

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

Lecture – 58
Spatio-Temporal Deep Learning for Video Analysis

Welcome. So, today in this particular lecture, what we will be doing is to understand, what we do exactly for one of the cases of Spatio Temporal Deep Learning?

(Refer Slide Time: 00:24)



The slide features the NPTEL logo and text at the top: "NPTEL ONLINE CERTIFICATION COURSES" and "Indian Institute of Technology Kharagpur | Department of Electrical Engineering". The main title is "Spatio Temporal Deep Learning". Below the title, the presenter's name "Dr. Debdoot Sheet" is listed, followed by his titles: "Assistant Professor, Department of Electrical Engineering" and "Principal Investigator, Kharagpur Learning, Imaging and Visualization Group, Indian Institute of Technology Kharagpur". A QR code is located in the bottom right corner, and a website URL "www.facweb.iitkgp.ernet.in/~debdoot/" is provided at the bottom center.

Now, in the earlier lecture, what I had you guys introduced on to was to understand videos as a tensor and the difference of how do you convert down all of these video frame, when you have colored video acquisitions, and what is that tensor dimensionality and priorities in terms of the arrangement for handling it out.

Now, one thing which was clear is that the meaning and interpretation of channels is quite different. When we are writing out these tensors and in fact; what you would effectively have is, if you have a spatial dimension and the temporal dimension, then for each of the color channels you have that packed down in terms of its own 3 D tensor and the whole resultant is a 4 D tensor which comes out.

And in the subsequent lecture we had studied about another newer kind of a neural network called as recurrent neural network which typically, deals on the aspect of

recurrence or if I have some sort of a dependency of my current state on the previous state.

As well as the input of the current state then, these kinds of networks are what are called as recurrent neural networks. And typically, what we studied was modern variant of these RNN called as the LSTM or Long Short-Term Memory. The advantage was that you could get down short term or very recent past kind of behaviors modeled out quite easily. And the other one was that you could also retain down long order relationships in terms of your temporal dimension.

Now, this did have a very significant advantage. So, one of these was, that initially they had come up for natural language processing, and for grammars and auto translate auto completion kind of works. But, subsequently when people did realize it is great advantage, we started using these for video analytics as well.

So, today's example is just to give you again a basic very brief revision of what we had done, so that you have all the context set down and then, I would enter into one of these case studies. So, we have two specific case studies and both of them are spatio temporal deep learning.

So, in one of these examples is where, we are going to use a Recurrent Neural Network, and we see how the performance comes out. And, the other one is, if in case I am not trying to use a Recurrent Neural Network then, how will I be modeling of the whole network over there? So, I would be showing you both of these examples and then that is how we keep on proceeding.

(Refer Slide Time: 02:40)



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

Organization

- Revision of Basic Concepts
- Spatio-Temporal Deep Learning
- Case Study – Surgical Tool Classification
- Spatio-Temporal Deep Learning without RNNs

Spatio Temporal Deep Learning [Deebot Sheet] 2

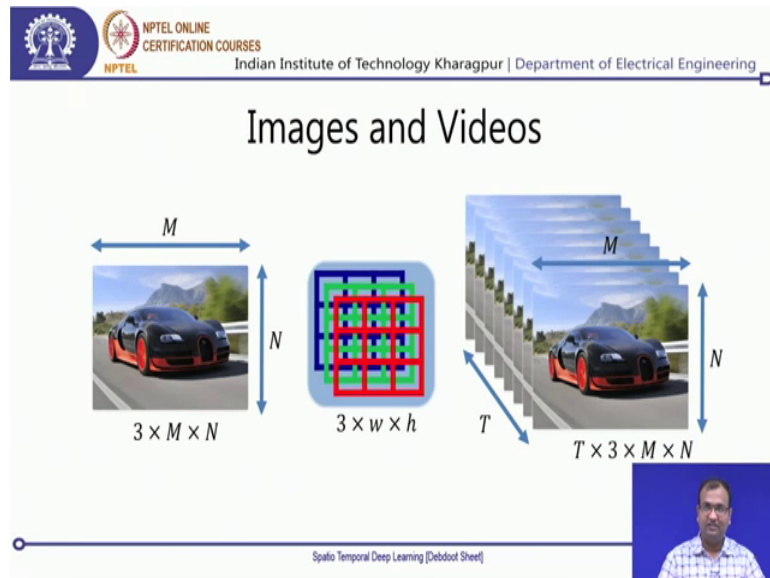
So, on the Organization side of it there is a Basic Revision of the Concepts one second, and then I would enter into the Spatio-Temporal Deep Learning. So, in the last lecture, where I was introducing you to this whole concept we did a very fast walkthrough and then, I said down that there is another aspect of recurrent model which, we would be doing in this subsequent one.

So, now, that you know about, how Recurrent Neural Networks work as well as; how the whole spatial data can now be processed and brought down into a lower dimensional tensor space on the temporal side of it we are going to review that, and come down to an actual model where we made use of it.

Now, there is a case study which we will do for Surgical Tool Classification. So, this is a very specific one from medical use cases and one of the publications which we had in CVPR 2017. So, we will open up the paper as well and go through some of these stages inside over there to explain you, what is a practical design problem which we had faced? And, how we thought of overcoming? And then, what were the solutions which we achieved out of it?

And then we would enter into spatial temporal deep learning without RNN as one of the case studies for a student project, which we had done where without using these LSTM's can we still end up doing. So, can there be a fully convolutional network equivalent for analyzing out videos as well.

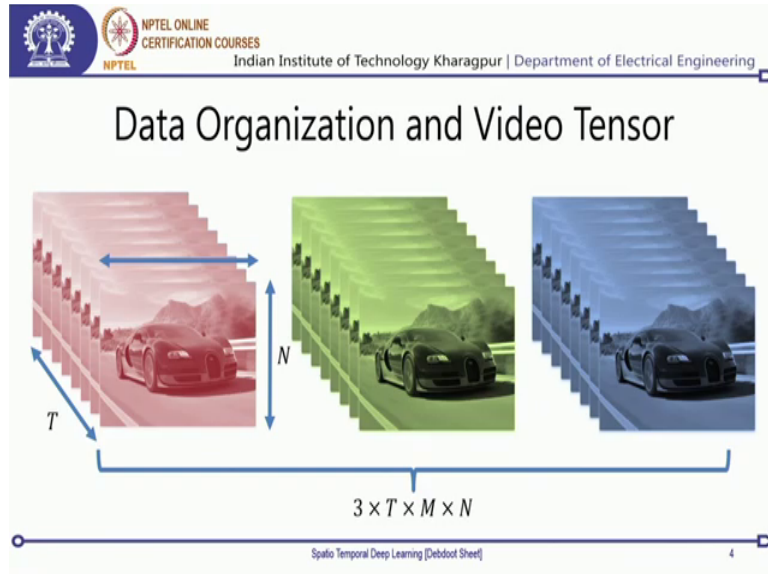
(Refer Slide Time: 03:58)



Now, a very basic introduction just a briefing of it; So, if you had your images as such of M cross N , and it was a three channel you had a 3 cross M cross N . And then, your convolution kernel was also a 3 D kernel, which had number of channels equal to the number of channels in the input image over there and it was 3 cross w cross h .

And then, when you convolve it out you have your resultant. But then in case of a video, what happens is that your frame dimension is still M cross N , but then you also have another component of time over there and each frame is a 3 cross M cross N . So, your resultant is T cross 3 cross M cross N .

(Refer Slide Time: 04:35)

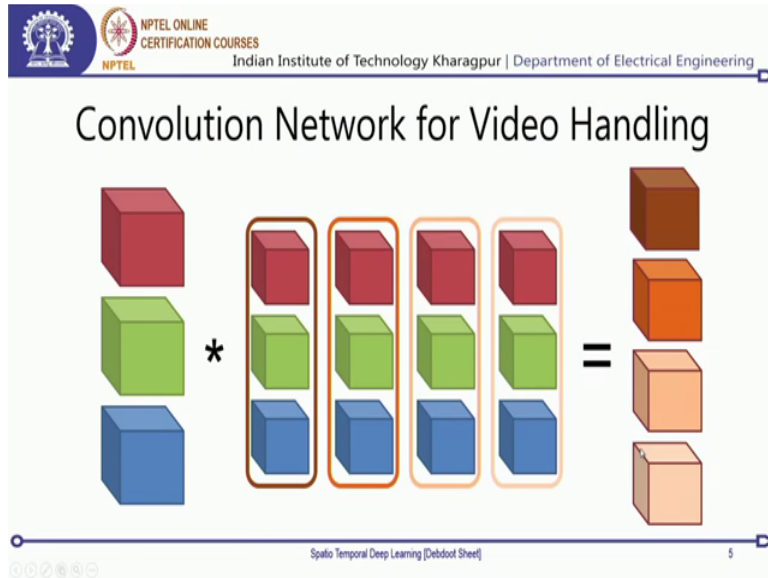


Now, when we get down into the video organization in terms of a tensor, what we need to do is that you have your overall organization given down as T cross 3 cross M cross N ; where 3 is the number of channels. Now, this number of channels can keep on varying. So, they not necessarily need to be 3 . Since we are dealing with RGB color images, so we still have this option of having just three channels over there, but that that can be different based on, whatever is the output from your previous stage.

Now, the only issue is it is not easy to handle the it down in that way because; number of channels is a unique quantity over there. Now, what we can do in that case is, we can break it down into a red special spatiotemporal tensor, which is T cross M cross N for the red channel, we break it down for the green channel and similarly, we break it down for the blue channel as well.

So, once we have this kind of representation coming down over there, then each channel itself is T cross M cross N . And there since there are three channels of T cross M cross N , so this whole tensor is now a 4 D tensor of 3 cross T cross M cross N , ok.

(Refer Slide Time: 05:42)



And if we enter into a convolutional network for this video handling then, what we would end up having is that you have a 3 D volume of T cross M cross N for the red channel green channel blue channel. Next we are going to convert it down with a convolution kernel, which is also 3 D in size.

Now, you have this 4 D collection. So, this is just a bucket on the 4 D space over there each of these kernels is a 3 D kernel. So, you have one kernel for the red channel, one for the green channel, one for the blue channel and then this together does a dot product. And for every point in this volume you are going to get down one output point over here. Ok. And, so this convolution is going to be on by giving your strides along the x, along the y, as well as along the T dimension over there. So, the resultant is also a 3 D volume, it is not a 2 D pianaar matrix in any way.

Now, if I have multiple number of these kernels, then they are going to define 1; 1 output volume for me, and that corresponds to 1; 1 output channel. So, your input was a three channel input, you had 4 such convolution kernels over here. So, you output has a four channel output over there, and everything is a 3 D volume which comes out. So, this was clear for us.

(Refer Slide Time: 06:57)

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

Convolution on Video Tensors

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

Spatio Temporal Deep Learning [Deebot Sheet]

The challenge however, was that when you are trying to do these kind of convolution using video tensor then, your total output size has this kind of a relationship which comes down.

(Refer Slide Time: 07:08)

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

Challenges

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

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

$$o_w = 222$$

Input = $3 \times 100 \times 224 \times 224$
Channels = 16
Output = $16 \times 98 \times 222 \times 222$

Spatio Temporal Deep Learning [Deebot Sheet]

And, the challenge which it actually poses is the data issue over there. There is the moment you have this massive amount of convolution coming down and the resultant. So, the size of this resultant is something, which is guided down by M minus w plus 2 p divided by S plus 1. Now, this padding along temporal will also be guided down over

there. So, we are taking only along one of this axis, so if you take along all the three axis then you get down a similar result and coming down.

So, as a typical case, what we had done in the last class was if we take down that M equal to N equal to 224, and w is equal to 3, and P_w is equal to 0, and S_w is equal to 1. In that case, we get down our O_w is equal to 222. Ok.

Now, if we take an input of this kind, which has three channels and on the time stamping, on the time axis over there, there are 100 time axis over there and the spatial size is 224 cross 224.

Now, if my convolution is there with convol 16 such channels coming down over there. Each of size 3 cross 3 cross 3, then my output comes down as 16 plus 98 cross 22 cross 222. Now, if you look down at this particular aspect, now this output volume is almost five times more bulky, than the input volume.

Though, we had just lost down by a reduction of 2 pixels on the time axis, 2 on the width and 2 on the height. Eventually, there was an increase in the total number of channels on the output, and that is about five times more. And this is the challenge which we were facing.

(Refer Slide Time: 08:44)

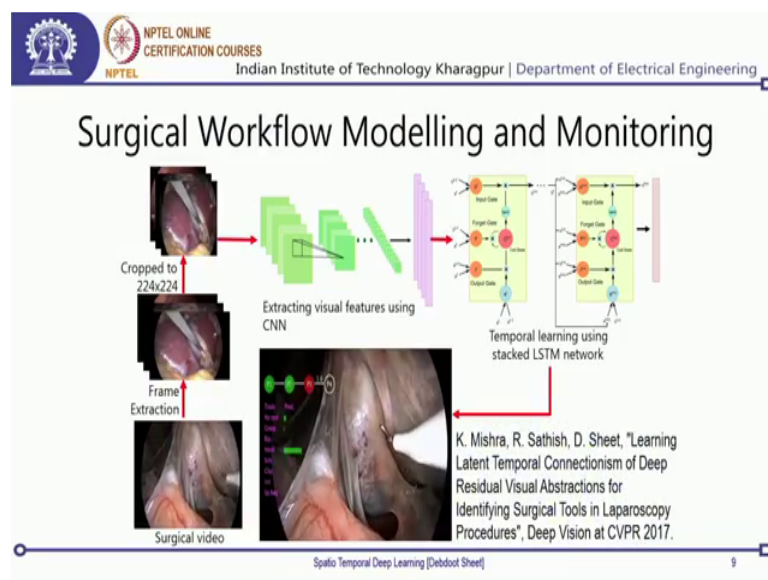
The slide is titled "Spatio-Temporal Deep Learning" and is part of NPTEL ONLINE CERTIFICATION COURSES from the Indian Institute of Technology Kharagpur, Department of Electrical Engineering. It features a diagram illustrating the process. On the left, a 3D volume represents the input, with dimensions labeled T (time), 3 (channels), M (width), and N (height), and the overall size is given as $T \times 3 \times M \times N$. This input is processed by a neural network, depicted as a series of interconnected nodes in blue, orange, and red. The output is shown as a 3D volume of size $T \times k$, where T is the time axis and k is the spatial dimension. A small inset video of a presenter is visible in the bottom right corner of the slide.

And what we said at that point of time is, let us move over to a other concept which is called a spatiotemporal deep learning such that; I can now consider my frames

individually and then reduce it down to some sort of a feature space, 1 D tensor feature space over there such that frame at a different point of time is what can consolidatedly be presented in terms of just a 1 D tensor. And that can in some way help me in solving this problem.

So, now, what happens in that case is that, you have your video which has T cross 3 cross M cross N frames over there. Now, what I do is, I take only one frame at a time from one particular instance of T. Now, I take one of these frames and I pass it down through one of my neural networks over there, and then, I get down my output tensor which comes over there. Now, this output tensor is of size k cross T which comes out.

(Refer Slide Time: 09:37)



So essentially now, if I use another kind of a concept of recurrent neural network standing at that point then, I should be able to use those features as input to it. Now, this is what we are going to exactly do in this example over here, from one of the case studies.

So, this is from our publication at division workshop in CVPR 2017, and the objective over here, what you see in this video is basically so, it is captioned out as a surgical video. So, this is an endoscopic surgery.

So, in it is a form of minimally invasive surgery, where you just drill a small keyhole into your body and then through this keyhole there are small surgical tools which are very thin. So, they are almost like the size of a refill of one of your ballpoint pens.

So, that is what enters, and there can be multiple tools. So, some of these are so, you see some sort of pinching tool over here or a grasper. You see some of these which are just ablating out over here. So, they are hooked like tools over there.

So, let us play that video again. And, there is a surgical operation which is going down. So, these are typically carried down within the gastric cavity or for operating down on your liver tissue, and this is being carried down in the gastrointestinal cavity as seen over here.

Now, the whole objective behind doing this was that, we have these different kind of tools. That can be multiple tools two or three tools at a time which come down, so can be identify those tools over there in some mechanism. So, that was the whole objective.

Now, one way is that you can typically say that, I have one frame and then I can process this frame and get down a classifier ready as a multi hot classification. So, it is not necessarily that you will have only one class coming up over there as the result, but you can have multiple classes also coming up.

Now, what we realized eventually was that, there within certain frames they can be discontinuities and you can; obviously, increase down on the total performance if you are trying to have some mechanism of incorporating and learning the spatial continuity between features which come down in a frame.

Now, being able to learn down this spatial continuity between features which come down in the frame would enable you to actually have a good grasp over, what is the possible tool which is present over there? Even if it is vanishing out for one frame or two frame or there are miss interpretations on if you are trying to classify it on a frame basis. Then still this long term context on the temporal side of it is what can solve it out. And that is what we solved out over here using a temporal problem.

Plus on top of it there is another extension of this work which we had done and that was more of to classify different phases of a surgery and that is the kind of action which is

taking place in this video. So, in the lab sessions what we do is, we will be taking down another standard video data set and where we will have to classify down the different action. So, it is going to be an exercise which all of us are going to collaboratively solve over there, and then find out which of the mechanism works out pretty good.

Now, here the whole idea based on what we had seen in the earlier slide; so, what we do is, we extract out the frames over there. So, now each of your frame is a 3 cross M cross N. Ok.

Now, for our work over here we needed to have a so, whatever idea was to get down a pre trained network, a very standard pre trained network which works out on image net scale resolution. So, that also meant that these frames will now have to be resized to fit down the 224 cross 224 image size for any of the world fit image net. So, that is the next step which is done.

Now, you can see over here that on these frames over. So, some of these boundaries are quite blanked out. And the reason is quite simple you have a camera which is placed on top of a circular aperture of a lens and this lens is a optically guided lens within the body cavity.

Now, since the camera sits on top of it and then this is done in an event so that you do not miss out on any of the information which comes out from the circular aperture. So, the camera is placed out in a way and such that this image is a small region within it. So, that is what you exactly see within this slide.

Now, we can actually chop off these extra perimeter parts over there, this peripheral black part. So, they are just truncated off. And then you near about have a square frame which there might be some sort of a squeezing needed at point of time. But, nonetheless that that does not introduce much of distortions as we had seen while, processing it out.

Now, that you have your frames which are resized and cropped down to come down to your image net scale resolution. Now, you can take down any kind of a standard model over them. So, you can try out with the VGG net, you can take an Alex net, you can take a Residual Network, you can take a Inception net Inception v 3 or the Googlenet, you can take Densely Connected Residual network, any of them are going to work down.

Now, once you pass it through it, then, you are going to get down this final layer of features; Now, in the domain adaptation lectures and the domain adaptation practical's which we were doing. So, what we had done is you had seen that we do a forward pass over there, and then the features are collected and somewhere at the last layer which is just the classification layer you truncate out instead of 1000 layers we were putting down our ten class classification. So, you just had 10 neurons instead of 1000 neurons on the final one.

Now, here the idea is quite similar. So, you can truncate off the last classification layer which had those 1000 neurons over there, but then you do not connect it to any further smaller number of neurons you just retain that over there. So, in case of your VGG net or something, you are going to get down 4096 neurons over there. Now, if you come down to say residual network, you get down just 512 neurons over there.

Now, this is your tensor, which represents the spatial characteristic at one point of time. However, you are going to have these kind of tensors at different point of time. So, corresponding to each frame you have one of these 1 D tensor which comes out over there, and then you stack this whole thing. So, this is stacked along the time axis, this is not stacked along the x or the y axis over here it is stacked along the time axis.

So, on the x axis you have a dimension of 1, on the y axis you have the dimension equal to say 4096 and 512, then along the third axis of the time axis you have the time dimension going down. Now, once you have these sequences available over here, so they can be fed down to your LSTM as features. Ok.

Now; obviously, at the start point over there, we do not have the option of a previous state coming down over here. So, anytime which we do, what we do is we typically take a bunch of frames over there, and then the last frame is whose state we try to predict out. Now, the previous frames are what just go as information over there.

Nonetheless, we are training for this one, what we do is? Since you have multiple modes of training a LSTM over there; So, you can have a many to one, you can have a one to many, you can have a many to one, many to many these kind of notifications. So, we trained it down as a many to many. So, given that you have n number of frames which are given as the input in terms of features, you will be trying to classify all the n frames

over there as a sequence of outputs. So, the class labels on all the frames are what I leveled out and then chunked out from the LSTM.

Having said that; what we more importantly put down is, now that you have this time stamping for each of so, classification label for each of the time stamps available to you, we do not consider all of them. So, we formulate this as a moving window kind of an approach or something which is time invariant approach in which provided certain frames have come down we are going to predict the only, what is my current frames classification label for that ?

So, if I am currently at the 100th frame, and I have a temporal length of say 5 then I am going to use my frame number 100, frame number 99, 98, 97, 96. So, all of these 5 frames over there, and now, my LSTM is going to give me an output of classification for each of the 5 frames. But then, I do not use the frame classification from frame number 96 to frame number 99.

So, instead I will just be using the classification for frame number 1000 and just given result. So, at any point of time what this method would necessarily need you to do is, even when you are training on testing over there you chunk in a block of frames, but then this block of frame has to be contiguous, and whichever is the recent most frame over there. So, the previous frames classifications are not taken only the current frames classification is what is taken down.

So, that is going to give you one one single scalar output for the current time frame based on whatever has happened in the previous few frames over them.

Now, once you do that this is what we were predicting out. So, there were these multiple tool classes present over there. So, on top there is something which is for the video activity classification, which is also called as phase classification problems for these kind of videos. Now, here the whole idea is the, you see this probability bar graphs coming down over there.

Now, whenever there is a new tool which comes in so, there is a grasper which comes in and then goes out. So, whenever there was this grasper coming down, you could see that one and then when it is out over there, you see there is no tool present. So, that is that is

what it is showing them. And all of these predictions are just for that current frame there is not a whole sequence of frames on which it is trying to predict it out.

Now, on the other side on this top, you have these phases of surgery. Now, this is the current phase of surgery which it is showing. Now, it also shows a possibility of the next phase and which may come up and that is an extension of this one what we do is since there is temporal context modeling down over there, there is a different field of sequence modellings called as Hidden Mark of Models or Markovian Field Processes.

So, what they do is based on, how many things have occurred in the past? It can try to predict, what may be a possible outcome in the future? So, that is what is going on over here as such.

Now, nonetheless we are just concerned about this first part, which is to find out, which is the kind of a tool which is present over there or just the activity, current activity stage not to look into the subsequent one. So, that is that is not part of the syllabus over here.

So, in the classroom exercises what you will be doing is with one of them, we are going to have your example tutorial ready in which in the next class, where you are going to get down a set of videos over there, and you would just be classifying, what is the activity shown down in that video over there ? Ok.

Now, this is one part where you have your spatial CNN working down, and then you have your features extracted and based on the features you are going to feed it down to an LSTM and then classify it out.

Now, there is also another kind of a problem.

(Refer Slide Time: 20:15)



So, this is what we had done with goal directed human videos and here, the idea was that you can extract out these frames over there and do the same kind of modification as in cropping and then, resizing it to 224 cross 224 and then, you feed all of them onto CNN for extracting out features. Now, once your features are extracted out the whole idea was that, can we have these at multiple time frames? And, then draw a fully connected neural network and then do a classification.

So, here it was just four different classes which needed to be classified and what we had was, we start with one of these frames at whatever level. Say this is at the T th frame. Then, I go back and take all the previous four frames, so; T minus 1, T minus 2, T minus 3, T minus 4.

So, these are the temporal context window of five frames available to me. Now, stacking down all the neurons together and in a linearized fashion I do a fully connection to the next layer which has 4096 neurons. Now, this is dependent on you. You can pretty much choose down and work out which would be a good combination of having the total number of neurons over there. And then from there, I connect it down for my, classification.

Now, what you see is that, only common denominator is the spatial features which you can extract out from these CNN's. But, then the rest of the temporal modeling is now a fully connected modelling over there. So, this particular kind of a model has a certain kind of a disadvantage. Because, even long order neurons and frames of activities in

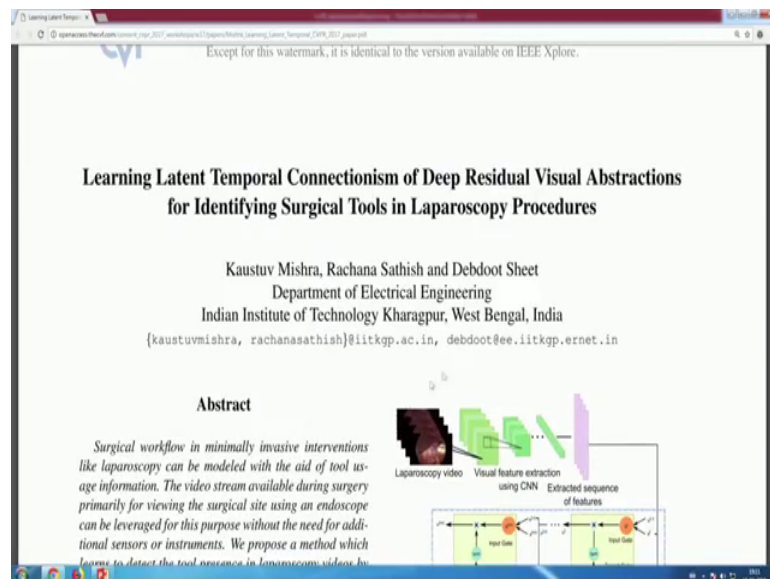
which have occurred much earlier, they are also given a similar kind of a preference. And everything is given down a similar preference and coming up over there; though network has to actually learn down over there.

The other downside is that the total number of weights which this whole network has to learn is much higher, then what would happen down within an LSTM. So, the network is definitely much more combustion. However, the good side is that you can now use just a fully connected network and try to do it out.

Now, what we have is on the green, you have the ground truth which is the actual class of activity which is going on which it is predicted. And, on the radio you have the actual predicted class which is coming out of this particular example of Spatio Temporal Deep Learning. Ok.

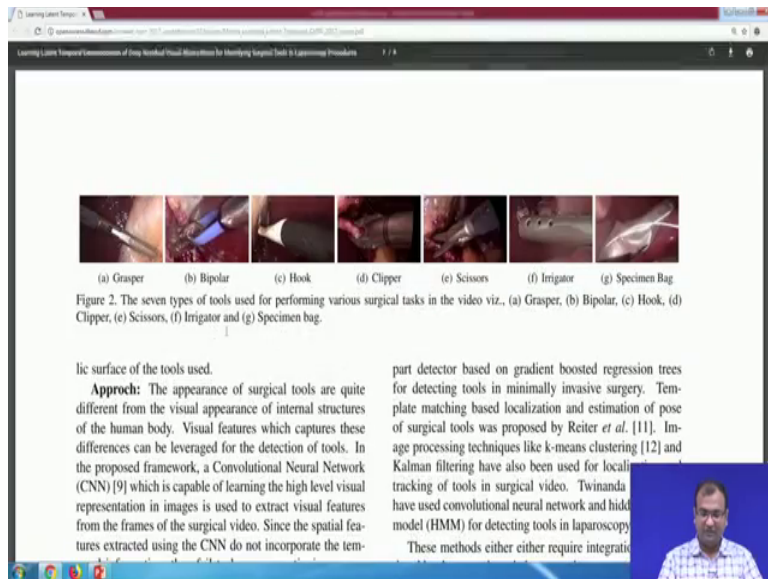
Now, these are two examples over there, of how you can solve it out? There are still much more problems over there. Now, if we get back one to a trying to understand some of these predominant problems then I would definitely refer you to this particular paper, which we had published out in CVPR in 2017.

(Refer Slide Time: 22:48)

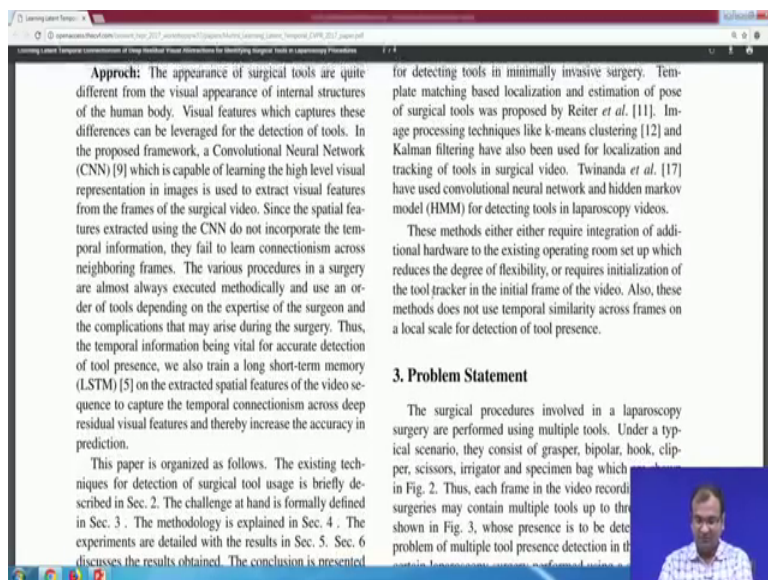


Now, here the overall idea was to get these kind of classifications done down for whatever tool is present over there.

(Refer Slide Time: 22:57)



(Refer Slide Time: 23:02)



Nonetheless, what we realized is that, training these networks over such, grossly newer kind of data is not so easy.

(Refer Slide Time: 23:07)

Now, whenever you see this codependency; say on these frames over here, there are few frames which have this kind of a tool grasper as well as clipper and bipolar. Ok. There are some of them which have a grasper as well as a specimen bank present down together. There are some which have a scissor and a grasper present down together.

Now, this does make it really challenging. And you would see that majority of your frames are just grasper, hook and clipper, and all the other ones other four categories over there are much lesser. Now, technically your network would tend to be heavily biased towards these particular classes and it would underperform on these classes. Now, for that you need to balance it out, and under balance it as it is something which would be looking down over here.

So, this is an extra investment which you have to do. So, that there is no neural network or a learning algorithm; in order to sample out from the data and actually create a balanced out dataset. Now, that is something you have to do, but the method is not so hard, though there is a lot of human intervention required over there.

So, what I would suggest is that, you can actually go down through this particular paper and on to understanding the design aspects, and what are the challenges which you would be facing down?

(Refer Slide Time: 24:55)

Baseline	Description	Train. time per epoch (min)	Test. time per frame (ms)
BL1	Modified (multi-label multi-class) AlexNet[6]	18.00	0.40
BL2	Modified (multi-label multi-class) AlexNet[6] (BL1) + LSTM	18.33	1.34
BL3	Modified (multi-label multi-class) GoogLeNet[14]	23.00	0.94
BL4	Modified (multi-label multi-class) GoogLeNet[14] (BL3) + LSTM	23.17	1.20
BL5	Modified (multi-label multi-class) ResNet-50[4]	35.00	1.95
Proposed Method	Modified (multi-label multi-class) ResNet-50[4] (BL5) + LSTM	35.30	2.42


Table 1. Baselines for performance comparison.

5.4. Baselines
To evaluate the performance of the proposed method, we have considered six baselines (BL) for comparison as summarized in Tab. 1.

5.5. Implementation
The proposed method was implemented and evaluated using Torch² and accelerated with CUDA 8.1³ and cuDNN 5.1⁴ on Ubuntu 14.04 LTS OS. The networks were trained on a system with 3×GTX TitanX GPU each with 12GB RAM, 2×Intel Xeon E5 2620 v3 processor and 176 GB of RAM. The codes used for implementing the framework is available at <https://github.com/kaustuv293/Tool-Detection>. The time taken for training and testing is also summarized in Tab. 1. We have trained the

CNN models (BL1, BL3 and BL5) for 2,000 epochs and the LSTM for adjucted models (BL2, BL4 and proposed framework) for 2,000 epochs.

5.6. Results
With baselines **BL2**, **BL4** and proposed framework, we experimented on the depth of the stacked LSTM network and the length of the sequence fed into the LSTM. We have evaluated the performance for the baselines with 2, 3 and 4 stacked LSTM networks and sequence lengths of 2, 3 and 50. The performance of **BL2** with the different network and sequence length is shown in Fig. 9(a). Performance of **BL4** is shown in Fig. 9(b). Performance of proposed framework is shown in Fig. 9(c). Performance comparison of **BL1**, **BL3** and **BL5** with the best

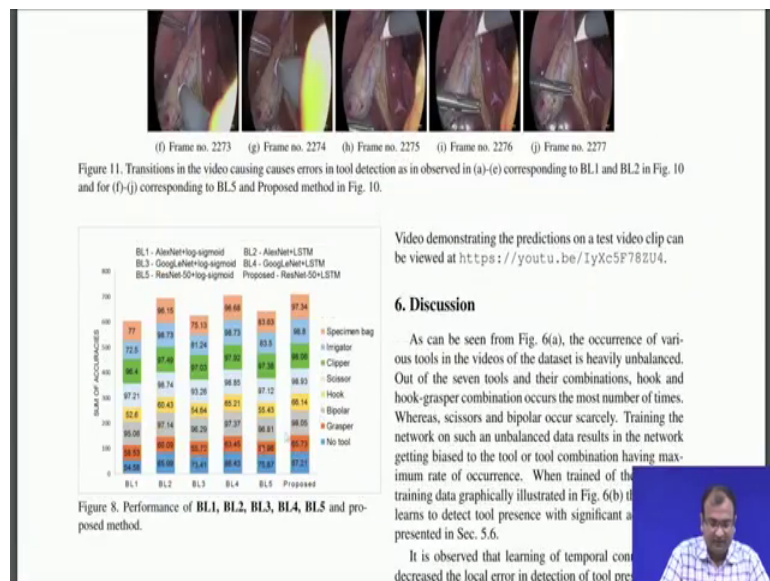


Now, the code is also available for this particular work, but it is it is not based on PyTorch it is on the earlier version of the library called as torch over there. Now, we had run down multiple experiments over there including different kind of network. So, we had tried down spatial feature extraction with Alexnet, Googlenet and Resnet.

And, then there was one where, you do not have any LSTM working. You just try to classify the tool in a frame based on what is present on that frame. So, it is a frame level classification. The other one is where you also have a temporal context modelling set down with using a LSTM for your recurrent modeling.

So, in together what comes out is an interesting aspect over here. So, these are four different baselines.

(Refer Slide Time: 25:43)



And, this is a full comparison over there. So, for each of these baselines is what is mentioned out over here, and this was the model which we finally, used for getting the maximum accuracy. And, this is where you have a Resnet 50 plus an LSTM coming down together in order to give you the highest accuracy per class and highest accuracy overall as well.

So, you can also have a quick look at the video over here which is on YouTube. And, these are few frames which were wrongly interpreted. And, and it was always found out that there is no surety on which frame will be wrongly depleted, it heavily depends on

what networks you are using and that is more of because every network tends to learn down a different kind of a feature and attribute over there.

So, having said that this is where we come to an end on understanding Temporal Deep Learning and Spatial Temporal Deep Learning on to a major context. So, in the next two lectures we will be doing a hands on with that one and, that would come to a closure of our course.

So, this is the last week of the course and, I wish you all the best for your exams as well, subsequently.

Thank you.