

**Deep Learning for Visual Computing**  
**Prof. Debdoot Sheet**  
**Department of Electrical Engineering**  
**Indian Institute of Technology, Kharagpur**

**Lecture - 56**  
**Understanding Video Analysis**

[noise]

Welcome. Ah. In this week, [noise] we are going to start with ah one of these most interesting aspects and that's about video analysis.

(Refer Slide Time: 00:17)



The banner features the NPTEL logo on the left, followed by the text "NPTEL ONLINE CERTIFICATION COURSES" and "Indian Institute of Technology Kharagpur | Department of Electrical Engineering".

# Understanding Video Analysis

**Dr. Debdoot Sheet**  
Assistant Professor, Department of Electrical Engineering  
Principal Investigator, Kharagpur Learning, Imaging and Visualization Group  
Indian Institute of Technology Kharagpur  
[www.facweb.iitkgp.ernet.in/~debdoot/](http://www.facweb.iitkgp.ernet.in/~debdoot/)



A square QR code is located in the bottom right corner of the slide content area.

Now, till whatever we have done, ah on the lectures as of now, they were to do only with 2 D images or single frames over there and ah it was frozen over time. So, you do not have this, ah [vocalized-noise] time varying nature of these ah images or anything created out over there [vocalized-noise]. Whereas, one of the major ah chunks of analysis which we end up doing, for ah current applications are all on videos and, what comes down is that, while you can always treat down videos as is just a bunch of frames over there and then keep on operating one frame at a time [noise] and ah still get down your

results [noise] coming out of it and use them, for your further analysis or summarization ah of [vocalized-noise] whatever analysis you want to do [vocalized-noise].

The main ah point, or or the main aspect which we often tend to forget is that, there is some sort of temporal consistency between these frames as they keep on coming [noise] [vocalized-noise] and, what that necessarily means is that, ah [vocalized-noise] we should be in some way able to use ah the features which come down on the feature space over here in [vocalized-noise] in terms of [vocalized-noise] just the spatial dimensions [vocalized-noise].

Now, do they have some sort of ah [noise] cross dependent relation between the [noise] ah between time frames? [noise] Now, nonetheless ah [vocalized-noise] what should also be another interesting meaningful thing over here, is that, ah [vocalized-noise] since we have ah these videos available, which can otherwise be just considered as a chunk of frames, [vocalized-noise] then ah we should also be in a position ah somehow to run down some sort of convolutions over there.

Now, [vocalized-noise] whatever convolutions you had done, ah you were doing till now, they were all ah spatial. So, [vocalized-noise] there was obviously, a three dimensional array, but then the third dimension was at the number of channels which comes down and as, as far as till we have been concerning over here in terms of these tensors [vocalized-noise]. So, the way these are represented is that the full dimension is actually the batch size, the second dimension is where you have the number of channels, input channels over there, [vocalized-noise] the third dimension is the x axis and the fourth dimension is the y axis present over here, [vocalized-noise] And if I if I am considering something in terms of video, I need to have this extra dimension over there, which is of time [noise].

So, what we do is, ah while typically video files as such are recorded in a way, where ah [vocalized-noise] you have the time axis as something which supersedes the color axis over there, which means that in our representation it should have been batch ah dimension, then the next dimension is what [vocalized-noise] represents the time and next dimension is what represents the color channels and then, you have x and y [vocalized-noise]. But then we do not do it for a very specific and obvious reason over there because, over channels over color channels, you cannot convolved anything

[noise]. So, you need to convolve it over the time dimension over there. So, how it works out is, what I am going to do [vocalized-noise].

So, it is it is ah really hard to just keep on explaining in terms of ah speaking out over here. So, I will just be using a few of these illustrations to get this concept clear. So, today we will be doing about ah [noise] understanding the very basic concepts of, how to handle down videos [noise] and ah can I get down something on the special side of it to work out on my videos, and the next lecture, I am going to get you introduced to a sequence modeling ah concept called as recurrent neural network, which is something which we use for ah [vocalized-noise] as such, is typically being used for ah speech and natural language [vocalized-noise] processing or any kind of a sequence data analytics and then, how do we end up treating these ah [noise] ah say images, image frames over here on a video volume, in terms of ah some and and extracting some features, which can now be fed down to my ah [noise] recurrent neural network for sequence modeling [vocalized-noise]

So, this is what [noise] ah how the flow will going down ah will be going down and then, then subsequently the next day we had be looking down into how to engage ah say ah spatial convolutional network, plus a temporal learning [vocalized-noise] network with respect of a [vocalized-noise] recurrent neural network [vocalized-noise] together, in order to get some sort of an analytics done over a whole video volume [vocalized-noise].

(Refer Slide Time: 03:57)



# Organization

- Images and Videos
- Data Organization and Video Tensor
- Neural Networks for Video Handling
- Convolution on Video Tensors
- Challenges
- Spatio-Temporal Deep Learning



So, [noise] [vocalized-noise] let us ah get into the Organization over here [noise]. So, how it is done is today I am going to [noise] rebrief you this ah sort of a similarity between [noise] images and video [noise]. So, we have already done ah the images part of it when we were looking into the early courses, early early lectures in this particular course in the first week itself [vocalized-noise]. And then, how do we deal with images and then in terms of convolutions what happens over there [vocalized-noise].

Now, today I am going to extend that [noise] whole concept over to videos how it goes down [vocalized-noise]. And then, ah I will enter into the Data Organization and Video Tensor representation. So, this is what I was telling you initially [vocalized-noise] just a couple of minutes ago that, ah [vocalized-noise] what dimension of my tensor is going to represent which aspect of the data [noise] [vocalized-noise]? Subsequent to that, ah I will be getting you introduced to ah Neural Networks for Video Handling and this [noise] this will be very specifically just convolution neural network. Because as such, [vocalized-noise] if you have a fully connected network over there, then it becomes easy because you can just [vocalized-noise] stand down all your neurons over there irrespective of whatever you are doing and then, ah [vocalized-noise] ah it can directly be used. But, the moment you are speaking about in terms of convolutions and it does not become so easy. So, here comes in a trick like what axis [noise] do you convolve around for each of them [vocalized-noise]? And, what happens to the number of channels, because now your data

technically becomes a four d data [noise] if you have a colored video representation [vocalized-noise] [noise].

So, that is what we will be doing down in this [noise] Neural Networks for video handling and then, enter into convolutions on video tensors and what is the input and output size relationships between each of them and take in a critical case [vocalized-noise] ah a very typical case and see down what is the criticality of [vocalized-noise] ah handling on direct temporal convolutions, ah spatiotemporal convolutions on these kind of ah data [noise]. And then, ah enter down into challenges and following that, ah I will give you a very brief introduction to spatiotemporal ah modelling for deep learning and [noise] what, where it actually comes down on the spatiotemporal concepts over there [vocalized-noise] [noise]? Ok.

(Refer Slide Time: 05:41)

NPTEL ONLINE CERTIFICATION COURSES  
Indian Institute of Technology Kharagpur | Department of Electrical Engineering

## Images and Videos

The diagram illustrates the dimensions of images and videos. On the left, a single image of a car is shown with width  $M$  and height  $N$ , labeled as  $3 \times M \times N$ . In the middle, a  $3 \times 3$  convolution kernel is shown with width  $w$  and height  $h$ , labeled as  $3 \times w \times h$ . On the right, a video is shown as a stack of frames with width  $M$ , height  $N$ , and time  $T$ , labeled as  $T \times 3 \times M \times N$ .

Understanding Video Analysis [Debdoot Sheel]

So, ah as far as ah Images and Videos are concerned [noise], so typically, ah if you look down on our ah earlier ah notions and this was an rgb image ah given down. So r, g and b were the three different channels [vocalized-noise] on the input [vocalized-noise]. Now, [vocalized-noise] ah with, whatever we have done till now you one thing which goes down clear to your mind is that, number of channels is ah [vocalized-noise] quite an independent aspect and it is just dependent on the operators [noise] and what happens

down with [vocalized-noise] number of channels is that, you can keep on changing [noise] the number of convolution kernels, [vocalized-noise] [noise] which would come down over there and that is something which is going to play around [noise] with the total number of channels.

And as a result, what would come down is, that [vocalized-noise] you all ah [vocalized-noise] these ah [vocalized-noise] it is it is not necessary that you just need r, g and b channels over here. So, you can have one channel data, you can have two input channels or say subsequently down the line [vocalized-noise] as you traverse across the depth of a convolutional neural network, you have even [vocalized-noise] more and more number of channels. So, there are 16, there are 6 channels, there are 32, 64 [vocalized-noise] and and this is a very generic concept.

So, [vocalized-noise] ah while in the earlier days, it was really ah [vocalized-noise] quite an effort for you to understand ah channels in terms of ah what happens when they are just increasing more than 3 over there, but [vocalized-noise] now we [vocalized-noise] have all of us are sort of ah used to this concept [vocalized-noise]. So, I am just going to refer down to color images with a simple example of ah three channel thing, because it makes it easier and [noise] intuitive to do, at the later on all of these are genetically ah [noise] scalable concepts. So, down the whole pipeline of our ah temporal video tensor you have everything going down in the same way [noise] [vocalized-noise].

So, now if this has ah M and N ah number of columns and rows over here. So, technically the size of it is represented as 3 cross M cross and N that is what an r, g, b [vocalized-noise]. So, if your number of channels is c, then you [vocalized-noise] in in a general case you would represent this as c cross M cross N [vocalized-noise]. Ok.

Now, if I have my convolution, what I would have is [noise] technically, 1 1 2 d matrix for each of these channels and and [noise] technically, [vocalized-noise] that makes up ah like you have [vocalized-noise] ah a 2 d matrix for each channel.

So, the number of [vocalized-noise] 2 d matrices over there, will be equal to the number of channels and then, what you do is you do a dot product for each of these 2 d [vocalized-noise] tensors and then, ah you take it is average over all of these tensors together, or or summation over all of these tensors together and that is a scalar value which you get down and this scalar value is what is representing [noise] one single x y

locations output for a particular given [vocalized-noise]. So, if you have multiple number of kernels over there, so that will be the total number of channels which you are going to create down [vocalized-noise] [noise]. Ok.

So, that is ah quite straightforward that the [noise] number of channels in your ah convolution [noise] volume over there is also going to [vocalized-noise] be the same as the number of channels in your ah ah [vocalized-noise] input data over there. Ok [vocalized-noise].

Now, if I look down at my video side over there. So, the difference what it comes down is, that it is going to be a collection of frames. Ok [vocalized-noise]. So over time, I have my camera and I am shooting down a video [noise]. So, what that means is that, I am looking forward over here, and and it keeps on shooting. And then, between ah [vocalized-noise] ah so, what I essentially get is, a series of 2 d frames and each arriving ah with a de fixed difference of time. So, if I have a 50 fps or 50 frames per second of a video acquisition system, then ah [noise] with the difference or 20 milliseconds, I will be getting down the subsequent frames coming down to me [vocalized-noise].

And what that means is that, ah this whole tensor is now going to have a size something called as  $T$  [vocalized-noise] cross 3 cross  $M$  cross  $N$ , by  $T$  is the total number of frames which I am considering [noise] along the time dimension over here [vocalized-noise]. And now, ah [vocalized-noise] given this whole understanding of  $T$  cross 3 cross  $M$  cross  $N$  over here, [vocalized-noise] what we do ah get to understand is that, in terms of my ah technical [vocalized-noise] way in which how this video is located, [noise] is that I have 3 cross  $M$  cross  $N$  this is a 3 d matrix over here [noise] and then, I ah on on my fourth axis over here, I am just going to pad it down.

So, for us the priority of the axis is that the major axis is something which comes at the first point over here [vocalized-noise] at the the first dimension for my data [noise]. So, this is where, I am going to use all of this and stack it down [noise]. So, this is [vocalized-noise] typically how it is represented in terms of a video [noise] tensor volume.

So, whenever you are trying to access a video, from a [vocalized-noise] video file, avi file or something and write it down to a math file or hdf five file, this is how it would be represent [vocalized-noise]. But, the problem is that, this is not how the [vocalized-noise]

tensor gets ah taken care of if I am trying to do a convolution over that. So, I will have to make some changes over there [noise] [vocalized-noise]. Ok.

(Refer Slide Time: 09:49)

The slide, titled "Data Organization and Video Tensor", illustrates the organization of video data into a 3D tensor. It shows three parallel 3D volumes representing the red, green, and blue color channels of a video. Each volume consists of a stack of frames. A vertical double-headed arrow labeled  $N$  indicates the height of the frames, and a horizontal double-headed arrow labeled  $T$  indicates the number of frames. Below the three volumes, a bracket spans all of them with the equation  $3 \times T \times M \times N$ . The slide header includes the NPTEL logo and text: "NPTEL ONLINE CERTIFICATION COURSES Indian Institute of Technology Kharagpur | Department of Electrical Engineering". A small video inset of a speaker is visible in the bottom right corner.

So now, for organizing this data and the video tensor, this is what we typically end up doing [noise]. So, I have my video which comes down to me, where the frame size is  $M$  cross  $N$ , and then I have ah [vocalized-noise]  $T$  number of such frames. So, so this is the, ah organization of it. Now, what I would technically do is, I would pull out each of these color frames separately [noise]. So, each channel is pulled out separately and then [noise] ah what I end up getting is, my channel is is my, is is going to be my ah priority axis over here. So, technically where [noise] I would be twiddling between these two dimensions. So,  $T$  comes [vocalized-noise] in place of 3 and 3 goes in place of  $T$  [vocalized-noise].

So, what that means is that, I have, ah  $T$  cross  $M$  cross  $N$  volume, for my red channel, I have a  $T$  cross  $M$  cross  $N$  volume for my green channel, and a  $T$  cross  $M$  cross  $N$  volume for my blue channel [vocalized-noise]. Such that; this is something, what it would be looking down? So, instead of having a 3 d matrix now ah at each time point, I am now going to have a 2 d matrix, such it each time point [noise]. So, first is going to be my red channel. Ok. So, this makes it as a [noise] 3 d matrix of size  $M$  cross  $N$  cross  $T$  or  $T$  cross  $M$  cross  $N$  anything what you [noise] want to do [vocalized-noise].



Now, I will have another volume of ah for my green channel and that will also be of the same size. I keep on constructing another volume, for my blue channel and that is also of the same size [vocalized-noise]. Now, that I have 3 of these volumes and if I am going to somehow ah [vocalized-noise] concatenate them along the major axis over here.

So, that would mean that, I need to create one more axis of the data. So now, instead of [noise]  $T \text{ cross } M \text{ cross } N$ , this red channel is going to [vocalized-noise] look like  $1 \text{ cross } T \text{ cross } M \text{ cross } N$ . [noise] This green will also be  $1 \text{ cross } T \text{ cross } M \text{ cross } N$ , and the blue will also be  $1 \text{ cross } T \text{ cross } M \text{ cross } N$ . And now, if I concatenate that along the first dimension [noise], so that is something which is going to look like  $3 \text{ cross } T \text{ cross } M \text{ cross } N$  [vocalized-noise].

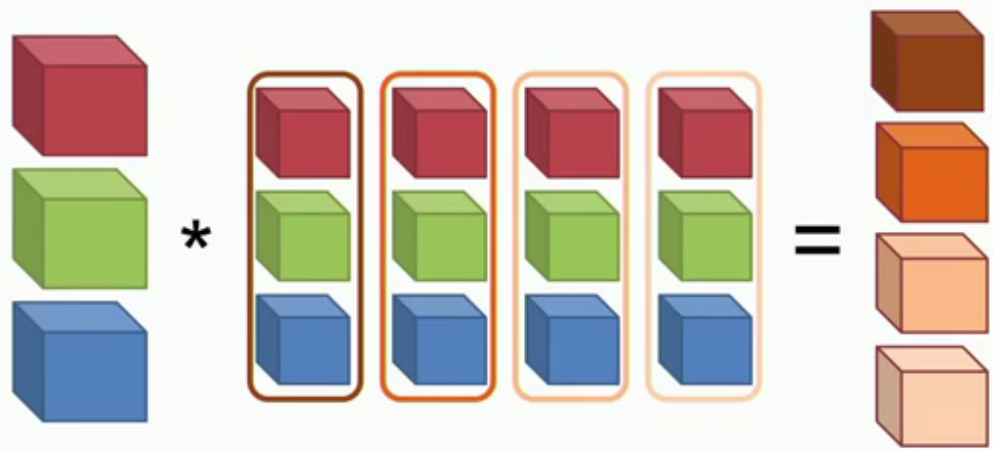
And now this again, ah boils down to the same ah kind of a philosophy as we had for our standard 2 d special convolutions [vocalized-noise]. So, in 2 d special convolution, what you had is the batch, ah number, the batch size or the [vocalized-noise] whatever is the index along the [vocalized-noise] batch of ah image frames or ah data coming down to you. Then, the next one was your number of input channels and then you had your ah first dimension and second dimension of the data.

Now, for us this data is now in 3 dimension. So, it is  $T \text{ cross } M \text{ cross } N$  [vocalized-noise]. So, that is what I have over here. This 3 is now my channels [noise]. Ok? And then, if I have another ah part of my batch handling over here, then this [vocalized-noise] preceding [vocalized-noise] part of the term is what would be representing my batches, and this is what comes down in terms of my ah 3 d volume, which I have ah for video.

So, this is a time space volume, and there are 3 channels of these time space volume. [noise] This is how you can imagine it pretty much [noise]. So, there is a red channel of my time and space volume, there is a green channel for my time and space volume, [noise] and there is a blue channel for my time and space volume [noise]. And then, this together creates out the whole gamut of my ah video, which is represented in terms of a tensor [vocalized-noise] [noise].

(Refer Slide Time: 12:46)

## Convolution Network for Video Handling



Now, let us get into, what will happen when I am trying to use a Convolutional Network for Handling of this kind of a Video? [noise] [vocalized-noise] So, I will be having these kind of volumes available to me. I have my red channels volume, I have my green channel volume, I have my blue channel volume [vocalized-noise] [noise].

Now, if I would like to convulse it ah over there, then I would need another ah set of kernel, ok [noise]. Now, as a result of this convolution, I am going to get down one block volume coming down. So, essentially what it means is that, this [vocalized-noise] small red volume cube, which is my kernel for convolution, [vocalized-noise] this does a convolution over, so it does a dot product, wherever it sits down on the first valid location over there [noise]. So, it sits [vocalized-noise] at this location it does a dot product, the green one sits over here does a dot product, the blue one sits over here does a dot product, [vocalized-noise] and then it sums up [vocalized-noise].

So, in case of your [noise] ah 2 d convolutions, what you are imagining as a 2 d ah [noise] [vocalized-noise] as a 2 d matrix, which sits on top of the image, one of these channels of the [noise] image and does a dot product. The second two d matrix for the next channel sits on top of the next ah channel over there, for video ah for your image and then does a dot product, and the third one sits over there and together you take it out.

So, now you have to imagine the same thing and [vocalized-noise] instead of for 2 d, [vocalized-noise] it is now a 3 d, which is fitting on top of it and this is generalizable. So, if I, if I have some ah 4 d matrix which is taken [vocalized-noise] over n number of channels, then I can still have the same thing, [vocalized-noise] that makes it hard to imagine on the first slot, but [vocalized-noise] typically that sets how it is going to be and then it is a generically ah [vocalized-noise] scalable of concept [noise] [vocalized-noise].

Now, my output is going to be as such, just one single channel and this will still be a time space volume based on, if I had a 2 d space over there and if I am convolving, my resultant is a 2 d thing. Ok [vocalized-noise]. Now, if I have a 3 d thing and I am convolving in 3 d, then my resultant is also supposed to be a 3 d. So, my convolution is 3 d convolution that is that is not a 2 d convolution as we had in the earlier case. So, here we are just going to treat all of these convolution sets typical 3 d convolutions [noise] [vocalized-noise].

Now from there, if I have another kernel, over here [vocalized-noise] [noise] then that is going to lead to another channel formation [noise]. Similarly, if I have another kernel, I get down another channel which is form, [noise] and another kernel which will get me ah another channel formed down [vocalized-noise].


So, this also goes in line with our original understanding and philosophy which is, [vocalized-noise] that the total number of channels on my output of my convolution is going to be dependent on the total number of convolution kernels I have in a particular given convolutional layer over there. So here, since I have four different convolution kernel, so it is going to lead down to four different [noise] ah channels coming down over here and that is pretty straightforward. Now, that there is not much to get confused in any way [vocalized-noise]. Ok.

Now, well ah comes in the interesting aspect is, [vocalized-noise] that you can technically see that I have increased one extra dimension [noise] of the data and associated with this one extra dimension of the data comes ah in its own computational complications as well. Ok.

So, let us look into one of these examples. Let us take one of these kernels over here, and try to convolve it with this whole time space video volume over here and then come

down as what comes on my output? And, then see [vocalized-noise] what is the net effect of this output as it comes down ah through this kind of a convolutional concept. And and, let us let us look into the width, height, ah depth and these calculations coming down. So, what is the resultant aspect of the convolution [vocalized-noise] [noise].



(Refer Slide Time: 15:57)



## Convolution on Video Tensors

\*=

$$o_w = \frac{M - w + 2p_w}{S_w} + 1$$


Understanding Video Analysis [Debdoot Sheel]


Now, as goes down over here. So, I have my, ah time space volume, and then the three channels of my time space volume are available [noise]. I take down my first ah convolution kernel, which is also a three channelled [noise] convolution ah kernel over here, and then [noise] I get my resultant in terms of a time space volume for one single channel [noise]. Ok.

Now, ah taking down the same kind of ah [vocalized-noise] analysis which we had done earlier, so in your earlier analysis what you had was, that your output over here is going to be  $M - w + 2P_w$  divided by  $S_w$ . Where [vocalized-noise] your  $O_w$  is basically width of the output. So, you consider any one of these axis over here, as your ah current width. So, I am I am taking this as my current width. I can also be putting down the same equation to do it along this, along this any any one of them. So, that is that is just which particular [noise] axis you are looking at you need to understand the padding and the stride along that particular axis. Ok [vocalized-noise].

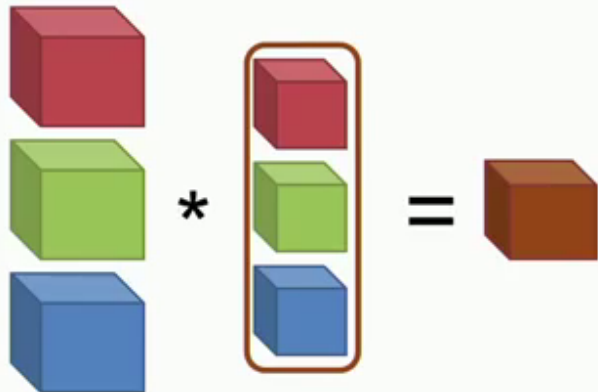
Now,  $w$  is basically width of the kernel along that axis. So, if I am looking at [noise] the  $x$  axis over here, then I would be looking at what is my ah [vocalized-noise] width of the kernel along the  $x$  axis. Then, I will have my padding over there. So, I can pad down along each of these axis in a different way. So since, it is a 3 d tensor over here [vocalized-noise]. So, in in case of your 2 d convolutions, you could see that you had to give down [noise] only 2 different padding [vocalized-noise] criteria over there, you could like explicitly mention down [noise] or if you are just mentioning only one padding criteria, then that is what is reflected across all the dimensions over there [noise].

Otherwise, if you are giving it within a tensor tuple, then ah the first one is for the first dimension, the second is for the second dimension [noise]. That is that is how could you could do. So, here your if you are giving it down in terms of a tensor tuple, then you have to have a 3 cross 1 ah size tensor, and not ah a 2 cross 1 size tensor as you had in the spatial convolution side [vocalized-noise]. So, that is the next difference which comes down over here [vocalized-noise]. Ok.

Now with that, you can also specify your stride and as we had seen, that you can have stride along each directions in a different way or you can put down isotropic stripes. And typically, if I am [vocalized-noise] not mentioning stride in any way or padding in any way. So, if there is no padding mentioned then it is a 0 [vocalized-noise], there is no 0 padding added down. So, there are no extra pads coming down [vocalized-noise]. If my  $S$   $w$  is also explicitly not mentioned, then it is just something which happens with the stride of 1. And [noise] this is what my output comes out [noise]. Ok.

(Refer Slide Time: 18:07)

## Challenges



$$o_w = \frac{M - w + 2p_w}{S_w} + 1$$

$$M = 224, w = 3, p_w = 0, S_w = 1$$

$$o_w = 222$$

$$\text{Input} = 3 \times 100 \times 224 \times 224$$

$$\text{Channels} = 16$$

$$\text{Output} = 16 \times 98 \times 222 \times 222$$

Now, where the challenge comes is quite straightforward. So, if I am trying to do this convolution [noise] over here, with one of these ah blocks and then, I get down my output. Now, as I get this output coming down over here, what would happen is that, this is my relationship which I had learnt on [vocalized-noise]. Ok.

Now, let us [vocalized-noise] take a very typical case and this typical case is where, my M is 224. So, ah my image net sized frames which I was taking till now, they were all 224 cross 224 ah [vocalized-noise] size freedom. So, they have a three number of channels, so 3 cross 224 cross 224. Ok.

Now, if I am taking [noise] say a video, which is shot on the same resolution as an image net over there. Ok. So, in that case; I can have some ah fixed number of frames over there. But, let us look into one of these accesses of now [vocalized-noise]. Ok.

So, [vocalized-noise] if M w, M is something like 22 [vocalized-noise] ah 4 ah cross 224 and this time, has some some other dimension. I will come down to that shortly [noise] [vocalized-noise]. And say, this w over here, this with is something of the size of 3. My P w is 0 and S w is 1. In that case; my O w actually comes down to be 22 ah 222, [vocalized-noise] that is pretty simple. So, I have 224 minus 3 plus 0 divided by 1. So, that is 220 ah ah 221 plus 1 which makes this [noise] 222.

So, that is my output width over here. So, my output width is directly related. Now, that is that straightforward. However, I have my time dimension. So, now, I need to get down into, what will happen on the [noise] three d case [vocalized-noise]?

Now, say my input over here is 3 cross 100 cross 224 cross 224, which technically means that I have 100 frames [noise] available over there, each of size 224 cross 224. And now, my total number of convolution channels over here, over here I was doing it just with one channel, but my total number of convolution channels, ah in in my convolution layer is 16. So, that is the total number of kernels I have.

So, now my output is going to be something like 16 cross [noise] 98. Now, where this 98 comes down is that, [vocalized-noise] my w over here is 3, and this is what I am going to consider isotropic across all of them, if I am not explicitly specified [vocalized-noise]. So, for this axis over here, ah where my time axis has 100 ah [vocalized-noise] is the size of the tensor dimension along the time axis is 100. In that case; this output along that one is going to ah be calculated something like 100 minus 3, because [noise] I am taking down 3 cross 3 cross 3 size kernels. So, its 100 minus 3 [vocalized-noise] ah plus 0 divided by 1. So, that makes it 97 plus 1 is 98. So, that is where I get down this one. And then, my 222 cross 222 is what is the straightforward calculation which I see over here [vocalized-noise].

So, this is where you have both the spatial aspect, as well as a temporal aspect being integrated together and it comes down into one single ah output. Ok [noise] [vocalized-noise]. So, if you look into this particular problem, what you can see is that, with increase in this extra dimension over here, there is a significant increase in the total data overhead. Now, if you [noise] write down ah as in the earlier cases we had been doing our pen and paper empirical solutions into, what is the total parameter space? [vocalized-noise] Then we found out, what is the model space in terms of bytes? What is the operational space requirement for the model? And, what is the operational compute requirement for the module [vocalized-noise]?

Now, if we go down with the same logic over here and place down, because of this extra increment of ah [vocalized-noise] dimension and this coming down to 98, now that is a huge burst up. So, it is almost 100 times more memory which it is going to take. And [noise] including these channel over here, this kernels over here are also going to get on

one extra dimension of parameters which it has to learn [noise]. So, it is not just that the parameter space is exploding, it is also that you have an explosion around the operational space complexity.

Now, this is an alarming factor. Because of these issues, it becomes really hard to actually operate on a [vocalized-noise] substantially sized video. So, if my [vocalized-noise] camera is acquiring something at 50 fps [vocalized-noise]. So, technically that means, in just in two seconds of time, I will have 100 frames acquired [noise] over there. Now, in two seconds of time, most likely my actions will not be changing significantly over there in any way [noise].

And, if that is the situation, then I am not getting to get down a substantial amount of description about the [vocalized-noise] whole action in a video [noise]. Maybe I can just comment on two second, what happened in these [vocalized-noise] two seconds, but not [vocalized-noise] what happened in the overall activity present over there.

So, these are challenges which we would be facing [vocalized-noise] while trying to do these analytics.

(Refer Slide Time: 22:28)

NPTEL ONLINE CERTIFICATION COURSES  
Indian Institute of Technology Kharagpur | Department of Electrical Engineering

## Spatio-Temporal Deep Learning

$T \times 3 \times M \times N$

$k$

$T$

Understanding Video Analysis [Debdoot Sheet]



And now, in order to get rid of this challenge, there is actually a very simple way, which is called a Spatio-Temporal Deep Learning and this is [noise] ah rather effectively used way which, a lot of people on the community work out on. Now, what it does is, ah say I have my frames available over here on my video, then I am going to extract out all the frames over there [vocalized-noise]. And now, instead of considering ah like reorienting my dimensions on my ah video over here, [noise] what I choose to do is? I choose to extract each frame and trying to operate on each of them [vocalized-noise].

Now, if I have this whole thing of ah  $T$  cross  $3$  cross  $M$  cross  $N$  [noise]. So, what that would mean is? I can pull out any one of this frame from here, which is going to be of the size of  $3$  cross  $M$  cross  $N$ . Ok.

So, let us pull out one of this frame. Now, what I can do is? I have these image net sized ah networks available to me, for for doing ah classification [noise] and if I, as I had done in the earlier experiments, in in most of our experiments what we had done is, we took down a model which was more of salt for the image net problem, and we were trying to solve something else [vocalized-noise] [noise] ah for a similar scale of data and to keep everything tractable we were trying to solve it with the  $c$  fact  $n$  problem over there [noise].

And, for that purpose what we had done is, we pulled out say vgg net or googlenet or res net, dens net [noise] any of these networks, and the image was taken down from sefa and it was scaled up. So, while I was loading the on a sefa image instead of for  $32$  cross  $32$  [vocalized-noise]. So, I scaled it up to  $224$  cross  $224$  [noise] sized and then I placed it into my network [noise] over here and then, I get down ah my my total output cover. So, I was truncating my total [noise] number of classification layers over there, instead of making it ah  $1000$ , which is typically for image net I chop it off and make it just  $10$  decision layers [vocalized-noise].

Now, if I am not adding this decision layer in any way, what I get down over there is, just my feature vectors and that is what I am going to consider over here as of now within spatiotemporal learning [vocalized-noise]. So, the idea is that you pass it down through a network, and then if you keep down sending each individual frame in the same way, [vocalized-noise] then you are going to get down an output [noise] tensor over here which has some  $k$  number of neurons. So, it is a  $k$  cross  $1$ , that is that is just before the

final ah [noise] [vocalized-noise] layer for ah doing the fully connected network for classification, you have a linearized out and then there may be some some extra connections given down. So, it depends on ah you can have ah [noise] 4096 neurons over there, you can have 512 neurons, you have 1024 neurons any of these number of neurons [vocalized-noise].

But then each [noise] tensor is what is corresponding to one frame. So technically, this whole thing becomes a  $T$  cross  $k$  size tensor, where  $T$  is this dimension of the [vocalized-noise] time length which goes down. Now, if I come down from this video space over here which has  $T$  cross 3 cross  $M$  cross  $N$  [noise]. Now, this whole thing is mapped down into a  $T$  cross  $k$  size [noise] tensor, where 3 cross  $M$  cross  $N$  dimension is what is mapped onto these  $k$  dimensions over here. So, that is linearized and mapped and to ah much reduced out space, whereas your time ah [noise] dimension still remains the same [vocalized-noise].

And this is a concept, which we are going to now [noise] ah make use of. So, now that I have this  $k$  number of neurons, each representing one timestamp of a tensor, now I can use this as some sort of a time stamped feature, [noise] which consolidates out my whole visual aspects on a frame and then, use it for passing through some sort of a sequence learning model [noise].

So, this is basically a sequence of features which I have [vocalized-noise]. So, in the next lecture, what I am going to cover is something which is called as a recurrent neural network. So, I will introduce you the whole theory of a recurrent neural network, [noise] and then enter into something called as a long short term memory, which is one of the most ah [vocalized-noise] viable meth ways of ah doing and solving out these recurrent neural networks. The input to this which I would be calling as, features or aspects or attributes over there [noise] and what are these  $T$  cross  $k$  tensors and each timestamp is [noise] one timestamp tensor which enters over there.

So, ah instead of waiting keeping you [vocalized-noise] waiting for long. So, we will be doing those things [noise] tomorrow. So, once we finish off in the next lecture, the series and understanding and the theory behind, what happens in a recurrent network for sequence modeling? In the subsequent one, we are going to show you a clear cut

example of, how this problem has been solved out? So, still ah till then ah stay tuned and ah.

Thanks [noise].