{3} \\ 3 \times 3 \times \mathbf{4} \end{bmatrix}$ conv → concat	$\begin{bmatrix} 1 \times 1 \times \mathbf{1} \\ 3 \times 3 \times \mathbf{3} \\ 3 \times 3 \times \mathbf{4} \end{bmatrix}$ conv → concat
	14 × 14 × 4	2 × 2 × 2 avg pool, stride 2		
3D Dense Block (3)	14 × 14 × 4	$\begin{bmatrix} 1 \times 1 \times 1 \\ 3 \times 3 \times 3 \end{bmatrix}$ × 24 conv	$\begin{bmatrix} 1 \times 1 \times 1 \\ 3 \times 3 \times 3 \end{bmatrix}$ × 24 conv	$\begin{bmatrix} 1 \times 1 \times 1 \\ 3 \times 3 \times 3 \end{bmatrix}$ × 32 conv
Transition/TTL (3)	14 × 14 × 4	1 × 1 × 1 conv	$\begin{bmatrix} 1 \times 1 \times \mathbf{1} \\ 3 \times 3 \times \mathbf{3} \\ 3 \times 3 \times \mathbf{4} \end{bmatrix}$ conv → concat	$\begin{bmatrix} 1 \times 1 \times \mathbf{1} \\ 3 \times 3 \times \mathbf{3} \\ 3 \times 3 \times \mathbf{4} \end{bmatrix}$ conv → concat
	7 × 7 × 2	2 × 2 × 2 avg pool, stride 2		
3D Dense Block (4)	7 × 7 × 2	$\begin{bmatrix} 1 \times 1 \times 1 \\ 3 \times 3 \times 3 \end{bmatrix}$ × 16 conv	$\begin{bmatrix} 1 \times 1 \times 1 \\ 3 \times 3 \times 3 \end{bmatrix}$ × 16 conv	$\begin{bmatrix} 1 \times 1 \times 1 \\ 3 \times 3 \times 3 \end{bmatrix}$ × 32 conv
Classification Layer	1 × 1 × 1	7 × 7 × 2 avg pool		
		400D softmax		

The bold numbers denote to the variable temporal convolution kernel depths applied to the 3D feature-maps.

Evaluation

- We present our evaluation on three challenging datasets: UCF101 HMDB51, and Kinetics.
- We use the standard training/testing splits and protocols provided as the original evaluation scheme.
- For HMDB51 and UCF101, we report the average accuracy over the three splits, and for Kinetics, we report the performance on the validation and test set.
- For video prediction, we decompose each video into non-overlapping clips of 32 frames, and average the predictions over all the clip.

Ablation study

- For T3D ablation study, the models were trained and tested on UCF101 split 1.
- We explore two versions of 2D-DenseNet with network sizes of 121 and 169 for designing the DenseNet3D, namely T3D-121 and T3D-169.

Model Depth	Accuracy %
121	69.1
169	71.3

- Temporal depth of series of input frames plays a key role in activity recognition tasks. Therefore, we have evaluated our T3D with configurations for different temporal depths.

Temporal Depth	Accuracy %
16	66.8
32	69.1

- T3D v.s Inception/ResNet 3D
  - We implemented Inception3D an ResNet3D

3D ConvNet	Accuracy %
ResNet3D-50	59.2
Inception3D	69.5
DenseNet3D-121 (ours)	69.1
T3D (ours)	**71.4**

Kinetics Dataset

- Trained from scratch.
- Comparison results of our T3D models with other state-of-the-art methods on Kinetics dataset.
- * denotes the pre- trained version of C3D on the Sports-1M.

Method	Top1- Val	Avg-Test
DenseNet3D	59.5	-
T3D	**62.2**	71.5
Inception3D	58.9	69.7
ResNet3D-38 [13]	58.0	68.9
C3D* [13]	55.6	-
C3D* w/ BN [4]	-	67.8
RGB-I3D w/o ImageNet [4]	-	78.2

UCF101 and HMDB51

- Pre-training on Kinetics and finetuning on all three splits of UCF101 and HMDB51 datasets.
- RGB-I3D, Kinetics pre-training 95.1 (UCF101),74.3 (HMDB51).
- Temporal segments network (TSN) with 5 non-overlapping clips of each video for encoding.

Method	UCF101	HMDB51
DT+MVSM [1]	83.5	55.9
iDT+FV [23]	85.9	57.2
C3D [20]	82.3	56.8
Conv Fusion [6]	82.6	56.8
Conv Pooling [27]	82.6	47.1
Spatial Stream-Resnet [5]	82.3	43.4
Two Stream [17]	88.6	–
F_{ST}CV (SCI fusion) [19]	88.1	59.1
TDD+FV [24]	90.3	63.2
TSN-RGB [25]	85.7	-
Res3D [21]	85.8	54.9
ResNet3D	86.1	55.6
Inception3D	87.2	56.9
DenseNet3D	88.9	57.8
T3D (ours)	**91.7**	**61.1**
T3D+TSN (ours)	**93.2**	**63.5**

L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016

Conclusion

- We clearly show the benefit of exploiting temporal depths over shorter and longer time ranges over fixed 3D homogeneous kernel depth architectures.
- Even though, in this paper, we have employed TTL to T3D architecture, our TTL has the potential to generalize to any other 3D architecture too.