AssembleNet++: Assembling Modality Representations via Attention Connections

Michael S. Ryoo[1,2], AJ Piergiovanni[1], Juhana Kangaspunta[1], and Anelia Angelova[1]

[1] Robotics at Google
[2] Stony Brook University
{mryoo,ajpiergi,juhana,anelia}@google.com

Abstract. We create a family of powerful video models which are able to: (i) learn interactions between semantic object information and raw appearance and motion features, and (ii) deploy attention in order to better learn the importance of features at each convolutional block of the network. A new network component named *peer-attention* is introduced, which dynamically learns the attention weights using another block or input modality. Even without pre-training, our models outperform the previous work on standard public activity recognition datasets with continuous videos, establishing new state-of-the-art. We also confirm that our findings of having neural connections from the object modality and the use of peer-attention is generally applicable for different existing architectures, improving their performances. We name our model explicitly as AssembleNet++. The code will be available at: https://sites.google.com/corp/view/assemblenet/

Keywords: video understanding, activity recognition, attention

1 Introduction

Video understanding is a fundamental problem in vision with many novel approaches proposed recently. While many advanced neural architectures have been used for video understanding [4, 43], including two-stream and multi-stream ones [4, 9, 34], learning of interactions between raw input modalities (e.g., RGB and motion) and semantic input modalities such as objects in the scene (e.g., persons and objects) have been limited.

Inspired by previous work, e.g. AssembleNet architectures for videos [34] and RandWire architectures for images [53] which proposed random or targeted connectivity between layers in a neural network, we create a family of powerful video models that explicitly learn interactions between spatial object-specific information and raw appearance and motion features. In particular, inter-block attention connectivity is searched for to best capture the interplay between different modality representations.

The main technical contributions of this paper include:

1. Optimizing neural architecture connectivity for object modality fusion. We discover that models with 'omnipresent' connectivity from object input allows the best multi-modal fusion.
2. Learning of video models with *peer-attention* on the connections. We newly introduce an one-shot model formulation to efficiently search for architectures with better peer-attention connectivity.

We test the approach extensively on challenging video understanding datasets, showing notable improvements: compared to the baseline backbone architecture we use, our new one-shot attention search model with object modality obtains +12.6% on Charades classification task and +6.22% on Toyota Smarthome dataset. Our approach also outperforms reported numbers of existing approaches on both datasets, establishing new state-of-the-art.

2 Previous work

Video CNNs Convolutional neural network (CNNs) for videos [38, 4, 8, 9, 31, 46, 18, 54, 42, 43] are a popular approach to video understanding, for example, solutions, such as 3D Video CNNs [40, 17, 41, 4, 42, 12], (2+1)D CNNs [43] or even novel architecture searched models [27, 28, 34] are widely used. Action recognition has also been the topic of intense research [11, 45].

Action recognition with objects Action recognition with objects has been traditionally studied years back [26]. The presence of specific objects in video frames, has been shown to be important for video recognition, even in the context of advanced feature learned by deep neural models, e.g., Sigurdsson et al. [37]; they are useful even if provided as a single label per frame. This is not surprising as many of the activities, e.g. 'speaking on the phone', or 'reading a book' are primarily determined by the objects themselves. Furthermore, clues about the location of persons, e.g., by 2D human pose has also been shown to be beneficial [5]. Recent video CNNs have also tried to integrate object-related information, from segmentation [2, 32] or pre-training from image datasets [6]. One-time late (or intermediate) fusion of object representation with RGB and flow representations has been widely used (e.g., [24]). Ji et al. [16] modeled scene relations on top of video CNNs using graph neural networks, for better usage of object information. However, we are not aware of any prior work that 'learns' the connectivity between among input modalities including object information, as we do in this paper.

Attention Use of attention within CNNs have been widely studied. Vaswani et al. [44] investigated different forms and applications of attention while focusing on self-attention. Hu et al. [14] introduced Squeeze-and-Excitation, which is a form of channel-wise self-attention. Researchers also developed other forms of channel-wise self-attention [50, 10, 15, 19, 47], often together with spatial self-attention. Attention was also applied to video CNN models [29, 22, 5]. However,

we are not aware of prior work explicitly searching for inter-block attention connectivity (i.e., peer-attention) as we do in this paper.

Neural architecture search Neural Architecture Search (NAS) is the concept of automatically finding superior architectures based on training data [58, 59, 20, 33, 39]. Multiple different strategies including learning of reinforcement learning controller (e.g., [58, 59] as well as evolutionary algorithms (e.g., [33]) have been developed for NAS. In particular, one-shot differentiable architecture search [3, 21] has been successful as it does not require a massive amount of model training. RandWire network [53] could also be interpreted as a form of differentiable architecture search, as it learns weights of (random) connections to minimize the classification loss.

However, architecture search for neural attention connectivity has been very limited. Ahmed and Torresani [1] searched for layer connectivity and Ryoo et al. [34] searched for multi-stream connectivity for video CNNs, but they were without any attention learning which becomes a crucial component when we have a mixture of input modalities. We believe this paper is the first paper to search for models with attention connectivity.

3 Approach

3.1 Preliminaries

This section describes the video CNN architecture framework, which will be used as a base for developing our approach.

We here adopt a multi-stream, multi-block architecture design from AssembleNet [34]. AssembleNet design allows learning of connections between modalities and their intermediate features. This architecture is similar to other two-stream models [4, 9], but is more flexible in two ways: 1) it allows the use of more than two streams, and 2) it allows connections to be formed (and potentially learned) between individual blocks of the neural architecture.

More specifically, the architecture we use has multiple input blocks, each corresponding to an input modality. The network blocks have a structure inspired by ResNet architectures [13]. Each input block is composed of a small number of pooling and convolutional layers attached directly on top of the input. The input blocks are then connected to network blocks at the next level. We follow the (2+1)D ResNet block structure from [43], where each module is composed of one 1D temporal conv., one 2D spatial conv., and one 1x1 conv. layer. A block is formed by repeating the (2+1)D residual module multiple times. This allows a fair and direct comparison between our approach and previous models using the same module and block [43, 9, 34].

Each network block (or block for short) can be connected to any block from any modality at the next level, including its own. Blocks are organized at levels so that connections do not form cycles. Connections can also be formed to skip levels. We note that since many connections between blocks are formed early,

the neural blocks themselves will often contain information from many input modalities as early as the first level of the network.

Figure 4 (a) shows one example architecture, where the structure of the network and example connectivity can be seen.

3.2 Input modalities and semantics

In addition to the standard raw RGB video input, motion information is added as a separate modality. More specifically, optical flow, either pre-computed for the dataset [55], or trained on the fly [7, 30], has been shown to be a crucial input for achieving better accuracy across the board [4].

We here propose to use object segmentation information as a separate 'object' modality. Objects and their locations provide semantics information which conveys useful information about activities in a video. Crucially here, semantic information is incorporated in the full architecture so that it is able to interact with other modalities and the intermediate features from them (as described more in Sections 3.3 to 3.5), to maximize its utilization for the best representation.

Input block details We construct an input block for each input modality. Each input block is composed of one pooling and up to two convolutional layers for raw RGB and optical flow inputs, and just one pooling layer for semantic object inputs applied directly on top of the inputs. In the object input block, a segmentation mask having an integer class value per pixel is converted into a HxWxC_O tensor using one-hot operation, where C_O is the number of object classes. The segmentation masks are obtained from a model trained on a non-related image-based dataset.

3.3 Learning weighted connections

Blocks in the network can potentially form connections with one or more blocks. While connectivity and the strength of the connectivity could also be hand-coded, we formulate our networks so that they are learnable.

Let G be the connectivity graph of the network where (j, i) specifies that there is a connection from the jth block to ith block. We allow each block to receive its inputs from multiple different input blocks as well as intermediate convolutional blocks, and generates an output. Specifically, we formulate the input of the block as a weighted summation over multiple connections where we learn one weight for each connection.

$$x_i^{in} = \sum_{(j,i)\in G} \sigma(w_{ji}) \cdot x_j^{out} \tag{1}$$

where i and j are block indexes, x_i^{in} corresponds on the final input to the ith block, and x_i^{out} corresponds to the output of the block. σ is a sigmoid function.

Learning of the connection weights together with the other convolutional layer parameters with the standard back propagation allows the network to optimize itself on which connections to use and which to not based on the training data. In our approach, this is done by initially connecting every possible blocks in the graph while using the block levels to avoid cycles, and then learning them. We consider every connection (j, i) from the jth block to ith block as valid as long as $L(j) < L(i)$ where $L(i)$ indicates the level of the block.

3.4 Attention connectivity and peer-attention

In addition to having and learning static weights per connection, we use attention to dynamically control the behavior of each connection. The intuition is that objects and activities are correlated, and using attention allows the model to focus on important objects based on motion context and vice versa. For instance, motion features of 'drinking' could suggest another network stream to focus more on objects related to such motion (e.g., 'cups' and 'bottles').

We formulate our connectivity graph G to have one more component for each edge: $((j, i), k)$, where k is the convolutional block influencing the connection (j, i) via attention. A channel-wise attention is used to implement this behavior. Let C_i be the size of the input channel of block i. For each connection (j, i), the attention vector of size C_i is computed per frame as:

$$A_i(x) = [a_1, \ldots, a_{C_i}] = \sigma(f(\text{GAP}(x))) \tag{2}$$

where f is a function (one fully connected layer in our case) mapping a vector to a vector of size C. GAP is the global average pooling over spatial resolution in the input tensor, making $\text{GAP}(x)$ to have a form of a vector per frame.

Using $A_i(x)$, the input for each block i is computed by combining every connection (j, i) while considering its attention from block k:

$$x_i^{in} = \sum_{((j,i),k) \in G} \sigma(w_{ji}) \cdot (A_i(x_k^{out}) \cdot x_j^{out}). \tag{3}$$

The simplest special case of our attention is self-attention, which is done by making x_k and x_j to be identical. In this form, the usage of attention becomes similar to Squeeze-and-Excitation [14].

Importantly, in our approach, we learn to select different x_k where $x_k \neq x_j$, which we discuss more in the following subsection. Attention with $x_k \neq x_j$ implies that the channels to use for the connection is dynamically decided based on another input modality and peer blocks. We more explicitly name this approach as *peer-attention*. In principle, we define a 'peer' as any block p that could potentially be connected to i. In our formulation where the convolutional blocks are organized into multiple levels (to avoid cycles), the set of peers P for a connection (j, i) is computed as $P_{(j,i)} = \{p \mid L(p) < L(i)\}$ where $L(p)$ indicates the level of the block p. We consider the attention connection $((j, i), k)$ to be valid as long as $k \in P_{(j,i)}$.

Figure 1 compares connectivity without attention and connectivity with self- and peer-attention.

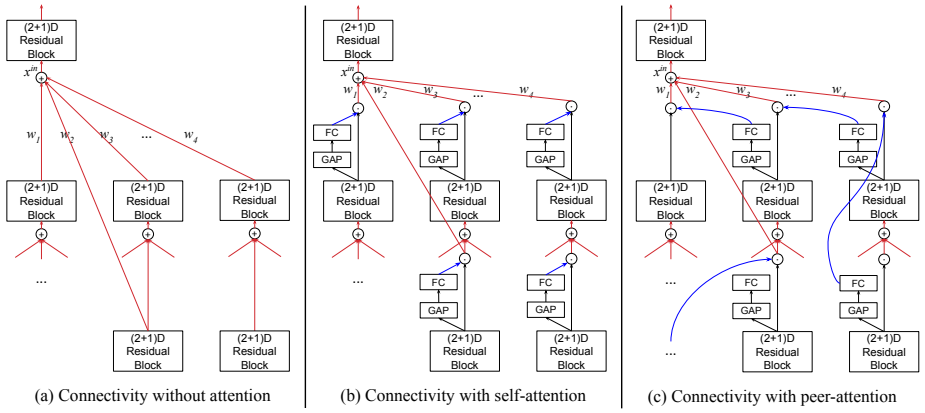


Fig. 1. Examples of convolutional block connectivity (a) without attention, (b) with self-attention, and (c) with peer-attention. Red lines indicate weighted connections from Section 3.3. Blue curves specify the attention connectivity. GAP is global average pooling and, FC is a fully connected layer. Our attention is channel-wise attention, and it is applied per frame.

3.5 One-shot attention search model

Given a set of convolutional blocks, instead of hand-designing peer-attention connections, we search for the attention connectivity. Our new one-shot attention search model is introduced, which optimizes the model's peer-attention configuration directly based on training data.

Our one-shot attention search model is formulated by combining attention from all possible peer blocks for each connection with learnable weights. The idea is to enable the model to soft-select the best peer for each block by learning differentiable weights, maximizing the recognition performance. All possible attention connectivity is considered as a consequence, and the searching is done solely based on the standard backpropagation.

For each pair of blocks (j, i) where $L(j) < L(i)$, we place a weight for every $k \in P_{(j,i)}$. Let h be a weight vector of size $m = |P_{(j,i)}|$, and $X_P^{out} = [x_1^{out}, \ldots, x_m^{out}]$ be the tensor concatenating x_k^{out} of every possible peer k in P. Then, we reformulate Equation 3 as:

$$x_i^{in} = \sum_{(j,i)\in G} \sigma(w_{ji}) \cdot (A(x) \cdot x_j^{out}) \quad \text{where} \quad x = \mathbf{1}^T \left(\text{softmax}\,(h) \cdot X_{P_{(j,i)}}^{out} \right). \quad (4)$$

$\mathbf{1}$ is a vector of size m having 1 as all its element values, making x to be a weighted sum of peer block outputs x_k^{out}. Use of softmax function allows one-hot like behavior (i.e., selecting one peer to control the attention) based on learned weights $h = [h_1, \ldots, h_m]$. Figure 2 visualizes the process.

The entire process is fully differentiable, allowing the one-shot training to learn the attention weights h together with the connection weights w_{ji}. This is

unlike AssembleNet which partially relies on exponential mutations to explore connections. Once the attention weights are found, we can either prune the connections by only leaving the argmax over h_k or leave them with softmax. We confirmed that they do not make different in practice, allowing us to only maintain one peer-attention per block as shown in Figure 1 (c). Peer-attention only causes 0.151% increase in computation, which we describe more in Appendix.

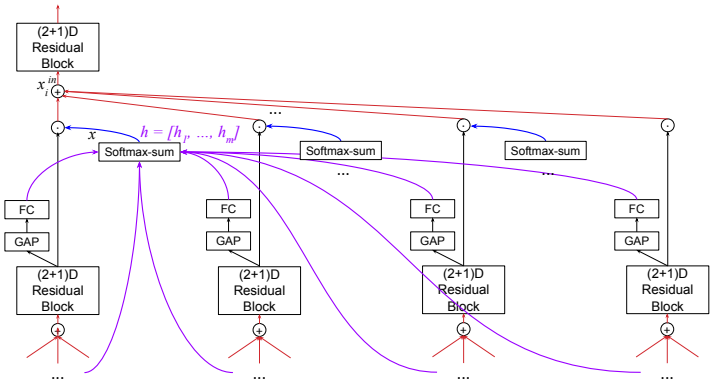


Fig. 2. Visualization of our one-shot attention search model. Magenta connections illustrate weights for the attention connection h. The softmax-sum module in the illustration corresponds to Eq. 4, fusing attentions from different blocks. These weights are fully differentiable and are learned together with convolutional filters, enabling the one-shot connectivity search.

3.6 Model implementation details

In order to provide fair comparison to previous work, we comply with the same block structure as AssembleNet [34], which by itself is comparable to (2+1)D ResNet-50.

We build two RGB input blocks (whose temporal resolutions are searched), two optical flow input blocks, and one object input block. RGB blocks and optical flow blocks have the same number of channels and layers as AssembleNet, while the object input block only has one max spatial pooling layer which does not increase the number of parameters of the model.

The object input block obtains its input from a fixed object segmentation model trained independently with the ADE-20K object segmentation dataset [57]. We treat this module as a blackbox and do not propagate gradients into it. Because this is an off-the-shelf segmentation module and was not trained on any video dataset, its outputs become noisy when directly applied to video datasets as shown in Figure 3.

Our model has convolutional blocks of four levels (five levels if we count input blocks). The sum of channel sizes are held as a constant at each level (regardless

the number of blocks), in order to maintain the total number of parameters. The total channels are 128 at input level, and 128, 256, 512, and 512 at levels 1 to 4 following the ResNet module and block formulation. As a result, all models have equivalent number of parameters to standard two-stream CNNs with (2+1)D residual modules.

Each convolutional block was implemented by alternating 2-D residual modules and (2+1)D residual modules as was done in [43, 34]. (2+1)D module is composed of 1D temporal convolution layer followed by 2D spatial convolution layer, followed by 1x1 convolution layer. The temporal resolution of each block is controlled using temporally dilated 1-D convolution, avoiding hard frame downsampling. More details of the blocks are in the supplementary material.

Although the number of blocks at each level could be hand-designed, we use AssembleNet architecture search (with an evolutionary algorithm) to find the optimal combination of convolutional blocks and their temporal resolutions. Once we have the blocks, we connect blocks with weighted connections (doing weighted summation) following Section 3.3. Finally, the one-shot attention search model obtained by implementing our peer-attention with softmax-weighted-sum, as described in Section 3.5.

Approach summary The overall process could be summarized as follows:

1. Prepare blocks. We use AssembleNet evolution to find convolutional blocks, roughly connected.
2. Initialize our one-shot search model by including all possible block connections as well as new attention connections, as described in Sections 3.3∼3.5.
3. Train the one-shot model, learning the attention connectivity weights.
4. Prune low weight connections to make the model more compact. We maintain only one peer-attention per block.

We name our final approach specifically as AssembleNet++.

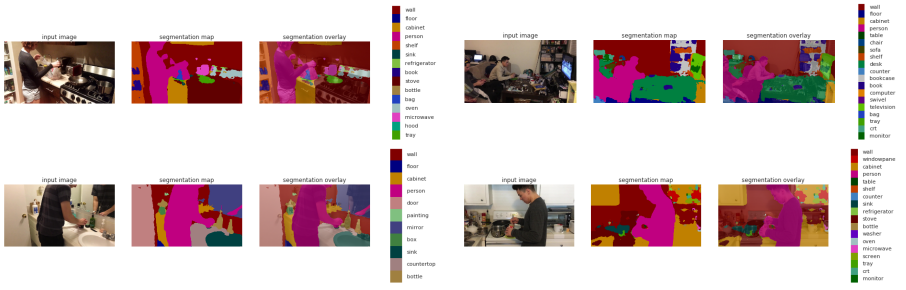


Fig. 3. Examples of the segmentation CNN applied directly on Charades video frames with in-home activities. These noisy masks serve as an input to the object input block, suggesting that our video model is required to learn to handle such noisy input.

4 Experimental results

We conduct experiments on popular video recognition datasets: multi-class multi-label Charades [36], and also the recent Toyota Smarthome dataset [5], which records natural person activities in their homes.

We note that we report results **without any pre-training** on a large-scale video dataset, which is unlike most of previous work. Regardless of that, AssembleNet++ outperforms prior work. We conduct multiple ablation experiments to confirm the benefit of our multi-modal model formulation with peer-attention and our one-shot attention search.

Charades dataset. The Charades dataset [36] is composed of continuous videos of humans interacting with objects. This dataset is a multi-class multi-label video dataset with a total of 66,500 annotations. The videos in the dataset involve motion of small objects in real-world home environments, making it a very challenging dataset. Example video frames of the Charades dataset could be found in Figure 3. We follow the standard v1 classification setting of the dataset, reporting mAP %. We use Charades as our main dataset for ablations, as it is a realistic dataset explicitly requiring modeling of interactions between object information and other raw inputs such as RGB.

Toyota Smarthomes dataset The Toyota Smarthomes dataset [5] consists of real-world activities of humans in their daily lives, such as reading, watching TV, making coffee or breakfast, etc. Humans often interact with objects in this dataset (e.g., 'drink from a can' and 'cook-cut'). The dataset contains 16,115 videos of 31 action classes, and the videos are taken from 7 different camera viewpoints. We only use RGB frames from this dataset, although depth and skeleton inputs are also present in the dataset.

Baselines As a baseline model, we use AssembleNet architecture backbone [34] which consists of multiple configurable (2+1)D ResNet blocks. Ablations, including our models without the object input block and without peer-attention, are also implemented and compared.

In our ablation experiments which compare different aspects of the proposed approach (Sections 4.1, 4.2, 4.4, and 4.5), we train the models for 50K iterations with cosine decay for the Charades dataset. When using the Toyota dataset, we train our models for 15K iterations with cosine decay as this is a smaller dataset than Charades (66,500 annotations in Charades vs. 16,115 segmented videos in Toyota Smarthome). Further, when comparing against the state-of-the-art, we use the learning rate following a cosine decay function with 'warm-restart' [23], which we discuss more in Section 4.3.

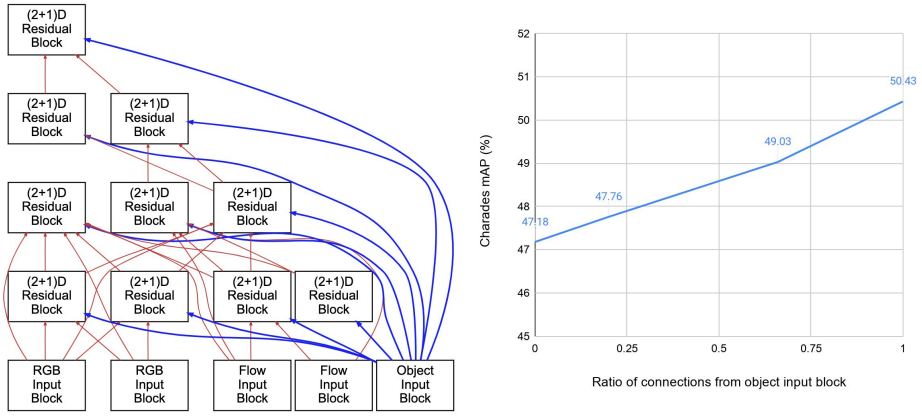
Since the model is a one-shot architecture search to discover the attention connectivity, training is efficient and takes only 20$\sim$30 hours.

4.1 Using object modality

In this ablation experiment, we explore the importance of the object input. For this study, our model learns the block connectivity from the Charades training, while not using any attention (i.e., they look like Figure 1 (a)).

Figure 4 (a) shows the best connectivity the one-shot model discovered. This is obtained by (i) evolving the blocks with 100 rounds of architecture evolution, (ii) connecting all blocks, (iii) training the weights in one-shot, and then (iv) pruning the low-weight connections. The connections weights w_{ji} with values higher than 0.2 are visualized. Interestingly, the best model is obtained by connecting the object input block to every possible block. The model with this 'omnipresent' object connectivity obtains 50.43 mAP on Charades compared to 47.18 mAP of the model without any object connections, which attests the the usefulness of the object modality. The learned weights of each object connection is more than 0.7, suggesting the strong usage of it.

Motivated by the finding that the usage of object information at every block is beneficial (i.e., omnipresent object modality connectivity), we ran an experiment to investigate how performance changes with respect to the best models found with different number of object connections. Figure 4 (b) shows the Charades classification performances of our best found models with full vs. restricted object input usage. X-axis of the graph corresponds to how often the model uses the direct input from the object input block. 0 means that it does not use object information at all, and 1 means it fuses the object information at every block. We are able to clearly observe that the performance increases proportionally to the usage of the object information.



(a) Learned connectivity between blocks

(b) Charades performance with respect to the number of object connections

Fig. 4. (a) Learned connectivity graph of the model and (b) Charades classification performance per object connection ratio. The highlighted blue edges correspond to the direct connections from the object input block.

4.2 Attention search

Next, we confirm the effectiveness of our proposed AssembleNet++ with attention search. Table 1 illustrates how much performance gain we get by using attention connections as opposed to the standard weighted connections.

In addition to attention connectivity with self-attention (Figure 1 (b)) and peer-attention (Figure 1 (c)), we implemented and tested 'static attention'. This is when learning fixed weights not influenced by any input. We are able to observe that our approach of one-shot attention search (with peer-attention) greatly improves the performance. The benefit was even higher (i.e., by ∼6% mAP) when using the object input.

Table 1. Comparison between performance with and without attention connections on Charades (mAP). The models were trained for 50K iterations.

Attention	without object	with object
None	47.18	50.43
Static	48.82	51.15
Self	51.91	55.40
Peer	52.39	**56.38**

4.3 Comparison to the state-of-the-art

In this section, we compare the performance of our AssembleNet++ model with the previous state-of-the-art approaches. We use the model with optimal peer-attention found using our one-shot attention search, and compare it against the results reported by previous work. Unlike most of the existing methods benefiting from pre-training with a large-scale video dataset, we demonstrate that we are able to outperform state-of-the-art **without such pre-training**. Below, we show our results on Charades and Toyota Smarthome datasets.

We also note that the proposed learned attention mechanisms are very powerful, as also seen in the ablation experiments in Section 4.2, and even without object information and without pre-training can outperform, or be competitive to, the state-of-the-art.

Charades dataset Table 2 shows the results of our method on the Charades dataset. Notice that we are establishing a new state-of-the-art number on this dataset, outperforming previous approaches relying on pre-training. Further, we emphasize that our model is organized to have a maximum of 50 convolutional layers as its depth, explicitly denoting it as 'AssembleNet++ 50'. Our model performs, without pre-training, even superior to AssembleNet with the depth of 101 layers that uses a significantly larger number of parameters. We also

Table 2. Classification performance on the Charades dataset (mAP).

Method	Pre-training	mAP
Two-stream [35]	UCF101	18.6
CoViAR [52] (Compressed)	ImageNet	21.9
Asyn-TF [35]	UCF101	22.4
MultiScale TRN [56] (RGB)	ImageNet	25.2
I3D [4] (RGB-only)	Kinetics	32.9
I3D from [48] (RGB-only)	Kinetics	35.5
I3D + Non-local [48] (RGB-only)	Kinetics	37.5
EvaNet [28] (RGB-only)	Kinetics	38.1
STRG [49] (RGB-only)	Kinetics	39.7
LFB-101 [51] (RGB-only)	Kinetics	42.5
SGFB-101 [16] (RGB-only)	Kinetics	44.3
SlowFast-101 [9] (RGB+RGB)	Kinetics	45.2
Two-stream (2+1)D ResNet-101	Kinetics	50.6
AssembleNet-50 [34]	MiT	53.0
AssembleNet-50 [34]	Kinetics	56.6
AssembleNet-101 [34]	Kinetics	58.6
AssembleNet-50 [34]	None	47.2
AssembleNet++ 50 (ours) without object	None	54.98
AssembleNet++ 50 (ours)	None	**59.8**

note that the use of object modality and attention mechanism proposed here, improves the corresponding AssembleNet baseline by +12.6%.

For this experiment, we use the learning rate with 'warm-restart' [23]. More specifically, we use a cosine decay function that restarts to a provided initial learning rate at every cycle. The motivation is to train our models with an identical amount of training iterations compared to the other state-of-the-art. We apply 100K training iterations with two 50K cycles while only using Charades videos, in contrast to previous work (e.g., AssembleNet [34] and SlowFast [9]) that used 50K pre-training with another dataset and did 50K fine-tuning with Charades on top of it. We note that the results of our model without 'warm-restart' and without pre-training (as seen in the ablation results in Table 1) at 56.38 are also very competitive to the state-of-the-art.

Toyota Smarthome dataset We follow the dataset's Cross-Subject (CS) evaluation setting, and measure performance in terms of two standard metrics of the dataset [5]: (1) activity classification accuracy (%) and (2) 'mean per-class accuracies' (%). Table 3 reports our results. Compared to [5], which benefits from Kinetics pre-training and additional 3D skeleton joint information, we obtain superior performance while training the model from scratch and without skeletons. We believe we are establishing new state-of-the-art numbers on this dataset.

Table 3. Performance on the Toyota Smarthome dataset. Classification % and mean per-class accuracy % are reported. Note that our models are being trained from scratch without any pre-training, while the previous work (e.g., [5]) relies on Kinetics pre-training.

Method	Classification %	mean per-class
LSTM [25]	-	42.5
I3D (with Kinetics pre-training)	72.0	53.4
I3D (pre-trained) + NL [48]	-	53.6
I3D (pre-trained) + separable STA [5]	75.3	54.2
Baseline AssembleNet-50	77.77	57.42
Baseline + self-attention	77.59	57.84
Ours (object + self-attention)	79.08	62.30
Ours (object + peer-attention)	**80.64**	**63.64**

Table 4. Comparing AssembleNet++ using peer-attention vs. a modification using 1x1 convolutional layer instead of attention. They use an identical number of parameters. Charades classification accuracy (mAP) and Toyota mean per-class accuracy (%) are reported.

Model	Charades	Toyota
Base	50.43	59.16
Base + 1x1 conv.	50.24	59.44
Random peer-attention	53.40	60.23
Our peer-attention	**56.38**	**63.64**

4.4 Ablation

In this experiment, we explicitly compare AssembleNet++ using peer-attention with its modifications using the same number of parameters. Specifically, we compare our model against (i) the model using 1x1 convolutional layers instead of attention and (ii) the model using peer-attention but with random attention connectivity. For (i), we make the number of 1x1 convolutional layer parameters identical to the number of parameters in FC layers for attention. Table 4 compares the accuracies of these models on Charades and Toyota Smarthome datasets. While using the identical number of parameters, our one-shot peer-attention search model obtains superior results.

4.5 General applicability of the findings

Based on the findings that (1) having 'omnipresent' neural connectivity from the object modality and (2) using attention connectivity are beneficial, we investigate further whether such findings are generally applicable for many different CNN models. We add object modality connections and attention to (i) standard R(2+1)D network, (ii) two-stream R(2+1)D network, (iii) original AssembleNet, and (iv) our Charades-searched network (without object connectivity and atten-

tion), and observe how their recognition accuracy changes compared to the original models. Our model without object and attention is obtained by manually removing connections from the object input block.

Table 5 shows the results tested on Charades. We are able to confirm that our findings are applicable to other manually designed, as well as, architecture searched architectures. The increase in accuracy is significant for all architectures. Note that our architecture itself is not significantly superior to AssembleNet without object. However, since its connectivity was searched together the object input block (i.e., Section 3.5), we are able to observe that our model better takes advantage of the object input via peer attention. 50K training iterations with cosine decay was used for this comparison.

Table 5. Comparison between original CNN models (without object modality and without attention) and their modifications based on our attention connectivity and object modality. The value corresponding to 'AssembleNet++' for the column 'base' is obtained by manually removing connections from the object input block and removing attention from our final one-shot attention search model. Measured with Charades classification (mAP, higher is better), trained from scratch for 50k iterations.

CNN model	base	+ object + attention
RGB R(2+1)D	36.51	**45.30**
Two-stream R(2+1)D	39.93	**47.74**
AssembleNet	47.18	**53.48**
AssembleNet++	47.62	**56.38**

5 Conclusion

We present a family of novel video models which are designed to learn interactions between the object modality input and the other raw inputs: AssembleNet++. We propose connectivity search to fuse new object input into the model, and introduce the concept of peer-attention to best capture the interplay between different modality representations. The concept of peer-attention generalizes previous channel-wise self-attention by allowing the attention weights to be computed based on other intermediate representations. An efficient differentiable one-shot attention search model is proposed to optimize the attention connectivity. Experimental results confirm that (i) our approach is able to appropriately take advantage of the object modality input (by learning connectivity to the object modality consistently) and that (ii) our searched peer-attention greatly benefits the final recognition. The method outperforms all existing approaches on two very challenging video datasets with daily human activities. Furthermore, we confirm that our proposed approach and the strategy are not just specific to one particular model but is generally applicable for different video CNN models, improving their performance notably.

References

1. Ahmed, K., Torresani, L.: Connectivity learning in multi-branch networks. In: Workshop on Meta-Learning (MetaLearn), NeurIPS (2017)
2. Baradel, F., Neverova, N., Wolf, C., Mille, J., Mori, G.: Object level visual reasoning in videos. In: Proceedings of European Conference on Computer Vision (ECCV) (2018)
3. Bender, G., Kindermans, P.J., Zoph, B., Vasudevan, V., Le, Q.: Understanding and simplifying one-shot architecture search. In: International Conference on Machine Learning (ICML) (2018)
4. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
5. Das, S., Dai, R., Koperski, M., Minciullo, L., Garattoni, L., Bremond, F., Francesca, G.: Toyota smarthome: Real-world activities of daily living. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019)
6. Diba, A., Fayyaz, M., Sharma, V., Paluri, M., Gall, J., , Stiefelhagen, R., Gool, L.V.: Holistic large scale video understanding. In: arxiv.org/pdf/1904.11451 (2019)
7. Fan, L., Huang, W., Gan, C., Ermon, S., Gong, B., Huang, J.: End-to-end learning of motion representation for video understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
8. Feichtenhofer, C., Pinz, A., Wildes, R.: Spatiotemporal residual networks for video action recognition. In: Advances in Neural Information Processing Systems (NeurIPS) (2016)
9. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019)
10. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
11. Girdhar, R., Ramanan, D., Gupta, A., Sivic, J., Russell, B.: Actionvlad: Learning spatio-temporal aggregation for action classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 971–980 (2017)
12. Hara, K., Kataoka, H., Satoh, Y.: Learning spatio-temporal features with 3d residual networks for action recognition. In: Proceedings of the ICCV Workshop on Action, Gesture, and Emotion Recognition. vol. 2, p. 4 (2017)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
14. Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
15. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019)
16. Ji, J., Krishna, R., Fei-Fei, L., Niebles, J.C.: Action genome: Actions as composition of spatio-temporal scene graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)

17. Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(1), 221–231 (2013)
18. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
19. Li, X., Zhong, Z., Wu, J., Yang, Y., Lin, Z., Liu, H.: Expectation-maximization attention networks for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019)
20. Liu, C., Zoph, B., Neumann, M., Shlens, J., Hua, W., Li, L.J., Fei-Fei, L., Yuille, A., Huang, J., Murphy, K.: Progressive neural architecture search. In: Proceedings of European Conference on Computer Vision (ECCV) (2018)
21. Liu, H., Simonyan, K., Yang, Y.: DARTS: Differentiable architecture seach. In: International Conference on Learning Representations (ICLR) (2019)
22. Long, X., Gan, C., de Melo, G., Wu, J., Liu, X., Wen, S.: Attention clusters: Purely attention based local feature integration for video classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7834–7843 (2018)
23. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. In: International Conference on Learning Representations (ICLR) (2017)
24. Ma, M., Fan, H., Kitani, K.M.: Going deeper into first-person activity recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
25. Mahasseni, B., Todorovic, S.: 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 (CVPR) (2016)
26. Moore, D.J., Essa, I.A., HayesIII, M.H.: Exploiting human actions and object context for recognition tasks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (1999)
27. Nekrasov, V., Chen, H., Shen, C., Reid, I.: Architecture search of dynamic cells for semantic video segmentation. In: CoRR:1904.02371 (2019)
28. Piergiovanni, A., Angelova, A., Toshev, A., Ryoo, M.S.: Evolving space-time neural architectures for videos. Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019)
29. Piergiovanni, A., Fan, C., Ryoo, M.S.: Learning latent sub-events in activity videos using temporal attention filters. In: Proceedings of AAAI Conference on Artificial Intelligence (AAAI) (2017)
30. Piergiovanni, A., Ryoo, M.S.: Representation flow for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
31. Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3d residual networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 5533–5541 (2017)
32. Ray, J., Wang, H., Tran, D., Wang, Y., Feiszli, M., Torresani, L., Paluri, M.: Scenes-objects-actions: A multi-task, multilabel video dataset. In: Proceedings of European Conference on Computer Vision (ECCV) (2018)
33. Real, E., Aggarwal, A., Huang, Y., Le, Q.V.: Regularized evolution for image classifier architecture search. In: Proceedings of AAAI Conference on Artificial Intelligence (AAAI) (2019)

34. Ryoo, M., Piergiovanni, A., Tan, M., Angelova, A.: AssembleNet: Searching for multi-stream neural connectivity in video architectures. In: International Conference on Learning Representations (ICLR) (2020)
35. Sigurdsson, G.A., Divvala, S., Farhadi, A., Gupta, A.: Asynchronous temporal fields for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
36. Sigurdsson, G.A., Gupta, A., Schmid, C., Farhadi, A., Alahari, K.: Charades-ego: A large-scale dataset of paired third and first person videos. arXiv preprint arXiv:1804.09626 (2018)
37. Sigurdsson, G.A., Russakovsky, O., Gupta, A.: What actions are needed for understanding human actions in videos? In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)
38. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 568–576 (2014)
39. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning (ICML) (2019)
40. Taylor, G.W., Fergus, R., LeCun, Y., Bregler, C.: Convolutional learning of spatio-temporal features. In: Proceedings of European Conference on Computer Vision (ECCV) (2010)
41. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2015)
42. Tran, D., Bourdev, L.D., Fergus, R., Torresani, L., Paluri, M.: C3d: generic features for video analysis. CoRR, abs/1412.0767 **2**(7), 8 (2014)
43. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6450–6459 (2018)
44. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (NeurIPS) (2017)
45. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3169–3176. IEEE (2011)
46. Wang, L., Li, W., Li, W., Gool, L.V.: Appearance-and-relation networks for video classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
47. Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., Hu, Q.: Eca-net: Efficient channel attention for deep convolutional neural networks. arXiv:1910.03151 (2019)
48. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7794–7803 (2018)
49. Wang, X., Gupta, A.: Videos as space-time region graphs. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 399–417 (2018)
50. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of European Conference on Computer Vision (ECCV) (2018)
51. Wu, C.Y., Feichtenhofer, C., Fan, H., He, K., Krähenbühl, P., Girshick, R.: Long-term feature banks for detailed video understanding. arXiv preprint arXiv:1812.05038 (2018)

52. Wu, C.Y., Zaheer, M., Hu, H., Manmatha, R., Smola, A.J., Krähenbühl, P.: Compressed video action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6026–6035 (2018)
53. Xie, S., Kirillov, A., Girshick, R., He, K.: Exploring randomly wired neural networks for image recognition. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 1284–1293 (2019)
54. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 305–321 (2018)
55. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime tv-l 1 optical flow. In: Joint Pattern Recognition Symposium. pp. 214–223. Springer (2007)
56. Zhou, B., Andonian, A., Oliva, A., Torralba, A.: Temporal relational reasoning in videos. In: Proceedings of European Conference on Computer Vision (ECCV). pp. 803–818 (2018)
57. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
58. Zoph, B., Le, Q.: Neural architecture search with reinforcement learning. In: International Conference on Learning Representations (ICLR) (2017)
59. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)