PDAN: Pyramid Dilated Attention Network for Action Detection

Rui Dai[1,2], Srijan Das[1,2], Luca Minciullo[3], Lorenzo Garattoni[3],
Gianpiero Francesca[3], François Bremond[1,2]

[1]Inria [2]Université Côte d'Azur [3]Toyota Motor Europe

{name.surname}@inria.fr {name.surname}@toyota-europe.com

Abstract

Handling long and complex temporal information is an important challenge for action detection tasks. This challenge is further aggravated by densely distributed actions in untrimmed videos. Previous action detection methods fail in selecting the key temporal information in long videos. To this end, we introduce the Dilated Attention Layer (DAL). Compared to previous temporal convolution layer, DAL allocates attentional weights to local frames in the kernel, which enables it to learn better local representation across time. Furthermore, we introduce Pyramid Dilated Attention Network (PDAN) which is built upon DAL. With the help of multiple DALs with different dilation rates, PDAN can model short-term and long-term temporal relations simultaneously by focusing on local segments at the level of low and high temporal receptive fields. This property enables PDAN to handle complex temporal relations between different action instances in long untrimmed videos. To corroborate the effectiveness and robustness of our method, we evaluate it on three densely annotated, multi-label datasets: MultiTHUMOS, Charades and Toyota Smarthome Untrimmed (TSU) dataset. PDAN is able to outperform previous state-of-the-art methods on all these datasets.

> ”Time abides long enough for those who make use of it.”
>
> *Leonardo da Vinci*

1. Introduction

Videos contain spatial and temporal information: they are composed of images in XY space stacked along the temporal dimension T. Action detection, often known as temporal action localization, is an important computer vision problem whose target task is to find precise temporal boundaries of actions occurring in an untrimmed video. Previous methods that use 3D XYT convolutional filters [4, 10, 32] have obtained a great success in action classification task for clipped videos. These filters learn spatio-temporal representations within a short period of time. But what about learning representations for videos with complex temporal relations? Extra layers such as Non-Local [35] and Timeception [16] have been proposed in recent years to improve the ability to model long temporal relationships. When placed on top of 3D Convolution Neural Networks (CNNs), these layers are capable of taking up to 1024 frames as input. However, this is not sufficient for learning representations of long untrimmed videos lasting several minutes.

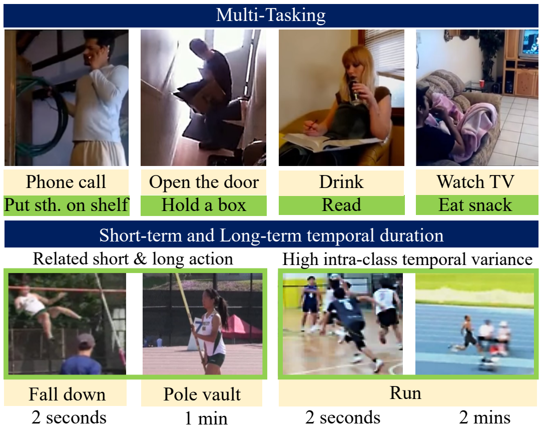


Figure 1. **Challenges:** (i) Multi-tasking: Actions can be performed concurrently (e.g. Watching TV while eating snack). (ii) Short-term and Long-term temporal duration: In the same untrimmed video, we may have related short action (e.g. *falling down*) and long action (e.g. *pole vault*). Different instances of the same action class can be short or long (e.g. *Running*) corresponding to high intra-class temporal variance.

To better understand the task, let us first understand the scenarios and challenges in action detection. In daily life, human actions are continuous and can be very dense. Every minute is filled with potential actions to be detected and labeled. There are 2 main challenges while handling densely annotated datasets (Fig. 1): (i) humans are great at multi-tasking, they can *drink* while *reading books* or *take a phone call* while *putting something on the shelf*; and (ii) the duration of actions can have a large variance. This holds true even for actions that are closely related (e.g., *pole vault* can last 1 minute, while its sub-action *falling down* only few seconds), and for instances of the same action class (e.g., *running* can last several minutes in a marathon, or a few seconds in a volleyball game).

Thus, to sum up, the challenges for action detection in long untrimmed videos include - (i) managing concurrent actions occurring at the same time, and (ii) modeling both long-term and short-term temporal duration in the video. In the next section, we discuss how the state-of-the-art (SOTA) algorithms attempted to address these challenges.

Most studies on action detection focus on videos with sparse and well-separated instances of actions [1, 17, 20, 39]. For instance, action detection algorithms on popular datasets like THUMOS [17] and ActivityNet [1] generally learn representations for single actions in a video. These actions may have only few instances in the same video. The learned representations from these videos mainly aim at discriminating the targeted actions from the background. On the other hand, algorithms targeted to daily living datasets like PKU-MMD [20] and DAHLIA [33] process input videos using a set of sliding windows. The videos in these datasets consist of several actions performed one after the other with a pause in between and no temporal relations among them. To perform well on this type of data, learning representations for small time windows is sufficient. Instead, we are interested in detecting actions occurring in the same video and pertaining complex temporal relationships among them. The videos in datasets like Charades [29] and MultiTHUMOS [38] possess such characteristics. However, algorithms designed for these datasets still struggle to model the complex temporal relationships among the densely distributed action instances [24, 25]. This generally results in a low detection accuracy.

To this end, we propose a Dilated Attention Layer (DAL). The main novelty of this architecture is how the attention weights are allocated to local frames at multi-temporal scales. Standard temporal convolution layer (i.e. Conv1D) allocates same importance to local frames in the kernel. This property prevents the temporal convolutional kernels from selecting the key information. This is a limitation especially when large temporal receptive fields are required for modeling long untrimmed videos. To overcome this limitation, we build a novel attention mechanism to explore the local context inside the kernel. The kernel ultimately processes the entire video, but at each time step, the input are only those frames comprised in the kernel size. DAL explores the relations between the center frame and the neighbouring frames in the kernel (called local context). This local attention mechanism enables the proposed framework to learn representations for short actions. Additionally, by introducing dilation in the aforementioned temporal attentional operations, we build a Pyramid Dilated Attention Network (PDAN) which consists of a hierarchy of DALs. These DALs are configured with different dilation rates to increase exponentially the size of the filter receptive field. This hierarchical structure allows PDAN to allocate attention weights to different temporal resolutions using the different DAL layers. This structure design is instrumental for the action detection of densely annotated videos.

To summarize, our contribution in this paper is in three folds: (i) We introduce DAL, which improves the quality of the local feature representation across time. (ii) We design PDAN, which can effectively learn the dependencies between action instances by applying DAL at different temporal scales. (iii) We extensively evaluate our proposed method on three densely annotated, multi-label datasets: MultiTHUMOS [38], Charades [29] and Toyota Smarthome Untrimmed (TSU) dataset, outperforming the state-of-the-art methods.

2. Related work

In this section, we review how previous studies learn temporal relations and attention for action detection.

2.1. Modeling temporal relations

Learning video representations for action detection has been popular over the years [27, 21, 40, 6]. Different from action classification, action detection needs to predict precise temporal boundaries from long untrimmed videos. In order to address the challenges of modeling complex temporal relationships within the actions in long videos, the current detection methods [19, 24, 25, 9] emphasize the temporal processing of these videos in an end-to-end manner. Such methods encode the videos by advanced 2D or 3D CNNs [31, 4, 15, 30] as a pre-processing step.

After encoding the video, action detection can be seen as a sequence-to-sequence problem. Recurrent Neural Networks (RNNs) [38, 8, 3] have been popularly used to model the temporal relation between the action instances. However, they only implicitly capture relationships between certain actions with high motion. Furthermore, due to the vanishing gradient problem, RNN based models can only capture a limited amount of temporal information and short-term dependencies.

Temporal Convolutional Networks (TCNs) are another group of temporal processing methods. In contrast to RNN based methods, TCNs can process long videos due to the kernels sharing weight for all the time steps. The result is a feature vector preserving the spatio-temporal information, along with contextual information from the neighboring frames. Some recent variants of TCNs for action detection include Dilated TCN [19] and MS-TCN [9]. Dilated-TCN [19] increases the temporal reception field by using dilated convolutions to model long temporal patterns. This is extended by MS-TCN [9] which stacks multiple Dilated-

TCNs to construct a multi-stage structure, where each stage refines the prediction of the previous one. However, standard convolutions allocate the same importance to each local feature in the kernel. This property prevents temporal convolution kernels from selecting the key information effectively, especially when dealing with dense actions and having large temporal reception fields.

With the introduction of datasets like MultiTHUMOS [38] and Charades [29] having dense labelling and concurrent actions (i.e. multi-label), more and more methodological attempts to model complex temporal relations between action instances have been made.

Ghosh et al. [12] proposed a method based on Graph Convolutional Network (GCN), namely stacked-STGCN, which extend STGCN [37] for action detection. Different from standard STGCN where the nodes of a graph represent the body joints, in stacked-STGCN, the nodes represent different elements related to the actions such as actors, objects, etc. Nodes are connected along the spatial and temporal dimensions to form the edges of the graph. Such a graph representation characterizes better the complex object-based actions in videos. But the challenge of handling actions over a long range of time still persists. Consequently, Piergiovanni et al. proposed a global representation, namely super-event [24]. In this model, Cauchy distribution based filters process the video across time to learn a latent contextual representation of the actions on particular sub-intervals of the video. The set of filters are summed by a soft attention mechanism to form the global super-event features. During prediction, the local I3D features are used with the super-event features to better model the global context. Similarly, Piergiovanni et al. [25] introduced Temporal Gaussian Mixture (TGM) layers. In contrast to standard convolution layer, TGM computes the filter weights based on Gaussian distributions, which enables TGM to learn longer temporal structures with a limited number of parameters. Although the above methods [24, 25] achieve state-of-the-art results in modeling complex temporal relations, the non-adaptive receptive field limits the ability of the models to capture the dynamics for both short and long patterns. Thus, we propose PDAN, which can capture simultaneously short and long range temporal relationships among action instances. In order to mitigate the limitations of standard temporal convolutional operations for the task of action detection, we provide a framework to learn suitable importance for neighboring frames in a video. This framework is based on self-attention mechanism.

2.2. Self-attention mechanism

Attention mechanisms focus on the salient part of a scene relative to a target task. The self-attention mechanism was proposed by Transformer Networks [34] for natural language processing. It enforces a network to establish one-to-one relations to understand the dependencies between their local representations. Employing self-attention mechanisms has gained popularity for different downstream tasks: Ramachandran et al. [26] proposed "fully attentional network", which achieves competitive prediction results on image classification tasks. This model replaces the standard 2D convolution layer with local attention layer in ResNet [15]. This layer learns the representation based on the relative position of the spatial features in the kernel. Similar to [34], Girdhar et al. [14] proposed the Action Transformer model for the task of action detection. This model inherits the transformer-style architecture to modulate features with attention weights from the spatio-temporal context within a video. This attention mechanism emphasizes the region-of-interest (e.g. actors' hands, faces), which are often crucial to recognize an action. However, Action Transformer is embedded in I3D [4] as the base network, which restricts its input size to only short video clips (i.e. 64 frames). Our target is to detect both long and short actions in a long video, far beyond 64 frames. Thus, we need a better attention mechanism that is dedicated to model temporal relations. Wang et al. [35] designed a Non-Local (NL) layer that achieves SOTA performance in action recognition task. This block leverages the self-attention mechanism to learn an attention map representing the spatial-temporal one-to-one dependencies of the 3D features. Extending NL layer, Cao et al. [2] introduced Global Context (GC) layer, which has same performance as the NL layer but with fewer parameters. While adapting NL layer and GC layer for action detection task, the receptive field of the layer is always the full video. The fixed global receptive field introduces more noise of the irrelevant actions in the attention map, thus can not provide effective attention information especially for the videos that concurrently have both multiple long and short actions. In this paper, we introduce DAL, a novel temporal filter based on self-attention mechanism. To the best of our knowledge, it is the first time that an attention mechanism is applied to varying reception fields to enhance the temporal modeling of actions with different temporal length. We further compared NL layer and DAL in Sec. 3.2.2.

3. Pyramid Dilated Attention Network (PDAN)

In this section, we introduce Pyramid Dilated Attention Network (PDAN), an end-to-end model for action detection. The main contribution of this architecture is how to allocate attention weights to all the frames at multi-temporal scales. We firstly define a Dilated Attention Layer (DAL), which is a temporal filter across time. DAL can extract better feature representation from the neighbouring frames using an attention mechanism within the kernels. By stacking DALs with different dilation rates, we design a Pyramid Dilated Attention Network (PDAN). This structure en-

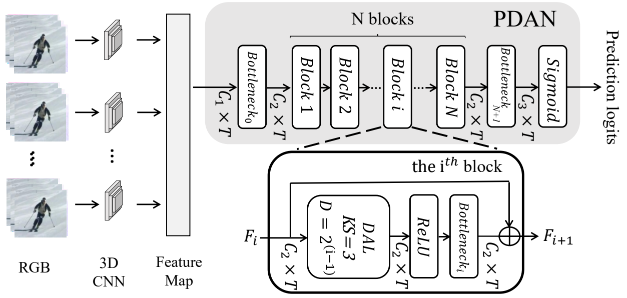


Figure 2. Overview of the Pyramid Dilated Attention Network (PDAN). In this figure, we present the structure of PDAN for one single stream. Note that RGB and Flow stream have same structure inside PDAN. Two streams are connected by late fusion operation before classification. DAL indicates the dilated attention layer, in which, KS is the kernel size, D is the dilation rate.

ables us to apply DAL on multiple temporal resolutions and therefore to effectively detect both long and short actions. Our primary novelty lies in the attention mechanism. In Sec. 3.2.2, we discuss how is it different from the state-of-the-art attention mechanisms, especially Non-Local layer. An overview of the proposed PDAN is shown in Fig. 2. The basic building block in PDAN is a DAL followed by a ReLU activation and a bottleneck with residual connection. Note that in this work, bottleneck indicates 1D convolution that processes across time and kernel size is 1. PDAN operates with both RGB as well as Flow modalities depending upon their availability. In the following sub-sections, we describe our model in detail.

3.1. Video feature extraction

Similar to most action detection models [19, 24, 25, 38], our model can process on top of video segment representations (usually from frame-level or segment-level CNN features). In this work, we use spatio-temporal features extracted from the RGB and Flow I3D networks [4] to encode appearance and motion information respectively. To achieve this, a video is divided into T non-overlapping segments, each segment consisting of 16 frames. The inputs to the RGB and Flow deep networks are the color images and corresponding Flow frames of a segment respectively. We stack the segment-level features along temporal axis to form a $T \times C_1$ dimensional video representation where each $1 \times C_1$ is the feature shape per segment. This video representation denoted as F_0 is further input to the RGB or Flow stream in our architecture. Below, we detail the basic component of our proposed PDAN, which is DAL.

3.2. Dilated Attention Layer (DAL)

In this section, we first describe structure of DAL, we then emphasize our novelty compared to non-local layer.

3.2.1 Structure

Standard temporal convolution layer (STCL) assigns the same importance to all the input features of the kernel. However, with multi-scale receptive fields, providing relevant attention weights can benefit modelling of complex temporal relationships. To this end, we propose DAL with multiple dilation rates that inherently learns the attention weights at different temporal scales. Similar to most temporal filters [25, 24], DAL processes the feature maps across the temporal domain only to preserve spatial information. To model complexity, compared to one 3×1 kernel in STCL, DAL has three learnable 1×1 kernels. Hence, DAL has a similar number of parameters compared to STCL, while providing salient attention for the detection task, owing to its structure design.

As shown in Fig. 3, the input features are processed in two steps in each kernel of DAL. Take the i^{th} block as an example: First, the elements (i.e. segment) around a center element f_{it} at time $t \in [1, T]$ are extracted to form a representative vector $f^{'}_{it}$. This feature representation is based on the kernel size: KS and dilation rate: D at i^{th} block. Note that: feature $f_{it} \in \mathbb{R}^{1 \times C_2}$, $f^{'}_{it} \in \mathbb{R}^{ks \times C_2}$. Second, the self-attention scoring system [34] is invoked by projecting the representative vector $f^{'}_{it}$ to a memory embedding (Key: K_i and Value: V_i) using 2 independent bottleneck convolutions: $K_i(f^{'}_{it}) = W_{K_i} f^{'}_{it}$, $V_i(f^{'}_{it}) = W_{V_i} f^{'}_{it}$, both W_{K_i} and $W_{V_i} \in \mathbb{R}^{C_2 \times C_2}$. Then, f_{it} is projected to the Query Q_i using another bottleneck convolution: $Q_i(f_{it}) = W_{Q_i} f_{it}$ and $W_{Q_i} \in \mathbb{R}^{C_2 \times C_2}$. The output of the attentional operation for the t^{th} time step is generated by a weighted sum of values V_i, with the attention weights obtained from the product of the query Q_i and keys K_i:

$$a_i(f_{it}) = V_i(f^{'}_{it})[softmax(Q_i(f_{it})K_i(f^{'}_{it}))]^{\mathrm{T}} \qquad (1)$$

DAL computes the correlation inside the kernel between the center element and the KS neighbouring elements. Thus for each time step t, we have a $C_2 \times 1 \times KS$ attention map. For example, while KS is 3, the local elements are frames at t, t-D and t+D. Finally, the output of DAL is obtained by concatenating the outputs for all the time steps t of the video.

$$attention_i(F_i) = [a_i(f_{i1})^{\mathrm{T}}, a_i(f_{i2})^{\mathrm{T}}, ..., a_i(f_{iT})^{\mathrm{T}}] \qquad (2)$$

where F_i is the input feature map of DAL for the i^{th} block. While concatenating the attention map for each t across time, we have the attention weights of the whole video $C_2 \times T \times KS$. In the following section, we compare DAL with the Non-local layer to emphasize its novelty.

3.2.2 Comparison with Non-Local layer

Transformer [34] is not directly applicable to action detection. Its extension to video, Action-Transformer [13] can

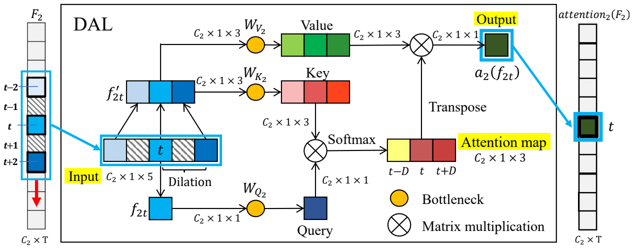


Figure 3. Dilated Attention Layer (DAL). Dilated Attention Layer (DAL). In this figure, we present a computation flow inside the kernel at time step t for layer i=2 (kernel size KS is 3, dilation rate D is 2). Afterwards, DAL processes one step forward following the red arrow at time $t + 1$.

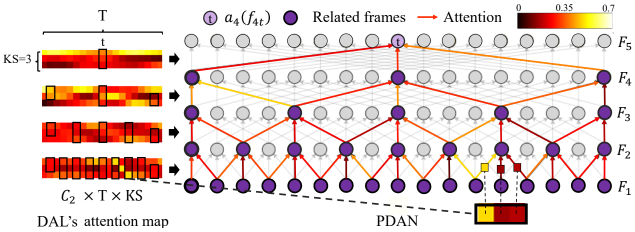


Figure 4. On the left, we visualize the attention map for DAL for four layers ($i \in$[1,4]). On the right, we present a group of frames at different temporal scales that are associated with $a_4(f_{4t})$ along with the corresponding attention weights. The circle represent the frame-level features (i.e. feature in F_i), and the arrow represents the attention-enhanced connection between the corresponding frames provided by DAL. The bounding box in the attention map corresponds to the colored arrow at right.

only process short video clips (i.e. 64 frames) and its attention mechanism is not designed to model temporal relations. Non-Local (NL) [35] has a similar structure to that of the attention head in Transformer, and is used in action detection task. Hence, we only compare DAL with the NL layer. The 1-dimensional NL layer's receptive field corresponds to the full video. These filters learn an attention map of dimension $T \times T$ reflecting the one-to-one dependency for every frame in the full video. On the other hand, DAL's receptive field at each time step t covers only the neighbouring frames in the kernel. The kernel ultimately processes the entire video, but at each time step t, the input are only those frames included in the kernel (of size KS) (see Fig. 3). Thus, DAL learns an attention map of dimension $T \times KS$, i.e. it explores the relations between the center frame and its KS neighbouring frames in the kernel. Moreover, by stacking multiple layers with different dilation rates, the receptive field is expanded gradually in higher layers to model longer actions. Consequently, both the DAL and NL layers explore the whole content of the video. Real-world untrimmed videos [38] have long duration, large temporal variance, and concurrent actions. While processing such videos, the fixed global receptive field of the NL layer implies that information linked to irrelevant actions happening potentially far away from the current frame will introduce noise to the representation of the current frame. In contrast, DAL reformulates the attention mechanism for detecting long and short actions in a sparse and hierarchical manner. This design enables the attention mechanism at each layer to focus on actions of different temporal lengths, thus providing better context information and filtering irrelevant information from the distant actions. Our ablation study confirm the effectiveness of DAL. In Fig 4, we give an example where DAL assigns different attention weights for local frames at every time step and at multi-temporal scales. The efficiency and effectiveness of NL layer and DAL is discussed in Sec. 4.3.3. In the following section, we describe how we use DALs at multiple-temporal scales.

3.3. Pyramid structure of temporal layers

Applying self-attention at multi-temporal scale is an essential ingredient for modeling complex temporal relations. PDAN is based on a pyramid of DALs with same kernel size and different dilation rates. The pyramid increases exponentially the size of the receptive field of the model. This structure allows the network to model short and long action patterns by focusing on the local frames in the kernel at the level of low and high temporal receptive fields.

As shown in Fig. 2, the input feature $F_0 \in \mathbb{R}^{T \times C_1}$ is firstly fed to a bottleneck layer to lightweight the model by reducing the channel size from C_1 to C_2. Then, N blocks are stacked, each block i is a cascade of a DAL with ReLU activation, *bottleneck* convolution and a residual link. This structure allows the receptive field to increase exponentially while keeping the same temporal length T as the input. In our experiment, we set the kernel size (KS) to 3 for all blocks, dilation and padding rate to 2^{i-1}, thus the reception field is up to 2^i+1 for the i^{th} block. The set of operations in each block can be formulated as:

$$F_{i+1} = F_i + W_i * ReLU(attention_i(F_i)) \tag{3}$$

where F_i indicates the input feature map of the i^{th} block. In the attention layer $attention_i$ the dilation rate varies with i. $W_i \in \mathbb{R}^{C_2 \times C_2}$ indicates the weights of the $bottleneck_i$. Finally, we compute per-frame binary classification score for each class (i.e. prediction logits). Therefore, the N^{th} block is followed by a bottleneck convolution with *sigmoid* activation:

$$P = sigmoid(W_{B_{N+1}} F_{N+1}) \tag{4}$$

where $P \in \mathbb{R}^{T \times C_3}$ is the prediction logits and $W_{B_{N+1}} \in \mathbb{R}^{C_3 \times C_2}$, C_3 corresponds to the number of action classes. To learn the parameters, we optimize the multi-label binary cross-entropy loss [23].

4. Experiment

The goal of these experiments is to verify that our proposed method can effectively model complex temporal relations. First, we perform an ablation study to validate the design choice of our model. Second, we compare our

Table 1. Frame-based mAP (%) to show the effectiveness of the components in PDAN. The ✓ indicates that we use this component in all the PDAN blocks. PDAN (DAL) is our proposed PDAN.

	Dilation	Residual link	DAL in block					Charades	TSU
			1	2	3	4	5		
Simple(STCL)	×	×	×	×	×	×	×	17.8	15.0
Simple(DAL)	×	×	✓	✓	✓	✓	✓	18.9	16.1
Dilation (STCL)	✓	×	×	×	×	×	×	21.8	24.0
Dilation (DAL)	✓	×	✓	✓	✓	✓	✓	23.2	26.1
Residua (STCL)	×	✓	×	×	×	×	×	21.8	24.3
Residua (DAL)	×	✓	✓	✓	✓	✓	✓	23.5	26.5
PDAN (STCL)	✓	✓	×	×	×	×	×	24.1	29.0
PDAN(Low)	✓	✓	✓	✓	×	×	×	25.3	30.1
PDAN(High)	✓	✓	×	×	×	✓	✓	25.4	30.1
PDAN (DAL)	✓	✓	✓	✓	✓	✓	✓	**26.5**	**32.7**

model with the current SOTA models on 3 densely annotated datasets to prove its effectiveness.

4.1. Evaluation datasets

We evaluate our PDAN on three challenging datasets: MultiTHUMOS[38], Charades[29] and an Toyota Smarthome Untrimmed (TSU) [5] dataset. All these three datasets are densely annotated with concurrent actions, allowing us to validate the effectiveness of PDAN in handling complex temporal relations. For all these datasets, we follow the original MultiTHUMOS and Charades evaluation settings for the action detection task, which is measuring the mean average precision (mAP) by predicting actions for each frame (frame-based mAP) of test videos. Note that TSU is a novel densely annotated action dataset. Different from Charades, TSU has much longer actions and still with complex temporal relations.

4.2. Implementation details

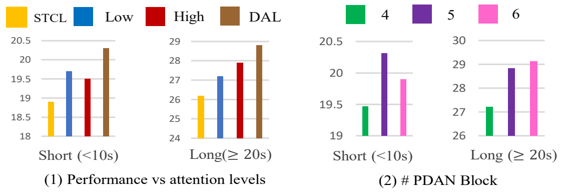
In PDAN, we set $N = 5$ blocks, $C_1 = 1024$ and $C_2 = 512$ (see Fig. 2). For each DAL in the aforementioned blocks, the kernel and stride size are set to 3 and 1, respectively. The dilation and padding rates are set to $2^{(i-1)}$ for block $i \in [1, N = 5]$. We use Adam optimizer [18] with an initial learning rate of 0.001, and we scale it by a factor of 0.3 with a patience of 10 epochs. The network is trained on a 4-GPU machine for 300 epochs with a mini batch of 32 videos for Charades, 8 videos for MultiTHUMOS and 2 videos for TSU dataset. Depending on the available modalities within the datasets, we use RGB-stream only for TSU dataset and two-stream structure for Charades and MultiTHUMOS datasets. Mean pooling of the prediction logits has been performed to fuse the RGB and Flow streams.

4.3. Ablation studies

In this section, we demonstrate the effectiveness of each component of our PDAN.

4.3.1 Block components

In Table 1, we first alternatively apply or remove dilation, residual link and DAL in all the blocks to show the effectiveness of these components (see Fig. 2). We test three configurations: (1) Simple: no residual link and no dilation[1] in any PDAN's block. (2) Dilation: no residual link but dilation in all the blocks. (3) Residual: no dilation but residual link in all the blocks. We indicate between the brackets when DAL or Standard Temporal Convolution Layer (STCL) is used in the blocks. Note that, DAL and STCL have the same kernel size and dilation rate.



(1) Performance vs attention levels

(2) # PDAN Block

Figure 5. The frame-based mAP performance for Short and Long actions on Charades with (1) different levels of attention, (2) different numbers of PDAN Blocks.

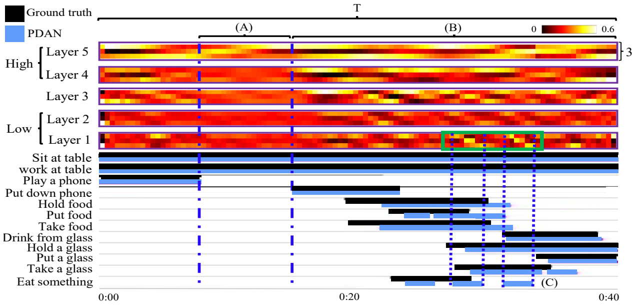


Figure 6. Qualitative analysis of the attention map. On the top, we visualize the attention map of DAL for 5 layers ($C_2 \times T \times 3$ for each layer). On the bottom, we present the corresponding **ground truth** and **PDAN** detection for this video.

Results show that for both datasets dilation and residual link lead to similar improvements (+4.0% on Charades). When accompanied by the residual link (i.e. PDAN (STCL)), dilation boosts the action detection performance by up to 2.3% on TSU w.r.t. dilation only. Using DAL in all the layers, PDAN outperforms all these ablation baselines (+1.1%, +2.1%, +2.2% and +3.7% w.r.t. Simple, Dilation, Residual and PDAN (STCL) on TSU). These results suggest that DAL is a more effective temporal filter than STCL and that dilation with residual link help boost DAL's performance. We then study to which block, attention should be integrated. We apply attention mechanism on different blocks to build four ablation baselines: PDAN (STCL), PDAN (Low), PDAN (High) and PDAN (DAL). Low and high indicates that instead of using STCL, we apply DAL in the first two blocks and last two blocks, respectively. PDAN (Low) and PDAN (High) correspond to a low (< 5.6 sec.) and high (> 24.8 sec.) receptive field respectively. Table 1 shows that both baselines can improve the performance (up to 1.3% w.r.t. Residual+Dilation on Charades). In Fig. 5 (1), we show that PDAN (Low)

[1] No dilation indicates that all the blocks are set with dilation rate 1.

can better detect short actions, and PDAN (High) can better detect the Long actions. PDAN incorporates the attention mechanism on all the blocks and achieves the best performance for both long and short actions (+2.4% w.r.t. PDAN (STCL) on Charades dataset).

In Fig. 6, we present the attention map of DAL for 5 layers (on top), and the corresponding ground truth vs PDAN detection results (on the bottom). In area (A), with only long actions (e.g.*work at table*), only the higher layers allocate high attention weights to the frames in the kernel. This reflects that the higher layers are more sensitive to long-term actions. In area (B), with both long and short actions, both higher and lower layers allocate high attention weights to the frames in the kernel. In area (C) (at the bottom), while detecting short actions, DAL allocates high attention weights at the lower layer, corroborating that the lower layer is particularly sensitive to short actions.

4.3.2 Number of blocks

Table 2. Ablation study to determine the number of blocks in PDAN. "Temp. Field" indicates the length of temporal reception field (expressed in seconds) for the kernel at the last block.

Num. Blocks	Temp. Field	Charades	TSU
3	15	23.3	29.4
4	31	25.0	30.3
5	63	**26.5**	**32.7**
6	127	25.6	30.5

Table 2 reports the performance while using different numbers of blocks in PDAN. This performance depends on the size of the temporal receptive field and the average action length in the videos. With more blocks, PDAN can have a larger temporal reception field. Here, 5 block structure indicates that PDAN's reception field explores up to 63 segments (i.e. about 1 min), which can satisfy the requirements of both datasets. In Fig. 5 (2), we analyse the performance of the number of PDAN blocks for actions with different duration. 5-blocks structure achieves the best performance for frame-based mAP (up to 2.4% w.r.t. 4 block structure on TSU). While increasing to 6-blocks improves the performance for long actions (+0.4%), it deteriorates the performance for short actions. This can be explained by the fact that having more layers tends to diminish the importance of local context.

4.3.3 DAL& NL layer

In Table 3, we measure the efficiency of DAL compared to the Non-Local (NL) layer [35]. While replacing all the DALs by STCLs in the PDAN block, we obtain PDAN (STCL) (see Fig. 2). We have tried two different ways of integrating the NL layer. NL-T1 indicates that we add one NL layer before the classifier in PDAN (STCL); NL-T2 indicates that we replace the DAL layer by a STCL and a NL layer in every PDAN block (see Fig. 2). As mentioned in Sec. 3.2, PDAN (STCL) and PDAN have similar parameters. Besides, DAL outperforms both NL-T1 and NL-T2 with large margin (+1.9% and +2.4% w.r.t. NL-T1 and NL-T2 on Charades), while having less parameters and less operations (i.e. FLOPs). This result reflects that DAL is more efficient and effective than NL layer for action detection in densely annotated videos.

Table 3. Frame-based mAP (%) to show the effectiveness of the components in PDAN. PDAN (STCL) indicates that we replace DAL in the PDAN block by the standard temporal convolution layer. NL-T1 indicates that we add one Non-Local layer before the PDAN (STCL) classifier. NL-T2 indicates that we add one NL-layer after every STCL in PDAN (STCL).

	#Param (M)	FLOPs (GMac)	Charades	TSU
PDAN (STCL)	5.9	0.59	24.1	29.0
PDAN (STCL)+NL-T1	6.4	0.65	24.6	29.2
PDAN (STCL)+NL-T2	8.5	0.88	23.9	28.5
PDAN (DAL)	5.9	0.62	**26.5**	**32.7**

Table 4. Frame-based mAP (%) to show the effectiveness of DAL integrated in Timeception structure.

	#Param	FLOPs	Charades	TSU
I3D+Timeception (STCL)	4.8 M	0.46 G	21.8	27.0
I3D+Timeception (DAL)	4.8 M	0.47 G	23.0	29.3

4.3.4 Timeception + DAL

Finally, we embed DAL in another structure based on temporal convolution [16] to confirm the effectiveness of DAL. Different from PDAN, Timeception [16] utilizes several temporal convolutions in parallel with different dilation rates. This design enables Timeception to explore multi-temporal scales in one layer. However, Timeception is designed for multi-label action classification, not for action detection. So, it applies max pooling to aggregate the temporal information and halve the temporal resolution at every layer. Hence, we remove the max pooling from the original Timeception structure to utilize the temporal information for the action detection task (i.e. Timeception (STCL)). Based on this new structure, we replace the standard temporal convolution with our proposed DAL (i.e. Timeception (DAL)) to demonstrate that DAL can be combined with other architectures. In Table 4, we report the mAP performance of 3-layer Timeception. We find out that Timeception (DAL) improves the base network performance (up to +2.3% on TSU w.r.t. Timeception (STCL)), but it underperforms compared to PDAN.

4.4. Comparison with state-of-the-art methods

The proposed PDAN is compared with previous methods on the MultiTHUMOS, Charades and TSU datasets in Table 5, Table 6 and Table 7. To be noticed, the I3D baseline (i.e. I3D in the tables) used for comparison is a classifier on top of the segment-level I3D features. Unlike the other SOTA, I3D baseline does not have further temporal processing after the video encoding part. Thus, this method cannot model long temporal information, which is crucial for action detection. In contrast, the other action detection base-

Table 5. Performance of the state-of-the-art methods and our approach on MultiTHUMOS. I3D model is two-stream, using both RGB and optical flow input. Note: cited papers may not be the original paper but the one providing this mAP results. *indicates the results obtained by running the available code.

	mAP
Two-stream [38]	27.6
Two-stream+LSTM [38]	28.1
Multi-LSTM [38]	29.6
SSN [40]	30.3
I3D [25]	29.7
I3D + LSTM [25]	29.9
I3D + temporal pyramid [25]	31.2
TAN [7]	33.3
I3D + Dilated-TCN* [19]	43.2
I3D + 3 TGMs [25]	44.3
I3D + MS-TCN* [9]	45.3
I3D + 3 TGMs + Super event [25]	46.4
I3D + **PDAN**	**47.6**

Table 6. Per-frame mAP on Charades, evaluated with the Charades localization setting. Note: cited papers may not be the original paper but the one providing this mAP results. *indicates the results obtained by running the available code.

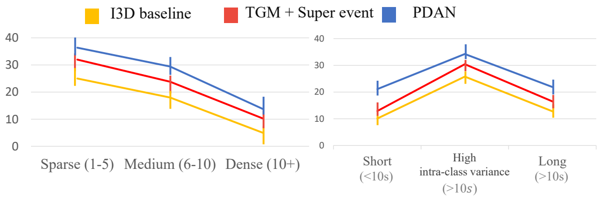
	Modality	mAP
Two-stream [28]	RGB + Flow	8.9
Two-stream+LSTM [28]	RGB + Flow	9.6
R-C3D [36]	RGB	12.7
Asynchronous Temporal Fields [28]	RGB + Flow	12.8
I3D [24]	RGB	15.6
I3D [24]	RGB + Flow	17.2
I3D + 3 temporal conv.layers [25]	RGB + Flow	17.5
TAN [7]	RGB + Flow	17.6
I3D + WSGN (supervised) [11]	RGB	18.7
I3D + Stacked-STGCN [12]	RGB	19.1
I3D + Super event [24]	RGB + Flow	19.4
I3D + 3 TGMs [25]	RGB + Flow	21.5
I3D + 3 TGMs + Super event [25]	RGB + Flow	22.3
I3D + Dilated-TCN* [19]	RGB + Flow	23.5
I3D + MS-TCN* [9]	RGB + Flow	24.2
I3D + **PDAN**	RGB	**23.7**
I3D + **PDAN**	RGB + Flow	**26.5**

Table 7. Frame-based mAP on TSU dataset. Note: I3D models are using only RGB stream.

	Parameter	FLOPs	mAP
I3D [4]	-	-	13.3
I3D + LSTM [22]	-	-	15.9
I3D + Super event [24]	-	-	15.6
I3D + 4 TGMs [25]	-	-	20.2
I3D + 4 TGMs + Super event [25]	2.1M	0.27 GMac	23.6
I3D + Dilated-TCN [19]	4.5 M	0.46 GMac	25.1
I3D + MS-TCN [9]	13.8 M	1.38 GMac	29.6
I3D + **PDAN**	5.9 M	0.62 GMac	**32.7**

lines as [24, 25, 38] focus on the temporal processing. The improvement over I3D baseline reflects the effectiveness of modeling temporal information. PDAN consistently outperforms the prior methods [38, 12, 24, 7, 25] for action detection on all the three challenging datasets. For Dilated-TCN, although it has less parameters, PDAN improves the performance with a large margin (up to +7.6% in TSU dataset). Compared with MS-TCN, PDAN achieves better performance (up to +3.1% in TSU) while having fewer parameters and FLOPs (Table 7).

We then study how our proposed method can tackle complex temporal relations. We perform this comparison with I3D baseline [4], and TGM + Super event [25]. In Fig. 7, we first study the performance along the multi-tasking challenge on Charades dataset and for detecting both long-term and shot-term temporal duration on TSU dataset with the appropriate metrics. To study the ability of the different approaches to handle concurrent actions, we created 3 groups of actions depending on the number of co-occurring actions per frame. Sparse: 1-5 concurrent actions, Medium: 6-9 concurrent actions and Dense: more than 10 concurrent actions. We compute the mAP for these three groups and find out that PDAN consistently achieves the best performance (see Fig. 7 (2)). Secondly, we study the performance along different temporal lengths of the actions. High intra-class temporal variance indicates the actions where the temporal variance is larger than 10 *seconds*. We then separate the remaining actions into short actions (≤10 sec) and long actions (> 10 sec). We find out that PDAN outperforms TGM + Super event for all these action types reflecting better handling of both short-term and long-term duration. Thanks to the use of the dilated attention layers with multi-temporal scales, PDAN can deal with actions of variable length. This comparison with SOTA methods confirms that PDAN can better handle complex temporal relations for actions from densely annotated untrimmed videos.



(1) Multi-tasking

(2) Short & Long temporal duration

Figure 7. Handling 2 challenges related to complex temporal relations on Charades dataset: (1) Multi-tasking, (2) Short and long temporal duration. We calculate the mAP for each group of actions for each challenge.

5. Conclusion

In this paper, we tackle the modeling of complex temporal relations in densely annotated video streams. We propose a Dilated Attention Layer (DAL) to learn better feature representation across time. We then introduce a Pyramid Dilated Attention Network (PDAN) that can effectively learn the dependencies between action instances by applying DAL at different temporal levels. We evaluate our method on 3 densely annotated multi-label datasets: MultiTHUMOS, Charades and an TSU dataset. Our experiments confirm that PDAN outperforms the state-of-the-art methods on all the datasets.

Acknowledgements. The authors are grateful to the OPAL infrastructure from Université Côte d'Azur for providing resources and support.

References

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

[2] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[3] Fabio Carrara, Petr Elias, Jan Sedmidubsky, and Pavel Zezula. Lstm-based real-time action detection and prediction in human motion streams. *Multimedia Tools and Applications*, 78(19):27309–27331, 2019.

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

[5] Rui Dai, Srijan Das, Saurav Sharma, Luca Minciullo, Lorenzo Garattoni, Francois Bremond, and Gianpiero Francesca. Toyota smarthome untrimmed: Real-world untrimmed videos for activity detection. *arXiv preprint arXiv:2010.14982*, 2020.

[6] Rui Dai, Luca Minciullo, Lorenzo Garattoni, Gianpiero Francesca, and François Bremond. Self-attention temporal convolutional network for long-term daily living activity detection. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–7. IEEE, 2019.

[7] Xiyang Dai, Bharat Singh, Joe Yue-Hei Ng, and Larry Davis. Tan: Temporal aggregation network for dense multi-label action recognition. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 151–160. IEEE, 2019.

[8] Roeland De Geest and Tinne Tuytelaars. Modeling temporal structure with lstm for online action detection. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1549–1557. IEEE, 2018.

[9] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3575–3584, 2019.

[10] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *CoRR*, abs/1812.03982, 2018.

[11] Basura Fernando, Cheston Tan, and Hakan Bilen. Weakly supervised gaussian networks for action detection. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[12] Pallabi Ghosh, Yi Yao, Larry S Davis, and Ajay Divakaran. Stacked spatio-temporal graph convolutional networks for action segmentation. *arXiv preprint arXiv:1811.10575*, 2018.

[13] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. *CoRR*, abs/1812.02707, 2018.

[14] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2019.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Timeception for complex action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 254–263, 2019.

[17] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. `http://crcv.ucf.edu/THUMOS14/`, 2014.

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[19] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017.

[20] Chunhui Liu, Yueyu Hu, Yanghao Li, Sijie Song, and Jiaying Liu. Pku-mmd: A large scale benchmark for continuous multi-modal human action understanding. *arXiv preprint arXiv:1703.07475*, 2017.

[21] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017.

[22] Behrooz Mahasseni and Sinisa Todorovic. Regularizing long short term memory with 3d human-skeleton sequences for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3054–3062, 2016.

[23] Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. Large-scale multi-label text classification—revisiting neural networks. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 437–452. Springer, 2014.

[24] AJ Piergiovanni and Michael S Ryoo. Learning latent super-events to detect multiple activities in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5304–5313, 2018.

[25] AJ Piergiovanni and Michael S. Ryoo. Temporal gaussian mixture layer for videos. In *International Conference on Machine Learning (ICML)*, 2019.

[26] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Standalone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019.

[27] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1049–1058, 2016.

[28] Gunnar A Sigurdsson, Santosh Divvala, Ali Farhadi, and Abhinav Gupta. Asynchronous temporal fields for action recog-

nition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 585–594, 2017.

[29] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision(ECCV)*, 2016.

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[31] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 4489–4497, Washington, DC, USA, 2015. IEEE Computer Society.

[32] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[33] Geoffrey Vaquette, Astrid Orcesi, Laurent Lucat, and Catherine Achard. The daily home life activity dataset: a high semantic activity dataset for online recognition. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 497–504. IEEE, 2017.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[35] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[36] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *Proceedings of the IEEE international conference on computer vision*, pages 5783–5792, 2017.

[37] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[38] Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision*, 126(2-4):375–389, 2018.

[39] Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan. Hacs: Human action clips and segments dataset for recognition and temporal localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8668–8678, 2019.

[40] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2914–2923, 2017.