Deep Learning
Prof. Prabir Kumar Biswas
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture – 53
Semantic Segmentation - I

Hello welcome back to the NPTEL online certification course on Deep Learning. In today's lecture we are going to talk about the Semantic Segmentation problem and we have said earlier that for coming few lectures we will mostly concentrate on applications of the deep learning techniques that we have done so for.

In our previous class we have talked about the deconvolution and the upsampling operations or un-pooling operations. So, what we said is for this kind of applications a semantic segmentation, where the operation you have to do on every pixel level or for every pixel you have to take a decision that two which of the classes the pixel belongs to.

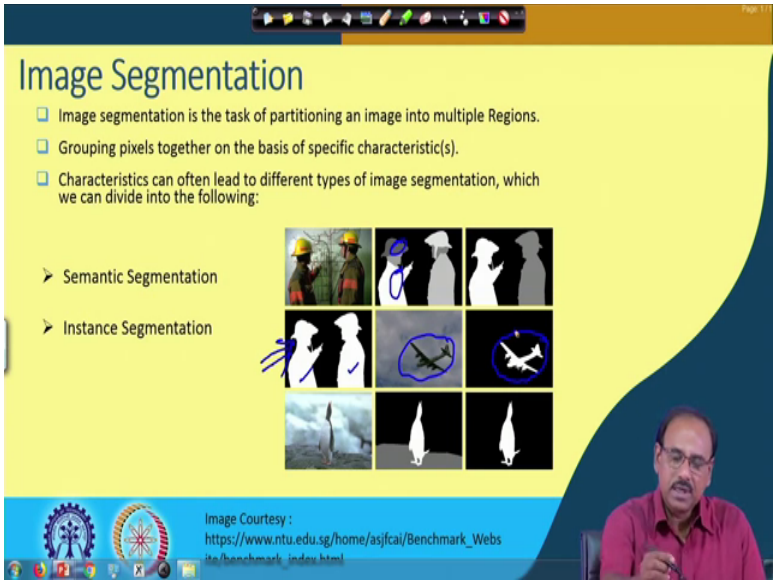
So, in order to have such a decision what is required is the abstraction of your input images which is given by the convolution layers, I have to blow it up to the size of your input image. And for that the kind of operations which are required is upsampling or un pooling and then we have to have the operation of deconvolution which is the reverse operation of convolution which is done in the forward direction.

(Refer Slide Time: 01:54)

So, in today's lecture we will talk about the semantic segmentation problem and we will try to discuss two particular approaches for semantic segmentation; one of them is called fully convolutional network based semantic segmentation and the other one is deconvolutional network based semantic segmentation.

So, before going to the problem of semantic segmentation or how to obtain semantic segmentation off and give an image using either fully convolutional network or using a deconvolutional network, let us see what is meant by image segmentation and what is semantic segmentation.

(Refer Slide Time: 02:41)



So, you see that image segmentation is basically a process of partitioning and input image into different regions, and this partitioning is done based on certain characteristics of the input pixels. So, partition the image in such a way that in every partition or every region within that image if I take 2 pixels then the characteristics of those 2 pixels will be very similar whereas, if I take 2 pixels from 2 different partitions or two different segments then the characteristics of those 2 pixels or the characteristics of those 2 pixels are likely to be very different.

So, based on similarity of the characteristics of the pixels and sometimes based on connectivity as well your partition the image into different regions and each of those regions is known as a segment. The purpose of the segmentation is that you can identify or you can group together the pixels belonging to a particular object or the pixels

belonging to a particular class, that class may be say buildings, the class may be persons present in the image, the class may be cars, the class may be cow, anything.

But what is expected is when I take 2 pixels belonging to a car then the characteristics of the region around that around those 2 pixels they will be very very similar. So, as shown in these figures you find that in the top left corner image you have two persons and the next two images in the same row as well as the image below it that identifies the regions or the pixels are the pixels belong to those persons. In particularly in this particular case you find that in this particular image you find that you do not have any differentiation between whether a pixel belongs to a hat of the person which the person is wearing or it belongs to the body of the person.

Similarly, there is no demarcation between this person and this person though there are two different persons. So, it says that these pixels belong to region how that region is occupied by persons whereas, if you look at this you find that this is the hat region and this is the body region where there is some demarcation though all these pixels belong to different persons, but within the persons there is further sub classification.

Similarly, if you come over here there is an aeroplane and here you find that it has grouped all the pixels which belongs to the aeroplane. So, depending upon the characteristics that we want to extract and the way we want to use those characteristics for segmentation purpose, we can have two different types of segmentation operations. One of the segmentation operation is what is known as semantic segmentation and the other kind of segmentation operation is what is known as instance segmentation.

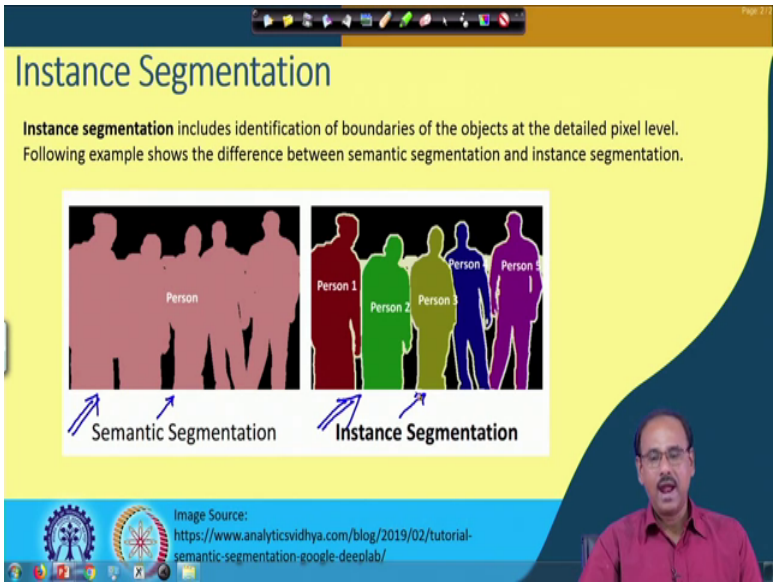
(Refer Slide Time: 06:10)

So, let us see that, what are these two types of segmentations so firstly, let us talk about what is semantic segmentation? Semantic segmentation is the process which links each pixel in an image to a class label. Say for example, we then an image I have five different cars right and the car is one of the classes which is present in the image similarly, in the same image I can have buildings and building is another class. In the image I can have different persons and the persons or the man or human being they become another class.

So, I can have an image where in the image I have three different classes or three different classes of objects, one class is car, one class is building, one class is say persons, another class maybe is a signboard road sign and many such things. So, in case of semantic segmentation it simply says or associates a pixel within the image to a particular class. So, it simply says whether a pixel belongs to car it does not identify which car it is or a pixel belong to a person or a pixel belong to a building or a pixel belong to a signboard and all that, but it does not differentiate.

If there are multiple instances of the cars semantic segmentation it does not tell you that to which of the cars particular pixel belong, or in other words what we can see is that all the pixels belonging to a particular class forms a semantic segment, ok. Or in case of semantic segmentation, we do not have any boundary of the different instances of the objects belonging to the same class.

(Refer Slide Time: 08:14)



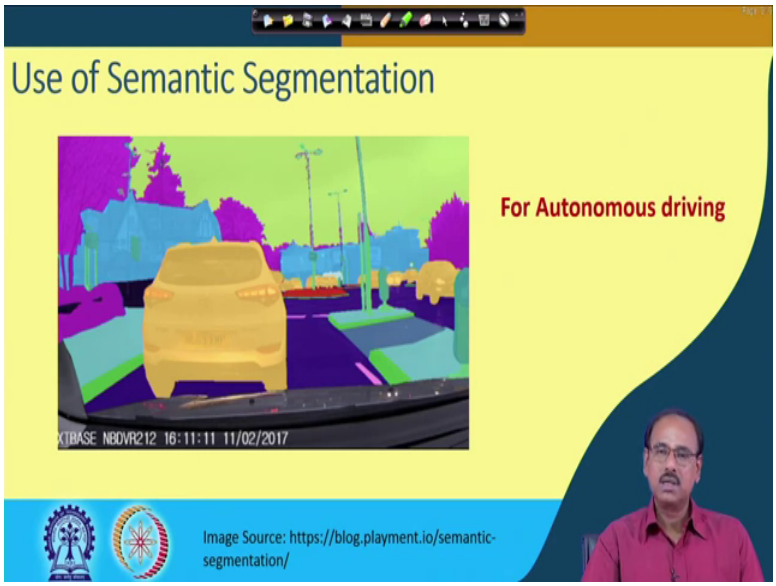
Whereas, if we talk about say instance segmentation say for example, over here you find that we have shown two different segmentation output of an image where within the image there are say five persons present within the image, right.

So, if you see the image which is the segmented output which is shown on the left so, this one. Here you find that all the pixels belonging to the persons have been classified into the same class that is a person class, it does not say that which pixel belong to which of the persons. Whereas, if you look at this particular segmented output here you find that in addition to classification it also tells you the boundary of different persons; that means, every instance of an object belonging to that class has been segmented out separately.

So, this first kind of segmentation the segments that you obtain this is what is known as semantic segmentation, which simply tells that the pixel belongs to which class and this kind of segmentation is known as what is instance segmentation, where it does not only tell you which class, but it also identifies that to which instance of that class our pixel belongs. So, this is the difference between your semantic segmentation and instance segmentation.

So, in our lecture today; in today's lecture maybe in the next lecture as well we will talk about the semantics segmentation problem and we will see that how this semantic segmentation problem can be solved using deep neural networks.

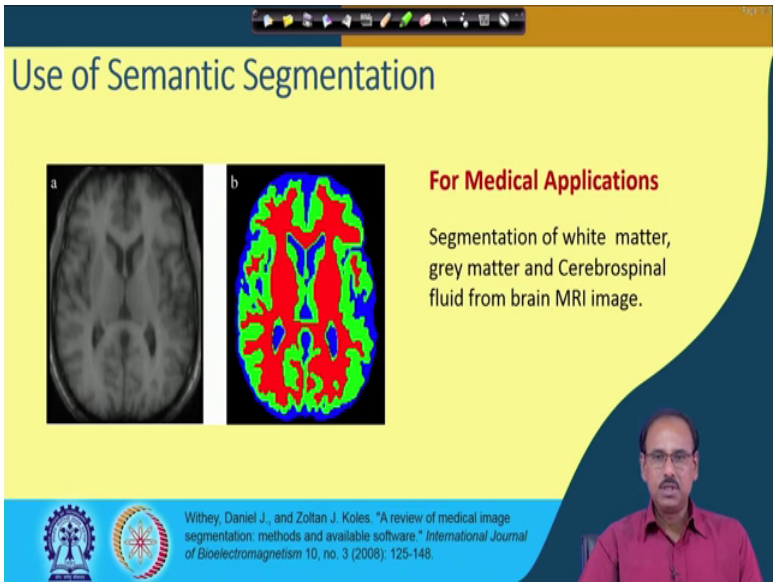
(Refer Slide Time: 10:14)



So, now let us see that what are the applications of such semantic segmentation? You take for example, navigation or driving and of an autonomous car. So, for an autonomous car when it is navigating in a particular region or it is navigating on a road, the car has to know it is surroundings. So, car has to know that which pixels belong to the road. So, that the car will move on the road only, it does not hit a building and in order to avoid hitting a building the car has to know that which are the pixels which corresponds to buildings.

If there is pedestrian moving around the car also has to know that which are the pixels which identifies the pedestrians which are moving around. So, that the car does not hit a pedestrian. So, this semantic segmentation of the scene is a very very important aspect of an autonomous car navigation problem.

(Refer Slide Time: 11:20)

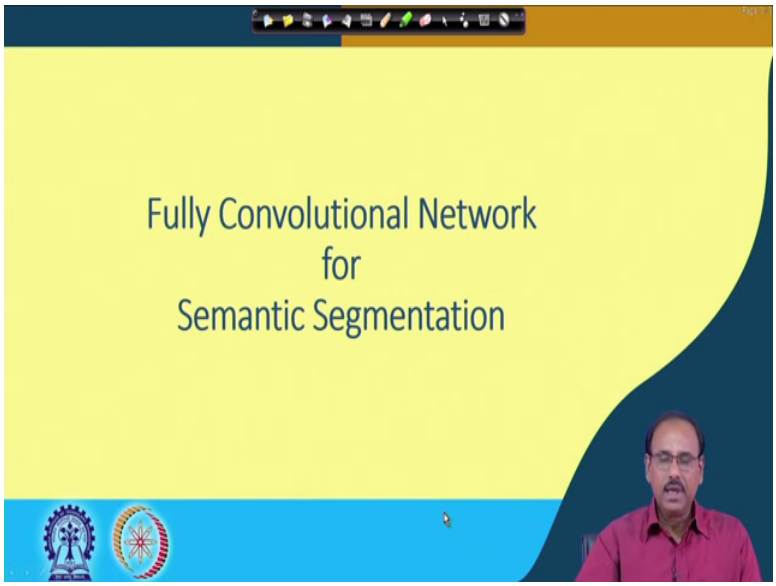


Similarly, the semantic segmentation can also be used in medical applications. So, this is this slideshows and slice of a brain MRI. So, a radiologist or a doctor may like to know that, what is the extent of white matter, what is the extent of gray matter or what is the extent of cerebrospinal fluid present in the brain.

And for this purpose this semantic segmentation which identifies the regions which belong to white matter or identifies the regions which belong to gray matter, similarly identifies the regions or the pixels which belong to cerebrospinal fluid, this information is very very important for diagnosis. So, the semantic segmentation plays a very very important role for such medical applications as well.

So, these are just two different applications that I have highlighted, but semantic segmentation has many many more applications right in wherever we think of application of computer vision. For example, if I want to have an automated assembly shop or say robot navigation, robotic vision in all cases the robot may have to identify a particular object or a set of objects which are present in the shop floor. So, semantic segmentation for identifying objects this plays a very very important role in machine vision applications in industries.

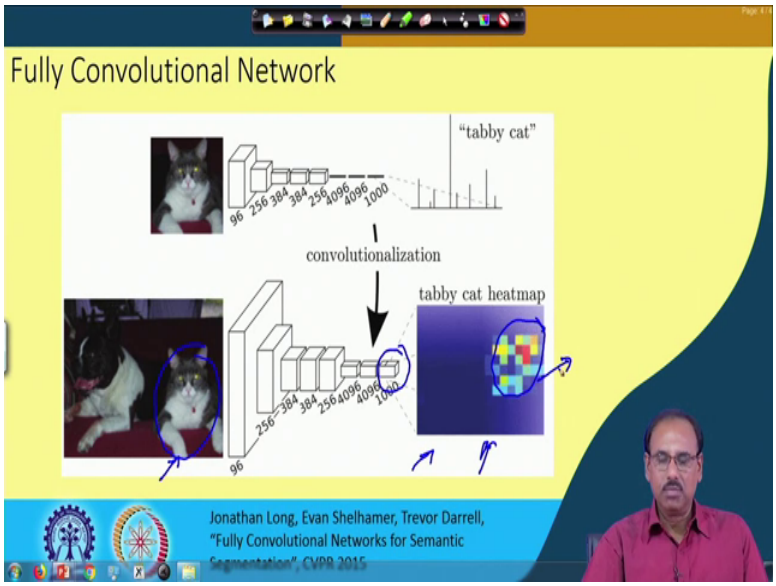
(Refer Slide Time: 13:10)



So, given these two applications now let us come to an approach of fully convolutional network or fully convolutional neural network used for semantic segmentation purpose. So, the work that I am going to present here, it was published in was presented in CVPR 2015 by persons Jonathan Long and colleagues.

So, earlier we have seen the convolutional neural network or deep convolutional neural network, and we have seen that whenever you have a deep convolutional neural network; the deep convolutional neural networks are basically used for classification or recognition purpose.

(Refer Slide Time: 13:25)



So, there we have a convolutional layer, then you have max pool layer; you have convolution layer, max pool layer appearing one after another, where the convolutional layer actually extracts the features from the input images or it extracts some abstract information from the input images and layer by layer the convolution networks actually gives you the information at different layers of abstraction; different labels of abstraction of the input signal that you have provided that we want to classify.

The purpose of max pool layers we said that the max pool layer tries to find out that which particular feature has got the prominent activation or prominent response the maximum response to a given filter in a given region which is known as receptive field. And based on this what the max pool layer does is it takes a max pool window and from that window it simply extracts the maximum activation and that maximum activation is passed to the next convolutional layer for further processing.

So, in the process the max pool layer identifies the maximum activation and also it reduces the dimensionality of the feature maps that helps in processing because your computation goes on reducing as the dimensionality of the feature maps reduces. So, as a result as you move along the convolutional layer from say shallow layers or the input layer to the deeper layers or the output layers, your abstraction information abstraction level goes on increasing or in other words we can say that every feature gets a global information within the receptive field of that particular node.

But in the process what you lose is the locational information, because if I take a particular feature in a feature map in a deeper layer the receptive field of that particular node when I come to the input layer is very large. So, you get the global information of that large receptive field, but at the same time in which area within this receptive field, your filters give maximum response that is lost.

So, what you get is you get the informations at different levels of abstraction through these convolution operations and the max pool operations. And we have seen earlier that the outer layer or the final layer or few layers at the final levels they are known as fully connected layers and this fully connected layers are just similar to the multi layer perceptron which are used for classification or recognition purpose. And, these fully connected layer or the output of the fully connected layer actually tells you that to which of the category or to which of the class your input signal is classified.

So, such a network is shown in this part of the figure. So, this is what is that convolutional network followed by here what we have is a fully connected network. So, this convolution networks actually gives you the feature map and finally, this feature map passes through the fully connected layer and the output of the fully connected layer is a decision about to which of the category or to which of the classes your input image belongs. So, here you find that this particular output which shows that this input image actually belongs to a category of a cat.

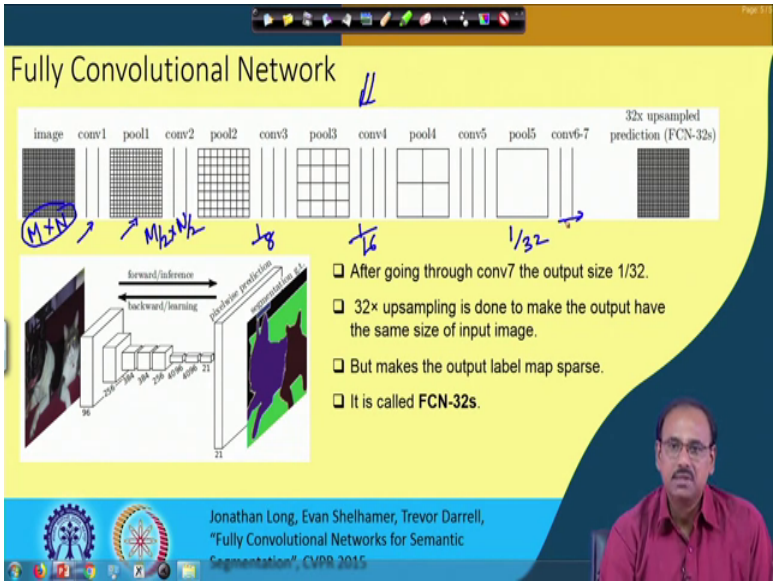
Now, in fully convolutional layer which these Jonathan Long and others have reported that they have said that instead of considering the output layer as a fully connected layer, what happens if we consider this to be a convolutional layer as well. So, there is nothing wrong I can consider this output layer which is otherwise a fully connected layer to be a convolutional layer and if I consider this to be a convolutional layer in that case you find that every node, in this final convolutional layer has a receptive field which is the entire feature map being fed to this particular layer, ok.

So, and if I consider this to be a fully connected layer then I can simply blow up in the outputs of this fully connected layer in the form of a feature map and that feature map looks like this. So, here in the fully connected layer if there are say 1000 nodes I can blow it up in the form of an image where image will be of size the M by N depending upon the number of nodes that I have in the fully connected layer.

And if you look at this that this feature map that you get that gives you some short of heat map of the cat or this region in this particular feature map, this shows that in this region I have this particular object which is the cat. But, the problem here is as the information has flown through a number of convolution layers and the number of max pool layers. So, the size of the feature map that you get over here is much less than the size of the input image which is actually fed.

So, in order to have the pixel wise decisions that is to take a decision at every pixel whether the pixel belongs to the cat or the pixel belongs to something else, I need to blow up or expand this image to a size which is of same size as that of the input image. So, that way it has been done or reported in this work is something like this.

(Refer Slide Time: 20:39)



So, if you look at the inter flow of the convolutional network, in this convolutional network you find that there are seven different convolution layers. So, these layers are as shown over here. So, I have an input image suppose the input image is of size M by N then you have the first convolution layer followed by a pooling layer.

So, this pooling layer if the pooling window is 2 by 2 and with size 2 then after pooling the size of the image which will be fed to the second convolution layer will be M by 2 by N by 2. So, you find that the image size simply becomes half, similarly the feature map, size of the feature map which is fed to this convolution 3 is M by 8 by N by 8 or it is

reduced by a factor of 8, similarly after pool 3 it is reduced by a factor of 16, after pool 5 it is reduced by a factor of 32.

So, if my original input image was of size M by N, the size of the feature map that you get after this convolution layer 7 is reduced by a factor 32. So, because of this if I want to get a pixel level decision; that means, for empty pixel I have to decide whether the pixel belongs to cat or the pixel belongs to something else what I need to do is, I have to expand or blow up this feature map by a factor of 32.

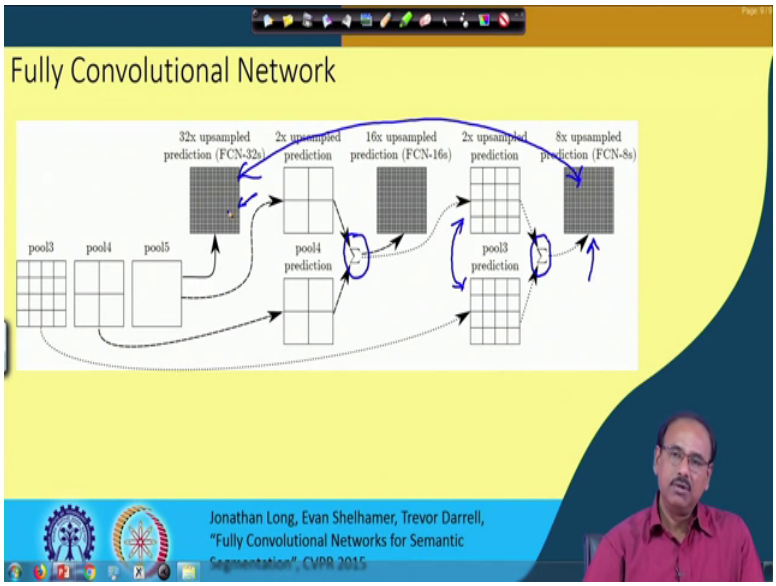
So, for that what I have to do is upsampling by a factor of 32 and this upsampling concept we have discussed in our previous class that what I can do is, I can go for deconvolution with a deconvolution filter with a larger stride greater than 1. So, if you perform a deconvolution operation with stride greater than 1, your output map that you get is larger than the size of the input map.

The other operation that we have discussed is that you perform the deconvolution which we said it is sub-pixel convolution. So, in sub pixel convolution what you have done is that you have expanded or upsampled the feature map and all the intermediate regions you filled up with 0. So, it is what is known as 0 upsampling and after upsampling the feature map what you have done is, you have performed the deconvolution using the deconvolution kernel. And, we have seen that this deconvolution kernel has certain parameters and these parameters are actually learnable parameters.

So, you learn these parameters of the network learn these parameters during the training operation following the back propagation algorithm or gradient descent algorithm. So, we will talk about how to train such networks maybe sometimes later. So, here what is shown is that what I want to do is. So, at this particular point whatever is the size of the image or the size of the feature map I know that size of this feature map is 1 by 32 of the size of the original image.

So, what I have to do is, I have to blow up this size to 32 times. So, that our output image matches with the input image and then every pixel within the some output image has to be classified to one of the classes which are actually present within this image. So, for this classification purpose, we have to plane this particular network and as I said that I will come back to the training of this network a bit later. So, when you go for this upsampling by a factor of 32, this can be done in various ways.

(Refer Slide Time: 25:03)



We can perform the upsampling in a single short operation that is upsampling by a factor of 2, and then after upsampling you go for deconvolution operation using the deconvolution filter or deconvolution kernel. But, the problem in that case is that if we are going for upsampling in a single step then you are upsampled image or the map that you get will be very various parts and the segmentation output that you get may not be very proper.

So, the other way in which the same can be done is that you go for multiple steps and here you use the concept of skip connection. So, what you do is, you take the output from pools of 5 in the convolutional part and you upsample it by a factor 2. Similarly take the output from pool 4 in the convolutional network after max pooling operation and then you know upsample by a factor of 4.

So, you find that output of pool 5 upsampling by a factor 2 and output of pool 4 upsampling by a factor 4, here the size of the 2 maps that you generate the match. And then what you do is this upsample for chain and this upsample version you simply add them together. And after adding them you again upsample it so, this output is now 16 times upsampled from here as 16 times upsampled from the output of your final convolution layer.

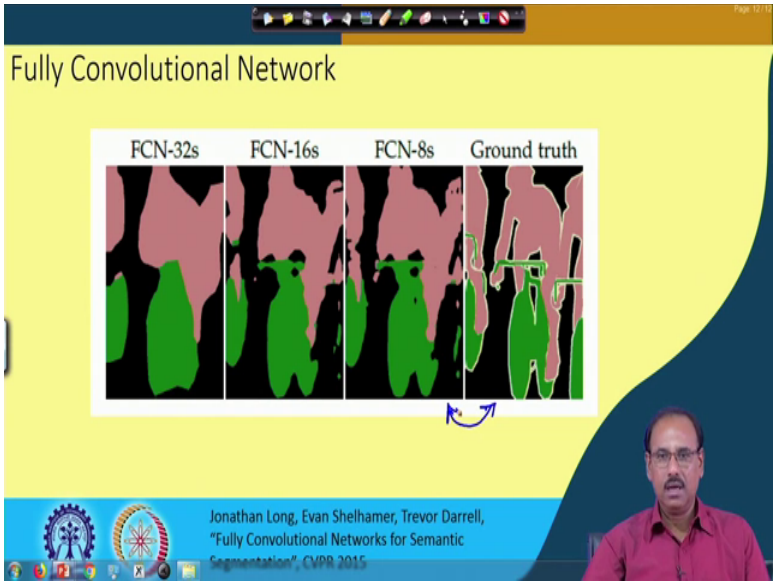
Next operation what you can do is you can take the output from pool 3 upsample it by a factor of 16 and this was already upsampled by 16 so, you upsampled it by a factor of 2

just a minute what we had done before is if you come to this after convolution 3, it was down sampled by a factor of 8; here it was down sampled by a factor 8 after pooling layer 3.

So, from the output of the pooling layer 3, we have to upsample it by a factor of 8 and this one the output that you get from here by combining these two you upsample it by a factor of 2. So, both of them taken together are now upsamples by a factor of 32 upsampled by a factor of 8. So, you add them together. So, here you get what is the upsampled version by a factor of 8 and then you can combine these two together.

So, whatever the sparse representation that you have got here by upsampling by a factor of 32 in a single step, now here what we are doing is hierarchically you are combining with that the upsampled versions which are at finer scales. So, as a result the output that you get that becomes much better than the output that you get using a single step upsampling.

(Refer Slide Time: 28:55)



So, now let us just see that, what are the kind of outputs that you can have. So, here is the output, it shows that if this is the ground truth of your semantic segmentation. Once you directly upsampled by factor 32 this is the output, this is what is FCN 16, now FCN 16 is output over here this is what is your FCN 16 and this is what is your FCN-8. So, you find that in these two cases your output of FCN 16 and output of FCN 8 is much better than output of FCN 32. So, this is much closer to your ground truth, this is over here.

So, this is an approach where a fully connected neural network can be conceived as a fully convolutional neural network and this fully convolutional neural network is used for the semantic segmentation purpose. So, we will stop here today, in our next class we will talk about a deconvolutional neural network that can be used for semantic segmentation.

Thank you.