Action Recognition with 3D Dilated Convolutions.

Rohit Venkata Sai Dulam
University of Rochester
Rochester, NY, USA
rdulam@ur.rochester.edu

Abstract

With the advent in the field of Deep Learning, many problems that have been thought unsolvable are turning out into reality. Most of the Vision related problems are being made possible by Convolutional Neural Networks. One such field is Action Recognition, where 3D Convolutional Networks with different modifications to them have become state-of-the-art models. I propose an architecture called 3D Dilated Convolutional Network, which as the name suggest, is based on 3D Dilated Convolutions. To my knowledge, this is the first time that Dilated Convolutions have been used to tackle the problem of Action Recognition.

1. Introduction

Activity Recognition involves the identification of various actions from a sequence of 2D frames. To solve tackle this issue, a novel architecture should be created which can identify what action is taking place and the interactions happening while the action is being performed. With the recent advent in the field of Deep Learning, most of the solutions or models created to tackle the problem are mostly based on Convolutional Architectures. 3D Convolutions are at the crux of all the models that have been proposed in the recent time. The only thing that is being sacrificed here is the time taken to train the model. If we take a normal 3D Convolutional Neural network, it takes around 3 days to train on the [10] Action dataset. That is a lot of time, nowhere close to being used in real time settings. Most of the recently proposed architectures use 2 Stream Convolutional Networks, which are even harder and take a longer time to train. Though efficient methods are coming up to make these huge models viable for real-time inference, they will be nowhere near that anytime soon. The above-stated reasons have driven me to try something new and something that can reduce the computation time and in turn, the inference time substantially. One more serious issue seems to be the capturing of long-range dependencies. It is well known that Action Recognition is one such task where important parts from all the frames in a sequence should be understood in order to determine a particular action, the minute interactions going on between different actors in the scene etc. The recent architectures have been trying to solve this issue, but they do not seem the best of the ways given the computation time. Therefore, I propose a similar 3D Convolutional Network, but with slight modification, which is the addition of Dilated Convolutions. Dilated Convolutions have originally excelled at Object Segmentation tasks where the output should be of the same size as the input and there shouldn't be any loss of resolution. The exponential increase in the receptive field with a linear increase in parameters has motivated to implement them for the task of Action Recognition.

My work is a study on how Dilated Convolutions effect the computation time and the accuracy of the model compared to a baseline 3D Convolutional Network. One other addition to this work is a new type of learning mechanism which will help the model learn better. The proposed architecture is trained in [13] action dataset, which contains 50 action categories. The use of [13] instead of [10] is justifiable as the data in the former is enough to train the model and get a respectable accuracy. The use of a new type of loss function, which makes the network learn faster, helps the model to converge faster. In order to understand if the model has understood what an action means, the model will try to reconstruct a few frames of the input video using the output from the 3D Dilated Neural Network. The Reconstruction network which is a simple De-Convolutional Neural network uses the same weights as that of the 3D Convolutional Network in order to capture better features. This reconstruction, I believe will help the network learn faster and better.

To summarize, the following are the improvements over a traditional 3D Convolutional Model used for the task of Action Recognition :

1. Introduction of Dilated Convolutions to test their impact.

2. Tackling the problem of Long range temporal dependencies using Non-Local blocks.

3. Devising a new loss function which intuitively makes sense and helps the model learn better.

With further research, this reduction in computation time can help inculcate better learning mechanisms and architectures. The second point couldn't be confirmed. Reasons are stated in the Conclusion section.

1.1. Related Work

Action Recognition. Action recognition aims to identify the actions and goals of one or more actors from a series of observations on the actors' actions and the environmental conditions. Considering temporal features will help the model better understand how an action changes the position of an object dynamically. Though we aren't explicitly telling the model to understand what the action is, the model will understand the change in position, orientation etc of that particular object. It is as if like teaching a kid to focus on a particular object, look closely how it is moving without telling what is causing that movement.

Recurrent Convolutional Networks. [3][20] Some of the initial works on Video understanding where based on Encoder-Decoder architectures, where the Encoder is a Convolutional Neural Network that extracts information from the input visual sequence, and the Decoder being a Recurrent Neural Network, since these architectures were somewhat related to Natural Language Processing. The 3-D CNN representation in [20] is trained on video action recognition tasks, so as to produce a representation that is tuned to human motion and behavior.

3D Convolution Networks. [8]3-dimensional Convolutional Neural Networks are very similar to that of 2 Dimensional Convolutions where the latter extracts features in the spatial domain, whereas the former extracts features both in the spatial along with the temporal domains. Extraction of features in the temporal domain becomes a very important aspect, as the model should truly understand how an action changes over a period of time, in order to truly recognize an action. This cannot be done if we feed in a single frame which best depicts the action performed by an actor, as they aren't as smart as we humans are. The temporal dependencies should be modeled, and that can be done if we give the required appropriate information. In the proposed work, I've stuck to a basic 3D Convolutional Network with Dilated Convolutions[12].

Two Stream Architectures.[9] [16] [4][18][5]Two stream architectures have been used by most of the state-of-the-art models in the field of Action Recognition. The two streams being the Spatial stream, extracting features in the spatial domain using a pretrained 3D Convolutional Network. The second stream is the Optical flow given as input to capture long-term temporal dependencies. A lot of works have focused on how these two domains could be combined to get the best results. Though this idea seems to work and is giving out promising results, that is not how we humans learn actions.This[21] paper seems to move away from the traditional 2 stream architectures, they stack another Convolutional Neural Network to capture the temporal dependencies. We humans do not look for a change in the position of the actor in each frame in order to understand an action. There's something else to it. With the computation power at our disposal, and the humongous amounts of data being available, has led to such approaches.

Dilated Convolutions. The use of Dilated Convolutions has been a proven fact in the field of instance segmentation from [12]. The same approach has been used in the proposed model as well, the addition being using Dilated Convolutions over the temporal domain to demonstrate or test its efficiency in the temporal domain. The advantage with Dilated Convolutions being the exponential increase in the receptive field increases the parameter linearly, which helps bring down the computational costs of the network. This is the first time that Dilated Convolutions have been used to tackle the problem of Action Recognition.

Encoder - Decoder Architecture. Encoder - Decoder architectures have been used for a variety of tasks ranging from Image Captioning methods to Image segmentation[1][14] and Video segmentation tasks. The second model in the paper is very similar to an Encoder-Decoder architecture, with a slight difference that will be talked about in the following sections.

Non Local Networks. Non-local networks [19] have shown good results in capturing the temporal information of a particular object. The proposed model uses the non-local network to help capture temporal dependencies in addition to the 3D Dilated Convolution operation. Non Local blocks couldn't be added to the network due to computational limitations.

The proposed model incorporates the advantages of all the above discussed methodologies and tries to create a novel solution to solve the task of Action Recognition in Videos. This method will also demonstrate if there is sizable difference between models trained from scratch and other pretrained models like [7] which are often used as a prior or base network. This model is in a way an experimental method which transforms 3 Dimensional Convolution to Dilated 3 Dimensional Convolution, which helps us observe what kind of an effect the latter has in the temporal domain. Also, in order to test the viability of the newly formed loss function, I've used two networks to be trained and tested on the same dataset, for the same number of epochs. The details about the architecture of both the networks is listed in tables 1 and 2.

2. Method Formulation.

Convolutional Neural Networks[11] have always been at the heart of solving problems related to the domain of Computer Vision. Convolutional Neural Networks have been so effective because of their power to extract features efficiently given enough data, and also their intuitiveness. 2 Dimensional Convolutional Neural Networks are widely used to capture local or spatial information given an image or scene. Most of the tasks involving Classification, Recognition, Segmentation etc make active use of these networks at their core. The same Convolutional Networks are at the heart of models tackling problems related to Videos or a sequence of frames. Similar to a 2D Convolutional Neural Network which captures information in the spatial domain, 3D Convolutional Neural Networks captures information in the spatiotemporal domain, i.e. both the Spatial and Temporal dimensions concurrently. Variations of Convolutional Kernels are shown in figure-1.

2.1. Dilated Convolutions.

Dilated Convolutions have shown great performance in image segmentation tasks[12]. My idea to try out Dilated Convolutions in place of normal Convolutions was a result of reading a number of papers which used Dilated Convolutions for various tasks. [2][6] These are the tasks that require the model to not reduce the resolution of the image and extracting features. Dilated Convolutions use lesser parameters but achieve far greater receptive field as they insert zeros in between. The increase in receptive field is exponential with a linear increase in the number of parameters. This inspired me to use them in the network to see if they are able to capture temporal dependencies based on the higher receptive field. The detailed explanation on how Dilated Convolutions have influenced the network is written in the results and conclusion section.

2.2. Reconstruction Network.

The second model uses a Reconstruction network to recreate few of the frames. The idea behind recreating frames is for the network to learn how an action is. How will the network learn? Since the recreated frames are compared to ground truth frames in a Supervised fashion, the network has to be better at recreating them in order to reduce the error between the recreated and the ground truth frames. The reconstruction network uses the filters of the Convolutional Layers in order to reconstruct frames. While training, these filters are optimized in order to be good at reconstruction, which inturn optimizes the Convolutional Network to extract features more efficiently to help reconstruct better.

2.3. Loss Function.

Since there are two methods, the loss for the first model is just straightforward cross entropy loss. Model - 2 uses the newly devised loss function. The motivation for this loss function came from its analogy to human beings. Human beings try to recreate things that they've learned in order to perfect them. This holds true for most of the things humans try to do. I wanted to use this analogy and devise a loss function. Though it is a Supervised Learning setting, the model is constructed in a way such that it can really understand the action that is taking place. For the model to be accurate, it has to understand what an action is, and should be able to reconstruct or reproduce that particular action. This is how the proposed model learns to recognize the actions of different sports. The Convolutional Model first extracts features out of the input. These extracted features are sent to a 3D De-Convolutional Network to recreate or reconstruct some of the frames. The reconstruction of the frames is done with the help of filters used in the initial Convolutional Network that extracts information from the initial input. One more thing to note here is, the first network which extracts information is trained to classify actions during the training phase by comparing its output with the ground truth label. This is done in order for the fully connected layers to learn, and for the network on the whole to extracts features better, which will help the reconstruction network to better recreate those frames. This intuitively is very encouraging, because, to tell whether a model has learned something can only be done if it can reconstruct the same thing again, which is being done here. The normal 3D Dilated Convolutional Network acts as a classifier, which is trained by comparing with the ground truth labels, the reconstruction network is trained by comparing its output to that of ground truth frames. The trick here is to use the same weights to extract information from the input and also using them to reconstruct the frames from the output of the Dilated network. The loss function is illustrated below.

$$l_{total} = l_{Classification} + \alpha \ \times l_{Reconstruction} \tag{1}$$

The classification loss is the Cross Entropy loss between the predicted and the ground truth labels, whereas the reconstruction loss is the Mean Square error between the reconstructed frames and the ground truth frames. Alpha is an experimental parameter, to help penalize the model if it is bad at reconstructing frames. This alpha helps make the reconstruction loss more important, which in turn will update the weights of both the reconstruction and the 3D Dilated Convolutional Network, since the same weights are used to extract features and reconstruct frames.

3. Experiments.

To determine the effectiveness of the proposed model, we use the UCF50[13] Sports Action Recognition Dataset. The dataset contains 50 sports action classes. The dataset has been annotated with the class numbers in order to make it a Supervised learning problem. The training phase consists of 3600 videos trained on 3 epochs due to hardware

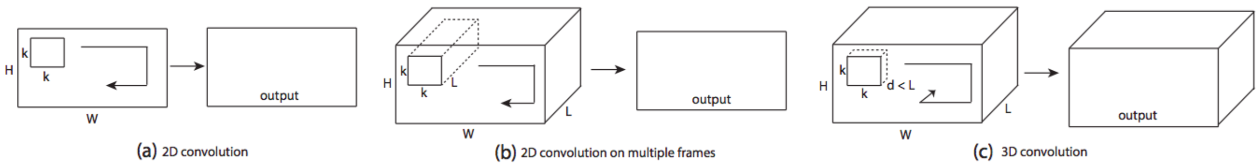

Figure 1: **(a)** shows the output of a 2D filter is convolved with an image. **(b)** shows when a 2D filter is convolved with multiple frames. **(c)** shows when a 3D filter is convolved with a sequence of frames. Source [17]

limitations.

3.1. Implementation Details.

Both the networks have been built using TensorFlow. In the second model, both the 3D Convolutional Neural Network and the Reconstruction Network share their weights, in order for the weights to be trained efficiently and also to capture more information. The videos from the [13] dataset are used as it is with the frames of the videos being clipped to 60 and but the frame size for the first model being set to 240X320. For the second model, the frame size was reduced to half i.e. 120X160. The Dilated Convolutional Network has 9 Convolutional layers followed by a Fully Connected Layer. The architectures of both the networks are illustrated in table-1 and 2. Due to the computational limitations, the batch size was restricted to just 4 set of sequence of frames or videos. Both the models were run on a NVIDIA Tesla K80 GPU with 11Gb of VRAM provided by Kaggle for a single user alongwith 20 GB of free disk space. Here, the time limit to run a particular model was 6 hours, which restricted the number of epochs I could train the network on. Few things to note about the first model.

1. The padding has been set to VALID since dilated convolutions in TensorFlow do not work with padding set to SAME.

2. Some of the choices of architecture had been made due to hardware constraints.

3. 1st, 3rd and the 5th rows are Dilated Convolutions. Explained in detail in the code.

4. 1X1 Convolutions have been used in order to aggregate the information from multiple channels and also to reduce dimensionality.

Few things to note about the second model.

1. The output from the 5th Convolution layer i.e. a tensor of shape 8 X 15 X 20 X 512 is used as input to the reconstruction network.

2. The filters used from second to fifth Convolution layers are used by the reconstruction network to reconstruct 16 frames.

3. The dimensionality of the inputs are reduce by the virtue of Max pooling layers.

4. The final Convolution layer is to aggregrate information and reduce the number of channels to 1.

3.2. Action Recognition Results.

3.2.1 UCF50.

This section consists of results of using different architectures for the same task. There are two types of architectures namely,

1. Action Recognition with a 3D Dilated Convolutional Neural Network.

2. Action Recognition with a 3D Dilated Convolutional Network and the Reconstruction Network with lower resolution video as input.

For the first architecture, I've retained the picture quality of the original videos from the [13] dataset i.e. 240 X 320, with the number of frames being set to 60. The number of frames is same for both the networks. The difference being, in the second network, the frame size has been reduced to half, i.e. 120 X 160. In both the networks, frames have been converted to grayscale from RGB as the point here is to recognize the action and converting the frame to grayscale wouldn't make a huge difference since the element changing from frame to frame will be the same. The first network gave out an accuracy of around 12.5% and the second network gave out an accuracy of 30%. For the testing phase, I randomly picked 8 videos for the first method and 10 videos for the second method and they where classified by the first and second network respectively. This was done for ease of computation. All the videos in the test set where randomly shuffled before being tested on, and there were many instances, where the second network got 3 of the 10 videos right. The highest the second model could classify at best was 5 videos. I personally checked which class they belonged to, and they were all random and different, which justifies the result. The first model could only classify 2 videos at best in the test set, and many of the other instances where it only could classify 1 video correctly out

	Architecture	output
Model - 1	16 X 3*3*3 filters	56 X 236 X 316 X 16
	32 X 3*3*3 filters	27 X 234 X 314 X 32
	64 X 3*3*3 filters	23 X 230 X 310 X 64
	128 X 3*3*3 filters	11 X 76 X 77 X 128
	128 X 3*3*3 filters	7 X 72 X 73 X 128
	256 X 3*3*3 filters	3 X 35 X 36 X 256
	64 X 3*1*1 filters	1 X 35 X 36 X 64
	16 X 1*3*3 filters	1 X 33 X 34 X 16
	1 X 1*1*1 filter	1 X 33 X 34 X 1
	FC Layer	

Table 1: Illustrates the architecture of the first network.

	Architecture	output
Model - 2	32 X 7*7*7 filters	60 X 120 X 160 X 32
	64 X 3*3*3 filters	60 X 120 X 160 X 64
	128 X 3*3*3 filters	30 X 60 X 80 X 128
	256 X 3*3*3 filters	15 X 30 X 40 X 256
	512 X 3*3*3 filters	8 X 15 X 20 X 512
	768 X 3*3*3 filters	4 X 8 X 10 X 768
	1024 X 3*3*3 filters	2 X 4 X 5 X 1024
	FC Layer-1	
	FC Layer-2	
Recon Net	Deconv-1	8 X 30 X 40 X 256
	Deconv-2	8 X 60 X 80 X 128
	Deconv-3	16 X 60 X 80 X 64
	Deconv-4	16 X 120 X 160 X 32
	32 3*3*3 filters	16 X 120 X 160 X 1

Table 2: Illustrates the architecture of the second network which has two parts, the first one being a 3D Convolutional Neural Network and the second part being a reconstruction network.

of 8. Though both the results aren't ground breaking, something that I have noticed from my experiments is, both of them take the same amount of time for training on the same number of videos and for the same number of epochs. But, the difference in the accuracy is the most interesting part, the proposed loss function seems to be helping the model learn better than just using a normal Dilated Convolutional Neural Network. Despite of having less number of samples to learn from, the proposed loss function seems to be working. The value of alpha has been set to 2.0 in my experiments. Nonetheless, both the results seem to be under impressive and some of the reasons for this abysmal performance have been discussed in the conclusion section.

Method	Accuracy
Model-1	12.5%
Model-2	30%

Table 3: Results. Show that the proposed loss function can yield better results, which implies the model is able to learn.

4. Conclusion

The paper talks about tackling the problem of Action Recognition in a different manner compared to traditional state-of-the-art two stream architectures. Due to lack of proper computation power, the results did not come out as expected. Though the network has the potential to learn, it wasn't trained to its fullest. The dataset chosen, the number of epochs that could be run on the train split and the testing data resulted in a lower accuracy than expected. Though this method gave a disappointing result, the second model talked about in the results section seems to be better that normal traditional 3D Convolutional networks. This goes on to show that the proposed model, though didn't perform well, has the ability to get better by the virtue of the new loss function. Some future work can be done to use this type of intuitive learning to train a model which can perform better than the state-of-the-art models. In my experiments, dropout wasn't added, which could've boosted up the accuracy. Other tasks that might be beneficial with this loss function are Image Classification, Image and Video Object Segmentation tasks. An image classifier that can better recreate an image could lead to even better classification model. Similarly, if the network is able to reconstruct the segmentation mask, along with the rest of the image, that will boost up the accuracy. One more observation here is the computational time taken by the Dilated Convolutions. Though there are lesser parameters to train, the computational time is comparable to that of normal convolution layer with a similar receptive field. There are no gains in computation time with the addition of Dilated Convolutions and in my experience, they seem to take a long time for initialization. Capturing Temporal dependencies is one of the main problems that hasn't been answered by me. Though my idea might be very similar to that of Capsule Networks [15], they do it differently compared to mine.

5. Future Work.

The devised Loss function is intuitive, and if used for the right task, and if properly trained, will give out some good results. Few things that can be made better are

1. Use a pretrained network, since this will give a better starting point for the network rather than starting from scratch.

2. Better way to determine the value of Alpha used in the newly devised loss function.

3. Dropout can be inculcated into the network.

4. The loss function can be tested first for Image Classification tasks, where a similar setting can be used to first extract features from an Image, and then trying to reconstruct it, similar to an Enocder-Decoder architecture, with the change being the Encoder is trained as a Classifier as well.

5. More training and testing data for the network to properly learn.

6. Use a better architecture. The above used architectures aren't the best to start with, since they lose a lot of information in the temporal domain, but some of the hardware limitations forced me to stick with them.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation, 2016.

[2] N. Cheema, S. Hosseini, J. Sprenger, E. Herrmann, H. Du, K. Fischer, and P. Slusallek. Dilated temporal fully-convolutional network for semantic segmentation of motion capture data, 2018.

[3] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description, 2015.

[4] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition, 2016.

[5] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell. Actionvlad: Learning spatio-temporal aggregation for action classification, 2017.

[6] R. Hamaguchi, A. Fujita, K. Nemoto, T. Imaizumi, and S. Hikosaka. Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery, 2017.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[8] R. Hou, C. Chen, and M. Shah. An end-to-end 3d convolutional neural network for action detection and segmentation in videos, 2017.

[9] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks, 2014.

[10] A. R. Z. Khurram Soomro and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild, 2012.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks, 2012.

[12] Liang-Chieh, C. George, P. F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation, 2017.

[13] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos, 2012.

[14] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[15] S. Sabour, N. Frosst, and G. E. Hinton. Dynamic routing between capsules, 2017.

[16] K. Simoyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos, 2014.

[17] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. learning spatiotemporal features with 3d convolutional networks, 2015.

[18] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool. Temporal segment networks: Towards good practices for deep action recognition, 2016.

[19] X. Wang, R. Girshick, A. Gupta1, and K. He. Non-local neural networks, 2018.

[20] L. Yao, A. Torabi, K. Cho, C. P. Nicolas Ballas, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure, 2015.

[21] Y. Zhu, Z. Lan, S. Newsam, and A. G. Hauptmann. Hidden two-stream convolutional networks for action recognition, 2017.